NASA Conference Publication 3198, Vol. 1

Goddard Conference on Mass Storage Systems and Technologies

Volume 1

Proceedings of a conference held at NASA Goddard Space Flight Center Greenbelt, Maryland September 22-24, 1992

NASA Conference Publication 3198, Vol. I

Goddard Conference on Mass Storage Systems and Technologies

Volume I

Edited by
Ben Kobler
Goddard Space Flight Center
Greenbelt, Maryland

P. C. Hariharan
STX Corporation
Lanham, Maryland

Proceedings of a conference held at
NASA Goddard Space Flight Center
Greenbelt, Maryland
September 22–24, 1992

National Aeronautics and
Space Administration

Office of Management

Scientific and Technical
Information Program

1993

Goddard Conference on Mass Storage Systems and Technologies

Program Committee

Ben Kobler, NASA/GSFC (Chair)
John Berbert, NASA/GSFC
William A Callicott, NOAA/NESDIS
Sam Coleman, Lawrence Livermore National Laboratories
Susan Hauser, National Library of Medicine
Sanjay Ranade, Infotech SA, Inc
Elizabeth Williams, Supercomputing Research Center
Jean-Jacques Bedet, Hughes STX
Alan Dwyer, Hughes STX
P C Hariharan, Hughes STX

Conference Coordinator

Nicki Fritz, Hughes STX

Production and Layout

Len Blasso, Hughes STX
Ann M. Lipscomb, Hughes STX

PREFACE

Papers presented at the Goddard Conference on Mass Storage Systems and Technologies that were submitted for publication in advance of the Conference appear in volume 1 of these Proceedings. Volume 2 contains additional papers and view graphs which were made available at the time of the Conference, as well as reports of the keynote address, the after-dinner speech, and the two panel discussions. We are grateful to all the authors for their contributions.

Dr. David Nelson, Director of the Office of Scientific Computing, Department of Energy, opened the conference with a keynote address that began by identifying projects and activities that are, or will be, generating massive volumes of data. Some of the grand challenge problems of the High Performance Computing and Communications initiative are likely to rival, or even surpass, the Earth Observing System in the amount of data they create. Managing such large archives is itself likely to prove a grand challenge. He referred to inaccessible data as the *"landfill of cyberspace."* Learning to answer unanticipated questions, revising data structures as requirements evolve, doing this in a cost-effective and practical manner in a hierarchical storage system, and dealing with distributed data bases that are networked together will tax both human ingenuity and resources.

Mass storage systems have now truly begun to be massive, with data ingestion rates approaching terabytes per day. At the same time, the identifiable unit for processing purposes (file, granule, dataset or some similar object), has also increased in size, and could begