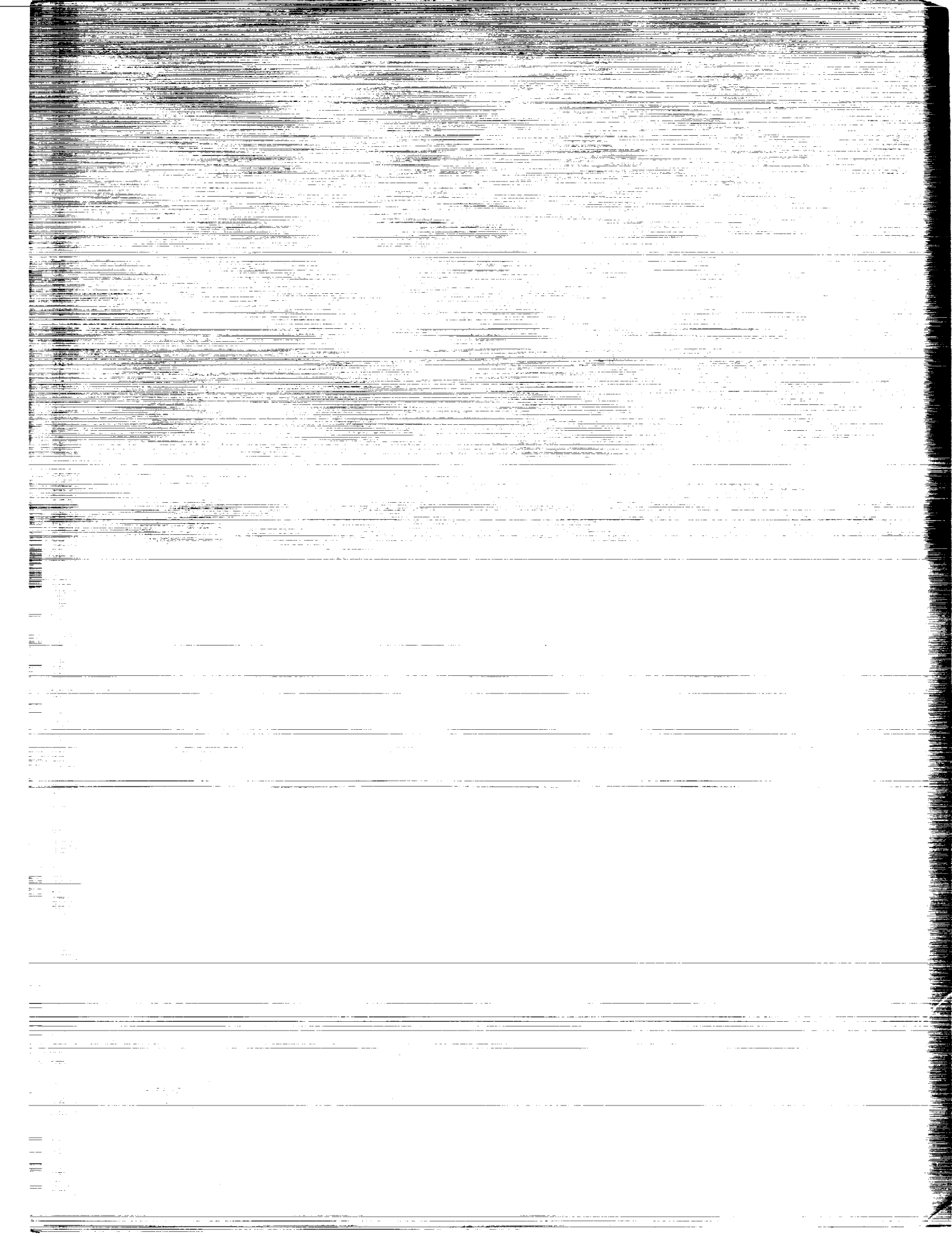
NASA Conference Publication 3262

# Third NASA Goddard Conference on Mass Storage Systems and Technologies



NASA Conference Publication 3262

# Third NASA Goddard Conference on Mass Storage Systems and Technologies

Edited by
Benjamin Kobler
*Goddard Space Flight Center*
*Greenbelt, Maryland*

P.C. Hariharan
*Systems Engineering and Security, Inc.*
*Lanham, Maryland*

*Proceedings of a conferance held at*
*the University of Maryland*
*University College Conference Center*
*College Park, Maryland*
*October 19 - 21, 1993*


National Aeronautics and
Space Administration

**Scientific and Technical**
**Information Branch**

**1994**

# Third NASA Goddard Conference on Mass Storage Systems and Technologies

## *Program Committee*

Ben Kobler, *NASA Goddard Space Flight Center (Chair)*
Jean-Jacques Bedet, Hughes STX Corporation
John Berbert, *NASA Goddard Space Flight Center*
Jimmy Berry, *Department of Defense*
William A. Callicott, *NOAA/NESDIS*
Sam Coleman, *Lawrence Livermore National Laboratories*
P C Hariharan, *Systems Engineering and Security, Inc.*
Terrence Pratt, *USRA/CESDIS*
Sanjay Ranade, *Infotech SA, Inc.*
Don Sawyer, *NASA Goddard Space Flight Center*
Elizabeth Williams, *Supercomputing Research Center*

## *Production, Copy Editing, and Layout*

Len Blasso
Ann M. Lipscomb

# Preface

The Proceedings of the Third Goddard Conference on Mass Storage Systems and Technologies include all the papers presented at the Conference which were made available for publication, as well as transcripts of the panel discussions and the after-dinner speech. We are grateful to the authors, and the panel members, for their time and effort.

Dr Hans Mark, currently Professor of Aerospace Science at the University of Texas in Austin TX, delivered a thoughtful keynote speech in which he pointed out that, in the last part of the last century, and the early half of this century, the focus was on solving the *infinitely small* (atomic, nuclear) and *infinitely large* (cosmological) problems which lent themselves to study without the aid of computers. The *infinitely complex* problems which are being studied now require the use of computers.

Dr John Simonds, Director of the National Storage Industry Consortium (NSIC), talked about the *Future of the US Digital Recording Industry*. By the end of the century, very much higher recording densities in both optical and magnetic recording media will be available, and the efforts of NSIC, whose members include industry, research institutions and universities, are directed towards attaining these goals more rapidly. Dr Simonds also announced the formation of a new Division, the National Storage System Foundation (NSSF), which will work to ensure that the hardware and media development efforts of NSIC will be complemented by work in the software and systems integration arenas to ensure that end users are able to realize the full potential of the storage devices being developed.

The next session focused on media. Steve Jewell, of Advanced Measurement Systems, spoke about risk factors to storage of magnetic data. Quarter Inch Cartridge (QIC) tapes, which are beginning to outsell most other formats, and their growth path, were described by Ted Schwarz of 3M. Ted Larsen, of Southwall Technologies, dealt with the write-once optical media developed by Dow, and Jeff Howell, of ICI Imagedata, explained the certification procedures used with the ICI optical tape which is currently the recording medium used with the CREO 1003 Optical tape recorder.

Standards are important for any large, long-term archive, and there were four papers devoted to this important issue. Bob Coyne, of IBM Federal Data Systems, and chairman of the IEEE Storage Systems Standards Working Group, explained the status of the IEEE Mass Storage Systems Reference Model (RM). Since its inception in 1982, the RM has undergone significant modifications, and is now in Version 5. A guide and standards are still being worked on. Don Sawyer, of NASA's NOST and NSSDC, explained the roles of standards in the various phases of the life of data. Lou Reich, of Computer Sciences Corporation, delivered a talk on *A Reference Model for Scientific Information Exchange*, and Fynnette Eaton, of the National Archives and Records Administration. addressed the problems peculiar to preserving electronic information.

The plenary session of the first day was terminated by a panel discussion on *User Experiences with Storage and Distribution Media*. This was moderated by Mr Jimmy Berry of the National Security Agency and the panel consisted of representatives from the Goddard V0 DAAC, NOAA's National Climatic Data Center in Asheville NC, NARA, NSSDC and the US Geological Survey's Eros Data Center.

Nine poster papers were on display that evening during the Conference reception. There were a number of exhibits and demonstrations by vendors during the first two days.

The second day of the conference opened with a session on system performance. Henry Newman, of Instrumental Inc., set the stage with a talk entitled *Emerging Standards for Collection of Performance Data*. Performance measurements made on automated cartridge libraries (STK 4400, Wolfcreek) from Storage Tek were presented by Gary Hull of Hughes STX. David Therrien of Epoch Systems reported on the use of magnetic tape technology for data

migration, and Jean-Jacques Bedet of Hughes STX presented a simulation study of the Goddard V0 DAAC. Data storage system concepts were covered in three talks. Large data systems at the National Security Agency in Fort Meade MD, and at the Mobil company facilities in Dallas TX were the subject of talks by Mike Shields and Mike Daily respectively. Dr David DeWitt, of the University of Wisconsin in Madison WI, gave a provocatively titled invited talk on parallel object-oriented database management systems and why they would doom standards like NetCDF! The Distributed Mass Storage System at NASA Langley was described by Juliet Pao. The day closed out with a panel discussion, chaired by Dr Sanjay Ranade of Infotech SA, on *User experiences with Unix-based Hierarchical File Storage Management Systems.* The participants included Mobil (E-Mass), the Department of Defense (Amass), Sandia National Laboratories (Unitree), Supercomputing Research Center (Epoch), NASA/GSFC (Unitree) and NASA Ames.

The Conference Banquet was held in the evening on October 20 and concluded with a talk by Dr David Parker of the Library of Congress on the archiving of movies.

Communications technologies will play a key role in enabling wider, easier access to data libraries, and in distribution and ingestion of data. Dr Nim Cheung, of Bellcore, described ATM (Asynchronous Transfer Mode) and SONET (Synchronous Optical Network) in his invited talk which was followed by one from Dr Yelena Yesha of the University of Maryland, Baltimore County, on Digital Libraries. There were five talks on Data Distribution Systems; it is obvious that, even in small systems, the problem of integrating technology and software from multiple vendors can lead to unanticipated problems and delays. Dr Jaideep Srivastava, of the University of Minnesota, presented a talk on *A Parallel Data management System for Large-scale NASA Datasets.* The final presentation at the conference was one on the *Importance of Robust Error Control in Data Compression* by Sandra Woolley of the University of Manchester in England. Data compression is the removal of redundancy in data; error detection and correction (EDAC), on the other hand, involves the addition of redundancy to combat errors caused by noisy channels or corrupted media. In the case of compressed data, extremely robust EDAC is required since small errors can propagate their effect far into a data set.

We are grateful to:

Jean-Jacques Bedet, Hughes STX Corporation
John H Berbert, NASA/GSFC
Jimmy F Berry, National Security Agency
William Callicott, NOAA/NESDIS
Terry Pratt, CESDIS/GSFC
Sanjay Ranade, Infotech SA
Don Sawyer, NASA/GSFC
Elizabeth Williams, Supercomputing Research Center

all members of the Conference Program Committee whose energetic efforts made this conference possible; the speakers, panel discussion members, and the attendees for their positive contributions which are collected together here; Jorge Scientific Corporation for the Conference arrangements; the staff at the University of Maryland Conference Center for the excellent facilities they provided; Len Blasso and Ann Lipscomb for help with the layout and publication of these Proceedings.

P C Hariharan
Systems Engineering and Security, Inc

Ben Kobler
NASA/GSFC

**Table of Contents**


Preface ..... iii

Table of Contents ..... v

Some Visions of Scientific Future, *Hans Mark, The University of Texas at Austin.* ..... 1

The United States Digital Recording Industry, *John L. Simonds, National Storage Industry Consortium* ..... 7

Toward a Digital Library Strategy for a National Information Infrastructure, *Robert A. Coyne and Harry Hulen, IBM Federal Systems Company* ..... 15

Magnetic Field Sources and Their Threat to Magnetic Media, *Steve Jewell, Advanced Measurement Systems, Inc.* ..... 19

High Performance Quarter-Inch Cartridge Tape Systems, *Ted Schwarz, 3M Company* ..... 31

A New Tape Product for Optical Data Storage, *T. L. Larsen, F. E. Woodard, and S. J. Pace, Southwall Technologies, Inc.* ..... 45

Certification of *ICI 1012* Optical Data Storage Tape, *J. M. Howell, ICI Imagedata* ..... 51

The IEEE Mass Storage System Reference Model: Update on Version 5, *Robert A. Coyne, IBM Federal Systems Company* ..... 57

Role of Formats in the Life Cycle of Data, *Don Sawyer, NASA/Goddard Space Flight Center* ..... 65

A Reference Model for Scientific Information Interchange, *Lou Reich, Computer Sciences Corporation, Don Sawyer, NASA/Goddard Space Flight Center, and Randy Davis, University of Colorado-LASP* ..... 75

Preserving Electronic Records: Not the Easiest Task, *Fynette Eaton, National Archives and Records Administration* ..... 99

Invited Panel: User Experience with Storage and Distribution Media, *Moderator: Jimmy Berry, DoD* ..... 103

NCDC Mass Storage Systems and Technologies, *Dick Davis, NOAA/National Climatic Data Center* ..... 121

A Petabyte Size Electronic Library Using the N-Gram Memory Engine, *Joseph M. Bugajski, Triada, Ltd.* ..... 125

Volume Serving and Media Management in a Networked, Distributed Client/Server Environment, *Ralph H. Herring and Linda L. Tefend, EMASS® Storage Systems* ..... 139

Improvement in HPC Performance Through HIPPI RAID Storage, *Blake Homan, Maximum Strategy, Inc.* ..... 149

Architectural Constructs of AMPEX DST, *Clay Johnson, Ampex Systems Corporation* ..... 153

Virtual File System For PSDS, *Tyson D. Runnels, The Boeing Company* ..... 163

Volume Server - A Scalable High-Speed and High-Capacity Magnetic Tape Archive Architecture with Concurrent Multi-Host Access, *Fred Rybczynski, Metrum, Inc.* ..... 169

The Growth of the UniTree Mass Storage System at the NASA Center for Computational Sciences, *Adina Tarshish, Goddard Space Flight Center, Ellen Salmon, Hughes STX Corporation* ..... 179

Hierarchical Storage Management System Evaluation, *Thomas S. Woodrow, NASA /Ames Research Center* ..... 187

Introduction of the UNIX International Performance Management Work Group, *Henry Newman, Instrumental, Inc.* ..... 217

Performance Measurements and Operational Characteristics of the Storage Tek ACS 4400 Tape Library with the Cray Y-MP EL, *Gary Hull, Hughes STX Corporation; Sanjay Ranade, Infotech SA Inc.* ..... 229

Using Magnetic Tape Technology for Data Migration, *David Therrien, Yim Ling Cheung, Epoch Systems* ..... 241

Simulation of a Data Archival and Distribution System at GSFC, *Jean-Jacques Bedet, Lee Bodden, Al Dwyer, and P C Hariharan, Hughes STX Corporation; John Berbert, Ben Kobler, and Phil Pease, NASA/Goddard Space Flight Center* ..... 257

Mass Storage - The Key to Success in High Performance Computing, *Richard R. Lee, Data Storage Technologies, Inc.* ..... 279

Storage System Architectures and Their Characteristics, *Bryan M. Sarandrea, Advanced Archival Products, Inc.* ..... 285

Storage Media Pipelining: Making Good Use of Fine-Grained Media, *Rodney Van Meter, ASACA Corporation* ..... 303

The Trend to Parallel, Object-Oriented DBMS, *David J. DeWitt, University of Wisconsin* ..... 313

Mass Storage At NSA, *Michael F. Shields, Department of Defense* ..... 325

ACE: A Distributed System To Manage Large Data Archives, *Mike I. Daily, Mobil Oil Corporation; and Frank W. Allen, E-Systems, Inc.* ..... 331

NASA Langley Research Center's Distributed Mass Storage System, *Juliet Z. Pao and D. Creig Humes, NASA/Langley Research Center* ..... 337

Invited Panel: User Experiences with Unix-Based Hierarchical File Storage Management Systems, *Moderator: Sanjay Ranade, Infotech S.A., Inc.* ..... 347

Evening Reception and Dinner Speech: Moving Images Archive, *David Parker, Library of Congress* ..... 369

ATM Technology and Beyond, *Nim K. Cheung, Bellcore* ..... 381

Issues for Bringing Digital Libraries into Public Use, *David W. Flater and Yelena Yesha, University of Maryland Baltimore County* ..... 389

A Data Distribution Strategy for the 90s (Files Are Not Enough), *Mike Tankenson and Steven Wright, Jet Propulsion Laboratory, Telos Systems Group* ..... 393

Management of the National Satellite Land Remote Sensing Data Archive, *(Text Not Made Available), Darla J. Werner, Hughes STX Corporation* ..... 405

Alaska SAR Facility Mass Storage, Current System, *David Cuddy, Eugene Chu, and Tom Bicknell, Jet Propulsion Laboratory, California Institute of Technology* ..... 407

Value Added Data Archiving, *Peter R. Berard, Battelle Pacific Northwest Laboratory* ..... 417

A Practical Large Scale/High-Speed Data Distribution System Using 8-mm Libraries, *Kevin Howard, EXABYTE (Presented by Kelly Sharfe)* ..... 427

Distributed Active Archive Center, *Lee Bodden, Jean-Jacques Bedet, and Wayne Rosen, Hughes STX; Phil Pease, NASA/Goddard Space Flight Center* ..... 447

User Interface Development and Metadata Considerations for the Atmospheric Radiation Measurement (ARM) Archive, *P. T. Singley, J. D. Bell, P. F. Daugherty, C. A. Hubbs, and J. G. Tuggle, Environmental Sciences Division , Oak Ridge National Laboratory* ..... 459

A Parallel Data Management System for Large-Scale NASA Datasets, *Jaideep Srivastava, University of Minnesota* ..... 469

The Importance of Robust Error Control in Data Compression Applications, *Sandra I. Woolley, University of Manchester* ..... 487

# Some Visions For Scientific Future

**Hans Mark**
The University of Texas at Austin
Department of Aerospace Engineering and Engineering Mechanics
401 Woolrich Hall
Austin, TX 78712-1085
Phone: (512) 471-5077
FAX: (512) 471-4070

## (1) Introduction

This conference brings together people whose work is at the heart of what Vice President Gore has called the "Information Super Highway". There has been much discussion about the value of this new communications concept but there is no doubt in my mind that many important new developments will result from its implementation. However, I want to narrow the discussion somewhat and ask how the technology that you are developing will be useful in extending the frontiers of human knowledge. How will computers and their associated devices be helpful in scientific research? What are the frontiers that we will be exploring? These are the questions I would like to examine today.

I do not have to tell this audience that computers are more than adding machines. The truth is that enough computer power can by itself be something that qualitatively enhances our ability to develop physical theories and to gain insights that could not have been gained otherwise. What I want to do is to speculate about some of the things that are still hidden behind the curtain of ignorance and to see where massive computing and data handling systems can be the key to finding answers. This is to me, perhaps the most exciting prospect before us today.

## (2) The Infinitely Large, the Infinitely Small, and the Infinitely Complex.

Some years ago Professor Victor F. Weisskopf of MIT presented a lecture on the subject that he called the "Infinitely Large, the Infinitely Small, and the Infinitely Complex". What he meant by this title was that during the first half of this century great progress was made in understanding the "small" on the one hand, and the "large" on the other. What he meant by "small", was the understanding of atomic and nuclear structure. The key to this understanding was of course, the development of quantum mechanics in the early years of this century. Once the subtleties of this way of looking at nature were understood, the mysteries of atomic and nuclear structure were resolved. In fact, today, seventy years after Werner Heisenberg established the uncertainty principle as the central theorem of quantum mechanics, everything we know still leads to the conclusion that quantum mechanics is applicable to all known phenomena, including the most esoteric high energy events. There is no question that this is a very real triumph of the human intellect. For all practical purposes, we understand the structure of matter and the formalism of quantum mechanics gives us the means to make predictions that are both scientifically valuable and of enormous practical utility.

By the "large", Weisskopf meant cosmology. In this field also, the early years of this century saw enormous progress. In 1917, Albert Einstein for the first time, wrote down his equations of general relativity that predicted a dynamic universe. It is ironic that Einstein himself did not believe his result and introduced his famous "fudge factor", the "Cosmological Constant" that made the universe static. It was only after Edwin Hubble and his collaborators discovered that the universe did in fact expand that Einstein's theory was triumphantly confirmed.

What is perhaps most fascinating is that in recent years Weisskopf's "small" and "large" have been coming together. The "big science" investments in high energy particle accelerators and space- based astronomy have revealed to us that the higher the energy, the more things seem to look alike. We now have evidence that the four forces of nature, the strong nuclear force, the electromagnetic force, the weak nuclear force, and gravity, become more alike as the energy of the interacting system is raised. My friend and colleague, Professor Steven Weinberg of The University of Texas at Austin and his colleagues, Sheldon Glashow and Abdus Salam, succeeded in showing that, at high energies the electromagnetic forces and the weak nuclear forces become identical. Strong nuclear forces and gravity exhibit some of the same trend. This is, of course, why building the new Superconducting Super Collider Particle Accelerator is so important. At the same time, we are learning more about the first few instants of the universe after the "big bang". It is at this time that the equilibrium energy that is the temperature, was high enough so that all forces probably looked alike, and that as the universe evolved, there was the differentiation toward the "infinitely complex" world as it exists today. This differentiation was of course Weisskopf's last theme to which I want to return in a few minutes.

All of the things that I have just mentioned depend on theoretical concepts developed in the early years of this century. Computing machinery was and is most important in exploring the "small" and the "large". But I can safely say that computers did not lead to the new fundamental insights in either one of these areas that dominated science in the first half of this century. It is in the understanding of the "infinitely complex" that computing machines have and will come into their own. It is in exploring these areas that new insights can be obtained only through the use of high speed computers and I would submit that this is what is really new in modern fundamental scientific research.

What I would like to do now is to discuss two examples in which computing machinery and data storage systems on a massive scale are the essential equipment necessary to make progress on an intellectual level.

**(3) Chaos Out of Order: The Understanding of Nonlinear Systems**

It is a remarkable fact that almost everything we think we understand about the universe is derived from solutions of linear systems of differential equations, or of differential equations that happen to be integerable. Isaac Newton was lucky. When he formulated the laws of motion and the law of gravitation more than three hundred years ago, he could apply these with great precision to the motions of the planets in the solar system. The precision was so astounding that people began to believe the universe operated like a clockwork. As great a mathematician as Pierre Simon de Laplace asserted, that if the conditions of the past were precisely established, then the future evolution of the universe could be completely predicted. As I have said, Isaac Newton was lucky. Laplace's statement depended on the fact that the solar system is, to a good approximation, a two-body problem. Thus the equations of motion as Newton formulated them were approximately integerable. It was this circumstance that led Laplace to make his statement. However, the key word is "approximately". Once people started to look at the details things turned out to be much more difficult than expected.

One hundred and seventy years after Newton's epochal work, James Clerk Maxwell had a similar stroke of luck. Maxwell was able to "unify" the electric and magnetic fields with a new system of equations that automatically explained the existence of electro-magnetic radiation. This system of equations turned out to be approximately linear and could therefore be dealt with using the principle of superposition. A vast and rich set of solutions followed that, once again, had enormous applications, both in generating new knowledge and in developing practical applications. Maxwell's work also seemed to point in the direction of the existence of a "clockwork" universe.

By the end of the 19th century, there were small hints on the horizon that the optimism generated by Laplace and by Maxwell was not really justified. The two people

most responsible for pointing this out were Henri Poincaré and Ludwig Boltzmann. Poincaré busied himself with the problem that Laplace thought he had solved. Was the solar system really a "clockwork"? In short, was it stable over long periods of time. Although Laplace thought he had proved this point, the fact is that he did not and in 1885 Poincaré showed that the solar system was unstable, or that at least stability could not be demonstrated. The equations that Newton had established were not integerable when all of the subtle forces acting on the bodies in the solar system were included. Even more important, when methods of perturbation theory were applied it was not possible to obtain a rigorous proof that the solar system was stable. Thus, from the viewpoint of rigorous analysis, the notion of the "clockwork universe" was no longer tenable. This statement is true even though it is possible to write down "exact" equations of motion for the solar system.

Another most important event that occurred at about the same time was Ludwig Boltzmann's monumental work in founding what we today call statistical mechanics. Boltzmann set himself the problem of explaining the macroscopic laws of thermodynamics on the basis of what was becoming to be the accepted microscopic or atomic view of the world. The first law was easy because it is simply a restatement of the conservation of energy. The difficulty lay with developing a way to deal with the concept of entropy on the microscopic level. Explaining the second law of thermodynamics was the critical problem. Boltzmann succeeded by developing the so-called "H Theorem" in which the thermodynamic entropy is related to the probability of the occurrence of a given "state" of the microscopic atomic system. What he did was to use a simple gas to start with the development of his theory. He then defined the statistical probability of a given "state" of the gas containing a very large number of atoms or molecules. Applying Newtonian mechanics to analyze the properties of the collisions between the atoms or molecules of the gas and, to describe the macroscopic system, averaging (or integrating to be mathematically precise) the behavior of individual atoms or molecules over a very large number of collisions he developed his famous relationship between entropy and the statistical probability of the state of gas.

In order to execute this program, Boltzmann had to make an assumption about the state of the gas in the past, that is, before the time at which Boltzmann's description of things starts. Not unreasonably, he assumed that the atoms or molecules moved completely randomly and that nothing could be said about the state of the gas before the "clock" in his calculation was started. He called this assumption "molecular chaos" and it is essentially equivalent to the "ergodic hypothesis" which states that the system (a gas in this case) will eventually assume all possible physical states energetically available to the system. Unfortunately, no one has been able to prove that the "ergodic hypothesis" is correct. In fact, we now know that it is not. Thus, Boltzmann's derivation also turned out to be only an approximation. Once again, an approximation that was extremely useful from the practical viewpoint but that contained an important flaw if theoretical rigor is the objective.

Maxwell's equations also led to some fundamental contradictions. The great Dutch physicist Hendrik Antoon Lorenz, struggled for many years with the problem of making Newtonian mechanics and the Maxwellian theory of electromagnetism consistent. This questions was also not resolved until Einstein proposed the special theory of relativity which was to have unforeseen consequences as well.

I have digressed, perhaps too much, by recounting all this history. The point is, that Victor Weisskopf's problem of the "infinitely complex" only became clear as the flaws in the theories that were to explain the "small" and the "large" became apparent. The accident that Newton was able to derive the law of gravitation because the solar system is approximately a "two body" problem and that Maxwell's equations turned out to be linear to a first approximation hid the true nature of things. Henri Poincaré, Ilya Prigogine, and a number of other people began to suspect that the real complexities of nature were hidden in the nonlinear equations that actually governed the real world. It

was of course, in developing solutions of these equations that high speed computers became absolutely essential.

In the past two decades, people like Edward Lorenz at the Massachusetts Institute of Technology, have investigated the behavior of simple nonlinear differential equations on relatively small computers. Lorenz is a meteorologist so he started by looking at very complex non-linear systems such as the atmosphere. However, he was the first to realize that complexity may be a property of the non-linearity rather than the complex nature of the atmosphere. Thus he looked at very simple non-linear equations. What he discovered was that there are domains in which chaotic solutions of simple and perfectly deterministic equations exist. These chaotic solutions can be traced on visual displays associated with modern computers. When this is done, new insights are apparent simply by looking at the pictures that are developed.

What Lorenz and his colleagues saw is that the chaotic solutions of nonlinear equations exhibit subtle patterns that seem to imply some kind of a more fundamental order or regularity. The beautiful patterns of the "strange attractors" on his computer screens, although they described chaotic motions, were themselves evidence of some kind of "order". No one quite understands this at present, but it is an approach requiring computer capacity to develop in the future.

At the same time, people like my colleague Harry Swinney at The University of Texas at Austin have set up experiments that are, if you will, "analog computers" which exhibit chaotic behavior in "simple" physical systems when the right circumstances are imposed. To completely understand these experiments, it will also be necessary to apply high speed computing systems of the most advanced kind. Perhaps the most beautiful example of chaotic behavior is the recent explanation of the orbit and the chaotic tumbling of Saturn's satellite Hyperion. Hyperion moves in Newtonian orbit around the planet Saturn, yet it tumbles in a most unpredictable way, due to the fact that the effect of Saturn's rings on the satellite put it in a region where the solution of its equation of motion is chaotic.

To those of us who received their scientific educations half a century ago, the fact that chaos results from order is surprising. This is but one of the startling consequences of the understanding of Weisskopf's "infinitely complex". An even more startling consequence is that order seemingly can arise from chaos.

**(4) <u>Order out of Chaos: The Genetic Code</u>**

In his monumental work on the origin of species, Charles Darwin commented on the infinite variety of life. If there ever was a phenomenon that is truly "infinitely complex", then it must be the existence of living things. Darwin pondered this question and after much thought, developed the theory of evolution through "natural selection". He maintained that the interaction between living beings and their environments shaped how they evolved. Darwin published his ideas in 1859 and a little less than a century later, they were triumphantly confirmed on the molecular level by Francis Crick, Maurice Wilkins and James D. Watson. What I am speaking of is, of course, the unraveling of the "genetic code".

What was discovered by Crick, Wilkins and Watson in 1954 is that the properties of living organisms are determined by a huge molecule called "deoxyribo- nucleic acid", or "DNA" for short. This molecule is a formidable and highly ordered structure of three billion units. It is wrapped around the chromosomes in the nuclei of the cells which constitute all living beings. When this molecule is unraveled, it stretches to the length of over six feet. The most important part of their discovery is that the molecule contains a code which is different for each living being. This code has to do with a rather simple sequence of four amino acid molecules arranged along the three billion units of the DNA structure. The "infinite variety" of life that Darwin speculated upon

can be explained by the absolutely enormous number of different combinations of arrangements of these four molecules distributed among the three billion units of the DNA. How did this exquisitely "ordered" molecule originate? We do not know.

We do know that the DNA molecule behaves precisely according to the laws of quantum mechanics that Victor Weisskopf talked about in his lecture on understanding the "small". Does the hint of "order" even in Edward Lorenz's chaotic "strange attractors" provide the clue? Is this how "order" arises from "chaos"? No one knows.

The study of the DNA molecule has led to enormously important practical consequences. We have actually been able to pinpoint how certain genetically linked diseases arise from the sequence of the four amino acid along the DNA molecule. Sickle Cell Anemia, Downs' Syndrome, and a number of other conditions result from slightly abnormal sequences of these amino acids along various sections of the molecule. It is of course, this discovery that led people to the conclusion that if we understood the sequence of the entire molecule we would be able to deal with many genetically related conditions that are now not within the reach of treatment. Thus the "Human Genome Project" was initiated.

Up to now, we have precisely sequenced perhaps five thousand of the three billion amino acid units along the chain of the DNA molecule. We have, so to speak, only scratched the surface. In order to do the sequencing job, two things need to be done: One is that the chemistry must be automated because it would simply take too long to perform the complex chemical operations that are necessary by the standard techniques. The second is that, in order to actually "decode" the molecule, an enormous computer program must be built to do what is essentially a cryptological analysis of the sequence. It is, of course, here where once again your expertise becomes important.

There are already some very intriguing results of the work done to date on the DNA molecule. I have already mentioned one which is that very small changes in sequences can have large effects in terms of the macroscopic health of the organism. The second discovery seems to be that there are large sections of the molecule in which the code is not telling us anything, or at least, so we think. These are called "nonsense" sections by some of the practitioners of DNA sequencing. If I had to guess, I believe that eventually we will discover that these "nonsense" sections of the molecule have a very precise purpose. Since I am speaking to computer experts, it may be that the specific sequence changes that characterize the diseases I have mentioned are akin to "machine language" in this code, but that there are also higher level languages that will become apparent only when a complete cryptological job is done. Is it possible that qualities of supreme importance, such as intelligence, judgment and human emotions comes from the existence of such "higher level" patterns in the DNA molecule sequence. I have already asked the critical question: How did this highly ordered molecule originate? Now I have to add another one: How did the patterns and the codes evolve? And if we try to understand the patterns that "cause" the more subjective (and probably more important) qualities of human beings, are we not treading on the feet of God? Are such things even within the realm of what is knowable?

There are many uncertainties, but I am certain of one thing: Computers and data storage systems will have a crucial role to play in decoding the DNA molecule and in understanding its workings.

### (5) <u>The End Game - If There is One.</u>

What does all this mean? Personally, I believe that Victor Weisskopf's scenario is correct. For most practical purposes, we have understood the "small" and the "large" although important questions still remain. There are people who believe that this is true not only for "practical" purposes but that it accurately reflects the state of

"theoretical" knowledge as well. Is it possible to create the "final theory" that explains both the "small" and the "large" and indeed shows that ultimately they come together? Many of our very best theoretical physicists believe so and are working hard to develop it. I tend to agree with this view provided that it is restricted to the domains of Weisskopf's "small" and "large". I also believe that when (or if, for those who are less optimistic) the "final theory" is put before us, it will not help us to deal with the " infinitely complex". We will then have to turn our attention to an entirely new territory and develop novel ways of dealing with the new situations we will confront.

The "infinitely complex" and its understanding is the new challenge with which we will grapple. What we now need to do is to re-examine some of the things that we thought we understood in this new context. What I have tried to do in the past few minutes is to take a short look at two aspects of complexity: the origin of chaos (complexity ) from order and the control of complexity (life) through highly ordered systems simple codes. Since dealing with complexity is really your business, your work will be critical to the success of the most important efforts to understand these things. To me, this is the real meaning and also the value of this conference.

N94-33792

# The United States Digital Recording Industry

**John L. Simonds**
Executive Director
National Storage Industry Consortium
9888 Carroll Center Road, Suite 115
San Diego, CA 92126-4580
Phone: 619-621-2550
FAX: 619-621-2551
jsimonds@ucsd.edu

The recording industry resembles the semiconductor industry in several aspects. Both are large (>$60 Billion/year revenues); both are considered critical technologies supporting national objectives; both are experiencing increased competition from foreign suppliers; they recognize significant opportunities for both technological and market growth in the decade to come; and both realize that a key to this future growth lies in alliances among industry, academia, and government.

The semiconductor industry has made significant investments in alliances relating to manufacturing technologies (SEMATECH) and to joint long-term technology research centered in universities (SRC). The federal government has provided funding support of these efforts in recognition of the critical roles semiconductor technologies play in national interests.

The recording industry is now also forming critical alliances, but has been slower in starting and in gaining broad recognition by government agencies and legislators that the industry needs federal support. Traditionally, the recording industry has been viewed as mature, stable, and, while critical to national interests, able to chart and fund its own course toward future national needs. That perception is fortunately changing.

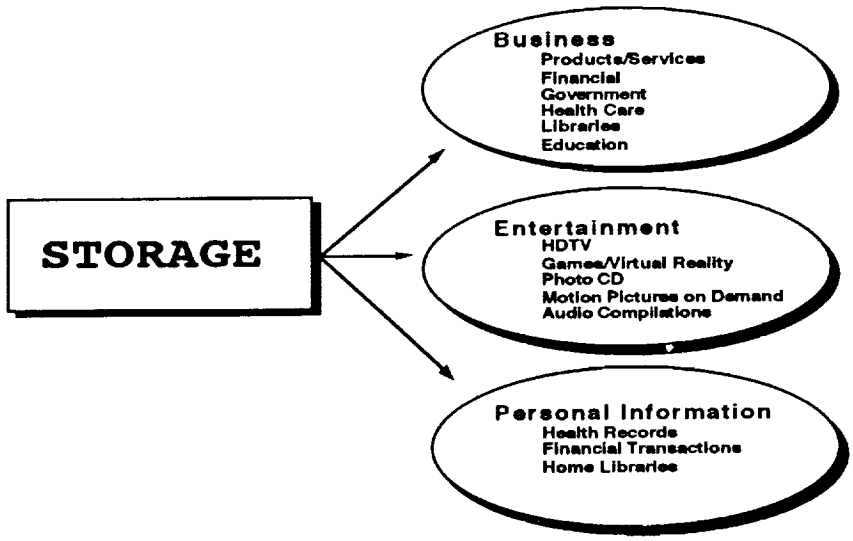
## Industry Challenges

In fact, the recording industry faces unprecedented challenges. Foreign companies play a dominant role in all aspects of consumer recording and in the supply of components and in manufacturing for other recording products. At the same time, U.S. recording industry profits have eroded (or gone negative), forcing personnel downsizing and even corporate failures. Manufactured quantities of hard disk drive units have been growing at a compound annual growth rate (CAGR) of 14%. In terms of recording capacities, the total aggregated storage capacity of the hard drives shipped is increasing at a CAGR of 44%. The result of all this is a current oversupply of disk drives, driving profits of some companies to new lows. Even if U.S. corporations can survive these problems, economic pressures have made it increasingly difficult to make long-term investments in research.

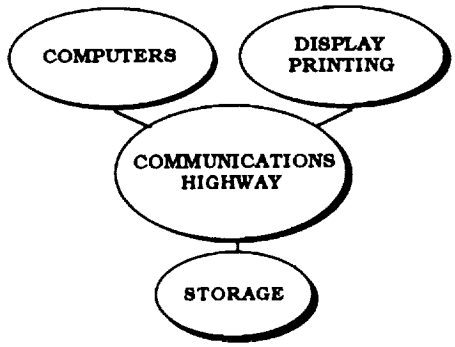
## Opportunities for Growth

On the positive side, the long-term market growth potential for recording is significant, assuming that the industry can survive the current problems.

The figure below depicts three main market growth vectors for recording: (1) expanding business, education, health, and governmental markets; (2) increased opportunities for recording in entertainment services (driven largely by HDTV); and, (3) a potential for a large growth in personal information systems.




These opportunities are consistent with the vision for the National Information Infrastructure that has been widely discussed by a variety of governmental sources. This concept involves a central communications highway which is fed by high-performance systems of computers, software architectures, displays/printers, and, of course, digital storage.




Combined, these opportunities for growth could result in a market size for recording within the next ten years which is more than an order of magnitude greater than at present. In fact, one executive of a large U.S. company speculated on a market size approaching a trillion dollars per year in that same time frame. The challenge for the recording industry is to build these future markets by developing systems solutions for the new storage-intensive applications.

Foreign corporate strategies clearly have targeted the growth opportunities in the entertainment and consumer segments, with the ancillary expectation that the resulting technology and manufacturing capabilities will feed product offerings in the commercial and governmental sectors. Further, foreign governmental support of the requisite alliances to make this all happen is both strong and mature.

Corporate planners in this country are no less perceptive of the opportunities, but they have traditionally experienced barriers to form the same level of alliances toward focused objectives. First, it was once common for the U.S. government to resist corporate alliances for anti-trust reasons. Fortunately, that situation has been significantly improved in recent years. Second, our litigious society raises barriers for organizations to work together[1]. Third, there is an inherent distrust among some organizations to work with each other. We come from

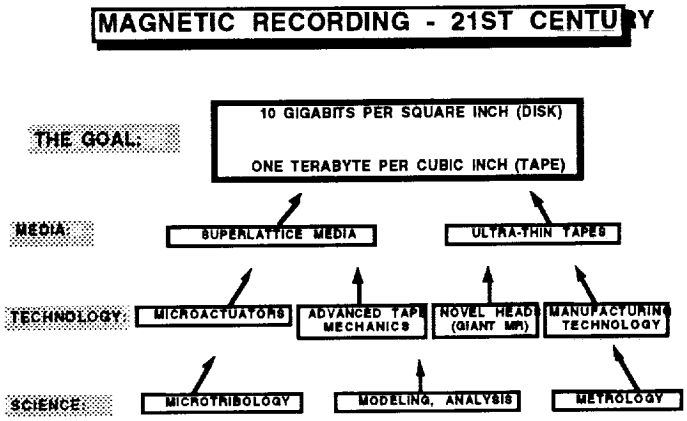
[1]The author can be induced with almost no urging to relate his harrowing experiences in trying to get corporate and university lawyers to agree on intellectual property agreements, for instance.

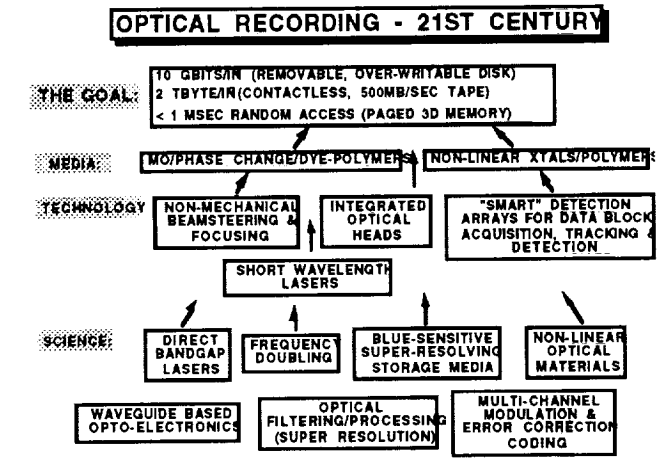
a history of each company acting independently toward market development and penetration. And fourth, our federal and state governments are still seeking the most effective ways to provide support for critical industry segments.

## Alliances

Like the semiconductor industry, the U.S. recording industry is taking steps to work together toward the opportunities that are commonly perceived. The National Media Laboratory, based in Minneapolis, is an effective organization which addresses government users and their needs for systems support and testing. More recently, the National Storage Industry Consortium (NSIC) has been formed[2] with the objective of enhancing the competitiveness of the U.S. recording industry through a strategic plan to form joint research programs on pre-competitive technologies and to coordinate technology developments among corporations, universities, and governmental organizations.

NSIC today has 36 member companies and over 30 universities which support the ongoing joint research programs that have been established. Early in its development, NSIC held workshops to prepare technology roadmaps for the storage industry. The following two figures summarize elements of NSIC's strategic plan for hardware and media technology developments; this plan is now being updated in detail and is being augmented by programs in software systems and in manufacturing technologies.



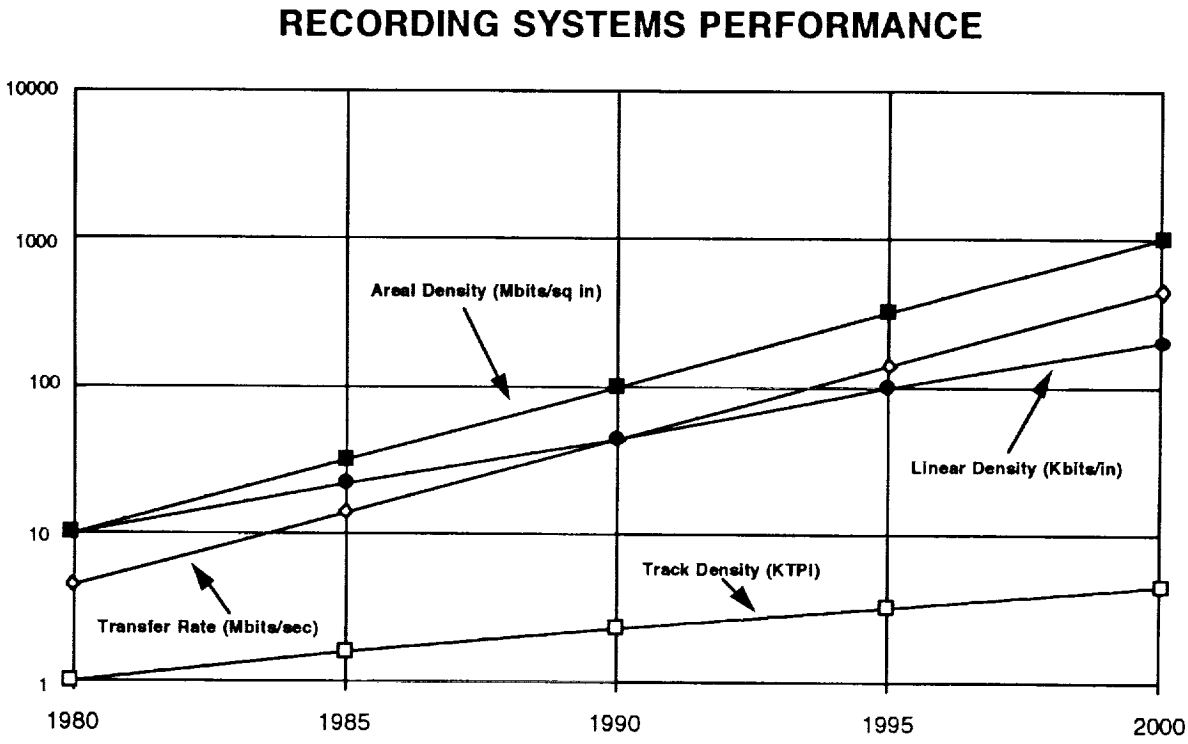




[2]Incorporated in California, April, 1991

**CURRENT NSIC JOINT RESEARCH PROGRAMS**

On the basis of this early work, members of NSIC proposed key joint technology research programs which have subsequently been established and which are currently partially funded by both NIST/ATP and ARPA. The goal of these programs has been to create pre-competitive technologies which will enable magnetic disk recording at 10 Gigabits/$in^2$, magnetic tape recording at one Terabyte per cubic inch, and optical recording at 10 Gigabits/$in^2$.

The figure below depicts a number of recording system performance parameters as a function of time. Our industry has traditionally produced technology advances which, for most of the




important performance parameters, plot linearly on the semi-log scale shown. These linear plots indicate, for instance, that, by the year 2000, the industry would be expected to provide products with areal densities of one Gigabit per square inch and track densities of less than 10,000 tracks per inch.

The NSIC goals postulate performance well beyond the usual industry trajectories in each of the parameters shown. By setting targets well beyond normal industry expectations, we encourage non-evolutionary approaches to technology development. To illustrate this point, current NSIC programs have goals for areal densities of 10 Gigabits per square inch and track densities (for magnetic disks) around 25,000 tracks per inch. Reference to the figure above shows that these targets are significantly beyond the progress that would traditionally have been expected within this industry.

The table on the following page is a financial summary of four present major NSIC joint research programs. All data are expressed as $K.

# NSIC 5-YEAR JOINT RESEARCH PROGRAMS

| PROGRAM | FUNDING AGENCY | FEDERAL FUNDING | NSIC FUNDING | TOTAL COSTS |
|---|---|---|---|---|
| SWAT (Start: 5/91) | | | | |
| Years 1 - 5 | ATP/NIST | $5,421 | $8,862 | $14,283 |
| HEADS (Start: 8/92) | | | | |
| Years 1 - 5 | ATP/NIST | $5,534 | $6,246 | $11,780 |
| UHD RECORDING (Start: 3/93) | | | | |
| Years 1 - 2 | ARPA | $10,700 | $11,633 | $22,333 |
| Years 3 - 5 | NOT YET IDENTIFIED | $17,335 | $20,360 | $37,695 |
| HOLOGRAPHIC RECORDING MATERIALS | | | | |
| Years 1 - 3 | ARPA | $6,272 | $6,272 | $12,545 |
| **TOTAL ALL PROGRAMS** | | **$45,262** | **$53,373** | **$98,635** |

The first program (SWAT - funded by NIST/ATP) is aimed at producing short wavelength sources for optical recording. The approach is to use nonlinear optical materials to effect frequency doubling of red diode lasers to produce blue integrated sources with an attendant reduction in mark sizes on optical media. The second ATP-funded program addresses new magnetic heads technology to meet the high-density goals set forth above. The third program on the list is a large ARPA-funded program with a variety of technical objectives in both optical and magnetic recording. The fourth program (PRISM - also ARPA funded) is aimed at development of holographic recording materials which will be stable and which can be manufactured reliably at low cost. Figure 5 substantiates that this industry and its government sponsors are seriously committed to using this process of joint research for advanced technology developments in this critical industry.

Still other NSIC programs are in the proposal stage. A 5-year program leading to digital optical tape recording capable of providing several Terabytes of data in a small cartridge (similar to a 3480 cartridge) has been proposed to ARPA as a TRP proposal. Another proposal is being put together relating to the development of prototype holographic data recorder systems which utilize the materials being developed in the PRISM Project. Yet another in process is a program addressing manufacturing technologies for the recording industry.

## SOFTWARE, SOFTWARE ARCHITECTURES

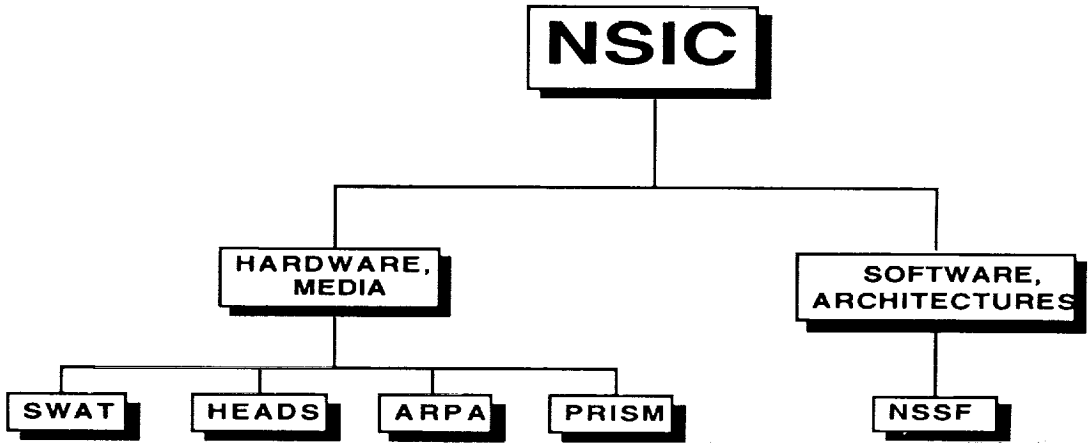
NSIC to date has focused on hardware and materials aspects of optical and magnetic recording. We recognize that there is an equally critical need for development of software relating to mass storage systems. Quoting from pending legislation:

*"NASA, ARPA, NSF and appropriate agencies shall develop technologies for "digital libraries" of electronic information. Development of digital libraries shall include ..... development of advanced data storage systems capable of storing petabytes of data and giving thousands of users nearly simultaneous access, .....development of database software capable of quickly*

*searching, filtering, and summarizing large volumes of text, imagery, data, and sound, .....development and adoption of standards for electronic data, .....technology for simplifying the utilization of networked databases distributed around the nation and around the world."*

These software developments are essential from two points of view: (1) they are necessary if we are to take advantage of the significant gains expected in the performance of recording technologies, and (2) achieving the goals of the National Information Infrastructure (NII) demands that these capabilities exist to enable efficient database sharing throughout the network.

For these reasons, NSIC, working with the National Storage Laboratory at Lawrence Livermore National Laboratory and other industry software developers, has recently formed a new division of NSIC, the National Storage System Foundation (NSSF).




This division, like the earlier NSIC structure, will have membership from NSIC industrial companies, universities, and the national laboratories. It is intended to augment and extend existing collaborations and standards organizations in standardizing, developing, and transferring technology for high performance storage systems.

The objectives of NSIC/NSSF are to:

- Create a United States digital library strategy
- Develop core technology for high capacity, high performance digital libraries and storage systems
- Develop technology for simplifying access to digital libraries
- Define a coherent storage system infrastructure
- Establish requirements for buildable components
- Promote interoperability among components from different developers
- Encourage the development and adoption of standards
- Organize and seek funding for collaborative research projects for next generation digital libraries and storage systems

The establishment of NSSF will encourage, through NSIC's central coordination, interactions between the software and hardware communities. Hopefully, both divisions of NSIC will influence and enhance each other's offerings in designing approaches to the needs of NII.

## So, what's next?

NSIC is presently sponsoring a detailed process of producing an updated and detailed technology roadmap for this industry - both software and hardware components. Participating in this process are industry, university, and national laboratories persons. Government persons are also invited to participate (any who would like to join are asked to call the author). In addition, a comprehensive National Plan for the recording industry is being prepared. This plan will describe the recording industry and its several segments in terms of both business issues and technology needs for future development. It will make use of the technology roadmap referred to above. It will conclude with a set of recommendations for government action in support of this industry.

At this time, it is premature to review the specific recommendations which will be made for government action. It's safe to say that these will include a request to assist this industry by a process that starts with a committed federal budget amount for investments in support of this industry. This budget would be used to fund new joint research programs, university or national laboratories work, or other activities which are judged to be important to NSIC's mission of enhancing the competitiveness of the U.S. recording industry.

# Toward a Digital Library Strategy for a National Information Infrastructure

Robert A. Coyne
Harry Hulen

IBM Federal Systems Company
3700 Bay Area Blvd., Houston TX 77058

## Abstract

Bills currently before the House and Senate would give support to the development of a National Information Infrastructure, in which digital libraries and storage systems would be an important part. A simple model is offered to show the relationship of storage systems, software, and standards to the overall information infrastructure. Some elements of a national strategy for digital libraries are proposed, based on the mission of the nonprofit National Storage System Foundation.

## 1. National Information Infrastructure Background

Two bills before the current session of Congress call for the creation of a National Information Infrastructure. The bill before the House of Representatives is called the "National Information Infrastructure Act of 1993" [1], and a somewhat similar bill before the Senate is called the "National Competitiveness Act of 1993" [2]. Whether or not either of these bills are passed, the fact that these bills have reached the level of serious committee discussion has a far reaching impact on the storage industry. The Senate bill states that "While the private sector must take the lead in the development, application, and manufacture of new technologies, the Federal Government should assist industry in the development of high-risk, long-term precommercial technologies which promise large economic benefits for the nation, ... and cooperate with industry and academia to help create an advanced information infrastructure for the United States. The term "information infrastructure" is defined in the Senate bill as "a network of communications systems and computer systems designed to exchange information among all citizens and residents of the United States."

Both bills propose to support the development of digital libraries as part of the information infrastructure. Some of the key provisions which relate to the underlying storage systems are,

- "Development of advanced data storage systems capable of storing hundreds of trillions of bits of data and giving thousands of users simultaneous and nearly instantaneous access to that information;"
- "Development of means for simplifying the utilization of networked databases distributed around the nation and around the world;"
- "Encourage the development and adoption of common standards and, where appropriate, common formats for electronic data."

The references to information infrastructure, storage systems, and digital libraries in the proposed legislation demonstrate an important shift in the perception of what constitutes our national information assets. In 1987, the Executive Office of the President, Office of Science and Technology Policy, published "A Research and Development Strategy for High Performance Computing" [3], which found its way into the High Performance Computing Act of 1991. The four areas of research supported by the program which came to be called the High Performance

Computing and Communications Initiative were high performance computers, software technology and algorithms, networking, and basic research and human resources. Storage systems were supported only indirectly, to the extent that they were needed by the computing and communications elements. The proposed legislation, which is heir to and which references the HPCC legislation, still emphasizes networking and various aspects of computation, but the acknowledgement is there that storage systems are an integral part of the information technology that forms our national information infrastructure.

## 2. A Model for National Information Infrastructure

A simple model for the components of a national information infrastructure is diagrammed in Figure 1. At the top of the figure is a layer representing users. As defined by Congress, the users are the American people - children in school, individuals in their homes, and entrepreneurs creating new opportunities and new jobs. To say this another way, the users are not just academic and government researchers.

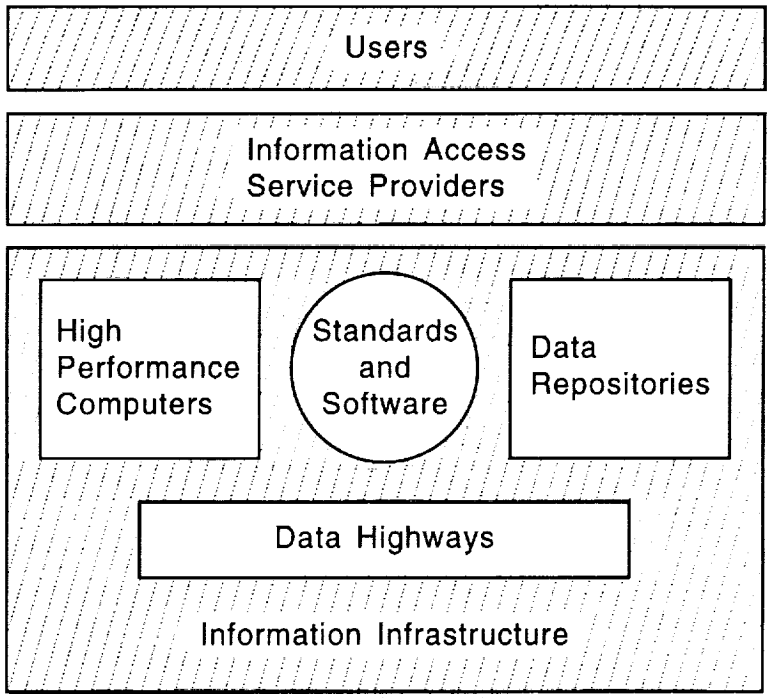



Figure 1. A Model for Information Infrastructure

To access the information infrastructure, there must be a layer of service providers. In our model the information access service providers are a layer of entrepreneurial service offerings which make the nuts and bolts of the infrastructure available to users. In a recent Business Week article entitled "The Cleavers Enter Cyberspace" [4], the lead sentence refers to this layer by asking, "Is Middle America ready for Internet?" The article describes how services such as Prodigy, CompuServe, GEnie, and America Online are offering Internet access to hundreds of thousands of users. Other examples of services available today are access to stock market information, airline reservations, banking, and thousands of forums and bulletin boards. Will industry find it profitable to deliver access to the medical research libraries of NIH, weather and environmental data bases of NOAA, earth observation data bases of NASA Goddard, and the astronomical catalog from JPL? The expectation is that by developing the infrastructure and educating the public, opportunities will abound to provide value-added access to the nation's information assets.

The information infrastructure layer itself is the focus of our interest in this paper. In our model, the infrastructure consists of four components: high performance computers, data repositories, the networks which it is now fashionable to call data highways, and the standards and software which enable the other components to work together.

High performance computers and networks have had center stage for the last several years. Our model takes a cue from the proposed NII legislation and elevates the data repositories to an equal billing. Repositories are the massive storage systems comprised of disk arrays, high density tape, optical media, and robotic media libraries. The fourth item, Standards and software, has been acknowledged by HPCC and NII sponsors to be a key enabling component. In our model, standards for information infrastructure would encompass standards for storage systems, such as the ones being developed by the IEEE Storage System Standards Working Group. Software would encompass the file systems, database management systems, storage servers, intelligent data movers, and physical volume managers.

## 3. Toward the creation of a United States digital library strategy

We must all seize the moment to formulate a strategy in plain, simple English to support the National Information Infrastructure with technologies for digital libraries, storage systems, and software. One organization which is working toward formulating such a strategy is the National Storage System Foundation (NSSF), a newly formed division of the National Storage Industry Consortium. NSIC is a not-for-profit business league chartered in California. The mission of the NSIC's NSSF division is to promote and support joint academic, industrial, and governmental research in information storage systems and software. The following outline for a digital library strategy is based on the mission statement of NSSF.

*Develop core technology for high capacity, high performance digital libraries and storage systems*

Digital libraries and storage systems can be very large, they can be geographically distributed using high speed networks, and they can be complex systems containing many kinds of data and many varieties of hardware and software components. We must, of course, be concerned with developing the core hardware technologies which provide the physical storage for digital libraries. We must also be concerned with developing the overall software architecture of the digital library systems. This includes research and development of key software components that are not yet available and integrating the hardware and software to create digital library systems. Other architectural issues are the ability to scale storage system both in size and in performance, to distribute them geographically, to make them secure, and to allow nondisruptive insertion of new technologies.

*Develop technology for simplifying access to digital libraries*

The successful deployment and utilization of the National Information Infrastructure is dependent upon massive amounts of data, stored in digital library systems, to be readily and easily available to consumers of this information. We must work to promote the development of software and systems which will make this possible. This includes technology to categorize and organize data, methods for optimizing data organization for rapid retrieval, and technology for extracting metadata. It includes technology to search, filter, and summarize large volumes of data. It includes technology for handling text, images, sound, and numerical data. It includes user interfaces using graphical and expert system technologies as well as automated access from other computers.

*Define a coherent digital library infrastructure*

The components which define the effectiveness of a digital library system, whether on a local or national scale, include computers, software, storage hardware and networks. Subcomponents include the security environment, the systems management environment and many other facets of

information access and retrieval. For a large number of these component and subcomponent areas, standards exist or are being developed. We must work to identify usable, coherent environments from the available choices and to identify areas where additional work needs to be done.

*Establish requirements for buildable components*

Not every company has the interest or resources to build an entire storage system. Our goal should be to define storage systems in such a way that specialists can build a software component with reasonable certainty that it can operate in a system with software components from other sources. As with hardware, the definition of components is a combination of historical precedent, feasibility, and standards. By defining building blocks and interfaces that conform to standards, we can enable different organizations to develop components in their areas of expertise with the confidence that these components will work with components from other developers.

*Encourage the development and adoption of standards*

We must encourage the establishment of standards for the storage industry through the IEEE Storage System Standards Working Group, ANSI X3, and other standards organizations. The IEEE Mass Storage System Reference Model has already taken significant steps to define the broad outlines of a future scalable standard for open storage system interconnection.

*Promote interoperability among components from different vendors*

A critical factor in the rapid market acceptance and deployment of digital library systems is the ability of hardware and software products from many vendors to seamlessly operate together. We must promote the idea of interchangeable and/or interoperable hardware and software components. To this end, we should support the development and use of standard test cases and reporting procedures to validate compliance with defined standards and interface definitions. Compliance enforcement is not a function of the IEEE and other standards organizations. This makes it possible to establish clearinghouses in which interoperability of digital library components and systems can be tested and demonstrated. Clearinghouses could be commercial operations, nonprofit organizations such as NSIC/NSSF, or an informal network of researchers and users.

*Promote collaborative research projects for next generation digital libraries and storage systems*

We need to initiate more joint research projects among industrial, university, and governmental research units to focus on the core technologies of the next generation of information management systems, including scalable, high performance, high capacity digital libraries and storage systems. Collaborative research proposals from the storage system community will help to focus governmental and industrial research funds on the development of digital library and storage system technology. Collaborative research will go a long way toward ensuring openness and interoperability among components. Collaborative research will lower the cost for everyone of building the national information infrastructure.

## References


1. U.S. Congress, *National Information Infrastructure Act of 1993 - H.R. 1757*, Washington, DC, July 13, 1993.
2. U.S. Congress, *National Competitiveness Act of 1993 - S. 4*, Washington, DC, January 21, 1993.
3. Office of Science and Technology Policy, Executive Office of the President, *A Research and Development Strategy for High Performance Computing*, Washington, DC, November, 1987.
4. Schwartz, Evan I., "The Cleavers Enter Cyberspace," *Business Week*, October 11, 1993.

N94-33794

# Magnetic Field Sources and Their Threat to Magnetic Media

**Steve Jewell**

12538 Rochester Dr.
Fairfax, VA 22030
(703) 631-0724

## Introduction

### General

Magnetic storage media (tapes, disks, cards, etc.) may be damaged by external magnetic fields. The potential for such damage has been researched, but no objective standard exists for the protection of such media. This paper summarizes a magnetic storage facility standard, Publication 933, [1] that ensures magnetic protection of data storage media.

### Background

Magnetic field sources can occur naturally (lightning) or unintentionally (ac line shorts, ground faults). In addition, the espionage threat exists that some unauthorized person or group could use high-energy magnets to destroy data from some distance away.

The existing standards on this subject [2][3] do not detail the magnitude of the magnetic fields which can be generated, nor the susceptibility threshold of the magnetic media. Instead, the fields are estimated and experimentally tested using magnets which are orders of magnitude less than those possible (and commercially available) today.

What are the threats to magnetic media? This paper summarizes research performed by the author to:

* Quantify the largest magnetic field that could be generated (now and in the near future);
* Characterize the magnetic susceptibility of a variety of magnetic media currently in use;
* Analyze the propagation of a magnetic field in conjunction with the susceptibility of the magnetic media;

and finally,

* Determine the spacing between hypothetical worst-case magnets and magnetic media to ensure that tape or disk erasure will never occur.

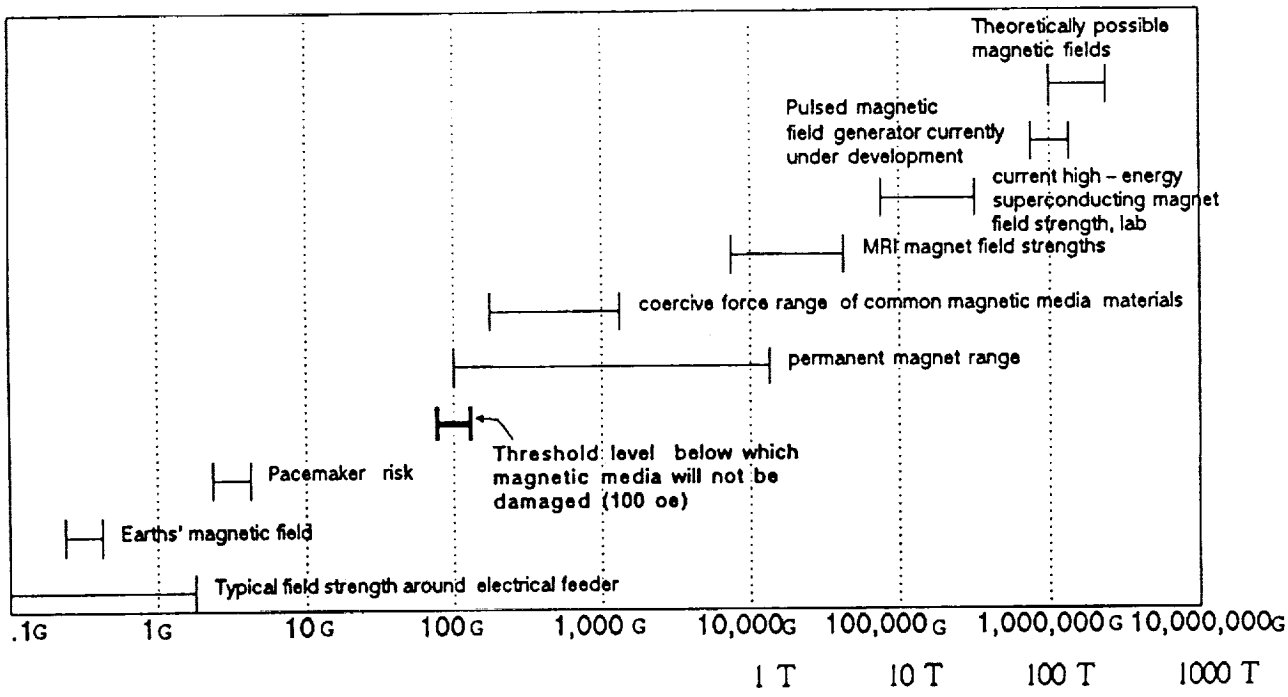
Publication 933 is a new standard that presents minimum spacing requirements between magnetic media and potential magnetic field sources. The procedure ensures both vendors and users of magnetic storage media that their data is safe from magnetic corruption.

### Magnetic Basics

Before we begin the discussion, let's review some common magnetic terms and units. Magnetic fields are created whenever a current flows. The amplitude of the magnetic field is proportional to the amount of current flow. The units of the magnetic flux (or magnetic field), **B**, are tesla (1 tesla = 1 weber/meter$^2$), or, for smaller magnetic fields, gauss (G), where:

1 T = 10,000 G

For reference purposes, the stationary magnetic field of the earth is about .5 gauss. The field of a small permanent magnet can range from 100 G to 13,000 G, and today's superconducting electromagnets can produce steady-state field strengths as high as 500,000 G (or 50 T) [4][5]. Some common magnetic devices and their corresponding fields are shown in figure 1.




MAGNETIC FIELDS AND MAGNETIC SOURCES
(G = value in Gauss, T = value in Tesla)

Figure 1. Magnetic Fields and Field Strengths

The magnetic field vector, **B**, is also known as the magnetic induction, or the magnetic flux density. It should be distinguished from the magnetic field intensity, **H**, which is different, but is also referred to as the magnetic field. The magnetic field intensity is expressed in Oersteds, which are equivalent to gauss in free space. For purposes of simplicity, the value for B and H shall be assumed equivalent for purposes of this paper.

## Magnetic Threat

This section discusses magnetic field sources that could destroy magnetic media. The magnetic fields could be generated by natural occurrences (lightning), accidental events (shorting of ac conductors), or other means. However, this chapter will focus on high-energy fields intentionally generated to destroy magnetic media at a distance.

As figure 1 shows, magnetic fields generators up to 50 T have been built, and fields as high as 380 T [6] are theoretically possible. Magnetic field generators of this size use superconductors to carry the enormous currents required.

High-energy, superconducting electromagnets require careful design to keep the coils at cryogenic superconducting temperatures. In addition, the internal magnetic stresses applied to the superconductors can tear the conductors apart, or cause the conductors to revert to a non-superconducting state.

Future advances in room temperature superconductors and the energy transfer capability of batteries and capacitors will significantly improve the ease with which high-energy magnets can be constructed. Indeed, the current state-of-the-art will improve as manufacturing processes are discovered for known high temperature superconductors. The net result will be an increased capability for generating high level magnetic fields that are portable. What then is the maximum field strength that could be developed by a hostile force with sufficient means and motives?

The data available today would limit the value to some field strength below about 50 T, even in a pulsed, destructive mode. However, it is naive to presume that energy storage technology will not also improve with time. In keeping with the intent of providing an absolute worst-case scenario, the highest possible instantaneous applied magnetic field strength is estimated to be 500 T.

The 500 T upper bound was selected because it is unlikely that superconducting magnet technology will yield a magnet exceeding this value in the near future. A more important reason is a fundamental premise of the 933 series of documents. The premise, stated more succinctly, is:

> **This standard (Publication 933-1) defines a process for determining the level of magnetic protection of a facility. The absolute level selected (500 T or 50 T), is not as important as the *process*. If a facility manager chooses to partition the facility space into zones that provide 500 tesla protection, or 5,000 tesla protection, he may do so, as long as the requirements of Publication 933-1 are met at the level specified.**

The purpose of determining the maximum magnetic field, then, is not to define the real threat today. The purpose of the process is to set a limit that magnetic storage media users and providers can agree upon is a worst-case magnetic field.

The upper bound provides a standard that represents the maximum protection required, as opposed to the maximum possible (which is currently much lower). The field applied is assumed to be a constant field (worst-case), to eliminate ambiguities about pulse characteristics or eddy current shielding.

The 500 T limit also may be zoned into lower levels of protection (such as 5 tesla and 50 tesla protection zones) in areas where the 500 T limit cannot be applied. It is then the customer's choice to purchase the magnetic storage protection that is required.

The 500 T limit depends on the magnetic susceptibility of the magnetic media. While 500 T (or 5,000,000 oersteds) is the maximum threat, the next section discusses the threshold for the minimum field that can cause magnetic damage.

## Magnetic Media Characteristics

Data can be stored on many types of magnetic media. Disks, floppy disks, tapes, and tape cartridges are the most commonly used magnetic media storage devices. Each of these devices consists of a substrate of poly (ethylene terephthalate) (PET) that is coated with a thin film of magnetic coating (nominally 2 to 5 um thick) [7]. The coating is comprised of a polymeric binder, lubricants, curing agent, solvents, and magnetic particles. The particles selected for a medium are dependent upon a variety of factors including cost, required storage density, and magnetizing force.

The coercive force of the magnetic media is the amount of applied field required to reverse the magnetic field in the material. As figure 1 shows, commercial magnetic media have a coercive force between about 200 and 2000 oersteds (or gauss, in free space). The traditional research [2][3] held that the coercive force is the minimum threshold to induce magnetic destruction. Discussions with industry leaders on this subject [8][9], however, indicated that "levels significantly below this value may induce data errors" [10].

The results of a detailed research effort indicated that the minimum field likely to cause magnetic damage to a tape with a coercivity of 200 oersteds or greater is approximately **100 oersteds**. The failures of disks and tapes that have been subjected to field strengths below this figure shall be considered non-destructive in most instances.

## Magnetic Field Generators and Models

Magnetic fields can be generated by many means, but electromagnets are the most common source of high energy magnets. Whenever current flows, a magnetic field is generated. The threat to nearby magnetic media is dependent upon:

* The amount of current being carried in the conductor(s) of the electromagnet
* The size of the electromagnet
* The number of windings or parallel current paths for a multi-conductor electromagnet

The most common magnetic field generator configurations are created by long, thin wires, and circular loops (with multiple windings). When fields are generated in this fashion, simple equations can be used to characterize their magnetic field patterns. Publication 933 discusses the derivation of fields in this manner.

In order to provide a consistent basis for the specification, a standard model was needed to model a variety of real-life magnetic field generators. The models and some simple formulas are presented below.

### Magnetic Fields from a Current Carrying Wire

The magnetic field produced by a thin wire of infinite length [11] is

$$\mathbf{B} = \frac{\mu_0 I}{2 \pi R} \quad \text{tesla} \qquad (1)$$

where:

R = perpendicular distance from the wire to the point in question (in meters)

$\mu_0 = 4\pi \times 10^{-7}$ H/m (the permeability of free space)

and,

$I$ = the current in the conductor in amps

This equation is valid for most single conductors when the point in question is close to a long wire. This equation is used to model the threat from lightning protection systems, ac short circuits, nearby power sources, and other high current line conductors.

### Magnetic Fields from a Wire Loop

The next case we review is that for magnetic fields generated from a wire loop. This type is the most commonly used to generate the high-energy fields of superconducting magnets. For ease of presentation, we will forego a discussion on the related subjects of heat dissipation and stress forces associated with high-energy superconducting magnets.

The field of a thin wire bent into a circular loop is easily modeled for the magnetic component on the axis of the loop (the worst case magnetic field). If we consider a magnetic loop with a radius $a$, the magnetic field at a point (on the axis of the loop) at a distance $z$ from the loop is:

$$\mathbf{B} = \mathbf{B}_z = \frac{\mu_0 I a^2}{2\left(a^2+z^2\right)^{\frac{3}{2}}} \quad \text{tesla} \tag{2}$$

where both $a$ and $z$ are expressed in meters

If we examine this equation, we see that in the far field condition ($z >> a$), the equation reduces to:

$$\mathbf{B} = \frac{\mu_0 I a^2}{2 z^3} \tag{3}$$

Also derived from equation (2) is the field in the center of the loop, given by:

$$\mathbf{B} = \frac{\mu_0 I}{2 a} \tag{4}$$

**Modeling Discussion**

The magnetic field equations presented are a simple means of determining how fields propagate. In order to create a standard for media protection, however, we also needed to define some fixed parameters and definitions. For example, the following primary assumptions were made:

* The magnetic field from an external, uncontrolled source (an intentionally generated source), shall be assumed to be generated from an infinitely thin loop of infinitely thin wire, with a radius of .1 meters. While a practical magnet must deviate from this value, the assumption allows us to use the equations presented above, and the radius provided is a realistic value.

* The realistic threat must be addressed. If a hostile entity wished to destroy magnetic media, it would be far easier to directly access the space with a small magnet (or other means) than use an expensive, cumbersome superconducting electromagnet. The protection against a magnetic threat should match the physical security already in place at the facility.

  In highly secure storage facilities, the magnetic threat can be divided into two distinct types of threat, defined as follows:

  Type I Magnet - The Type I magnet is small enough to fit into a briefcase, and can be carried into common areas and unprotected areas of the storage facility itself. The Type I magnet has a maximum applied field of 5 Tesla

  Type II Magnet - The Type II magnet is capable of generating field strengths of up to 500 Tesla. Since this type of magnet would require much more preparation time and energy storage components, Type II magnets can only be placed in locations which are not continuously patrolled or inspected (i.e. adjacent office spaces, exterior walls/roof, tenants above and below the magnetic storage area (MSA).

* Damage to magnetic media can come from a variety of sources, including natural and man-made sources. The threat from a lightning strike is at least as likely (and as potentially damaging), as the threat from hostile forces. Electrical power sources, and structural steel members can also produce high level magnetic fields which could damage magnetic media.

- The key notion for magnetic protection is the magnetic protection zone level. This level is the minimum magnetic field which could damage magnetic media in the MSA. The area is partitioned into magnetic protection zones, each of which are rated based on their magnetic protection characteristics. For example, a tape storage rack next to an external wall may have a magnetic protection zone level of 1 tesla, while the center of a huge vault would have (no greater than) a 500 T level.

The computational model and associated variables for each of the magnetic field threats described in Publication 933 are presented in Table I.

## Table 1. Summary of Magnetic Field Sources and Modeling Parameters

| Magnetic Field Generator/Source | Computational Model Model (Equation #'s) | Minimum Spacing[1] | Comments |
|---|---|---|---|
| Type I Electromagnet | Infinitely thin wire loop, with .1 meter radius, equiv. field in center. Use (4) to calculate I, and (2) to determine field propagation range. | 2m | Field equivalent to magnetic protection zone required. No fields > 5 T, or < 1 T. |
| Type II Electromagnet | Same as above | 4m | Same as above, but no fields > 500 T, or < 1 T. |
| Lightning rod grounded conductor | Infinitely thin wire of length equivalent to conductor. Use 200,000 A current or less. | 4m | Reduce current if parallel grounded conductors are present. |
| AC shorts and related conductors | Infinitely thin wire of length equivalent to conductor. Use short circuit current rating of distribution panel or 10,000 Amps, which-ever is greater | 1m[2] | Use this value for ac outlets, lights, phone lines, power feeders, thermostat and control lines. greater |
| Metallic beam or member | Infinitely thin wire loop with radius equiv. to 1/2 the longest diagonal of beam or member | 1m[3] | Use for any beams that are not bonded, use B = 2 Tesla. |
| Water pipes | See ac shorts | 1m | |

[1] Ignore conductors or magnetic sources that are further than this distance from the MSA.

[2] Use .5 meters for conductors completely contained within controlled space.

[3] For beams with diagonal < .3 meters. Use .1 meters for rebar with diagonal < 2 cm

Publication 933 provides additional guidance on selecting potential threats, modeling their field propagation, and specific guidance on using the equations.

## Magnetic Shielding

Some Magnetic Storage Area (MSA) vendors may choose to shield certain portions of their space to improve their magnetic media protection. Other vendors may have existing steel rooms that can be characterized according to the calculations presented herein. A full treatise on magnetic shielding is far beyond the scope of this paper, but characteristics of magnetic field shields are presented in Publication 933.

### Summary of Magnetic Shielding Properties

The following statements summarize the characteristics of magnetic shields, and lend insight into their design.

1) If a Magnetic Field shield is exposed to a high level magnetic field, and is insufficiently thick, the material will saturate (become permanently magnetized), and provide little or no magnetic shielding.

2) Constant or low frequency magnetic fields can only be shielded with steel or ferrous metal. The magnetic properties (B/H curve, saturation induction) of the material, as well as its thickness must be used to ensure that the steel will not saturate before it performs its desired shielding goal.

3) Magnetic fields fall off as 1/distance, 1/distance$^2$, or 1/distance$^3$. Therefore, the threat magnet must be accurately modeled to determine that field to which the shield will be exposed.

4) Shielding against high-energy magnetic fields is impractical at close range because of the requirement for thick magnetic material.

To illustrate the field strength reduction from magnetic shielding, we now review figure 2. Figure 2 shows several curves which illustrate the free space falloff distance for electromagnets of various field strengths. For example, a .01 Tesla (100 gauss) may destroy data at a separation distance of 0 meters, while a 10 Tesla field can destroy data at distances up to 1 meter (about 39 inches). These distances are based on a .1 meter radius electromagnet as described above.

When shielding steel is correctly installed to the walls of the magnetic storage area, the protection level of the space may increase. Figure 2 shows the minimum separation distances between various thicknesses of steel (cold-rolled, low-carbon strip steel [12]) and the magnetic shielding separations and field strengths [13]. For example, if a shield is required to increase an MSA up to 50 Tesla protection, and if the separation distance (distance between the exterior threat and the shield) is about .5 meters, a minimum of 2 inches of steel will be required for the shield.

As the example shows, shielding can be an expensive proposition, especially when the shield must be close to the external threat or the field to be shielded is excessive. The shield must also be constructed to provide good magnetic flux transfer, which usually involves welding and additional attachment mechanisms. Structural loading can also be a problem.

For this and other reasons, the best magnetic shielding mechanism is physical separation between the magnetic storage area (MSA) and the magnetic threat. While it may be difficult to obtain such separation in some existing facilities, careful design and site selection may reduce or eliminate the cost of protecting against magnetic threats. The next section discusses the magnetic protection certification, a means of verifying that magnetic media stored in a facility could not be adversely affected by magnetic sources of any type.

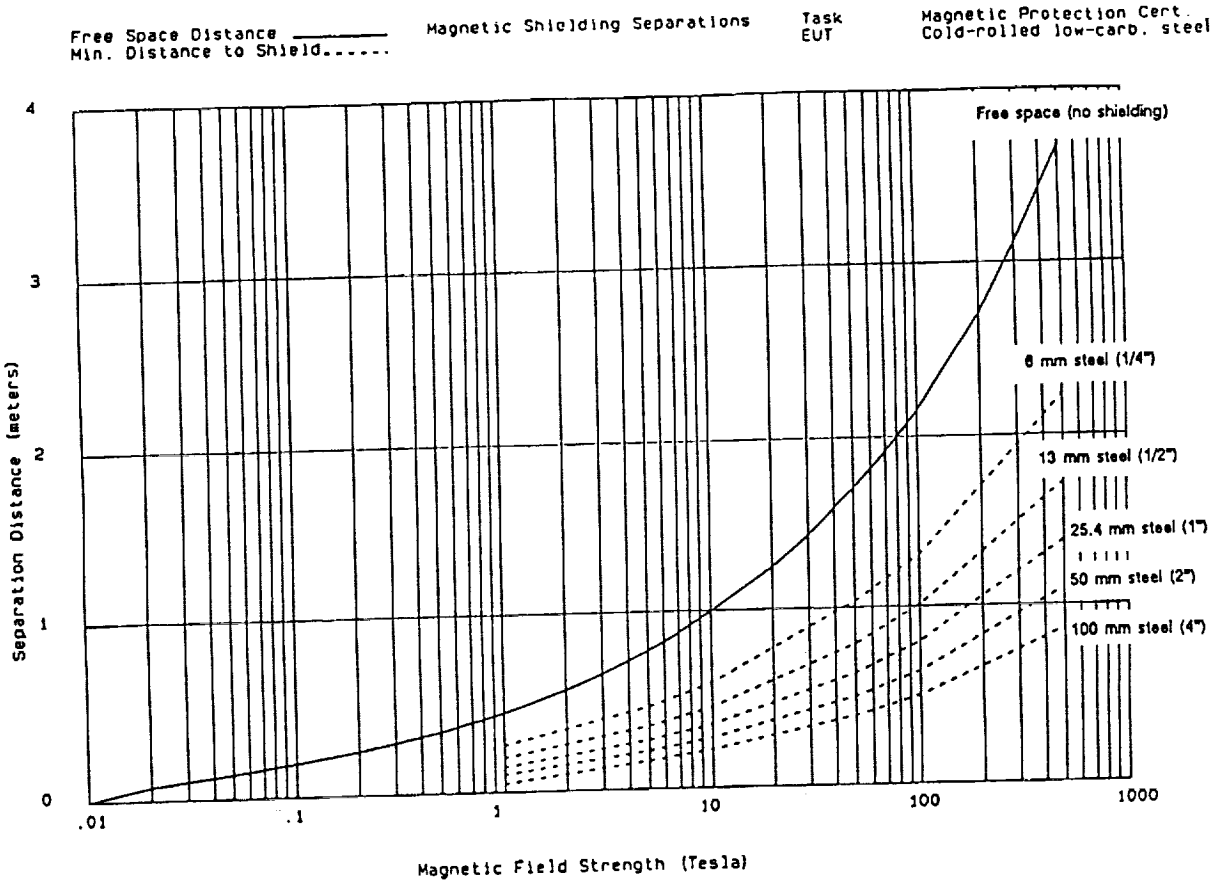



Figure 2. Magnetic Shielding Separation Distances

## Magnetic Protection Certification

The purpose of Publication 933 is to provide an objective standard for the magnetic protection of magnetic media storage facilities. The previous sections of this paper discussed the nature of magnetic fields and how magnetic media could be destroyed. We now review a sample magnetic protection certification, which shows how a facility is certified.

### Facility Layout and Magnetic Protection Zone Map

In this example facility certification, company X owns an MSA in a facility as shown in figure 3. The L-shaped MSA is shielded on two walls, and has good physical separation on the remaining walls. The roof of the facility is over 4 meters above the MSA. The company X controlled space is the area which the company owns and has alarmed on a 24 hour basis. The MSA has sufficient physical security to preclude unauthorized access, but desires a 50 Tesla magnetic protection level.

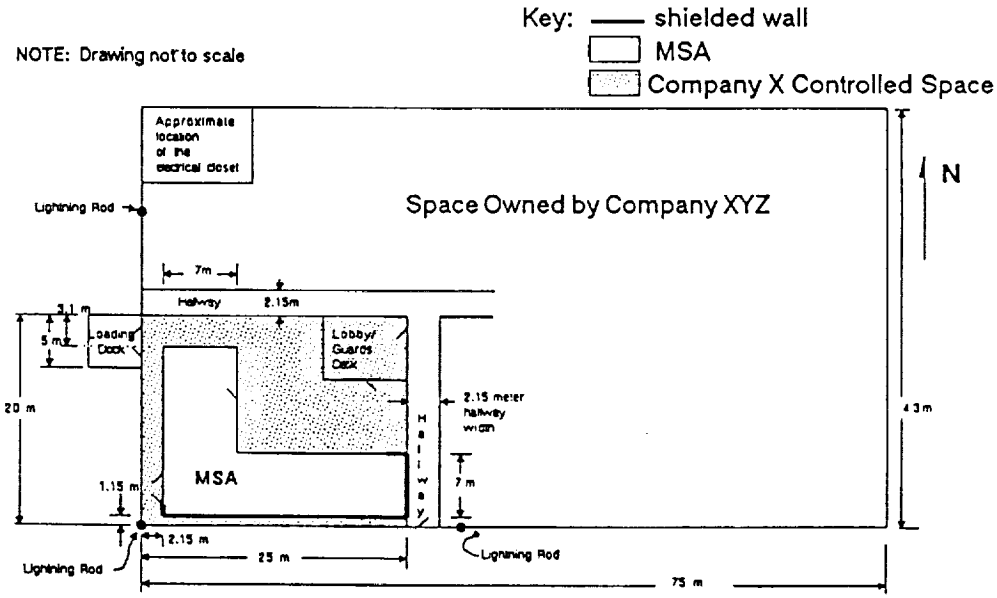



Figure 3. Facility Layout

After a site inspection and authorization to proceed, a magnetic protection zone map is created (in conjunction with a magnetic protection certification report). An example of a magnetic protection zone map is shown in figure 4. The magnetic separation zone map shows the areas within the MSA and their corresponding magnetic protection zones.

As figure 4 shows, some area on the east wall of the space is unsuitable for magnetic storage, even though it is shielded. On the south wall, the 6 mm (1/4") steel shield is performing its intended function of increasing the interior space within the MSA that is rated for 50 T protection. A vast majority of the northern part of the MSA is rated for 500 T protection.

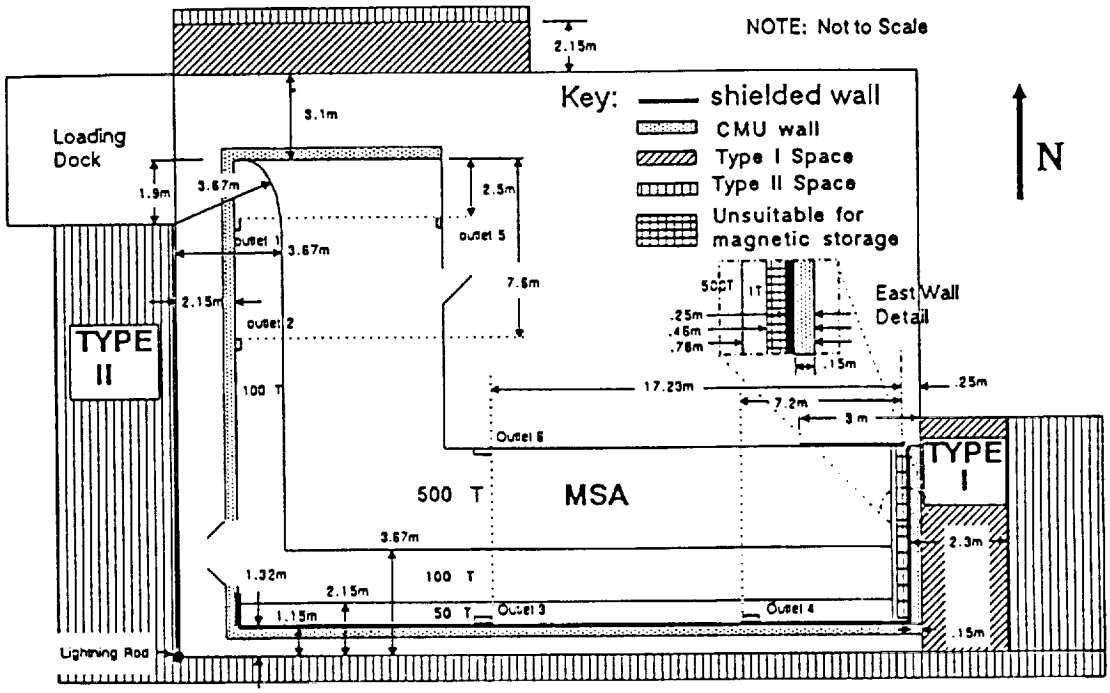



Figure 4. Magnetic Protection Zone Map

Details for the floor and ceiling, and outlets are presented in figure 5. Figure 5 shows the areas which are unsuitable for magnetic storage. The areas which are unsuitable for magnetic storage are based on the following guidelines:

The floor - the floor of the MSA is poured concrete with rebar reinforcement. If an externally applied magnetic field coupled to the rebar, the ends of the rebar (which are assumed to be randomly oriented in the slab) could be subjected to field strengths of up to 2 Tesla. The separation distance shown (.05 meters, 2 inches) is valid for rebar with diameters up to 13 mm (1/2").

The ceiling - the ceiling of the MSA is lighted with fluorescent lights which are interconnected with flexible conduit. If a light shorted instantaneously to structural steel, the current in the conductor could reach 10,000 amps (limited by the short circuit current of the lighting breaker). The separation distance shown is based on this current, and is present at any point on the ceiling since the cabling runs are flexible.

The outlets - the outlets in the MSA are fed from two breakers, each of which used a separation distance derived in a manner similar to that in the ceiling. The flexible conduit to the outlet can be oriented between the 18" on center studs in any random fashion, therefore, the field radiation is as shown. The separation distance is common for outlets fed from breakers with a similar short circuit rating.

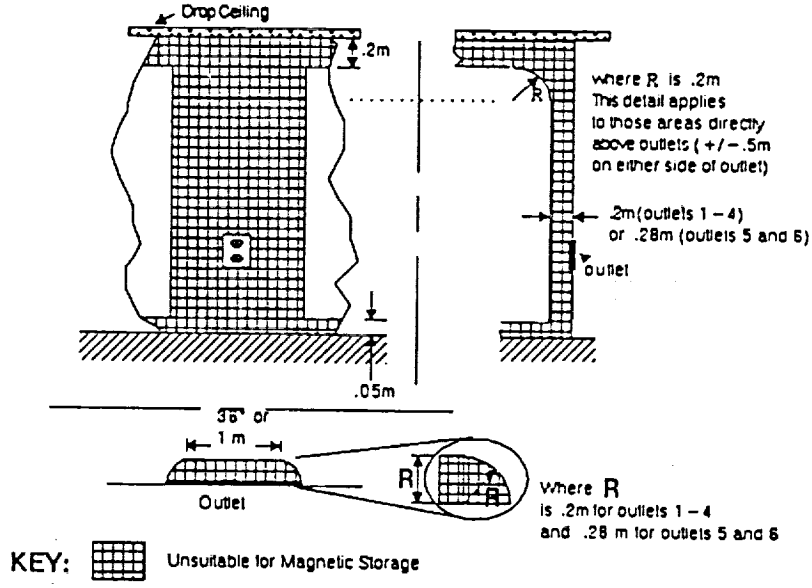



Figure 5. Magnetic Protection Certification Details

The magnetic protection certification report provides a means for MSA vendors and users to ensure magnetic protection. The design and solicitation documentation can now reference Publication 933 and a certain threat level (50 Tesla, for example), instead of subjective terms such as "magnetic protection" and "ferromagnetic shielding". The certification can be performed by a "qualified assessor", or be "self-certified" by a representative of the MSA.

While there may be questions about a specific facility, or the derivation of specific processes, Publication 933 presents a standard which is comprehensive, and can be tailored to a customers' specific application.

## Summary

The intent of this paper is to introduce a method for determining the magnetic threat to magnetic storage media. An additional intent has been to introduce Publication 933, an objective standard for magnetic protection of magnetic storage media. In this paper we have learned the following:

Magnets may destroy magnetic media via natural, unintentional or intentional means.

The highest magnetic fields which can currently be generated have a field strength of about 50 Tesla, and fields as high as 380 Tesla may be theoretically possible.

Magnetic media can be damaged at field strengths as low as 100 gauss (.01 tesla). The coercive force of the magnetic media determines how high the field must be to damage the media.

Simple formulas can be used to estimate the propagation of magnetic fields from simple wire and loop magnets.

Magnetic media can be damaged at distances up to 4 meters (about 13 feet).

Magnetic shielding can be used to reduce the magnetic field and subsequent separation distance between magnetic sources and magnetic media.

Publication 933 is a new standard that presents minimum spacing requirements between magnetic media and potential magnetic field sources. The procedure ensures both vendors and users of magnetic storage media that their data is safe from magnetic corruption.

## References


[1] Jewell, S., Publication 933-1, Objective Standards for Magnetic Shielding of Magnetic Media Storage Facilities, (Chantilly, VA: Advanced Measurement Systems, Inc., 1993)

[2] Geller, S.B., The effects of magnetic fields on magnetic storage media used in computers, NBS Technical Note 735, COM-72-50873, U.S. Department of Commerce, 1973, 1-5.

[3] Geller, S.B., Care and handling of computer magnetic storage media, NBS Special Pub. 500-101, PB83-237271, U.S. Dept. of Commerce, 1983, 37-51.

[4] Nakagawa, Y., Kido, G., Miura, S., Hoshi, A., Watanabe, K. and Muto Y., A design of 50 T hybrid magnet for quasi-stationary operation, digest of papers, abstracted from Winter Annual Meeting of the American Society of Mechanical Engineers, San Francisco, AC, Dec. 10-15, 1989 633-638. from Superconductivity advances and applications, 1989, (New York, NY: ASME, 1989).

[5] Weggel, R.J., Leupold, M.J., Williams, J.E.C., and Iwasa, Y., 45 T, Steady State, digest of papers, abstracted from Winter Annual Meeting of the Americans Society of Mechanical Engineers, San Francisco, AC, Dec. 10-15, 1989 627-632. from Superconductivity advances and applications, 1989, (New York, NY: ASME, 1989).

[6] Palmer, D.N., Forward for Superconductivity advances and applications, 1989, (New York, NY: ASME, 1989).

[7] Bhushan, B., Tribology and mechanics of magnetic storage devices (New York: Springer-Verlag, 1990).

[8] Vinstra, L., Manager of the Data Diskette Lab, 3M corporation, St. Paul, Minnesota: March 10, 1993. Interview.

[9] Goldfarb, R., Technical Staff, Magnetic Field Measurements Department, NIST, Colorado Springs, Colorado: March 8, 1993. Interview.

[10] Hoagland, A., Professor of Electrical Engineering and Computer Science, Santa Clara University, Director of Institute for Information Storage Technology, Santa Clara, California: March 9,1993. Interview.

[11] Plonus, M.A., Applied electromagnetics (New York: McGraw Hill, Inc., 1978).

[12] Metals Handbook, 8th Edition, Volume 1, p. 792, Prepared under the direction of the ASM Handbook Committee (Metals Park, Ohio: American Society for Metals, 1976).

[13] Assorted Product literature and Shop Talk abstracts, (Bensenville, Illinois: Magnetic Shield Division of Perfection Mica Company, 1986).

N94-33795

# High Performance Quarter-Inch Cartridge Tape Systems

**Ted Schwarz**

3M Company
3M Center 236-GN-06
St. Paul, MN 55144-1000
Phone: 612-733-3367
Fax: 612-737-2801

## The Industry

More Quarter-inch Cartridge (QIC) tape drives are sold than all other data tape recorders combined.[1] By the end of 1993, the installed base will be over ten million units. In terms of unit volume, QIC tape drives are second only to consumer video systems, although that separation is measured by orders of magnitude. A comparison of estimated volumes of data systems shipped in 1993 is shown in Figure 1. QIC's unique self-contained tape transport and guidance system within the cartridge allows for low-cost, highly reliable, small form factor transports which meet the needs of small to medium computer systems such as the IBM AS/400, workstations and personal computers. QIC systems provide solutions for computer systems which require the back-up of a few hundred megabytes of data for less than $200 (single user) in the 3.5 inch form factor Mini-Data Cartridge to 5 gigabytes of data for under $2000 (less than $1000 in large OEM quantities) in the 5.25 inch form factor Standard Data Cartridge. The proliferation of 100+ MByte disk drives in PCs is driving a large increase in the penetration of Mini-cartridge tape drives as the back-up storage device of choice.

In the workstation, LAN server, and small-to-medium computer system, however, 4 mm and 8 mm helical scan devices have made significant inroads on the larger capacity Standard Data Cartridge. In the past few years, their sales growth has been flat and is predicted to decline slightly in the next few years. Helical scan systems have been able to provide large multi-gigabyte storage capability in the $2000-$5000 range, suitable for unattended back-up for small-to-large systems. Two years ago helical scan solutions were able to offer 10X storage capacity for a 2X-4X price over the top-of-the-line Data Cartridge systems. This capacity difference is shrinking as new technologies are introduced into Data Cartridge systems. Today, the capacity ratio is 2:1 with the introduction of the five gigabyte (5 GB) systems by several QIC drive manufacturers. With the next generation, the Standard Data Cartridge will exceed its 8 mm rival in capacity while maintaining its lower cost advamtage.

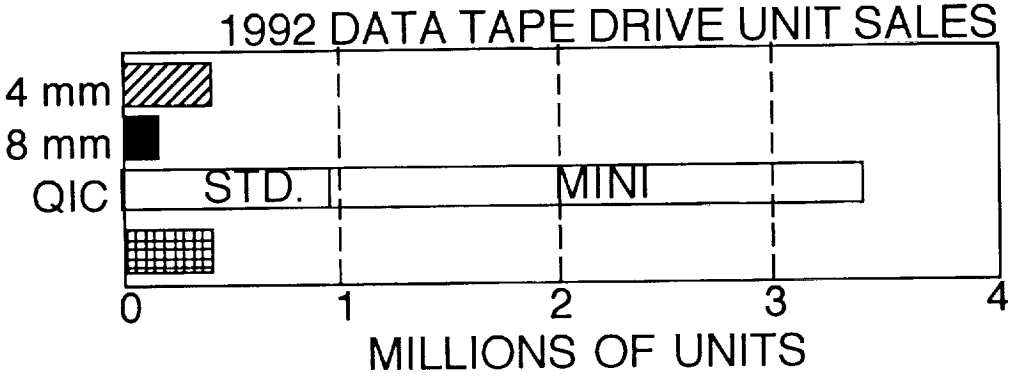



Figure 1. 1992 Tape Drive Sales

## Quest for 1 Gbit/in$^2$

The migration path of the Data Cartridge Technology, DCT, to higher areal density and, hence, higher capacity is shown in Figure 2. Data Cartridge Technology shares the National Industry Consortium's (NSIC) goal for tape of achieving an areal density of one $10^9$ bits per square inch (1 Gb/in$^2$). The diagonals correspond to lines of constant areal density. For reference, the published path of 4 mm helical scan . The 10 Gb/in$^2$ NSIC target for rigid disks along with its present achievements is also shown. The usable range along the 1 GB/in$^2$ is bounded by the maximum demonstrated bit density recording, approximately 500,000 transitions per inch (500 kfci or 20000 fcmm), and the maximum demonstrated rigid disk track density of almost 10,000 tracks/in (400 trks/mm). Given the issues of tolerances and media substrate instability, it is expected that for tape, the track density will be in the range of 3000 to 4000 tracks per inch. To achieve 1 Gb/in$^2$ the corresponding bit density will need to be 250 kbpi to 333 kbpi. Current QIC technology under development and due as products in 1994 is the 13GB generation. Its predecessors, the 1.35GB generation and its derivatives are already in production. Future technology generations code named "Hawk", "Condor", and "Eagle" represent areal densities of approximately .150, .500, and 1.000 Gbit/in$^2$, respectively.

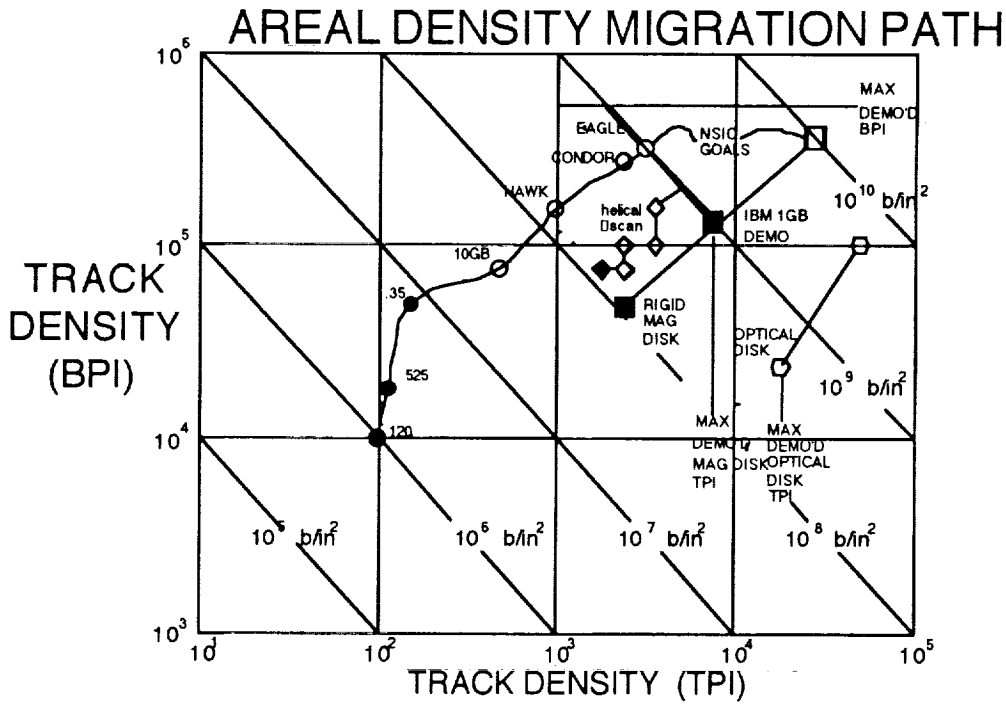



Figure 2. QIC Areal Density Migration Path

The areal density curve in Figure 2 is nearly vertical from the QIC 150 drives to the 1.35GB generation. In that period, nearly all the increase in areal density is the result of increases in bit density. Bit density increases have been made possible, primarily, by increasing the coercivity of the media and changing the data modulation code to provide more data bits per flux transition in the media. The horizontal break in the curve between the 1.35GB and the 13GB technology families has come about by the introduction of track following servo systems and thinfilm magnetoresistive, MR, heads. The plan is then to progress with increases in both track and bit densities towards the "Eagle" generation goals.

The quest for areal density is reflected in Figure 2. The real measure, however, is the increase in capacity. This is more difficult to predict because of the increasing length (area) of the tape. The media substrate thickness, which plays a dominant role in the tape length, has been

plunging. It has decreased from 25 microns in the IBM 3480 tape to about 6 microns (.00025 inch) for the current generation. New substrate materials such as Polybenzoxazole, PBO, may make it possible to achieve thicknesses of 1-2 microns. The length of tape has increased from 760 feet in the 1.35 GB system to 1200 feet for the 13GB family. Future capabilities for the 5.25" Standard Data Cartridge and the 3.5" Mini-data Cartridge are illustrated in Figure 3 as bands reflecting potential tape lengths. Because of this variation and other modifications to the areal density, the advanced technology generations have been identified by code names rather than capacity or areal density designations. The Mini-data Cartridge holds about 30 percent of the tape length of the Standard Data Cartridge.

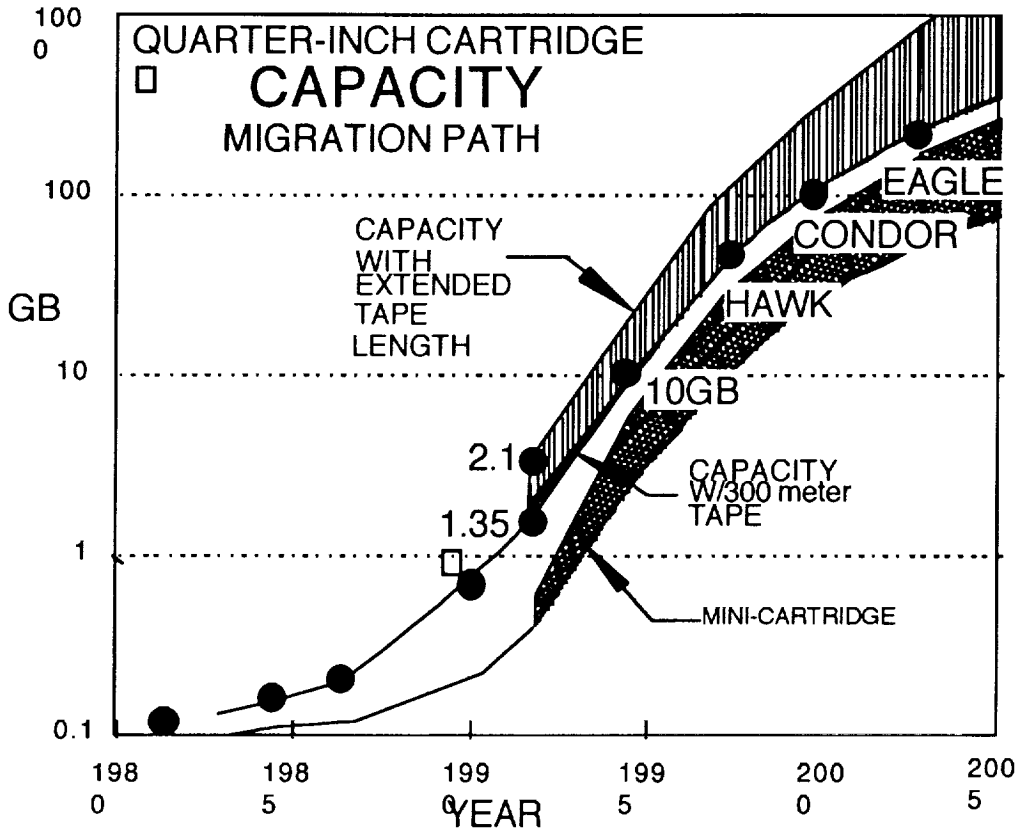



Figure 3. Data Cartridge Capacity Growth.

## New Technologies

New technologies in the following areas will fuel the capacity growth in the current development and future Data Cartridges:

**Heads**
**Servo**
**Encoding**
**Media**

It is planned to introduce one or two major technology advances in each new generation. In this way, the absorption of technology is reasonable and yet, the growth in areal density and capacity is strong and sustainable. The specific key technologies for each generation are shown in Table 1.

Table 1. Key Generational Technologies

| Technology Family | New Technologies |
|---|---|
| 1.35GB | 900 Oe Co-$Fe_2O_3$ Media<br>1,7 RLL Encoding |
| 13GB | MR Thinfilm Heads<br>Track following servos |
| Hawk | Barium Ferrite (BaFe) Media<br>Partial Reponse Encoding |
| Condor | ME Media/Advanced BaFe Media<br>Ultra-Thin substrate |
| Eagle | Perpendicular Recording Media<br>Perpendicular Recording/playback Heads |

From this list the critical technologies that differentiate linear scan Data Cartridges from helical scan systems are MR heads and track following servos. The other technologies are equally applicable to both philosophies. The DCT track following servo allows it to approach helical scan in track density by reducing the difference from an almost 20:1 factor down to about a 1.3:1 factor. The MR head allows the Data Cart to achieve very high data rates inexpensively through multiple data channels and operate over wide speed ranges, unlike inductive heads, since its output signal voltage is speed insensitive. Because the MR head is DC powered and is built as a planar array, it does not lend itself to multi-element azimuth recording or operation through transformer coupled systems. It is unlikely that MR heads will be incorporated into helical scan systems.

## Head Technology

The thinfilm MR head being introduced in the 13GB generation represents a quantum jump in technology and capability. It allows the system to be approached differently than with conventional inductive heads. This head senses dF/dx versus dF/dt. Hence, its output is the same, barring any separation losses, from 0.1 inch/sec to 2000+ inches/sec. It is also a much more sensitive transducer, producing approximately 1000 microvolts, uV, per mil, .001 inch, of head width versus 100 uV/mil for an inductive head when the latter is optimized for the head-to-media speed at which it is intended to operate. An inductive head output, $e_o$, is compromised by its maximum operating frequency, $f_{max}$, which is limited the number of turns, N.

$$f_{max} \sim 1/N^2$$

$$e_o \sim M_r \cdot t \cdot N \cdot W \cdot v$$

| | |
|---|---|
| N | is the number of turns |
| $M_r$ | Remanent Magnetization of the Media |
| t | effective recording depth |
| v | is the head-to-media velocity |
| W | is the read track width |

Running at a sub-optimal speed, i.e. multi-speed drives, results in an even greater difference between the output of inductive and MR heads. The much greater output of the MR head results in a system performance that is dominated by the magnetic medium's signal-to-noise-ratio,

SNR, rather than the system noise. Hence, the raw signal output from the medium is less important than its intrinsic SNR. The configuration of the 13GB head is illustrated in Figure 4.

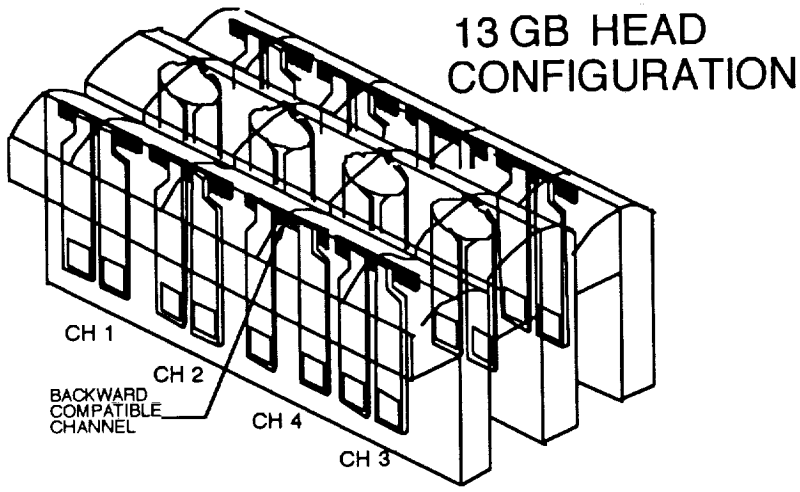



Figure 4. 13GB Head Configuration

The head contains three "13GB" data channels, capable of read-while-write for tape motion in either direction, arranged in an asymmetrical 2:1 spacing separation. Two channels are used for data and the third is used to read servo data. The functions of the channels are alternated between servo and data as shown in Figure 5 to minimize both the number of elements and the overhead for the servo band. Only 14 percent of the tape surface is devoted to servo information in this configuration. Two servo bands are required, as long term tape substrate instability of the current PET media and current head manufacture/assembly tolerances would cause excessive misregistration of the outermost head because of the additional separation if only one servo band were utilized.

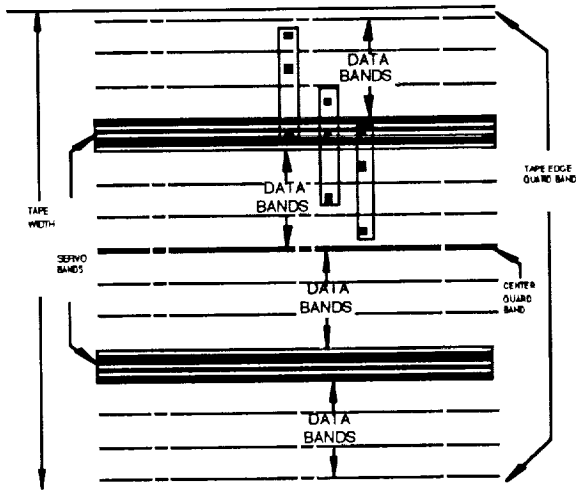



Figure 5. Servo Band and Head Configuration

In the wider separation region between the two 13GB channels, a 1.35GB compatible channel is inserted. This provides backward compatibility for several generations of systems, allowing old tapes as far back as QIC-24 (60 MBytes) to be readable in this system.

## Track-Following Servo

As previously illustrated in Figure 2, the track density for Data Cartridges has been limited to about 150 tracks per inch by tape tracking and distortion in the cartridge. The non-repeatability in track has both a quasi-static component associated with the direction of the tape motion and a dynamic component resulting from the composite of rotating parts. The 13GB system incorporates a full-time tracking system. The tape is divided into two servo bands to reduce the effect of tape distortion due to time variations in temperature, humidity, tension, and creep. The servo pattern and resultant "on track" signal are illustrated in Figure 6.

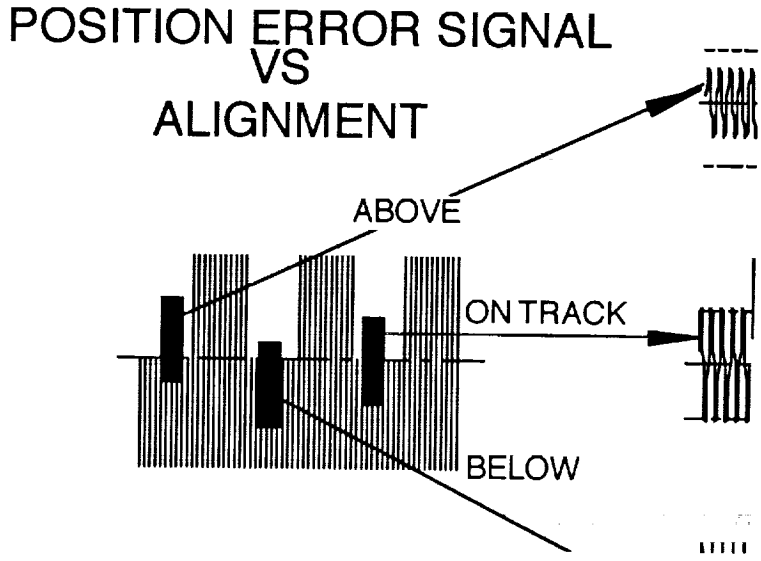



Figure 6. Servo Pattern and "On-Track" Position Error Signal

In the "on-track" condition, the "B" signal is 50 percent of the reference "A" signal. The system has allowed the tracking error to be reduced from .001 inch ( 1000 microinches) to 30 microinches. The number of data tracks has been increased from 30 in the 1.35GB to 144 in the 13GB drive. The latter is the equivalent of about 750 tracks per inch (tpi). This track density breakthrough is illustrated in Figure 7. Azimuth recording used in helical scan still allows an extra 30 to 40 percent in track density but, is inefficient in other ways such that its future areal density advantage is only 15 to 25 percent.

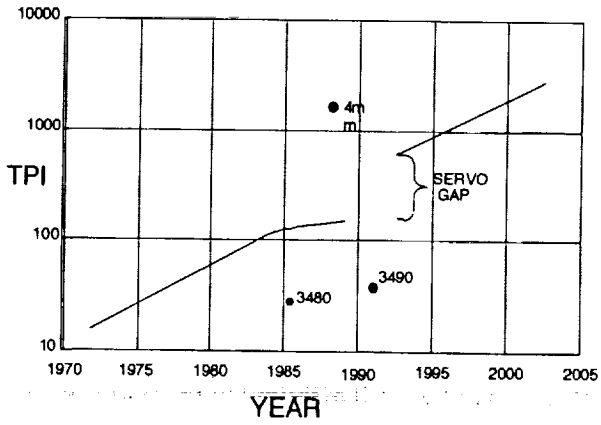



Figure 7. Data Cartridge Track Density Migration Path

## Data Encoding

The path for data encoding for Data Cartridges has generally followed that of rigid disk drives. Historically, the codes employed were MFM, 4/5 GCR, 1,7 RLL. In the near future, a form of Partial Response may be employed. The 2,7 RLL disk drive code was not incorporated because of its much smaller clocking window which is incompatible with the poorer time base instability (jitter) of tape. These codes and their data bit to transition on the tape efficiency are shown in table 2.

Table 2. Data Encoding

| Type | Efficiency |
|---|---|
| MFM | 0.5 |
| 4/5 GCR | 0.8 |
| 1,7 RLL | 1.33 |
| 2,7 RLL | 1.5 |
| Partial Reponse | 1.5 - 2.0 |

Partial Response encoding is the first non-peak detecting system. It operates by sampling the waveform and determining from a sequence of sampled amplitudes what the signal was. It has several variants (PR4, EPR4, and $E^2PR4$) .These are differentiated by the number of times the waveform is sampled. The most effective configuration is highly dependent on the system characteristics. For a given error rate, the efficiency ranges from 0.9 to 2.0 Disk drives utilizing thinfilm inductive heads have not achieved efficiencies much beyond 1.6-1.7 because of anomalies in their isolated pulse reponse characteristics. The MR head, however, gives the smoother, more classical response, of a conventional ferrite head and efficiencies approaching 2.0 are anticipated. This technology change will have the least impact on increasing the capacity.

## Media

For all forms of magnetic recording, the magnetic medium is the key ingredient in propelling increases in areal density. The most important characteristics of tape medium are its magnetic properties and its surface finish. The surface roughness, which is semi-independent of the magnetic characteristics, impacts head-to-media separation, hence, output and density response. Its other key characteristic is "defects" which cause drop-outs in the signal resulting in errors.

13GB generation of Data Cartridges employ the standard SVHS 900 Oe cobalt doped iron oxide (Co-$Fe_2O_3$) particle which was also employed for the 1.35GB generation. It was selected at that time because of its superior environmental stability, availability, and noise characteristics relative to metal particles (MP). Since then, improved passivation has greatly reduced the corrosion concern for MP. Future generations will require new media. It is anticipated that either a super-fine MP or BaFe particle will be used. While metal-evaporated, ME, media such as NiCo or CoCr hold out the promise of the higher performance required for future systems, the issues of sufficient durability and chemical stability to meet long term data storage requirements have yet to be demonstratedfor these media. Progress has been reported and, perhaps, ME media will be ready by the very late 1990's.

MR heads with their much greater sensitivity have altered the prioritization of characteristics for magnetic media. Where the system noise dominates the SNR, such as with inductive heads with a few hundred microvolts of signal or less, the output from the tape is paramount. Among the particulate media, MP has the highest magnetic moment, hence, the highest output. ME films offer even higher outputs. In Data Cartridges, where MR heads are utilized, however, the medium's SNR dominates the overall system SNR. Here particle size, packing density, and uniformity of dispersion are the most important characteristics. Based on the data cell size for

the projected track and transition densities for current and future generations and assuming an effective recording depth of one fourth (1/4) the minimum spacing between transitions, the calculated SNR is shown for each of the media in Figure 8. The high performance "HDTV" MP particle is assumed to be .01 micron (um) in diameter by 0.1 um long with an .005 um thick passivation layer. In all cases it is assumed that a 50 percent by volume fractional packing density is used since this is dependent on the process capability of the manufacturer. This is optimistic for Co-$Fe_2O_3$ and MP which are typically closer to 40 percent and pessimistic for BaFe which is closer to 60 percent. Both NiCo and CoCr ME media are also compared. Here the SNR is derived based on the surface granularization. The NiCo is assumed to have magnetic domains with an approximated dimension of 300 angstroms (1.2 microinch) and the CoCr to have hexagonal crystals of 150 angstroms in diameter.

The SNR in Figure 8 is calculated. Practical experience indicates that the actual SNR will be about 6 dB lower. This can be attributed to head-to-media separation, surface roughness, non-uniform distribution, etc. Figure 8 does give the relative SNR, however. The required SNR to for a raw error rate needed for a $10^{-15}$ corrected error rate (error bursts between bits) in conjunction with other system parameters is approximately 27 dB. 900 Oe Co-$Fe_2O_3$ is adequate for both the 1.35GB and the 13GB generations. A new medium is required for the Hawk generation. This is likely to be BaFe whose calculated SNR is 3-6 dB greater than that projected for MP. The additional advantage resulting from dual layer coating may allow BaFe to suffice for both the Hawk and Condor generations.

Use of particulate media is very desirable in that it provides substantial cost savings over sputtered or evaporated metal film media. ME media is expensive because of both the cost of the equipment and the overall throughput of that equipment. Vacuum deposition equipment is both expensive initially, requires a significant amount of downtime for maintenance, and has coating speed rates in the hundreds of feet per minute instead of thousands of feet per minute for particulate coaters.

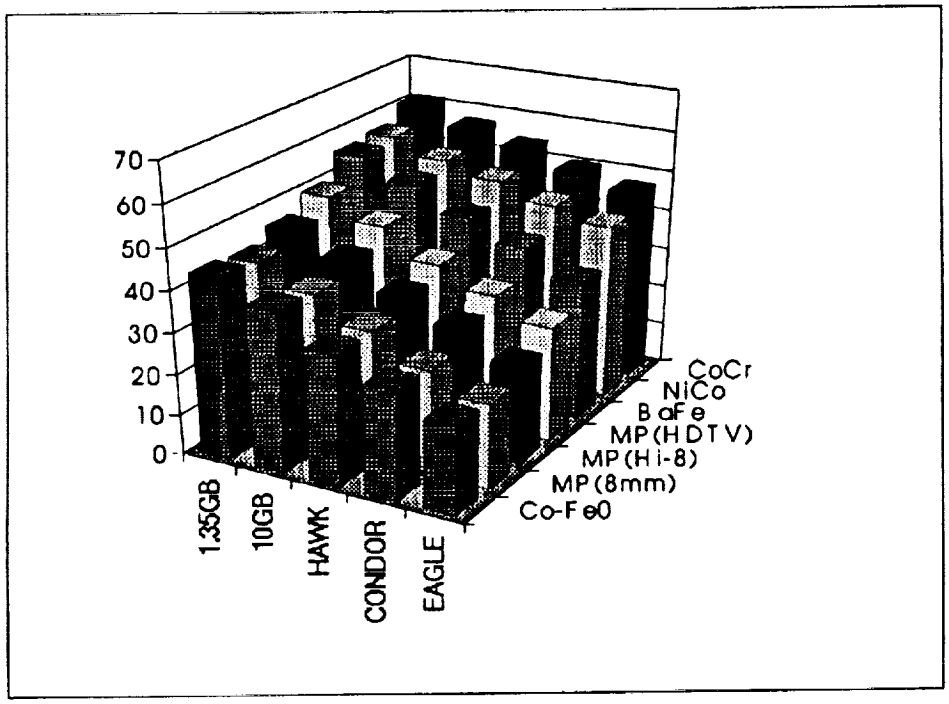



Figure 8. SNR for Various Media

The transition to ME media would appear necessary for the Condor technology generation (approximately 500 Mb/in$^2$). However, Fuji has demonstrated dual layer media in which a smooth thin 0.2-0.5 um layer of MP is coated over a thicker layer of lower coercivity media or a non-magnetic layer with special characteristics such as titanium oxide. Dual coating has the potential of achieving a magnetic layer as thin as 0.1 um. The output of this medium at very high densities approaches within a few dB of that of the ME media with the density response and overwrite characteristics of thinfilm media. Yet, it retains the producibility, lower cost, and tribological characteristics of the current particulate media systems. This thin layer characteristic may reduce one of the more difficult characteristics of BaFe, namely the difficulty in overwriting this very high coercivity (>1500 Oe) material. Hence, the migration to the more expensive ME media may be delayed until after the turn of the century. The construction of such a medium for Data Cartridges is shown in Figure 9.

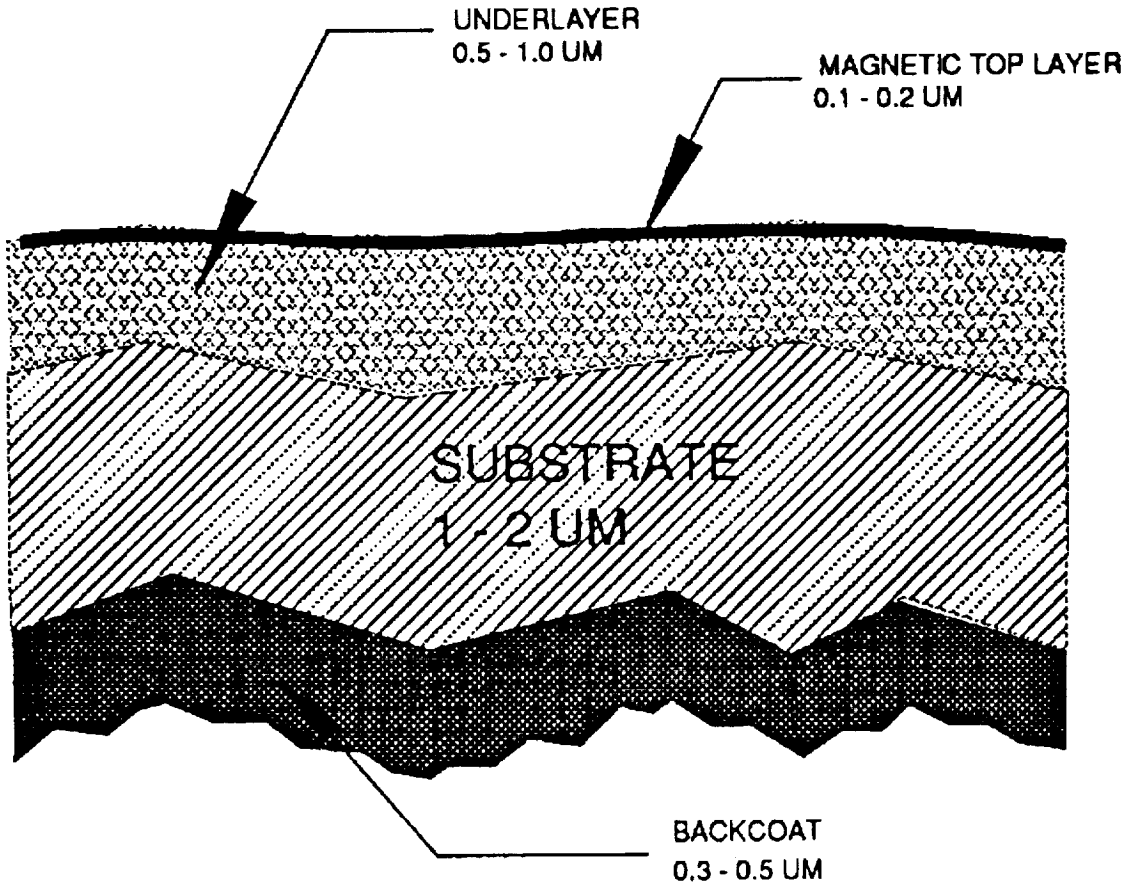



Figure 9. Potential Dual Layer Particulate System

A straightforward method of increasing the capacity is to make the tape longer. This can be achieved by making the magnetic layer, substrate, and backcoat layers thinner. The front and back layers are already quite thin and further reductions will provide little opportunity to lengthen the tape significantly. The current PET and PEN substrates are reaching their mechanical limits as the substrate thickness approaches 4 um. Materials such as the polyaramides or Polybenzoxazole, PBO, offer promise of higher modulus and, in the case of PBO, other improved characteristics. A comparison is shown in Table 3.[2] PBO or a material with similar coefficients of creep and expansion will be necessary for Data Cartridge Technology to reach 3000-4000 tpi.

**Table 3. Substrate Characteristics**

| Characteristic | units | PEN | PET | ARAMID | PBO |
|---|---|---|---|---|---|
| Density | $g/cm^3$ | 1.395 | 1.355 | 1.420 | 1.54 |
| Melting Temperature | $^oC$ | 263 | 272 | 350 | None |
| Young's Modulus | $kg/mm^2$ | 500-850 | 650-1400 | 1000-2000 | 4922 |
| Tensile Strength | $kg/mm^2$ | 25 | 30 | 50 | 56-63 |
| Tensile Elongation | % | 150 | 95 | 60 | 1-2 |
| Long Term Heat-Pool Temp | $^oC$ | 120 | 155 | 180 | >300 |
| Heat Shrinkage (200° x 5 min) | % | 5-10 | 1.5 | 0.1 | <0.1 |
| Coeff of Thermal Expansion | $10^{-6}/^oC$ | 15 | 13 | 15 | -7 |
| Coeff of Hygroscopic Exp. | $10^{-5}$/%RH | 10 | 10 | 18 | <1 |
| Moisture Absorption | in/in/% RH | 0.4 | 0.4 | 1.5 | <1 |

The mechanical strength of PBO may allow the thickness of the substrates to approach 1.0 um. Figure 9 illustrates the effect of total tape caliper on length. It is expected that the magnetic and back coatings will add from 0.6 to 1.5 um to the substrate thickness for total caliper.

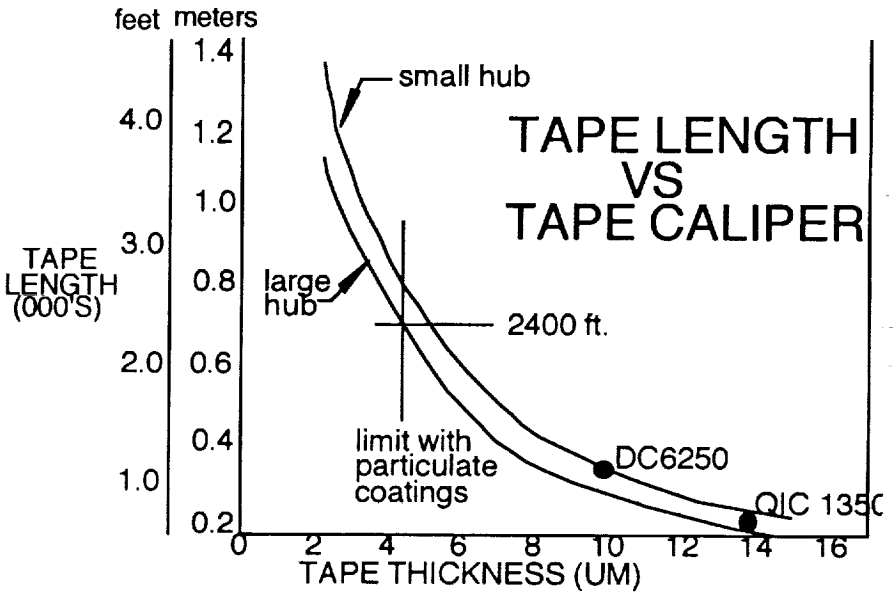



Figure 9. Tape Length versus Caliper for 5.25" Data Cartridge

Other factors such as tape handling, tension variation, achievable substrate roughness, etc. also figure into determining the maximum length achievable.

## The Marketplace

In the past, Data Cartridge Technology has carved out a large segment in the PC to medium system marketplaces for back-up and data exchange where the emphasis has been focused on low initial cost and compactness. While retaining these characteristics, the enhanced capacity and potential for high transfer rate opens up new markets. The "Tertiary Storage " market

graph by Ann Drapeau from last year's NASA Conference on Mass Storage[3] has been updated with helical scan and Data Cartridge next generation capacity, due in late 1993 or early 1994, in Figure 10.

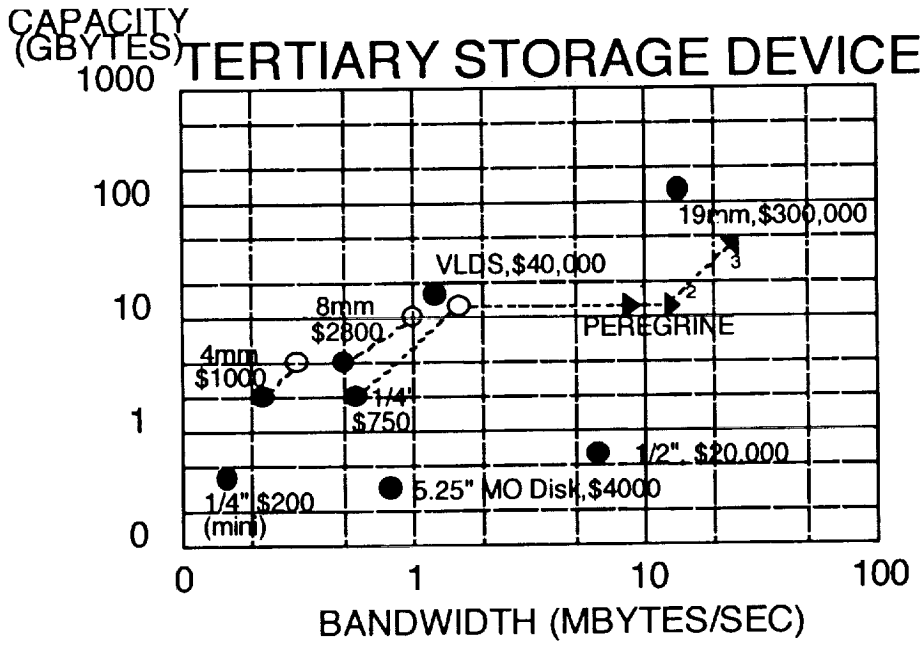



Figure 10. Capacity and Transfer Rates for Assorted Removable Technologies

It should be emphasized that the growth paths for certain technologies are not shown as they have not been announced by their developers. Undoubtedly, they too will be growing in capacity and data rate. The latter, though, is likely to be much slower for helical technologies. Their improvement rates will be primarily paced by the increases in bit density which is similar for all the above technologies with the exception of optical.

## High Performance Systems

"Linear Scan" technology such as Data Cartridge Technology offers the opportunity to incorporate many parallel channels at minimum additional drive cost. A 12-channel system compatible with the 13GB cartridge has been proposed. This class of drives is called Peregrine. There are two versions. The first is simply a 12-channel version of the 13GB drive. It increases the maximum data rate from 1.54 MBytes/sec to 9.3 MBytes/sec. A later second version, called Peregrine II, uses an 180 ips 13GB cartridge to achieve a sustained data rate of over 100 Mbits/sec or more precisely, 14.3 MBytes/sec. These are shown as the breakout points in Figure 10. The multi-channel approach is also applicable to the Mini-cartridge. Because of the size of the cartridge, the tape speed is limited to 120 ips. Hence, the maximum data rate for the Mini-cartridge is 9.3 MBytes/sec. The twelve-channel configuration was selected because of its compatibility with the 13GB systems and a feature geometry that is relatively economical to produce. Utilizing only half the width of the tape because of the tape related track distortion, TRTD, the elements are set on a 204 um (.008 inch) pitch. Use of advanced substrate materials and tighter tolerances in the head fabrication may allow future generations to easily expand to 24 channels. It is not anticipated that the channel pitch will be reduced much below 200 um. Multiple gaps might increase this number to 48 channels but, the structure and the termination of the elements would be very complex. One possible configuration of the 12 channel head is shown in Figure 11. Contrasted to the 13GB head in Figure 4, it appears very complex, and even more so when compared to a single-channel head. However, a good analogy

can be drawn from the individual transistor and integrated circuits. The fabrication processes and dimensions between IC's and thinfilm heads is quite similar.

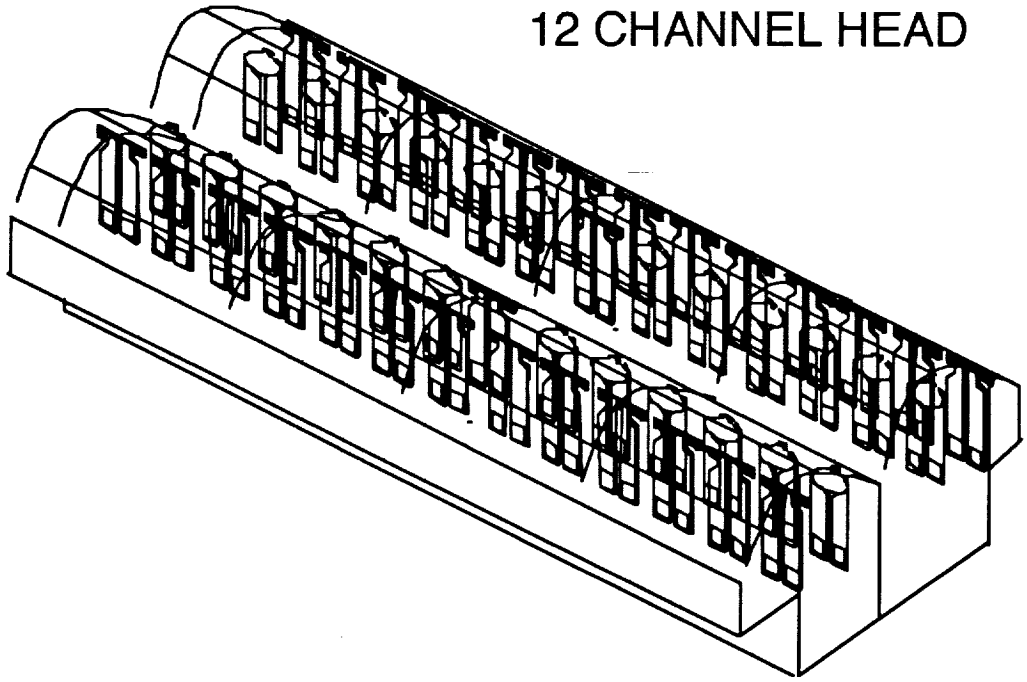



Figure 11. Twelve-Channel Peregrine Head

The Peregrine head is symmetrically divided into two groups of six channels centered about a servo channel. The outermost data channel is no farther away from the tracking servo channel in the Peregrine configuration than it is in the 13GB configuration except for the small delta displacement of one quarter of a 13GB servo band. Hence, the issues of TRTD on the head-to-track misregistration is little worse than in the 13GB. The layout of the Peregrine head to the 13GB head is illustrated in Figure 12.

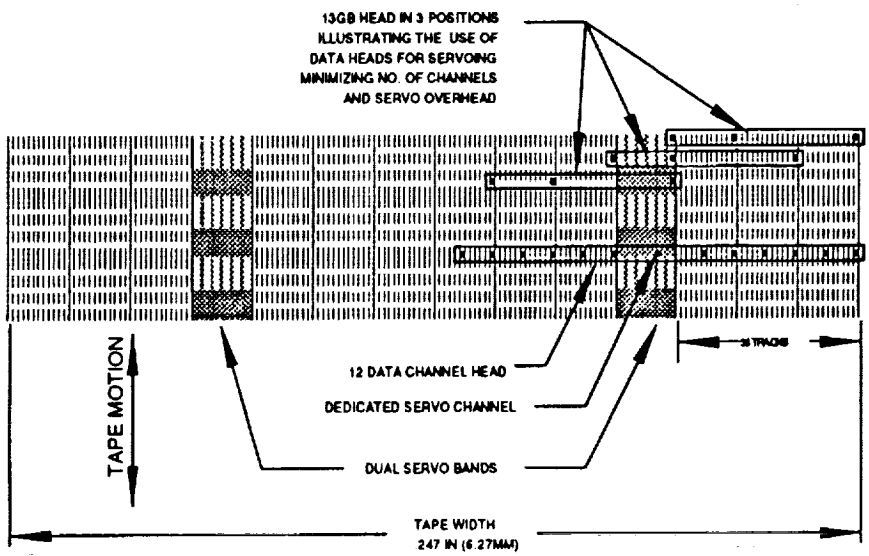



Figure 12. Tape, Peregrine, and 13GB Head Layout

The bit and track configurations for the next generation "Hawk" are fairly well defined at this time. The Peregrine version of that technology will provide over 250 Mbits/sec and 56 Gbytes of data in the maximum configuration. This is illustrated in Figure 10 as the P3 extension. The principal characteristics for the first three members of the Peregrine family are listed in Table 4.

**Table 4. Peregrine System Characteristics**

| Parameter | Peregrine 1 | Peregrine 2 | Peregrine 3 |
|---|---|---|---|
| Technology | 13GB | 13GB | Hawk |
| Capacity (GBytes) | 13.5 | 13.5 | 56 |
| Data Rate (MB/s) | 9.3 | 14.3 | 31 |
| Data Rate (Mb/s) | 76 | 114 | 253 |
| No. of Channels | 12 | 12 | 12 |
| Bit Density (kbpi) | 67.733 | 150 | 150 |
| Data Tracks | 144 | 144 | 216 |
| Media | Co-$Fe_2O_3$ | Co-$Fe_2O_3$ | BaFe |
| Media Speed (ips) | 120 | 180 | 180 |
| Tape Length (ft) | 1200 | 1200 | 1500 |

The progression of Data Cartridge Technology to higher bit density, more channels, and a practical upper limit of tape speed to the region of 200+ ips, should make Data Cartridge systems with 1-2 Gbit/sec data rates readily achievable in the future.

## Summary

Within the established low cost structure of Data Cartridge drive technology, it is possible to achieve nearly 1 terrabyte ($10^{12}$) of data capacity and more than 1 Gbit/sec (>100 Mbytes/sec) transfer rates. The desirability to place this capability within a single cartridge will be determined by the market. The 3.5" or smaller form factor may suffice to serve both the current Data Cartridge market and a high performance segment. In any case, Data Cartridge Technology provides a strong sustainable technology growth path into the 21st century.


1. International Data Corporation
2. National Media Lab Newsletter
3. NASA Conference Publication 3198, Vol. II pg. 203

A proprietary process has been developed to "stabilize" the media against corrosion and phase separation observed previously in this alloy system under environmental testing at elevated temperatures and humidities. Although having the effect of decreasing the sensitivity of the media somewhat, excellent stability results have been demonstrated for this process, while still retaining sensitivity adequate for current applications. The subbing layer sets a smooth, hardened surface for the optical media and is particularly important in reducing optical defects in the PET surface. The backcoat enhances web or tape handling properties and can significantly reduce wear on the front surface of the tape.

## Vacuum Depostion

A laboratory size sputtering machine (13" web width) is used with 3 separate minichambers located around the coating drum. Each minichamber is operated independently in either the metal or reactive sputtering mode for discrete and sequential layer depositions. Optical properties of the deposited layers are measured in-situ and used to computer control the operation. Optical monitors provide frequent wavelength scanning from 360-2200nm of both reflection and transmission, and the computer provides down-web data logging of all process parameters. Typical optical properties of the media at two wavelengths are shown in Table 1.

TABLE 1: Optical reflection and transmission of media at 670/830nm on PET during sputter deposition.

| LAYER TYPE | THICK | R (670/830) | T (830) |
|---|---|---|---|
| 1. Optical Metal Only | | 75/75% | 3% |
| 2. With Oxygen Dopant | ~ 35nM | 55/55% | 10% |
| 3. With Activation Layer | ~ 50nM | 43/50% | 8% |
| 4. With Abrasion Layer | ~ 65nM | 37/45% | 7% |

## APEX WRITE/READ Test Results

An APEX OHMT-300 instrument is used to measure recording characteristic with an 830nm laser. For purposes of evaluation, modulation depth is defined as (R0-R1)/(R0-R2), where R0, R1 and R2 are reflectivities of unwritten media, written data bits and bare PET (typically 6%), respectively. Full modulation (i.e., 1.0) is defined as the data bit reflection equal to base PET reflection. For the current product, modulation depth as a function of writing energy for various pulse widths is shown in Fig.2 for two different lasers in the system, 10mW and 50mW, and a disk rotation rate of 5rps (about 850mm/s velocity). Write sensitivity of the media increases substantially with decreasing pulse width. At the shortest pulse width tested (50ns), full modulation is reached near 0.5nJ. The offset of the power curves at 250, 300 and 350ns to lower energies (about 0.35nJ) for the 50mW laser vs. the 10mW laser is attributed to the spacial differences in the energy output of the two lasers incident on the media surface and/or a possible difference in calibration.

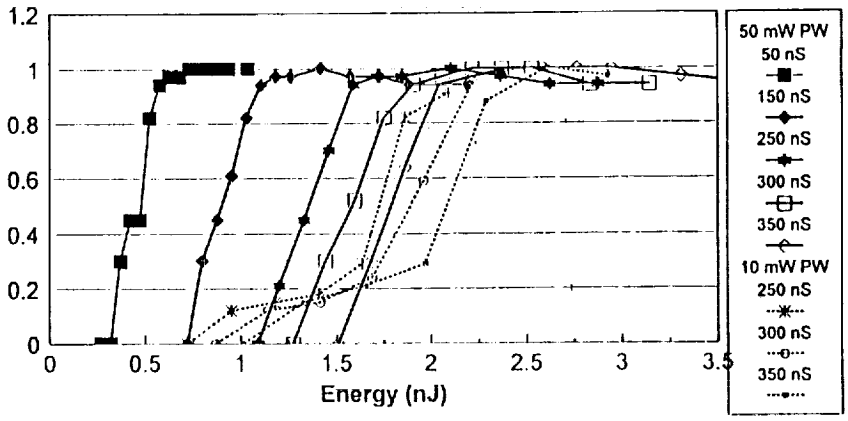



FIG.2: Modulation Depth vs. Writing Energy at Various Pulse Widths

The APEX test results (50mW laser) of writing energy as a function of pulse width at various fixed modulation depths are shown in Fig. 3. Over the range tested, the writing energy (P) at a fixed modulation depth decreases linearly with decreasing pulse width (t) and can be described by the function, P = A + Bt. At 50% modulationdepth, the intercept (A), at t = 0 for an infinitely narrow pulse width, is 0.2nJ, and the slope (B) is 0.0045J/s. The increase in writing energy for a fixed modulation depth is attributed to time dependent power dissipation by thermal conduction in the media. This characteristic implies a significant energy advantage (as well as speed) for shorter pulse writing.

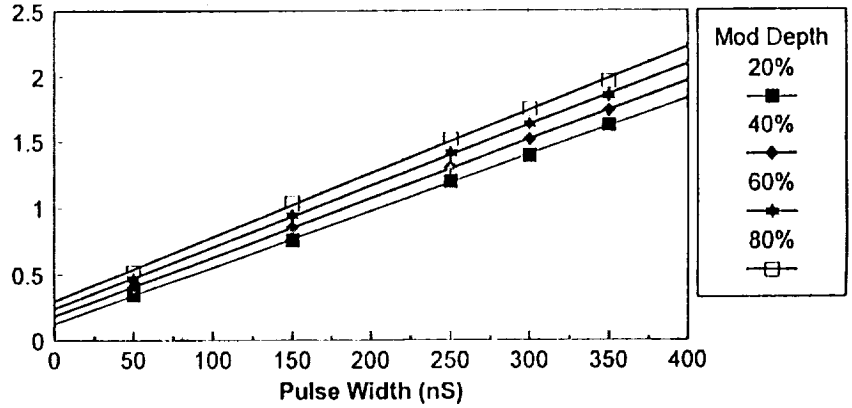



FIG. 3: Writing Energy vs. Pulse Width at Various Modulation Depths

## Data Bit Morphology

CREO written data bits of approximately 1 micron diameter are shown on a section of tape in the photomicrograph of Fig.4. Formation of such data bits are shown in more detail by atomic force microscopy in the micrograph of Fig.5. With the exaggerated scaling in the z-axis, the bits appear as mounds with a somewhat faceted center pit. The cross-section of the lower bit in Fig.5 shows the pit with a flat bottom, which may extend down to the substrate. The small rectangular features in the surface of the media are believed to be metallic crystallites.

FIG. 4: Photomicrograph of CREO Written Tape

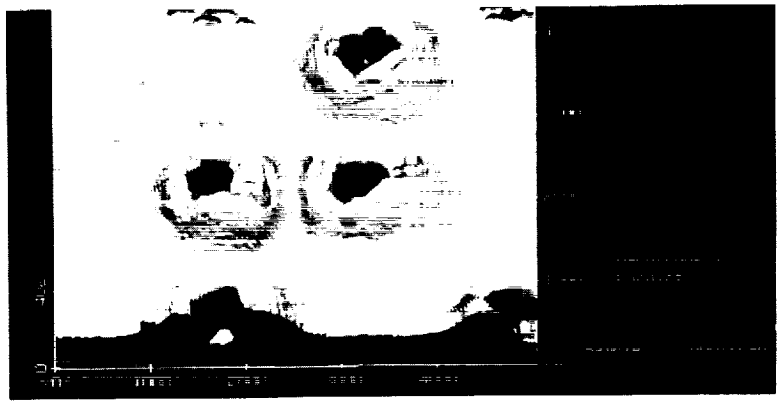


FIG. 5: AFM Micrograph of CREO Written Data Bit Scale = nm

## Wear Test Results

As a guide to developing wear resistance, a simple abrasion test with a hard eraser was employed to determine the number of strokes required to produce an observable removal of media from the substrate. The results in Table 2 indicate a major improvement in wear resistance for the combination of subbing and abrasion resist layers. The contribution of the subbing layer in enhancing abrasion resistance is attributed to a firmer base for the media provided by its hard-coat properties and stronger adhesion to the optical metal. The abrasion resist layer is then most effective under these conditions. There appears to be an optimal thickness of the abrasion resist layer, i.e., abrasion resistance improves significantly with increasing thickness, however, media sensitivity is somewhat diminished by thicker layers. Wear tests conducted at CREO on the optical tape recorder show very promising results. Initial (at zero search cycles) raw and corrected bit error rates are below 5x10exp-5 and 10exp-22, respectively. At search speeds of 5m/s, more than 5,000 searches are readily obtained on tapes without backcoats before corrected bit error rates of 10exp- 12 are exceeded. With the current top side structure, the major contribution to wear appears to come from the back or slip-coated side of the tape. Very promising results have been obtained from samples with anti-abrasion back side coatings.

TABLE 2: Simple Erasure Abrasion Resistance of Media Showing Effects of Subbing and Abrasion Resist Layers

| PET SUBSTRATE | COATING TYPE | # OF STROKES |
|---|---|---|
| WITHOUT SUBBING LAYER | ACTIVATION LAYER ONLY | 7 |
| | WITH ABRASION RESIST | 8 |
| WITH SUBBING LAYER | ACTIVATION LAYER ONLY | 12 |
| | WITH ABRASION RESIST | 30 |

## Environmental Test Results

Accelerated aging tests were conducted by suspending unprotected tape strips in a temperature/humidity chamber at 70°C and 95%RH. Typical optical property changes with time for stabilized media are shown in Fig.6. Only slight changes in transmission and reflection were observed after 384 hours. At each test interval, the characteristics of modulation depth vs. writing energy were measured on the APEX tester (with the 10mW laser and 250ns pulse width), as shown in Fig.7. No noticeable changes in writing characteristics were observed. All of these results are in sharp contrast to earlier experience with these media before the stabilization process was developed; i.e., the media would be totally degraded after weathering for 24 hours. Other tests are in progress to determine potential for 100 year archival life.

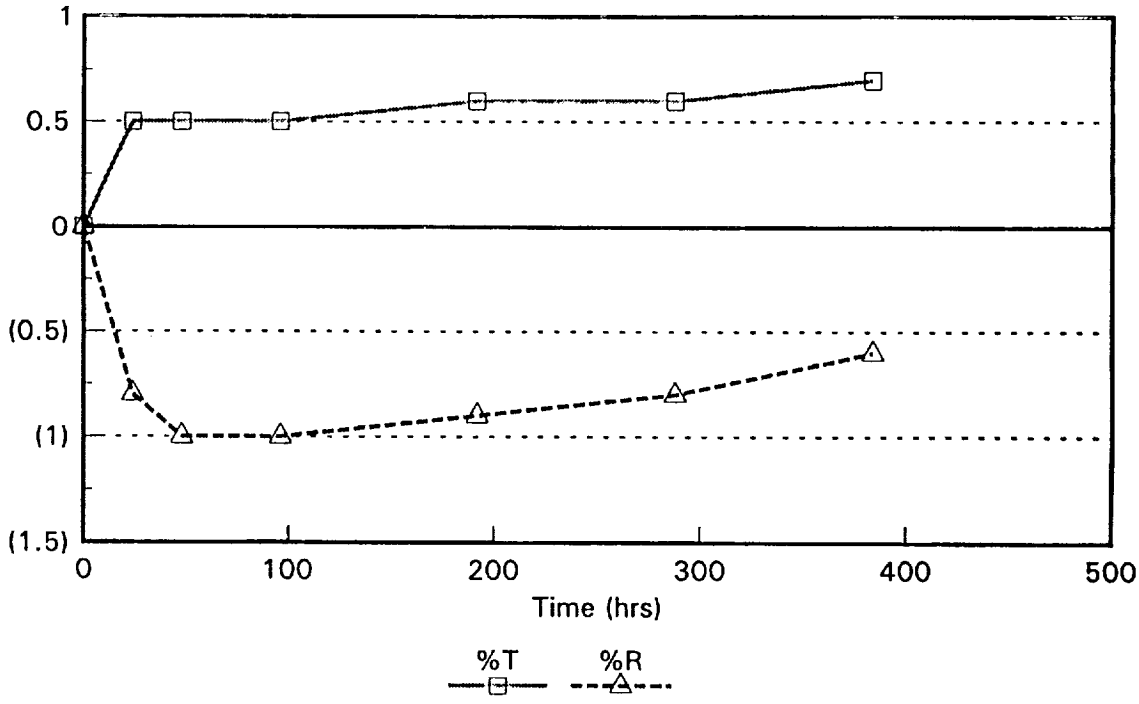



FIG.6: Change In Transmission & Reflection vs. Time at 70°C and 95%RH Change in Percent @ 850nm

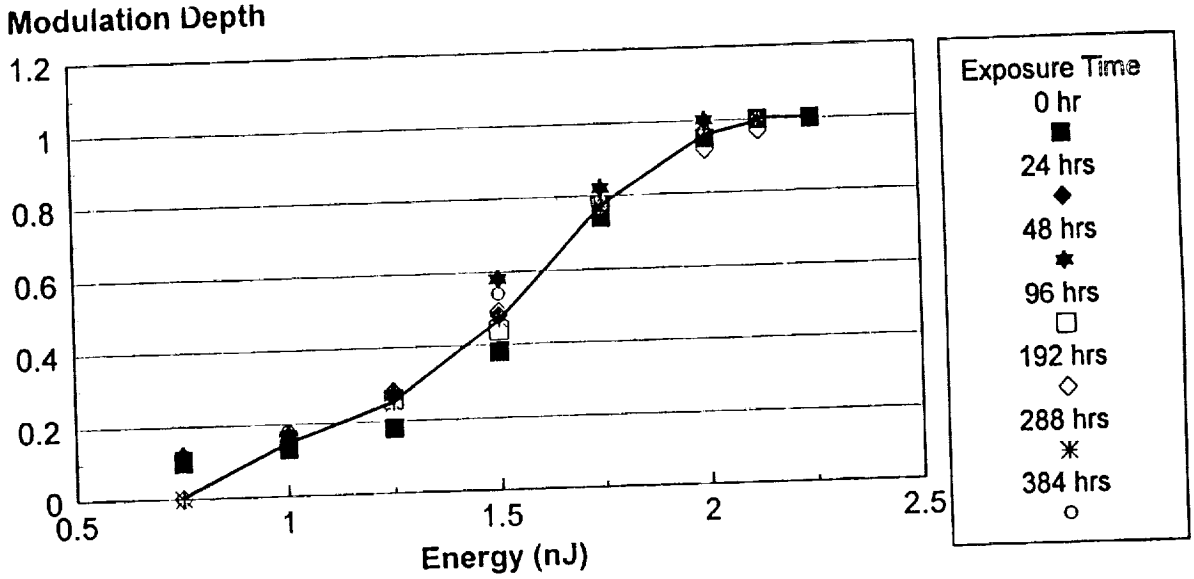



FIG. 7: Modulation Depth vs. Writing Energy at Various Times in T/H Chamber at 70°C and 95% RH

## Conclusion

An optical recording media has been developed for permanent digital storage application in tape formats. Useful characteristics of write sensitivity, durability and environmental stability have been demonstrated. The sputter coating process can be scaled to large, wide-web roll coaters for future high-volume production. Although tuning of overcoats will be required for blue lasers, the media is well suited for such developments in future higher speed, higher density applications.

## Acknowledgements

For invaluable contributions and assistance, the authors thank D. Perettie, A. Strandjord, S. Webb and D. Hawn of Dow, D. Smythies of CREO and L. Peck and D. Willis of Southwall Technologies.

## References

1. A.Strandjord, S.Webb, D.Beaman and S.Carroll, "Thin Film Coatings for Flexible Optical Data Storage", Proc. SPIE Optical Thin Films III: New Developments, San Diego, Vol. 1323, 1990.

N94-33797

# Certification of *ICI 1012* Optical Data Storage Tape

**J. M. Howell**

Senior Engineer, ICI Imagedata
Brantham, Manningtree, Essex CO11 1NL, England.
Phone 044-206-392424 Ext 6432, Fax 044-206-391472

## Introduction

ICI has developed a unique and novel method of certifying a Terabyte optical tape. The tape quality is guaranteed as a statistical upper limit on the probability of uncorrectable errors. This is called the Corrected Byte Error Rate or CBER, and is defined below.

We developed this probabilistic method because of two reasons why error rate cannot be measured directly. Firstly, written data is indelible, so one cannot employ write/read tests such as used for magnetic tape. Secondly, the anticipated error rates need impractically large samples to measure accurately (Smythies and Woodley [1]). For example, a rate of 1E-12 implies only one byte in error per tape.

The archivability of *ICI 1012* Data Storage Tape in general is well characterised and understood; see for example Ruddick [2]. Nevertheless, customers expect performance guarantees to be supported by test results on individual tapes. In particular, they need assurance that data is retrievable after decades in archive. This paper describes the mathematical basis, measurement apparatus and applicability of the certification method.

## Tape format and error correction

See figure 1. Data is stored as records written transversely across the tape. Each record on the tape is built up from 1024 codewords. Each codeword contains 64 bytes of user data and 16 error correction code (ECC) bytes. The ECC algorithm is a Reed-Solomon code which can completely correct up to 8 defective bytes within a codeword. If there were 9 bytes or more, then the codeword would be flagged as uncorrectable and the regenerated data would very likely contain errors.

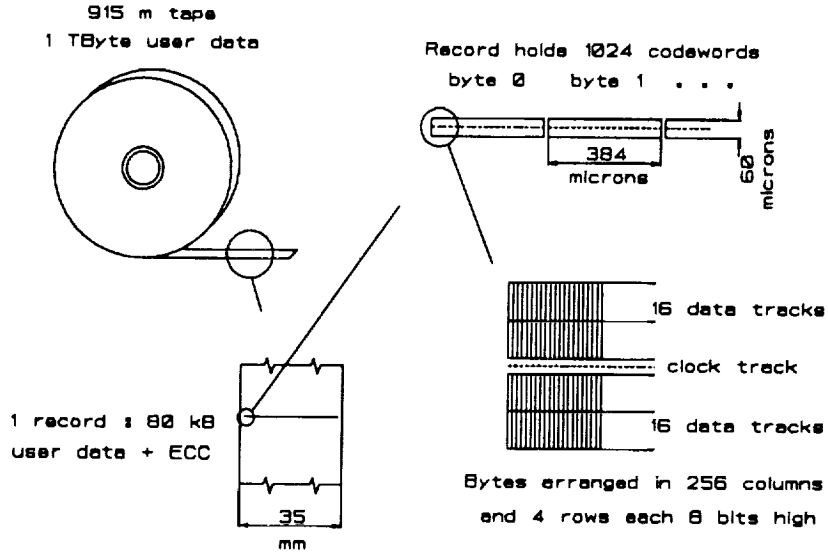



Figure 1. Data format on *ICI 1012* Data Storage Tape

The CBER is the probability that a byte regenerated from the raw data in any codeword is corrupt. Because of the ECC, and the way the data is packed in each record, the number of defective bits in each byte does not affect the ECC; it is thus appropriate to measure errors in bytes rather than bits. Statistically, the number of errors in the data increases with the raw byte error rate, or BER.

The ECC works on expanded codewords 255 bytes long consisting of the 80 user and ECC bytes plus 175 bytes filled with zeroes. The algorithm generates 8 correction bytes and 8 correction addresses in the expanded codeword. With more than 8 corrupt bytes in the original codeword, the corrections and addresses are wrong, and there is a high probability that some of the 175 padding bytes will be toggled non-zero. When this happens ECC breakdown is detected and the recovered data is issued as it was read.

## Origin of Errors

Optical tape is written by short bursts of intense light from a laser which reduces the dye-polymer recording layer thickness, making it appear dark. The data is read by another lower power laser which discriminates the dark and light regions. Errors arise if either a written region appears bright or an unwritten region appears dark. There are many potential sources of errors, but for media certification we are only interested in those from physical anomalies in the recording layer.

Bright spots occur when there is a break in the dye-polymer coating so that the underlying aluminium alloy reflector is exposed. Alternatively, the coating is so thick that the write laser cannot 'punch through' to sufficient depth. Dark spots are generally due to scratches and debris which scatter the read laser light. They can also be seen if the dye-polymer layer is thinner than normal, so that it is comparable in thickness to a written area.

## Calculating the CBER from measurable properties

For reasons given above, one must use statistical probability to compute CBER. This section starts with the general case and develops a practical tool for deriving the CBER from the background defects and observable point defects.

1. General case.

The probability of a given codeword being corrupt is the sum of the probabilities of that codeword containing y=9, 10, .. 80 corrupt bytes. The CBER for this codeword is the sum of the products of

i) the probability $p_{y,80}$ of a codeword having y corrupt bytes before correction, and
ii) the fraction y/80 of corrupt bytes in the codeword (since correction is inhibited).

$$CBER = \sum_{y=9}^{80} p_{y,80} \cdot \frac{y}{80}$$

2. Effect of background defects.

If the defective bytes are completely random, then the binomial distribution can be used to compute the $p_{y,80}$ from the probability p of any individual byte in the codeword being corrupt. The fraction p is the byte error rate, or BER.

$$p_{k,n} = \frac{n!}{k!.(n-k)!} . p^k . (1-p)^{n-k}$$

It was shown in Howell [3] that the CBER calculated in this way is the most pessimistic value; if the defective bytes are not random, i.e. they are clustered in the record, then the CBER will be less. It represents an upper bound on CBER when the raw BER in a record measured by the drive is used directly in the binomial distribution.

In practice the CBER estimated from data from large scale experimental trials [3] is substantially less than the upper bound. A lot of perfectly serviceable material would fail if this simple test was applied indiscriminately. The reason is clear when the tape surface is examined more closely. A relatively small peak in BER would be amplified by the non-linear binomial expression so it dominates the CBER result. However we find that these peaks are not showers of defects spread over the tape width. They are compact clusters caused by well-defined circular or 'point' defects. Such defects corrupt nearly every byte within their boundary, but do not affect those outside.

A practical certification method must take the morphology of these point defects into account. As the tape is not preformatted it is impossible to know exactly which codewords will be affected. We can still obtain an important statistical CBER estimate from the size alone. This will now be derived.

3. Effect of individual 'point' defects.

Consider codewords within a single record containing a defect which is exactly one byte block (384 microns) wide. This defect is superimposed on the random background errors. Every codeword in the record will have at least one corrupt byte from the defect. The probability of 9 corrupt bytes in total is then the binomial probability of 8 out of the remaining 79. For a defect which is 2 byte blocks wide, we must find the binomial probability of 7 out of 78, and so forth. In general, consider a 'point' defect which is (w+w') blocks wide, where w' is the fraction and w is the integer part. The CBER for a record containing this defect is given by the following formula:

$$\begin{aligned} CBER &= \sum_{y=9}^{80} \left( w' . p_{y-w-1,80-w-1} + (1-w') . p_{y-w,80-w} \right) . \frac{y}{80} \\ &= w' . \sum_{y=9}^{80} p_{y-w-1,80-w-1} . \frac{y}{80} + (1-w') \sum_{y=9}^{80} p_{y-w,80-w} . \frac{y}{80} \\ &= w' . B_{w+1} + (1-w') . B_w \end{aligned}$$

where $B_W$ is the cumulative binomial expansion from background errors for the CBER of a codeword which has w bytes within a 'point' defect:

$$B_w = \sum_{y=9}^{80} p_{y-w,80-w} \frac{y}{80}$$

4. Effect of multiple 'point' defects.

If several 'point' defects occur in the same record, their combined effect is calculated from the probability of the defects coinciding in any codeword. Consider two defects each less than one block wide of sizes w' and v' blocks. Assume they are randomly placed in the record. The total CBER will be the sum of the contributions from codewords with none, one or two bytes affected, as follows:

$$\begin{aligned} CBER = &(1-w').(1-v').B_0 \\ &+ ((1-w').v' + w'.(1-v')).B_1 \\ &+ (v'.w').B_2 \end{aligned}$$

The calculation can be developed iteratively to include any number of such defects, of any width (w+w') blocks. Let $d_i^n$ be the probability of a codeword having exactly i corrupt bytes from n 'point' defects. Imagine a codeword which is initially free of point defects. Here $d_0^0 = 1$ and $d_1^0 \ldots d_{80}^0 = 0$. We include the effect of each defect which might affect the record in turn. For the k'th defect the vector $\mathbf{d}^k$ is generated from its predecessor $\mathbf{d}^{k-1}$ and defect width (w+w') thus:

$$d_i^k = 0, \text{ for } i < w$$

$$d_i^k = (1 - w').d_{i-w}^{k-1}, \text{ for } i = w$$

$$d_i^k = (1 - w').d_{i-w}^{k-1} + w'.d_{i-w-1}^{k-1}, \text{ for } i > w$$

The expression for the CBER in a record with n point defects is then the sum of the contributions from the codewords with 0, 1, 2, ... 80 bytes corrupted by 'point' defects.

$$\text{CBER} = \sum_{i=0}^{80} d_i.B_i$$

5. Practical implementation.

The vector $\mathbf{d} = (d_0, d_1, \ldots d_{80})$ is a complete description of the effect of 'point' defects in a record on the CBER for that record. The vector $\mathbf{B} = (B_0, B_1, \ldots B_{80})$ is only a function of the background BER. One can therefore compute **d** as an arithmetic mean over all records in a region of tape (say 1 metre) and use it in the above expression for an average CBER. If the BER is invariant along the tape the vector **B** will be constant for all regions. Thus one has a computationally efficient method of predicting the CBER profile for a tape from the sequence of **d** vectors and the background BER.

**Measuring Point Errors**

'Point' defects are regions of the tape which introduce substantial clusters of errors, and are due to irregularities in the active layer. A fully automatic system is required which will measure these in line for every tape pancake (figure 2).

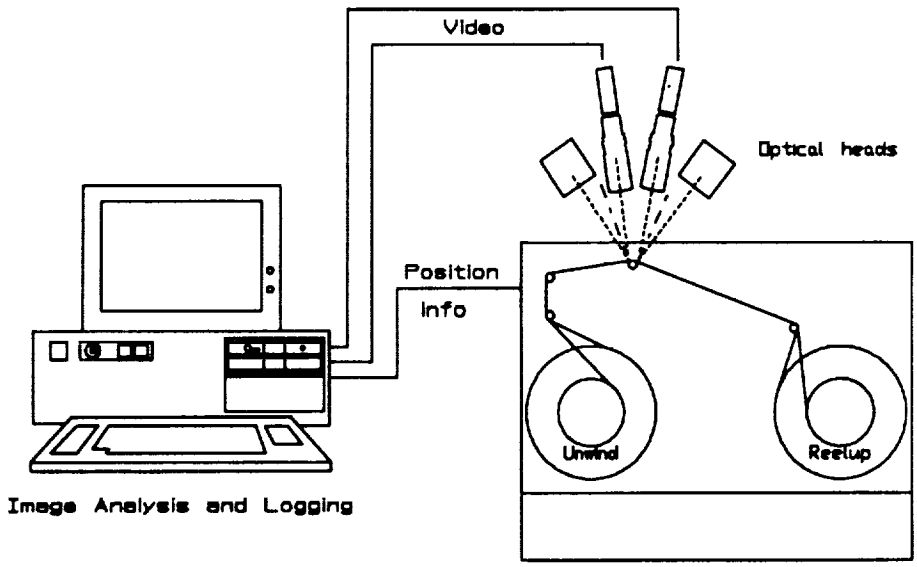



Figure 2. Automatic inspection for *ICI 1012* optical tape

The system uses a pair of line scan cameras which observe the tape in the near infra-red as it passes at constant speed. Defects show up in both cameras, but the wavelengths have been carefully chosen to measure the reflectivity in the unwritten and written states. The latter measurement is clearly an indirect one. The optics incorporate many innovative features to reduce sensitivity to other disturbances, such as tape movement. This means that the system can be made very sensitive to variations in coating which are known to affect read/write performance.

The cameras will resolve to 20 microns at the tape surface. This means that any defect can be measured to 5% of a byte block width. The software analyses the image in real time, including correlation of the two camera images to avoid double-counting of defects. The codeword defect vector **d** is built up for each transverse scan which approximates to a record width. The cumulative codeword defect vector may then be used to compute the CBER directly for each metre run of tape, as described above. This is also a traceable record for quality assurance.

The software further analyses the image to classify the physical form of the defects. Thus, scratches, thin and thick dye-polymer, debris and exposed reflector are individually logged and displayed. This map then becomes a permanent record of every tape which is produced, so that in the unlikely event of problems the product is fully supportable.

## Conclusions

The method described is a non destructive, statistically valid prediction of the CBER for certification of individual tapes. It is flexible in three respects.

Firstly, it is not a simple pass or fail quality check. Different applications will have different tolerances to uncorrectable errors. One inspection provides all the information needed to determine the suitability of a tape. It is then possible to grade tapes if the need arises.

Secondly, the CBER is known along the whole length of the tape pancake. This means that cutting positions can be optimised to minimise waste when removing any substandard regions of the pancake.

Finally, the defect vector is a compact record which is independent of subsequent changes in service[1]. The CBER can be recalculated easily and accurately for any background BER. This could be BER measured by the tape drive after writing. Equally it could be BER at some point in the future as estimated by models from accelerated ageing experiments.

## References


1. Smythies, DC and Woodley, BR. Bit Rate qualification Criteria for ICI Optical Media, Creo Products Inc. internal report, May 1991.

2. Ruddick, AJ. ICI Optical Data Storage Tape - an archival mass storage medium, Goddard Conf. on Mass Storage Systems and Technologies, NASA Goddard Space Flight Centre, September 1992.

3. Howell, JM. The Effect of Defect Distribution on Error Correction in 1012 Tape, ICI Imagedata internal report, September 1992.


[1]This is because in practice the point defects themselves are invariant. Any changes during the life of the tape occur inside the measured defect boundary which the theory already assumes is unusable.

# The IEEE Mass Storage System Reference Model: Update on Version 5

Bob Coyne

IBM Federal Systems Company
3700 Bay Area Blvd., Houston, TX 77058

## Overview

Work in Progress
Key Concepts
The Big Picture(s)
Fundamental Abstractions
The Mover
The Physical Volume Repository
The Physical Volume Library
The Storage Server
The Bitfile Server
Environmental Services

## Work in Progress

Latest document integration and editing
by Rich Garrison of Martin Marietta
Version 5 unapproved draft 1.2,
Oct 18, 1993
Current draft available on request after
October meeting
Collecting public comments, send to
ieee+mss@larc.nasa.gov and/or
coyne@houvmscc.vnet.ibm.com

### Key Concepts and Features

Abstract model for open storage systems
interconnection (OSSI)
Modularity
Transparency
Separation of policy and mechanism
Logical separation of control and data flows
Third-party transfers
Layered object naming via name services
Enable automated storage hierarchy
management
No scalability limits
OSI system management model
General security model

## Fundamental Abstractions

**Sets/Containment/Groups**
used for managing sets of storage system objects
many types of homogeneous and heterogeneous sets

**Stores**
structured address space with operations
fundamental object manipulated by the Model
contain all storage data
includes physical stores, virtual stores, virtual volumes

**Physical Volume** directly maps to storage media

**Cartridge** contains physical volumes

**Device** contains read/write interfaces and mount points

**SOID** typed name for all Model-visible objects

## Mover

Media Access Point

a read/write interface with state (position, etc.)

Dual Role:

as *Device Manager*
controls media access points
as *Data Transfer Manager*
controls peer data flow between two movers

Commands: copy, load/unload, position

## Physical Volume Repository (PVR)

Contains cartridges and device mount points
Mounts cartridges onto device mount points
Supports location-independent access to contained cartridges and devices
Transfer mechanism may be human, robotic, or a combination
Commands: mount/unmount, stage, inject/eject

## Physical Volume Library (PVL)

Location-independent mounting of a set of cartridges resident in various PVRs
Secure and reliable mounting of removable media
Single uniform physical volume name space
Global resource allocation and compatible device selection
Lifecycle management for cartridges and physical volumes (assigned, scratch, maintenance)
Commands: mount/unmount, stage, import/export

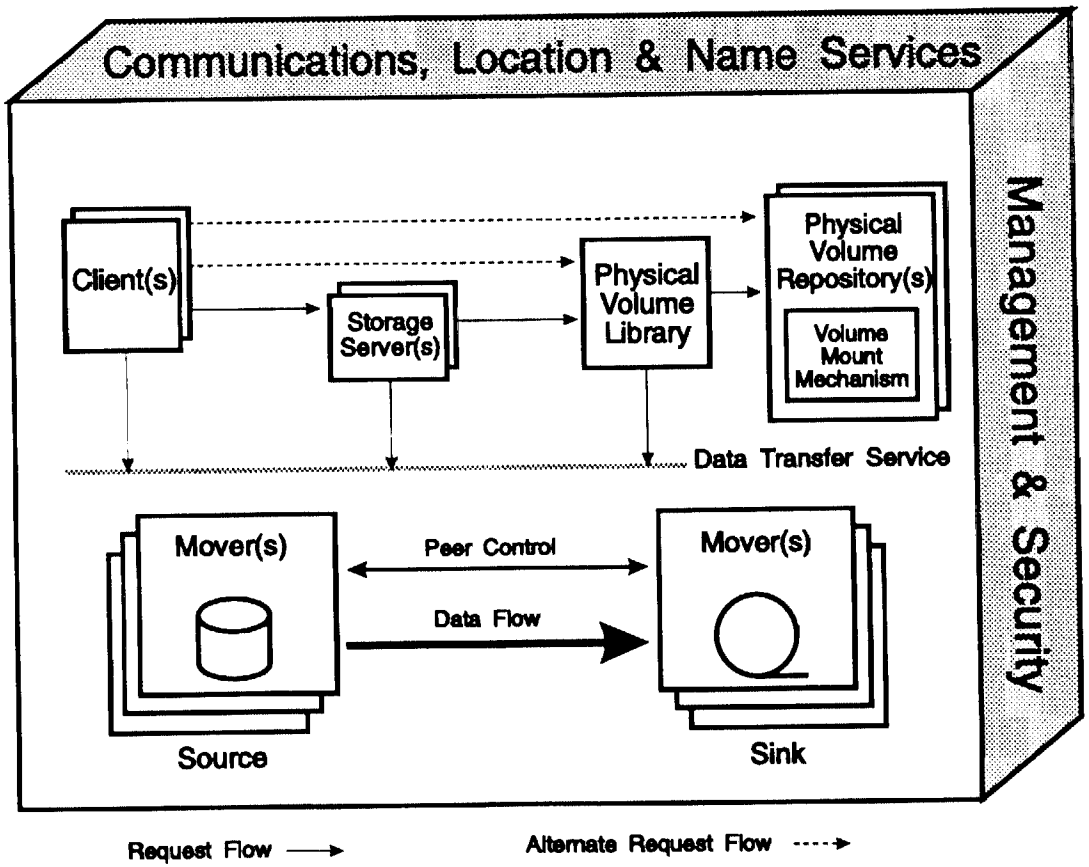
**Storage Server**

Composes physical and virtual stores into other virtual stores
Supports concatenation, replication, RAID, etc.
Virtual stores are managed and exported via storage groups
Range of access semantics:
  fine to coarse grained access & allocation
  shared (locking) to unshared access
Commands: copy, create/delete/reconfigure, mount/unmount, import/export

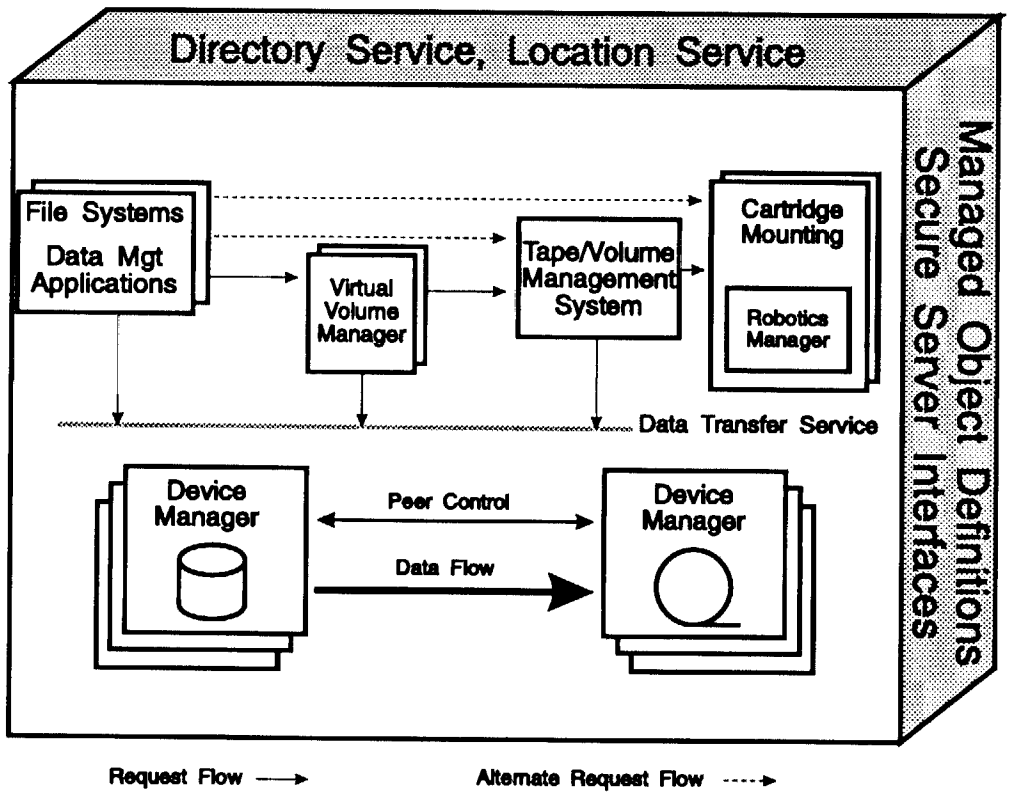
## Environmental Services

Communications
Location by SOID Name
Security
Layered Name Services (Directory Services)

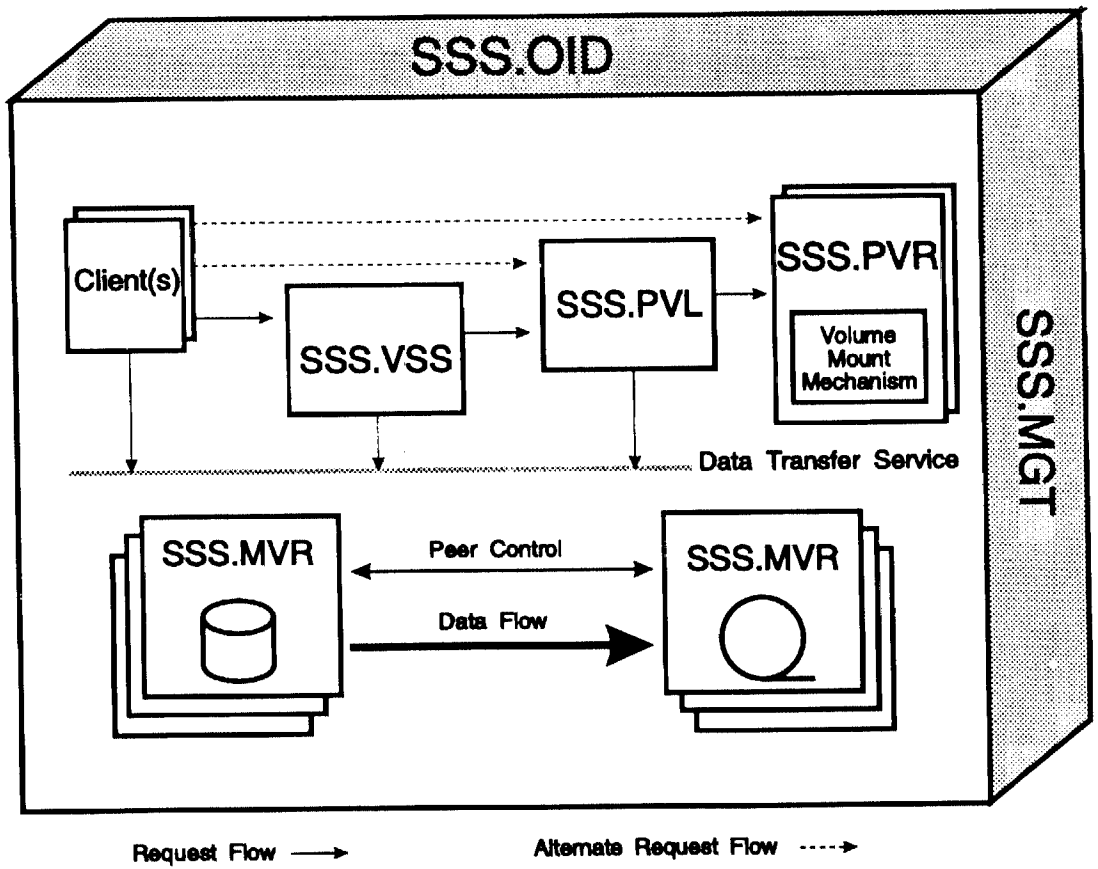
# Storage System Reference Model, Version 5



# Mapping the Reference Model to Existing Components



# Authorized Standards Projects in the SSSWG



N94-33798

# Role Of Formats In The Life Cycle Of Data

**Don Sawyer**
Code 633
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771
Phone: (301) 286-2748
Fax: (301) 286-1771
sawyer@nssdca.gsfc.nasa.gov

## 1.0 Introduction

This paper's perspective is based on the author's experience generating, analyzing, archiving, and distributing data obtained from satellites, and on the experience gained in data modeling and the development of standards for data understanding under the Consultative Committee for Space Data Systems (CCSDS).

Data formats are used to represent all information in digital form, and thus play a major role in all interchanges and access to this information. The need to more efficiently manage and process rapidly growing quantities of data, and to preserve the information contained therein, continue to drive a great interest in data formats.

The purpose of this paper is to examine the role of formats as they support the use of data within a space agency. The life-cycle identified is only one of many variations that would be recognized by those familiar with the 'space business', however it is expected that most of the issues raised will be pertinent to other 'space business' life cycles and to other 'non-space' disciplines as well.

## 2.0 Space Data Life Cycle Outline

This life-cycle has a clear beginning, but an ill-defined end, if it exists at all. In outline form, as shown in Figure 1, it begins with the generation of an Instrument Bit Stream (IBS) by a science instrument flown on board a space platform, the collection of these bits, and their transmission to a ground system. Ground processing is performed to recover the original IBS and to produce an Instrument Bit Stream Product (IBSP). This is followed by the application of various types of processing to correct for instrument characteristics and to produce a First Calibrated Product (FCP). The further application of various types of processing to compare the data with models and data from other sources usually results in some Archival Products (AP) suitable for long term archiving. The IBSP may also be a suitable product for long term archiving although this is not shown explicitly in the figure. Parts or all of these products are distributed over time spans of decades to data requesters for further analysis and for the archival of additional products derived from this analysis. Finally, at various points in time, there is the expected removal of these archival products from the archive. This demise of data is currently ill-defined because to date most data draining from scientific archives has been accidental. It is an ongoing issue to arrive at effective policies and practices for data retention in such archives.

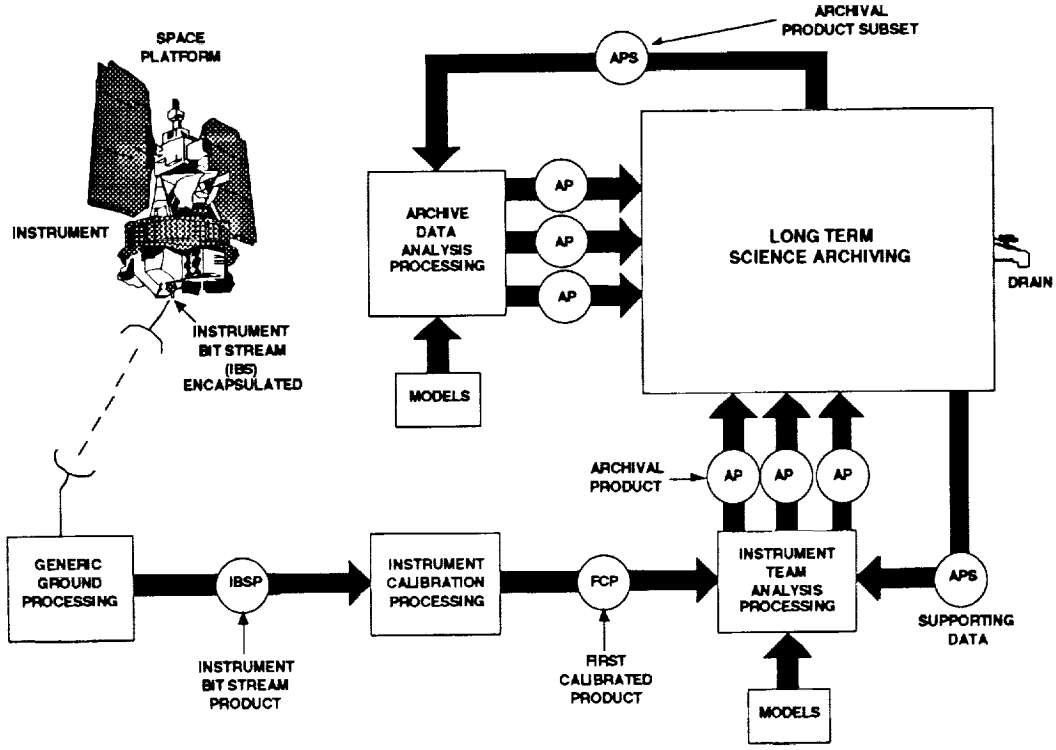
## 3.0 Role of Formats

### 3.1 Information, Not Just Data

The primary purpose for the birth of data in this life cycle is to provide new information that will be used to advance scientific understanding of our universe. (Throughout this paper,

"information" is understood as any kind of knowledge that can be transferred among users, while "data" is understood as the representation forms of that information.)

**FIGURE 1: A SPACE-DATA LIFE CYCLE**




Instruments onboard a space platform generate large volumes of data bits having repeating structures containing numbers representing a variety of observations and conditions. As the information represented by these number moves through the life cycle, it is augmented with other information and processed into new types of information. For example, a spinning particle detector on board a spacecraft counts particle events, but eventually this information is turned into count rates and then into particle fluxes. By combining this information with instrument looking direction information, particle fluxes in various directions are obtained. It is such particle fluxes that may be readily compared with our models of this space environment to validate and extend the models, thus increasing our understanding of the universe.

## 3.2 Formats and Metadata

At each stage in the life cycle, the information is represented in some way by data bits. "Format", or "Data Format" information typically refers to the way data bits are organized into recognizable data types (e.g., integers, reals, characters) and the way sequences of these data types are constructed to form ever more complex structures including whole data products that may cross multiple files on a physical volume. While this format type of information (i.e., metadata) is essential, much more metadata is needed to fully understand the information carried by a digital data stream or data product. For example, data types usually need additional attributes such as meaning (a text description), units, precision, and meaningful ranges or valid values. Information on the relationships among the data types, and the data structures, can be complex but must be known. Additionally, information on the context in which the data were obtained (e.g.,mission, processing history, instrument locations and pointing directions) is also required if the data are to be fully understood.

The amount and types of metadata that need to be recorded and formally associated with a data product, in order to fully understand it, depends on the knowledge of the intended users of the product. Clearly more metadata (supporting information) is going to be needed by high school students than by graduate students in the science discipline associated with the instrument being flown. Further, information that seems 'obvious' to those familiar with the production of a data product can rapidly become 'cloudy' when they have not worked with the product for many months or years. Experience suggests that clear categories of required metadata need to be defined, and then supposedly conforming instances need to be checked by independent reviewers to be sure the information is understandable and complete if the information is to be useable by others.

It is also true that space data products tend to become better understood as they are used over time. Some long term instrument drifts may become apparent and correlations with other data may improve the understanding of the space data product. Further, there may be new sets of metadata created to efficiently search the data in new ways. Thus the mechanisms used to represent and manipulate the metadata must support augmentation of the metadata over time.

Given the large volumes of data that need to be handled in the life cycle, efficient computer processing is a major consideration. Since writing, testing, and maintaining new software is a major expense, one might postulate that an ideal scenario is one in which all (or most) of the types of information, including relationships, that need to be represented should have generally agreed structural representations, or formats. This would mean that a computer interpretable language, capable of representing much of the information desired to be expressed in scientific data products, would be available. The extent to which such a language can be developed, and still provide sufficient storage and processing efficiency, is not clear to this author. That such a language is not available, coupled with the costs of unique software, can be viewed as the primary reasons for the great interest in data formats.

## 3.3 Access to Formatted Data

As there is no standard, formal, language for representing the kinds of scientific information we have been addressing, a number of techniques, each supporting a type of access to the information, are currently being used.

The most basic type of access is to build an understanding of a unique data structure, representing some set of scientific information, into the access software. This has the advantage of being very efficient for access, but has the severe disadvantage of being very costly since it promotes the generation of lots of unique software. It also makes information interchange difficult because recipients need to make modifications to their local software to be able to 'read and understand' each new data structure. For long term preservation of such data, good human understandable documentation going down to the bit level is needed to enable new access software to be written when needed. The existing software languages are inadequate for this task because it can be very hard to infer the underlying data structure from the software. This is understandable because these languages are designed primarily to transform data and not to provide an understanding of the data. In addition, the software languages usually do not address the needed bit level information.

Another type of access is to use software that supports a 'particular data model' (e.g., an n-dimensional array with dimensional and global attributes) having a private internal data structure. Information to be exchanged or stored is mapped to the data model, and then loaded into the internal data structure. This includes the data and some of the metadata needed to understand the data. The advantage for information interchange is that local software can be prepared to work with the data model, and thus be able to work with a variety of information as long as it can be usefully represented by the data model. The disadvantage is that no current data model can usefully represent all the types of scientific information that need to be exchanged, and the information's representation must be converted to the internal form of

the data model. Further, for long term preservation, the information must be carefully checked to ensure no loss when changes occur to the local hardware, operating system, or version of the data model access software. It is not clear that this can even be accomplished for large data volumes. When the data model and its software is no longer supported, the information will need to be extracted and moved to a new model or a new technique for information preservation.

A third type of access is provided by a variation of the second type of access. In this type the data model is represented by a standard, not private, internal data structure. This has an additional advantage for long term preservation in that the information content's dependence on hardware and operating system should be clear from the standard, and thus much more easily controlled against information loss when hardware and operating systems change. Further, there is no need to move the information to a new model or mechanism when software supporting the model is no longer to be maintained. The creation of new access software or other techniques can wait until the information needs to be used as long as the document representing the standard still exists. In other words, the lack of working access software does not mean the information has been lost.

A fourth type of access is provided by software that understands a standard data description capability that is embedded with the data. This differs from the 'particular data model' in that it is able to support a much more varied set of data structures, but typically (as far as the author is aware) does not support the relationships that would provide the capabilities of the 'particular data models'. The lack of 'particular data model' support is likely to be addressed as these description languages mature. The advantage to this approach is the great flexibility of data structures that can be supported. The disadvantage is that embedding the description capability with the data may cause considerable data expansion and thus may not be practical for large data volumes. Further, access to the information may be less efficient than for 'particular data models' with data structures tuned to their needs. However data description capabilities tare especially good for information preservation since the information is preserved as long as the data description standard is available. It should be noted that the 'particular data models' use some type of internal data description capability, but only to the extent needed to support their data model.

The last type of access described here uses software that understands a separable data description language. This is like the embedded capability described above, except the description may be separated from the data. This has the distinct advantage of not expanding the data volume, and of allowing this metadata to be independently managed and updated. This also allows the structure of the data to be efficient for representing the information. The disadvantage is that access to the data, using the description language software, is likely to be less efficient than for the 'particular data model' case.

Some of the current constraints on the use of data formats and their access mechanisms can be seen by a closer examination of the format and metadata issues in the space data life cycle.

## 3.4 Format (and Metadata) Issues in the Life Cycle

The approach in this section is to examine the environments suggested by Figure 1, and to determine a number of data format considerations that can affect the identified data products

### 3.4.1 Space Platform Environment

An instrument on board a space platform generates an Instrument Bit Stream (IBS). There may be several constraints that determine how the data are formatted, including: 1) space platform resource limitations such as telemetry bandwidth, on board power and weight, 2) available space platform data handling services, and 3) reuse of data structures from previous versions of the instrument or from similar missions.

This resource constrained environment drives a very efficient use of bits to represent numbers, flags, modes, and other conditions. Seldom are these numbers 16 or 32 bits in length as they usually are when generated by ground based computers, and often they are also scaled in various ways. Complex instruments will use mode indicators to signal the presence of different data structures, or different interpretations of the numbers in a given data structure. To reduce the telemetry burden, only information not easily added on the ground will be included in the IBS. Such data structures will need a lot of additional metadata, not found in the IBS, to be fully understandable.

The IBS is encapsulated in some manner before it is transmitted to the ground. Traditional space platform major and and minor telemetry frames force an overall structure that each instrument must share, with the result that one instrument's data stream is multiplexed with that of others unless there is only one primary instrument on the platform. The new standards for the Consultative Committee for Space Data Systems (CCSDS) packets and frames allows an instrument to own individual packets containing hundreds or thousands of bits. This greatly simplifies space platform data handling on the ground and gives the instrument developer much greater freedom in designing a data format for the packet content. However the instrument packet designer must now explicitly insert time tags as needed because, in general, there is no guaranteed relation between packet generation and an external clock.

If similar instruments have been flown on previous missions by the same instrument team, then it is likely that the same or similar data formats for the IBS will be used and this will facilitate some software reuse.

In summary, an IBS tends to have a great variety of data type representations for numbers, and to be quite instrument specific in the organization and meaning assigned to these data types. Nevertheless the types of information represented, such as images, time series, and spectra, are more common across different instruments and missions than are the various representations or formats used. The use of data description languages may be the best approach to providing common access (i.e.,reusable software) while supporting a variety of bit efficient representations.

### 3.4.2 Generic Ground Processing

The function of Generic Ground Processing in this life cycle is to remove the artifacts of the space to ground transmission domain, to recover the original IBS, and to put out an IBSP which has a basic structure that is the same from mission to mission. Typically the IBS is collected for a previously agreed period, such as an orbit or a day, before an IBSP is released. The IBSP will include, in addition to the original IBS bits, attributes related to the accumulation period such as orbit number or time period covered, and possibly some quality information relating to the reliability of the recovered IBS. This information will be appended without altering the format of the IBS, which in general is transparent to Generic Ground Processing. This stage of the life cycle is a reasonable place to add one or more identifiers of the metadata needed to convert the IBSP into useful information. This has the benefit of stimulating the documentation of this metadata (which might include a formal description of the format using a data description language), and making the IBSP much more archivable as well as useful if it needs to be shared with a distributed set of colleagues. This will also provide a good start to the metadata that will be needed to support other products in the life cycle, and will provide a source of information to stimulate reuse in other mission's products. Note that it appears less desirable to actually include most of the metadata, as opposed to identifiers of the metadata, in the IBSP. Including this metadata may significantly expand the size of the product, and it freezes the metadata at an early stage of understanding of the product.

An IBSP instance may be one or a few files, and may be distributed via networks or via physical media (sequential or random access).

This environment may include a temporary, or semi-permanent, archive for the IBSP instances from many instruments and missions. It is assumed that any such temporary archiving is done without needing to understand the content of each IBS. For example, the archive catalog would be populated with information such as spacecraft identifications and orbit numbers that are obtained from sources other than the IBS. Considerations for permanent archiving are addressed in sections 3.4.4, 3.4.5, and 3.4.6.

In summary, the format of the IBSP includes that of the IBS, but adds additional attributes associated with the collection interval, mission, instrument, etc. to make the resulting product readily recognizable and archivable without having to parse the content of the IBS structure. The format should be efficiently accessible whether the distribution is via networks or physical media (both sequential and random access) since the next stage of processing will most likely be done in a pipeline approach and a media independent format will have the best chance of being a long lived output format from this Generic Ground Processing environment.

### 3.4.3 Instrument Calibration Processing

The functionality envisioned in this stage of the life cycle is primarily the conversion of the raw IBSP numbers (actually the IBS numbers) to more meaningful quantities. These conversions are most likely to be reversible (e.g., multiplying values by a constant), although some non-reversible calibrations may also be performed at this stage. Although not shown explicitly in Figure 1, this processing may require incorporation of other data streams derived from the space platform, such as orbit and attitude information, or data from ground observations, to complete the calibrations. Typically the result is an Initial Calibrated Product (ICP) whose format is organizationally similar to the IBSP, but with some information (perhaps quality) eliminated and other information (such as location and pointing direction) added.

This product may, or may not, have most of its values converted to 8, 16, or 32 bit quantities to be more easily processible by software. Even greater changes in the format are likely to take place if there has been prior agreements to push all data into a particular data model whose implementation software maintains its own internal format, or if there is a need to conform to input requirements for the next stage of processing. These decisions will most likely depend on trade-offs among the volume expansion that would take place, the availability of storage, and use of common mechanisms for access to this data product.

This processing would most likely be done at a mission or project facility, and typically an ICP instance would be one or a few files.

The metadata associated with the IBSP should be a good starting point for the metadata needed for this new product, but it needs to be suitably updated. It is very important that the insight gained from overseeing the calibration processing be recorded as supporting metadata. A new metadata identifier, for this new set of metadata, can be inserted in forming the ICP. Again it appears desirable not to include substantial amounts of metadata directly within the product for the same reasons given in section 3.4.2. The use of an overall format organization that is efficiently accessible from all types of media or from networks is also desirable if the mission is long lived and evolution of systems is a concern.

In summary, the format of the ICP may, or may not, be substantially different from that of the IBSP. This appears to depend primarily on the nature of the calibration performed, on the relative availability of storage space for this product, and on the planned use of access mechanisms such as use of a single data model or use of a data description language. As envisioned in this life cycle, the basic nature of the information contained in the ICP is not

substantially different from that in the IBSP or IBS. In other words, the presence of an image, time series, or spectrum, for example, would be present throughout and through a simple mapping (calibration function) be related to the output of the instrument. More complex transformations of the information are assumed to take place in the next stage of the life cycle.

### 3.4.4 Instrument Team Analysis Processing

The two primary objectives of this stage of the life cycle are to extract new understanding of our environment from the data, and to produce useful APs that support future analysis. The functions envisioned to meet the objectives are wide ranging. They include reprocessing the IBSP with improved calibration information, the application of physics models to effect substantial transformations of the ICP into what are often referred to as "higher level products", and the selective subsetting of the products for incorporation into a variety of favorite data analysis and display tools. Subsets of data (Archival Product Subsets, or APS) from external archives may need to be folded into this processing either to create a new product or for comparisons. Papers for publication in the literature would be generated, and some of the products (including possibly the IBSP and FCP) generated should be suitable for preparation for long term archiving.

This processing environment will need to have its own archive to hold the ICP instances, support products ingested from long term archives, and intermediate products generated during the processing. This archive requirement is not explicitly shown in Figure 1, but may be met by some combination of project archive support and local archive support. In general, it must be assumed that data products in multiple formats will need to be archivable and accessible to this analysis processing environment. The formats of these products can ease the archiving function if they include an easily accessible set of attributes (e.g., time period covered, mission, instrument) that can be used to populate an archival catalog, and if they did so in a way that allowed them to be updated and accessed without having to modify or parse the rest of the data product.

The analysis processing environment's desires for the formats of the input products (ICP and APS) shown in Figure 1 can be widely varying. While this environment may be able to significantly affect the ICP format, such as ensuring that arrays are stored in an efficient way for the local hardware, this is much less likely for data acquired from long term archives. Thus this environment must work with data in multiple formats.

Given that this environment is likely to have a good set of resources, the detailed formats at the record level (bits and bytes) are less of a concern than overall product organization and a clear understanding of the information present in the data. Maintaining an up-to-date set of metadata linked to the ICP will help to avoid information loss, particularly when this stage of processing involves sending data to distributed colleagues, and it will aid in preparing data products for subsequent archiving. Processing the ICP and APS will generally involve applying selection criteria to values within these products, and the extraction of subsets of values when the selection criteria are satisfied. The focus will be on software to perform this subsetting and extraction. The output formats for these extractions may be driven by the input formats required by favorite analysis tools.

The generation of Archival Products (AP) will be constrained by the requirements of the intended archive, as well as by the internal formats used in this analysis environment. The archive should require formats that are as media independent as possible in order to facilitate management of the products within the archive. The archive may require a specific set of attributes with each instance of an AP to support automated data ingest cataloging and future subsetting. The extent of metadata, needed to support use by some minimally trained potential user (e.g., high school student, graduate student) and going down to the bit level, needs to be included with the AP.

A typical AP instance, submitted to an archive, may range up to tens of thousands of files. A great many such instances many be sent over several years to complete the archiving of a single AP.

In summary, the information processing within this stage is likely to deal with a number of formats. Constraints on these formats typically come from the input constraints of commonly used data analysis tools, and from archive constraints on the types of formats the archives will provide. Further constraints may come from satisfying local data management needs by providing a set of attributes with each product instance to ease local cataloging. Finally, for those products which are intended for long term archives, additional constraints are likely to be imposed by particular archives, including the association of complete sets of metadata with each product instance. The impacts of these constraints can be minimized if formats are adopted which support the archive's data ingest and metadata needs since these formats should also support local data management and cataloging needs and aid in the distribution of meaningful products to colleagues.

### 3.4.5 Long Term Science Archiving

The two primary objectives of this stage of the life cycle are to preserve information (not just data bits) for an indefinite period (assumed to be many decades, at least), and to provide requesters with a range of access services. The role of formats is a key element in both of these objectives.

The information preservation objective has proven to be quite difficult and tends to be greatly underestimated by new archives that have not felt the full impacts of technological change. For APs ingested into the archive, full data product metadata, down to the bit level, is still needed. Software access, alone, as the way to understand and work with bit structures has proven to be inadequate. It is very expensive to ensure that software, archived with a data product, performs properly against changing hardware and operating systems. Even software which supports multiple data products is unlikely to have sufficient resources behind it to permit the extent of testing needed to be very sure that all the data products are accessible without information loss. Large, stable vendors stand the best chance of having the resources to do extensive testing, but even here there is no guarantee against some data/information loss. An archive relying on such software must also have an extensive test plan involving accessing and comparing data values with the new and old software. Software which provides many types of information is particularly difficult to adequately test and this becomes more difficult as the archive data volume grows. In addition, software (i.e., data manipulation languages) are inadequate as data description languages because they rely on local representations of bit level data types which tend to change with new hardware and operating systems. Therefore it appears that the use of standard data description languages, coupled with human readable descriptions intended to be complete and understandable 50 years into the future, is the best current approach to addressing these aspects of information preservation.

Archives which do format conversions on AP ingest to an internal data model also risk information loss unless they have done a very careful mapping of the incoming information to that model. This can be difficult since archive personnel may not sufficiently understand the incoming information to ensure against information loss. Therefore it is safest to avoid format conversions for archival copies of APs, and to limit conversions to the provision of special data access services for some APSs. The provision of some of these services, such as rapid online access to large amounts of this information, is likely to require some format conversions given current technology. This implies a trade-off must be made among information preservation , efficient online access services to the information , and storage volume. The author believes that only two of these three can be optimized in any single system, at least with current technology. This appears t o be a major challenge for archives that is easily overlooked since the lack of desirable access services is felt immediately while

the loss of information may not be noticed until there has been major technology evolution (e.g., a decade or more).

AP formats should also support the updating of metadata over time without having to rewrite the associated data. For example, new calibration coefficients may be defined, new interpretations of certain types of observations may need to be documented, and metadata errors may need to be corrected.

AP formats, including the associated metadata and its linkage to the data, should avoid or at least isolate, any media dependence. For example, the embedding of directory and file names in the data or metadata can produce name conflicts when the information must be moved, in response to technology evolution, to new media types within the archive. It also makes subsetting of the information (i.e. creating an APS) for distribution to requesters difficult to accomplish. Since all references to directory and file names can not be eliminated, the best approach appears to be to use formats which allow these names to be isolated and readily updated as needed. Directory names and file names should never be used in metadata text as it become very hard to update. Unfortunately this is common practice among data producers because it is convenient in a local system and there is no standard way accepted to name, and thus refer, to other data objects.

A difficulty in producing an APS with proper metadata results from having to extract that set of metadata, from the total metadata associated with the original AP, that is pertinent to understanding the APS. This puts constraints on the format of the metadata and suggests that the metadata should be broken into separate objects, some of which will apply to all possible APSs and some that will be need to be shaped for particular APSs.

The efficient ingest and cataloging of APs, which is needed to support access to these data as APSs, requires that catalog attribute objects accompany the data products and be linked to this data at a useful level of granularity. The data producers, after they understand the ingest requirement of archives, are in the best position to prepare products that include these attribute objects. Ideally, all data products would include such objects to facilitate local cataloging in both temporary and long term archives.

APs which are submitted to an archive on media volumes containing a great many data objects also need to include standard table-of-contents and/or index objects. The purpose of these objects is to permit efficient subsetting of the AP into an APS in response to requests. Again, it is much easier for the data producer to create these objects in consultation with the archive than it is for the archive to produce them after a great volume of data has been received.

In summary, archives should require AP formats to be as media independent as possible and to include complete metadata, down to the bit level, in order to maximize information preservation. This metadata should be updateable without having to rewrite the data. The inclusion of catalog attribute objects, including table-of-contents and index objects, can greatly improve archive efficiency and access services at little cost to AP producers. The provision of efficient online access to large amounts of information may require format conversions and special software, with the attendant increase in data storage volume over that devoted to information preservation. The use of data description languages can support information preservation while allowing access to a great variety of data structures, but current data description languages are not yet providing very efficient access.

### 3.4.6 Archive Data Analysis Processing

There are two primary objectives for this stage and they are the same as for Instrument Team Analysis Processing; to extract new understanding of our environment from the data, and to produce useful APs that support future analysis. The primary differences are that the data

come from an archive as one or more APSs, and the available processing resources may be much less than for Instrument Team Analysis Processing.

An APS needs to have a complete metadata set associated with it as it may be decades since the information was put into the archive and there may not be anyone who is familiar with the data or even the mission. APS formats need to support incremental access to the information, such as through table-of-contents and index objects, when the volume of data in the APS is large (e.g., CD-ROM).

It is a great benefit to this analysis processing if there exists working access software for a given APS. An APS that includes a format description written in a standard data description language stands a good chance of having working software that may be used to access the data. An APS that conforms to a particular data model may also have working software, but this software should be supplied as an addition to the metadata, not as a substitute.

## 4.0 Summary

This space data life cycle is only one of many variations seen in the 'space business'. However it is expected that most of the issues and concerns raised will also be applicable to 'non-space business' life cycles.

The preservation of information (not just bits) throughout this cycle is a primary objective, and this requires appropriate metadata, down to the bit level, at each stage of the cycle. Software alone is not suitable for information preservation. The required metadata grows throughout this cycle, and must be associated with the data in ways which permit both the data and metadata to move easily to new types of media, including both random and sequential.

Data formats are used to represent the data and the metadata, and to link the two. The data formats are subject to various constraints as the information moves through the life cycle, and no single bit representations for science objects (e.g., image, time series) is practical at all stages of the cycle. The need to support subsetting of both the data and metadata is apparent in several of the stages, as is the need to support archival or repository ingest.

It is suggested that data description languages may be a good approach to supporting information preservation and some automated access, but are not yet up to providing efficient access for a range of archive online services. Therefore it may be necessary, given current technology, to use one copy of the information for preservation and a somewhat differently formatted version (in some cases) for efficient online access services. However the products sent out from these access services may be well served to have associated standard data description language metadata to support automated access. The use of particular data models have a role to play, particularly in terms of efficient access, but they can get in the way of information preservation if archives try to use them as their primary storage mechanism.

N94-33799

# A Reference Model For Scientific Information Interchange

**Lou Reich**
Computer Sciences Corporation
Code 502
4600 Powder Mill Road
Beltsville, Maryland 20705
Phone: (301) 572-8445
FAX: (301) 595-1774
lreich@gsfcmail.nasa.gov

**Don Sawyer**
Goddard Space Flight Center
Code 633
Greenbelt, Maryland 20771
Phone: (301) 286-2748
FAX: (301) 286-1771
sawyer@nssdca.gsfc.nasa.gov

**Randy Davis**
Laboratory for Atmospheric and Space Physics
University of Colorado
Campus Box 590
Boulder, Colorado 80309
Phone: (303) 492-6867
FAX: (303) 492-6444,
davis@aquila.colorado.edu

## I. Introduction

This paper presents an overview of an Information Interchange Reference Model (IIRM) currently being developed by individuals participating in the Consultative Committee for Space Data Systems (CCSDS) Panel 2, the Planetary Data Systems (PDS), and the Committee on Earth Observing Satellites (CEOS). This is an ongoing research activity and is not an official position by these bodies.

This reference model provides a framework for describing and assessing current and proposed methodologies for information interchange within and among the space agencies. It is hoped that this model will improve interoperability between the various methodologies. As such, this model attempts to address key information interchange issues as seen by the producers and users of space-related data and to put them into a coherent framework.

Information is understood as the knowledge (e.g., the scientific content) represented by data. Therefore, concern is not primarily on mechanisms for transferring data from user to user [e.g., compact disk read-only memory (CD-ROM), wide-area networks, optical tape, and so forth] but on how information is encoded as data and how the information content is maintained with minimal loss or distortion during transmittal. The model assumes open systems, which means that the protocols or methods used should be fully described and the descriptions publicly available. Ideally these protocols are promoted by recognized standards organizations using processes that permit involvement by those most likely to be affected, thereby enhancing the protocol's stability and the likelihood of wide support.

## II. Issues In Scientific Information Interchange

Figure 1 presents an overview of what is meant by information interchange. The left side indicates the existence of several pieces of information in various local forms and knowledge of the relationships among them. The objective for the data producer is to assemble these pieces and the appropriate knowledge in a way that can be transferred across a spatial and temporal gap to a consumer system where any or all of the pieces of information and the relationships between them can be identified, extracted, and used in processing and display.

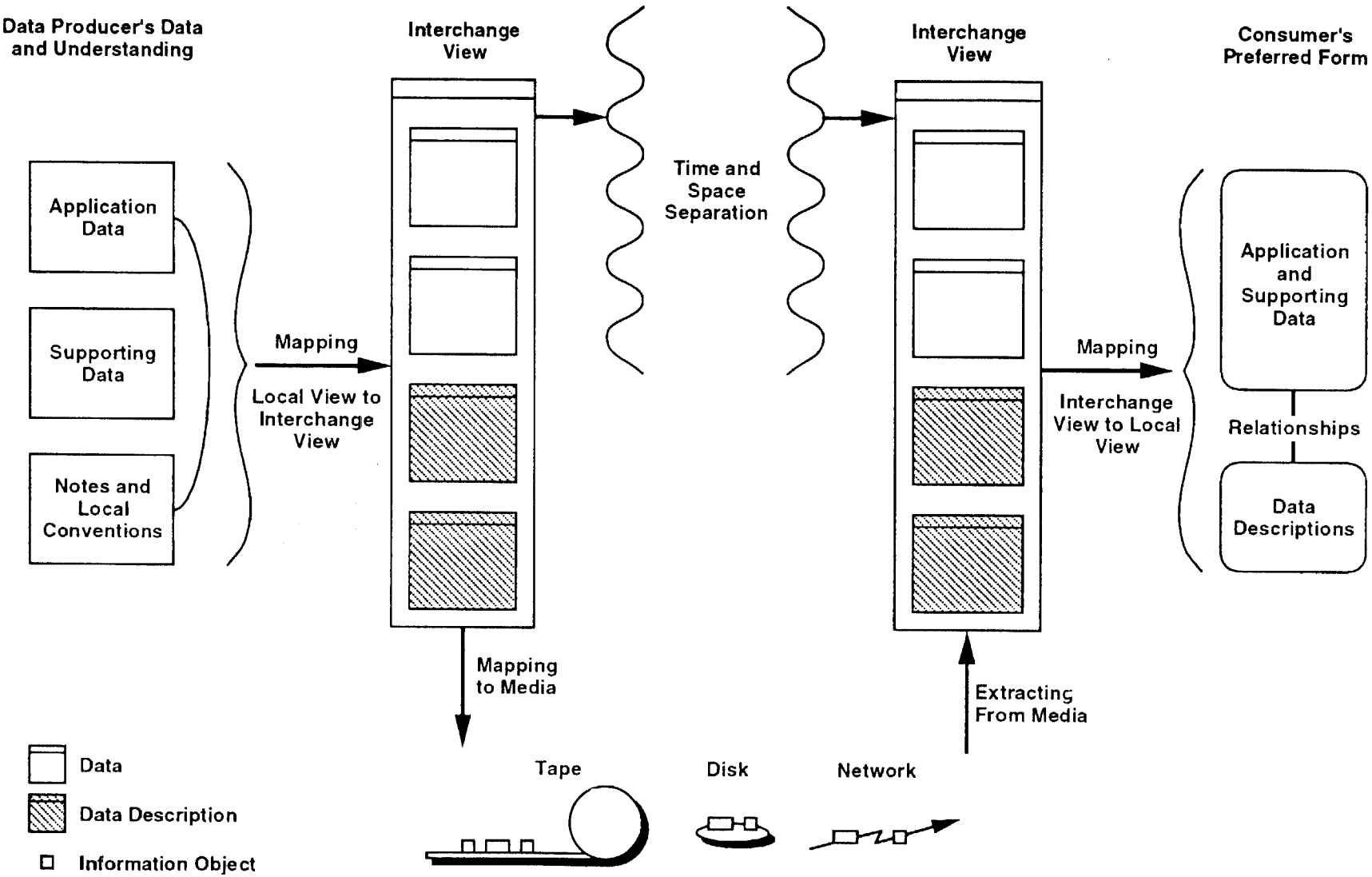



**Figure 1. Information Interchange Process**

An essential element in this view is the physical (spatial) separation of the two systems. Temporal separations can range from a fraction of a second to many decades.

The problem of moving strings of bytes reliably from senders to receivers has been successfully addressed by several suites of standards. The Open Systems Interconnection (OSI) model and the standards that adhere to it provide a solid framework for understanding and implementing systems that move data across networks. The packet telemetry and telecommand standards developed by the CCSDS supplement OSI-compliant protocols with capabilities specifically designed for communications with satellites. The Sony-Philips Red and Green Books provide the basis for encoding information on CD-ROM media so that the resulting disks can be read in any CD-ROM reader. These protocols can assure that a transmitted byte string is received completely and in the correct order (or if it is not, that the failure is reported to the receiver).

The protocols cited above do not, however, address all the needed aspects of encoding and interpreting the information within byte strings. They transport blocks, packets, and frames, whereas end users in the space sciences deal with images, spectra, tables, and maps. How then do we address the transport of the information objects, such as images and tables, to scientists? The OSI model allows for applications-level protocols that provide the rules for encoding and interpreting information within an applications domain. It is the applications-level protocols (sometimes with assistance from presentation layer protocols) that allow recipients to extract information from the bytes of data they receive. Few formally standardized data transfer methods for scientific information exist, but the need for them is growing. Most science disciplines within NASA are developing or seeking standard ways to transfer complex scientific information. The IIRM provides a mechanism for characterizing data transfer methods (with emphasis on those for scientific applications) so that users can describe the similarities and differences between existing or proposed methods. This may provide a basis for discussing the way individual science disciplines view their data and perhaps result in greater uniformity between data transfer methods for scientists. The more standardized the methods are the more automated the services can be for dealing with the information in both producer and consumer environments.

Several characteristics of space science applications complicate the information interchange process, including:

- Highly heterogeneous computing environments
- Voluminous data and metadata
- Wide variations in the level of user sophistication
- A large--and expanding--set of information relationships

The remainder of this paper discusses some of the key issues of information transfer in space science applications. This list is a first pass and is probably not comprehensive. Interested readers are encouraged to submit any additional issues or comments on current issues.

## A. Encoding Information Into a Data Stream

Whenever information is stored or processed, it is encoded as a series of primitive data elements. Use of heterogeneous computer hardware and bit-efficient coding schemes for data from satellites and science instruments results in a wide variety of bit sequences to represent primitive data types (e.g., integer and floating point numbers). For efficiency, the data processed by a computer should be encoded in the formats that the computer hardware supports; however, these formats often differ for the computer systems used by the producer and consumer of a data stream. There are several ways to address this problem:

- The producer's system may know the local representation of the consumer's system and convert data to the consumer's local representation before sending

- The producer's system can inform the receiving system about the data representation and require the consumer to convert the data it receives to its local representation

- The producer's system can convert data into an agreed-on format, and the consumer's system can convert from the agreed format to its local representation

No single solution is best for all situations. Despite today's sophisticated and fast computer hardware, converting large volumes of information from one format to another for interchange or archiving is often impractical; data volumes appear to increase as rapidly as processing power. This means that a scientist's access to information may be limited simply by the difficulties of data translation.

Encoding issues also arise in every layer of software through which information must pass when transferring data streams. For example, some operating systems impose private record encoding schemes within files that can restrict or complicate the flow of data files within an open system.

Programming languages present an additional problem: a programming language's set of base types is usually richer than the primitive data types represented in hardware (for example, arrays and enumerated types). However, different programming languages use different conventions to encode the same base type, and encoding information as a sequence of base data types that can be recognized and manipulated by all the languages that might be used to process the information is often difficult. For example, exchanging arrays across different languages is often difficult because arrays for some languages are implicitly column major, and for others they are row major. These kinds of problems have led to specialized data definition languages (DDLs) that allow data to be fully described in a way that is independent of any particular programming language. Even with DDLs, some modification of the information may be required before the information is used with a specific programming language (the array majority issue is such a problem).

## B. Identifying and Accessing the Information in a Data Stream

The receiver of a data stream must be able to locate, identify, and access each major information unit in the data. These units are called information objects; however, use of this term does not imply that the systems producing and consuming them necessarily conform to the principles of object-oriented programming.

For open systems, a very large number of different types of information objects may be transmitted. The producer knows the identity and order of objects within any data stream it transmits, but it is presumed that the consumer has no prior knowledge of the data contents. A mechanism is therefore required to identify and describe each information object in the data. The usual mechanism is to provide supporting information, or metadata, that identifies and describes the information objects. Some of the metadata acts like a table of contents or index in helping to locate and identify the information objects. Software is then provided to browse through a large set of information objects to find the specific objects required for an application or to create a useful subset of objects. Metadata are also used to describe the attributes of information objects and to describe the relationships between information objects.

Numerous issues are associated with metadata. First, the mechanism for encoding and supplying the metadata must be determined. Second, the amount and completeness of metadata needed to describe information objects and their relationships is inversely proportional to the inherent level of the consumers understanding of the information objects received. Producers must determine the metadata needed to make the transmitted data understandable and accessible to the intended audience. The requirements are particularly

stringent for archived data, where a data stream may be preserved beyond the life of any hardware or software that created it or that can access it. In such cases, sufficient descriptive information must be available to allow deciphering of the entire data stream.

Metadata are data, and like other kinds of data, generally require their own metadata to allow receivers to understand and interpret them. This meta-metadata must also be provided. A current mechanism for storing and providing some of this meta-metadata is a database called a Data Dictionary or Data Entity Dictionary (DED). The DED defines information in a consistent format.

### C. Interpreting Information in a Data Stream

Received information must often be placed into a context broader than the containing data stream. A common problem is unambiguously identifying and naming an object so that it can be distinguished from all other information objects that exist in a large system. A traditional method of naming the information objects held in computer systems is by location, for example, directory path names for files. This method causes problems, however, when the location of an information object changes, then references to the object (e.g., a file reference appearing within a text document) must also change.

Another issue is how received information objects relate to other information objects within a large system. Software reusability depends in large measure on this issue, for if a piece of software has applicability to a wide variety of data objects, a mechanism is needed for determining which objects the software can and cannot handle. The inheritance mechanism used in object-oriented programming addresses this problem by providing a hierarchy or network to determine how each type of object is related to all other types. Typically in an object-oriented system, software that works for one type of object will work for all objects of that type and for all types of objects derived from the original type.

## III. Overview Of An Information Interchange Reference Model

The IIRM consists of three layers, as shown in Figure 2. The layers support information partitioning and a degree of information hiding, which grows as one moves from the lowest layer to the top layer. This structure allows the functionality assigned to each layer to be addressed separately and allows users to assume that the functionality of the lower layers is provided in support of a given upper layer. An implementation need not adhere to strict information hiding to be consistent with this model; access to information at a lower layer may be needed to meet special circumstances. For a given implementation, the three layers work together. Note that not every implementation will interoperate with other implementations at the adjacent (lower or upper) layers.

The top layer of the IIRM is based on the object-oriented paradigm. This schema includes the definition of base types, a type hierarchy, and relationships that model the process of information interchange. Use of an object-oriented data model, by identifying the specific objects defined and supported (either implicitly or explicitly) by various information interchange methodologies, makes it possible to identify similar objects across implementations and to compare the capabilities and mechanisms of each implementation. This technique allows analysis of non-object-oriented methodologies through the identification of the implicit objects that a methodology supports. In addition, an object-oriented view allows for explicit definition of complex relationships among scientific data and metadata. Current object-oriented data models do not discuss underlying representation of data. Because such representation is an important aspect of science data exchange, the IIRM augments the object-oriented data model with the additional (lower) layers that deal with data representation issues.

The functionality addressed in each of the layers is described in the sections that follow.

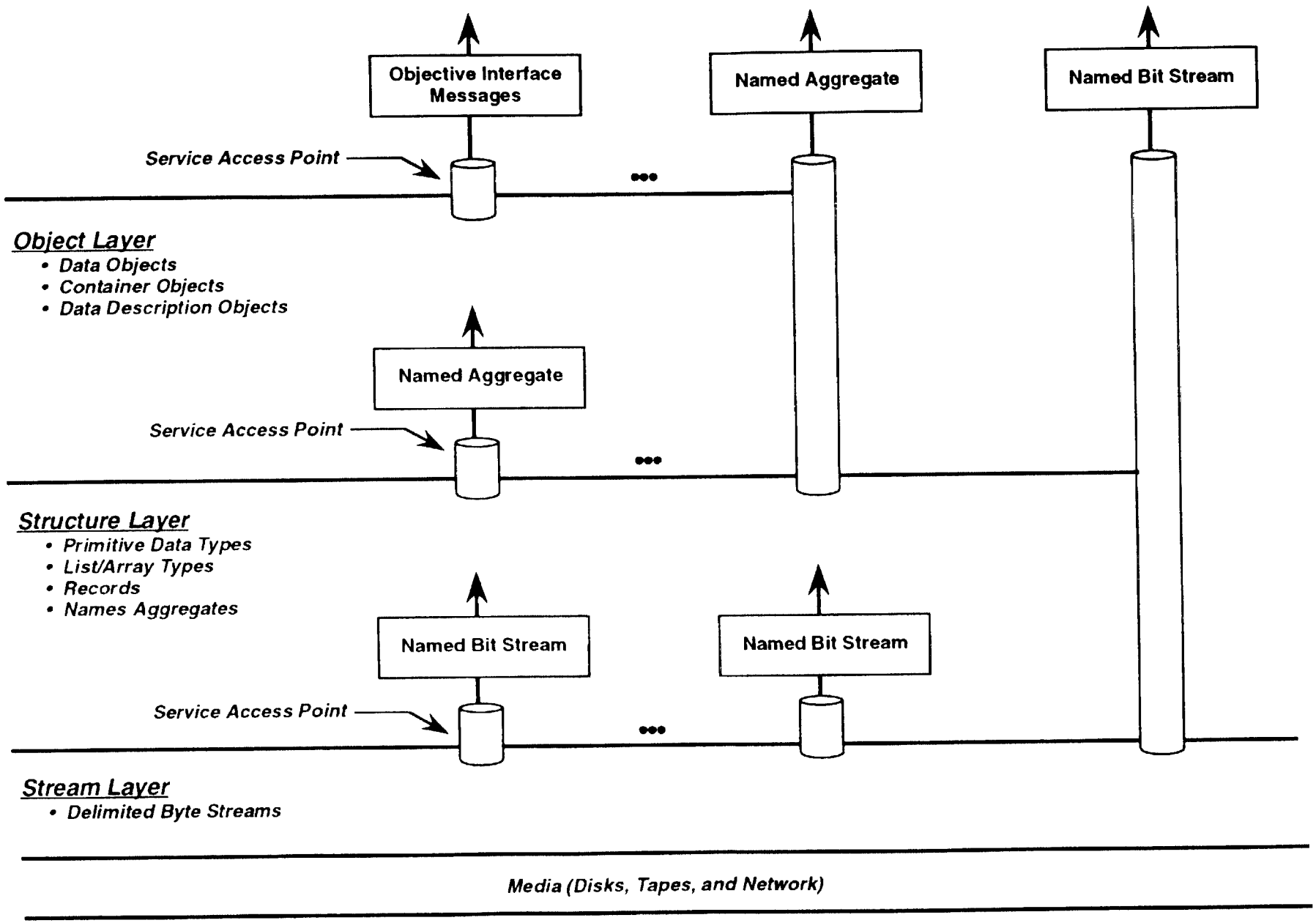



Figure 2. Information Interchange Core Model

## A. Stream Layer

As noted previously, the IIRM augments existing models of the data transfer process, like the OSI model. Because the IIRM addresses issues found in the user-oriented top layers (the applications and presentation layers) of the OSI model, the IIRM can assume the existence of protocols for the lower five layers of the OSI stack (the physical through session layers) and need not duplicate the functionality of those lower layers. However, the IIRM applies not just for information interchange over networks; it is for information transported on media like tape and CD-ROM as well. The stream layer—the lowest layer of the IIRM—provides the interface between the IIRM and medium-dependent standards, protocols and mechanisms for data transport. It hides the unique characteristics of the transport medium by stripping any artifacts of the storage or transmission process (such as packet formats, block sizes, inter-record gaps, and error-correction codes) and it provides the higher levels of the IIRM with a consistent view of data that is independent of its medium. This common view is that data are simply collections of named sequences of bits. The term name here means any unique key for locating the data bytes of interest, including path names for files, a virtual channel ID for CCSDS telemetry, and so on.

Examples of standards and protocols that provide the functionality needed in the stream layer are ISO 9660 for CD-ROM, ISO standard labels on magnetic tapes, and file transfer protocol (FTP) on networks. For example, the ISO-9660 standard provides the volume and directory information needed to locate a file on a CD–ROM volume and sufficient information about the file format that a user retrieve the file as a sequence of bits. It ignores issues such as record structure (fixed length or variable length). The returned file is simply a sequence of bytes at this point; access to the information encoded within this file (or any other data stream) is addressed in the structure layer, described in the next section.

## B. Structure Layer

As mentioned previously, information must be coded into primitive data types that can be recognized and accessed by computer hardware and operating systems. In the structure layer, information is viewed as a sequence of primitive data types. For any implementation, the structure layer defines the primitive types that are recognized. This usually means at least characters and integer and real numbers. Primitive types can also include the aggregation types typically supported in computer languages, including the array (where each element consists of the same type of data) and a record or structure that can (potentially) hold more than one type of data. An enumeration type is also often provided as a primitive type. As noted earlier, because of the efficiency constraints often imposed on space science data, users sometimes create their own representations for primitive data types (e.g., 6-bit integer numbers). Issues relating to the representation of primitive data types are resolved in this layer.

All types of information are built from these primitive types. Through the structure layer, the information is mapped into primitive types and then into the corresponding bits and bytes of a data stream. Note that a single structure may be distributed among several streams. The issues of the structure layer are often thought of as data format issues and are handled automatically by DDLs.

## C. Object Layer

The highest layer in the IIRM is the object layer, wherein information is represented as objects that are recognizable and meaningful to end users. For scientists, this includes objects such as images, spectra, and histograms. The object layer adds semantic meaning to the data treated by the lower layers of the model. Some specific functions of this layer include the following:

- Recognizing data types based on information content rather than on the representation of those data at the structure layer. For example, many different kinds of objects—images, maps, and tables—can be implemented at the structure level using arrays.

Within the object layer, images, maps, and tables are recognized and treated as distinct types of information.

- Presenting applications with a consistent interface to similar kinds of information objects, regardless of their underlying representations.

- Providing a schema mechanism to identify the characteristics of objects that are visible to users along with the relationships between objects.

To characterize information in the object layer, the IIRM uses concepts and terminology that have been developed in the object-oriented community. Agreement is not unanimous about what constitutes an object-oriented approach, but most models of object-oriented systems currently in use or in development share the key features needed. One such model, the concrete object model developed by the Object Data Management Group (ODMG), is being used to facilitate the standardization of Object Database Management Systems (ODBMSs). This paper uses the ODMG's approach to describe the entities at the object layer of the IIRM. This model can be briefly summarized as follows:

- The basic modeling primitive is the object. As with real-world objects, information objects can be arbitrarily complex. For example, in the real world, both a bolt and an automobile are objects, although the latter is significantly more elaborate than the former. Similarly, a pixel of an image, an entire image, and the entire dataset containing the image can all be treated as objects.

- Objects can be categorized into types.

- Instances of objects are created using object types as templates. Each object instance possesses all the characteristics of its type. The set of all instances of a specific object type is called that type's extent.

A type has one interface and one or more implementations. The interface defines the external public behavior supported by all instances of a type. The components of the interface are as follows:

- Attributes—Characteristics of the object for which an external user can get the values for any instance of the object

- Relationships—Logical paths an external user can traverse to move from an object instance to related object instances

- Operations—Actions an external user can invoke on an instance of an object

An implementation defines the internal or private data structures and procedures that support the externally visible states and behaviors. A single interface may have several alternative implementations.

Object types are related to one another using the supertype/subtype (or parent/child) relationship. This relationship links all object types according to their shared characteristics and is commonly represented as an acyclic graph. For example, a type called Faculty Member may have subtypes called Instructor and Associate Professor, and Faculty Member may in turn be a subtype of Person. All of the attributes, relationships, and operations defined for a supertype are inherited by the subtype. The subtype may add attributes, relationships, and operations to introduce behaviors or states unique to the instances of the subtype. A subtype may also refine the attributes, relationships, and operations it inherits to specialize them to the behavior and range of state values appropriate for instances of the subtype.

## IV. Model Schema For Scientific Information Interchange

The three-layer model just described is general and can describe many data interchange problems. The goal of the IIRM, however, is to have a model specifically suited to describing scientific data interchange. In this section the model adds a domain-specific object-layer schema that allows characterization and comparison of systems for scientific data interchange.

To show what the description of an object looks like, Figure 3 presents a formal description of an image as represented in the object layer of a hypothetical data system. The descriptions of each component are given in plain English, although for a real data system the descriptions of attributes, operations, and languages will typically be in a formal, computer-readable language.

A key point about scientific data in general can be found in the description of relationships in the sample: Manipulation of a primary scientific data object such as an image frequently requires substantial auxiliary data. For example, interpretation of image objects requires a knowledge of the camera detector calibration as well as geometric information—orbit position, spacecraft inertial attitude, and the mounting and pointing of the camera on the spacecraft. These kinds of information may be of scientific interest in their own right (for example, the trajectory of a spacecraft reveals something about the number, position, and masses of objects in the solar system), but if in a scientific application they are primarily used to analyze of other information objects such as images and spectra, these kinds of information are auxiliary data. Auxiliary data can be collected into a set of objects. The attributes, operations, and relationships for each type of auxiliary data object are highly dependent on the object's role in data analysis. With orbit/attitude/pointing information, for example, there may be attributes that indicate the inertial frame of reference (e.g., ecliptic and equinox of date) and there may be operations to return spacecraft position at a specific time.

Another key point arises from the requirement that the IIRM be applicable to an open system environment. In such an environment, it should be possible to devise software that can receive and manipulate new types of objects with little or no reprogramming To do so such software must have access to the metadata that describes the interface to each new object. A database of interface definitions for objects is sometimes called an Object Interface Repository (OIR) or an Object

Dictionary (OD); these are specific cases of a DED. Such a DED can identify the interface components—attributes, operations and relationships—for the known types of objects. The DED can also provide a formal definition of each of these components. A DED and the definitions within it can be considered objects called metadata objects. Transferring metadata objects from one DED to another or from a DED to an end user may require that the metadata objects be encapsulated for transport other kinds of objects, so that metadata objects may exist outside of the framework of a DED.

Given the complexities of scientific data, typical data requests may require the transfer of several types of primary objects (for example, some images and their associated image-intensity histograms), along with associated auxiliary objects, such as calibration files and orbit/attitude/pointing data, and metadata objects that describe each of these other kinds of objects. Thus mechanisms must be available for collecting other kinds of objects and encapsulating them during transport; such mechanisms are called container objects. Container objects may contain their own kinds of metadata: for example, they may provide a sort of table of contents that identifies and locates each object within a container.

Figure 4 provides a preliminary class hierarchy. Each downward arrow indicates a subtype relationship. For example, both Container Object and Data Object are subtypes of Object and they inherit all the methods of Object.

When applying the IIRM in the analysis of a data system or a data interchange methodology, seek to identify the types of objects that are used by the system. Examples of this analysis are given in the next section. Some data systems can be best described by modeling from the top (i.e., object layer) down, whereas others are better suited for modeling from the bottom (i.e., stream layer) up. Either a top-down or bottom-up approach may be used when applying the IIRM model.

| **Object Type** | **Image** |
|---|---|
| Description | An image represents a mapping of the intensity of electromagnetic radiation in two or three spatial dimensions. Digital images consist of a set of picture elements, or pixels, with the value of each pixel proportional to the intensity of light measured by the camera system within the areal extent of the pixel. |
| Supertype | Image is derived from type Array, which describes homogeneous multi-dimension data structures. Type Array is in turn a subtype of the most basic type called Object |
| Subtype | Subtypes of this type can be created to characterize images taken by specific camera systems. |
| Attributes | The following are the attributes—the visible characteristics—of images:<br>• Number of dimensions (2 or 3) in the image [positive integer numbers]<br>• Number of pixels in each dimension [positive integer number]<br>• Number of bits per pixel [positive integer number]<br>• Content [character string]<br>• Time that picture was taken [date/time]<br>• Exposure time [time]<br>• Wavelength or frequency range [real numbers]<br>• etc. |
| Operations | The following are the operations that can be performed on all images. These augment the set of operations that are inherited from the parent type Array. |
| Subsample | Create a new image consisting of a contiguous set of the pixels from an image. |
| Average | Create a new image by averaging a specified number of contiguous pixels from an image. |
| Generate Histogram | Create a Histogram object for which each element is the total number of pixels within an image with a given intensity value. |
| Relationships | The following are relationships involving image objects: |
| Calibration | This relationship relates an image to a characterization of the sensor that took the image. |
| Pointing | This relationship relates an image to where the camera is pointing. |

**Figure 3. Sample Type Description of an Image**

## V. Applying The Reference Model

In this section, the IIRM is used to characterize current data exchange methodologies as follows:

1. Identify the primary object types defined by the methodology at the object layer, along with the auxiliary, metadata, and container objects used.

2. Identify the primitive data types defined in the structure layer and the way the object-layer entities map to the primitive types in the structure layer

3. Identify the media and data exchange mechanisms supported at the stream layer.

The following data interchange methodologies are described here:

- Hierarchical Data Format (HDF)
- Planetary Data System (PDS)
- Standard Formatted Data Unit (SFDU)

Figure 5 summarizes the key characteristics of these methodologies.

### A. Hierarchical Data Format

The HDF was created by the National Center for Supercomputing Applications (NCSA) to provide access to common types of scientific data. An HDF is a self-describing file format that contains a set of tagged objects. NCSA provides a comprehensive library of routines in C and FORTRAN to create and to retrieve data from HDF files. In addition, there is a sizable body of applications software, both public domain and commercial, for accessing data in HDF format.

HDF has been selected as the baseline standard data format for the Earth Observing System Data and Information System (EOSDIS). Consequently, the HDF data model is undergoing significant evolution to provide high-level data types commonly used by scientists to model Earth-related phenomena. The following analysis is based on Version 3.3 of HDF, released in September 1993.

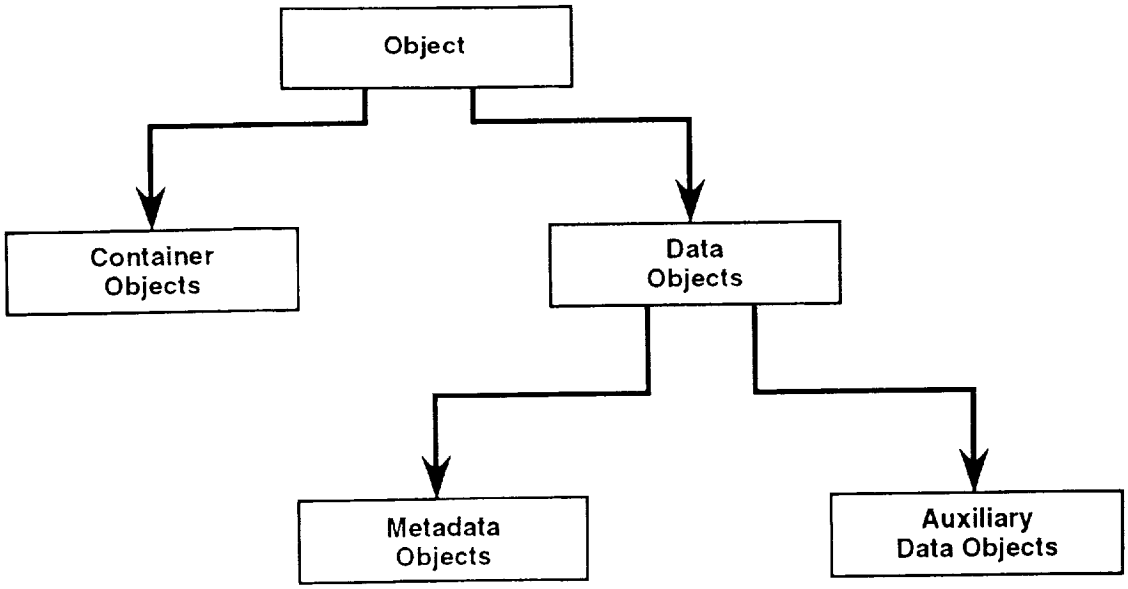



**Figure 4. IIRM Object Layer Class Hierarchy (Preliminary)**

| | PDS | HDF | SFDU |
|---|---|---|---|
| Stream Layer | • Requires file structure<br>• Uses FTP/DECNET or disk structure | • Requires direct access file structure | • Allows any level of service that supports conversion of bits to bytes |
| Structure Layer | • ODL labeled objects<br>• Machine dependent datatypes, IEEE datatypes | • Tagged record structure<br>• Machine dependent datatypes, IEEE datatypes | • Stream of Label-Value Objects<br>• Data Definition Language allows wide specification of primitive types and "record structures" |
| Object Layer | • Limited class hierarchy<br>• No methods defined other than attribute retrieval<br>• Data Objects<br>– Images<br>– Histograms<br>– Spectra<br>– Tables<br>• Container Objects<br>– Files<br>– Volumes<br>• Metadata Objects<br>– Catalog<br>– Data Entity Dictionary<br>• Auxiliary Data Objects<br>– SPICE Kernals<br>– Gazeteer Objects | • No current class hierarchy<br>• Formal Application Program Interface (API) for each data type<br>• Data objects<br>– Raster Images<br>– Palette<br>– Multidimensional Array (SDS)<br>– Tables (Vdata)<br>• Container Objects<br>– Vgroups<br>– Files<br>• Metadata Objects<br>– Annotation<br>– Attributes with SDS | • No current class hierarchy<br>• No methods defined other than object insertion/retrieval from containers<br>• Data Objects<br>– Application Data Objects<br>– Supplementary Data Objects<br>• Container Objects<br>– Exchange Data Units<br>– Application Data Units<br>– Description Data Units<br>• Metadata<br>– Data Description Packages<br>– Data Entity Dictionary Objects<br>– Catalog Attribute Objects<br>• Auxiliary Data Objects<br>– Supplementary Data Objects |

**Figure 5. Preliminary Descriptions of HDF, PDS, and SFDU Using IIRM**

*Object Layer*

HDF provides a set of Application Program Interfaces (APIs) through which all application data access must occur. The primary data objects within HDF are classified by the relevant API. These APIs are equivalent to defining the external interface (i.e. operations and relationships) of objects at the IIRM object layer in that they are independent of the internal implementation of the objects within HDF files. The APIs currently defined are:

- Raster Image API: Allows the user to store and access raster images and optional color palettes. Three optional forms of image compression are supported: JPEG, run-length encoding and IMCOMP compression.
- Palette API: Defines color tables for 8-bit raster image data.
- Scientific Data Set (SDS) API: Allows the storage and access of multidimensional arrays with specific attribute data. The interface provides the ability to slice an array and work with the resulting subset of the data.
- NetCDF API: Also allows storage and retrieval of multidimensional arrays. This API supports the netCDF data model, developed by the Unidata program of the University Corporation for Atmospheric Research, which is a richer data model than SDS. Additional features include an "unlimited" dimension and global and local attributes.
- Vdata API: Allows storage and retrieval of collections of data that can be viewed as record structures. This includes data meshes, polygonal data with connection information, packed data records, and sparse matrices.
- Vgroup API: Allows general hierarchical grouping of HDF objects.
- Annotation API: Allows labels and unstructured text to be associated with any HDF object or with an entire HDF file.

HDF does not support the concept of type hierarchies and formal inheritance. NCSA's commitment to backward compatibility with previous versions of HDF has led to some features that would probably be implemented differently if the system had been engineered to be object-oriented from the outset. For example, the NetCDF API is a pure superset of the SDS API, since these two APIs developed separately, the relationship between the SDS and NetCDF is not a true subclass/superclass relationship.

*Structure Layer*

The structure layer in HDF supports a standard set of primitive data types including real numbers (IEEE floating point), integer numbers (unsigned and signed 2's compliment), and character strings (big-endian byte ordering). In addition, HDF can store the machine-specific representation of reals, integers, and character strings for supported platforms.

The basic building block of an HDF file is the data object, which contains both data and information about the data. A data object has two parts: a 12-byte data descriptor (DD) and a data element. Figure 6 below illustrates two data objects.

A DD has four fields: a 16-bit tag, a 16-bit reference number, a 32-bit data offset, and a 32-bit data length. The tag of a DD tells what kind of data is contained in the corresponding data element. A tag and its associated reference number uniquely identify a data element within an HDF file.

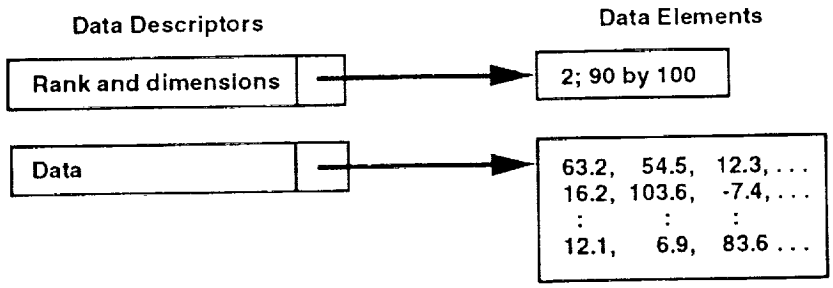



**Figure 6. Two HDF Data Objects**

DDs are stored in a linked list of blocks called data descriptor blocks, or DD blocks. The file header, DD blocks, and data elements appear in an HDF file in the following order:

- File header
- First DD block
- Data elements
- Additional DD blocks and data elements

*Stream Layer*

HDF depends on a stream layer that provides direct access capabilities. The tagged structure in the structure layer requires efficient seeking to specific locations in a single HDF file. HDF files may be stored or transmitted on sequential media, but they must be moved to direct access media before they are accessed.

## B. Planetary Data System

The PDS acquires, archives, and distributes much of the data that NASA collects on bodies in this solar system other than Earth, including planets, comets, and asteroids. When the prototype of the PDS began in 1983, it inherited substantial amounts of existing planetary science data in many different formats. It was not practical to reformat all of those data into a standard representation. therefore, the PDS developed a methodology for describing data in a way that both human users and computers could identify and understand the content of a data file or stream. This methodology describes data objects that are set forth in a language called the Object Description Language (ODL). A label (typically called a PDS label) encoded in ODL is attached to every data file or data stream that flows into or out of the PDS to identify the objects in the file or stream. Gradually the PDS evolved a relatively comprehensive set of standard objects and data providers are encouraged, even required, to submit data in a format that is consistent with the standard objects definitions. The standard objects are defined through the Planetary Science Data Dictionary (PSDD).

*Object Layer*

PDS object model is still in development and the description below includes some new facets to the model that are currently being adopted and formalized through the PSDD.

*Primary Objects*

The two simplest types of objects, called Element and Bit Element, can hold a single instance of a primitive data type. The two are similar, but the Bit Element type can handle primitive data that are not aligned on byte boundaries. There are two general aggregation objects—Array and Collection that hold element objects. An array is homogeneous—all elements must have the same underlying primitive data type—while the collection can be heterogeneous, which makes it analogous to the record or structure data type found in many data models.

The PDS also provides several primary data objects that are specialized for space science applications. These include:

- Histogram
- Image
- Table
- Spectrum

PDS does not use the inheritance mechanism to define subtypes of these objects. Instead, each of these object classes provides all the attributes needed to describe nearly all instances of the object. For example, all images are objects of type Image. Figure 7 describes the image object.

Three aspects of the PDS object model, as illustrated above for images, deserve elaboration. First, there are only a few PDS objects that have formal subtypes. Specifically, there are several important subtypes of the Table object, including a Palette object to hold color table information for image display and a Series object to hold time series (or similarly organized) data.

Second, no currently no formal operations defined for images or any other type of PDS object exist. There are several reasons for this omission, including the difficulty in agreeing on what the standard operations should be and neither the PSDD nor the ODL used for PDS labels currently have the syntax or semantics necessary to describe operations. A unique problem with defining standard operations arises when PDS object types like Image are designed to cover a vast extent of object instances, with no use of subtyping to provide specialization. This means that some PDS object types are so complex that there is no single piece of software that can account for all the possible permutations of their optional attributes. For example, no single piece of software can handle all instances of PDS images.

Third, there are no formal relationships defined for PDS objects, except for the limited use of supertype/subtype as noted above and a simple relationship called Contains indicates an object holds other types of objects. The most notable example of the Contains relationship is the Table object, which contains one or more Column type objects. In general, if two or more instances of PDS objects are related—for example, an image and its associated histogram together within a file—this relationship is only implicitly indicated by the objects that are contained within the same file and described together by the same PDS label.

*Auxiliary Objects*

The planetary community has developed a standard representation for orbit/attitude and pointing auxiliary data. This standard is called SPICE, where the letters of the acronym stand for the kinds of information that are handled: spacecraft, planets, instruments, coordinates,

and events. The Navigation and Ancillary Information Facility (NAIF) at the Jet Propulsion Laboratory (JPL) provides auxiliary data to projects in SPICE format. The NAIF also maintains the SPICE standard and provides an extensive Fortran library of operations to support SPICE-encoded data. SPICE files (called SPICE kernels) are considered to be PDS objects and their attributes are defined through the PSDD.

| **Object Type:** | **Image** |
|---|---|
| Description: | An image represents a mapping of the intensity of electromagnetic radiation in two or three spatial dimensions. Digital images consist of a set of picture elements, or pixels, with the value of each pixel proportional to the intensity of light measured by the camera system within the area extent of the pixel. |
| Supertype: | PDS has no formal inheritance mechanism, hence there is no formal supertype for type Image. |
| Subtype: | There are no formal subtypes since there is no formal inheritance mechanism. In practice there are numerous subtypes of images, since the standard image format produced by each of the cameras abroad a planetary spacecraft can be considered to be a subtype of type Image |
| Attributes: | The following attributes are mandatory and must appear in each description of an image object instance:<br>• Lines—number of scan lines in image<br>• Line_Samples—number of scan lines in image<br>• Sample_Type—Type of primitive data that makes up a pixel of the image<br>• Sample_Bits—The length of a pixel. There are also a large number of optional attributes which may or may not appear in a description for an image object instance, depending upon whether or not they are needed (if omitted, they each have a default value). A representative set of the optional attributes for Image are given below:<br><br>There are also a large number of optional attributes that may or may not appear in a description for an image object instance, depending on whether or not they are needed (if omitted, they each have a default value). A representative set of the optional attributes for Image are given below:<br>• Bands—The number of spectral bands in an image<br>• Band_Storage_Type—Method used to interleave spectral bands in a multi-spectral image<br>• Encoding_Type—The method used to compress an image, if any<br>• Line_Prefix_Bytes—The number of bytes at the beginning of a scan line that contain non-image data (for example, gain information or timing data)<br>• Line_Suffix_Bytes—The number of bytes at the end of a scan line that contain non-image data |
| Operations: | The PDS does not formally define operations upon objects. |
| Relationships: | There are no formal relationships defined for Image objects. |

**Figure 7. Description of PDS Image Object Type**

Another type of PDS auxiliary data is the Gazeteer object, which is a subtype of the Table object that provides information about geographical features on planets and satellites. For example, it provides the name of a feature or region, the body on which it is found, and its coordinates on the body.

*Metadata Objects*

The PDS defines a set of metadata object classes called Catalog Objects. They are used primarily to provide a template for data providers who are supplying information to be placed into the PDS catalog of data holdings. Some catalog objects are also used to augment the standard attributes of data objects. A prime example is the Map Projection catalog object, which provides a set of attributes that define a map projection. Frequently the raw images from planetary spacecraft are processed by mapping their pixels onto a standard map projection grid. When an object of this kind is created, a Map Projection catalog object is placed within the Image object in a PDS label to describe the map characteristics of the data. Users can correlate each pixel of the image with its location on the planet from information from the Map Projection object.

*Container Objects*

The PDS has several objects that serve to collect other objects. The most important is the File object, since most PDS data are transferred within files. Since much of the data that the PDS distributes is on volume-oriented media like CD-ROM, there is also a Volume object to provide information on the organization of a collection of files.

PDS container objects often have their own metadata. There is a Header object, which defines the headers that in turn describe the contents of data files. Aside from the standard PDS labels, this includes the VICAR labels found on many planetary images and the FITS headers found on many planetary datasets derived from observations with earth-based telescopes.

*Structure Layer*

The PDS has a fairly ordinary set of primitive scalar types: character strings, integer, and real numbers, enumeration types. It also uses the CCSDS format dates and times, allowing these to be considered primitive types as well.

There is no single required representation for primitive types. It is the instantiation of a primitive type as an Element type object, or as a component of some other kind of object (like a pixel of an image), that determines its format. Thus primitive types like numeric values can be represented in nearly any computer's native format. The PDS label that describes a data object provides information on the encoding of the primitive data types within the object. For example, a PDS label will identify whether or not the real number values that make up a histogram object are encoded in VAX format, IEEE format, or another type of format.

There is no separate data definition language for PDS-labelled data, because the PDS labels contain information needed to understand the structure layer, A PDS label does not as a rule provide a complete structure layer mapping: it does not rigorously establish the position of every data item in the object. Users have to rely upon numerous implicit rules to map from the PDS label's description of objects to the underlying representation of those objects within the structure layer.

*Stream Layer*

Small amounts of data are sometime provided to users over the NASA Science Internet. Typically FTP or DECNET file copy is used to transfer files over the network. Larger quantities of data are typically provided to users on CD-ROM. There are many CD-ROM titles that adhere to PDS standards. These disks adhere to the ISO-9660 standard. There are currently no specific stream layer services provided by the PDS to access data files in a way that is transparent of the medium of transport.

## C. Standard Formatted Data Units

The CCSDS Panel 2 has been developing, adapting, and adopting standards to improve information interchange within and among space agencies. CCSDS standard recommendations have been developed in support of a methodology called SFDUs. Briefly, this methodology involves the association of a small label with a collection of data values, forming a labeled value object (LVO), and the incorporation within the label of a globally unique identifier (i.e., Authority and Description Identifier, or ADID) of a description of the data values. This description may be a CCSDS Panel 2 standard and thus be found in a formal CCSDS recommendation document, or it may be defined by a user and be found at a Control Authority Office (CAO) set up by a participating agency conforming to the CCSDS standard titled "Control Authority Procedures." The primary function of a CAO is to register, archive, and disseminate data descriptions in response to user requests. These descriptions may themselves be composed of several labeled objects, including a formal (computer interpretable) description of the format of the data values, a text description of the mission and instrumentation involved in the creation of the data values, and software that may be used to obtain particular services from the data values. As such, these description LVOs may also be packaged with the data LVOs to form a self-describing data package.

*Stream Layer*

The SFDU standards assume the existence of stream layer services such as those provided by the volume/directory file system on a CD-ROM, the sequence of files on an ISO/ANSI standard labeled magnetic tape, and FTP for network file transfer. The provision of a sequential byte (8-bit) stream is the minimum requirement of the SFDU standards, while the use of named (e.g., directory/file names) byte streams permits the construction of sequences of labeled data objects that cross multiple files on random access media. This functionality is described in the Structure Layer.

*Structure Layer*

The standard titled "SFDU Structure and Construction Rules" is the primary CCSDS Panel 2 standard that interfaces with stream layer services. It defines an SFDU 20-byte label to support three primary functions:

1. Provide mechanisms to determine the end of a sequence of data values (i.e., encapsulate the data values) associated with the label

2. Provide a code which gives a general classification (e.g., data, data description package, supplementary data) to the encapsulated data values

3. Provide a globally unique identifier of a description (e.g., data description package) of the encapsulated data values. It also defines a number of standard descriptions and assigns globally unique 8-character standard identifiers (e.g., "CCSD0001") to them.

Application of this standard to the stream layer converts the byte stream view into a view of a sequence of hierarchically organized labeled value objects. This sequence may span multiple files on both sequential and random access media. One or more such sequences may be defined on a physical volume, or within a single file. There is no explicit provision for crossing multiple physical volumes with a single sequence, but it is possible if this is supported by the stream layer. It should be noted that the standard can be applied in such a way that many files are not required to contain labels. Thus the standard can also be applied to pre-existing data streams and to files conforming to other standards.

The labeled value objects at the lowest level of the hierarchy have a content that appears as a sequence of bytes from the stream layer. The structure layer function of interpreting this sequence of bytes into a sequence of primitive datatypes (e.g., integers, characters, and reals) is accomplished by interpretation of the Data Description Record (DDR) found within the Data

Description Package (DDP) identified in the label. This linkage of information is illustrated in Figure 8.

The DDR can be expressed in a number of standard languages that have been documented in CCSDS standards. Currently these include "ASCII Encoded English (CCSD0002)", "Parameter Value Language (CCSD0006)", and the draft standard "Enhanced Ada Subset (EAST)." The level of language-related automated support for access to the labeled value object depends on the language selected and ranges from presentation (e.g., ASCII/English) of a text description of the record structure(s) within the value to full parsing of record structures (e.g., EAST). Alternative support may be obtained from software associated with the particular ADID. This software may be provided as an additional object within the DDP.

DDPs are archived in a CAO so that any DDPs not present in the data stream may be obtained from the CAO. DDPs are expected to provide a complete description of the values whose labels contain their ADID, and in addition to the DDR which supports the structure layer function, they are likely to include a DED object and other semantics which may be used to support object layer services as described in the next section.

*Object Layer*

The SFDU standards provide a very general mechanism for representing and transmitting data objects. The SFDU standards do not currently provide a fully object-oriented approach: there is no class hierarchy; nor are methods defined, other than services for insertion and retrieval of data from containers. But SFDUs can be used to encapsulate data objects complete with their attributes and methods. SFDUs also provide container objects for combining collections of primary objects with the auxiliary data and metadata needed to interpret them. Thus the SFDU concept is one of a very few data interchange mechanisms that are designed to encapsulate and transmit all of the kinds of information contained in a scientific data system, whether object-oriented or not.

*Primary Objects*

Unlike the PDS and HDF methodologies described above, there are no specific primary data objects in the SFDU concept. Instead the SFDU standards provide a general object class called an Application Data Object (ADO). (Each SFDU object class has a one-letter identifier and an ADO is also called an I class object. As described in the structure layer discussion, the ADID in the label points to a DDP that fully describes the LVO. The Data Entity Dictionary (DED) with the DDP gives all the attribute names for the LVO type. In the future the DED will also contain relationship information about the LVO type. The DED is further described later in this section. For example, a scientist can use the ADID of an ADO to determine whether the data in the SFDU is an image, map, spectrum, or whatever, and to tell whether the object is the FITS format, PDS format, or some other format.

*Auxiliary Objects*

Since the SFDU standards have been developed with scientific applications in mind, there is a specific class of SFDU called the Supplementary Data Object (SDO) (or S class) that is used to contain auxiliary data. For example, if a spectrum is transferred in an ADO the calibration information for the spectrum can be placed into a SDO and the S class supplemental SFDU can then be transferred with the I class SFDU that holds the spectrum. As with ADOs a SDO may contain virtually any kind of data in any format desired, and the ADID for the SDO provides the key to determining the content and format of the object.

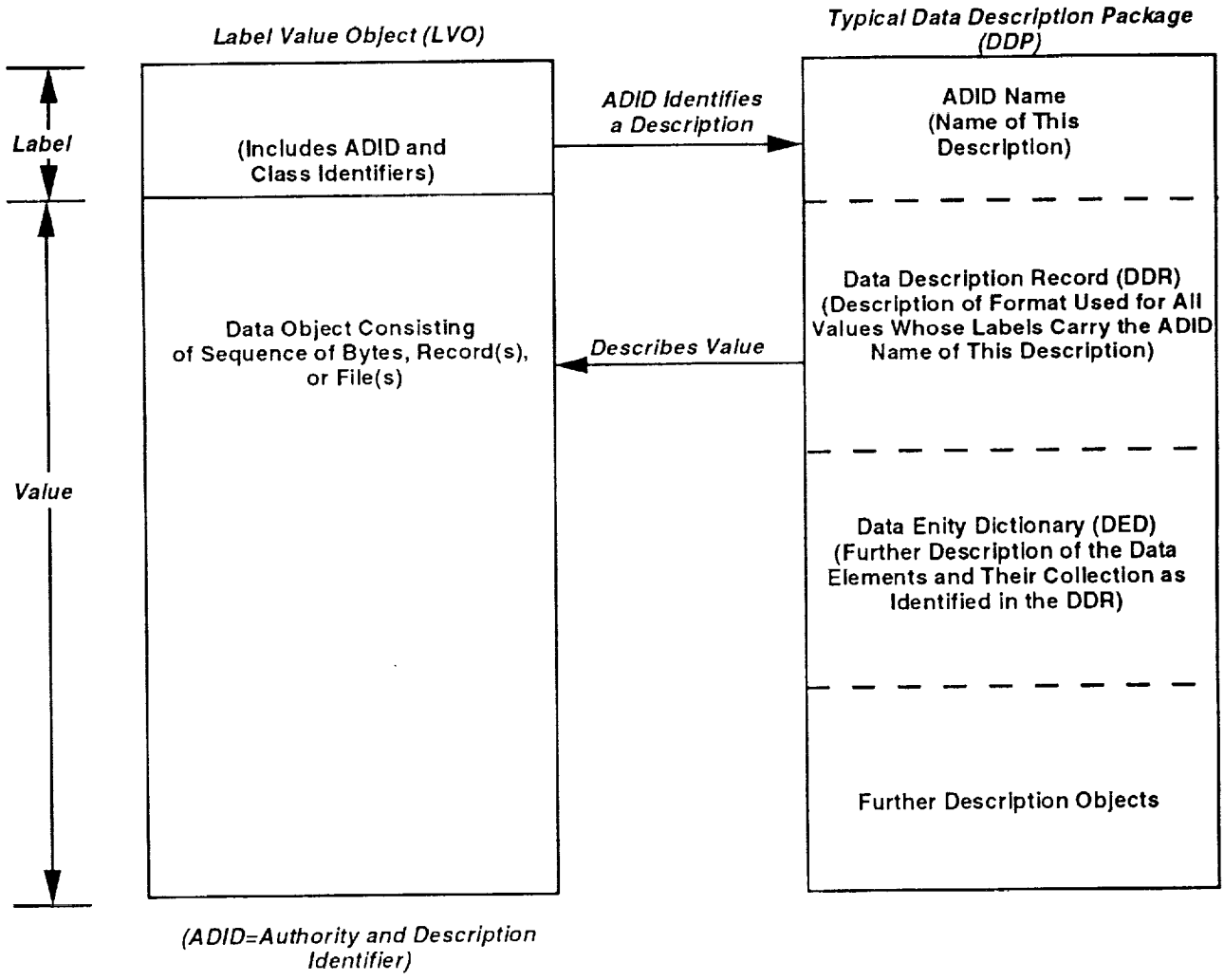



Figure 8. SFDU Label Value Object and Its Description

*Metadata Objects*

An important aspect of the SFDU concept is the ability to encapsulate metadata as well as data. There are three types of metadata objects defined by the SFDU standards:

- DDO (or D class)—These objects are used to hold the data descriptions that map an SFDU object—for example, an ADO—into the structure layer. The definition is given in a DDL. A DDO provides the mapping for a specific instance of an SFDU object. For example, a DDO may provide the data definition for a specific data table. Other data tables may have very different representations and hence would have their own DDO to describe them.

- DED Objects (or E class)—These objects are used to hold descriptions from a DED. The descriptions define types of objects rather than specific object instances. They can also define the terms used in object type definitions. For example, if an object has an attribute called Length, a DED object can specify the minimum and maximum values allowed for Length. The CCSDS is currently completing work on a standard representation for the information within DED objects. This standard representation uses the Parameter Value Language (PVL) to encode the DED information.

- Catalog Attribute Object (CAO) (or K class)—Data systems often maintain a catalog—a database that describes the data held within the system. The CAO can be used to transfer information to and from a catalog or a similar database. When a data system transfers applications data to a user it will often provide the pertinent catalog information or other attributes for the transferred data objects. The CAO supports this by holding the attributes of a set of ADO wrapped within a container SFDU. As with other types of SFDUs, the form and content of a CAO are not constrained by the SFDU standards. The information might be given in tabular format, where the columns are the attributes of the objects that are being described and each row of the table contains all the attributes for one data object. Alternatively, catalog attribute information can be given using PVL or a similar keyword/value notation, where there is a keyword/value pair for each attribute of each object.

*Container Objects*

The SFDU methodology provides three types of container objects:

- Exchange Data Units (or Z class)—These objects are the most general encapsulation mechanism for SFDUs. An Exchange Data Unit (EDU) can hold essentially any combination of the SFDU objects described in this section, including other EDUs.

- Applications Data Units (or U class)—These container objects can be used to aggregate a set of related ADOs and SDOs. An Applications Data Unit (ADU) may include a CAO that describes the other objects in the container. An ADU can also hold other ADUs.

- Description Data Units (or F class)—These container objects can be used to aggregate DDO, DED Objects, and any other metadata objects.

## VI. Relationships With Other Reference Models

This section provides a comparison of the IIRM and two other models: the IEEE mass storage system reference model and the familiar OSI reference model for communications.

### A. IEEE Mass Storage System Reference Model

Information on the Mass Storage System (MSS) Reference Model (RM) was obtained from the paper "Mass Storage System Reference Model: Version 4", which was published in the

proceedings of the Goddard Conference on Mass Storage Systems and Technologies, Volume 1, 1992.

The MSS RM establishes a client server environment to provide access to a (potentially) distributed system that accepts and returns named Bitfiles. This storage model addresses data interchange over time (i.e., storage), but not over space (i.e., an instance of a MSS is not moved to a new location). In contrast, the IIRM addresses data interchange over both time and space. Since data moved over time and space may end up stored in a MSS, it is useful to perform a mapping between the IIRM and the MSS RM.

The MSS RM named Bitfiles appear to be virtually identical to the named bit streams that the IIRM Stream Layer provides to the Structure Layer. The one exception is that the MSS RM Bitfiles also have a set of attributes such as file creation date, file owner, etc. Such attributes have not been called out explicitly in the IIRM, although they must exist and be accessible to the Structure and Object Layers. In other words, the entire MSS RM addresses functionality covered in the IIRM Stream Layer.

## B. ISO Open Systems Interconnect Reference Model

The ISO OSI RM addresses the interchange of information over time and space using electronic networks. In contrast, the IIRM applies to both networks and physical media as interchange mechanisms.

The OSI model is a seven-layer model, which makes use of the information hiding principle of layers. The functionality of layers one through five (Physical through Session Layers) is to establish a connection between two communicating nodes and effect the transfer of data bits between them. This is similar to the functionality of the IIRM Stream Layer, although the name capability associated with this bit stream as output from the Session Layer appears to depend on the particular protocol standards defined for this layer.

The sixth layer of the OSI model, called the Presentation Layer, is intended to convert a bit stream into recognizable data types. While it is hard to determine from the OSI model itself the extent of this functionality, a clearer picture emerges from an examination of the ASN.1 protocol defined for this layer. For this layer, the functionality is similar to the IIRM Structure Layer, which includes the identification of common data types, and their aggregation into named structures.

The seventh, and top, layer of the OSI model, called the Application Layer, is intended to provide user applications with a number of common services. The types of services to be provided, as shown by some of the protocols defined for this layer, include electronic mail, a directory service, and a file transfer service. There is considerable parallel with the IIRM Object Layer, as these layers are intended to provide user applications with a service view of the underlying data structures. Differences include the object orientation of this layer in the IIRM (although an object view of the Application Layer should be possible) and the IIRM focus on understanding scientific data by focusing on identifying objects of scientific interest. The fact that the OSI model addresses network functionality leads to identifying Application Layer services for what are highly common network service needs (e.g., electronic mail). The types of objects (and their services) being addressed by the IIRM Object Layer could, in principle through standardization, enter an expanded OSI Application Layer.

The OSI Application Layer file transfer service, differs from the IIRM file transfers that are handled within the Stream Layer. This is not a contradiction to the mapping between the models just described. The functionality requested from a file system in the IIRM is to provide named bit streams. The functionality provided by an FTAM file transfer in the OSI Application Layer includes the recognition of common data types. The IIRM views the recognition of data types, and the provision of services from them, are more usefully obtained from an object view, not from a file view. Mechanisms that take this object view could use an FTAM service, in principle, in either of two ways: 1) by not using the capability of ASN.1 to

describe the data types, and instead describing the file content as a bit string, thereby reducing FTAM to simply providing named bit streams, or 2) by using FTAM to include the functionality of the IIRM Structure Layer, and then providing an object view of the FTAM file content. These variations in mapping reflect options on the level of services requested, and the ways they may be combined.

## VII. Summary And Future Plans

The IIRM provides a basis for comparing data systems and data interchange methodologies at three levels: as represented by a stream of bits (the stream layer); as a stream of primitive elements (the structure layer); and as a collection of objects. By applying this model similarities and differences can be called out in the systems that are used for scientific data interchange and data analysis. The object layer of the model is unique as it accounts for primary scientific data like images and spectra that require auxiliary data for interpretation, metadata for description, and containers for encapsulation. The IIRM allows the user to describe how all these elements fit together for a specific data system or application.

In the future the IIRM will be refined and the model applied to data interchange systems other than the three that were analyzed in this paper. This analysis should permit data system designers and implementers to improve the compatibility and uniformity of information interchange where practical. This may, for example, make it possible for a scientist to compare spectra of the Earth's atmosphere with those from other planets, even though the spectra may be retrieved from different data systems in quite different formats. Capabilities like these will be especially important if we want to reduce the burden on scientists from dealing with the form rather than the content of scientific data.

# Preserving Electronic Records: Not The Easiest Task

**Fynnette Eaton**
National Archives and Records Administration
7th and Pennsylvania Avenue, N.W.
Washington, DC 20408
fez@cu.nih.gov

The National Archives and Records Administration has had a program for accessioning, describing, preserving and providing reference service to the electronic records (machine-readable records) created by Federal agencies for more than twenty years. Although there have been many changes in the name of the office, its basic mission has remained the same: to preserve and make available those records created by Federal agencies that the National Archives has determined to have value beyond the short-term need of the originating agency. A phrase that I once coined for a preservation conference still applies: the National Archives, when it decides to accept the transfer of records into its custody, is committing itself to preserving these records for perpetuity.

Most people think of the National Archives as the keeper of the Constitution and the Declaration of Independence. Even the most experienced researchers are unaware of the growing number of files that have been accessioned by the National Archives in electronic format. Since the creation of the Center for Electronic Records in 1988, the number of files transferred has literally skyrocketed: in 1988 the Archives received 150 files from Federal agencies; in fiscal year 1991, the number was 1500, an order of magnitude increase in two years. This number jumped again this last year to 8700 files. Unfortunately, the number of files that we should be receiving completely overshadows our accomplishments. Beginning in the 1970's, NARA signed agreements with Federal agencies specifying that at certain times or under certain conditions these agencies would transfer files to the National Archives. The Center in 1990/91 developed a database to determine how many of these files have been transferred. As you can see on the graph, we have received very little: less than 10% of what has been anticipated. The second chart is even more daunting. It combines what we knowingly have not received from agencies with projected numbers of new files currently in use at agencies that warrant permanent retention by the National Archives. The National Archives asked the National Academy of Public Administration to perform a study, identifying the most important current databases in use in the Federal government, and to provide s in a study commissioned by the National Archives in 1990 and 1991 identified as candidates for preservation by the National Archives. This second graph has four components. First, almost a blip on the chart are the files received. The second component is what has been projected that we knowingly should have received, based on schedules developed with Federal agencies. The third area is the continuation of series beyond the first file, that should have been received, but have not. Only the fourth component are the files identified by NAPA that NARA should target for transfer.

Consequently, although the Center has successfully increased the number of files being transferred, we have barely made a dent in the ensuring the preservation of Federal records that have clearly been designated as important enough to be transferred to the National Archives.

The successful and almost ubiquitous use of computers for more kinds of record keeping activities in ever increasing quantities poses serious and extensive problems for the National Archives and Records Administration. We share many of these problems with organizations represented in this audience.

The problems posed by electronic record keeping include fragility of the media, rapid obsolescence and incompatibilities between makes and models, and even between different releases of the same product. The magnetic media most commonly used to store electronic

records off-line, open reel tape or tape cartridge, are physically fragile and easily erasable and reusable, presenting a serious challenge to the preservation of electronic records.

The lack of standardization coupled with hardware and software dependencies of electronic records means that even if we can identify and physically preserve the records, we may not be able to access the data they contain. And, finally, rapid and unending change in computer technology exacerbates these problems.

The purpose of my paper is to discuss the types of problems we have encountered in trying to preserve the electronic files at the National Archives. But, before I provide specific examples of our problems, I want to quickly acknowledge the fact that my institution's holdings are tiny in comparison to the size of most of this audience's. The amount of information currently in our custody is roughly 1 Terabyte. But NARA makes up for size in the diversity of files that it has accessioned. As of October 1st, 1993, the Center has electronic files from 91 agencies: 19,278 data files in a wide range of formats. For example, although our regulations clearly state that agencies should transfer files that are hardware and software dependent in either ASCII or EBCDIC character code, we have in our holdings files that are BCD binary coded decimal, binary data, EBCDIC and binary data, EBCPARK. EBCZON, Multipunch, NIPS (National Military Command Information Processing System), SAS, and SPSS.

I would characterize our problems as falling into three categories. First there is the very serious problem with metadata. Secondly, the hardware and software dependencies of files, which prohibit us from being able to properly process and preserve the information and thirdly, the medium itself. I would like to spend time on each of these categories.

First, documentation. I would like to use a true anecdote to bring into clearer focus how wide-ranging the problems relating to metadata can be. There was one accession, which we could not properly process, because the documentation was only half-way complete. The agency had made an incomplete copy of the documentation, providing us with only the even-numbered pages, because they had failed to make a proper two-sided copy of the documentation. Fortunately, we were able to secure a full copy of this information, before it was dispersed. Although this appears to be a silly example, it illustrates why it often takes archivists within the Center between six months and a year to secure all the necessary documentation for files that are being accessioned into the National Archives.

Unfortunately, the Center is not always successful in receiving sufficiently complete documentation to ensure a proper understanding of the file. There are numerous examples of files that have been rejected for transfer because documentation was simply too incomplete. We have rejected surveys of taxpayer attitudes, trade statistics, military personnel statistics, and a collection of Vietnam War data, because the archivists could not make sense of the data, due to the lack of documentation.

The likelihood of encountering these problems increases dramatically if the files being transferred are either program files that have not been prepared for distribution or older files that have been in offsite storage for some time. If a file is active, it is easier to find reliable documentation. If the file has been distributed to the public, documentation would have been prepared for those ordering the files.

Yet there is also a middle ground, where NARA has received documentation, but it is incomplete and causes problems when trying to provide reference on these files. One file, the Combat Area Casualty file, which is one of our most frequently requested files, provides a good example of these types of problems in documentation.

The Combat Area Casualty file was maintained by the Department of Defense. The system recorded those people, either military or civilian who were wounded, captured or killed in the conflict in southeast Asia. One of the problems with the documentation appears to have been caused by clerical error. In preparing the list of codes for types of injury, the person creating the documentation skipped one of the letters of the alphabet that represented a code for a type

of injury. Thus there were records that, according to the documentation prepared by the office using the file, were invalid. A second problem encountered was the replacement of one set of codes with another, without identifying this change in the documentation. This is a common problem with files that are frequently updated. The documentation must also be updated as codes are modified or replaced. Otherwise the documentation will not accurately reflect the information in the file. This is a serious problem for the National Archives, because we do not actively maintain these files: we simply preserve the information for access by researchers or the agency interested in the file.

The Center continues to confront the problem of hardware and software dependencies when working with agencies to transfer files to the National Archives. Some of our knottiest problems come from records created on systems developed during the Vietnam War. Perhaps the best example is the Filesearch IV system, which was used by the Combined Document Exploitation Center, to film captured Vietnamese documents. In 1977 the National Archives acquired a 106 reels of motion picture film which had 16 mm images superimposed on 35 mm motion picture film, with the sound track used to record a digitized index to the images. There was a major problem. The FileSearch retrieval equipment had ceased to be manufactured in 1969. There have been several efforts to find a way to get access to the index, but there are few systems still in existence (we have one that has never been operational) and we found that there were two incompatible coding structures. Thus, the index to the 3 million images and 1 million documents is inaccessible.

The issue of hardware dependency is still with us, even today. One agency with financial data was interested in transferring the records to us, but this agency had used "Tall Grass" tape drives to create the file and their drives had not been used for many years. When one of our computer systems analysts visited the site, he reported that is was unlikely that the drives would function, and that the data would be inaccessible unless placed on a tall grass tape drive and then outputted to another format.

A third example is provided by a large statistical agency that has permanent files that were created in a proprietary system with both hardware and software dependencies, that they have not been able to successfully convert into a format that could be used outside of that agency. We have been working with this agency for more than 15 years on this issue, yet we have not yet received a file from this system. (Census Input/Output [CENIO ]).

Our major problem with software dependencies is found with military files created during the Vietnam War. The military used NIPS software, (National Military Command Information Processing System) to compact files that had numerous repeating fields. Unfortunately, at some point in the 1970's IBM ceased to support this software, and since no one else was a major user of this system, it disappeared. The records, however, are still with us. Currently the Center has approximately 150 files that are in this NIPS format. In this case, we are hopeful that the functionality of relational databases will provide the tools necessary to decode the files recorded in this format.

Some additional examples of software dependent files that we are unable to provide normal access are the National Economic Commission Computer Budget Gameor simulation, which was recorded in a spreadsheet and implemented via macros; the United States Railray Administrations Tracking/Document Management System, which was created in Basis and transferred to us in system backup format. These are problems we currently have. If we did not have the regulations requiring files to be hardware and software independent, this list would be unending.

The last category of problems, fragility of the medium, are the ones that my branch deals with since we have the responsibility for making preservation copies of files transferred by Federal agencies to the National Archives. We often find, particularly with older tapes, that we are unable to read the files because of excessive data checks. We have several accessions of older files that we have not been able to copy because of this problem. Since this is probably the only extant copy of the file, we are in the uncomfortable situation of either rejecting outright these

files, or trying to copy as much of the information as possible onto newer media. The problems encountered in trying to copy older tapes is one of the best justifications for working with agencies to provide copies of data that are to be transferred while they are still in use, so if there are problems, the agency can easily supply another copy of the information

Another problem we are finding with older files is the condition that we refer to as "sticky" tapes. On a few occasions we have been notified by the computer center we use that one of our tapes has stuck to their tape drive. So far they have not threatened to kill us, but we must carefully monitor the conditions of the older tapes that we send to the computer center for processing. The National Media Lab has been very helpful in providing us with advice and training in how to screen older tapes.

The fragility of the media has been well documented. Storage conditions are always cited as a major reason for data loss in magnetic recordings. In many cases, the Center has no knowledge as to the storage conditions of the tapes before they are sent to the National Archives for copying. A couple of years ago we received some tapes from Wright Patterson AFB that had files relating to the Vietnam War. The Center has not been able to process most of the records transferred from the Department of Defense on the Vietnam war because the files are in the NIPS format and there is only limited documentation for most of these files. We had hoped to gain additional information about our records from these recently received files, but we discovered that the tapes had not been stored properly: there was the possibility of fungus on the tape and most of these tapes were created before 1975 and exhibited tendencies of sticking.

Again, this is a more dramatic example than what we normally confront, but it helps to convey the wide range of problems facing us as we receive files from the very large universe that is the Federal government.

The Center has techniques for dealing with these problems. For example, to cope with lack of standardization, NARA has issued regulations requiring that any electronic files scheduled for transfer to the National Archives be written in a simple format that is not dependent on any specific hardware or software. We are attempting to find more sophisticated solutions to compatibility by promoting standardization and collaborating with other organizations such as the National Institute of Standards and Technology, the National Environmental Satellite Data and Information Service and the Federal Interagency Coordinating Committee on Digital Cartography. To overcome the fragility of magnetic media, we have implemented controlled storage and testing procedures, and we require other agencies which retain permanently valuable electronic records to do the same. We actively encourage agencies to give us copies of permanently valuable records at the earliest opportunity, and we return the agency's tapes to them for possible reuse. But these are means of coping rather then solving the problems. So the task of preserving electronic records for future generations is not the easiest task: but it is an essential one. That is why we, at the National Archives, look to other agencies and experts in the field of magnetic recording to help us confront the problems of electronic records. By working together we might actually find some solutions to the problems confronting us today. Thank you.

# Invited Panel: User Experience with Storage and Distribution Media

MR SAWYER: The panel chairman for today is Jim Berry. He received his BA from the University of Maryland and his master's from American University. He also has an MBA from the University of Southern California. He's held various positions at the National Security Agency, the Department of Agriculture, and the Office of Personnel Management.

Currently, he's the user representative to a processing office which supports one of the major operations groups at NSA. His areas of specialization are massively parallel computing, high speed networking, and mass storage.

Jim Berry.

MR BERRY: This afternoon, each member of the panel is going to make a very short presentation, and then we'll open it up with questions. I invite questions from the floor. Each person is now going to explain to you what they do at their respective organizations.

The first speaker is Lee Bodden. Lee is the Hughes STX manager of the Goddard Space Flight Center Version 0 Distributed Active Archives Center. He has been one of the system engineers for the DAAC since its inception in 1991.

Lee?

MR BODDEN: (Off microphone.)

Thank you. My name is Lee Bodden, as Jim said earlier. I'm with the Goddard Space Flight Center V Zero DAAC. We've been asked to give a short two-minute presentation on what the DAAC is; I understand that there will be other talks on the Goddard DAAC later on.

We're just getting rolling as an on-line, active archive. We -- right now we've got about two terabytes of data on line, but we anticipate growing to twenty terabytes minimum. That's just what we're looking at now. It may be more than that, because there are projects that are coming at us from all directions.

Our computers, we went with SGIs. The mass storage hardware for our archive -- we're going to be talking about Cygnet optical jukeboxes with 12-inch optical platters; and we've got the Metrum RSS600, which is an updated VHS cassette library system, which has been performing very well for us. We've only been using it about six months, but we're very happy with it.

Our storage medium: for the Cygnet, the corresponding media is the optical disk, and the Metrum uses VHS cassettes with a capacity of 14.5 gigabytes per cassette. The distribution that we're currently supporting that we're holding up to is what you see here. There are other types of media available upon special request.

And, as I say, we're just getting going. Our ingest volume right now is 25 gigabytes per month, and our monthly distribution is 125 gigabytes. That's going to increase. By fiscal year 1997 we are looking at perhaps getting up to 60 gigabytes distribution per day. We have a long way to go from where we are and where we're going to be.

Just very quickly, this is an architecture of our system, and these two rectangles, sort of in the middle there, those are our computers with the peripherals surrounding what we call our DADS (Data Archival and Distribution System) systems. So, we've got one computer dedicated to running the archives. Then we have another computer, our second one, which is dedicated to our information management system (IMS). This system is where the users will log into, and users, using our IMS, will be able to browse the summary records, which we call metadata, of all our current data holdings, and select the data.

Once selected, the IMS/DADS interface is automated so that the IMS will take that request from the user, feed it into the DADS and from there on --we are slowly automating the system-- so the users will not see anything beyond that, except for the fact that their request has been filled onto one of these different types of media, sent back to them. We are supporting on-line FTP distribution of the data for limited amounts of data.

So that's the situation that we have at the Goddard DAAC.

MR BERRY: Thank you. The next person on the panel is Richard Davis, who is the data administrator and records officer of the National Oceanographic and Atmospheric Administration's National Climatic Data Center in Asheville, North Carolina. He's been in government service for 47 years -- I didn't know anybody had been in that long -- in the field of meteorology and climatology.

He's responsible for the management of 50,000 cubic feet of manuscript records, 1.3 million microfiche, and over 100,000 reels of magnetic tape, which are depicted on his slides. He's also currently the project manager for the receipt and archiving of data from the new Doppler radars (NEXRAD) that will generate in excess of 88 terabytes of data per year.

Dick?

MR DAVIS: (Off microphone.)

Thank you. Can you hear that all right? At the National Climatic Data Center, we receive climatological and meteorological information from the National Weather Service (NWS), all the Department of Defense agencies, and the FAA. This has been going on for many years.-- We started in 1938 in New Orleans, and then we moved to Asheville, North Carolina, in 1952, and we've been there ever since.

We are an officially designated records center for the Department of Commerce. As such we work very, very closely with the National Archives and Records Administration (NARA). We do have a significant amount of data today, about 140 terabytes now. The NWS modernization will generate about 100 terabytes per year, and then who knows in 1999, what it will be like. So there's a fair amount of data to manage..

So right now this is the way it's looking like in gigabytes, how we've been doing. And from 1986 to 1993, you see here we -- right here is the NEXRAD program. We'll be getting about 33,000 eight millimeter EXABYTE tapes per year -- somewhere in the neighborhood of 88 to 90 terabytes per year from this one project alone.

Now, the good news is that within five years, we would start to migrate these things to 3480 cartridges, and we're going to give Fynnette (Ms Eaton of NARA) about 990,000 3480 cartridges each year.

MR BERRY: The next person is Fynnette Eaton. She is going to get another chance to talk to you about her problem, which she described in her previous talk. So maybe she can say a few more words about it now. She is the chief of the Technical Services Branch of the Center for Electronic Records, and has been an archivist at the National Archives and Records Administration since 1977.

MS EATON: Yes, if it's alright -- is this working?

MR BERRY: Why don't you use the one in front of you?

MS EATON: I do not have any overheads. I am wearing two hats for this meeting. Margaret Adams, who is the chief of the reference services, supplied me with the information about the Center's reference services.

As I said in my talk, we have approximately 19,000 files. They range from military files from Vietnam to the 1990 Decennial census files. The Bureau of the Census constitutes almost 30 percent of our holdings. About 50 percent of our requests are for the census files.

The date span of our files is from the 1960s up to this past year. We do not have information on line. As I was telling my host at lunch, I think NARA is the dinosaur of the group. We do not provide on-line access. People can get documentation about our files and then we will supply copies of the files either on 3480 or 9-track open reel.

The National Archives has a new facility that is very close to this campus. Our office will move into this new building in January 1994. There are plans to provide access to Internet for NARA staff at this building. Once Internet is available to the Center for Electronic Records, we will explore ways for making information available to users through the Internet.

MR BERRY: Thank you. The next panelist is Jordan Gottlieb, who works at the NASA Goddard Space Flight Center. For the past ten years he has participated in the design, development, implementation and maintenance of small and large software projects, project implementation and management, and also had extensive involvement in system integration and archive management of near-line and off-line data.

MR GOTTLIEB: I work at the National Space Science Data Center, and we have an enormous range of media we deal with, most of which are listed. We consider 7-track no longer a current media, but it will show up on the next slide as having significant holdings.

What we do is we act as a twofold division. We actually are chartered with archiving data, scientific data, but we also have taken on a new role in the not too distant history of also providing distribution services. Distribution services get to be rather interesting because we support an on-line distribution service for that data which is electronic, but we also have to support the analog portions, scientific photographs, fiche film of very old history, as well as alternate media for electronic data.

The current estimates of the holdings are listed there, and right now the holdings are probably somewhere around three and a half terabytes. We're looking to have that continue up to six terabytes in the coming year, and then we're looking at a projected growth of approximately six terabytes every year after that.

It gets to be a rather large problem as the equipment gets more and more sophisticated and the data rates keep going up, and it becomes a problem to manage these issues.

7-track is rather interesting because we are in the process of migrating that to a more modern media, and one of the things we are challenged with at the Data Center is continually finding ways to promote data up into more current media. As was stated in many of the presentations, the media changes so rapidly that by the time we actually can do a study and acquire new media technologies, the next media is out and touted as better and more efficient.

MR BERRY: Thank you. The next panelist is Laura Potler, who has been with Goddard for nine years. She started with compiler debugging on the massively parallel processor and has transitioned to systems analysis of data systems for satellite projects, including ROSAT, SeaWiFS, and TRMM.

MS POTLER: Hello. Well, I'm really surprised to be with such a distinguished group. I'm in a very different category-- I do not work for an archive and distribution center. My job is one step removed. I'm a systems engineer, and I've been working at Goddard on an assortment of projects over the last... close to ten years, actually. I half suspect I was invited because a few years ago at one of these conferences I shot my mouth off about the state of eight millimeter, and I think they're getting even with me.

Anyway, I have worked on a variety of projects. I started out on ROSAT, which is short for Roentgensatellite. This is x-ray astronomy; high-energy astrophysics. It was launched in 1990. Then I worked on the design of SeaWiFS, which stands for the Sea-Viewing, Wide-Field-of-View Sensor, which is the follow-on to CZCS (Coastal Zone Color Scanner). SeaWiFS is scheduled to launch next year. Very recently I switched to TRMM, which is the Tropical Rainfall Measurement Mission. I've been doing system design of the data systems which produce all the data that goes to these folks here.

I wrote up a few notes about the projects themselves, the formats of the data. As you see, the ROSAT data is in FITS format. SeaWiFS and TRMM, HDF. We're hoping to feed both of these data sets to the Goddard DAAC. The size of the holdings, as the years go by, changes dramatically from ROSAT to TRMM. I have it broken up by *proprietary* and *public*. It could be *intermediate* and *final* products or however you want to term it. In terms of proprietary data holdings, ROSAT is currently working towards 300 gigabytes. TRMM is expecting 66 terabytes over the life of the mission. So you can see how dramatically the data holdings size increases.

The archive medium for ROSAT, both the proprietary and the public, is 12-inch WORM. SeaWiFS has an intermediate (or proprietary) archive on 5.25-inch magneto-optical platter and is planning to have the DAAC as the public archive, which would mean a combination of 12-inch WORM and VHS. TRMM is still TBD.

So, as you see, we've got projects that are in really very different stages. ROSAT has been operational for several years, SeaWiFS is about to be operational, and TRMM is still very much in the design stages. So I have a real interest in the discussions going on here.

In terms of volume distributed, I called up ROSAT. I haven't actually been involved with ROSAT for several years. I called up Cynthia Cheung and she told me for the month of August, 334 requests were made. These comprise 6,000 files ranging from 1 to 10 megabytes per file.

For SeaWiFS and TRMM, we haven't started distributing data yet, so we don't really know what we're going to be up against. We have estimates based on their predecessors.

Mode of distribution available: ROSAT is shipping out uncompressed data, mainly electronically. That's the way they prefer to deal with it. Again, the volume is such that it's manageable at this point. They do get some 8-millimeter and 9-track requests, but the demand is minor, really, compared to the network.

SeaWiFS' mode is dependent on the DAAC, and I suppose TRMM's will be, also; and, as well, of course what mode the science communities wish to receive the data by.

Wishes and responses to problems: maybe we'll get into these as we get further discussion. I don't want to ramble on here, but there are a lot of wishes that the various groups have. So I'll wrap it up with that.

MR BERRY: And the last panelist is Darla Werner, who is section manager of integration and technology assessment, affiliated with the Hughes STX Corporation. She is project manager for the Landsat Digital Archive Conversion System and managed EDC's digital archive computer operations and technical support areas for over ten years. She implemented tape baking in April 1993, which I believe is a solution for some of the problems where the lacquer on the tapes is peeling off.

MS WERNER: The EROS Data Center is a U.S. Geological Survey Facility. It was established in 1972 in Sioux Falls, South Dakota, to receive, process and distribute Landsat data. EDC was designated as a national land satellite remote sensing data archive in 1992. EDC archives over 10 million space and aircraft images of the earth's land surfaces. Three million of those images are Landsat.

The national archivists focussed on developing advanced data archiving and retrieval to permit more efficient storage and retrieval of the large amounts of data that we will be receiving in the next 10 to 15 years.

As far as the digital archive, the facility is over 12,000 square feet of environmentally controlled storage space in the lower level of our facility. It is accessible to the computer room via an elevator, and all other accesses are card key. The original 4,400 square feet were constructed in 1978; and 10 years later, we finished off another 8,000 square feet to accommodate the early historical Landsat wide band videotapes and also to create an overflow area for 9-track and 3480.

As far as security controls and environmental controls, we do follow the National Archives' Code of Federal Regulations for Electronic Records Management and also use the Care and Handling of Magnetic Storage Media publication from NIST.

The EROS Data Center has made a major commitment to the long-term preservation of data. We currently have two media conversion projects in process: copying 9-track tape to 3480 and also transcribing Landsat data from the 1-inch high density tape to DCRSi digital cassettes. We began baking sticky tapes in April of 1993, and we have experienced a 100-percent success rate with that venture.

As far as our current storage media, our primary media is still the 9-track tape; however, we are copying to 3480. We did start with about 105,000 9-track tapes 3 years ago, so we have made some significant progress. As far as 3480, we have about 50,000.

The 8-mm cassettes are used as system backups, and they are used basically on all of our major systems throughout the building. QIC tapes are used by the users on their work stations, and I do believe that we have more QIC tapes in the building. This is all that is registered in our digital archive, but users tend to keep them in their desk drawers and wherever.

We currently have at the Data Center about 37,000 1-inch high density tapes. We have another 26,000 tapes that are stored in Alexandria, Virginia, that are being incrementally shipped out to the Data Center. We will be transcribing the 26,000 tapes which hold the TM Landsat data and also these 13,000 high density tapes holding some of the older, or the more recent, Landsat MSS data.

The DCRSi cassettes are the result of a conversion that we are doing, which began in December 1992. By the way, these 200 cassettes hold the data that was on 7,200 high density tapes.

CD-ROM: we have what's reported here just a small number less than 200, but again, this is what's registered in the digital archive. If you look in the offices at the EROS Data Center, I believe that we've got hundreds and hundreds of CD-ROMs. They're quite popular.

We have two robotic systems, an EPOCH file server that is used for browse images and electronic file transfers, and also a newly installed STK silo, which is used for raw data sets to be used later for image processing.

As far as distribution, our primary media is 9-track tape. We put out very few 3480s on a monthly basis. We'd like to see that increase. And 8-mm cassettes have been requested more often, even just within the last six months.

As far as issues, problems, and challenges, the first is the rapidly changing technology and the challenges of technology obsolescence. I believe we need to put more emphasis on retrieval. Due to the changing technology, the question I often ask myself is: are we going to be able to play back the data 10 or 15 years from now that we've recorded today?

Also, our distribution data sets are getting larger. Because of the large size, the distribution media options are more limited. We are starting to put out more one gigabyte-sized data sets,

and naturally, that's a lot of 9-track tapes, which most universities are still using. There are some universities that have been requesting 8 mm, but because we like to verify our products before they go out the door, that's a very slow process for product generation.

We wish there were more 3480 users. We are watching advances in the 3480 technologies. Media management and maintenance are expensive. As our digital archives grow, there's more and more tasks that are associated with maintaining an archive. We can't just put a tape on the shelf or in the archive and then call it done. Media maintenance and management require people and specialized equipment, and that costs money.

As our digital archives get older, we have to convert to newer media or advanced recording technologies, and conversions cost money and take a lot of time. I also believe that conversions are a never- ending process.

Data management is a science. It involves many tasks and considerations. It's not just archiving and it's not just data handling. It's defining metadata and knowing the environmental conditions and specifications for good archiving practices; making sure that data is going to be available and useable by future users.

I believe that data management is a very complex system of processes that depend on one another. For many data facilities, I believe that data management is a number one problem as far as having funds allocated. I think that more budgets need to have data management as a line item versus just a category under a project. It seems as though when funds are allocated for data management, that the moneys tend to go into the systems that are used to record the data; and there's very little resources that are available for archiving and the tasks that are associated afterwards.

Thank you.

MR BERRY: Thank you.

Now you know a little bit about the panel. First of all, we would like to entertain questions from the floor or, alternatively, we'll discuss a series of points. This is your opportunity to ask questions of the panel, to get their opinions or find out more about what they've been doing.

I'm going to start it off with a question. What I'd like to know from the panel is the following: I notice you are using things like 3480, 7-track, 9-track. Where do you anticipate going in the future, let's say two years from now, three years from now? Are you still going to be in the same place or will you be in a different place?

MR DAVIS: We'll be still 3480 or perhaps 3490, if we get that capability.

MR BODDEN: For the Version 0 DAAC, which is part of the EOS project, we anticipate staying with the Metrum, which is giving us a lot of good use. And also with the Cygnet jukebox. But the problem for Version 1 becomes a lot more complex, and they will be receiving up to one terabyte of data per day to process. So the EOS project itself has to still be looking at what kind of options and alternatives are out there that can handle this kind of load. So there is somewhat of an open page here as to where we're going to go in the long run.

MS WERNER: For our long-term preservation of data in our lower density archives, we have made a commitment to go with 3480s, but we are also looking at the advances in that technology. The 3490 looks to be a promising substitute for that.

In our higher density archives, we will be using the DCRSi cassettes, but longer term, we are keeping our options open.

MR BERRY: Could each of you speak to the question?

MS EATON: Most of our files are much smaller, and what we are interested in doing is actually downsizing it. What we would like to do for most of our users, who are not -- they're more interested in specific information from the files as to try to move to floppies to actually send it out, so that it can be more exactly what they want.

We currently use 9-track and 3480. Most of our users are from universities, so they have the mainframes. But if we could move to other modes so the PC could be used, we could get a much wider distribution.

MS POTLER: Again, in my situation I'm not so much looking at archiving and distribution of the final products but finding ways to keep data, large amounts of data, near-line so that I can do reprocessing in order to get the final product and give it to these folks.

We are keeping our eyes on the market, trying to figure out what the best solution is. I haven't seen it yet, but we're looking. We're trying different things.

MR GOTTLIEB: The Data Center has an idea of what to do in long term. We currently are supporting 9-track, 3480/90, 8 mm, 4 mm, 12-inch optical, CD- ROM, and adhering to the ISO 9960, and we're looking to progress into other areas. We are currently looking into D2, D1, D3, when and if it becomes available, quad density platters, optical platters, blue laser CD-ROM, which will be a 2 gig CD-ROM.

But we also have to continue to support the user base, which may be regressive in the current technology trends, so that right now we can't plan on migrating everything to a forward technology and lose the capability of being able to provide data to our user base.

MR BERRY: Okay. Thank you.

VOICE: (Off microphone.) The issue of archive came up with six panelists; they probably identified about eight different long-term archive media. The first question is, I guess: Is anybody working the issue to say the United States shall go to -- in some kind of synchronous fashion to D2 or whatever? And if that were to ever occur, what would be the impact?

MR GOTTLIEB: The answer is that the decision of how an archive actually manages and stores its data is an internal question. And yes, there are standard activities being processed by the National Institute of Standards and Technology. The question is whether or not excluding certain media out of the marketplace would present problems in the U.S. and whether or not there could be more than one sanctioned archival media.

So what the data centers tend to do is look at those media which are currently providing adequate storage and recovery, as well durability for the long-term archiving, with philosophies of future promotion into more sophisticated media.

MR DAVIS: I would take perhaps a little exception to the internal situation with the archiving. In our case, we must go ahead and keep our archives in a format that will be transferable to NARA in the future. Therefore, it does not become a totally internal situation of how we're going to keep those.

Right now, they accept the 9-track or 3480 in ASCII or EBCDIC. So we attempt to go ahead and do that for those files that we know may well be transferred to NARA at some time in the future. So there's always a problem with the term archive and long-term retention.

Archive is for permanency, real, in perpetuity, where long-term retention *is temporary.*-- NARA says that temporary can be up to 75 years. Well, we're in the process of trying to keep our data at least for 75 years before we turn it over to the Archives in many instances. But we must follow their dictates.

MR BODDEN: Let me add just one more thing. I'm not so sure that it makes sense to go to just one standard media for archives, and I just want to quickly point out that in the Goddard DAAC we have selected two different types of media for two very different reasons. We went with the optical platters for what we consider our most highest priority data, most important data, and the data that we could spend money on.

For the VHS system, our Metrum system, what we selected there was a media that provided us a very economical amount of storage per terabyte or per gigabyte, whatever you want to call it. So there are two different types of media that are being selected for very different reasons. So that's some of our justification.

MS WERNER: In 1988 we went through a lengthy process of reviewing the types of media that were available at the time for both our low-density and our high-density archives. At that time we elected to go with the 3480 to replace the 9-track tapes. But, with the large amounts of data that we have at the Data Center, we have to be more conservative with our choices rather than choosing maybe the newest technology at the time. So we have made a commitment to go with 3480.

MS EATON: If I can second with the last two comments that speakers have said and then add an additional thought. Because of the diversity of formats used by agencies, the National Archives uses standards to ensure that we will be able to process those files deemed to have long-term value and that is why our current regulations cite 9-track and 3480 cartridge. We can find drives that can read the data, so unless there is a problem with the tape itself, we know we will be able to process the file. There is also, though, the issue of what do you need the information for. If the information is scheduled to be transferred to the National Archives, then it must be in a format that we can process. But if it is current information that your agency will need for five or ten years, and it has been scheduled as temporary, then you should use whatever format is best for your institution. We don't feel that we have a right to impose standards on temporary information. We can give suggestions, but it is really up to the individual institution as to what they use.

MS POTLER: Well, I keep coming to these things hoping that I will hear from various committees the answer we are all seeking. There is no one good answer. I agree with everything that's been said so far. It's interesting that every time I come to these, I hear about more and more technologies. It's diverging instead of converging. It's exciting, it's interesting. I'd like to see more work done in terms of committee work or various organizations getting together to try and do some more standardization of what's already in existence, because once we do commit to something, we have to stick with it and make it work. And I'd like to see more elegant software to support a lot of the hardware technology and so forth. But I agree with you that standards is a big issue right now.

VOICE: My question deals with the use of compression and what experience either using the industry standard 3480 -- I don't know the name of the compression algorithm *(Editor: IBM calls this IDRC, Improved Data Recording Characteristic, and has licensed it to other manufacturers under names such as Improved Character Recording Characteristic, or ICRC)*-- for some of the more specialized algorithms which might be applied, especially considering the cost that CPU power is decreasing at a significant rate now. Has anybody had any experience with applying compression techniques to the data? And does it impact your error rates?

MR DAVIS: We are just experimenting now with compression techniques on the 8-mm tapes on the EXABYTE drives. In some recent tests we found a compression rate of about 8 to 1 for the NEXRAD radar data, which *meant* we got about 38 gigabytes on a tape, which sounds great. But then when we read it back, it took a little over 20 hours to read it (*Laughter*). We didn't have any error rate problems, but our concern is, of course, that if you get into the middle of that thing and get some sort of little burp or something like that, you've got a lot of time invested in that one tape.

We'll probably use that compression and go to something less than the 8 to 1, maybe 5 tapes to 1 or something like that, where it would really be beneficial to us.

MR KOBLER (NASA): If I could just interrupt for one quick second. I know Sandra Woolley is in the audience, and she will be doing a paper the last day. I invite you to listen to that paper. That might address some of your questions, unless she wants to respond to that now perhaps.

MS WOOLLEY(Manchester University, England): Thank you. Yes. Data compression does impact your error rates. If you have, say, a single uncorrected bit pass through the system, it can scramble all data to follow, and that's the main theme of my talk. Robust error control is absolutely essential to preserve data integrity. Thank you.

MR BODDEN: Continuing to talk about compression, at Goddard DAAC we've also just started looking at compression as our on-line system is just really getting going now over the last few months. We are looking to compress at three different points. We transfer data over the network from different data projects, and we were looking to compress the data at these points. We've had difficulty getting that started so far.

The second point that we're looking to compress is, as we receive the data, process it, then we put it to the archive, we were looking to compress it at that point. And that has yielded some very good results. For one class of data, like AVHRR (Advanced Very High Resolution Radar) data, we're getting a 70- to 80-percent compression rate. Each file is about 240 megabytes in size, and we're able to compress that down quite nicely. So that has worked there.

We've also been trying to compress the data as we write it out to media to send out to researchers and scientists, and we're just getting started with this. In all of these first three compression techniques, we're using just a very standard UNIX compress. We're not going into any fancy compression algorithms, but we have looked into them. And we've found that the UNIX compress doesn't give us as much as some of these other algorithms, but, given the difficulty that the researchers would have out there in handling the different kinds of compression, we stuck with just a pure UNIX compress.

MS WERNER: We've attempted to apply IDRC on our STK 3480 rack mountable tape drives, interfacing with SGs and DGs, and have not been successful at doing so.

PANELIST: We have not tried to compress our data yet. There hasn't been the need.

MS POTLER: ROSAT doesn't compress. SeaWIFS is just starting to look at compression algorithms, so I don't have anything to say about that. TRMM will definitely have to compress at 60 gigabytes a day, but I don't have results yet.

MR GOTTLIEB: The Data Center actually uses compression but not on most of the scientific holdings. The places we've encountered compression and had it successfully implemented and then extracted again was on the CD-ROMs that the Data Center distributes. There is a concern that taking compression techniques and applying them to data, there is a possibility that you will find some data loss. And when dealing with pure bit streams where every bit is either meaningful or unmeaningful and having that change and become meaningful is a real concern. So there is a danger in that compression will not yield 100-percent accuracy all the time.

DR HARIHARAN (Systems Engineering and Security): What was the problem in writing out the data in compressed form on distribution media?

VOICE: Right. The problem wasn't so much in trying to compress the data. It was just -- it's a new function for us and we haven't got it working yet. That's all it is. We will get it working.

MR BERRY: The question was: what are the problems that he has experienced in using compression on his data?

VOICE: A question for the National Archives. Why is it necessary to centralize all the data at National Archives? Why not network the data and have each agency who owns the systems that are unique to their data archive them in place with you maintaining some index?

MS EATON: That's an interesting possibility. We have not considered that because we don't have access to a network. There is also the concern about documentation. It is only by working with agencies when they transfer files to us, that we're able to determine what the problems are with the documentation. Further, it is only when we can compare the documentation to the file that we know everything is complete. Too often, the documentation is incomplete and additional work is required. So, there would be a problem with the agency maintaining the file, unless they ensured that the documentation was complete, which has not been the case to this point.

As I alluded to in my talk, when the National Archives commissioned the study about current federal data bases, they specifically excluded scientific data bases. And actually, my friend on the far right has been dealing with the National Academy of Sciences' study in which they're looking at what should be done with the scientific records. My personal view is let the agencies keep them, since the agencies have the expertise. I don't know what the study will recommend, but perhaps it will be close to what you recommend.

There is also, with the National Information Infrastructure initiative, the idea of creating a government information locator system. So that might be a way of going about it as well. There's a lot going on with these issues, and I'm not real sure what the direction for all of this will be in five years. But at least for now, the National Archives accepts custody of files in order to maintain the integrity of these records.

DR ANDREW OGIELSKI (Bellcore): Our panelists represent publicly funded archives. What projects are under way in your institutions to improve access over the data networks?

MR DAVIS: Right now we are working on and have several files on line through Internet that are free to the scientific community over Internet. These are both metadata files, inventories, where you can browse and in some cases you can go ahead and go a step further and actually order off-line data that way, but then we also have actual data files out there for, in some cases, the most recent period, like the last month or two, that you can access and use. That effort is expanding fairly rapidly at our center.

MR BODDEN: For the Goddard DAAC, part of the mission for EOS is to try to bring the Earth science data that NASA and the affiliated agencies, such as the U.S. Geological Survey, hold, to bring this data on line. So, that's our mission. As we bring it on line, we are also going to provide network access for the world to log into the Goddard DAAC and browse through our data holdings, which will be represented by metadata records, summary records, of the data.

The researcher will then be able to select, during this session, some samples of data up to a certain limit. That limit we haven't really set yet. And that data, if it's small enough, the amount that the user has selected, can be FTP'd back to the user during that same session. So we are trying to set up a system where you can research your data, you can access it, and retrieve it, all during the same session.

MS WERNER: The EROS Data Center has developed a system called a Global Land Information System (GLIS), and it is available to users on the network for access to US and foreign Landsat data, AVHRR, and other miscellaneous earth sciences datasets. The GLIS, as we call it, has the capability to allow research; and some browse files are available, so that the scientist or user can actually see what their area of interest might be like.

If you would like information on GLIS, please talk to me afterwards and we can give you the address for that.

MR BERRY: Will the panelists make sure you speak into the mikes so everybody can hear you?

MS EATON: What I'm going to say next really applies to the entire National Archives. The National Archives has begun to put some of its selective guides on the Internet so people can get a taste of what is available. We have our title list that is available on the Internet, but that is the only thing that we can provide across networks at this point.

We are hoping in the next couple years to determine if there are some files that possibly should be included so that people could access it that way.

MS POTLER: We certainly have been working on the browse capabilities in conjunction with the DAAC. The hardware itself is not the problem. SeaWIFS has both Ethernet and FDDI connectivity, which is more than adequate to handle the load right now. The problem is the users don't have the high-speed connections, and they have so many hops to go across and so forth. So it's more in terms of what we have to deal with our audience, our user base, and what they have to deal with.

We do support anonymous FTP and so forth to get the data and the browse capabilities. We're trying to improve the software so that they can work with it more easily.

MR GOTTLIEB: Well, it turns out that the Data Center actually has an on-line system that is actually -- to the users it is a mail interface system. So by simply sending a mail message to the system, it...

MR BERRY: Could you speak into the mike a little bit more?

MR GOTTLIEB: -- will actually stage your data into an anonymous FTP area and you can come and get it. As stated before, the Goddard net is a T1, and, as we go outside -- actually it's a hypernet -- as we go outside, we find that 9.6 may become very painful to transfer, you know, three or four megabyte files to a 9.6 station.

There is a selective process that a committee meets on as to which data files are put into the on-line system; but also as a distribution center, the entire holdings that are cataloged are available through other means and media requests. So you can actually write to the Data Center, the User Support Office, which I can give you more information on, and acquire any of the catalog holdings through the National Space Science Data Center.

MS POTLER: I'd like to add one more thing. Even though the users don't have the FDDI or the even higher speed networks, by putting some of the Goddard systems on the FDDI, we can send data to the DAAC via FDDI, and therefore limit the contention on the Goddard Ethernet, which is a dramatic improvement. So there is that advantage.

VOICE: I have a two-part question. The first part is for Laura. If you had to make a decision now as to what type of storage technology you were going to use, do you think you would go for the 3480s? Or would you go with perhaps the Metrum, or another optical?

MS POTLER: Can you be more specific? Are we talking about TRMM launching in '97?

VOICE: Yes.

MS POTLER: If I had to make a decision now for archiving, for distribution?

VOICE: For archiving.

MS POTLER: For archiving. I -- you're really putting me on the spot here in front of all of these people. I would -- if I had to make a decision now, I would go with WORM.

VOICE: Okay. I'm afraid I'm going to put somebody else on the spot. For Darla and Dick, you both have a commitment to the 3480s, and it sounds like you have an awful lot of data to store.

Doesn't that become kind of a management nightmare for all those cartridges that you're storing?

MS WERNER: Actually, copying the 9-track tapes to 3480, is a lot less of a nightmare using the 3480. We do have a lot of data, but we have seen about a four time decrease in space requirements. The advantages of the 3480, as far as speed, reliability and just the easy handling make that an easy choice over what we have right now.

MR DAVIS: And my answer is yes. (*Laughter*)

VOICE: Lee has gone on to the Metrum and I think -- it sounds like he's satisfied with it, where you get a lot higher density per cartridge. Have you all thought about that? It probably wasn't out when you made your decisions.

MR GOTTLIEB: Is this question to me?

VOICE: (Off microphone.)

MS WERNER: Excuse me. Could you please repeat the question?

VOICE: Well, I guess the Metrum technology -- I'm asking a lot of questions about it because I'm looking at it. It probably wasn't out when you all were making your decisions. I guess, if I understand correctly, it would probably decrease the number of cartridges even further. Have you all thought about that at all? Or are you all firmly committed to the 3480s now, so it can't really be an issue?

MS WERNER: Right now we are committed to 3480s for our lower density data. We are always keeping our eye open and watching for current technology and future advances. As I stated before, we tend to be a little bit more conservative because of the large datasets and large data volumes that we deal with.

MR BERRY: Let me give a corollary question to that, because you all have picked a technology that's relatively expensive from a per bit standpoint. There's sort of an implication here that cost is not a driver. If you look at the relative cost of storing something in one media versus another -- that's some of the dramatic differences between media -- is the cost of storing or even sending data to someone.

And it also tends to preclude -- most workstations, for example, don't have that capability. Certainly almost no PCs do. So are those kinds of considerations having any impact in your systems? I mean, you're using historically what have been sort of the mainframe kind of approach.

MR DAVIS: From our standpoint, of course, cost is always a concern. But the initial cost of the new systems, when you're talking *big bucks, is important.* Well, if you take a CREO system, a dual drive CREO system, for example, which we've been looking at, and we'd like the optical tape option. We'd like the permanency for our function, which is primarily archive, and has good recall. But you're talking about a situation there where you've got about $750,000 to $800,000 initial price in order to buy a two-station system, and that's a pretty good chunk of change, particularly for fairly new technology, even though I think it's very viable technology.

The cost of media for 3480s is down about $5 a cartridge. We're paying more than that even for the 8- mm stuff right now. We, too, are constantly looking at new technology. We're always open to new ideas, but we do have to take the more conservative approach for our basic archives.

Now for the NEXRAD system where we're using 8 mm, we have no choice because the National Weather Service has installed and are installing those recorders in the field, and we had no option on that. That's strictly an economical situation there. The drives are reasonably

priced. The media is reasonably priced for the amount of data that you can get on one of those. So we're forced into that. We really are not going to convert those to 3480 and give them to NARA.

DR MARIA ZEMANKOVA (MITRE): My question is on data base management; that is, are you satisfied with the current state of the art of data base management systems so that given that we can store all this stuff out there, we can actually get the information from it that we need?

MR BODDEN: I'll start that response for the Goddard DAAC. We anticipate that, as we approach our 20 terabyte goal, that our DBMS problems are going to become very severe. Right now our data base is performing quite well, and we don't see that changing in the very near future. But we have started talking with vendors, and I'm not going to tell you which ones, but we have started talking with vendors who have new and innovative approaches to data bases where they will distribute one data base over several platforms and control that data base through several other CPUs. So there are new ideas out there, and we are exploring them for the not too distant future.

MR BERRY: Anybody else?

MR GOTTLIEB: There are two aspects of the information you're talking about. I think maybe what you were addressing was both the pointer to retrieve the actual information, but also what, at the Data Center, we might call metadata, which is information about the file you might want to retrieve. And the answer is that I don't think you have to position yourself to use a single data base to answer that question. So that would be the first approach I would say, if somebody were to say "my data base can't handle it anymore" -- perhaps you have to reengineer it into several data bases.

But also with medium proof -- there's also a host of other technologies that are improving, perhaps a little more slowly or even more quickly, and data bases are progressing in some fashion. I know object bases have become available, and object bases, I think, start handling the metadata questions where a traditional RDBMS could actually point into the archive and direct you to retrieve a file.

So I think that the answer to your underlying question is that we will have to rethink how the traditional data base is implemented as the archives grow.

MS POTLER: Well, I think the DAAC, in particular, that I'm familiar with is doing a lot of work in terms of metadata and organizing with the user in mind. I know they're holding the projects to the line on these things.

My concern is more with how to get the data out of the hardware, the media, to locate it and get it out rather than what particularly you're looking for within the data base and how the data is organized. I would like to see more elegant handling of the data as it's stored and as it's retrieved, that sort of thing, is what I'm seeing. I see a lot of jukebox-type mechanisms, but not necessarily elegant software to use it.

MS EATON: We have developed a data base for capturing the metadata so that we can do automated validated files that are transferred to us. That project has been ongoing for about a year and a half. It's still not fully operational, but we're trying to capture the metadata in that way.

And we are also considering building a system that we call X-WEF so that we can get specific information about files, across the files, to do a reference system, and we're just beginning looking into that now.

MS WERNER: We have several metadata data bases to handle the different data sets that we have at the Data Center. It doesn't seem to be an issue for us right now.

MR BERRY: We have had a request from the audience. If all of you would be willing to provide the Internet addresses where some of this data could be reached, then we could make it available to the participants. What we will do is make it available on a sheet and have it out front so that you can pick it up later on during the conference. We will do that for you for those people that can supply the information.

VOICE: There have been a lot of questions as to whether in ten years will there be the drives to read particular types of media. What about the flip side of the question, the software format of the data on the media? Are any of you facing the issue of supporting either computers or software packages simply for backwards compatibility to be able to access data that you would just as soon retire?

MS EATON: We cheat a little bit. When certain agencies transfer files to the National Archives, they give us two copies of the file, one in ASCII, the other in SAS or SPSS format. We will copy both formats. We will make the dependent format available to researchers for the first ten years, because we will feel that that version will still be supported by the software that's out there. When we recopy the file after ten years, we do not recopy the dependent file. We just keep the software-independent file at that point. So that is how we cheat.

MS POTLER: I'm not facing that problem; that is up to the DAAC.

MR GOTTLIEB: Actually, we are facing that problem, and the way we're dealing with it is we currently have another committee to attempt to describe and conclude what the best formatting techniques will be. One of the ways we are thinking about solving that is basically internalizing a standard format for the archive so that we could create an *in* filter and an *out* filter, which could either ingest or export any of the required formatting necessary.

MS WERNER: This has been a big problem for us. We have tried likewise to develop an internal archive format, but many of our datasets are in a native format. So that's what we've elected to go with so we don't have to rewrite documentation and such. But we are trying to go with an internal archive format that would include an ANSI standard label, and therefore, to reduce the amount of software required to use the data.

MR DAVIS: We're basically doing the same thing. Of course, we're obligated to provide data to customers, as well as have data for use in our own center for as far as we can see into the future. So it's important for *users* to be able to go ahead and read these data.

We have basically an internal format for our standard records that we use; but we also get a lot of stranger tapes that we get from other organizations, and we have to be able to read those and define what's on there and that sort of thing. So it's a pretty good maintenance programming job to keep up with all that.

One hopes that the industry will go ahead and, as it progresses, will allow you to progress in some reasonable fashion rather than a fruitbasket turnover-type thing.

MR BODDEN: For the Goddard DAAC in dealing with this issue, we have to look at it in two different parts. For the data that is supported by commercial software, that we access through commercial software, that's sort of setting a standard for the data, and, by that, I mean that we are now asking projects that transfer data to us to put the data in a standard format, such as HDF, which is supported by NCSA. Or there are some other formats that we would be willing to take but are not sanctioned by the EOS project, such as CDF.

One concern that we have is that our archive is being placed under a file management system called UniTree, which so far we've been doing okay with. But our concern is: where is UniTree going down the road? And is our archive going to be able to evolve with UniTree? Or are we going to have to at some point move into some other kind of file management system?

So that's a concern for us. And I might say that the people who we buy UniTree from, Titan, have been very responsive in trying to meet our goals with Unitree.

And then one last aspect of this question is the application software that is used to access certain types of data files, and that is a real concern. That's something we want to avoid in the DAAC in the future. We have a lot of old data that is being accessed by programs that have been written 10 or more years ago. So that, we want to try to shy away from, and want to try to move towards standard formats for the data that we're receiving.

DR KING (National Space Science Data Center): Our data centers have a dual responsibility to provide convenient access today to data in the data centers, as well as to preserve the data for the long term so they will be available 50 years or whatever downstream. In your discussions of data media choices, I've not heard that particular distinction brought out. In fact, might the optimal scenario be one where one type of media are in our jukeboxes, providing that convenient, current access, and perhaps the same data on either the same type or a different type media in our deep archives?

MR DAVIS: Well, we keep statistics on what our customers want. Of course, there are some customers who want everything and they want everything on-line, and we're not able to go ahead and do that. We just don't have that capability from hard disk or jukeboxes or anything else at this point.

However, we have noticed some trends. For example, we no longer get any requests for 7-track tapes, and we are getting fewer and fewer requests for 9-track, 1600 BPI tapes. We are seeing an increase in the requests for 3480s, and 2 months ago for the first time, we discovered now the most popular, most frequently requested format is on floppy disk. We are also seeing a great increase in the request by customers for data on 8-mm tape. We do have some CD-ROMs. We don't produce those at will, but we have about seven of those that we distribute.

So we have seen a definite trend towards the users of particularly floppy and 8 mm, and our obligation is, and what we do is, we take data from our 3480 files, we will go ahead and put out data files in any of those formats that the customer wants.

MR BODDEN: Yes. The same thing for the Goddard DAAC. Internally, we store the data on optical platters and VHS tapes. We used two different types of media because of the importance of the data. We don't want to put all the data in just one type of media and have a single point of failure. We view the data as very important, number one, and it is very hard to replace. So a lot of this data is coming down from satellites that once they're -- it's just irreplaceable.

As far as distributing the data, we distribute on popular media, such as 4 mm, 8 mm, 3480s and the 9-tracks. Let me cancel the 3480s. That is available but only through special request. And the same is true with optical disks. That is available through special requests. So we handle quite a full range of media.

MS WERNER: Our Landsat data will all be transcribed to DCRSs is, so we are being consistent there with one media choice. And there will be about 50 terabyte of data.

Our lower density, as I've mentioned, we're using 3480. We've got a commitment there, and we've got about 35 terabyte of data that will be going to 3480. However, our distribution media, we, too -- it's driven by user request, user demand, and somewhat by what we offer. We are still putting out a lot of 9-track tapes. We would like to see that move toward more 3480 and 8 mm. In the last 6 months, we have seen a big increase in the use and demand of 8 mm for file transfers.

MS EATON: We have had to use the computer center for doing both our preservation work and reference work, so we've been very limited in what we could offer. We are trying to build an in-house preservation system. If that works, we would then build a reference system, and we

would probably try to hang at least floppy drives from this system, as well as other types of drives, if there are enough requests for another format. It's one of the issues we are looking at.

MS POTLER: Obviously, there has to be different criteria for distribution and archiving and backup, which is something we really haven't talked about here and is a major concern, as well as internal reprocessing and so forth. I spoke with ROSAT project recently to see how they're doing, now that they're in operation, and how they feel about it. One of the things they said that they're looking into is that they have the proprietary and the public archive on WORM, because they have the most confidence in WORM. But at the same time, they have all their eggs in one basket, and if something goes wrong or the company goes out of business or whatever, they're stuck. It's certainly a major concern.

SeaWIFs took the opposite tact. They took everything. They have 8 mm and 4 mm and 9-track and MO. As long as the budget allowed, we got a little bit of everything to be covered. But it would be nice if there were some way to narrow that down a little bit.

MS WERNER: Good enough.

MR GOTTLIEB: I guess I was kind of set up by the questioner, but at the Data Center we use dual media philosophy and that is we actually take in a media and as best we can immediately produce the second media and even go one step further: try to get it to a secondary storage site so that we have an original and a safe copy which is termed off site.

Unfortunately, the world is not utopian, so we are struggling with how to back up some of the data that has arrived electronically, and we did not have an original media choice.

The other dilemma that we're faced with is, how to back up half a terabyte of data in a reasonable fashion without impacting our requesting community's throughput. So the Data Center is currently supporting a dual media philosophy and are coming up to speed on getting the dual media throughout the entire archive and also supporting off-site storage.

MR BERRY: I think we have time for one more question. You'll notice that even the panel has started to leave, so one more. (*Laughter*)

VOICE: I have a question for the panel. How important is backwards compatibility in your decision-making process? If you take as an example 3480, 3490, 3490E, you've got a clear migration path for where you are going to head with your capital investment over quite a number of years. Or, are you concerned about that you downsize your archives to the extent you can forego that migration path and go into what we call leapfrogging technologies?

MR DAVIS: Backward compatibility is critical to us because of our customer base. We've got -- although we get about 90,000 requests a year at the Center, that's not all for digital data. We have somewhere in the neighborhood of 2,800 requests or so per year for digital data. All you've got to do is to stop being able to provide one type of media or one format, and you hear about it very, very quickly.

So, it's important to us, the backward compatibility. But, on the other hand, that means that we've got to be able to do that to the customers. The only thing we've stopped in the last 38 years that I've been there is punched cards and 7-track tape. We can still do everything else, and I think we will continue to have to do that for some time to come.

MR BODDEN: Backward compatibility is also important for the Goddard DAAC, but we are using the EOS project as a breakpoint where there are certain data sets that we will no longer support the backward compatibility or the old version of these data sets. And we're actually migrating them forward to a new technology. An example of that is the coastal zone color scanner system data, CZCS, some of you may know. This data was produced and was available through a VAX VMS system.

We are now in the process of moving this data over, going through all the bit and byte conversions to move it over to a Unix system. That is our intention, really, for all of the old data to little by little bring it on line in our new Unix system.

MS WERNER: Backward compatibility is very important to the EROS Data Center. Even though we are upgrading to new technologies, the data migration, data conversions are a lengthy process. So, for several years, we need to make sure that we can use the data on the older technology to allow our users to access what they need for their project.

MS EATON: Since we often get the older technologies, it is very important to us. As I was telling someone, we still receive 7-track tapes from agencies, so we have to have that functionality.

MR GOTTLIEB: The issue of backward compatibility, I think, can really be addressed as to whether or not it is an archive concern or a distribution concern. And if you manage your archive as progressive and even conservative, I think the entire issue of backward compatibility goes out to a distribution concern.

At the Data Center we do live and breath with that concern every day. I don't think we've had a recent request for a 7-track tape, but it wasn't too far back where somebody actually did request a 7-track tape. Fortunately, we still had a functional drive with which we could fulfill that request.

So I think you have to look at the whole scenario, and the question you need to ask is: is the backward compatibility an issue for the archive? I think the answer is no, if you manage your archive progressively. But it always remains an issue for your community support, depending on how flexible you wish to be in supporting that community.

VOICE: (Off microphone.)

MS EATON: Ten years.

MR BERRY: Let me repeat the question so that other people can hear it. The question is: "Would you consider a system that did not have a clearly defined migration path from where it is today to where it would go out into the future?"

MS EATON: No. We always look at things that we know we will be able to access in 10 years.

MR BODDEN: The answer for Goddard DAAC is also no, we would not look at a system that did not have a clear migration and evolutionary path.

MR DAVIS: I believe that we would (*laughter*) if there's such an animal out there. Yes, we'd definitely look at that and actually have been looking at it. I would say, for example, a jump from 3480 to a CREO system would be the kind of thing you're talking about as a leapfrog, and I'd have no objection to something like that at all.

MS WERNER: The EROS Data Center would need to have a technology with a clear migration path. However, in our long-term archive for the Landsat data, we did go to a different technology. So I don't know if that would be a leapfrog or not, going from high density to DCRSi.

MR GOTTLIEB: I guess the Data Center would answer -- I don't -- it would have to answer: do you mean a media migration path or a migration strategy?

VOICE: Migration strategy.

MR GOTTLIEB: Then the answer is without a migration strategy, no. But without a clear media migration, it's possible.

MR BERRY: Okay. I'd like to thank the panel. I'd like to thank the audience for your participation, and I think we have a poster session scheduled for 6:00. Do you have any announcements?

# NCDC Mass Storage Systems and Technologies

**Dick Davis**
NOAA/National Climatic Data Center
Federal Bldg
Asheville, NC 28801
Phone: (704) 271-4384
FAX: (704) 271-4246
ddavis@NCDC.noaa.gov

## 1. Size Of Holdings:

Current holdings at NCDC are 107.8 terabytes of digital data and about 0.3 terabytes of manuscript data.

The nexrad radar system is expected to generate approximately 88 terabytes per year by 1996.

## 2. Nature Of Holdings:

Basically environmental data. Climatological observations at varying temporal scale from 1 minute to monthly values for both surface and upper air. Analyzed grids of surface and upper air, summarized climatological information, surface marine observations and gridded values, selected satellite data. There are approximately 400 different tape decks within the archives.

## 3. How Long Site In Existence

NCDC was first established in 1938 as a wpa project to use punched cards to tabulate climatological information. In 1952 the center was relocated to asheville, nc. So, for 14 years we were the new orleans tabulation unit, then for the next 41 years we became known as the national weather records center, the national climate center and then in 1976, the national climatic data center.

## 4. Popularity Of Data Sets:

The popularity of our basic climatological data sets, hourly surface, summary of the day, and hourly precipitation has remained essentially constant. The top ten are:

Surface airways hourly
Daily cooperative summary of the day
First order summary of the day
Surface/land summary of the month
Datsave surface hourly
NCDC us upper air
Hourly precipitation data
National solar radiation data base
Mixing height studies
Surface marine observations

## 5. Media/Technology Used For Storage:

Current policy is that archives will be on 3480 square cartridges. We have just completed a three year effort to copy all round tapes in our primary archives to the 3480 media. This has included merging two or three tapes to one cartridge where possible.

In addition to the reduced storage space needed, the read/write reliability is much enhanced. With increasing acquisition of archive files, we have been able to maintain a viable tape library without an increase in storage area.

We are also involved in a data rescue effort, transferring satellite data from round tapes to cartridge tapes. Over 50,000 tapes have been "rescued" so far and the remaining 20,000 or so tapes will be processed prior to July 1994.

The choice of storage media has influenced the distribution of data but generally user capabilities have also kept pace with the newer tape technologies. While we no longer provide data on 7-track tape, we do maintain the ability to furnish data on 9-track tape at 1600 or 6250 bpi, as well as on 3480 cartridges.

We also are able to extract data from the archives and provide data to customers on 8 mm tape or 3 1/2" and 5 1/4" diskette. These are the preferred media by many customers who have pcs and who do not work with very large data files.

## 6. Volume Distributed Per Month:

447 media units/month over the past two years.

## 7. Mode Of Distribution:

Principal modes of distribution remain as magnetic tape, diskette, 8 mm tape, or cd-rom by mail. On-line capability is increasing and is available through internet. Most of these on-line data sets are special projects such as profiler, or the most recent (perhaps month) period of record from the principal climatological files e.G. Summary of the day or hourly airways observations.

The on-line system also includes several inventory sets and in some cases allows the user to request copies of data sets to be copied off-line, and mailed to him/her.

Another increasingly popular mode of dissemination is by spectra-fax. Several of the most requested publications and data sets are kept on line and the user has only to dial into the fax machine, enter his account number or credit card and then specify the data he wants. The hard disc is searched and the data transferred directly. No human intervention at the NCDC is required.

## 8. Most Frequently Encountered Problems:

Archives: acquisition of "stranger" tapes that do not conform to stated formats, have internal labels that conflict with our tape management system, or that do not contain the data purported to be on the tape.

"older" tapes that are difficult or impossible to read on the high speed drives of the main frame and which must be copied on other, slower speed drives, before converting to cartridge tapes.

Binary tapes with ebcdic labels. These must be copied onto other round tapes in order to convert the labels to ascii and then copying to cartridge tapes can proceed.

Most problems encountered in using the 8 mm tapes from the nexrad system seem to be procedural or system errors rather than problems with the tapes or drives themselves. We have, however, encountered some difficulties during the write process that have been attributed to tape debris. This whole system is in its infancy and it remains to be seen how viable the 8 mm technology is for continuous drive operation and long term retention. Economics dictate the use of this technology.

Customers: we guarantee readability of digital data for a period of 60 days. Very few customers experience difficulty with the output media. Probably the most frequent complaint is with the documentation that is provided.

## 9. Type Of Media Requested/Used:

Over the past two years requests for floppy disks have over taken those for magnetic tape.

Requests for 1600 bpi tapes have virtually disappeared. Most tape customers still want round reels at 6250 bpi although we do have some who request 3480 cartridges.

There has also been an increase in the number of customers asking for data on 8 mm tapes.

We will soon have seven cd-roms available. There has been a 900% increase in requests for data on this media over the past two years.

## 10. Evolution Of Media:

Through the years, storage media has basically kept pace with newest technology. This has provided the opportunity to systematically migrate data sets in order to ensure the readability of the data, as well as decrease the number of media units required to hold the ever increasing amount of data in the archives. The progression has been:

Punched cards
7-track 3/4" mag tape
7-track 1/2" mag tape at 200 bpi
7-track 1/2" mag tape at 556 bpi
7-track 1/2" mag tape at 800 bpi
9-track 1/2" mag tape at 800 bpi
9-track 1/2" mag tape at 1600 bpi
9-track 1/2" mag tape at 6250 bpi
3480 cartridge mag tape
8 mm helical scan tape*

* this media is being used for archiving of nexrad data only.

We are also producing and distributing special data sets on cd-rom.

## 11. Wish List:

We are looking at an hierarchical mass store subsystem that can be upgraded as needed and as funds permit. This piecemeal approach, may not be ideal, but in the real world sometimes one has to use innovative techniques in order to secure to the desired end result.

A dual or quadruple creo optical tape system that could be used both as permanent storage for the archives and as an on-line access for our principal data sets, using raid technology.

Development of an optical tape system using 1/2" film cartridges that could be used in a robotic system. This would be more salable than the 35 mm format now in existence. The recording densities and access times developed by the creo corporation, coupled with the shelf life of the media make this a most attractive approach.

And of course, my pet wish - a truly operational holographic storage/recall system. To an archivist, this would be the ultimate permanent media providing high density recording and nearly indestructible data files.

## 12. Words Of Wisdom:

Caution but not inertia as you attempt to solve the problems of storing multi-terabytes of data. The next technological break-through is ***always*** just around the corner. At some point you have to go with your best intuition and declare that you have reached the corner.

Changing technology must be kept in mind however, and one should assume the mantle of omniscience, planning for the inevitable migration to yet another "new" system.

# A PETABYTE SIZE ELECTRONIC LIBRARY USING THE N-GRAM MEMORY ENGINE

Joseph M. Bugajski
Triada, Ltd.
4251 Plymouth Road
Ann Arbor, Michigan 48105
(313) 663-8622
(313) 663-7570
triada@middlec.convex.com

ABSTRACT: A model library containing petabytes of data is proposed by Triada, Ltd., Ann Arbor, Michigan. The library uses the newly patented N-Gram™ Memory Engine (Neurex™), for storage, compression, and retrieval. Neurex splits data into two parts: an hierarchical network of associative memories that store "information" from data, and a permutation operator that preserves sequence. Neurex is expected to offer four advantages in mass storage systems. (1) Neurex representations are dense, fully reversible, hence, less expensive to store. (2) Neurex becomes exponentially more stable with increasing data flow, thus, its contents and the inverting algorithm may be mass produced for low cost distribution. Only a small permutation operator would be recalled from the library to recover data. (3) Neurex may be enhanced to recall patterns using a partial pattern. (4) Neurex nodes are measures of their pattern. Researchers might use nodes in statistical models to avoid costly sorting and counting procedures.

Neurex subsumes a theory of learning and memory that the author believes extends information theory. Its first axiom is a symmetry principle: learning creates memory and memory evidences learning. The theory treats an information store that evolves from a null state to stationarity. A Neurex extracts information from data without *a priori* knowledge; i.e., unlike neural networks, neither feedback nor training is required. The model consists of an energetically conservative field of uniformly distributed events with variable spatial and temporal scale, and an observer walking randomly through this field. A bank of band limited transducers (an "eye"), each transducer in a bank being tuned to a sub-band, outputs signals upon registering events. Output signals are "observed" by another transducer bank (a mid-brain), except the band limit of the second bank is narrower than the band limit of the first bank. The banks are arrayed as n "levels" or "time domains, td." The banks are the hierarchical network (a cortex), and transducers are (associative) memories.

A model Neurex was built and studied. Data were 50 MB to 10 GB samples of text, data base, and images - black/white, grey scale, and high resolution in several spectral bands. Memories at td, $S(m_{td})$, were plotted against outputs of memories at td-1. $S(m_{td})$ was Boltzman distributed, and memory frequencies exhibited Self-Organized Criticality (SOC) [Bak *et al.* (1987) Phys Rev Lett: *59*, 381-384]; i.e., "$1/f^{\beta}$" after long exposures to data. Whereas output signals from level n may be encoded with $B_{output} = O(-\log_2 f^{\beta})$ bits, and input data encoded with $B_{input} = O([S(td)/S(td\text{-}1)]^n)$, $B_{output}/B_{input} \ll 1$ always, the Neurex determines a canonical code for data and it is a (lossless) data compressor. Further tests are underway to confirm these results with more data types and larger samples.

## 1. Introduction

Electronic libraries holding $10^{15}$ bytes (one petabyte, PB) of information are being planned. The Library of Congress' Global Knowledge Network, NASA's EOS/DIS, the Sequoia earth science project, and seismic data collections at major oil companies may be measured in petabyte units within ten years [1][2][3]. These large libraries will adopt information system technologies that compress data, store and retrieve information from very high density storage devices, and answer queries using knowledge of the information in the library. The Neurex™ memory engine for mass storage applications, being developed by our firm Triada, Ltd., Ann Arbor, Michigan, should provide features large libraries will require. And it is being considered for beta installation by several large libraries. Here we introduce the technology behind Neurex; N-Gram™, learning and memory theory. We review the N-Gram associative memory form that equates information with storage locations. We report results of tests using data samples provided by prospective Neurex users to show that Neurex losslessly compresses data at rates up to 200:1. In the attachments we illustrate the N-Gram learning transform and the Neurex machine.

How will petabytes of information be stored? How will users retrieve information from a petabyte library? Is it possible to just automate card catalogs or expand the scale of file based or database management systems? The first question appears to have been answered. The other questions are actively debated under the rubric of *metadata*.

Data storage technology now can support petabyte storage systems using mini-supercomputers running UNIX and UNITREE, redundant arrays of inexpensive disks (RAID), and petabyte libraries comprising helical scan tape [4][5][6]. A large storage system model is being built at the National Storage Laboratory at the Lawrence Livermore National Laboratory[7][8]. With it data storage technology advances from a role subservient to computers to an egalitarian role in a network of computing devices. But key issues are unsolved, including support for high performance computing [9].

The *metadata* problem requires integrating storage management with data management and current technology does not solve the problem [10]. First, databases do not extend to tertiary stores [11]. Second, unstructured data requires many file names. Suppose text files are .01 MB and image files are 20 MB. The catalog for a 1 PB system then has 1 billion names. 2.5 kilobytes per name requires a 2.5 terabyte card catalog on fast storage. The naming problem can be experienced today firsthand. Issue a global query on Internet. It may be days before the system contacts tens of thousands of nodes and it might not come back [12].

*Meta*-data is an intelligence modeling problem; data must become information. Researchers are attacking it from two directions. We call one the Turing paradigm; the other we call the connectionist paradigm [13].

The Turing paradigm works from the *top down*. One studies a phenomenon, e.g., intelligence, to deduce an algorithm that will operate on input data and output the phenomenon of interest. Ostensibly a metadata transformation is sought to map data into information by a finite number of instructions that can be executed on a computer in polynomial time, and the program can be self modifying. Artificial intelligence (AI) attempts to provide a complete solution, while database theory (DBT), information retrieval (IR), and information filtering (IF) attack parts of the problem.

Although AI, DBT, IR, and IF have progressed during the past twenty years, a general transform for changing data into information has not been discovered [14]. Notwithstanding the problems inherent in intelligence modelling,

research according to the Turing paradigm is robust and new publications are numerous. [15] is about (AI) implementation issues. [16] is a classic AI reference. [17][18] review problems in image representation and understanding. [19] and accompanying articles review database theory. [20] defines a general IR system model. [21] explains basic concepts in IR and compares these with IF, and [22] reviews an AI application at the U.S. Census Bureau. An intriguing extension of AI learning models, which has a flavor of fuzzy logic and poses interesting issues when juxtaposed with semantic logic, is relevance feedback theory [23]. Finally, no review of AI is complete without referencing Japan's Fifth Generation Language Project [24].

Solutions following the Turing paradigm that employ indexing methods could exacerbate the storage problem and not solve the metadata problem. Database keys and indices within text and images must be in primary memory but primary memory costs are high. If indices measure $10^{10}$ bytes and more, total system costs could measure ($ U.S.) $10^{7}$ or more. Indices in tertiary storage expand storage costs and they are useless until data is moved to primary storage.

The connectionist paradigm works from the bottom up and is a branch of cellular automata theory. Cellular automata are "discrete dynamical systems whose behavior is completely specified in terms of a local relation" [25]. The phenomenon exhibited by a cellular automaton is expressed by a behavior rule for the individual components. Hence, a researcher who wants a cellular automaton to act intelligently must discover a local relation that globally will make the automaton seem intelligent. Most current research defines local relations as either the spin glass model of John Hopfield, or the Boltzmann machine model of Terrence Sejnowski [26][27]. An alternative to the energy function models is the autocorrelation model [28]. Kevin Knight surveys the field, and he contrasts the Turing and connectionist paradigms [29]. Three survey works are [30][31][32]. Self-organizing systems and a review of several of the problems mentioned here is in [33]. Marvin Minsky wrote rules for a novel automaton that departs from the connectionist model [34].

The connectionist paradigm also does not solve the metadata problem. First, memory is not invertible and given the continuous functions of the local relations the capacity is unknown in general [35]. Second, neural networks can fall into spurious minima and not yield correct answers [36]. Third, they are not entirely bottom up because behavior derives from *a priori* training procedures. Example: A network taught to recognize type written characters will not recognize hand print. [37] gives a more complete introduction to problems in machine learning including an introduction to the literature of machine learning paradigms.

The above argues that the metadata problem cannot be solved following either the Turing paradigm or the connectionist paradigm. The crux of the metadata problem is that its solution may depend on answering a more profound question, *what is meaning*, which begs another profound question, *what is mind*? [38] Study of these go to the heart of philosophical enquiry dating back to antiquity, and have been investigated by the world's greatest minds: in jargon, the problem is highly non-trivial.

Triada is developing what we believe to be a robust solution to the metadata problem. It is obtained by attacking the metadata problem as a learning transform problem. Learning in our model is a metric tensor that under suitable conditions reversibly maps vectors of data into memories that are forms, i.e., information, and thus departing philosophically from the above paradigms. We study a general model of an observer equipped with a bank of band limited transducers attached to a hierarchical memory structure. The observer randomly walks through a region bounded by its lifetime and containing objects that reflect photons thereby allowing the observer to "see" the objects. The observer's input transducers register events within their frequency band limit by outputting a signal to the discrete learning transform. A set of ordered signals is a vector that is mapped into a memory form by the learning

transform. The set of all forms recorded this way describe the path taken by the observer, and transforming these into their dual space equivalent constitutes a faithful memory of the objects along the path in the neighborhood of the observer. Thus, memories are *p-forms* and electromagnetic events are *n-vectors*. Our conclusion is that information is a form while data is a vector, and the learning tensor is the desired metadata transform, that is, *memory and information are the same phenomenon*. The transform in hand we introduce the Neurex memory engine that embodies it. We present results of tests using a Neurex prototype and discuss the benefits afforded by this new technology. In particular, we will show results indicating 85:1 compression of text and 341:1 of fax image data. We will conclude with a review and talk about future research directions.

## 2. N-Gram Learning and Memory Theory

The learning transform acting on a field of electromagnetic events and registering differential patterns, or forms, is called a Poisson process [39]. Individual memories accumulate at each level of the memory hierarchy at a rate that decays exponentially, their probability of occurrence within any subregion of the entire region bounded by the observer's lifetime is Poisson distributed, the length of the path required to completely map all objects into the observer's memory is gamma distributed. Because sums of Poisson distributed random variables are Poisson distributed the growth of the entire memory is readily characterized.

Energy values (the memory forms) as memory is well accepted; minimal energy states are memories in both Hopfield and Boltzmann neural networks. Recently Friedland and Rosenfeld recognized a class of objects using an energy function [40]. Their work followed Geman and Geman who showed the Gibbs (Boltzmann) distribution and the characterization of an image as a *Markov Random Field* (MRF) were equivalent, where an image is a pair of matrices, the matrix of grey levels, and its dual, the edge matrix. Eugene Margulis applies a related concept in multiple Poisson models of word distributions in full text documents [41]. He demonstrated empirically that the meanings of particular words are multiply Poison distributed according to distribution parameters $\pi_i$ and $\lambda_i$; where i counts the number of subjects, $\pi_i$ is the probability the i'th subject is covered in a document, and $\lambda_i$ is the mean occurrence of a word in the i'th subject.

We hypothesize the existence of measures $\lambda_{\beta,\alpha}$ of local information content, and other measures $\mu_{r,\beta}$ of global information content. The measures $\mu_{r,\beta}$ are the boundaries of the r volumes that contain the $\lambda_{\beta,\alpha}$, both sets of measures are found during a point-wise continuous random walk through all parts of an energetically conservative data field. Should a path of the walk be restricted to a surface of constant energy then only events with the same information will be found. But, these are elementary results in probability theory where the gamma and Poisson distributions are shown to be related, and the Boltzmann distribution is a special case of the gamma distribution [42][43]. In particular, the sum of t Boltzmann distributed random variables with parameter $\lambda$ is gamma distributed with parameters (t, $\lambda$), and the probability that there are k occurrences of an event, say a particular word appears in an interval of length t is Poisson distributed. The equivalence of Markov and Poisson processes then obtains by [44]. Hence Markov $\Leftrightarrow$ Boltzmann $\Leftrightarrow$ Poisson.

The N-Gram memory model is an elementary implementation of the above ideas. A data stream is input to the N-Gram algorithm. The stream is parsed into sets of words according to rules that are empirically determined to be appropriate for the data type. The processor receiving the input word pattern searches its local memory is to determine if the input word pattern has previously occurred. If it has previously occurred, a counter is incremented and a signal representative of the storage location of the pattern is output to the subsequent processing level. If the pattern has not previously occurred, it is assigned a place in storage, a signal representative of its new location is output to the subsequent processing level, and a counter is incremented to the value 1. The signals output to the next

processing stage are similarly treated.

We want to know the size of the output stream after n levels and we want to know the size of the hierarchical memory after x bytes of data have been read. We first determine the size of the memory structure.

The N-Gram Memory can be represented an arrays of numbers. The numbers may be from the set of integers ($\mathbf{I}$), rationals ($\mathcal{Q}$), real ($\mathbf{R}$), or complex ($\mathbf{C}$). Elements in each row, or level, in the network are mapped into the level immediately above it, and each element in a level is the image of a mapping of elements in the level immediately below it. Let us assume that the level elements are rank ordered by relative frequency from most to least frequent.[1] Let $X$ be a data stream comprised of signals $\mathscr{E}_j$, $0 < j \le \mathfrak{I}$, $\mathfrak{I}$ a nonzero integer, from a nonempty range of signals measured by (real or complex valued) frequencies, $f_i < \mathscr{E} < f_f$, $0 < |f_f - f_i|$. Thus, $\mathscr{E}_j$ is a signal (most commonly, an n bit binary code) representing any frequency in the $j'th$ partition of the range $|f_f - f_i| \; / \; \mathfrak{I}$. Define a recognition event in an N-Gram Associative Memory Network as the image of a function $\mathfrak{L}$ from any nonempty string $S$ of signals $\mathscr{E}_j$ along a data stream $X$. Hence, in the most general case, the N-Gram Associative Memory Network is the codomain of $\mathfrak{L}$ where the domain of $\mathfrak{L}$ is any "piece wise continuous" stream of signals.

Now, let $T = |t_{final} - t_{initial}|$ be any nonzero time interval. Let $\mathfrak{L}$ be any invertible function that rank orders its image by relative frequency, from most to least frequent. Above we said the N-Gram Memory, $N$, can be represented by an array of size $Cl_{max}$ *by* $TD$ with integer elements. Let the first level of $N$ be the image of $\mathfrak{L}$ operating on a data stream $X$ comprised of signals $\mathscr{E}_j$, where each signal is n bits long. Suppose $\mathfrak{L}$ begins sampling $X$ at time $t_{initial}$ by consistently selecting $s$, $s \in \mathbf{I}$, $0 < s$, nonoverlapping contiguous signals from $X$. Hence, every $S^1$ has word length $W = s \times n$ bits. Let $x_1, x_j \in \mathbf{I}$, be the number of words $S^1$ sampled by $\mathfrak{L}$ during an interval $T$. Note, $x^1 = 0$ at time $t_{initial}$. Then the first level of $N$, $M_1$, is the set

$$M_1 = \{ m_{1,i} \mid m_{1,i} = \mathfrak{L}(S^1);\ |a| < |m_{1,i}| < |b|;\ \text{a, b, and } m_{1,i} \in \mathbf{R} \},\ \text{where}$$

$||b| - |a|| \ge \lceil Cl_{\max}(1) \rceil$, $Cl_{\max}(1)$ is an empirically determined constant, and $\lceil\ \rceil$ is the greatest integer function.

We call an element $m_{td,i}$ a "memory," and the level number is $td$, $1 \le td \le TD$. Note, also, that $\mathfrak{L}$ is invertible and its image is discrete and rank ordered, therefore, without loss of generality we define a new function I that substitutes for each $m_{1,i}$ its integer position, i.

Define the second level in $N$ like the first level as the rank ordered image of $\mathfrak{L}$, $m2_{,i} = \mathfrak{L}(S^2)$. Here $S^2$ contains $s^2$ contiguous signals $\mathscr{E}$ from a data stream $X$. Every $S^2$ is now a digital word of length $W = s^2 \times n$ bits. Suppose, we define a binary function $\mathfrak{L}^*$, that has as its image the position values i of the elements of the second level $M_2$ of $N$, and $\mathfrak{L}^*$ takes as its arguments the two recognition events (position values) of the elements of the first level $M_1$ of $N$ that are the level one images of the first and second halves of the signal $S^2$. Let $S^1(x^1_u)$ and $S^1(x^1_v)$ be the first and second halves, respectively, of a signal $S^2$ from $X$: u and v are indices. Then,

$i_2 = \mathrm{I}(m_{2,i}) = \mathfrak{L}^*[\ \mathfrak{L}(\mathrm{k,l})\ ] = \mathfrak{L}^*[\ \mathfrak{L}(\ m_{1,k}\ ), \mathfrak{L}(\ m_{1,l}\ )] =$

$\mathfrak{L}^*[\ \mathfrak{L}(\ S^1(x^1_u)), \mathfrak{L}(\ S^1(x^1_v))] = \mathfrak{L}[\ S^1(x^1_u) \wedge S^1(x^1_v)\ ] = \mathfrak{L}[\ S^2\ ]$, where $\wedge$ is the concatenation operator.

Therefore, the second level of memories, $M_2$ in $N$, is the set $M_2 = \{\ i_2 \mid i_2 = \mathrm{I}(m_{2,i}) = \mathfrak{L}\{S^2) = \mathfrak{L}^*(p,q)\ \}$, where $p,q$ are recognition events in level one, i.e., $p = \mathfrak{L}(S^1(x^1_u)$ and $q = \mathfrak{L}(S^1(x^1_v)$;

---

[1] If the $m_i$ are integers, i.e., $m_i \in \mathbf{I}$, then $\mathfrak{L}$ is an indexing function. If the elements of the array are real (or rational), i.e., $m_i \in \mathbf{R}$ ($\mathcal{Q}$), and $a = 0$, $b = 1$, and the relation above is $a \le m_i$, then $\mathfrak{L}$ is a correlation function. If the elements are complex $\mathfrak{L}$ is a contraction.

$i_2 \in \mathbf{I}$; $|a| < |m_{2,i}| < |b|$; a, b, and $m_{2,i} \in \mathbf{R}$ }; $||b| - |a|| \geq \lceil Cl_{max}(2)$, and $Cl_{max}(2) \rceil$, is an empirically determined constant

We can now define any memory level as the ordered set of integers $M_{td} = \{ i_{td} \mid i_{td} = I(m_{td,i}) = \mathcal{L}\{S^{td}\} = \mathcal{L}^*(p,q) \}$, where the signal $S^{td}$ is a binary word of length $W = s^{td} \times n$ bits; $p,q$ are recognition events in level td - 1; $i_{td} \in \mathbf{I}$; $|a| < |m_{td,i}| < |b|$; a, b, and $m_{td,i} \in \mathbf{R}$ }; $||b| - |a|| \geq \lceil Cl_{max}(td) \rceil$, and $Cl_{max}(td)$ is an empirically determined constant.

N-Gram technology is the study of the N-Gram Memory to better understand human knowledge, and to invent and develop more efficient information management systems. We obtain the empirical constant $Cl_{max}(td)$

$$CL_{max}(td) = \frac{CL(x_{td})}{(1 - e^{-\lambda x_{td}})} \tag{1}$$

where, $\lambda$ is the mean of the information density of the data $X$, $Cl(x^{td})$ are the number of memories accumulated after $x^{td}$ events, and $0 \leq x^{td}$ is the number of nonoverlapping contiguous signals $S^{td}$ from $X$.

Equation (2) shows a relationship between the relative frequency of a memory at level $td$, $m_{td,i}$, and its rank in the relative frequency ordered list of memories at that level. This equation is related to (1) by the information mean density value, $\lambda$.

$$\begin{gathered} 2\lambda = f^{c,i} N_{c,i}, \\ \textit{whence,} \\ I(m_{TD,i}) = i_{TD} = \lceil \frac{2\lambda}{f^c} \rceil, \end{gathered} \tag{2}$$

$f^{c,i}$ is the (relative) frequency of the memory $m_{td,i}$ and $c$ is the class number, therefore, $N_{c,i}$ is the i'th memory at level $td$. $c = \lceil \log_2(f^{c,i}) \rceil$. The total number of classes, $C_{td}$, that form at level $td$ is exactly

$$C_{td} = \lceil \frac{1 - f^{0,0}}{2\lambda} \rceil \tag{3}$$

Therefore, the total number of memories at level $td$, is

$$Cl_{max}(td) = 2\lambda \sum_{c=1}^{C_{td}} \overline{f^{-c}}, \tag{4}$$

where $f^c$ is the class frequency.

Suppose $X$ has a density $\lambda$ at every $td^2$. Then using either (4) or (1), we calculate the number of memories in $N$ formed after it has observed $X$. The length of $X$, $|X|$, must be much longer than $Cl_{max}(TD)$, the number of unique signals $S_{TD}$ that occur in $X$; say that the length of $X$ is greater than an integer N > 10: i.e., let the bit length measure be $|X| > N\ Cl_{max}(TD)$ ($s^{td}$ x n). Thus, the number of memories $M$ contained in a network $N$ is $M = TD\ x\ Cl_{max}(td)$.

The N-Gram algorithm $N^*$

(i) parses a data stream $X$ into signals $S_{td}$ that are binary words of size $W$, as defined above,
(ii) maps every $S_{td}$ in $X$ into one and only one element $m_{td,i}$ of $N$; and
(iii) outputs a data stream $N^*(X) = m_{TD,i}(x)$, where $x$ is the number of signals $S_{TD}$ input to $N^*$, and the output is ordered as $x = 1,2,3,...$

Each signal $S^{TD}$ has word length $W$. The length of an output word $N^*(X)$ is $W^* = \lceil \log_2(Cl_{max}(TD)) \rceil$. Hence, the density improvement ratio $\rho$ achieved by $N^*$ as it processes $X$ is simply, $\rho = W/W^*$. If $N$ contains fewer than $M$ memories then the density improvement ratio is degraded by a factor $r$, where $r$ is of the order $O_{td}(r) \approx 2^{-c+1}$, where $td$ is the lowest level at which $Cl_{td}(x) < Cl_{max}(td)$, and $c$ is the corresponding frequency class. In this case the density improvement becomes $(1-O(r))\rho = W/(W^*+r')$, where $r' = \log_2(Cl_{td}(x))$.

## 3. Neurex System Tests

The machine embodiment of N-Gram learning and memory theory is called Neurex™ and it is patented [45]. Two prototype Neurex were built and tested using samples of data to (1) test predictions of N-Gram Theory, (2) measure memory populations, and (3) determine performance parameters. They were not designed to benchmark I/O performance nor to reduce data samples for compressed storage. Rather, both were designed to gather statistics to determine the relationship among the size of the memory structure, the amount of density improvement obtained with a given memory structure, the amount of physical storage that would be needed for a memory structure, and the distribution of the memories within lists of memories created by the N-Gram algorithm.

The first prototype was a set of boards with four Inmos Transputers installed in a 500 megabyte solid state disk (SSD) loaned to us by Zitel Corporation. The N-Gram algorithm was written in the "C" programming language. The SSD held a partial N-Gram Memory. The Neurex was linked by serial ports on the Transputers to Transputer boards installed in two IBM AT compatibles. The compatibles provided the programming environment, and they were used to load programs and test software, to supply test data, and to hold statistics gathered during test runs.

The N-Gram algorithm mapped patterns in the input data stream into the N-Gram memory array stored in the Zitel RAMDisk. Two memory classes were created: those having met a predetermined threshold value and which are stored permanently, and those which have not met the threshold and are stored temporarily. Memories that have not met the threshold value, and are thus kept temporarily, are eventually excluded into the output stream. Memories that have met the threshold value are mapped into the next higher level in the memory array to determine more complex features in the data stream. The amount of space available for memories bounded the length of the data stream that could be viewed; i.e., a window was created that reduced the exposure of the Neurex to low frequency data patterns slowing the growth of the permanent memory structure. The prototype permitted periodic measurements

---

[2] The assumption that the mean information density exists over a range of levels $TD$, is valid whenever the longest signal $S_{TD}$ is small compared to the "field of view" of an N-Gram associative memory network $N$.

of the memories accumulated as a function of the number of events.

We also built a prototype consisting of N-Gram algorithm running on a Convex mini-supercomputer. Convex provided time on their laboratory machines and access to tape drives to load large data files. The algorithm was modified to process data in sections where every section contained only those data stream patterns that would be within the section of the memory structure in the primary memory.

**Description of Test Data Samples**

We tested samples of text, 10 bit four color images, black/white images, travel time data, data base data, a 10 gigabtye sample of 32 bit floating point numbers from a numerical analysis project at NASA Ames, and multiple spectral band data from the LandSat and NOAA 12 satellites. The text sample was 1.5 gigabytes of ASCII coded files from the University of Michigan's collection of weekly USENET Internet service articles. A 1 gigabyte sample three of LandSat scenes was provided by NASA Goddard Space Flight Center. A single scene consists of seven roughly equal sized segments, each of which represents a spectral view of the same area on the surface of the earth as viewed from the LandSat satellite. The black/white fax images were a 3.2 gigabyte sample of bank check images. The relational data base contained typical corporate records. The sample was 4.4 gigabytes long.

**Test Results**

The tests were designed to measure the information density of the data samples, and to calculate a compression ratio using the above equations.

The information density for each data sample was obtained and it was used to extrapolate compression results shown in Table I. The fax image sample required approximately 500 million memories to achieve a density improvement ratio of 341:1. The text data sample reached 85:1 with a 1.6 billion memories. To obtain a 43:1 density improvement the commercial data base required only 280 million memories. The samples that were most dense with information were the satellite images. We were estimated the size of a memory structure for these high resolution images would be 3.6 billion memories and it would achieve a density improvement of 73:1. The worst performance was with the seismic and floating point matrix samples, however, these were said to be incompressible using standard compression techniques (according to the owners of the data).

Table I: Neurex Data Compression Performance

| Data Type | No. Memories | Output Code Word Length | Input Code Word Length | Compression Ratio |
|---|---|---|---|---|
| ASCII Text | $1.6 * 10^9$ | 24 bits | 2048 bits | 85:1 |
| Fax Image | $5.0 * 10^8$ | 24 bits | 8192 bits | 341:1 |
| Seismic | $5.2 * 10^7$ | 24 bits | 64 bits | 2.7:1 |
| LandSat (8-bit pixels) | $3.6 * 10^9$ | 28 bits | 2048 bits | 73:1 |

| | | | | |
|---|---|---|---|---|
| NOAA 11 (8 bit pixels) | $3.6 * 10^9$ | 28 bits | 2048 bits | 73:1 |
| Commercial Database | $2.8 * 10^8$ | 24 bits | 1024 bits | 43:1 |
| Floating Point matrix (32 bit) | $7.0 * 10^7$ | 26 bits | 32 bits | 1.23:1 |

## 4. Neurex Model Library

A model library with 36 terabyte capacity is illustrated in the attachments. Key to the feasibility of the library are the above compression results and the application of the N-Gram memory form to pattern recognition.

## 5. Conclusions

The N-Gram learning and memory model holds for a large range of data types. The compression possible with the large memory structure is significantly greater than that achieved using state-of-the-art methods. While additional test are required using data samples that are significantly larger than the memory structure size, given the stationarity and ergodicity of the samples we tested there is no reason to believe a larger sample will produce significantly different results than those given above.


1. J.A. Adams, "Multimedia Repository," *IEEE Spectrum*, p. 29, March 1993.

2. C. D. Benjamin, "The Role of Optical Storage Technology for NASA's Image Storage and Retrieval Systems," in *Storage and Retrieval Systems and Applications* (1990), SPIE, vol. 1248, pp. 10-17.

3. M. Stonebreaker, J. Frew, K. Gardels, J. Meredith, "The Sequoia 2000 Storage Benchmark," in *Proc. 1993 ACM SIGMOD* (Wash., DC), P. Buneman & S. Jajodia, eds., SIGMOD Record, 22, Issue 2, June 1993, p. 3.

4. R. H. Katz, G. A. Gibson, and D. A. Patterson, "Disk System Architecture for High Performance Computing," *Proc. IEEE*, Vol. 77, No. 12, December 1989, pp. 1842-1858.

5. D. Lancaster, "Convex and the Fileserver Market," in *Proc. Seminar Series, Mass Storage and Fileserver Solutions for Networked Computer Systems*, Convex Computer Corporation publication, Richardson TX, 1992.

6. K. Ishii, T. Takeda, K. Itao, and R. Kaneko, "Mass Storage Technology in Networks," in *Storage and Retrieval Systems and Applications* (1990), SPIE, Vol. 1248, pp. 2-9.

7. S. Coleman and S. W. Miller, eds. "Mass Storage System Reference Model: Version 4." IEEE Technical Committee on Mass Storage Systems and Technology, May 1990.

8. S. Louis, S. W. Hyer, "Applying IEEE Storage System Management Standards at the National Storage Laboratory," in *Proc. Twelfth IEEE Symposium on Mass Storage Systems*, IEEE Computer Society Press, Los Alamitos, CA, S. S. Coleman, ed., 1993, 55-62.

9. W. Myers, "Supercomputing 92 reaches down to the workstation," *Computer*, IEEE Comp. Soc., Vol. 26, No.1, January 1993, pp. 113-117.

10. S. S. Coleman, R. W. Watson, R. A. Coyne, and H. Hulen, "The Emerging Storage Management Paradigm," in *Proc. Twelfth IEEE Symposium on Mass Storage Systems*, S. S. Coleman, ed., IEEE Computer Society Press, Los Alamitos, CA, 1993, pp. 101-110.

11. M. Carey, L. Haas, M. Livny, "Tapes Hold Data Too: Challenges of Tuples on a Tertiary Store," in *Proc. ACM SIGMOD*, P. Buneman and S. Jajodia, eds., ACM, NY, 1993, pp. 413-417.

12. J. J. Ordille and B. P. Miller, "Database Challenges In Global Information Systems," in *Proc. 1993 ACM SIGMOD*, P. Buneman and S. Jajodia, eds., ACM, NY., 1993, pp. 417.

13. M. Conrad, "Molecular Computing Paradigms," *Computing*, Vol.25, Nov. 1992, pp. 6-9.

14. M.V. Wilkes, "Artificial Intelligence as the Year 2000 Approaches," *Communications of the ACM*, Vol. 35, No. 8, August 1992, pp. 17-25.

15. B.R. Gaines and M. L. G. Shaw, "Eliciting Knowledge and Transferring It Effectively to a Knowledge-Based System," *IEEE Trans. on Knowledge and Data Engrg*, Vol. 5, No. 1, February 1993, pp. 4-14.

16. P. R. Cohen and E. A. Fiegenbaum, eds., *The Handbook of Artificial Intelligence*, Vols. I-III, William Kaufmann, Inc., Los Altos, CA, 1982.

17. S-K Chang and A. Hsu, "Image Information Systems: Where Do We Go From Here," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 4, No. 5, Oct. 1992, pp. 431-442.

18. C. C. Weems, E. M. Riseman, A. R. Hanson, "Image Understanding Architecture: Exploiting Potential Parallelism in Machine Vision," *Computer*, Vol. 25, No. 2, February 1992, pp. 65-68.

19. A. Silberschatz, M. Stonebreaker, J. D. Ullman, "Database Systems: Achievements and Opportunities," *SIGMOD RECORD*, ACM Press, Vol. 19, No. 4, December 1990, pp. 6-22.

20. J. Tague, A. Salminen and C. McClellan, "Complete Formal Model for Information Retrieval Systems," in *Proc. 14'th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, 1991, pp. 14-20.

21. N. J. Belkin and W. B. Croft, "Information Filtering and Information Retrieval: Two Sides of the Same Coin?," *Communications of the ACM*, December 1992, Vol. 35, No. 12, December 1992, pp. 29-38.

22. R. H. Creecy, B. M. Masand, S. J. Smith, D. L. Waltz, "Trading MIPS and Memory for Knowledge Engineering," *Communications of the ACM*, Vol. 35, No. 8, August 1992, pp. 48-64.

23. IJ. J. Aalbersberg, "Incremental Relevance Feedback," *Proc. 15'th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, NY, 1992, pp. 11-22.

24. E. Shapiro and D. H.D. Warren, eds., "The Fifth Generation Language Project: Personal Perspectives, *Communications of the ACM*, Vol. 36, No. 3, March 1993, pp. 46-103.

25. T. Toffoli and N. Margolus, *Cellular Automata Machines*, MIT Press, Cambridge, Mass., 1987, p. 5.

26. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. National Academy of Sciences*, Vol. 79, pp. 2554-2558.

27. D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, Vol. 9, pp. 147-169.

28. S. Amari and K. Maginu, "Statistical Neurodynamics of Associative Memory," *Neural Networks*, Vol. 1, 1988, pp. 63-73.

29. K. Knight, "Connectionist Ideas and Algorithms," *Communications of the ACM*, Vol. 33, No. 11, November 1990, pp. 72-74.

30. J. A. Anderson and E. Rosenfeld, eds., *Neurocomputing*, MIT Press, Cambridge, Mass., 1988.

31. D. E. Rumelhart, J. L. McClelland and the PDP Research Group, *Parallel Distributed Processing*, Vols. I-II, 1986.

32. I. Aleksander, ed., *Neural Computing Architectures*, MIT Press, Cambridge, Mass., 1989.

33. S. S. Iyengar and F. B. Bastani, eds., "Special Section on Self-Organizing Knowledge and Data Representation in Distributed Environments," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 4, No. 2, April 1992.

34. M. Minsky, *The Society of Mind*, Simon and Schuster, NY, 1986.

35. D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks," *Phys. Rev. Lett.*, Vol. 55, No. 14, 30 September, 1985, pp. 1530-1533.

36. M. Zak, "Terminal Attractors in Neural Networks," *Technical Support Package* for *NASA Tech. Brief*, Vol. 15, No. 7, July 1991, p. 1.

37. A. M. Segre, "Applications of Machine Learning," *IEEE Expert*, June 1992, pp. 30-34.

38. J.E.T., "Meaning," in *The Oxford Companion to the Mind*, Oxford Univ. Press, Cambridge, UK, 1985, pp. 450-454.

39. R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Macmillan Publishing, N.Y., 1978, pp. 99-102.

40. N. S. Friedland and Azriel Rosenfeld, "Compact Object Recognition Using Energy-Function-Based Optimization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 7, July 1992, pp. 770-777.

41. Eugene L. Margulis, "N-Poisson Document Modelling," in *Proc. 15'th ACM SIGIR Conf. on Research and Development in Information Retrieval*, ACM Press, 1992, pp. 177-189.

42. R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Fourth Ed., Macmillan, NY, 1978, pp. 99-109.

43. P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Probability Theory*, Houghton Mifflin, Boston, Mass., 1971, pp, 129-130, 146-7, 230-3.

44. S. Geman and D. Geman, Section IV, pp. 724-26.

45. J. M. Bugajski and J. T. Russo, "Data Compression with Pipeline Processors Having Separate Memories," U.S. Patent No. 5,245,337, Sept. 14, 1993.

# Volume Serving and Media Management in a Networked, distributed Client/Server Environment

**Ralph H. Herring and Linda L. Tefend**

EMASS®Storage Systems
Solutions from E-Systems
P. O. Box 660023
2260 Merritt Drive
Dallas, TX 75266-0023
Phone: (214) 205-6478
Fax: (214) 205-7200
lindat@Emass.Esy.COM

# Volume Serving and Media Management in a Networked, Distributed Client/Server Environment

**Ralph H. Herring and Linda L. Tefend**

EMASS® Storage Systems
Solutions from E-Systems
P. O. Box 660023
2260 Merritt Drive
Dallas, TX 75266-0023
Phone: (214) 205 - 6478
Fax: (214) 205 - 7200
lindat@Emass.Esy.COM

## 1. Introduction

Data storage requirements have increased exponentially in the last 10 years. While many things have contributed to this explosive growth, perhaps the biggest single cause is the increase in data processing capability brought on by the wide acceptance and use of supercomputers and large networks of workstations. This added processing power allows work on complex problems such as medical, digital imaging, modeling, and satellite data analysis that could not be tackled in the past. More significantly, added processing capability results in major increases in both the quantity and size of data files to be managed. Prior memory architectures have been out-dated by these changes. This results in a whole new field, the field of mass storage.

Figure 1 shows the major functional blocks of a classic mass storage system. The application program processes data and prepares it for initial storage. Access to the data by the application program is by the file name established when the data was initially stored. Other applications can share the mass storage system by common access to the file names or by use of their own names. These application programs can be on the same computer system (supercomputer, minicomputer, main frame, or workstation) or networked to the parent computer system.

The file server accepts file requests by file name. Because file storage is hierarchical, a file may be on solid state (RAM) memory, magnetic disk, optical disk, or tape. When the file server is asked to retrieve a file, it determines the file/medium relationship. If the file is stored on a medium managed by the volume server, the file server generates a media request to the volume server. After the medium is mounted, the file server receives file data from the storage drive.

The volume server accepts media requests, by media name, from the file server. The volume server maintains the relationship between each medium it manages and the associated media type and location. The volume server works with a variety of sizes and types of media. Although the volume server does not control read/write operations with the storage drives, it knows drive status and can maintain mount statistics and request queues for each drive.

A mass storage system can consist of several robotic and manual archives offering storage for several media types chosen for a variety of reasons (cost, convenience, speed, reliability, etc.) An archive houses the storage drives and delivers media to the drives. An archive recognizes media by external labels, so it has no need to know the information on the media. Storage drives provide the means to store and retrieve individual files. Storage drives interact directly with the file server to pass file data. Several drives can be associated with a single archive.

The E-Systems Modular Automated Storage System (EMASS) is a family of hierarchical mass storage systems providing complete storage/"file space" management. The EMASS volume server provides the flexibility to work with different clients (file servers), different platforms,

and different archives with a "mix and match" capability. This volume server implementation encompasses the mass storage functions shown in figure 1. The EMASS design considers all file management programs as clients of the volume server system. System storage capacities are tailored to customer needs ranging from small data centers to large central libraries serving multiple users simultaneously. All EMASS hardware is Commercial-off the Shelf (COTS), selected to provide the performance and reliability needed in current and future mass storage solutions. All interfaces use standard commercial protocols and networks suitable to service multiple hosts. EMASS is designed to efficiently store and retrieve in excess of 10,000 terabytes of data. Current clients include CRAY's YMP Model E based Data Migration Facility (DMF), IBM's RS/6000 based Unitree, and CONVEX based EMASS File Server software.

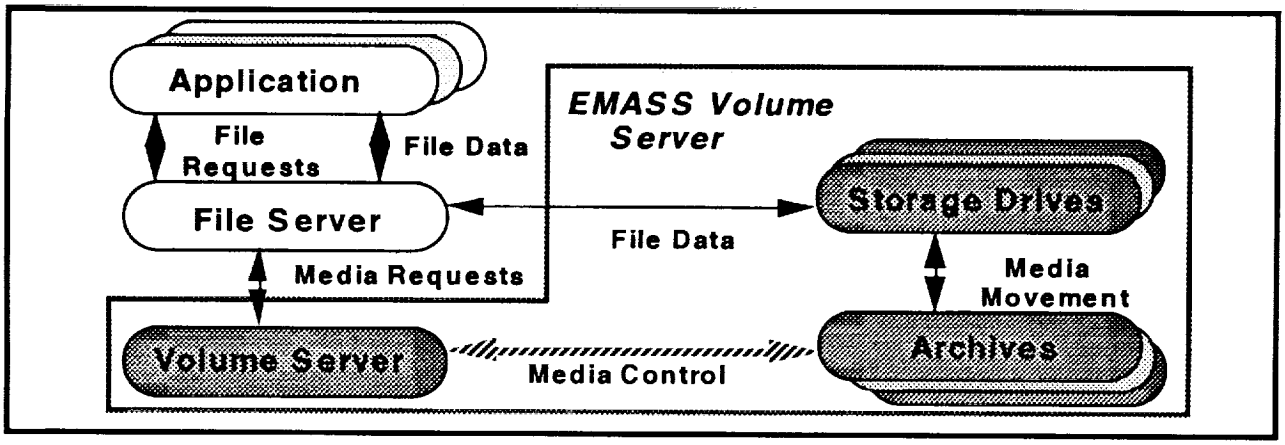



Figure 1 Mass Storage System Components

The VolServ™ software provides the capability to accept client or Graphical User Interface (GUI) commands from the Operator's Console and translate them to the commands needed to control any configured archive. The VolServ system offers advanced features to enhance media handling and particularly media mounting such as: automated media migration, preferred media placement, drive load leveling, registered MediaClass™ groupings, and drive pooling.

## 2. Mission

Provide Transparent Media and Drive Management The EMASS volume server provides the ability to accept and execute defined commands for media within its domain. The volume server finds and moves media based on logical name. If the request involves a mount, the volume server finds a storage drive compatible with the medium and accomplishes the mount. If the request involves media movement between archives, the volume server manages the move without involving the client. The volume server system can be applied to a large range of configurations with storage options involving data rates, media types, number of storage locations, and number of drives. The volume server system provides data in readily accessible, near immediate storage for purposes such as: history (archival backup), redundancy (data security), overflow (near line recovery of data as needed),buffer (temporary storage for later processing), and data distribution and transfer. Many applications require a single volume server system to provide for multiple networked clients. These clients do not have to be the same type of computer and may or may not share data, drives, or media. The volume server satisfies this mission with a design that can be used equally well for any of the listed purposes.

Minimize Impact of Utilizing Emerging Storage Technologies The EMASS volume server employs an object-oriented implementation to provide the ability to add new archives, drives, and interfaces in a modular manner. Existing applications can be preserved while new ones are added by including their specific control and status interface characteristics. The evolution in storage systems has been so rapid that any other approach would doom a volume server to obsolescence in the near future. The volume server uses COTS archives and drives

with close attention to industry standards. The file server interface (volume name) need not change even to add new robotic archives, because this interface incorporates the "transparent" media location capability. Further, the VolServ software is both modular and portable as demonstrated through added archive types and porting to multiple process control computers, including SUN, IBM RS/6000, and CONVEX.

Conform to Industry Standards The IEEE recognized the need for a standardized way to structure memory storage systems through the development and release of its IEEE Mass Storage System Reference Model, Version 4 and the on-going work of Version 5. While this model is not yet released as an industry standard, it is being developed to allow and encourage vendors to develop inter-operable storage components that can be combined to form integrated storage systems and services. This model recognizes separation of the memory architecture into component elements including file management and volume management. The EMASS VolServ software provides the major Physical Volume Library (PVL) functions of centralized management of storage media, control of storage media architectures (PVRs), and automation of mounting and dismounting media into drive devices. The VolServ software is also designed to support multiple independent client systems. The VolServ software conforms to the concepts of the IEEE MSS Reference Model, ensuring it can readily adapt to future innovations in media storage architecture.

## 3. Library Services

The EMASS VolServ software represents the most complete media and drive management package available in the industry. The VolServ software provides the capability to accept client or Graphical User Interface (GUI) commands and translate them to the appropriate commands to control any supported archive. The VolServ software can service a variety of robotic archives and manual archives as shown in figure 2. Client commands are received through a layered Ethernet™ interface featuring a Remote Procedure Call (RPC) communication path; multiple clients can share the same interface. Operator commands are provided on a series of screens using the OSF/Motif GUI.

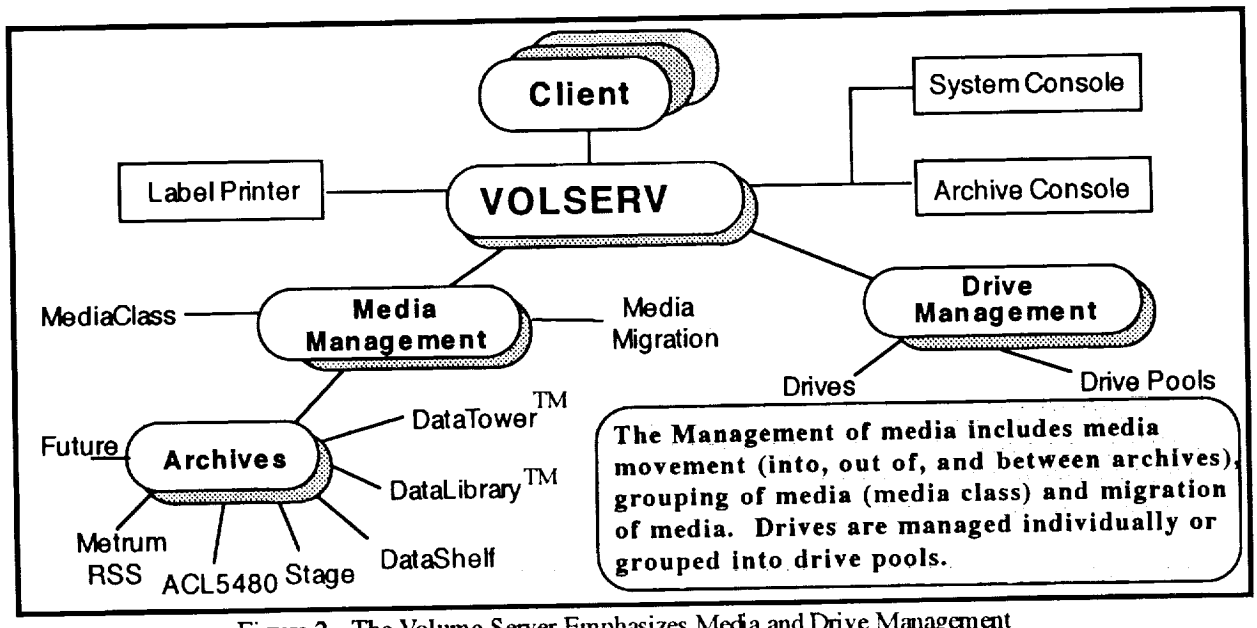



Figure 2 The Volume Server Emphasizes Media and Drive Management

**Centralized Media Management** The volume server provides a complete media management capability. The VolServ software automatically locates media within its domain. For example, the volume server receives a mount request, locates the requested medium, mounts the medium, and returns to the client the drive on which the medium was mounted. The volume server supports media migration between archives based on media type or media class.

Either source or destination migration archive can be robotic or manual. If multiple archives support the same media type, media can be migrated from one archive to another and, upon reaching the migration threshold of the second archive, to still a third archive.

**Centralized Management of Storage Drives** The VolServ software allocates storage drives for use by the client and controls placement of media into and out of the drives. The client provides control of read/write activities to/from the medium and releases the medium from the drive when finished. In automated archive systems, the use of storage drive types is restricted by the archive architecture. Manual archives can include drives of a variety of types. Each Archive Manager console has a screen which supports mounting and dismounting media. The VolServ software identifies which medium and drive to use.

**Categories of Storage** The volume server supports three categories of media storage. These are:

- Media within a robotic archive and available for near immediate data recovery by the client. The volume server assures media contained within an archive are suitable for mounting on drives associated with the archive.

- Media in a manual archive handled by an archive operator. The volume server provides clear operator instructions via Archive GUI consoles for media mounting on drives considered part of the manual archive and/or movement associated with drives contained in a robotic archive.

- Media which has been checked out and currently belongs to no archive. The VolServ software maintains the history of all checked out media to simplify future check in of the media and return to active control. The check out capability is separate from the "export" capability which removes the media from all databases.

**Media Classes** The EMASS volume server provides the ability to segregate media both physically and logically. Physical separation is done through archives and is enhanced by the ability to select preferred media placement within an archive. Preferred placement is implemented through the use of media classes. Media classes are a logical segregation of media based on client control and security needs. Media classes can be assigned to span multiple archives that support the associated media type(s). When a media class spans archives, media can be freely moved between these archives and automatic media migration can be used. The volume server provides the capability to define, modify, or delete media classes during initial system configuration or subsequently. Every medium known to the VolServ system must be associated with a media class. Media classes can segregate media by date, backup, inventory, per cent of medium filled with data, type of data or any other organizational need. Media classes figure prominently in the media mount algorithm.

Figure 3 shows a system configuration with four clients, two of which connect directly to the VolServ system and the tape drives. Four MediaClass groupings and two archives are shown. Since any client recording data on media needs access to scratch media, the "Scratch" media class is associated with both archives. Client A needs access only to Seismic data. If the "Seismic" media class is limited to Archive A, Client A needs no connection to drives in Archive B. Client B needs access to all four media classes and needs access to drives in Archive A and in Archive B. Clients C and D receive data through Client B. Neither client desires Seismic data, so they create traffic primarily for Archive B. A growth path could provide drive interfaces between Client C and the drives in Archive B. Media class "Maps" spans both archives and is ideal for migration and drive load balancing.

Membership in a media class is exclusive. Every medium belongs to one media class. A media class supports one media type. Media enter a media class as they are imported into the VolServ system. A default class can be specified (usually the "scratch" media class) for media auto-imported into robotic archives. The class for a medium can be entered via the Import command. The class associated with a medium can be changed via the reclassify command. A medium can, optionally, be reclassified as part of a mount operation.

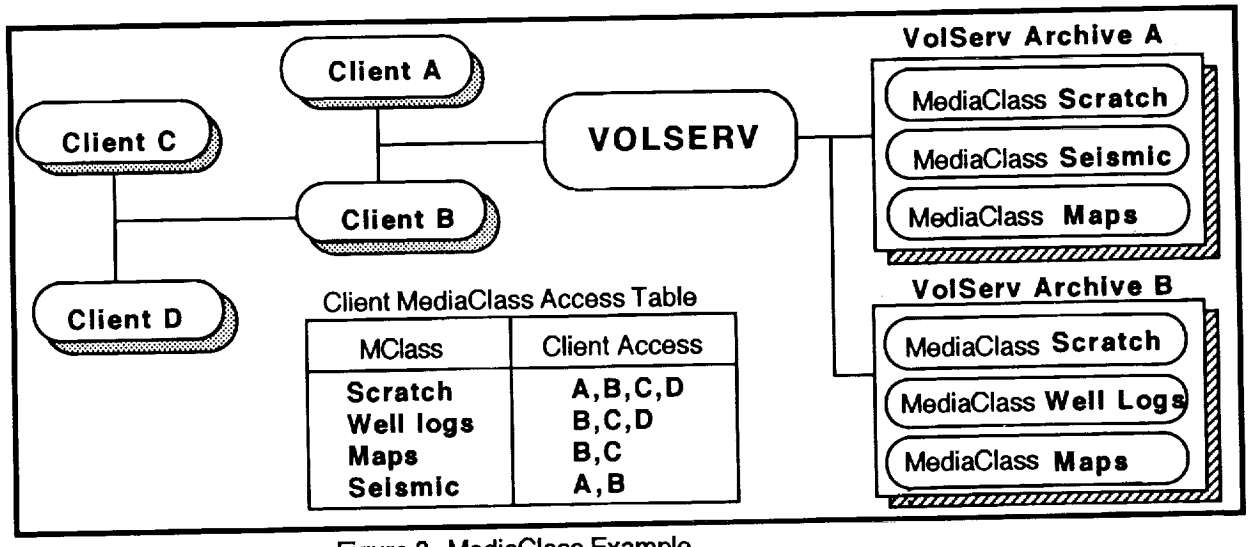



Figure 3 MediaClass Example

**Media Migration** The EMASS volume server system supports migration between archives based on media type or media class. Migration can be established at three levels: 1) automatic identification of the media to be migrated and their destination archive(s), 2) an operator notification when a user-specified threshold has been reached, and 3) no migration activity. A destination archive must be specified for each migratable media class.

A simple migration hierarchy includes a robotic archive and a manual archive. Scratch media are used and reassigned to a "permanent" media class in the robotic archive. When the media class high threshold is reached, the least-recently-used media are migrated to the manual archive. A medium can be recalled from the manual archive for use in the robotic archive. A medium can be exported (removed from the VolServ system) when it is no longer needed. Media migration can be used to accommodate other purposes:

- Migration from a robotic archive to another robotic archive - useful when one robotic archive provides better performance than another, or because the uses for the media change and different clients have access to one archive and not the other.

- Migration from a manual archive to another manual archive - useful when one manual archive is closer to the robotic archives than the other, because one manual archive has drives while the other has fewer or no drives, or because one archive is an organized DataShelf™ while the other is a "keep-it-for-awhile-longer" stage type archive.

- Migration to balance the load between similar archives with several clients as shown earlier.

- Migration set at a quantity of one (or two media) to provide one set of backups on-line while automatically migrating the previous backups to a manual archive or to a degauss and reuse category.

**MediaClass Migration** Each archive media class has associated migration parameters including capacity, high threshold, and low threshold. High and low thresholds are specified as a percentage of capacity so do not have to be updated when the capacity is changed. Capacity is the maximum number of media desired in the archive media class. High threshold is used to trigger migration processing. When automatic media migration is executed the VolServ software determines how many media must be removed to reach the low threshold and places those media on the ejection list. (Media can be removed from the eject list via the clear eject command.) An archive operator supports media migration by selecting media to be physically

ejected from the eject list on the archive's console. Once ejected, these media appear on the destination archive's enter list. An operator completes migration by physically placing them into the entry port or manual interface for adding media to the destination archive.

**MediaType Migration** The volume server also provides automatic media migration for media types. Media type migration is conducted one media class at a time. When media type migration is triggered, the media class with the highest migration priority has its fill level lowered to its low threshold. This processing is applied, iteratively, to the media class with the next highest migration priority until the fill level for the media type reaches its low threshold. Depending on migration priorities and thresholds, migration may not be applied to all media classes.

**Use of the Low Threshold** The VolServ system offers the capability to notify an operator when the number of media in a media class decreases below the low threshold. Low threshold notification can be used when scratch media are used and reassigned to "permanent" media classes or when media are exported, moved, or reclassified down to the low threshold. The client may use this information for inventory management, to keep a minimum number of scratch or in-work tapes, or other purposes. The client can ignore the notification or take an appropriate action such as adding media, reclassifying media, lowering the archive media class capacity, etc.

**DrivePools** A drive pool is a logical grouping of drives associated with one or more archives. A drive pool can frequently offer more rapid media mounting than the standard mount on a client-specified drive. Drive pools allow the VolServ software to select the best drive to satisfy a mount request. The volume server provides automatic media movement to get a medium in the same archive as the selected drive. A preferred solution is to satisfy a mount request within the archive that contains the medium. This solution is enabled by constructing a non-exclusive drive pool with at least one drive in each archive. If the medium has been relocated to a manual archive, a human is required to mount and dismount the medium. Figure 4 shows an example of drive pool organization for a system configuration with two archives, each with four drives. Drive pool 1 contains all four drives in archive A. This corresponds to the MediaClass grouping of figure 3 where all media for Client A are held in Archive A. When a drive pool contains all drives in an archive, it allows a medium to be mounted immediately on any available drive. Further, if the mount is queued, it is a candidate for the first available drive.

Drive pool 2 has two drives in each archive, ideal when the client has media classes in each archive, for example, Client B of figure 3. A client could include all drives in one archive and some drives in the other if this improves system operation. In this case, only two were chosen to ensure Client B never takes all the resources in Archive A. Drive pool 3 has two drives in Archive A. Pool 3 could be a second pool for Client A or for Client B. Client A may use pool 1 for data capture and hence want access to all four drives. Client A may use pool 3 for a less critical activity (data playback) to ensure playback operations never take all the resources. Pool 4 has only one drive. Drives can be added or deleted from a pool, so this could be a temporary state. A client may have committed to request all mounts by drive pool, but this function is of lower priority. The DataLibrary™ and the manual archives allow drives to mount more than one media type. If Archive B is a DataLibrary, pool 4 may be for D2 medium, while pool 2 is for D2 small. Drive 8 can only be mounted by requesting the specific drive.

**Medium and Drive Mounting Options** The volume server's goal is to mount a desired medium on any acceptable drive as quickly as possible by offering several options on the way the medium and the drive are chosen. These options take advantage of the media class and drive pool groupings. In each case, the mount will be queued if either the medium or the drive is busy. A sophisticated media-drive pairing algorithm selects media as follows:

- A specific medium - The VolServ software finds and mounts the user-specified medium on the nearest available on-line drive (if given a choice) or on the requested drive.

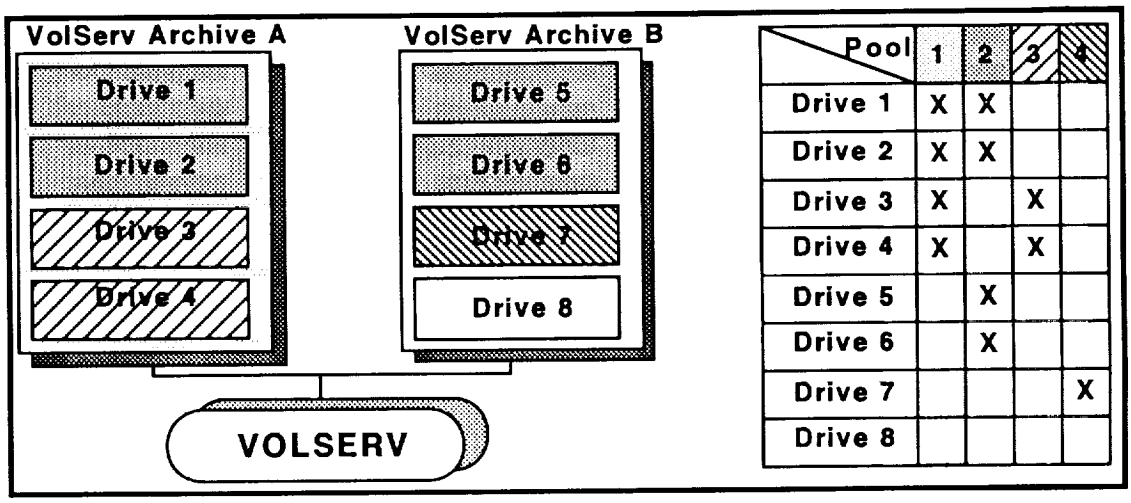



Figure 4 Example Use of Drive Pools

- A list of media - The VolServ software pairs each medium in the user-specified listed of media with an available on-line drive (if given a choice) and selects the medium/drive pair requiring the least robotic movement.

- A MediaClass - The VolServ software exercises its drive-media pairing algorithm, using any medium in the user-specified media class, to find an available mount with the least robotic movement.

The EMASS volume server allows selection of drives in several ways to assist in satisfying clients requests:

- A specific drive - If the user-specified drive is available and in a different archive from the medium, the VolServ software initiates and controls an inter-archive media movement to get the medium to the drive.

- A drive pool - The VolServ software looks for an open drive in the user-specified drive pool that requires the least media movement.

- A drive pool with exclusions - After excluding the specified drives from the drive pool, the VolServ software processes this mount request the same as a mount by drive pool request.

**Robot Allocation** For most automated archives, a specific medium, a specific drive, and a specific load port can only be reached by a single robot. To move a medium, the supporting robot is scheduled to perform the movement. The DataLibrary has an enhanced capability when two or more robots are used. If a medium and a suitable drive or load port can be accessed by more than one robot, the VolServ software determines the first robot to become available. The goal is to match a medium, a drive, and a robot to minimize movement activity time.

## 4. Archives

A volume server archive is based on the Physical Volume Repository (PVR) definition in the IEEE Mass Storage Reference Model. A single instance of the EMASS system can be composed of one or more archives of the same or different architectures. Automated archives have self-contained, robotically-accessed media storage and retrieval providing a mechanical interface to the storage drives and providing load/unload ports for entering and ejecting media. Manual archives contain no robotics, so a human operator processes each media request in accordance with commands from the VolServ software to the appropriate archive console. Manual

archives can include drives of a variety of types. Each Archive Manager console has a screen which supports mount and dismount of tapes.

The VolServ software supports four types of automated archives and two types of manual archives:

- **DataTower** stores 227 small 19 millimeter (D2) cassettes, has a single robot, supports up to four ER90™ tape drives, and provides 6 Terabytes of storage. Up to four towers can be interconnected as one archive with pass-through ports.

- **DataLibrary** provides expandable storage in increments of 4-foot cassette storage modules(CSMs), uses ER90 tape drives, and provides up to 5,000 Terabytes of storage. Each CSM can store 240 small, 192 medium, or a combination of small and medium 19 millimeter (D2) cassettes. A DataLibrary can be constructed with up to 20 aisles with up to 20 CSMs on each side of the aisle. Each aisle consists of a robot with access to any cassette in the CSMs on the aisle. Internal CSMs, drives, and load ports associated with internal aisles can be accessed by two robots. Drives can be located at either end of an aisle.

- **ACL5480** stores 288 3480 cassettes, uses one or two 3480 tape drives, and provides 58 Gigabytes of storage. Up to four 5480 units can be interconnected as one archive with pass-through ports.

- **RSS 48** and **RSS 600** store 48 and 600 T120 1/2 inch helical scan tapes each holding up to 14.7 Gigabytes of data for a total of 0.7 and 8.8 Terabytes respectively. The RSS 48 uses one or two drives, the RSS 600 uses one to five drives.

- **DataShelf**, a manual archive, stores 3480 cartridges, T120 cassettes, all three sizes of 19 millimeter cassettes, and up to 16 user-defined media types. A single DataShelf archive can support multiple media types and sizes. Storage is organized into rows, columns, shelves, and bins. Total storage capacity is limited only by facilities. This archive uses any storage drive type compatible with any supported media type.

- **Stage**, a manual archive, has the same capabilities as the DataShelf except the storage is free form. The stage archive can be used as an area to receive media for import or export or as a processing station for media needing cleaning, degaussing, certification, or other client desired processes.

The VolServ software design provides for addition or removal of individual archives from an established volume server system. Reconfiguration of an archive does not interfere with the operations of other archives. New archives and drives can easily be added to the volume server family with a minimum of effort. The volume server is modular so archive-dependent changes are constrained to the archive manager software. Clients can use multiple archives in a variety of ways: archives can be shared to provide client interusage, archives can be operated independently to provide data privacy and control, or archives can be structured in a hierarchy so media can be migrated between any combination of compatible archives. Media movement between two archives is directed by software, but performed manually. An operator can put an archive in an unattended mode. When a movement request involves an unattended archive, the VolServ software can cancel the request or wait until the archive is again attended.

## 5. Client Interfaces and Relationships

The VolServ software provides a control/status interface to the client software over a network. The VolServ software is connected to client program computers through an Ethernet or Fiber Distributed Data Interface (FDDI) connection using standard Remote Procedure Call (RPC) protocols. This connection allows multiple clients to share a volume server system. Figure 2 shows a stylized volume server environment with several potential file serving clients and the currently supported EMASS volume server archives. The client provides three

hardware/software capabilities to use the EMASS volume server, namely the file server software application, the data and control interface to each drive, and the VolServ connect interface. These capabilities reside on each client machine connected via network to the volume server. EMASS imposes no limit to the number of clients that could be connected on the network path to the VolServ control processor.

A client system is a hardware/software package performing data management services for the client's own use or as an intermediary to other client programs. The VolServ software provides a high level interface relieving the client system of the need to know the storage architecture. The volume server offers transparency by locating and moving media based on its internal database. VolServ software provides a programmatic command set that allows a client to integrate any file management program with Client Interface Software (CIS). Through this CIS, the VolServ software may communicate with current and future file systems. In addition, EMASS has an Application Program Interface (API) and a command line interface (CLI) which simplify the client's interface design by residing on each client's computer and providing the RPC network interface.

The client interface can be implemented as an application program or as a modification to an operating system. Commands sent by an application program pass the required information to the volume server. For example, the VolServ software mounts the appropriate medium to allow the application to perform its read and write activities. One volume server can simultaneously perform similar services for several clients. Operating systems may include file management functions. If the operating system is providing file management, the VolServ CIS would be included as part of the operating system. Application programs would issue commands through the operating system which access the volume server transparently and the user would not be required to learn a new application.

The client provides file server software that determines what data is recorded on each medium and tracks data location with a cross reference to specific volume name(s). The client migrates data files onto media, identifies which medium contains which file, and requests the medium to restore data. The client provides the drive interface (data and control path) to each drive and knows which files are placed on which media and at what security or privacy level. Finally, the client provides the VolServ connect interface. This interface emphasizes standards so the client machine can use the RPC interface for all command and status data passed to/from the volume server. This interface represents only one load on the client's Ethernet or FDDI network. All traffic internal to the Volume Server is conducted over a separate Ethernet path. The client needs no direct interface with the archives for robot control.

## 6. Operations and Administration

System administrators and operators work directly with the VolServ software through the GUI for configuration, reconfiguration, archive management, media management, resource allocation, and daily maintenance operations for the volume server system. The System Administrator initializes and configures the volume server system and defines the associations between volume server components. The System Administrator login ID provides access to all VolServ software functions, while the operator login ID allows access to a subset. System Administrator/Operator interface GUI runs under any Motif window based manager. The VolServ software accesses the GUI directly from the VolServ control processor's console or remotely via a network. The control processor uses the OSF/Motif™ windowing system. The VolServ software must be installed on the host and INGRES® and the X-window manager must be running. The UNIX® shell used to initiate the GUI must be configured with environmental settings established through the software installation script.

The GUI operations are grouped and accessed through three types of consoles: the System Management Console, the Archive Console, and the System Log Console:

- **System management console** - provides access to logical operations and administrative functions. This console is generated by a System Administrator/System Operator.

Access is controlled through the use of passwords. Multiple consoles can be used simultaneously by System Administrators/Operators. Console GUIs are grouped into four functional categories: Media Operations, Administration, Configuration, and Queries/Reports.

- **Archive console** - used to execute media movement commands and manage the media stored within an individual archive. A volume server system has a separate archive console for each configured archive.

- **System logging console** - displays system messages. Logging consoles are generated automatically by the VolServ software. Message levels are defined during configuration by the System Administrator. System messages, generated during system operations, provide information about events occurring within the volume server system. The logs can be displayed on one or multiple consoles and/or directed to one or multiple files. Operator options for managing the display provide for clearing the display, printing the text in the display buffer, saving the information into a file, and setting auto-scroll on and off. The upper limit of displayable information is configurable. Once the limit is reached, the oldest messages are removed as new messages enter the display buffer.

## 7. Summary

The EMASS Volume Server provides all the media management capabilities to build a mass storage system *now* using a design and current hardware that can be used well into the future. These capabilities include:

1. **Transparent media and drive management**. The client provides the volume server with a media name or class and a drive or drive pool. The volume server advises the client when the medium is mounted (or moved).

2. **Multiple archive control**. The volume server is configured to manage several types of archives. Storage flexibility is enhanced by the capability to manage multiple copies of each archive.

3. **Multiple media (and drive) types**. The volume server supports standard media types and 16 user-specified types. The volume server supports robotic mounting of drives in robotic archives and operator mounting of drives in manual archives.

4. **Easily expandable**. The same VolServ software supports one archive or a variety of archives. Archives can be added with minimal impact to existing operations. Based on designed-in modularity, when new archive types are added to the volume server design, these archives could also be added to an on-going site.

5. **Advanced mounting algorithm using media classes and drive pools**. The mounting algorithm considers media, drives, drive-load balancing, robots, and operator support (for cross-archive mounts) in choosing the best media-drive pairing for each mount.

6. **Media migration between archives**. The volume server uses media type and class capability to support automated or operator-directed media migration from any type of archive (manual or robotic) to any other type.

7. **Full-featured manual archives**. Manual archive support includes a simple table-top used for media entry, up to a multi-row shelf archive with thousands of volumes, multiple media types, and a variety of drive types.

8. **Software portability (now runs on Sun, IBM and Convex platforms)**. The software emphasizes basic UNIX concepts. It can be readily ported to other platforms running versions of the Unix operating system.

# IMPROVEMENT IN HPC PERFORMANCE THROUGH HIPPI RAID STORAGE

Blake Homan
Maximum Strategy, Inc.
801 Buckeye Ct.
Milpitas, CA 95035
Tel: (408) 383-1600
Fax: (408) 383-1616
blakeh@maxstrat.com

## RAID History

In 1986, RAID (Redundant Array of Inexpensive [or Independent] Disks) technology was introduced as a viable solution to the I/O bottleneck. A number of different RAID levels were defined, in 1987 by the Computer Science Division (EECS) University of California, Berkeley, each with specific advantages and disadvantages.

With multiple RAID options available, taking advantage of RAID technology required matching particular RAID levels with specific applications. It was not possible to use one RAID device to address all applications. Maximum Strategy's Gen 4 Storage Server addresses this issue with a new capability called *Programmable RAID Level Partitioning*. This capability enables users to have multiple RAID levels coexist on the same disks, thereby providing the versatility necessary for multiple concurrent applications.

## Architecture

Gen 4 is essentially a parallel computer. Multiple CPUs work in parallel to facilitate the asynchronous data transfer to and from up to 20 IPI-2 channels and one or two HIPPI channels.

Gen 4 utilizes a Motorola 68040 microprocessor running a powerful real-time, multitasking operating system. The 68040 is the centralized task manager for all command and control.

Each dual IPI-2 interface also uses a Motorola 68000 microprocessor running a real-time operating system and two independent microcontrollers to control the IPI-2 channels.

Gen 4 may be configured with one or two HIPPI channels, the open systems standard for high performance computing (HPC), and utilizes the IPI-3 command set.

Internally, the HIPPI interface controls data mapping between its high-speed buffers, and all IPI-2 channels. Additional dedicated hardware performs the functions required for RAID 3 or 5 data recovery.

Three additional interfaces, one Ethernet port and two RS-232 ports are available for external communication. The Ethernet is also capable of transferring data, however it is not well-suited for transferring at high-performance levels. The Ethernet port is best suited for third-party or complex data transfers where the command/status information is sent over Ethernet, and data only is sent over the HIPPI channel. This provides the capability to interface with various distributed computing solutions that are now becoming available.

Gen 4 may be managed and configured by the host using the IPI-3 command set, or from a system management console via RS-232 or Ethernet, allowing the operator to monitor real-time status of the system through a menu-driven interface.

## Investment Protection

The Gen 4 has been designed with the future in mind, utilizing industry standards. First, with the standardized HIPPI interface, users attaching a system directly to a host today, can move that same storage to a new host or to a distributed, network-attached storage architecture in the future.

Additionally, as higher performance standard channels and larger capacity disks become available, previously developed applications can be easily ported to take advantage of new storage capabilities.

## Summary

RAID technology has become the accepted solution to the I/O bottleneck in the HPC community. As HPC becomes more mainstream, it is important that users have the flexibility to mix and match hosts with the highest performance peripherals. The HIPPI standard has been a major milestone in this process. Other fabrics such as Serial HIPPI and Fibre Channel are in the early stages of standardization.

Maximum Strategy is dedicated to improving the ability of the high-performance computing marketplace to provide solutions by increasing the transaction rates, throughput, and mean time to data loss of its storage solutions. This will allow researchers and businesses to solve problems much faster, save money and resources, and make HPC more interactive for multiple users.

N94-33805

# Architectural Constructs of AMPEX DST

**Clay Johnson**

Ampex Systems Corporation
2345 Vantage Drive
Colorado Springs, CO 80919
Phone (719)590-9758
Fax (719)590-7526
Clay_Johnson@ampex.com

## Abstract

The DST™ 800[1] automated library is a high performance, automated tape storage system, developed by AMPEX, providing mass storage to host systems.

Physical Volume Manager (PVM) is a volume server which supports either a DST 800, DST 600 stand alone tape drive or combination of DST 800 and DST 600 subsystems. The objective of the Physical Volume Manager is to provide the foundation support to allow automated and operator assisted access to the DST cartridges with continuous operation. A second objective is to create a data base about the media, its location and its usage so that the quality and utilization of the media on which specific data is recorded and the performance of the storage system may be managed.

The DST Tape Drive architecture and media provides several unique functions that enhance the ability to achieve high media space utilization and fast access. Access times are enhanced through the implementation of multiple areas (called System Zones) on the media where the media may be unloaded. This reduces positioning time in loading and unloading the cartridge. Access times are also reduced through high speed positioning in excess of 800 megabytes per second.

A DST cartridge can be partitioned into fixed size units which can be reclaimed for rewriting without invalidating other recorded data on the tape cartridge. Most tape management systems achieve space reclamation by deleting an entire tape volume, then allowing users to request a "scratch tape" or "non-specific" volume when they wish to record data to tape. Physical cartridge sizes of 25, 75, or 165 gigabytes will make this existing process inefficient or unusable. The DST cartridge partitioning capability provides an efficient mechanism for addressing the tape space utilization problem.

## Overview

This paper will provide an architectural overview of the DD-2 (Ampex DST™) magnetic tape storage subsystem. The areas discussed within this paper will cover the functionality of the tape formatting and tape drive design, automated library, the device specific software, and physical volume management software. The DST library subsystem will then be discussed in the context of the National Storage Laboratory environment.

The design of DD-2 transport is a novel approach in helical recording. The unit utilizes proven transport technology developed for the video industry as it's core technology. The remainder of the system is designed specifically for the data processing industry.

[1]DST is a Trademark of Ampex Systems Corporation

## Ampex DD-2 Core Technology

The Ampex DD-2 design meets the stringent requirements for long media life in its approach to tape handling. The design of the Ampex transport was concurrent with the SMPTE/EBU standards development for D2 Composite Video. This allowed the adoption of several cartridge features to match specific transport requirements. One example is the size and shape of the cavity for the threading apparatus, which accommodates the large direct-drive capstan hub, in addition to four large diameter threading posts.
The unit uses this large direct-coupled capstan hub similar to high performance reel-to-reel tape drives instead of the usual pinch-roller design found in most helical transports. The advantages include fast accelerations and direction reversal without tape damage, plus elimination of the scuffing and stretching problems of pinch roller systems. Since a direct drive capstan must couple to the backside of the tape, it must be introduced inside the loop extracted from the cartridge. In order to avoid a "pop up" or moving capstan and the problems of precise registration, the capstan was placed under the cartridge elevator, so that it is introduced into the threading cavity as the cartridge is lowered onto the turntables.

In order to prevent tension buildup and potential tape damage, none of the tape guides within the transport are conventional fixed posts. Air film lubricated guides are used throughout with one exception which is a precision rotating guide that is in contact with the backside of the tape.

All motors are equipped with tachometers to provide speed, direction, or position information to the servo system, including the gear motors which power the cartridge elevator and the threading apparatus. End position sensors are used only at beginning-of-tape and end-of-tape while formatting the tape. In other situations, the servo learns the limit positions of the mechanisms and subsequently applies acceleration profiles to drive them rapidly and without crash stops. This approach also permits the machine to recover from an interruption during any phase of operation without damage to the machine or the tape.

The tape transport also features a functional intermediate tape path that allows high speed searches and reading or writing of the longitudinal tracks without the tape being in contact with helical scan drum. If tape is already threaded, high speed operations are also performed without having to unthread tape from the scanner. Thread/unthread operations are not performed over user data except when the user write performance over-rides error rate performance requirements.

The DD-2 Tape Drive architecture and media provides several unique functions that enhance the ability to achieve high media space utilization and fast access. These core technology designs allow a set of advantages unique to DD-2 tape transports:

### Multiple Unload Positions

Access times are improved, as compared to traditional 3480 type cartridges, through the implementation of multiple areas (called System Zones) on the media where the media may be loaded and unloaded. Full rewind is therefore unnecessary. This reduces positioning time in loading and unloading the cartridge, eliminates mechanical actions of threading and unthreading/loading and unloading over recorded data and eliminates the wear that is always inherent in any design that requires a return to beginning of tape.

### Head Life

Using DST media, helical heads are warranted for 500 hours, however, experience with helical head contact time can exceed 2000 hours. Because of the system zones and the ability to move between system zones without tape loaded to the helical scanner drum, the actual head life with tape mounted on the drive may be considerable longer.

### Head Replacement

DD-2 head replacement in the field requires about 0.5 hours for fault isolation, replacement, alignment checks and repair-verification of a single head. The time required

to do the same actions on the complete head set is one hour. Some competitive helical scan implementations do not provide for head replacement in the field.

## Safe Time on Stopped Tape

Whenever the flow of data to or from the tape drive is interrupted, after a pre-determined period of time, the media is moved to a system zone and unloaded from the helical drum. When data is being written, this should be a rare occurrence because of the minimum 64 (maximum 128) million bytes of buffering per drive. When in retrieval mode, returning to a system zone whenever the access queue is zero should be standard practice. In this way, if the drive is needed for a different cartridge, it is available sooner and if another access is directed at the same cartridge, the average access time is not adversely impacted by positioning the tape to the nearest system zone. Half the time it will be closer to the next access and half the time it will be farther away. With this type of drive management, the cartridge may remain mounted indefinitely without exposure to the tape or head wear.

The SCSI-2 implementation offers an interface programmable option, providing the ability to override the use of system zone for thread/unthread operations. This feature is made available for use in selected sequential access applications where optimum time-line performance is more important than achieving the best overall media life or system error rate performance.

## Media Usage Life

One of the major applications for DD-2 technology is its use as a storage level in a hierarchy of storage devices, a Disk/Tape hierarchy for example. As such, the number of tape load and unload cycles, thread/unthread cycles and searches may be significant. The expected usage capabilities for Ampex DD-2 media should exceed 50,000 load/unload cycles. An even larger number of tape thread/unthread cycles spread across system zones (assuming at least three system zones on a cartridge) can be expected, and up to 5,000 shuttle forward and rewind cycles. The number of end-to-end reads using incremental motion (less than 15 MB/sec) should exceed 2,000 while the number of reads of 1 Giga Byte files using incremental motion should exceed 5,000.

An operating environment of 20 ± 2° C with relative humidity of 50 ± 2% will provide best overall results.

## Environmental Archival Stability

Assuming cartridges are always stored within the operating environment recommended above, 20 ± 2° C with relative humidity of 50 ± 2% non-condensing, computer room storage of over 10 years is expected. For even longer archival stability, an environment maintained at 10° C or lower and a relative humidity of 40 % non-condensing or lower should result in archival stability exceeding 15 years.

## High Speed Search

DD-2 data formats include full function use of the longitudinal tracks that can be read in ether the forward or reverse direction. One of these tracks contains the geometric address of each physical block of data. This track can be searched at speeds of greater than 300 inches per second, equivalent to searching user data at more than 800 megabytes per second. Another longitudinal track is automatically recorded as user data is written to tape and provides the user with the ability to address either data block or byte offset within a user file. No user action is required to cause these tracks to be written and they provide high speed search to any point in the recorded data, not just points explicitly recorded at the time of data creation.

## Data Processing Design

The DD-2 design is the only high performance cartridge based helical data storage peripheral based upon modern digital platforms (D1, D2, D3) that is commercially available today. As such, DD-2 based products will set new performance and functionality benchmarks for the data storage industry.

A DD-2 cartridge can be partitioned into fixed size units which can be reclaimed for rewriting without invalidating other recorded data on the tape cartridge. Most tape management systems achieve space reclamation by deleting an entire tape volume, then allowing users to request a "scratch tape" or "non-specific" volume when they wish to record data to tape. Physical cartridge sizes common in helical recording devices will make this existing process inefficient or unusable. The DD-2 cartridge partitioning capability provides an efficient mechanism for addressing the tape space utilization problem.

DD-2 formatting provides for three levels of Reed-Solomon error correction. In addition, data is shuffled across the 32 tracks that make up a physical block, and interleaved within the physical track so that each byte of a block has maximum separation from every other byte that make up an error correction code word. Data is then recorded using an Ampex patented process called Miller Squared. This process is self clocking, DC free rate 1/2 coding process that approaches 100% probability of flagging a burst error. This has the effect of doubling the efficiency of a Reed-Solomon code by knowing where the power of the code should be applied.

The following table summarizes the error management system.

| FORMAT ITEM | FORMAT DESCRIPTION |
|---|---|
| BYTES PER TRACK | 48,972 |
| USER BYTES PER TRACK | 37,495 |
| C1 DIMENSIONS | RS (228,220,8) T=4 |
| C2 DIMENSIONS | RS (106,96,10) T=5 |
| C3 DIMENSIONS | RS (96,86,10) T=5 |
| CHANNEL CODE | MILLER-SQUARED (RATE 1/2) |
| C1-C2 PRODUCT CODE ARRAY | IN-TRACK BLOCK INTERLEAVE WITH DIMENSIONS 456 X 106 (TWO C1 WORDS BY ONE C2 WORD) |
| C3 CODE CROSS-TRACK INTERLEAVE DESCRIPTION | C3 CODE WORDS INTERLEAVED ACROSS A 32-TRACK PHYSICAL BLOCK |
| OUTER CRC ERROR DETECTION OF C1-C2-C3 FAILURE | FOUR 64 PARITY BIT CRC CODE WORDS INTERLACED OVER 32 TRACKS WHICH PROVIDE UNDETECTED ERROR PROBABILITY OF $10^{-20}$ |
| WRITE RETRY | YES |
| CODING OVERHEAD | 28% |

The complete redesign of the D-2 video recorder into the second generation data recording technology known as DD-2 provides the following characteristics:

## Error Rate

Using DST cartridges, the number of bytes read per permanent read error should exceed $10^{14}$, (achieved when factoring in the effect of the interleave, write retry, write bias, and the use of system zones for all thread/unthread and cartridge load/unload operations). C3 error correction is disabled during read back check when writing in order to bias the write process. Any time C2 is unable to correct the error of any one byte, a re-try is invoked.

## Error Correction Code

Reed-Solomon C1 is (228,220,8), C2 is (106,96,10), C3 is (96,86,10). All three codes are always applied, therefore, data rate, capacity, search speeds, and maximum data reliability are all achieved concurrently.

## Technology

Using the core D-2 technology, a new second generation product called Data D-2 (DD-2), was created.

## Data Rate
Interface rate of 20 MB/sec, 15 MB/sec device sustained, even with full error protection applied.

## Head Azimuth
+14.97 degrees, -15.03 degrees for adjacent tracks provides sufficient suppression of head cross talk without undue loss of signal strength.

## Data Modulation
Ampex Patented Miller Squared Code, (Self clocking, DC Free, Rate 1/2 Code).

## Tape Tension
An appropriate tape tension (3+ oz.) is maintained by the tension arm and the servo system to insure stager free wraps and precise tracking registration.

## Physical Format
DD-2 uses a 19 mm wide tape media. The track angle is 6.13 degrees yielding a track length of 150.8 mm. Head rotation speed is 100 RPS, four read and four write heads are mounted on the scanner, 37,495 user data bytes are recorded on each track yielding a sustained data rate of 14.998 MB/sec. Track pitch is 0.0395 mm which converts to about 69 six inch tracks per linear inch. This dense track format and rotation speed is what achieves the significant volumetric efficiency of the DD-2 design and it's superior data rate.

## Data Capacity
Data capacity of the DD-2 media are 25, 75 and 165 Giga Bytes for the Small, Medium and Large size cartridges.

## Data Compression
Current implementation does not provide data compression. Current system implementations tend to provide data compaction and data compression at the source of the data to gain the benefits of improved capacity and data rate throughout the system. The DD-2 format standards proposal does include provision for data compaction and compression, should it become a requirement.

## Drive Configuration
DD-2 is configured with a one-to-one, transport to control unit relationship. Helical tape devices with very high data rates are streaming devices which achieve best results with large data transfers. DD-2 control units have at least 64 Mega Bytes of high speed memory buffering for each tape drive, ensuring high probability of efficient usage of the device and the channels to which they are attached.

## Interface
Current interface capabilities include IPI-3 and SCSI-II available from Ampex, HIPPI (available from Maximum Strategy), ESCON, FDDI and others via an IBM RS/6000 attachment and others such as FCS and ATM fabrics as they become available.

## DST 800™ ACL
DD-2 can be acquired with a varied array of automation devices. Ampex provides a 6.4 Terabyte Library (DST 800) and two other offerings are provided by other vendors, the first configurable from 3.5 to 25 Terabytes and the second configurable from 10 to 10,000 Terabytes. The DST 800 is a high performance automated cartridge library storage system utilizing an Ampex designed cartridge handling system in conjunction with a special design of Ampex's 19 mm DST helical scan tape drives. The DST 800 is designed to provide fast access to 6.4 Terabytes of on-line data within a footprint of 21 square feet.

The DST 800 utilizes the 25 gigabyte cartridge which has been specifically designed for automation. The cartridge is equipped with an indentation the cartridge picker can access to withdraw the medium from the storage bin. This eliminates the need for the complexities of a

gripping mechanism and all the moving parts associated with gripping the cartridge. This design facilitates the high performance of the operation. The cartridge actually reaches speeds of 50 inches per second while being withdrawn from it's storage bin. The cartridge is drawn into a holder that is then accelerated to speeds of 90 inches per second moving between storage bins and the drive.

The DST 800 comes in two versions, Version 1 with physical volume manager (Figure 1) or Version 2 without physical volume manager (Figure 2).

## Version 1:

The interface to the ACL is through an Application Programming Interface (API) Figure 1. This interface supports a set of APIs in several categories including administrative, error handling, general information and operation, media management, mount and dismount, resource reservation and operator interface. Over this interface, a tape is identified by a cartridge ID or a partition ID. The cartridge ID is the ID for the cartridge. The partition ID is either the entire cartridge or the ID of a partition on the cartridge depending the cartridge format. A cartridge can be formatted either as a one partition cartridge or with multiple partitions on the same cartridge. Each partition acting as a logically contiguous independent storage space.

The DST 800 system delivered and installed at the National Storage Laboratory is as described in version 1 above. The File Management Software is the NSL version of UNITREE. The ACL system attached to an IBM RS/6000 which is in tern connected to a HIPPI network.

## Version 2:

The physical interface to the ACL is Ethernet. The logical interfaces uses the open system interconnection (OSI) model as is shown in Figure 2. RPC over TCP/IP is used on the session layer. The version of RPC used is SUN RPC version 2. On the Application Layer, SCSI-2 Cartridge Changer commands are used. This interface is described in Ampex document PD-19920918-1, NetSCSI specification. Over this interface a cartridge is accessed by a SCSI-2 Move Cartridge command specifying a source and destination address. The cartridge is identified by its location.

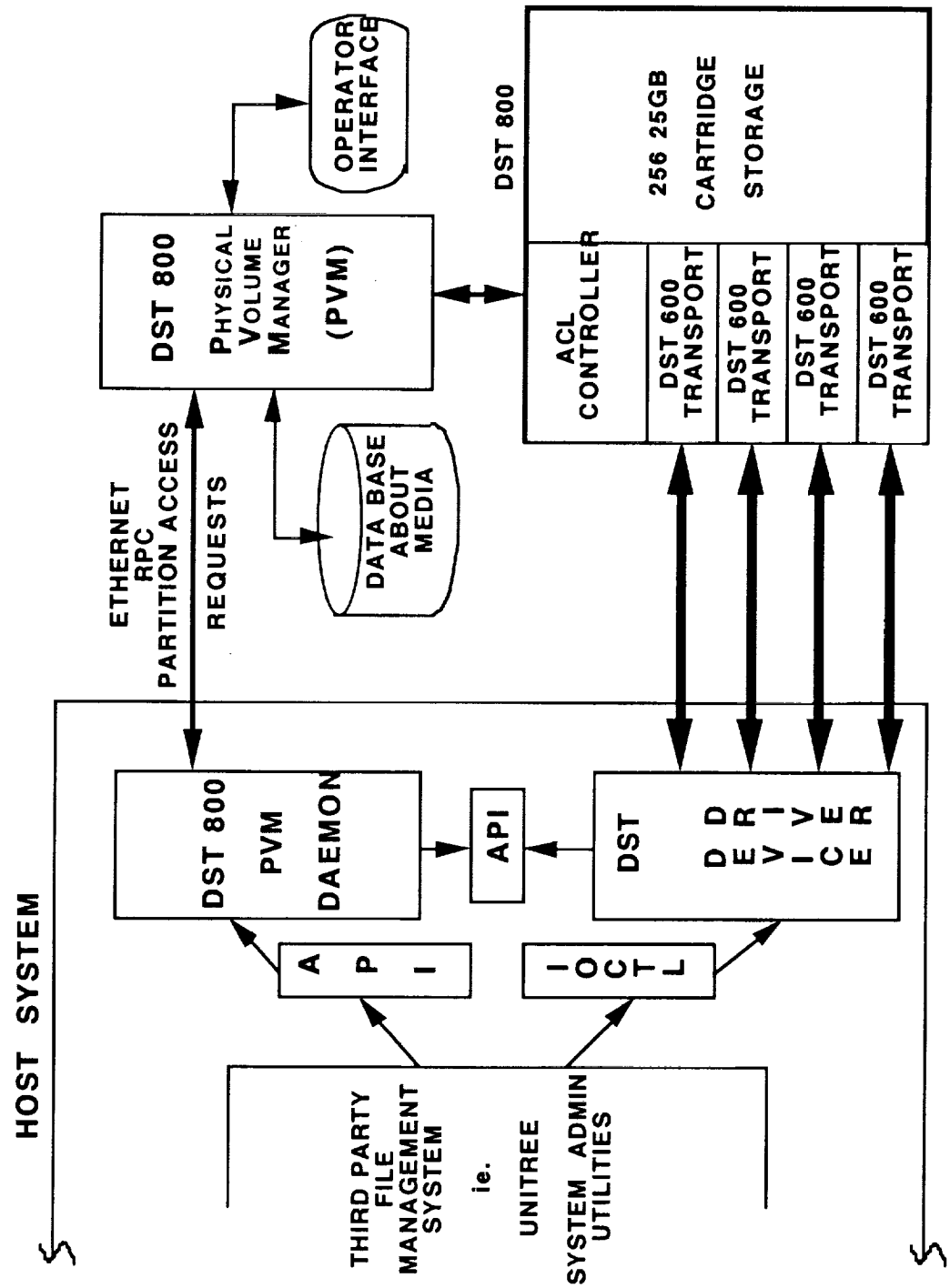



Figure 1. DST 800 With Tape Management Software

Whenever a cartridge that is not known to the DST 800 database is entered into the system for information exchange, the following procedure is performed.

1. The information contained in the cartridge manifest describing the attributes of the cartridge is entered in the DST 800 database.
2. A DST 800 barcode label is placed on the cartridge.
3. An operator command is issued to enter the cartridge into the DST 800.
4. The operator places the cartridge in an Import/Export slot.

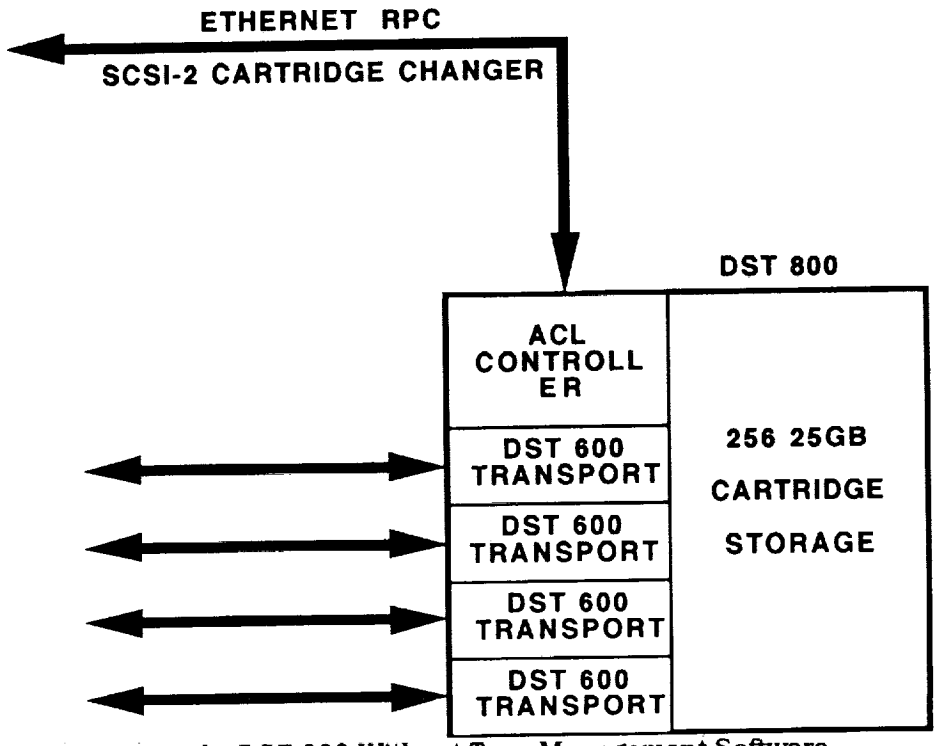



Figure 2. DST 800 Without Tape Management Software

The ACL automatically notifies the Physical Volume Manager (PVM) whenever a cartridge is introduced to one of its 8 Import/Export slots. The PVM automatically commands the ACL to read the barcode after receiving this notification. However, there is no requirement for the PVM to prioritize reading the barcode of a newly introduced cartridge over other operations. Barcodes are not automatically read when cartridges are loaded into tape drives. The ACL and PVM both maintain state information which includes a map of all cartridge locations (storage slots, import/export slots, tape drives) and, for all occupied locations, the barcode values (6 numeric characters). The barcode is an interleaved two out of five code format (ANSI standard MH 10.8-1983) and is a high density, self checking code with parity bits and accommodates bi-directional reading. The data is a six digit number with low optical density for reliable reading. The narrow bars are 30 mils and the wide bars are 75 mils.

Each cartridge is identified by a human readable cartridge ID on one end of the cartridge and a barcode on the other end. The IDs are placed on the short sides of the cartridge. The barcode is read horizontally from left to right and right to left. The DST library drive is designed to allow loading from the front of the ACL by the operator and at the back of the drive by the cartridge handling system. In the event of cartridge handling system failure, the PVM will communicate with the operator via a graphical interface on the front of the library and the operator can retrieve cartridges from the storage bins and mount them in the library drives. When automatic operation is restored, the ACL system will read the barcode on each cartridge to re-establish the integrity of the cartridge positions. This action takes just over 32 seconds.

All power supplies are equipped with over temperature sensors, that turn a particular supply off if an over temperature exists within that supply. In addition cooling fans are equipped with motion sensors that are monitored by the system. If a fan quits working, the system will shut down the subsystem affected. The environmental specifications for the ACL are the same as specified under Media Usage Life for operational environments.

Eight Import/Export slots can be used to input cartridges into the system or export cartridges out of the system. The DST 800 doors are only opened for repair or to remove cartridges for

manually loading of tape drives when the cartridge accessor is down for repair. The cartridge accessor always returns the cartridge to its original position.

The maximum number of drives that have been coupled with the ACL is four. In the television marketplace over 100 ACLs operate in the maximum configuration. The original design goal for the cartridge accessor was to support 4 transports, continuously playing back to back 7 second commercial spot segments. Today's typical cartridge accessor cycle times are summarized below for back to back operation.

| | Min. | Ave | Max. | Description |
|---|---|---|---|---|
| Move | 0.5 | 0.5 | .05 | Go to a drive |
| Picker (in/out) | 1 | 1 | 1 | Get a cartridge |
| Move | 0.5 | 0.7 | .09 | Go to a bin |
| Picker (in/out) | 1 | 1 | 1 | Put cartridge |
| Move | 0.5 | 0.7 | .09 | To another bin |
| Picker (in/out) | 1 | 1 | 1 | Get cartridge |
| Move | 0.5 | 0.7 | .09 | Go to a drive |
| Picker (in/out) | 1 | 1 | 1 | Put cartridge |
| | 6 sec | 6.6 sec | 7.2 sec | |

# Virtual File System For PSDS

**Tyson D. Runnels**

The Boeing Company
P.O. Box 24346 MS 7F-73
Seattle, WA 98124-0346
Office: 206-865-4128
Fax: 206-865-2982
runnels@zuben.ca.boeing.com

## Abstract

This is a case study. It deals with the use of a "virtual file system" (VFS) for Boeing's UNIX-based Product Standards Data System (PSDS). One of the objectives of PSDS is to store digital standards documents. The file-storage requirements are that the files must be rapidly accessible, stored for long periods of time - as though they were paper, protected from disaster, and accumulate to about 80 billion characters (80 gigabytes). This volume of data will be approached in the first two years of the project's operation. The approach chosen is to install an hierarchical file migration system using optical disk cartridges. Files are migrated from high-performance media to lower performance optical media based on a least-frequently-used algorithm. The optical media are less expensive per-character-stored and are removable. Vital statistics about the removable optical disk cartridges are maintained in a database. The assembly of hardware and software acts as a single virtual file system transparent to the PSDS user. The files are copied to "backup-and-recover" media whose vital statistical are also stored in the database. Seventeen months into operation, PSDS is storing 49 gigabytes. A number of operational and performance problems were overcome. Costs are under control. New and/or alternative uses for the VFS are being considered.

## Introduction

The conceptual architecture of the Product Standards Data System (PSDS) includes large-scale file storage. The plan calls for storing 80 billion characters representing the digitization of the Boeing Company's standards documents. These documents must remain rapidly available with all revisions for the lifetime of any product built using the standards. The current documents must remain immediately available for reference and revision.

Project requirements include that the system be deployed on UNIX-based computers. The preferred UNIX-based systems had, at design time, upper limits of file storage that were significantly lower than the projected maximum. Additionally, the file-management software stored files in one single directory unless manually overridden. This limitation posed problems for fixed-capacity disk drives. Given the above requirements, a solution was sought that provided large-scale storage capacity, archival storage, disaster recovery, and flexible disk-space management. This solution is called the Virtual File System for PSDS.

## Project

The project, in more detail, includes a number of components. They are illustrated in Figure 1. An acronym list is provided to decipher them. Authors, using the Authoring Workstations, create or modify the digital standards documents. The documents are stored on the Standards Authority Database platform. Each digital document consists of multiple files that, together, may be displayed or reproduced on paper as a formal corporate standard. Subsets of files are routinely downloaded to subordinate platforms. One set is named Master Local Authoritative Databases. The other is named Local Authoritative Databases. Customers of PSDS retrieve and display the standards using Retrieval Workstations. A set of platforms named Derived

Authoritative Databases store subsets of the information in a form retrievable by computer applications other than the PSDS retrieval subsystems. These other applications may, in turn, support their own form of retrieval that may or may not be strictly a reproduction of the printable standard. An example is an expert system that, when posed a question by a design engineer, reasons about the information stored in an aggregate of standards - including those stored in PSDS.

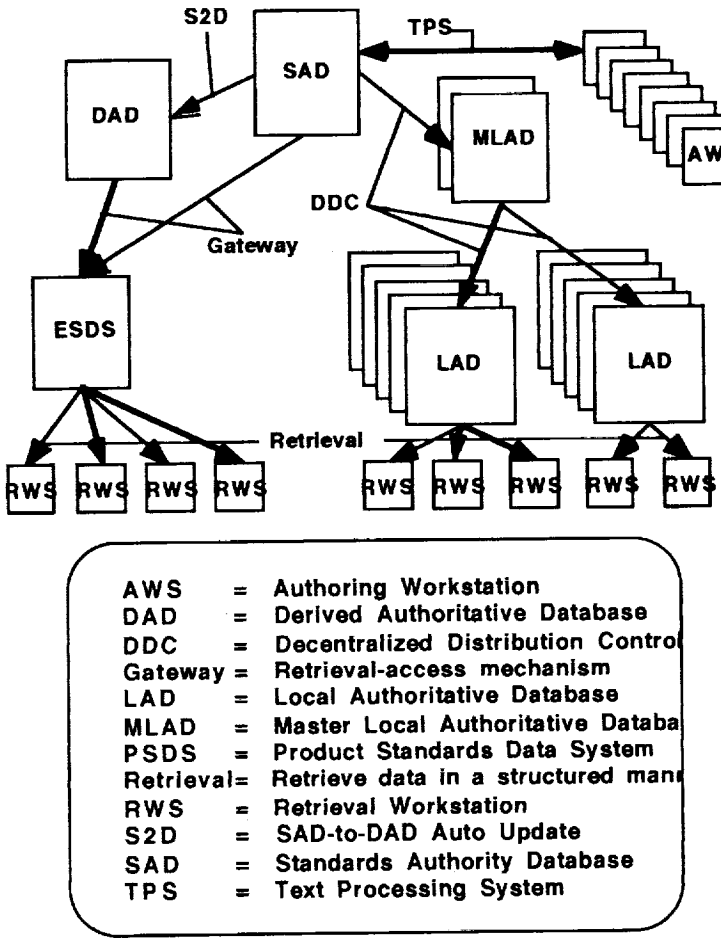



Figure 1. Global Architecture of PSDS

## Objectives

The Virtual File System is a component underlying the Standards Authoritative Database platform. The objectives of the VFS include:

- Store tens of gigabytes of information (80 gigabyte projection)
- Store the information as a database and as flat files
- Support a file manager that clusters the flat files densely in a small number of directories
- Support a commercially-available database management system
- Provide "immediate" access to the information
- Behave as a permanent archive
- Secure the information through disaster-recover processes
- Be cost effective

## Alternatives

The alternatives analysis was an exercise in matching cost, performance, and functionality with the objectives. Preceding decisions about architecture also constrained the choice of alternatives. A major concern for PSDS is to use technology that is available at the time of need. Although the time of need was 4 months in the future, the procurement activity alone - in a large corporation - would use 3 of them. Thus, the first decision was to use "off-the-shelf" technology.

The main UNIX server was limited, at the time, to 32 disks of no more than 1 gigabyte each. Projected storage volumes exceeded this value. Backup required 1 hour per gigabyte. A weekend would not provide enough time for a backup. UNIX-based backups also required that the applications be shutdown during backup. Backups would, therefore, exceed the shutdown time available. Additionally, the high-performance and high-capacity disk drives available for the server were relatively expensive and ill suited for archival storage.

Pure backup-and-recover software and specialized hardware did not meet the objectives either. They did not provide the required online capability. And, pure large-scale data storage products did not provide the embedded backup-and-recover functionality. The Virtual File System approach was chosen. A vendor's product met all of the objectives.

## Solution

Figure 2 illustrates the interrelationships among the PSDS file-management components.

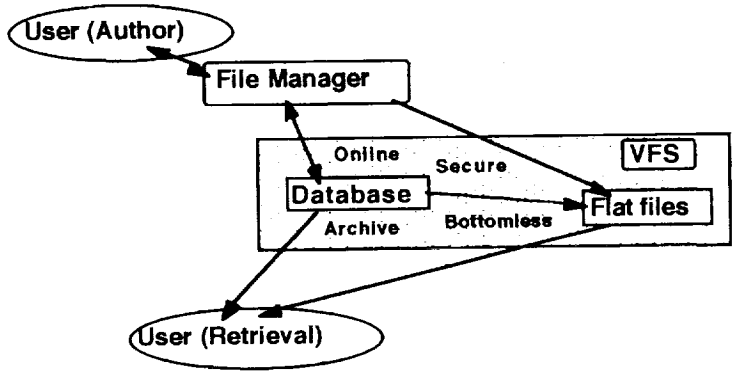



Figure 2. Interrelationship of PSDS File-Management Components

The vendor supplied the following VFS functionality:

- Hierarchical file management: The automatic migration of files from one storage media to another using a least-frequently-used algorithm.
- Embedded backup and recovery: Backups were designed to take maximum advantage of optical hardware to reduce the time necessary to perform a backup. Backup could also run while other applications were running. Recovery was optimized for disaster recovery in such a way as to reduce downtime by a factor of 10 over standard UNIX utilities.
- Lower cost-per-byte: Files are migrated to optical disk. Given careful planning, the cost per byte for data storage on the optical cartridges is less than that for the central UNIX server's spinning magnetic disks.
- Online visibility: Regardless of whether the data are on magnetic or optical, the access is transparent to the applications.
- Disk partition limits are relaxed: Disk partitions mounted on the VFS have data-storage limits extended by a least a factor of 40 over those on the central UNIX server.

Limitations and compromises are still required. First, the VFS does not handle database management systems (DBMS) that manage their own files using the disk as a "raw device" - that is, without using the UNIX file system. The PSDS database is such a DBMS. The UNIX server's disks are used by the DBMS. The PSDS file manager uses the DBMS to store the location - UNIX path name - of all of its files. These files are stored on the VFS. Second, the VFS is a separate device with its own operating system. It must be managed separately and independently yet in coordination with the central UNIX server.

## Performance

Performance was estimated during the design phase to be acceptable for a network-based system such as PSDS. Actual performance was at first not as good as the estimate.

Backup and recovery are particularly slow. That is, they cannot perform their work in the time estimated to be required - or in the time available. Their performance is a function of the UNIX file structure imposed by the PSDS file manager. The file structure also slows performance of NFS and a host of other UNIX utilities and PSDS modules. The PSDS file manager's design tends strongly toward placing all files into a single directory. The UNIX file system is optimized for a tree-like structure of directories, sub-directories, and files. The VFS is optimized in the same way. Performance drops off geometrically with the number of files in a single directory. See Figure 3, next page.

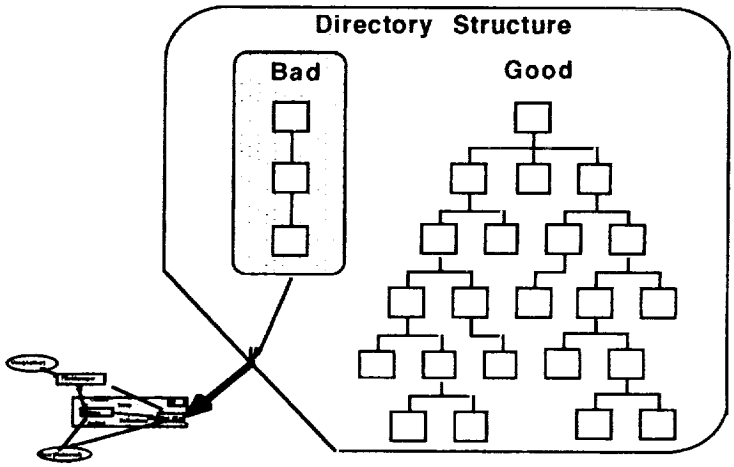



Figure 3. Major Source of Performance Problems

The Network File System (NFS) performance was also slower than expected. The NFS performance is partly a function of the speed of the central processing unit (CPU). The VFS is not a fast CPU in comparison with the main UNIX server and the PSDS load is large. This problem is overcome in the current system by using the central UNIX server's disks as a work area for the most time-critical files. This means unanticipated system management.

## Futures

The VFS is being used as a bottomless archive with backup and recovery for the SAD. At least three other distinct possibilities exist for use within the capabilities supplied with the VFS. See Figure 4.

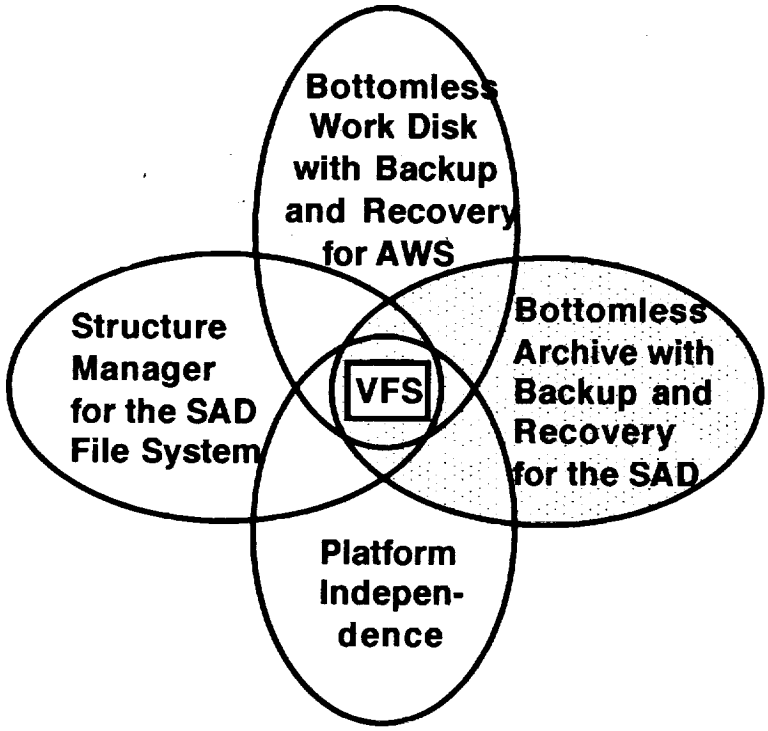



Figure 4. Potential Future Applications for a VFS

The VFS capability may be extended to components of PSDS other than the SAD. The Authoring Workstations (AWS) receive original work from authors on a daily basis. The disks on the workstations may be made into virtual file systems just as are those on the SAD. Linking the AWS to the VFS would provide the bottomless partition feature and, most importantly, a centralized backup and recovery mechanism. A drawback to this process is that both the local-area and wide-area networks will receive more traffic.

Another feature of the VFS is to manage the distribution and proliferation of sub-directories in a way transparent to the PSDS file manager. Thus the "bad" directory structure can be made into the "good" directory structure independently from the requirements of the application (the PSDS file manager). Tests on PSDS data show this to be a 100% improvement in performance for UNIX utilities, NFS, and backup-and-recovery. A drawback to this process is that the system administrators have an added burden of maintaining a mapping of files from the "bad" structure to the "good" structure.

The third additional way to use the VFS is to distribute it among the far-flung platforms that comprise PSDS. The original VFS acquired by PSDS was an independent "turnkey" system - hardware and software. Evolution of the VFS is moving it toward a more software-only architecture. Limitations of CPU speed, memory constraints, number of I/O busses, and sundry become less restrictive. Each local-area-network could have a VFS. System-administration tools are also expected to evolve in support of a more distributed architecture. Traffic on the wide-area-network could be reduced. Drawbacks are cost and training. The VFS is not trivial to manage. It is not trivial in cost.

## Summary

Given its requirements and constraints, PSDS picked a solution for large-scale file storage that worked. Conversely, a solution was available that satisfied the PSDS requirements and constraints. Opportunities for wider use of the VFS exist and are being considered.

Problems were encountered after installation. They included issues involving cost, performance, and reliability. The issues were attacked vigorously by PSDS and vendor staff and resolved.

Future management of PSDS data will be supported by enhancements from the VFS vendor, improvements in the PSDS software, and improvements in UNIX-based systems. It appears, though, that the volume of data will continue to exceed the currently available "simple" storage systems and that a VFS in some form will be required.

# VOLUME SERVER - A SCALABLE HIGH SPEED AND HIGH CAPACITY MAGNETIC TAPE ARCHIVE ARCHITECTURE WITH CONCURRENT MULTI-HOST ACCESS

Fred Rybczynski

Metrum,Inc.
10948A Beaver Dam Road
Hunt Valley, Maryland 21030
Voice: (410) 771-9207
FAX: (410) 771-9210
email: rybski@metrum.com

## INTRODUCTION

A major challenge facing data processing centers today is data management. This includes the storage of large volumes of data and access to it. Current media storage for large data volumes is typically off line and frequently off site in warehouses. Access to data archived in this fashion can be subject to long delays, errors in media selection and retrieval, and even loss of data through misplacement or damage to the media.

Similarly, designers responsible for architecting systems capable of continuous high-speed recording of large volumes of digital data are faced with the challenge of identifying technologies and configurations that meet their requirements. Past approaches have tended to evaluate the combination of the fastest tape recorders with the highest capacity tape media and then to compromise technology selection as a consequence of cost.

This paper discusses an architecture that addresses both of these challenges and proposes a cost-effective solution based on robots, high-speed helical scan tape drives, and large-capacity media.

## DATA CENTER PERSPECTIVE

Significant advances in magnetic tape drives, media capacities, and the integration of robotics now make it possible for most sites to maintain a significant portion, if not the entire set, of data in the computer center. Using these new technologies, the amount of floor space required for data storage is significantly reduced. (For example, the Metrum RSS-600b robot system contains 10.8 terabytes in less than 20 square feet of floor space. This is equivalent to approximately 60,000 reels of 9-track tape.) Media, housed within computer-controlled and robot-accessible carousels, is accurately identified by barcode readers integrated within the robotics. Following identification, media is rapidly retrieved and loaded by the robot. The high speed tape drives quickly locate the data and convey it to the computer host.

These new technologies enable the co-location of large volumes of data with computer hosts, thereby expediting data access and analysis. However, they are but mere tools, incapable of performing any data management function by themselves. Management of the data is performed by software that can manipulate the previously-described tools to achieve efficient data storage and retrieval. Various software data and storage management solutions are available. Some, such as UniTree™, perform a data migration function. They transfer files through a hierarchy of storage technologies measured by speed, capacity and cost (Figure 1) under the direction of a software-implemented algorithm responsible for managing the computer system's mass storage resources.

Others, such as AMASS™, perform a network-attached archival function (Figure 2). They present the entire archive storage capacity as if it were a huge disk-based file system. Data, referenced by a path-qualified file name, is transferred only in response to explicit commands received from a user or an application process. Although these archival systems are typically not delivered with software to perform behind-the-scenes file migrations, they can be used to accomplish a limited version of these functions.

It may occasionally be desirable to connect two or more tape drives, located within a single robot, to one host so that read and write operations can proceed independently, or in order to support multiple concurrent accesses. The additional tape drives could share the same interface bus or each could be connected using separate buses to maximize bandwidth. Decisions to utilize separate buses must consider the portion of time the tape drives might be involved in input and output versus the portion of time spent positioning to a new location on the tape. A tape drive places no load on the bus bandwidth during positioning functions.

## EXTENSIBILITY OF THE ARCHITECTURE

Metrum tape drive features and robot configurations are identified and described in Tables III and IV in order to enhance the relevancy of further discussion. The information in these tables establish a real-world measure of capacity and transfer rates. All referenced components are available as commercial off-the-shelf (COTS) products. Their capacity and throughput numbers do not represent theoretical possibilities, but reflect actual system performance measures.

Extensibility is illustrated in Figures 3 and 4. Figure 3 shows the simple case of one robot shared by three hosts. Figure 4 shows how multiple hosts can access data residing in multiple robots.

**Table III.** Metrum Tape Drive Features.

| | |
|---|---|
| RSP-2150 Tape Drive | 2 MB/S Sustained |
| | 4 MB/S Burst |
| | Track Addressable |
| | Record Addressable |
| | Robot-Compatible Media (S-VHS) |
| S-VHS Cartridge Media | DDC-258 (14.5 GB) |
| | DDC-343 (18 GB) |
| | $1.30 per Gigabyte |

The American National Standards Institute Small Computer System Interface specification (SCSI, ANSI X3.131 1986) indicates that a host, through a single SCSI host bus adapters, could be connected to up to seven tape drives on a single SCSI bus. If each one of these tape drives were located in a different RSS-600b robot, the computer host would have immediate and unattended access to more than 75,000 gigabytes (ie, 7 tape drives * 10,800 GB/robot) of data storage capacity within a total floor space of less than 140 square feet. This is equivalent to approximately 420,000 reels of nine-track tape occupying more than 11,000 square feet of floor space. In essence then, each SCSI host bus adapter represents the potential for up to 75,000 gigabytes of robot-accessible data storage. At the same time, the seven RSP-2150 tape drives represent a total I/O bandwidth of 14 MB/S for that host.

Data archive capacity can easily be increased if the host is able to accommodate additional SCSI host bus adapters. Three host bus adapters represent the potential for 225,000 gigabytes of data storage capacity in 420 square feet of floor space. The equivalent volume of data on nine-track tape would require 1,260,000 reels of tape (at 6,250 bpi, 2,400 feet long, and 180 megabytes per reel) and more than 30,000 square feet of floor space. The entire 225,000 gigabytes of storage capacity can be shared with six additional hosts using the same principals as shown in Figure 3 and Figure 4.

[NOTE: The ANSI SCSI specification x3.131 1986 gives specific cable length limitations. These can be extended through the use of SCSI bus repeaters and/or fiber optic bus extenders.]

**Table IV.** Metrum Robot Configurations.

| | RSS-48b | RSS-600b |
|---|---|---|
| Tape Drives | 2 | 6 |
| Cartridges | 48 | 600 |
| Capacity (18 GB/Cartridge) | 864 GB | 10,800 GB |
| Robot Cost Per Megabyte | $ 0.08 | $ 0 02 |
| Control Interface | RS-232 | RS-232 |
| Floor Space Required | 6 $Ft^2$ | 19 $Ft^2$ |
| Equivalent 9-trk Reels | 4,800 | 60,000 |
| Equivalent 9-trk Floor Space | 125 $Ft^2$ | 1,600 $Ft^2$ |

The robot command language can be extended to enable a computer host to specify a robot identifier. In this way the host is able to access cartridges in automated storage systems configured with more than one robot.

- The host could QUERY if a data set is available in the cartridge inventory of a specific robot. The response could be "AVAILABLE", "NOT AVAILABLE", or "AVAILABLE IN rr", where "rr" identifies the robot containing the specified data set.
- The host can request that a cartridge be loaded into a specific drive in a specific robot. The robot identifier would be mandatory. Absence of a robot identifier would generate an "INCOMPLETE COMMAND" response.

The database and the robot command language can be extended to support controlled access to multiple copies of a data set. For example, the results of the database query may report that the primary tape cartridge copy with the requested data set is in use, but that another copy is available. If both the primary and secondary copies reside in the same robot, the robot-controlling computer simply loads the secondary copy. However, if the secondary copy resides in a different robot, the robot-controlling computer might respond to the LOAD request with "AVAILABLE IN rr". This efficiently conveys that the load request could not be satisfied and that robot "rr" has an available copy of the data set. The requesting host can then decide if it wants to issue a revised LOAD command, probably determined by the availability of its tape drive in robot "rr".

## ADVANTAGES OF THE ARCHITECTURE

A significant advantage is the ability to scale system storage capacity in response to system needs:

- The number of tape drives within a single robot can be as few as one or as many as the robot can contain. (The Metrum RSS-600b robot can contain a maximum of six tape drives.)
- The number of robots is limited by the number of tape drive connections a host can support. It can range from as little as one robot and scale up to the maximum connections possible. Since each robot represents up to 10,800 gigabytes, a very wide range of data storage capacity is possible.

System bandwidth can be scaled in response to system needs:

- The implementation of multiple tape drives installed in one or more robots can dramatically increase the number of file transfer operations that can occur concurrently. For example, if seven tape drives were connected to a single host, up to seven I/O operations could occur concurrently in any combination of read, write and/or data search.
- The number of tape drives can be increased over time in response to rising system load in order to increase bandwidth. A host system can be configured with one Metrum RSP-2150 tape drive or up to the system-supported maximum. Table III lists the RSP-2150 tape drive features. Seven RSP-2150 tape drives can support a sustained transfer rate of 14 MB/S and burst rates of up to 28 MB/S for markedly less cost than a single 19mm tape drive. (This analogy is only valid if either the mandated data acquisition rate is in the range of the RSP-2150 or if the overall data stream can be demultiplexed and the resultant sub-streams of data have rates in the range of the RSP-2150.)
- A SCSI bus interface can transfer data faster than most network media-plus-protocol combinations. SCSI-connected tape drives with robot-assisted access to media in the storage system can be used to transfer data at sustained rates significantly faster than the network could support.

The ability to directly connect multiple hosts to the same data reservoir optimizes overall system throughput:

- 2 MB/S at each of 7 sites represents a significantly higher bandwidth than 14 MB/S at a single site if the data from the single site subsequently has to go through the restrictive bandwidth of a network in order to reach the other 6 remote sites. For example, 300 KB/S is typically the maximum sustainable transfer rate for a 10 megabit baseband Ethernet network.
- Data access times will be faster, since each tape drive can position directly to a data starting location on tape without having to first wait for data transfer processes on other tape drives to complete.

Improvements in operations:

- Automated media management means reductions in errors, loss of data, and associated recurring costs.
- Automated media management performs accurate and very fast media retrieve/store operations.
- The automated storage system can periodically analyze all media for wear without operator intervention. A computer host initiates the process and analyzes the final results. No data from the media needs to traverse the SCSI bus, therefore there is neither bus bandwidth nor host CPU impact.
- Excessive media wear can be determined before the data becomes unreadable. If a host detects media showing excessive wear, the data can be transferred to a new cartridge and the old cartridge identified for removal by an operator.

Costs are minimized:

- Less floor space means reductions in the amount of leased media storage space and related insurance and media transportation costs.
- Incrementaly augmenting the number of tape drives in response to rising system load affordably increases bandwidth. For example, seven RSP-2150 tape drives can support a sustained transfer rate of 14 MB/S and burst rates of up to 28 MB/S for markedly less cost than a single 19mm tape drive. Increasing bandwidth to 16 MB/S with the acquisition of one additional RSP-2150 is significantly less expensive than acquiring another 19 mm tape drive.
- The volume serve architecture is cost effective for a distributed computing environment because it allows sharing of the most expensive component (the robot) while still providing lights-out, operator-free support, minimizing recurring operating costs.

Limited risk:

- Configurations with multiple tape drives and robots distribute risk. Failure in a single component does not shut down the entire system.
- The Metrum components and storage media identified by way of example in this paper are based on standards.

- The Metrum RSP-2150 tape data format has been submitted to ANSI and is going through the standardization process.
- The Metrum RSP-2150 tape drive uses the standard SCSI-1 interface, supported by virtually all computers on the market through direct manufacturer support or through third-party offerings.
- Media wear and aging can by monitored dynamically with software by interrogating Metrum RSP-2150 tape drive registers.

## DISADVANTAGES OF THE ARCHITECTURE

- The desired tape cartridge could be in use by another host/drive, resulting in a delay in access of indeterminate duration. It may be possible to minimize the number and frequency of these delays by making multiple copies of data cartridges for which access conflicts occur. The database structure would then have to be extended to support the concept of multiple copies of a single data set.
- The situation may arise when the cartridge containing the data set is available but the host's tape drive in that robot is already busy servicing a data transfer request. This situation may cause an unacceptable access delay. It may be possible to resolve this by any of the methods listed below. Some of these methods may require that additional database information fields be generated before they can be implemented.
    - Additional copies of the cartridge could be placed in other robots. If one tape drive is busy, perhaps at least one of the others may not be, thereby permitting immediate access to the data.
    - An additional tape drive could be added to the robot, space permitting.
    - Implement a data storage architecture capable of passing cartridges from one robot to another automatically.

## SUMMARY

This paper has proposed hardware configurations that support the construction of large computer-accessible data archives. These configurations minimize storage costs and data access latencies while they maximize data transfer rates. Simple database constructs and a minimal robot control language have been presented. Commercial off-the-shelf hardware components were identified, by way of example, to demonstrate the feasibility and capability of this architecture.

Computer programs needed to implement this architecture, while not exceedingly complex, are not commercially available at this time. Non-commercial versions with limited functionality are currently under development.

# The Growth of the UniTree Mass Storage System at the NASA Center for Computational Sciences

**Adina Tarshish**
NASA/GSFC Code 931
Greenbelt, MD 20771
(301) 286-6592

**Ellen Salmon**
Hughes STX Corporation
NASA/GSFC Code 931
Greenbelt, MD 20771
(301) 286-7705
XREMS@CHARNEY.GSFC.NASA.GOV

## ABSTRACT

In October 1992, the NASA Center for Computational Sciences made its Convex-based UniTree system generally available to users. The ensuing months saw the growth of near-online data from nil to nearly three terabytes, a doubling of the number of CPUs on the facility's Cray Y-MP (the primary data source for UniTree), and the necessity for an aggressive regimen for repacking sparse tapes and hierarchical "vaulting" of old files to freestanding tape. Connectivity was enhanced as well with the addition of UltraNet HiPPI. This paper describes the increasing demands placed on the storage system's performance and throughput that resulted from the significant augmentation of compute-server processor power and network speed.

## I Introduction of UniTree at GSFC

The NASA Center for Computational Sciences (NCCS) is a scientific computing center serving more than 1200 users with a range of needs from supercomputing to data analysis. The UniTree file storage management system first arrived at the NCCS on July 6, 1992. As UniTree was to be the primary system for mass storage management, the Convex C220 was upgraded to a C3240 with four CPUs, 512 megabytes of memory, and 110 gigabytes of disk. Also included in this initial configuration were 2.4 terabytes of nearline robotic storage provided by two StorageTek 4400 silos. Although UniTree supported both NFS and ftp as access methods, access to UniTree was permitted only through ftp in order to meet the throughput demands of our Cray Y-MP (UniTree's primary storage client), IBM 9000 users, and workstation clients.

The mass storage contract under which Convex/UniTree was obtained required that it be able to handle 32 concurrent transfers while 132 other sessions supported users. The size of the transfers done in testing was realistically large, about 200 megabytes each. UniTree ultimately showed itself able to manage this workload, and by the third week in September it had passed acceptance.

In the following sections we describe the extensive efforts of the NCCS in supporting the initial configuration to bring UniTree to a robust production-level file storage system.

## II Getting Ultranet Access

The NCCS computing environment supports an UltraNet network configuration between seven buildings serving more than 750 scientists at the Goddard Space Flight Center. This network

provides gigabit-per-second access to the Cray Y-MP and the IBM 9000. At acceptance time, the UniTree system had not yet been compiled and tested under ConvexOS version 10.x, requiring that we backlevel the operating system down to 9.1. ConvexOS UltraNet support, however, required a minimum level of 10.0. When 10.x-compatible UniTree executables were finally available in early October, we upgraded directly to 10.1 and installed the new UniTree routines. A month later, the HiPPI UltraNet hardware interface arrived, followed in a few weeks by the beta UltraNet 4.0 software. The next few weeks were spent stress-testing the system, ultimately uncovering the same bug that had been reported by other beta-test sites, i.e. the native Ultra path (-u) could only handle a maximum of sixteen concurrent transfers, refusing to connect the seventeenth at all. A fix for this problem had been released by Ultra and had been proven to cause the Convex to crash. Crashes also occurred when too many concurrent transfers over the host stack (-uh) path were attempted. By early January of 1993 an Ultra microcode fix was finally available which managed to avoid this problem. The fix allowed up to 28 simultaneous transfers to take place, and Ultra access to UniTree was now enabled for users on the same port used for UniTree Ethernet transfers.

Within a week, we discovered that when Ethernet transfers used the same executables as UltraNet transfers (as intended by the developers), all Ethernet communication throughout the machine would go into a hang state. To overcome this problem, Ultra access via the default UniTree ftp port was removed and enabled via a different port, employing locally-written software to enforce that Ethernet transfers could not be started up on the UltraNet port. Although patches have been applied to address the original problem, the local software has remained in place pending stress-testing of the patches. UltraNet access to UniTree has since enjoyed relative stability in this configuration.

## III Usage Rises, Functionality Increases

Once stability on UltraNet was realized, overall demand for UniTree increased sharply. In February alone users added nearly as much data to the UniTree system as they had added in November, December, and January *combined* (fig. 1).

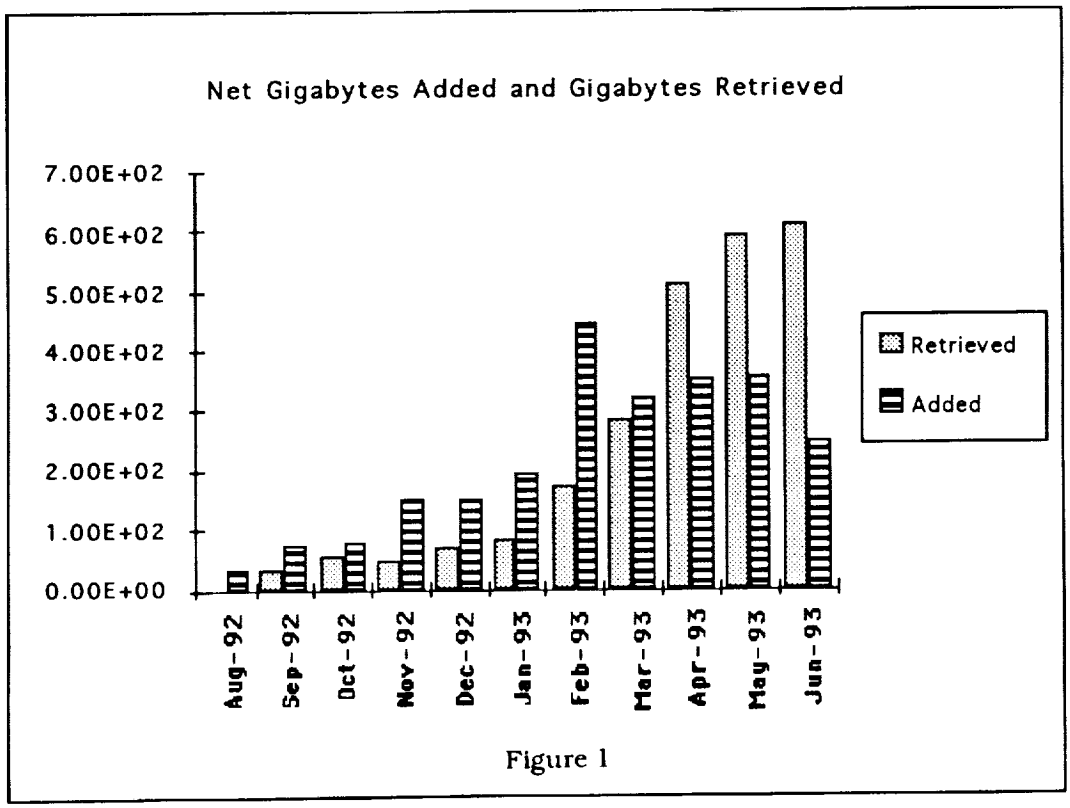



Figure 1

By the end of February, more than 7500 silo tapes out of an available total of 10,000 had been filled with UniTree data (fig. 2).

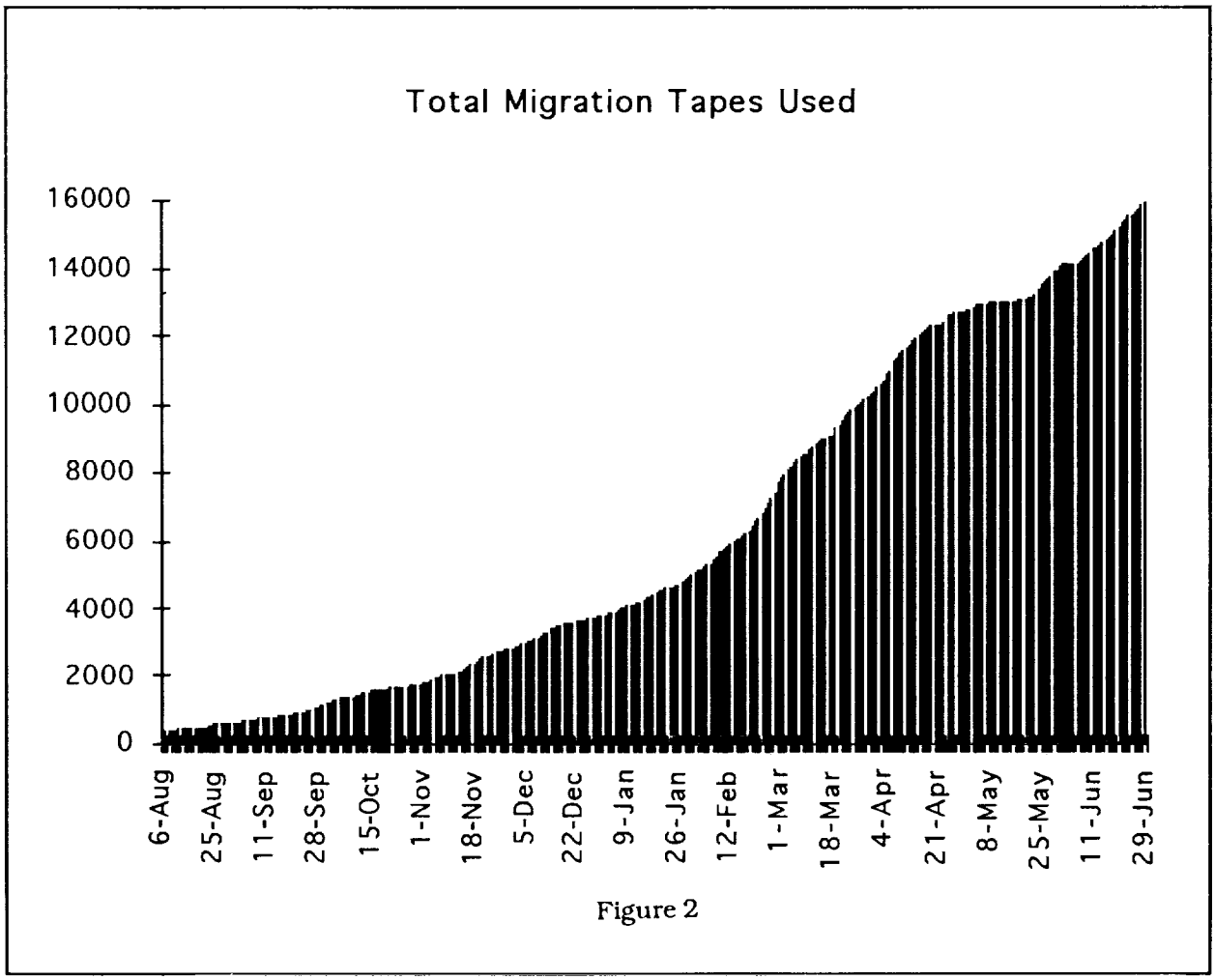



Figure 2

But UniTree's growing popularity soon placed us in a potentially disastrous situation - we were quickly running out of storage. The reason for this was that UniTree 1.5, the only production-level version of Convex/UniTree that existed at that time, did not allow for more than 10,000 tapes to be managed by the system. Not until early March was Convex able to install a modification to allow for up to 100,000 tapes, 18,000 of them for nearline storage and the rest for vaulting, or deep archive.

UniTree vaulting and repacking remained a concern. Our version of UniTree 1.5 did include executables for repacking, or removing the "holes" from tapes caused by deleted files, as well as those for vaulting, or the copying of little-used files onto free-standing tape, for deep archive, but neither of these worked properly at our site. We soon realized that the additional 8000 nearline tapes that could be accommodated by the software would not last for more than a couple of months, and that even if they lasted longer, without repacking or vaulting, we would not be solving our storage crisis but merely postponing it. Another catastrophe we were facing at that time was that both of ours silos were nearly full. Without vaulting, most of the additional 8000 tapes for nearline storage would actually have to be offline, mounted by operators. On busy days, that would amount to hundreds of tape mounts a day. We did not have the operations staff necessary for such an undertaking, nor did we want to slow UniTree down while humans located and mounted the tapes. For these reasons, we found ourselves clamoring

for repacking and vaulting executables that worked, so that we would have a measure of control over the number of free silo tapes.

By April 5 we finally had a working repacker with UniTree 1.5. We began immediately to repack in earnest, freeing hundreds of tapes for new data. By April 22 we had also succeeded in vaulting to free-standing tapes. Working with Convex, we assembled utilities that operators could invoke to place a UniTree label on new free-standing tapes, so that they could be used for vaulting. Both repacking and vaulting are now fully operational and running in a production mode.

By April 14 we had obtained a second modification to UniTree 1.5 that allowed up to 36,000 tapes to be used for nearline storage. In early May we were able to acquire a third silo for UniTree, and as of this writing, we have already used over 3500 of the tapes stored within it.

## IV Current Trends in Usage

Recently we have observed that the total number of retrieves for a given period of time outstrip the number of files stored (fig. 1). Since we have discouraged the use of UniTree as a "black hole" from the beginning, we are encouraged by this finding to believe that users are making use of the data they have stored. Accordingly, we have begun to see a considerable increase in the number of silo tape mounts, many of which are done for the purpose of retrieving data from archive (fig. 3).

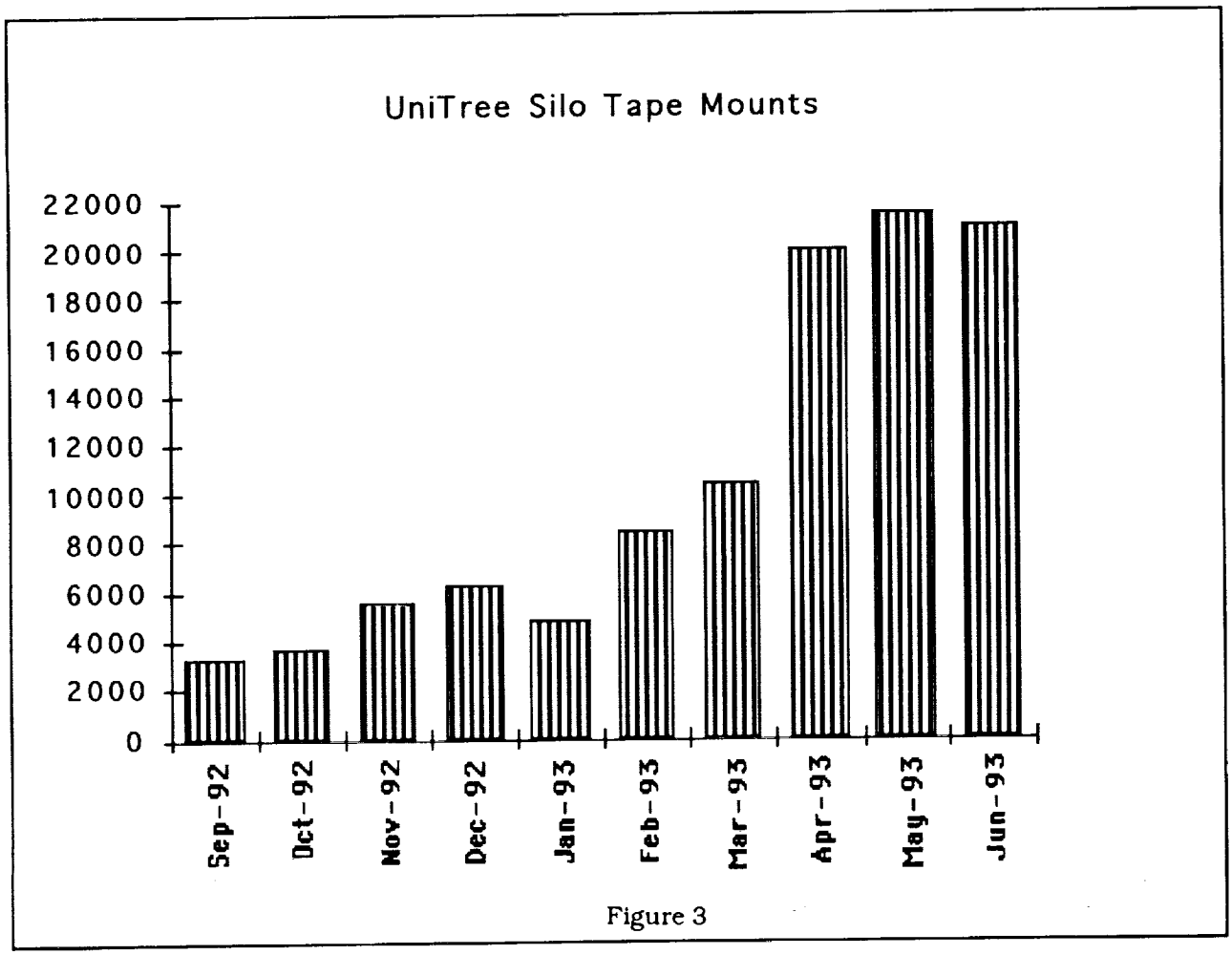



Figure 3

We are also gratified to find that users are making increasing use of the UltraNet interface where available (fig. 4), which frees the heavily-used Ethernet for telnet sessions and transfers from machines lacking UltraNet hardware.

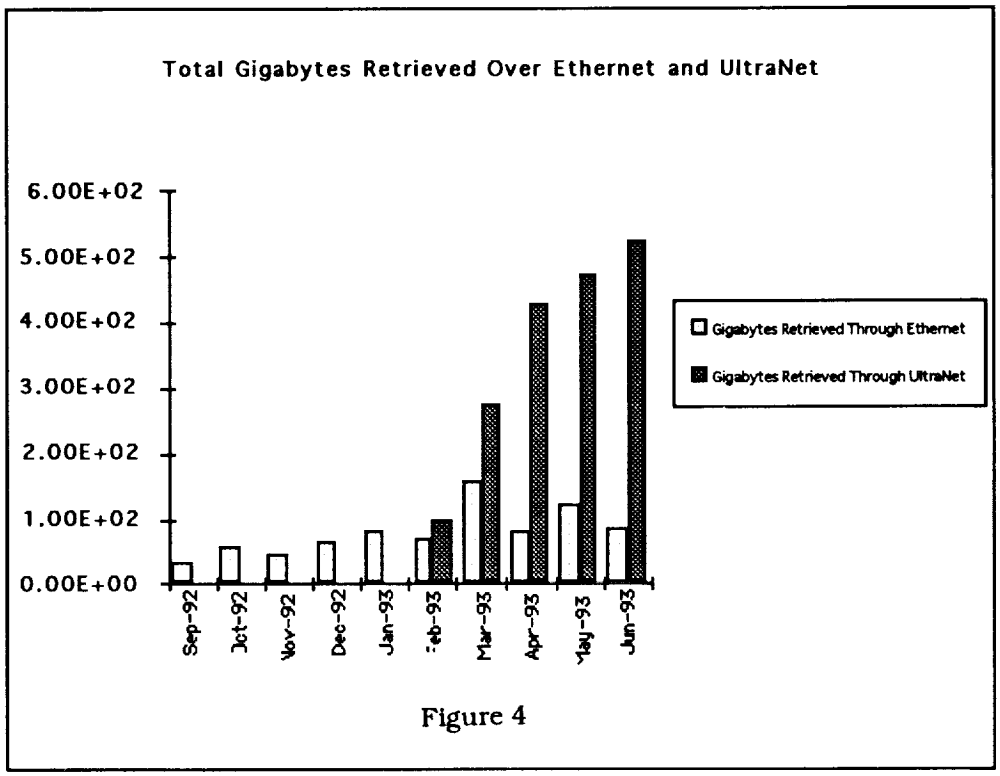



Figure 4

At this time, UniTree contains nearly two-and-a half terabytes worth of data (figure 5).

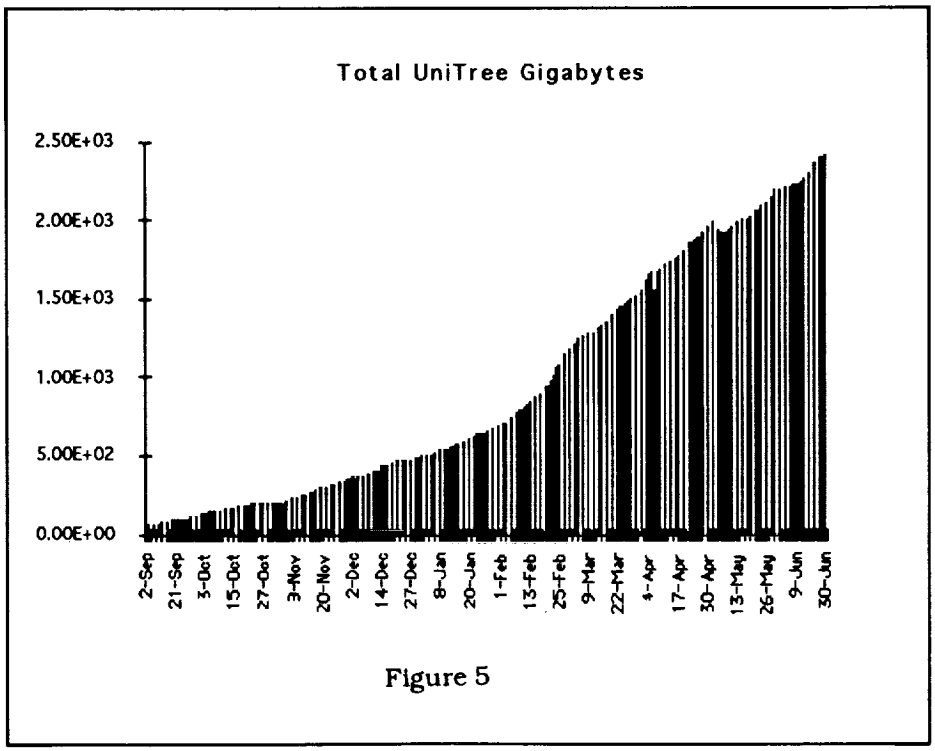



Figure 5

## IV Near-Term Challenges

Since late February, our disk cache has been 77.5 gigabytes in size. We have seen many days since then when the cache has been so busy that its contents have completely turned over within 24 hours (fig. 6).

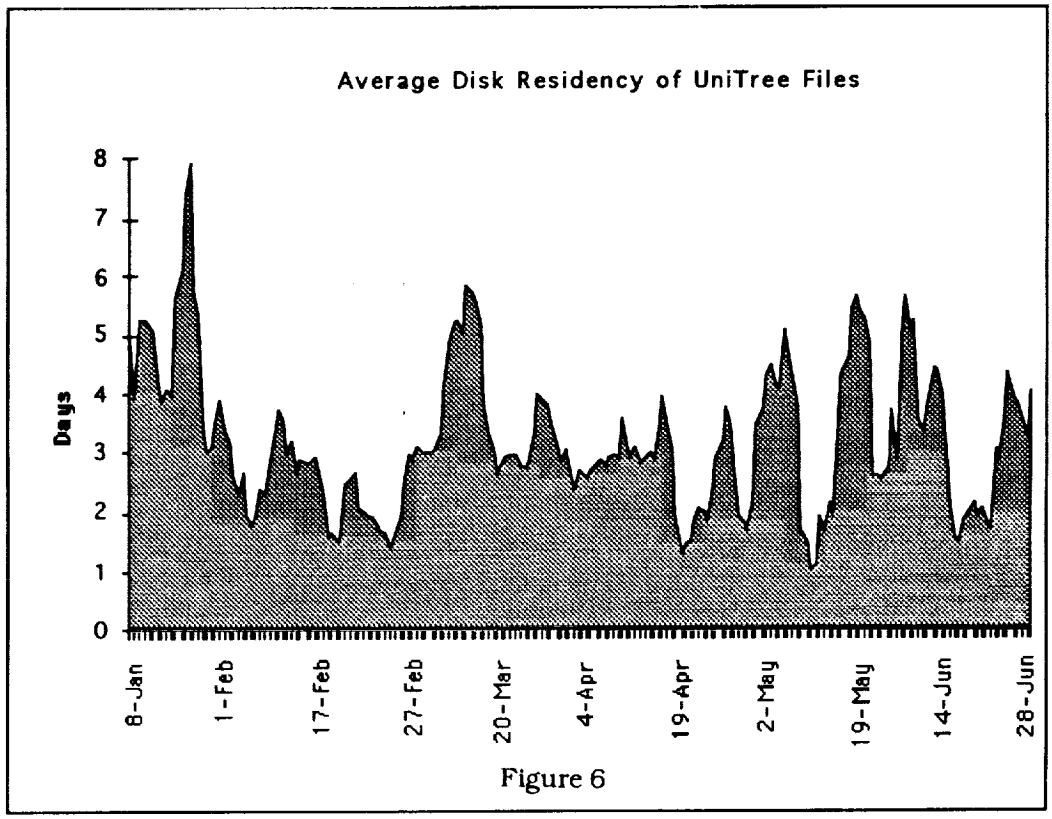



Figure 6

It is very inconvenient for users to find that their data which was stored only the day before must now be retrieved from tape, and it puts a considerably greater load on the UniTree tape processes. To address this problem we ordered an additional 40 GB of disk, bringing our total disk capacity to 150 GB (fig. 7).

However, the inherent difficulty of recovering from disk crashes under UniTree 1.5 prompted us to allocate the newly acquired disk for RAID use and for hot spares instead of using it to enlarge our disk cache, opting for reliability over performance. We have yet to determine the effects RAID has on our UniTree system.

Our real concern of late has been UniTree's current inability to migrate to more than a single tape at a time. In our experience, migration has never been able to proceed faster than one 3480 cartridge tape every six minutes. If migration performed at that peak rate round-the-clock, we would have no more than 240 tapes filled by the end of a 24-hour period. For a cartridge tape with a 200-megabyte capacity, this would mean no more than 48 gigabytes could be migrated each day. There have already been individual days when more than 35 gigabytes have been added to UniTree. The arrival of a new Cray C98 in late August will likely mean a three-fold increase in data production at our site, and if migration to tape cannot keep pace with the arrival of new data, UniTree will crash, irrespective of the size of the disk cache.

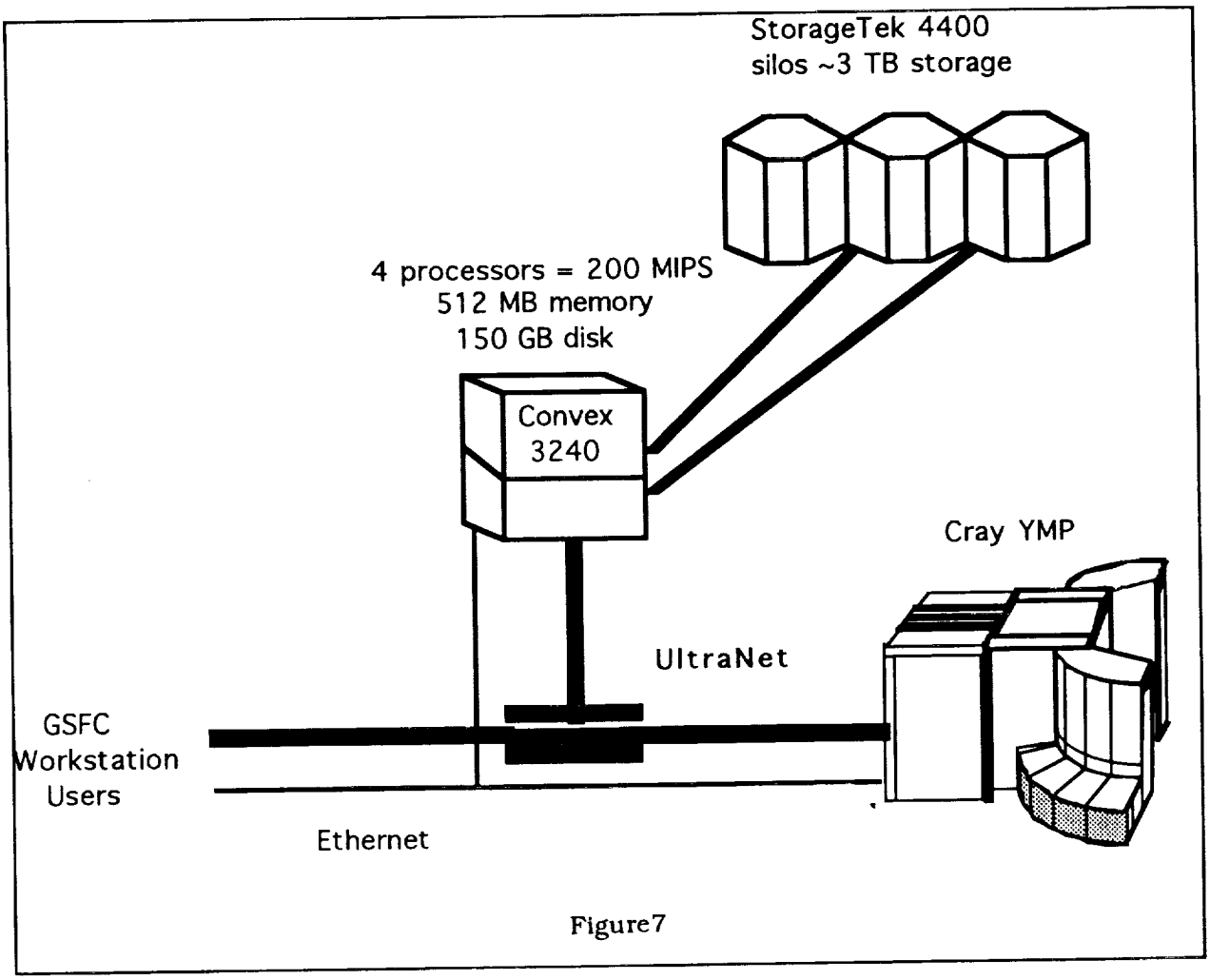



Figure7

For this and other reasons, we are eagerly looking ahead to future releases of UniTree. Convex/UniTree 1.7, released just recently, includes a new feature known as *family of files*, which will allow selected files to be migrated directly to offline tape, bypassing robotic storage entirely. In the same vein, large files could be automatically selected for denser archival media. By the end of the year, a performance release of UniTree 1.7 is expected that we hope will include faster writing to tape and will allow us to accommodate the storage rate anticipated from the new Cray.

## V SUMMARY

User acceptance of UniTree has been high, as evidenced by the rapid turnover of our disk cache (figure 6). We have had no complaints about the integrity of the data stored. Although users have found UniTree's instability to be frustrating, we believe that with time UniTree will prove to be the valuable and reliable storage system that mass storage sites have anticipated.

# Hierarchical Storage Management System Evaluation

**Thomas S. Woodrow**
NAS Systems Development Branch
NAS Systems Division
NASA Ames Research Center
Mail Stop N258-5
Moffett Field, CA 94035-1000
woodrow@nas.nasa.gov

**Abstract**

*The Numerical Aerodynamic Simulation (NAS) Program at NASA Ames Research Center has been developing a Hierarchical Storage Management System, NAStore, for some 6 years. This evaluation compares functionality, performance, reliability and other factors of NAStore and 3 commercial alternatives. FileServ is found to be slightly better overall than NAStore and DMF. UniTree is found to be severely lacking in comparison.*

## 1. Introduction and Problem Definition

The Numerical Aerodynamic Simulation (NAS) Program has been involved with Mass Storage Hardware and Software since its inception in 1984. In 1985, the Mass Storage Subsystem (MSS) Project was initiated to create an Hierarchical Storage Manager (HSM) to meet the needs of the NAS Program. Since 1985, there have been several releases of MSS software running under MVS and UNIX on Amdahl hardware and currently under UNIX on Convex hardware. During this period, several commercial alternatives appeared. These alternatives have now been available in the market for some time and have been subjected to the testing rigors of the marketplace. It is a good time to evaluate these packages, compare features and performance, and make a determination whether to continue internal development of NAStore or embrace one or more of the commercial alternatives.

The following packages are compared: the Data Migration Facility from Cray Research, FileServ from E-Systems, NAStore from the NAS Program, and UniTree from Open Vision.

This paper is arranged as follows: configuration, functionality, performance, hands-on experience, detailed observations, and conclusions. Weights are assigned for each category and totaled to give a final recommendation. The evaluation weights for each category are shown below:

| Category | Weighting Factor |
|---|---|
| Functionality | 11 |
| Performance | 23 |
| Reliability | 23 |
| Stability | 23 |
| Ease of Use | 9 |
| Outstanding Problems | 4 |
| Miscellaneous | 7 |
| **Total** | **100** |

## 2. Test Configuration

Three systems were used during testing; 2 Cray Y-MPs and a Convex C3820. The two Crays are both running DMF in production and have slightly differing configurations and loads. The Convex is one of two recently acquired systems for the NAS Mass Storage Subsystem.

Columbia YMP 2E 1/16, HiPPI UltraNet, Model E IOS, 1 - 4 channel TCA (total of 4 -3 MB/s paths), 16 - 3480 tape drives in 3 STK 4400 Silos, located in the Ames Research Center - Central Computing Facility
Filesystem: composed of DS-42s and DD-62s connected to one controller - not striped
UNICOS 7.0.4.3
Sun 4/330 running ACSLS 4.0
DMF 2.0
This system functions as a dedicated file server.

Reynolds YMP 8/256, HiPPI UltraNet, Model D IOS, 1 - 4 channel BMC (total of 4 - 3 MB/s paths), 16 - 3480 tape drives in 2 STK 4400 Silos, located in the Ames Research Center Numerical Aerodynamic Simulation Facility
Filesystem: composed of DS-42s and DD-49s connected to one controller - not striped
UNICOS 7.0.4.1
Sun SPARCstation running ACSLS 3.0
DMF 2.0
This system functions primarily as a compute engine. File serving functionality is of secondary importance except in support of the computation capability.

Pancho Convex C3820, 1 GB RAM, HiPPI UltraNet, 2 - 2 channel TLI interfaces (total of 4 - 4.5 MB/s paths), 16 - 3480 tape drives in 2 STK 4400 Silos, located in the Ames Research Center Numerical Aerodynamic Simulation Facility
Filesystem: composed of 4 wide stripes using DKD-504 disks across 4 IDCs
ConvexOS 10.2
Sun SPARCstation running ACSLS 3.0
FileServ 2.1.5, NAStore 2.1.1, UniTree 1.7.1.14
This system will function as a dedicated file server.

## 3. Features

### Major Software Components

DMF (from Cray Research Inc.)
- tpdaemon
- dmdaemon
- Media Specific Processes (MSP)
- Applications
  - dmput, dmget, dmlim, dmmode, several others

The Data Migration Facility is primarily composed of the dmdaemon, MSPs, some kernel modifications and some application programs. The dmdaemon handles all of the requests initiated by the user applications and initiates MSP actions. The system tape daemon, tpdaemon, allocates devices and controls tape mounts and dismounts. There are only two applications a user is ever likely to use: dmput and dmget. However, any UNIX file open will automatically cause a migrated file to be restored to disk. There are numerous other applications which are used primarily by operators. DMF uses

CTREE routines to maintain database information. DMF is a relatively simple implementation and has a comparatively small number of functional pieces.

DMF makes use of UNICOS kernel hooks to initiate automatic file restoration on file opens. The structure of the inode has been modified to show the migration state of a file. File reads and writes will block until a file is resident on disk.

FileServ (from E-Systems Inc.)

INGRES database daemons
(iidbms, dmfacp, dmfrcp, iigcn)
FileServ daemons
(fs_cpyreq, fs_media, fs_mcontrol, fs_monitor, fs_cpyresp, fs_resource, fs_alloc_s, fs_fcontrol, fs_rem_s, fs_admin, fs_data)
Applications
fsstore, fsretrieve, fschstate, fsmedinfo, fsmedlist, fsaddrelation, fsaddclass, others

FileServ uses the INGRES commercial database. There are many daemons and specialized processes which run to make FileServ function. There are a large number of applications available. One of the strengths of FileServ is the wealth of applications to check/modify status information and control variables. Another benefit is the ability to track the state of individual stores and retrieves while they are in progress.

FileServ makes use of the ConvexOS kernel modifications which cause a file open to automatically restore files from tape. File reads and writes will block until a file is restored to disk.

NAStore (developed at NAS)

voldaemon
rashd
Repository Controllers - Manual 3480, Manual 3420, ACS 3480
Applications
forcearc, frestore, arcbuild, archive, reclaim, volstat, volvary, vls, rls, vol, others.

NAStore has the following functional elements: the Volume Manager, *voldaemon*, uses repository controllers to mount and dismount tape volumes, Rapid Access Storage Hierarchy, *rashd*, uses the Volume Manager to move files in and out from various media through Repository Controllers. There are several applications, but only a few that users are likely to use: *forcearc* and *frestore* move files in and out from tape, *volstat* checks the queue of requests and *volvary* checks the state of tape devices. It is difficult, but possible, to see if your archive or restore is in progress. One of the strengths of NAStore is the relatively few pieces that must be running. NAStore uses BTREE routines to maintain file and tape information. One of the unique features of NAStore is that the tapes are standard ANSI format and are readable by ANSI tape readers.

Like DMF and FileServ, the ConvexOS kernel modifications are used to cause restoration on a file open. NAStore adds several fields to the Convex inode structure and therefore modifies several file related utilities to make use of the additional information stored in the inode (newfs, newst, ls, others).

UniTree (from Open Vision)

tpdaemon
UniTree daemons
(unamed, udiskd, utaped, uftpd, unfsmntd, tapesrvr, disksrvr, namesrvr, pdmsrvr, tapemovr, diskmovr, pvrsrvr)
Applications
uftp

UniTree is composed of a collection of specialized daemon processes, the Convex Tape Daemon, and a modified ftp daemon. There are a large number of running daemons. Under periods of high activity, we might expect high CPU load and lots of context switching. We were not able to check this due to several bugs encountered during testing. UniTree uses BTREE routines to maintain file and tape information. UniTree restricts all access to user data by forcing access through ftp or NFS. This is a restriction unique to UniTree.

UniTree formats, labels and controls its own file system on the Convex. This means that UniTree file systems are quite different than other file systems on the Convex. Rather than take the approach of using the kernel, UniTree runs entirely in user space with a number of daemons responsible for all event handling and file movement.

One of the problems with this approach is that access is limited to an application which links the UniTree library. UniTree provides NFS and FTP which have been modified to use the daemons for file opens, reads and writes.

An obvious strength to this approach is that it is very easy to port UniTree to new platforms. There are versions of UniTree available on systems ranging from Convex and Amdahl to IBM RS6000, SGI and Sun. There is also support for a wide range of output devices as well: IBM 3480 cartridges, Metrum VHS, 9-track round tape, others.

One artifact of a large number of UniTree ports is that each may be implemented from a different base version of the software. Most were ported and are supported by separate vendors. Many of the points brought up during testing are likely to be indicative of the large number of UniTree installations even though the implementation hardware may be different. Convex is the largest installed UniTree implementation with over 20 of 54 installations according to Max Morton of Open Vision.

## Storage Model

Each of the packages manages files in collections or groups. These collections govern how files are segregated on tapes (i.e. files within the collection can be mingled on tape).

| Package | Grouping Mechanism |
|---|---|
| DMF | filesystem |
| FileServ | class |
| NAStore | user |
| UniTree | family |

DMF

All files within a filesystem are managed together. This results in files from several users on a single tape.

When a file is disk resident, it appears as any other UNIX file in an *ls -la* listing:

```
-rw-r--r--  1  woodrow     npo          309848      Jul    20     23:27  file
```

When it is only available on tape, this is indicated by an "m" appearing in the first column of an *ls -la* listing:

```
mrw-r--r--  1  woodrow     npo          309848      Jul    20     23:27  file
```

FileServ

There are system defined *classes* which are mapped to directory trees. All files in a class are managed together. Similar to DMF, this means that user files within a class can be intermingled on a single tape.

When a file is disk resident, it appears as any other UNIX file in an *ls -la* listing:

```
-rw-r--r-- 1  woodrow     npo          309848      Jul    20     23:27  file
```

When it is only available on tape, this is indicated by an "a" appearing in the first column of an *ls -la* listing:

```
arw-r--r-- 1  woodrow     npo          309848      Jul    20     23:27  file
```

The *fsfileinfo* command will also list the state of the file, the number of tape copies and whether the file is resident on DISK, TAPE or DISK and TAPE.

NAStore

All files for a specific user are managed together. NAStore uses a hot tape model for tape writes. This means a "hot" primary and backup tape, will be written and filled before a new tape is used. There are special cases where multiple hot tapes can occur; for example when a file will not fit at the end of the current hot tape. There may be space available at the end of the tape, just not enough for the current file. In this case, a new hot tape is selected, resulting in multiple hot tapes. Files which will span tapes (files larger than 200 MB) are exempted from the hot tape mechanism and will start a new tape immediately.

NAStore, unlike DMF, FileServ and UniTree does not know how much space is left on tape. This means that NAStore may try to put a file on tape before finding out if it will fit. The other packages require the system operator to configure the tape size and then use this to determine whether a file should fit before trying.

When a file is disk resident, it can appear in one of the following two ways in an *ls -la* listing: 1) as any other UNIX file, or 2) with an "a" in the first column.

```
-rw-r--r-- 1  woodrow     npo          309848      Jul    20     23:27  file
arw-r--r-- 1  woodrow     npo          309848      Jul    20     23:27  file
```

When it is only available on tape, this is indicated by an "A" appearing in the first column of an *ls -la* listing:

```
Arw-r--r-- 1  woodrow     npo          309848      Jul    20     23:27  file
```

Since the archive state information is available from a file listing, it is easy to track the file archival state without learning any new commands.

UniTree

Files are grouped in system defined families and used similar to UNIX groups (i.e. users enter a *setfam* directive within FTP similar to *setgrp* in UNIX - all subsequent stored files are associated with the family). All files in a family are managed together and user data can be intermingled on tape. UniTree uses a hot tape model for archival. Unlike NAStore, however, UniTree proceeds linearly through tapes regardless of file size. This means that a spanning file (a file greater than 200 MB) can start in the middle of a tape. This is a decision unique to UniTree.

NFS does not display the archival state of a file. Within FTP, a *dir* or *ls* command will display whether a file is in disk cache (DK) or archived (AR). There is no mechanism to see if a file is on both disk and tape.

## Some Useful Functionality

Here is a short list of some basic functions which were used or would have been used during the evaluation. Most of these are provided by several of the packages, some are unique to one package or another. Some of these are not available and are included as suggested capability.

| Function | Description | D | F | N | U |
|---|---|---|---|---|---|
| List user tapes | List all tapes (primary and backup) in use by a user, class, family, filesystem. | X | X | | X |
| List files on a tape | List all files on a specific tape. | X | X | | |
| File Status | List the archival state of a file | X | X | X | X |
| | List the tapes which hold a file (primary and backup) | X | X | | |
| List Tape Blocks Used | This is a modification of the UNIX *du* utility to list blocks used for migrated files. | | | | |
| Drive Status | List the system defined drives and their current state. | X | X | X | X |
| Tape Status | List the % full or bytes on tape | X | X | | X |
| | List the files known to reside on the tape. | X | X | X | |
| Archive/Restore Status | List the percent complete for all archive or restore operations | | | | |
| | Given a user/family/class/filesystem id, identify all archives or restores active and their completion state | | | | |
| Media Lists | Provide information on tape media in use (virgin/free, labeled, allocated, bad, etc.) | X | X | X | X |
| | Provide the option for a long or tabular presentation of this data | | X | | |
| Media Labeling in Parallel | A Parallel labeling utility. | X | X | | |
| Media Recycle/Reclaim in Parallel | A Parallel Recycle/Reclaim utility - i.e. the operator should not have to manually parallelize this function. | X | X | | |
| Media compaction | The ability to compact out the old dead files from a tape and hence realize some savings in the tape inventory. | X | X | | X |
| Database Lists | List all versions of a file stored in system | X | | | |
| Misc. Configuration Stuff | Specify tape quota | | X | | |
| | Specify user files to keep on disk | X | X | X | |

There is a difference between file migration, the act of copying a file to tape, and file truncation, the act of removing the file from the disk after migration. UniTree treats

this as a single logical operation, i.e. its *forcemig -all* command. DMF, FileServ and NAStore all have separate utilities which perform migration and truncation. As long as migration is performed as files are written to disk, truncation can be performed as necessary to keep the disk usage within a controlled range. For DMF, FileServ and NAStore file truncation is performed very quickly and is not measured. Each HSM has a different utility to build the list of candidate files for truncation. These provide a sorted list of files eligible for truncation from disk. The list is then used to maintain a file system percent utilization.

Tape compaction is provided by all of the packages except NAStore currently. This is an important capability and will be required for production use.

FileServ and DMF are very complete in functionality. NAStore needs tape compaction. UniTree is hit for the lack of analysis tools and capability.

**Scores**

| | | | | |
|---|---|---|---|---|
| **Functionality [11]:** | **DMF: 11** | **FileServ: 11** | **NAStore: 10** | **UniTree: 8** |

## 4. Performance

NAS sizes the Mass Storage System to hold the latest 30 days of data on disk. This is based on the assumption that the most current data has the highest usage and should have the fastest access. File access from disk or the highest level of the storage hierarchy is therefore one of the most important performance elements of any HSM. Of course, some files will be accessed which are not resident on disk. This makes the tape system read performance of secondary importance. The ability of the system to process multiple requests simultaneously is important. For this reason, we measure both individual user and system aggregate performance for tape operations.

Tape read performance is usually more important to the user than write performance, since most tape writes are initiated by the system and can be scheduled. Tape reads are initiated in response to an immediate need by a user. Tape read performance should therefore be as fast as is possible on the media. Tape write performance is measured for individual files and under various system loads. File migration and truncation performance from disk is measured to see the relative ability of these packages to keep up with a stream of input data.

A user workload of file reads and writes using ftp with a mix of resident and migrated files is used to make an overall relative assessment of the HSMs. This is probably the most realistic measure of user performance of all the tests since it incorporates disk, tape and network performance.

### 4.1 Individual file, disk read and write performance

Test Description
Files of various sizes (256K, 10M, 75M, 300M) are read from and written to disk. Sequential I/O tests are used, since file access will typically require a network transfer and will involve reading an entire file. A program called *multirate* is used to measure disk performance on the Crays and the Convex. This utility was acquired from Convex and has been used on site for some time.

There are several factors to note when looking at the disk results:

1) The DMF file systems installed on the 2 Crays deliver 8 - 10 MB/s for individual file reads and writes because of the current hardware configuration.

2) The file systems installed on the Convex can deliver 28-31 MB/s for individual file reads and writes.

Given these points, we are also interested whether the HSM delivers the native filesystem performance or degrades it.

Results
From Figure 1, NAStore and FileServ both deliver the highest performance to and from disk. Based on the measurements, DMF, FileServ and NAStore all deliver native filesystem performance. However, UniTree degrades the native file system performance by about 66%. The DMF is given 2 scores: 1) compared to the installed Convex filesystem the performance is lower - however this is not a DMF problem 2) if the filesystem were at the same or higher performance then we would expect DMF to deliver this performance - however this is an assumption. Both of these scores will be discussed in the performance summary.

**Scores:**
**Disk Read/Write** **DMF: 0.31 (1.0)** **FileServ: 1.0** **NAStore: 1.0** **UniTree: 0.33**

## 4.2 Individual file, tape read and write performance

Test Description
Files of various sizes (256K, 10M, 75M, 300M) are written to and read from tape using the HSM user commands (dmput/dmget, fsstore/fsretrieve, forcearc/frestore, put/get[1]). Elapsed wall clock time is used to measure the duration of the write/read. Measurements include the time to archive a primary and backup copy of a file. All timings take into account tape mount activity. The intent is to report rates which are as close as possible to user experienced rates.

It is important to note the following in this test:
1) the peak 3480 tape drive performance is 3.0 MB/s
2) it is interesting to note the ability of each package to attain tape drive peak
3) higher average performance is expected for larger files as mount activity is reduced as a proportion of the total elapsed time

Results
During testing, the following was observed
DMF sustained 2.0 MB/s on writes and reads
FileServ sustained 2.0 MB/s on write and 2.7 MB/s on reads
NAStore sustained 2.7 MB/s on writes and reads
UniTree sustained 1.5 MB/s on writes and reads

All HSMs exhibit an increase in performance as file size increases. Tape mount time was observed to be 20 - 50 seconds depending on silo load.

Figure 2 shows that NAStore dominates tape read and write performance, especially as file size increases. UniTree is quite a bit slower than the rest. Also, Columbia outperforms Reynolds in read tape performance with DMF. This is most likely attributable to the newer silos which likely have had engineering improvements since the installation of the silos at the NAS facility. The effect of the engineering improvements is faster tapes mounts. Unfortunately, we have incomplete numbers for Columbia due to the scarcity of dedicated Cray time.

[1] UniTree access is through ftp. This is true even when accessing files on the local system. *Put* and *get* are the ftp commands used to move files in and out of a UniTree controlled filesystem. *Get* is also used when retrieving a file from tape.

**Scores:**

| | | | | |
|---|---|---|---|---|
| **Tape Write** | **DMF: 0.71** | **FileServ: 0.81** | **NAStore: 1.0** | **UniTree: 0.43** |
| **Tape Read** | **DMF: 0.74** | **FileServ: 0.94** | **NAStore: 1.0** | **UniTree: 0.55** |

### 4.2.1 Tape handling optimizations

Following is a list of the kind of optimizations observed among the different systems. Given the fixed performance of the tape media, it is desirable to maximize the delivered performance from that media. Some of these optimizations are simple to implement. We hope all of the packages will continue to improve in this area.

| **Optimization** | **Exists in Package** |
|---|---|
| Parallel primary and backup tape writes. | DMF, NAStore |
| Asynchronous primary and backup tape writes. | DMF |
| Pre-mount tapes for spanning file reads | NAStore |
| Check pending requests for a tape prior to a dismount | DMF, FileServ |
| | NAStore has implemented this feature but it has not yet been tested |
| Optimize wild card transactions (reads and writes) | DMF, FileServ, NAStore |
| Cache data on disk and perform only full tape writes | DMF |
| | UniTree - this was not working in our version |
| Use any available tape for a write (not just the "HOT" tape) | DMF, FileServ |
| Spanning files start with a new tape | DMF, FileServ, NAStore |
| Optimal use of tape system data paths. | NAStore |

Listing an optimization does not mean that it is necessarily the best choice, it was simply observed. NAStore and UniTree both sacrifice some tape write performance in using HOT tapes. However, both may reap improved performance during restores because improved locality of user or family files should reduce tape mounts.

NAStore tries to make optimal use of the system data paths by rotating mounts through controllers. This has noticeable effect during periods of light load.

## 4.3 Multiple file tape read performance

Test Description

Multiple files are written to tape in simultaneous streams. Each stream is composed of consecutive 256K,10M, 75M, or 300M files. Streams have differing numbers of files based on file size, in an attempt to keep all streams active for the entire test. The test is run two different ways:

1) with separate store commands for each file
2) with a single wild carded store command for an entire stream.

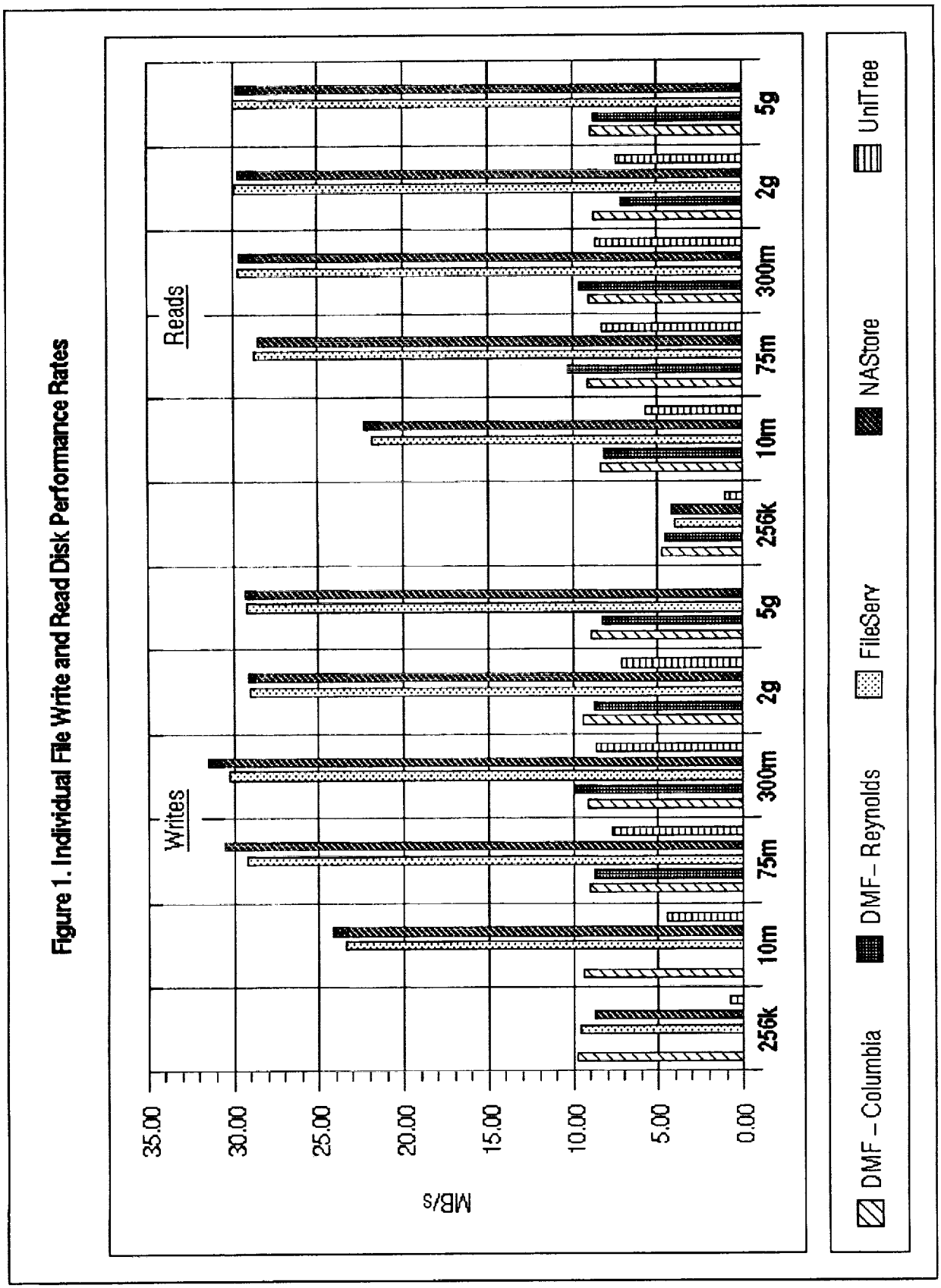
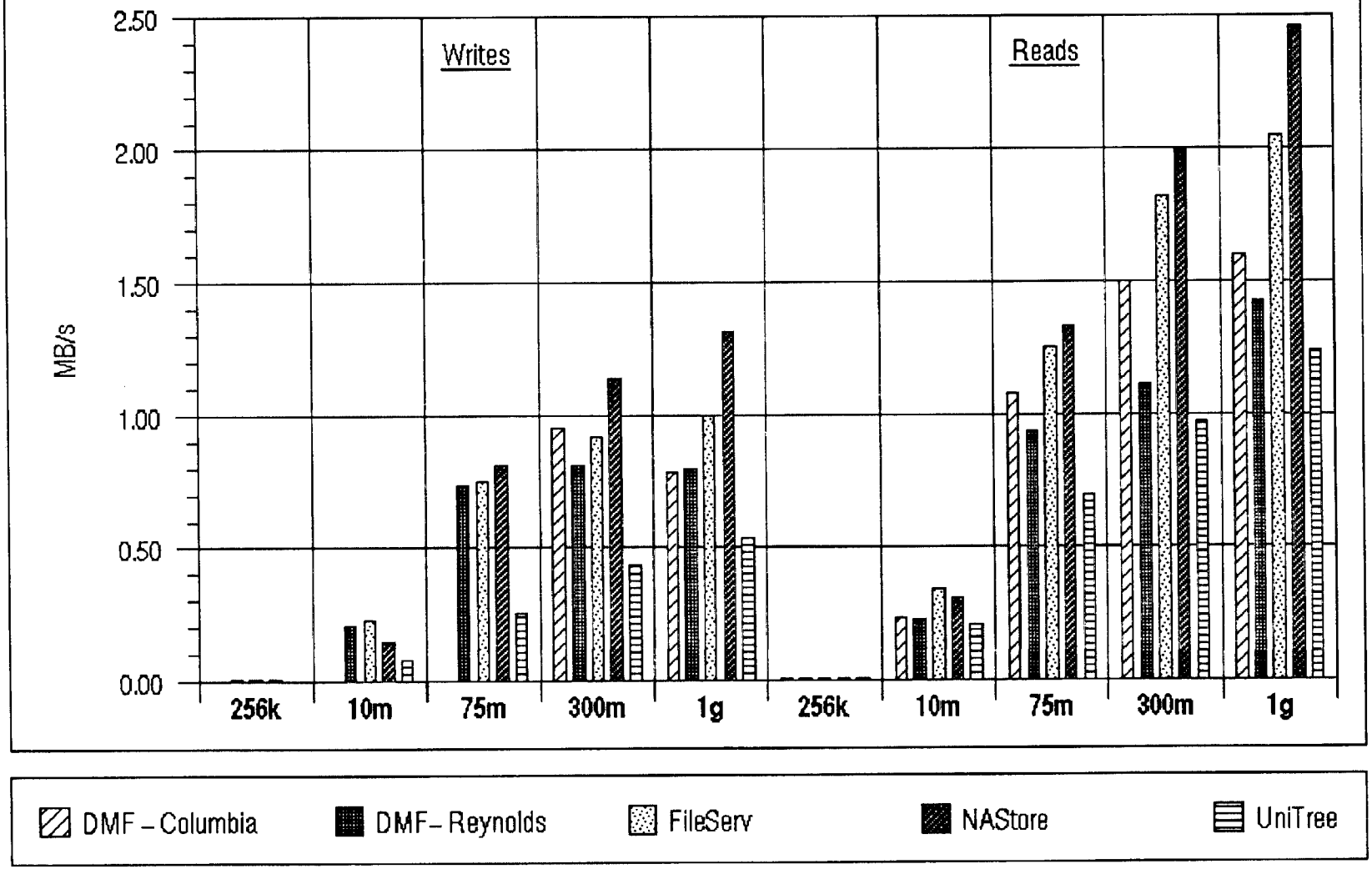



**Figure 1. Individual File Write and Read Disk Performance Rates**

**Figure 2. Individual File Write and Read Tape Performance Rates**



Data in Figures 3 and 4 show the average rate across all streams against the simulated load level. Figure 3 shows the performance on the system when issuing a separate store command for each file. In this test, a higher level of tape mounts and dismounts was expected. In Figure 4, a single wild carded command performs all of the writes. In this case, the system aggregate performance is expected to rise due to reduced mounts and dismounts.

System aggregate performance over time data is also gathered for these tests, although only on the Convexes. These charts are included as an appendix. They are not used to formulate scoring, but are discussed briefly in section 6, **Miscellaneous Points and Observations**.

Results
All candidates were tested in this area, but UniTree had a problem on the Convex which caused FTP to block during file reads from tape. This made it impossible to automate testing of a multiple file restore. We were unable to work around this problem. As a result, UniTree has rather few measured results from here on. This problem does not mean that users cannot get files from UniTree file systems using FTP, but it does mean that several commands may be necessary to retrieve a single file. This was determined to be unworkable for performance testing purposes.

Figure 3 shows the average system throughput to restore files for various simulated user loads. Comparing the results from Figure 3 and Figure 2, we might expect that the system throughput to be higher. The primary reason system throughput appears lower is that we average across all file sizes. FileServ has a slight edge in performance over both DMF and NAStore in this test.

Figure 4 shows the wild carded restore system performance. FileServ dominates at lower system loads and then NAStore outperforms all others at higher loads. Both FileServ and NAStore exhibit significantly improved system performance when file restores are wild carded. DMF (reynolds) does not display improved performance in this category which is puzzling, especially since DMF does optimize mount/dismount activity. One possibility for the performance behavior may be a rewind/seek between files restored from the same tape - this is not confirmed. The DMF columbia result at 8 simulated users suggests that columbia could outperform reynolds in this test. We were unable to complete the test matrix for columbia because of lack of additional dedicated time.

The performance difference between Figures 3 and 4 suggests that users would be well advised to aggregate file reads and writes from an HSM.

**Scores:**

| | | | | |
|---|---|---|---|---|
| **Separate:** | **DMF: .84** | **FileServ: 1** | **NAStore: .79** | **UniTree:0** |
| **Aggregated:** | **DMF: .48** | **FileServ: 1** | **NAStore: 1** | **UniTree:0** |

### 4.4 Migrate and Truncate data from Disk

Test Description
This test measures the speed the of each package to archive and truncate files from disk. This is an important measure of how well an HSM can keep up with data as it hits the system. This test differs from the previous test, only in how it was initiated. A single, wild carded command is used to initiate the store. After the store completes, a truncate command is issued. The total elapsed time to archive and truncate the files from disk is measured.

**Figure 3. Average System Throughput – Separated Reads, Various Simulated Loads**

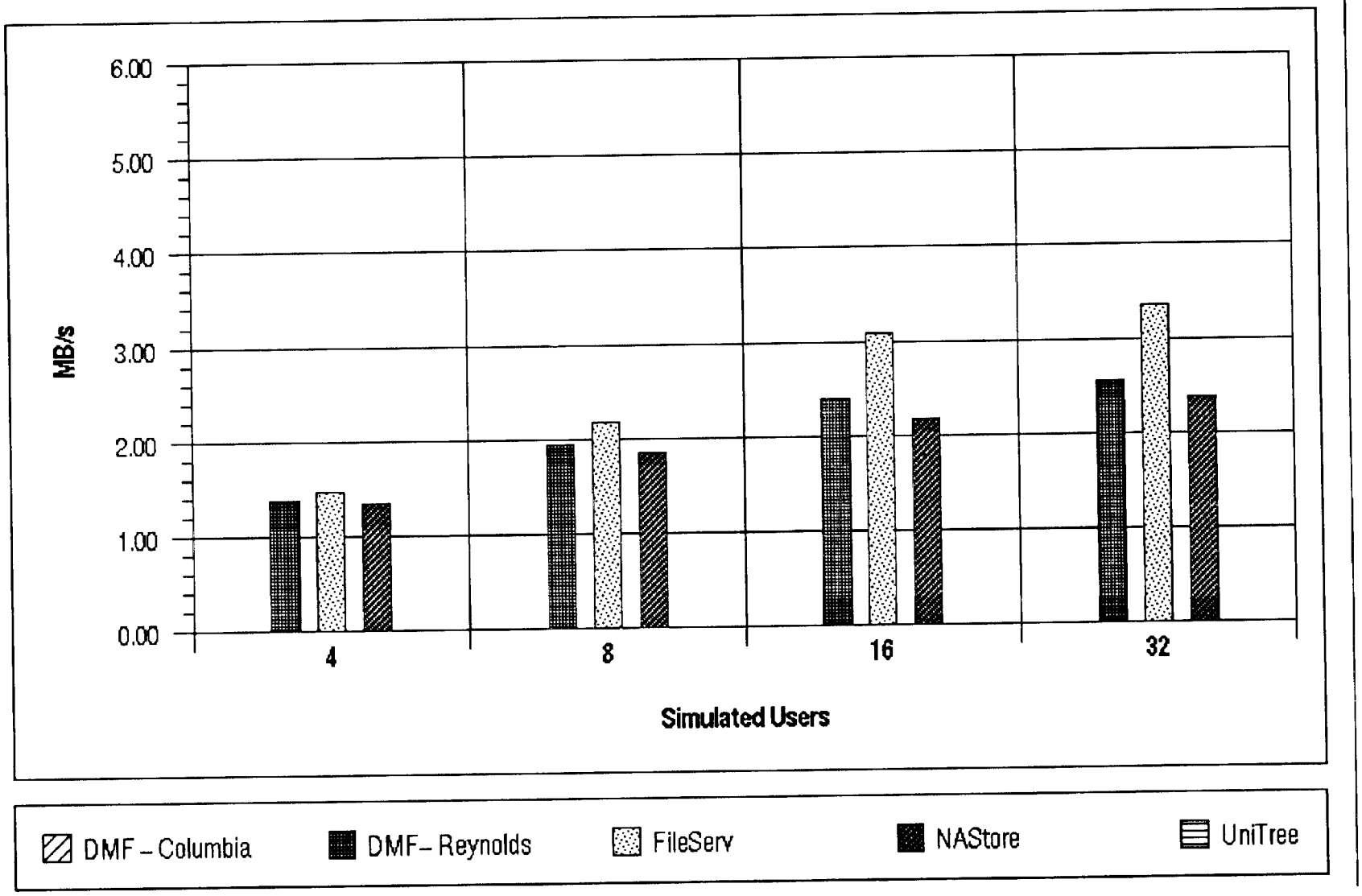
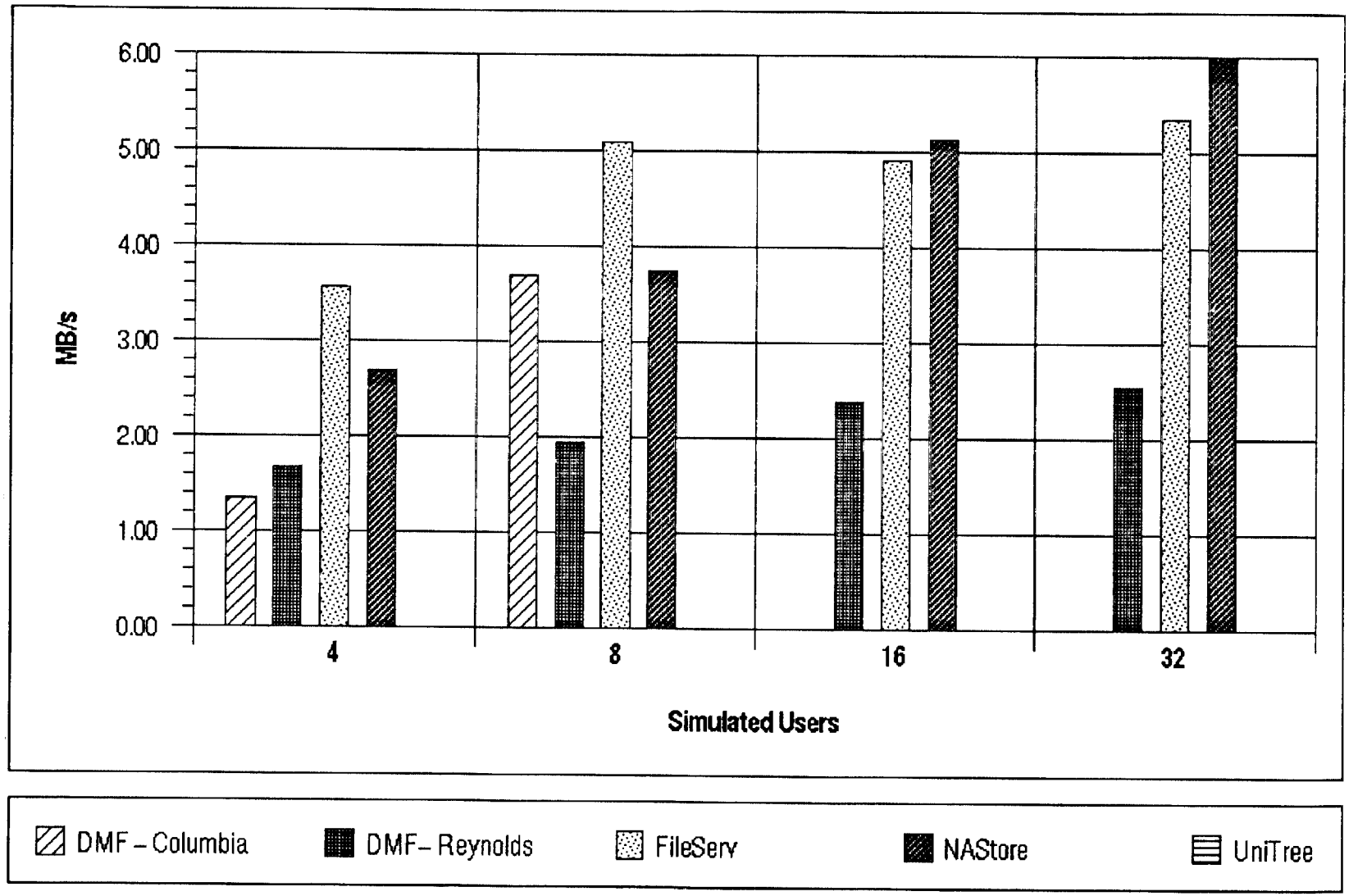


**Figure 4. Average System Throughput, Aggregated Reads, Various Simulated Loads**



Results
DMF, FileServ and NAStore all perform in the same approximate range here, although FileServ has a slight edge. UniTree is significantly slower. One of the primary reasons for this is that UniTree writes a single stream of data (1 tape drive) and does the primary and backup copies sequentially. This seems to be a feature in all UniTree versions according to discussions with other sites.

**Scores:**
**Migrate/Truncate: DMF: 0.81 FileServ: 1.0 NAStore: 0.74 UniTree: 0.17**

## 4.5 Workload performance disk and tape read and write

Test Description
This test incorporates elements of network, disk and tape performance by simulating the activity of 26 simultaneous users using ftp over UltraNet. The user data population was examined on the previous storage system and users were consulted to define a workload profile for the system. The workload definition follows:

| User Type | Description | File Volume | % of population |
|---|---|---|---|
| 1 | Workstation Backup | 1 - 150K files | 15 |
| 2 | Miscellaneous Small | 50 - 20K files | 20 |
| 3 | CFD steady state | 5 - 10M files | 45 |
| | | 5 - 75M files | |
| 4 | CFD small unsteady | 20 - 75M files | 15 |
| 5 | CFD large unsteady | 50 - 300M files | 5 |

Ideally we would have run increasing loads of users based on this breakdown till we saturated the system. In the future, perhaps we will do this. In the interest of timeliness we selected a representative load; 26 users.

Each storage management system is initially configured with a predetermined number of files resident on disk and migrated to tape. Simulated users issue ftp get and put commands to move files in and out of the system under test. The test is driven from a Cray C-90 and uses native UltraNet paths to ensure that neither the driving system CPU nor the network are the bottleneck. High, low and average performance are calculated and plotted by file size.

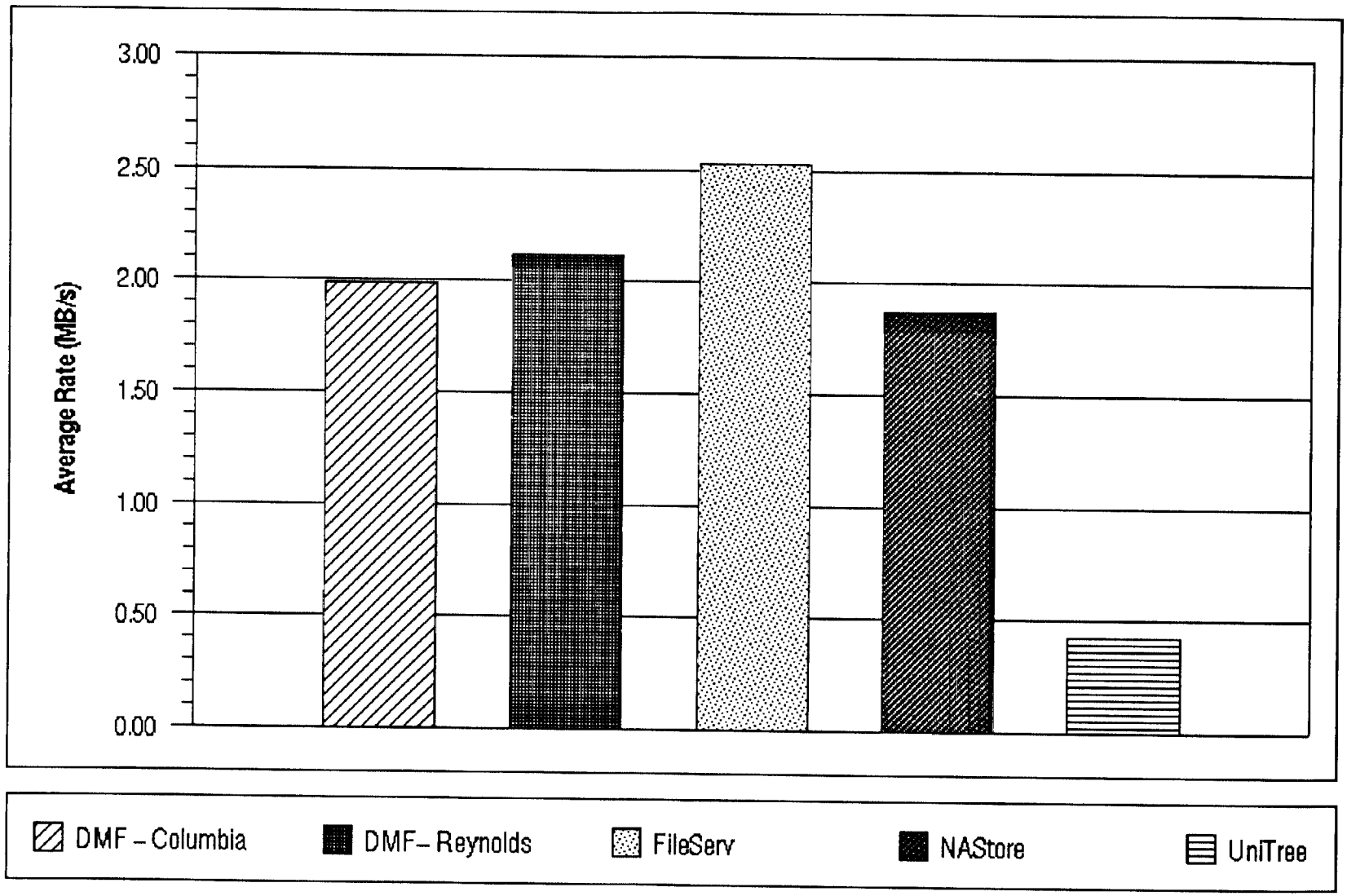
One unintended limitation imposed on this test is the C-90 filesystem. Since gets and puts utilize the C-90 file system, this creates a ceiling for individual file transfers. In the future, we will use a faster file system on the driving system.

The following are considered in ranking the packages:
1) the average performance at each file size
2) the variation in performance at each file size

Results

Figure 6 shows the high, low and average rate for reads and writes by file size. The data points on the X axis are broken out by package at each file size. DMF, FileServ and NAStore results are shown from left to right.

**Figure 5. Effective System Rate to Archive/Truncate 8.0 GB of User Files**



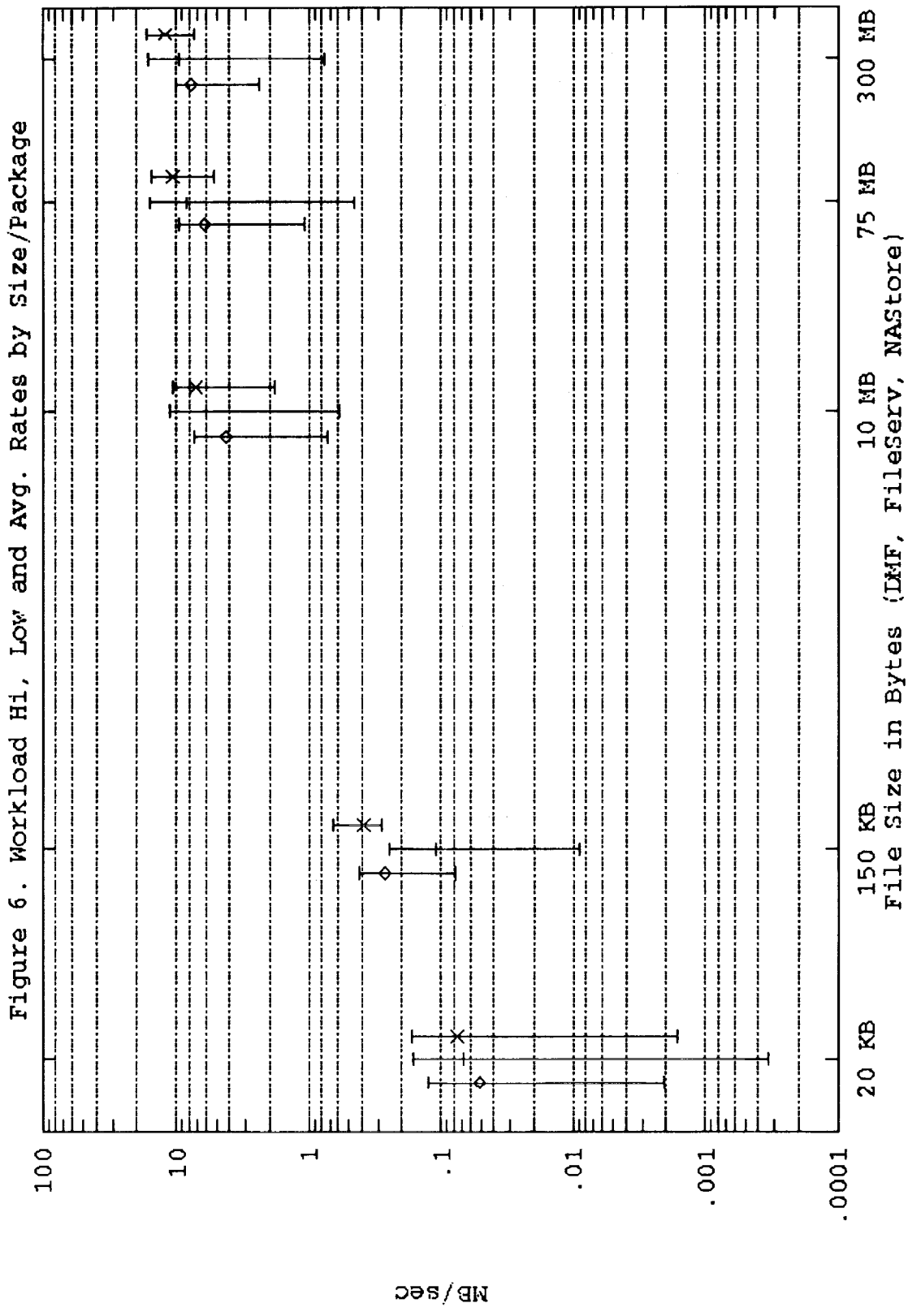


Figure 6. Workload Hi, Low and Avg. Rates by Size/Package
MB/sec
100
10
1
.1
.01
.001
.0001
20 KB
150 KB
10 MB
75 MB
300 MB
File Size in Bytes (IMF, FileServ, NAStore)

NAStore performs better than both DMF and FileServ at all file sizes. In almost all cases, NAStore has less variation between the high and low rate than the others. FileServ, although usually 2nd in average performance, has a large fluctuation in performance at each file size.

Given the C-90 file system limitation mentioned above, we still see significant performance differences between the packages.

**Scores:**
**Workload: DMF: 0.65 FileServ: 0.82 NAStore: 1.0 UniTree: 0.0**

## 4.6 Performance Totals

Performance tests were broken into 4 functional areas and given the following weights:

| | |
|---|---|
| Disk Read and Write | 100 |
| Tape Read | 55 |
| Tape Write/Migrate | 20 |
| Workload | 25 |

These weights reflect the relative importance of disk read and write performance and tape read performance. As long as the tape write performance can keep up with the daily load, it is adequate. The workload reflects all of these categories, but is perhaps a more representative measure of user performance.

Existing systems were utilized for performance testing. We are well aware that both Convex and Cray are capable of configuring systems with more or faster hardware. We made a conscious choice to measure existing systems and rank them based on the current configurations. File system performance, within reason, is largely a cost issue.

The table below shows only the totals for the performance scoring. The complete, weighted scoring table is Figure 7.

| | |
|---|---|
| **Total Points** | **210** |
| **DMF** | **100.25* (169.25)** |
| **FileServ** | **192.27** |
| **NAStore** | **192.42** |
| **UniTree** | **40.47** |

From a performance standpoint, FileServ and NAStore are identical. DMF is also a strong contender. The DMF performance total is 169.25 - a strong third, were we to assign full points for the disk performance test. Again, we ran against existing hardware configurations. UniTree would definitely be higher without functional problems on tape restores.

# 5. Hands-On Experience

This sections summarizes experiences during installation, configuration and usage of the packages.

## Figure 7. Performance Rank of HSMS Alternatives

| | Disk Read/Write | | | Indiv. Tape Read | | | Indiv. Tape Write | | | Syst Tape Read (sep) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | weight | act/max | points | weight | act/max | points | weight | act/max | points | weight | act/max | points |
| DMF | 100 | 0.31 (1.0) | 31 (100) | 15 | 0.74 | 11.1 | 7 | 0.71 | 4.97 | 20 | 0.84 | 16.8 |
| FileServ | 100 | 0.99 | 99 | 15 | 0.94 | 14.1 | 7 | 0.81 | 5.67 | 20 | 1 | 20 |
| NAStore | 100 | 1 | 100 | 15 | 1 | 15 | 7 | 1 | 7 | 20 | 0.79 | 15.8 |
| UniTree | 100 | 0.27 | 27 | 15 | 0.55 | 8.25 | 7 | 0.43 | 3.01 | 20 | 0 | 0 |

| | Syst Tape Reads (agg) | | | 8.0 GB migrate | | | Workload | | |
|---|---|---|---|---|---|---|---|---|---|
| | weight | act/max | points | weight | act/max | points | weight | act/max | points |
| DMF | 20 | 0.48 | 9.6 | 13 | 0.81 | 10.53 | 25 | 0.65 | 16.25 |
| FileServ | 20 | 1 | 20 | 13 | 1 | 13 | 25 | 0.82 | 20.5 |
| NAStore | 20 | 1 | 20 | 13 | 0.74 | 9.62 | 25 | 1 | 25 |
| UniTree | 20 | 0 | 0 | 13 | 0.17 | 2.21 | 25 | 0 | 0 |

| Total Points | 210 |
|---|---|
| DMF | 100.25 (169.25) |
| FileServ | 192.27 |
| NAStore | 192.42 |
| UniTree | 40.47 |

## 5.1 Reliability

An exhaustive reliability test including testing of failure modes was not undertaken. Therefore results in this section are limited to experience during testing. There was no data loss or corruption by any of the packages during performance or functionality testing.

During NAStore beta test, it was discovered that exec did not block correctly on files which were non-disk resident. This resulted in erratic behavior until the file was entirely disk resident. A simple kernel change was implemented and this was cured.

**Scores**
**Reliability [23]: DMF: 23 FileServ: 23 NAStore: 23 UniTree: 23**

## 5.2 Stability

Each package has unique stability issues.

***DMF***

- We encountered difficulty when bringing down DMF with tape write processes active. This turned out to be a cockpit error, but the experience points out the complexity involved with this systems.
+ Overall DMF is rock-solid stable.

***FileServ***

- Out of 20 - 30 sessions, we did cycle the software several times to clear hung behavior. This was related to a bug in the first tested release. In the second release, this problem was fixed.
- There was a problem involving badly formed silo addresses, by the mount utility. This had the effect of flipping some address information and making tapes unavailable. The work around involved running an SQL script continuously in the background. This script worked fine, but added a significant load to the database. During heavy load, we could get into a confused state. This was also repaired in the second release we received to complete testing.
+ A system crash occurred during FileServ testing. We were able to restart the software after reboot without problem.
+ FileServ is rather simple to start and stop. Once started, it is very stable.

***NAStore***

- Corrupted individual user rash index files. Rash recreated these indexes later with no data loss. This bug bas been repaired and tested.
- Tape devices were vary'ed off during high levels of activity. This meant that we closely monitored activity during NAStore tests. This was attributed to an error prone TLI driver on the Convex. There were several instances when the voldaemon core dumped and disappeared during high incidence of TLI errors. There were some changes made in NAStore to back off on a drive when a high number of errors were seen, this did reduce the number of drives which were vary'ed off significantly and removed the voldaemon core dumps.
+ NAStore stayed up and worked well after it was initiated.

***UniTree***

- Encountered problems getting NFS to work at first, eventually got things working with help from Convex.

- There were numerous times when it was difficult or impossible to dismount the UniTree NFS partition during shutdown. This had an effect on the stability of the entire system, even after a reboot.
- UniTree has a tendency to halt completely when it encounters a bad or badly labeled tape. This meant constant monitoring of the log files during testing.
+ In general UniTree could be trusted to stay up if it did not encounter and tape problems.

It is difficult to determine the stability of software over a short period of time. There is extensive local experience with DMF in production usage on several Cray systems. There is also less extensive, but still significant experience with NAStore during beta test (4 - 5 weeks) and during unit and integration testing (2 - 3 months). Our experience with UniTree and FileServ has been for a short period of time (2 - 4 weeks). This experience certainly influences our impressions on stability.

**Scores**
**Stability[23]: DMF: 23 FileServ: 21 NAStore: 21 UniTree: 15**

## 5.3 Configuration, Documentation and Ease of Use

***DMF***
+ Documentation is terse, but complete
+ Tape recycling/compaction is very simple to use
- Some operator utilities seem needlessly difficult to use (ex: dmvdbgen)
- If a file is on disk, the user may not be able to determine if it is also on tape.

***FileServ***
- Adding tape media to the system requires a pass through the cap door. This means that even if you already have a silo full of virgin tapes, FileServ would like you to remove them and define them by entering them through the cap. This problem has been fixed for the next version.
+ Tape labeling is done in parallel using all drives - painless and fast.
+ Tape recycling is very simple.
+ Easy to define classes, add relations, change configuration, administer package.
+ Easy to view archive activity (fschstate, fsqueue, fsmedinfo)

***NAStore***
- Tape labeling is manual, sequential, single stream (however, the user can start up multiple streams with separate lists) - user supplies a list to a utility
- Tape recycling is manual.
- On-line documentation is not as strong as the other packages.
+ Strong software architecture documentation.

UniTree
+ Configuration is mostly localized to just a few files
- FTP/NFS access is maddeningly restrictive.
- knowledge and viewing of UniTree log files is essential to monitor activity

**Scores**
**Config, Doc, Ease of Use [9]: DMF: 7 FileServ: 9 NAStore: 7 UniTree: 5**

## 5.4 Outstanding Issues

***DMF***
• Dmdidle is required to force data to tape, when there is not enough data waiting for archival. This is a root only command. If a user determines that he wants to force a file to tape, but the system does not have a tape full ready to write, the user will block

until the system has enough data to fill a tape. It is not possible for a user to find out how much data must be sent to work around this situation.
- Limit of 8 processes per MSP. There is a primary and secondary Media Specific Process. Although there are 16 tape drives, during restores all files are required to go through the primary MSP, unless a tape error is encountered. This places a seemingly arbitrary limit on the number of simultaneous restores possible. This has apparently been fixed in a patch to be released very soon.

***FileServ***
- Flipped silo identifiers on each tape with the initial version of FileServ we tested. This was resolved with a bug fix.
- Tape entry through cap door in this version of the software. This has been fixed in the next version.

***NAStore***
- Rash indices were corrupted several times. This has been fixed and tested.
- Tape devices were vary'ed off during high load. A work around is in place.
- Voldaemon died several times - sometimes requiring a reboot to clear a hung named pipe. This has been fixed and tested.

***UniTree***
- Single tape failure halts migration
- "quote wait get" hangs
- Extremely difficult to measure performance
- Wrapping files > 2 GB. During an ftp put of a 5 GB file, the system was observed to wrap the file at 2 GB back to 0 and then continue. The final file size was 1 GB. There were no errors reported.

**Scores**
**Problems[4]: DMF: 3 FileServ: 4 NAStore: 4 UniTree: 0**

## 6. Miscellaneous Points/Observations

- It is difficult to tell the state of an individual user transaction with DMF. During "puts", files are gathered in a caching directory until a full tape is ready, then written. Although the dmput may return immediately, a user's files may not get out to tape for some time. During "gets", files may be in the process of restoration, but unless the user can make the association between his tape ids and what is currently mounted, he is unable to tell if his transaction is active or not (this is especially true on a busy system).

+ After doing a DMF dmput, a user's disk utilization is immediately reduced by the size of the file, even though the file may not have left the disk cache. This is certainly a desirable feature for the user who is running at or near his quota on a Cray.

+ DMF used a small percentage of the system CPU. During the simulated user activity to disk, system usage ranged from 1 - 5%. This percentage is based on an 8 CPU system.

- DMF uses a proprietary tape format.

- When the FileServ daemons are not present (i.e. running) directories and files under FileServ control cannot be listed or read.

- FileServ's reliance on INGRES means the FileServ administrator should be versed in SQL. INGRES is also one of the major bottlenecks in FileServ performance, since

all transactions must refer to the database. DMF, NAStore and UniTree all utilize BTREE or CTREE for database functionality rather than use a commercial database.

- FileServ developed a high system load on the Convex. During the 32 Simulated User restore, system load averaged 50 - 80%. While this is not critical, it is a warning sign. NAStore was well below 50% utilization. We were not really able to drive UniTree hard enough to know how it behaved under load.

- FileServ uses a proprietary tape format.

+ NAStore is the only package which delivers bytes immediately as they hit system memory.

+/- NAStore sacrifices some archival performance to group files by user. This is based on the assumption that the individual user will benefit during restores, since his files will be closer together (fewer mounts during restores). I think the positive aspect of faster restores far outweighs the negative aspect of slower archives.

- NAStore is only in use at NAS. The only support for NAStore is provided locally. The only users who have experience with the system are at one location.

- It is difficult, but possible to track the state of an individual *forcearc* or *frestore* transaction within NAStore.

+ NAStore produces ANSI standard formatted tapes.

- UniTree is very difficult to measure. Using FTP for all transactions, makes measurement of the individual components of a store almost impossible. Although FTP reports the time to restore a file, it does not begin measurement until a file is resident on disk cache. Therefore the numbers FTP reports are suspect. Many times we were reduced to watching the UniTree log files.

- Although UniTree does give the user access to data through NFS, it is not possible to determine disk residency using NFS. File read and write performance using NFS, even on the local system, is poor (significantly lower than FTP) and is discouraged.

- UniTree uses a proprietary tape format.

All of the packages were hit for a proprietary tape format except NAStore. Many people seem to agree that this is important, however few have implemented using a standard.

System Aggregate vs. Time Plots give a picture into the system of how performance changes over time. These were produced on the Convex because of an existing utility called *syspic* and access to the source code. A utility exists on the Cray which could be modified to provide this data, but there was insufficient time prior to testing.

From these plots (included as an appendix), it is clear that there is large variation in performance over time. One performance goal for these packages could be that the average system tape performance during load approach (number of data paths * peak drive performance). On all of the systems tested, this number is 4 * 2.7 MB/s = 10.8 MB/s. Tape mounts and dismounts will always reduce the system aggregate performance from a theoretical peak, but with 16 drives, it should be possible to keep the data paths well utilized during periods of high load. A more aggressive goal would be to approach (number of data paths * data path rate) = 4 * 4.5 MB/s = 18 MB/s.

**Scores**

**Misc [7]:** **DMF: 6** **FileServ: 4** **NAStore: 6** **UniTree: 3**

## 7. Conclusions

### Summary Scoring

| Category | Weight | DMF | FileServ | NAStore | UniTree |
|---|---|---|---|---|---|
| Functionality | 11 | 11 | 11 | 10 | 8 |
| Performance | 23 | 11.04 (18.4) | 21.16 | 21.16 | 4.6 |
| Reliability | 23 | 23 | 23 | 23 | 23 |
| Stability | 23 | 23 | 21 | 21 | 15 |
| Ease of Use | 9 | 7 | 9 | 7 | 5 |
| Outstanding Issues | 4 | 3 | 4 | 4 | 0 |
| Miscellaneous | 7 | 6 | 4 | 6 | 3 |
| Total | 100 | 84.04 (91.4) | 93.16 | 92.16 | 58.6 |

FileServ is the most well rounded product, based on all of the factors considered. NAStore is a strong second. DMF comes in third unless the filesystem performance is factored out. Removing the penalty for a slow filesystem, DMF still falls 3rd just behind NAStore. UniTree loses many points in the performance area on tests which were not completed due to a bug. If this bug had been fixed, UniTree would be a stronger contender but would still place fourth due to lower performance and stability.

The decision of which Storage System to put into long term production at NAS is a judgment call which will involve the technical comparison, cost information and other factors. A discussion of costs would have restricted the availability of this report and was therefore dropped. One of the other factors that will be considered in the production decision is the ability to influence or make change in the Storage System in response to local requirements. NAStore is certainly the easiest product for NAS Program to influence and change. There is no major internal development required to use NAStore in production service. There are some features which can and will be added, but these can be accomplished with a sustaining level of effort.

## Acknowledgments

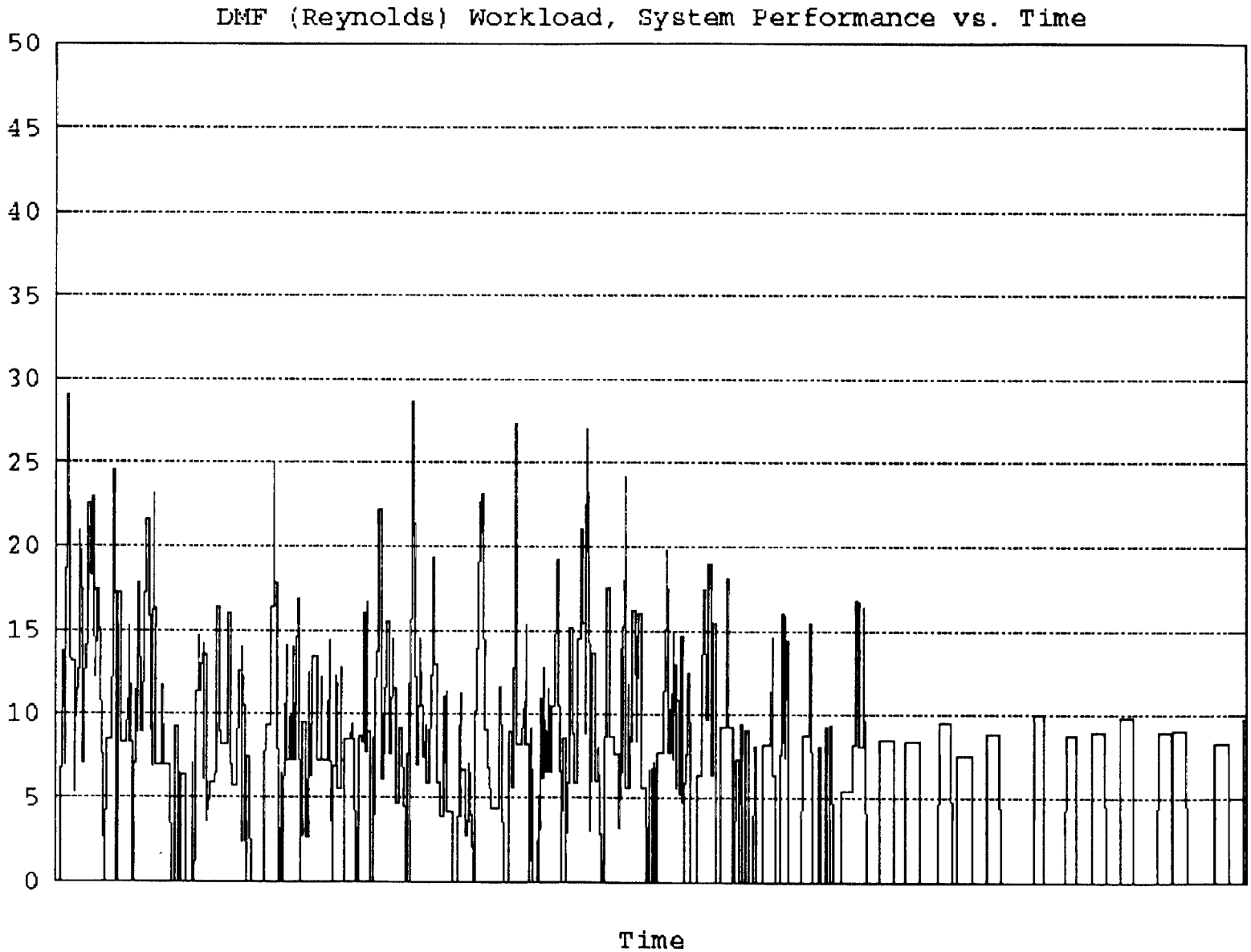
This report would not have been possible without the efforts of quite a number of people. I would like to thank the following people who played some part in data gathering or support with any of the packages:

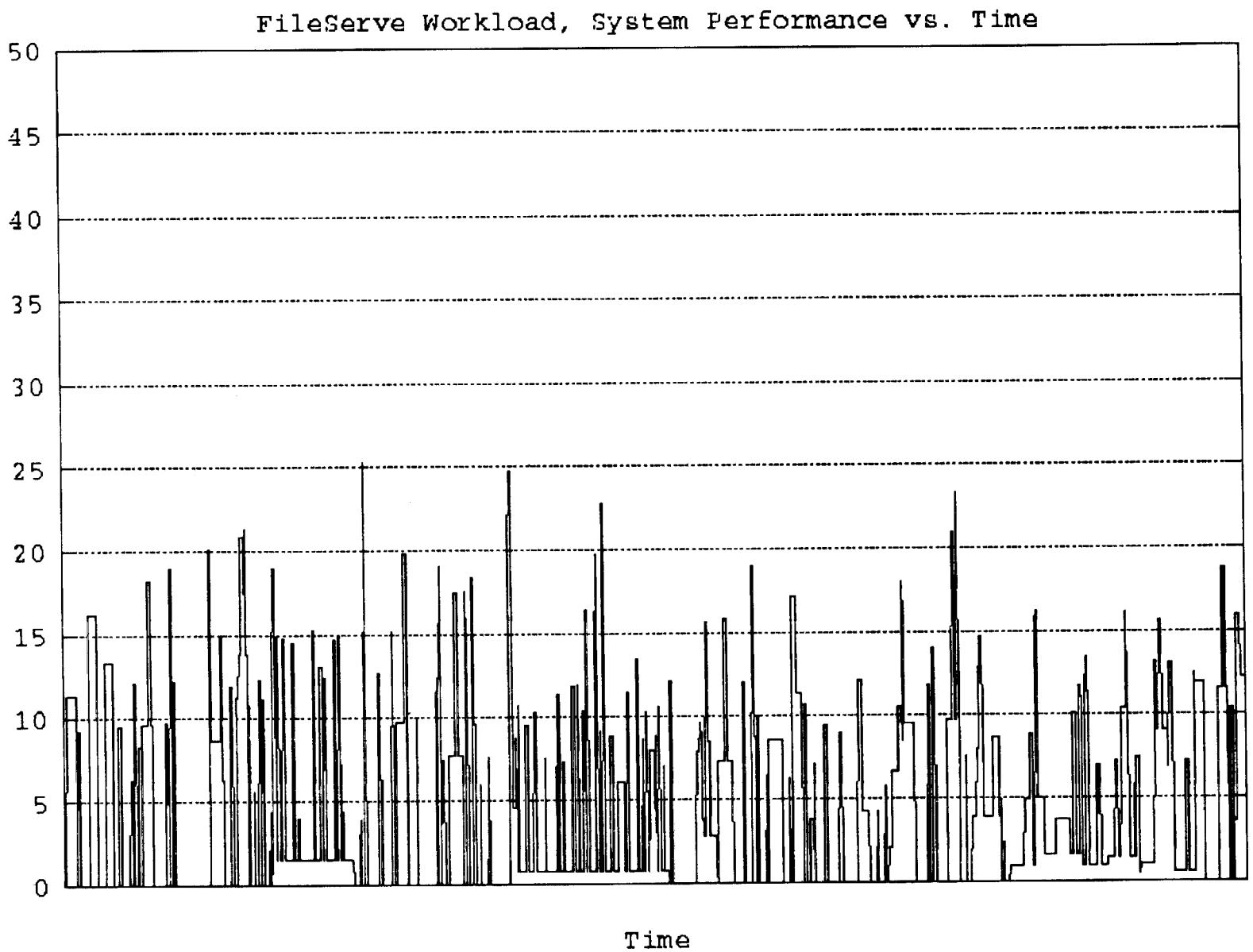
Alfred Nothaft for composing the workload code and running it
Eugene Miya with testing on various packages and platforms
Mike Selway and Eric Powell of E-Systems with FileServ
Larry Spoo and Sharon Bayne of Convex with UniTree
Nick Cardo, Winston Lew and Alan Powers of Sterling with DMF
Ruth Iverson, Alan Poston, Tom Proett, Bill Ross of CSC with NAStore

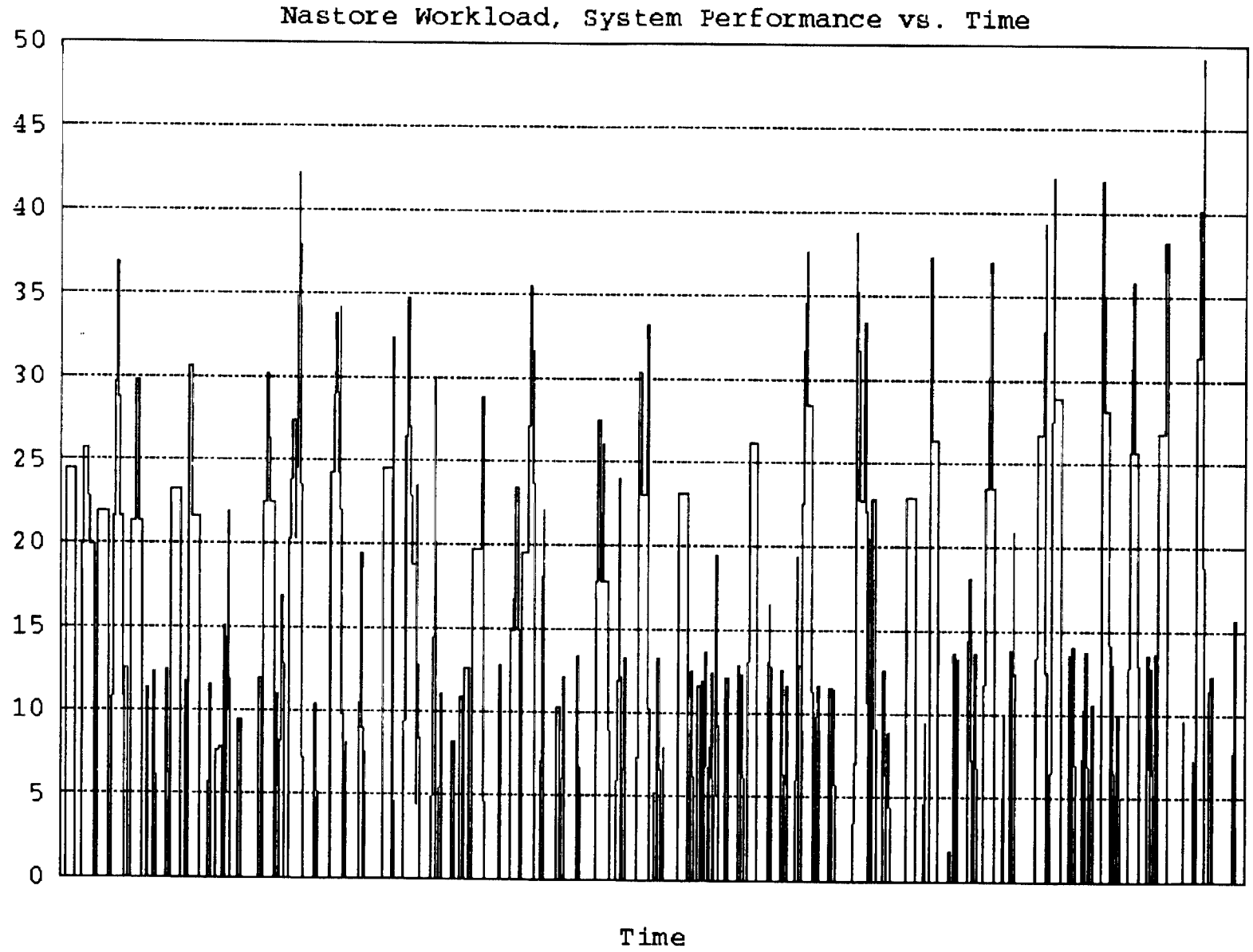
## Contacts

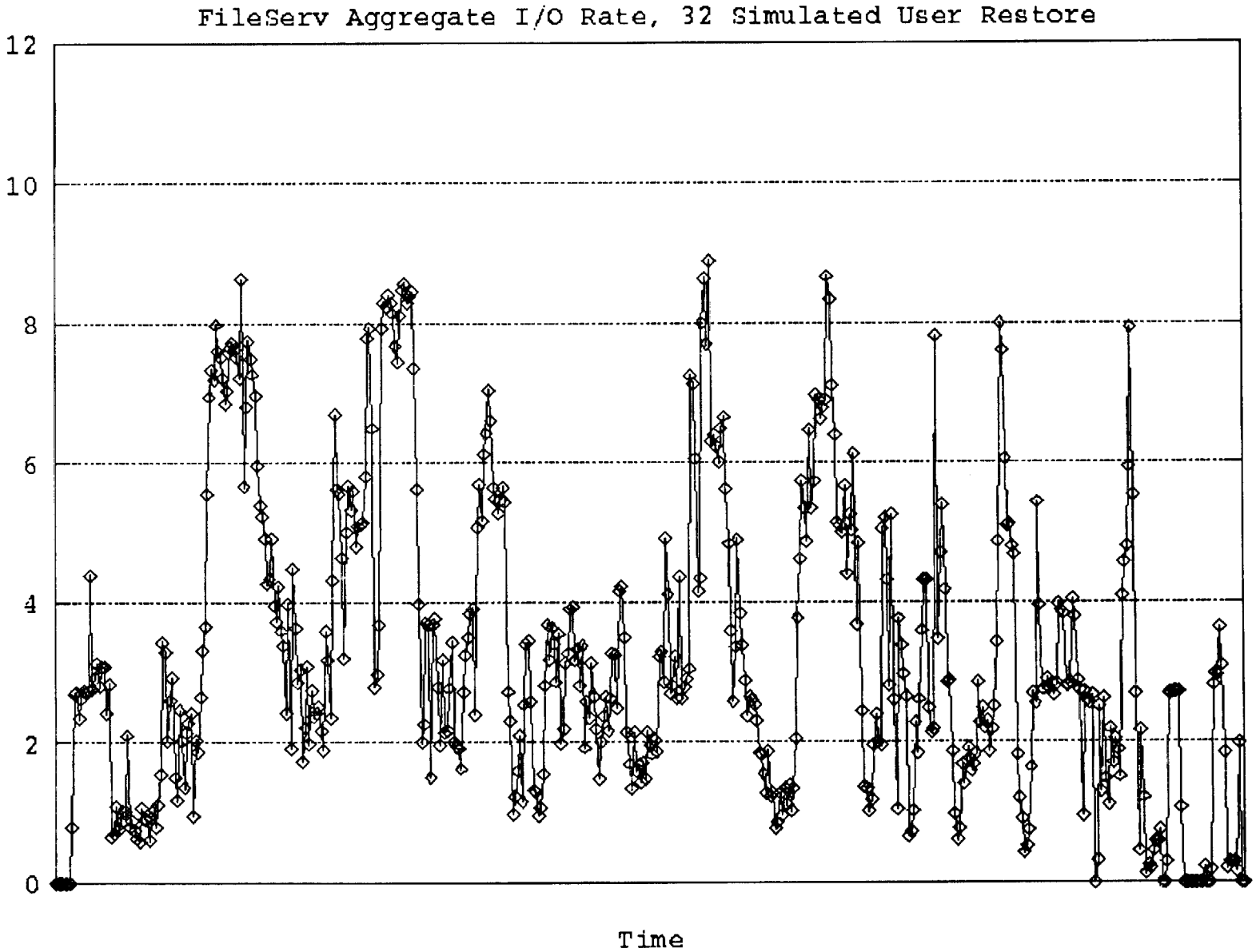
Several individuals are listed below for follow up or additional information.

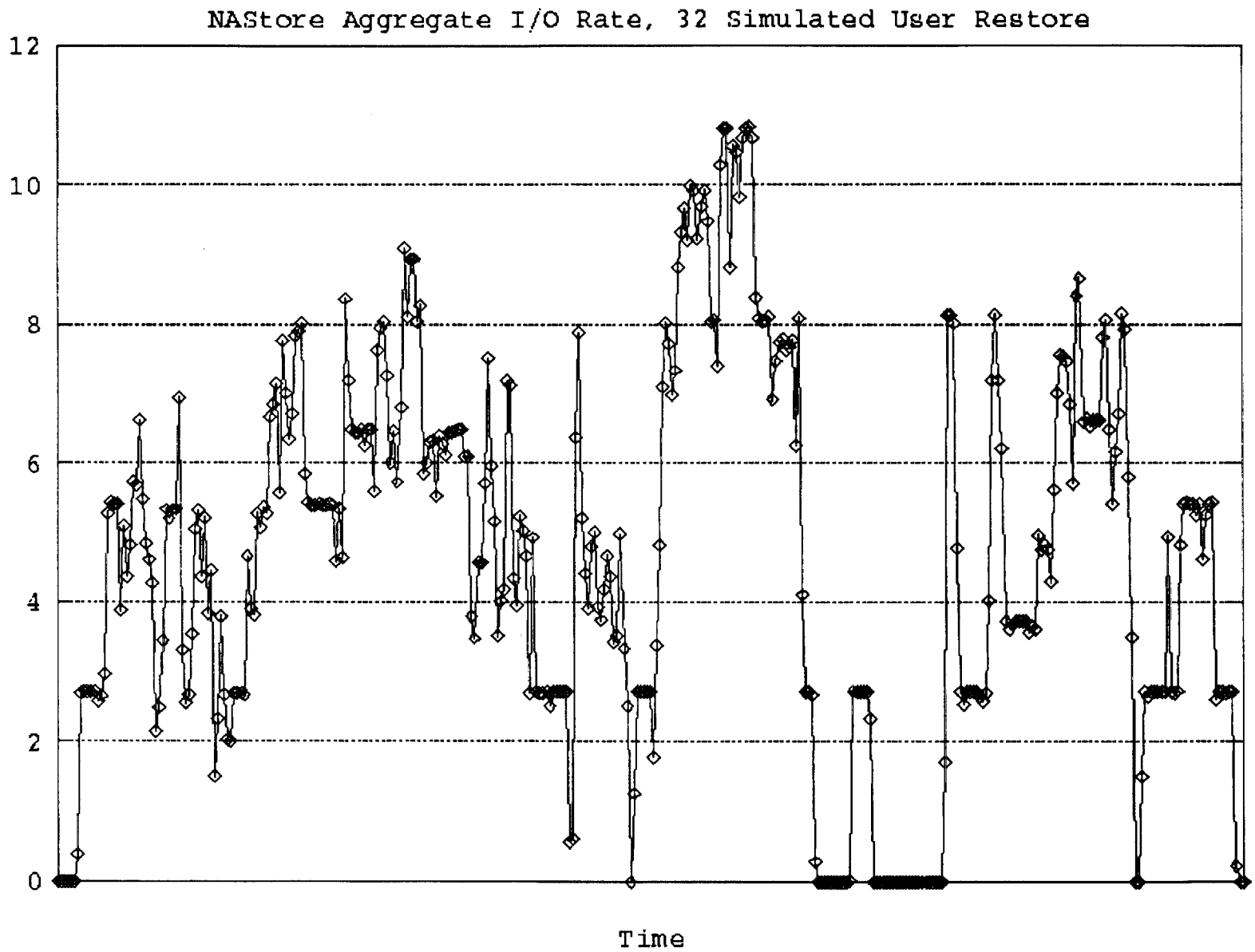
| | | |
|---|---|---|
| Data Migration Facility | Steve Banks<br>Cray Research Inc.<br>(408) 745-6466<br>banks@renaissance.cray.com | |
| FileServ | Steve Frazier<br>E-Systems Corp.<br>(404) 980-6685 | John George<br>Convex Computer Corp.<br>(408) 453-5700<br>jgeorge@convex.com |
| NAStore | John Lekashman<br>NASA Ames Research Center<br>(415) 604-4359<br>lekash@nas.nasa.gov | Alan Poston<br>CSC - NASA Ames<br>(415) 604-4307<br>poston@nas.nasa.gov |
| UniTree | Max Morton<br>Open Vision<br>(510) 426-6452 | John George<br>Convex Computer Corp.<br>(408) 453-5700<br>jgeorge@convex.com |


DMF (Reynolds) Workload, System Performance vs. Time
50
45
40
35
30
25
20
15
10
5
0
MB/sec
Time


FileServe Workload, System Performance vs. Time
50
45
40
35
30
25
20
15
10
5
0
MB/sec
Time


Nastore Workload, System Performance vs. Time
50
45
40
35
30
25
20
15
10
5
0
MB/sec
Time


FileServ Aggregate I/O Rate, 32 Simulated User Restore
12
10
8
6
4
2
0
MB/s
Time


NAStore Aggregate I/O Rate, 32 Simulated User Restore
MB/s
12
10
8
6
4
2
0
Time

N94-33810

# Introduction of the UNIX International Performance Management Work Group

**Henry Newman**
Instrumental Inc.
4500 Park Glen Road, Suite 390
Minneapolis, MN 55044
hsn@instrumental.com

The Performance Management Work Group (PMWG) was first convened four years ago, and its work is now out for public review. Both OSF and USL are implementing this work as are a number of companies. XOPEN and POSIX 1003.7 have agreed to accept the work after the public review has been completed. The following White Paper is an overview of this work, and describes the group's motivations and requirements.

## Performance Management Activities Within UNIX International

**UNIX International**
Waterview Corporate Center, 20 Waterview Boulevard
Parsippany, NJ 07054
Phone: +1 201-263-8400, Fax: +1 201-263-8401

## 1. Introduction

The primary output of the UNIX International Work Group on Performance Measurement is a set of requirements and recommendations to UNIX International and UNIX System Laboratories for the development of standard performance measurement interfaces to the UNIX System. Requirements will be based on the collective, non-vendor specific needs for a standard performance architecture. Currently the lack of this standard causes undue porting and kernel additions by each UNIX System vendor as well as a great variety of approaches to gain the same basic performance insight into the system. Building tools to monitor, display, model, or predict performance or its trends is a frustrating and currently single vendor enterprise. By providing standard data structures, types of performance data gathered, and a common kernel interface to collect this data, the whole UNIX system vendor community along with the UNIX software vendors can develop performance tools which last more than UNIX release and work on multiple UNIX platforms.

Some of the PMWG findings may be in the form of recommendations rather than requirements as a mechanism to stimulate the creation of a common base technology for performance measurement or reporting that is more tool oriented and provides a rallying point rather than a rigid standard imposed on the UNIX system performance measurement, end-user system tuning, capacity planning, and benchmarking areas.

In summary, the requirements and recommendations of the UNIX International Work Group on Performance Measurement can be a driving force behind the advancement of UNIX system performance technology allowing the end-users of UNIX systems to better understand and answer questions such as: what system to buy, how to tune the system, when to upgrade the system, and when to move to a faster system.

## 2. Organizational Statement of UI Performance Management Work Group

It is our desire that the Performance Management Work Group be composed of a balanced team of performance professionals representing the users prospective, as well as the development prospective in the area of Performance Management. We have invited a number of system management as well as development professionals from a number of systems data centers,

large systems manufactures, small systems manufactures, performance analysis organizations, and the US government users community to join the UI Performance Management Work Group. We are pleased to have in attendance at our Work Group meetings, a number of user and development professionals representing a broad cross section of the UNIX industry.

It has also proved to be quite valuable to have in attendance at our UI Work Group meetings, the performance professionals from other organizations outside of the UNIX International community. The experience they bring to the team in the performance management research areas, as well as their desire to develop and adhere to proposed performance management standards, makes the results of our efforts more acceptable throughout the industry.

With this prospective of having developers, users, and a broad representation of UNIX interested professionals attending our UI Performance Management Work Group meetings, the following document is a consensus of our views for making proposals to UNIX International to include Performance Management functions into the UNIX System V Roadmap.

## 3. Statement of UI Performance Management Work Group

The objective of this work group is to examine the area of performance management as it pertains to the UNIX Operating System and to make recommendations on performance management to UNIX International and to UNIX System Laboratories. In addition, this organization will also exchange information and ideas regarding performance management, with other related groups in the UNIX industry including, but not limited to, the IEEE Posix 1003.7 Committees, the Open Software Foundation, and X/Open. In particular, our results shall be made available to these organizations.

### 3.1 Scope

This Performance Management Work Group will be concerned with defining requirements and standards for the collection, presentation and distribution of performance data in large-scale distributed systems. Here, "performance data" is defined to include:

1. Interval or sampled data describing hardware and software resource usage or times, either globally or by some logical entity

2. Count data representing system or applications queue lengths, events, and system resource states

3. Data representing execution traces of processors

4. Data notifying of events occurring at a system, subsystem, or application levels

A layered model of function and interfaces for acquisition and use of such data is shown in Figure 1 to further assist in the delineation of the scope of concerns for this Performance Management Work Group.

- *Measurement Application Layer*

The uppermost level of the model (layer 4) contains the application primitives and tools used to present currently captured and archival performance data to the end-user (or potentially, to an automated stand-in). These application implementations will be called Measurement Application Programs (MAPs).

- *Data Services Layer*

This level of the model (layer 3) is responsible for data simulation, archival data storage, management or services and resources required for distributed measurement access and

control, for measurement requesting, and for data transformations required for analysis and data recording.

- *Measurement Control Layer*

This layer of the model (layer 2) is responsible for managing the capture of data, including the synchronization, and for providing any necessary buffer or queue management for data assembled by the data capture mechanism. A portion of this layer and the next lower (data capture) layer may be functionally replicated in a subsystem or application for synchronized data collection from such entities.

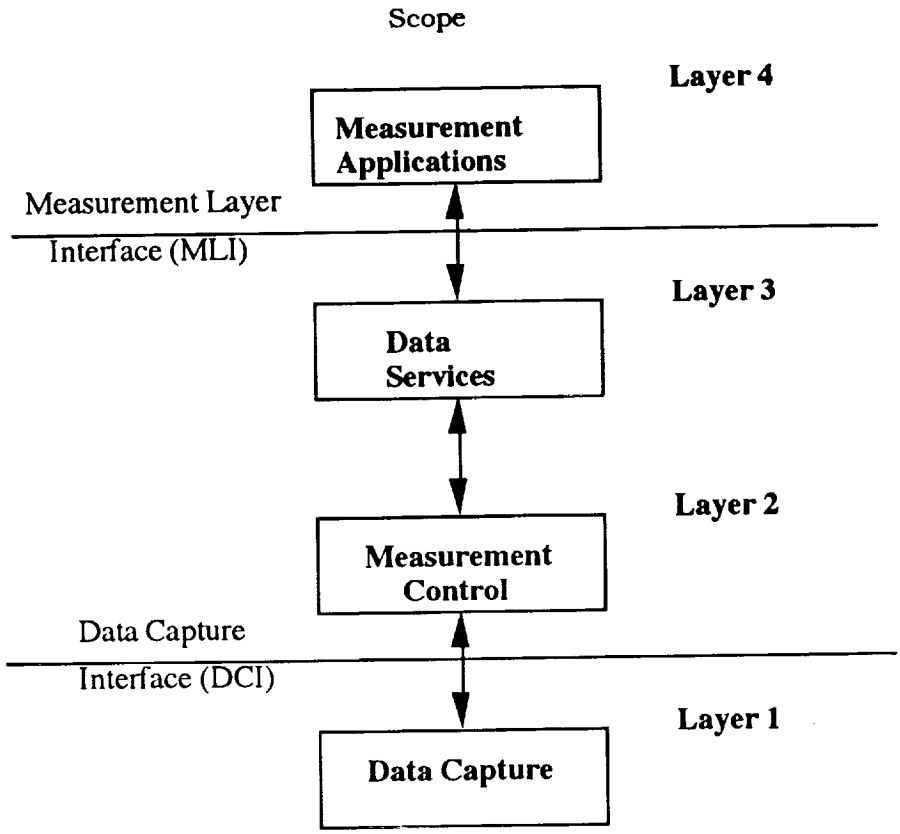



Figure 1. A Measurement Model for UNIX-Based Systems

- Data Capture Layer

This layer of the model (layer 1) is responsible for capture of data manifested in system or hardware counters or structures. Data is considered captured when it exists assembled into data structures of predefined class and type in storage controlled by services contained in the measurement model.

- Interfaces Defined by the Measurement Model

The interfaces between the layers are defined in a way that frees an upper layer from concern about how services are provided below it.

- The model provides a Measurement Layer Interface (MLI) for requesting measurement services. The MLI enables MAPs to be implemented without knowledge of the underlying measurement procedures.

## 5.1 Performance Management Systems - Technology

### 5.1.1 Technology Overview - Large System Facilities

Currently, the most developed performance and accounting data management facilities for large-scale systems are to be found in proprietary operating systems such as IBM's MVS and DEC's VMS on its VAX computers.

In general, the modes of capturing data for either presentation as reports or subsequent use by other tools includes:

- Sampled Data - Data which is measured by repetitive capture (at the sampling rate) and presumably accumulated in a counter.
- Interval Data - Data which represents the *incremental* activity within a certain time interval.
- Event Data - Data which provides notification of the occurrence of a particular state within a subsystem.
- Trace Data - Data which captures a succession of subsystem states, usually in substantial detail.

IBM's MVS provides selectable recording of accounting and performance data through SMF (System Management FAcilities) extended by high resolution performance data through RMF (Resource Management Facility). Other MVS facilities provide for the acquisition of trace data. Since these sources have well-defined data contents and formats, third parties have created management tools (especially for SMF/RMF data) that provide extensive reporting capabilities for accounting, security functions, and performance analysis (e.g. MICS, JARS, TSO/MON). Some modeling tools, such as BEST/I and CMF/MODEL make direct use of these same data sources for model definition and validation. Lastly, data manipulation and statistical analysis packages such as SAS have provided a basis for both "home-grown" and vendor-supplied tools, again based on these same data sources.

## Performance Management Systems - Technology

DEC provides a set of tools for VAX/VMS, each using its own data collection mechanism and maintaining separate logs for each VAXcluster node. These DEC products include:

MONITOR: This tool provides on-line reporting of system-wide information for a running system. Allows viewing of combined usage from VAXclusters on a single terminal.

ACCOUNTING: As part of VMS, provides basic accounting information and optional information on user jobs or processes, on images or programs executed, and on batch and print jobs. An included utility produces reports.

SPM: The SOFTWARE PERFORMANCE MONITOR provides more extensive data collection and reporting and includes an Event Trace Facility which permits the triggering of custom written trace code capturing data from both the OS, the Record Management Services, device drivers, or applications. SPM can maintain a historical database of information over multiple nodes. Both system-wide and per-process statistics are supported. SPM software does not provide synchronization among nodes of a VAXcluster.

VPA: VAXPerformance Advisor - Collects and analyzes system-wide performance data using a knowledge base of rules and thresholds. VPA synchronizes clocks among nodes in a VAXcluster (to within 0.5 sec).

It is important to recognize the benefits that these and similar facilities offer, however, it is not our intention to replicate either the specific methods or data content.

### 5.1.2. Accessing Performance Data in A Vendor-Independent Way

This Performance Management Work Group believes that accessing of performance and accounting data through well-defined, standard and non-proprietary interfaces is essential for the creation and wide availability of a toolset that is suitable for large-scale UNIX-based systems management. Such interfaces and their related functions will promote:

- Mutual insulation of client measurement applications from implementation details in the measurement provider or its sources. This facilitates version independence and ease of measurement application maintenance which benefits system vendors, software creators, and ultimately, the system owner.

- Portability of tools. Applications built to a standard, vendor-independent interface can function on various implementations. Well-designed performance and accounting applications can include awareness of both common data and that specialized to a particular architecture.

- Data capture efficiency. Requesting of measurements through a common measurement interface makes it possible to service requests for the same data from multiple measurement applications by distributing data obtained from a single data capture.

- Extensibility of instrumentation. A standard interface for data capture make it possible to add instrumentation in a well-defined and thus more easily maintained way.

- Distributed control of measurement and access to measurement data, even across heterogeneous hardware architectures. Such distributed control and access facilities should also provide the means for achieving a level of coordination and synchronization between dispersed measurements sufficient to make possible a coherent logical view of the data.

- Increased third party applications development. Portability of tools encourages third party interest due to the increased size of the potential market.

## 5.2. Performance Management Tools

Performance management covers a wide area of related activities and can be grouped into the following three task categories:

- The first category of tasks is related to capacity planning and quality of services as specified in the Service Level Agreements (SLA).

- The second category embraces maintenance, tuning and elimination of bottle-necks, and deals with planning on a weekly or a monthly scale.

- The third category consists of ad hoc operations in order to keep the systems alive and to solve user problems.

The performance management tools provide for configuration planning , capacity planning, on-line performance measurement/monitoring, and expert systems to analyze, interpret and to predict computer systems performance. It is important to note that these performance tools require accurate data in terms of resource and system utilization and this white paper deals with descriptions for the performance data gathering facilities. An example use of the performance management tools in traditional data processing is illustrated in the figures 2 and 3. Figure 2 shows the expected and actual usage in a specified peak period (e.g. 9:00 A.M. -

11:00 A.M.) of application packages in a given production data (computer) center. Figure 3 shows the detailed usage of DBMS commands. Based on the information presented in Figures 2 and 3, the data (computer) center management can easily identify the top running applications and users and, adjust the computer systems resources (CPUs, Memory, Disks, Tuning etc.)

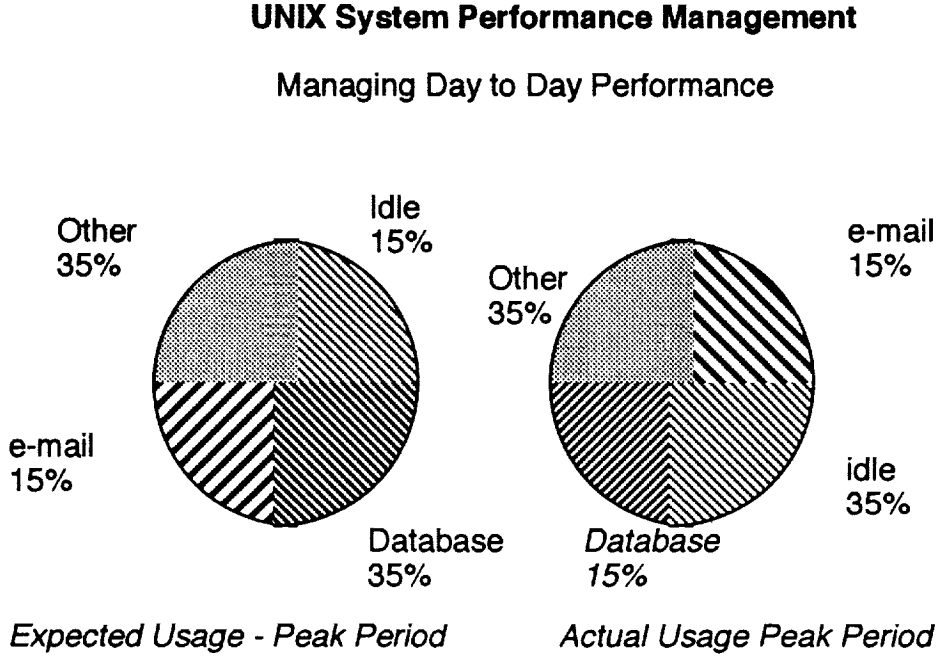



Figure 2. UNIX Performance Management Tools/CPU Usage Comparison

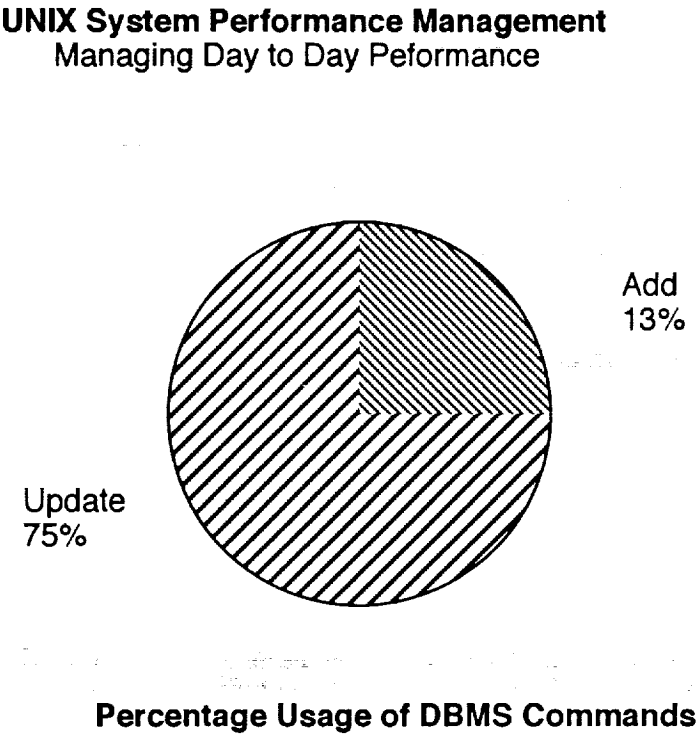



Figure 3. UNIX Performance Management Tools/DBMS Commands Usage

# 6. Performance of systems: A users perspective

## 6.1. Overview

Performance methodologies have evolved considerably over the last two decades from an analysis of system utilization, to a degradation analysis of manageable subcomponents of end-user response time (or batch process compete time). The primary focus of the performance analyst has shifted from the resource to the workload. This is sometimes called workload analysis. After workload analysis has been completed, and the critical resource(s) have been identified, the performance analysts secondary focus shifts to dividing the time spent at the resource(s) into subcomponents.

A critical requirement for subcomponents analysis of end-user response time is an architected definition of what constitutes the beginning and ending of a transaction. In the UNIX environment this is not the beginning and ending of a process but must be defined from an end-user perspective. For management reporting and Service Level Agreements it is imperative that response distribution buckets be maintained so percentiles may be reported. This is because response times do not fall into statistically 'normal' distributions making average times difficult to understand.

## 6.2 Granularity

The required granularity of the subcomponents of response time is dependent upon the level of analysis being done.

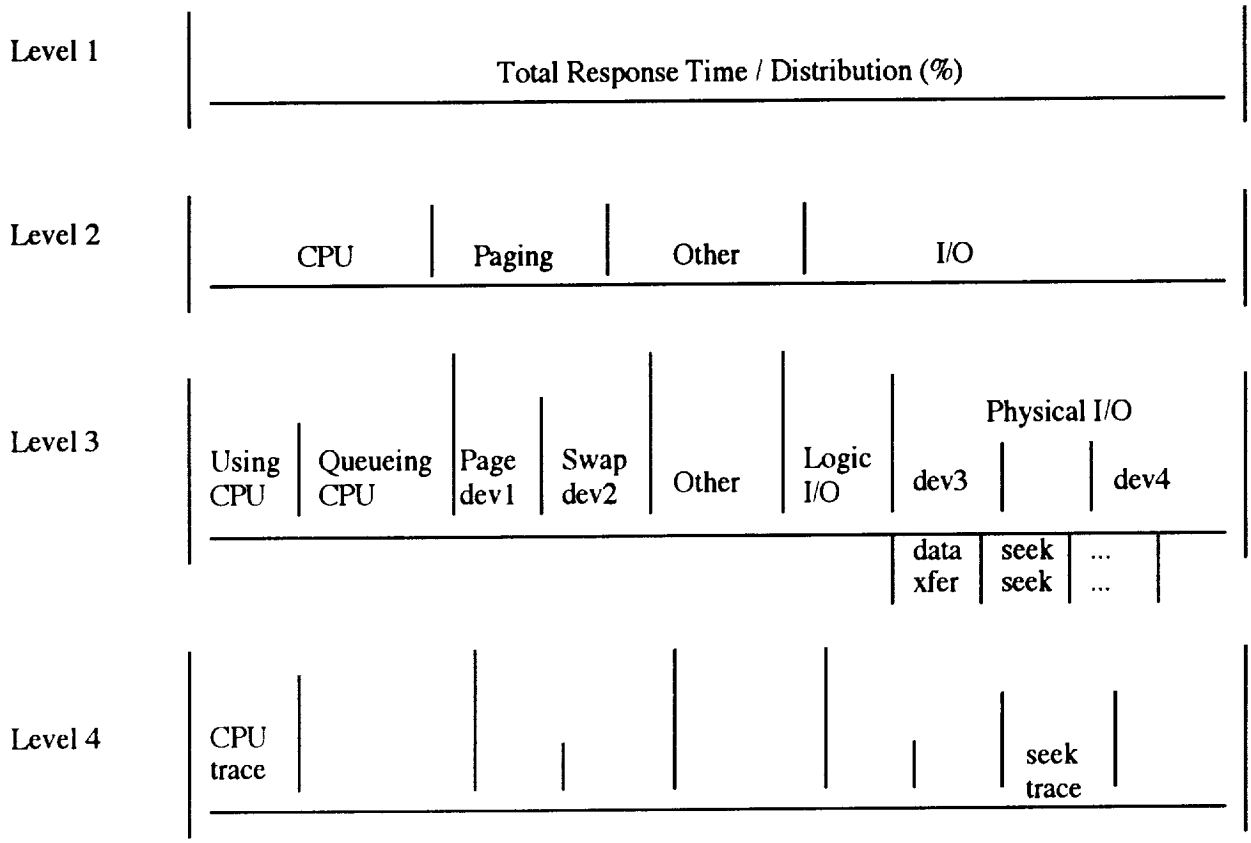



Figure 4. Workload Analysis By Level

At the highest level (what we will call level 1) the total response time or response distribution is all that is required to determine if further analysis is necessary. This information is best gathered by event driven mechanisms.

At the next level (level 2) it may be sufficient to see the delays for CPU, I/O, Paging and 'Other' divided out. These times could include both using and queuing times for each resource. This information is best gathered by high priority state sampling techniques.

At the next level (level 3) each component can then be subdivided into its component parts. For example I/O can be divided into logical and physical. Physical I/O can then be split, by device, into it's measurable subcomponents. This information may be gathered by either high priority state sampling techniques and/or event driven mechanisms.

At the lowest level (level 4) detailed traces can be used to further divide a subcomponent into smaller manageable parts.

Measurement controls should be flexible enough to allow monitoring of individual end-users and groups of end-users by transaction type. Information should be available for both real-time and historical analysis.

## 7. Summary

In this paper we presented the planned direction of the UNIX International Performance Management Work Group. This group consists of concerned system developers and users who have organized to synthesize recommendations for standard UNIX performance management subsystem interfaces and architectures. The purpose of these recommendations is to provide a core set of performance management functions and these functions can be used to build tools by hardware system developers, vertical application software developers, and performance application software developers.

**Published by:**

UNIX International
Waterview Corporate Center
20 Waterview Boulevard
Parsippany, NJ 07054

for further information, contact:
Vice President of Marketing

Phone: +1 201-263-8400
Fax: +1 201-263-8401

International Offices:

UNIX International
Asian/Pacific Office
Shinei Bldg. IF
Kameido
Koto-ku, Tokyo 136
JAPAN

Phone: +81 3-3636-1122
Fax: +81 3-3636-1121

UNIX International
European Office
25, Avenue de Beaulieu
1160 Brussels
BELGIUM

Phone: +32 2-672-3700
Fax: +32 2-672-4415

UNIX International
Pacific Basin Office
Cintech II
75 Science Park Drive
Singapore Science Park
Singapore 0511
SINGAPORE

Phone: +65 776-0313
Fax: +65 776-0421

# Performance Measurements and Operational Characteristics of the Storage Tek ACS 4400 Tape Library with the Cray Y-MP EL

**Gary Hull**
Hughes STX Corporation
4400 Forbes Blvd.
Lanham, MD 20706
ghull@flyfish.stx.com

**Sanjay Ranade**
Infotech SA Inc,
12303 Sandy Point Court
Silver Spring, MD 20904
infotech@access.digex.com

## Abstract

With over 5000 units sold, the Storage Tek Automated Cartridge System (ACS) 4400 tape library is currently the most popular large automated tape library. Based on 3480/90 tape technology, the library is used as the migration device ("nearline" storage) in high-performance mass storage systems. In its maximum configuration, one ACS 4400 tape library houses sixteen 3480/3490 tape drives and is capable of holding approximately 6000 cartridge tapes. The maximum storage capacity of one library using 3480 tapes is 1.2 TB and the advertised aggregate I/O rate is about 24 MB/s.

This paper reports on an extensive set of tests designed to accurately assess the performance capabilities and operational characteristics of one STK ACS 4400 tape library holding approximately 5200 cartridge tapes and configured with eight 3480 tape drives. A Cray Y-MP EL2-256 was configured as its host machine. More than 40,000 tape jobs were run in a variety of conditions to gather data in the areas of channel speed characteristics, robotics motion, timed tape mounts and timed tape reads and writes.

## Background

The major objectives of this study, part of the High-Performance Computing and Communications Project (HPCC), were as follows:

1. To establish a set of tape I/O performance measurements associated with the current Y-MP EL hardware configuration for comparison with future technology. The STK ACS 4400 Tape Library is the first magnetic tape library system available to the project.

2. To utilize the Cray Y-MP EL as a research tool dedicated to I/O performance measurements.

3. To apply the results of this research to the user community.

## Test Environment

This section discusses the computer, disks, tapes, library hardware, software and the system configuration used for the STK ACS 4400 tape library performance tests.

*a. Hardware Configuration*

The hardware configuration for these tests is shown in Figure 1. The Cray Y-MP EL is a two-processor machine configured with 256 MBytes of central memory. Connected to the main

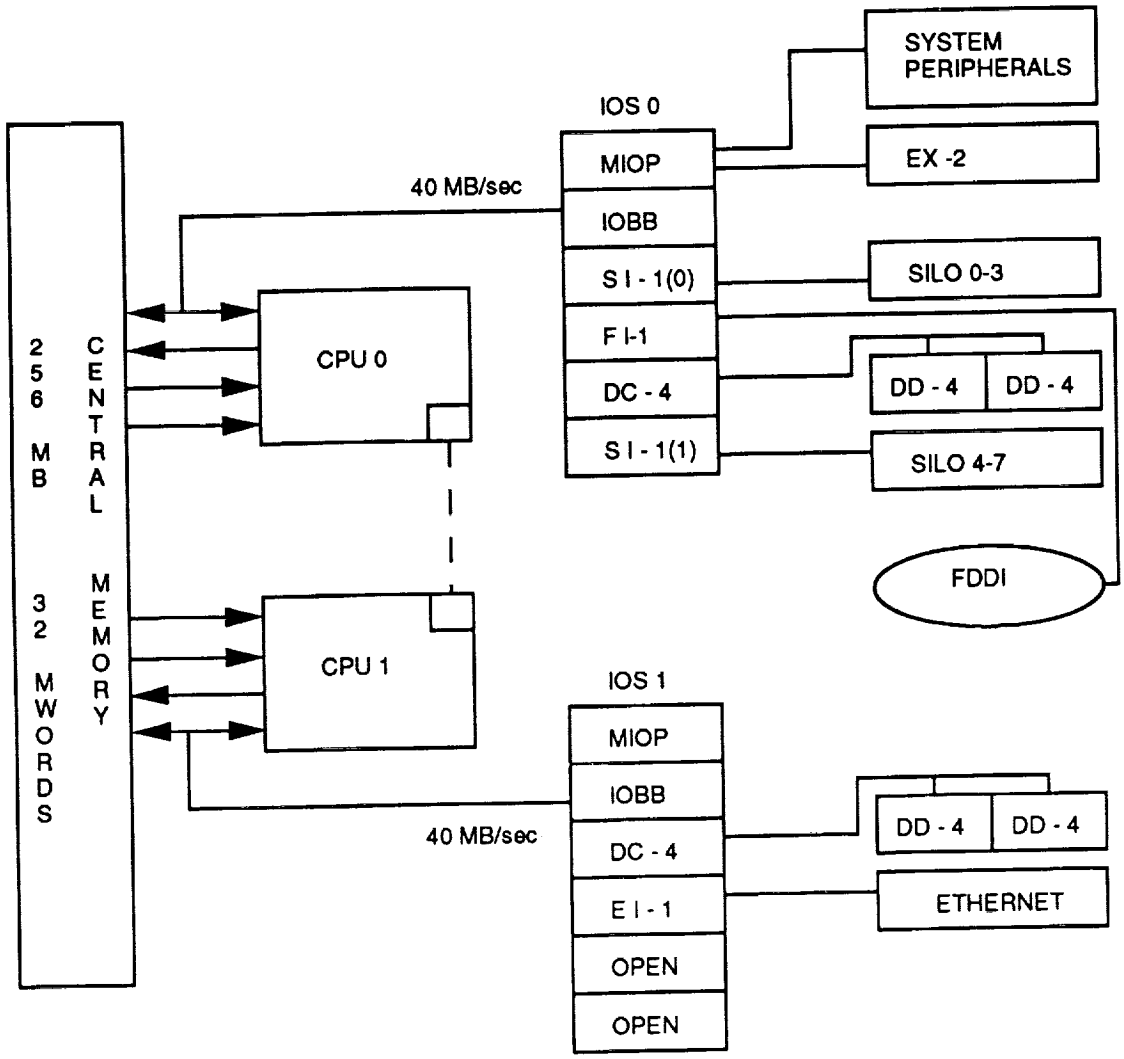



**Figure 1. NASA/GSFC Y-MP EL/2-256**

memory via two 40 MByte/sec channels are two Input/Output Buffer Boards (IOBB). Connected to the disk controller DC-4 are four DD-4 disks, each with 2.7 GB formatted capacity. The specified peak transfer rate for the DD-4 is 7.5 MByte/sec. These disks are distributed across two controllers and each controller is connected to its own Input -Output Subsystem (IOS).

The tape drives in the STK ACS 4400 library are connected to the CRAY Y-MP EL by two Ciprico SCSI interfaces, each capable of sustaining an advertised transfer rate of 4.5 MByte/s. Both 4781controllers are connected to IOS 0. Each Ciprico controller manages four STK tape drives. The data buffer size for the tape controller is large, but nevertheless limited by the 128K maximum allowed by the IOBB. (Note that the data transfer rate obtainable with the STK ACS 4400 SCSI tape drives is dependent on the specific SCSI interface used to connect the host to the drive. The Ciprico SCSI interface does not have the SCSI incompatibility problem identified with other SCSI interfaces).

The Cray Y-MP EL shared the same Ethernet rib as the STK Sun 330 server and the two systems communicated with each other using standard TCP/IP protocol.

The FDDI connection links the Cray Y-MP EL to the NASA/Goddard campus-wide network.

*b. Software Configuration*

Release level 6.1.6 of the UNICOS operating system was run on the Cray Y-MP EL and included Cray proprietary software subsystems: Cray tpdaemon, stknet and Data Migration Facility (DMF). Stknet is the software interface which communicates directly with the ACSLS Client Server Interface (CSI) running on the STK Sun 330 server. The ACSLS server software was run at level 3.0. Tape requests from the Cray Y-MP EL are initiated and managed by the Cray tpdaemon, forwarded to stknet and processed as level 3 TCP/IP packets. These packets are then sent to CSI on the STK Sun 330 server using standard TCP/IP protocol (see Figure 2).

The software subsystem DMF manages on-line mass storage space and implements data retrieval and storage to and from the tape library. DMF requests are initiated and managed by the dmdaemon and utilize the same transfer path described above.

*c. Test Methodology*

More than one hundred programs written in C and Bourne shell scripts were designed and implemented to gather systems and user performance characteristics of the Cray Y-MP EL using the STK 4400 tape library. Emphasis was placed on learning what the general user might experience so accurate predictions could be developed. Snapshots of the Cray Y-MP EL resource allocation characteristics were captured during individual components of each timed test. The data for CPU time, user time and system idle time were used to predict tape I/O resource requirements.

Actual mount times, data write and data read rates were timed under various system load conditions and block sizes. Mount time is defined as the time from the initial mount request by the Cray Y-MP EL to when the STK Sun 330 server replies , acknowledging that the requested tape volume is ready for a write or read operation. Mount time was calculated for and expressed as follows:

The Cray UNICOS tpmnt command with the -w option was used to control command processing during timed mount testing.

1. Mount time in seconds.

2. Mount rate: total number of mounts per hour.

A 51 MB data file was used for movement of data during the timed write and read operations. The Cray UNICOS tpmnt command with the -w option was needed to define the beginning of a

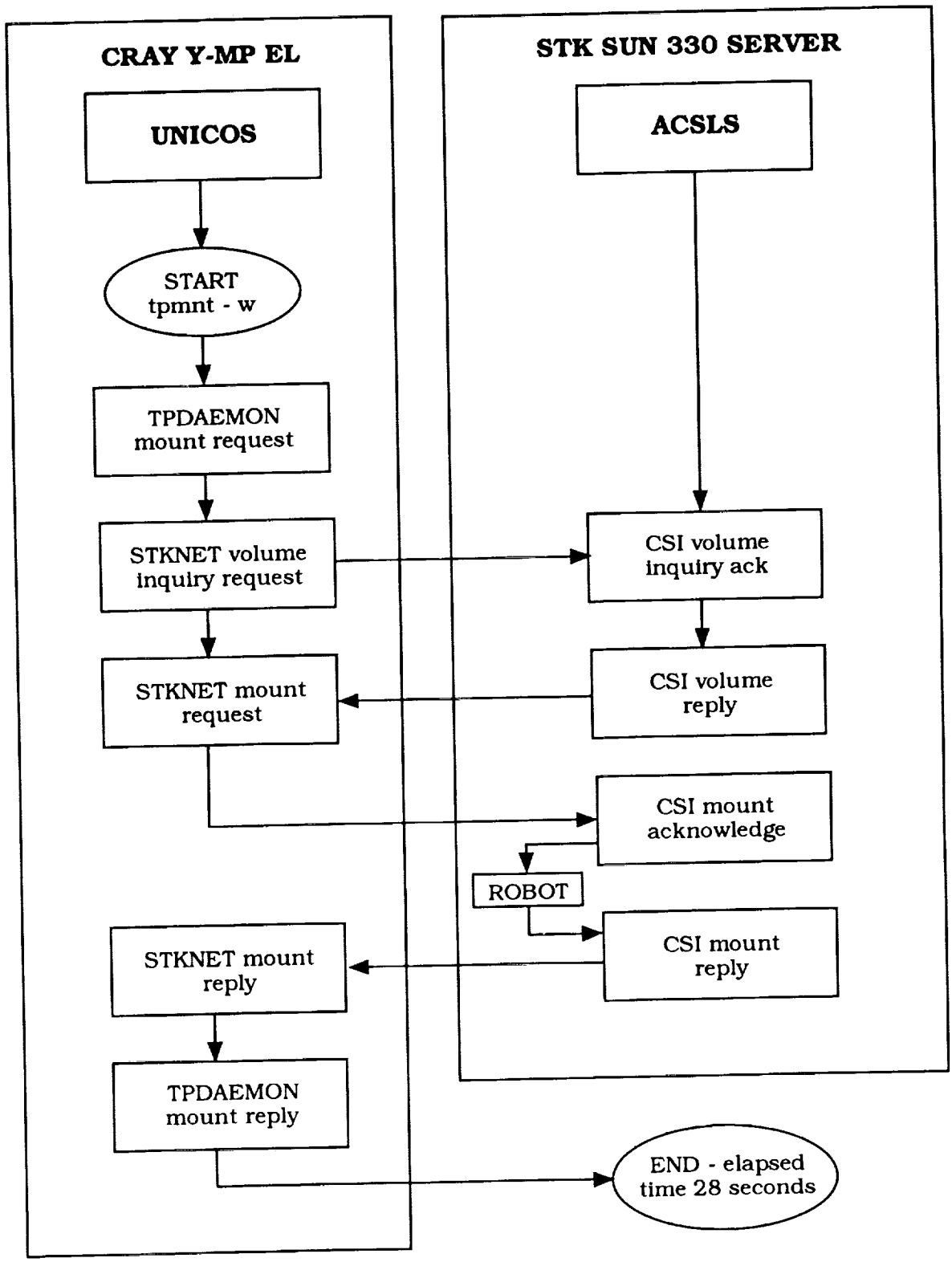



**Figure 2. CRAY-STK Sun Server Communications Path (Example - Mount Request)**

write/read operation. It was used in conjunction with the Cray UNICOS gettime system command issued before and after the transfer operation to capture actual data transfer time. Transfer rate was calculated and expressed in terms of MB/s.

The transfer rate to tape for a large data file using DMF was also timed. The DMF command, dmput was used with the Cray UNICOS timex command to measure transfer rate for DMF of a 207 MB data file. This measure was expressed in MB/s and included both tape rewind and unload time.

Potential maximum SCSI 1 channel speed was also measured for our configuration. A 207 MB data file was written to tape using the Cray UNICOS tar command with the -b option. The block size specified with the -b option was 128 and corresponds to a 64K byte block size. The tape was read using tar with -b equal to 128 and with the -t option which instructs tar to read only the tape label, but also forces the read process to go to the EOT marker. The time in which tar accomplished this was defined as the potential maximum channel speed for our configuration and was expressed in MB/s.

## Test Results

This section summarizes the various test results recorded in this study. Mount times, disk -to -tape read/write transfer rates, tape -to- disk read/write transfer rates, DMF transfer rates, channel speed transfer rates and robotics observations are presented. Average transfer rates are also computed for these functions.

*a. Mount Times*

It took the Cray Y-MP EL/STK 4400 tape subsystem an average of 52.15 seconds to mount one tape. The Cray Y-MP EL averaged approximately 40-50% user activity during the periods in which the timed mount tests were run. Figures 3, 4 and 5 further break down the results of the timed mount tests by tape device and the controller path to which each was configured. These figures compare mount times for the first and last tape device on each controller, as well as the last tape device on controller 0 to the first tape device on controller 1. Although not significant, the first device configured to each controller averaged slightly lower mount times than the last device configured to the same controller. The average mount times appear to be more a function of position within the controller rather than to the port the controller is configured. The average mount times by controller and tape device are as follows:

| | tape device | mount time |
|---|---|---|
| controller 0 - | silo00 | 51.25 seconds |
| | silo03 | 52.47 seconds |
| controller 1 | silo04 | 51.96 seconds |
| | silo07 | 52.93 seconds |

*b. Disk -to -Tape (Tape write operation)*

The data transfer rates for disk reads/tape writes are shown in Figures 6 and 7. The block sizes tested are 4KB, 8KB, 16KB, 32KB, 64KB and 128KB. Transparent buffered I/O, a technique which produces tapes with a block size equal to the maximum block size specified by the -b option of the UNICOS tpmnt command was used to move 51 MB's of data from disk to tape in all tests. The last block may vary in size from 1B to the maximum block size. Note that the three smaller block sizes (4KB, 8KB and 16KB) exhibited average transfer rates of less than 1 MB/s while the three larger block sizes (32KB, 64KB and 128KB) exceeded average transfer rates of 1 MB/s. Movement of data to tape in our environment appears to be fastest when using a block size of 64KB. Running these tests consistently caused the test job to use 12% additional system resources regardless of block size. Average tape write operation transfer rates by block size are as follows:

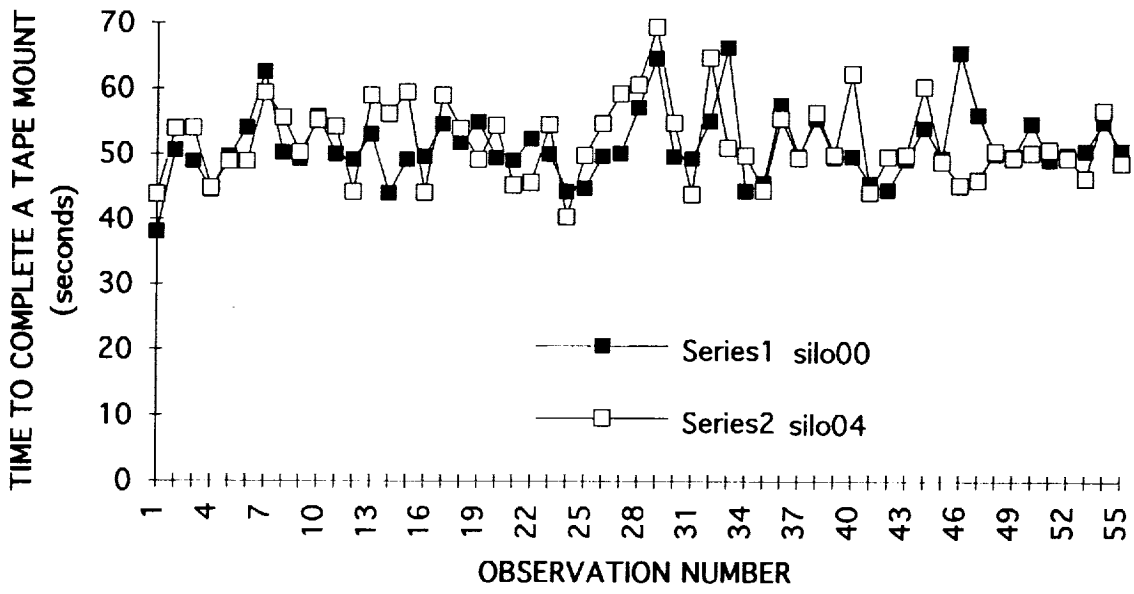



**Figure 3. Tape Mounts Silo00 and Silo03 (Timed Mounts for 1 Hour)**

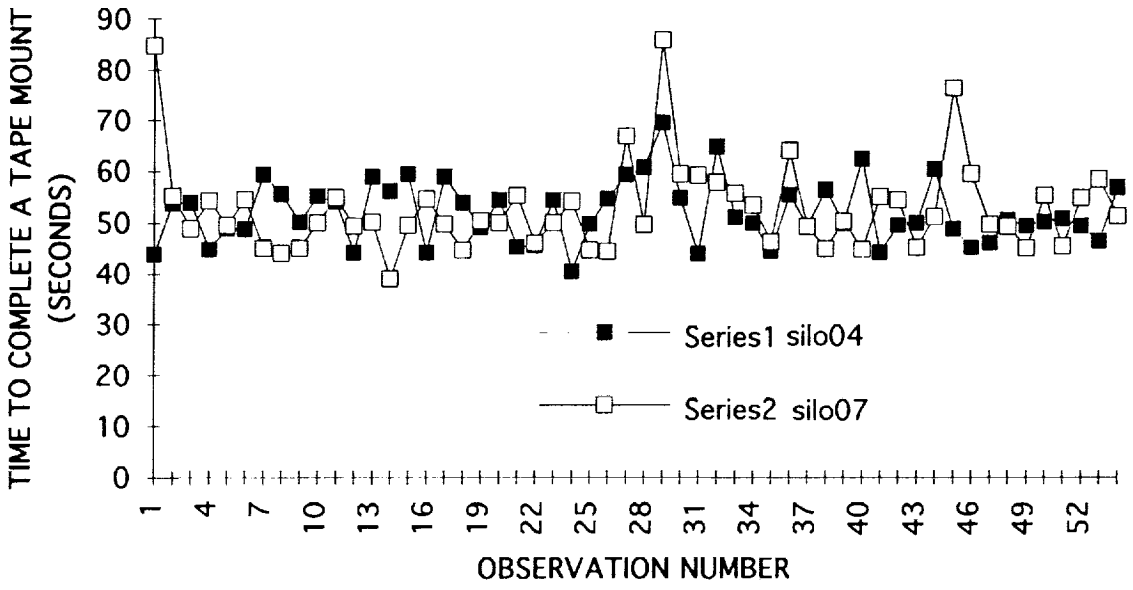



**Figure 4. Tape Mount Timings for Silo04 and Silo07 (Average 1 Hour Timed Mounts)**

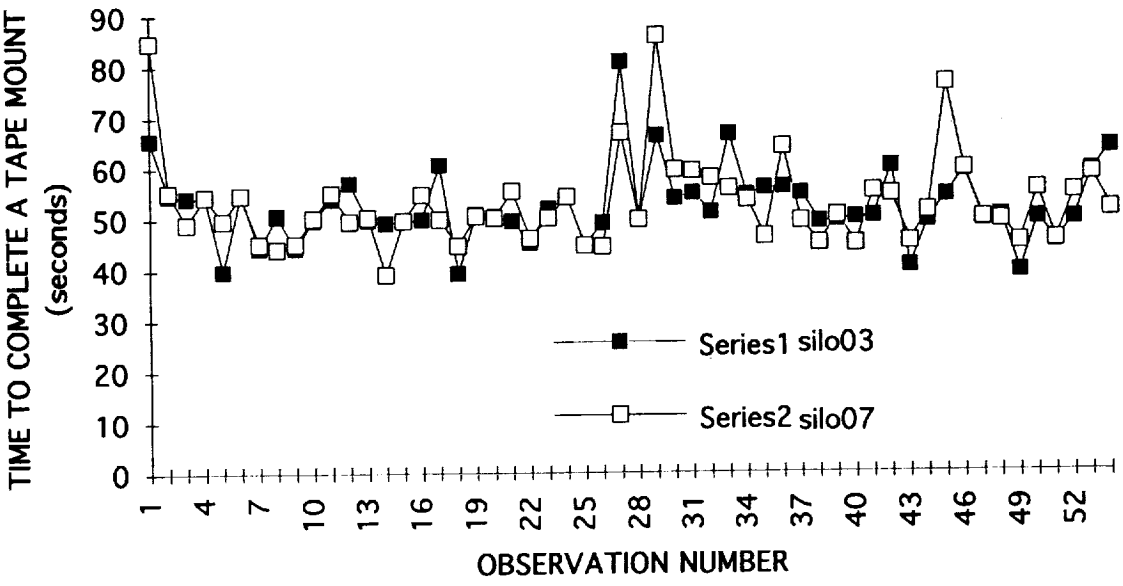



**Figure 5. Tape Mount Timings for Silo03 and Silo07 (Averages for 1 Hour Timed Mounts)**

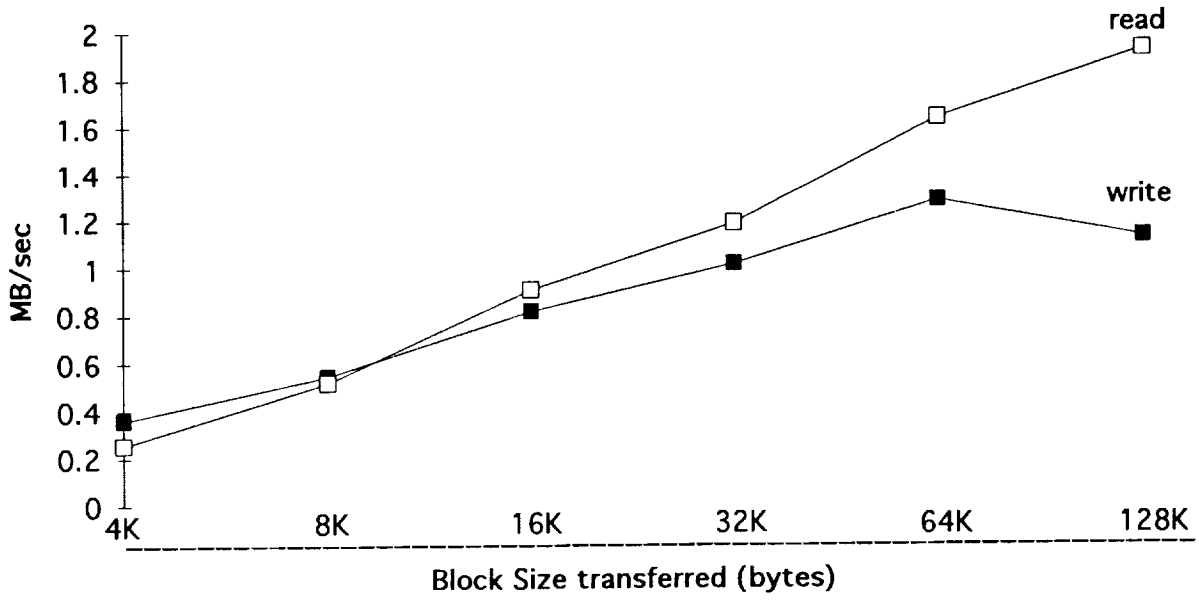



**Figure 6. Tape I/O Write/Read Transfer Rates**

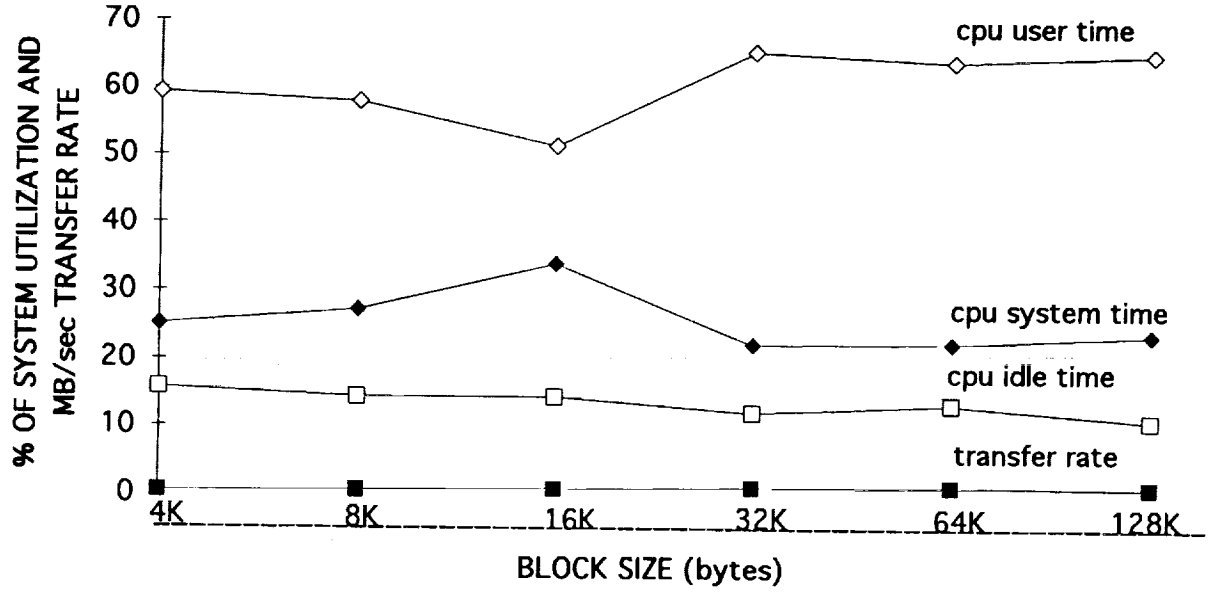



**Figure 7. Tape I/O Write and Systems Utilization Time**

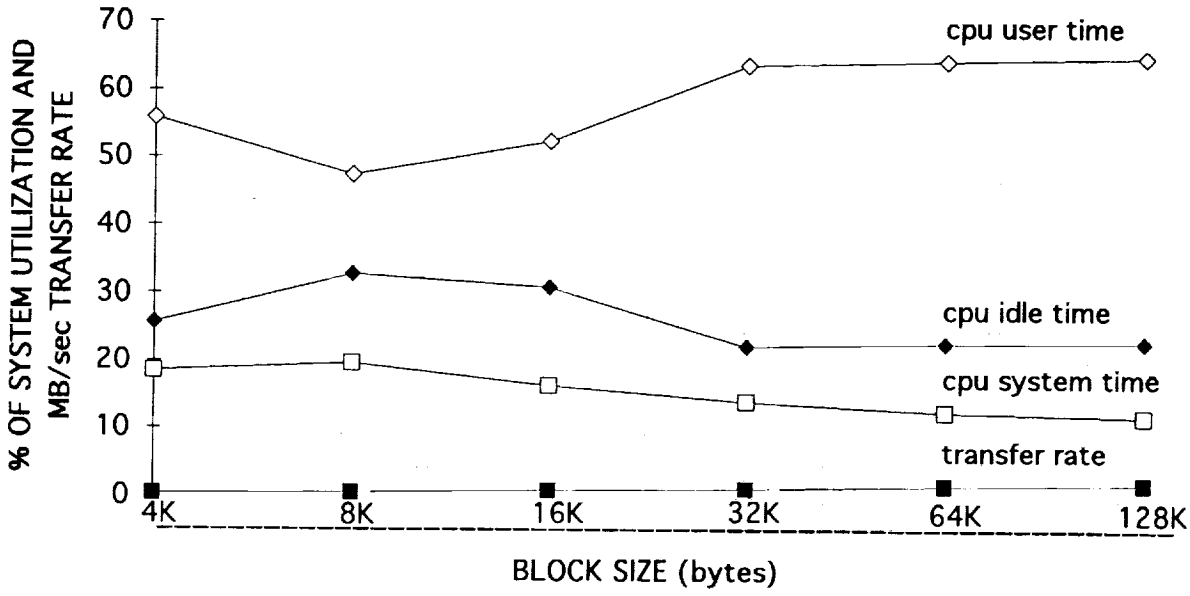



**Figure 8. Tape I/O Read and System Utilization Time**

| Block Size | Transfer Rate |
|---|---|
| 4KB | 0.36 MB/s |
| 8KB | 0.54 MB/s |
| 16KB | 0.91 MB/s |
| 32KB | 1.01 MB/s |
| 64KB | 1.28 MB/s |
| 128KB | 1.13 MB/s |

*c. Tape -to -Disk (tape read operations)*

The data transfer rates for tape reads/disk writes are shown in Figures 6 and 8. The same block sizes used to write a tape were used to read the tape. The technique of transparent buffered I/O was also applied. The three smaller block sizes consistently averaged less than 1 MB/s transfer rate and beginning with 32KB block sizes, the average transfer rate exceeded 1MB/s. For the block sizes tested in our environment, a 128KB block size achieved the fastest transfer rate of 1.92 MB/s. In our configuration, tape read s consistently required 10% more CPU time than would otherwise be required regardless of the blocking factor used. Average tape read operation transfer rates by block size are indicated below:

| Block Size | Transfer Rate |
|---|---|
| 4KB | 0.26 MB/s |
| 8KB | 0.52 MB/s |
| 16KB | 0.91 MB/s |
| 32KB | 1.19 MB/s |
| 64KB | 1.63 MB/s |
| 128KB | 1.92 MB/s |

*d. DMF*

The achieved data transfer rate for DMF was 1.39 MB/s. This transfer rate remained consistent regardless of controller and tape device. The time measured included tape volume rewind and unload time.

*e. Channel Speed*

The measured channel speed for the SCSI 1 ports remained consistent at 2.59 MB/s regardless of controller and device selected. Because this figure included tape volume rewind and unload time we suspect that the SCSI 1 port channel bandwidth of 4.5 MB/s was approached if not realized. However, a measure of tape rewind and unload time was not taken to confirm this.

f. *Robotic Observations*

Considerable effort was taken to observe the motion characteristics of the STK 4400 robotics during the timed tape mount testing. These observations proved invaluable during the design phase of the C programs used to time the tape mounts and contributed to the following design changes:

1. Use the -w option on the UNICOS tpmnt command.

2. Use at least two tape volumes in each timed mount test job.

3. Use tape volumes not housed in the same general area of the STK 4400 ACS library.

4. Write a zero byte tar file to the tapes used for timing mounts and read them during the test phase by issuing the UNICOS tar command with the -t option.

These decisions eliminated the inherent bias imposed by the design of the STK 4400 hardware and firmware.

The robot would remain "at home" in the position it was last instructed to go. Using only one tape volume would always put the robot in front of that tape regardless of location within the library.

When the Cray Y-MP EL executed a tpmnt request without the -w option it would immediately begin processing the next command without waiting for acknowledgment from the STK Sun 330 server that the tape was in fact ready for a read or write operation. If the next command was a UNICOS rls -a (to release all resources) the robot would never physically mount the tape. It would sit in front of the tape volume still in its cell location within the library and never complete the fetch operation.

Once the design changes were implemented,, as many as four timed mount sessions were able to be run simultaneously, using eight different tape volumes. The intelligence built into the STK 4400 robotics system did not then appear to bias or impact our results.

## Discussion

*a. Hardware Considerations*

The transfer rate for tape I/O depends on several factors, the most important of which is the block size used for each transfer. However, to a much larger extent than would be evident at first sight, this transfer rate also depends on the characteristics of the hardware being used.

The I/O capabilities of the Cray Y-MP EL is impacted by its hardware design. The IOBB (see Figure 1) is designed to support a maximum block size of 128KB. In addition, the IOBB must be used to support all I/O transfers for all peripheral devices configured on the Cray Y-MP EL system. This is a limitation which cannot be modified by a user. All simultaneous I/O operations, tape and disk alike, compete for IOBB resources. In our configuration, this built in contention was exacerbated by the lack of available disk space (4 disk drives configured, see Figure 1). One job writing to tape while at the same time another job is trying to read from tape, will always force the second job to be put in a wait for I/O state until the first job completes. This wait time accumulates as part of the transfer time and results in a slower overall transfer rate for the second job.

Some of this contention can be "programmed out" by insuring that the file (i.e. being written to tape) resides on a file system that is not heavily used. We did not use the /tmp file system for this reason.

In our configuration, both tape controllers were attached to the same Cray Y-MP EL IOS and used the same IOBB. Contention for tape resources were evident in multiple session tape jobs and exhibited by both higher mount times and lower transfer rates. In a single session mount job we were able to achieve an average mount time of 28 seconds, but while running multiple mount jobs this time approached 52 seconds.

The Ciprico SCSI controller, with the IOBB does not permit block sizes larger than 128 KB. While the read transfer rate increased progressively for block sizes up to 128K, the write rate actually *decreased* for the same block size. We did not expect the disk -to -tape transfer rate to peak out at the 64K block size, since Cray supports block sizes of up to 4 MB in Unicos 6.1.6. The buffering in the 4781 caused the write rate to drop because the buffer set-up time increased the amount of time required to transfer data. The 4781 has a total of 1 MB of buffer space which is shared equally among all drives configured. Since we have eight tape drives we had only 125KB of buffer space allocated for each tape device. A 128K block size caused a delay in write operations, due to preparing the 125KB buffer space. In read operations, the transparent buffered I/O transfer technique worked well and allowed us to successfully stream our data.

*b. Robotic Considerations*

During this project we uncovered several important factors surrounding the motion and responsiveness of the robot. Our goal was not only to measure the time for various operations, but also to characterize the performance of the robot under different operational conditions.

Immediately noted was the fact that the robot would always find as its home position, the location in front of the last cartridge serviced (i.e., on a tape mount, it would remain in front of the tape drive; on a release, it would remain in front of the tape's cell location). Such a home location would only change once a new command to mount or dismount a different tape was received.

*c. UNICOS Issues*

During the testing, two significant problems related to UNICOS on the Cray Y-MP EL were noted with consistency. The first involved crashing the system when trying to read a previously written 1K block. This problem was attributed to Unicos internals and promptly resolved by Cray.

The second problem involved the tape daemon. The tape daemon hung consistently while trying concurrent tape I/O with four sessions by the same user. This problem was not resolved and is currently under investigation.

Another aspect of library usage also involved the tape daemon and its technique of allocating drives to user jobs. The Unicos tape daemon assigns tape drives in a round-robin fashion and does not take into consideration the location of a requested volume when assigning a drive to this volume. In library configurations of multiple silos this can have a serious impact on efficiency. Given the tape daemon's round-robin policy of assigning drives, a volume could be assigned a drive not attached to the silo in which that volume resides. The net effect would be to at least double the effective mount and dismount times because of "pass-throughs" that would be required to complete the tape request.

*f. Software Considerations*

Overall performance as perceived by a user is dependent upon the storage and file management system used to store data in the library. Cray's Data Migration Facility was used to manage data stored in the STK ACS 4400 library. An additional goal of this project was to measure and characterize DMF. We were able to achieve a very respectable transfer rate of 1.39 MB/s. Unfortunately, the ACS 4400 library became unavailable to us in the test environment shortly after completing this phase of our testing and we were unable to further examine DMF as a file management system.

With DMF we anticipated faster transfer rates due to its larger internal block size (49 KB) as compared to other popular file management systems, such as UniTree. Some versions of UniTree use a 15.5KB blocking factor, which may result in slower transfer times.

## Summary

This study of the STK ACS 4400 Tape library revealed important information concerning on-line mass storage space. While the STK Tape Library performed well, it did not achieve the manufacturer's advertised specifications in our test environment. We see channel port type as the primary limiting factor. Maximum throughput could be enhanced by upgrading to a SCSI 2 port, if available, or to a Block Mux port, which was used by the manufacturer in a highly controlled IBM test environment to achieve the figures they report.

Transfer rates from our study, which more closely emulate a real user environment, showed a direct correlation with block size. Other constraints, primarily due to inherent hardware

limitations, were circumvented by code modifications that optimized system commands and by using alternate file systems to reduce I/O contention.

The speed at which data can be transferred is affected by both hardware and software considerations as shown in our study. However, by applying the techniques reported in this paper, the requirement for reliable movement of data from disk to tape and from tape to disk would certainly be achieved.

## Acknowledgements

The authors appreciate the numerous helpful questions, comments and suggestions from Ben Kobler and Dr. P.C. Hariharan. Ray Yee and Frithjov Iverson of Cray Research gave valuable assistance in identifying and understanding specific UNICOS I/O features. Steve Cranage of STK offered useful insights into the ACS 4400 Library operation.

## References

1. S. Ranade, Mass Storage Technologies ( Meckler Corp. Westport, CT 1990)

2. UNICOS File Formats and Special Files Reference Manual, Publication SR2014, Cray Research Inc.

3. UNICOS User Commands Reference Manual, Publication SR2011, Cray Research Inc.

N94-33812

# Using Magnetic Tape Technology for Data Migration

**David Therrien and Yim Ling Cheung**

Epoch Systems
8 Technology Drive
Westboro MA 01581
Phone: (508)-836-4711
Fax: (508)-836-4884
dave@epoch.com

## Abstract

Magnetic tape and optical disk library units (jukeboxes) are satisfying the demand for high-capacity cost-effective storage. The choice between optical disk and magnetic tape technology must take into account the cost limitations as well as the performance and reliability requirements of the user environment.

Library units require data management software in order to function in an automated and user-transparent way. The most common data management applications are backup and recovery, data migration and archiving. The medium access patterns that these applications create will be described. Since the most user visible application is data migration, a queue simulator has been developed to model its performance against a variety of library units. The major subject of this paper is the design and implementation of this simulator as well as some simulation results. The relative cost and reliability of magnetic tape versus optical disk library units is presented for completeness.

## Data Management Applications

There are three main data management applications that library units are used for:

- The *Backup/Recovery* application enables data that has been lost due to magnetic disk failure or accidental user file deletion to be recovered from backup media. During backup, magnetic tape is preferred over optical disk for the following reasons:

  - Magnetic tape has a lower cost per megabyte than optical disk.
  - Magnetic tape can provide higher write data transfer rates than optical disk.
  - Backup is a sequential access process, so the random access feature of optical disk is not an advantage.

  When a large number of files must be recovered from a backup medium, optical disk could significantly speed up the recovery time. For optical disk, file to file access time is measured in milliseconds as opposed to seconds and even minutes on magnetic tape. However, recovery software that can sort the list of files to be recovered by physical location on magnetic tape has been developed, thereby minimizing search time. This sorting operation also reduces magnetic tape medium wear.

- *Migration* is a high-capacity, lower performance, user-transparent extension of a system's magnetic disk file system. A system that supports migration can provide a storage capacity that is well in excess of reasonable magnetic disk subsystems at a fraction of the cost. During the stage-out process, the migration application automatically identifies least-recently-used data on magnetic disk and moves that data

to a lower cost staging medium. Since data is staged-out periodically in bulk form and written to the staging medium in sequential form, magnetic tape is as effective as optical disk. Stage-in moves data from the staging medium back to magnetic disk when requested by a user. The fast drive load/unload and seek times of optical disk make it the preferred medium over magnetic tape for stage-in. These user requests for stage-in are random and unpredictable, making software optimizations ineffective for general storage systems. Since stage-in is the most user-visible application, it was chosen as the application to model against a variety of library units using the queue simulator.

- *Archiving* moves data from magnetic disk to a lower cost archive medium when it is either not being requested by users or it needs to be replicated for increased data availability. Users expect an access time of hours or days to acquire data that has been archived. Magnetic tape provides the following advantages over optical disk for archiving.:

  - The storage density of magnetic tape is higher than optical disk.
  - The cost per megabyte of magnetic tape media is significantly lower than optical disk media.
  - Data compression minimizes the physical storage space for off-line volumes. Hardware data compression is available within most tape drives and is not found in any optical disk drives today because disks are direct access devices that create operating system dependencies.

  The advantages of optical disk over magnetic tape in an archiving application include:

  - Longer archive life. Optical disk archive life is measured in tens to hundreds of years. Magnetic tape is measured in units to tens of years.
  - Lower medium maintenance. Most magnetic tape formats require retensioning to repack the tape onto the storage reels. Magnetic tape must also be periodically cycled from the archive environment back into the active-use environment in order to monitor medium quality and expire volumes with higher bit error rates. Optical disk requires no recycling of volumes in this manner.

Data management servers today that run these applications usually employ magnetic tape for backup/recovery & archiving. Optical disk has been the preferred medium for migration. With the recent availability of cost-effective magnetic tape library units, users are requesting that servers be configured with just tape library units, thereby eliminating the purchase of optical library units. Although this solution is attractive from a cost standpoint, there are significant performance and reliability concerns that must be addressed. The stage-in simulator has been used to quantify the performance differences between these two technologies.

## Performance Comparison and the Stage-In Simulator

### *Motivation for Developing the Stage-In Simulator*

Since stage-in is the most user-visible application of data management, the primary purpose of the stage-in simulator is to quantify the library unit service rate of various magnetic tape and optical disk library units. Optical disk provides a stage-in service time to the user of approximately twenty seconds, even in high request rate environments. Idle magnetic tape library units can service requests within minutes, but in high user request rate environments, the service time would extend to hours and possible days in extreme cases. The motivation for developing the simulator was to define the acceptable user request rate limits for a variety of library units.

### *Simulation Methodology*

The stage-in simulator is a discrete queue simulator. The steps involved in the development of this simulator follow typical simulation methodology [3] which includes planning, modeling, verification and validation and finally running applications against it.

*Simulator Planning*

The statement of the problem was formed during the planning phase. Initially, the simulator was going to be designed to model all data management applications being serviced by a single library unit. This problem statement was simplified to develop a model for just the migration stage-in application. This application was chosen since it is the most user-visible application and it exhibits the most unpredictable user-access patterns.

*Simulator Modeling*

During the modeling phase, the following activities were undertaken:

- The model of a library unit was developed
- The data model describing input, output and simulation variables was defined
- The simulator was written based on the library unit and data modeling.
- Performance data from real devices was measured and accumulated for input to the simulator.

*Library Unit Modeling*

Each user request that is sent to the stage-in simulator requires that a volume be mounted in a library unit drive so that data transfer can take place. The simulator uses a two-level library unit service model where *some* requests require a robot to mount the medium into one of the available drives and *all* requests require the use of a drive to access the data from the mounted volume. A queue is created when the user requests arrive faster than the library unit can process them, because either all of the drives and/or the robot are busy servicing an outstanding request. As shown in Figure 1, the stage-in simulator takes a single stream of user requests and attempts to satisfy them based on the utilization of a single shared robotics element feeding a number of drives.

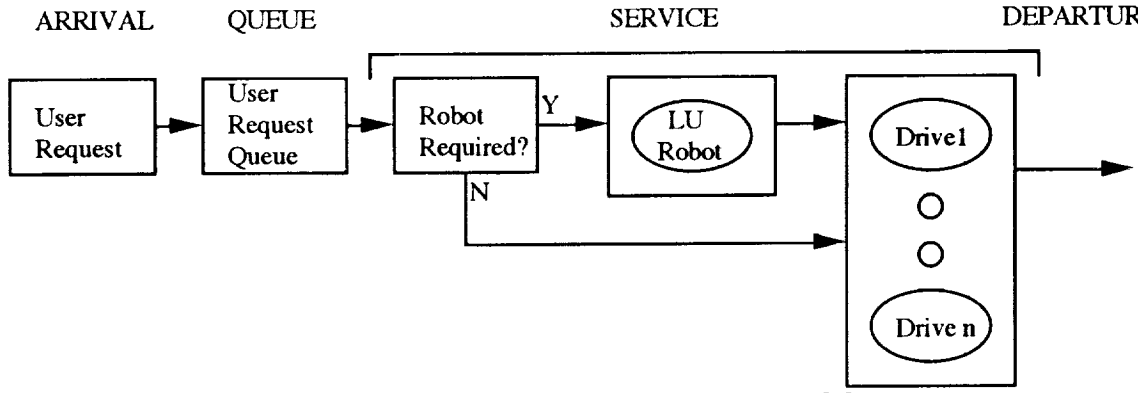



*Figure 1. Stage-in Simulator Service Model*

The service time for a user request involves a number of robot and drive service time components as shown in Table 1. When user requests require the use of a robot, the service time is the sum of all of the library unit and drive service time components If a user request arrives that can be satisfied by a drive that already has the right medium loaded, only the drive's access time and data transfer time are included in the service time for that request.

*Table 1: Robotics and Drive Components of Service Time*

| Magnetic Tape (Optical Disk) | Robot Is Required | No Robot Is Required |
|---|---|---|
| Rewind to BOT (Spin-down) Medium | √ | |
| Eject Medium from Drive | √ | |
| Robotics Exchange, drive->slot, slot-> drive | √ | |
| Drive Medium Load to BOT (Spin-up) | √ | |
| Drive Access Time | √ | √ |
| Drive Data Transfer Time | √ | √ |

*Data Modeling*

The simulator data model is comprised of input data, simulation variables and output as shown in Figure 2. The stage-in simulator accepts laboratory-measured library unit and drive performance data as input. It produces information on the percent utilization of the library unit robotics and drive(s) as well as the overall library unit service rate, average service time and maximum queue length as output. During simulation, simulation variables such as the user request rate and file size are varied to simulate different user environments.

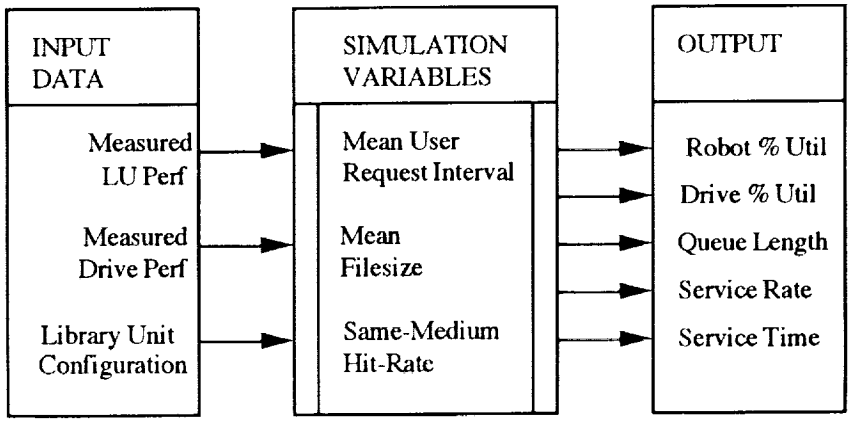



*Figure 2: Data Model of the Stage-In Simulator*

Simulator Output:

- Robot % Utilization - the percentage of time that the robot is busy during the simulation. Logged values near 100% indicate that the performance of the unit is limited by the robot.
- Drive % Utilization - the percentage of time that the drives in the LU are busy during the simulation Logged values near 100% indicate that the performance of the unit is limited by the drive.
- Queue Length - the size of the user request queue after servicing fifty user requests is logged to quantify the degree to which certain LU configurations fall behind in servicing simulated user request rates. For very high user request rates of very large files, the queue length of user requests to be serviced could reach into the thousands at the point in time where just the first fifty requests have been serviced.
- Service Rate - the number of user requests serviced per hour by the library unit.
- Service Time - the average service time per user request.

Simulator Variables:

- Mean User Request Interval - This variable represents the rate that user requests arrive for stage-in at the server. During simulation, mean user request intervals of 512, 256, 128, 64, 32, 16, 8, 4, and 2 seconds per request were run. This range was selected because it showed the region of user request rate that created drive and robot bound conditions for both magnetic tape and optical disk library units. During simulation, a Poisson distribution was applied to this mean user request interval to induce variability in arrival time. This distribution has been widely used to model arrival distributions and other seemingly random events [3].

- Mean File Size - mean file sizes of 10KB, 100KB, 1MB, and 10MB were selected for simulation. A Poisson distribution was applied to this mean file size to induce variability in user request file size. The drive's measured data transfer rate was multiplied by the file size during simulation to create the data transfer service time component of the total user request service time.

- Same-Medium-Hit-Rate (SMHR) - This variable allowed the simulator to model the behavior of servicing user requests that either exhibit a high degree of same-medium locality (SMHR = 100%) or a low degree of same-medium locality (SMHR = 0%). Each user request that arrives is tagged with a flag that indicates whether or not it requires the use of the robot based on the SMHR % value. Any SMHR percentage can be modeled. When the SMHR is 100%, the service time only includes a drive access time and a drive data transfer component. When the SMHR is 0%, the service time is the sum of all possible drive and robot times as shown in the "Robot Required" column of Table 1.

Simulator Input:

The first real application of the simulator was to model the stage-in performance of a number of magnetic tape and optical disk library unit configurations. For these devices, the following data was collected as input to the simulator:

- Library Unit (LU) Performance - Each real library unit that was modeled had its robotics exchange time measured to be used directly by the simulator. The exchange time includes the time to move a medium from a drive to a storage slot plus the time to move another medium from a storage slot into a drive. For the purpose of this simulation, some conceptual library units were created. Their exchange time was set to exchange times of similar commercially available library units.

- Library Unit (LU) Configuration - The number of media and drives associated with commercially available as well as conceptual library units.

- Drive Performance - the following drive parameters were measured for input to the simulator:

  - Drive Load Time - the time it takes a drive to load and spin up an optical disk or to load and get a magnetic tape to its BOT point.

  - Drive Unload Time - the time it takes a drive to spin-down and eject an optical disk or to eject a tape that was already rewound and at BOT.

  - Drive Data Transfer Rate - the rate at which the drive transfers data to/from the host computer. This rate was measured while servicing stage-in requests for all simulated drive devices. The measured data rate is generally lower than the manufacturer's published data transfer rate, due to drive and host latencies. For this reason, it was important to provide this measured data to the simulator.

- Drive Access Time - For optical disk drives, access time is the sum of seek time plus the rotational delay and is usually well under one second. The access time for magnetic tape drives is its search time which can be measured in minutes. Since magnetic tape drive search time is a major service time component for random stage-in requests, it was important to accurately model search characteristics for magnetic tape. The method of capturing this data involved first writing to the entire medium with a fixed file size and then performing random file reads on that volume while recording the time for each access. Six-hundred random access time samples were taken for a number of storage technologies. Table 2 shows the calculated mean and standard deviation of these six-hundred random access times.

*Table 2: Measured Mean and Standard Deviation for Various Device Random Access Times*

| Medium Type | Tape Length (Opt. Disk Diam.) | Median (seconds) | Standard Deviation (seconds) |
|---|---|---|---|
| Eraseable Optical Disk | (5.25") | 0.044 | 0.011 |
| WORM Disk | (12" ) | 0.429 | 0.199 |
| 8mm Tape | 54m | 31 | 15 |
| 4mm Tape | 90m | 47 | 25 |
| 8mm Tape | 112m | 53 | 31 |
| DLT Tape | 1100' | 54 | 31 |
| VHS Tape | T120 | 67 | 19 |

Random access times could have been generated for the simulator using the mean and standard deviation values in Table 2, but these two values alone did not capture the inherent skew visible in some of the distribution histograms (see Figures 3 and 4). When a random access service time component was required, one of the six-hundred random access time data points was selected.

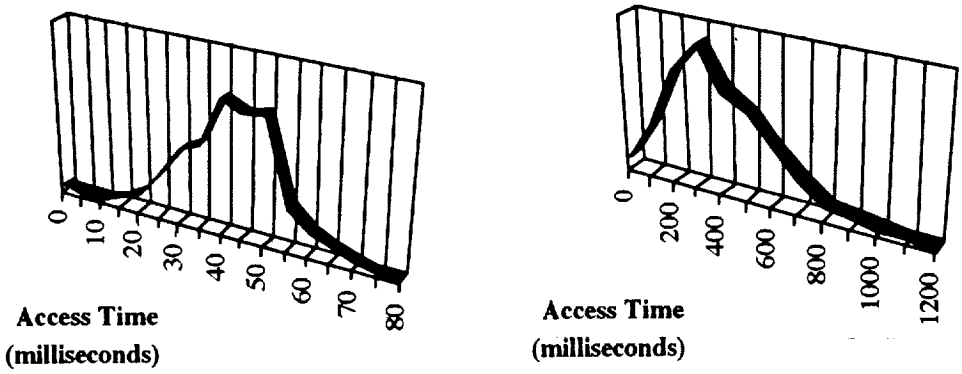



*Figure 3: 5.25" Eraseable Optical Disk and 12" WORM Disk Random Access Time Distribution*

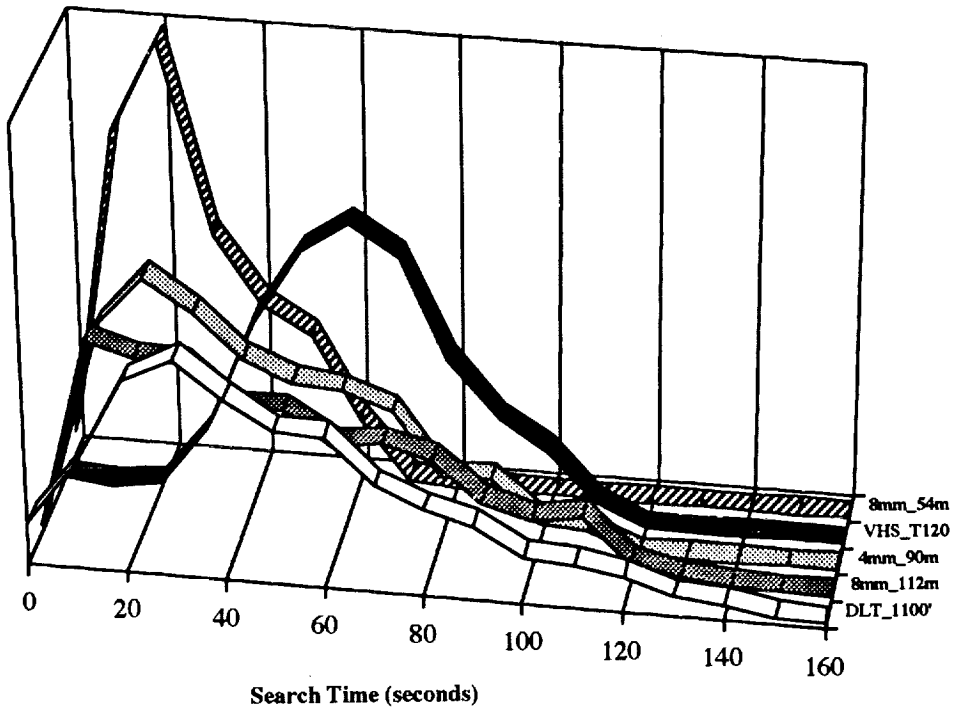



*Figure 4: Magnetic Tape Drive Random Access Time Distributions*

*Simulation Verification and Validation*

During the verification and validation phase, the program produced a significant amount of logged data to allow the servicing of each arrival to be studied. This data was helpful in identifying functional bugs in the early implementations of the simulator. Special simulation runs were executed that modeled the operating extremes of a device so the simulated results could be compared against calculated results for validation purposes. The simulator was executed over the same input data and simulation variables repeatedly to ensure the results produced were within a reasonable deviation from all other simulation runs. Also, by varying simulation variables and simulating different library unit configurations, sanity checks of the change in the output data revealed that the simulator was functioning properly.

During this phase of simulator development, it was important to identify the number of departures that had to be produced to provide consistent output data. Simulation runs of 25, 100 and 500 departures were executed with similar output results. For this application, the simulator was run for each user request rate, file size and SMHR value until 50 departures were completed.

*Simulator Application*

The simulator has the capability of modeling the performance of commercially available library units as well as those that are only conceptual. For this application, a total library unit capacity of 300GB was selected as a product normalizing criterion. Also, each library unit had a configuration of four drives. The library unit and media configurations are shown in Table 3:

*Table 3: 300GB Library unit configurations used during simulation*

| Medium Type/Size | Media/LU | Media Cap. (GB) | LU Cap. (GB) | Real /Conceptual LU |
|---|---|---|---|---|
| EO_5.25" | 215 | 1.3 | 279 | Real (DISC w/HP1.3GB) |
| WORM_12" | 47 | 6.5 | 307 | Real (Sony WDA-930) |
| 4mm_90m | 150 | 2.0 | 300 | Conceptual |
| 8mm_54m | 116 | 2.5 | 300 | Real (Exabyte EXB120) |
| 8mm_112m | 60 | 5.0 | 300 | Real (Exabyte EXB120) |
| VHS_T120 | 20 | 14.5 | 290 | Conceptual |
| DLT_1100' | 50 | 6.0 | 300 | Conceptual |

For this application of the simulator, the three SMHR percentages shown in Table 4 were run.

*Table 4: Effect of SMHR on Service Time*

| SMHR | Effect on Service Time |
|---|---|
| 0% | All user requests require a robotics exchange, drive load and unload |
| 50% | Half of the requests do not require robotics exchange and drive load/unload |
| 100% | Robot only used to load each drive once |

The theoretical maximum library unit service rate (requests per hour) is bounded by the user request rate as shown in Table 5. This is a units conversion from seconds per user request to library unit service rate expressed in requests per hour. For example, a user request every two seconds generates a theoretical maximum library unit service rate of 1800 requests per hour.

*Table 5: Maximum LU Service Rate based on the User Request Rate*

| User Request Rate(Sec/Req) | 64 | 32 | 16 | 8 | 4 | 2 |
|---|---|---|---|---|---|---|
| Maximum LU Rate (Req/Hr) | 56 | 112 | 225 | 450 | 900 | 1800 |

After running the simulator across many library unit models while varying the mean file size, mean user request rate and SMHR, it was observed that the library unit service rate was file size insensitive (from 10KB to 10MB) for lower SMHR percentages (0%, 50%). When SMHR approached 100%, file sizes at 10K, 100K and 1MB had similar service rate performance and 10MB files had measurably lower service rate performance, due to the significant service time component associated with data transfer. For this reason, the simulator output data was condensed to the four cases shown in Table 6.

*Table 6: Effect of File size on Service Rate for various values of SMHR*

| SMHR | File sizes (Bytes) | Service Rate Computation |
|---|---|---|
| 0% | 10K, 100K, 1M, 10M | average of service rate for 10KB, 100KB, 1MB and 10MB files |
| 50% | 10K, 100K, 1M, 10M | average of service rate for 10KB, 100KB, 1MB and 10MB files |
| 100% | 10K, 100K, 1M | average of service rate for 10K,100K and 1MB files |
| 100% | 10M | service rate for 10MB files |

Tables 7 through 10 display the simulated service rate of a number of library units expressed in requests per hour. This data represents the capability of each library units to service user requests that arrive at various request rates and file sizes.

The values in Tables 7 through 10 are coded with an indication of whether the unit was *drive bound* (shown in italics) or **robot bound** (shown in boldface). In either case, user requests were being placed in a queue for service and the overall library unit service rate was limited. Drive-bound service rates indicate that the library unit could not service requests at the required user request rate because the drive access time and data transfer characteristics were the limiting factor. Robot-bound service rates indicate that the unit was dominated by robotics exchanges and drive load/unload/search/rewind times.

The queue size data in the rightmost column of Tables 7 through 10 indicates the number of user requests that were waiting in the queue at the point in time when 50 requests were serviced and when the user request rate was at 2 seconds per request which is the worst case user-request rate condition.

*Table 7: LU Service Rate - SMHR = 0% - all file sizes*

| **User Req Rate(Sec/Req)** | 64 | 32 | 16 | 8 | 4 | 2 | **Queue Size** |
|---|---|---|---|---|---|---|---|
| **Max. LU Rate (Req/Hr)** | 56 | 112 | 225 | 450 | 900 | 1800 | |
| EO_5.25"_215c4d | 57 | 112 | 220 | **230** | **232** | **232** | 330 |
| WORM_12"_47c4d | 71 | 112 | 210 | 440 | **470** | **480** | 160 |
| 4mm_90m_150c4d | **45** | **45** | **47** | **48** | **47** | **47** | 1900 |
| 8mm_54m_116c4d | 45 | **48** | **48** | **48** | **48** | **48** | 1800 |
| 8mm_112m_60c4d | **36** | **37** | **37** | **37** | **37** | **38** | 2400 |
| DLT_1100'_50c4d | **30** | **30** | **30** | **30** | **30** | **30** | 3000 |
| VHS_T120_20c4d | **39** | **39** | **40** | **41** | **40** | **40** | 2200 |

*Table 7 Observations:*

- All magnetic tape library units were able to service user requests at a rate of 128 seconds per request (this user rate was simulated, but not shown in the table). Magnetic tape library units are limited to servicing only 30 to 50 requests per hour for SMHR = 0%.
- 12" WORM disk outperformed 5.25" eraseable optical disk in this model primarily because the 12" library unit robotics exchange time was faster. Optical disk technology can service user requests in the 8-16 second per request range.
- All library units became **robot bound** as the user request rate increased.
- After servicing only 50 user requests, very significant request queues were created for magnetic tape. With the average service time per user request at ~100 seconds for magnetic tape, the last user requests in the queue of ~2000 entries would not be serviced for 2.3 days. The first 50 requests to magnetic tape library units were serviced in approximately one hour.

*Table 8: LU Service Rate - SMHR = 50% - all file sizes*

| **User Req Rate(Sec/Req)** | 64 | 32 | 16 | 8 | 4 | 2 | **Queue Size** |
|---|---|---|---|---|---|---|---|
| **Max. LU Rate (Req/Hr)** | 56 | 112 | 225 | 450 | 900 | 1800 | |
| EO_5.25"_215c4d | 56 | 110 | 210 | 400 | **500** | **420** | 150 |
| WORM_12"_47c4d | 56 | 112 | 236 | 472 | 700 | **1050** | 55 |
| 4mm_90m_150c4d | 58 | **78** | **84** | **97** | **95** | **88** | 1000 |
| 8mm_54m_116c4d | 55 | **82** | **103** | **90** | **110** | **105** | 700 |
| 8mm_112m_60c4d | 56 | **63** | **65** | **70** | **70** | **70** | 1300 |
| DLT_1100'_50c4d | 56 | **62** | **60** | **62** | **60** | **60** | 1600 |
| VHS_T120_20c4d | 50 | **70** | **70** | **90** | **80** | **80** | 1500 |

*Table 8 Observations:*

- All magnetic tape library units were able to service user requests at a rate of 64 seconds per request.
- 12" WORM disk outperformed 5.25" eraseable optical disk in this model primarily because the 12" library unit robotics exchange time was faster. Either of these technologies is capable of servicing user requests at a rate of 8 seconds per request.
- All library units became **robot bound** (as shown in boldface) as the user request rate increased.
- Magnetic tape library units are limited to servicing only 60 to 100 requests per hour.
- Using shorter 54m tapes instead of the longer 112m 8mm tapes improved the LU service rate from ~90 request per hour to ~105 requests per hour.
- After servicing only 50 user requests, very significant request queues were created for magnetic tape. With the average service time per user request at ~60 seconds for magnetic tape, the last user requests in the queue of ~1500 entries would not be serviced for ~1 day. The first 50 requests to magnetic tape library units were serviced in approximately one hour.

*Table 9: LU Service Rate - SMHR = 100% - file size <= 1MB*

| **User Req Rate(Sec/Req)** | 64 | 32 | 16 | 8 | 4 | 2 | **Queue Size** |
|---|---|---|---|---|---|---|---|
| **Max. LU Rate (Req/Hr)** | 56 | 112 | 225 | 450 | 900 | 1800 | |
| EO_5.25"_215c4d | 56 | 110 | 230 | 450 | 860 | 1900 | 0 |
| WORM_12"_47c4d | 55 | 110 | 200 | 470 | 880 | 1670 | 2 |
| 4mm_90m_150c4d | 54 | 110 | 212 | *285* | *285* | *285* | 230 |
| 8mm_54m_116c4d | 56 | 114 | 190 | 370 | *450* | *440* | 160 |
| 8mm_112m_60c4d | 75 | 110 | *175* | *260* | *290* | *270* | 270 |
| DLT_1100'_50c4d | 56 | 104 | 200 | *270* | *270* | *263* | 290 |
| VHS_T120_20c4d | 56 | 112 | *190* | *200* | *200* | *200* | 400 |

*Table 9 Observations*

- Magnetic tape library units were able to service user requests at a rate of ~16-32 seconds per request.

- All magnetic tape library units became *drive bound* (shown in italics in the table) due to long drive search time as the user request rate increased.

- 5.25" eraseable had a performance advantage over 12" WORM, primarily due to the faster seek time of the smaller 5.25" medium (see Figure 3). It should be noted that a 5.25" medium contains only one-fifth the data of a 12" WORM medium. Either of these technologies is capable of servicing user requests at a rate of 2 seconds per request.

- Magnetic tape library units are limited to servicing only 200 to 400 requests per hour for this SMHR and mean file size range of 10K-1MB.

- Using shorter 54m tapes instead of the longer 112m 8mm tapes improved the LU service rate from ~270 request per hour to ~440 requests per hour.

- After servicing only 50 user requests, significant request queues were created for magnetic tape. With the average service time per user request at ~15 seconds for magnetic tape (because 4 user requests are being serviced simultaneously), the last user requests in the queue of ~275 entries would not be serviced for ~1 hour.

*Table 10: LU Service Rate - SMHR = 100% - file size = 10MB*

| **User Req Rate(Sec/Req)** | 64 | 32 | 16 | 8 | 4 | 2 | **Queue** |
|---|---|---|---|---|---|---|---|
| **Max. LU Rate (Req/Hr)** | 56 | 112 | 225 | 450 | 900 | 1800 | **Size** |
| EO_5.25"_215c4d | 56 | 101 | 218 | 417 | 843 | *1130* | 29 |
| WORM_12"_47c4d | 54 | 97 | 236 | *396* | *582* | *504* | 111 |
| 4mm_90m_150c4d | 59 | 103 | 172 | *201* | *219* | *202* | 232 |
| 8mm_54m_116c4d | 60 | 92 | 219 | *236* | *278* | *267* | 259 |
| 8mm_112m_60c4d | 52 | 113 | *184* | *169* | *191* | *184* | 442 |
| DLT_1100'_50c4d | 60 | 121 | *158* | *228* | *203* | *196* | 381 |
| VHS_T120_20c4d | 58 | 85 | *164* | *194* | *180* | *196* | 993 |

*Table 10 Observations:*

- All magnetic tape library units were able to service all requests at a rate of ~16-32 seconds per request.

- All magnetic tape library units became *drive bound* (shown in italics in Table 10) due to search rate and low data transfer rate as the user request rate increased. The 5.25" eraseable and 12" WORM library units became *drive bound* because of their relatively low read data transfer rate.

- 5.25" eraseable optical and 12" WORM are capable of servicing user requests at a rate of 16 seconds per request. This simulation set of parameters produced lower performance than that from Table 9, indicating the increased contribution of data transfer rate to the overall service time and the low data transfer rate characteristics of optical disk drives.

- Magnetic tape library units are limited to servicing only 200 requests per hour for this SMHR and file size.

- Using shorter 54m tapes instead of the longer 112m 8mm tapes improved the LU service rate from ~184 request per hour to ~267 requests per hour.

- After servicing only 50 user requests, significant request queues were created for magnetic tape. With the average service time per user request at ~18 seconds for

magnetic tape (because 4 user requests are being serviced simultaneously), the last user requests in the queue of ~350 entries would not be serviced for ~2 hours.

*Summary of Simulation Application*

Tables 7 through 10 indicate that any library unit can be driven to either being drive or robot bound under various user request load characteristics. If a user can determine a mean file size for the environment and estimate a user request rate, the SMHR percentage can be varied from 0% to 100% across a number of library unit models to determine the best technology fit for that environment.

## Cost Comparison

Today, most systems that support data management applications employ optical disk library units for migration and magnetic tape library units for backup/recovery. From the overall system cost perspective, there is a strong motivation to have all data management applications running on a single magnetic tape library unit to eliminate the cost of the optical disk library unit altogether.

When comparing various library unit options, the total cost of the library unit, its drives and its media must be considered. Magnetic tape library units with their media and drives are two to five times more cost effective than optical disk library units of a similar capacity.

The cost of library unit drives becomes a major factor in deciding on a storage technology for data management applications. Random stage-in requests from users can be serviced more effectively when more drives are available to service requests simultaneously. Middle and high-end magnetic tape drives (VHS, 3480, D2) and larger optical disk drives (12", 14") can be from three times to hundreds of times more expensive than smaller form-factor drives (3.5", 5.25"). For servicing high-volume stage-in requests, the preferred library unit configuration would house many low-cost drives as opposed to a few large drives. This assumes that the outstanding requests are serviced by as many different media as there are drives.

The cost per megabyte of optical disk media can be from three times to twenty times more expensive than magnetic tape media, depending on the two specific media types being compared. The cost of having to replace worn magnetic tape should be factored into the comparative media cost calculation. Media cost comparisons become important for environments where a significant amount of data will be archived off-line outside of the library unit.

The simulation data presented in Tables 7-10 represented the service rate performance of a variety of 300GB library units, each having four drives. The range of service performance that a single library unit can exhibit can be plotted against the estimated cost of the sum of the library unit, its drives and media to create a stage-in performance versus cost chart as shown in Figure 5. The service rate minimum and maximum values were taken from the 2 seconds per request column of Tables 7-10.

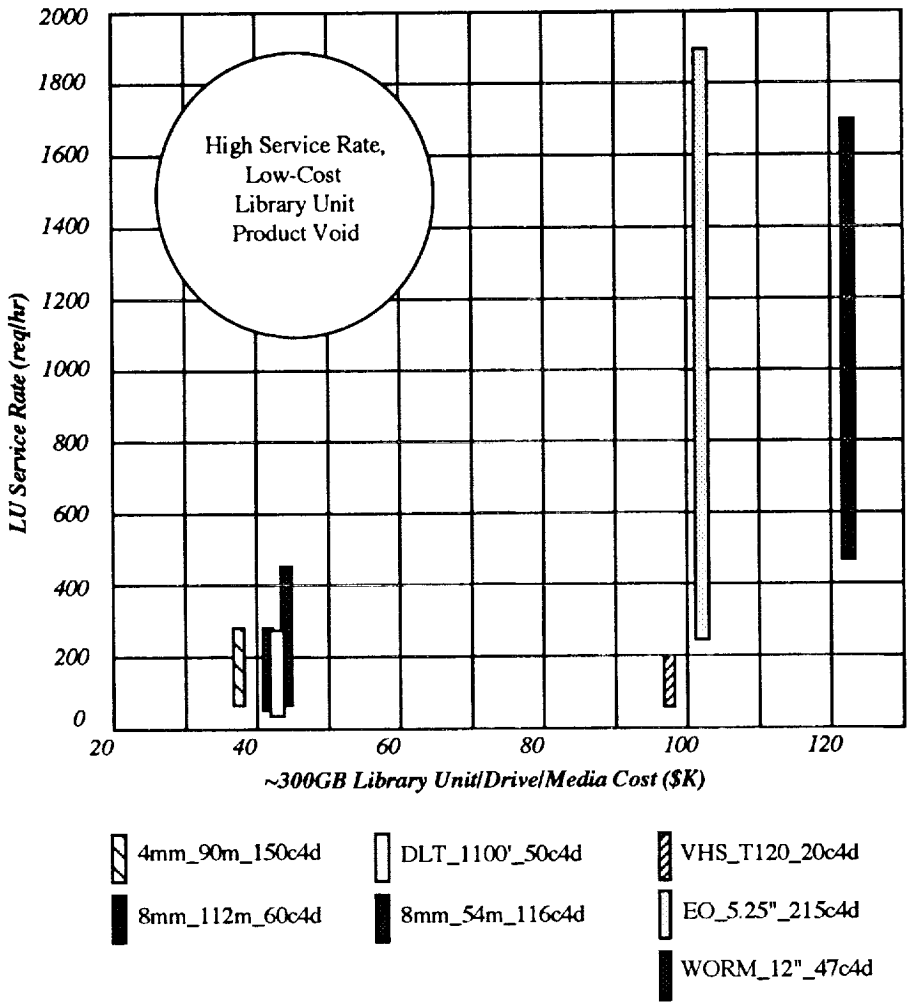



*Figure 5: Cost vs. LU Service Rate Performance of Simulated 300GB Library Units*

From the data presented in Figure 5, the following cost & performance observations can be made:

- Of the devices simulated, only optical disk library units provide service rate capability over 500 requests per hour.
- EO_5.25" is faster than WORM_12" for high SMHR values because its access time and data transfer rates are greater than WORM_12". EO_5.25" can also have a lower service rate than 12" WORM in very low SMHR environments, since WORM_12" has faster robotics exchange, drive load and unload times.
- Most magnetic tape technologies are clustered in the low service rate, low cost corner of the chart, with the exception of VHS. VHS tape drives are expensive, but they can transfer data faster than any other tape drive that was simulated.
- VHS produced the narrowest range of stage-in performance. This can be explained by the shifted random search distribution for VHS as compared to 4mm, 8mm, and DLT (see Figure 4). Although VHS did not perform well for stage-in, it would most likely outperform all other tape technologies when used with backup/recovery and stage-out data management applications.

- The 8mm configuration that used the shorter 54m tape had a high-end service rate that was significantly higher than the same library unit with fewer cartridges and longer 112m tapes. This is primarily due to the reduced search/rewind times of shorter tapes as shown in Figure 4. This may be an option for customers who are willing to significantly reduce the library unit capacity for an increase in overall stage-in performance.

- 4mm tape library units can provide service rate performance similar to DLT and 8mm library units at a reduced cost. This is primarily due to the lower cost of the 4mm drives.

- There is a high-service rate, low cost library unit product void that has not yet been filled by new library units as shown in Figure 5.

## Reliability and Data Availability Comparison

There are many optical disk and magnetic tape library units available that provide high reliability and high availability of user data. The critical reliability features of a library unit include:

- Robotics MEBF - the mean exchanges between failure of the robotics mechanism. A mean of one million exchanges has become the standard that most library units are expected to perform to.

- Drive MIBF - the mean insertions of media into the drive before drive failure occurs. For optical disk drives, MIBF is usually greater than 400,000. Magnetic tape drive MIBF values are usually much lower.

- Adaptive robotics system that can compensate for robotics wear or mechanical alignment drift over time.

- Robust robotics retry mechanisms to compensate for marginal physical alignment. Some tape library units exceed optical disk library units in their ability to recover from soft robot-movement errors.

The critical data availability features of a library unit include:

- Safe operator access to media and drives when the robotics fails. This allows an operator to "play the robot" while spare robotics parts are in transit for replacement. Most optical disk library units do not provide user access to media and drives while many magnetic tape library units do.

- Standard drives that can be installed in the library unit without drive modification. Because of the complicated medium loading mechanism of certain tape drives, some tape library units require that the standard drive be modified before installation into a library unit.

- Customer replaceable drives with foolproof drive alignment during drive replacement. Most optical disk library units are not designed with customer replaceable drives, but some tape library units do have this feature.

- No required periodic maintenance for drives, media and robotics.

Periodic maintenance is required on many magnetic tape drives and optical disk drives. Magnetic tape drive heads wear as the medium is passed over them. Helical scan drives like 8mm, 4mm, D2, and VHS have low head life ratings between 1,000 and 5,000 hours [1] while non-helical scan drives like QIC, DLT, and 3480 tape technology have head life ratings between 5,000 and 10,000 hours after which drive heads have to be replaced. Certain optical disk

drives require periodic maintenance in the form of an adjustment to the laser "head" that is responsible for writing and reading data. In either the magnetic tape drive case or the optical disk drive case, the cost of adjusting or repairing a worn head is usually a significant cost-of-ownership for lower-volume larger form-factor drives.

Overall media reliability can be segmented into archive reliability and active-use reliability. The archive life of most magnetic tape media is between 10 to 30 years and is significantly affected by temperature and humidity conditions in the archive environment. Many tape medium formats require retensioning in order to repack the tape onto the cartridge reel to eliminate stresses or to separate tape that is beginning to adhere to adjacent layers. For example, Exabyte suggests rewinding 8mm tape once every three years if kept in an archive environment of 20°C, and once every three months if kept in an archive environment of 30°C [2]. Optical disk media can provide stable archive storage from 25 to 100 years.

Active-use magnetic tape media reliability is mostly affected by the amount of wear between the drive head and media. Helical scan technologies like 4mm, 8mm, VHS, and D2 specify the number of passes against the head at ~1500 [1], where a pass is any forward or backward movement that creates contact with the head. Non-helical scan technologies like QIC, 3480, and DLT specify the number of passes of media at 5,000 to 20,000.

The limited medium pass count for helical scan tape media has not been a significant problem for use in a backup/recovery application, since backup is sequential and recovery is infrequent. When data is staged-out, it creates sequential access to magnetic tape which minimizes tape wear. **Stage-in requests, on the other hand, are random and unordered, and will impose a high number of passes over a tape during routine stage-in activity. Most tape technology cannot withstand this random-access activity.** To compensate for this lack of medium durability, data management software must be developed that provides improved media quality monitoring, data replication, and volume expiration features. From a hardware reliability and data integrity standpoint, the medium with the highest number of head to medium passes is preferred for the stage-in application.

## Summary

Magnetic tape library units are more cost-effective than optical disk library units. Unfortunately, magnetic tape drives and media are less durable and reliable than optical disk drives and media. Magnetic tape library units should only be used with user-request rates that don't cause the library unit to be drive or robot bound as shown in Tables 7-10.

The stage-in simulator has been used during system planning exercises to estimate the overall performance of very high capacity system configurations. It has been effective in quantifying the weakness of sequential devices that are perceived to be "high performance" but have been designed for high data transfer rate, not fast random-access to data.

Improved data migration software needs to be developed as the use of magnetic tape as a migration device becomes more widespread. Because of the relatively low magnetic tape medium and head reliability and durability, data management software must perform more media defect management and historical soft error logging to find the "best" point in time to expire a volume. In terms of performance, improved data placement algorithms must be developed that provide a high degree of data locality during stage-in.

## Future Simulation Activity

The simulator has been used for a number of other applications since its development. It has been effective in assisting library unit vendors in planning their next generation library units. The stage-in simulator can model the effect of changing the number of drives, cartridges and robotics elements within the library unit. The simulator can also assist in migration data management research by modeling a variety of stage-out data placement algorithms against real library unit devices. The goal of this research is to increase the locality of stage-in data.

A number of simulator enhancements are planned. These enhancements include:

- Adapting the current simulator to model library units with more than one non-conflicting robotics element to increase low SMHR performance.
- Producing a UNIX version of the program and providing it to customers for what-if analysis. It is currently written in ThinkC for an Apple Macintosh.
- A graphical output of the simulator progress as well as direct program charting of simulation results.
- Continued data acquisition of performance parameters for newer devices.
- Consideration for drives like VHS that allow the medium to be ejected without rewinding.

Also, the following applications are planned:

- Perform simulation of many library units in the 50-100GB range and the 1-10TB range to compare against the results of the 300GB simulation presented in this paper.
- Assist library unit vendors in planning their next generation library units. For instance, it is simple to create library unit configurations that show the results of changing the number of drives in the library unit from 1 to n drives to arrive at an optimal number of drives to robotics elements.
- Model the user-perceived effect of modifying library unit service rate components and configurations. For instance, if the drive load time could be cut in half from the present time, what effect would that have on the user-perceived service rate.
- A number of papers have been written on the subject of data placement on media during stage-out in order to optimize stage-in performance in the future [5-10]. Using the simulator, various data placement algorithms could be modeled against a variety of library units, user request rates and mean file sizes to quantify the effectiveness these algorithms. For example, a simulation could be run that quantifies the stage-in performance when data is staged-out *across* all magnetic tape volumes within a library unit instead of filling each volume to end-of-tape before starting the next volume. This scheme would be effective for library units that have fast robotics exchange times and magnetic tape drives that have fast load/unload/rewind times but relatively slow search times.

N94-33813

# Simulation of a Data Archival and Distribution System at GSFC

**Jean-Jacques Bedet, Lee Bodden, Al Dwyer, P C Hariharan***
Hughes STX Corporation
7701 Greenbelt Road, Suite 400
Greenbelt MD 20770
jbedet@nssdca.gsfc.nasa.gov

**John Berbert, Ben Kobler, Phil Pease**
NASA/GSFC
Greenbelt MD 20771
berbert@nssdca.gsfc.nasa.gov
kobler@nssdca.gsfc.nasa.gov

## Abstract

A version-0 of a Data Archive and Distribution System (DADS) is being developed at the Goddard Space Flight Center (GSFC) to support existing and pre-EOS Earth science datasets and test Earth Observing System Data and Information System (EOSDIS) concepts. The performance of the DADS is predicted using a discrete event simulation model. The goals of the simulation were to estimate the amount of disk space needed and the time required to fulfill the DADS requirements for ingestion (14 GB/day) and distribution (48 GB/day). The model has demonstrated that 4 mm and 8 mm stackers can play a critical role in improving the performance of the DADS, since it takes, on average, 3 minutes to manually mount/dismount tapes compared to less than a minute with stackers. With two 4 mm stackers and two 8 mm stackers, and a single operator per shift, the DADS requirements can be met within 16 hours using a total of 9 GB of disk space. When the DADS has no stacker, and the DADS depends entirely on operators to handle the distribution tapes, the simulation has shown that the DADS requirements can still be met within 16 hours, but a minimum of 4 operators per shift were required. The compression/decompression of data sets is very CPU intensive, and relatively slow when performed in software, thereby contributing to an increase in the amount of disk space needed.

## Introduction

The Goddard Space Flight Center (GSFC) is building a Version 0 Distributed Active Archive Center (V0 DAAC) to support pre-EOS projects and test Earth Observing System Data and Information System (EOSDIS) concepts. This system will consolidate management and provide access, archiving, and distribution functions for Goddard's Earth Science data. This paper describes a study of the performance of one of the elements of the DAAC; the Data Archive and Distribution System (DADS). The DADS is responsible for the ingestion, archiving and distribution of pre-EOS data. To assess the storage needs and performance capability of the DADS, a discrete event simulation model has been developed using the NASA Data Systems Dynamic Simulator (DSDS) package. This study has identified potential bottlenecks in the utilization of the selected ingest, archival, and distribution devices (on-line disks, automated tape libraries, jukeboxes, and magnetic tape drives), and has identified the performance benefits to be gained by adding one or more stackers to the 4 mm and 8 mm tape drives.

---

* Present address: Systems Engineering and Security, Inc.
7474 Greenway Center Drive, Suite 720
Greenbelt, MD 20770

The GSFC DADS is expected to ingest 14 GB/day of data and distribute an estimated 48 GB/day of data over various media (4 mm, 8 mm, and 9 track tapes) and over the network. With these large volumes of data to be ingested and distributed, the GSFC DADS wanted to assess the amount of staging disk space and the number of tape drives required to meet the estimated DADS workload. To address these issues, a discrete event simulation model of the DADS has been developed using the NASA DSDS package. The model simulates the ingestion of regular and reprocessed data, and the fulfillment of standing orders and user requests for data distribution.

First, the GSFC hardware configuration and the main DADS activities that are simulated are described. A high level view of the DADS model is presented and the results obtained from the model are discussed. The contention for the robots of the Metrum RSS-600 Automated Tape Library (ATL) and the Cygnet optical disk jukebox, and the various tape and disk drives is explained. In particular, we looked into the effect of having human operators in the distribution process and quantify how 4 mm and 8 mm stackers could improve the performance. The impact of using compression and decompression techniques has also been studied. Finally, the lessons learned and future work are summarized in the last paragraph.

## V0 GSFC DADS Configuration

First we examine the storage devices used to ingest, archive, and distribute data.The current hardware configuration of the V0 GSFC DADS, as of August 1993, is illustrated in Fig. 1.

*Ingestion*

Most of the data to be ingested at the GSFC DADS is received over an FDDI network (100 Mbits /s) and copied to Unix staging disks (2.7 MB/s). The ingestion operation is performed overnight to minimize the impact on the network. A small amount of data is received on 3480 cartridges.

*Archival*

To automate the archival and retrieval process, the GSFC DADS has acquired a Cygnet 1803 jukebox with 2 ATG WORM drives and an RSS-600 Metrum Automated Tape Library (ATL) with 4 RSP 2150 VHS drives. Based on the data type, a data set is either stored on the Cygnet jukebox, which can hold up to 131 12" WORM platters (9 GB per platter), or on the Metrum ATL which can accommodate 600 magnetic T120 VHS cassettes (14.5 GB per cassette). The file management is controlled by Unitree 1.7, which is running on an SGI 4D/440 workstation. Files are automatically migrated from the Unitree magnetic disk cache, which holds 13.8 GB, to either the jukebox or the ATL. Similarly requests for data already residing on the jukebox or the Metrum ATL are handled by Unitree, which retrieves the data and puts them in its cache. Table 1 provides the specifications of the two archive devices selected for the DADS.

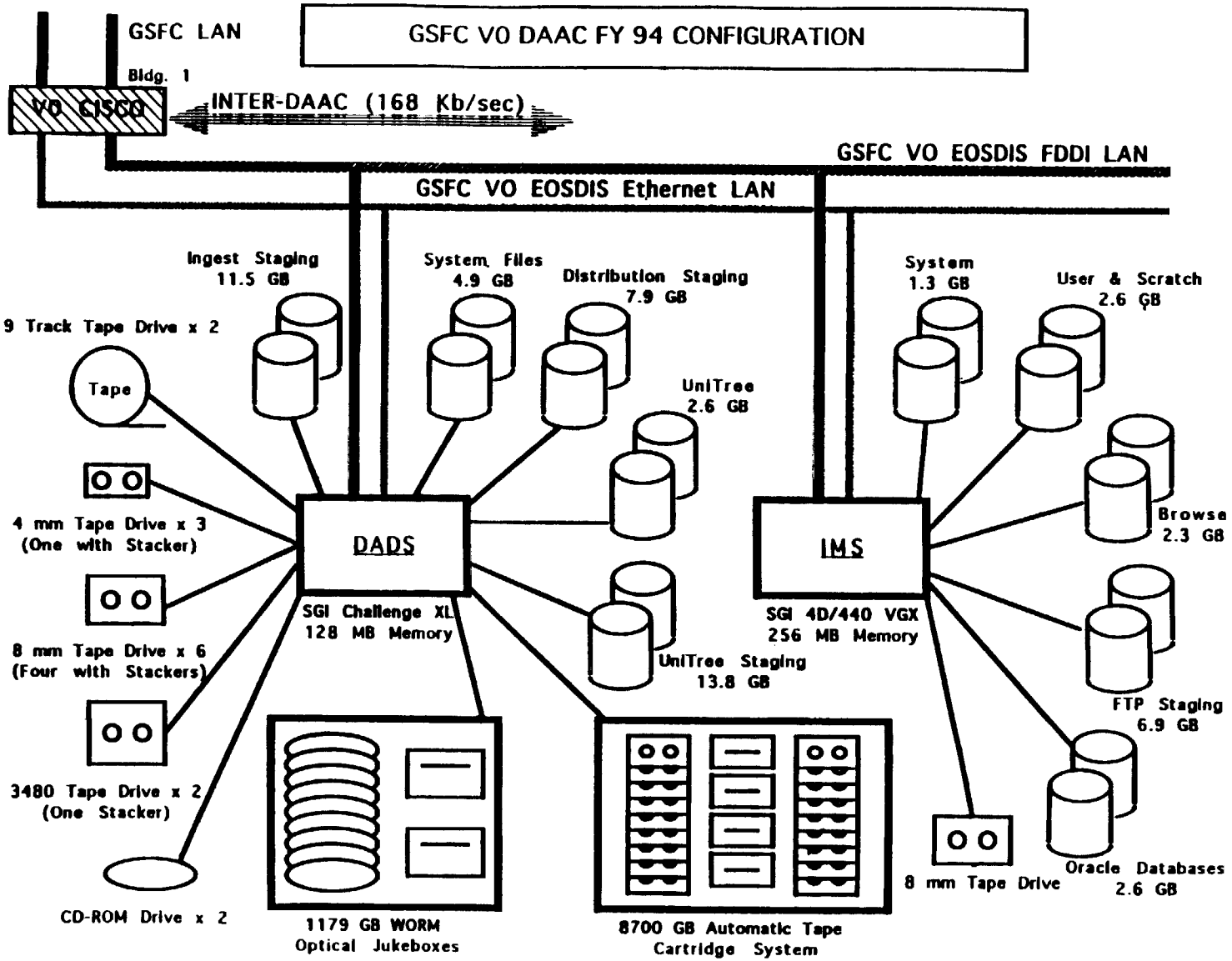



Fig 1. GSFC V0 DAAC Configuration

| | 1803 Cygnet jukebox | RSS-600 Metrum ATL |
|---|---|---|
| Media used | 12" WORM platters | T120 VHS tape cassettes |
| Drive type | ATG WORM | Metrum RSP 2150 VHS |
| # of drives available | 2 | 4 |
| drive read/write rate (MB/s) | 0.5 | 1 |
| Media capacity (GB) | 9 (4.5 GB/side) | 14.5 (T120), 16 (T160) |
| Number of media | up to 131 | 600 |
| System capacity (GB) | 1179 | 8700 (T120), 9600 (T160) |
| Number of robot arms | 1 | 1 |
| Avg robot access time(s) | 8 | 8 |

Table 1. Specification of DADS archive devices

*Distribution*

It is expected that most data will be requested on 8 mm and 4 mm cassettes. To automate the distribution process, the DADS has an 8 mm stacker and is investigating the possibility of purchasing additional 4 mm and 8 mm stackers. For users who still need their data on 6250 bpi tapes, the DADS has two 9 track drives. For quick delivery and for small files, the data may also be sent over the network. The characteristics of the distribution devices are summarized in Table 2.

| | 4 mm DAT | 8 mm Exabyte (8500) | 9 track drive |
|---|---|---|---|
| Number of drives | 3 | 4 | 2 |
| Manual fetch time (min) | 1 or 3 | 1 or 3 | 1 or 3 |
| Stacker fetch time (s) | 60 | 60 | N/A |
| Load time (s) | 14 | 42 | 60 |
| Unload time (s) | 10 | 21 | 20 |
| Manual return time (min) | 1 or 3 | 1 or 3 | 1 or 3 |
| Stacker return time (s) | 60 | 60 | N/A |
| Search rate (MB/s) | 13 | 22.6 | 0.15 |
| Rewind rate (MB/s) | 25 | 28 | 1 |
| Read transfer rate (MB/s) | 0.17 | 0.40 | 0.17 |
| Write transfer rate (MB/s) | 0.17 | 0.43 | 0.17 |

Table 2. DADS distribution parameters

## DADS Activities Simulated

The two main activities simulated in the model are the ingestion/archival and the distribution. The ingested data are subdivided into two categories: regular processing data and reprocessing data. For both categories the data are first copied to disks (Unix disks), compressed (optional), and transferred to the Unitree cache (referred to as Unitree disks) and then migrated automatically, under the control of Unitree, to the Cygnet jukebox or the Metrum ATL. In the case of ingested data, the metadata containing information about the data, are first extracted before being sent to the Unitree cache. In addition, some of the new regular ingested data are known in advance to be requested for distribution. The data used to satisfy these advance requests (called "standing orders") are kept on-line on the Unix disks until all the standing orders have been fulfilled. For the distribution requests that are not standing orders, the data are retrieved from one of the robotic devices (Metrum ATL or Cygnet jukebox), copied to the Unitree cache, decompressed (optional), staged to the Unix disks, and finally copied to one of the distribution media.

The sequence of actions, for each activity performed at the DADS, is as follows:

*Ingestion/Archival*

- Write incoming data to Unix disks
- Compress(optional) and copy data to Unitree cache
- If data are used in standing orders
  First complete all standing orders and then delete data from Unix disks
- If data are not used in standing orders
  Delete data from Unix disks
- Migrate data from Unitree cache to robotic devices archive
- Mark file as purgeable from Unitree cache

*Distribution (non-standing orders)*

- Retrieve data from robotic devices
- Copy data to Unitree cache
- Decompress (optional) and copy data to Unix disks
- Mark file as purgeable from Unitree cache
- Copy data to distribution media
- Delete data from Unix disks

*Distribution (standing orders)*

- Read staged data from Unix disks
- Write data to distribution media
- Remove staged data from Unix disks

## Simulation model

Using DSDS, a model has been developed to simulate the various activities and devices at the DAAC. The block diagram illustrated in Fig 2, has four main components. The first one contains the elements that generate the files to be ingested or distributed. File sizes and inter-arrival times are both randomly computed by the use of appropriate distributions (e.g. uniform). This first component models the expected data volume to be ingested and distributed by the DADS. The second component (initialization), identifies the source and the destination of each file as well as the disk to which the file is temporarily stored. The third component acts as a switch, directing the file to the right device. The fourth component (devices) models the various storage devices and the resource allocation. After leaving the devices component, the step is incremented by the counter and the file is once again directed to the appropriate device by the switch component. This process is repeated until the file reaches the end component.

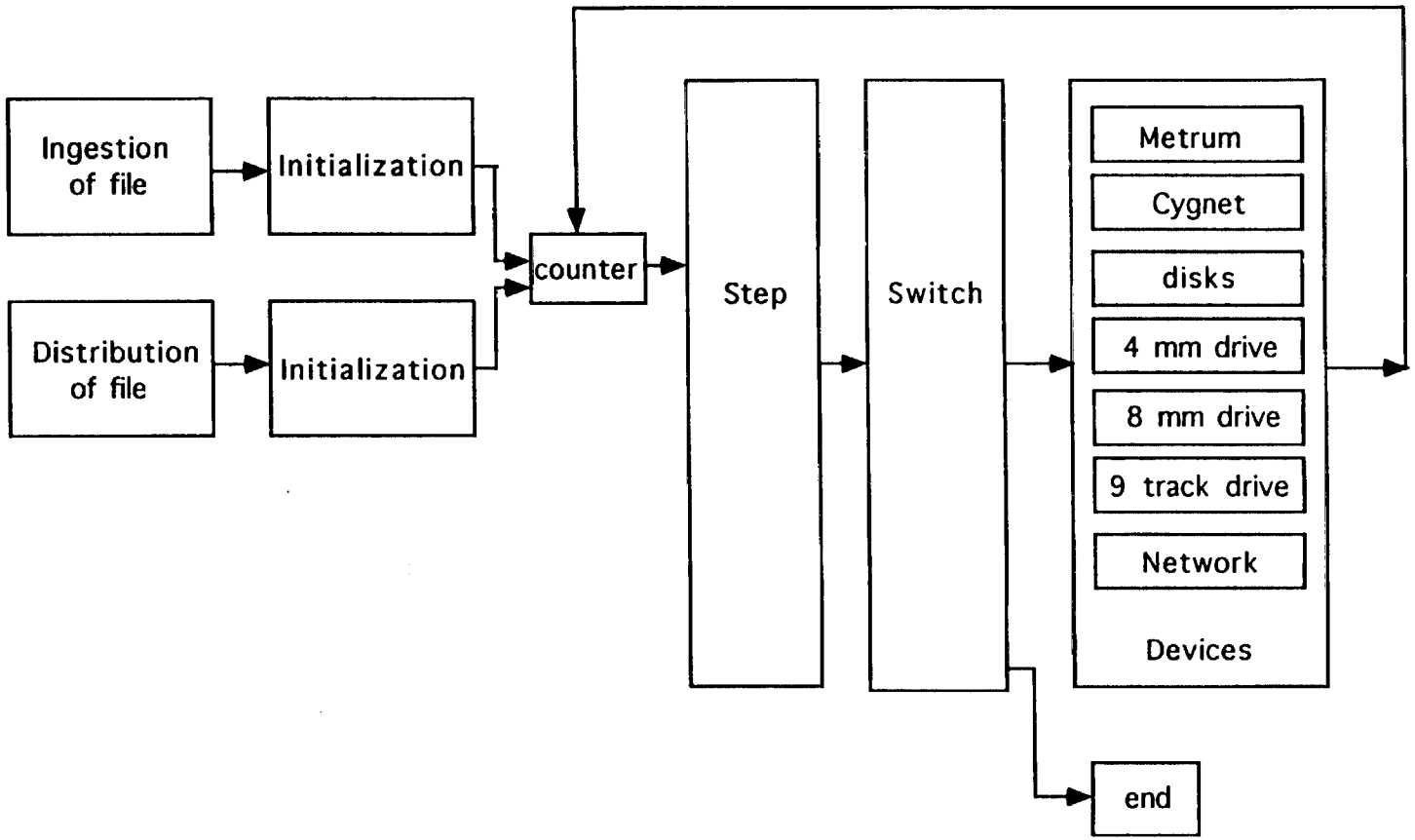



Fig. 2. Schematic Diagram of the DADS Model

*DADS Simulation Model Assumptions :*

- Any file ingested is first copied to the Unix staging disks and then to the Unitree cache disks for migration into the archive storage devices.
- Any file retrieved from the archive devices for distribution, is first copied to the Unitree cache disks and then to the Unix staging disks for tape copy.
- The simulation allows all the various DADS disk and tape storage devices to have different data transfer rates in read and write modes.
- UNITREE 1.7 supports multiple simultaneous read operations but can only support a single write operation at a time. This Unitree restriction has been implemented in the current version of the model.
- Each file read from or written to the jukebox requires a load and unload of a platter.
- Each file read from or written to the Metrum storage module, requires a load, and unload operation of a cassette.
- When not using a stacker, each file copied to a 4 mm drive, 8 mm drive, or 9 track drive requires a manual fetch of the blank tape to the drive load mechanism and a return of the copied tape.
- In the first two scenarios examined the requests are assumed to be distributed with an equal probability on each of the three types of media (8 mm, 4 mm, and 6250 bpi) in the proportion of 33%, 33%, and 33%. For later scenarios this was changed to 50%, 33%, and 17% after a survey of potential users was made.
- Distribution request files (non-standing orders) are uniformly distributed over 12 hours.
- Ingestion files for the SeaWiFS regular processing are uniformly distributed over 2 hours (except in scenario 2, when the 2 hours is changed to 16 minutes).
- Ingestion files for the SeaWiFS reprocessing are uniformly distributed over 16 hours.
- Ingestion files for the non-SeaWiFS regular processing are uniformly distributed over 6 hours.

## DADS WORKLOAD

The largest volume data set to be ingested, archived and distributed by the Goddard DAAC is that from the SeaWiFS project (see Tables 3 and 4). The SeaWiFS project regular processing operation will send 1.59 GB/day to the GSFC DAAC over the network. In addition, periodically, the SeaWiFS project will reprocess all the data and redeliver replacement data at a rate of 8.9 GB/day. The total estimated distribution data volume for SeaWiFS (including the standing orders) is 40 GB/day (see Table 4).

In addition to SeaWiFS data, the GSFC DAAC will also service a number of other projects. These non-SeaWiFS data add 4.23 GB/day of ingest and 7.97 GB/day of distribution. In this report the SeaWiFS regular ingest and non-SeaWiFS ingest have been referred to as ingestion (regular). The workload was modeled to represent these separate categories so as to facilitate model validation with actual measurements of the DAAC operation with SeaWiFS test data.

In the simulation, using the Workload Model for Archive and Distribution of SeaWiFS Data (November 16, 1992), the daily volume of ingested data of each data type, has been estimated and is tabulated in Table 3. This table indicates also the percentage of this volume from each of the two sources to each of the two archive destinations . For instance, SeaWiFS L1A product is expected to have a volume of 694 MB per day. All SeaWiFS L1A data will be received over the network, and will be stored on the Cygnet jukebox. For simplicity, the simulation model assumes that all data ingested is transmitted over the network. This will have the effect of adding 1.23 GB/day to network ingestion which are currently assumed to be ingested by reading 3480 cartridges. Similarly Table 4 represents the workload for the distribution.

Table 3. Ingestion workload

| | | Source % | | Destination % | |
|---|---|---|---|---|---|
| **Data Type** | **Volume (GB/day)** | **% From network** | **% from 3480** | **% to Jukebox** | **% to Metrum** |
| | | | | | |
| *SeaWifs (regular)* | | | | | |
| L1A | 0.694 | 100 | | 100 | 0 |
| L2 | 0.461 | 100 | | | 100 |
| L3 | 0.43 | 100 | | | 100 |
| **Total** | 1.585 | 100 | | 43.79 | 56.21 |
| | | | | | |
| *SeaWifs (reprocessing)* | | | | | |
| L2 | 4.61 | 100 | | | 100 |
| L3 | 4.3 | 100 | | | 100 |
| **Total** | 8.91 | 100 | | | 100 |
| | | | | | |
| *Non-SeaWiFS (regular)* | | | | | |
| AVHRR | 1 | 100 | | 0 | 100 |
| TOVS | 0.233 | | 100 | 100 | 0 |
| UARS | 1 | 100 | | 0 | 100 |
| DAAC Climate data | 1 | | 100 | 100 | 0 |
| CZCS | 1 | 100 | | 100 | 0 |
| **Total** | 4.233 | 70.87 | 29.13 | 52.75 | 47.25 |
| | | | | | |
| **Grand Total** | **14.728** | **91.63** | **8.37** | **19.87** | **80.13** |

Table 4. Distribution workload

| | | Source % | | | Destination % | | |
|---|---|---|---|---|---|---|---|
| **Data Type** | **Volume (GB/day)** | **% from Disk** | **% from Jukebox** | **% from Metrum** | **% to 4 mm** | **% to 8 mm** | **% to 9 track** |
| | | | | | | | |
| *SeaWifs* | | | | | | | |
| Global data order | 16 | 100 | | | 33 | 33 | 33 |
| large chunks | 14 | 50 | 20.91 | 29.09 | 33 | 33 | 33 |
| small chunks | 8 | 25 | 49.80 | 25.20 | 33 | 33 | 33 |
| level 3 | 2 | | | 100 | 33 | 33 | 33 |
| **Total** | 40 | 62.5 | 17.28 | 20.22 | 33 | 33 | 33 |
| | | | | | | | |
| *Non-SeaWiFS* | | | | | | | |
| AVHRR | 5 | 80 | | 20 | 33 | 33 | 33 |
| TOVS | 0.466 | 100 | | | 33 | 33 | 33 |
| UARS | 1 | 50 | | 50 | 33 | 33 | 33 |
| DAAC Climate data | 1 | | 100 | | 33 | 33 | 33 |
| CZCS | 0.5 | | 100 | | 33 | 33 | 33 |
| **Total** | 7.966 | 62.34 | 18.83 | 18.83 | 33 | 33 | 33 |
| | | | | | | | |
| **Grand Total** | **47.966** | **62.47** | **17.54** | **19.99** | **33** | **33** | **33** |

## DADS Performance

In order to estimate the amount of disk space necessary to ingest/archive and distribute data, and to determine the time required to satisfy the daily activities at the DADS, the discrete events model has been run for scenarios with varying assumptions. These are summarized in Table 5:

Table 5. Summary of assumptions by scenarios

| | Scenarios | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1-A | 1-B | 2 | 3 | 4 | 5 | 6 | 7-A | 7-B |
| **Assumptions** | | | | | | | | | |
| Regular SeaWiFS ingestion (hours) | 2 | 2 | 0.26 | 2 | 2 | 2 | 2 | 2 | 2 |
| Reprocessing SeaWiFS ingestion (hours) | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| Non-SeaWiFS ingestion (hours) | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| Distribution SeaWiFS (hours) | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| % Distribution on 8 mm tapes | 33.3 | 33.3 | 33.3 | 50 | 50 | 50 | 50 | 50 | 50 |
| % Distribution on 4 mm tapes | 33.3 | 33.3 | 33.3 | 33 | 33 | 33 | 33 | 33 | 33 |
| % Distribution on 9 track tapes | 33.3 | 33.3 | 33.3 | 17 | 17 | 17 | 17 | 17 | 17 |
| Ingestion (GB/day) | 14.7 | 14.7 | 14.7 | 14.7 | 29.4 | 14.7 | 14.7 | 14.7 | 14.7 |
| Distribution SeaWiFS (GB/day) | 25 | 40 | 40 | 40 | 80 | 40 | 40 | 40 | 40 |
| Distribution non-SeaWiFS (GB/day) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 |
| Number of operators | ∞ | ∞ | ∞ | ∞ | ∞ | 1-∞ | 1 | 1 | 1 |
| Number of 8 mm stackers | 0 | 0 | 0 | 0 | 0 | 0 | 0-2 | 2 | 2 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Number of 4 mm stackers | 0 | 0 | 0 | 0 | 0 | 0 | 0-2 | 2 | 2 |
| Compress/Decompress (Y/N) | N | N | N | N | N | N | N | N | Y |
| Operator Avg. response/fetch time (min) | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 |

- Scenario 1-A: Disk space requirement for the ingestion and SeaWiFS standing order distribution.

First, the DADS has been examined ingesting all data over the network and processing the SeaWiFS standing orders. The disk space used on the Unix disks and the Unitree cache is illustrated in Fig 3. During the first two hours, the DADS receives regular SeaWiFS data, migrates them to the archive, and retains a copy of the data (1.6 GB) on Unix disks in order to fulfil the standing orders. All the other ingested data are rapidly migrated to the archive and do not accumulate on the Unix disks. Due to the large volume of standing orders (25 GB) to be copied to slow devices such as 8 mm and 4 mm tape drives, the standing order distribution operation continues up to 10 hours. At that time, the standing orders are completed and regular SeaWiFS data are deleted from disk, creating a big drop in the Unix disk space.

The Unitree disk space is also illustrated in Fig 3 and shows a peak of approximately 400 MB. During the ingestion process, data are migrated to the robotic devices archive as soon as possible. In this scenario the total ingestion rate approximately matches the archival rate (including robotic access times as well as jukebox and Metrum ATL write rates), so that only a small amount of data is retained in the Unitree cache. After migrating the files to the robotic archive devices, they are marked as purgeable in the Unitree cache. Only the non-purgeable files are plotted in the Unitree disk space in Fig 3.

- Scenario 1-B: Disk space requirement adding SeaWiFS non-standing order distribution.

Figure 4 represents the disk space used as a function of time when the SeaWiFS non-standing orders are added to the previous workload. With a non-standing order, the data are first retrieved from the Cygnet jukebox or the Metrum ATL, copied to the Unitree disk cache, and then copied quickly to the Unix disks. After writing the data set to the Unix disks, the space used in the cache is marked as purgeable. The Unitree cache used with non-purgeable files remains small (~400 MB) over time and is similar to the previous case (Fig 3). The daily volume of non-standing SeaWiFS orders to be distributed is quite large (15 GB) and the distribution tape device write rates are rather slow (see Table 2). This creates a bottleneck and the files are staged in the Unix disks for several hours, waiting to be copied to tapes. The Unix disk space used is illustrated in Fig 4 and it shows a peak of 5.5 GB and a sudden drop at 11 hours, when the standing orders are completed. The backlog of requests staged on Unix disks disappears at about 15.5 hours.

- Scenario 2: Effect of ingesting regular SeaWiFS data over a shorter time interval.

In the previous scenarios, the daily SeaWiFS ingestion was assumed to occur over a 2 hour period. A question of interest is to examine the DADS system when the ingestion is performed during a shorter interval of time. Fig 5 illustrates the case of an ingestion over 16 minutes. As expected, the completion of the standing orders, indicated by a sudden drop of 1.6 GB, occurs earlier (9 hours instead of 11 hours). The non-standing orders backlog disappears at 14.5 hours, or about 1 hour sooner than before. The Unitree cache during the first 2 hours is also much larger (1.4 GB). The data are ingested at a rate which exceeds the archival rate to the Metrum ATL and the Cygnet jukebox. This causes the data to be delayed in the Unitree cache. By controlling the ingestion schedule of the data (i.e., spreading it out), it is possible to keep the Unitree cache used at a minimum, but this increases the time required to eliminate the backlog in the distribution operations.

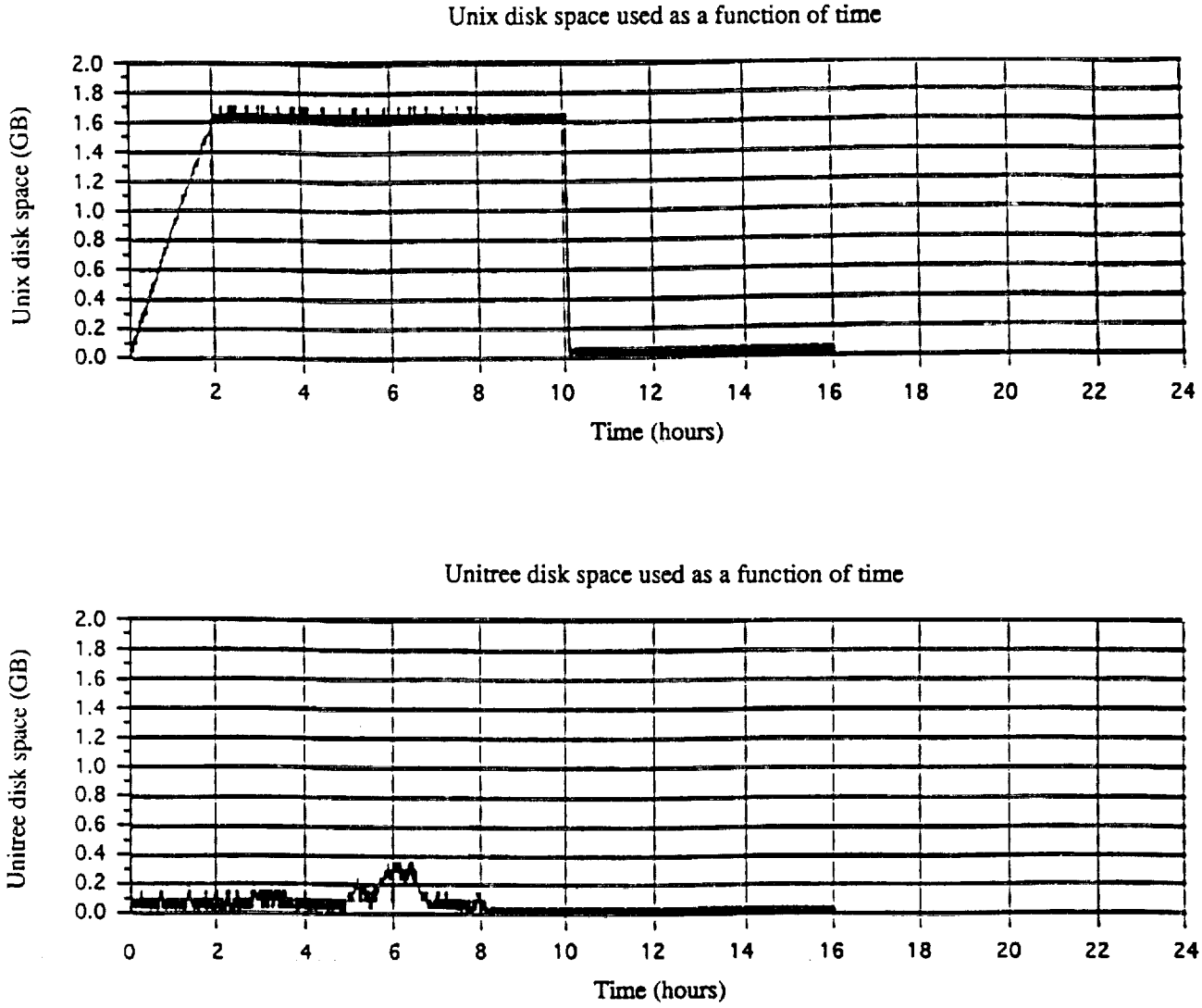



Fig 3. Disk space used in scenario 1-A

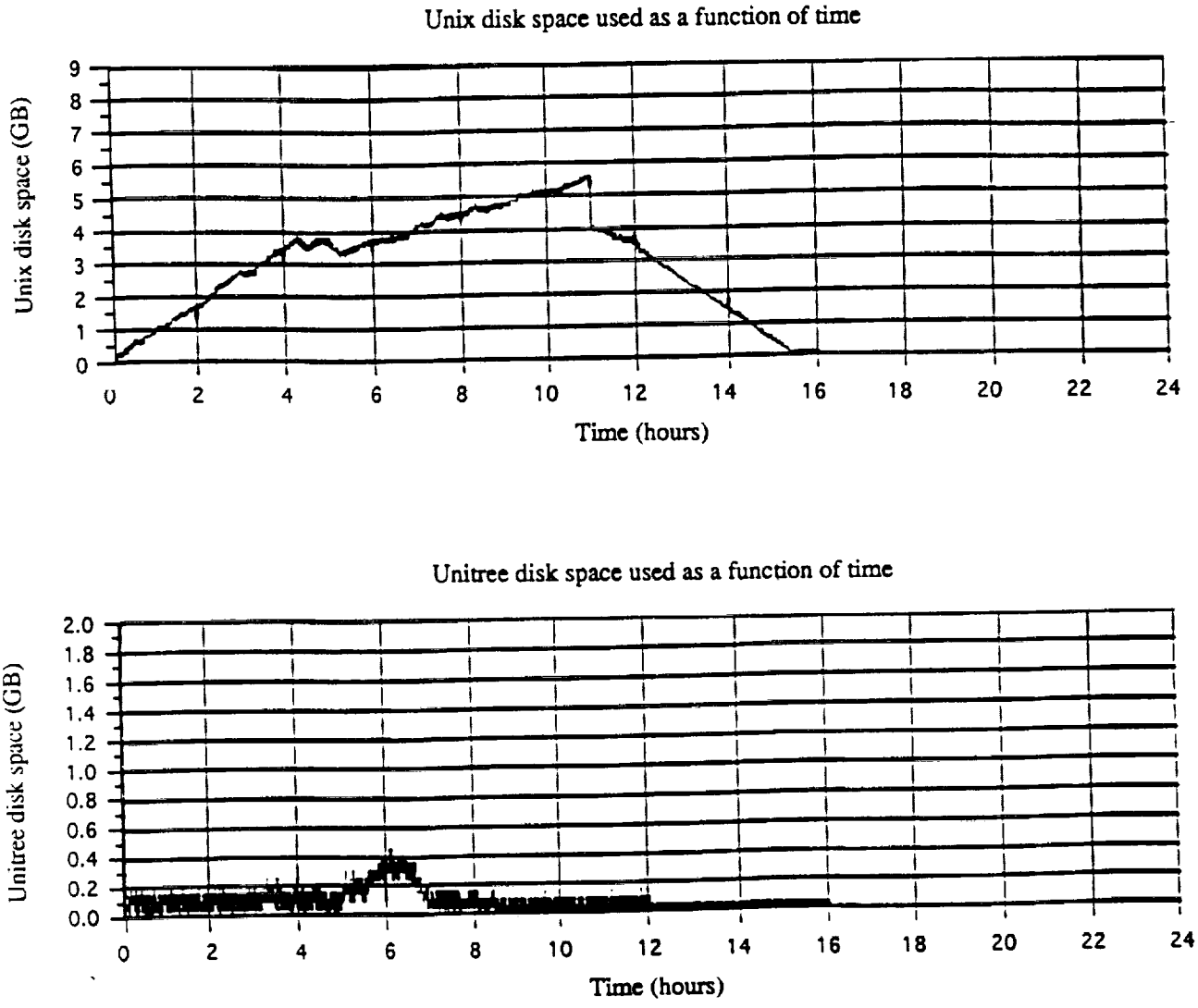



Fig 4. Disk space used in scenario 1-B

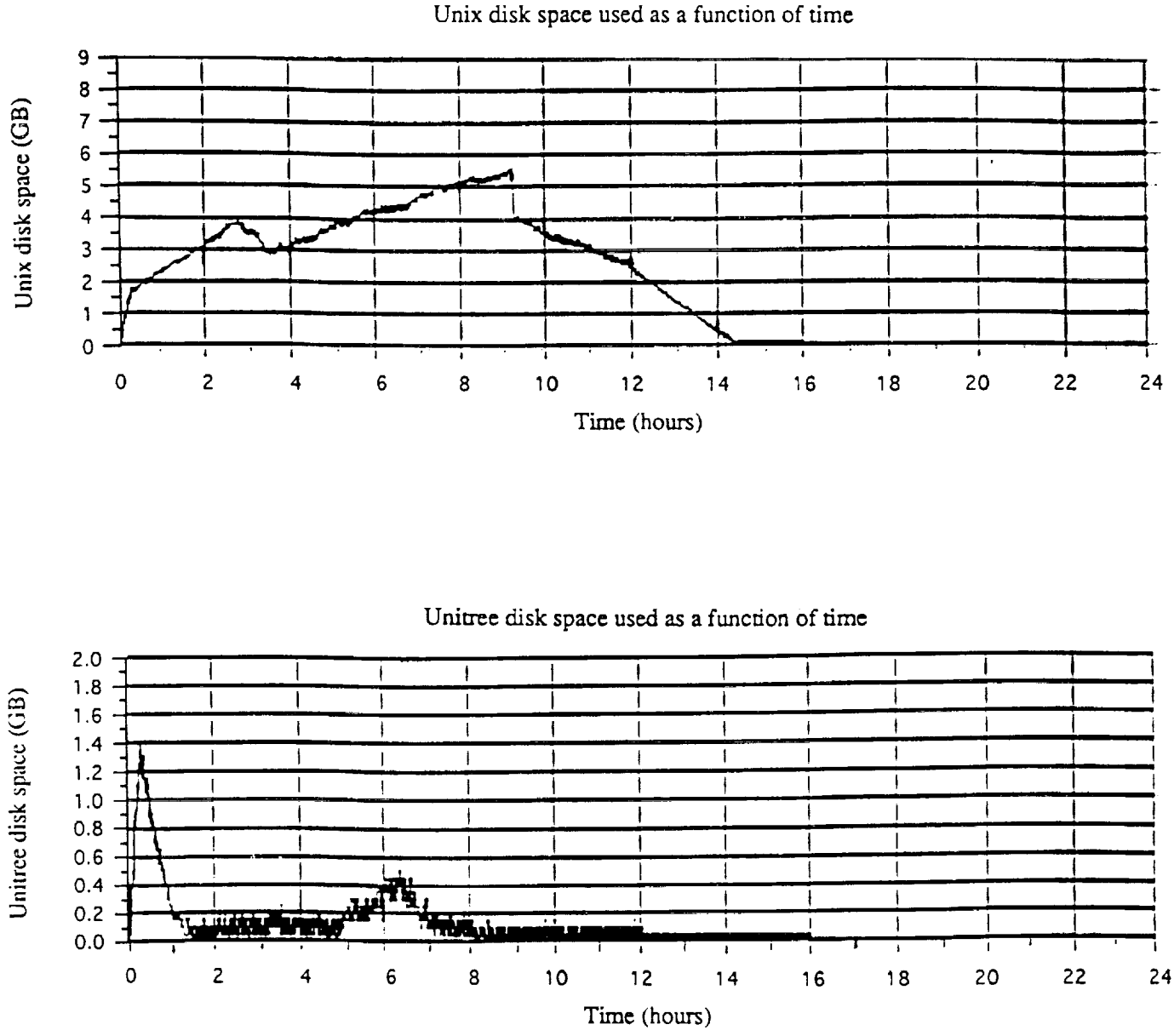



Fig 5. Disk space used in scenario 2

- Scenario 3: Effect of varying the proportion of distribution media requested.

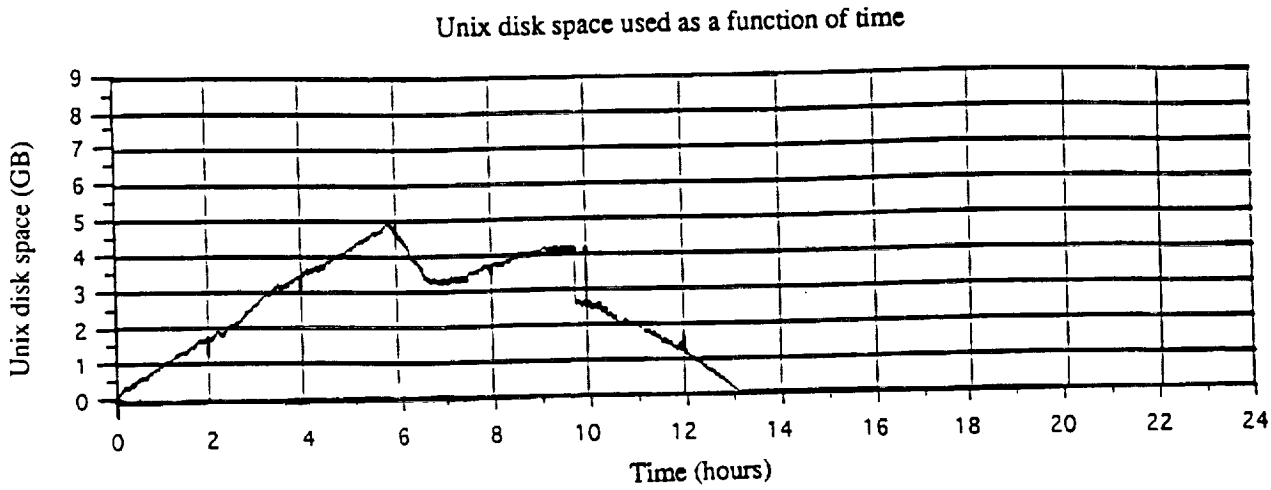
When the model was originally developed, it was assumed that 1/3 of the requests were copied on 8 mm tapes, 1/3 on 4 mm tapes, and 1/3 on 9 track tapes. A small survey of scientists indicated that a more realistic proportion of media requested may be 50% for 8 mm, 33% for 4 mm, and 17% for 9 track tapes. The DADS model has been simulated for the same volume of data ingested and distributed as before (in scenario 1-B) but with the new proportion of distribution media (Fig 6). The change affected only the distribution process and the Unitree disk space remained the same. When compared to scenario 1-B, the Unix disk space needed is smaller (5 GB instead of 5.5 GB) and the distribution requests backlog was eliminated sooner (13 hours instead of 15.5 hours). By changing the proportion of media requests, the volume of data copied to 8 mm drives increased. The 8 mm drive write rates are about 2 to 3 times faster than the 4 mm drives. Consequently it took less time to fulfil the requests and the data were staged in the Unix disks for a shorter period of time.

- Scenario 4: Effect of processing 2 days worth of data in one day.

In this scenario we analyzed the ability of the DADS to be unavailable for 24 hours and to recover in the following 24 hours. In order to examine this case, the model has been fed with two days worth of data. The doubled amount of data to be ingested and distributed, results in a substantial increase in the disk space required (Fig 7) for both the Unix disks (18 GB) and the Unitree disks (7.5 GB). It requires almost 26 hours to complete the requests, thus slightly exceeding the 24 hours planned recovery period. The fallover could be easily accommodated the next day. If the disk space available at the DADS is less than the amount specified as used in the simulation (18 GB for Unix disks and 7.5 GB for Unitree disks), the ingestion and distribution functions would require additional time.

- Scenario 5: Effect of the number of human operators at the DADS

Human operators play a critical role in the performance of the DADS, since it may take them several minutes, on an average, to respond to a request and fetch distribution cassettes. In the previous scenarios, the simulation assumed that there was no restriction on the number of operators and each of them took 1 minute, on an average, to mount or dismount a tape. In the simulation, this 1 minute average was represented by a uniform distribution from 0 to 2 minutes. After discussion with the DAAC operation staff, this 1 minute delay to fetch and mount was found to be too optimistic, based on their experience, and has been replaced by an average of 3 minutes (uniform distribution from 1 to 5 minutes). The proportion of media requested are assumed to be respectively 50% for 8 mm, 33% for 4 mm, and 17% for 9 track tape, and the number of operators has been varied from unrestricted to 1.
Table 6 summarizes the results of these tests. Table 6, case # 1, differs from the scenario 3 assumptions only in that the operator response/fetch time was increased from 1 to 3 minutes. This resulted in an increase in total disk space (Unitree disks and Unix disks combined) from 5.4 GB to 7 GB, and an increase in the time to eliminate the backlog of requests from 13 hours to 16 hours. In cases 2 and 3, restricting the number of operators to 8 or 4 has little effect on the results. In case 4, with only 2 operators, the total disk space required, and the time to complete the requests, both begin to increase noticeably. In case 5, with a single operator, the total disk space is large (14.5 GB) and, even after 30 hours, the requests were still not completed. Thus, the DAAC needs more than one operator to keep up with the daily workload.
In summary, with 2 operators instead of 1, there is a substantial decrease in the disk used (11 GB) and a significant improvement in the time required to fulfill the distribution requests (17.5 hours). Having more than 4 operators does not change considerably the disk space requirement or the request completion time.



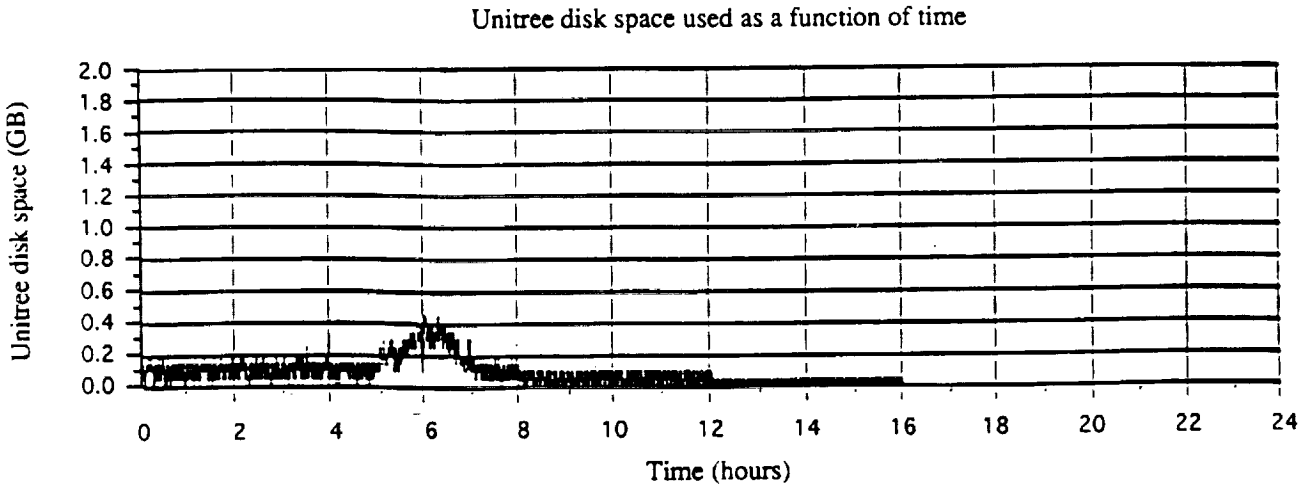
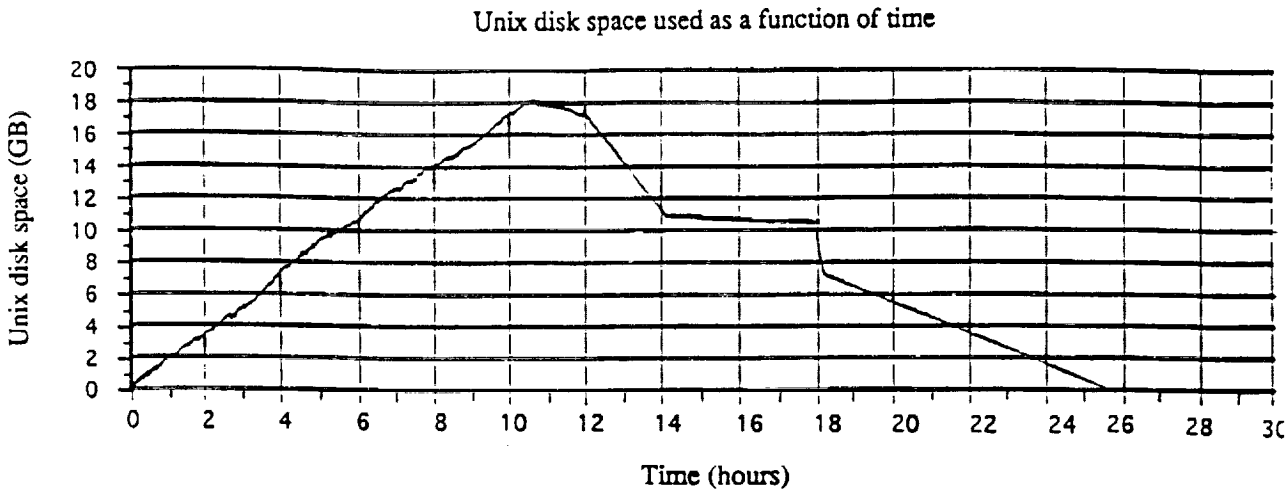




Fig 6. Disk space used in scenario 3



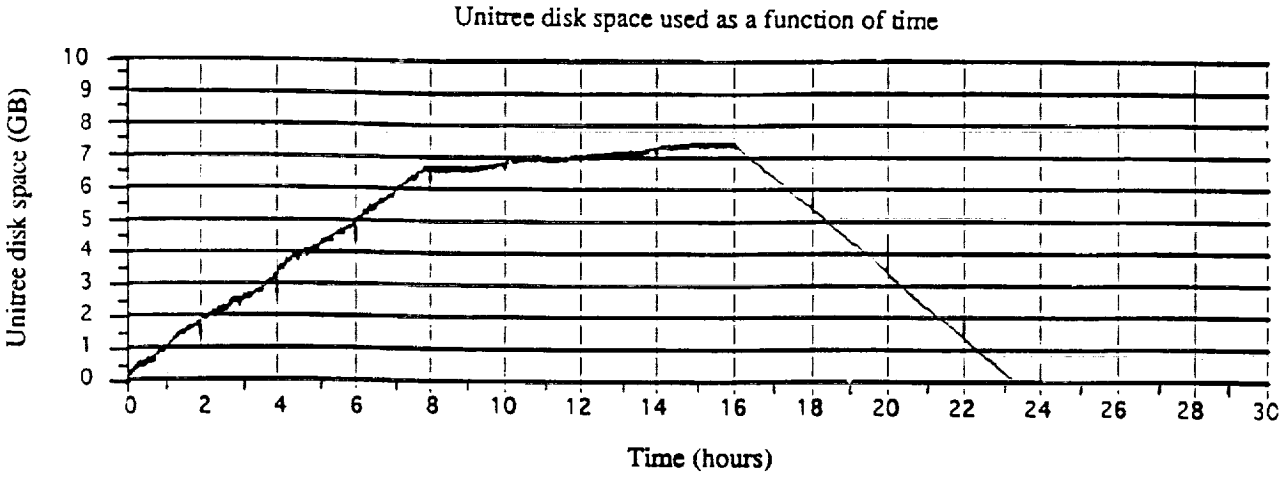




Fig 7. Disk space used in scenario 4

Table 6. DADS performance as a function of the number of operators

| case # | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Number of operators | unrestricted | 8 | 4 | 2 | 1 |
| Max disk space used (GB) | 7 | 7 | 7.5 | 11 | 14.5 |
| completion of standing orders (h) | 8.5 | 9.5 | 9.5 | 11.7 | 21.5 |
| completion of all requests (h) | 16 | 16 | 16 | 17.5 | >30 |

- Scenario 6: Effect of having stackers (4 mm and 8 mm) at the DADS

In order to automate the system and provide faster tape handling, the DADS model has been examined with one or several 4 mm and 8 mm stackers. It is assumed that the total number of stand-alone and stacker drives, for each type of drive remains constant ( four 8 mm, three 4 mm, and two 9 track ). There is a single operator to mount/dismount tapes from the stackers or the stand-alone tape drives.

The results obtained from the different cases examined are summarized in Table 7. Case 1 in Table 7 is the same as case 5 in Table 6 (i.e. one operator and no stackers). Adding a single 8 mm stacker and a single 4 mm stacker (case 4) has a significant impact on the performance of the DADS. The combined disk space required is reduced (13 GB instead of 14.5 GB), and the requests are completed much sooner (19 hours instead of > 30 hours) than when there is no stacker. As more stackers are installed at the DADS, the disk space used is decreased and the requests are finished over a shorter period of time. For cases 7,8, and 9, with 3 or more stackers, the results do not differ much from each other. In these 3 cases, the amount of disk space used is 9-10 GB and all requests are completed within 16-17.5 hours.

Table 7. DADS performance with and without stackers

| Case # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Number of operators | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| # of 4 mm stackers | 0 | 0 | 1 | 1 | 2 | 0 | 1 | 2 | 2 |
| # of 8 mm stackers | 0 | 1 | 0 | 1 | 0 | 2 | 2 | 1 | 2 |
| Max disk space (GB) | 14.5 | 13 | 14 | 13 | 11.5 | 9.5 | 9.5 | 10 | 9 |
| Completion of standing orders (h) | 21.5 | 19 | 18.5 | 13.7 | 16.5 | 15.5 | 12.25 | 12.25 | 10.7 |
| Completion of all requests (h) | >30 | 24.5 | 26.2 | 19 | 23.7 | 21.5 | 16.5 | 17.5 | 16 |

- Scenario 7-A: Effect of ingesting all data and distributing all data without compression and decompression.

In the previous scenarios, the model had been executed when all data were ingested and when all the SeaWiFS data were distributed. For this scenario, the estimated distribution workload of AVHRR, TOVS, UARS, DAAC climate, and CZCS have also been included (8 GB/day). Disk space used when all data are ingested and distributed is illustrated in Fig 8. Comparing Fig 8 with Table 7, case 9, indicates that the additional 8 GB/day distribution workload results in an increase of total disk space required ( from 9 GB to 16 GB) and takes longer to complete the SeaWiFS standing orders (from 10.7 to 13 hours). However, the time required to complete all requests (16 hours) is not changed.

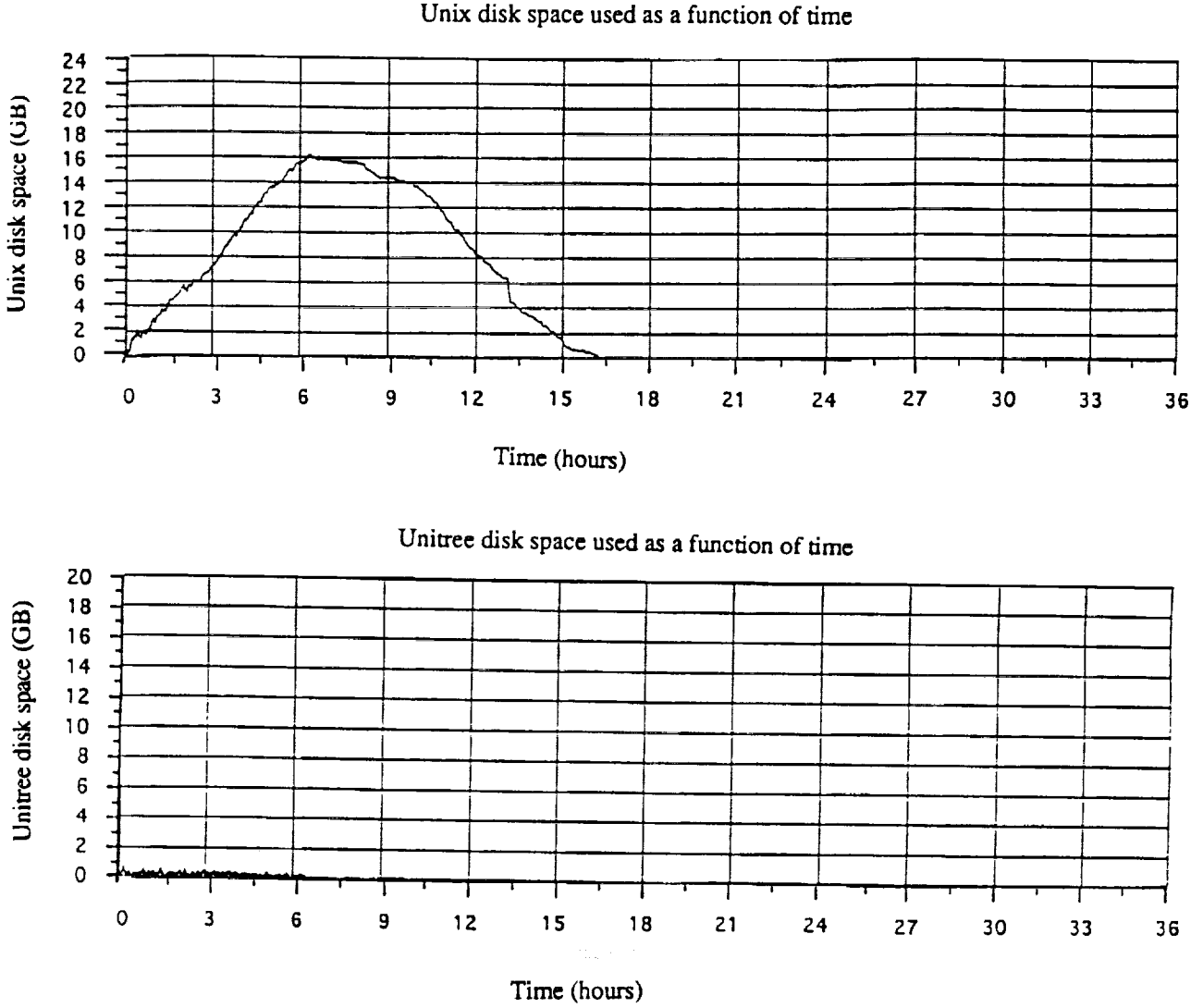
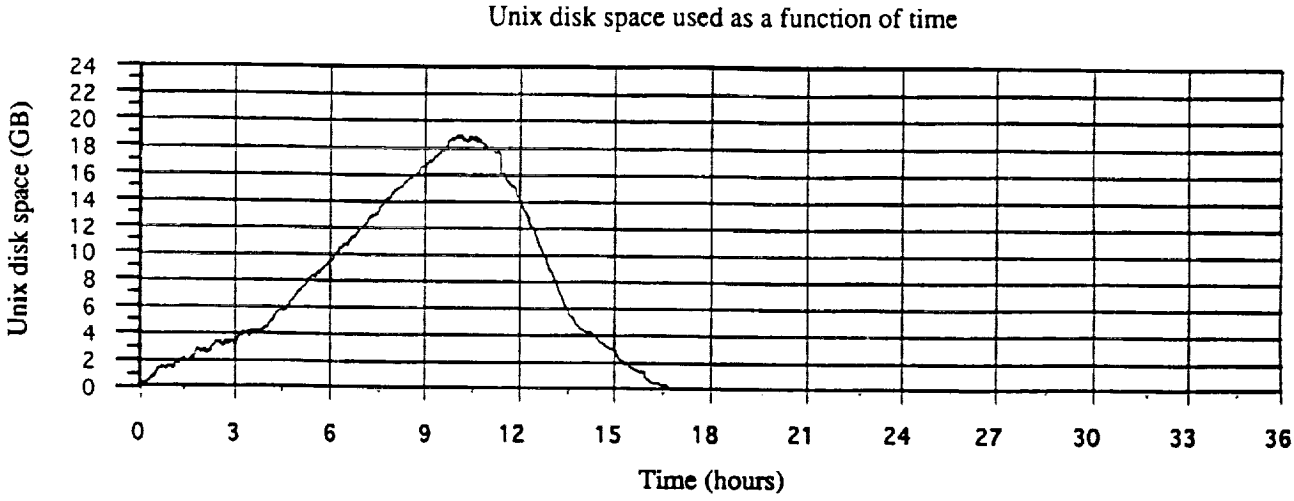



Fig 8. Disk space used in scenario 7-A

- Scenario 7-B: Effect of ingesting all data and distributing all data with compression and decompression.

The GSFC DADS is investigating the prospect of using data compression techniques to save storage space (1). Depending on the data set and the compression algorithm used, the original file can often substantially be reduced, thereby contributing to the mass storage solutions. However compression is a CPU intensive operation and can be rather slow if the compression is performed with software rather than hardware. The goal of this simulation is to estimate the impact on the DADS performance when using compression/decompression. In the DADS model the compression algorithm selected is the Unix compression (LZC), which does not have the best compression ratio, but is faster than the other algorithms evaluated and is a quasi-standard. The compression rate varies from data set to data set and, based on the results of the compression investigation, a compression rate of 200 KB/s was chosen for this simulation. It is assumed that all files ingested are archived in compressed form. The standing orders and the other distribution requests are sent to the users in an uncompressed form. With a slow compression/decompression rate it is expected that this may cause a bottleneck in the system.

The penalty for performing compression/decompression is indicated in Fig 9. The Unix disk space has increased from 16 GB to 19 GB, and the Unitree cache, which was under 1 GB, has now a peak of 10 GB, so that the total disk space is now 29 GB. The time to fulfill the distribution requests has increased slightly from 16 hours to about 17 hours. The large increase in disk space required is due to the slow compression/decompression rate assumed, which delays the ingestion and distribution processes, thereby causing data to build up on the disks. If the total amount of disk space had been constrained to less than 29 GB, the time required to fulfill the distribution requests would have been increased.



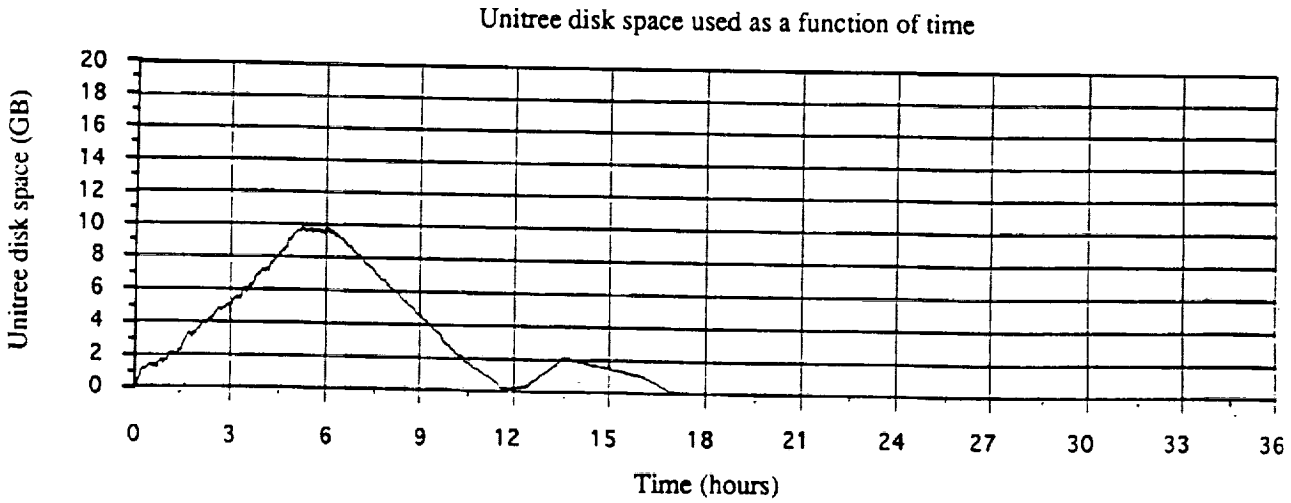




Fig 9. Disk space used in scenario 7-B

## Conclusion

The discrete event simulation model has proved to be a very useful tool to evaluate the performance of the V0 DADS. The amount of disk space necessary to fulfill the daily ingestion and distribution requests at the DADS has been estimated under various conditions. The model has demonstrated the importance of matching the ingestion rate to the archival rate to prevent data build-up in the Unitree cache, thus minimizing the amount of cache disk space required (scenario 2). Human operators play a critical role at the DADS since the time of about 3 minutes required to manually mount/dismount tapes is a limiting factor in the DADS performance. However, having too many operators (4 or more) does not improve the performance of the DADS (scenario 5). Stackers (4 mm and 8 mm) can substantially help in automating and processing the DADS requests more quickly (scenario 6). With a single operator, under the assumptions of scenario 6, a single 8 mm stacker reduces request completion time from > 30 hours to 24.5 hours. The combination of two 8 mm stackers and two 4 mm stackers further reduces the request completion time to 16 hours. The compression and decompression operations are very CPU intensive and, if performed at slow software rates, will require substantial additional disk space (scenario 7-B).

# Mass Storage - The Key to Success in High Performance Computing

**Richard R. Lee**

Data Storage Technologies, Inc.
Post Office Box 1293
Ridgewood, New Jersey USA 07451-1293
Phone: (201)-670-6620
Fax: (201)-670-7814
rrl@dst.com

**Abstract:** There are numerous High Performance Computing & Communications Initiatives in the world today. All are determined to help solve some "Grand Challenges" [1] type of problem, but each appears to be dominated by the pursuit of higher and higher levels of CPU performance and interconnection bandwidth as the approach to success, without any regard to the impact of Mass Storage. My colleagues and I at Data Storage Technologies believe that all will have their performance against their goals ultimately measured by their ability to efficiently store and retrieve the "deluge of data" created by end-users who will be using these systems to solve Scientific Grand Challenges problems, and that the issue of Mass Storage will become then the determinant of success or failure in achieving each projects goals.

In today's world of High Performance Computing and Communications (HPCC), the critical path to success in solving problems can only be traveled by designing and implementing Mass Storage Systems capable of storing and manipulating the truly "massive" amounts of data associated with solving these challenges. Within my presentation I will explore this critical issue and hypothesize solutions to this problem.

**Topics to be discussed:** To properly lay the foundation for this paper I must briefly discuss the history of Mass Storage in respect to high performance computing. Once these background materials are discussed, I will then focus the body of the presentation; "Current and Future HPCC Initiatives and the Impact of Mass Storage on Their Success or Failure"

The areas of background to be discussed are as follows;

i.- Basic Definitions and Key Underlying Factors
ii.- The Early Days of Mass Storage and its Role in Advancing the art of Computing
iii.- The current Role of Mass Storage in High Performance Computing & Communications

## Basic Definitions and Key Underlying Factors

Mass Storage per my definition is: "Any type of Storage System exceeding 100 GB in total size (not off-line), and operating under the Control of a Centralized or Distributed File Management Scheme.

CPU Power has Increased at a rate of 25% per year (CAGR) for the Past 10+ years, While I/O Bandwidth/Rates have Remained Constant. 1 MIP of CPU power should correspond to 1 MB/s, of I/O bandwidth performance. This fundamental relationship has not been adhered to since the early days of the mainframe, and is not found anywhere today in HPC.[1]

---

[1] The following are a partial listing of the HPCC Coordinating Offices "Grand Challenges" research teams projects; Computational Quantum Materials, High Resolution Operational Weather Forecasting, Numerical Tokamak, Multidiscipline Simulation of High Speed Civil Transport and Performance Aircraft, etc.

I/O and Network Bottlenecks, along with OS and other software inefficiencies are crippling all types of computing systems today, and jot just those utilized in the world of HPC.

There are no panaceas to solve the "Mass Storage Crisis" found in HPC today. A new paradigm in Systems Architecture and Design Philosophy is required to meet the requirements of future HPC environments. [4], [5], [6], [7]

As the "deluge of data" continues to grow (25+ CAGR)in the HPC data center, many end-users will be faced with the dilemma of not being able to store Critical Data due to increasing economic constraints. Not only is the Cost-per-MB of On-Line and Secondary storage too high, but the CPU cycles required for off-loading and accessing large files (Multi-GB) is quickly becoming unaffordable! [2], [5]

## The Early Days of Mass Storage and its Role in Advancing the Art of Computing

Early Mass Storage systems consisted of removable hard disk packs, and magnetic tape drives-freestanding or serving large off-line round tape repositories. These early peripheral based systems were augmented by unique, proprietary storage systems such as the IBM 1360 photo-store, the IBM 3850 helical scan tape library, the Ampex terabit memory , the Braegen automated tape library, and others from CDC, Remington Rand, etc. Although these early systems offered increased capacities over stand alone peripherals, none were commercially successful and most were sold into US Government labs or to the Intelligence Agencies.

Surprisingly thought, these early systems were much better matched to their accompanying CPU's I/O bandwidth than that found today and they truly did provide very good performance and value to the customer during their heyday, given the lack of practical alternatives.

## The Current Role of Mass Storage in High Performance Computing

Mass Storage systems today range in size from 100 GB to 30+TB, with all under the control of some type of dedicated File Server CPU. Most of these systems are; slow in performance, woefully under powered in terms of I/O Bandwidth, and utilize very immature Hierarchical File Management software schemes. These systems provide cost reductions in terms of storing a variety of bitfile data set types, but do very little to actually improve the performance of the overall system. This problem is further exacerbated by the divergence between CPU and Network Operating Systems (MVS, UNIX, OSI, etc.). and their fundamental differences in approach to the task at hand and the hardware interfaces supported.

All of today's' Mass Storage systems utilize dedicated, and very expensive components in order to optimize performance capabilities and most are based on technologies developed in the 1980's which are now just becoming commercialized e.g. RAID, HiPPI, FDDI, DD-2 , UniTree, etc. These systems will be the benchmark in the early '90's but will be replaced by radically new approaches scheduled to become available in the mid-'90's.[5], [9], [3]

## "Current and Future HPCC Initiatives and the Impact of Mass Storage on Their Success or Failure"

### 1.0 The HPCC Initiatives

High Performance Computing and Communications or HPCC has become the buzzword acronym of the early 1990's. In its simplest form it refers to Public Law 102-194 1991 The High Performance Computing Act/Initiative Of 1991, signed into law by President, George Bush (12/91). It is broken down into four constituent parts;

1.- TeraFlop (now referred to as "Teraop") Computing
2.- NREN (National Research & Education Network)
3.- Advanced Software and Algorithm Development
4.- Training & Research

In its most complex form HPCC is a catchall for every advanced computing activity in the world today. It has been widely promulgated as fundamental to the Clinton administrations' endeavors to improve the US's competitiveness and productivity in respect to Japan and Europe, and is deeply mired in party politics. Many new initiatives have been tacked on to the original legislation[2] and funding is anticipated to increase in out years regardless of the wrangling by each political party that continues to go on..

## 2.0 Mass Storage's Role in the Success or Failure of HPCC

In spite of its politicization, HPCC has provided a focal point for addressing all issues relevant to the future of computing. In monitoring this focus, it is painfully obvious that the issue of Mass Storage has been largely ignored, with the exception of the National Storage Laboratory @ LLNL and a few other small projects spread around the HPCC community. [7], [6], [2]

When the issues of "how to achieve" the levels of performance necessary to solve "Grand Challenges" scientific computing problems are addressed by all parties involved at conferences and symposia as well as in articles and abstracts and testimony to Congress; it is painfully clear that Mass Storage is forgotten altogether or minimized in importance in the grand scheme of things This is a critical error in my opinion.

As computing moves quickly towards client-server topologies in every imaginable application, the network will essentially become the computer. Numerous heterogeneous computing resources will be linked together over "data superhighways" (multi-gigabit links) to form large on-line computing capabilities. These systems will range from clusters of high-end workstations to numerous supercomputers in many locations linked together i.e. the NSF MetaCenter. These meta-type systems are touted as having the capability to finally begin to address some of the really difficult "Grand Challenges" problems that many believed could only be solved by Teraops type machines of the future (Table Number 1 lists the capabilities of many of the network topologies being discussed to form the '"data superhighways".) This approach has been widely endorsed as of late, but within those endorsements there is no mention of how these meta-type systems will store and manage the avalanche of data created by "the system", much less how one can practically afford the cost associated with the task.[3]

The NSF MetaCenter is one of these systems and will utilize the capabilities of some 21 supercomputers (vector, scalar & parallel), linked together over an optical network ( NSFNET). The amount of data to be generated by this system begins to boggle the mind, and yet is is treated as a secondary issue by many in the MetaCenter development group. What is clear is that when these types of systems are finally up and running is that they will all essentially swamp their local storage capabilities and that the data sets generated by the meta-computer will not be able to be stored and further manipulated due to cost, bandwidth and capacity constraints at every link of the "MetaCenter chain". This is a quandary not only for the meta-

---

[2] As of this writing, the following new bills and acts regarding add-ons to the original HPCC legislation are in process;

1.- "the National Information Infrastructure Act of 1993 (formerly known as "the High Performance Computing and High-Speed Networking Applications Act" - HR 1757
2.- "the National Competitiveness Act of 1993", HR 820, S.4
3.- "the Electronic Library Act of 1993", S.4 Attachment

[3] It has been said by many that current costs in the data center are split 50-50 between the CPU and the peripherals. This has been fairly accurate until recently when, scientific visualization and the use of more on-line archives has produced a phenomena where peripheral costs are now climbing to 60+% of the overall cost and we predict that in the future this may rise to almost 75% if not abated by a new paradigm in systems design.

computer types, but those involved in visualization, parallel computing and scientific activities such as CD, etc. The quandary is as follows: What makes more sense; to utilize the entirety of the data centers available resources (storage capacity and CPU cycles) to store the results of a complex computational problem, or to throw the data away and re-calculate the results on another day, often without the same results achieved or computational resources available? In its simplest form this quandary speaks to the fact that we have spent the last 20 years pursuing the Holy Grail of CPU power and speed, but cannot utilize it to its fullest capabilities, because we have nowhere to store the data!

**Table 1**

**Emerging Networking Standards**

| Network: | Type: | Data Rate(s): | Data Type(s): | Max. Distance: |
|---|---|---|---|---|
| ***Fast Ethernet*** | TP- Cu | 100 Mb/s | Digital | 25m |
| ***CDDI*** | TP - Cu | 100 Mb/s | Digital | 50-100m |
| ***FDDI*** | Opt. Fiber | 100 Mb/s | Digital | 60 km |
| ***FDDI-II*** | Opt. Fiber | 100 Mb/s | A, V & Digital | 60 km |
| ***HiPPI*** | TP - Cu | 800/1600 Mb/s | Digital | 25m |
| ***Fibre Channel*** | Opt. Fiber | 1000 Mb/s | Digital | 10 km |
| ***SONET/ATM/B-ISDN*** | Opt. Fiber | 51-2488 Mb/s | A, V & Digital | LD Network Limits |

In spite of it looming over the future of HPCC, the issue of Mass Storage is not insurmountable by any means. What is needed are new approaches to the problem and new storage devices capable of storing, manipulating and retrieving vast sums of data at faster speeds, with higher volumetric efficiency and will attendant incremental reductions in cost-per-unit stored.

Many of the storage technologies shown in Table Number 2 have been around for some time now, but have been recently adapted to offer orders of magnitude increases in capacity and bandwidth, while increasing volumetric efficiency (in terms of physical space utilized) as well as having unbefore seen low costs-per-unit of data stored.

**Table 2**

**High Performance Data Storage Devices**

| Name/Std.: | Storage Technology: | Data Rate(s): | Data Capacity: | Device Cost: |
|---|---|---|---|---|
| ***IBM 3490E*** | 1/2" Longitudinal OT | 4.5 MB/s | 500 MB (Native) | $70K |
| ***ANSI DD-1*** | 19mm Helical OT | 15 - 45 MB/s | 15, 50, 100 GB | $250K |
| ***Ampex DD-2*** | 19mm Helical MT | 15 MB/s | 25, 75, 186 GB | $200K |
| ***STK DD-3*** | 1/2" Helical MT | 15 MB/s | 20 GB | $65K |
| ***Metrum 2150*** | 1/2" Helical OT | 2-4 MB/s | 14.5 GB | $35K |
| ***CREO 1003*** | 35mm Optical Tape | 3 MB/s | 1000 MB | $250K |

These devices when wedded with robotics and advanced Data Management Software schemes can begin to meet the challenge of the MetaCenter and other such initiatives. They provide almost infinite capacity, with wide bandwidth (for time is money) and extremely low cost relative to the service that they are providing.

The issue of Data Management cannot be overlooked when challenging the "deluge of data" to be found in the future. This class of software and its influence on the systems architecture cannot be relegated to the role of freeing up more DASD, and therefore temporarily abating the data centers capital problems. (see Table Number 3 for a listing of currently available File System and File Management S/W) It must instead become the central director of all activities within the network and its attached resources (CPU's, peripherals, etc.). The orderly flow of data within the hierarchy of storage devices and networks will ultimately control the overall capabilities of the entire computational system.. The need for this class of software is made self-evident by the MetaCenter concept. Much attention is currently being paid as to how to break big problems up into large parallel pieces, but this effort will be futile if not supported by the Data Management S/W mandated by this type of challenge.

**Table 3**

**File Systems/Data Management Software**

| Trade Name.: | Developer(s): | Type: | OS Baseline: | OSI/IEEE Oriented: |
|---|---|---|---|---|
| *Network File System - NFS* | Sun Microsystems | F.S. | UNIX | No |
| *Andrew File System - AFS* | CMU/Transarc | F.S. | UNIX | No |
| *OSF DCE/DFS* | OS Foundation | O.S./F.S. | OSF/1 | OSI Model |
| *DataTree* | LANL/DISCOS | F.M.S. | MVS | Yes (early) |
| *EpochServ* | Epoch Systems | F.M.S. | UNIX | No |
| *DFSMS/DFDSM* | IBM Corp. | F.M.S. | MVS Family | No - Proprietary |
| *Open Vision UniTree V1.8X* | LLNL/DISCOS | F.M.S. | UNIX/NFS | Yes V3.0 |
| *NSL UniTree V1.X* | LLNL/IBM | F.S./F.M.S. | UNIX/AFS | Yes V5.0 Oriented |

## Conclusions and Recommendations

Mass Storage has become a critical path driver in the success of all HPCC Initiatives. To achieve the level of Systems Performance required to solve "Grand Challenges" computing problems, all elements of system must be optimized, with special emphasis on the role of Mass Storage in controlling the performance of the entire system.

The cost of storage will be a critical factor in determining the allocation of resources in the HPCC Initiatives. To meet the challenge, many orders of magnitude of cost reduction in -per-unit data stored must achieved. Part and parcel to these cost reductions will be increases in storage device bandwidths, volumetric efficiency and overall capacity. The hardware costs will be supported increasingly efficient Data Management S/W systems who manage and optimize the flow of data within the entire system.

I strongly advocate that Mass Storage and its attendant issues be brought to the forefront of the HPCC Initiatives. Only by applying this level of visibility and sensitivity to the issue will there be success in utilizing the HPCC Initiatives to solve "Grand Challenges" problems. Mass Storage can no longer be a secondary issue.

## References


1. Lee, R. and Dan Mintz, "Grand Challenges in Mass Storage - A Systems Integrators Perspective", Second NASA Goddard Conference on Mass Storage Systems and Technologies, Greenbelt, MD, September 1992

2. Lee, R., "The Future of Mass Storage", THIC Winter Meeting, San Diego, CA, January 1993

3. Lee, R., "Interfacing 19mm Helical Scan Recording Systems to Computing Environments", THIC Spring Meeting, Annapolis, MD, March 1990

4. Kuhn, T., *"The Structure of Scientific Revolution"*, University of Chicago Press, Chicago, IL 1970

5. Lee, R., 19mm Helical Scan Recording Technology for Data Intensive Computing Environments", 10th IEEE Symposium on Mass Storage Systems (vendor poster session), Monterey, CA, May 1990

6. Coleman, S. and R.W. Watson, "The Emerging Paradigm Shift in Storage System Architectures", review copy for Proceedings of the IEEE, April 1993

7. Coyne, R. , H. Hulen and R. Watson, "Storage Systems for National Information Assets", Proceedings-Supercomputing '92, Minneapolis, MN, November 1992

8. Lee, R., "Mass Storage - the key to success in high performance computing" (early version), Convex File Server Seminars, Milan/Rome, Italy, February 1993

9. Lee, R., "19mm Data Storage Applications", THIC Fall Meeting, Annapolis, MD, October 1990

N94-33815

# Storage System Architectures and Their Characteristics

**Bryan M. Sarandrea**

Advanced Archival Products, Inc.
6595 S. Dayton Street, Suite 1200
Greenwood Village, CO 80111
Phone: (303) 792-9700
Fax: (303) 792-2465
bryan@aap.com

## 1. Abstract

Not all users storage requirements call for 20 MBS data transfer rates, multi-tier file or data migration schemes or even automated retrieval of data. The number of available storage solutions reflects the broad range of user requirements. It is foolish to think that any one solution can address the complete range of requirements. For users with simple off-line storage requirements, the cost and complexity of high end solutions would provide no advantage over a more simple solution. The correct answer is to match the requirements of a particular storage need to the various attributes of the available solutions.

The goal of this paper is to introduce basic concepts of archiving and storage management in combination with the most common architectures and to provide a some insight to how these concepts and architectures address various storage problems. The intent is to provide potential consumers of storage technology with a framework within which to begin the hunt for a solution which meets their particular needs. This paper is NOT intended to be an exhaustive study or to address all possible solutions or new technologies, but is intended to be a more practical treatment of todays storage system alternatives.

Since most commercial storage systems today are built on Open Systems concepts, the majority of these solutions are hosted on the UNIX operating system. For this reason, some of the architectural issues discussed focus around specific UNIX architectural concepts. However, most of the architectures are operating system independent and the conclusions are applicable to such architectures on any operating system.

The problem:

The explosion of information storage requirements is being driven by more on-line data collection, data intensive applications such as imaging, government regulation over data availability and maintenance, and other needs. As this explosion takes place more and more users are realizing that disk (magnetic disk, DASD) is not an ideal solution for many reasons. Relative to other technologies, disk is more expensive, more prone to mechanical failure or data loss, requires increased administration, and other limitations.

Organizations are continually looking for solutions which meet their individual storage and retrieval needs which solve some of the problems associated with disk. However, disk has many advantages such as high transfer rates, random access, readily available file system interfaces and others. These advantages mean that disk almost invariably plays some role in meeting the storage requirements. The architectures discussed in this paper are all built around systems where disks play a primary role in the architecture. These are the most common solutions available and they leverage the good attributes of disk while utilizing alternative technologies to minimize the less attractive aspects of disk.

## 2. Criteria

This paper is forced to limit the scope of parameters discussed to only a few. In keeping with the stated goals of providing high level guidance, these parameters will be dealt with in general terms and as such should be useful as guidelines in evaluating or selecting technologies.

The parameters we will try to address are:

cost
performance
transparency of data access
administrative burden
distributed access

The conclusions reached on any given architecture will be arguable. For example, it is impossible to provide detailed performance information for tertiary storage systems. Nearly all systems discussed handle multi-user or multi-tasking environments where system throughput will vary by access patterns. A system with no contention and pre-staged media can provide nearly instantaneous response while the same system could require 2 minutes to begin providing data under other circumstances. Instead of providing details, an attempt will be made to present the issues. Only a detailed analysis or even evaluation period can determine actual performance under a given situation.

Costs will be given in relative terms along with some insight into the elements driving system pricing.

## 3. Terminology

The terms defined below are useful for understanding this paper. In no way do these definitions attempt to resolve the confusion over the correct use of these terms in the industry. Other industry terms will be introduced in context.

On-line:

Data which is on-line is accessible without human intervention.

Off-line:

Data which is off-line is inaccessible without human intervention.

Tertiary storage:

Storage which is accessible without human intervention but which is not RAM or directly addressable hard mounted media such as a magnetic disk drive. Tertiary storage devices in this paper will typically refer to removable media auto-changers such as optical disk jukeboxes or tape libraries.

Transparent Access:

Transparent access implies that a file can be accessed using the standard file system calls of the native operating system. Under transparent access, an application written to be capable of creating, reading, writing, etc. on the operating systems standard magnetic disk file system could perform these same functions on the tertiary storage without modification.

## 4. Non-Transparent Access Systems

While most storage solutions offered today are stressing transparent access, there are still many non-transparent access solutions on the market.

### 4.1 Backup Systems

Backup is only mentioned here for completeness. While backup systems are typically used in conjunction with any storage solution to protect from data loss in the event of failures in the storage system, in and of itself backup would not typically be considered a storage solution. Backup is, however, often used as simple archival mechanisms as referenced under "Simple Archive" below.

### 4.2 Simple Archival

For many applications and users, the usage characteristics of files (data sets) is either well known whereby a specific determination is made of which files should be archived, or specific user or application control over the process is desired. This environment could most easily be characterized as manual or demand archival. Typically the device used is a removable media device such as a cartridge tape drive which would require manual loading of the tape for retrieval. Some newer systems integrate the archival software with tertiary storage devices, providing unattended access to the files.

In some systems, certain data types are always considered archival data and are only retrieved to on-line devices when accessed. Not only is automated file/data management not necessary, it is often undesirable in these systems.

In order for these systems to operate effectively, mechanisms need to exist which support the storage and retrieval of these files to and from the archival system storage. Older systems accomplished this by providing specific archival functions which enable applications or users to copy file data from its current location to the archival system. Those systems typically would either provide an archival name or allow the user to specify an archival name. With these systems, an archived file could not be accessed in place by an application. Access required the file to be copied from the archival media to on-line storage. In addition, the archived files did not appear in the standard system file name space (file system). This required that the application or user remember the "archival name" of the file and learn new access methods.

Nearly all modern archival solutions will at a minimum, manage and track media volume allocation. This allows the non-transparent access simple archival process to be minimized to as few as two commands equivalent to "store the file" and "retrieve the file". Manual systems will typically then interface to a system administrator or the user to satisfy the media load function. The identification of the correct media is provided by the archival software which tracks the file to volume relationships.

There are many similarities between these archival solutions and backup solutions. The advent in recent years of simplified identification and retrieval of individual files from backup volumes, combined with the ability to specify individual files for backup, has all but duplicated the functionality previously provided by archival software. In fact, many backup vendors sell their solutions for both backup and archival purposes.

### 4.3 Automated Archival

Automated archival in these systems provides a function which automatically identifies which files should be archived. Typically such a function would be used to groom the on-line disk file systems for files which have not been accessed for long periods and/or which are large files. These procedures are often un-popular since the lack of transparent, automated access

combined with the lack of user control over what is archived can create various problems for both users and their applications.

## 4.4 Characteristics of Non-Transparent Access Systems

Clearly these solutions do not provide the benefits of transparent access to files. However, they can be very cost effective solutions when used with stand-alone tape drives or low cost autoloaders.

Performance of these systems ranges from essentially off-line to low performance since even automated systems will require restoration of the file from the archival media before access to any data can take place. These systems provide no ability to access the data without restoring the entire file, thus sufficient on-line storage capacity must exist in order to get access to the data. If sufficient space does not exist, it is up to the user or system administrator to move other files to create space or to find an alternative location for the file.

The administrative burden of these systems is placed on the user or application to determine which data should be archived. In addition, manual load system will require manual interaction to handle media load functions. Clearly data archived using these systems should be data which is very infrequently accessed.

# 5. Transparent Access Systems

The remainder of this paper discusses systems which attempt to provide transparent access to files held in tertiary storage. By transparent access, we mean that the access methods used to read or write these files is identical to those used in accessing the operating systems standard magnetic disk resident files. In order for this to happen, files on tertiary storage must be available in the file system name space and the standard operating system calls must be supported for access. There are numerous architectures available providing this functionality. Each architecture has been designed to provide certain features and functions. We will try to identify these as well as any trade-offs made by a specific implementation.

## 5.1 Virtual File Systems Interface (VFS)

In tertiary storage systems implemented on UNIX systems, the most common approach to achieving transparent access is to utilize the Virtual File System Switch as a mechanism for inserting new file system types or for extending the functionality of existing file systems. The VFS provides a standard interface to the internal OS calls dealing with file data. The purpose of this interface is to allow multiple file system types to co-exist within a UNIX kernel. Because the calls are well defined and adhered to by all underlying file systems, it is theoretically impossible for the users, applications, and network protocol packages to identify which file system is being accessed. This allows file systems to be added to the kernel which are designed to deal with the idiosyncrasies of the underlying hardware, transparently to the users.

FIGURE 1 shows how the VFS hides the file systems from the users, applications, and network protocols. The system calls to the operating system and the VFS call layer provide a set of standard interfaces which allow the introduction of new file system types for handling new functionality. The VFS is the facilitator which allows many of the architectures on the market today to provide the transparent access that is so desirable. In addition, since the underlying file systems are isolated from and invisible to the network protocols, systems implemented using this layer can be compatible with existing network software, allowing the vendors to utilize the standard protocol packages available from the OS and network software vendors. This minimizes the development efforts of the storage system vendors, reducing price, complexity, and cost.

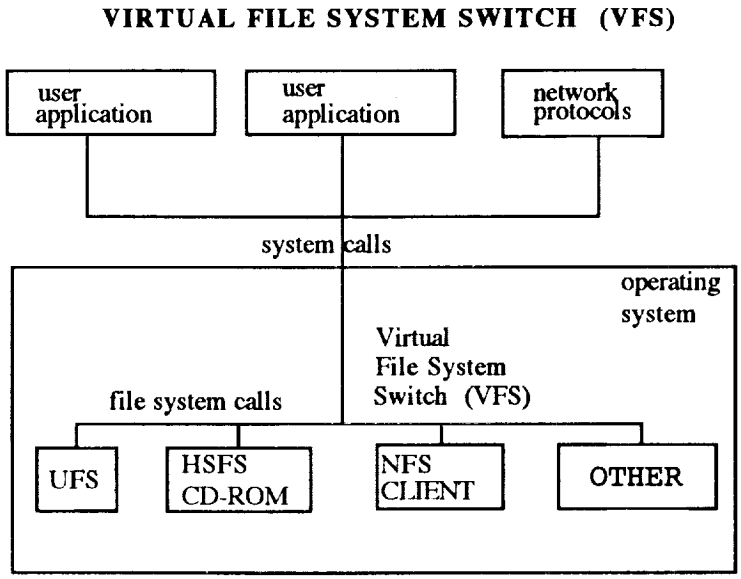



FIGURE 1

While the VFS depicted above is specific to UNIX, new operating system designs, including Microsoft's NT and most micro-kernels such as MACH are incorporating these concepts.

The transparent access systems discussed below can also be broken down into two classes of systems, those which try to provide automated data management and those which do not. Automated data management has come to be known as Hierarchical Storage Management (HSM). In HSM, the system attempts to manage the data, keeping it on the appropriate class of storage for its type and access patterns. A simplistic model of this is to move the least frequently accessed data from magnetic disk to tape as the magnetic disk fills up and more space is required. For vendors providing HSM, the goal is to keep the data management function hidden from the user. This is only possible if file retrieval is truly automated and the users access methods remain unchanged. Any solution will only be successful if the user population accepts it. Users are less likely to subvert the data management solutions if the solutions are designed well, make good selections, provide good performance, and do not interfere with their applications.

Systems which are not trying to attempt data management typically will utilize the native operating systems ability to specify the device as part of the file name specification. In UNIX this is done by allowing the device to be "mount"ed into the file system tree. From then on, data can be created or accessed on that device by using the path name which includes the mount point of the device. Through these methods, users and applications can control the device on which their data resides while maintaining system usability by having identical access methods to those of the primary data storage devices. While no industry accepted terminology exists for this class of tertiary storage systems, we will call them "direct access" systems in this discussion.

Even within these sub-classes there are a number of possible architectures which can be found. These architectures will be discussed below in order to see how they impact the criteria which we are trying to address here.

## 6. Direct Access Systems

These systems share many attributes with archival systems. The user or application of direct access systems makes the determination of which device(s) the data should reside on. Unlike

archival systems, however, the data files can be created directly on the tertiary storage (or copied there) using the standard file access methods of the host operating system. Some vendors call these systems "Direct Access Secondary Storage" since the data is created and accessed in place on direct access devices with random access characteristics.

Direct access systems exist which will cache the most frequently used data on high speed devices in order to provide better system performance. Such systems will typically allow repeated accesses to data to be satisfied from the cache. These systems differ from data management systems in that the primary copy of the data is on the specified tertiary storage device, while a copy of the data may exist on a high performance cache device only when it has been recently accessed. Data management systems, on the other hand, would have the primary copy on the high speed device and would only create a copy of the data on the tertiary storage device when the primary copy is about to be deleted. The sophisticated caching methods, of these direct access systems, blur the distinction between direct access and data management systems.

Direct access systems are particularly useful in environments where users or applications need control over data location or where the best location of data can be better determined by the user or application. Such situations may include:

Secure environments

The applications or users require deterministic location for data integrity, performance, transportability or other reasons.

Large file situations where the files or the aggregate size of the files being accessed at once do not fit on the available magnetic disk storage.

Situations where large data streams would force a data management system to purge all recently accessed data to handle the creation of the resulting large data files thus overriding the benefits of the data management algorithms.

System contains primarily small files where the overhead of tracking the data management function would mitigate the savings potential of tertiary storage.

Environments where the cost of storage is an overriding concern over the performance of repeated file access.

## 6.1 Server Based Direct Access Architectures

In a server based system, FIGURE 2, the tertiary storage is connected to a system which is responsible for allocation of drives, loading of media, and movement of data to/from the drives.

In these systems the data for all storage or access functions flows through the server architecture. Thus this architecture can create certain bottlenecks in the system. Careful consideration of the software architecture must be given in order to allow the simultaneous operation of the library and each of the drives to maximize total system throughput. In addition, the server must be closely matched to the performance requirements of the application and the hardware devices. A server handling 4 optical disks capable of 1 MBS each is clearly a different class of machine from the server which would be required to handle 4 tape drives capable of 20 MBS each.

In some environments all data processing, at least for a given set of data, is done on a single machine and no distributed access to the data is required. A direct access tertiary storage system for such an environment would be typically be configured similarly to a Server Based.

### 6.1.1 General Characteristics of Server Based Architectures

Server Based systems implemented as shown in FIGURE 3 have the benefit of providing the transparent access discussed in relationship with the VFS. Such systems can support local file access or remote file access using the OS vendors protocols or third party network protocols available for the host computer.

Some Server Based systems are available which provide transparent access by re-implementing specific network protocols. These implementations will typically limit the flexibility, since most only support a limited number of protocols.

A Server Based system will usually provide acceptable performance and functionality in any environment where file servers are now used to provide distributed access to a single shared name space. However, since the data stored on tertiary storage is often less frequently accessed or it is archival data, the applicability of this architecture can be far greater than those served by file servers alone. Often it is the data rate performance that prevent this type of architecture from being used. For example, if the system was required to support multiple simultaneous 10 MBS data streams to/from high performance helical scan tape drives, it may be difficult, or too expensive, to configure a single server to handle the requirements.

This architecture is particularly cost effective. Since direct access is being used, the large front end disk farm usually associated with HSM systems is avoided. Further, since the data rates of many of the devices used in tertiary storage systems is not particularly high, an inexpensive server class machine is usually quite effective as the underlying host hardware.

Administrative burden is variable with the specific implementation and is treated below where specific implementations are addressed.

The distributed access, server approach used, particularly for VFS based systems, allows these servers to be utilized in a very broad range of applications and heterogeneous network environments.

SERVER BASED DIRECT ACCI

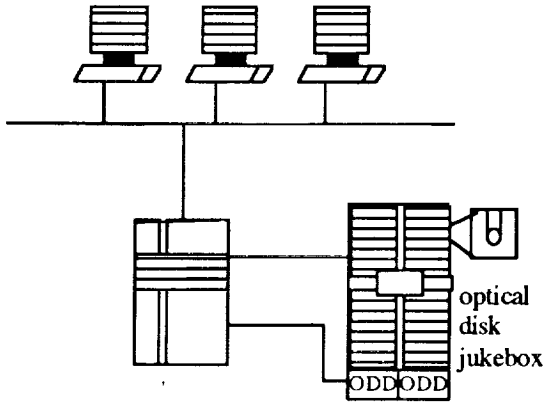



FIGURE 2

### 6.1.2 Direct Access Using a Magnetic Disk File System

Since a number of the devices used in removable media storage libraries have similar characteristics to magnetic disks, many vendors provide system solutions which use magnetic disk file systems to provide identical access methods for the tertiary storage. FIGURE 3, below,

shows a UNIX system architecture which implements this approach and a alternative file system approach. As shown in this optical disk jukebox system, the standard magnetic disk file system can be used to write the on-media format of the optical disks in the system. An additional device driver, the jukebox device driver, is layered between the file system and the device driver which operates the optical disks. This allows the block I/O requests to be intercepted so that the correct media for the given request is loaded before actually issuing the I/O request to the optical disks drive(s).

In particular, this method is used by a number of vendors for optical disk devices. The implementation of the device drivers is relatively simple to the implementation of a special file system and some compatibility with the standard OS's file system is achieved. However, this approach can not be used for media types which are not substantially the same as magnetic disk, e.g. WORM optical disks or tape devices since the file system is designed for random access of re-writable media.

### 6.1.2.1 Characteristics of Magnetic Disk File System Implementations

As mentioned above, one of the limitations of this approach is that certain types of devices and media types cannot be used. This stems from the fact that by re-using a file system written for magnetic disks, we are restricted to operating with device which can emulate the magnetic disk.

These systems also suffer from restrictions in the design of the file systems used as follows:

> The file system typically cannot cross media boundaries. This means that a minimum of one physical file system will exist per media volume. For two sided media, this will usually create two physical file systems per volume. This can be a burden for the system administrator who must now allocate space in much smaller fragments and potentially move data sets around the volumes as volume space becomes scarce. This will also typically limit the maximum size of files and will prevent files from being created in file systems and directories when media volumes become full.
>
> The file system does not know that the underlying media is removable and will schedule block I/O in random fashion, potentially causing thrashing of the jukebox. In addition, since the media is removable, standard file system sync mechanisms will not function correctly and system crashes may cause extensive file system damage.
>
> The caching mechanisms are also designed for non-removable media and may be inadequate for the long delays associated with loading and unloading media in a multi-user or multi-tasking environment.

These factors are typically weighed against system software cost. Systems which are implemented around custom file systems designed to solve these problems require significantly more R&D and typically carry a higher price.

Performance of these systems is typically acceptable if the above problems can be avoided. Thrashing in particular will drive system performance down. In this situation, the random block I/O patterns keep the jukebox constantly changing media volumes and very little I/O actually gets done.

## 6.1.2 Direct Access Using Alternative File Systems

As seen above, it is possible to use the embedded standard file system when the devices used have characteristics similar to a magnetic disk. But what if the vendor is interfacing non-standard devices or wishes to solve some of the problems pointed out in the discussions above. This is where new file systems have been introduced by several vendors. The VFS layer is called into play to allow a new file system to be added into the architecture where "alternative file system" is identified in FIGURE 3. This approach has allowed some well known file

systems to be added, as shown in FIGURE 1, which are considered to be a "standard part" of the UNIX kernel, but which in fact are really add on file systems for dealing with different device types or even networks. For example:

> High Sierra File System (HSFS). Most UNIX systems implement this CD-ROM file system under the VFS layer allowing the CD-ROMs to be mounted as a separate file system type.
>
> Network File System (NFS). As a network file system connecting two hosts, the NFS implementation consists of two pieces. The NFS client software which makes a remote file system appear as local is implemented under the VFS as a unique file system type. The NFS server software is implemented as "application code" which makes calls into the file system services of the VFS just as does any other protocol. This allows the server code to make any file system under the VFS umbrella available to the network.

These are just two examples of well known VFS implementations. Just as the HSFS implementation was designed to handle the differences associated with CD-ROM, third party tertiary storage solutions using this implementation are free to implement whatever mechanisms are appropriate for the device being integrated. The designers of these systems have much more flexibility in handling the idiosyncrasies of the various devices and can add custom scheduling algorithms, add additional caching, adjust the on-media formats, etc. as needed to provide good solutions for these devices. Packages are available to deal with re-writable and WORM optical disks as well as tape devices.

DIRECT ACCESS USING VFS

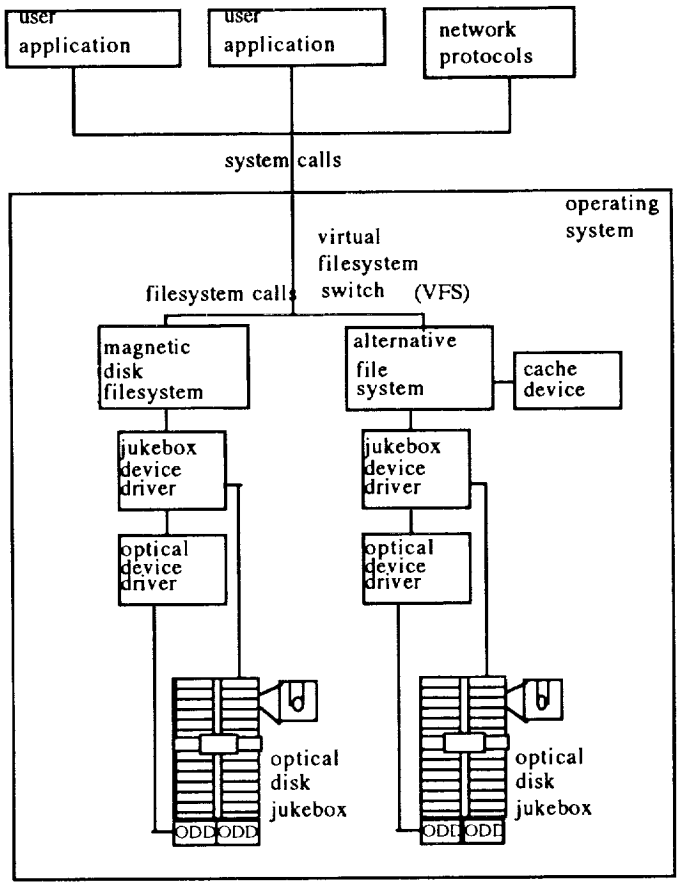



FIGURE 3

### 6.1.2.1 Characteristics of Alternative File Systems Implementations

The virtual file system approach has allowed vendors to directly address the unique characteristics of removable media devices while providing a standard file system type interface, making these devices appear as magnetic disks. The flexibility of this approach is born out by the availability of UNIX style file system interfaces for tape libraries as well as optical disk jukeboxes with WORM media.

The cost of these solutions is typically higher than others discussed thus far. This is due to the extensive software development effort required to write a complete UNIX file system. In addition, many of these file systems provide features not found in traditional file systems and invest a great deal in tuning the file system designs to obtain the best possible performance.

Additional caching is available in most of these systems. This caching allows the systems to provide a higher overall system performance, particularly in mutli-user and multi-task access environments. Here the cache can be used to implement read ahead and write behind algorithms which reduce wear on the auto changer hardware and increase overall system throughput by minimizing the number of volume exchanges required.

Some of these solutions also provide the ability to concatenate the media, thus providing a single file system view of the entire system. This can greatly alleviate the system administrators burden when dealing with space allocation. Single file system views also typically provide full file size support and eliminates the problems associated with full media volumes discussed above.

## 6.2 Direct Access with Network Attached or Switched Peripherals

In order to provide high data transfer rates and avoid the potential bottlenecks of server based systems, several vendors provide access to the tertiary storage peripherals through a high speed network, e.g. fibre optics, or high speed switch such as HPPI. Such a connection allows the data to be transferred direct from the data storage device direct to the requesting station. These systems are only employed in environments where distributed processing and very high performance distributed data access is required. There would be no need for such a system if all data was accessed or processed on the server.

WITCHED OR NETWORK ATTACHED PERIPHERALS

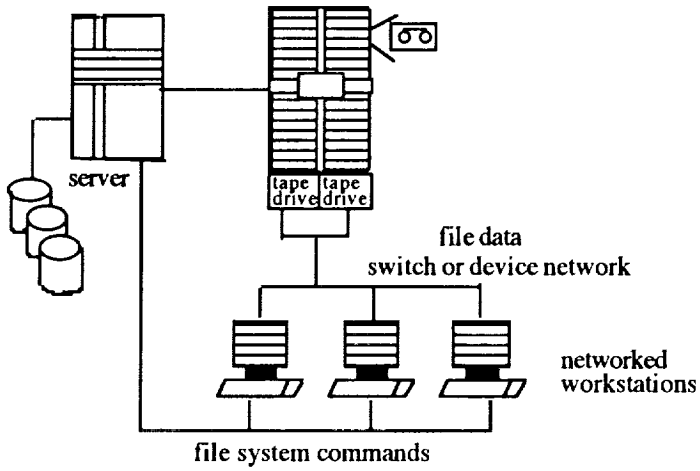



FIGURE 4

In these systems, in order to support shared access to the files stored on the tertiary storage, a central file system name space must still be maintained. This central processor is then also responsible for allocation of the drives, and operation of the library. In a system like this caching can get very complicated if shared simultaneous access across multiple processors is supported. In fact, the problem becomes identical to the caching problems addressed in such distributed file system implementations as DFS and NFS. Since the tertiary storage system has now become not only a storage subsystem but also a complete distributed file system implementation, the products tend to be incompatible with other network protocols and much more complex to administer.

If the files are not shared, each system on the network may maintain its own file system information and provide specific volume load requests to the library controller. Since each system tracks its own files, the files are not available to other users on the network. This model looks much like the server based implementation, without other network users, except for the fact that the library and devices become shared resources.

The benefits of these systems are that once the data is located, the transfer takes place directly between the peripheral device and the users processor. With a high speed connection, this configuration is much more capable of achieving the high transfer rates that are available on todays latest tape drives. Typically these systems are found handling the high speed data streams associated with todays super-computers. Since todays workstations are incapable of driving a 20 MBS tape drive at full speed, using these configurations on a network of workstations will not typically provide enough benefit to justify the increased complexity and cost.

### 6.2.1 Characteristics of Network/Switched Peripherals

Since direct peripheral device access and control is required to implement this architecture, custom software must be loaded onto both the server and each of the systems requiring data access. The architectures presented thus far have avoided this by utilizing exiting network solutions to provide the data access. This add a high degree of administration and support overhead to these systems. It also requires the implementation of custom data transfer protocols and control protocols by the software vendor.

Since the peripheral device control is turned over to the remote machine once the media volume is loaded, it is also not possible to share the peripherals. Once allocated, a drive will be dedicated to the remote machine until all of its I/O is complete. Therefore, although the I/O is most likely happening at a much higher rate, there is no ability to use caching as a solution to providing mutli-user access to the tertiary storage.

The extra software components, hardware components, and complexity of these systems relegates them to environments where very high performance transfers are required. This is usually associated with real time data streams or super computer centers in combination with very fast peripherals such as D1 or D2 tape drives. These same issues also mean that these systems are perhaps the most expensive to configure.

The requirement of dedicating access to drives and media volumes in this configuration makes comparison with cached systems difficult. Since contention can create significant delays in allocation of the drives or access to a particular media volume, the time saved in data transfer must more than compensate for the ability of a cached server type system to provide shared interleaved access and cached data. This implies that these systems also work best in environments with very large data sets.

## 7. Hierarchical Storage Management (HSM) Data Management

HSM or data management systems attempt to provide automated administration of data such that data is safeguarded against loss, shared access is provided where necessary, and large data repositories are managed in a performance/cost tradeoff fashion which minimizes the costs of

keeping the data available. Most HSM systems provide a view of the managed data that makes all data appear as though it is magnetic disk resident. Thus, the magnetic disk systems act as the primary storage device and, in fact, tertiary storage resident data which is accessed is typically moved back to the disk in order to provide access.

Unlike Direct Access systems, HSM systems will nearly always provide a first tier of storage which is a magnetic disk file system. The HSM then controls whether, and when, files are moved from the primary storage to tertiary storage. This is often referred to as "file migration", another name used to specify an HSM system. Users are given some level of control over the factors used to determine which files are migrated, however, it is the migration software which will perform the move automatically. Typically files are moved from the primary magnetic disk storage to the tertiary storage in order to maintain free space for new files on the magnetic disks. When a file which has been moved is accessed, the file is moved back to the magnetic disk for access.

Much like direct access systems, HSM systems can be configured in a large variety of architectures to meet various needs. A typical HSM system today would have only two tiers of storage, being magnetic disk and an optical disk or tape library. However, more vendors are now offering multi-tier systems with three or more tiers of storage. For example a system might move a file from magnetic disk to an optical disk jukebox and then if still not accessed after an additional time period move the data again to a tape library.

### 7.0.1 General Characteristics of HSM Systems

HSM systems can be very complex. The analysis of HSM systems requires an in-depth look at the architecture used and some general issues concerning HSM and how the systems and data are used. However, it can be generally stated that HSM systems can provide significant savings in storage costs when used in the right environment. In addition, the automation of the data management function frees system administrators from the chores of managing disk space and data archives.

Much like Direct Access systems, the performance, flexibility, network compatibility and other issues are determined by the specific packages architecture. We will try to address these below.

In evaluating HSM systems there are several hidden aspects to be aware of. First, HSM performs its function by being aware of which files have been used recently and which have not. For this reason it is imperative that nothing on the system destroys this information or the HSM system will not operate correctly. However, it is common for backup software to routinely go through file systems "looking" at if not accessing all files. This can cause the HSM system to see all files as recently accessed.

Some vendors solve this problem by providing special local and client/server backup packages which can work with their HSM solutions to solve the problem. With these vendors, the customer may be locked in to that backup solution, in which case it is imperative that the backup package also be evaluated as to functionality, cost, performance, etc. since it is the only backup package which can be used.

Other vendors solve this problem by making their HSM software compatible with third party backup packages. This is usually done by providing a framework within which to run the third party package such that it does not destroy the vital information needed by the HSM system. With such a system, the entire market of available backup packages can be used, leaving more flexibility for the consumer.

In addition, there are other backup issues;

How do you prevent the backup package from migrating in all files on a "full" backup.

How do you restore a migrated file.

What happens if you restore an old backup tape on a client. Are the references to migrated files still accurate.

Select an HSM package which has addressed these issues in its design.

HSM systems also impose additional storage overhead on files they migrate, and sometimes even on files which are not migrated. An HSM system which puts a file on tertiary storage will potentially require the following storage:

An inode to track the migrated file,
A data block for information about the file,
A local database entry for additional information,
A database entry on the tertiary storage system to track the files location,
An inode on the tertiary storage system for identifying the data file,
and of course the data file itself.

Look at how much storage is used for migrated files and how much disk is required just to set up the file migration. For most systems it is not economical to migrate small files.

It is also important to determine whether there are functions on your system which will defeat the data management system. For example, does anything periodically sift through your file systems reading files which would cause all files to migrate back. If users use the "file" command, does the HSM system allow for that without causing all files to migrate in.

For systems that are composed primarily of very large files, HSM may also be a bad choice. If each file access causes some other file to migrate out to make room to migrate this file in, then a direct access system may be a better choice. For example, a 2 GB disk HSM system where all files are 200 MB and which has 11 simultaneous users will thrash the system trying to get all 11 200 MB files read in at the same time.

The following are some features to look for in HSM systems.

File data is available as it is migrated in, as opposed to waiting for the complete file to be resident on the magnetic disk.

It is possible to determine if a file is resident or non-resident.

Manual migrate in & out is possible.

The system supports high and low watermarks for controlling available disk space.

The system supports pre-migrated files which can have their space freed up quickly.

The system catches "out of space" and begins "demand migration" creating space instead of returning an error.

Files migrate in both directions, some system migrate out and only provide Direct Access to these files after being migrated.

The software can manage pre-existing file systems. File systems are compatible with the native operating systems file system software.

### 7.0.2 VFS Versus Non-VFS HSM Implementations

The implementation approach taken implies a great deal about the functionality of the package. In HSM systems we find, as we did in the Direct Access discussions, that some

implementations utilize a VFS approach while others re-implement specific network protocols. These later system suffer from the same flexibility issues, in that all protocols or access methods supported must be re-implemented and cannot build on the existing system capabilities. These systems may also suffer from performance degradations associated with trying to implement file system type functions as application level software. In the UNIX community, the non-VFS implementations are nearly always Server Based HSM only and do not support the Client/Server HSM model. One benefit of non-VFS systems is that the opportunity exists to use mechanisms other than traditional inodes to track files. This could alleviate some of the duplicated overheads and prevent file systems from becoming full by running out of inodes.

Those systems implemented as VFS code will typically either overlay data management onto an existing file system or insert a file system with data management capabilities. The former approach allows the use of the vendors standard magnetic disk file systems. By layering the data management function over the existing file system, it is possible to maintain complete compatibility with all file system utilities and other software.

The approach of adding a unique file system, whether through the VFS or not, means that the resulting on-media formats will be incompatible with the vendors and a complete set of separate file system maintenance utilities will have to be used.

## 7.1 Server Based HSM

In a server based HSM configuration, FIGURE 5, all managed storage is centralized on a file server. These systems utilize file transfer protocols such as FTP or distributed file systems such as NFS or DFS to provide decentralized file access services. Any data not resident on the server cannot be managed by the HSM solution.

For network environments where centralized storage is already in use with network file servers and where the network clients are usually diskless, this type of system fits well. The HSM system will continue to manage storage as a central resource and the tertiary storage is shared as a function of data management of the shared disk farm.

### 7.1.1 CHARACTERISTICS OF SERVER BASED HSM:

The performance of Server HSM varies with the implementation. A good implementation should have a minimal impact on the performance of the magnetic disk file system on non-migrated files. In fact, the managed file system should perform with less than 1 to 2 % degradation.

It is usually possible to add Server Based HSM to an existing file server and obtain acceptable performance, assuming that the performance was already acceptable. These systems should place a minimal amount of load on a functioning system.

The cost of Server Based HSM should be close to that of Server Based Direct Access systems of the VFS style. The level of complexity of the two products is roughly equivalent. However, prices will reflect the type of tertiary devices being supported as well as the type of server.

The administrative load of these systems should be quite low. In fact, most vendors claim the reduction in system administrator load as one of the cost justifications of HSM systems. However, there is certainly an initial setup and learning curve on systems this complex.

SERVER BASED HSM

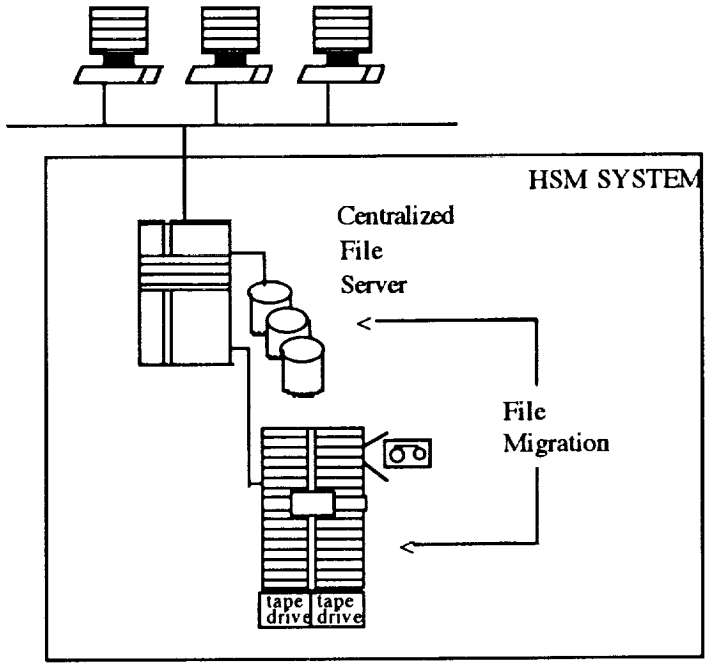



FIGURE 5

## 7.2 CLIENT - SERVER HSM:

Client Server HSM, FIGURE 5, provides the ability to manage data at both the client level as well as at a server level. By creating a client-server network interface, the client HSM software can manage local data repositories and utilize the storage capacity of centralized tertiary storage devices. This allows the most recently accessed file data to be moved close to the client by moving it to the clients local disk while sharing the high capacity lower cost storage devices for the data which has been migrated from the primary storage.

CLIENT - SERVER HSM

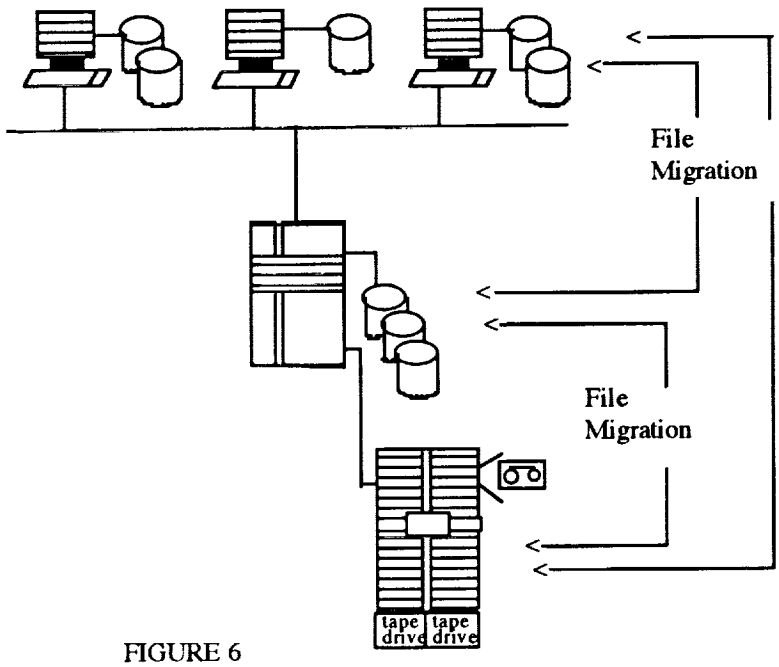



FIGURE 6

The primary consideration when deciding on a Server Based or Client - Server configuration is the impact on accessing the file systems. Most HSM solutions, in order to provide high performance in combination with transparent access, keep some type of reference to the migrated files on the disk, and the file system, where the file was originally created. When using Client - Server configurations this implies that in order to get distributed access to each of the diskfull clients files, their individual disk file systems must be "exported" to be made available for access from other clients. This in effect makes each client a server as well if shared access to each clients files is required. However, for environments where shared access is not required, this method can provide data management to each diskfull machine, high local file system performance, and access to shared inexpensive tertiary storage.

Since the Client - Server architecture typically supports the client software running on the server as well, data management of the servers disks is also provided as depicted in FIGURE 6. This allows shared access to the servers disk file systems combined with data management. Some systems will also allow the servers disks to be the first "tier" of storage used for migrating files from the clients. This provides two tiers of high performance storage before files migrate to tertiary storage.

It is easy to see the flexibility of Client - Server architectures. If the workstations shown in FIGURE 6 were configured as servers, and each was serving a network of diskless workstations, we would have a configuration which would support a number of decentralized department or work group servers where each server's local disks had HSM software managing the local data and moving less frequently used data to a shared tertiary storage system.

### 7.2.1 Characteristics of Server Based HSM

When the client software is used in conjunction with the server based tertiary storage components, this system is identical to a Server Based HSM.

For true Client/Server configurations, this system provides the performance benefits of local disk file systems combined with the benefits of HSM. The client software can also be written to

be consistent with the native operating systems capabilities. On clients with such features as access control lists (ACLs) or typed files, the client software can retain such functionality while implementing the data management function. Under Server implementations, the clients may only see those file system characteristics available on the Server.

The single biggest factors affecting the performance of access to migrated files will be the network interface used to send commands and data between the components and the performance of the tertiary storage system. The network interface should be flexible enough to support a wide range of client implementations. No standard exist at this time for these interfaces and vendors have all gone their own way. However, standards are expected in the future, in the hope that the components from various vendors will be able to work together.

System pricing for client server varies widely. Some vendors price by client, others by GB of tertiary storage and yet others by GB of disk managed. (I'm sure I missed some.) These systems will typically cost more for equivalent servers and storage devices because of the higher complexity involved in supporting remote clients. However, the current market forces have created a broader selection of these systems than high end Server Based HSM systems and competition seems to be quickly driving prices lower.

## 7.3 Multi-Tier HSM

In multi-tier configurations, data can be moved from the primary storage through multiple levels of tertiary storage according the migration parameters. The first tier might be an optical disk jukebox providing fast random access, fast media load times and a lower cost per MB than the primary storage. Once resident on the optical disks, data may later become eligible to be moved again to less available storage. The next tier might be a tape library system. This tier would feature a very low cost per MB but at the expense of much longer file retrieval times should the data be requested.

Some systems will have off-line media as the lowest level of managed storage. In such a system, data moved to the off-line media will require operator assisted media loads for any requested data. While this is certainly the least expensive, it does require an operator, and data access is no longer automated.

### 7.3.1 Characteristics of Multi-Tier HSM

Multi-tier HSM provides a very sophisticated ability to tune the cost of data storage to the access patterns of the data itself. However, setting up, managing, and tuning such systems can be a significant effort. In addition, the cost of these systems is higher due to the added functionality and flexibility. Look carefully to see that the added savings on storage capacity of the lowest tier(s) justifies the administrative effort and initial purchase differentials.

These systems should be capable of moving data from any tier direct to the user systems primary storage. The performance impact of having to move the file through multiple tiers to retrieve the file would create severe performance penalties.

## 7.4 Distributed HSM Architectures

While we have demonstrated some of the configurations available in HSM solutions from a file system perspective, we have yet to discuss the flexibility some vendors offer on the tertiary storage side. Whether or not the system consists of multiple tertiary storage devices as discussed in the mutli-tier configuration above, it may be desirable for the tertiary storage to be remote from the primary storage or from other tertiary storage components. FIGURE 7 shows a distributed storage HSM configuration.

The ability to distribute the storage elements of an HSM system can spread the processing, data transfer and the network loads. This type of capability creates a more scalable system and one

where additional capacity can be added in smaller increments. The power of any given component does not need to be scaled to match the total systems capacity.

**DISTRIBUTED MULTI-TIER HSM**

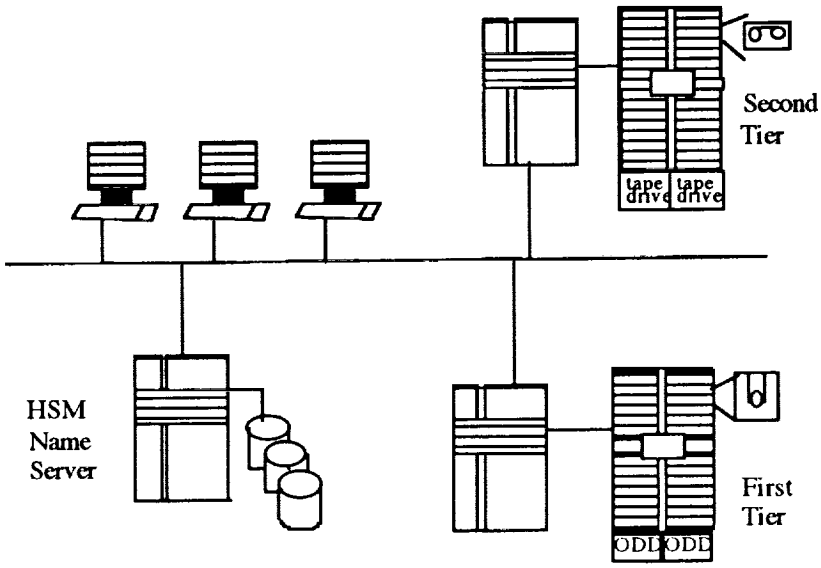



FIGURE 7

## 8. Combined Architectures

It is possible to configure many different system architectures from the individual concepts presented in this paper. For example, a multi-tier HSM system consisting of a RAID as the primary storage device coupled with an optical jukebox could be used in a network attached peripheral system in order to provide the high performance advantages of both architectures and the central name space attributes of a server based implementation.

## 9. Summary

It is not possible to address all of the possible architectures. I hope that this paper has, however, introduced the basic architectural concepts.

By having the basic understanding of the various type of implementation available, it is possible to analyze ones requirements with respect to performance, cost, transparency, complexity, etc. and to be able to evaluate the competitive offerings on the market with a more objective set of criteria. No one vendor offers all of the configurations discussed, it is therefore important to understand the technology to determine if a given vendor's proposal meets your requirements.

N94-33816

# Storage Media Pipelining: Making Good Use of Fine-Grained Media

**Rodney Van Meter**

ASACA Corporation
Tokyo, Japan
3-2-28 Asahigaoka, Hino-Shi, Tokyo, 191 Japan
rdv@alumni.caltech.edu

## Abstract

This paper proposes a new high-performance paradigm for accessing removable media such as tapes and especially magneto-optical disks. In high-performance computing, striping of data across multiple devices is a common means of improving data transfer rates. Striping has been used very successfully for fixed magnetic disks, improving overall system reliability as well as throughput. It has also been proposed as a solution for providing improved bandwidth for tape and magneto-optical subsystems. However, striping of removable media has shortcomings, particularly in the areas of latency to data and restricted system configurations, and is suitable primarily for very large I/Os. We propose that for fine-grained media, an alternative access method, media pipelining, may be used to provide high bandwidth for large requests while retaining the flexibility to support concurrent small requests and different system configurations. Its principal drawback is high buffering requirements in the host computer or file server.

This paper discusses the possible organization of such a system, including the hardware conditions under which it may be effective, and the flexibility of configuration. Its expected performance is discussed under varying workloads, including large single I/Os and numerous smaller ones. Finally, a specific system incorporating a high-transfer-rate magneto-optical disk drive and autochanger is discussed.

## 1. Introduction

*"Life does not give itself to one who tries to keep all its advantages at once."*
*Leon Blum*
*For Dr. Kim "Wombat" Korner, 1953-1993, a good and marvelously unconventional friend and teacher.*

We propose that, for fine-grained media, a new access method, which we have dubbed **media pipelining**, can be used to dramatically increase the aggregate bandwidth available. Media pipelining operates much like pipelining in a CPU with multiple functional units[1], overlapping multiple requests (or portions of a single large request) to improve system throughput and resource utilization. Many of the analysis techniques applied to processor pipelines, including space-time diagrams and pipeline reservation tables can usefully be applied to media pipelining.

Pipelining can benefit single large jobs in a manner comparable to striping, while retaining the flexibility to accommodate smaller requests that striping may sacrifice. It also easily supports different system configurations, allowing the system to operate effectively with any number of drives. This flexibility also means that dynamically changing workloads can be handled effectively. The principal drawback to media pipelining is high buffering requirements in the file server or filesystem cache to achieve maximum throughput.

Section 2 presents some definitions for discussing the performance of such systems. Section 3 briefly explains striping for removable media. We present a contrasting discussion of pipelining in Section 4. Section 5 briefly covers host requirements for pipelining. Section 6 describes in detail one possible implementation, and section 7 presents our conclusions.

We define **granularity** as the ratio of the capacity of a medium to its transfer rate. The result is the amount of time it takes to read the entire medium. Some removable media have a very high capacity-to-bandwidth ratio. As an example, the ASACA AMD-1340NS HSMO disk drive, with a medium capacity of 600 MB per side and a transfer rate of 10 MB/s can read an entire medium in under one minute, which is approximately 4 times the time necessary for an autochanger to exchange the medium and a drive to perform its load and unload operations. We refer to such media as **fine-grained media**, as opposed to those whose read times may be on the order of hours (for example, the new optical tape has a capacity of one terabyte per reel, and a transfer rate of 3 MB/s, making a granularity of $3.3 \times 10^5$ seconds, or more than 900 hours), which we call **coarse-grained media.** Some example granularities of both common and experimental removable media are summarized in table 1 (the capacities for HSMO and ISO MO are for one side of a double-sided disk). (Note that this simple chart does not take into account drive type and host interface, which may result in different apparent granularities for the same medium.) It is easy to see that granularity varies by orders of magnitude. The impact of this feature on system design has not been fully explored.

| Media Type | Capacity (MB) | Transfer Rate (MB/s) | Granularity (sec) |
|---|---|---|---|
| 3480 tape | 200 | 2 | 100 |
| VHS T-120 tape | 14,000 | 3 | 2,700 |
| D-2 S tape | 25,000 | 15 | 1,700 |
| HSMO disk | 600 | 10 | 60 |
| ISO MO disk | 300 | 0.6 | 500 |
| optical tape | 1,000,000 | 3 | 330,000 |

*Table 1: Example Granularities of Removable Media*

It is also sometimes desirable to talk about the effects of media granularity on the total performance of an autochanger system. In this case the important measure is the (dimensionless) ratio of the media granularity divided by the cartridge exchange time, $G/t_x$.

## 2. Assumptions and Definitions

We will discuss both striping and pipelining in the context of two different access patterns. The first is for an assumed linear scan of a very large ($>>c_m$) dataset. It will be analyzed primarily for its steady-state behavior rather than startup or latency. The principal metric is $r_t$, the total apparent throughput for a complete system.

The second workload is small requests located randomly in the entire available dataspace. In any removable-media system, with an average request size $O(r_m * t_r)$, both p and $r_t$ are low; a better metric is $x_n$, the number of requests that can be serviced in a given time. Flexibility and the ability to support dynamically varying workloads will also be discussed.

We assume throughout this paper that the number of media in use is much larger than the largest possible stripe set; typically hundreds of media in a single autochanger (or "cart machine"). We further assume that it is desirable for this entire collection of media to be composed into a single dataspace (or a small number of large dataspaces). For fine-grained media this appears to be a reasonable assumption, freeing applications from managing data in chunks that may be unnatural and result in wasted capacity. However, this does force fixed addressing, as demonstrated later, making compressing media or other media with highly

variable capacity poor candidates for pipelining. This also essentially constrains the management of the media to automated handlers, but fine-grained media are unlikely to be used for human-handled dataset import/export anyway. While removing or adding media from/to the middle of the dataspace is impossible once the addressing is fixed, simple expansion is straightforward -- new addresses are allocated past the end of the existing dataspace.

If a user process is transferring very large amounts of data, >> $c_m$, a fine-grained system is to a certain extent handicapped. The apparent aggregate transfer rate using one drive, $r_a$, is limited to $r_m * p$, where

$$p = G/(G + t_x) \quad \text{for} \quad t_x = t_u + t_r + t_l + t_s.$$

It is clear that for coarse-grain media this ratio p is close to 1, meaning that over the long term for very large requests (>>$c_m$) that the cost of media exchanges is negligible. However, for fine-grained systems such as the ASACA HSMO, p may be significantly less than one. Thus, low granularity would appear to be a significant handicap; can it be turned into an advantage?

| Abbr.. | Description |
|---|---|
| B | buffer space necessary |
| $c_m$ | capacity of a single medium |
| $c_s$ | capacity of a stripe set = S * cm |
| p | percentage of time a drive spends transferring data |
| $r_m$ | transfer rate of a single drive |
| $r_s$ | transfer rate of a stripe set |
| $r_p$ | transfer rate of a pipeline configuration |
| $r_a$ | aggregate transfer rate including media exchange times |
| $r_t$ | total throughput for a multi-drive system |
| $t_m$ | robot cartridge move time |
| $t_l$ | drive load time |
| $t_u$ | drive unload time |
| $t_i$ | time for the robot handler to actually insert/remove a medium |
| $t_d$ | data transfer time |
| $t_s$ | seek time |
| $t_r$ | robot round trip time, perhaps 2*tm |
| $t_x$ | time to exchange a medium, including eject, setup, and seek |
| $n_b$ | number of blocks per medium |
| $n_d$ | number of drives in system |
| S | striping factor (ignoring ECC additions) |
| $s_b$ | block size (in bytes) of media |
| $s_s$ | logical block size for stripe set = S * sb |
| $s_p$ | logical block size for pipeline set = sb |
| $x_n$ | request (transaction) rate (dimension #requests/time) |
| G | granularity = $c_m$ / $r_m$ (dimension is time) |

*Table 2: Definitions*

## 3. The Shortcomings of Striping

One possible way to increase system throughput for large requests is to stripe the data across multiple media, increasing the available data size and multiplying the data rate. This approach is fine for fixed disks with stable configurations, but in a more dynamic system with removable media it presents severe management and use problems and may substantially increase the vulnerability of the user's data to access problems or media failures.

Hard disk systems that perform some form of striping must take steps to ensure the integrity of the data. The simplest approach, simply copying the data across two disks, improves the safety of the data but does not help either the transfer rate or the capacity. Larger RAID systems improve all of the above by using more than two disks and designating one or more disks to store error correction information[2].

Striping of data across two or more removable media is being investigated[3]. The principal problem with striping of removable media, especially in a robotic system designed to reduce the latency of access to the data, is that the whole set (perhaps, depending on the management scheme, minus the error control tape) must be on-line at once to read or write. Figure 1 shows the logical block layout of two stripe sets laid back to back as a single address space. A minor consideration is that the block size also goes up by a factor of S.

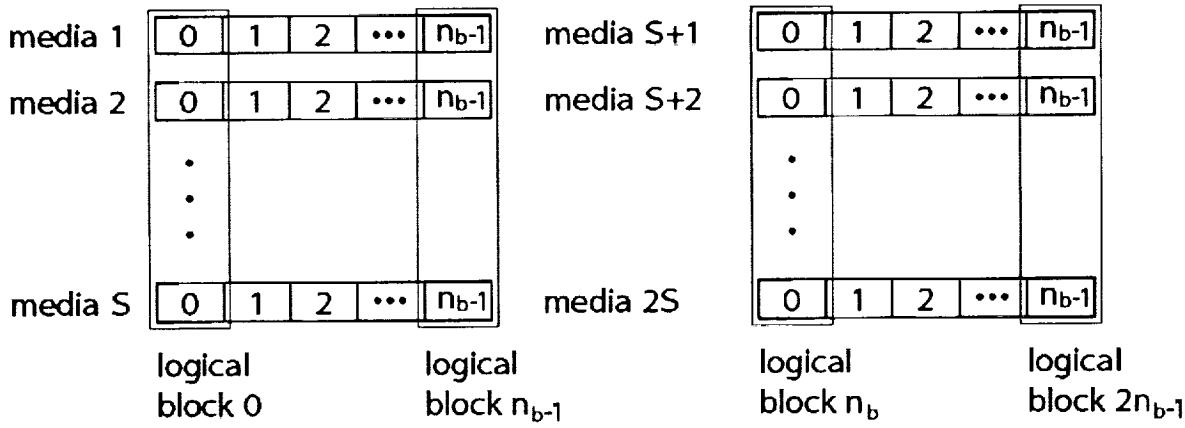



*Figure 1: Logical Block Layout for Striped Media*

Assuming intra-cart machine striping, bringing a stripe set online involves several operations by the robot to fetch multiple media, and forces the drives to sit idle while other drives in the set are loaded. Also, if the set consists of S striped tapes, if only S-1 drives are available at the time, the dataset may be unavailable. In a striped system drives must generally be allocated and used in sets of size S, meaning that the addition of a single drive does little good but the removal of a single drive may prevent access to data.

For a single cart machine servicing small requests, the maximum rate of requests that can be serviced, $x_n$, is $1/(t_r * S)$.

Inter-cart machine striping can be used to eliminate the increased latency for media access and increase the throughput of the system for small requests, but this again is very limiting in system configuration (requiring S cart machines exactly) and increases the vulnerability of the system to robot failures. It may be effective for small autochangers (10-tape stackers, for example) but becomes a very expensive solution for larger autochangers.

Figure 2 shows the access timing for a 4-way intra-cart machine stripe set.

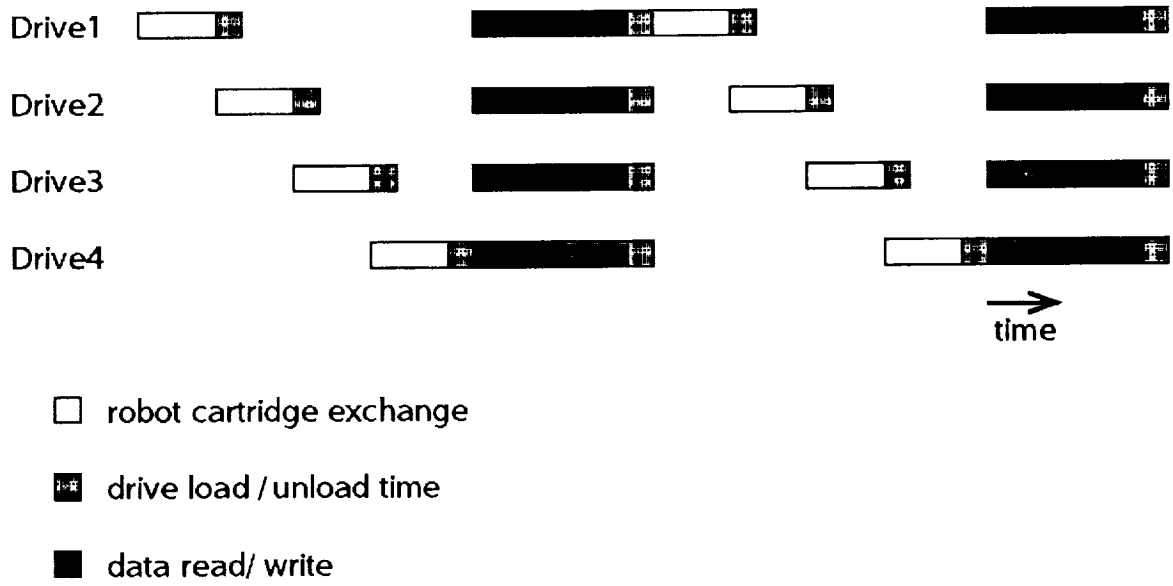



*Figure 2: Striping of Removable Media*

## 4. An Alternative Solution

One answer which appears to address some of the problems of striping for fine-grained media is **media pipelining**. It can use a higher percentage of available drive bandwidth, increasing total throughput for large requests, and retain the flexibility to accommodate small requests.

The logical block layout for media pipelining can be the "obvious" one, with blocks counting up on the first medium in the system and continuing consecutively across media boundaries, as shown in figure 3. A key assumption of this paper, as mentioned earlier, is the desirability of maintaining a single addressable space (the dataspace).

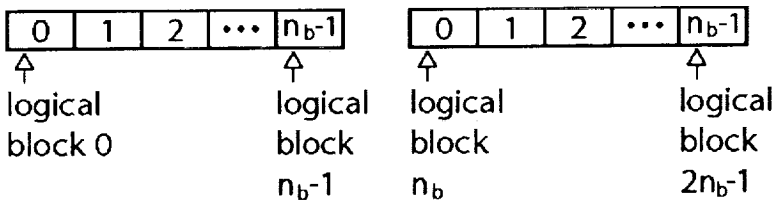



*Figure 3: Logical Block Layout for Pipelined Media*

The concept of media pipelining comes into play when requests move into the range of the media size $c_m$, on up to the terabyte range. How can multiple drives be used to increase the speed at which such requests are serviced? The simple answer is to recognize that the request spans a disk boundary, and preload the next disk in the request and have it ready to read when the first disk completes. This in itself is a simple form of pipelining, with the loading of a disk overlapping the reading of another. However, it commits two drives to the read and provides the effective bandwidth of one full drive. This we call **linear pipelining** (our definition is

somewhat different from that of Hwang and Briggs). Linear pipelining is illustrated in figure 4. Even in fine-grained systems, two drives should be sufficient to provide linear pipelining.

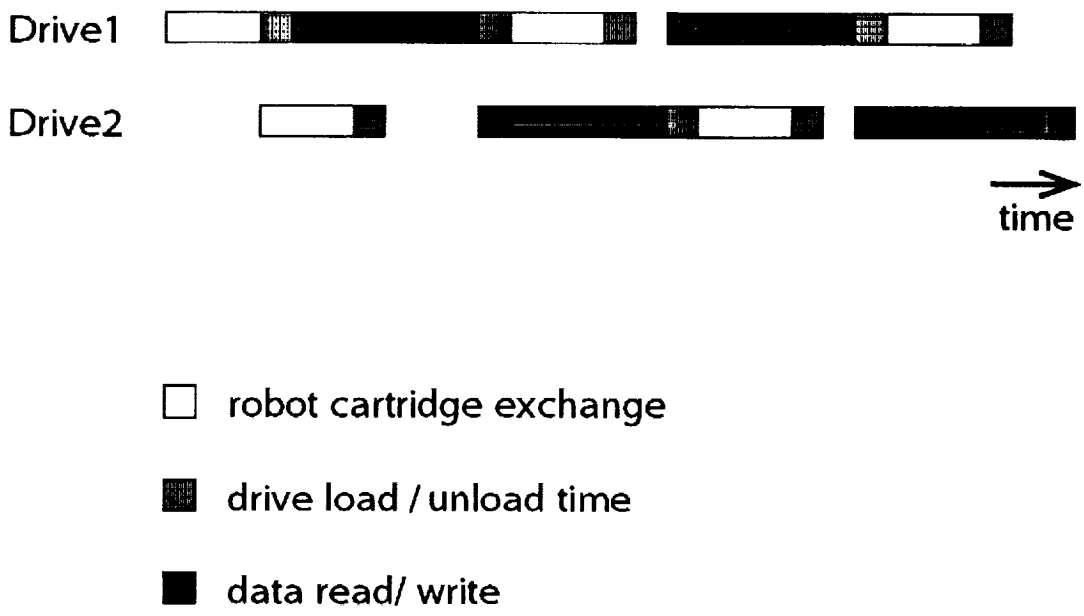



*Figure 4: Linear Pipelining*

It is possible, with a little care and appropriate driver software, to increase the drive utilization with pipelining by allowing the second drive to begin reading as soon as it comes on line. This we refer to as **superlinear pipelining**, as in figure 5. The diagram shows the overlapping reads, and below, a graph of the amount of data delivered to the application, assuming that the application is infinitely voracious but insists on data being delivered in order. It can be seen that both drives run at p efficiency, resulting in a total sustained throughput $r_t = 2 * r_a$.

The data that is read off of the second medium before the end of the first medium must be buffered (represented graphically as the gray area above the data delivered curve). It can be seen that this peaks right before the end of the first medium. The maximum amount of buffer space necessary is $B = (G - t_r) * r_m$. It should be readily apparent that this requirement can be reduced by having the system pause before starting the transfer from the second disk, reducing the total overlap. This obviously delays somewhat the delivery of the data, but may be acceptable depending on the rate at which the application is truly consuming data (in very long reads, the impact on average throughput would be negligible anyway). The most efficient configuration is of course device- and system-dependent.

As the size of the request grows, the full resources of the system can be brought to bear on the problem. For reads of $N * c_m$ or larger, we reach steady-state transfer rates utilizing the maximum percentage of the bandwidth of each drive.

A key feature of pipelining is its flexibility of configuration, allowing dynamically varying drive allocation depending on user needs and work load. For example, in a four-drive system, the first user to begin using the system may receive data at the full transfer rate, just as in a striped system. A second user entering the striped system may find his access to his data blocked, while the pipelined system can reconfigure (presumably on a medium boundary in the first transfer) dynamically, reallocating the drives 2-2 or 3-1 in favor of either user, allowing both to continue working in a manner very analogous to dynamic multiprocessor configurations. The system should autoconfigure, allowing up to as many users as there are

drives available (at one drive per user, this would be sublinear pipelining, equivalent to a "stalled" processor pipeline). It is not even a requirement that the different operations be the same kind of operation; one may be an excellent candidate for pipelining, one may be a concentrated series of operations within a single medium, while another is randomly placed reads and writes throughout the entire dataspace.

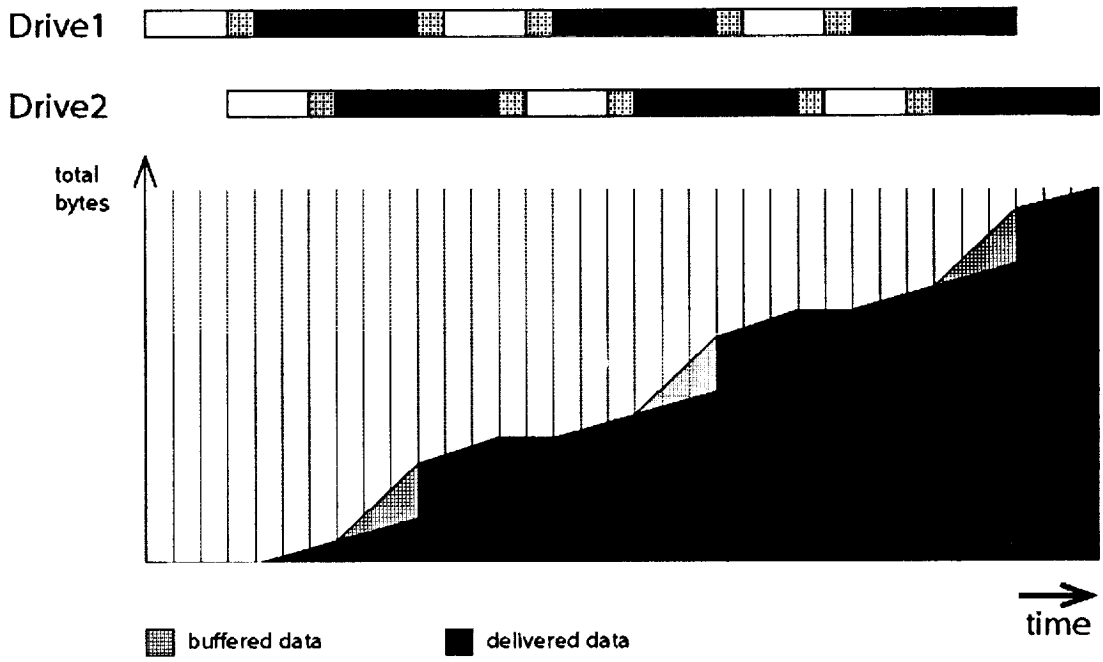



*Figure 5: 2-Way Superlinear Pipelining*

The pipeline overlap can be decreased when necessary (i.e. when the host cannot afford such a large buffer) for the simple tradeoff of reduced throughput for pipelined requests.

The system hardware configuration is extremely flexible with respect to the number of drives in the system. Unlike a striped system, a pipelined system can run comfortably with any number of drives, allowing drives to be moved, allocated, or maintained without necessarily forcing the unavailability of the entire system.

For small reads and writes (where "small" in this context may be less than a few hundred megabytes), $<< c_m$, the probability of the entire read residing on a single medium is high. In this case, the operation will be dominated by the media load time rather than the transfer time (as mentioned above, the load time becomes substantially longer in a striped system), negating any advantage in improved transfer rate from a stripe set. A single drive may be allocated to the request, leaving the other drives free for other operations, thus allowing the system to concurrently process as many requests as it has drives.

## 5. Operating System and File System Requirements

The operating system and file system have numerous demands made of them in a pipelined system. Both must be able to address large spaces. The device driver must be able to support multiple physical devices and manage and reassemble data as it comes in. Naturally the system should support transfers at such high rates.

If the requests to be pipelined arrive at the driver as a linear collection of smaller requests rather than a single huge request, the driver should still be able to pipeline the requests by prefetching data. If the file system provides "hints" it eases the process of determining when a

prefetch and pipeline setup will be worth the extra trouble. As discussed above, the system must be able to provide adequate buffer space, which should be readily available in a configuration with a minisupercomputer as a file server for a supercomputer.

High-speed interfaces are required to make such a system work. HIPPI and SCSI-2 and SCSI-3 are examples. The peak supported I/O burst rate (sum for the I/O busses used for pipelining; there is no requirement that all the drives be on the same bus) must be at least $n_d * r_m$ in order to maximize pipelining.

## 6. A Specific MO System

Evaluating real-world systems is of course substantially more complex than the abstract concepts presented above. Probably pipelining is best suited to removable disks, since their capacity is fixed, although uncompressed 3480 tape, with its low granularity, is also a possibility.

Asaca has developed the world's fastest magneto-optical disk drive, the AMD-1340N,[4] with a native data transfer rate of 12.24 megabytes/second, and a cartridge capacity of 1.2 GB (600 MB/side). These represent, respectively, twenty times and two times the values for most ISO-standard 5.25" MO drives. The speed advantage comes primarily from ASACA's 4-beam head technology, in which two heads each focus four lasers, for a total of eight beams lifting data off the disk concurrently. This tremendous speed improvement results in an entire side of a disk being readable in only 50 seconds. Using the SCSI-2 fast-wide interface, sustained transfer rates of approximately 10 MB/s are expected, and this is the number used throughout this paper.

The Asaca ADL-450 HSMO library contains 450 disks, 900 sides, 14,790 sectors of 40,448 bytes each, for a total capacity of 538 GB. It can hold up to four AMD-1340NS drives. Its potential in mass storage has already been discussed[5].

With the ASACA cart machine, the first disk comes on-line in approximately 15 seconds, and the second in approximately 23. The disk handler can hold two disks, meaning that during steady-state striping operations, the handler can prefetch the next disk while a drive is finishing a read and ejecting the disk, and have the second disk ready to load when the first is ejected. Thus, although the round-trip exchange time $t_r$ is 15 seconds, the load time $t_l$ is 8 seconds, the unload time $t_u$ is 3 seconds, and the insert time is approximately two seconds, a new disk can be online in approximately 15 seconds $t_x = t_u + t_l + 2 * t_i$).
For $p = G/(G + t_x)$, using $G = 60$, $p = 0.8$, so during steady-state pipelining, we can expect to receive approximately 80% of the drive bandwidth.

Assuming that the driver makes the intelligent choice of mounting the disk with the most data on it first (a pipelining reordering operation), the worst case for a one gigabyte read is when the data is split 500 MB each on two disks. In that case, the read should be completed in 50 seconds of read plus the 23 to mount the disks, for an average aggregate throughput of 12 MB/s.

This transfer rate amounts to an aggregate of 32 MB/sec. across the four drives in a complete system. The difficulty is in managing the data, drives, and robot to provide fast access in a relatively transparent manner. A crucial part of the problem is coordinating multiple drives so that a user may take the best advantage of all the resources the system has to offer.

## 7. Conclusion and Future Work

We have presented here a new concept, the granularity of media, and shown how fine-grained media can be used in a method called media pipelining, which offers some advantages over striping for fine-grained removable media. It offers additional flexibility and improved response time compared to striping for small-request and dynamic workloads. For very large

requests, it can offer improved throughput compared to striping at the cost of high buffering requirements.

Further work calls for simulations and an implementation to verify predicted performance, increased formalization of the analytic model, and possibly extensions to the concept to allow pipelining to be used for coarse-grain media as well. The interaction of media pipelining with the host operating system also offers challenging work.

---

[1]Hwang, Kai and Briggs, Faye' A., *Computer Architecture and Parallel Processing*, McGraw-Hill, 1984, pp. 145-212.

[2]David A. Patterson, Garth Gibson, and Randy H. Katz, "A Case for Redundant Arrays of Inexpensive Disks" *Proc. ACM SIGMOD*, June 1988.

[3]Drapeau, Ann L. and Katz, Randy H., "Striped Tape Arrays," *Proc. Twelfth IEEE Symposium on Mass Storage Systems*, Monterey, CA, April 1993.

[4]Nakagomi, Takashi et al, "Development of High Speed Magneto-optical Disk Drive Using 4 Beam Optical Head," *IEEE Translation J. on Magnetics in Japan*, Vol. 6, No. 3, p. 250, March 1991.

[5]Nakagomi, Takashi, et all, "Re-Defining the Storage Hierarchy: An Ultra-Fast Magneto-Optical Disk Drive," *Proc. Twelfth IEEE Symposium on Mass Storage Systems*, Monterey, CA, April 1993.

# The Trend to Parallel, Object-Oriented DBMS

David J. DeWitt
Professor and Romnes Fellow
Computer Sciences Department
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706
Phone: (608) 263-5489 / (608) 262-1204
FAX: (608) 265-2635

1

## Background

- Supercomputers users and vendors are finally discovering the importance of I/O!
- Recently I read a paper titled "Satisfying the I/O Requirements of Massively Parallel Supercomputers"
- Nice paper, but not a single reference to any work in the parallel database system field
- I found this:

2

# AMAZING!

3

## Why Amazing?

- Parallel DBMS community has been working on this problem for 15 years and essentially has it solved
- Teradata has systems in the field with over 300 processors and 1000 disk drives!
- Other vendors include NCR/Sybase, Tandem, IBM SP2, & DEC (soon)
- All vendors use the same basic architecture
- None of the supercomputer vendors use it.

4

## What am I going to talk about?

I wondered the same thing when I saw the program for this conference

5

## Talk Outline

- Hardware and software architectures used by today's relational DBMS products
- DBMS trends - the transition from relational to object-oriented
- What is an object-oriented (OO) DBMS?
- OODBMS and standards such at NetCDF
- Future OODBMS directions
- 1 slide sales pitch

6

## Database Systems Today

- Relational data model and SQL dominate
- Targeted at commercial applications
- A relational database: set of relations
- A relation: a set of homogenous tuples

Telephone_Book

| Name | Address | Number |
|---|---|---|
| Jones | 110 Main St | 255-4834 |
| Smith | 2164 Lake Lane | 238-5936 |
| Smith | 5 Roby Rd | 746-0192 |

- SQL used to create, update, and query

```
SELECT Number
FROM Telephone_book
WHERE Name = "Smith"
```

7

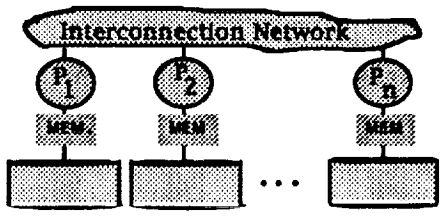
## Relational Database Systems

- SQL is easy to optimize and parallelize
- Terabyte databases, consisting of billions of records, are becoming common
- Databases of this size require the use of parallel processors
- Teradata and other commercial parallel DBMS employ what is termed a shared-nothing architecture

8

## Shared-Nothing

- Each memory and disk is owned by some processor that acts as a server for that data

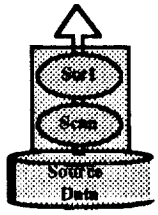


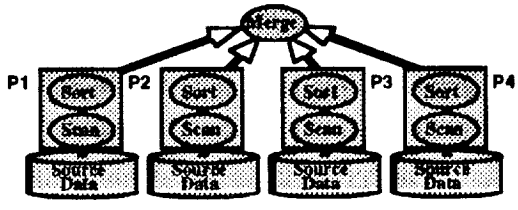

- Teradata, NCR 3600, Tandem, IBM SP2
- Actual interconnection network varies: trees, hypercubes, meshes, rings, ...

9

## Relational DBMS Parallelism

- 3 key techniques: pipelining, partitioned execution, & data partitioning
- Pipelined parallelism
- Partitioned execution




Telephone_Book Relation

10

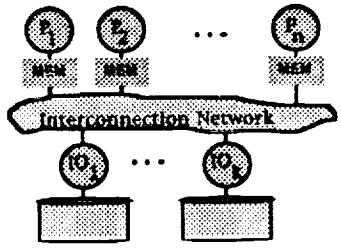
## Relational DBMS Summary

- Shared-nothing approach has proven to scale very successful providing both linear speedup and scaleup on I/O intensive applications
- The largest Teradata systems have over 300 processors and 1000 disk drives (1 terabyte)
- While NASA has lots of satellite data, K-Mart and WalMart have lots of cash registers!
- Despite all the talk about an I/O bottleneck, the vendors supplying parallel processors to the scientific community are not following suit.
- What are they doing?

11

## Dedicated I/O Nodes (Shared-Disk)

- Each processor has a private memory and access to all disks




- CM-5, Intel Paragon, Cray T3D, IBM 3090
- DBMS community have rejected such architectures
  - coordinating access to shared data is complex
  - extra cost required for I/O nodes and their network interfaces

12

## Common Gripes about Shared-Nothing

- **Packaging problems**
  - **total nonsense. Teradata is sufficient proof**
  - **by 2000, 1", 1 gigabyte drives will be common**
- **Interference with application code**
  - **Assume:**
    - » 50 MIP cpu
    - » 20 ms. to do a disk I/O
    - » Each "remote" disk request consumes 3000 instructions locally (1000 to accept message, 1000 to start I/O, 1000 to send page back to requestor)
  - **So every 20 ms., 3000 instructions are stolen from application. These 3000 instructions account for 0.3% of the available CPU cyles!!!**
- **The future of parallel computing may be commodity computers connected by commodity networking technology (e.g. ATM)**

13

## Another Observation

- **DB community has totally accepted message passing for both parallel computation & parallel I/O**
- **Scientific community has accepted message passing as the standard communication mode (though HPF may hide a lot of ugly details)**
- **But, is holding on to a "shared-disks" architecture for the parallel I/O system**

14

## Talk Outline

- Hardware and software architectures used to today's relational DBMS products
- DBMS trends - the transition from relational to object-oriented
- What is an object-oriented (OO) DBMS?
- OODBMS and standards such at NetCDF
- Future OODBMS directions
- 1 slide sales pitch

15

## The Transition from Relational to Object-Oriented

- Why?
- Relational DBMS:
  - Modeling capabilities too limited:
    - » Tuples (records) of base types only!
    - » No arrays let alone polygons or polylines
    - » No nested tuples or structure-valued attributes
  - Application interface (i.e. SQL with cursors) is simply wrong for manipulating scientific or CAD data
    - » CAD applications love to chase pointers around
  - No support for tertiary storage

16

## What is an Object-Oriented DBMS?

- The marriage of a modern programming language such as C++ and a modern DBMS.
- From the programming language world:
  - Rich type system including classes with encapsulation and inheritance
  - Computational completeness
- From the DBMS world:
  - Persistence
  - Bulk types ( sets, lists)
  - Transactions (concurrency control and recovery services)
  - Associative queries (balance < $100)
- Transparent Access to Persistent Objects

17

## Example

- Given the following type definition

```
class raster_data {
    int     time;
    int     frequency;
    float   image[4096][4096];
}
```

- Can declare persistent variables of this class:

```
persistent raster_data X, Y;
```

- Transparent access to data on disk:

```
for (i=0;i<4096;i++)
      for (j=0;j<4096;j++)
        Y[i][j] = f(X.image[i][j]);
```

- A year's worth of data:

```
persistent raster_data GeosDataSet[365];
```

18

## OODBMS & Standards like NetCDF

- Data model provided by a typical OODBMS is much more general than that provided by a standard
  - Typical OODBMS, in addition to arrays and records, provide sets, lists, and relationships as type constructors
- Transparent access to persistent data makes manipulation of data residing on secondary and tertiary storage trivial
  - Current products have sufficient performance to satisfy even the most demanding CAD applications
- Persistent objects are strongly typed with their type descriptors stored as persistent objects in the database

19

## Talk Outline

- Hardware and software architectures used to today's relational DBMS products
- DBMS trends - the transition from relational to object-oriented
- What is an object-oriented (OO) DBMS?
- OODBMS and standards such at NetCDF
- Future OODBMS directions
- 1 slide sales pitch

20

## Future OODBMS Directions

- **Standardization via either ODMG or SQL3**
- **Integrated support for tertiary storage**
- **Extension to parallel processors**
  - **Current products architected for client-server environments**
  - **Only "small" databases supported: 10s of gigabtyes and not terabytes**
  - **Two possible directions**
    - » Relational products will adopt a richer type system such as SQL3
    - » OODBMS products will be extended to operate on parallel processors
    - » Joint project between KSR and Intellitic to parallelize Matisse is the first such effort

21

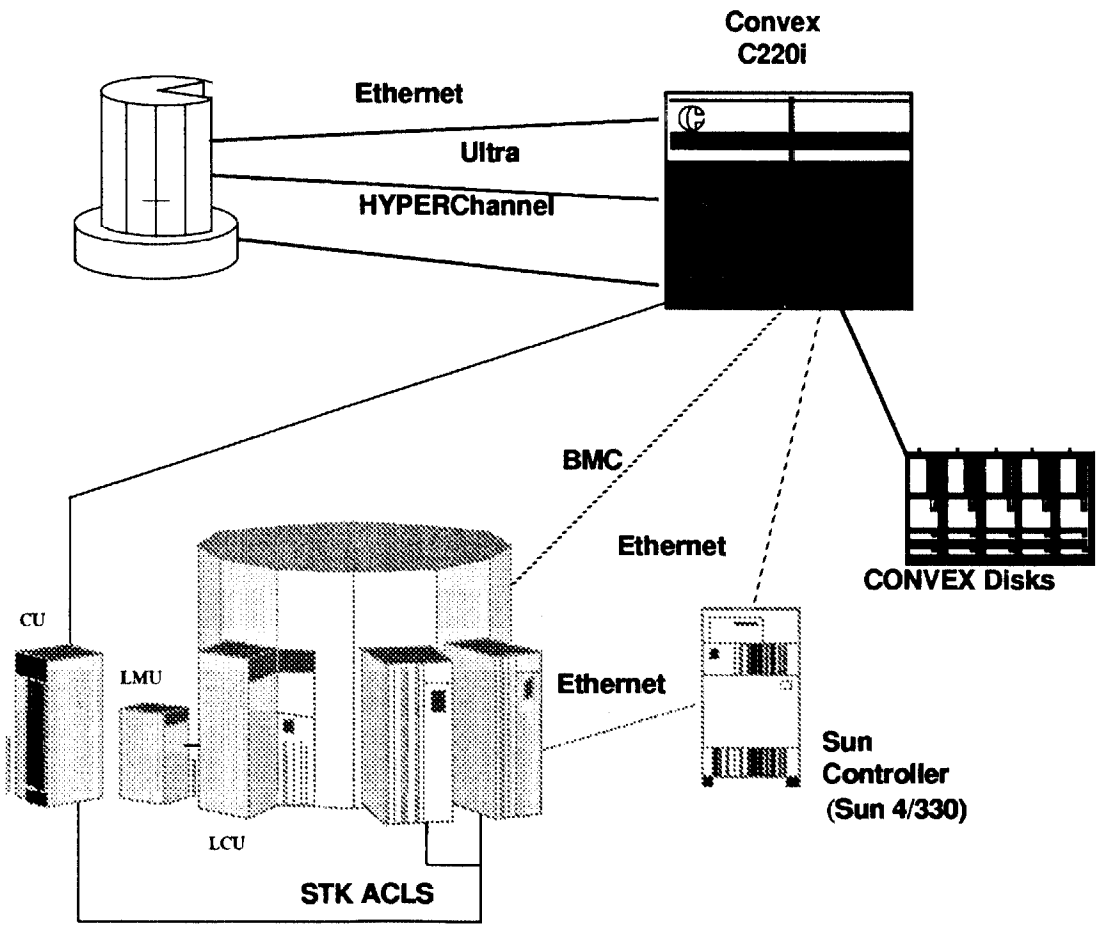
## Parallel-Sets (ParSets)

- **Proposed by Kilian, basis of Matisse/KSR effort**
- **Employs a data-parallel approach to object-oriented parallel programming**
- **ParSets extend set type constructor as follows:**
  - **ParSets are partitioned across multiple processors/disks to facilitate CPU and I/O parallelism**
  - **Provide 4 basic operations:**
    - » Add()
    - » Remove()
    - » SetApply() - invokes a method on all the objects in parallel
    - » Reduce() function - calculates a single value from all objects in the ParSet
- **Most promising proposal to date. Can be extended to other bulk types such as arrays, lists, trees, etc.**

22

# ABUNDANT PROGRAM

## R1 MSL ARCHITECTURE



1991, the first instantiation was installed and tested. It was based upon the architecture depicted in Figure One and employed a Convex control processor, the STK silo, and custom file server software developed by our integration contractor. The purpose of the interim system was to test the proof of concept and functionality of the product, and most importantly, to develop lessons learned which would help shape improvements for the larger 1015 bit system. The system was tested with Cray client systems during late 91 and early 92 and the desired lessons learned were captured.

The second release would also use a Convex control processor, but would have more functionality and increased performance. To handle the marked increase in capacity, an aisle based robotic tape archive with the desired modularity and capacity was developed. Built under a subcontract, this product was designed, developed, and tested by our integration contractor. This archive would use the ER90 D2 helical scan recorders and would be fully compliant with all of our stated goals and requirements. As we all know, what started as a custom development has now become a commercial product known as E-MASS. In addition, varying size robotic libraries are now commercially supported which include the STK Silo, the Odetics Data Tower (6 TBs) as well as our Data Library (150 TBs).

Taking advantage of the rapid expansion of commercially available Mass Storage product offerings, the Agency began restructuring the ABUNDANT program. In late FY92 a separate contract was awarded to field a network attached Volume Server based on the D2 recorders and the Odetics Data Tower. This system would use HiPPI switch technology to suppor
t tertiary tape storage needs of the recently acquired Cray YMP C-90 computer system used by our research organization. Once again it should be noted that we were acquiring a COTS, vice custom system. In addition, it was determined that the File Server EMASS product was mature enough so that the current ABUNDANT contract type (for the second release) could be changed from developmental to firm fixed price. Other operational changes have allowed this system to be utilized as a shared resource vice a dedicated system to a specific user group. Finally, to accelerate the fielding of the file server system, we decided to first implement it as a Data Tower this summer, and to then field the larger Data Library in late FY94. These changes have now been contractually implemented and final planning is being conducted to provide for a smooth installation this fiscal year.

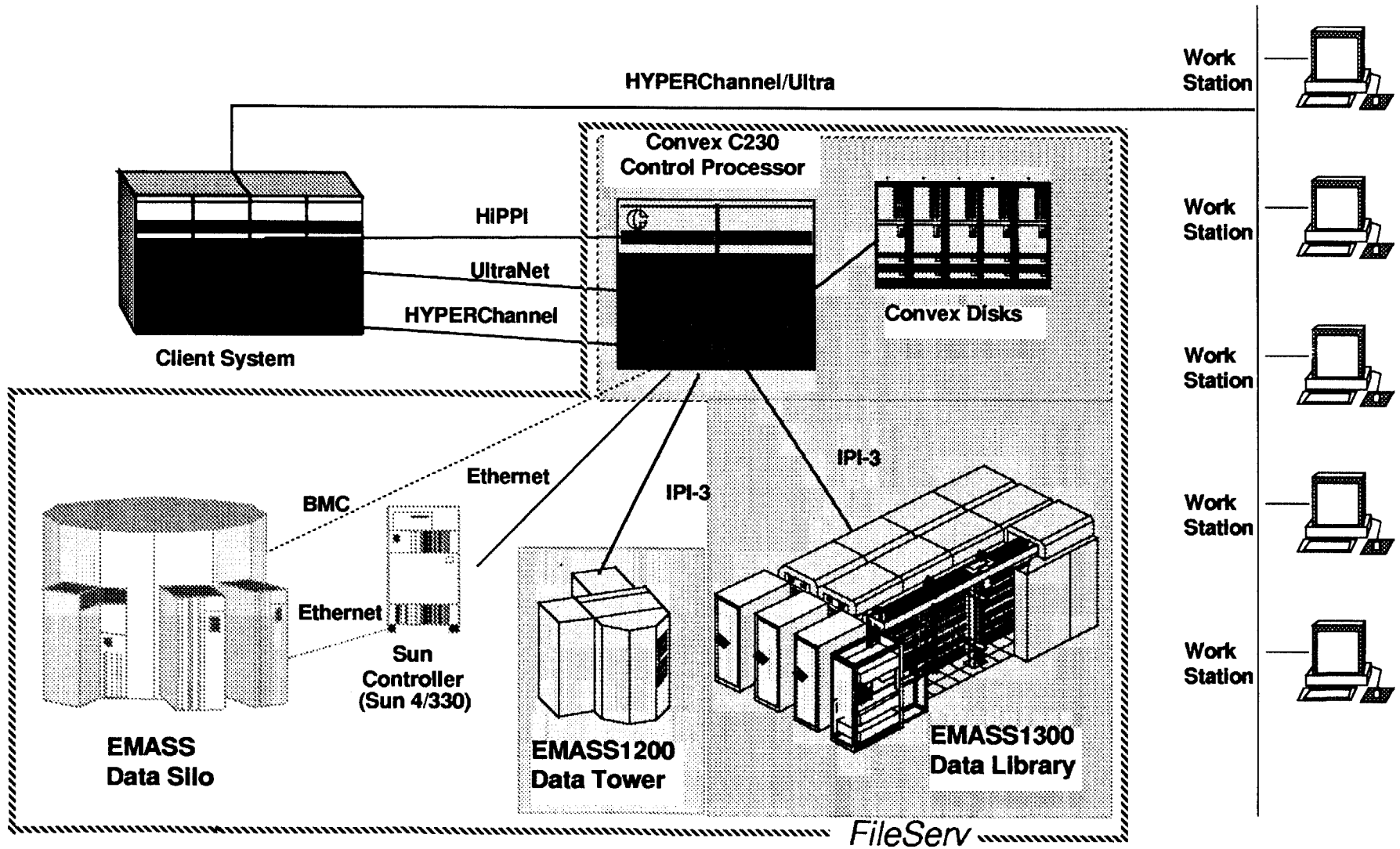
Figure Two contains a depiction of both the interim and final file server configurations. It is important to note that this architecture is totally modular, offers significant flexibility for future change and upgrades, and clearly satisfies our COTS, footprint, and flexibility program goals. Figure Three contains a similar depiction of the HiPPI network attached Volume Server Data Tower system. Our principal activities over the next year will be to perform significant testing, of both approaches to Mass Storage so as to determine their optimal employment. Detailed plans are being developed with our customer, support, and operations organizations to fully evaluate both products.

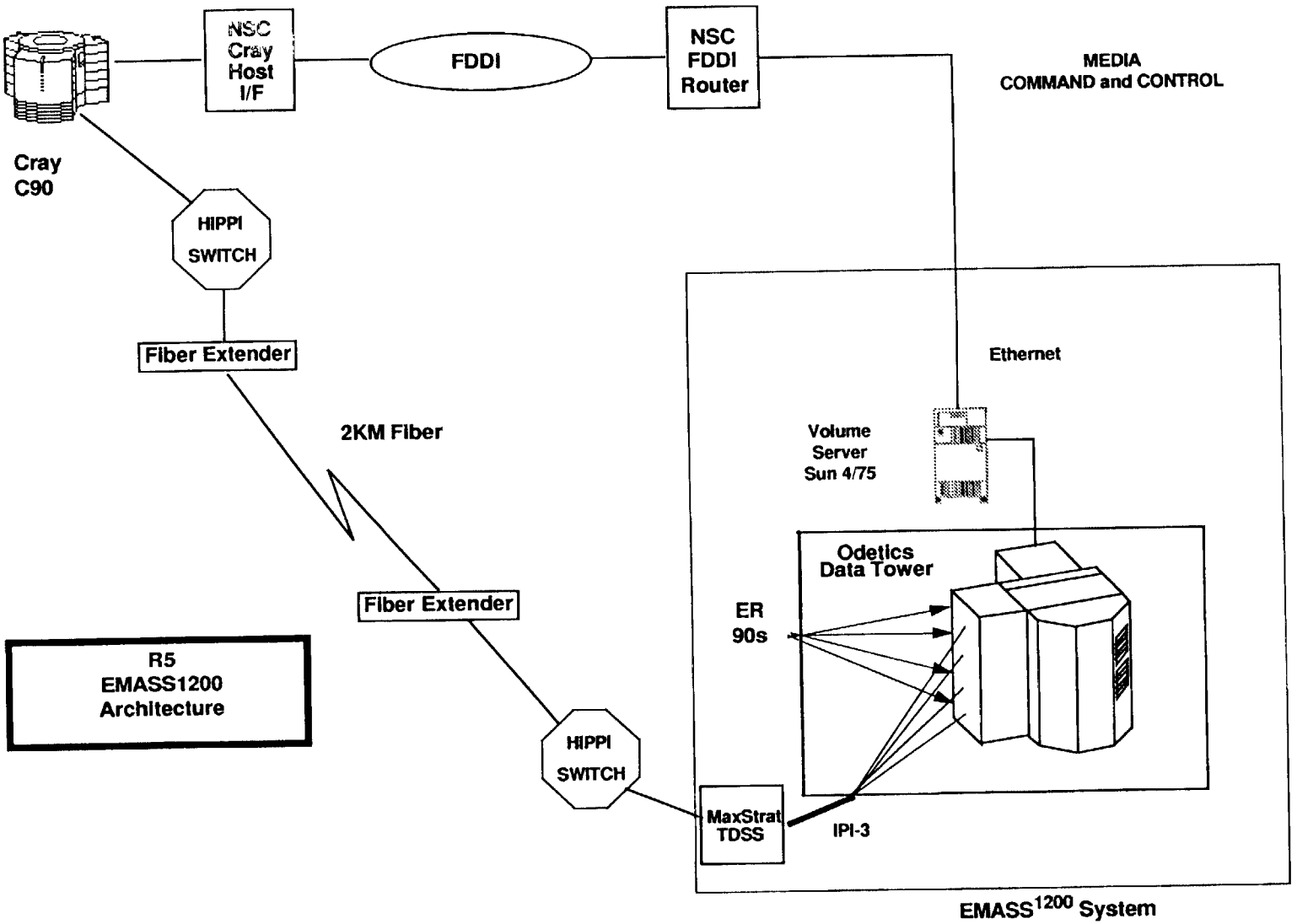
## 3.0 Current Environment

I alluded to the appearance in the commercial marketplace of a wide range of products that have, in most instances, found their way into our current operational environment, since the inception of ABUNDANT. Let me outline some of these that are used in our daily computer production. Perhaps the STK Silo could best be described as today's Mass Storage System of choice at the high end of the spectrum. Numerous silos are employed for the Cray, Convex, IBM and Amdahl, and other high end processors that we utilize. Silos are used as volume servers and are usually clustered in groups of two or three. They are cross connected to insure high availability and permit data interchange.

Next, we use both the Metrum RSS 48 and 600 SVHS robotic tape systems for mid-range processors. All of these are used as file servers and run the AMASS commercial file server product. Other AMASS uses employ robotic 8mm tape and optical disk libraries. In addition, Exabyte robotic controlled 8mm systems (EXB-120s, 10i's, and carousels) are used as volume

# CONVEX EMASS1200 FileServ Interim - 1300 FileServ Final at IOC




NSC
Cray
Host
I/F
FDDI
NSC
FDDI
Router
MEDIA
COMMAND and CONTROL
Cray
C90
HIPPI
SWITCH
Fiber Extender
2KM Fiber
Ethernet
Volume
Server
Sun 4/75
Odetics
Data Tower
ER
90s
Fiber Extender
R5
EMASS1200
Architecture
HIPPI
SWITCH
MaxStrat
TDSS
IPI-3
EMASS 1200 System

servers principally to perform backup function. A few user groups employ the Epoch file server software to manage their files.

## 4.0 Near Term Environment

Later this summer, the first UniTree evaluation will occur. This test will use the Amdahl as the control processor and the STK silo as the robotic library. Another user group is acquiring the TriPlex STX controller and the Sony ID-1 robotic library for use with a Cray YMP system. As previously stated, both the EMASS Data Tower FileServ and VolServ systems will be installed this year for evaluation. The Data Library EMASS FileServ product will replace the Data Tower in 1994. In addition several IBM 9570 RAID disk devices will be fielded; some will be tested as HiPPI attached network storage. Plans are being developed to evaluate shared file systems among multiple client computers with these devices.

## 5.0 Future Environment

One of our principal goals for the future is to match massively parallel processing with network attached storage. We have been active with the IEEE Mass Storage Reference Model Committee and other forums and will continue to participate. We intend to deploy Mass Storage technology to selected field sites in order to reduce our network bandwidth requirements. We intend to field an architecture that employs network storage devices which are readily accessible by any of our processors, yet has directly attached storage in those areas where security and access requirements dictate. We will closely monitor research work in these areas as well as that done in tape striping. We will also continue to closely monitor optical tape.

One of our principal lessons learned over the years is that we can no longer afford to enter into a custom development effort for Mass Storage. We must and will rely upon published standards, COTS products, and open systems architectures. We believe that the computer OEMs must accommodate storage product support as a price to do business. Finally, we believe that system reliability, manufacturer warranty, and support costs are just as important as any other acquisition consideration.

# ACE: A Distributed System To Manage Large Data Archives

**Dr. Mike I. Daily**
Mobil Oil Corporation, P. O. Box 650232, Dallas, TX 75265-0232
Phone: 214-951-2651 Fax: 214-951-3923
and
**Dr. Frank W. Allen**
E-Systems, Inc., P. O. Box 660023, Dallas, TX 75266-0023
Phone: 214-205-7510 Fax: 214-205-7200

## Introduction

Competitive pressures in the oil and gas industry are requiring a much tighter integration of technical data into E&P business processes. The development of new systems to accommodate this business need must comprehend the significant numbers of large, complex data objects which the industry generates. The life cycle of the data objects is a four phase progression from data acquisition, to data processing, through data interpretation ending finally with data archival (Figure 1.) In order to implement a cost effective system which provides an efficient conversion from data to information and allows effective use of this information, an organization must consider the technical data management requirements in all four phases. A set of technical issues which may differ in each phase must be addressed to insure an overall successful development strategy.

The technical issues include standardized data formats and media for data acquisition, data management during processing, plus networks, applications software and GUI's for interpretation of the processed data. Mass storage hardware and software is required to provide cost effective storage and retrieval during the latter three stages as well as long term archival.

Mobil Oil Corporation's Exploration and Producing Technical Center (MEPTEC) has addressed the technical and cost issues of designing, building and implementing an Advanced Computing Environment (ACE) to support the petroleum E&P function, which is critical to the corporation's continued success. Mobil views ACE as a cost effective solution which can give Mobil a competitive edge as well as a viable technical solution.

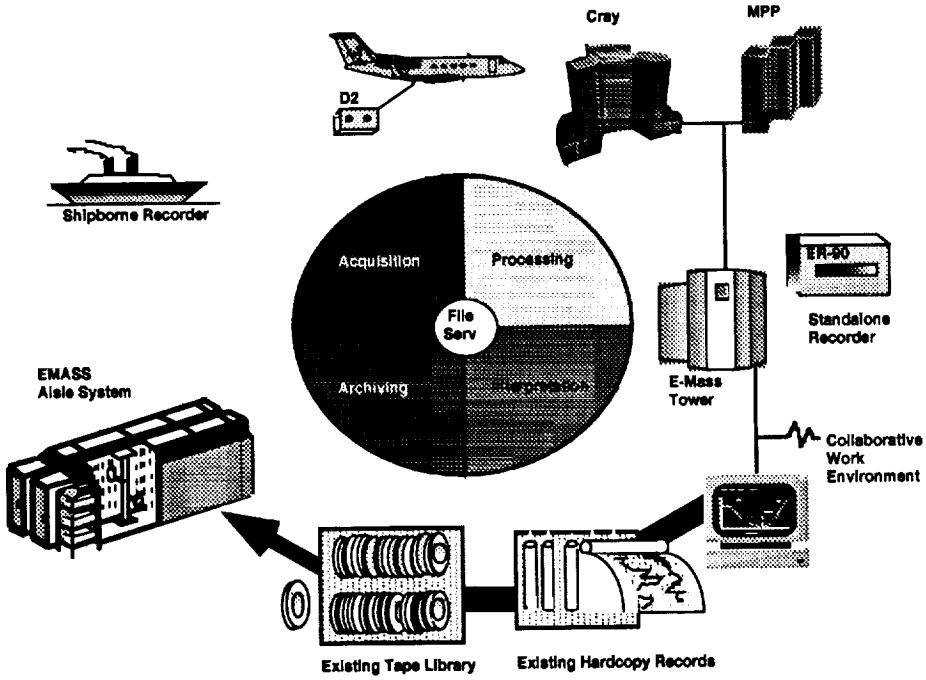



**Figure 1. Data Life Cycle**

**Acquisition**

The search for hydrocarbon accumulations requires an analysis of the earth's subsurface using the seismic reflection technique. Seismic data sets are acquired by land and marine crews over areas of interest and organized into surveys which are then transformed to 2-D or 3-D images of the subsurface. The increasing use of 3-D surveys in field exploitation has reduced the percentage of dry holes drilled from approximately 70% to 80% in the 1970's to 20% to 30% in the 1990's by providing more accurate and comprehensive geologic information. This reduction is significant when the cost of drilling a well in deep water exceeds $100 million. But the trend to 3-D, and denser spatial data sampling has resulted in survey data sets which are terabytes in size. A single seismic acquisition vessel (there are approximately 90 in operation today) may collect 240 channels of seismic data every 12.5 meters using a 2 millisecond sampling rate. This amounts to 4 GBytes of data collected each hour or a terabyte (TB) every 10 days. As Figure 2 indicates, the trend since 1965 is a 5-10 fold increase each year in the amount of seismic data collected per square kilometer surveyed.

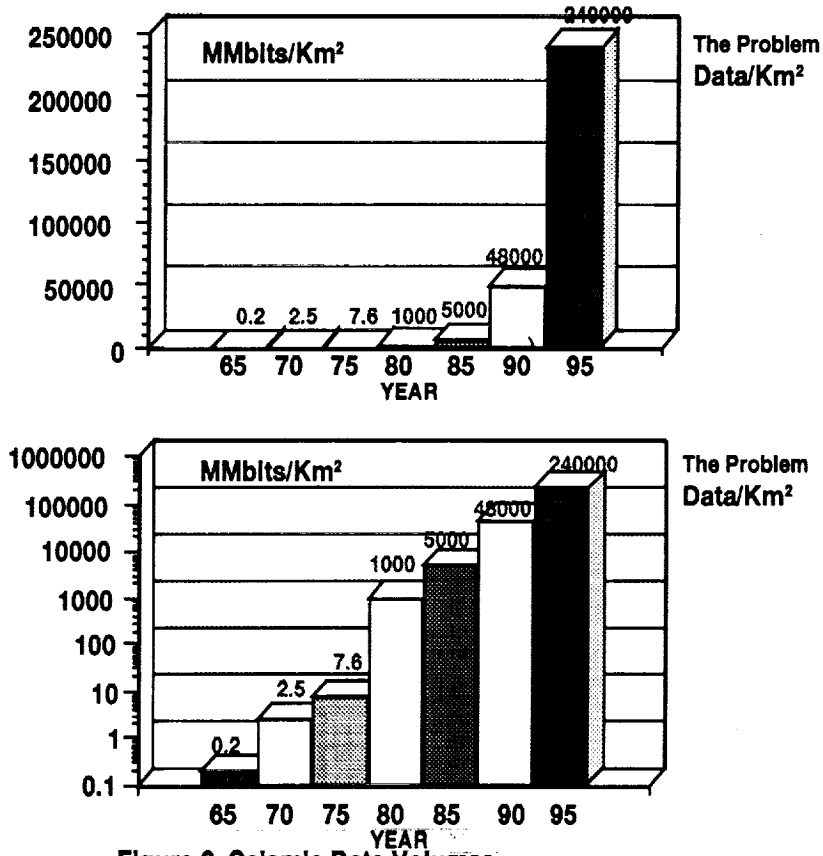



**Figure 2. Seismic Data Volumes**
**from Martin Thompson & Ian Jack, Seismic 92**

The challenge in the acquisition phase is how to contain the increasing cost of seismic data collection and storage using a standard mass storage technology which is generic to the follow-on processing, interpretation and archiving phases. The storage media technologies in use today are 9-track tapes and 3480 cartridges. A 10 TB survey requires approximately 66,000 9-track tapes or 50,000 3480 cartridges with a total media cost of $500K to $1000K. To transport the survey to be processed, as well as to replenish the supply of media, requires a costly port call by the seismic vessel. Then the land/air transportation to the seismic processing center may well exceed $100K. The bottom line to the acquisition contractor and ultimately the end user is a 3-D survey which may approach $100 million in acquisition costs!

The acquisition contractors are always evaluating the latest storage media technologies for adaptation to seismic vessels. Of particular interest today are the helical scan technologies because of the higher media densities, faster transfer rates and increased reliability. Mobil is cooperating with acquisition contractors to evaluate the 19mm D-2 media technology. The D-2 technology enables the storage of a 10 TB survey on 400 small (25 GB) cassettes which cost $17K, 140 medium (75 GB) cassettes which cost $13K or 70 larger (165 GB) cassettes costing $15K. Port calls will not be required to off-load the D-2 cassettes or to replenish the D-2 supply nearly as often and the projected transport costs to the processing centers will be two orders of magnitude lower than current costs. The ability to make a backup copy of the survey field data, something not done today, prior to transporting to the processing center is a key advantage to insuring the security of the survey against catastrophic loss. In the past, entire surveys have been lost in transit and had to be reshot. The backup of the survey will result in lower insurance premiums.

The data transfer rate and reliability of the D-2 technology is also important to the data acquisition process. Faster sampling rates and increased number of input channels in the future translate to higher bandwidth requirements. The D-2 recorder is capable of sustained transfer rates of 15 MB/sec and the reliability of the D-2 recorder has been measured at one permanent write error per TB with a 99% confidence. The features of the D-2 technology have led major oil producers such as Mobil, Shell and British Petroleum to request the development of a standard D-2 tape exchange format for seismic data by the industry standardization bodies including the Society of Exploration Geophysicists (SEG), International Association of Geophysical Contractors (IAGC) and Petrotechnical Open Software Corporation (POSC).

**Processing**

The processing of seismic field surveys to develop 3-D images of the subsurface consists of several computation steps. But before the computations begin, the field data media must be manually mounted and the data transferred into the computational engine. This step can take months in the case of a 10 TB survey stored on 9-track or 3480 media due to the thousands of manually intensive tasks required and the relatively slow transfer speeds from the 9-track and 3480 recorders to the compute engine. Estimates of the cost of this step range from $1 to $2 per media when the manual handling, data administration and storage of active data are taken into account. Mobil has minimized these costs through the use of D-2 media and the EMASS® DataTower™ from E-Systems. The DataTower™ is a robotically controlled mass storage device about the size of a soft drink vending machine with a capacity of 5.7 TB of data stored on 226 small D-2 cassettes. The D-2 cassettes are accessed within 30 sec and loaded into one of four ER90™ D-2 recorders contained in the tower, each of which can transfer data at up to 15 MB/sec to Mobil's Convex C3220 file serving computer. In the future, Mobil plans to migrate all active and archived data to an EMASS® DataLibrary™ which is scalable to a 10,000 TB data capacity and bandwidth capacity which matches any commercially available supercomputer or MPP.

Each computational step to convert field surveys to image data requires careful analysis with intermediate data sets and partial test data sets created by different algorithms with multiple analytical parameters tuned for differing geophysical subsurface properties. A large 3-D survey can take months to process on the largest vector supercomputers. Mobil is reducing the time required for each of the computational steps by using a CM-5 massively parallel processor (MPP) from Thinking Machines. The EMASS file-serving Convex platform is connected to the CM-5 by an Ultra Network Technologies HiPPI channel which sustains a bandwidth of about 10 MB/sec.

It is desirable to store the interim results of the computational steps because the process is recursive, plus the results of step n+3 may indicate that a return to step n is necessary because geophysical parameters used on step n+1 were not optimal. Today the output of the current processing step is transferred to 3480 tape media because the amount of disk required to store these results is cost prohibitive and the earlier, interim steps are therefore deleted. The output of the current step normally requires many 3480 or 9-track media and results in additional manual intervention. Another I/O bottleneck occurs when the output data set on 3480 or 9-track media is used as the input stream for the next processing step. The supercomputer incurs an I/O wait while the slower I/O device transfers the data to the CPU and this I/O wait can amount to a yearly cost amounting to hundreds of thousands of dollars. This I/O wait is reduced significantly by using the ER90™ recorders as a virtual disk, storing the output stream and then transferring the data as an input channel to the next compute process.

The long, compute intensive processing steps are susceptible to errors inherent in the data storage media. Permanent write errors in the input data stream to the seismic processing procedures can cause abnormal termination of the processing and require restarts and/or reprocessing. The improved reliability of the ER90™ recorders and D-2 media reduces the risk of these occurrences and is a major reason why other major producers such as Aramco and Exxon are seriously considering the D-2 technology.

**Interpretation**

Elements of the modern interpretation environment include high-performance X-based desktop displays, fast networks, tools for collaboration between remote sites, and seamless access to data. An advanced prototype environment, named MobilView, has been constructed to demonstrate key aspects of this environment.

Subject areas covered in the interpretation environment include:

*-Relational (drilling, geoscience, and engineering)*
*-Vector (downhole sensors, hydrography, political boundary)*
*-Array (raw seismic, processed seismic, scanned photos and microfiche, scanned paper documents)*
*-Other (grid/CAD files, multimedia, compound documents)*

The desktop user interface is geographical in nature, in line with users's mental models of the world. The user interface conforms to draft versions of an extension to the Motif style guide, which has been developed by an oil industry consortium known as the Petrotechnical Open Software Corporation (POSC). At the physical level, the underlying cartographic database has been organized using tree-structured tiling methods to ensure rapid data access over a wide dynamic range of scales.

The primary objective of MobilView is rapid viewing of large complex data objects, spanning a variety of formats. Secondary objectives include low-volume data ingest, file routing, and project archiving/recovery. Little emphasis is placed on actual computational processing, 3-D visualization, or hardcopy output. The viewing environment consists of a collection of Motif display programs ranging from purchased oil industry-specific tools to publicly-available image viewers. These are all integrated under a common shell and launcher environment that is fed by disk and cassette-based components of the storage hiearchy.

Image scanning and ingest in low volume are supported by a software environment that enables the user to pick an object (e.g., a seismic line) from the map or from a list, then scan in one or more hardcopy documents or images using a deskside scanner. The primary key association is made transparently and the user may then key in ancillary information about the scanned hardcopy. A browse-and-route function allows the user to browse through thousands of images and other large data types, and then route a file to destinations including the user's local workstation, a high-end processor (such as an MPP), or to a plotter.

Archiving functions have been developed to capture the results of long-running multidisciplinary studies. A named archive can be created and associated with a site, and files entered. Bulk data in any of the supported formats may be written to D-2 cassette and their associated metadata updated. Mandatory metadata includes an archive's geographic boundaries, thus enabling placement on the electronic map and use for later browsing. Upon selecting the archive, its contents are displayed for detailed browsing, display, or file routing over wide-area networks.

**Archiving**

By emerging standards for archive size, the needs of a large oil company represent a medium-size (one Petabyte) problem. Data ingest consists of a mix of low-rate scanning input, medium-rate transcription from low-density tape, and direct insertion of D-2 tapes from offsite acquisition and processing activities. The upper limit of a petabyte is projected to consist of:

*-hardcopy scanning = 400 TB total*
*-existing tape library = 200 TB total*
*-future (15 year) inflow = 400 TB total*
*interpretation results @10 TB/yr*
*acquisition/processing @15 TB/yr*

The requirements for an archiving function include long effective media life, scalability, and reliability. Typical data has value for 15-20 years, comparable to the nominal lifetime of most magnetic tape media. Given the large number of files per tape on D-2, the likely failure mode becomes one of mechanical wear and tear. This occurs at approximately 1000 mount/dismount cycles, estimated at 2-3 years. The EMASS FileServ software enables automated transcription to be invoked after a specified number of cycles or at a given error rate threshold.

Scalability is important for supporting physically remote offices having relatively poor data communications service. Current plans are to configure non-robotic servers consisting of a pair of ER-90 TM recorders managed by a RISC processor. With the ability of the recorders to use the large (165 GB) cassette, this gives a respectable 300 GB 'slow disk' facility. Bulk data transfer from the central archive could then be done on off-hours. The usual issues of synchronization, federation, etc. found in distributed database environments exist here as well,

At the high end, the archive must be designed to scale up from the present 10 TB systems to 1000 TB library systems. D2 Cassette replication will be needed to ensure backup and disaster recovery. Long-range technology planning for future media (optical tape, holographic) is simplified in a robotically-accessed environment having computer managed metadata. With increased data density from 3-D seismic data acquisition and the growth of full-motion video, the nominal one petabyte case may be overtaken by events later in the 90's.

**Conclusion**

The oil and gas industry is currently one of the largest application areas for high-density mass storage technology. Current immaturity of the technology and standards forces the use of rather custom systems; by late in the decade, however, off-the-shelf one petabyte systems should be readily available. At the high end, Grand Challenge problems will spur the development of large integrated systems, while sub-petabyte systems will be commodity items in use by thousands of organizations. The seismic problem is a challenging one. The global competitiveness of the U.S. oil industry depends on solving this problem.

# NASA Langley Research Center's Distributed Mass Storage System

**Juliet Z. Pao and**
**D. Creig Humes**

MS157A
NASA/Langley Research Center
Hampton, VA 23681
pao@subserv.larc.nasa.gov
humes@quickdraw.larc.nasa.gov

## Abstract

There is a trend in institutions with high performance computing and data management requirements to explore mass storage systems with peripherals directly attached to a high speed network. The Distributed Mass Storage System (DMSS) Project at the NASA Langley Research Center (LaRC) is building such a system and expects to put it into production use by the end of 1993. This paper presents the design of the DMSS, some experiences in its development and use, and a performance analysis of its capabilities. The special features of this system are: 1) workstation class file servers running UniTree software; 2) third party I/O; 3) HIPPI network; 4) HIPPI/IPI3 disk array systems; 5) Storage Technology Corporation (STK) ACS 4400 automatic cartridge system; 6) CRAY Research Incorporated (CRI) CRAY Y-MP and CRAY-2 clients; 7) file server redundancy provision; and 8) a transition mechanism from the existent mass storage system to the DMSS.

## 1. Introduction

The Distributed Mass Storage System (DMSS) project at the NASA Langley Research Center (LaRC) integrates emerging technologies from the areas of data storage hardware, high speed communications, and mass storage system software into a system that overcomes the limitations of the current approach to mass storage. The DMSS is characterized by peripherals attached directly to a network, and a workstation acting as the file server. The file server will no longer be an active participant in most data transfers because they will occur directly between the peripheral and the requesting client.

The first phase is a prototype system to provide a proof of concept. It will also provide a base for testing ideas, and measuring and tuning performance. Once the prototype system is successfully completed, the production phase of the project will be initiated. This phase will include the procurement of necessary production storage and the addition of other functionality, such as network-attached tape.

## 2. Background

The Analysis and Computational Division (ACD) is responsible for providing a Mass Storage System (MSS) to meet the storage needs for both central and distributed computing systems at the NASA LaRC. The current production MSS is implemented on LaRC's CRAY Y-MP. The system consists of a CRAY disk and three STK 4400 robotic tape libraries. The disk is managed by CRI's Data Migration Facility (DMF) software. When it fills to a site specified threshold, the DMF automatically moves selected files to the STK libraries. Files that reside on tape are transparently moved back to disk upon access.

The main access method to the MSS is through a set of LaRC-developed Explicit Archive and Retrieval System (EARS) commands (masput, masget, masls, etc.) which allow the users to put,

get, list, move, remove, make and remove directories, and change attributes of MSS files. Files are transferred over the local area network to and from the CRAY disk. Users may also use the File Transfer Protocol (FTP) which is available for most network-attached machines.

The current MSS is typical of large scale mass storage systems in use today. Each transfer results in data flowing through the file server before arriving at its destination. In order to meet high performance demands, this server is usually a supercomputer or mini-supercomputer. Because of the high cost of this class of machine, the current system has limited expandability, scalability, performance, and availability.

## 3. Goals

The primary goal of the DMSS project is to move away from costly proprietary hardware and software solutions towards an open systems approach that does not limit expandability or scalability. The hardware and software purchased and developed for the DMSS must adhere to industry standards. This will facilitate expandability, scalability, and changes to hardware and software platforms. Software used and developed must be portable so that LaRC efforts and experiences can benefit other sites with common mass storage requirements. The system must be capable of providing high-speed access to files for selected client machines (i.e. the supercomputers), while not penalizing the performance of other clients.

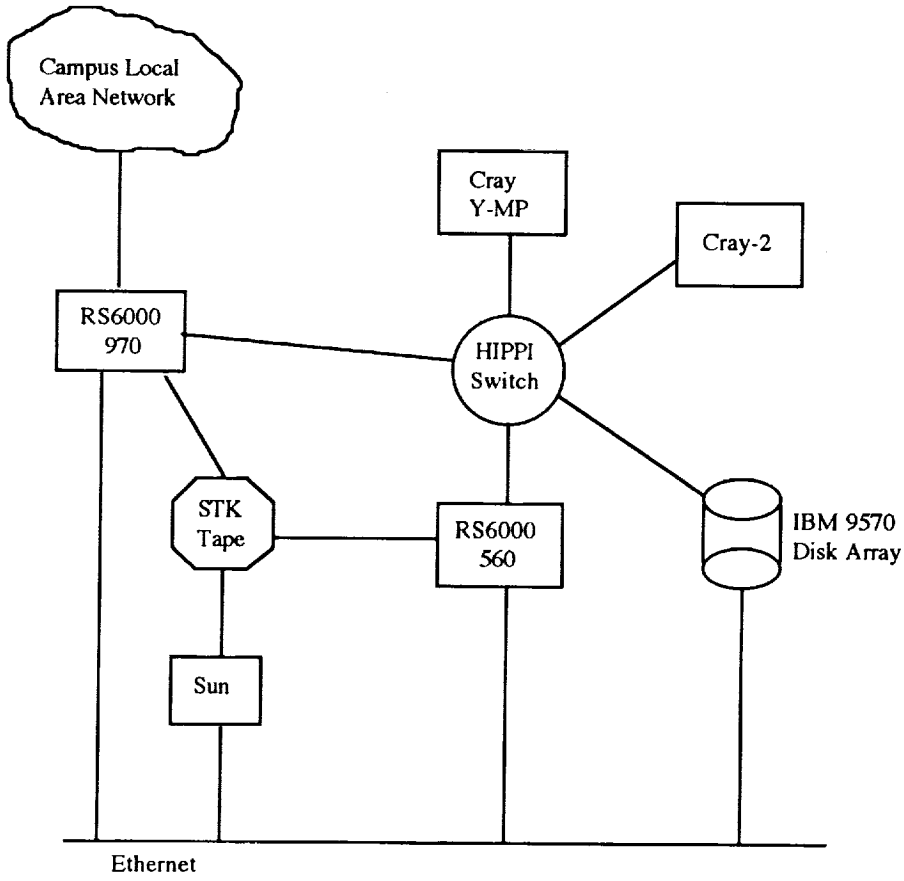



Figure 1

DMSS Prototype

# 4. DMSS Prototype

## 4.1 Equipment

The DMSS prototype [Figure 1] consists of an International Business Machines Corporation (IBM) 9570 disk array, two IBM RS6000 workstations (models 560 and 970), a CRAY Y-MP, and a CRAY-2. All of these pieces are connected to a Network Systems Corporation (NSC) PS32 High Performance Parallel Interface (HIPPI) Switch [1,3]. The workstations are also connected to the existing STK 4400 tape libraries through a SCSI interface. A separate ethernet network connects the workstations and the disk array. This ethernet is used for disk array control and tape mount requests to the STK Sun workstation.

The disk array uses the Intelligent Peripheral Interface (IPI3) protocol [4]. IPI3 commands may be submitted to the disk array via either the HIPPI interface (using HIPPI/IPI3) or the ethernet interface. Data can be directed to flow through either interface. The current disk array supports the Redundant Array of Inexpensive Disks (RAID) level 3 and supplies 40 GB of storage.

The file servers for the prototype system are IBM RS6000s. Each file server currently has 3.5 GB of local disk, 128 MB of memory, and HIPPI and ethernet connections.

The CRAY supercomputers act as clients in the DMSS prototype system. They request data transfers from the file servers. The CRAY-2 has one HIPPI channel and the CRAY Y-MP has two.

The PS32 HIPPI Switch allows up to 32 machines or peripherals to be connected. The switch allows multiple HIPPI connections without any degradation to standard HIPPI performance. Switches may be hooked together to provide more connections.

UniTree, a product of OpenVision, is a mass storage system software package which manages a storage hierarchy for files. UniTree is available on almost all open system platforms. We are currently running version 1.0 of the National Storage Laboratory (NSL) UniTree. The NSL modified version 1.7 of the general UniTree product and made numerous enhancements. The enhancements of particular interest to the DMSS project are support for HIPPI-attached disk arrays and multiple dynamic storage hierarchies. UniTree provides FTP and NFS interfaces to its filesystem and also supports distributing pieces of the system to different machines (i.e. one machine can support tape functions while another supports the disk cache).

## 4.2 Data Flow in the DMSS

Throughout the rest of this paper, components of the DMSS will be discussed in terms of the IEEE Mass Storage Reference Model (MSRM), Version 4, and the current evolution of Version 5 [5,7].

Clients of the DMSS that have HIPPI channels and the appropriate software drivers can take advantage of the speed of the disk array. These machines have bitfile client software which sends UniTree file transfer requests to the file server. UniTree then instructs the disk array to transfer data to/from the HIPPI port specified in the file transfer request. The disk array then initiates the data transfer with the requesting client's software component, called the mover, which moves data between the proper memory address and the HIPPI channel. The protocol used to accomplish the data transfer is IPI3 third-party [8].

Other clients of the DMSS, which do not possess HIPPI channels, cannot trade data directly with the disk array. For these clients, one of the file servers acts as an intermediary. The file server receives requests from them through a standard protocol (FTP or RCP). The file server then transfers data between the client (through FTP or RCP) and disk array (through IPI3 third

party). It is worth noting here, while hundreds of these clients exist and make use of the current MSS, they only account for approximately twenty percent of all data transferred.

The STK libraries are connected to the file servers and do not have HIPPI connectivity. During a file migration, a file server acts as a HIPPI client (as described above) to get data from the disk array before it writes the data to the tape. During a file recall a file server reads the data from tape before sending it to the disk array.

The initial user interfaces supported by DMSS include FTP, RCP, and EARS. All of these interfaces are explicit file transfer mechanisms which transfer complete files sequentially.

## 4.3 Redundancy

The approach for providing high availability is through redundant equipment. The production system will consist of two disk arrays, two workstations, and two HIPPI switches. This allows for the loss of any single piece of equipment without incurring lengthy down time. There are external SCSI disks that house the NSL/UniTree databases. Upon the loss of one server the other can be reconfigured to take over the functionality of the unavailable server, with access to the most up to date databases. The redundancy of equipment also allows for new system testing and development without impacting production use.

# 5. Prototype Development Work

The prototype system required LaRC to undertake development and integration work. The areas that needed development were IPI3 third party movers for the CRAY machines, user interfaces, and a mechanism to transition our current production system data to DMSS in an efficient manner.

## 5.1 Mover for the CRAY Y-MP with Model E Input/Output Subsystem (IOS)

In order to provide third-party transfer for the supercomputer client, movers have been developed for both user space and kernel space. The kernel version has been chosen for production use because it allows access to DMSS from multiple processes and fair sharing of the mover's system resource, the HIPPI channels. The user space version only allows one process to access the HIPPI channel at a time.

**Mover Interface**

The bitfile client, which is a set of NSL UniTree functions, communicates with both UniTree and the mover. It communicates with the mover by issuing transactions which consist of the following information:

function - action to be performed (such as read, write, or cancel)
transaction identifier - a 32-bit integer which uniquely identifies the transaction
buffer - a pointer to a buffer
length - the data length in bytes of the transaction
device index - the device index of the HIPPI device used for this transaction
status - pointer to a status structure associated with this transaction

When the bitfile client issues a transaction to the mover, it also issues a companion request to the file server which results in the file server issuing one or more IPI3 third-party transfer requests to the disk array system. The disk array system then sends the waiting client's mover one or more Transfer Notification Responses (TNR), each of which contains a Transfer Notification Parameter (TNP) with the following information:

transaction identifier- a 32-bit integer which uniquely identifies the transaction
offset- offset in bytes of this segment relative to the beginning of the transaction

length- data length in bytes of this segment
last_transfer_flag - flag to indicate that this request is the last transfer for the transaction identifier

The mover uses the TNP information to take action to complete the third-party transfer. One transaction request from the UniTree bitfile client may result in multiple TNRs due to file segmentation and system resource sharing requirements. The mover makes no assumptions as to the order of arrival or segment length of these TNRs. It also does not assume that all TNRs for a particular transaction identifier must arrive before it can handle the TNR of another transaction identifier. [8]

**Mover Design**

The mover maintains transaction queues and other information necessary to manage requests from multiple processes. The mover also maintains two kinds of internal buffers. It owns three large buffers used to receive the TNR and data, and many small ones used to store the HIPPI-FP (Framing Protocol) header and IPI3 command for a write request. The buffers are necessary because the mover must always be ready to accept a TNR for any transaction in the system.

The size of the large buffer limits the amount of input data coming from the disk array system via UniTree. As the buffer size increases, the number of HIPPI packets needed to perform the transfer decreases. An appropriate buffer size must be chosen to maximize performance and minimize waste of memory. The raw HIPPI driver on the CRAY Y-MP can handle a HIPPI write that has data split between two buffers. Therefore, the mover only needs to provide small buffers for the HIPPI-FP header and IPI3 command, and the user data does not need to pass through an intermediate buffer on a write. The size of the output packet is slightly larger than the user buffer size and is only limited by the maximum size of a HIPPI packet supported by the Model E IOS.

There is a set of commands to provide the following operational capabilities for the control of the mover:

- Initialize the mover environment.
- Halt all mover operations immediately (without shutting down the supercomputer client).
- Disable the submittal of transactions.
- Drop all active transactions.
- Close all HIPPI devices.
- Clear mover internal tables.
- Disable the submittal of transactions; all current transactions will be allowed to complete.
- Re-enable the submittal of transactions.
- Provide dynamic configuration capability for message logging options.
- Provide dynamic configuration capabilities for changing the time interval length for a transaction to be considered as timed-out and the time interval length to do the periodic checking.

## 5.2 Mover for the CRAY-2

The mover for the CRAY-2 is similar to that of the CRAY Y-MP, except for the handling of the third-party write. The raw HIPPI driver does not support a two buffer write. As a result, the mover's large buffers are used to pack the HIPPI-FP header, the IPI3 write command, and data into one contiguous area to be sent out with one HIPPI packet to the disk array system. So the bitfile client on the CRAY-2 can only submit requests to UniTree for transfers of size equal to or less than the large buffer size., Currently, the user space mover for the CRAY Y-MP has been ported to the CRAY-2. The porting of the kernel code began in June, 1993.

## 5.3 User Interfaces

The EARS commands have been rewritten for DMSS clients with HIPPI channels. These commands submit requests to NSL/UniTree using the supplied libnsl library. This library acts as the bitfile client and uses the LaRC developed mover for data transfer. This version of EARS is supported on the CRAY Y-MP, CRAY-2 and IBM RS6000.

Non-HIPPI attached machines have to retrieve their files from one of the file servers. These machines can get data either through FTP, RCP, or EARS. FTP is provided with UniTree. Two options are currently under investigation for providing RCP access. The first uses a locally modified version of RCP that understands how to talk to UniTree and the disk array (much like the EARS commands for the CRAYs). The second is to NFS mount the UniTree file system and use the regular RCP. The modified RCP currently works, but NFS with the disk array does not, so no comparison of performance is available at this time. The EARS interface is available to all distributed machines and is built using RCP for file transfers.

## 5.4 Transitioning From the Present DMF/UNICOS System to NSL/UniTree

The current LaRC MSS has more than a million files which comprise 1.5 terabytes of data on the STK ACS 4400 tape library under DMF management. LaRC has developed software that provides a mechanism for users to access any data in the current mass storage system on the first day of DMSS usage. The transition of DMF data into the DMSS is transparent to the users and requires minimal down time for the current system.

The day before DMSS production, the current mass storage system will be shut down for the transition process to take effect. First, on the CRAY Y-MP, a database called LaRCDB will be created using inode information of the current mass storage file system, the DMF daemon database, and the tape catalog database. The LaRCDB will then be moved to the file server. For each entry in LaRCDB, an entry will be created in the UniTree name server with a special flag set, indicating that it is a DMF formatted file. When a DMF file is accessed by a user via UniTree, the DMF flag will result in the tape file being staged onto UniTree disks using locally-developed routines incorporated into UniTree. After the staging, the DMF file becomes a bona fide UniTree file and its entry in the LaRCDB will be marked as soft-deleted.

While all the DMF files are available for UniTree users when they access them, not all of those files will be accessed by the users. So after DMSS is in production, a utility will be run on non-prime shifts to transition DMF files, cartridge by cartridge, into bona fide UniTree files until all files have been transferred.

# 6. Current Status

The prototype system is currently in a functional state. Test files are constantly being transferred, compared, and migrated. A majority of the effort now is spent testing and stabilizing the locally developed software and NSL/UniTree. The major items still in development are the CRAY-2 kernel mover and the transition software.

## 6.1 Performance of the DMSS

The initial tests of accessing DMSS data on the disk array system have been encouraging. The performance figures are grouped into three parts: disk array performance, file transfer performance to and from the CRAY Y-MP with Model E IOS, and file transfer performance between a Sun workstation and DMSS. The Sun is connected to the local area network via ethernet. The supercomputer's statistics were gathered on an idle machine, whereas the statistics for the local area network access were gathered in a normal production traffic environment. The IBM 9570 disk array system is configured using a 64K block size. All file transfer performance measurements include the whole transfer time between the client disk and the UniTree-managed disk array.

### Disk Array Performance

Figure 2 shows the performance for the IBM 9570 disk array in both the first-party and third-party modes. Third-party performance was gathered using the CRAY Y-MP as the client and the IBM RS6000 560 as the file server. The performance includes the overhead of the command and response packets sent over the ethernet for control.

### Complete File Transfer Between CRAY Y-MP and the DMSS

The timing measured is for file sizes of .5MB, 2MB, 16MB and 64MB, which are all block-aligned. Transfers that are block-aligned occur directly between the disk array and the CRAY. For non-aligned parts of a transfer, the file server is responsible for performing the transfer with the disk array [8]. In this case, the file server gets data from the CRAY's mover and places it on the disk array. This part of the transfer has been observed to take between 0.06 and 0.5 seconds.

Figure 3 compares the DMSS read transfer rates of different file sizes using large buffer sizes of 1MB, 2MB and 4MB. The graph for the 4 MB buffer case shows a decrease of performance as the file size increases from 16MB up to 64MB. This is due to the time necessary to flush the CRAY disk cache buffer. The performance of the current system is also plotted to show the increase of performance of DMSS.

Figure 4 compares the DMSS write transfer rates of different file sizes using large buffer sizes of 1MB, 2MB and 4MB. The write scenario is not limited by the large buffer size but rather the user level program's, namely masput's, buffer size. The graph shows that changing the user level buffer size from 2MB to 4MB did not yield a proportional increase of performance. The performance of the current system is also plotted for comparison. The CRAY's disk buffer cache was cleared before each transfer.

Figure 2 shows that larger buffers give increasingly better results. This is true for data transfers between the disk array system and the client's memory, but not for disks to disk file transfers. Both Figures 3 and 4 support the choice of 2MB for the mover's internal large buffer and user level program's buffer. Choosing buffer sizes larger than this gives rapidly diminishing returns due to the CRAY disk speed and the size of the CRAY disk buffer cache.

### Complete File Transfer Between the LaRC Local Area Network and the DMSS

Figure 5 gives the statistics for DMSS access from a Sun workstation on the LaRC campus local area network. Masput and masget make use of the modified RCP (on the file server) which talks directly to UniTree. The performance of the current system is also plotted for comparison.

## 6.2 Schedule

Development will continue through the summer of 1993, along with debugging efforts for existing components and NSL/UniTree. Internal test users will begin making use of the system sometime in August and will use the system for a two month evaluation period. If the system is stable at this point selected users from the research community will be invited for a one to two month beta-test, followed by full production use by the entire research center. A second 40 GB HIPPI-attached disk array, external SCSI disk, and second HIPPI switch will be added to the configuration before production usage is initiated.

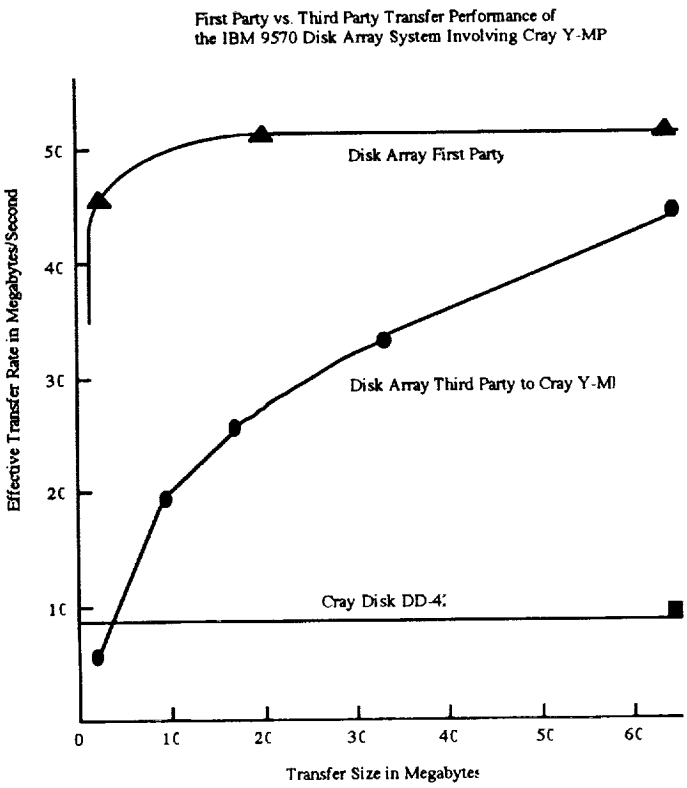



Figure2. Performance comparison among the first party disk array transfer rate provided by IBM, the third party disk array transfer to/from Cray Y-MP using LaRC mover, and the sustained transfer rate of the Cray DD-42 disks.

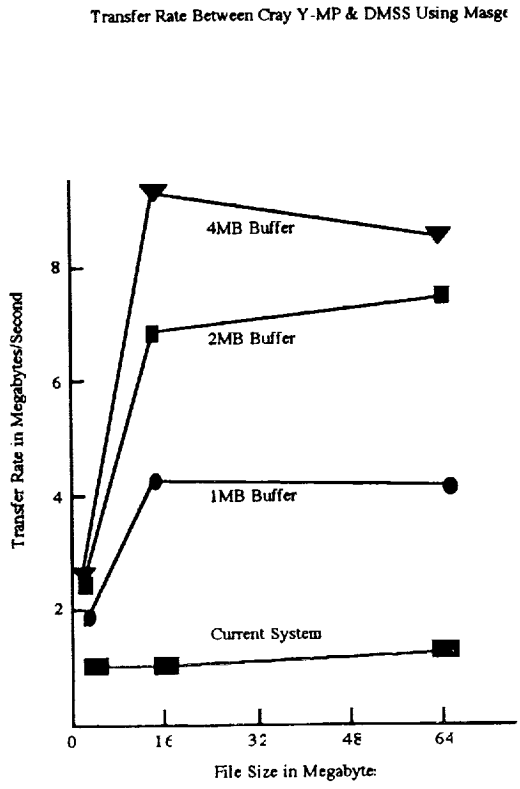



Figure 3. Transfer rate comparison of masget using differe sizes of buffers on the Cray Y-MP.

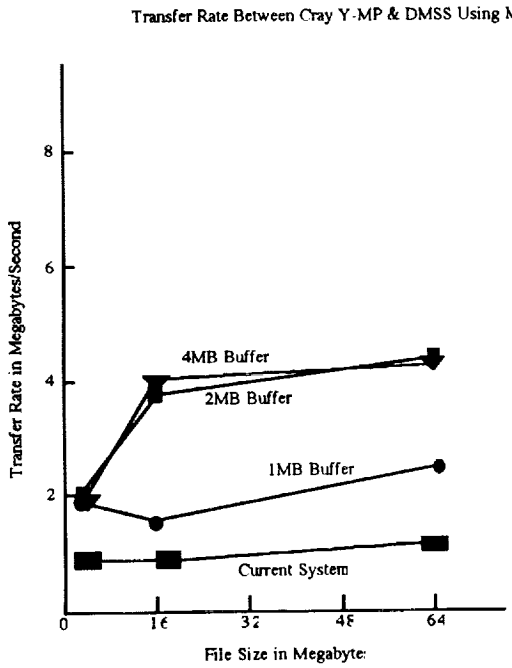



Figure 4. Transfer rate comparison of masput using differe buffer sizes on the Cray Y-MP.

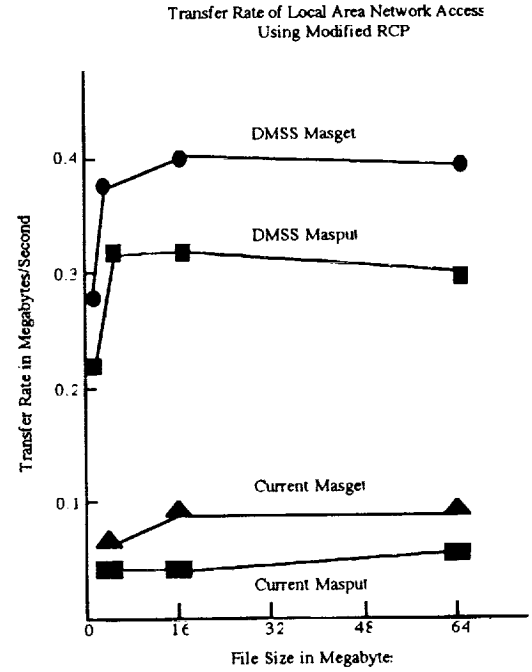



Figure 5. Transfer rate comparison of masput and masg used from machines on the LaRC local area network.

## 7. Future Plans

Once DMSS is stable, other features will be added. Of particular interest is a file system interface (using vnodes). The first supported interfaces are all disk-to-disk file transfers. There is also a need for high performance data transfers directly between an application on the CRAYs and the disk array. Currently the only way to do this is to incorporate the libnsl routines directly into a program. This does not give the users file location transparency, thus placing an unnecessary burden on the users. A transparent file system interface would allow for extremely good performance for jobs running on the CRAYs, while maintaining location transparency. In this way all permanent file storage for the CRAYs can be managed by DMSS.

Also of interest is a site-wide distributed file system that will be able to use the DMSS to store data. For example, this could be based on OSF's DCE/DFS.

Other machines with HIPPI attachments will have movers developed to enable high speed DMSS access. The next machine targeted is the Intel Paragon.

LaRC will also pursue adding network-attached tape to DMSS. This will relieve the workstations of more than 95 percent of the data transfer responsibilities of the current CRAY Y-MP based MSS. Migrations and recalls will occur directly between network peripherals. As the multiple dynamic hierarchies mature, applications, such as backup and visualization, will move data directly to and from the network-attached tape.

## 8. Conclusion

When DMSS goes into production in the fall of 1993, it will relieve the CRAY Y-MP of its function as a file server. Users of DMSS will experience performance three times better than the current system. Their access to DMSS will no longer be interrupted by the file server's unavailability due to various system maintenance functions, malfunctions, or system time. The system will be expandable and scalable. Disk and tape will be added directly to the network as the need grows. If one file server is not powerful enough to handle the workload, then the function can be split among two or more file servers.

## 9. Acknowledgments

The LaRC prototype DMSS system has gone through the cycle of design, acquisition, testing and software development since January 1991. The acquisition took the initial one and a half years. We would like to acknowledge Everett C. Johnson and David E. Cordner of the Computer System Branch at NASA LaRC for their help in the design and acquisition of DMSS equipment, the Unisys Cooperation for their support in software development and testing, and CRAY Research Inc. for their support on the UNICOS internals. We also appreciate the cooperation of DISCOS of General Atomics (presently OpenVision) and IBM Federal Systems Company.

## References

1. ANSI, "High Performance Parallel Interface - Mechanical, Electrical, and Signaling Protocol Specification (HIPPI-PH)", American National Standards Institute, X3.183-1991.

2. ANSI, "High Performance Parallel Interface - Framing Protocol (HIPPI-FP) Preliminary Draft", American National Standards Institute, X3.210-199x.

3. ANSI, "High Performance Parallel Interface - Physical Switch Control (HIPPI-SC)", American National Standards Institute,X3.91-023-1991.

4. ISO/IEC, "Information Technology - Intelligent Peripheral Interface Part 3: Device Generic Command Set for Magnetic and Optical Disk Drives", ISO/IEC 9318-3, September, 1990.

5 Coleman, S. and S. Miller, eds., "Mass Storage System Reference Model Version 4", IEEE Technical Committee on Mass Storage Systems and Technology, May 1990.

6. Coyne, R. and H. Hulen, "An Introduction to the Mass Storage System Reference Model, Version 5", Proc. Twelfth IEEE Symposium on Mass Storage Systems, Monterey, April 1993.

7. Merrill, J., "Toward a Standard IEEE Mover", Proc. Twelfth IEEE Symposium on Mass Storage Systems, Monterey, April 1993.

8. Hyer, R., R. Ruef, R. Watson, "High-Performance Data Transfers Using Network-Attached Peripherals at the National Storage Laboratory", Proc. Twelfth IEEE Symposium on Mass Storage Systems, Monterey, April 1993.

# Invited Panel: User Experiences with Unix Based Hierarchical File Storage Management Systems

DR PRATT: The Panel moderator will be Dr Sanjay Ranade, who has a bachelor's degree in aeronautics and a Ph.D. in computer science. He worked at NASA/Goddard for eight years. He helped to design and develop a high-performance network fileserver for Hughes STX, and now has his own company, Infotech S.A., Incorporated.

Sanjay?

DR RANADE: Thank you. Can everybody hear me okay? I'd like to start off by introducing the panel. The topic is User Experiences with Unix-based Hierarchical Storage Systems, and we're going to refer to these things as HSM or File Servers or whatever. But that's the main topic-Unix-based only.

The first person I'd like to introduce is Mike Daily. I won't go into a big discussion of him because he was already introduced earlier. Mike is from Mobil, and he has experience with the FileServ software.

The next person is Ellen Salmon, who works with Hughes STX supporting NASA's Center of Computational Science. She's a principal systems programmer, and she has worked one and a half years with the UniTree system on the Convex machine at Goddard. Prior to that, she has eight years software support experience, also at Goddard.

John Garon is a computer scientist at NSA. He has an MS in computer science and a BS in mathematics. He's been developing software for data archive data bases and software analysis, and he has experience with Advanced Archive Products'AMASS software.

Thomas Woodrow is from NASA Ames Research Center. He's a Scientific Analysis Software group leader. He has a BS in computer science from Hobart College and some very apt experience here, because he was recently asked to perform an evaluation of the Unix-based HSM software and he has written up a nice paper which we had a chance to look at yesterday. I am sure he will be telling us of his experiences. Included in his evaluation were DMF from Cray Research, UniTree, FileServ and Nastor.

Joe Marsala is from the Supercomputing Research Center in Bowie, Maryland. He has a BS in mathematics from Texas A&M, and he has worked with the EPOCH storage management software over the last few years.

Suzanne Kelly is from Sandia National Labs in Albuquerque, New Mexico. She is a Distinguished Member of the Technical Staff there. She has a BS in computer science from the University of Michigan and an MS in computer science from Boston University. Sue is the president of the UniTree Users' Group. She has ten years' experience maintaining HSM software storage systems. She's very well known in the UniTree community. She's involved in the HPSS software development work for the National Storage Lab.

So, having introduced everybody on the panel, I just want to give you a summary of how we are going to try and do this panel discussion. The first thing is I'd like each panel member to just introduce themselves, what they do, what their installation is like, basically give a little synopsis of their experience there.

Then we have a bunch of discussion topics. After we've been through the panel, each one describing their experience and so on, we have ten discussion topics. We will step through each one, one by one, and I will ask the panel members to comment on it. Anybody in the audience who wants to, can chime in and say whatever you like. You can ask questions at any time. Don't be shy. Just raise your hand and ask whatever you like.

Let's try to keep this really informal and productive and interactive so that we have more of a dialogue rather than people here lecturing to people over there. Let's try to keep it informal.

So, why don't we start with Mike? Do you want to say a few words about yourself and your installation, and we will go on down the line here?

DR DAILY: Well, I'm a geologist by training, so I don't know that much about all the technical aspects. As I said in the talk, we're FileServ based, with a Convex front end. The evolution that we see is that we will have direct connection in due course to things like the Connection Machine. Our installation is intended to be very diverse, so it is supporting not only supercomputer-type processing but also wide-area access by workstations, and also data archiving.

Our definition of archiving is not deep storage; it's more sort of a back end store for what will eventually be several hundred terabytes of data. We are committed entirely to open systems. So we started this thing in the Unix world and have no intention of moving from there. So in that sense, I guess we're not carrying a whole lot of baggage with us.

What were some other -- we're not going to turn to the ten questions yet, are we?

DR RANADE: No.

DR DAILY: Okay. So those will come out in due course. I guess that will do as a capsule summary of what we're up to.

MR WOODROW: I'm Tom Woodrow. I am a manager for Computational Fluid Dynamics (CFD) Visualization Developers and Parallel Software Tool Developers. I provide support for users who are trying to analyze CFD data sets which range in size from 50 GB - 1 TB. In an attempt to support users with very large data sets, I borrowed a Storage Technologies robotic tape silo, attached it to an existing Convex Visualization System and ran a UNIX-based HSM called Convex Storage Manager (CSM).

Later, when our organization needed to make a decision on whether to go into production with a home grown HSM, NAStore, or a commercial alternative, my experience and the fact that I was not involved with Storage Development made me an ideal candidate to conduct the review.

Our environment consists of 2 Cray C-90s which generate CFD data sets. We currently have 2 production HSM systems deployed at the center, one is a dedicated Cray YMP2E running Cray's Data Migration Facility (DMF), the other is one of the C-90s which runs DMF to keep scratch disks relatively free. The use of DMF on the C-90 system is tolerated because it allows us to keep scratch disk space free and the CPU load does not appear excessive. We are about to place 2 dedicated Convex C3820s into service running NAStore, a locally developed UNIX-based HSM. The volume of data and daily flow into these systems is approximately as follows:

| | |
|---|---|
| YMP2E | 1.3 M files, 5 TB, 7 GB/day |
| C-90 | 31 GB/day |
| Convex | 2.2 M files, 3.7 TB, 4 GB/day |

MS KELLY: Hi. I'm Sue Kelly, and I wanted to talk to you about what Sandia National Labs' Scientific Computing Directorate has for file servers. We have four file servers, two in Albuquerque, New Mexico, two in Livermore, California. In each site, one is doing classified file serving and the other is doing unclassified file serving.

All four systems are pretty comparable in architecture. They're all based on Convex C2 or C3 CPUs. They have on the order of 100 gigabytes of disk on each of them, and they have one or two Storage Tek silos as the archive. They interface to networks via FDDI and two of them that interface also have an UltraNet connection to Cray Y-MPs.

For client nodes, there are approximately 500 on the classified network, and 500 on the unclassified network. The nodes are one Cray on each of those networks and an assortment of HPs, Suns, Macintosh, Silicon Graphics. And that's pretty much the hardware side of it.

On the software side, all file servers use the UniTree 1.5 version from Convex. That means the access methods are NFS and FTP to the UniTree system. In the case of the Crays we have UltraNet, because we use native Ultra FTP to communicate, and that provides us higher performance. So in deference to previous speakers, we do not have performance requirements problems with UniTree systems. They satisfy our needs, and in all cases it has been networking, protocol stacks or the client that has been the slow part of the file transfers to the UniTree system.

For the rest, when I discuss these systems -- because there are four of them, it gets confusing to differentiate to you, who certainly don't care about my four systems -- I'm going to refer only to the one system which has the longest lifetime. It's been around for 18 months, has 1.2 terabytes of data on it. It averages only about 5 gigabytes a day of traffic and grows by about 1 gigabyte per day.

This system manages about 277,000 files. I think that's about all I wanted to say for the hardware and the software environment. I look forward to your questions.

MR GARON: I'm John Garon. I have been working at NSA for 18 years and, for the last 12 years, one of my responsibilities has been in the area of mass storage systems. I began by developing software to interface to the Bragaen Automatic Tape Library systems attached to CDC Cyber 176 and Cyber 84's. Although we were using commercial equipment, we had to develop all of our own software since we used a home-grown operating system and programming language. With the introduction of UNICOS around 1987, we began to explore the use of commercial software to replace the storage control software used to drive our hardware storage systems.

In the late 1980's, my office initiated the ABUNDANT requirements that Mike Shields talked about earlier. My office is no longer the customer for that project, and it is now being developed for another NSA customer.

We are still using internally developed software to perform file and volume management on our main storage products, the STK silos, using Crays as the host. My concern is that the software will become unmaintainable as people leave the project and that it will eventually not satisfy our growing requirements. In the late 1980's, we thought that hardware would be the limiting factor in solving our storage needs, but over the last few years, it appears that industry will develop the hardware and storage capacity to satisfy our requirement. The problem seems to be in the software to control the hardware, and to perform file management to those high density robotic systems.

I have had experience with the AAP product in our office on optical and Metrum VHS tape systems. Although the AAP applications do not store nearly the amount of data that goes to our silos, the functionality of the product is very close to our requirements. My problem is in the control software, and I think the product that will satisfy my requirement has its basis in the AAP product. I have plans to begin working with my systems developers at NSA to determine whether it is desirable to have the AAP software enhanced to work on Crays to interface to our silos.

MR MARSALA: Hi. I'm Joe Marsala of the Supercomputing Research Center. We're a relatively small research house, about 140 people. We have 300 workstations, a Cray 2, a TMC CM2 and a TMC CM3. Our backbone network is FDDI. We've got about 13 or 14 relatively large servers, and one of those servers, our archive server, is an EPOCH 1 Optical Jukebox System. After hearing all the massive storage requirements here, I think I'm here to provide the comic relief.

MS SALMON: Hi. I'm Ellen Salmon. I work for Hughes STX supporting the NASA Center for Computational Sciences at Goddard. We have about 1,200 space and earth science researchers who are users of our facility. So they have a great divergent group of requirements themselves.

The facility itself has a primary compute server in the form of a Cray C98, and it has six processors. That itself has a Storage Tek silo and runs DMF for a 21-day archive. After 21 days, the data is purged from that system.

We are running UniTree 1.5 on a Convex C3820. Our Convex/UniTree system has three Storage Tek silos that are within a couple of hundred tapes of being completely full. We have about 5,000 vault tapes from our UniTree system. We've got about 105 gigabytes of disk cache. We have about 3.3 terabytes stored at this point.

Our UniTree system has been operational a little over a year at this point, so we've gone from nothing to 3.3 terabytes in a year's time, and one of our big issues, of course, is handling that volume of data. We do have UltraNet connectivity between the C98 and the Convex. UltraNet is the route where most of our data comes, and the Crays are the primary storage client.

We are expecting, as far as requirements are concerned, that our transfer requirements are going to have to be even bigger than they are now. At the moment, we see in the neighborhood of 50 to 70 gigabytes of traffic a day into and out of our Convex/UniTree system. Depending on the day, we can see more gets than puts. We allow only FTP access for reasons of performance. Recently, we've been seeing on the order of 30 to 50 gigabytes of new data a day.

Probably our primary concerns at this point are issues of network robustness and the ability to write enough data tapes fast enough to keep up with the data coming in from the Cray; we're also very concerned about the fact that we also have an IBM system from which we have 1 to 2 terabytes of data to transfer into our UniTree system. Right now, we're going from 3480 technology on the IBM system to 3480 technology at this point with our UniTree system, and we'd like to see higher density. So, at least we would have our storage in a smaller area and not just moving the data from one kind of system to another.

DR RANADE: Okay. Before we get to the questions, I just want to say a couple of things to set the background, as it were. First of all, the market for mass storage systems, many people look at it as being composed of three segments: the small segment, your workstation, LAN, file server; the middle segment, which is often commercial market; and the top end, high performance, high capacity, which some people call the lunatic fringe of the market.

So that's one way to break down the mass storage picture. Another way is by the type of system that's really needed in a given case, and the four cases that I can say, there is what is called the virtual disk, which is just one machine with extended storage. There's the network file server. There's the backup and recovery, and there's client migration. There are four different kinds of software there.

I just said that because we're not comparing apples and apples, and we're not talking about the same thing. We're talking about different kinds of software for different requirements. So having said that, I'd like to ask each of the panel members how they developed their requirements. What process did you go through to come up with your requirements? Or did you go through a process to come up with your requirements?

Mike?

DR DAILY: I guess looking back into the deep dark past -- five or six years ago is when we started doing this. Originally, our use of this technology was envisioned in the grand scale, which is kind of how it's turning out. And then about halfway through its life or the development cycle, it got sort of pinched down, and then it has subsequently re-expanded. So let me just mention that.

Five or six years ago we looked at it primarily as a back end to supercomputers and as a replacement for the tape library. So the idea was that we had this pretty compelling economics of projection of a couple million tapes sitting off the tape library with capital costs of that running $20 million or $30 million just for media and $4 million to $5 million a year for managing those tapes.

I don't know if any of you have ever worked with round tapes especially. You actually have to be like these people at brindle champagne where you go in and quarter turn them every three months to straighten out the magnetic flux lines and all that, some sort of weird physics involved in these large amounts of magnetic media.

So the two drivers at the time were replacement of the tape library and the back end for the supercomputer. With E Systems we did a lot of numerical simulation about how many recorders and latencies and all that sort of thing and put that case together.

At the time we also recognized that there would be a future need for things like serving workstations over wide area networks, but that was not explicitly part of the justification. About halfway through the project the focus narrowed just to replacing the tape library, so there was little attention paid by the people that were managing at that time on these other things.

Then about a year ago, things opened back up again. So I guess the long and the short of it was that there was a lot of thinking done, constructive thinking, and now it has rewidened with all these opportunities which have come available, especially with faster workstations.

DR RANADE: Is it possible for you to say what proportion, what is the ratio between the money you spent on developing your requirements compared to the money you spent in buying the system? The reason I ask is my own experiences, having worked with the procurement, which is about a million dollars, the government spent $200K on developing the requirements and doing the spec. So how does this compare with your experience?

DR DAILY: Multiply that by ten in both cases and you're about right on.

DR RANADE: Okay.

DR HARIHARAN: (Off microphone.)

VOICE: Is that a later question?

DR RANADE: Yes, we can come to that.

DR DAILY: Do you want me to go ahead, though?

DR RANADE: Go ahead, yes.

MR WOODROW: Okay. I wasn't around at least to develop or participate in the requirement discussion for how we got going. I can talk a little bit about what model I know we use. We've been driving the requirements for how much storage space we needed basically by the solution development capability that we have in the Crays. We have an idea of how fast the systems are, what the canonical grid size is for a CFD data set and about what we can produce per day.

Unfortunately, most of the data that is produced is saved forever, whether it's good or not. So we're not terribly aggressive to go out and get people to throw away the data set that they don't really need to keep. At least I know that that's part of the model. So we're talking with users to identify how big their data set is and we multiply by the capability that we have to produce on the Cray.

One of the factors that's making things more difficult for us now is we're going from a situation where people are generating a single time step to generating a hundred or ten thousand time steps. So we're seeing that individual users are increasing their output tremendously, and what they want to look at later.

Okay. So that talks about at least how I believe we derive the requirements for the production systems that we have on the floor. For the purpose of the evaluation that I ran, I did the same kind of evaluation. I sat down with users. I talked to them about data set sizes, and I also took a look at the population breakdown for what we have on our production system. Then I put together a workload that reflected user needs and population breakdown.

DR RANADE: Does anyone in the audience have anything to say at this point on requirements? Any comments? No? Sue?

MS KELLY: Yes. We did a very detailed requirements study in order to purchase the system. It was a competitive bid, so the requirements study was translated into a Request For Proposal. We spent approximately $300,000 for that requirements study which resulted in an acquisition of $3.3 million. Different color money, however, capital versus expense.

DR RANADE: So it's 10 percent roughly.

MS KELLY: Yes.

MR GARON: I have no idea what it cost to gather our requirements. We have two sites that I am familiar with using the AAP AMASS product, and I was not involved in either procurement process. My office has the AAP product controlling two optical units. The other site has two Metrum systems, a 600-cassette and a 48-cassette system.

How we got the AAP product in our office was rather by chance. I stumbled on the two optical units that were a by-product of the ABUNDANT program. They were not being used, so I borrowed them and discovered that the systems were managed by the AAP software. So we re-initiated the AAP license and found, to our surprise, how functional the software actually was. Now we are investigating other platforms where the AAP product may be useful.

MR MARSALA: Well, see, at SRC, my group's function is primarily to support our research user population. So we basically developed requirements by talking to those users, looking at some historical data, and a scan of available technology. I couldn't give you any idea what it wound up costing. The evaluation assistance later wound up being a whole lot more than gathering the requirements.

MS SALMON: I wasn't in on the whole procurement process, but I understand ours was one that started five years before the final product was accepted, kind of a large-scale government procurement type of thing where, at least initially, I think the need for storage, *et cetera*, was greater than what was available on the market.

As far as requirements, we had an existing, and still have an existing, IBM MVS-based HSM system. Processing done on that system was primarily satellite data calibration, *et cetera*, very I/O intensive work. The other big use of data, of course, is our supercomputers, the Cray C98 at this point in time. I know the procurement process was pretty thorough in trying to understand what the satellite processing requirements were going to be and including major users and asking for their trends and trying to look into the crystal ball and seeing what the computers, the supercomputers, of the future were going to require.

That's pretty much what I can tell you about our requirements.

DR RANADE: Anybody in the audience from Goddard who has something to say on the requirements development? Because Goddard had an interesting experience. They purchased one mass storage system, and then they bought another one. And I think a lot of it had to do

with the requirements being reformulated or whatever. Anybody want to comment? No? Okay.

The second one -- and let's start with Tom -- how did he develop acceptance tests or benchmark tests? Did you have a need to have acceptance testing or benchmark testing? Did you write your own? Did you go and talk to other people, borrow theirs?

MR WOODROW: For the HSM evaluation I just completed, I created my own set of benchmarks to stress disk, tape performance and that of the HSM product. These tests included individual peak performance tests as well as a simulated user workload. I was interested in pointing out differences between several alternatives rather than testing out a system before it went into production. Our Mass Storage Groups ran extensive acceptance tests on NAStore, the system we recently placed in production. These tests were oriented towards verifying functionality and performance, reliability, stability and failure testing, and an extensive beta testing period. We had access to the experience of two production HSM capabilities on site and were able to develop extensive test suites.

DR DAILY: Going back to requirements for just a second, I wanted to know if any of the panelists had the experience of, in the process of the requirements, having seriously underestimated their total capacity? Have they filled up their systems dramatically faster than they had originally anticipated? Or were they always aware that they were dealing with a very short time constraint? Because it sounds like a number of the panelists are already pushing up against the limits of their existing systems.

MS SALMON: It's my understanding -- and once again, I wasn't in on all the details of the procurement for our particular system, but if the money had been available-- by the time things finally came through, we would probably have initially obtained two to three times the storage we have now with, of course, growth capability. So, to a certain extent some people felt that we had overestimated the rate at which we would be storing data. But it's pretty much gone according to those who felt we were going to be storing more data than what prevailed budgetwise.

DR RANADE: Okay. Anyone else?

MR WOODROW: We're also seeing data coming in from other sources than we had earlier anticipated, so that's not a major increase. But we did not expect to see the volume of data saved on the Cray that we are seeing, and it's causing us to rethink the way that we stay in production with our service.

MS KELLY: For our requirements, every capacity requirement also had a requirement for an order of magnitude expansion beyond what was already there. So we bought a system that can be expanded quite readily.

DR DAILY: I think that's pretty much our experience, too. We chose the solution that we did because of the very large dynamic range and size. I think we are pretty much on track for the sizing that we did but for the wrong reasons in the sense that we anticipated 200 or 300 terabytes a few years out. That was based on the idea that we were going to transcribe the existing million and a half tapes in the tape library, because no one has the guts to throw away existing data.

Well, since then, with the travails of the oil industry, people have gotten a lot more courageous about it. We're putting them into deep storage in salt mines. So, our guess now is we're only going to transcribe about 20 percent of that one and a half million or so, but it's being made up for by the much higher data densities that we're getting in seismic acquisition now, gigabytes per kilometer of line mile, that sort of thing.

DR RANADE: Moving to the next topic: how did you evaluate the software? What process did you go through? Can we step through, let's have a bit more speed, because we've got a lot of topics, and we're at 5:00 o'clock now, I think, aren't we?

VOICE: 5:30.

DR RANADE: But right now it's about 5:00?

Okay. Sue, do you want to start on this one?

MS KELLY: Well, it was a competitive bid, so that's how the evaluation was done. To kind of pick up on question number two, we did develop a set of benchmarks for evaluating the various solutions that were offered to us. The evaluation and the benchmark criteria were part of the $300K investment we made in the requirements.

MR WOODROW: I can say a couple things.

MR MARSALA: Well, we didn't do a benchmark per se. We took the requirements that were at a more functional level and did a validation/evaluation of all of those, including some transfer times and that sort of thing. But that was basically the extent at which we evaluated it.

MS SALMON: For us, the part of the procurement was also acceptance criteria. Basically, the product had to satisfy the acceptance criteria, and there's a list of them. We kind of had to go through one by one and show that they could be met.

DR RANADE: Mike, do you want to go?

DR DAILY: Our selection was really driven by some of the requirements for the media itself. We have pretty stringent requirements on bit error rate, like $10^{-12}$. We needed bandwidths of 10 to 15 megabytes per second. The large capacity per cassette is to minimize the handling, so we wanted these 10-, 20-, 30-gigabyte cassette sizes instead of sub-gigabyte, and the scalability left up to libraries.

At the time that we really got into writing checks and things like that, about the only thing that we saw out there was the stuff that our cousins in Fort Meade are doing. So I think it ended up being pretty much of a sole-source sort of thing.

MR WOODROW: We had to justify why we would continue going with Nastor as opposed to one of these commercial alternatives that certainly are getting a lot of use. So we brought in UniTree, we brought in FileServ and DMF and ran them on systems in-house for about three months while also running Nastor. We ran a number of different benchmarks across all of them, and then we basically rated all of them for performance, functionality, ease of use, stability as much as we could determine in a short period of time, and ranked them and made a decision: in the end, to stick with Nastor since there is no additional development that needs to be done. Basically, because of a cost decision at the end, it's the lowest cost one for us to go with.

DR RANADE: It was the lowest cost one?

MR WOODROW: It was the lowest cost at the level of functionality and performance that we wanted. Basically, the result of our evaluation was that DMF, FileServ, and Nastor were all very, very close in terms of performance, ease of use, functionality, and that DMF was behind primarily on a performance basis. I'm sorry, UniTree was behind primarily on a performance basis.

Question?

MR JIMMY BERRY (DoD): (Off microphone.) How much did it cost the government?

MR WOODROW: I'm sorry. Could you repeat that?

MR JIMMY BERRY (DoD): You indicated that your own internal system was the lowest cost. What value did you assign to the government resources that were used to produce these?

MR WOODROW: We assigned a cost of $0 to NAStore. This clearly does not take into account any of the development costs that have gone into it. However, given that we are faced with a choice of several alternatives, all of them cost real dollars for us to acquire, except NAStore. These costs are not trivial, especially when dealing with a tape inventory of significant size. Most of the commercial packages are priced based on size of the inventory or on the number of robotic tape units. For an installation like ours where we have eight 3480 silos, the cost of a commercial license is large.

MR JIMMY BERRY: How do you account, then, for the subsequent releases in the operating system, changes in the environment? I mean, for example, there's some of the other people that are running on like a release 1.5, which is about two releases back on even the commercial products.

MR WOODROW: We recognize that whether we run a commercial or home-developed HSM, we need a staff who understand the product in detail. In fact, we require that the local staff can build the product from source code on site. With this in mind we believe that OS upgrades for a home developed HSM can be accommodated locally without significant additional cost.

For the four packages in the HSM Evaluation, we looked at startup and recurring costs. We estimated that we would require a local staff of 2 for a commercial package and 3 for NAStore, to find and repair problems (yes we do this for commercial packages too) and add features as required..

Based on a one-person difference between a commercial HSM and NAStore, significant start-up costs and the fact the NAStore was very strong from a performance standpoint compared with the other alternatives, we chose to go into production with it. This decision makes sense today. When we started development of NAStore in the mid 80s, there were no UNIX-based commercial alternatives. The Storage decisions we make in the future will again be a cost/performance tradeoff and will likely be tipped in favor of a commercial package.

DR RANADE: Are you happy with that answer?

MR BERRY: (Off microphone.) Well ... (*laughter*)

DR RANADE: I'm not either. I mean I'm not --

MR WOODROW: You're not.

DR RANADE: Well, let me rephrase it. I'm not unhappy with it, but what I'm thinking, isn't this the case everywhere? I mean, wouldn't this be the justification in any place where they have a home-grown mass storage system? For example, does the Census -- go ahead.

MR WOODROW: We recognize that continued development on an in-house package makes less sense in light of current commercial alternatives. We do not intend to continue development of NAStore. It is useful as is and can be sustained at a competitive cost. At this time, factoring in cost, performance, and features, the balance is in NAStore's favor. As time goes on, the commercial alternatives should improve, and the balance will tip in their favor. We welcome this and will continue to evaluate our situation in light of the market.

DR RANADE: Anyone else on the panel? No? Okay. Let's move to number four. We have now developed the requirements, we've done the benchmarks, we've evaluated and now we're up to installation. Were there any special events or something you wanted to communicate to

potential buyers about the installation phase, something that you learned and which you wouldn't know otherwise about any of the software packages?

MR MARSALA: Well, at SRC we sort of learned remembering back that our primary function is supporting our research users. While we go a lot of input about the functional things that they wanted to do, when we implemented it, we implemented it about as user unfriendly as we could have, and, of course, the users didn't use it, which brought to our attention that it wasn't being used. After a little checking, we found out that maybe if we did a little more homework, we'd have it right. We now have our archive mounted as normal user UNIX file systems, and users don't seem to have any problems anymore.

VOICE: (Off microphone.)

MR MARSALA: FTP, telnet, and, of course, they hated it. I mean, it sort of makes the assumption that you have a knowledgeable UNIX user with lots of time, and both of those assumptions are bad.

VOICE: (Off microphone.)

MR MARSALA: Right. It's now NFS-mounted.

MS KELLY: When the system went into production, I had a 3-month hard deadline for decommissioning the old file system, which had about a terabyte of data. That was by far the most painful experience, migrating the old data to the new system, while we were still learning how to operate it. Of course, we didn't quite have our administration guide and all our procedures down pat on day one. So the conflict between getting the data off the old system at the same time we were trying to learn how to run the new system was a very painful experience.

I don't know if I should elaborate too much, but we didn't spend enough effort on the scripts for transferring the data. And yes, we chose to transfer the data rather than a cut-over date where the old system went away and the new system came on-line. We didn't spend enough time on recovery on the scripts. We didn't spend enough time on statistics to tell how we were doing. Operations had to dedicate one person 24 hours a day for those 3 months, and during that time an analyst worked 7 days a week, just making sure that everything was running all right.

DR RANADE: Mike, do you want to say something?

DR DAILY: I guess the only lessons learned were the typical things that you learned when you've got a complicated system: a fair amount of finger pointing, problems with software revs with mismatches, FTP daemons misbehaving and all that sort of stuff. I think if we had to do it over again, we would have tasked E-Systems a little bit harder to be the total system integrator rather than maybe doing a few end runs around them, or we would go chat with Convex about something. Pinpointing accountability and this sort of stuff is important, especially if you're not trying to be in this business.

MR WOODROW: Two points: 1) make sure data gets out to tape daily (don't allow a backlog to develop); 2) do regular backups of the file systems. Both of these are things that seem obvious, but can pass you by a little at a time..

DR RANADE: Okay. Well, both sides learn lessons, I'm sure. Question?

VOICE: When you're dealing with a terabyte of data, how long does it take to back up a system like that, or multiple terabytes of data? It strikes me as a significant problem.

DR DAILY: In our case, a substantial amount of what is on the system is data that's been transcribed in from external sources, and we transcribe in duplicate and pull the duplicate cassettes. As we start working more with intermediate data sets that get shed out of the supercomputers, that problem is going to become much more severe. I agree.

MR WOODROW: We use a primary and backup tape for all user files. User file system backups only save metadata (the node information) to tape and are quite fast. We also do regular backups of system file systems directories, but these are quite small and the backups are similar to most UNIX systems.

MS KELLY: We only backup the metadata, also. We only make one copy of the actual user data.

MR GARON: We don't back up the data. Most of our metadata is in Sybase data bases, and we just back that up as regular Sybase backups.

MR MARSALA: We just do a rotational kind of thing. It takes us about a week before we finish backing up our optical jukebox.

MS SALMON: Well, we back up the data bases that control where things are on tape, *et cetera*, but we've made it very clear to our users that we can only afford to keep one copy and can't make backups of the user data.

DR RANADE: How about the lessons from the other side of the fence, the vendors? I'm sure they learned lots of lessons in installing big systems and small systems. Would somebody from the vendor community like to say something? Dale, would you like to say something?

MR LANCASTER (Convex): (Off microphone.)

I was just saying that I don't know if it's a lesson learned, but it's just that you want to have the customer expectations well defined so that there's no mismatch in what you're trying to do. Also, try to bring these systems up slowly, rather than try to turn them on overnight. I think that's probably one lesson that I have seen out of many installations that we've done.

DR RANADE: Yes.

MR BENDER (Convex): I'm Ed Bender of Convex, and one of the things that I've seen is that data management customers are a hell of a lot more maintenance for us, a lot more work than typical computer servers. So that's one thing we've learned. We've had to put a lot of people into keeping things working.

DR RANADE: Can you tell us why that is, I mean elaborate a bit?

VOICE: (Off microphone.)

DR RANADE: A lot of different technologies are coming together in one system, and, therefore, you have these things.

MR LANCASTER: (Off microphone.)

I guess to summarize what your question is: why is there so much work, it's that we're really a system integrator now, rather than just a computer vendor, and that's really a big step.

DR RANADE: How about -- Dave, would you like to say something from the lower end of the market? I mean, you don't have as big a system as Convex does, for example, but --

MR THERRIEN (Epoch): (Off microphone.)

DR RANADE: Well, your lessons learned from installations with your customer base. I mean, yours is more or less a shrink-wrapped thing, isn't it? I mean, it goes in and --

MR THERRIEN (Epoch ): Right. EPOCH was -- I think when you go from being a turnkey supplier to being a software supplier and expecting the hardware to come from somewhere else,

the problems are magnified even more so. Because now you're dealing with product revisions that are sitting in some dusty distributor's site that don't really match your minimum requirements, and you've got to kind of manage all that.

Those are some new problems that we're facing as we're moving toward being a software-only supplier: hardware incompatibilities. So we have to maintain quite a bit of information on which revisions of which storage products and which platforms actually do work with our software on a revision-by-revision basis. It's a big problem, but it's not impossible.

I guess what it produces is a limit to how many products you can support. If we go back to some of the talks today, you just don't want to support everything out there. What you want are a collection of things that you know work from release to release. So you've got to limit what you support.

DR RANADE: And you guys do very thorough testing before you actually support it in your product.

MR THERRIEN: Sure. Sure. Right. You have to do that. If you don't, you spend all your days on the phone in customer support problems.

DR RANADE: Anybody from a systems integration company? Do you want to say something on this?

VOICE: The prototyping seems to be very important, especially when you're working with new hardware. Also, simulation seems to be a good tool. We use quite a bit of that, but the real key is when you're experimenting with new types of hardware, HiPPI switches, if that's the case, MaxStrat disk arrays, or even at the lower end, the newer disk drives, that prototyping is pretty critical to understanding how user requirements relate to system sizing.

DR RANADE: How about somebody in that segment of the audience? That's a pretty quiet segment over there. No? All right. Well, let's move on to the next one, performance, which is a big issue for many people. Whose turn is it? Joe, is it your turn to start? We are on question six.

MR MARSALA: Well, what I'd like to say about the Epoch 1 is the performance met our expectations.

DR RANADE: Okay.

MS SALMON: For us, performance is a continuing concern. I think, in general, we've gotten some strong performance out of all parts of our system. We're handling 50 gigabytes of new data a day and up to 70 or more in and out of the system. So clearly, it's not that any of these pieces is a fly-by- night kind of thing. But our users' performance requirements continue to grow, so the level of the fence that you have to jump over keeps getting raised, as well. It's something that we have to continually work in concert with the vendors to try to solve, and the users.

DR RANADE: Mike?

DR DAILY: I'd say for most operations we're within a factor of two of the nominal numbers for these things, which is pretty good for being only a year or so out of the gates. There's still plenty of room to improve, and I think in many cases we're still technology-limited. Things like the CM5 are sufficiently fast that we're going to have trouble feeding it no matter what we use.

DR RANADE: What about this problem of small files and the D2 tape drive? Is that something you've experienced?

DR DAILY: No, we tend to have different classes for the large data files that get stuffed into the Connection Machine and smaller files that sit off on other classes that serve as workstations. And we've been experimenting with some of the things that the Sequoia folks have thought up, like abstracting, and our own crude forms of clustering of data to kind of intuit what the user is going to do next to improve the perceived performance there.

MR WOODROW: One of the surprises in the HSM Evaluation was that although the same underlying storage media was used there was great variation in the disk and tape performance. Apparently simple things like keeping a slow tape device streaming were accomplished by only two of the four packages.

Another surprise was the extent to which the disk performance differed between UniTree and the other candidates.

There is a lot of variation between commercial HSMs in the types of performance optimizations built in to the package. There appears to be a lot of room for improvement for some of the packages and extensive benchmarking appears to be a very wise investment.

I spent a lot of time on performance in the evaluation report and you can see the specific differences for yourself in the proceedings.

DR RANADE: Sue?

MS KELLY: Well, I've already given my two cents' worth on performance. The UniTree system satisfies our performance needs. But I guess to give four cents' instead, when we had originally done the requirement study, we had selected the protocols of NFS and FTP. They were given. And we began an early campaign of recommending NFS for directory management and for small files and using FTP for any large file transfers.

So when we think of performance, we tend to focus in on the FTP performance. UniTree is a poor NFS server. Our NFS transfer rates with UniTree are about 250 kilobytes per second, whereas we can get up to 6 megabytes per second with FTP. Did I say that right? Six megabytes per second with FTP; 250 kilobytes with NFS for reads and writes.

MR WOODROW: That's from disk.

MS KELLY: From disk. Well, yes, that's where they come from. For our tape activity, we have approximately four new gigabytes that are written a day. I said we have five gigabytes a day of I/O; four is writes and one gigabyte is reads. With the four gigabytes per day, our tape system has no trouble keeping up. Our migration is idle a good part of the day. So it's somewhere between four gigabtyes and five that there's a problem.

MR GARON: The system that's using the Metrum AMASS software, they're very happy with what they have. They just bought it and plugged it in and it sort of worked just the way they expected it be. They're storing about eight gigabytes a day. I talked to them and interviewed them, and they just can't imagine anything much better than what they're getting.

And there are improvements coming with AMASS software. Those improvements, I'm hoping, will help me solve some of the problems that I'm going to try to use another Metrum system for coming this fall. I'm going to try to store 25 gigabytes a day and see what comes of it, see how well it does in that environment. Call me up in 6 months and I'll tell you.

DR RANADE: Well, since we have about 10 minutes left, let's skip number seven and go on to number eight. This is: what are your thoughts on cooperating servers, different mass storage systems being able to talk to each other, so to speak? This will lead into our next topic, which is the IEEE Mass Storage Reference Model.

MR GARON: The only problems I have with the AMASS software is that it does have a proprietary format on a tape and the disk, but I think that's all done for performance issues. What eventually I'd like to be able to do is be able to move that media into other software management systems.

DR RANADE: Right. What most of these do -- I mean, all of them do.

MR GARON: Right. That's the problem.

DR RANADE: They just get locked into their universe of data formats and then it's impossible right now to move data between one system and another. So in whose interest is it to have that happen and are we likely to see it? Does anybody want to comment?

For example, if there's a UniTree system or some system and there's an Epoch system or some other system, is it useful to expect them to talk to each other? Does anybody have a need like that? Yes? Do you have a need?

VOICE: I have a question about proprietary formats. By definition, a format is proprietary if it is used by one company to store its data.

DR RANADE: Okay.

VOICE: (Off microphone.)

Is it still proprietary if that plan is public, even though it's only used by one vendor? If you have access to the formats so that you could translate the data if you need to, then is it still proprietary?

DR RANADE: Well, I don't know what the definition of proprietary is, but I see what you're saying. If the format is public, then anybody who wants to can write in that format. But what I'm asking is: is there a need for this to happen? I mean, are there installations where they have two different types of storage systems and they have a need for one of them to talk to another one?

I would think that there would be such a need, but I don't know if any -- yes, Jim?

VOICE: (Off microphone.)

DR RANADE: Any comments from NASA/Langley?

MR BERRY: Not NASA-Langley, but I can give you a different comment. We went through an evaluation on doing backup and recovery for a bunch of file servers. In the paper by Mike Shields of the National Security Agency which appears in this volume, you could see there were a lot of systems back in there. One of the primary criteria for the selection was that the tapes be readable through the standard Unix utilities, which means we could take a tape that was made through the backup system, move it somewhere else, restore it, and put it back up. From the system administrators' standpoint, that was very significant for their selection criteria. There were relatively few systems that did that, but that was one of the reasons why Bud Tool was selected, for example, because it produces that type of format that you can then use through a standard utility.

DR RANADE: Right.

MR BERRY: So in that particular situation, that was a very important criteria, and it also let you exchange tapes between Bud Tool systems. So you -- or you can even -- well, the other thing we were concerned with was being able to read a tape if we didn't have Bud Tool installed on a given server so that we can move files around.

So there is a very specific situation where that's true. It's also -- in some of the situations, one of the reasons why we don't have some of the systems on our supercomputers was the ability to share those files and not wanting to be locked up inside somebody's format, so that multiples of those systems can read the same data.

And actually, as we go to a more scattered processing, that becomes even more important. We don't want to funnel it through one thing.

DR RANADE: Right.

MR BERRY: So in both of those cases where we've got production processing, we think that's an issue. And backup and recovery, I think it's an issue that's turned out to be fairly important.

DR RANADE: Backup and recovery is a big issue. So are these open systems under Unix-based HSMs -- but how many of them are really open systems? I think to my knowledge there's only one HSM that writes migrated data in a standard format.

MR SARANDREA: What format? (No reply) Which is?

DR RANADE: Which is NetStore. They write standard format optical disks when they send data off the magnetic disk.

Yes? Go ahead.

MR SARANDREA: With reference to NetStore, just to comment. You said they write open format optical disks, but what they're really writing is the UFS file, magnetic disk file system, of the system they're on, which is far from standard. UFS file system on media format changes from vendor to vendor, so that's not an open standard.

DR RANADE: Okay.

VOICE: Our FileServ product --

DR RANADE: Writes tapes.

DR DAILY: It writes tapes and it writes standard ANSI tapes.

DR RANADE: Okay.

DR DAILY: So any utility that can read an ANSI tape can read our formatted tapes. Also, there's work with POSC to standardize an interchange format for tapes, so that it's not just the format that FileServ might use on D2, but it would be a standard that anybody that wishes to adhere to could use.

DR RANADE: Okay. Moving on to number nine, we have 5 minutes left, 6 minutes left, I think. I've purposely left that one vague. It says: IEEE MSS RM - practical import. So I think when we discuss that question in the panel, what we mean is: if the IEEE Mass Storage Reference Model has been an ongoing activity for a long time, and who knows how much longer it will go on. So what is the practical relevance of it to buying a system today? I mean, if it were ready and done, would it affect the way you buy something today or would it not?

I'd like to hear from the panel and also from the audience on this, because almost every spec from the government that one sees, it says the system shall be IEEE MSS RM-compliant or something to that effect.

DR DAILY: Well, we're big fans of standards, and we're willing to pay a certain performance penalty for it. But I don't think it would be a make or break in anything we're doing. This area is still awfully immature and there are other bigger fish to fry right now. But longer term, yes.

DR RANADE: Brian, did you have something?

MR SARANDREA: Yes, Sanjay. Can you define mass storage reference model-compliant?

DR RANADE: No, I can't. That's why the question is there. Why do you think it's on the list of things to talk about?

MR WOODROW: Yes, that's why I think the problem everybody puts in their spec, but how do you determine whether when the vendor says yes, this is compliant, what is it? Certainly, this is what we look for, one of the things that we look for, but it's not one of the things that we've been terribly rigid about enforcing.

DR RANADE: Well, yes. I think the goal of it is great and we want that, but how can the user community move towards it? I mean, is there a way for the users to accelerate that? I don't know. Sue?

MS KELLY: We used the IEEE MSS Reference Model during our requirements study. We had first done our requirements in more traditional areas of functionality, performance, and capacity. We then turned it around and looked at the system, looking for requirements based on the components of the reference model. We were not able to identify any new requirements by looking at it from the reference model viewpoint.

MR GARON: We would certainly ask the question, but I don't think it would have any impact on what we bought or didn't buy.

DR RANADE: It would or wouldn't? What did you say?

MR GARON: It would not impact what we bought.

DR RANADE: Okay.

MR GARON: I think it would satisfy the requirements, and it wasn't -- it satisfied what Mike Shields was talking about: solid company, they're going to be in business for a while and we can work with these people. Then we will continue to -- that would be a big plus, not necessarily the IEEE model.

DR RANADE: Joe?

MR MARSALA: I don't think I can add anything to what has already been said. I mean, it's just not defined enough yet.

DR RANADE: Ellen?

MS SALMON: Well, I can pretty much only speak for myself and not for the folks that went through the procurement. I think that the Reference Model is an important basis, but perhaps for us it was more important that the product we ended up with could run on multiple platforms from multiple vendors. So the product being "open" was probably more important than the Mass Storage Reference Model itself.

DR RANADE: Anyone in the audience?

MR BERRY: Yes, I can give one comment. Probably the most practical import that we've seen from our basis is early on and almost continuously they've emphasized the separation of control and data. And for at least the high-performance applications, I think we've validated that that is a concept that must be present if you're going to get performance. It's absolutely critical. You can't move this data across the networks with the control. You literally need to set up things. So when -- in Mike's charts you saw HiPPI switches, eventually fiber channel

kinds of things in which the data is going to move in a path that's not out contending with network traffic; it's running TCP.

So in that sense, I would say that's -- from our standpoint, we've seen that that's really a critical factor and is how you get high performance.

DR RANADE: Now that's a very specific application.

MR BERRY: It's a very specific thing in terms of model, but in terms of the whole model, no. There's lots of things in there that don't seem to be -- you know -- who knows?

DR RANADE: Yes, sir?

VOICE: How do you verify compliance?

DR RANADE: With something that doesn't exist?

VOICE: How do you verify compliance with things like compilers, POSIX, for instance? It seems to me what you're going to need to do is you're going to have to come up with a series of tests by some group affiliated with the people that come up with the standards or the models, and the products are simply going to have to be -- you're going to have to be able to run these tests to guarantee that all of those requirements are met when in operation.

DR RANADE: Right. It's a big job, isn't it, to say if something is compliant or not and actually prove it or certify something like that.

Dale?

VOICE: I think Mike --

DR ISAAC (MITRE): Just having some experience with the reference model, I felt obligated to stand up and say something about it. There's three or four comments I'll make. I'm not sure they're all connected.

Of practical importance, I'm not sure any reference model has any practical importance, and perhaps it shouldn't. Maybe the only practical importance a reference model would have is that one of the goals of the reference model establish a common vocabulary; this way, we can sit around here and talk about migration, and everybody knows that we mean something different than caching.

So just having a common vocabulary can be practical importance, but that's about as close as a reference model can get. Its goal is, especially if you read the fine print in the front, that this is not a document that one can establish compliance with.

The goal of the reference model, the second goal besides establishing a common vocabulary, is to establish a framework for the standards that are to follow, and that's where you should look for the compliance, the rigor, the benchmarking, and compliance testing. There are three or four dots that have been spun off the IEEE P1244 project. PVR will be the first one out of the gate. You can look to have active work on that towards a standard that will get you a physical volume repository, and the major vendors are actively involved in that.

It is yet to be seen whether or not such a standard is successful. It's quite a challenge to develop a standard rather than accepting the product.

DR RANADE: See, that's my point. Go ahead. Sorry.

DR ISAAC: That's about all I said. As for the other ones, storage systems management, the identifier, storage object identifier, and storage server, there is a dot spun up that has been

launched to establish standards in those areas. So maybe down the road in another few years, we can start looking at standardization that will actually get you interoperability and some of the other things that we'd like to see.

DR RANADE: Thanks.

Dale, do you want to say something?

MR LANCASTER I think -- I was going to make one of the points that David pointed out, that the reference model is really not the standard. It is, you might say, a fleshing out of the thinking behind the need for a standard. The standard is really called P1244 dot whatever and is currently being developed. How you do compliance is one of the goals of the National Storage System Foundation, which is having somebody that says yes, you really are compliant to the P1244 dot whatever standard.

I think mainly it benefits the vendors, rather than the users. I think the users have a secondary benefit, but the vendors, you know, we're pulling our hair out trying to have five different PVMs and PVLs and PVRs and all this other stuff that we have to integrate day to day with each of these systems. So it benefits us more than the users. The users just want a system, and I think I heard that a little bit earlier today, maybe from Mike, that you just want to store lots of data quickly and easily access that, whatever that means. And you're not going to hear, I don't think, a customer say "I think I need to buy another PVL today." That just won't happen, even though the PVL will be P1244 dot something complaint. So I don't think I -- I think there's no practical import to the user, but there's a lot of practical import to the vendor, which in the end will probably save money to the users buying the stuff.

DR RANADE: Sam, would you like to add something?

DR SAM COLEMAN: (Off microphone.) ...

In the software area, UniTree is an implementation of one of the earlier versions of the reference model, and it points out some of the strengths and weaknesses of the model. But the success of that product is demonstrating the importance of the reference model.

The National Storage Lab was a direct result, an outgrowth, of the IEEE effort, and that's a collaboration with 27 companies at this point working on new architectures that were suggested by the reference model.

There's a new project in the National Storage Laboratory which is specifically chartered to be an implementation of Version 5 of the Reference Model, and that system is going to provide performance of a couple of orders of magnitude greater than what can be achieved today. That project will become one of the projects of the National Storage System Foundation that John Simonds described yesterday, and the software division of the National Storage Industry Consortium is a direct result of the work in the IEEE.

I think the most important value is that the vendors have deemed this to be sufficiently important that several dozen companies are willing to send people to meetings every two months, and we have forty to fifty people that come together to talk about the best ways to design a storage system. The IEEE provides the forum and the reference model is the basis for those discussions. And that's very important, because we brought together a lot of traditional competitors in this area. We have all of the major software developers that are working on these systems. We have IBM and DEC and HP talking about how to build storage systems. We have Ampex and Storage Tek talking to each other. We even have Convex and Cray in one room having friendly discussions on how to build storage systems.

I think that the real importance is that this storage problem has gotten to be so big that no one vendor, not even Convex and not even Cray, is going to be able to solve this problem when we have large networks of heterogeneous, massively parallel systems, and we're talking about

terabytes a day and many petabytes of storage. This is an enormous problem, and the only way we can solve it is by collaborating and cooperating. And we see good cooperation among the vendors, and I think with that kind of effort being applied to the problem, that we will be able to solve it. So I think that's the main importance of the model.

DR RANADE: Any more on the model? Okay. Let's go on to the last one, metadata.

DR HOWELL (ICI Imagedata): Sanjay?

DR RANADE: Somebody on the model? Okay.

DR HOWELL: This makes me a little horrified, hearing that the standard is actually just a vocabulary. I would agree with the previous speakers that standards, in my book, are an agreed solution to a common problem. If it's a vocabulary, let's not have it masquerading as a standard.

DR ISAAC: Should I respond to that?

DR RANADE: Yes, absolutely.

DR ISAAC: (Off microphone.)

So you'll see IEEE documents that say guidelines four, blah, blah, blah, and standard four, and this is a Reference Model four. So it's not -- there will be a standards to come, and that's what you'll get, lots more than vocabulary. But the reference model has -- besides the Reference Model activity, I think Sam pointed out well that half of the importance of the Reference Model is the Reference Model activity in the working group. Establishing common vocabularies and establishing the major components as a framework for the follow-on standards is the most important activity of the Reference Model itself.

DR RANADE: Any final thoughts on the model before we leave it for another year? All right.

On metadata, anybody on the panel want to start? We talked about it yesterday. But let me just explain what we mean by that. Metadata, we mean data about data. You have lots of files, lot of information, but how do you access it? Must you use the file name every time? Or is there a way to intelligently index what you've got stored? I think we have somebody who has actually done a pilot system. Do you have a DBMS that --

MR GARON: The only data that we store in the one main system we work with is all -- there's a Sybase data base and it points to every entity of data. The analysts never pull by file name. Well, they don't know what the file names are. They query the data base, and they query in certain columns and get their information; that gives them their file name. We have built a level of software above that does the queries for them if they know what they're looking for. It goes out and retrieves the data for them.

DR RANADE: So they ask for certain types of data and the files come to them.

MR GARON: It could be.

DR RANADE: Right.

MR GARON: By various reasons, dates, whatever. I can't tell you the rest of it.

DR RANADE: It could be content-dependent, also, like what type of data is it; you could say for a simple example--cloud cover. You know, if you want data with X percent cloud cover, you could pull those, for example.

I would think that, Ellen, in your system, where you have 60 gigs going in and out for a day, something like that would be useful, right?

MS SALMON: Well, I think one of the problems with implementing that system-wide for our facility is the wide diversity of users and the reasons that they use the data. I think the division is looking towards at least providing the tools for users to organize their data in that way, and at some point it may be the logical step for us to step up to the management of that. But that's almost going to have to be something that the user labs explicitly come to us for and say we need this, and by the way, here are the funds from headquarters to go purchase the software and things.

DR RANADE: Well, there are actually two efforts that I'm aware of that are going on to define metadata standards. One is the one in Austin, and Bernie -- is Bernie O'Lear here? He left? He just left, okay.

MR LANCASTER: (Off microphone.)

DR RANADE: Could you tell us about it, about both of them?

MR LANCASTER: There are two efforts that are actually combining. I just found out this afternoon, because I talked to Paul Singley from Oakridge, who was on that committee with Bernie O'Lear. Basically, the IEEE, the same group that Sam and I and all are involved in, especially the one that was responsible for doing the Mass Storage Reference Model, started a series of workshops to deal with intelligent access to large amounts of data. Now, I'm not sure exactly what the titles were, but that's what I call it. Or what we call simply the metadata problem, which is: you've got ten million files-- how do you find what you're looking for?

Even people at NASA retire eventually, but their data doesn't. And you wonder: well, do I need to delete this file or keep it? And you don't know, because you didn't generate it originally.

But anyway, we had a workshop in Austin that Jim Almond and I set up down at his center, and we had several people come who were highly motivated to try to get a handle on this. We have some minutes from that workshop that have been generated, and a white paper is being written by a couple of people. Robyn Sumpter and I think even Sanjay is working on that as well -- to try to define what the problem is and where we might want to go with that.

Parallel with that, there was supposed to be a workshop at Oakridge sometime in '94 to deal with something that they thought would be the data base-type problem. Well, they had their first meeting to set up the workshop, and they realized that they were really more interested in metadata; that's what they really want to talk about.

So Paul Singley and I got together just a while ago -- and I don't know if he's in here or not. There he is -- to say: well, gosh, we're skinning the same cat; let's go skin it together rather than try to reinvent the wheel.

Then I saw some papers on the Information Interchange Reference Model, again maybe defining some vocabulary; but the idea that -- it's a big problem. In fact, I think that's public enemy number one, because I think that anybody can store lots of data, but not anybody can effectively use it. And I think this is a step to get there.

So that's my 25 cents' worth, Sanjay.

DR RANADE: Thanks.

VOICE: (Off microphone.)

DR RANADE: Well, if there's no more, I'd like to thank each of the panelists for being here with us and sharing your experience, and the audience for being here and listening to us and participating. Thank you.

EVENING RECEPTION AND DINNER

# Moving Images Archive

**David Parker**
Acting Head
Curatorial Section
Division of Motion Pictures, Broadcasting and Recorded Sound
Library of Congress
Washington, DC 20540

MR. PARKER: -- for lower check, for the way things were. I'll try not to make this autobiographical and dull; I'll try to make it official and dull instead. But I got something in the mail Saturday. It was one that didn't say "occupant" or "resident." It said something to the effect that if you get to the Library of Congress by 3:00 a.m. on Wednesday morning and stand in line or bring a cot, as you would for a Rolling Stones concert tickets, you were eligible to retire. It puts me in a retrospective mood tonight.

Well, came Wednesday and a lot of people were in line, including our assistant chief. He's been there for 30-some years. So it was retirement, retirement all day long. People who hardly knew each other, who were barely colleagues at the Library, would pass in the halls, and one of them said, "I don't want to hear another word about retirement."

At the end of the day, one of the last researchers came in, somebody I knew, and he didn't know anything about this. He hadn't been reading the paper. So he saw the assistant chief's secretary putting on her coat, and he said, "Oh, are you leaving early?", as one would ask, "how's the weather?" She said, "I'm not retiring." And neither am I.

But it does take me back to 1969 when I came to the Library of Congress. I was a film maker, and somehow they convinced me it was more important to save the original negative of ***Citizen Kane*** than whatever I might turn out next year.

I was also there in the early '70s when they changed the name of our division. It's a little bit of immortality for me, because the word "broadcasting" in the name, "The Division of Motion Pictures, Broadcasting and Recorded Sound was my suggestion. And about 15 minutes after it was officially adopted, it became obsolete for reasons that may have to do with what you were talking about during this conference.

I hold here a printout. This is my security blanket. I'm a bureaucrat, I bring this. This is ultimate truth. This is a count of everything we had as of last October. If you want a count of everything we hold in our division as of *last* October, that's why I may look a little more frayed than usual today. We're still working on it. We're going to have it ready tomorrow. It's four days overdue right now.

I guess an important milestone would be in 1964, when we got a film scholar as head of the film division, not a retired military person, which had been the tradition up till then. I mean, pledge of allegiance to the flag first thing every morning.

The film scholar decided it would be good to retain more films than fewer. So the idea of selecting only the very best of the best, chosen by whatever the standards of that year, reverted to what Archibald MacLeish, a Librarian of Congress in 1943, had envisioned at the establishment of the film area: instead of sending films in for copyright , having a clerk note some information from the film and sending them back to be dispersed and perhaps never to be collected again, the Library of Congress, as the national library, should select every year for the permanent collection films that tell us about living in that year.

And Archibald MacLeish didn't just want the best of the best of that year; in addition to films of great news events, he also wanted newsreels about whatever would be the 1943 equivalent of the hula hoop, and he spells that out, the range of production, I guess, as if it were to go into a time capsule. And curiously enough, the University of South Carolina, which now has the collection of films of the Movietone News, most of the requests are not for the hard-core news features; they're for the other parts of the newsreel: the dog who ice skates, the guy with the wooden garden, and the hula hoop. Because a newsreel of 1943 was made up of all sorts of things, and that's the mix he wanted.

We were able to select a lot of films because of the U.S. copyright law, one of the best in the world, if we wanted a film for the permanent collection, it must be surrendered. One copy instead of two. For books, two copies are required, but the Motion Picture Association and The Library of Congress made a deal, not the last one.

In the late '70s we had a shotgun marriage, and all media was put in one area. It's sort of the concept that I understand was used by the University of Maryland library. You could dial the media number -- probably still can -- and they answer the phone, "non-book." So I guess I'm in the Library's "uncola" division.

Well, that's the way it is. That's the library. the books and the media, these Johnny-come-latelies. Perhaps the reason film has become thought of as an art is that there is now television to trash, you know, because it's newer.

So we're now the Division of Motion Pictures, Broadcasting and Recorded Sound. (Presumably that's sound isn't wandering around, bouncing off the walls but is actually engraved on a support base.) In the division, they came up with the Curatorial Section. They already had the standard library administration, acquisition, processing and cataloging and added something called "curatorial". (That's not "custodial", but some days I can't tell the difference.)

I'm up to '92. The official count: In our curatorial division alone -- omitting the books and the electronic media, (machine-readable documents and CD ROM) -- only counting moving image and audio -- we hold 3,328,589 items, which take up linear shelf feet of 263,875 feet and 7/10s of a foot.

I can't give you the cubic feet they fill; because we have many formats, from miniature home movie formats to 70 mm copies of films such as "Lawrence of Arabia," each reel of which is counted as one item -- and don't drop a reel of 70 mm on your foot! In fact, if you've been with the projectionists' union so long that you have the seniority to be projectionist at the house where they show 70 mm, you might get a hernia. They ought to assign 70 mm work to projectionists in reverse order of seniority.

So we have several hundred pictures that are in 70 mm format, including reels that came from Elizabeth Taylor Warren's residence of a motion picture called ***Around the World in 80 Days***. A film studio has accessed that material to put together a new 70 mm version of that film, in the same restoration procedure done with ***Lawrence of Arabia*** and ***Spartacus***.

That kind of holdings help make us an archives, not just a library. It's not getting just having video copies for home viewing; it's also having the original camera negative of ***Casablanca***.

The Library and copyright started about 100 years ago copyrighting films. We have now some film copies manufactured made 100 years ago that are still in good shape. The others are not and I want to get to that right away, because that's the part that worries us in the janitorial part of the Library here.

We also collect 45rpm records, although there's a guy who says he has more 45 records than the Library of Congress. His trick is that he bought up all stock from the regional exchanges as they went out of business, because the computer let the record companies ship nationally from a single location -- Terre Haute, I think -- so he may have many copies of the same 45. But it's true, he has more copies of 45s than the Library of Congress. We're talking to him.

Now, when a format becomes obsolete, we don't throw it away. We give it to the Library of Congress. For instance, recorded sound on cylinders. We've got 10,500 of them in last year from one collector. So when people clear out their attics and basements and they find something very valuable, the Library of Congress has to have something to play it back on, into whatever new wonderful equipment is now the technology for the next decade or so.

Maybe the name I should have come up with in 1970 was "the Division of Motion Pictures, Broadcasting, Recorded Sound and Laser-Etched Saran Wrap, and whatever they invent next"..."non-book". the book side seems a bit more stable.

I've seen some things along the way that even with my poor eyesight I knew weren't going to pass muster. Somebody was explaining to me -- I think it was a film manufacturer-- the advantages of something new called super eight (How many remember super eight?)

He was explaining the advantages of super eight over standard eight. Does anybody remember standard eight?) He said you couldn't recognize your own grandmother on standard eight, but with super eight, you could.

So somewhere between ***Lawrence of Arabia*** or ***Far and Away*** or some showing in IMAX format and the poorest half-inch videotapes we've ever been offered, we have to decide what is appropriate or acceptable quality of preservation for the moving image. Does the film still survive when you can barely make it out as if it's transmitted by wirephoto? Or do we require a 70 mm original copy?

You can look at a movie called ***Love Story*** and hardly make out the figures of the actors, and it can still make you cry. But if you're looking at an Anthony Mann western, the landscape is very important to what the filmmaker is trying to communicate. Some film makers use such strong geometric forms in their pictorial compositions that you could send it by thermofax and the idea would get across. But the more detailed the physical surface, the more the sensuous parts of the medium are used to tell the story, in contrast to diagrammatic plots and cliche'd dialogue, the more important it is to retain the resolution, the technical quality, of the original, at least in one format so it could then be translated into the other forms in which it's going to be distributed and viewed.

So ideally the problem is getting a print from the original negative of ***Casablanca*** over the fiber optics network to Los Ceritos, California, where it can be picked up in the viewer's own home, and still look and sound like ***Casablanca***. There are perhaps one hundred shades from the whitest to the blackest black in ***Citizen Kane***. To reduce that to 20 shades of gray gives you the equivalent of a smudged carbon copy or something even worse. Let me take an example from music: listening to Mahler's ***Symphony of a Thousand*** over a 50 cent, two-inch loud speaker (like those used in cars at the drive-in movie) may work fine if you've already heard Mahler's ***Symphony of a Thousand*** in a concert hall or on a fine CD. You can bring your earlier experience to what's actually there from the 50 cent speaker from the drive-in. But if the drive-in quality of sound is your *first* experience with the work, your filling-in of what's actually not there may be relatively unsuccessful.

The Library of Congress has to worry about such considerations when we talk about compression and sampling rates, when we talk about translating it into any other formats. But mostly we worry about the condition of the physical material on which the content is recorded. Digitally we can now recopy every five years and theoretically lose virtually nothing. But if we've got 700,000 safety films, all of which may be attacked in the present or future by the vinegar syndrome, that's 770,000 cans that have to be opened by somebody has to

do something physical to each can, even if it's just to stick the rubber nipple in the first time so that a probe can be used with the nipple every subsequent time to record information about gases in the can and not have to open the can itself again.

We have 110,000 cans of nitrate film. When nitrate film is ignited by a spark or an open flame -- it doesn't explode; it just burns so fast, even under water, that you can't tell the difference, -- With nitrate film, we try to open each can for inspection once every six months. But the irony now is with the vinegar syndrome problem, we have movies made on safety film in the '50s and '60s, the original negatives of which are showing -- not on a large scale yet, but on a small scale -- throughout that entire 20-year period, deterioration characteristics quite similar to those of nitrate made from the late 1890's to 1952.

We've found there are not that many differences between the new safety films of 1915 or '52 and the nitrate, if we're talking about long-term keeping and storing and their total lifetime. Let's move closer to the present time.

Remember you couldn't recognize your grandmother in the straight eight? Now let's go to something I saw a couple years ago that made me very excited and made we want to be part of this group here to learn what I could learn. All this time we've been hearing something just as good as 35 mm and then we've been seeing, and it does not meet large auditorium, large screen showings. It may work in some other kind of presentation environments.

I've seen Kodak's new system, where you convert a 35 mm  mm image -- not 70 mm, not IMAX, but 35 mm to a digital record, manipulate it in that form and etch it back onto back 35 mm film. , at least on a reasonably-sized screen, some pretty remarkable digitization. First the Kodak tests and then Cinesite, the company that restored ***Snow White and the Seven Dwarfs***.

Now we're back to an area in which I'm some kind of an expert, having memorized "Snow White" over many viewings. I've been seeing that film -- I won't tell you how many times and for how long. Every time it was reissued, I saw it, and I have some clips of a print at home which I could compare what the digital form was like with the original. If I were an art historian, I could quibble about this shade and that hue and that intensity and say that the blacks are too gray and the saturated reds are not there. But it's amazing what *is* there.

What is there is a pristine copy. If you saw it in its last reissue in 1987, produced with conventional printing techniques, in the scene where the prince first meets Snow White and she's singing to the doves -- well, in that 1987 print you could see the doves fly off to the left and the field of dust go over to the right, and both were about equally prominent visually. In the current reissue the dust has been now removed digitally, except for two specks they left on the surface of the magic mirror because, you know they made it look more like a mirror. Without the dust, it looked too transparent. That's referred to as the inability of a dog to pass a fire hydrant without stopping.

That's not fair to the people who have done a wonderful job, and they showed the "before and after" of the first reel to us at the Library of Congress,. They had an idea in mind that tied right in with something we'd been talking about ever since we knew in 1969 what NASA could do visually that was not possible for us. When large pieces of the original film emulsion with the original information fall off of that 1895 picture, leaving only the clear base. And if what's been lost is redundant information, if it's present in adjacent frames, and if you could capture it from those frames and put it back in the frame suffering the diverticulation, it could look as if it had been shot yesterday.

When Frank Capra, the director visited the Library of Congress in his later years I was privileged to set up a screening for him of one of his movies made in the mid-thirties; we had struck a print from the original negative. It was a test print, and I thought it was terrible due to shrinkage of the native -- the sharpness was not good. But Mr. Capra said he was impressed with what he saw because there were no visible scratches, and without the scratches it seemed

that the action was happening not in 1933, but right now as we were looking at it. The illusion of the movies was sustained. That could hold good for a sound recording, too, where the processing allows the original to come through with its own kind of sound.

We had wet gate to make the grain less visible. You couldn't see the grain. You could blow up 16 mm, Disney's ***True Life Adventures***, ***The African Lion*** or ***The Endless Summer***, **On Any Sunday**, and you didn't see an oppressive grain structure; that was removed. You didn't have a very sharp image there either, but you had the cues of color and shadows in your own mind to help separate planes of action and foreground and middle ground from background. As for what's not there, but it doesn't seem to matter because the psychological effort of the person who are reading the image or listening to the sound image compensates for it.

Maybe we may decide to do exactly what they did with Disney's ***African Lion*** shot in 16 mm. We can't just save everything in 70 mm, although the technology to do it is there. Here is one thing which the Library of Congress is somewhat slowed up a bit, and it's the same factor as in 1969, when we were talking about diverticulation and the patches and what NASA could do to restore visual information at that time:

With ***Snow White*** it goes something like this. I may not be quoting this directly, but this is what I remember hearing them say: For each frame manipulated, it takes thirty seconds and costs $8 to etch that amount of information into the digital format, and then when you're finished manipulating it, getting rid of the holes and patches and creases and all or maybe touching up the color a bit -- for instance, it's monochrome down to the bottom of the ocean, so you add red coloring to the coral so the audience doesn't see such a boring all-blue image. (That's being done for a new Tom Cruise picture. Look for the coral; it's digitally enhanced -- and then you get it back onto motion picture film so it can be projected in 35 in a regular-size theater, that's $6 more. Twenty-four frames a second, 90 feet a minute -- well, you get the idea. And that isn't paying for the 100 people who worked three shifts around the clock to get "Snow White" ready.

So the difference between the potential and possibilities and what resources the Library might have available for that seems to be a great chasm to bridge.

There is another demonstration I saw that cheered me up as much as seeing the digital ***Snow White***. This was a development by a professor and his graduate student, working with limited resources, using off-the-shelf materials at a university brought to the Library of Congress. It was a particular jolt for me because the man who had just been given the assignment at the Library of Congress to look into what might be technologically possible for such an application, was watching the demonstration and could see that this system was already up and running, and we were starting far behind.

Positioned on the West Coast you could view the cracks and gouges on the surface of a disc recording that we hold in the Library of Congress in Washington. It's a 78 rpm record. (How many people know about 78 rpm records?) Maybe they're in your attic, if you're not tidy and haven't done spring cleaning.

We have become reconciled at the Library that our 78 rpm records are going to get fully cataloged just about the time all those 45s also get cataloged by conventional means. It isn't going to happen soon. So let's talk about applying the low tech of 1975. We photographed each label, front and back, on each disk onto a frame of 16 mm film. It may not have the best resolution, but if they can use 4 mm for photographing recording instrument panels for test planes, we can use 16 mm film for photographing record labels.

Now, we haven't cleaned up the mistakes on the record, the typos. And beyond mere typos, you may not believe everything stated on the label: we have a Decca record that says, "Bill Haley and the Comets, `Rock around the Clock'. Foxtrot."

But catalogers can worry about what it is if it isn't a foxtrot later. What you can do now is punch a four digit number and retrieve by composer, by artist or by title every 78 rpm that we have in the Library of Congress and in four other U.S. sound archives, up until the time when the project was over, when they quit photographing labels onto those 16 mm frames. Accessing a huge data base is possible, thanks to a meat packer who'd made a lot of money, who liked opera, who wanted to find out what there might be in the way of opera on 78 rpm records. And he was convinced that everything on 78 rpm ought to be treated the way he wanted opera treated.

Now they're working on getting that data onto a CD-ROM so it can go out with all the other things the Library makes available on CD-ROM. We have videodisks of San Francisco, of New York, photographed at the turn of the century. These are the paper prints, contact sheets made for purposes of copyright. Until 1912, the only way moving pictures could be copyrighted was as still images. And between 1912 and 1943, when the Librarian of Congress said, "We ought to be keeping some of these films here in the national collection," that's the period we were trying to fill in by getting the original negatives from the major studies and making a master film copies on 35 mm to match the originals as closely as we could with the silver content of emulsions today, to retain the ability to recreate a large screen theatrical experience.

Yes, you can get a copy of ***Casablanca*** on a half-inch video copy, but it's not quite the same thing. The size, the dimension, a lot more is lost than one would know unless one saw it in reverse order, on the big screen first as I've been privileged to do, as we did for all of these films.

As you may have suspected by now, I am lost somewhere in the past, selecting films made before 1952 to be copied, because they're on a nitrate base and going to crumble into dust early. And now we're also concerned about the pictures made in the '50s and '60s because of the recently discovered threat of disintegration.

The one thing that it seems to me that all this boils down to that I've seen since '69 is the technology changes every decade or less. The Library of Congress has to keep all the information we might want to access that's recorded on the cylinders -- Brahms playing-- down to the present day. And the physical materials that these recordings are on is so fragile. If the consumer audiotape is projected to last 20 years plus however lucky we get -- and, of course, we don't control the materials chosen. We don't have the materials of our choice to work with. Often we have just what the collectors give us, .

In a play by Brendan Behan titled the ***Choir Fellow***, which takes place in a bar, The woman who runs the bar sees a man dandling a girl on his lap, and says, "Put that girl down! You don't know where she's been."

We don't know where the collections have been before they come to us, so it's harder to figure out what their additional life span may be now. We know about a man who owned the organ company and wanted to have something to look at while he played his magnificent theater organ. He got a wonderful collection of the great silent films. He lived in a castle by the sea. In it he had a vault near the seacoast in which he kept the films. By the time we learned about it, there was only one of his films that could be salvaged. It was ***Salome***. We hung it up around a room in the nitrate film building, and dried it out like wet wash. When it was dried out we were able to print it.

So we worry about compression, we worry about sampling rate. But mainly we worry about the tendency of all things laminated to delaminate, whether we have 20 years or 30 years and whether the accelerated aging tests of materials done at Kodak and elsewhere are accurately predictive. We do know tests for the longevity of films, done in the '50s at one of our sister archives, didn't prove to be accurate. So there must be other factors, such as "where it's been", that couldn't be taken into account.

We have somebody who believes in cryogenics, digs the film a hole, buries it in the hopes that the technology to bring it back will come from this group or others one of these days. That's a faith in science maybe, but beyond my powers of willing suspension of disbelief.

All of this audible and moving image material is the memory of the world or at least, the memory of the Continental United States and its territorial possessions, et cetera, et cetera, as of certain times. Of this memory of the world, we never know for certain what is going to be wanted next. Although we keep a great deal of it, we have to make "triage" decisions every now and then.

The fragility of the material, the lack of backup copies, that's the sort of thing that bothers me. But the excitement is what is possible even if the Library of Congress doesn't have the resources yet to play in that particular high-tech, high-expense ball game in that club, in that league.

The disk that you can not only hear played for you but can also look at its notches and cracks, as well as the label stating that it's a foxtrot called "Rock Around the Clock", from across the country, that's a little more exciting than just the offerings on pay TV, as easy as selecting something from your local video store. It's an example of what the Librarian of Congress may be talking about when he speaks of "getting the champagne out of the bottle", so the super digital highway is a wonderful dream of possibilities and we're all following that dream.

But the time and cost of getting from here to there is a problem, and I suppose I'm an arch conservative. I'll end with repeating what I heard at the East German film archive: (remember East Germany?) And if I suspect that it was chosen because they didn't have the high tech resources available to them, it still may be the right choice.

But what the head of the archive there said is something like this: "we've built good vaults with proper temperature and humidity controls to keep the film and tape alive for 100 years. And when you with the high technology figure out what are the optimum means of re-recording this material might be, the material will be here. We'll know where it is, we can find it and we'll make it available to you. It will have been saved."

So those are the two paths, to what I like to call "archivery". It's "thievery" and "sorcery". A bit of everything in it. I think it sounds better than "janitorial".
(*Applause*)

If there's someone I've not confused totally by what I'm saying or where I seem to be going, raise your hand. I'm open to questions.

VOICE: (Off microphone.)

MR. PARKER: What is the relationship between the Library of Congress and the National Archives? Do you mean from the firing on Fort Sumter or after that? I get this asked all the time, when I don't get asked about Kemp Niver, the guy who got the Academy Award for the paper prints being transferred to 16 mm film.

There are gray areas, which I'll not go into, but roughly it is that what the government produced, the documentation the government produces, like your Army record from 1915 or films about activities of the government -- that's how they sneak in newsreels with hula hoops into the collection -- material generated by the government goes there.

The private sector, largely, I guess, because of the books we buy and the fact that the copyright office gets materials to us, the private sector is represented in the Library of Congress. We're always getting mistaken. You know, it's like the actresses, Gale Sondergard and Judith Anderson: which one played the sinister housekeeper in ***The Cat and the Canary***

and which one played the sinister housekeeper in ***Rebecca***? After a while, the one that wasn't in ***Rebecca*** just wearily thanks the fan for the compliment and doesn't try to correct anybody.

VOICE: (Off microphone.)

MR. PARKER: Surrender the copyright to the government? Did I say that?

VOICE: No.

MR. PARKER: Sounds good. Surrender copies, two copies. Should there be a legal case, then this would be evidence. We've even sent out a videotape of a movie that was evidence in a copyright infringement case, and we put a ribbon around it and stated ion a note, "We verify this was a true copy of the movie."

VOICE: (Off microphone.)

MR. PARKER: Yes, we have two copies of one film, ***Johnny Guitar***, because one copy came from its star, Joan Crawford, and they didn't say no to her. And there is a problem of how much backup is desirable. If you have one copy and it gets torn up during the next screening, where are you?

VOICE: (Off microphone.)

MR. PARKER: Everything until about 1955 might have been shot in the three-color process. (***Foxfire*** was the last movie shot in the three-color process, that's ***Foxfire*** with Jane Russell, not ***Firefox*** with Clint Eastwood.)

VOICE: (Off microphone.)

MR PARKER: Yes and no. We are storing them for the archive they belong to, along with three vaults of other materials. Well, let me just explain about these three million items by way of Technicolor and Warner Bros.:. If you want to see ***Robin Hood***, it runs about two hours. If you want a print from the original negatives, that's forty reels. For every reel of picture you look at for ten minutes, you've got a cyan record, a magenta record, a yellow record, and the soundtrack.

So if you lose one of those -- and it happened to a reel of ***The High and The Mighty***, I'm told -- then you've got to try to reconstitute what should be there from the surviving elements, and that happens too, as was done with the restoration of ***Becky Sharp***.

Yes, we have -- we're storing MGM color pictures made during the nitrate era to my knowledge, but they're not ours. We're storing them temporarily for another archives.

VOICE: (Off microphone.)

MR. PARKER: You hold onto it as long as possible, because still and yet again, (***Snow White*** on digital notwithstanding), it is the best source material to copy from. Of course, if it *does* start ticking, it is put under water, because if it crumbles into pieces, it is much like gunpowder. If it becomes a safety hazard, it goes under water. But you try to save as much of the picture as possible. You *don't* say: "oh, this reel smells bad. I'll throw it away." You carefully cut out the deteriorating parts. It's a bit like running a cancer ward.

VOICE: (Off microphone.)

MR. PARKER: Well, we hope, you know, it will go by fiber optics to Los Ceritos, California, but we're a little way -- but with $8 a frame going out of film and $6 a frame going back to film, we're not quite there yet.

The other thing, of course, are the copyright owners. Oftentimes we have to send access seekers to the donor of the material, if it is on deposit at our place, to their lawyers to find out what rights are involved. Paranoia in the industry is classic and has not been mollified by the discovery people who have been active selling
video copies that are unauthorized. The copyright office is located one floor above us, so we're very circumspect about that sort of thing.

But we always have the success story of the guy who did everything we told him to do, instead of trying to find a way to beat the system to get access. The rights owners said yes, the publisher said yes, and he got what he wanted. It takes a little longer maybe than you wish and a little patience, but it works.-- I think the last line in my job description says: "get the stuff out so people can see it and hear it. So we've found new ways of doing that. We're having some touring shows next year for the centennial of the motion picture. Nearly every state will have a showing, over two years. The details are not worked out yet.

We're making the first batch of films available to the public. Early films by early film directors, women -- some important black cast films that are otherwise not being distributed. And those will be out in February for rental on 16 mm, 35 mm, and for sale by mail on half-inch videotape.

VOICE: (Off microphone.)

MR. PARKER: What I heard, they don't store it on digital video for "Snow White." It takes up too much space, it's impractical. If you're talking about full 35 mm resolution.

VOICE: (Off microphone.)

MR. PARKER: Well, yes. We're working with -- that's why I was interested in last year's transcripts. One thing I may have in common with this group is interest in the longevity of D1. We worry about the moisture content of the tape at the Library, too.

We make -- the analogy, I think, for our policy, quickly, would be when we make a transfer on audio, we make both an analog and a digital copy because we're trying to have something retrievable for 200 years, and because we have anecdotal evidence accumulating that's not cheering, such as not being able to read time codes and things like that. In fact, I guess our most extreme position would be the one we've taken with the Marlboro Music Festival. They've been sending us recordings of the festival for years. When they started sending us digital recordings, we said, in effect, "Thanks for the recordings. Now we want the machine they're recorded on, too," because we've got to be sure we'll be able to play them back." That may be an extreme position, but I guess that's the way our thinking goes.

VOICE: (Off microphone.) ...Movietone News

MR. PARKER: I didn't see it personally. I've talked to fellow archivists about it. I have a prejudiced, bigoted opinion of it without having enough information to even be worthy of having an opinion. However, would you like to hear my opinion?

So far, it has nothing to do with preservation. It has to do with access. The preservation part of it does not meet our criteria, to put it mildly. These films go back to c. 1919. There's yet to be any test of film in shrunken, curled or otherwise unsatisfactory physical condition being transferred. I don't know whose criteria it might meet. We'll find out as they work it out. It may mean that a lower level of preservation is acceptable for some applications. But if you've got gorgeous, breathtaking 35 mm images, to reduce them to that kind of quick, easy access only does part of the job, I think.

Although, that would be half the solution that I would see as ideal somewhere along the way. But I would say you start with retaining the information that is there in some kind of master copy and then make it available for prompt use that way. And my boss, who just

retired, was once called a bad name by a frustrated documentary film maker at the top of his voice because the Library, then working through an outside lab -- we didn't have our own in those days -- couldn't meet his deadline for television.

So that part of the problem, the Fox has got -- let me say something nice about the studios. You know, we're not -- I feel like Teddy Roosevelt: "Alone in Cuba" should be the name of my address here.

There are four archives that conserve this same kind of material in the United States, as well as the film companies. I saw something wonderful in last year's program about assets, preserving and protecting assets. That's a new idea, instead of nitrate just being this stuff that explodes on you and costs a lot of money. And one of the major companies that just built a beautiful restored vault for nitrate films, state-of-the-art facility, calls it "asset protection". Why didn't we think of calling it that in 1969?

VOICE: (Off microphone.)

MR. PARKER: It's here. I can work it out with you afterwards.

VOICE: (Off microphone.)

MR. PARKER: I could have gone on with several more formats, you know, after super 8mm and 78 records. By a reel, the industry, since the '30s, has considered a reel about 8 to 10 minutes of running time. When we get these reels, they may come off the airplane in 3,000' reels. Typically, with original 35 mm negatives, you don't store anything larger than 900 to 1,000 feet a reel.

So the average A budget picture in the '30s runs 10 to 12 reels. A Fred and Ginger musical may run 10 to 12 reels, although its running time may be only 90 minutes, because they don't want to cut right in the middle of one of the numbers of where the reel breaks go.

However, once when the Library of Congress had a total of three people working on motion pictures and the industry had changed over from 1,000 foot as its standard length to 2,000 foot, because everybody now had projectors with take-up reels with 2,000 feet capacity, there was one guy, I understand -- and I've seen some of the musicals, so I think it's true -- who had a machete. If he had a 2,000-foot reel that came in for copyright, about 1,000 feet in, he would whack it with the machete so it would fit in the 1,000 cans. He only had 1,000-foot cans. We couldn't buy 2,000 foot. And he didn't miss a musical number; it was sort of amazing--whacked right in the middle of each one. I don't know about the others, just the musicals I went through.

VOICE: (Off microphone.)

MR. PARKER: It's difficult having to operate many kinds of equipment at once, and we've had special programs transferring cylinders.

Let me tell you about the amateurs who recorded wax cylinders, because what's semi-soft wax and what's stamped celluloid, what's original wax recordings, is one of the more exciting stories we have.

Indian rituals that would be lost to the memory of the tribe today are sometimes documented and retrievable by amateurs who went out with their portable cylinder machines. Long before the folk song project of the '30s that the Library of Congress is noted for, when they took tons of recording equipment in a truck right out in the field and recorded folk songs on site.

We had a special project for transferring these disks in the late `70's, and I remember vividly when we became part of the recorded sound division in a shotgun marriage. We would have a meeting around the great green table in a recording studio. But the project couldn't stop.

In the same room they were transferring native American chants at the same time. Yes, we don't deal with all obsolete formats the same way, but yes, we try to cover the waterfront.

VOICE: (Off microphone.)

MR. PARKER: Do we buy hardware?

VOICE: (Off microphone.)

MR. PARKER: Yes. In the case of the Vitaphone system, the disk system that brought sound movies, to popularity -- they'd been around forever, like 3-D, but they weren't popular -- the Vitaphone system we now share is in a lab in Hollywood with one of the other archives.

You see, they -- this is a symbiotic relationship. They have the soundtracks for these movies and we have the movies without any soundtracks. And there is a third factor: ***Ali Baba and the 40 Thieves***, I left them out. These are the collectors, bless their hearts, without whom I'd be out of business, because a lot of these things are not available at the studios or from copyright deposits, if the movies we're talking about are from the silent era or the very first years of sound.

And there's a record collectors group now, a consortium, which negotiates with the Library of Congress, because their collectors have the soundtracks and we have the films. It's getting more interesting. If you want to know what the Ed Sullivan Show would have looked like in 1927, we're about to be able to show it to you. Because in the first years of sound, twenty-four hours a day in a studio in Brooklyn, they set up four cameras, and anybody in show business who was appearing in town came in and did their act. They didn't cut away to Alice Faye kissing Don Ameche or keep the plot going during the act. You get to see the act ungilded. So you get to see somebody who had done the same act for 30 years on the stage, and in their thirtieth year they're recorded picture and sound for the vitaphone in 1926. That's sort of a reach- back.

Not as amazing as seeing a pope who was born in 1830 on a motion picture, which we can do with the paper prints of films made before the turn of the century, but we're getting back there. We're finding out that we're not necessarily better in every way than anybody else who ever lived in this country. I guess we learned that from Ken Burns' television series on the Civil War. He found people who were sensitive and intelligent and admirable and their experiences could be moving to us from that kind of presentation, and we're finding the same sort of thing as we go back to these obsolete formats and bring them back to life.

Not all of the films and recordings are equally wonderful, but enough are so that the pride of discovery is still there and the delight in finding something that communicates to us today is there.

Well, we've got Mickey Mouses now. Disney has deposited at the Library important material, World War II and -- the people who acted out "Snow White" live before the cameras, the animators translated it into drawings and much more. Please do come to visit our division at the Library of Congress. And if you give me enough advance notice, I'll try to crack out a Mickey Mouse to look at. Thank you.

# ATM TECHNOLOGY AND BEYOND

**Nim K. Cheung**
Bellcore
445 South St., Rm 2K-118
Morristown, NJ 07960-6438
Phone: (201) 829-4078
Fax: (201) 829-5886
nkc@faline.bellcore.com

## Abstract

Networks based on Asynchronous Transfer Mode (ATM) are expected to provide cost-effective and ubiquitous infrastructure to support broadband and multimedia services. In this paper, we will give an overview of the ATM standards and its associated physical layer transport technologies. We use the experimental HIPPI-ATM-SONET (HAS) interface in the Nectar Gigabit Testbed to illustrate how one can use the SONET/ATM public network to provide transport for bursty gigabit applications.

## Introduction

The phenomenal progress in telecommunication and computer technologies in the past decade has created an emerging demand for broadband and multimedia services. Such demand stimulated the development of the Broadband Integrated Services Digital Network (B-ISDN) [1-4]. B-ISDN is a standardized public switched telecommunications network infrastructure capable of supporting both broadband and narrowband services on a single flexible network platform. A key element of the standards is the use of Asynchronous Transfer Mode (ATM) for multiplexing and switching. ATM combines the advantages of both circuit- and packet-switching techniques, allowing many services to be transported and switched in a common digital format. Furthermore, the current recommendations include the Synchronous Optical Network (SONET) [5,6], also known internationally as the Synchronous Digital Hierarchy (SDH), as the physical layer transmission standard. ATM, together with SONET, is expected to provide the reliable high-speed transport, bandwidth flexibility, and integrated transmission and switching for a diverse set of traffic characteristics as required by B-ISDN.

This paper provides an overview of ATM and SONET standards and explores how the current broadband standards can be extended to accommodate bursty gigabit applications for supercomputers and high performance workstations. We use the experimental HIPPI-ATM-SONET (HAS) interface in the Nectar Gigabit Testbed as an example to illustrate how one can use the ATM-based public network to provide end-to-end transport for gigabit applications.

## Asynchronous Transfer Mode (ATM)

ATM is a high-speed and low latency packet-like switching technique. It employs fixed size packets, or "cells", with a 5-byte header and a 48-byte information payload. It is a connection-oriented technology whereby user data is transported through a network of switches over pre-defined routes. Because it is a statistical multiplexing and switching technology, ATM can operate over a wide range of data rates with different physical media. The latter can be optical fiber, twisted copper pair, or wireless medium. Depending on the media, the data rates can vary from 1.5 Mb/s to 2.5 Gb/s and beyond.

The protocol stack for ATM is shown in Fig. 1. Each layer performs the function corresponding to a similar layer of the International Standards Organization (ISO) Open Systems Interconnect

(OSI) communications protocol stack. The physical layer deals with the framing and physical transport of the data. The ATM layer addresses such issues as switching/routing and multiplexing. The ATM Adaptation Layer (AAL) converts the information from the higher layers to ATM cells and vice-versa. It handles the segmentation and reassembly of data packets into cells as well as service dependent functions such as timing and synchronization. The user plane deals with user application information while the control plane handles call and connection control. The management plane supports operations, administration, and management functions.

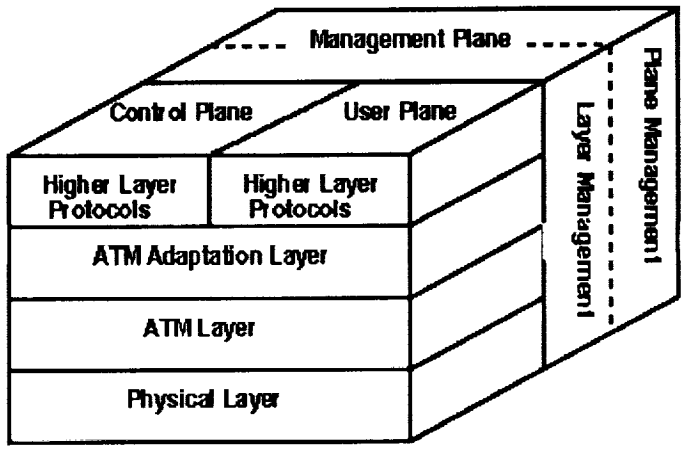



Fig. 1 ATM Protocol Reference Model

Unlike other communications technologies, ATM is designed to accommodate a variety of traffic types with different quality of service characteristics. ATM assigns all traffic to one of four basic classes as shown in Fig. 2. In Class A service, the data rate is constant and a timing relationship exists between the source and destination. Examples of such services are voice or fixed rate video. Class B services differ from those of Class A in that the bit rate can be variable. Examples include differentially encoded video. In Class C service, the connection between the source and destination is asynchronous and the data rate can be variable. An example is file transfer between computers. Finally, Class D is similar to Class C except that it is connectionless. An example of such services is connectionless data transport (e.g. UDP/IP or SMDS).

| | Class A | Class B | Class C | Class D |
|---|---|---|---|---|
| Timing Between Source and Destination | related | | unrelated | |
| Bit Rate | Constant | Variable | | |
| Connection Mode | Connection-oriented | | | Connectionless |

Figure 2: ATM Service Classes

## Advantages of ATM

ATM offers significant advantages over earlier networking technologies in supporting broadband and high performance computer networks. A key advantage of ATM is its high speed and associated low-latency. Many organizations are beginning to store documents in image formats. A high resolution image may contain 4096 x 4096 pixels each with 16 bits of gray scale resulting in a file size of 256 megabits. It would take over 25 seconds to transmit the image over a 10 Mb/s Ethernet. However, an 155 Mb/s ATM link would require less than 2 seconds. This difference becomes more pronounced when a set of data contain a large number of images. Furthermore, these images might only constitute a part of a complete file such as a detailed medical record. The amount of data could reach gigabit range and lower-speed technologies perform inadequately. Also, unlike Ethernet or FDDI, which are shared media technologies, ATM connected workstations have dedicated links to the networks. As a result, a user performing network-intensive operations is unlikely to affect other network users.

Another advantage of ATM is its flexibility. Since it was designed to support different types of traffic over the same network, ATM can be used as the single network technology for the various systems within an organization. Image transfer applications, which require high bandwidth and low loss, can use the same network as video conferencing system, which requires low latency exchanges. A common network platform would facilitate the interconnection of the critical information systems of the organization.

An aspect of this flexibility is that ATM is both a local area network (LAN) and a wide area network (WAN) technology. It can potentially offer seamless connectivity between an organization's internal network and the public network. This would not only facilitate the interconnection of geographically distributed facilities allowing for greater sharing of data but would also spur "telecommuting" whereby services could be provided to remotely located business sites. Because there would be no need for special internetworking units connecting the LAN to the public network, this could result in decreased startup costs and increased performance and reliability. Furthermore, ATM has a well-defined set of performance and management functions. For end-to-end applications, a homogeneous management structure can be employed to manage most of the public and private networks. This could result in considerable savings in cost and management effort.

ATM has become an international standard with the support of both the telecommunications and computer industries. In addition to the international standard bodies such as ITU-TS (International Telecommunications Union -- Telecom Sector, formerly called CCITT), the ATM Forum appears to be evolving into a new standard body guiding the implementation of ATM products into the marketplace. These emerging standards (for example, Ref. 7) are under intense scrutiny from both equipment manufacturers and users to help ensure that the specifications are implementable. Even though the specifications are not yet completed, many companies are currently offering products. For example, almost every major router and bridge vendor is offering, or has announced, support for ATM LAN interconnectivity either by marketing small (up to 64 ports) switches or by adding ATM interfaces to their existing products. Several companies are also offering ATM adapter cards for workstations and PCs. For WAN connectivity, most large switch manufacturers are marketing first generation switches and many local and long distance carriers in the U.S. have announced plans for offering ATM services within the next several years.

## Synchronous Optical Network (SONET)

Most of the ATM applications will likely be carried over the Synchronous Optical Network (SONET) [5,6] as the physical layer. SONET defines a standard set of optical interfaces for network transport in interoffice transmission and cross-connects, switching, local distribution, and local area networks in customer premises. It is a hierarchy of optical signals which are multiples (called OC-N) of a basic signal rate of 51.84 Mb/s called OC-1, or Optical Carrier at Level 1. The electrical counterpart of these optical signals are called STS-N, or Synchronous Transport Signal at Level N. The STS-N signals have standardized frame formats with a frame

duration of 125 microseconds (8 kHz). The STS-1 frame consists of 90 columns and 9 rows of 8 bit bytes. The STS-N signal is formed by synchronously byte-interleaving N STS-1 signals. The OC-3 (155.52 Mb/s) and OC-12 (622.08 Mb/s) have been designated as the customer access rates in future B-ISDN networks. Other important SONET rates are OC-48 (2.488 Gb/s) and, in the future, OC-192 (9.953 Gb/s). Of special interest to B-ISDN and gigabit networking is the Concatenated Synchronous Transport Signal Level N (STS-Nc) which is an STS-N signal in which the N STS-1s have been combined together as a single entity and is transported not as several separate signals but as a single channel [5]. The concatenated signal provides a contiguous high speed channel to support services that require large bandwidths.

The orderly, synchronous structure of the SONET/SDH concept simplifies multiplexing, and reduces significantly the amount of network equipment for each node. As a result of international standardization, SONET/SDH allows the interconnection of different manufacturers' products at the optical level, and facilitates optical mid-span meets. The flexible payload structure can accommodate virtually any type of digital signal, and provides a flexible platform for future services. The definition of SONET also includes provisions for standardized operation and maintenance support which will become a key consideration in the implementation of future broadband networks.

Recently, there has been considerable standards activities at the ATM Forum to propose methods of sending SONET and ATM signals over unshielded twisted copper pairs (UTP) at data rates of up to 155 Mb/s (STS-3c rate) for short distances (up to 100 meters). A key application of SONET over UTP is to provide low cost ATM connectivity all the way to the desktop within an organization. The basic SONET rate of STS-1 (51.84 Mb/s) has further been extended to subrates of 25.92 and 12.96 Mb/s in modulo 2 fashion to support lower speed applications at the desktop. The most important SONET rates and their equivalent in the international SDH hierarchy are listed in Fig. 3.

| OC Level | STS Level | SDH Level | Line Rate (Mb/s) | Remark |
|---|---|---|---|---|
| | Scalable SONET | | 12.96 | Modulo 2 on UTP |
| | Scalable SONET | | 25.92 | Modulo 2 on UTP |
| OC-1 | STS-1 | | 51.84 | OK on UTP |
| OC-3 | STS-3 (c) | STM-1 | 155.52 | B-ISDN UNI OK on UTP |
| OC-12 | STS-12 (c) | STM-3 | 622.08 | B-ISDN UNI |
| OC-48 | STS-48 (c) | STM-16 | 2488.32 | |
| OC-192 | STS-192 (c?) | STM-64 | 9953.28 | |

Fig. 3: Key SONET rates and their SDH equivalents

## ATM over Satellite and Wireless Transport

In addition to the terrestrial networks, one may also carry ATM over satellites or other wireless means. Although satellites can support link rates that are orders of magnitude less than fiber links, the basic ATM transport rates — DS-3 (45 Mb/s), SONET STS-1 (51 Mb/s), and STS-3 (155 Mb/s) — can be supported over the current generation of satellites or other wireless media. It is expected that the next generation of satellites can support link rates of STS-12 (622 Mb/s) and higher data rates [8]. A hybrid fiber-satellite ATM-based computer network would significantly extend the reach of a terrestrial network to remote areas.

## Extension of SONET/ATM to Support Gigabit Data Services

With the advent of gigabit networking, networks capable of transporting bursty gigabit/sec data packets will be necessary to satisfy the increasing bandwidth demands for communications among supercomputers and large data archives. An example of such a high speed network is the local area network employing the High Performance Parallel Interface, or HIPPI [9-11]. HIPPI was proposed by the ANSI X3 standards committee for transmitting digital data at peak rates of 800 or 1600 Mbit/s between high performance computer equipment. HIPPI, however, is defined only for twisted-pair copper cables over a maximum distance of 25 meters, or serial point-to-point HIPPI extenders over private fiber links. To take advantage of low-cost shared facilities of the ubiquitous public network, it would be highly desirable to interconnect HIPPI hosts over much longer spans across the public metropolitan and wide area networks. The SONET/ATM-based B-ISDN networks offers an attractive solution for such applications [12].

## The HIPPI-ATM-SONET (HAS) Interface

In this section, we outline the experimental HIPPI-ATM-SONET (HAS) interface which is one of the first attempts in investigating the transport of bursty gigabit/sec data packets over a SONET/ATM-based public network. The HAS is a key component of the Nectar Gigabit Testbed [13], and is implemented in a collaboration between Bellcore and Carnegie Mellon University. The role of the HAS interface in the Nectar Testbed is shown in Fig. 4.

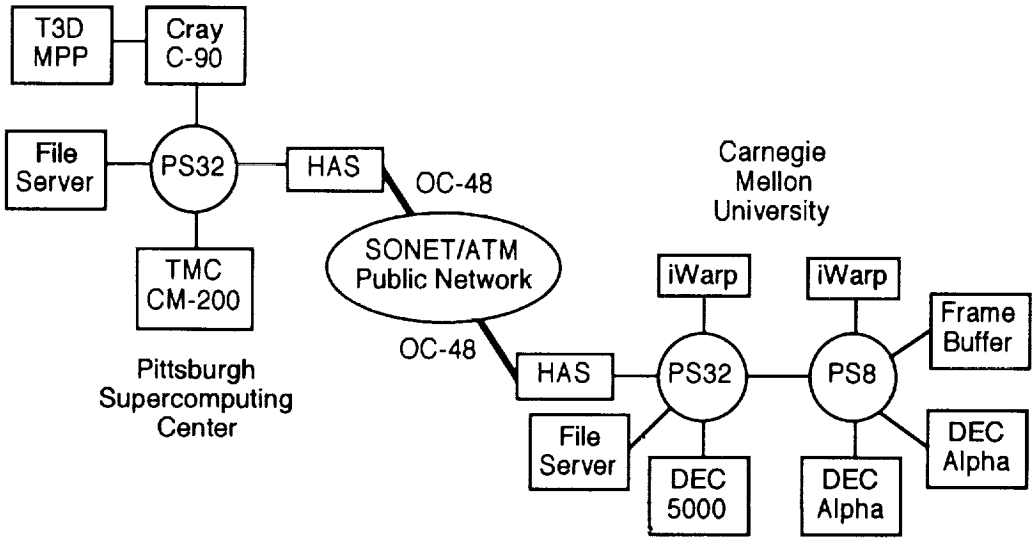



Fig. 4 The Nectar Gigabit Testbed

The basic functions performed by the HAS are as follows. In the transmit direction HIPPI packets from a HIPPI-based host computer or router are terminated on the HIPPI module (Fig. 5). The data within the packet, along with routing information is passed to one of eight ATM/ATM Adaptation Layer (ATM/AAL) modules, each corresponding to one SONET STS-3c channel. The ATM/AAL modules convert the routing information to ATM virtual circuit information, segment the HIPPI data into ATM cells and provide for various forms of error checking. The ATM cells are then passed to the SONET module where they are mapped into SONET STS-3c (155.52 Mb/s) channels. Eight STS-3c channels are used for one HIPPI channel; the testbed will accommodate up to 16 STS-3c channels so that eight additional channels could be used for a second HIPPI channel. The sixteen parallel STS-3c channels are then multiplexed up to the STS-48 rate (2.488 Gb/s) and converted to an optical OC-48 signal for transmission across the network. At the receive side, the OC-48 signal is received, converted to the electrical STS-48 signal and demultiplexed to sixteen parallel STS-3c channels. ATM cells are extracted out of the SONET payload and rassembled into HIPPI data units. ATM routing information is converted to HIPPI routing information and the HIPPI packet is then reconstructed. The architecture of the HAS interface is modular so that the HIPPI module could be replaced by another high-speed data communication interface.

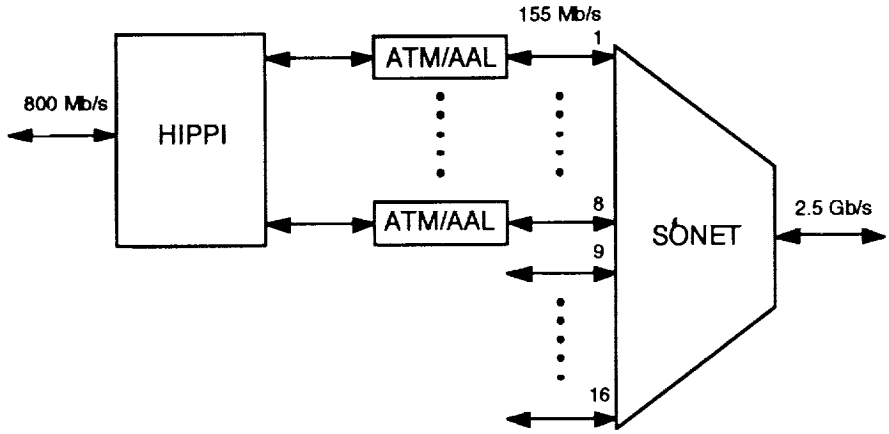



Fig. 5 Schematics of HIPPI-ATM-SONET Interface

As described above, the method chosen for transmitting HIPPI packets is to terminate HIPPI physical-layer signals, such as handshaking control signals, at the network interface rather than transporting them across the network. This approach has the advantage that it does not depend on HIPPI's link layer flow control mechanism, which resembles the window-based flow control mechanisms used in lower-speed data networks. Window-based flow control mechanisms may not scale well to gigabit/second speeds because they control only the number of outstanding packets in the network, not the traffic flow rate. The approach of terminating HIPPI and transporting the data in the form of ATM cells allows rate-based traffic control mechanisms to be used and also provides the flexibility to adapt the HAS interface to transport other types of data traffic.

## Conclusion

We have reviewed the key features of the ATM and SONET standards for the B-ISDN public network which is expected to provide an ubiquitous infrastructure for the emerging broadband and multimedia applications. These applications will undoubtedly include gigabit/sec data transfer among high performance computing devices ranging from supercomputers to mass storage archives.

## Reference


[1] "Broadband ISDN Switching System Generic Requirements", Bellcore Technical Advisory, TA-NWT-001110, Issue 1, Aug. 1992.

[2] "Broadband ISDN User to Network Interface and Network Node Interface Physical Layer Generic Criteria", Bellcore Technical Reference, TR-NWT-001112, Issue 1, June 1993.

[3] "Asynchronous Transfer Mode (ATM) and ATM Adaptation Layer (AAL) Protocol Generic Requirements", Bellcore Technical Advisory, TA-NWT-001113, Issue 2, July 1993.

[4] "Generic Requirements for Operations of Broadband Switching Systems", Bellcore Technical Advisory, TA-NWT-001248, Issue 1, Oct. 1992.

[5] "Digital Hierarchy — Optical Interface Rates and Formats Specifications", American National Standard for Telecommunications, ANSI T1.105-1988, Sept. 1988.

[6] "SONET Transport Systems: Common Generic Criteria", Bellcore Technical Reference TR-NWT-000253, Issue 2, Dec. 1991.

[7] "ATM User-Network Interface Specification, Version 2.0", ATM Forum Document, June 1, 1992.

[8] N. R. Helm, H. J. Helgert, and B. I. Edelson, "Supercomputer Networking Applications", NASA Advanced Communications Technology Satellite Conference, Washington DC, November 18-19, 1992.

[9] "High-Performance Parallel Interface — Framing Protocol (HIPPI-FP)", ANSI X3.210.199x, March 23, 1992.

[10] "High-Performance Parallel Interface — Encapsulation of ISO 8802-2 (IEEE 802.2) Logical Link Control Protocol Data Units (HIPPI-LE)", ANSI X3.218.199x, September 14, 1992.

[11] "High-Performance Parallel Interface — Physical Switch Control (HIPPI-SC)", ANSI X3.222.199x, February 10, 1992.

[12] N. K. Cheung, "The Infrastructure for Gigabit Computer Networks", IEEE Communications Magazine, Vol. 30, No. 4, pp.60-68, Apr. 1992.

[13] R. Binder, "Networking Testbeds at Gigabit/second Speeds", Optical Fiber Communications Conference Digest, paper TuE1, p.27, San Jose, CA, Feb.2-7, 1992.

# Issues for Bringing Digital Libraries into Public Use

**David W. Flater**
University of Maryland Baltimore County
Baltimore, MD 21228 U.S.A.
Phone: 410-455-3000 Fax: 410-455-3969 davidf@cs.umbc.edu

**Yelena Yesha**
University of Maryland Baltimore County
Center of Excellence in Space Data and Information Sciences
NASA Goddard Space Flight Center
Greenbelt, MD 20771 U.S.A.
Phone: 410-455-3000 Fax: 410-455-3969 yelena@cesdis1.gsfc.nasa.gov

## Introduction

The U.S. Government has recently begun a renewed effort to support the research and development of digital libraries. This happened as part of a larger package intended to hasten the evolution of a new and improved information infrastructure. The project is intended to benefit everyone to some extent, not just the institutions that currently have wide-area network access[1]. The questions facing us now are therefore not of an exclusively technical nature. People hearing the phrase "digital library" are bound to worry about things like the relative availability of romance novels when the public library on the corner decides to go electronic.

Such worries are not entirely unfounded, and we will attempt to address them in this paper. We will also try to answer the questions, What exactly *is* a digital library? How long have they been around? What are they likely to become, and what has that got to do with romance novels, anyway?

## Digital Libraries Today

Depending on what sort of digital library is being discussed, one may claim that they already exist or that they could not exist for at least ten years. To some, a digital library is any collection of electronic books. This phrase can be taken loosely to include any kind of structured data, so that existing archives and even databases could be construed as digital libraries. However, in the future, the phrase electronic book is likely to refer to a particular standardized encoding for arbitrary collections of data. Electronic books, or e-books, can encompass quite a lot more than what is currently thought of as a book. Standards already exist for how to encode pictures, sound, and movies for storage, retrieval, and playback on a computer. Furthermore, e-books can be interactive. An e-book can give you all of these elements together in something reminiscent of a video game, it can be the ASCII text of the world's most boring stereo instructions, or it can be any combination, however well orchestrated or poorly pulled-off. Today's personal computers already have the processing and storage capacity to support full-fledged e-books, let alone today's low-cost workstations. What remains for the future, some would say, is to make use of these resources with digital libraries. Digital libraries are destined to be the first true archives of *integrated* multimedia data. Today we have corpora of text, image, and sound files; soon we will have corpora of electronic books which contain all these things in a single organized package.

To others, a digital library is not just any collection of e-books, but a facility which makes use of a limitless number of resources which are accessed through a wide-area network. The focus is not upon individual "books," but on the data repositories which provide them. The digital library as such would be the unification of all the available repositories through a single interface. That interface would naturally be able to cope with whatever kind of data, multimedia or otherwise, is retrieved. While it might seem as if the production of multimedia

e-books would be the greatest challenge, that technology already exists. The intelligent management of networked resources is what remains a difficult topic for research. This research has been going on for some time in an attempt to manage the already existing repositories of data which are public on the Internet. With the new initiative to build a better information infrastructure, it has become even more important that we improve the ways in which we manage, locate, and retrieve information over wide-area networks.

The above two "definitions" of digital libraries are only mutually exclusive in the sense that they both lay claim to the same buzzword. There is nothing really preventing those technologies from being integrated. Individual "digital libraries" will be needed to provide e-books to the world, and we will need some way of providing access to those libraries that allows users to find the repositories they need and to access them without undue trouble. Although the latter topic is more interesting from a research perspective, the former is more important to ordinary people who are more interested in what digital libraries can do for them than in how it is accomplished.

## Digital libraries in the Future

It is probably a safe prediction that many serious efforts will be made to build digital libraries. A lot of money will be spent on getting books and other items onto computer media, on the physical media themselves, and on the software and hardware required to read these electronic books. However, after this startup period, very little of the cost of a real life digital library will have to do with the cost of building or maintaining the library itself. The real cost will be licensing fees.

Most proponents of digital libraries want to see them benefit everyday people, not just academicians and engineers. In order for digital libraries to be a popular success, they must contain popular e-books. In order for publishers to provide e-books, they must make a comfortable profit. Therefore, we may predict that if popular entertainment shifts into the realm of digital libraries, it will be after sufficient propagation controls and billing procedures have been put in place to insure that whoever has to pay for the e-books will pay for all those who make illicit copies as well. With sufficient effort, the number of illicit copies could be small indeed. The propagation controls could be as draconian as any which have ever been applied to computer software, and all the more sinister for having had so many years to be perfected. It could be insured that a customer has the use of a data object exactly *once* with a sufficiently encrypted data object and an obfuscated, self-destructing piece of software which is necessary to make use of the data. The software could insist on contacting an authentication server maintained by the publisher before executing, so that copying the software itself would achieve nothing. First-run movies and premium sports broadcasts would probably be handled this way, and the companies providing the copy protection stand to make as much money as their clients.

On a more pedestrian level, publishers will want to sell copies of more ordinary e-books to people and hope that the cost of blank media will be sufficient to discourage truly rampant piracy. When digital phone service becomes widely available in the U.S., a new species of 900 numbers will appear. Small computer companies, always in search of a niche with profit potential, will offer access to well-maintained and well-stocked digital libraries. These libraries will offer reasonably new movies, music, periodicals, and books, similar to today's video, record, and book stores. Customers may pay a specific price for each item they retrieve, a per-kilobyte charge, or both, but the objects retrieved will not be subject to draconian controls and customers will be able to use them as much as they please. A large portion of the proceeds from these operations will be returned to the publishers in the form of licensing fees.

Digital libraries owned by educational institutions and research companies, containing specialized scientific and professional e-books, will probably be more open to the public than such libraries are today, but few of these institutions will miss the opportunity to charge for access. In fact, they may be forced to charge just to cover the licensing fees for the e-books, which, like the subscription rates for today's scientific journals, will be high in order to compensate for their relatively small circulation. In the present time, it is already the case that

institutions are charging for access to databases which merely catalog the titles, authors, and abstracts of current periodical literature. A researcher who does not have access to such a database is at a noticeable disadvantage when trying to find related work. The free access databases cannot afford to be as current or as complete as the pay-databases. In the future, the situation will be the same with digital libraries of a technical nature, and those without access will be a step behind in their research.

Public libraries will presumably retain their traditional, underfunded role after they become digital and concentrate on whatever they can afford to provide -- maybe romance novels, maybe shareware e-books. Most government funding will go to educational institutions and to public libraries, and that funding which goes to public libraries will be a pittance compared to what the public at large will spend on premium entertainment through private companies.

It may sadden some to believe that quality news and entertainment will be just as expensive in the future as they are today due to licensing fees, but let us consider the alternative. Suppose that public digital libraries, funded by the government, were the only game in town. Let us even suppose that they were *well* funded and could afford to house whatever they chose. Having spent federal funds on the deployment of a digital library, the government will set standards on what sorts of e-books are worth procuring. We may boldly predict that this will include mostly educational material, but with a large section of romance novels. Without independent digital libraries, e-book publishers will have no incentive to produce e-books other than tutorials and romance novels. Romance novels would effectively become the only officially sanctioned form of entertainment.

Let us give thanks, then, for the independent retailers and for the commercialized publishers that bring us so many choices in what to read. Indeed, e-books could take commercialism to previously inconceivable levels of shallowness as advertisers cash in on the element of future shock. Imagine reading a newspaper in standard black print on a white background, turning the page, and being assaulted by a full color, moving, speaking advertisement with CD-quality Surround Sound [1] blasting subliminal messages from seven different directions. Had enough? Not yet: it could be interactive. An advertisement could *argue* with you until it gets your credit card number. If there is any vestige of human compassion left in corporate society, it will be possible to dismiss these solicitations, or at least to iconify them until they give up and go away.

## Help Wanted

Bringing digital libraries to life will require us to make significant progress in the following areas:

- Integration of multimedia data. An extensible standard must be developed for the seamless integration of multimedia data objects into e-books. Hawking proprietary formats will only lessen the chances of long-term success.

- Licensing and copyright issues. While the digital library medium must be open to the public, the contents of e-books must be protected in law and in practice from unauthorized use and duplication. Researchers are still thinking of new problems in this area as they seek to solve the old ones[2,3].

- Wide-area networking. While our desktop computers have enormous capacity for processing and storing information, our wide-area networks have limited bandwidth and accessibility. Public use of digital libraries will require much greater bandwidth. Upgrading to faster network technology will help, but we must also learn to make more efficient use of network resources lest we again use up all the available bandwidth. Alibi[4] is a software system now in development to help us achieve this goal.

[1] Dolby Surround is a registered trademark of Dolby Laboratories.

- Resource discovery. Users must be able to locate the digital libraries which would be useful to them. A number of systems are already in place to support resource discovery through browsing or through cataloging of network resources. Alibi uses a more flexible method for finding resources which may prove to be more useful in the long run. Knowbots[1] are another strategy which is being investigated for resource discovery and information retrieval in digital libraries.

- Information retrieval. How are we to keep track of all our e-books and find the ones that users want? After years of research into ways of retrieving textual documents, we are still not able to automate the process of accurately cataloging arbitrary documents for later retrieval. Research into image retrieval is relatively young, and sound retrieval may be unresearched. Presumably publishers of e-books would be willing to provide catalog entries for them if we could agree on a format for the catalog as well. This is a complicated problem in itself, since the catalog must contain enough data to be useful but not enough to overwhelm the search. There is great potential for innovation in the development of algorithms for retrieving e-books.

## Conclusion

In much the same way that the field of artificial intelligence produced a cult which fervently believed that computers would soon think like human beings, the existence of electronic books has resurrected the paperless society as a utopian vision to some, an apocalyptic horror to others[5]. In this essay we have attempted to provide realistic notions of what digital libraries are likely to become *if* they are a popular success. E-books are capable of subsuming most of the media we use today and have the potential for added functionality by being interactive. The environmental impact of having millions more computers will be offset to some degree, perhaps even exceeded, by the fact that televisions, stereos, VCRs, CD players, newspapers, magazines, and books will become part of the computer system or be made redundant. On the whole, large-scale use of digital libraries is likely to be a winning proposition.

Whether or not this comes to pass depends on the directions taken by today's researchers and software developers. By involving the public, the effort being put into digital libraries can be leveraged into something which is big enough to make a real change for the better. If digital libraries remain the exclusive property of government, universities, and large research firms, then large parts of the world will remain without digital libraries for years to come, just as they have remained without digital phone service for far too long. If software companies try to scuttle the project by patenting crucial algorithms and using proprietary data formats, all of us will suffer. Let us reverse the errors of the past and create a truly open digital library system.

## References


[1] R. E. Kahn and V. G. Cerf. The Digital Library Project, Volume 1: The World of Knowbots (DRAFT). March 1988.

[2] J. R. Garrett and J. S. Alen. Toward a Copyright Management System for Digital Libraries. 1991.

[3] Workshop on the Protection of Intellectual Property Rights in a Digital Library System. May 1989.

[4] D. W. Flater and Y. Yesha. An Efficient Management of Read-Only Data in a Distributed Information System. International Journal of Intelligent and Cooperative Information Systems, Special issue on Information and Knowledge Management. To appear, 1993.

[5] Discussions seen on comp.infosystems and related newsgroups, June 1993.

N94-33822

# A Data Distribution Strategy for the 90s (Files Are Not Enough)

**Mike Tankenson and Steven Wright**

Jet Propulsion Laboratory, Telos Systems Group
4800 Oak Grove Drive, 525-3670
Pasadena, CA 91109-8099

Virtually all of the data distribution strategies being contemplated for the EOSDIS era revolve around the use of files. Most, if not all, mass storage technologies are based around the file model. However, files may be the wrong primary abstraction for supporting scientific users in the 1990s and beyond. Other abstractions more closely matching the respective scientific discipline of the end user may be more appropriate. JPL has built a unique multimission data distribution system based on a strategy of telemetry stream emulation to match the responsibilities of spacecraft team and ground data system operators supporting our nations suite of planetary probes.

The current system, operational since 1989 and the launch of the Magellan spacecraft, is supporting over 200 users at 15 remote sites. This stream-oriented data distribution model can provide important lessons learned to builders of future data systems.JPL's Multimission Ground Data System (MGDS)

## JPL's Multimission Ground Data System (MGDS)

JPL's MGDS is a distributed, workstation based, ground data system that provides on-line, near-line and off-line storage for all telemetry, ancillary and processed data in support of the Voyager, Magellan, Galileo, Mars Observer, and Ulysses missions. In the future the MGDS will support the MESUR Pathfinder mission, the CASSINI mission to Saturn, and the mission to Pluto currently in the early planning stages. The MGDS began development in 1985 as the Space Flight Operations Center (SFOC) software upgrade following the successful prototyping effort to apply workstation technology to support the Voyager encounter with Uranus and continues through today as part of the Advanced Multimission Operations System (AMMOS) with mission support and maintenance activity.

The MGDS provides a Project Data Base (PDB) for each mission Consisting of two parts:

- A Telemetry Record-Based System.
- A File-Based System to support data products processed at levels 2 and above.

The file-based system is in close harmony with systems proposed for EOS. The file storage system consists of science and engineering file data products, and a catalog constructed using relational database technology (Sybase). The MGDS supplies a variety of tools for browsing the catalog and importing and exporting products to and from the system.

The telemetry-record based system, the subject of this paper, consists of the set of all Level 0 and selected Level 1 mission telemetry products and related ground data system information. Specifically, the telemetry-records based system contains:

- Spacecraft Engineering Data
- Decommutated (channelized) Spacecraft Engineering Data
- Level 0 and Level 1 Science Data
- Deep Space Network (DSN) Monitor Data
- Radio Science Data
- Quality, Quantity and Continuity (QQC) Data.

The telemetry-record base system is implemented as the Telemetry Delivery Subsystem (TDS) and supported by the Central Database Subsystem (CDB).

The MGDS System Architecture is based around a set of project Local Area Networks (LANs) interconnected over a high speed backbone (Figure 1). Wide Area LANs are supported to the Magellan spacecraft team in Denver, and to PIs/CoPIs all over the country for Mars Observer. Each project LAN has a CDB for non-real-time data storage, and a TDS for near real-time (NERT) and real-time telemetry data access. The basic architectures of these two systems are common among projects -- typically, project specific adaptations require changes to tables along with minimal software changes.

## The Spacecraft Team and Operations Support

The Telemetry Delivery Subsystem's primary role is to support the daily activities of the individual Mission Spacecraft Teams, the supporting Multimission Control Team (MCT) and Data System Operations Team (DSOT), and to provide science investigators with access to their primary data. The function of the Spacecraft Team is to operate the spacecraft, monitor its health, perform routine calibration and maintenance and deal with spacecraft anomalies. Spacecraft teams consist of spacecraft subsystem analysts (power, propulsion, command and control, ... ), a navigation team, telecommunications analysts and others typified by the Mars Observer team with over 40 personnel (including management and staff). During periods of routine operation, the Spacecraft Teams at JPL operate on a 40-hour 5-day work week as a baseline. On a typical work day, MGDS users will review engineering and ground data system data received since the end of the previous working day, and continue with real-time data throughout the day. Each element within the Spacecraft Team will summon data related to their area of responsibility from the TDS for processing and analysis.

The MCT and DSOT provide 24-hour monitoring of all spacecraft and operation and control of the JPL ground data system. For these teams, the primary role of the TDS is to provide operators with data to support problem resolution.

## Query Requirements

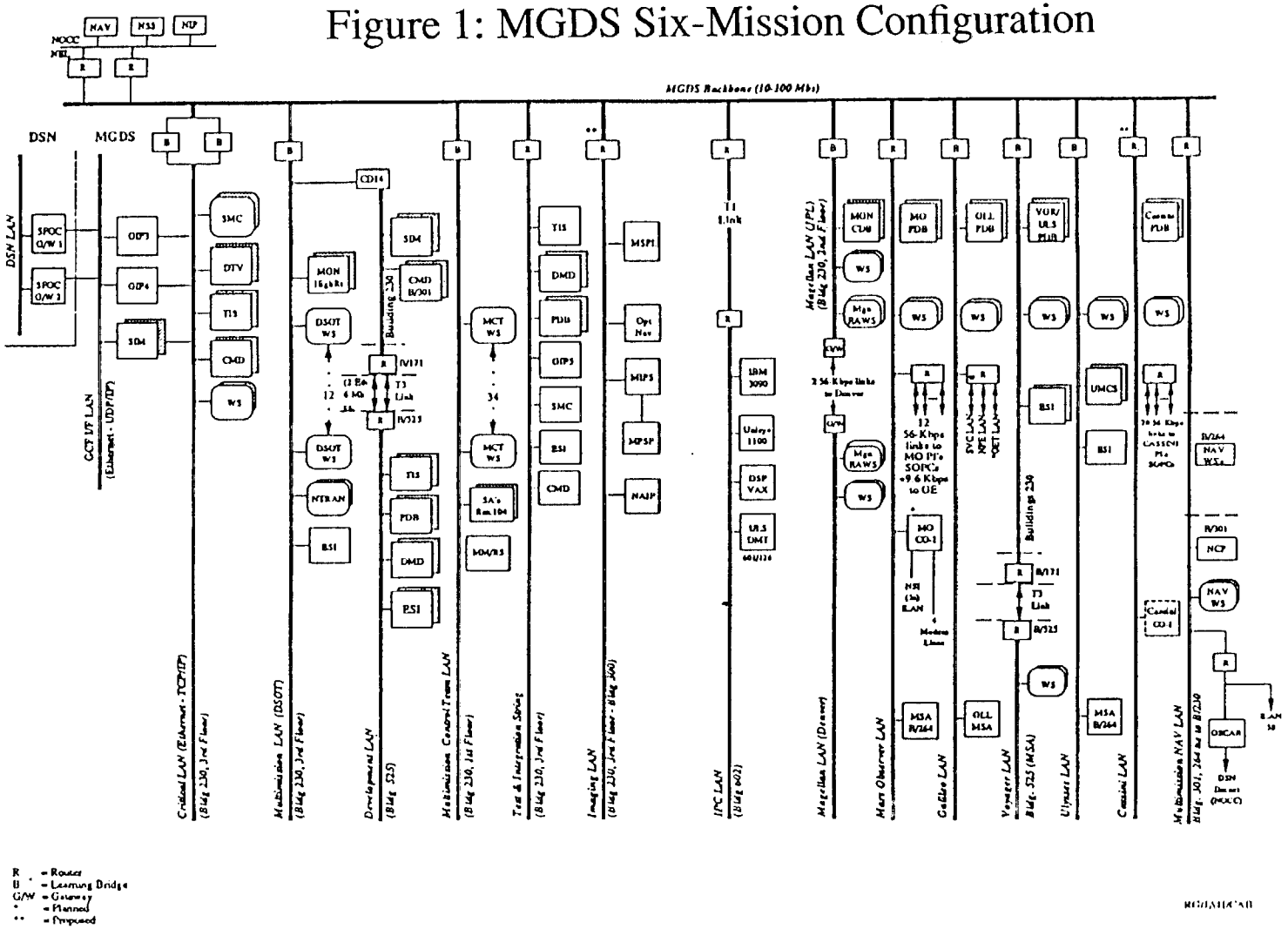
Typical scenarios supported by the Telemetry Delivery System include daily queries from the end of the previous working day right into the current real-time stream of return link telemetry; a 10- to 60-day trend analysis study; a query for retransmitted data from the DSN; ad hoc queries of on-line engineering data to support anomaly resolution on the spacecraft or ground data system and a query for science data from a local or remote principle investigator. To support all of these responsibilities the data distribution strategy of emulating telemetry streams was devised. A telemetry stream consists of a subset of processed telemetry data tailored to the needs and responsibilities of the user.

The strategy to emulate telemetry streams allows TDS to support on-line, interactive access to telemetry, access to real-time return link streams, and to provide seamless queries that transition from non-real-time to real-time telemetry data. Gap filling, overlap removal, besting are supported automatically and transparently. Because the data distribution model is based on a functional rather than an implementation model of the system, users can interact with the data system based on their operational view of the system with little or no knowledge about file systems, database manager internals, or data transmission protocols.

## Telemetry Record System Organization

To support our distribution strategy the telemetry data system is organized by mission, telemetry record type and, more fundamentally, by time. There is a plethora of clocks within the scheme of mission telemetry to support. Spacecraft Clock (SCLK), Spacecraft Event Time (SCET), Earth Receive Time (ERT), Record Creation Time (RCT), Monitor Sample Time (MST), Radio Science Sample Time (RSST), and even orbit number was proposed as a clock. Each of these clocks have unique behaviors which affect the ordering of data. SCLKs are subject to

# Figure 1: MGDS Six-Mission Configuration



spacecraft reset and tape recorders anomalies such as the "crap-in-the-gap" phenomena where old data was recorded and remained in between new recording periods. This old data is streamed back imbedded in the latest recorded data. SCETs are corrected for reset, but will be effected by "crap-in-the-gap". ERT is a simple, well-behaved clock, but is not homomorphic with the spacecraft clock because of recorder playback.

The clocks of primary interest to the spacecraft teams are SCET, and ERT. The preference typically reflects whether a mission has a short one-way light time (Magellan) where team members tend to work in terms of SCET, or a long one-way light time (Voyager) where team members seem to prefer ERT. The data system must handle both on an equal footing and produce a stream of telemetry that is ordered "as it occurred". Thus, queries by ERT are ordered by SCET unless specifically requested otherwise. To support this level of functionality, the telemetry database had to go through several incarnations.

## The Magellan Telemetry Database

Magellan was the pathfinder system and our first attempt to implement the telemetry stream access strategy. The approach taken on the Magellan system was to implement a Channel database (Borgen, [3]) in addition to a telemetry record database. To understand the Channel database we need some background. Planetary spacecraft (and presumably earth orbiting spacecraft) use the concept of commutation and decommutation to pack and unpack telemetry data during transmission to Earth. Commutation occurs on the spacecraft, and involves systematically sampling several sources of data and constructing a single telemetry frame from these samples. Each sample occupies an assigned position (as specified in a decommutation map) in a regular, repeating fashion. Decommutation occurs on the ground, where separation of the single telemetry frame into its component parts takes place based on their assigned position (as specified in the same decommution map) in a data frame. The channelization process is performed on the data after acquisition by matching each sample value with an explicit channel identifier. Thus, a channel is the output data from a single instrument or sensor, uniquely identified by the MGDS.

In the JPL telemetry world, there are various types of channels. Engineering channels correspond directly to spacecraft instruments and sensors. Monitor channels are added to the telemetry stream by the Deep Space Network (DSN) where tracking data, radio science and other quality indicators are produced. QQC channels are added to the telemetry stream by the Product Generation System and represent the processing analysis done on the raw data. Header channels are those values that correspond to the SFDU CHDO headers (discussed below) that are added to the data as part of telemetry processing. All of these channel data are packaged in the same manner and archived for later distribution to users.

This Channel database was an experiment where "channelized" telemetry was disassembled, and the individual channel records were stored into a relational database. The premise of the system was that users would be able to perform complex operations on channels using a relational model, and that the performance would be superior. In terms of performance, the Channel database was fairly fast for queries, but the loading suffered due to the overhead of loading thousands of data items into an RDBMS (compared to loading a file of data). There was also the problem that channels are dynamic and can be added to the system at any time by changing the commutation process on the spacecraft, or by introducing new channels through the DSN. Thus the PDB had to be able to deal with both time and channels and dynamic variables. This early experimental system was not pretty, but was able to formulate a tailored stream of telemetry in response to a request by a user. This capability formed the basis of our fundamental strategy.

## MGDS Multimission Telemetry Database Architecture

After the Magellan system went operational (and it is still in operation), the telemetry database was redesigned to enhance its multimission nature and resolve the issues associated with the Channel database. The redesign took advantage of the VANESSA prototyping effort to

support Voyager's Neptune Encounter. VANESSA placed greater emphasis on storing and retrieving telemetry records rather than individual channels in satisfying the needs of the science community for near real-time (NERT) access to data. Any channels needed were extracted "on-the-fly", either by the Query Server or by the users analysis tools. The channel database was dropped in the redesign and a simpler storage mechanism was implemented for near real-time data based on files (the NERT cache). To maintain our distribution strategy in the new system, data is separated by telemetry record type as it is recorded into files. These files are cataloged according to record type, and the start and end clocks of interest for that type. Data is queried from the NERT cache through a process of ordering the files according to their starting clocks and time merging the data across them.

The most challenging complication in this approach has been dealing with clock anomalies. In order to support queries by any of the clocks mentioned above, and to be able to order the data by any of those clocks "on-the-fly", the clocks associated with the data within any file have to be well behaved. This means that for all clocks of interest, the end time has to be greater than the start time, and clock values have to be monotonically increasing. To guarantee this behavior, algorithms were devised to detect clock anomalies as the data is being loaded. When anomalous behavior is detected, loading to the current file of data is closed out (and cataloged) and a new file started. If the new file has well-behaved clocks (just disjoint from the previous file), loading continues. If the clocks are poorly behaved they are isolated and query processing may or may not ignore them based on the query request.

The VANESSA prototype also had the capability to provide real-time access to data as it entered the system. Users of the VANESSA prototype were able to query from the past and into the future and receive stored and real-time data in the same query. This capability had been specified for the original Magellan system but was never implemented. Although the PDB provide near real-time loading of telemetry data, access to future data was impossible to implement in the context of an RDBMS because these systems will only support queries of data already existing within the database. The simpler NERT cache storage model has made it possible to implement a real-time query capability and provide data to end users directly as it is received and processed from the DSN.

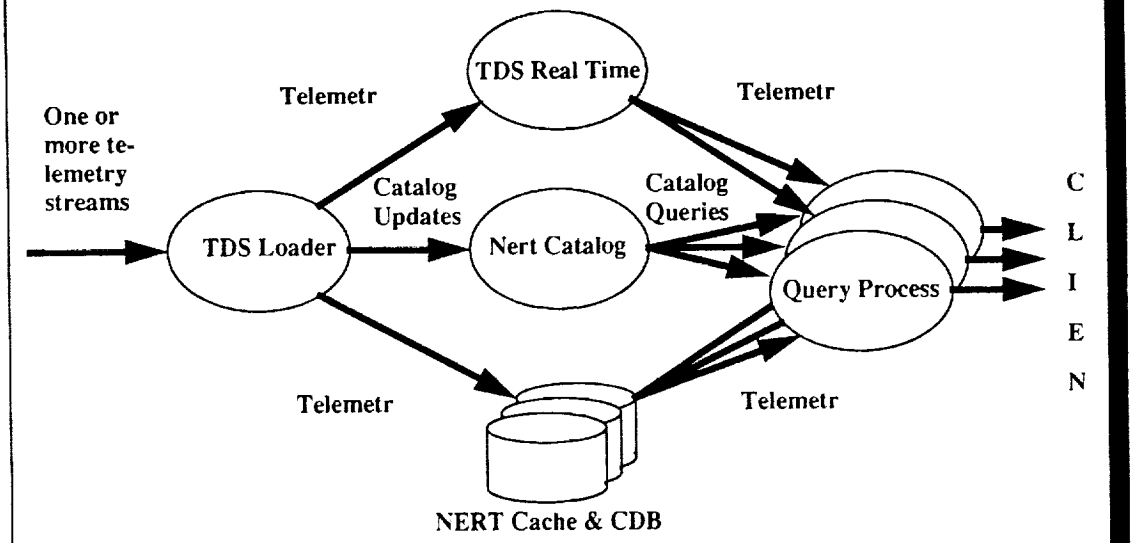
Finally, the initial concept of the NERT cache was as a shortterm storage location. Responsibility for longer term on-line storage was retained by the CDB subsystem. The NERT cache was intended to provide quick access to data and smooth out the operational irregularities of loading data into the CDB. The NERT cache has proven to be very robust and use of the CDB telemetry record storage system is starting to wane. Nevertheless, the final query system as implemented in the TDS provides seamless access to all three sources of data (CDB, NERT cache and real-time). Its architecture is illustrated in Figure 2.

## Application of Standards, The Standard Formatted Data Unit

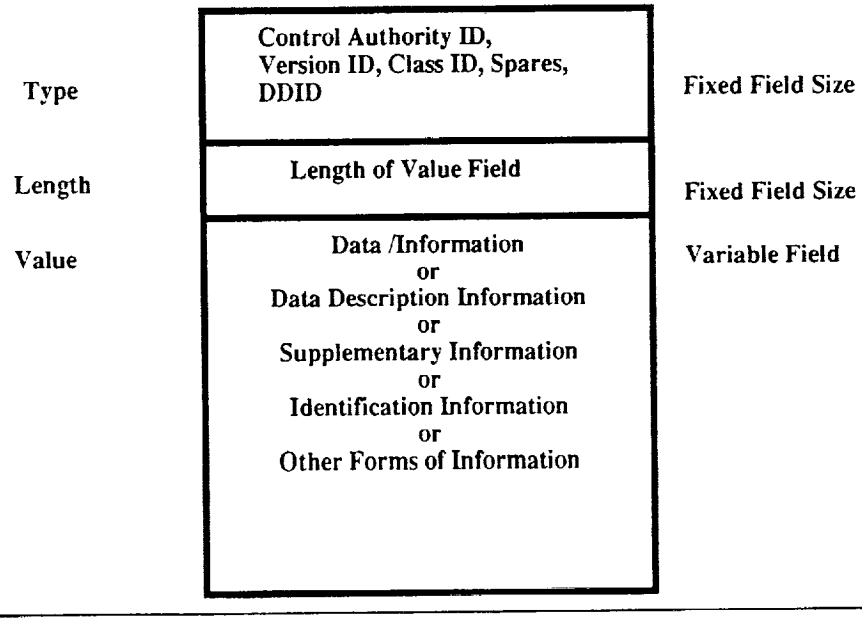
The Standard Formatted Data Unit (SFDU) has been critical to the development of the MGDS and its data storage system. SFDUs provide a way to globally define and identify data products for interchange among various software applications and international organizations (Miller, Elgin, [2]). The SFDU concept provides a means for globally defining and identifying data products; a means for aggregating instances of these data products; and a means for administering the data products definitions and descriptions to ensure their accessibility and understanding. The abstract nature of the SFDU has proven itself time and time again in constructing software to meet JPL's multimission requirements by providing sufficient polymorphic richness to characterize all telemetry data within the system.

The SFDU structure is derived from Label-Value-Objects (LVO) which are self-identifying and self-delimiting data records that follow the labeling rules of the Consultative Committee on Space Data Systems (CCSDS) or one of its Control Authorities. LVOs have a label element to identify the data object and give it length, and an element that contains the data values (data fields). High level SFDU structure guidelines are determined by the CCSDS and focus on standard labeling of data. These guidelines include rules to enable individual agencies to

## Figure 2: TDS Server Architecture




## Figure 3: Compressed Header Data Object



define their own detailed formatting specifications. JPL has adopted or developed several standards for formatting data within the SFDU including the Compressed Header Data Object and the Parameter Value Language (PVL).

## Compressed Header Data Objects

A Compressed Header Data Object (CHDO) is an LVO except that it has a shortened, 4-byte label to provide a compact envelope structure for telemetry, monitor and QQC data. The CHDO structure is used only for data exchange between MGDS subsystems. The CHDO label contains a 2-byte type field and a 2-byte length field. The fixed size of the length field places a 32-kilobyte limit on the size of CHDO-structured SFDUs. The type field contains an integer representation of type information sufficient for MGDS purposes (Figure 3).

CHDOs at JPL are enveloped within SFDUs with standard CCSDS labels, making the SFDU readable by other systems that use the SFDU standard. Within the SFDU header itself, JPL further defines subheaders (Figure 4):

- Aggregation subheader CHDO
- Primary subheader CHDO (required: data type, mission ID)
- Secondary subheader CHDO (optional: mission independent metadata)
- Tertiary subheader CHDO (optional: mission dependent metadata)
- Quaternary subheader CHDO (optional: mission dependent metadata)
- Data CHDO

The data ("metadata") fields of the primary, secondary, tertiary, and quaternary subheader CHDOs further define and identify the data. The headers are produced by the MGDS Product Generation System, and may be mission independent or mission specific. The content of these subheaders are defined by the projects.

## Parameter Value Language

The Parameter Value Language (PVL) is a simple ASCII language of the form "keyword = value;" plus some delimiting constructs. PVL provided a standard for expressing query requests, in ASCII, that could be encapsulated within an SFDU in a standard fashion (Figure 5, TDS Query Protocol).

## Data Aggregation

The Version 3 SFDU label provides the ability to create a variable length information product without requiring byte counts of the product's length. This was utilized by TDS to create an SFDU compliant query product that could be constructed and transmitted to the end user, on-the-fly, without having to stage the product locally to measure its size and fill in the label of the encapsulating SFDU.
The Version 3 SFDU labels support (in addition to others) the notion of delimiting an SFDU by an End Marker. The marker is embedded in the length field of the encapsulating SFDU and is paired with an End Marker Label at the end of the data product ( Figure 6, TDS Data Product).

## Stream-based Versus File-based Data Distribution

"Get away from files and filenames" (Dozier, [4]).

The easy way to manage data distribution problems involving extremely large datasets is to use files. The file model is universally understood and supported by all operating systems, storage systems and network transfer services. In addition, once the requested data is staged into files, there is nothing more for the data system to do other than to notify users to retrieve them. Presumably, users will have access to plentiful file transfer tools (commercial or public domain) and can perform the actual transfer themselves. Once the files are transferred, the job

## Figure 4: CHDO Aggregation, MGDS SFDU Stream Data Structure

| | | | |
|---|---|---|---|
| T | NJPL1I001231 | | |
| L | 699 | | |
| V | T | 1 (Aggregation CHDO ) | |
| | L | 132 | |
| | V | T | 2 |
| | | L | 4 |
| | | V | Primary Subheader |
| | | T | 15 |
| | | L | 80 |
| | | V | Secondary Subheader |
| | | T | 50 |
| | | L | 26 |
| | | V | Tertiary Subheader |
| | | T | 27 |
| | | L | 6 |
| | | V | Quaternary Subheader |
| | T | 28 | |
| | L | 180 | |
| | V | Data | |

## Figure 5: TDS Data Product

| | |
|---|---|
| Primary Label | CCSD3ZS00001<br>TDSQDATA |
| K-HEADER Label | NJPL3KS0L009<br>TDSQUERY |
| Data Product Identification | OBJECT = QUERYNAME;<br>...<br>END_OBJECT = QUERYNAME; |
| K-HEADER End Marker | CCSD3RE00000<br>TDSQUERY |
| Data & TDS Status Messages | NJPL... |
| End Marker Label | CCSD3RE00000<br>TDSQDATA |

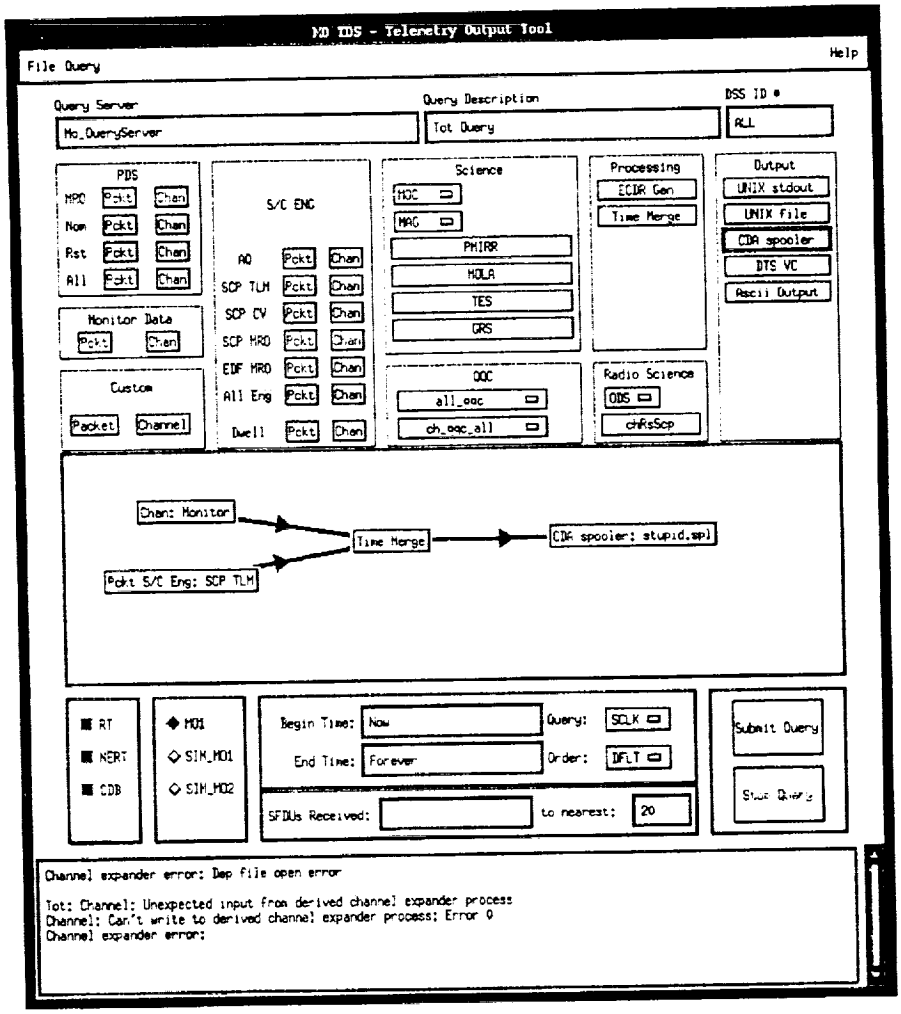
## Figure 6: TDS Query Protocol

| | |
|---|---|
| Primary Label | CCSD3ZS00001 |
| PVL SFDU Label | NJPL3ISOL009 |
| PVL Query Spec. | OBJECT = QUERYNAME;<br>DESCRIPTION = " ... ";<br>...<br>END_OBJECT= QUERYNAME; |
| PVL End | CCSD3RE00000 |
| End Marker | CCSD3RE00000 |

**Sample Query PVL**

```
OBJECT = Mo_Query;
DESCRIPTION = 'Tot Query';
REQUESTER_NAME = Al Sacks;
MISSION_NAME = MO;
SPACECRAFT_NAME = Mo1;
TIME_TYPE = ERT;
START_TIME=91/352T20:09:;
END_TIME = 91/352T21:09:;
GROUP = FRAME;
DATA_TYPE = sci_tes;
DSS_ID = ALL;
END_GROUP = FRAME ;
END_OBJECT = Mo_Query;
```

# Figure 7: Telemetry Output Tool



of the data system is complete -- it is the user's problem to get at the scientific data within the files.

## XBROWSE, from the University of Rhode Island (URI)

In contrast, data systems based on streams or other abstractions require more processing and system administration support, but enhance the usefulness of the system to end users. The 'xbrowse' system provides one such example.

The 'xbrowse' system, developed at URI (Kowallski, Gallagher, et al [11]) is a stream-based layer over a data system whose basic data abstraction presented to the end user is the image. Users make requests for images that, because of their size, are broken up into 'chunks' by sampling the high resolution images and transmitting a stream image of progressively higher resolution. The data is transmitted directly into data visualization tools at the client site (which may be local to URI, or over the Internet). The system allows users to view images and to throttle the incoming data interactively if the image is examined at low resolution and rejected. A file-based system would not be able to provide either of these capabilities directly.

## Telemetry Output Tool

Users of JPL's MGDS are provided with an interactive, point and click (and type a little bit) telemetry query tool called the Telemetry Output Tool (TOT). Users are presented with an abstraction that closely models the Telemetry problem domain. Figure 7 shows the TOT graphical user interface with widgets for selecting packets, channels, channel sets, time ranges, spacecraft, clock types, and so on. Once users have specified the query parameters for TOT (including the desired output), transfers occur 'in the background'. The requested data is packaged into standard SFDU objects and, if requested, delivered directly into workstation analysis tools [such as the MGDS Data Monitor & Display, (DMD)] over local and wide area networks. Users interact with the system via the telemetry stream abstraction with no knowledge of the underlying file or database management systems involved.

The 'look and feel' of the TOT interface is the same for all JPL missions. Each mission "adapts" the TOT through MOTIF resource files (TOT is constructed using the public domain Widget Creation Library, WCL, which affords considerable flexibility) rather than constructing new query applications for each new mission because the underlying abstraction is derived from the model for doing business at JPL.

## Building Custom Client Tools

As mentioned above, abstract views of a data system require extra processing by the system. Both TOT and 'Xbrowse' required custom software, at the client side, to properly present the system and ingest their data products. Unlike the file transfer model where standard FTAM and FTP tools can be assumed, no standards exist to construct these client tools. A first step in developing standard data system presentations in client-side software is to adopt some existing standards for data packaging (SFDU, HDF, etc.), and then provide enhanced client/server tools that understand the formats. To some extent, the NCSA tools supporting HDF are built on this model.

Although neither XBrowse nor TDS provide a general solution to representing space data systems to users, both are good examples of developing presentations to data system users which more closely model their particular problem domain.

## References


1. J.G. Kowalski, J.H.R. Gallagher, A. Reza Nekovei, P.C. Cornillon. Remote Access to Multi-Inventory Oceanographic Digital Data Archives. See also: J. Gallagher, P.C. Cornillon. "AVHRR Imagery and In-Situ Data Accessed Via Internet", EOS: Transactions: American Geophysical Union, Volume 74, No. 17, April 27, 1993, p. 204.

2. D. Miller and B. Elgin. Introduction to the Advanced Multimission Operations System (AMMOS) Lecture Course. December 18, 1992 M6 MOPS0511-00-05.

3. R. Borgen. "The Unique Problems of Using Relational Databases for Space Missions", DBA ShopTalk, Database Programming and Design, December 1991.

4. J. Dozier. Presentation the at the August 1992 EOSDIS Quarterly Review, Greenbelt MD.


## Acknowledgments


This work was done under government contract to the Jet Propulsion Laboratory and the California Institute of Technology.

# Management of the National Satellite Land Remote Sensing Data Archive

**Darla J. Werner**

**Hughes STX Corporation**
**Mundt Federal Bldg.**
**Sioux Falls, SD 57198**
**Werner@edcserver1.cr.usgs.gov**

## TEXT WAS NOT MADE AVAILABLE FOR PUBLICATION

# Alaska SAR Facility Mass Storage, Current System

**David Cuddy, Eugene Chu, and Tom Bicknell**

MS 300-319
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109, USA
Phone: (818) 354-6277
Fax: (818) 393-5184
dcuddy@sakai.jpl.nasa.gov

## Abstract

This paper examines the mass storage systems that are currently in place at the Alaska SAR Facility (ASF). The architecture of the facility will be presented including specifications of the mass storage media that are currently used and the performances that we have realized from the various media. The distribution formats and media will also be discussed. Because the facility is expected to service future sensors, the new requirements and possible solutions to these requirements will also be discussed.

## Introduction

Synthetic Aperture Radar (SAR) is an imaging radar technique that achieves high resolution through synthesizing the performance of a large aperture radar with a small aperture by processing the data from multiple samples of the beam footprint across the target in the azimuth direction. Typically the SAR data is collected from spacecraft or aircraft. To achieve high resolution and wide coverage, large volumes of data are collected from the SAR, and large data sets are produced when the data is processed into images.

The ASF is a National Aeronautics and Space Administration (NASA) sponsored project to collect data from space borne SAR instruments, to process this data into SAR images, and to operate an active archive in support of scientific investigations which use these images for the study of geophysical processes. Currently, the ASF is collecting SAR data from two satellites: the European Space Agency's (ESA) first Remote-Sensing Satellite (ERS-1) and the Japanese space agency's (NASDA) first Earth Resources Satellite (JERS-1). ERS-1 was launched in July of 1991, and JERS-1 was launched in February of 1992. Both satellites have been transmitting SAR data steadily to the ASF since their initial validation period. The ASF will evolve to support ERS-2 which is ESA's second in the series of Remote-Sensing Satellites with a launch in late 1994 or early 1995, and to support the Canadian Space Agency's (CSA) RADARSAT which is scheduled to launch in early 1995. The ASF was designed and built by Jet Propulsion Laboratory and is located at the University of Alaska in Fairbanks (UAF) and is operated by the UAF. The ASF is now one of the Distributed Active Archive Center (DAAC) sites by the Earth Science Data and Information System (ESDIS) project.

## ASF Architecture

The overall functional diagram of the ASF is shown in Figure 1. The ASF consists of four major components [1]: the Receiving Ground Station (RGS), the SAR Processor System (SPS), the Geophysical Processor System (GPS), and the Archive and Operations System (AOS). The RGS functions are to track the satellites with a 10 meter dish, to receive and record the SAR data sent by the satellites on high density recorders for both the local archives and for the

respective flight agencies. The SPS functions are to read and decode the data that has been recorded on high density tapes by the RGS, to process the SAR signal data into SAR image data products, and to deliver these products to the archives. The GPS functions are to create geophysical products such as ice motion vectors, ice classification images, and wave products from the SAR images, and to deliver these products with their metadata to the archives. The AOS functions are to manage the archive, to provide a user interface for searching the catalog of the metadata and for ordering data from the archives, to manage the data distribution, and to provide mission planning capabilities including the ability for a user to request a data acquisition.

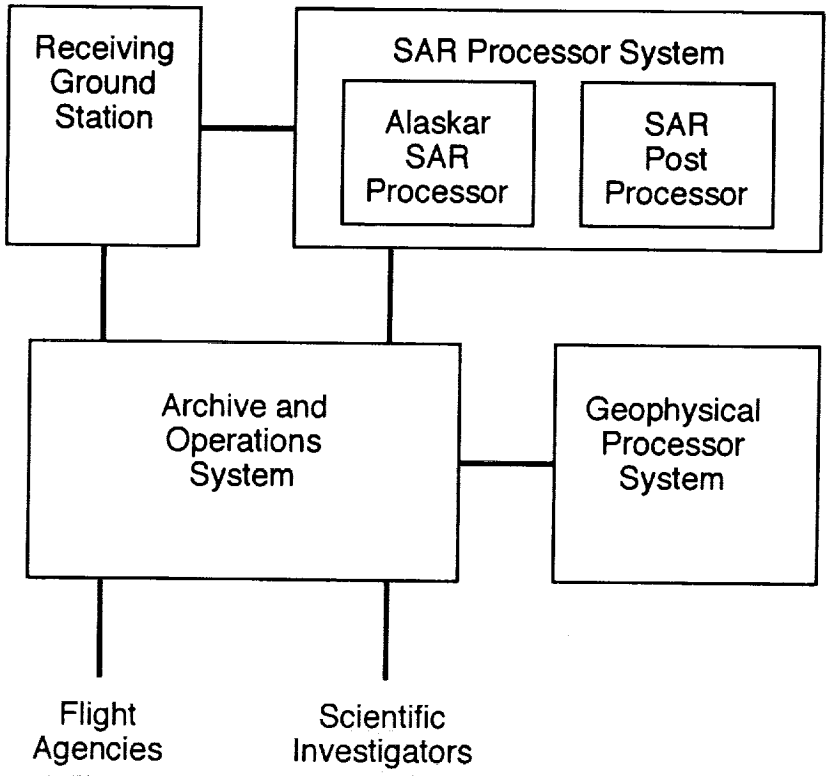



Figure 1: The Alaska SAR Facility Functional Block Diagram.

## Radar Instruments

The ERS-1 satellite generates data at 105 Megabits per second during a data take which can last up to 15 minutes. The ASF receives multiple data takes per day. For 60 minutes of collected SAR data, the volume amounts to 47.3 Gigabytes (105x60x60 Megabits) of the raw data. The JERS-1 satellite has a lower data rate of 65 Megabits per second, but it also has an on-board recorder which can concurrently dump data recorded from anywhere around the earth with the data collected in real time.

For standard processing of ERS-1 data, each minute of data generates about 4.5 images, or 9 images from each 2 minutes of data. Each image is 8kx8k pixels covering an area on the earth of 102.4km x 102.4km with pixel spacing of 12.5 meters in each dimension. One of the standard products derived from each image is a 1kx1k low resolution image generated by performing an 8x8 average of the full resolution image. For each image there are also metadata that describe the image and the processing that produced the image. Each metadata set is about 30 kilobytes (KB) of information. The total storage required for each image is about 68.2 Megabytes (MB), or 20.5 Gigabytes (GB) per 60 minutes of data. For a 1000 day mission, this translates to 36.7 Terabytes of input data and 20.5 terabytes of output data. Mass storage is one

of the major cost drivers in the operation of the ASF, as it will be for all SAR data systems. This has become especially true with computing systems rapidly increasing in power. A few years ago, the cost of high volume and high speed storage systems were secondary to the cost of high performance computing systems. In recent years, the relation has reversed.

## The RGS

The RGS records the telemetry and SAR signal data to many combinations of high density recorders including: two Honeywell HD-96 for data sent to Japan, one Thorn-EMI Digital tape recorder for data sent to ESA and six AMPEX DCRSi recorders for the local archives and for inputs and outputs of the subsystems that operate on the data. Typically at transmission time, the RGS will record two DCRSi tapes, one for the archives and one for the working copy, and one tape for the respective flight agency. Because the JERS-1 has an on-board recorder, it can transmit dual data streams - one from the on-board recorder and the other from the SAR instrument in real-time, so the RGS will record data to both Honeywell recorders simultaneously as well as to two DCRSi drives.

## The SPS

The SPS consists of two subsystems: the Alaska SAR Processor (ASP) and the Post Processor (SPP). The ASP is composed of a custom hardware for reading, decoding, and processing of SAR data from DCRSi tapes and a MASSCOMP computer which controls the custom hardware and the DCRSi tape drives. It uses one DCRSi drive to input SAR data and one to record the output results. The ASP reads ancillary data from the DCRSi to generate processing parameters which are used to set up the custom hardware for processing the SAR data.

The SPP is composed of two computers that work together for specialized input and output and for making the products compatible with VMS-RMS file system. A DEC MicroVAX is used to control all of its functions, while an APTEC IOC-24 is used to perform some of the high data rate functions. Its functions include reproducing high volume data from the DCRSi, receiving averaged image data from the ASP, converting all data into the VMS format for the AOS, and recording image data onto films. The SPP uses a disk system shared with the AOS to quickly exchange large volumes of data. In addition, the SPP uses local storage to keep its copy of metadata and ancillary information.

## The GPS

The GPS, which produces the higher level geophysical products, is implemented on a SUN-4 workstation with a high-speed array processor. It communicates with the ACS via an Ethernet connection. It needs enough local storage for its own database of metadata and for work space to handle input image data and output geophysical products. This space amounts to 3 GB of magnetic disk storage.

## The AOS

The AOS consists of two subsystems: the Archive and Catalog Subsystem (ACS) and the Mission Planning Subsystem (MPS). The ACS maintains the archive and the catalog of the archive, and the ACS provides the user interface so that they can search the catalog, order data from the archives, and request data be acquired from a satellite. ACS is providing the IMS and DADS functions in the ESDIS DAAC model. The MPS provides the mission planner with tools to perform conflict resolution of satellite schedules and to create weekly operations schedule of data acquisition times.

The MPS is implemented on a VAXstation 4000/90 and the ACS is implemented on a VAX 8530 which will soon be augmented with another VAXstation 4000/90 to perform its database operations. The ACS has a large volume (24 Gigabytes) of magnetic disk storage to serve several storage requirements. The largest use is to cache data as it is moved to and from the optical disk jukebox, which stores all of the low resolution data and all of the higher level products in the near-line archive. Other uses for the magnetic disk include staging area for data that is to be transferred to the user electronically, housing of the database for the archive catalog, and providing work space for some of the auxiliary processing. For digital image distribution, the ACS has two 9-track tape drives with multiple density capabilities (800 bpi, 1600 bpi, and 6250 bpi) and two 8-mm tape drives. The ACS can make prints of low resolution images on a Lasertechnics printer. One of the major auxiliary processing that ACS must perform is to transform the images onto a map projection such as Universal Transverse Mercator or Polarstereo projections by using an array processor and a large portion of its 96 Megabytes of memory.

## ASF High Density Storage Systems

The ASF currently uses two types of high density tape recording formats for data storage in the RGS and the SPS. One format used in the Honeywell and EMI high desnity data recorders (HDDRs) is the familiar reel to reel 1 inch wide high density data tape (HDDT) with multiple linear data tracks that are recorded by moving the tape over a fixed head assembly. The other format is the AMPEX DCRSi which uses a rotating head assembly that records data tracks across the width of the tape as it is moved over the heads.

The reel to reel HDDTs were traditionally used for recording high rate data collected from instrumentation and satellite systems. The tapes are one inch wide, and each reel contains approximately 9600 feet, providing approximately 15 minutes of recording time at the maximum data rate. The reels are approximately 14 inches diameter and each reel of tape weighs about 20 pounds. Both the EMI and Honeywell drives are capable of recording data at various rates from a minimum of less than 1 megabit per second to a maximum of about 150 megabits per second on up to 32 data tracks. The drives can record data onto each track with a maximum density of 33 kilobits per inch. With all 32 tracks in use, this provides an aerial density of approximately 1 megabits per square inch. This translates to approximately 12 GB of data capacity per reel. In practice, some of the data tracks are used for recording error checking and correcting codes (ECC), so the effective data density and capacity are somewhat less.

The recorders input data via a serial data line and use a separate clock line that synchronizes the data. The drive electronics track the input clock and set the tape speed and density accordingly to keep up with the data. The recording rates are changed by varying the speed of the tape motion; the drives are capable of moving the tapes at a minimum of 15/16 inch per second (ips) to a maximum of 120 ips, in binary increments. As the tapes require different recording and playback equalizations at each different speed, a different set of equalization and biasing circuits in the recorder are required for each speed of operation. As a new tape speed is selected, a different set of recording electronics, tuned for that speed, is selected. Each set of recording electronics can tolerate small variations of data rates, and the recorder can compensate for some of these variations by slewing the tape speed and changing the data density on the tape. However, large variations of data rates will cause the recorder to select the next higher or lower tape speed, and the associated recording electronics. But because of the size and weight of the reels, the recorders can not change the speed of the tape motion very quickly. Therefor, during recording, the input data rates must remain fairly constant since some data will be lost as the recorder selects a new speed and a different set of recording electronics.

When preparing for a recording, the tape drives must be started at least 15 seconds before the input data becomes available in order to spin up the reels to move the tape at the proper speed. In practice, however, much more lead time is used for the spin-up and pre-roll functions in

order to insure that no data is lost. Similarly, during playback, the tape motion must be started from a point well before where the data begins to allow the reels and tape to reach their correct speeds. Then the data is output by the recorder at a constant rate with a clock signal, and the device receiving the data must be capable of ingesting the data at that rate. Typically, in order to process the data, the receiving device is a general purpose computer system. These can only accept the data at rates far lower than when it was recorded. In addition, the computer most likely could not be performing any other tasks at the same time or it could be diverted from the data receiving task long enough to lose some of the data. Often in practice, when a tape is played back at a much slower speed than when it was recorded, the playback process is not reliable, and is subject to very high error rates.

In order to facilitate location of data on a HDDT, time codes are recorded onto the tape with the data. During playback, a time code decoder is used to find the approximate starting location of the data area of interest. Then, as the recording is played back to the processing computer, it will contain the desired data as well as unneeded information. It is the task of the processing computer to separate these by locating the desired data.

The ASF provides data recording services for ESA and NASDA when their respective satellites are within its station masks using the tape formats that each agency uses. The JERS-1 satellite uses an on-board recorder to store some of the data that it collects. The recorder does not rewind its tape when it dumps the recorded data, so it is dumped in reversed order from when it was collected. The two Honeywell recorders have the ability to play its data in reverse order as well, so the reversed data that was recorded onto it can be played back in the correct order.

The ASF uses the AMPEX DCRSi cassettes for storing its own long term archives of ERS-1 and JERS-1 satellite data and all processed data generated by the SPS by processing the SAR data. The digital cassette format is a relatively new entry in the field of high density data recording. The DCRSi is derived from an AMPEX broadcast video cassette format. The tape itself is 1 inch wide and each cassette contains about 1600 feet, providing approximately 1 hour of recording time at its maximum data rate of 107 megabits per second. This translates to approximately 48 gigabytes of data capacity. The record/playback head assembly consists of 6 heads mounted on a drum which rotates in a direction perpendicular to the direction of tape motion. As the tape is moved across the head assembly, successive "swipes" of data are recorded across the width of the tape. Each block receives its own address during recording, so it can be referenced during playback. The recorder also records a linear track onto the tape containing coarse address markers which aid in location searches.

The DCRSi moves the tape at only one speed, the maximum speed, and records data onto the tape at one density, the maximum density. The recording heads scan across the width of the tape, recording 4356 bytes, or 34848 bits, of user data in each block, with a linear density of 606 blocks per inch. This represents an aerial density of over 21 megabits per square inch. This is effective data density, as actual density is somewhat higher to record ECC data, time codes, and block addresses in addition to the user data.

The DCRSi recorder uses a front-end buffer to catch input data and spool output data. During the recording process, as the buffer becomes half filled with the input data, the tape drive pre-rolls the tape and begins recording the buffer data at its maximum speed. As the buffer empties, the tape motion stops, and the tape is repositioned, ready to start moving again when the buffer becomes half full. The tape motion is simple and quick enough that it can be accomplished before the buffer fills up. During playback, the drive first plays data into the buffer. If the destination device cannot accept the data fast enough, the buffer will fill up, and the tape will be stopped and repositioned. As the buffer reaches half empty, the tape is pre-rolled and started again, filling the buffer. This allows the external data rate to vary continuously between 0 and the maximum rate during recording or playback with no restrictions on how often the rate varies. In addition, there is no need to start recording on the tape before the input data becomes available, as there is with the HDDTs. The recorder can respond to incoming data and outgoing data requests with almost no latency. This allows the data to be recorded at, and later to be

retrieved from, exactly the specified location on the tape. This also has the benefit of helping to make more efficient use of the tape.

The flexible data flow characteristic of the DCRSi is established once the recorder is put into the proper recording or reproducing mode. This step requires that the recorder move the tape to the requested block, which could require up to two minutes, depending on what position the tape was in initially. But once the initial seek is completed, the recorder response to incoming or outgoing data is nearly instantaneous. Effectively, the DCRSi behaves like a 48 gigabyte disk drive with long seek times, fast transfer rates, but no latency times.

The versatility of the DCRSi makes it much easier to use than the reel to reel HDDRs. Its capabilities have made it the only recorders used in the operational data processing in the SPS. The DCRSi cassette itself is capable of storing about four times as much data as a full reel of HDDT, but it requires approximately one-fourth the physical volume for storage. This makes it a much better medium for long term archives in the ASF. The one function that the DCRSi cannot perform for the ASF is to play its data in reverse, as the linear track recorders can. So in order to process the JERS-1 data collected from a dump from the on-board recorders, the data must first be dubbed onto a DCRSi cassette from a Honeywell HDDR playing in reverse.

The DCRSi does share some inconvenient features with the older HDDRs. Because of the high data rates of these types of recorders, they cannot be easily interfaced to a general purpose computer for access of their data. The DCRSi currently requires a customized interface for communicating with any other device. On the RGS, the standard serial data and clock lines adequately communicates with the drives. On the ASP, a custom interface was designed in-house to make use of the DCRSi's parallel interface. On the SPP, a special interface was designed by AMPEX to communicate with the APTEC IOC-24, also using the parallel interface.

The ASF is currently investigating a high speed SCSI interface for the DCRSi. It consists of a DCRSi interface installed into the VME chassis of a small computer system based on a SUN SPARC processor. The system also includes a fast SCSI interface for interfacing to general purpose computers. This will enable the DCRSi to be attached to most computer systems with a fast SCSI interface.

The ASF currently operates 6 DCRSi drives. The RGS requires at minimum 1 DCRSi drive, although the normal operation is to record on two drives at down link time to concurrently make both the archive and working copy. The SPS requires three DCRSi drives, one for input and one for output during SAR image processing and one for playback of the processed data. The sixth drive is kept for a working spare which allows quick change when a drive fails. A parallel switch helps to easily configure which drive is connected to which port of a specific computer.

To store all of the data on-line would be prohibitive and the archive strategy to this date is to store all large volume data sets on high density recorders in the off-line manner. The location (tape identity and addresses on the tape) and other key information are kept in an on-line data base in the ACS.

The ACS uses a jukebox with a capacity of 89 platters of Write-Once-Read-Many (WORM) laser disks for storage of low resolution data. With a 2 GB capacity per platter, the juke box has a capacity of 178 GB. The jukebox contains two drives each capable of addressing only one side of the platter at a time. The robotic mechanism in the jukebox moves the platters between their storage bins and the drives, and positions the selected side of the platter in the drive. To prevent excessive robotic action, the input data is cached on magnetic disks on the VAX, and periodically flushed to the jukebox. Since the disk platters are normally stored in their bins, the initial access time to each platter is approximately 30 seconds, the time required by the robotic mechanism to retrieve the platter, insert it into the drive, and spin it up to speed. Then, the optical disk drive behaves like a slow magnetic disk drive; they are capable of transferring data at 250 kilobytes per second, with seek times on the order of tens of milliseconds.

The ACS has a total of 24 GB of magnetic storage to provide storage space for caching of data to the archive, to provide storage space for the catalog and the needed work space for the database system, to provide working space for geocoding of images, to provide cache space for staging of data to users and to the GPS, and to provide work space to all users. It uses 6 GB of magnetic disks for caching data for the optical disk jukebox. In addition, it shares 8 GB of disk storage with the SPP for buffering of data exchanged in between the two systems.

Initially, the ACS used physically large disk drives connected to the VAX through VAXBI controllers using a proprietary DEC interface. These drives and controllers were costly, not very reliable, and consumed large amounts of physical space and power. The ACS has now been upgraded to new disk drive technologies. In the last two years, disk manufacturers have released a number of technologically advanced disk drives into the market. These drives are physically smaller, transfer data at higher rates than older drives, and are far less expensive in terms of cost per unit of storage. These drives also use a new version of the Small Computer System Interconnect (SCSI-2) interface standard for connection to the host computers. This open standard interface allows any of these drives to be connected to any host computer with a SCSI adapter.

The SPP uses SCSI-2 disk drives almost exclusively for its operations. Its 8 GB of shared disks are composed of 4 SCSI drives which have dual host attachments. The SPP writes the data it produces onto the shared disks for the ACS to retrieve, and the ACS writes processing requests and data onto the disks for the SPP to read. The two systems are using the disks in a static dual-port manner; one disk is configured to allow only the ACS to write to and the SPP to read from while the rest are configured to allow the SPP to write to and the ACS to read from. The VAX clustering feature on the VMS operating systems on both the MicroVAX and the 8530 can be enabled to perform full active dual-porting of the shard disks. However, it was found that the overhead of VAXclustering was too great, and that the current configuration is more than adequate. The SPP also keeps its own repository of all full resolution metadata on its local storage of 4 GB of SCSI disk drives.

The ASP MASSCOMP currently uses disk drives with the old Storage Module Device (SMD) interface. There are a total of 1 GB of storage used for operations. The ASP is in the process of replacing the MASSCOMP with one of the newer workstation based systems which will be much faster, easier to maintain, and will also use the more efficient SCSI disk drives for storage.

## Data Distribution

The ASF is responsible for distribution of data on various media to the scientific users. Currently, the ASF delivers data on the following media: nine track magnetic tape at 1600 bytes per inch (bpi) and 6250 bpi, 8 mm tape at 2.3 gigabytes or 5.0 gigabytes per tape, file transfer of s all files, and hard copies of images on low or high resolution films. The data is stored and distributed in a format that is in compliance with the specifications of the Committee on Earth Observations Satellites (CEOS) [2], to which ESA, NASDA, NASA, and CSA have agreed as the format of data exchange. The data that is put on tape also have a CEOS volume description that identifies the contents of the tape(s). The CEOS specifications allow for leader and trailer files which describe the details of the data, the processing, the sensor, and the satellite. The hardcopy and film products also are in compliance with the guidelines set by the CEOS specifications.

The digital data distributed on the various recording media include low and high resolution multi-look detected image data, high resolution single-look complex data, and reformatted SAR signal data. Initially, the ASF supported only the 9 track magnetic tape as its sole form of data distribution media. These tapes are usually packed with 3600 feet per reel, providing up to 270 MB of storage in 6250 bpi and up to 69 MB at 1600 bpi. In the higher density, this provides the ability to record about 80 low resolution detected images, two high resolution detected

images, or one complex image. The reformatted SAR signal data files need to be spread across two reels, or volumes of tapes.

More recently, the ASF began distributing data on the 2.3 and 5 GB 8 mm cassette tapes. Because of its high capacity, a single 8 mm cassette can usually store all the data requested by any user. It requires less operator assistance since it is rare to need to change tapes to make more than one volume for any request. The media is also much smaller and lighter than the 9 track tape, making it easier to handle store, and ship. In addition, all of the 8 mm tape drives available are produced by the Exabyte Corporation, which has created a de-facto standard in the 8 mm format. All drives also use the SCSI standard interface, making it readily usable by most computing platforms.

The 8 mm tape format does have some drawbacks compared to the old 9 track tapes. First, its drive mechanism is much slower than those of the 9 track machines. Typically, any given tape motion on the 8 mm drive can take up to an order of magnitude longer to complete than the 9 track drives. Then, the data transfer rates of the 8 mm drives are about half as fast as the 9 track drives. Finally, the one operation that is the most time consuming on the 8 mm drives is the creation of a new tape file. The Exabyte tape format uses a very large header for each file. Each header can be several MB in length, and requires on the order of tens of seconds to create. Using the example of a CEOS formatted tape, each product contains a leader file, the data file, and a trailer file, each of which would require a physical file header to be created by the drive. In addition, the CEOS tape also has a volume directory file, and a null descriptor file. So the overhead of creating a CEOS formatted 8 mm tape usually requires more time than transferring the actual data. In contrast, file headers are very efficient on the 9 track drives, and are usually executed very quickly.

The hard copies of image data are distributed on two types of films. The low resolution data are recorded onto dry silver films using a LASERTECHNICS 300D printer. The high resolution data are recorded onto Kodak photographic films using a ColorFIRE 240 film recorder made by MacDonald Dettwiler and Associates (MDA), now Cymbolic Sciences Incorporated (CSI).

The 300D printer takes 8 bit image pixels as its input. The data is used to modulate a Bragg crystal which in turn modulates the intensity of a laser beam that exposes the pixels on the dry silver film. The modulated beam is scanned across the width of the film by an oscillating mirror. The film is developed by heat rollers within the printer, providing a nearly instant hardcopy on the image. The printer is capable of writing up to 2048 pixels across a 8.5 inch wide film, and up to 1500 lines of pixels. The transparency film provides 128 effective gray shades, or 7 bits of dynamic resolution. The paper film provides 64 shades, or 6 bits of resolution.

The ColorFIRE 240 is capable of making use of full 24 bit color (8 bits red, 8 bits green, 8 bits blue) image data to generate color films. However, the SAR image data that the ASF currently produces has only one channel at 8 bits per pixel, so only black and white films are being generated from existing data. The recorder is capable of writing up to 8800 pixels across the 240 mm (9.44 inch) film, and up to 9600 lines of pixels. It also uses a Bragg crystal to modulate the intensity of a beam of light which exposes the film. The beam is directed across the width of the film by a rotating mirror riding on air bearings. This allows the printed image to have extremely accurate geometric fidelity. The ColorFIRE 240 in the ASF is set up to also provide very accurate radiometric accuracy while maintaining the full 8 bits of dynamic resolution.

## RADARSAT Support

The ASF facility was originally sized to collect and process 5 minutes per day of ERS-1 data, 10 minutes per day of JERS-1 data, and 30 minutes per day of RADARSAT data. During the first year of ERS-1 operations, ASF collected on the average more than 30 minutes of data per day and for JERS-1 the volume has also exceed the original sizing. For RADARSAT, the system is now being specified with a requirement of 80 minutes of processed data from 120 minutes of

collected data. This translates roughly to 5 terabytes of processed and 33 terabytes of raw data per year. The remaining unprocessed data is to be collected for other stations who will be responsible for processing their own data.

Currently the ASP processor at ASF is being upgraded to have the ability to process 60 minutes of data per day. For the RADARSAT, additional processors will be delivered to ASF so that the processing throughput will be increase significantly to keep up with the data volume. The storage capacity is quickly being consumed even at the processing rate of 40 minutes per day. The storage of on-line data will exceed capacity the current archive system within 6 months, and ASF is investigating different technologies and archive strategies which can alleviate the expected overflow in the near future, but the studies are looking also at the big picture of the many years of data to come and also at the ESDIS schemes to handle storage with their DAAC contract.

## Conclusion

SAR data is a very high volume form of data which requires a processing and archive facility which can accommodate multiple terabytes of input and output data per year. The data distribution also must be able to handle large volumes of data to a very diverse community of users whose requirements on the data are almost unique on a user by user basis. The ASF has been able to meet the demands for the large volumes of data by employing not only a variety of storage media but an archive strategy that tries to keep the more frequently accessed and smaller data sets either on-line or near-line while the less frequently accessed and larger data set are kept in the off-line, high volume storage media. Granted, the system is tasked rather heavily, but it has exceeded its original requirements and will be able to grow and evolve with the ever increasing requirements of future.

## Acknowledgments

This work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## References

[1] R. Berwin, D. Cuddy, J. Hilland, and B. Holt, "Design, Test, and Applications of the Alaska SAR Facility," Space Technology, Vol. 12, PP 91-104, (1992).

[2] CEOS WGD on SAR Data Standards, "SAR Computer Compatible Tape Format Specifications", Rev 2.1, January 1, 1992.

# Value Added Data Archiving

**Peter R. Berard**

Battelle Pacific Northwest Laboratory
P.O. Box 999, MS K1-87
Richland, WA 99352
(509) 375-6591
(509) 375-6631 (fax)
pr_berard@pnl.gov

## Abstract

Researchers in the Molecular Sciences Research Center (MSRC) of Pacific Northwest Laboratory (PNL) currently generate massive amounts of scientific data. The amount of data that will need to be managed by the turn of the century is expected to increase significantly. Automated tools that support the management, maintenance, and sharing of this data are minimal. Researchers typically manage their own data by physically moving datasets to and from long term storage devices and recording a dataset's historical information in a laboratory notebook. Even though it is not the most efficient use of resources, researchers have tolerated the process.

The solution to this problem will evolve over the next three years in three phases. PNL plans to add sophistication to existing multilevel file system (MLFS) software by integrating it with an object database management system (ODBMS). The first phase in this evolution is currently underway. A prototype system of limited scale is being used to gather information that will feed into the next two phases. This paper describes the prototype system, identifies the successes and problems/complications experienced to date, and outlines PNL's long term goals and objectives in providing a permanent solution.

## Introduction

Researchers in the Molecular Sciences Research Center (MSRC) of Pacific Northwest Laboratory (PNL) spend a considerable portion of their time on the encumbering task associated with managing their scientific data. Automated tools that support the management, maintenance, and sharing of this data are minimal. Researchers typically manage their data by physically moving datasets to and from long term storage devices and recording a dataset's historical information in a laboratory notebook. While this process has been tolerated, it is not acceptable for managing the amount of data researchers will be generating in the near future.

The Environmental and Molecular Sciences Laboratory (EMSL) is currently under development at PNL for the U.S. Department of Energy (DOE). The goal of this construction project is to field a fully functional, equipped, and staffed research facility in early 1997. The EMSL will be operated by PNL as a DOE Collaborative Research Facility open to scientists and engineers from the academic community, industry, and other government laboratories for collaborative research in the molecular and environmental sciences. Major facilities within the EMSL include the Molecular Sciences Computing Facility (MSCF), a laser/surface dynamics laboratory, a high-field nuclear magnetic resonance laboratory, and a mass spectrometry laboratory. The MSCF will consist of the High Performance Computer System (a massively parallel processor), the DataBase Computer System (described below), and the Graphics and Visualization Laboratory.

With the development of EMSL, it is anticipated that by the turn of the century, data to be archived annually will be on the order of seven terabytes. The size of individual datasets will reach tens of gigabytes and the total amount of data each researcher will manage is expected to

increase significantly. Manually managing this scientific data and maintaining historical information about individual datasets will prove to be cumbersome, if not impossible. The need for high-speed, large-scale data transfer and long-term storage and retrieval of scientific data is critical to the MSCF. Large data streams will be produced by multiple computational experiments and instruments. The data archival and retrieval required to support the post-processing for these experiments and instruments is the primary driver for the high performance DataBase Computer System (DBCS).

Given these observations, researchers have two basic needs: 1) a data archiving facility that allows immediate access to any given datasets and 2) an automated means by which to maintain and access historical information about individual datasets. The solution to this problem will evolve over the next three years in three phases. As part of the MSCF, the DBCS will be used for scientific data management. PNL plans to add sophistication to existing multilevel file system (MLFS) software by integrating it with an object database management system (ODBMS) (i.e., value added data archiving). The goal is for DBCS to provide researchers with a completely automated facility in which both datasets and the associated historical information will be electronically accessible. Each phase of DBCS will be implemented on increasingly more sophisticated and powerful hardware architectures. The first phase in the evolution of DBCS is currently underway. A prototype system of limited scale is being used to gather information that will feed into the next two phases.

The sections that follow provide an overview of DBCS and describe the activities associated with each of the three phases of the DBCS development.

## Database Computer System (DBCS)

DBCS will be a scientific information management "instrument." DBCS will provide data archival services over a backbone network connecting most offices, users, workstations, and servers. In addition, data archival services will be provided for very high bandwidth data transfers using a High Performance Parallel Interface (HIPPI) based high speed network. Research scientists using the EMSL will access theses services via a graphical user interface. Behind the scenes, an ODBMS will be integrated with MLFS software to provide a sophisticated data archiving package for managing scientific data (i.e., datasets) and files. DBCS will be rich in methods to store, manage, and effectively search and browse information about datasets that are part of the file system.

The MLFS will provide virtually infinite file size and file system size. This is made possible by automatically moving or *migrating* files up and down a hierarchy of successively faster but lower-capacity storage devices (levels). High speed storage will be provided by a Redundant Array of Inexpensive Disks (RAID). Medium and low speed storage will be provided by "robotic removable" media devices/robots. The IEEE Storage Systems Standards Working Group is in the process of developing the IEEE Mass Storage System Reference Model [1] that will eventually result in a set of standards for mass storage system software. It is important for the long term viability of DBCS that the MLFS be based on the Reference Model. This requirement ensures that future upgrades of one or more components of the MLFS software will be feasible due to the industry standard interfaces between components.

The ODBMS component will provide persistent storage of information about datasets and files in the MLFS. This information, often referred to as *metadata*, will allow associative access to datasets and files by information other than filename. In addition, the ODBMS will provide sophisticated and extensible querying facilities, support for versioning, views, and additional security.

DBCS will be implemented in three distinct phases:

1. DBCS-0 prototype system

2. DBCS-0

3. DBCS-1

The DBCS-0 prototype is a system of limited scale and is described in detail below. DBCS-0 will be an interim system that will provide a hardware and software platform on which to develop a scientific database application, as well as tools for users and application developers. The final production system, DBCS-1, will be acquired and implemented in the third phase.

## Phase I: DBCS Prototype System

The DBCS prototype system includes an ODBMS, MLFS, and minimal supporting hardware. This system is being used to gain hands-on experience and knowledge with these types of products. The prototype system's hardware includes a host computer system, SCSI disks, and an 8-mm tape robot archive. The National Storage Laboratory's version of UniTree (NSL UniTree) (hereafter referred to as UniTree) is used as the MLFS software and ObjectStore is the ODBMS.

The host computer system is an IBM RS/6000 980 POWERserver running version 3.2.3e of the AIX operating system and has 128 megabytes of memory, one 970 megabyte and two 1.37 gigabyte internal SCSI disk drives. A disk farm consisting of four Seagate SCSI-2 disk drives (connected to a SCSI-2 I/O controller) providing a total of 8 gigabyte of formatted disk space is also part of this system. However, none of the SCSI-2 disk space is being used for UniTree support. Instead, this disk space is used for the Andrew File Server (AFS) in support of other EMSL software development efforts. A total of 2 gigabytes of the internal SCSI disk drives serves as UniTree's disk cache. While this is a minimally sized disk cache, it serves the purpose for the near term prototyping efforts. A Comtec ATL-8, Model 54 8mm tape robot archive supports UniTree's long term storage needs. This robot contains two EXABYTE 8500 5gigabyte disk drives and is capable of holding 54 tape cartridges in its carousel. The host computer system is connected to other workstations through a Fiber Distributed Data Interface (FDDI) network.

An NSL UniTree license to manage 250 gigabyte of data has been purchased for the prototype system. For this initial prototype, three users have been identified for MLFS support. Many of their files are in the rangé of 50 to 250 megabytes in size, with the largest file size approaching 500 megabyte. It is expected that files produced by these users will reach 1 to 2 gigabytes within the next year.

### Current Status

Due to several unfortunate events, the delivery and installation of the prototype system is behind schedule at the time of this writing. The original schedule called for delivery and installation of the system by February 1, 1993, with an additional 30 days scheduled for a series of acceptance tests. Thus, the system was to be available for general use beginning the first part of March 1993. In addition to problems encountered during system installation, the acceptance tests identified several problems that required resolution. Consequently, progress towards developing an intelligent data archiving system for DBCS has been severely impacted. In any event, the anticipated near-term work required to implement value added data archiving is described below.

### Near Term Direction

It is highly desirable that researchers using DBCS have a user-friendly interface by which to access their data in UniTree. The first logical step in achieving this is to develop a layer of software that minimizes the need for users to become familiar with NSL UniTree. In the

prototype system, several scripts will be provided to shield the user from the implementation details of UniTree. These scripts will mimic standard UNIX commands (e.g., **utcp** and **utmv** for moving files to and from UniTree, **utrm** for removing files in UniTree, **utls** for listing files that are in UniTree, etc.). Where appropriate, the scripts will query the user for the file's metadata and store this information into the ObjectStore database. Lessons learned from this initial implementation will feed into future, more robust versions of DBCS.

Perhaps the ultimate mass storage solution is to provide users with what appears to be a virtual file system or file "space" in which mass storage services are performed automatically. One such implementation can be found at the Pittsburgh Supercomputing Center, where AFS has been successfully extended to provide mass storage support and multiple copies of data to users [2, 3]. In the prototype system, this concept will be tested by attempting to implement a file space in which a user's UniTree files are accessed as local files. In reality, this may be as simple as using scripts or C programs that automatically shuttle files to and from a specific directory under the user's home account (using anonymous FTP). Alternatively, it may be possible to use existing tools such as Alex [4]. In Alex, anonymous FTP sessions are disguised as a pseudo-file system that allows users to access FTP sites as a local file system. Since Alex is freeware, it is unlikely that any implementation using this product will provide a long term solution for DBCS.

## Problems Encountered and Lessons Learned

Buying an off-the-shelf integrated mass storage system that will meet the needs of the EMSL is simply not possible. Integrating hardware and software to support mass storage needs is a challenging task and requires considerable resources and talent. Like many systems' integration efforts, the time required to implement such a system can easily and grossly be underestimated due to unforeseen difficulties. Integrating this seemingly simplistic mass storage system for the DBCS prototype proved to be such a challenge. PNL is fortunate to be in a situation where time has been appropriated for experimentation. While installing the prototype system resulted in a rather significant schedule slip that was not budgeted for, the lessons learned will prove to be a valuable asset in the next two phases of DBCS and will be carried forward into future procurements. Likewise, it is intended that these lessons be disseminated to others who are undertaking similar efforts.

Perhaps the most significant negative impact was due to the fact that no attempt was made to integrate this system prior to delivery to PNL. The prototype system was perceived as a simple implementation compared to other systems in use today. The fact that the prototype system's hardware is virtually identical to that used by the NSL UniTree development team indicated that the system could be integrated on site. However, certain acceptance tests on the prototype system identified problems that required an update from version 3.2.2 of the AIX O/S to version 3.2.3e, as well as several Program Temporary Fixes. This resulted in distinct differences in the versions of the AIX O/S used by PNL and NSL, and caused severe complications in integrating NSL UniTree. In addition, PNL's tests identified several unknown bugs in UniTree.

The extent of the bugs detected indicate that more extensive testing is required prior to releasing future versions of the software. While PNL does not perceive NSL UniTree to be a mature product at this time, the advanced features it offers for mass storage systems, including third party transfer and support for multiple dynamic hierarchies, certainly warrant it as a worthy candidate for investigation. In any case, the efforts spent on integrating NSL UniTree into this prototype system have not been wasted. In order to avoid these types of problems in future procurements for DBCS, the integrator who is awarded the contract will be required to submit a detailed integration plan, integrate the system prior to delivery, and demonstrate the system by successfully running a predefined set of acceptance tests.

## Phase 2: Database Computer System - Level 0

The DBCS prototype system is implementing one hierarchy of storage devices (SCSI disks and an 8-mm tape robot archive). Phase 2 in the evolution of the DBCS, DBCS-0, will expand the capabilities of the prototype system by adding a second hierarchy of more powerful devices (RAID disks and a fast tape robot archive). NSL UniTree will support these multiple dynamic hierarchies of storage devices.

The files and datasets to be stored in DBCS-0 will range in size from less than 1 megabyte to multiple gigabytes. DBCS-0 will efficiently support data archiving of this vast range of file sizes by providing multiple hierarchies of storage devices. The smaller files will be assigned to the hierarchy containing the slow devices (i.e., SCSI disks and 8mm tape robot archive) and the larger files will be assigned to the hierarchy containing the faster devices (i.e., RAID disks and the fast tape robot archive). Determining the optimal match of file size-to-hierarchy will be achieved by experimentation. A RAID Level 3 configuration will result in maximized data throughput and efficient management of large files. It is possible to manage small files on the RAID disks under this configuration by utilizing a log-structured file system [5, 6], that groups the small files into a segmented log. The advantages and disadvantages of using this approach in DBCS-0 have not yet been identified, but will be investigated.

Each hierarchy will consist of two levels of successively faster but lower-capacity storage devices (Figure 1). The "first level" within each hierarchy will consist of disk drives that provide relatively high-speed storage. The "second level" within each hierarchy will consist of one or more tape robot archive machines which provides relatively medium/low speed storage. The "first hierarchy" (i.e., Hierarchy 1) will consist of RAID disks and one or more high capacity tape robots for long term storage of datasets and files. The "second hierarchy" (i.e., Hierarchy 2) will consist of SCSI disk storage and an 8-mm tape robot, both of which are directly connected to the host computer system. Among other things, Hierarchy 2 will be used for intermediate storage of datasets and files. UniTree, the ODBMS, and the 8-mm tape robot, which are part of the prototype system, will be transferred to the DBCS-0 archive computer system.

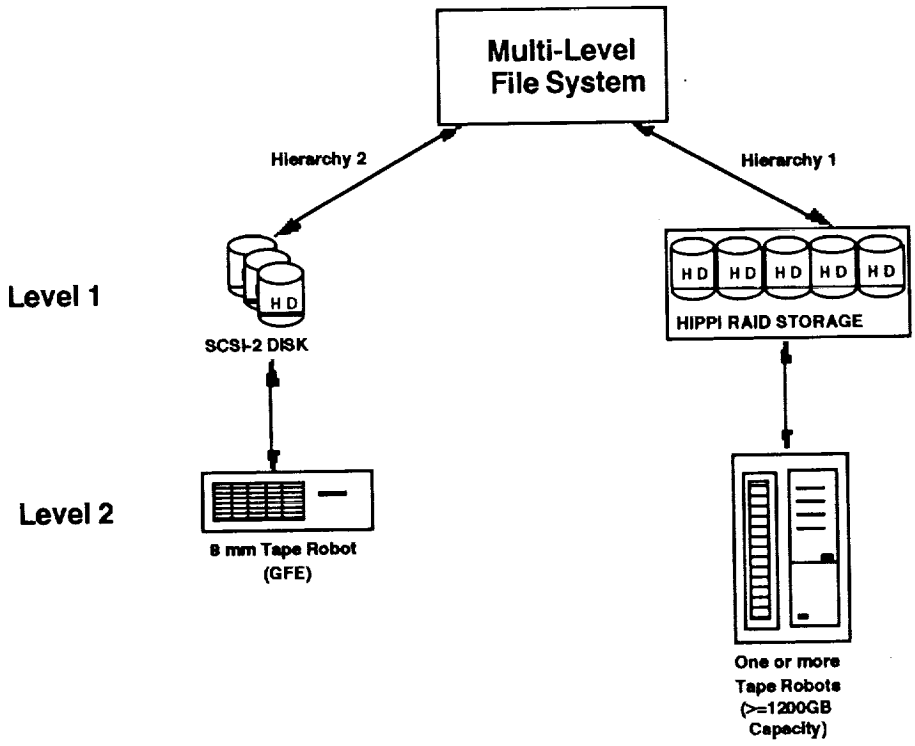



Figure 1: DBCS-0 Levels and Hierarchies

DBCS-0 will support a variety of client platforms, including the High Performance Computer System (HPCS) (a massively parallel processor), High Performance Graphics stations, workstations, etc. (Figure 2). Those clients requiring high speed data transfer (e.g., HPCS, High Performance Graphics stations) will be integrated with DBCS-0 through a HIPPI based high speed network. A slower speed backbone network (i.e., FDDI/Ethernet) will be provided for client platforms that only require medium speed data transfer. The DBCS-0 computer system will be devoted to managing the datasets and files on these client platforms and will be reconfigurable as required (both software and hardware). Third party transfer, as implemented by NSL UniTree, will be utilized for data transfers from HIPPI connected clients. This implementation of third party transfer passes control packets to the host computer system over the slower speed network and passes the data packets over the HIPPI network. This should prove to be a very efficient way in which to move data between clients and DBCS-0.

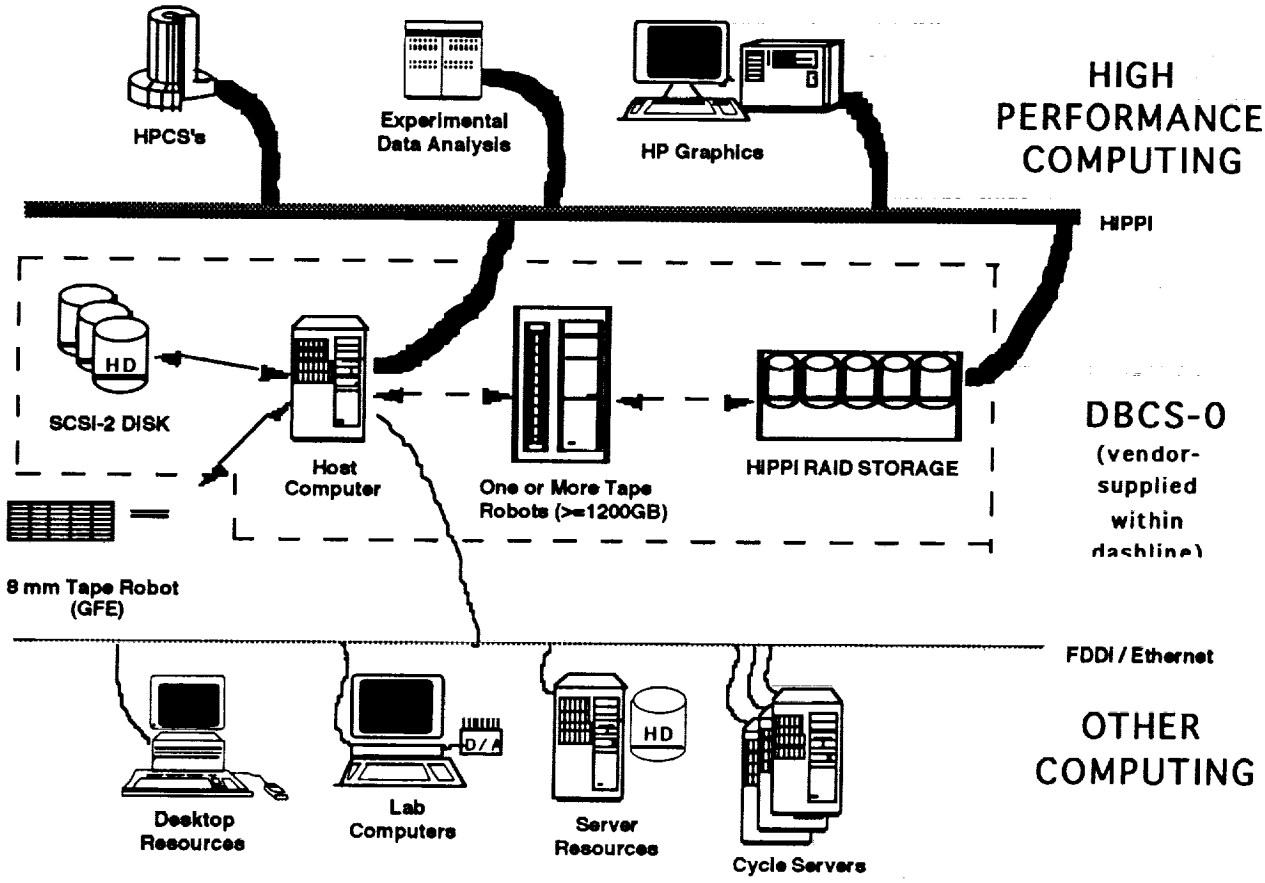



Figure 2: DBCS-0 Connectivity

One of the greatest challenges anticipated in implementing DBCS-0 will be supporting a massively parallel processor (MPP) such as HPCS. The problems associated with supporting the mass storage needs of an MPP are well documented in [7]. NSL UniTree in its present form will not be able to support an MPP because it only supports one logical stream of data. A new breed of MLFS software that is capable of supporting scalable, parallel storage systems is required. While there are no such products readily available today, any efforts in developing a reasonable solution will be closely tracked.

## Current Status

A Request For Proposal for DBCS-0 has been prepared and distributed to companies interested in bidding on providing a solution. Hardware procured for DBCS-0 will include a host computer system with 256 megabytes of memory and 8 gigabytes of SCSI-2 disk space, a HIPPI connected RAID with at least 30 gigabytes of usable disk space and a minimum sustained data transfer rate of 40 megabyte/second, and at least 1200 gigabytes of storage on one or more fast tape robot archives. The fast tape robot archive(s) will have at least four tape drives, each capable a sustained data transfer rate of 2 megabyte/second. The NSL UniTree license procured for the prototype system will be upgraded to support the desired amount of storage and the ObjectStore license will be transferred to the new host computer system. The integrated DBCS-0 system is expected to be delivered in October 1993 and acceptance tests will begin thereafter.

## Future Direction

Data that is currently generated in the MSRC is manually managed and maintained by researchers. Pertinent historical information about this data is usually recorded in a laboratory notebook. When disk space shortage mandates, data is manually archived to tape for long term storage via standard backup procedures. These tapes are then physically maintained by the researcher. The degree of reliability in this process is directly dependent on the researcher's ability to maintain and coordinate the tapes and notebook. Over time, inventory control of data in long term storage becomes a time consuming task.

It is expected that researchers will initially be reluctant to relinquish control of their data and its associated historical information (i.e., metadata) to an automated facility such as DBCS-0. Researchers feel secure in having the ability to control the physical media and laboratory notebook at all times. Developing a similar level of trust in an automated system must be achieved in order for DBCS-0 to be successful. Extensive, but fair acceptance tests for DBCS-0 will ensure a high degree of reliability in all hardware and software components. Lessons learned in testing the DBCS prototype system will be used in developing the scripts and procedures for the DBCS-0 acceptance tests. Likewise, all procedures and software developed for DBCS-0 will undergo a rigid set of predefined tests that will ensure all files/datasets and metadata are maintained in a highly reliable manner. Recovery from failures must be handled gracefully.

Like any other system, DBCS-0 will require a certain level of administration. The individual(s) responsible for administering DBCS-0 must be intimately familiar with each hardware and software component. Administration policies will be written to ensure that users' data is maintained in the most reliable manner. Researchers will be polled for their concerns and this input will be incorporated into the administration policies. Regular and routine maintenance of the MLFS and ODBMS will be performed to ensure recovery in the event of system failures. Once developed, the policies will be automated to the extent possible.

The administration policies for DBCS-0 will also account for management of media used for long term storage. To reduce the amount of human intervention in administrating DBCS-0, it is necessary to provide a maximum amount of storage capacity within the robot archive(s) at all times. While redundant copies of datasets/files will be allowed, only the media containing the primary copy of files/datasets will typically reside in the robot for an extended period of time. Media containing secondary copies of files/datasets will periodically be removed from the robot along with the necessary backup of the MLFS software's databases. The number of times each individual media is used will be automatically tracked in DBCS-0. When any media has outlived its expected lifetime, all of the files and datasets it houses will be transferred to new media and the old media will be destroyed. All information supporting the media management policies will be stored in the ODBMS and will be readily available for the DBCS-0 administrator.

Future implementations of DBCS-0 will provide researchers with a Graphical User Interface (GUI) for managing and manipulating files. This interface will allow users to move files and datasets to and from the mass storage system, enter the file/dataset's metadata, and search and browse the metadata for any publicly accessible files/datasets. Researchers will be able to access files and datasets within DBCS-0 by querying the ODBMS for attributes of the data (e.g., all datasets on a "class" of molecule). These features will save a researcher considerable time compared to the present day practice of keeping notes in one's laboratory notebook and will allow researchers to readily exchange information, thereby increasing their overall productivity and effectiveness. While some types of metadata will be entered by the user via the DBCS-0 GUI, it is important that the user not be required to enter any metadata that can be collected automatically (e.g. user's name, date, etc.).

Restricted access to files/datasets and metadata will be supported. Some metadata maintained by researchers are private notes and are not intended to be shared with others. Also, some datasets/files produced or collected may be immature or meant to be used only on an interim basis. This type of information will be protected from unauthorized access by allowing each researcher to determine what data are to be shared with others. Access to a researcher's data may be granted on a user and/or group basis. In order to support this requirement, the underlying MLFS software must support restricted access to data which has been archived.

A toolkit will be provided that will allow developers of scientific applications to manage and manipulate files and datasets from within their code. Access to the advanced features and local extensions of the MLFS software will be provided within this toolkit. Likewise, information in the ODBMS will be accessible through this toolkit. This *back door* into DBCS will provide application developers the means to search the ODBMS for the required information and stage and migrate files/datasets within the mass storage system.

Supporting the features described above requires that the MLFS software and the ODBMS be integrated. A considerable amount of analysis and experimentation is required before attempting this integration. Much work has been done in an attempt to provide extended capabilities and intelligence to existing MLFS software. As an example of one such effort, Isaac describes a prototype in which the UniTree is integrated with a DBMS [8]. A standard Structured Query Language interface is provided for accessing data in the file system. Data are automatically staged from the file system to the DBMS. As Isaac points out, utilizing the Bitfile Identifier found in UniTree's Name Server database for referencing datasets warrants further investigation. This use of a Bitfile Identifier will likely prove an efficient way in which to access files/datasets within DBCS-0. Also, expanding Isaac's concept to include maintaining metadata about each dataset in the DBMS is consistent with the requirements for DBCS and worth investigation. PNL plans to leverage efforts such as these when integrating the MLFS software with the ODBMS.

## Phase 3: Database Computer System - Level 1

The third and final phase will yield a production mass storage system (DBCS-1) for the EMSL. The planned November 1995 delivery of DBCS-1 will result in a fully operational system in March 1996. Development and enhancement of DBCS-1 is planned to continue after the system is put into operation. Currently, it is not assumed that DBCS-1 will simply be an extension of DBCS-0. While some components of DBCS-0 may be reused in DBCS-1, emerging technology may dictate that DBCS-1 be an entirely new system. It is imperative that evolving technology in both hardware and software be tracked closely. The specifications for DBCS-1 will be prepared based on information collected during the first two phases, input solicited from EMSL users, recent trends and developments in the mass storage community, etc. It is important that the communication channels with others undertaking similar activities, as well as vendors of mass storage systems/products, be open and active at all times. Future papers will describe this phase in more detail.

## Conclusions

Intelligent data archiving services will be provided to researchers in the EMSL in 1996 in the DBCS-1 system. MLFS software will be integrated with an ODBMS to provide researchers with a convenient and efficient way in which to manage their files/datasets and electronically maintain its associated historical information. Emerging technology and information gathered in two previous phases of development will drive the specifications for DBCS-1.

The first phase is currently under way. A DBCS prototype system of limited scale is being used to gain hands-on experience and knowledge of NSL UniTree and ObjectStore software. The second phase, DBCS-0, will expand the capabilities of the prototype system to include more powerful storage devices in multiple dynamic hierarchies later in 1993. DBCS-0 will be faced with the challenge of supporting a variety of clients, including a massively parallel processor.

## Acknowledgments

The work described in this paper is based on the efforts of several people at PNL. I wish to acknowledge my colleagues for their input, support, and previous work in this area.

Pacific Northwest Laboratory is operated for the U.S. Department of Energy by Battelle Memorial Institute under contract DE-AC06-76RLO 1830.

The National Storage Laboratory is a collaborative effort of various industry partners and DOE Laboratories at Lawrence Livermore National Laboratory (LLNL). LLNL is operated for the U.S. Department of Energy under contract W-7405-Eng-48.



## Bibliography

1. S. Coleman and S. Miller, *Mass Storage System Reference Model Version 4*, IEEE Technical Committee on Mass Storage Systems and Technology, May 1990.

2. D. Nydick, K. Benninger, B. Bosley, J. Ellis, J. Goldick, C. Kirby, M. Levine, C. Maher, and M. Mathis, "An AFS-based Mass Storage System at the Pittsburgh Supercomputer Center," *Digest of Papers, Eleventh IEEE Symposium on Mass Storage Systems*, IEEE Computer Society Press, October 1991.

3. J. Goldick, K. Benninger, W. Brown, C. Kirby, C. Maher, D. Nydick, B. Zumach, "An AFS-Based Supercomputing Environment," *Proceedings, Twelfth IEEE Symposium on Mass Storage Systems*, IEEE Computer Society Press, April 1993.

4. V. Gate, "Alex - a Global Filesystem," *Proceedings of the Usenix File System Workshop*, 1992.

5. M. Rosenblum, J. Ousterhout, "The Design and Implementation of a Log-Structured File System," *Proceedings of the 13th ACM Symposium on Operating Systems Principles*, February 1992.

6. M. Seltzer, K. Bostic, M. McKusick, C. Staelin, "An Implementation of a Log-Structured File System for UNIX," *Proceedings of the 1993 Winter Usenix*, January 1993.

7. S. Coleman, R. Watson, R. Coyne, H. Hulen, "The Emerging Storage Management Paradigm," *Proceedings, Twelfth IEEE Symposium on Mass Storage Systems*, IEEE Computer Society Press, April 1993.

8. D. Isaac, "Hierarchical Storage Management for Relational Databases," *Proceedings, Twelfth IEEE Symposium on Mass Storage Systems*, IEEE Computer Society Press, April 1993.

N94-33825

# A Practical Large Scale\High Speed Data Distribution System Using 8 mm Libraries

**Kevin Howard**
EXABYTE
1685 38th Street
Boulder, CO 80301

## Introduction

8 mm tape libraries are known primarily for their small size, large storage capacity and low cost. However, many applications require an additional attribute which, heretofore, has been lacking -- high transfer rate. Transfer rate is particularly important in a large scale data distribution environment -- an environment in which 8 mm tape should play a very important role. Data distribution is a natural application for 8 mm for several reasons: most large laboratories have access to 8 mm tape drives, 8 mm tapes are upwardly compatible, 8 mm media are very inexpensive, 8 mm media are light weight (important for shipping purposes), and 8 mm media densely pack data (5 gigabytes now and 15 gigabytes on the horizon). If the transfer rate issue were resolved, 8 mm could offer a good solution to the data distribution problem. To that end Exabyte has analyzed four ways to increase its transfer rate: native drive transfer rate increases, data compression at the drive level, tape striping, and homogeneous drive utilization. Exabyte is actively pursuing native drive transfer rate increases and drive level data compression. However, for non-transmitted bulk data applications (which include data distribution) the other two methods (tape striping, homogeneous drive utilization) hold promise.

Tape striping is the tape analogue to disk arrays. However, there are many problems associated with tape striping, especially for the data distribution application. These problems include the need to distribute multiple tapes per data set and the requirement that each receiving site have a tape array to reconstitute the data, as well as data synchronization problems. For these reasons and others, tape striping was not deemed suitable for the distribution application.

The final mechanism for transfer rate enhancement explored by Exabyte, homogeneous drive utilization (HDU), offers the potential speed of a tape striping system, without the multiple tape transmission and receiving site burdens or the data synchronization problems associated with tape striping solutions. The primary premise of HDU is, that for large scale data systems, there is a significant amount of data concurrence. This data concurrence can be exploited so as to read or write the concurrent data in parallel. By reading or writing data in parallel, the speed of the system becomes the sum of the speed of the attached drives. The easiest exploitable concurrent data movement activities occur when performing the data duplication work needed to distribute large amounts of data.

This paper will discuss the various concurrence types, exploitable tape drive effects, exploitable tape library effects, a priori table manipulation, the Exabyte controller and describe a practical system using these techniques; that is HDU.

## Concurrence Types for Multiple Tape Drive Systems

In order to move data in parallel to a system of tape drives, one must understand when and how data set concurrence occurs. In addition, the amount and type of data set concurrence must be known for each specific application so that those concurrences can be exploited as parallel activity. My studies have led

to the conclusion that there exist at least four different types of generalized concurrence which can be exploited to produce multiple tape drive band width enhancement.

1. Type 0 Concurrence ignores the structure of the data and imposes concurrence by fracturing the data.

2. Type 1 Concurrence uses the physical hardware determined data structure to provide the concurrence structure.

3. Type 2 Concurrence uses the logical structure of the data to provide the concurrence structure.

4. Type 3 Concurrence uses the application determined data structure to provide the concurrence structure.

For brevity the equations that I am using to compute the type x concurrence throughput limits are published in the proceeds of the Goddard Conference on Mass Storage Systems and Technologies, September 22 - 24 1992, "High Speed Data Duplication/Data Distribution - An Adjunct to the Mass Storage Equation" by Kevin Howard.

***Type 0 Concurrence***

Type 0 concurrence is what is usually considered in discussions of tape striping. Type 0 concurrence works best in those applications where the media remain in a fixed or near fixed location such as a hard disk array. The first limit placed upon type 0 concurrence in tape systems is found in a consequence of tapes' removable nature. Removable media offer the opportunity for data set discontinuity which is not present for fixed media systems. This makes type 0 concurrence unsuitable for any application where the media will be frequently removed as is the case with data duplication and distribution. The second limit to type 0 concurrence is also a consequence of removable media; this is the total system throughput limit. The total system throughput limit can be seen in equation 1 below:

**(equation 1)**

Total average time = average load time + average unload time +
(2 * (average pick time + average place time)) +
(average drive speed * amount of data)

This equation shows that load/unload/pick/place must be included when discussing total system time. With a removable media system a large amount of the total time is spent as system overhead with the overall throughput limit defined to be a function of the maximum capacity of the media used, the average data set size, the average drive speed, the number of pick and place devices and their average speed, et cetera. Below is the type 0 concurrence throughput limit for an Exabyte EXB-120 tape library with EXB-8500 tape drives starting in the normal situation (the normal situation assumes that all four tape drives are full and must be emptied prior to starting.)

average drive speed .5 MB/s * 4 drives = 2 MB/s
amount of data 5 GB
number of pick and place devices 1

| | | |
|---|---|---|
| unload | = | 11 + 1 + 1 + 1 |
| pick/place1 | = | 17 + 17 + 17 + 17 |
| pick/place2 | = | 20 + 20 + 20 + 20 |
| load | = | 32 + 12 + 12 + 12 |
| overlap effect | = | 0 - 1 - 1 - 1 |
| overhead | = | 227 seconds |
| streaming | = | 2500 seconds |
| Total time | = | 2727 seconds |

Type 0 concurrence throughput limit = 1.8 MB/s

This means that under the very best case our throughput was limited from 2 MB/s to 1.8 MB/s only because of the concurrence type used. As the size of the dataset decreases the system overhead has a greater and greater effect. For example, if we limit the data set size to 2.5 GB instead of 5 GB, the Type 0 concurrence throughput limit is 1.7 MB/s, and if the data set size is 1.3 GB the Type 0 throughput limit is 1.5 MB/s. In addition to this speed degradation, any redundancy required for data integrity compounds the problem.

***Type 1 Concurrence***

Type 1 concurrence occurs in a number of applications, some of which are listed below:

1. Disk farm applications
2. Multiple channel telemetry applications
3. Multiple server networks.

The primary attributes of type 1 concurrence are:

1. The primary or secondary storage systems have a natural discrete boundary condition.
2. Those natural discrete boundary conditions can be retained in the tertiary storage system.

Exploiting type 1 concurrence is far easier than type 0 concurrence for the following reasons:

1. There is no need for data synchronization. If the natural discrete boundary conditions are maintained as a boundary condition of each tape, then each tape drive can act independently.
2. A full data set can be placed on a single tape thus eliminating both the data duplication problem and the potential data set discontinuity problem. Again, if the natural discrete boundary conditions are maintained as the boundary conditions for each tape, then each tape drive can act independently.

This still leaves the problem of increasing the total system throughput. Total system throughput can be thought of in terms of the total amount of data which needs to be transferred from one portion of the system to another. Given this definition of system throughput only three requirements need to be met to exploit type 1 concurrence for throughput enhancement. These requirements are given below:

1. Multiple tertiary drives are available.

2. The system of tertiary drives can be run in parallel.

3. The load on the system of tertiary drives can be balanced.

If these three conditions are met, then the throughput for the system can be as much as the sum of the speeds of the attached drives minus the system overhead. The throughput problem thus becomes a problem of keeping the multiple drives streaming; that is, keeping the maximum number of drives simultaneously streaming. To insure the maximum system throughput two additional conditions must be met. These conditions are:

1. The number of discrete boundary conditions is an even multiple of the number of tertiary drives or

   if the number of discrete boundary conditions is not an even multiple of the number of tertiary drives, then other "filler" data can be used to balance the system or

   the sum total of all associated data sets per drive times the native drive throughput is equivalent for all attached drives.

2. The sum total of all associated data sets per drive times the native drive throughput is at least equal to the total system overhead.

An interesting side note is the fact that type 0 concurrence can be treated as type 1 concurrence if the following conditions are met:

1. The data sets are fractured into sections which correspond to the number of tertiary drives.

2. Retrieval order for the fractured sections is not relevant.

This second condition is met when each fractured section is treated as an individual data set then recombined outside of the tertiary system.

There is a type 1 concurrence total system throughput limit. However, because we do not have to wait for the total array of drives to become available before we start the data transfer, this limit is less severe. As we did with the type 0 concurrence we will show the type 1 concurrence throughput limit for an EXB-120 tape library with EXB-8500 tape drives starting from the normal situation.

average drive speed .5 MB/s * 4 drives = 2 MB/s
amount of data 5 GB
number of pick and place devices 1

| | | |
|---|---|---|
| unload | = | 11 + 1 + 1 + 1 |
| pick/place1 | = | 17 + 17 + 17 + 17 |
| pick/place2 | = | 20 + 20 + 20 + 20 |
| load | = | 32 + 32 + 32 + 32 |
| overlap effect | = | -37 - 38 - 38 - 1 |
| overhead | = | 53 + 32 + 32 + 32 = 149 |
| streaming | = | 2500 |
| Total time | = | 2649 |

Type 1 concurrence throughput limit = 1.9 MB/s

This means that under the very best case our throughput was limited from 2 MB/s to 1.9 MB/s only because of the concurrence type used. As the size of the dataset decreases, the system overhead has a greater and greater effect. For example, if we limit the data set size to 2.5 GB instead of 5 GB, the Type 1 concurrence throughput limit is 1.8 MB/s, and if the data set size is 1.3 GB the Type 1 throughput limit is 1.6 MB/s. This compares to the 1.5 MB/s of maximum throughput for type 0 concurrence or an approximate speed difference of 10%. The difference in speed occurs because type 1 concurrence (because of its data set independence) does not need all of the tapes loaded to start transferring data.

***Type 2 Concurrence***

Type 2 concurrence can most easily be seen in network applications. Within a network server, the data is stored as a function of departments, groups, and/or users. Each department, group or users represents a discrete boundary condition which can be exploited in the same way as type 1 concurrence. In fact, because of the discrete nature of the data, the type 2 concurrence throughput limit is the same as that for type 1 concurrence. The following conditions must be met in order for type 2 concurrence to yield throughput enhancements:

1. Multiple tertiary drives are available.

2. The system of tertiary drives can be run simultaneously.

3. The load on the system of tertiary drives can be balanced.

4. The system of tertiary drives can be accessed using multiple access threads.

The first three conditions are the same as that required for type 1 concurrence. The fourth condition can be met by the tertiary system having the capability of disconnecting from one discrete boundary condition and reconnecting to another discrete boundary condition. This occurs with many hardware bus protocols -- the Small Computer System Interface (SCSI) being one.

***Type 3 Concurrence***

Type 3 concurrence occurs when the application itself determines the discrete boundary conditions. An example of this can be seen with data duplication systems. A data duplication system performs four types of duplication; they are:

1. Copy data from a single source to a single target.

2. Copy data from a single source to many targets.

3. Copy data from many sources to a single target.

4. Copy data from many sources to many targets.

This leads to two defining conditions:

1. When reading data to be copied, the number of sources define the amount of system parallelism.

2. When writing data, the number of targets define the amount of system parallelism.

These defining conditions support the discrete boundary condition which can be exploited in the same way as type 1 or type 2 concurrence. Like type 1 and type 2 concurrence the type 3 throughput limit is the same. The following conditions must be met in order for type 3 concurrence to yield throughput enhancements:

1. Multiple tertiary drives are available.

2. The system of tertiary drives can be run simultaneously.

3. The load on the system of tertiary drives can be balanced.

4. The system of tertiary drives can be accessed using multiple threads.

5. The direction of data exploited is stated.

The first four conditions above are the same as that required for type 2 concurrence. The fifth condition can be met by the application itself. For example, in the data duplication application, the number of parallel reads and writes can be known prior to committing system resources.

**Exploitable Tape Drive Effects**

A tape drive sits between the media and the host. However, unlike disk drives, a type drive can fill its internal buffer much faster than it can read or write to the media. Generally there are two measures of speed for a tape drive -- the burst transfer rate and the sustained transfer rate. The burst transfer rate can be thought of as the time it takes the tape drive to fill its buffer either from the media or from the host. The sustained transfer rate can be thought of as the time it takes the tape drive to read from or write to the media. Since many tape drives (including 8 mm) are able to disconnect after its buffer is full (the disconnect threshold is reached) and reselect after its buffer is empty (the reselect threshold is reached), we have an important tool which can be used to exploit type 1 through type 3 concurrences -- the burst transfer rate to sustained transfer rate ratio. When a tape drive disconnects from the host, the host is able to select or be reselected by other tape drives. This means that either a single or multiple hosts can cause parallel transfers of data on a single bus and keep those parallel drives streaming.

**(equation 2)**

$$D = 1 + (B / S)$$

Where:
- D = The number of drives which can stream in parallel
- B = The average burst rate of the attached drives
- S = The average sustained transfer rate of the attached drives

An example of using the burst transfer rate to sustained transfer rate ratio is given below:

EXB-8500 tape drives:

| | |
|---|---|
| Burst transfer rate | = 4.0 MB/s |
| Sustained transfer rate | = 0.5 MB/s |
| The number of parallel streaming drives | = 9 |

This means that, for each bus, up to nine EXB-8500 drives could stream if the bus can attach that number of drives, the transfer rate of the bus is at least the burst rate, and total system overhead is kept to a minimum. This example implies several interesting ways to optimize the use of a system based upon this effect. The first optimazation implication is that the wider the gap between the burst transfer rate and the sustained transfer rate, the more drives that can stream in parallel. The second optimization implication is that the gross throughput is the sum of the sustained transfer rates of all attached parallel streaming drives. The third optimization implication is that the number of channels required to achieve a given transfer rate can be computed. For example, if the required transfer rate is 8 MB/s and the drives are EXB-8500 tape drives, then two channels are required to achieve this rate with each channel able to move data a 4.5 MB/s. The number of channels required for a given number of drives (which defines the required transfer rate) is defined in the equation below:

**(equation 3)**

$$C = 1 + \text{int}(d/D) + \text{mod}(\text{rem}(\text{int}(d/D)))$$

$$d > \text{or} = D \text{ implies } C = C - 1$$

where:
- C = Number of channels required
- d = Number of drives
- D = Burst transfer rate to sustained transfer rate ratio
- int = Function which returns only the integer value of a real number
- rem = Function which returns only the decimal value of a real number
- mod = Function which returns a zero when given a zero or a one when given any other value

Note: All values are assumed to be positive real numbers.

An example of this equation in use is given below using 12 EXB-8500 tape drives:

$$C = 1 + 1 + 1 = 3$$

$$12 > \text{or} = 9 \text{ which implies that } C = C - 1 = 2$$

As can be seen, the burst transfer rate to sustained transfer rate ratio combined with the disconnect/reselection methods of many tape drives gives a practical way to insure parallel reads and/or writes of multiple tape drives on a single bus.

**Exploitable Tape Library Effects**

Tape libraries are an automated way to load and unload tapes from one or more tape drives. There are several parameters associated with the speed with which a tape library can move tapes into or out of tape drives. These parameters are given below:

1. Native robot speed and average travel distance
2. The number of drives which require tape service
3. The number of tapes within the library
4. The number of robotic tape handlers in the system.

The worst case for a multiple tape drive library system occurs when all of the tape drives require service. If there is only one handler (as is usually the case) the library itself becomes a system throughput impediment because it imposes sequential activity upon the system. Within the worst case scenario it is possible to increase the total system throughput without changing the speed of the robotic tape handler. This throughput increase is accomplished by increasing the number of handlers and using parallelism to decrease the total system overhead. To accomplish this the following conditions must be present:

1. All robotic tape handlers must be under a central control unit.
2. Parallel movement of the robotic tape handlers must be possible.

One of the most cost effective ways to insure condition 2 is to use a disconnect/reselection protocol at the robotic tape handler level. Because sending the movement commands takes so little time compared to the movement itself, near parallel robotic tape handler movement can be accomplished without the extra expense of extra bus controller circuitry, cabling, or controller firmware. If the tape libraries are inexpensive and a way to parallel control multiple libraries could be found, then it is possible to move tapes in parallel across multiple libraries, which would greatly decrease the total system overhead. For each additional library added to the system of libraries, the system overhead decreases according to the following equation:

**(equation 4)**

n > 2

$$T = \frac{\sum_{x=2}^{n} \frac{O}{X}}{n}$$

n < or = 2

$$T = \sum_{X=1}^{n} \frac{O}{X}$$

| | | |
|---|---|---|
| Where: | T | = Total system overhead |
| | O | = Overhead with one robotic tape handler |
| | n | = Total number of robotic tape handlers |
| | x | = Additional robotic tape handler number |

As can be seen, as more robotic handling systems are added their effectiveness decreases. The following sequence shows the effect of adding additional robotic handling systems:

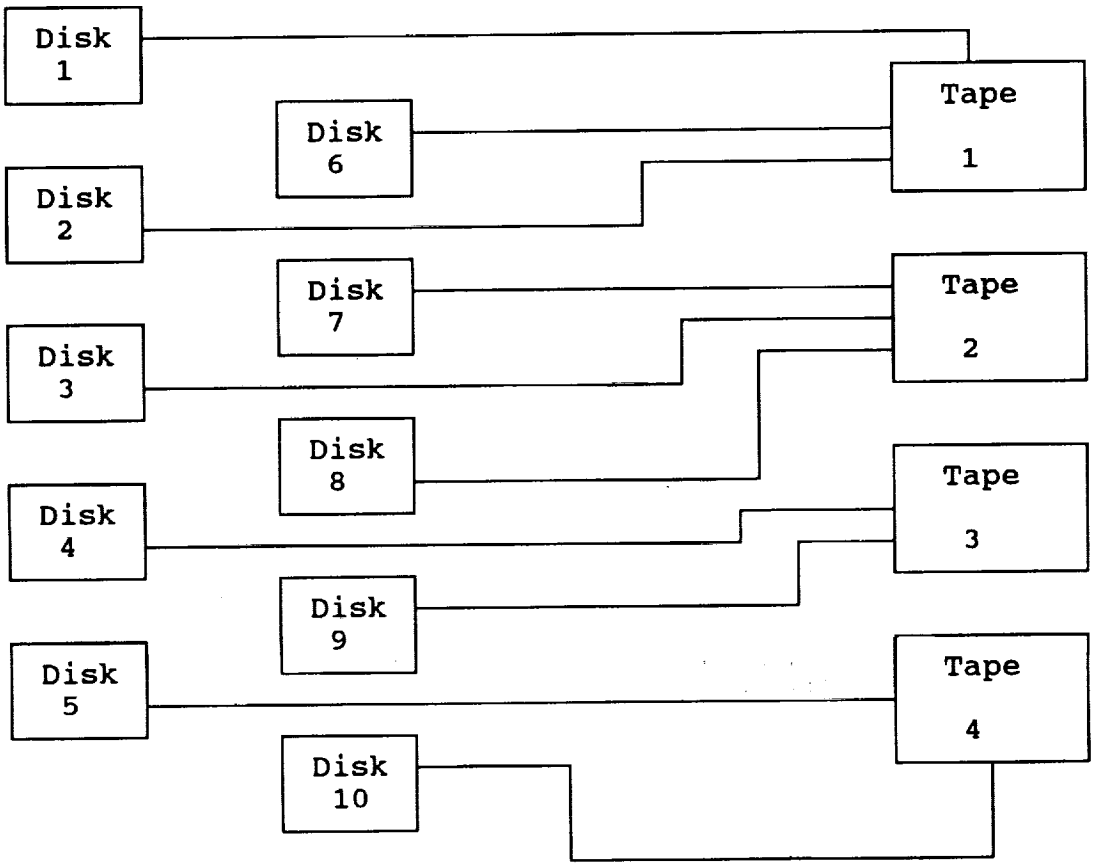
{100%, 50%, 27.7%, 27%, 25.6%, ...}

With only one robotic handling system we get 100% of the system overhead, with two we get 50% of the original system overhead, with three we get 27.7% of the original system overhead, et cetera. Adding either more libraries to the system or adding additional robotic handling devices to the system can greatly decrease the overall system overhead.

**A Priori Table Manipulation**

In type 1, type 2 and type 3 concurrence only a priori knowledge can guarantee that parallelism is maintained. Since the operative feature of types 1, 2, and 3 concurrence is that concurrence is derived from the physical world concurrence, an a priori map of this concurrence should facilitate its accurate distribution throughout the array of tape drives. For example, a disk farm which contains 10 1 gigabyte disk drives which are mapped into an array of 4 tape drives might look like the following diagram:

**Type 1 Concurrence Mapping**




The question now becomes, "How do I guarantee the above mapping?". The answer to this question is to provide the tape array system with adequate information. Adequate information in this case is given in the table below:

**A Priori Table**

| Link # | Data Set Name | Data Set Size | Links |
|---|---|---|---|
| 0 | disk 1 data | 1.0 GB | NULL |
| 1 | disk 2 data | 1.0 GB | NULL |
| 2 | disk 3 data | 1.0 GB | NULL |
| 3 | disk 4 data | 1.0 GB | NULL |
| 4 | disk 5 data | 1.0 GB | NULL |
| 5 | disk 6 data | 1.0 GB | NULL |
| 6 | disk 7 data | 1.0 GB | NULL |
| 7 | disk 8 data | 1.0 GB | NULL |
| 8 | disk 9 data | 1.0 GB | NULL |
| 9 | disk 10 data | 1.0 GB | NULL |

The link # is a simple index number for the data set name. The data set name is used to tell the host which data set is required next. The data set size allows the tape system to determine when the current drive will complete next. The links allow different discrete data components to be placed on the same media. Now the tape system can use the drive information, sustained transfer rate, burst rate/sustained rate ratio, internal tape drive buffer size to compute where each data set should go. The above map would tell the tape system that each data set would go on separate media. This would not be the optimum use of the system because there are more loads, unloads, picks, and places involved with ten tapes versus four. A better mapping would be given by the following table:

**A Priori Table With Links Filled**

| Link # | Data Set Name | Data Set Size | Links |
|---|---|---|---|
| 0 | disk 1 data | 1.0 GB | 1, 5 |
| 1 | disk 2 data | 1.0 GB | NULL |
| 2 | disk 3 data | 1.0 GB | 6, 7 |
| 3 | disk 4 data | 1.0 GB | 8 |
| 4 | disk 5 data | 1.0 GB | 9 |
| 5 | disk 6 data | 1.0 GB | NULL |
| 6 | disk 7 data | 1.0 GB | NULL |
| 7 | disk 8 data | 1.0 GB | NULL |
| 8 | disk 9 data | 1.0 GB | NULL |
| 9 | disk 10 data | 1.0 GB | NULL |

This mapping would give the best overall performance for the system of tapes. To read back the data quickly and to know which tape to load initially, one needs another column which specifies the volume number. This augmented table is shown below:

**A Priori Table With Volume Number**

| Link # | Data Set Name | Data Set Size | Links | Volume |
|---|---|---|---|---|
| 0 | disk 1 data | 1.0 GB | 1, 5 | 000001 |
| 1 | disk 2 data | 1.0 GB | NULL | NULL |
| 2 | disk 3 data | 1.0 GB | 6, 7 | 000002 |
| 3 | disk 4 data | 1.0 GB | 8 | 000003 |
| 4 | disk 5 data | 1.0 GB | 9 | 000004 |
| 5 | disk 6 data | 1.0 GB | NULL | NULL |
| 6 | disk 7 data | 1.0 GB | NULL | NULL |
| 7 | disk 8 data | 1.0 GB | NULL | NULL |
| 8 | disk 9 data | 1.0 GB | NULL | NULL |
| 9 | disk 10 data | 1.0 GB | NULL | NULL |

For robotic tape library systems which do not retain volume information, simple location information could be substituted. For data sets which can be placed on any of the currently available volumes, the "links" data can be computed a posteriori. All that is needed is a symbol to inform the tape system that a posteriori processing is required. Below is a table which would allow the tape system to assign media location:

**A Priori Table With Automatic Assignment Operator**

| Link # | Data Set Name | Data Set Size | Links | Volume |
|---|---|---|---|---|
| 0 | disk 1 data | 1.0 GB | * | 000001 |
| 1 | disk 2 data | 1.0 GB | * | NULL |
| 2 | disk 3 data | 1.0 GB | * | 000002 |
| 3 | disk 4 data | 1.0 GB | * | 000003 |
| 4 | disk 5 data | 1.0 GB | * | 000004 |
| 5 | disk 6 data | 1.0 GB | * | NULL |
| 6 | disk 7 data | 1.0 GB | * | NULL |
| 7 | disk 8 data | 1.0 GB | * | NULL |
| 8 | disk 9 data | 1.0 GB | * | NULL |
| 9 | disk 10 data | 1.0 GB | * | NULL |

These table entries would force link # 0 to volume 000001, link # 2 to volume 000002, link # 3 to volume 000003, and link # 4 to volume 000004, all other volume assignments could be made by the controller of the multi-tape drive system. After assignment the links field would be changed to reflect the final status. Below is an example:

**A Priori Table With A Posteriori Computed Assignment**

| Link # | Data Set Name | Data Set Size | Links | Volume |
|---|---|---|---|---|
| 0 | disk 1 data | 1.0 GB | *,1,8 | 000001 |
| 1 | disk 2 data | 1.0 GB | *,NULL | NULL |
| 2 | disk 3 data | 1.0 GB | *,5,9 | 000002 |
| 3 | disk 4 data | 1.0 GB | *,6 | 000003 |
| 4 | disk 5 data | 1.0 GB | *,7 | 000004 |
| 5 | disk 6 data | 1.0 GB | *,NULL | NULL |
| 6 | disk 7 data | 1.0 GB | *,NULL | NULL |
| 7 | disk 8 data | 1.0 GB | *,NULL | NULL |
| 8 | disk 9 data | 1.0 GB | *,NULL | NULL |
| 9 | disk 10 data | 1.0 GB | *,NULL | NULL |

To ensure that this information is retained regardless of the state of the tape system or its controller, the above table must made available to the host. If automatic assignment is selected for some or all of the data sets, the there must be a computation done which takes into account the data set sizes and native tape drive throughput. If we assume that the native tape drive throughput is the same for all tape drives in the array then that calculation simply assigns the next data set to the tape drive which will be depleted of data the soonest. Below are two tables
which show the effect of different data set sizes on the links field outcome:

**A Priori Table With Automatic Assignment Operator and Varying Data Set Sizes**

| Link # | Data Set Name | Data Set Size | Links | Volume |
|---|---|---|---|---|
| 0 | disk 1 data | 1.0 GB | * | 000001 |
| 1 | disk 2 data | 0.5 GB | * | NULL |
| 2 | disk 3 data | 1.0 GB | * | 000002 |
| 3 | disk 4 data | 1.0 GB | * | 000003 |
| 4 | disk 5 data | 1.0 GB | * | 000004 |
| 5 | disk 6 data | 1.0 GB | * | NULL |
| 6 | disk 7 data | 0.5 GB | * | NULL |
| 7 | disk 8 data | 0.5 GB | * | NULL |
| 8 | disk 9 data | 1.0 GB | * | NULL |
| 9 | disk 10 data | 0.5 GB | * | NULL |

**A Priori Table With A Posteriori Computed Assignment and Varying Data Set Sizes**

| Link # | Data Set Name | Data Set Size | Links | Volume |
|---|---|---|---|---|
| 0 | disk 1 data | 1.0 GB | *,5 | 000001 |
| 1 | disk 2 data | 0.5 GB | *,NULL | NULL |
| 2 | disk 3 data | 1.0 GB | *,8 | 000002 |
| 3 | disk 4 data | 1.0 GB | *,1,7 | 000003 |
| 4 | disk 5 data | 1.0 GB | *,6,9 | 000004 |
| 5 | disk 6 data | 1.0 GB | *,NULL | NULL |
| 6 | disk 7 data | 0.5 GB | *,NULL | NULL |
| 7 | disk 8 data | 0.5 GB | *,NULL | NULL |
| 8 | disk 9 data | 1.0 GB | *,NULL | NULL |
| 9 | disk 10 data | 0.5 GB | *,NULL | NULL |

As can be seen the algorithm assigns the largest data sets first and then fills in the rest of the information as required to balance the tape drive load. Type 2 concurrence would work similarly to type 1 concurrence except the logical rather than physical concurrence would be used. An unlinked data set which has a NULL volume field assignment in the a priori table is requesting the tape system controller to assign a volume. For a reasonable assignment to take place two assumptions must be made. The first assumption is that the user does not care where the data set is placed. The second assumption is that load/unload and pick/place times are expensive overhead. The natural consequence of these two assumptions is to use the normal assignment algorithm and use the currently loaded volumes. If a volume is full another volume assignment must be made.

**The Exabyte Controller**

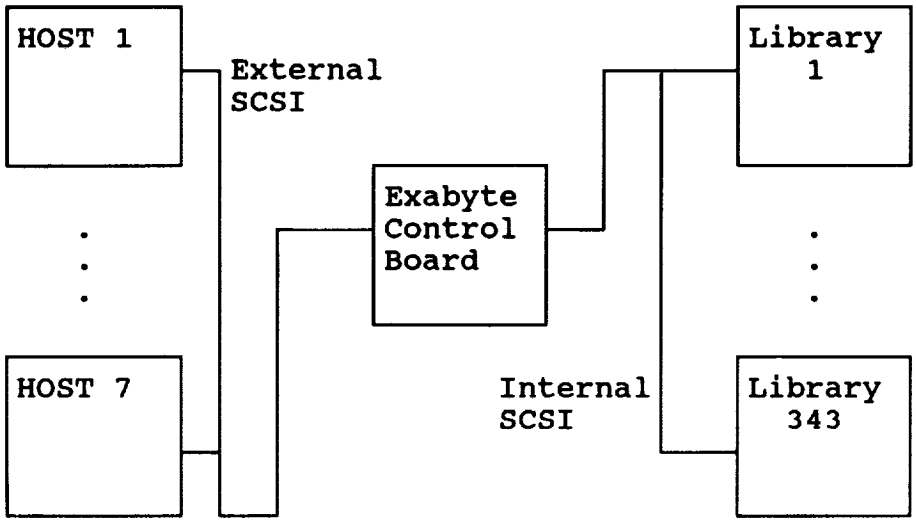
Since Exbyte tape drives and libraries are SCSI devices, the SCSI bus is an integal part of this discussion. However, any bus protocol can be used to acheive these results.

The SCSI bus protocol allows only eight devices per bus. Since one EXB-120 uses five devices, and there must be at least one host, only one EXB-120 can be attached to that host. This problem is overcome by placing a controller between the host and the tape libraries. This accomplishes two things:

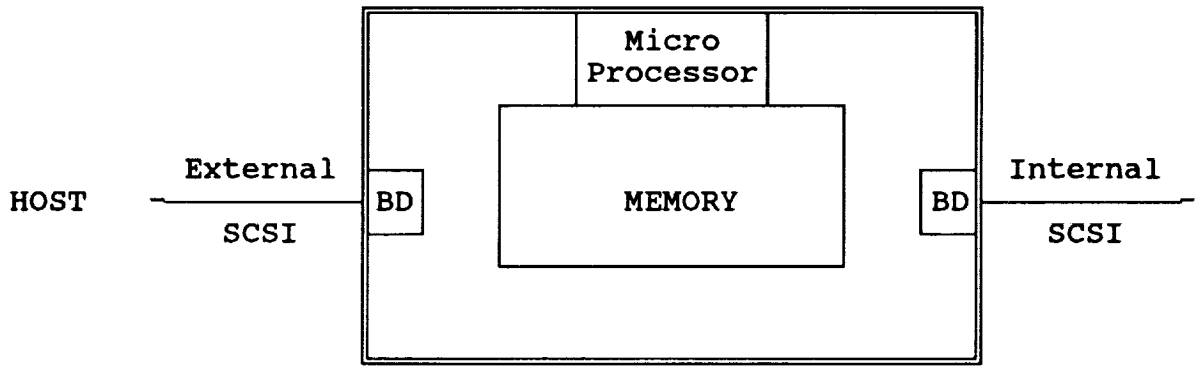
1. Provides a single control point for the various devices and

2. Allows multiple EXB-120s to reside on a single SCSI bus port.

The basic system architecture is as follows:

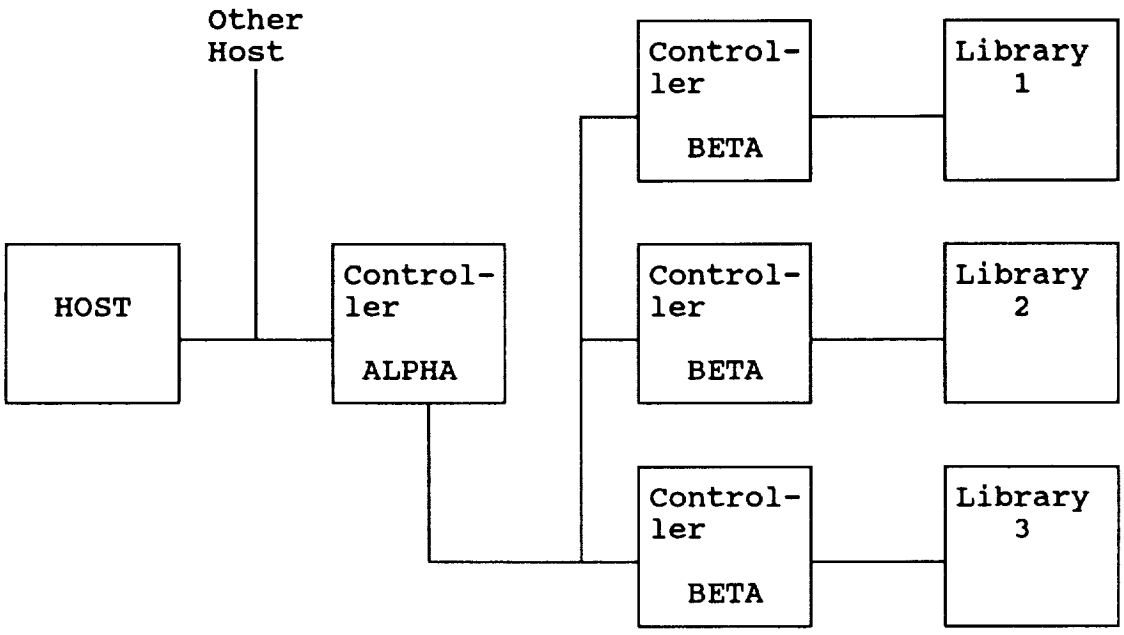



The Exabyte controller board can communicate with up to 343 EXB-120 class libraries by chaining multiple controller boards together. A controller board has the following basic architecture:




Where: BD = SCSI bus driver hardware/firmware

As can be seen, the host communicates to the Exabyte controller board via one SCSI bus (the External SCSI bus), and the Exabyte controller board communicates with the EXB-120s via another different SCSI bus (the Internal SCSI bus). This allows the SCSI device limit to be overcome. For example, if three EXB-120s are to function as a single device we would connect them together as follows:




As can be seen, there are two types of Exabyte controllers -- the alpha and beta controllers. The only difference between these controller types is the firmware.

The alpha level controller passes commands and messages through to the next level controller. Since the alpha level is the only level which has access to the host identity, it passes this host identity down to the beta level controller. The first alpha level controller, virtue of its central location, receives and interprets all system level commands. For the data distribution application we have the following system level commands:

1. Device Selection command

2. Initialize Blank Cartridges command

3. High Speed Data Duplication command.

The device selection command is an index command which allows the host to select the internal bus identity to which subsequent commands are to be passed. The format of the device select command is as follows:

| BIT<br>BYTE | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 01 | L U N | | | Reserved | | | | |
| 02 | Reserved | | | A | P | Device ID | | |
| 03 | Reserved | | | A | P | Device ID | | |
| 04 | Reserved | | | A | P | Device ID | | |
| 05 | Reserved | | | | | | | |
| 06 | Reserved | | | | | | | |
| 07 | Reserved | | | | | | | |
| 08 | Reserved | | | | | | | |
| 09 | Reserved | | | | | | | |
| 0A | Reserved | | | | | | | |
| 0B | Reserved | | | | | | | |

The first byte is the command identification byte. The LUN field is for compatability with an older protocol. The 'A' fields are the active bits. Because an alpha level controller can be attached to another alpha level controller, we support three levels of communication. This provides us with the ability to communicate with up to 343 EXB-120 class tape libraries. The 'P' fields are the pass-through bits. The pass-through bits determine if subsequent commands and messages will be passed through to the next level.

The initialize blank cartridges command allows the host to designate where in the system of libraries the blank cartridges are located. This is important for the subsequent analysis required to optimize the data duplication effort. The format of the initialize blank cartridges command is as follows:

| BIT<br>BYTE | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 01 | L | U | N | Reserved | | | | |
| 02 | Reserved | | | | | | | |
| 03 | Reserved | | | | | | | |
| 04 | Reserved | | | | | | | |
| 05 | Reserved | | | | | | | |
| 06 | Reserved | | | | | | | |
| 07 | Reserved | | | | | | | |
| 08 | Reserved | | | | | | | |
| 09 | Reserved | | | | | | | |
| 0A | Number of Data Pages Field | | | | | | | |
| 0B | Reserved | | | | | | | |

If the number of data pages field is zero, then this command deletes the memory of all blank cartridge locations. If the number of data pages field is not zero then the data pages are used to define the location and number of blank cartridges in the system of libraries. Below is the format of the data pages:

| BIT<br>BYTE | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | Page Number | | | | | | | |
| 01 | Reserved | | | | A | Device ID | | |
| 02 | Reserved | | | | A | Device ID | | |
| 03 | Reserved | | | | A | Device ID | | |
| 04 | Blanks Location (000-007) | | | | | | | |
| 05 | Blanks Location (008-015) | | | | | | | |
| 06 | Blanks Location (016-023) | | | | | | | |
| 07 | Blanks Location (024-031) | | | | | | | |
| 08 | Blanks Location (032-039) | | | | | | | |
| 09 | Blanks Location (040-047) | | | | | | | |
| 0A | Blanks Location (048-055) | | | | | | | |
| 0B | Blanks Location (056-063) | | | | | | | |
| 0C | Blanks Location (064-071) | | | | | | | |
| 0D | Blanks Location (072-079) | | | | | | | |
| 0E | Blanks Location (080-087) | | | | | | | |
| 0F | Blanks Location (088-095) | | | | | | | |
| 10 | Blanks Location (096-103) | | | | | | | |
| 11 | Blanks Location (104-111) | | | | | | | |
| 12 | Blanks Location (112-115) | | | | | | | |

The page number field gives the current data page identity. The 'A' field indicates that the associated device identity is active. The device identity fields provide a path to the correct library and the blanks location fields are a bit map which show the location within a library of each blank cartridge.
The blanks location fields of the page data is a bit map which designates the position within the library of the blank cartridge.

The high speed data duplication command allows the host to analyze and duplicate files in a system coordinated manner.

| BIT / BYTE | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 01 | L U N | | | Reserved | | | | |
| 02 | Reserved | | | | | | ANA | H |
| 03 | (MSB) | | | | | | | |
| 04 | HOST DATA LENGTH | | | | | | | |
| 05 | | | | | | | | |
| 06 | | | | | | | | (LSB) |
| 07 | (MSB) | | | | | | | |
| 08 | NUMBER OF HOST DUPLICATIONS | | | | | | | (LSB) |
| 09 | Reserved | | | | | | | |
| 0A | Allocation Length | | | | | | | |
| 0B | Reserved | | | | | | | |

The 'ANA' field is a bit which designates that either analysis or actual duplication is to take place. The 'H' field is a bit which designates that the data is to come from the host. The host data length field allows up to 4 GB to be duplicated. The number of host duplications field allows up to 65,000 copies to be made. We will not discuss tape to tape copying, blank tape location rejection, or host multiple data set analysis for brevity. However, it is clear how the use of a prior tables can be used in the multiple data set analysis to guarantee parallelism.

If analysis is selected, the Exabyte controller will divide the duplications by the number of attached libraries which contain blank cartridges. This is to decrease the duplication overhead by performing the cartridge loading and unloading in parallel. We then map the duplications to the specific tape drives within the libraries. A list of the cartridge locations which will receive the duplicate data is then sent back to the host. If the host approves of the selection it retransmits the high speed data duplication command with the 'ANA' field set to duplicate. The data to be duplicated is then sent to the controller and, if larger than the controllers buffer, is saved on a scratch tape. Finally, the data is duplicated in parallel to the designated blank tapes.

## A Practical Data Distribution System

A practical data distribution system using 8 mm tapes consists of the following elements:

1. Host computers with access to the data to be duplicated
2. Two or more Exabyte controllers
3. One or more Exabyte libraries.

Because of the application (data distribution), type 3 concurrence can be assumbed. To perform the data duplication portion of the data distribution system, the host first gives the location of the blank cartridges to the lead controller. Next the host requests that duplication analysis be done by the lead controller. Finally, the host sends the data to be duplicated to the lead controller which duplicates the data. This concludes the data duplication effort. Now the tape cartridges are collected and mailed to their respective sites.

The speed of the duplication effort is limited only by the bus limits. In the case of the SCSI bus these limits are 5 MB/s for normal SCSI, 10 MB/s for fast SCSI, 20 MB/s for fast/wide 16 SCSI, and 40 MB/s for fast/wide 32 SCSI.

In conclusion, HDU offers tremendous throughput for the data distribution application and indeed for all other bulk data applications.

N94-33826

# Distributed Active Archive Center

**Lee Bodden**
Hughes STX
Code 902.2, GSFC, NASA
Greenbelt, MD 20771
Telephone: (301) 286-9413
bodden@eosdata.gsfc.nasa.gov

**Phil Pease**
NASA
Code 902.2, GSFC
Greenbelt, MD 20771
Telephone: (301) 286-4418
pease@eosdata.gsfc.nasa.gov

**Jean-Jacques Bedet**
Hughes STX
Code 902.2, GSFC, NASA
Greenbelt, MD 20771
Telephone: (301) 513-1646
bedet@eosdata.gsfc.nasa.gov

**Wayne Rosen**
Hughes STX
Code 902.2, GSFC, NASA
Greenbelt, MD 20771
Telephone: (301) 286-3526
rosen@eosdata.gsfc.nasa.gov

## Abstract

The Goddard Space Flight Center Version 0 Distributed Active Archive Center (GSFC V0 DAAC) is being developed to enhance and improve scientific research and productivity by consolidating access to remote sensor earth science data in the pre-EOS time frame. In cooperation with scientists from the science labs at GSFC, other NASA facilities, universities, and other government agencies, the DAAC will support data acquisition, validation, archive and distribution. The DAAC is being developed in response to EOSDIS Project Functional Requirements as well as from requirements originating from individual science projects such as SeaWiFS, Meteor3/TOMS2, AVHRR Pathfinder, TOVS Pathfinder, and UARS. The GSFC V0 DAAC has begun operational support for the AVHRR Pathfinder (as of April, 1993), TOVS Pathfinder (as of July, 1993) and the UARS (September, 1993) Projects, and is preparing to provide operational support for SeaWiFS (August, 1994) data. The GSFC V0 DAAC has also incorporated the existing data, services, and functionality of the DAAC/Climate, DAAC/Land, and the Coastal Zone Color Scanner (CZCS) Systems.

## Introduction

This paper presents the architecture of the DAAC which includes two SGI 4D/440 mini-supercomputers and numerous smaller computers including: an HP 730, MicroVAX II, VAX 3900, SGI 4D/35 and three SUNs all configured in a distributed environment. The DAAC contains two different mass data storage systems, a Cygnet 1803 12" WORM Optical Jukebox and a Metrum RSS 600 VHS Automatic Tape Cartridge System. Both systems are being configured under the UniTree File Management System. The DAAC also supports a host of peripheral devices including two 9-track tape drives, three 8 mm tape drives, two 3480 tape drives, two 4 mm, two CD ROM drives, over 40 GB of magnetic disk storage, ten X-terminals and over 25 Macintoshes and personal computers. The DAAC's distributed environment includes two ethernet Local Area Networks, an FDDI network interface, two appletalk networks, and a T1/T3 link. This paper presents the advantages and disadvantages of the chosen architectural approach of the DAAC including a discussion of the cost trade-off analyses justifying the decisions made by the DAAC. This paper also discusses the system performance characteristics in terms of throughput rates and volumes for the data ingested into the DAAC's archive and for data distribution conducted by the DAAC. The percentages of data distributed on different media, and the medias popularity is also discussed.

## GSFC V0 DAAC Mission

The Distributed Active Archive Center (DAAC) is a component of NASA's Earth Observing System (EOS) Data and Information System (EOSDIS). The EOSDIS acquires Earth science data, derives scientifically useful data products, archives the data products and makes them available to the Earth science researchers. The EOSDIS currently includes eight DAAC sites. These DAAC sites are generally oriented around scientific disciplines and are multi-agency.

A DAAC consists of three components, a Product Generation System (PGS) that generates derived data products, a Data Archive and Distribution System (DADS) that stores the data products and distributes requested products to a researcher, and a Information Management System (IMS) that are used by researchers as a catalog of all the DAAC products from which he/she can select specific data files of interest. The IMS allows the user to select data based on time, spatial location, geophysical parameter and/or instrument. The IMS will also provide a capability to browse interesting data products as an aid to ordering the data. The IMS at all the DAACs are interoperable so the user sees the holding of all the DAACs and can order them from any DAAC he/she logs into.

This paper focuses on the DADS component. Data are ingested into the DADS primarily over the EOSDIS dedicated computer network either from instrument data capture facilities, other DAACs, or from the DAAC's own PGS. Metadata information is extracted or created from each data file and loaded into the IMS database. The data are archived to on-line (magnetic disk), near-line (robotics storage system), or off-line (on the shelf) storage. When an order for data is received via the IMS the data are copied from the archive to either magnetic disk for network (FTP) distribution or to magnetic tape (8mm, 4 mm, and 9 track are standard media supported).

The EOSDIS and the DAAC elements are being developed in an evolutionary manner with Version 0 being the initial system. Version 0 is intended to demonstrate the concept of an interoperable set of distributed archive centers and to prototype various aspects of the system. The version 0 will operate with pre-EOS satellite data sets, either currently existing or missions between now and the EOS flights.

## Requirements

The GSFC V0 DAAC archive will contain about 20 Terabytes by FY97. The amounts of data expected from the projects and sources interfacing with the DAAC is shown in Figure 1. Rates for data delivery into the DADS are expected to reach 17 GB/day via a computer network. FTP data distribution and other networking activity is expected to double this figure for a total network load of 30 to 40 GB/day. Estimates are that distribution volumes may reach 50 to 60 GB/day. It is estimated that for tape distribution, 50% will be on 8mm cartridges, 33% on 9-track 6250 bpi round tapes and 17% on 4 mm cartridges. Distribution on prepublished CD-ROMs will also be supported.

The researcher will be able to order and receive small amounts (TBD) of data via network transmission during an interactive session while logged on to the DAAC's computers. Larger amounts of data will be available for distribution on the various media supported by the DAAC. The guideline is that all orders will be filled within 30 days, with 3 days response time being a desirable goal. Specific data sets have been identified by a Science User Working Group as being high priority (expect a lot of scientific interest) and with this prioritization, the DAAC have organized their on-line, near-line, and off-line archive storage to have the higher priority data more readily available.

One group of data that will also be stored on-line and accessed interactively by a user are the browse products. Browse products are reduced resolution images used as an aid for selecting and ordering data. The user will need to have the analytical tools required to display these browse images. Other data products such as scientific documentation describing the data sets will also be available for ordering.

Data compression is planned prior to archiving in order to reduce storage needs. The DAAC will encourage users to accept data in compressed form but will decompress the data prior to distribution to the user if desired. Data compression is also being recommended for the data being transmitted into the DAAC from the various supported science projects.

## Strategy and Approach

The approach being used to meet the above requirements begins with an analysis of the system capabilities. This analysis initially was done using crude spreadsheet calculations of overall bandwidth for networks and published write rates for various peripheral devices. These simple calculations were used to arrive at a hardware configuration that was "in the ballpark" and the computer and two each of each peripheral were ordered. This initial configuration provided a platform for software development and for making performance measurements to gain better throughput figures.

We then initiated the development of a computer simulation of the workload, configuration, and operation of the DADS. Performance measurements were made to determine parameter values to be used in the computer model (e.g. actual device transfer rates achieved using operational software) and overall system throughput was calculated and compared with the simulation results (e.g., simultaneous ingest and multiple distribution activity).

With the validation of the computer model, using the benchmark measurements, the model can then be selectively modified to assess changes to the system configuration (e.g., faster processors, more tape drives, more disk space, use of data compression, number of operators) or workload (e.g., different proportion of distribution media types requested, greater number of requests for data).

Finally, after all hardware and software development and integration is completed the DAAC will perform a formal test of the systems ability to meet the performance requirements. These tests will also include stress testing to determine the upper limits of the processing capabilities of the DAAC.

## Trade-off Analyses

The trade-off analyses for the DAAC began with the evaluation of different computers and operating systems. The types of media that would be supported and the drives were also analyzed. Most importantly, the DAAC evaluated the mass data storage hardware currently available and the file management systems that will support the hardware.

All major computers available on the market today were evaluated during this analysis. Each computer was evaluated against the following criteria:

- MIPS
- Internal BUS throughput
- Individual magnetic disk storage capacity
- Magnetic disk transfer rates
- Power requirements
- Drives supported including interface mode
- Total number of drives supported
- Long-range maintenance
- Network connectivity
- Product reliability
- Applications s/w supported (DBMS,tools)
- Cost
- MFLOPS
- SCSI & IPI channel throughputs
- Total magnetic disk storage capacity
- Operating system/planned upgrades
- Space requirements
- Availability of device drivers
- Upgrade path
- Total memory capacity
- Product quality
- Procurement vehicle
- File management systems supported
- Standards supported(x-window/motif)

The information for the criteria listed above was collected and compared, and the SGI 4D/440 S and 4D/440 VGX computers were selected. This computer provided a very cost-effective MIPS/$ with an eight CPU expansion capacity per computer. The SCSI channel and internal BUS throughput rates were fast enough to meet the requirements of the DAAC and the disk storage capacity could also be expanded to meet the DAAC's needs. The space efficient SGI had high marks for quality and reliability with a low maintenance record. The SGIs also provided fddi and ethernet network connectivity.

The peripheral drives and corresponding media were also investigated as part of the overall computer system using the following additional criteria:

- Drive transfer rates
- Media capacity
- Available device drivers
- Popularity of media and drive
- File search time
- Media longevity
- Compatibility with host computer
- Cost of media and drives

Using this criteria, the DAAC selected SGI 8 mm, 4 mm, QIC, 9-track, and CD-ROM drives. Many of these drives were third party hardware sold by SGI. The risk of having interface problems with the computers was greatly reduced by selecting drives that have been thoroughly integrated. Two Fujitsu 3480 drives (one with a stacker) were also procured. This wide range of peripherals allows the DAAC to provide support to a broad base of users, an important consideration for the DAAC.

The DAAC analyzed mass storage hardware systems and the corresponding file management systems. The criteria used in this analysis were:

- Drive transfer rates
- Media capacity
- Mass storage system capacity
- Available device drivers
- Power requirements
- Reliability in the field
- Procurement vehicle
- Cost of file management system/licenses
- Maturity of file management system
- Data format and standards
- Adherence to IEEE Mass Storage Reference Model
- Multiple mass data storage systems supported
- File search times
- Media longevity
- Expandability and upgrade paths
- Compatibility with host computer
- Quality
- Maintenance costs
- Cost of hardware, media and drives
- Space requirements
- Functionality provided
- Supports hierarchical file migration
- Vendor support
- Integration support

The results of some of these analyses are shown in Figures 2 and 3. The Cygnet 1803 12" WORM Optical Jukebox and the Metrum RSS 600B VHS Automated Tape Library were selected along with UniTree as the file management system. The optical media inside the Cygnet jukebox provides the DAAC with a long-life substrate for its most important data. The Cygnet jukebox also provides rapid access for files that require it such as browse data. The media cost does prohibit all of the data from being place on the Cygnet jukebox. The Metrum provides a slightly higher throughput than the Cygnet jukebox with a very cost-effective $/TB ratio. The low cost of the media makes the Metrum the DAAC's selection for where most of the data will be stored. The UniTree file management system is the only system that can support both mass storage systems, although support for the Cygnet jukebox and the dual support capability were introduced into UniTree at the request of the DAAC. The selection of a file management system was essential in avoiding expensive development and maintenance costs associated with providing this functionality as part of the software development effort.

## Hardware and Software Selected

The Silicon Graphics Inc. 4D/440 VGX computer is a four CPU machine that was selected for the IMS. It can be upgraded to an eight CPU version (4D/480) by simply plugging in additional boards. The ease of expansion and the relatively inexpensive cost was a factor in the selection of this system. Other factors are the commercial software packages available for this platform.

The database manager product used in the IMS is Oracle. Oracle was chosen primarily because in had been successfully used previously on other data systems that the DAAC organization continues to operate. Another factor is that the Oracle product on the SGI computer can use any and all of the processors available and thus as the need requires additional CPU boards can be added. A feature used with oracle is configuring for separate tables and interface from remote machines for software development activities, system testing activities, and for operational activities. The large number of platforms for which Oracle is available provides flexibility in future system configuration changes.

The IMS user interface was implemented using the JYACC Applications Manager (JAM). This product allowed us to create both the interface for alphanumeric users and for graphical users without needing to develop separate programs. JAM also supported interface with the Oracle database product and allowed running the interface from remote systems without additional license

costs. The wide variety of platforms for which JAM is available provides flexibility in future configuration changes.

The SGI 4D/440 S is a four CPU machine in a server configuration that is the computer system selected for the DADS activities. This server configuration substitutes the graphics hardware [in the VGX model] with additional I/O capacity. Like the IMS machine this system is expandable to eight CPUs. Cost and availability of software was a factor in selecting this computer system. Also, having the same operating system for both the IMS and DADS makes system support easier.

The DAAC selected two mass data storage systems for its archive. The first is a Cygnet model 1803 12" WORM Jukebox with two ATG Gigadisc model GD9001 WORM drives. The ATG WORM platters hold 4.5 GB per side. With the two drive configuration, the Cygnet holds 131 platters providing a total storage capacity of 1179 GB. The second mass storage system is a Metrum model RSS-600B Automated Tape Library system with four model RSP-2150 VHS Cartridge Tape Drive Subsystems. The DAAC is currently using ST-120 VHS cartridges, that hold 14.5 GB per cartridge. The RSS-600B holds 600 cartridges providing a total storage capacity of 8700 GB. The Metrum system can also be used with ST-160 VHS cartridges that hold 18 GB/cartridge yielding a storage capacity of 10800 GB. The DAAC will be storing its low level (L1) data on WORM because of it's reported long life characteristic and the higher level (L2, L3, and L4) on VHS tape because this data is more likely to be reprocessed and replaced as better scientific processing algorithms are developed.

UniTree Central File Manager (UCFM) from Titan Client/Server Technologies and Open Vision was selected to manage the archive. Agreements were reached to introduce into this version (1.6.1) of UniTree support for mixed mass storage media. It also has been enhanced to support asynchronous I/O and thus can take advantage of the multiple CPUs of the SGI 4D/440 machine to give improved performance for simultaneous archive and multiple distribution activities.

## Hardware and Network Architecture

The hardware architecture of the DAAC is shown in Figure 4. The functionality of the DAAC was distributed over two computer systems in the operational configuration; the Information Management System (IMS) and the Data Archive and Distribution System (DADS).

For distribution of the large number of data orders, anticipated for 8 mm cartridges and 4 mm DAT media, several of the distribution tape drives are configured in a tape stacker configuration. This will reduce the workload on the operations staff for mounting and dismounting of media.

## Functional Capabilities

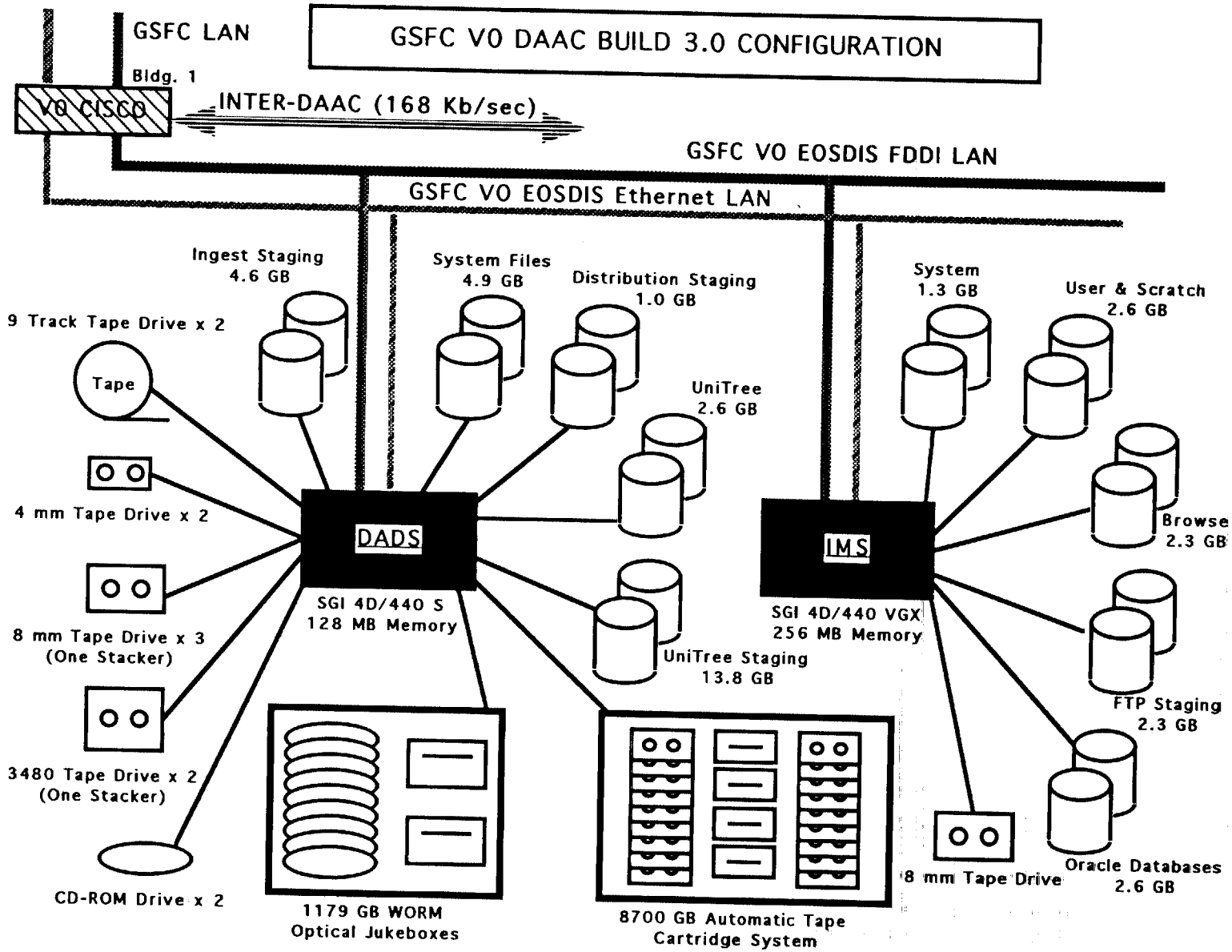
*Information Management System (IMS)*

Users will connect to the IMS computer through the GSFC V0 EOSDIS Ethernet LAN [network]. The user interacts with the IMS system through an interface program that support either an alphanumeric or graphics terminals. There are actually two IMS interface programs; one is an EOSDIS IMS interface that interacts with all of the DAAC sites (there are currently eight) and the other interacts only with the local (i.e., GSFC) DAAC. With the EOSDIS IMS the user sees the holding at all the DAACs while the GSFC local IMS only sees the GSFC DAAC holdings.

Either of these IMS user interfaces is used to search a database containing metadata information for the DAAC data holdings in order to identify and then request desired data. Users may also order browse data, for data sets of interest, that may be viewed on his/her local workstation, and to directly order the corresponding data file from the browse viewer program. Orders for data are stored in an order database. Ordered data may be retrieved over the network or copied to media and mailed to the user.

*Data Archive and Distribution System (DADS)*

The DADS provides two main functions; the ingest and archiving of data and copying of data from the archive to a disk for network distribution or to media for distribution by that mechanism. Most


GSFC V0 DAAC BUILD 3.0 CONFIGURATION
GSFC LAN
Bldg. 1
V0 CISCO
INTER-DAAC (168 Kb/sec)
GSFC V0 EOSDIS FDDI LAN
GSFC V0 EOSDIS Ethernet LAN
Ingest Staging
4.6 GB
System Files
4.9 GB
Distribution Staging
1.0 GB
System
1.3 GB
User & Scratch
2.6 GB
9 Track Tape Drive x 2
Tape
UniTree
2.6 GB
4 mm Tape Drive x 2
DADS
IMS
Browse
2.3 GB
8 mm Tape Drive x 3
(One Stacker)
SGI 4D/440 S
128 MB Memory
SGI 4D/440 VGX
256 MB Memory
UniTree Staging
13.8 GB
FTP Staging
2.3 GB
3480 Tape Drive x 2
(One Stacker)
CD-ROM Drive x 2
1179 GB WORM
Optical Jukeboxes
8700 GB Automatic Tape
Cartridge System
8 mm Tape Drive
Oracle Databases
2.6 GB

# User Interface Development and Metadata Considerations for the Atmospheric Radiation Measurement (ARM) Archive

**P. T. Singley, J. D. Bell, P. F. Daugherty, C. A. Hubbs, and J. G. Tuggle**

Environmental Sciences Division
Oak Ridge National Laboratory
Bldg. 0907, Mail Stop 6490
P.O. Box 2008
Oak Ridge, Tennessee 37831-6490
Phone: (615) 574-7817
Fax: (615) 574-4665
sin@ornl.gov

Based on work performed at the Oak Ridge National Laboratory Oak Ridge, Tennessee 37831 managed for the U.S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

## Introduction

This paper will discuss user interface development and the structure and use of metadata for the Atmospheric Radiation Measurement (ARM) Archive. The ARM Archive, located at Oak Ridge National Laboratory (ORNL) in Oak Ridge, Tennessee, is the data repository for the U.S. Department of Energy's (DOE's) ARM Project. After a short description of the ARM Project and the ARM Archive's role, we will consider the philosophy and goals, constraints, and prototype implementation of the user interface for the archive. We will also describe the metadata that are stored at the archive and support the user interface.

ARM is a part of the global research effort directed toward understanding weather and climate change. The current generation of climate simulations, general circulation models (GCMs), cannot treat the physics of radiative transport and cloud behavior at the relevant distance scales. DOE has initiated ARM to characterize the physical and dynamical structure of the atmospheric column well enough to significantly improve the modeling of the radiative flux of the earth. This entails measuring radiative fluxes and a wide range of atmospheric conditions at five highly instrumented sites worldwide. The ARM sites constitute the Cloud and Radiation Testbed (CART). Each site will collect data from all its instruments for transmission to the ARM Archive. The first site, in Lamont, Oklahoma, has been operational since June 1992. Other sites will come on-line over the next few years.

The ARM Archive stores ARM data and will provide the scientific community with data taken from the sites, along with data developed from the merger of site data with data from external sources, information describing the quality assurance (QA) checks, and contextual information. The archive will eventually use a mass storage system architecture based on the National Storage Laboratory (NSL) architecture to manage and store these data. This system uses a relatively small computer that controls a group of mass storage devices linked by a high-speed data network.

## Philosophy and Goals of the ARM User Interface

ARM Archive users—students, government-funded researchers, and policymakers—span a fairly large range of interests and capabilities. Initially, the user interface will be primarily designed to support professional researchers in atmospheric science and related disciplines. As more information about the data is available at the ARM Archive and more summary and



C-6

4

value-added data products are created, the focus of the user interface will expand to support a broader user community.

To support users, both initially and over the long term, the principal goal of the user interface is to provide enough information about data and products that are housed in the ARM Archive so the users can select exactly the data they need. To make the necessary information as accessible as possible, the user interface is designed to address the users in terms familiar to them. For instance, the user interface offers complete instrument names rather than the cryptic abbreviations that instruments are known by within the CART Data System. In addition to providing the users with familiar terminology, the user interface hides the details of how data are managed at the ARM Archive. The users do not need to know about file names, data base structures, or how to develop a data base query to get access to archive data. To successfully retrieve data from the archive, the users indicate the instruments of interest, date ranges, and other criteria (such as data processing level or QA level) that will narrow their selection. Then the users request that the data be retrieved.

Another goal of the user interface is to make sure that it is inexpensive enough so that it can be given away to anyone. Furthermore, every effort will be made to port the interface to a wide variety of computer hardware. Input from users working with the initial interface will help refine this system. This should ensure that the interface will continue to provide easy access to the ARM Archive.

## Constraints on the User Interface

Constraints as well as goals shape all systems. In addition to the usual limits on money, time, and resources, the ARM Archive has several technical constraints, some unique to the ARM Archive and some that affect any large scientific data archive.

One of the leading constraints for a large scientific data archive is simply its size. Although not as large as several of the NASA data centers, the ARM Archive will hold as much as 100 terabytes of data and metadata by the time the ARM Project ends. With that amount of data and metadata, maintaining all or even a significant portion of it on spinning disk in a data base management system is not feasible, either technically or economically. Almost all of the data, and a good fraction of the metadata, will be maintained only as files in a tape-based mass storage system. The user interface will be based primarily on the metadata that are kept in a relational data base management system (RDBMS). In addition, some metadata will be managed and accessed in a Wide Area Information Server (WAIS), a search/retrieval system for computer networks. The smallest data granule given to the user will be a file. Because of the volume of information, users will not be able to directly browse the data. Summary or value-added products may be created so users can browse.

Another aspect of keeping the data files in a mass storage system is that the user can only request that data be retrieved from the system and not examine these data using the user interface. The user must later retrieve the requested files via "ftp" (electronic transfer) or wait for surface mail to deliver the physical media containing the requested data to the user's computer. Direct interactive exploration of the data in the ARM Archive is not available.

Additional difficulties are imposed by the fact that not all of the data or metadata arrive at the archive produced in the same format, written with the same degree of formality, or subject to the same level of quality control. This diversity and how it is dealt with by the ARM Archive system are discussed later in the paper. Briefly, most of the data and the formal metadata, such as instrument, location, and current calibration readings, arrive packaged in highly structured NetCDF files generated by the Site Data System (SDS). Some data arrive in the file format of the instrument that produced them, with little associated metadata. Finally, there are logs describing conditions of instruments and other information about the site that affect instrument operation. These are human generated and fairly informal, and there is little or no quality control of the entries. All of this variability makes presenting the users with the necessary metadata in a uniform fashion a very challenging problem.

## Implementation of the User Interface

We have chosen to build the user interface on a client/server model shown schematically in Fig. 1. The user interacts with screens provided by a client application. This client sends requests to, and receives data from, a server system, which includes the archive's RDBMS and file-retrieval system. Currently, TELNET is used to access the archive's host computer and the user interface. In the future, the client will reside on the user's local computer while the server runs on the archive's host computer. Our prototype interface is based on the X-windows protocol using the Motif window manager.

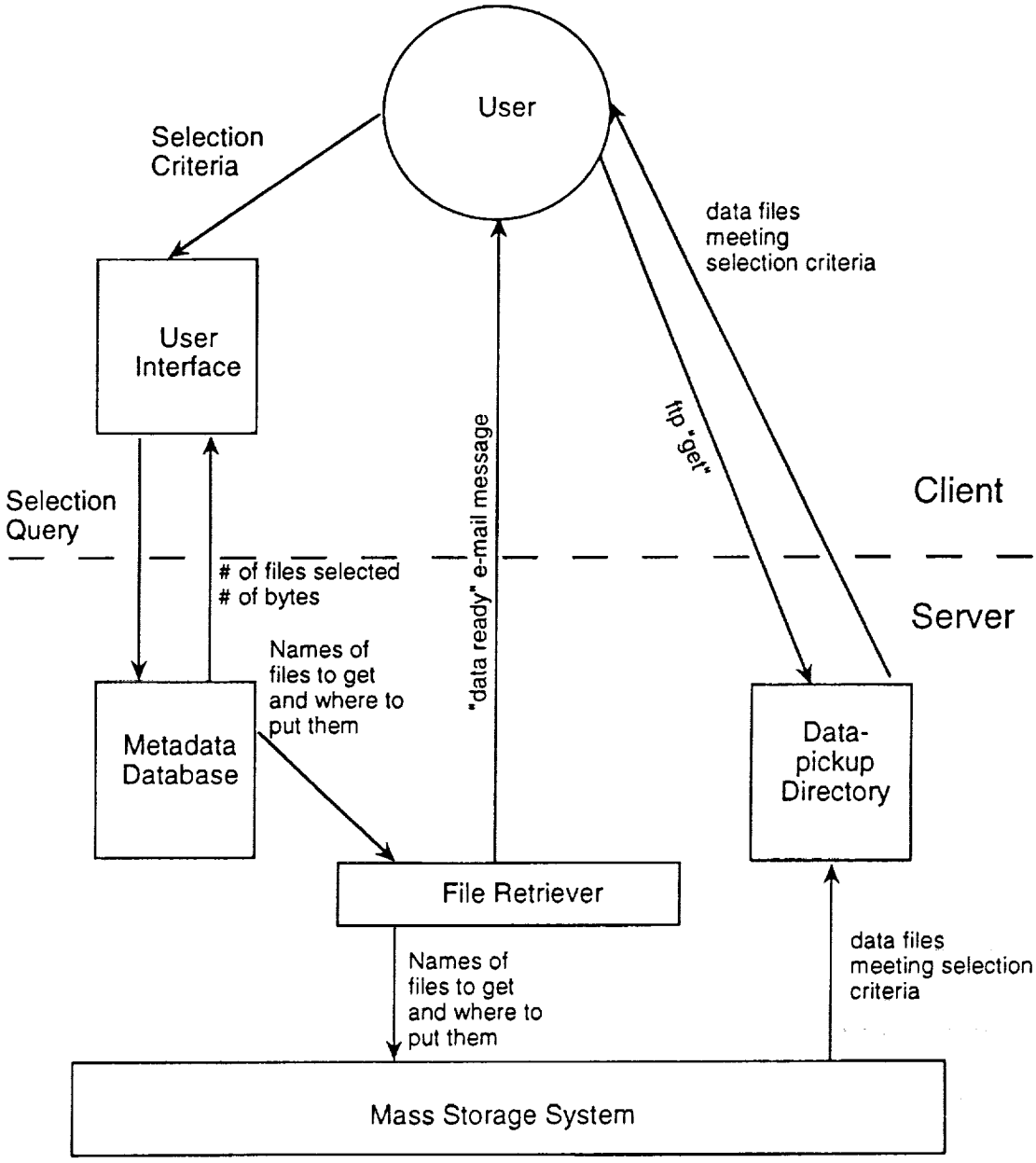
The initial screen is shown in Fig. 2. Here the user is asked for a user ID and an easily remembered password. On the second screen of each session (Fig. 3), the current identifying data about this user are displayed for verification: name, phone number, electronic-mail address, and surface-mail address. New users will be prompted for these data when they first log on to the archive.

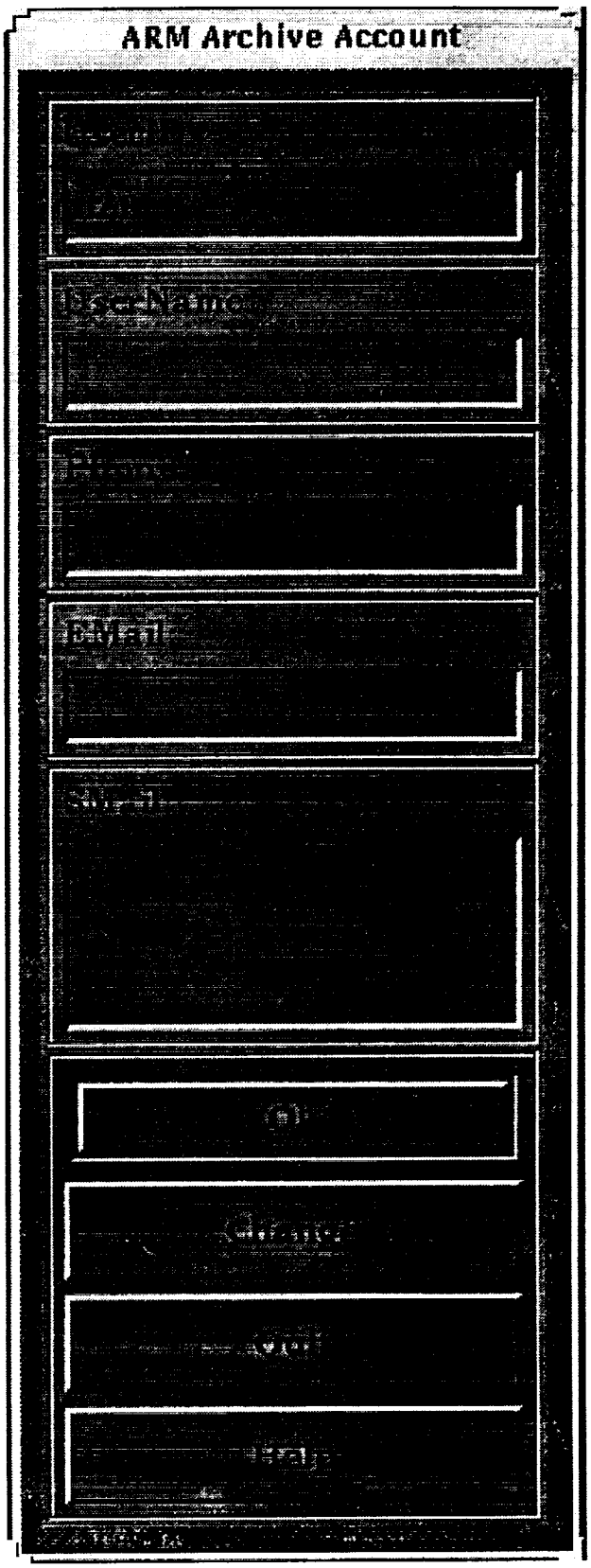
The SDS packages data into files labeled by platform and date. The Data Criteria screen (Fig. 4) is hence composed of sections for selecting a set of platforms and a range of dates. The user can select platforms based on either the SDS platform name or its equivalent "plain English" instrument name (and accompanying location information). This prototype assumes that the user already knows which named data streams are contained in the data files for a single platform or instrument; an RDBMS table and interface screens for this mapping will be added later. The user can type or use a scroll bar to specify the start and end of the date interval for which he or she wishes to get data. The system will return all those data files that include at least one data point within the interval specified.

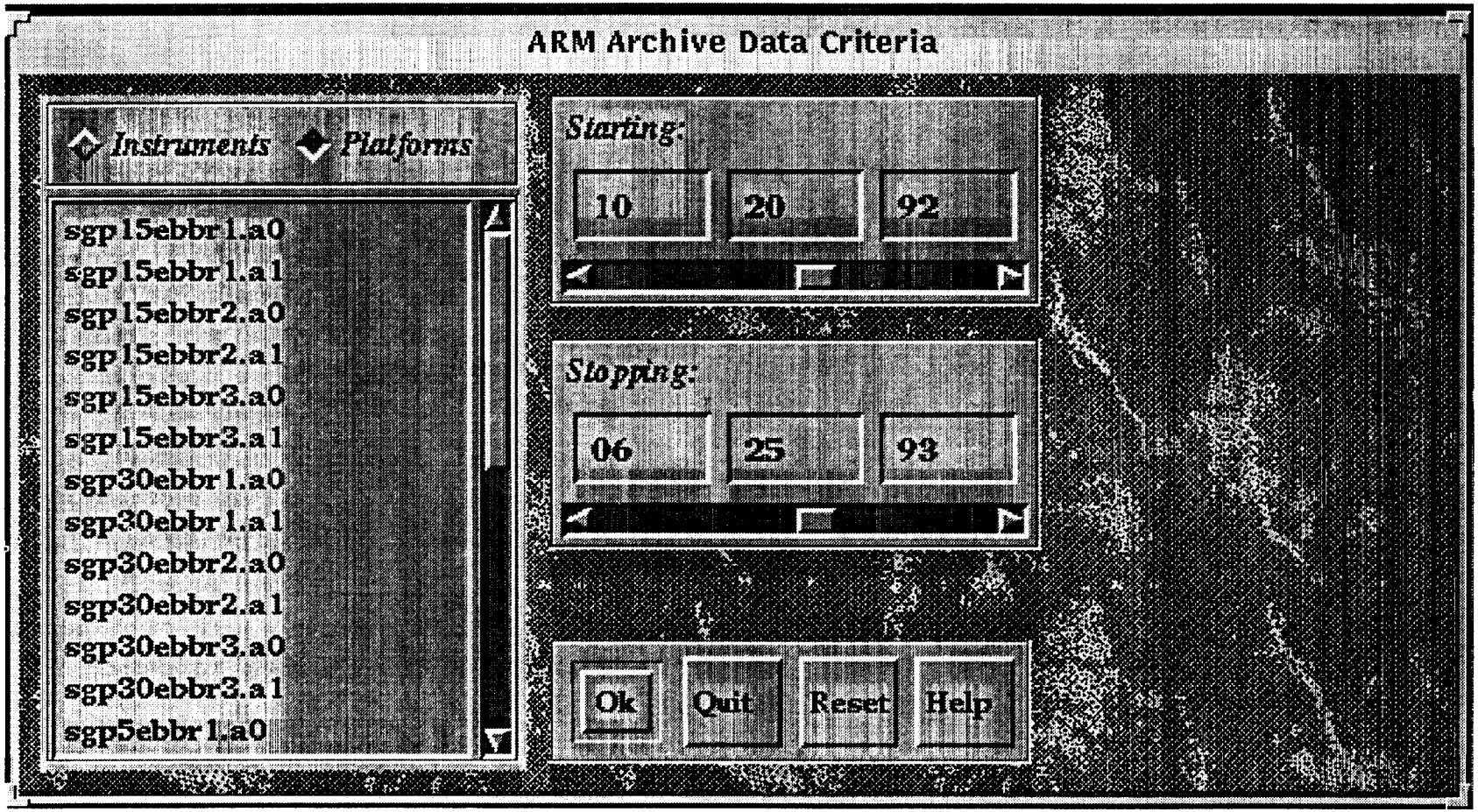
The restricted set of queries that users can presently make (based on the small number of search parameters) is stored as string templates in the client C-program. They are filled in on the basis of the user's choices from the Data Criteria screen, then executed against the metadata data base. When that query is executed, a Transfer Confirmation screen (Fig. 5) displays the number of files (and the total number of bytes of data contained therein) that meet the selection criteria. The user may then choose Initiate or Cancel at this screen. "Initiate" causes the client program to request that the files meeting the selection criteria be retrieved from mass storage and delivered to a pickup directory on the server computer system; "Cancel" returns the user to the Data Criteria screen directly for a new query without requesting data retrieval. Back at the Data Criteria screen, the user may refine the existing query by altering a few values or start afresh (Reset) by clearing all of the settings chosen for the previous query.

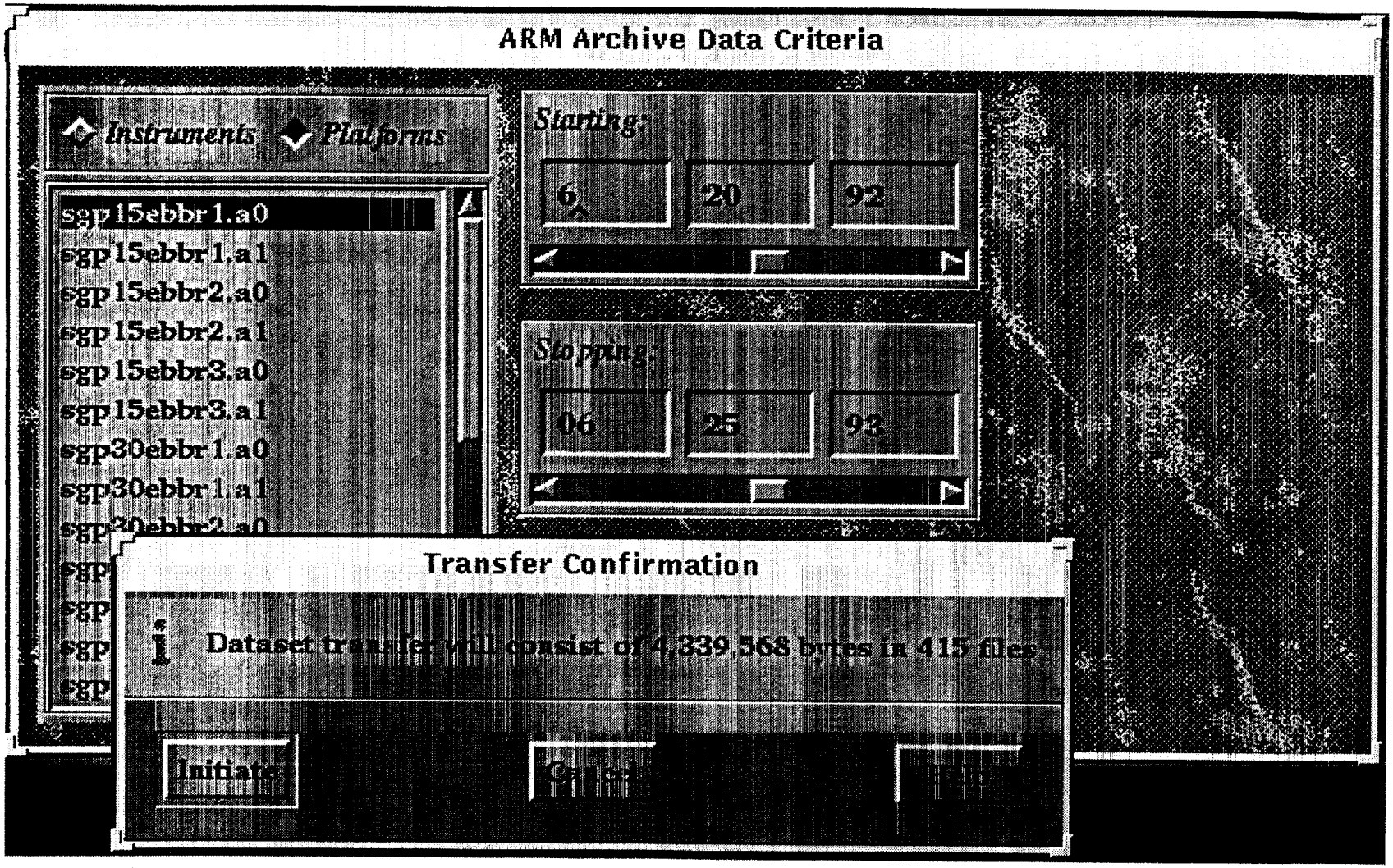
The server portion of our user interface consists of the metadata data base and the file-retrieval and user-notification system. The metadata data base is implemented as a set of tables in the Sybase RDBMS, which are queried by the client program through Sybase-provided library functions. The metadata that pertain to data files stored at the archive are described below; information about user addresses and tables for processing requests for file retrieval are also stored in this data base. When the user initiates a request for data from the client, the file names requested are written into a Sybase table, which is then read by a process that submits a retrieval request to our mass storage system (currently, a Storage Tek Silo managed by an IBM 3090 computer). The files are retrieved via ftp and then put into a holding directory on the host computer to await pickup by the user.

When all of the files requested by a user in a single session have been retrieved from the mass storage system, an electronic-mail message is sent to the user, informing him or her of the availability of the requested data . The user can then copy those data files by ftp to his or her local computer. After an interval of a few days, the files will be deleted from the pickup directory.


User
Selection Criteria
data files meeting selection criteria
User Interface
ftp "get"
"data ready" e-mail message
Client
Selection Query
# of files selected
# of bytes
Server
Names of files to get and where to put them
Metadata Database
Data-pickup Directory
File Retriever
Names of files to get and where to put them
data files meeting selection criteria
Mass Storage System




ARM Archive Data Criteria
Instruments
Platforms
sgp15ebbr1.a0
sgp15ebbr1.a1
sgp15ebbr2.a0
sgp15ebbr2.a1
sgp15ebbr3.a0
sgp15ebbr3.a1
sgp30ebbr1.a0
sgp30ebbr1.a1
sgp30ebbr2.a0
sgp30ebbr2.a1
sgp30ebbr3.a0
sgp30ebbr3.a1
sgp5ebbr1.a0
Starting:
10
20
92
Stopping:
06
25
93
Ok
Quit
Reset
Help


ARM Archive Data Criteria
Instruments
Platforms
sgp15ebbr1.a0
sgp15ebbr1.a1
sgp15ebbr2.a0
sgp15ebbr2.a1
sgp15ebbr3.a0
sgp15ebbr3.a1
sgp30ebbr1.a0
sgp30ebbr1.a1
Starting:
6
20
92
Stopping:
06
25
93
Transfer Confirmation
Dataset transfer will consist of 4,339,568 bytes in 415 files
Initiate

## Metadata for the User Interface and the Archive

As noted above, the primary unit of data given to the user is a single data file, which contains data recorded over a specified time interval from a specified set of sensors (an instrument or platform). In order to allow the user to select the desired files, retrieve them, and use the data properly, we must deal with three classes of metadata: (1) data extracted directly from the individual data files, (2) other data about individual files (e.g., file size and storage location), and (3) site operations logs and other documentary information that are not keyed to a specific data file.

To extract needed metadata from the data files themselves, we have a copy of the suite of programs that produces the NetCDF files for the SDS. We use this code to extract the data-start and data-end dates and times, and the number of samples for each variable from each NetCDF data file, for entry into our data base. As files arrive at the archive, we record their file name, arrival date/time, and file size in our Sybase data base. Further information about storage locations and dates is collected as the files are sent to our mass storage system and as a permanent, vault-archived copy of the files is written by that system. Other metadata include the Site Operations Log, platform and data dictionaries for instruments, and other textual information that may affect the correctness or usability of data files but are not directly linked to them. The Site Ops Log is being stored in a table in the metadata data base (based on date of entry), as well as being archived as entries arrive. The other text files will be managed (and accessed by the users) through the WAIS system, which is specifically designed to allow browsing of large free-text data files for keywords.

The current stock of metadata may be significantly expanded as the ARM Project continues. As discussed above, all of the header information in the NetCDF files from the SDS could be incorporated into the metadata data base. We are also starting to explore schemes to allow users or persons responsible for specific instruments to comment on data files, perhaps at the single data point level. (In this case, we intend for users requesting data with existing comments to receive the comments along with the data files.) We further expect that some metadata will arrive in a form that is not computer-readable, and we are pondering our response.

## Future Improvements to the User Interface

To make the ARM Archive user interface as helpful as possible, the capabilities that have been discussed in this paper must be extended to provide the users with more information about the available data. In addition to making more information available about the holdings of the ARM Archive in general, more selection criteria need to be available for the users to refine their requests for data.

Additional selection criteria will be derived from the formal metadata transmitted with the data from the CART sites. As with the current selection criteria of platform and date, this data will be managed with the use of the metadata RDBMS. In order to implement additional selection criteria, we need to work with the user community to identify the useful selection criteria for each platform and to extract that metadata from the data files and place it in the metadata data base. We also envision a desire to select data files for one platform on the basis of the availability of data from another platform for the same time interval: "Give me the BSRN1 data for June 1993 where EBBR9 data exist," for example. The user interface is designed so that new selection criteria can be easily added to the user interface screens.

Most of the information about the ARM Archive that explains the contents to users is in textual form. If ARM data are to be accessible to relatively naive users, this textual information must be made available on-line. The current plans are to manage textual information about the ARM Project with a WAIS server. A WAIS client will become part of the client portion of the user interface to make all ARM text available for perusal and downloading through the user interface.

Several potential metadata sources, such as operators' descriptions of instrument status in the site operations log, are textual with little formal structure. This type of information can be critical to the user in deciding if a particular data file is desired or not. In order to make text information part of the selection parameters, links need to be developed between the metadata kept in the RDBMS and those kept in the WAIS system. We are developing a design to provide this connection using a common identifier in the RDBMS records and the text record that will allow textual information as part of the selection criteria for requesting data files.

As a final assistance to users in selecting data, we are exploring the possibility of logically linking textual comments to data files. The proposed implementation would allow the users to see brief comments on the data files that they are about to request. On the basis of those comments, they might elect to remove some data files from their request. The comments would deal with data quality and use issues that were not captured in other parts of the metadata system. One proposal is that some of these comments might be from previous users of the data.

## Conclusions

Scientific data are useful only when they are producing scientific or policy results. The ARM Archive user interface is designed to make the ARM data quickly and easily available to the user community. To accomplish this goal, the user interface will provide the users with information in the terms of the atmospheric science discipline. Over time, it will also provide users with extensive documentation about the condition of the data, why and how the data were collected, and other information to make data selection and use easier.

Metadata that provide clear, accurate, and precise information about the context of the basic data are necessary to support this type of user interface. The ARM metadata are being organized with the use of an RDBMS for data that are very formally organized and lend themselves to management in tabular format. For those data that are textual and do not easily fit the row/column format, a WAIS system will be used for data management and access. In the future the ARM Archive will explore ways to link the metadata in the RDBMS and WAIS systems to provide users with both a rich set of selection parameters and concise descriptions of the data they are requesting.

# A Parallel Data Management System for Large-Scale NASA Datasets

Jaideep Srivastava

*Computer Science Department*
*4-192 EECS Building, 200 Union St. S.E.*
*University of Minnesota*
*Minneapolis, MN 55455*
*srivasta@cs.umn.edu; (612)625-4012*

**Abstract**

The past decade has experienced a phenomenal growth in the amount of data and resultant information generated by NASA's operations and research projects. A key application is the *reprocessing problem* which has been identified to require data management capabilities beyond those available today [PRAT93]. The Intelligent Information Fusion (IIF) system [ROEL91] is an ongoing NASA project which has similar requirements. Deriving our understanding of NASA's future data management needs based on the above, this paper describes an approach to using parallel computer systems (processor and I/O architectures) to develop an efficient parallel database management system to address these needs. Specifically, we propose to investigate issues in low-level record organization and management, complex query processing, and query compilation and scheduling.

## 1. Problem Understanding: NASA's Future Data Management Needs

The past decade has experienced a phenomenal growth in the amount of raw data and resultant information generated by NASA's operations and research projects [ROEL91]. The need for significant improvement in information technologies to manage, identify, and access this data has been clearly identified [ROEL91, CROM92, CAMP90a, CAMP90b]. This section present's our view of NASA's future data management needs (at least in part). It is based on (i) the description of the *reprocessing problem* given in [PRAT93], (ii) published descriptions of the *Intelligent Information Fusion (IIF)* system [ROEL91], and (iii) miscellaneous NASA publications.

### 1.1 A View of NASA's Data Management Architecture

Figure 1 shows the schematic of a system architecture where the principal emphasis is on the path data takes, and the transformations it goes through, from sensor collection to the scientific user. This architecture borrows from that of the IIF system [ROEL91]. The aim of this diagram is principally for problem understanding purposes and to establish a context for the subsequent discussion. It is by no means a proposal of what the complete architecture for NASA's data management system should be, and is much wider in scope than that of the present paper.

Sensor data first goes through some very low-level processing to generate 'raw data' [PRAT93] which is stored in a Parallel Raw Data Archive (PRDA). The reprocessing activity creates 'data products' [PRAT93] which are managed by a Parallel Relational Database Management System (PRDBMS). Metadata about both raw data and data products is stored in a Metadata Database (MDB). The three different types of data stores, i.e. the PRDA, PRDB, and MDB, reflect the three basically different types of usage of the data and metadata in such an environment [PRAT93]. The raw data is expected to be used mostly by reprocessing algorithms running on vector supercomputers and massively parallel processors (MPPs), and hence is shown managed by a high-performance file system. Since existing data products can also be inputs to the reprocessing activity [PRAT93], direct access to the Parallel Record Management Layer (PRML) of the PRDBMS by the machines running the reprocessing algorithms is shown. A typical user of the data products is a remote scientist who logs in and *browses* the metadata searching for data relevant to a research project. While most browsing involves interaction with the metadata, the scientist may periodically access data products as well as raw data to identify interesting data. Upon selecting

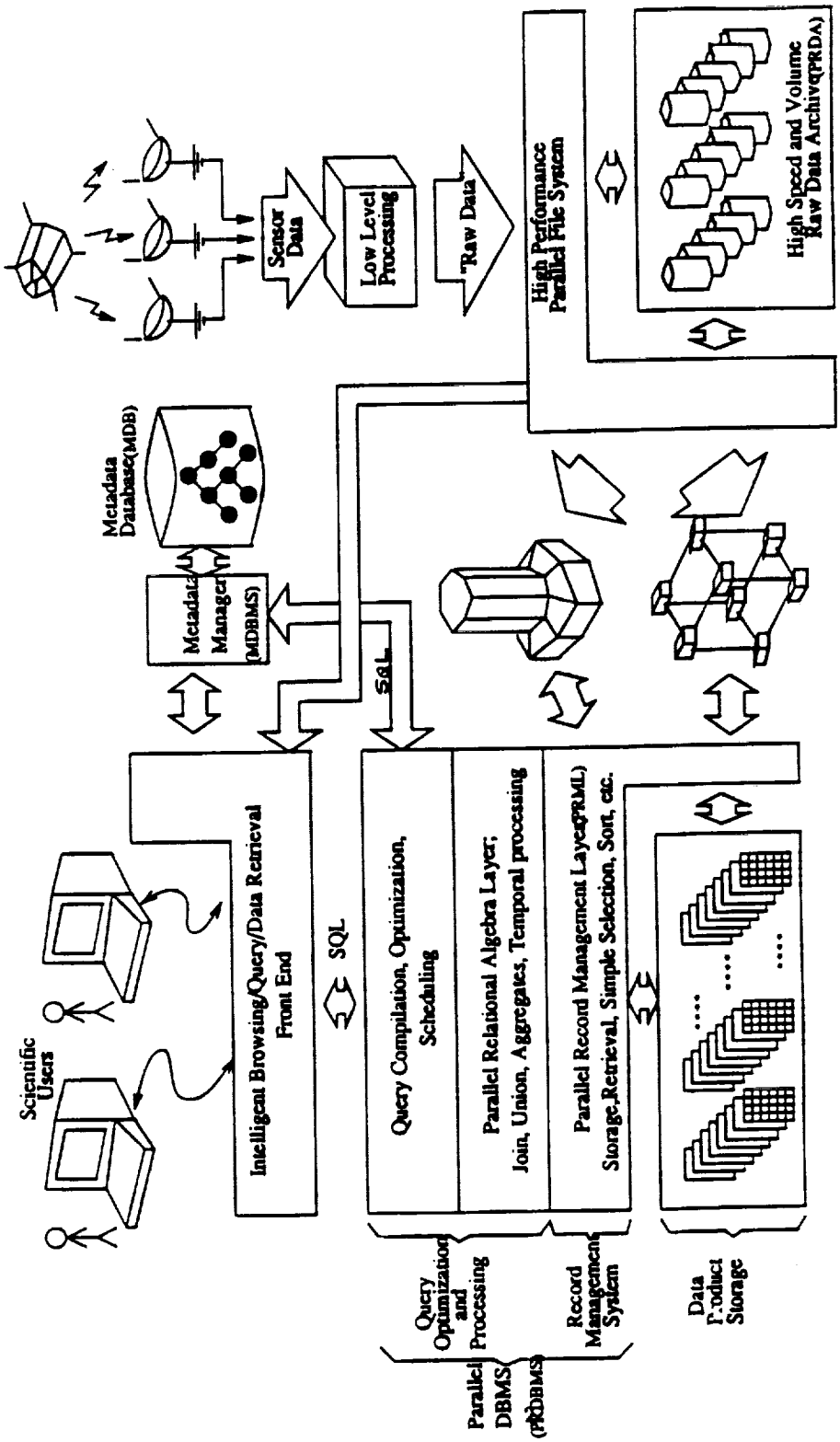



**Figure 1. Our View of NASA Data Management Architecture**

the needed data, the appropriate portion is downloaded into the scientists home location. To support such pattern of user behaviour the MDBMS should support large numbers of interactive browsing sessions, each posing mostly small queries against the MDB, interspersed with occasional queries against the PRDA or PRDB. While interactive response time is needed, the bandwidth required is expected to be small during the browsing. Once browsing is complete, the user will issue a series of requests to extract the data to be downloaded to his home location. These requests can be SQL queries to the PRDBMS or file access requests to the PRDA. These requests are expected to have high bandwidth requirements since a large volume of data may be extracted. Since execution times of different plans for a SQL query can differ by a few orders of magnitude, query optimization is critical to ensure both interactive response time and reduced system workload.

## 1.2 Parallel I/O: Key to NASA's Data Management

Given the volume of data/information in NASA's applications, the use of multiple disks for storage is well accepted. In a database processing environment, the fact that disk I/O is the main bottleneck has been a consensus among researchers. Recent years have seen phenomenal increase in processor speeds, while the 'disk access time' has not shown much improvement, exacerbating the 'access gap' problem. The advent of multiple processor machines has added to this problem. Fortunately, the computer architecture community has started addressing the needs of data intensive applications by developing parallel I/O architectures, e.g. Redundant Array of Inexpensive Disks (RAID) [PATT88] and Disk Arrays [GORD91]. This promises future parallel I/O systems which can feed data to the multiprocessor at a high sustained bandwidth.

Along with the development of parallel I/O hardware, there is a need to develop efficient parallel I/O algorithms to exploit their full potential. The main focus of research in parallel algorithms has been on main memory resident data, where processor parallelism has been of primary concern [LEWI92]. With I/O bandwidth being a principal concern, high performance parallel databases require parallel algorithms for disk resident data. Parallel processing of database operations was first addressed by the database machine community, where the focus was on designing special-purpose hardware [SU86]. No single architecture was found suitable for all database applications, and the cost of building special purpose hardware for specific applications led to only limited success in this direction [DeWI92]. In the past few years there has been renewed interest in looking at database issues for general purpose parallel machines. The availability of a variety of commercial parallel machines, which has eliminated the expense of building special purpose hardware, is in large measure responsible for this [DeWI92].

A crucial factor in our choice of the relational model for the PRDBMS component of the architecture in Figure 1 is that the set-oriented, non-procedural nature of the relational model provides opportunities for massive parallelization [DeWIT92]. This choice is further supported by the fact that the IIF system has already proposed using a relational DBMS for its low-level record management system (LLRMS) [ROEL91].

## 1.3 Scope of Our Project

Realization of the architecture shown in Figure 1 is a major task and requires research and development in many areas. The scope of our project is limited to addressing problems in the PRDBMS component of the system. Specifically, we address the following problems:

- Data organization, loading, sorting, and retrieval, and index creation and maintenance, in the Parallel Record Management Layer. The proposed solutions must consider that access requests to this layer will be a mix of (i) very high rate of large size access requests from the reprocessing algorithms, and (ii) low to medium to sometimes large size requests from the upper layers of PRDBMS.

- Parallel algorithms to support expensive operations, e.g. join, union, etc., in the Parallel Relational Algebra Layer.

- Compilation and optimization of SQL queries, and resource allocation and scheduling of operators in the resultant plan. Minimizing response time and maximizing throughput will be considered as the optimization criteria.

The rest of the paper is organized as follows: Section 2 presents the technical details of our approach. Section 3 presents a list of goals that must be met, including specific technical problems that must be solved, to make such a system a reality. Section 4 provides the conclusions and section 5 contains the list of references.

## 2. Technical Details of the Proposed Approach

Our overall goal is to investigate techniques for building a parallel database engine which could fulfill the needs of the PRDBMS component of Figure 1. Following are the key ideas behind our approach:

- Tuples in a relation (or records in a file) are modeled as points in a multi-dimensional space, with each attribute representing an axis.
- This multi-dimensional space can be divided into (overlapping or non-overlapping, nested or non-nested) subdivisions.
- The subdivisions are allocated to different I/O units (e.g. disks) of a parallel computer, with usually many subdivisions going to a single unit, and possibly a single subdivision replicated on multiple units for reliability. This has been termed *declustering* [DEWI90]. The aim is to provide good (close to optimal) load-balancing for query processing.
- New *declustering-aware* parallel algorithms for basic data retrieval operations, e.g. relation/file scan, as well as complex operations, e.g. join and sort, are built to take advantage of the underlying declustering.
- The query compiler/parallelizer/scheduler takes considers architectural parameters and declustering information, in addition to the traditional query and database parameters, in minimizing the execution plan cost. In addition, it generates an initial resource allocation schedule for plan execution.

The remainder of this section is organized as follows: Section 2.1 presents an architecture for the PRDBMS. Sections 2.2 through 2.4 describe our approach to solving specific problems in the *record management*, *relational algebra*, and *query compilation* layers of the PRDBMS.

### 2.1 Parallel RDBMS Architecture

As shown in Figure 1, the PRDBMS has a layered architecture. The *parallel record management layer* provides the abstraction of relations/tables which can be created, deleted, populated, sorted, and on which simple selections (predicates involving single relations only) can be performed. This abstraction is used both by the higher layers of the PRDBMS and by the reprocessing algorithms. The *parallel relational algebra layer* contains algorithms for complex operations such as join, union, difference, aggregation, etc. It uses the abstractions provided by the record management layer. The *query compilation layer* provides a declarative interface (SQL) to PRDBMS users (the intelligent front-end and metadata manager in Figure 1), and does the necessary translation and optimization of declarative queries into a sequence of relational algebra operations.

### 2.2 Parallel Record Management Layer

The parallel record management layer uses the services offered by the operating system to provide an abstraction of relations/tables containing records.

#### 2.2.1 Requirements

We first identify the characteristics of data stored in the record management system as well as of the retrieval requests on it. Datasets for many large-scale scientific applications, including those of NASA, exhibit the following characteristics [ROEL91, CAMP90a]:

- The basic data unit is an observation, e.g. from a satellite, with various attributes such as latitude, longitude, temperature, time, etc.
- The data is multi-dimensional, e.g. the three spatial dimensions, the temporal dimension, and various other attributes.
- The database is fairly stationary, i.e. new data can be appended or results of analyses can be added. However, the basic data once added is rarely, if ever, updated.
- High speed and volume of reprocessing requires support for efficient creation and population of relations, both in terms of bandwidth and response time.
- A very high rate of large size retrieval requests is expected from reprocessing algorithms. Large size requests are also expected from the intelligent front-end working on the users' behalf, albeit not at quite the same rate as reprocessing algorithms (though it really depends on user load).

### 2.2.2 Approach

In the following we describe our approach to the specific problems listed below. Comparisons with related work are included where appropriate.

- Data declustering, i.e. partitioning a file of records across multiple disks of a parallel I/O system.
- Parallel algorithms for range query processing on a single relation/table.
- Parallel algorithms for loading large data files into relations/tables.

Unit datum is modeled as a tuple/record whose attributes/fields represent various facets of the datum such as latitude, longitude, temperature, time, etc. Relations/Files, i.e. a collection of records of the same type, model sets of observations of the same type. A general request on a collection of observations of the same type is modeled as a multi-attribute range query, with predicates defined on one or more attributes.

Let $D_i$ $(1 \leq i \leq d)$ be an ordered set. A *record* is an ordered d-tuple $(r_1, r_2, ..., r_d) \in D_1 \times D_2 \times ... \times D_d$. $D_i$ is defined to be the domain of the $i^{th}$ attribute, and $r_i$ is the value of the $i^{th}$ attribute of the record. A d-dimensional file, $F$, is a non-empty set of records, stored on a parallel disk system with $M$ disks.

The most general retrieval operation, the range query, is denoted by $Q = ([L_1, U_1), ..., [L_d, U_d))$, with $[L_i, U_i)$ being the desired range on the $i^{th}$ attribute. The answer to the range query $Q$ is $A(Q) = \{(r_1, ..., r_d) \; F \mid L_i \leq r_i < U_i, \; 1 \leq i \leq d\}$. Note that the exact-match query and the partial-match query can be treated as special cases of the range query. For a query $Q$, let $Work_i(Q)$ be the number of blocks required from disk $i$ to answer the query, $1 \leq i \leq M$, and let $Work(Q) = \sum_{1 \leq i \leq M} Work_i(Q)$ be its total work. Assuming parallel operation of individual disk units, and the performance of the I/O subsystem being the critical factor in system performance - which is a reasonable assumption given trends in parallel machines, the response time of the query is $Rsp(Q) = MAX_{1 \leq i \leq M} \{Work_i(Q)\}$. The optimal (minimal) response time for the query $Q$ by distributing data over $M$ disks is then $\lceil Work(Q)/M \rceil$.

Now, the data declustering problem for a parallel record management system is to develop a strategy such that it provides (i) optimal parallelization of individual queries (*speed-up*) as well as (ii) good parallelization of all possible queries (*robustness*). In the last few years a number of declustering strategies have been proposed [DeWI90, GHAN91, GHAN92, HUA91 LI92, FALO93, ABDE93]. A survey of a some of these is given in [DeWI92]. The focus of [DeWI90, GHAN91] is to decluster based on a single attribute, thus improving performance only of queries containing a predicate on that attribute. [GHAN92]

improves upon their previous proposal by selecting a *typical query* and using information about it to improve declustering. [HUA91] considers multiple attributes but optimality is not addressed. [ABDE93, FALO93] identify specific subsets of queries for which their schemes have optimal performance, but the issue of robustness is not addressed. Our work [LI92] has developed the *Co-ordinate Modulo Declustering (CMD)* techniques (i) which is optimal for a very large percentage of all possible single relation SQL queries, (ii) has a small deviation from optimality for the rest, and (iii) whose deviation from optimality decreases as the size of the query result grows. Complete details of CMD and its comparison with other schemes is given in [LI92]. Here we provide a brief overview.

For illustration assume that all files are subsets of the unit space $S = [0, 1)^d$, $d \geq 1$. Divide each dimension of $S$ into $nM$ equal sized intervals for some integer $n$:

$$[0, 1/nM), [1/nM, 2/nM), ..., [(nM-1)/nM, 1).$$

Let the $i^{th}$ interval of the $k^{th}$ dimension be denoted by $I_{ki} = [l_{ki}, h_{ki}) = [i/nM, (i+1)/nM)$, for $0 \leq i \leq nM-1$, with its *interval coordinate*, $ic_k$, being $i$. Given a region $[l_{1i_1}, h_{1i_1}) \times [l_{2i_2}, h_{2i_2}) \times ... \times [l_{di_d}, h_{di_d})$ of $S$, its *region coordinate*, $rc$, is defined to as an ordered set of its interval coordinates, i.e. $rc = (ic_1, ic_2, ..., ic_d)$. Now, a region, i.e. partition of the multi-dimensional data space, with region coordinate $rc$ is assigned to disk $CMD(rc, M)$, where the allocation function $CMD$ is defined as:

$$CMD(rc, M) = (ic_1 + ic_2 + ... + ic_d) \ mod \ M.$$

**Example 1:** Let $S = [0,1)^2$, $M = 4$ and $n = 2$, i.e. each dimension is divided into 8 intervals with length 0.125 each. The partitions of $S$ and their allocation to disk units is shown in Fig. 2.

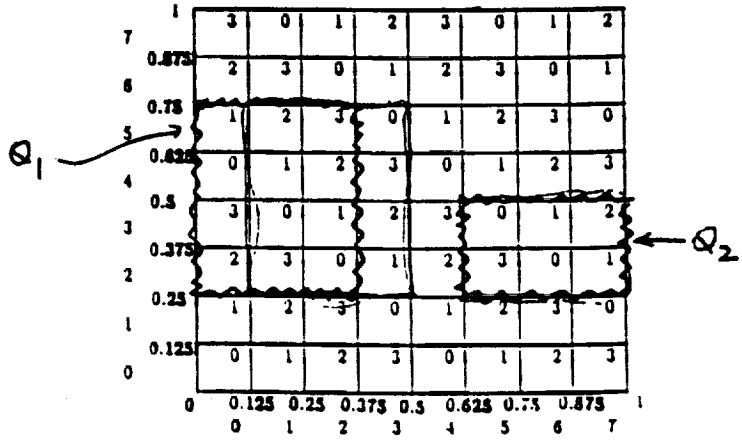



**Figure 2. The partition and allocation of S = [0, 1) x [0,1) among 4 disks with M = 4 and n = 2**

We have developed parallel algorithms for multi-dimensional range queries on data with CMD partitioning. The following theorems describe the key properties of the algorithms. Proofs are given in [LI92].

**Theorem 1** (*Speedup*): The $CMD$ method is optimal for all range queries whose length, in terms of the number of regions covered, on some dimension is equal to $kM$ where $k$ is an integer.

**Corollary 2.1.** The $CMD$ method is optimal for all range queries in which at least one attribute is unspecified (since the query length on that attribute is the complete range, automatically an integral multiple of $M$).

**Example 2:** Consider query $Q_1 = ([0.000, 0.375), [0.250, 0.750))$ in Figure 2. Assuming each region can be fetched in a single disk access, $Work(Q_1) = 12$ disk accesses. Since exactly 3 accesses need to be made to each of the disks $0, 1, 2, 3$, the response time for $Q_1$ is optimal. The condition

in Theorem 1 is sufficient but not necessary since optimal response time is also achieved for query $Q_2 = ([0.625, 1.000), [0.250, 0.500))$.

**Theorem 2** (*Robustness*): For any arbitrary range query $Q$ the response time, $Rsp(Q)$, is bounded by $\lceil Work(Q)/M \rceil + (M-1)^{d-1} - 1$.

Theorem 2 gives an approximate upper bound, and the actual performance of $CMD$ is much better. For example, for 2 and 3 dimensions the worst case upper bounds are $M/4$ and $M^2/16$, respectively. Note that range queries usually examine a very large subspace of $S$, i.e. $Work(Q)$ is usually large. Thus $\lceil Work(Q)/M \rceil$, the fraction that is optimal, is much more significant than $(M-1)^{d-1} - 1$.

**Parallel Data Loading Algorithms:** Our recent work [LI93] is developing efficient parallel algorithms for loading files of records into a CMD format. Initial results show that almost linear speedup of the process, in terms of the number of disk units, is achievable. Detailed algorithms and their properties are discussed in [LI93].

## 2.3 Parallel Relational Algebra Layer

The *parallel relational algebra layer* contains algorithms for complex operations such as join, union, difference, and aggregation. It uses the abstractions provided by the parallel record management layer.

### 2.3.1 Requirements

Descriptions of various NASA projects, including the *Intelligent Information Fusion (IIF)* system [ROEL91, CAMP90a], the *Intelligent User Interface for Catalog Browsing* system [CROM89], etc., have identified the need for performing complex comparisons across different types of data sets. Thus, the requirements for this layer are:

- Efficient algorithms for complex operations such as join, union, set difference, etc.
- Efficient algorithms for various kinds of aggregate operations.
- Since space and time are special types of attributes, correlations on them can be potentially treated in a more specialized and efficient manner, e.g. by supporting temporal joins [McKE92].

### 2.3.2 Approach

In the previous section we presented results about the efficacy of the CMD approach in processing queries accessing a single relation. A vast body of work [DeWI92, WOLF91, FRIE90, SCHN89, DeWI92, CHEN92, NICC92] has shown that join continues to be one of the most expensive relational operations in the parallel environment. Our recent work [NICC92] has shown that an approach to achieving efficiency for complex database operations in a parallel environment is to make them *declustering aware*, i.e. an algorithm implementing a complex operation (e.g. join) will perform better if it is aware of the underlying declustering strategy. [NICC92] describes and analyzes in detail the benefits of making hybrid-hash join algorithm [DeWI84] aware of CMD declustering. We outline the approach here.

For a relation stored using CMD declustering, we define the following:

**Definition** (*Join Axis, b*): The axis of the multi-dimensional space representing the join attribute 'b'.

Each interval $(l_i, h_i)$ along the join axis denotes a subrange of the join attribute domain.

**Definition** (*Joining Region, JR(R,B,i)*): The $d-1$ dimensional subspace, of the $d$ dimensional space, created by fixing the subrange of the join axis, $b$, to have values in the interval $(l_i, h_i)$ and allowing the other axes to be free.

$JR_{R_{a_i}}$ is the $i^th$ joining region of relation $R$ along attribute axis $a$.

As shown in Figure 3(a), consider $R$ and $S$ as relations to be joined on attribute $b$. $JR(R, b, 2)$ and $JR(S, b, 1)$ are example joining regions of relations $R$ and $S$, respectively. A joining region of $R$ must join with every joining region of $S$ with which it overlaps on the join axis. Thus, $JR(R, b, i)$ and $JR(S, b, j)$ must be joined *iff*:

$$(l_i \leq l_j \leq h_i) \mathbf{or} (l_i \leq h_j \leq h_i)$$

The following results describe the properties of our declustering aware approach, details of which are presented in [NICC92]:

**Theorem 3**: If there is enough aggregate buffer memory, i.e. among all processors together, to hold the largest joining region of the smaller relation, plus one disk block per processor, then no data need be read from the I/O system more than once.

**Corollary**: There exist cases where declustering aware algorithm will read a disk block exactly once while a non declustering aware algorithm will read it more than once.

In addition to reducing disk accesses, a declustering aware algorithm may entirely eliminate part of the computation, by skipping over entire joining regions of either relation, if there is no intersecting joining region of the other relation, as shown in Figure 3(b). Essentially, a declustering-aware approach to query processing has the following advantages:

- A large problem is broken down into a set of subproblems, such that the sum of the work for the set of subproblems is usually lesser than that for the original. For example, the work for an equi-join between relations $R$ and $S$, with sizes $|R|$ and $|S|$ respectively, is roughly proportional to $|R||S|$, say with a nested-loops join. If, however, the join axis has $k$ partitions, a declustering-aware nested-loops algorithm is required to do only $k(|R||S|)/k^2$ total work for the $k$ subproblems.
- The performance of most database algorithms, e.g. join, sort, etc., is highly sensitive to the amount of main memory buffer available, with performance often increasing dramatically as the ratio $BufferSize/ProblemSize$ increases [CHOU86, YU93]. For a given amount of aggregate main memory buffer (of the parallel machine), breaking a problem into smaller subproblems has the net effect of increasing this ratio.
- Skewed data distribution causes serious performance problems for most database algorithms [DeWI92a DeWI92b], mainly due to improper load balancing. Declustering aware algorithms provide one way to handle this [NICC92].

## 2.4 Parallel Query Compilation and Scheduling Layer

Database query compilation for sequential machines provides the functionality of translating a high-level (declarative) query into an optimized sequence of relational algebra and record management level operations. For a parallel machine, the additional decisions of (i) determining the type and degree of parallelization, (ii) an estimation of resource requirements, and (iii) an initial assignment of resources, must be made [GANG92, SRIV93].

### 2.4.1 Requirements

Descriptions of various NASA projects, including the *Intelligent Information Fusion (IIF)* system [ROEL91, CAMP90a], the *Intelligent User Interface for Catalog Browsing* system [CROM89], etc., have identified that the interface between the applications, e.g. intelligent front-end of Figure 1, and the database of data products be a high-level one, e.g. SQL. Query compilation and scheduling for parallel databases is currently an active research area [DeWI92a, WILS91, GANG92, SCHN90, HUA93, SRIV93, NICC93]. While detailed survey and comparisons are provided in [SRIV93, NICC93], the basic requirements for this layer are:

- Translation from SQL to an internal form (not a research issue).
- Optimizations performed on the internal form based on the desired objective, e.g. minimize work, minimize response time, etc., to generate a 'good' query execution plan.

- Determining the type(s) and degree of parallelization of the query plan.
- Estimation of resource needs for a query plan to help resource managers during query execution.
- Determining an initial resource allocation for the plan, which may potentially be modified during execution.

### 2.4.2 Approach

Our overall approach to query compilation is shown in Figure 4. It is a 2-phase approach, where in Phase 1 a compiler that optimizes SQL for sequential machines is used, which (heuristically) minimizes work. *This is not a research issue since good sequential optimizers exist.* The output is fed to Phase 2 which (i) parallelizes the sequential plan, (ii) estimates its resource needs, and (iii) generates an initial resource allocation schedule. The output of Phase 2 is a set of tasks schedulable on a parallel machine. An example input query, represented as a query graph, and its corresponding set of tasks, $t_1$ through $t_{11}$, is shown in Figure 5. In each of the seven time slices, numbered 0 through 6, the total resources allocated for this query's execution are shared between the tasks allocated to the slice. Further details are in [NICC93]. While in general it is not true that the parallelization of a 'good' (or even the optimal) sequential plan will yield the best parallel plan, a 2-phase approach such as ours has the advantages of (i) drastically reducing the search space size, and (ii) leveraging off the existing technology in sequential optimization. We share the belief with [STON88, HONG91, HONG92] that a 2-phase approach is a viable heuristic and worth a detailed investigation.

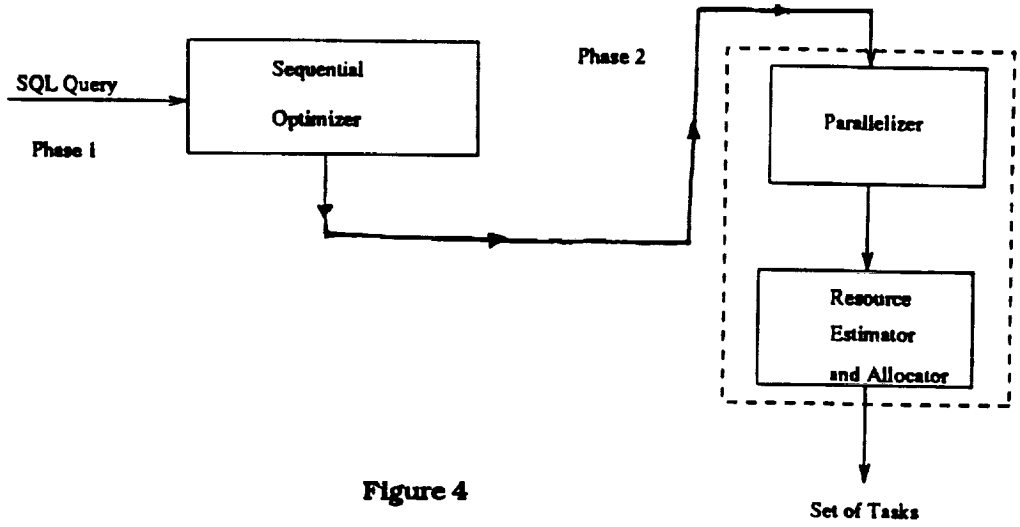



**Figure 4**

We now briefly describe the key elements of our approach to query compilation and scheduling. Details are provided mainly in [NICC93] and some in [SRIV93]. Specifically, we propose (i) a parallel query plan representation, (ii) a new cost model to incorporate parallel execution, and (iii) heuristic search algorithms.

**Query Plan Representation:** A parallel query plan can exploit the following kinds of parallelism:

- *Intra-operator parallelism*: A relational operator, such as select, project or join, can be performed by multiple processors simultaneously.
- *Inter-operator parallelism*: Different relational operators of a query, eg. different joins, can be performed in parallel by different (sets of) processors.
- *Pipelining*: Different relational operators can be performed in a pipelined manner using separate (groups of) processors. The result of one is pipelined to the other.

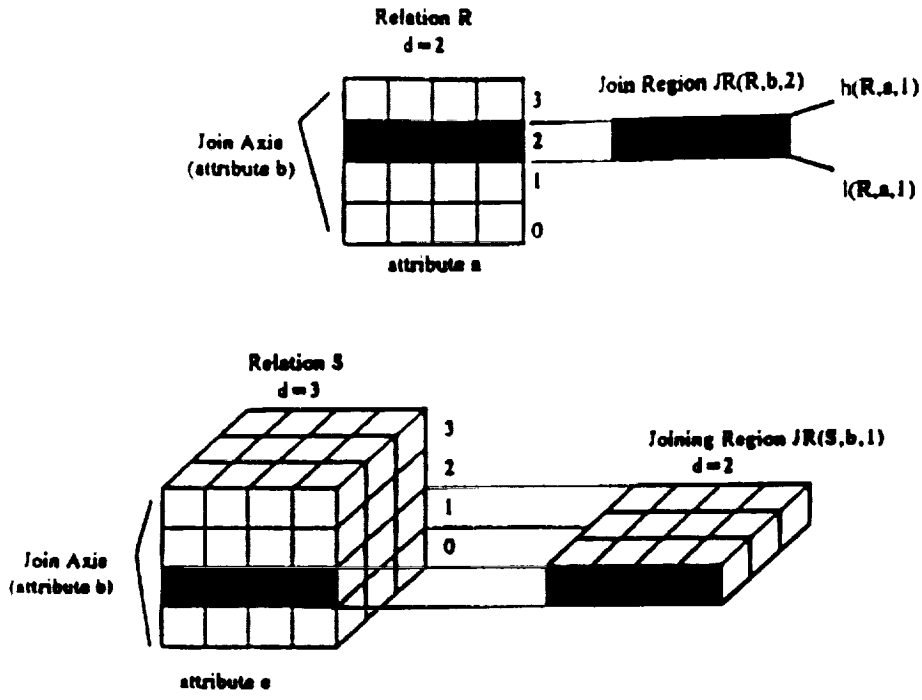



**Figure 3 a.- 3-Dimensional Relation and a Single Join Region**

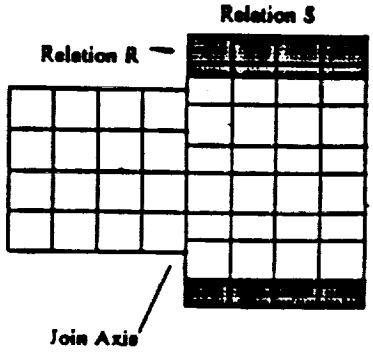



**Figure 3 b - A join of situation where some partitions can be ignored**

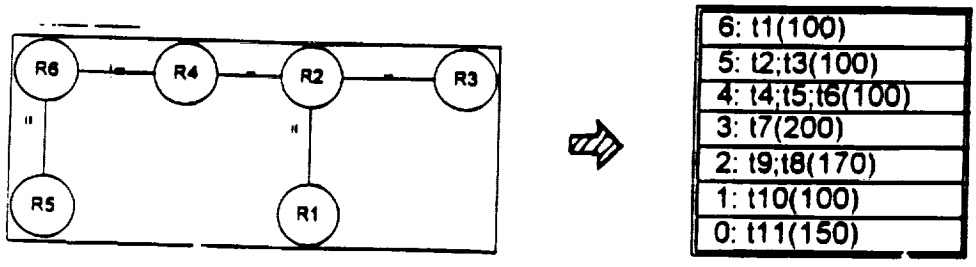



**Figure 5**

In our model a parallel query plan is represented as a *capacitated labeled ordered binary tree.* The shape represents inter-operator parallelism, the orientation represents operand ordering, the node labeling represents intra-operator parallelism, the M (P) branch labeling represents materialization (pipelining) of results between operators, and the branch capacity represents the size of the main-memory (producer-consumer) buffer when materialization (pipelining) of intermediate results is being done.

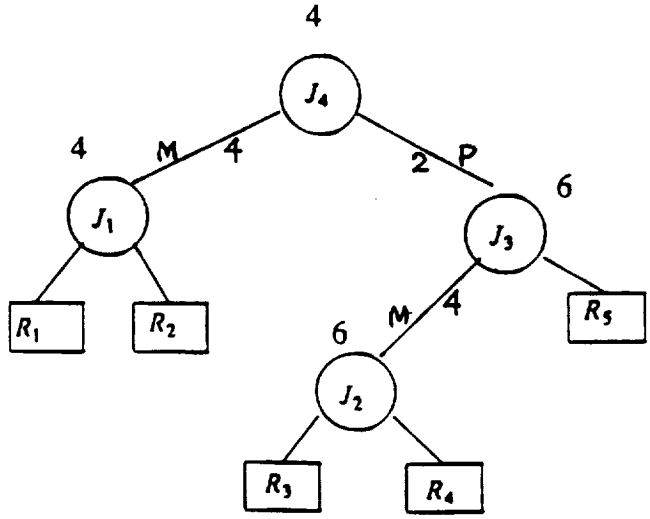



**Figure 6**

Figure 6 shows a plan for a query with four joins, i.e. $J_1$, $J_2$, $J_3$ and $J_4$, between five relations, i.e. $R_1$, $R_2$, $R_3$, $R_4$ and $R_5$. $J_1$ has inter-operator parallelism with $J_2$ (and with $J_3$). Operations $J_1$ and $J_4$ are on a root-leaf path and thus do not have inter-operator parallelism. The same holds between $J_2$, $J_3$ and $J_4$. Since the branch between $J_1$ and $J_4$ is labeled with $M$, $J_1$ must complete *before* $J_4$ can begin. The same holds for $J_2$ and $J_3$. The branch between $J_3$ and $J_4$ is labeled $P$, and thus the two are pipelined, with $J_4$ beginning as soon as $J_3$ has produced the first result tuple. The labels 4, 4, 6 and 6 on $J_1$, $J_4$, $J_2$ and $J_3$, respectively, represent the number of processors assigned to each. Note that the processors assigned to operators at the opposite ends of a branch labeled $M$ are the same set, i.e. they first perform the child task and then proceed to the parent task. The processors on the opposite ends of a branch labeled $P$ are distinct sets since the operations are pipelined. The 4 processors will first perform the join $J_1$ and then $J_4$. The 6 processors will first perform the join $J_2$ and then $J_3$. The two processor sets will be working independently while performing the joins $J_1$ and $J_2$. While performing $J_3$ and $J_4$, the 6 processor set will be the producer while the 4 processor set will be the consumer. The capacity of 2 on the branch $(J_4, J_3)$ means that the intermediate buffer is assigned 2 units of memory. A capacity of 4 on the other branches indicates that each materialized intermediate result has been assigned 4 units of buffer space. Upon overflow the results must go to disk. Total system memory is 10 units.

**Cost Model for Parallel Query Plans:** A cost model for parallel query plans requires (i) developing analytical cost expressions for individual operators such as select, project, join, etc., and (ii) combining the expressions for individual operators to obtain costs for entire plans. Special care has to be paid in combining costs for operators executing in a parallel or pipelined manner. The two key components are:

- *Cost of Individual Operators*: A number of simulation and experimental evaluations of parallel algorithms for relational operators exist [DeWI90, BARU88, SCHN89, FRIE90]. For query optimization, however, an analytical parameterized cost model is needed. In addition to conventional parameters such as database size, query selectivity, indexes, algorithm used, etc., the cost of an operator depends on (i) its degree of parallelization, (ii) its resource allocation, (iii) parameters of the machine architecture, e.g. costs for unit processing, I/O, and communication operations, and (iv) data declustering.

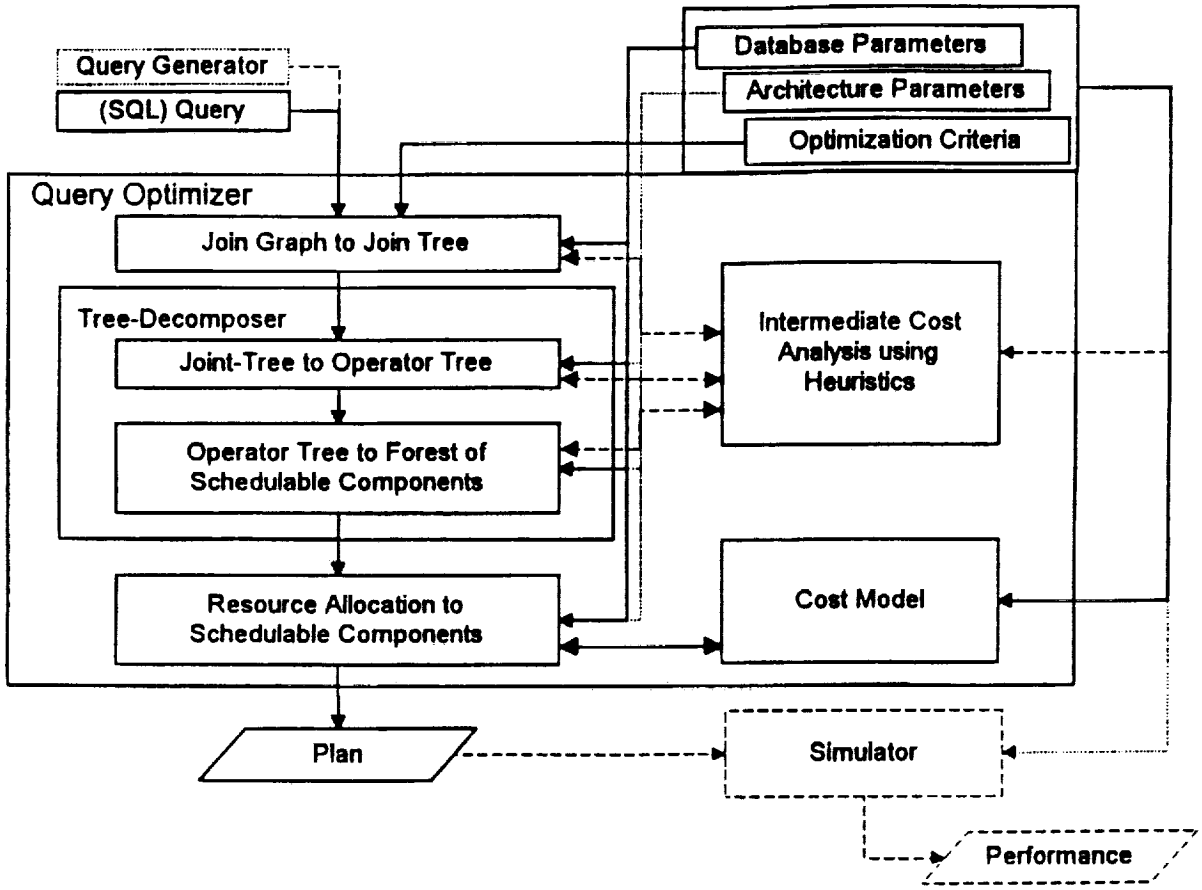



**Figure 7. Architecture of an Optimizer**

- *Combining Operator Costs*: For parallel query processing the plan with total minimum work and the one with the shortest critical path may not be identical [GUST89]. Maximizing overall throughput in a multiprogrammed environment requires minimizing a query's total work, while minimizing individual response time requires reducing the critical path. Calculating the critical path in a plan can be quite tricky as it needs to consider data flow dependencies and resource allocation [GANG92, SRIV93, NICC93].

In [SRIV93, NICC93] we describe the details of a cost model that addresses the above issues. It provides means of labeling nodes of the query plan tree with various cost metrics such as work, response time, etc., and lends itself to efficient bottom-up evaluation.

**Search Algorithm:** It has been argued by [SWAM88,SWAM89,IOAN90] that exhaustive enumeration techniques such as dynamic programming [SELI79] are not likely to be successful for queries with large number of joins, i.e. 100 or so, and have proposed heuristic combinatorial optimization techniques such as Simulated Annealing, Iterative Improvement, and Successive Augmentation. The size of the search space for parallel query plans will be much larger than that for sequential ones [SRIV93]. This makes the need for efficient search algorithms of paramount importance. In [SRIV93] and [NICC93] we present two search heuristics to reduce the search space. The key elements of our approach are the following:

- The join-tree output from the sequential optimizer is converted into an operator tree.
- Decisions is made about which branches, i.e. intermediate results, will be pipelines and which will be materialized.
- Resource estimation for various tasks is done.
- Resource allocation for various tasks is carried out.
- At each step some heuristic choices are made to reduce the search space size.

We have built a prototype query optimizer and performed its initial evaluation [SRIV93, NICC93]. Figure 7 shows a schematic of our prototype optimizer. It is a *customizable* optimizer in the sense that it is table-driven and takes architectural parameters from a file as an input to its cost model. Thus, it is customizable to various architectures.

## 3. Goals & Specific Research Issues

### 3.1 Research Issues in the Record Management Layer

For the record management layer, the following specific research problems must be addressed:

- Evaluate the CMD approach with NASA data sets.
- Based on above evaluation tune/modify CMD, and if need be create new declustering strategies for NASA's data sets.
- Enhance our approach to provide better declustering by including information about a *core set* of NASA application queries. Many applications often have such a set, and we would like to identify such a set for the reprocessing algorithms.
- Since the relations are partially sorted on each dimension, its benefit on parallel external sorting algorithms needs to be examined.
- CMD provides an implicit indexing because of partial ordering of various domains. How this affects and is complemented by explicit indices, e.g. tree or hash based, needs exploration.
- Development of specialized indices for the parallel I/O system to speed-up the evaluation of aggregates [SRIV89], temporal selections [KOLO89], etc.
- Develop efficient parallel algorithms for loading large data files into relations in the PRDBMS, since this expected to be a frequent operation [PRAT93].
- Develop algorithms to perform operations along the temporal and spatial dimensions efficiently.

### 3.2 Research Issues in the Parallel Relational Algebra Layer

In this layer the following research issues must be addressed:

- Evaluate our *declustering aware* join algorithm on NASA's data sets.
- Based on above evaluation tune/modify the join algorithm, and if need be create new ones, for NASA's data sets.
- Apply the declustering aware approach to other algorithms in the relational algebra layer, e.g. union, difference, aggregation, etc.

### 3.3 Research Issues in the Query Compilation Scheduling Layer

Query compilation and scheduling is a wide open area of research today, and a number of issues remain open. Given the fact that it took almost a decade to get satisfactory sequential database query compilers, this is likely to be an area of active research for a few years. Specifically, the following research issues must be addressed:

- Evaluate the effectiveness of our optimizer on some typical queries found in NASA applications.
- Customize our prototype optimizer for a parallel architecture that NASA may be considering for building/acquiring a parallel DBMS on.
- Evaluate and validate the optimizer cost model, which is one of the keys to building a successful optimizer [DeWI92a]

## 4. Conclusions

In the past decade there has been a tremendous growth in the amount of data and resultant information generated by NASA's operations and research projects. This growth is expected to continue in the future. Use of parallel computers, both processing and input-output, will be a key to solving the resultant data management problem. In this paper we have described the architecture of a parallel data management system which is based on visualizing data as points in space and query processing as geometric operations. The architecture is highly parallel and is quite generic, i.e. can be realized on a wide variety of parallel machines. We provided an overview of our results and pointed out a number of open research issues.

## 5. References


[BARU88] C. K. Baru, O. Frieder, D. Kanflur, and M. Segal, "Join on a cube: Analysis, Simulation and Implementation", in *Database Machines and Knowledge Base Machines*, M. Kitsuregawa and H. Tanaka, Eds. Boston: Kluwer, 1988, pp 61-74.

[BORA83] H. Boral, and D. DeWitt, "Database Machines: An Idea Whose Time has Passed? A Critique of the Future of Database Machines", in *Proceedings of the 1983 Workshop on Database Machines*, Springer-Verlag, 1983.

[CAMP90a] William J. Campbell, and Robert F. Cromp, "Intelligent Information Fusion for Spatial Data Management", in *Proceedings of the 4th International Symposium on Spatial Data Handling*, 1990, Zurich, Switzerland.

[CAMP90b] William J. Campbell, and Robert F. Cromp, "Evolution of an Intelligent Information Fusion System", in *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No. 6, June 1990, pp. 867-870.

[CHEN 92] M. S. Chen, M. L. Lo, P. S. Yu, and H. C. Young, "Using Segmented Right-Deep Tress for the Execution of Pipelined Hash Joins", in *Proceedings of the 18th VLDB Conference*, August 1992, Vancouver, B.C., Canada.

[CHOU86] Hong-Tai Chou and David J. DeWitt, "An Evaluation of Buffer Management Strategies for Relational Database Systems", in *Proceedings of* $12^{th}$ *International Conference on Very Large Data Bases*, 1986.

[CROM89] R.F. Cromp, Sharon Crook, "An Intelligent User Interface for Browsing Satellite Data", in *Proceedings of 1989 Goddard Conference on Space Applications of AI*, Greenbelt, MD.

[CROM92] Robert F. Cromp, "An Intelligent Information Fusion System for Handing the Archiving and Querying of Terabyte-sized Spatial Databases", in *Proceedings of International Space Year Conference on Earth and Space Science Information Systems*, 1992, Pasadena, CA.

[DeWI84] D.J. DeWitt, et al, "Implementation Techniques for Main memory Database Systems", in *Proceedings of ACM SIGMOD Conference*, Boston, MA, June 1984.

[DeWI90] D.J. DeWitt, et al, "The Gamma Database Machine Project", in *IEEE Transactions on Knowledge and Data Engineering*, Vol 2, No 1, March 1990.

[DeWI92a] D. J. DeWitt and J. Gray, "Parallel Database Systems: The Future of High Performance Systems," in *Communications of the ACM*, Vol. 35, No. 6, June 1992.

[DeWI92b] D. J. DeWitt, J. F. Naughton, D. A. Schneider, and S. Seshadri, "Practical Skew Handling in Parallel Joins", in *Proceedings of the 18th VLDB Conference*, August 1992, Vancouver, B.C., Canada.

[ABDE93] K.A.S. Abdel-Ghaffar, A. El-Abbadi, "Optimal Disk Allocation for Partial Match Queries", in *ACM Transactions on Database Systems*, Vol 18, No 1, March 1993.

[FALO93] C. Faloutsos, P. Bhagwat, "Declustering Using Fractals", in *Proceedings of 2nd International Conference on PDIS*, San Diego, CA, January 1993.

[FRIE90] O. Frieder, "Multiprocessor Algorithms for Relational Database Operations on Hypercube Systems", in *IEEE Computer*, Vol 19, No 4, December 1990.

[GANG92] S. Ganguly, W. Hasan, R. Krishnamurthy, "Query Optimization for Parallel Execution", in *Proceedings of ACM SIGMOD Conference*, San Diego, CA June 1992.

[GHAN91] S. Ghandeharizadeh, L. Ramos, Z. Asad, and W. Qureshi, "Object placement in Parallel Hypermedia Systems", in *Proceedings of the* $17^{th}$ *International Conference on Very Large Data Bases*, Barcelona, Spain, 1991.

[GHAN92] S. Ghandeharizadeh, David J. DeWitt, and Waheed Qureshi, "A Performance Analysis of Alternative Multi-Attribute Declustering Strategies", in *Proceedings of ACM SIGMOD Conference*, San Diego, CA June 1992.

[GORD91] David Gordon, etc., "Disk Arrays: Are They of Use for Database Processing?" *Panel in Proceedings of First International Conference on Parallel and Distributed Information Systems*, Dec.1991, Miami Beach, Florida, pp. 117-118.

[GUST89] J.L. Gustafson, "Challenges to Parallel Processing", talk given at University of Minnesota, Minneapolis, September 1989.

[HONG91] Wei Hong and M. Stonebraker, "Optimization of Parallel Query Execution Plans for XPRS", $1^{st}$ *International Conference on Parallel and Distributed Information Systems*, Miami, Florida, 1991.

[HONG92] Wei Hong, "Exploiting Inter-Operation Parallelism in XPRS", in *Proceedings of ACM SIGMOD Conference*, San Diego, CA June 1992.

[HUA91] K. A. Hua and C. Lee, "Handling Data Skew in Multiprocessor Database Computers Using Partition Tuning", in *Proceedings of the* $17^{th}$ *International Conference on Very Large Data Bases*, Barcelona, Spain, 1991.

[HUA93] K.A. Hua, Y. Lo, and H.C. Young, "Including the Lad Balancing Issue in the Optimization of Multi-Way Join Querys for Shared-Nothing Database Computers", in *Proceedings of the 2nd International Conference on Parallel and Distributed Information Systems*, January 1993, San Diego, CA.

[IOAN90] Y. E. Ioannidis, and Y. Kang, "Randomized Algorithms for Optimizing Large Join Queries," in *Proceedings of Intl. Conf. on the Mgmt. of Data*, Atlantic City, NJ, May 1990.

[KOLO89] C. Kolovson and M. Stonebraker, "Indexing Techniques for Historical Databases", in *Proceedings of the* $5^{th}$ *International Conference on Data Engineering*, Los Angeles, LA, 1989.

[LEWI92] Ted G. Lewis, and Hesham El-Rewini, "Introduction to Parallel Computing", *Prentice Hall, Englewood Cliffs, New Jersey*, 1992, ISBN 0-13-498924-4.

[LI92] Jianzhong Li, Jaideep Srivastava, and Doron Rotem, "CMD: A Multidimensional Declustering Method for Parallel Databases Systems", in *Proceedings of the 18th VLDB Conference*, August 1992, Vancouver, B.C., Canada.

[McKE92] Edwin L. McKenzie Jr. and Richard T. Snodgrass, "Evaluation of Relational Algebras incorporating the Time Dimension in Databases", in *ACM Computing Surveys*, Vol. 24 No. 4, December 1991.

[MERC93] A. Merchant and P. S. Yu, "Issues in the Design of Multi-Server File Systems to Cope with Load Skew", in *Proceedings of the 2nd International Conference on Parallel and Distributed Information SYstems*, January 1993, San Diego, CA.

[NICC92] Thomas Niccum, Jaideep Srivastava, and Jianzhong Li, "DA-Joins : Declustering Aware Parallel Join Algorithms," Computer Science Dept. Univ. Of Minnesota, TR92-71.

[NICC93] Thomas Niccum, Jaideep Srivastava, Bhaskar Himatsingka and Jianzhong Li, "A Tree Decomposition Approach to the Parallel Execution of Relational Query Plans", AHPCRC Univ. of Minnesota, TR93-019.

[PATT88] David A. Patterson, Garth Gibson, and Randy H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)', in *Proceedings of ACM International Conference on Management of Data (SIGMOD)*, June 1988, pp. 109-116.

[PRAT93] Terrence W. Pratt, "CESDIS Proposal Guidelines", February, 1993.

[ROEL91] Larry H. Roelofs, and William J. Campbell, "Applying Semantic Data Modeling Techniques to Large Mass Storage System Designs", Presented at *the Tenth IEEE Symposium on Mass Storage Systems*, 1991.

[SCHN89] D.A. Schneider, D.J.DeWitt, "A Performance Evaluation of Four Parallel Join Algorithms in a Shared-Nothing Multiprocessor", in *Proceedings of ACM SIGMOD Conference*, Portland, OR, June

1989.

[SCHN90] D. A. Schneider and D. J. DeWitt, "Trade-offs in Processing Complex Queries via Hashing in Multiprocessor Database Machines", in *Proceedings of* $16^{th}$ *VLDB Conference*, Brisbane, Australia, 1990.

[SELI79] P. P. Selinger, et al, "Access Path Selection in a Relational Database Management System", in *Proceedings of ACM SIGMOD International Conf. on Mgmt. of Data*, 1979.

[SRIV89] J. Srivastava, J.S.E. Tan, V.Y. Lum, "TBSAM: A Tree-Based Access Method for Processing Aggregate Queries", in *IEEE Transactions on Knowledge and Data Engineering*, Vol 1, No 4, December 1989.

[SRIV93] Jaideep Srivastava, and Gary Elsesser, "Optimizing Multi-Join Queries for Shared-Memory Multiprocessor", in *Proceedings of the 2nd International Conference on Parallel and Distributed Information SYstems*, January 1993, San Diego, CA.

[STON86] M. R. Stonebraker, "The Case for Shared Nothing," in *Database Engineering*, Vol. 9, No. 1, 1986.

[STON88] M.R. Stonebraker, et al, "The Design of XPRS", in *Proceedings of 14th VLDB Conference*, Los Angeles, CA, August 1988.

[SWAM88] A. Swami, and A. Gupta, "Optimization of Large Join Queries," in *Proceedings of ACM SIGMOD Intl . Conf. on Mgmt. of Data*, Chicago, IL, June 1988.

[SWAM89] A. Swami, "Optimization of Large Join Queries: Combining Heuristics and Combinatorial Techniques," in *Proceedings of Intl. Conf. on Mgmt. of Data*, Portland, OR, June 1989.

[SU86] S.Y.W. Su, "Database Computers: Principles, Architecture and Techniques", *New York: McGraw-Hill*, 1986.

[TDMS90] "Third Generation Database Manifesto," In *ACM SIGMOD Record*, Vol. 19, No. 3, September 1990.

[TMC91] "Data Vault", *talk given by TMC*, 1991.

[WILS91] A.L. Wilschut and P.M.G. Apers, "Dataflow Query Execution in a Parallel Main-Memory Environment", in $1^{st}$ *International Conference on Parallel and Distributed Information Systems*, 1991, Miami, Florida.

[WOLF91] J.L. Wolf, D.M. Dias, P.S. Yu, and J. Turek, "Comparative Performance of Parallel Join Algorithms",in $1^{st}$ *International Conference on Parallel and Distributed Information Systems*, 1991, Miami, Florida.

[YU93] P. S. Yu and Douglas W. Cornell, "Buffer Management Based on Return on Consumption in a Multi-Query Environment", in *VLDB Journal*, Vol. 2, No. 1, January, 1993.

N94-33829

# The Importance of Robust Error Control in Data Compression Applications

S.I.Woolley
Dept. Electrical Engineering
University of Manchester
Oxford Road, M13 9PL, U.K.
Phone: 061-275-4538
Fax: 061-257-3902
sandra@hpc.ee.man.ac.uk

## Abstract

Data compression has become an increasingly popular option as advances in information technology have placed further demands on data storage capacities. With compression ratios as high as 100:1 the benefits are clear, however; the inherent intolerance of many compression formats to error events should be given careful consideration.

If we consider that efficiently compressed data will ideally contain no redundancy, then the introduction of a channel error must result in a change of understanding from that of the original source. Whilst the prefix property of codes such as Huffman enables resynchronisation, this is not sufficient to arrest propagating errors in an adaptive environment. Arithmetic, Lempel-Ziv, discrete cosine transform (DCT) and fractal methods are similarly prone to error propagating behaviours. It is, therefore, essential that compression implementations provide sufficient combatant error control in order to maintain data integrity. Ideally, this control should be derived from a full understanding of the prevailing error mechanisms and their interaction with both the system configuration and the compression schemes in use.

## Introduction

Data compression is essentially the process of identifying and extracting source redundancy in order to reduce storage requirements. Since the nature of encountered redundancy is dependent upon the type of source, e.g. image, audio, video, text, program source, database, instrumentation, etc., the best compression performance is achieved by a source-specific algorithm. For example, an image source may contain a large amount of positional redundancy (e.g. a raster scan of a vertical line) which could be efficiently exploited. However, an algorithm of this type would be of little benefit if applied to a textual source. Accordingly, each of the main source categories have their own families of compression algorithms.

The following text provides an introduction to some important compression concepts, methodologies and behaviours, with the aim of demonstrating the effects of compression on data integrity and the need for adequate error control. Where possible, examples have been used to illustrate this information such that a prior understanding of compression methods and behaviours is not required.

## Advantages and Disadvantages of Data Compression

In addition to the increase in data storage capacity, real-time compression implementations can enhance data transfer rates, improve network performance by reducing traffic loads on lines and, via the encryption process, provide additional data security against unauthorised access [1].

There are, however, disadvantages involved in data compression. Firstly, the compressed data is significantly more vulnerable to error, as will be demonstrated below. Secondly, the majority of software compressors do not work in real-time such that data retrieval and storage can be significantly delayed. This is particularly noticeable with most image compression methods, although the delay is, of course, dependent on system performance and source size. In addition, the plethora of algorithms currently available can result in compatibility problems when transferring data between systems. For example, the absence of an early compression standard for DAT (Digital Audio Tape) drives resulted in two competing and incompatible implementations, such that compressed DAT tapes cannot be freely transferred between systems which would otherwise be compatible [2]. Compatibility is also deteriorated, to some extent, by the existence of a variety of patents which make bespoke compressors more attractive.

## Channel Errors and Data Integrity

There are two types of channel errors; amplitude errors and phase errors. In digital form amplitude errors appear as inversions of bit values whilst phase errors appear as the insertion or extraction of one or more bits in the data stream. As shown in Figure 1, amplitude errors affect only those bits which are directly afflicted, whereas phase errors result in a stream of amplitude errors which propagate to the end of the information source or until the next resynchronisation marker.

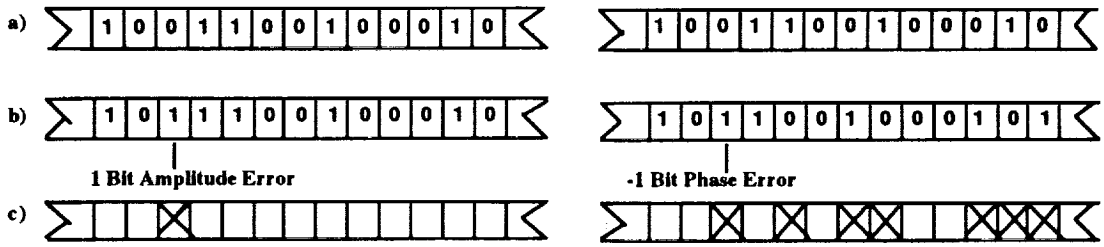



**Figure 1 : Example of Channel Errors**
**a) Source data, b) Source data with channel error imposed, c)Resultant bits in error.**

All systems are prone to error mechanisms of one sort or another, and with an understanding of these phenomena the subsequent, or secondary, effects on data integrity can be ascertained. For example, in magnetic tape recording systems a primary source of errors is the existence of embedded asperities, introduced either via the manufacturing process or via external contamination. The secondary effect of these mechanisms is the burst error syndrome caused directly by a departure of head-to-tape contact which attenuates signal amplitude. This phenomenon is referred to as a dropout, and where the attenuation results in effective signal loss then, as well as a burst of amplitude errors, the system may also experience a loss of synchronisation, i.e. a phase error. Therefore, although these events are comparatively rare their potential for error propagation makes them the dominant contributors to the overall error rate. So whilst bit/byte error rates are important indicators of system performance, the distribution and characterisation of error events are similarly important, and can be used to determine suitable error control measures i.e. interleaving, resynchronisation markers and error control coding.

## Lossy and Lossless Compression Techniques

The amount of compression that can be achieved by a given algorithm depends on both the amount of redundancy in the source and the efficiency of its extraction. The very high compression ratios often quoted generally relate to high-redundancy sources such as databases or to lossy compressed formats such as fractal image representations.

Lossless techniques are, of course, required by many applications, such as computer disk compression, where exact replication of the original is essential. Alternatively, lossy compression techniques, where only a close approximation of the original is reconstructed, can be successfully applied to many image, video and audio applications where losses outside visual/aural perception can be tolerated, and where the additional compression achieved is highly desirable. For example, the philosophy behind the lossy DCT (Discrete Cosine Transform) compression of images is that the human eye is less sensitive to high-frequency information. Further compression can, therefore, be achieved by more coarsely quantising high-frequency components of an image without significant visual deterioration.

There are, however, some image compression applications where certain quality and/or legal considerations dictate the use of lossless compression. For example, medical imaging **[3]** and deep space communication of imagery data **[4]**.

## Static and Adaptive Implementations

Compression algorithms remove source redundancy by using some working definition of the source characteristics, i.e. a source model. Compression algorithms which use a pre-defined source model are referred to as static, whilst algorithms which use the data itself to fully or partially define this model are referred to as adaptive.

Static implementations can achieve very good compression ratios for well defined sources; however, their inability to respond to changes in source statistics limits their usage. If applied to a source significantly different from that modelled, a static algorithm could result in source expansion rather than compression.

In contrast, adaptive algorithms are more versatile, and will update their working source model according to current source characteristics. However, adaptive implementations have lowered compression performance, at least until a suitable model is properly generated. In addition,

*"A major problem with any adaptive compression method is that a single error on the communication channel or storage medium can cause the sender and receiver to lose synchronisation and can result in error propagation that in the worst case can corrupt all data to follow."***[5]**

## Compression Methodologies

The following text presents a selection of commonly used compression methods and investigates the effects of channel errors on data integrity. Image sources and bit error maps are used to illustrate these phenomena since the effects of any errors are more readily appreciated.

## Huffman Coding

Huffman compression techniques are examples of statistical coding, where character frequency statistics are used to allocate appropriate codewords for output. The basic philosophy is that compression can be achieved by allocating shorter codewords to the more frequently occurring characters (a simple example of this technique is the Morse Code where, for example, E= • and Y= -•--). As shown in Figure 2, by arranging the source alphabet in descending order of probability, then repeatedly adding the two lowest probabilities and resorting, a Huffman tree can be generated. The resultant codewords are formed by tracing the tree path from the root node to the codeword leaf.

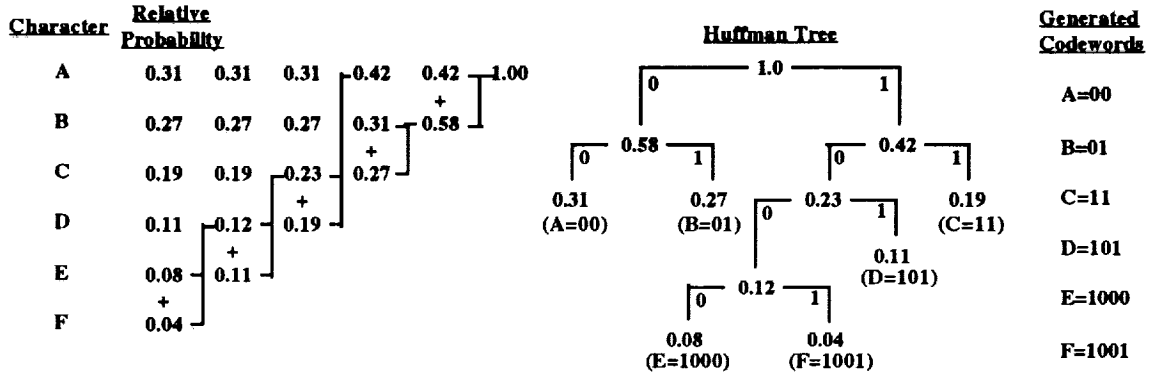



**Figure 2 : An Example of Huffman Coding**

The relative probabilities and, hence the Huffman tree, can be derived by the compressor in three ways. Firstly, in static implementations the probabilities are predefined. This enables rapid and efficient compression and decompression but only for sources whose character frequencies are accurately described by those of the algorithm. Secondly, the probabilities can be generated by an initial pass of the data to count the character frequencies, but this requires an additional second pass to perform the compression. Thirdly, the probabilities (and hence the codewords) can be adjusted dynamically during the compression process. This adaptive Huffman method, although more complex, can still perform quite rapid compression and decompression and has the advantage of being more versatile as well as being able to respond dynamically to changes in the source.

Compression performance can be improved by increasing the order of the source model (i.e. considering the context of incoming characters). For example, when the letter "q" is parsed from a stream of english text there is a very high probability (approx. 95%) that the letter "u" will follow.

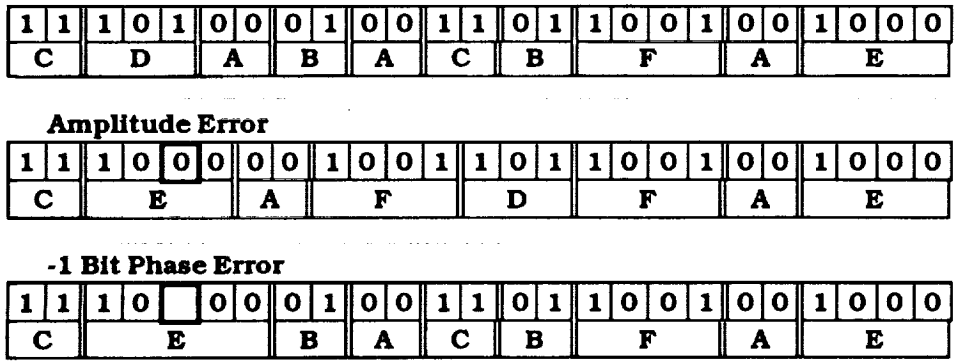
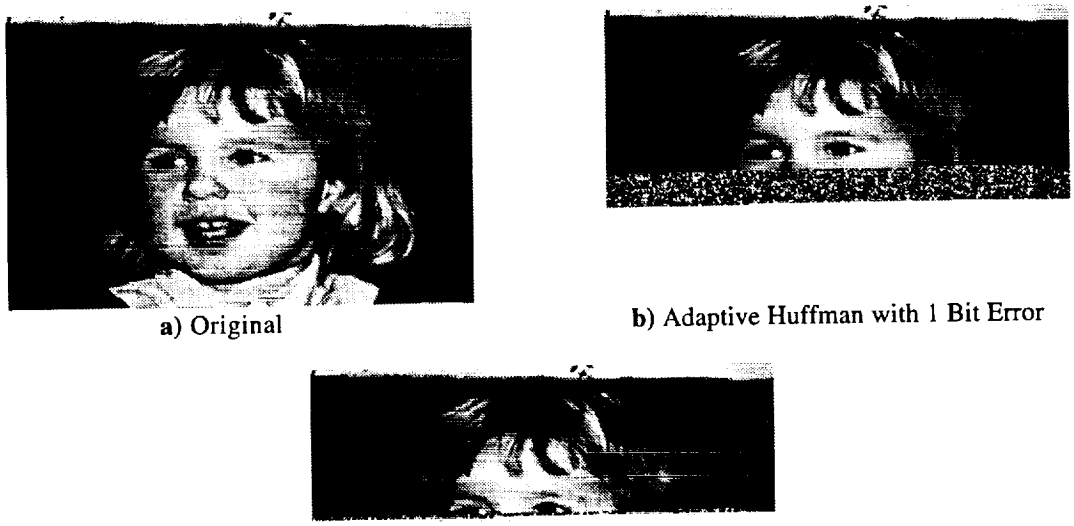



**Figure 3 : Resynchronisation of Huffman Coded Data in the Event of Channel Errors**

There is, therefore, no single Huffman method, but rather a whole family varying in adaptivity and order[6]. The specific effects of any transmission errors on Huffman compressed data would therefore depend on the particular implementation as well as the source. Fortunately, however, some generalisations can be made. Firstly, static implementations have an in-built resistance to error propagation. This is due to the prefix property common to all Huffman codes, whereby no code is a prefix of any other. For example, using the generated codewords in Figure 1, if 00 is a permitted codeword then no other codeword may begin with the sequence 00, similarly if 101 is a permitted codeword then no other codeword may begin with the sequence 101, and so on. Figure 3 illustrates the effects of an amplitude and a phase error, demonstrating the code's self-synchronising ability.

Adaptive Huffman implementations, however, are not protected by this facility since; *"The fact that the sender and receiver are dynamically redefining the code indicates that by the time synchronisation is regained, they may have radically different representations of the code"* **[7].**

Figure 4b below shows the effects of a single bit error on an image source compressed by an adaptive Huffman scheme.



a) Original

b) Adaptive Huffman with 1 Bit Error

c) Arithmetic Compression with 1 Bit Error

**Figure 4** : Examples of Bit Errors on Arithmetic and Adaptive Huffman Formats

## Arithmetic Coding

The method of compression employed by Huffman compression coding involves the allocation of shorter codewords for more frequently occurring characters. It is, however, unable to allocate fractional codeword lengths, so that a character must be allocated at least a one-bit codeword no matter how high its frequency. Huffman coding cannot, therefore, achieve optimal compression.

Arithmetic coding [8] offers an alternative to Huffman coding, enabling characters to be represented as fractional bit lengths. This is achieved by representing the source as a real number, greater than or equal to zero, but less than one, denoted as the range [0,1). As shown in Figure 5, each character of the source alphabet is allocated a proportion of this range according to its probability. As data symbols are parsed from the source, the working range is reduced and a real number is generated which can then be transmitted and decompressed via a parallel process. The source sequence shown in column four of Figure 5 illustrates this process. As can be seen the initial range is [0,1) which is reduced to [0.2,0.5) when a "B" is encountered. The next symbol, an "A", requires the range [0,0.2), i.e., the first 20% of the working range, [0.2,0.5), hence the new range [0.2,0.26).

| Source Characters | Relative Probabilities | Allocated Range | Source Sequence | Denoted Range |
|---|---|---|---|---|
| A | 0.2 | [0,0.2) | --- | [0,1) |
| B | 0.3 | [0.2,0.5) | B | [0.2,0.5) |
| C | 0.1 | [0.5,0.6) | A | [0.2,0.26) |
| D | 0.2 | [0.6,0.8) | C | [0.23,0.236) |
| E | 0.1 | [0.8,0.9) | C | [0.233,0.2336) |
| F | 0.1 | [0.9,1.0) | F | [0.23354,0.2336) |

**Figure 5 : An Example of Arithmetic Coding**

The handling of the necessary floating point arithmetic and the need for special terminating sequences makes arithmetic coding more complex than Huffman coding. However, the algorithm achieves close to optimal compression, and, like Huffman coding, can be increased in order and adaptivity.

Unfortunately, arithmetic coding, whether static or adaptive, is particularly vulnerable to errors, whereby a single bit error can result in a complete scrambling of all subsequent data [**Teuhola91**], as demonstrated in Figure 4c.

| DATA SEQUENCE : "THE THREE TREES" | | | | | |
|---|---|---|---|---|---|
| Previous Character or String | New Character | Generated Dictionary Codeword | Meaning of Dictionary Codeword | Code Output | Meaning of Output |
| --- | T | --- | --- | --- | --- |
| T | H | D1 | TH | T | T |
| H | E | D2 | HE | H | H |
| E | T | D3 | ET | E | E |
| T | H | The string "TH" already exists so a new dictionary codeword is not generated. | --- | --- | --- |
| TH | R | D4 | D1+R =THR | D1 | TH |
| R | E | D5 | RE | R | R |

| E | E | D6 | EE | E | E |
|---|---|---|---|---|---|
| E | T | The string "ET" already exists so a new dictionary codeword is not generated. | --- | --- | --- |
| ET | R | D7 | D3+R =ETR | D3 | ET |
| R | E | The string "RE" already exists so a new dictionary codeword is not generated. | --- | --- | --- |
| RE | E | D8 | D5+E =REE | D5 | RE |
| E | S | D9 | ES | E | E |
| S | End of data | --- | --- | S | S |

**Figure 6 : Simplified Example of LZ78 Compression Process**
**The original source contains 13 8-bit characters (=104 bits) and the compressed output contains 10 9-bit codewords (=90 bits).**

## Lempel-Ziv Methods

There are two Lempel-Ziv compression algorithms: LZ77 [10] and LZ78 [11], both of which compress data by replacing repeated strings by defined codewords. LZ77 uses a so-called sliding window from which repeated strings are identified and referenced by a coded block indicating the position and length of the string. In this way LZ77 creates a compressed format which comprises uncompressed data interspersed with pointers to recognised strings. This simple format enables rapid decompression; however, compression is comparatively slow since continuous searches of the sliding-window are required.

Alternatively, the LZ78 algorithm (and similarly LZW [12], an improved and patented version) creates a dynamic embedded dictionary designed with a self-referencing structure. The algorithm parses the data source for unique strings (i.e., strings not previously encountered) for which it allocates dictionary codewords that can be used to replace the string if it occurs again. The implementation of this method is best explained by means of a simple example as shown in Figure 6 (reading left to right, row by row). In its simplest form all of the output (characters and dictionary codewords) are in 9-bit form, but can be increased to 10, 11 or 12 bits as required by sending reserved codewords (control flags). These are also used to fully or partially reset the dictionary when it fills or when significant deterioration in compression performance is detected. The simplified example shown in Figure 6 describes dictionary entries as D1, D2, D3 etc., and a high-redundancy data sequence ("THE THREE TREES", with spaces ignored) is used in order to demonstrate the compression of repeated strings.

During compression, the self-referencing nature of the dictionary enables longer and longer strings to be replaced by just one dictionary codeword, and there is no need to explicitly write the dictionary contents since it can be regenerated via the same process on decompression. Initially the compression performance of the algorithm is poor, but, as strings are re-encountered and replaced with dictionary codewords performance increases rapidly.

The speed and versatility of the Lempel-Ziv implementations have made them particularly popular and have led to the development of a large and growing family of related algorithms

([13] refers to 12 variants of Lempel-Ziv methods). Implementations of Lempel-Ziv type algorithms can be found in most computer disk compressors and on tape drives including DAT.

## Channel Errors in LZ78 Compressed Data Formats

The introduction of amplitude errors into the compressed format can have the following effects:

1. a small number of amplitude errors restricted to the locality of the original,
2. a propagation of amplitude errors from the point of error location to the end of the entity, and
3. a complete loss of data either occurring abruptly or following a variable-length burst of amplitude errors.

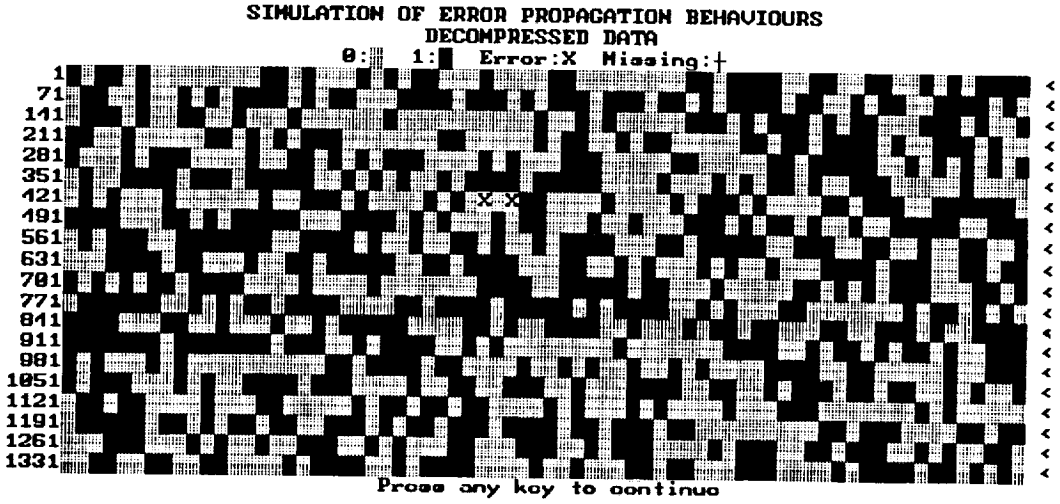
These outcomes are explained in Figure 7 by considering the types of compressed source involved.

| The Effects of Channel errors on LZ78 Compressed Data | | |
|---|---|---|
| **Location of Error** | **Description** | **Resultant Effect on Data Integrity** |
| a) In an output character. | (i) The character value is altered. | The error would remain in the decompressed output and be replicated for each repeating string in which the character was parsed. |
| | (ii) The character is transposed into a dictionary codeword. | *The difference in length alone will produce a synchronisation loss. |
| | (iii) The character is transposed into a control flag. | **The source will be corrupted or lost since control flags adjust byte lengths, manipulate the dictionary, and indicate the end of data. |
| b) In an output dictionary codeword. | (i) The error transposes the dictionary codeword into another [recognised] dictionary codeword. | The referenced string is replaced by an erroneous string. If these are of different lengths then synchronisation will be lost, and any future references to the newly created dictionary codeword will be affected in the same way. |
| | (ii) The error transposes the dictionary codeword into an unrecognised dictionary codeword. | The decompressor will be unable to identify the codeword and subsequent data will be lost. |
| | (iii) The error transposes the dictionary codeword into an output character. | As *. |
| | (iv) The error transposes the dictionary codeword into a control flag. | As **. |

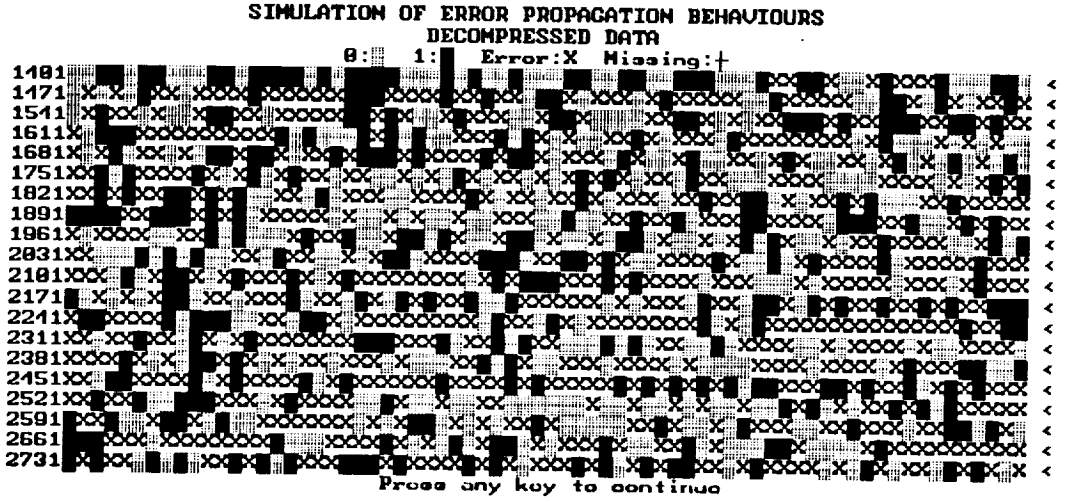
| | | |
|---|---|---|
| c) In a control flag. | (i)The control flag is mistaken for an output character or a dictionary codeword. | As **. |

**Figure 7 : A Description of the Potential Effects of Channel Errors on an LZ78 Compressed Source.**

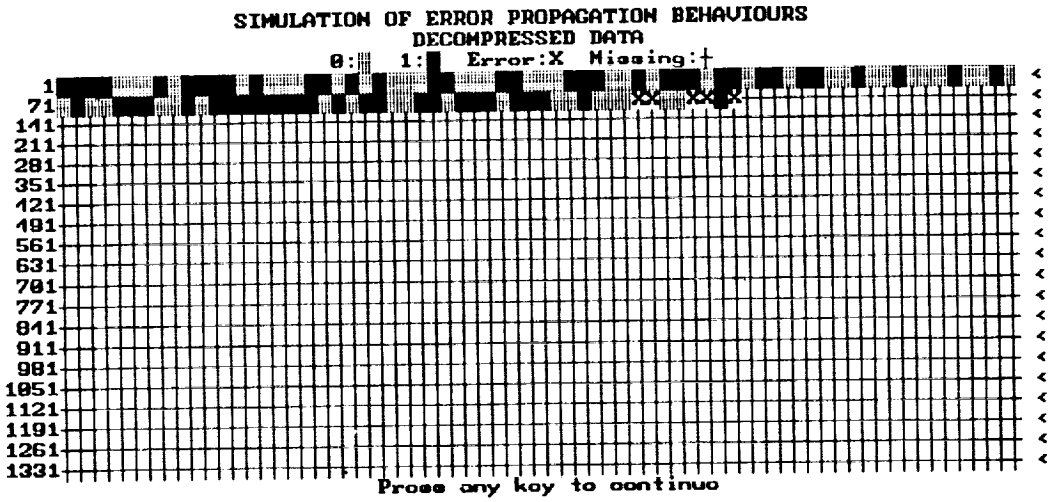
**Figure 8** below shows 3 different bit error maps: a), b) and c), of LZ78-type compressed sources which correspond to **1.**, **2.** and **3.** above. These results were generated via the comparison of the original source with the decompressed output of the same source, but with a single-bit error introduced into the compressed form.




a)




b)




c)

Figure 8 : The effects of single amplitude errors on LZ78 compressed data.
a) Recovery, b) Error Propagation and c) Data Loss

## Discrete Cosine Transform (DCT) Image Compression

The philosophy behind DCT image compression is that the human eye is less sensitive to high-frequency information (and also more sensitive to intensity than to colour), so that compression can be achieved by more coarsely quantising the large amount of high-frequency components usually present. Firstly, the image must be transformed into the frequency domain. Since it would be computationally prohibitive to transform even a low resolution image in one full block, the image is subdivided. The developing JPEG (Joint [CCITT and ISO] Photographic Experts Group) standard algorithm [**14**] for full-colour and grey-scale image compression uses 8x8 blocks.

The DCT itself does not achieve compression, but rather prepares the image for compression. Once in the frequency domain the image's high-frequency coefficients can be coarsely quantised so that many of them (>50%) can be truncated to zero. The coefficients can then be arranged so that the zeroes are clustered and Run-Length Encoding (RLE), whereby repeated values are referenced and followed by a counter indicating the number of successive occurrences, can be applied. The remaining data is then compressed with Huffman coding (arithmetic coding has also been proposed but implementation has been hampered by patent issues).

DCT compression, therefore, involves a number of processes, all of which combine to allow extensive error propagation. An example of the effects of a single bit error is shown in Figure 9g. The RLE coding alone is prone to error propagation since the length or value of referenced symbol repetitions can be altered by a single amplitude error. The JPEG standard addresses this problem by providing "restart" markers which allow the decoder to resynchronise after a transmission error [**JPEG Version 4:USAGE**]. This facility does not correct errors, but arrests their propagation. The number of markers used is user defined and should be determined by the channel error statistics, the error tolerance of the system, and the reduction in compression which can be tolerated as a result of their insertion.

The lossy nature of the DCT method is shown in Figure 9. As can be seen in Figure 9b, significant compression can be achieved without any visible loss in picture clarity. However,

by increasing the compression performance further losses are seen to develop (particularly lossy examples were chosen since the effects were required to be visible after reduction for inclusion in this document).

The presence of artifacts around sharp edges is referred to as Gibb's phenomenon [15](pg225). These are caused by the inability of a finite combination of continuous functions to describe jump discontinuities. As shown in Figure 9e, at higher compression ratios these losses become mores apparent, as does the blocked nature of the compressed form. Figure 9f shows a difference mapping of the original and the highly lossy decompressed image in which the loss of edge clarity can be observed.

This type of lossyness makes JPEG and other DCT-based algorithms unsuitable for non-realistic images, e.g. line drawings, cartoons, etc., as can be seen by the large amount of deterioration in the geometric example used in Figure 10.

## Fractal Image Compression

A fractal, in simplest terms, is an image of a texture or shape expressed as one or more mathematical formulae. It is a geometric form whose irregular details recur at different locations, scales and angles, and which can be described in terms of formulae called affine or fractal transformations [16]. Fractal image compression is achieved by dividing an image into sub-blocks, each of which is then compared to scaled and rotated versions of the other sub-blocks in the image. When sufficiently similar sub-blocks have been found for all of the sub-blocks in the image, they can be referenced geometrically so that a fractal description is obtained.

Unlike other image compression techniques, the fractal transform is a very asymmetric process. The exhaustive searching for self-similarity requires intensive computational power, but once expressed in terms of fractal transforms the image can be quickly decompressed. Fractal compression is therefore best suited to WORM applications. For example, the Microsoft multimedia encyclopaedia Encarta makes use of fractal compression to store hundreds of colour maps and thousands of photographs on a single CD which also contains extensive audio, animation and textual data. [15].

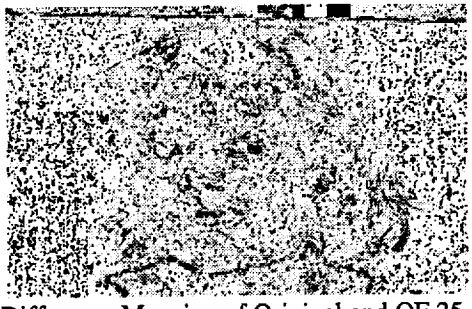
a) Original

b) DCT : QF 3 : CR 8:1

c) DCT : QF 10 : CR 11.6:1

d) DCT : QF 20 : CR 13.6:1

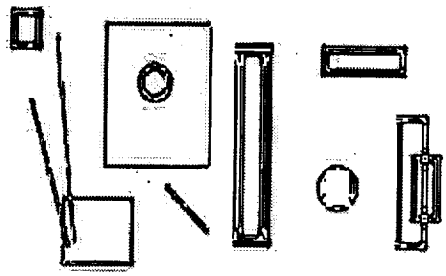
e) DCT : QF 25 : CR 14.2:1



f) Difference Mapping of Original and QF 25

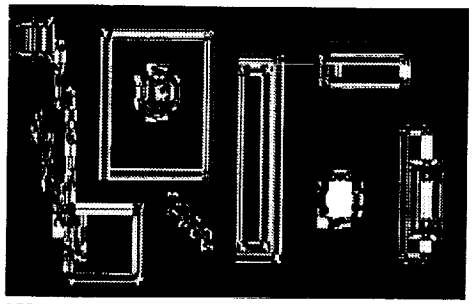
g) DCT (8:1) with 1 Bit Error

**Figure 9**: **b)-e)** DCT Compressed Images of **a)** : QF (Quality Factor) in the range [1-25] (best-worst)
**f)** Comparison of a) and e)
**g)** Data Loss due to a Single Bit Error



Original

DCT : QF 25 : CR 11.6:1



Difference Mapping of Original and QF 25

**Figure 10** : Edge Distortions Produced by DCT Compression

The large amount of self-similarity in "real-world" images enables fractal image compression to achieve very high compression ratios, usually significantly higher than those for DCT compression. But like DCT compression, fractal compression is unsuited to geometrical images where self-similarity is not evident [17].

The effects of small numbers of bit errors on fractal compressed images can result in severe degradation of the afflicted sub-blocks and propagate into the referenced self-similar sub-blocks. However, the iterative nature of fractal image generation means that these distortions will have reduced contributions.

Since fractal compressed images are represented as mathematical structures they are size-independent. Compression ratios can therefore be increased by comparing the size of the fractal form to the size of the enlarged original, rather than the original itself.

## Compression of Instrumentation Data

The compression of instrumentation data is a comparatively neglected area. This can be explained by the variance in byte lengths and also the application specific nature of generated sources. However, one method which can be more generally employed is difference modulation, whereby data symbols are encoded as the difference (positive or negative) from the previous symbol. For example:

**42 41 43 43 45 44 42 39 40**

could be represented as;

**42 -1 +2 0 +2 -1 -2 -3 +1**

which, of course, could be easily compressed by using fewer bits to represent the differences. Instrumentation data is suited to difference modulation since it often involves either visible trends relating to gradual changes in a monitored process, or relatively small perturbations

centred about some mean value. Difference modulation is also useful in audio compression applications, where trends can be identified in the source waveform. In applications were losses can be tolerated, the quantisation of differences can achieve further compression.
In the presence of channel errors difference modulation will result in error propagation since each of the symbols measures itself against its predecessor. The resultant shift caused by a bit error will be replicated in all subsequent symbols on decompression, propagating until the end of the data sequence or until the next true measurement is parsed.

## Conclusion

In the absence of data compression many systems, when afflicted by uncorrected channel errors, will suffer only localised losses in data integrity, i.e. will fail gracefully.. However, similar errors in systems using data compression can have disastrous results. For this reason users of computer disk compression are advised to take regular backups since: *"if something does go wrong, it is likely to be major"*.[18].

The results of transmission errors on different compression methodologies has been demonstrated, and the need for robust error control emphasised. This control should be determined by the channel error statistics and the error tolerance of the system. In addition to error control coding, piece-wise compression, resynchronisation markers and deep interleaving can also be employed to limit the propagation of errors and reduce the correction burden placed on the error control coding.

## Acknowledgements

Much of the work presented here was sponsored by British Gas Plc, whose financial and technical support has been greatly appreciated. I am also very grateful for the interest and support given by ICI Imagedata which has enabled the publication of this work. Finally, I would like to thank to my research supervisor, Prof.B.K.Middleton, and Dr.B.Bani-Eqbal (Dept. Computer Science, University of Manchester) for his technical advice on fractal and DCT compression implementations.

## References

[1] **The Hidden Benefits of Data Compression**
D.Powell
Networking Management, Vol.7, No.10, October 1989, pp46-54

[2] **The Compression Wars**
J.McLeod
Electronics, Vol.64, September 91, pp27-28

[3] **Performance Analysis of Reversible Image Compression Techniques for High-Resolution Digital Teleradiology**
G.R.Kuduvalli and R.M.Rangayyan
IEEE Transactions on Medical Imaging, Vol.11, No.3, September 1992, pp430-445

[4] **Recent Advances in Lossless Coding Techniques**
G.S.Yovanof
Proc-27th Int. Telemetric Conf. ITC91, pp7-19

[5] **Data Compression**
A.Bookstein and J.A.Storer
Information Processing and Management, Vol 28, No.6, 1992, pp675-680

[6] **Overview of Huffman Encoding as [a] Compression Technique**
K.Anderson
Computer Technology Review, Vol.11, No.6, 1991, pp97-101

[7] **Data Compression**
D.A.Lelewer and D.S.Hirschberg
ACM Computing Surveys, Vol.19, No.3, September 1987, pp261-296

[8] **An Introduction to Arithmetic Coding**
G.G.Langdon
IBM Journal of Research and Development, Vol.28, No.2, March 1984, pp135-149

[9] **Piecewise Arithmetic Coding**
J.Leuhola and T.Raita
Proceedings of DCC'91, pp33-42

[10] **A Universal Algorithm for Sequential Data Compression**
J.Ziv and A.Lempel
IEEE Trans IT, IT-32, No.3,May 1977,pp337-343

[11] **Compression of Individual Sequences via Variable-Rate Coding**
J.Ziv and A.Lempel
IEEE Trans IT, IT-24, No.5,Sept 1978, pp530-536

[12] **A Technique for High-Performance Data Compression**
T.A.Welch
IEEE Computer, Vol.17, No.6, June 1984, pp8-19

[13] **Text Compression**
T.C.Bell, J.G.Cleary and I.H.Witten
Published by Prentice Hall, Englewood Cliffs,NJ. 1990

[14] **The JPEG Still Picture Compression Standard**
G.K.Wallace
Communications of the ACM, Vol.34, No.4, April 1991, pp31-44

[15] **Fractal Image Compression**
M.F.Barnsley and L.P.Hurd
Published by AK Peters Ltd, 1993, ISBN 1-56881-000-8

[16] **Advances in Digital Image Compression Techniques**
G.Lu
Computer Communications, Vol 16, NO.4, April 1993, pp202-214

[17] **Fractals Transform Image Compression**
A.Wright
Electronics World and Wireless World, March 1992, pp208-211

[18] **Data Compression Software**
R.Milton
Computing , 22 July 1993, pp19-20

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | April 1993 | Conference Publication, October 19-21, 1993 |

**4. TITLE AND SUBTITLE**
Third NASA Goddard Conference on Mass Storage Systems and Technologies

**5. FUNDING NUMBERS**
505

**6. AUTHOR(S)**
Benjamin Kobler and P. C. Hariharan, Editors

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS (ES)**
Goddard Space Flight Center
Greenbelt, Maryland 20771

**8. PEFORMING ORGANIZATION REPORT NUMBER**
94B00057

**9. SPONSORING / MONITORING ADGENCY NAME(S) AND ADDRESS (ES)**
National Aeronautics and Space Administration
Washington, DC 20546-0001

**10. SPONSORING / MONITORING ADGENCY REPORT NUMBER**
NASA CP-3262

**11. SUPPLEMENTARY NOTES**
Kobler: Goddard Space Flight Center, Greenbelt, Maryland;
Hariharan: Systems Engineering and Security, Inc., Lanham, Maryland.

**12a. DISTRIBUTION / AVAILABILITY STATMENT**
Unclassified - Unlimited
Subject Category 82

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

This report contains copies of nearly all of the technical papers and viewgraphs presented at the Goddard Conference on Mass Storage Systems and Technologies held in October 1993. Once again, the conference served as an informational exchange forum for topics primarily relating to the ingestion and management of massive amounts of data and the attendant problems involved. Discussion topics include the necessary use of computers in the solution of today's infinitely complex problems, the need for greatly increased storage densities in both optical and magnetic recording media, currently popular storage media and magnetic media storage risk factors, data archiving standards including a talk on the current status of the IEEE Storage Systems Reference Model (RM). Additional discussion topics addressed System performance, data storage system concepts, communications technologies, data distribution systems, and finally, a talk on data compression and error detection and correction.

**14. SUBJECT TERMS**
Magnetic tape, magnetic disk, optical disk, mass storage, software storage, digital recording, data compression

**15. NUMBER OF PAGES**
514

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |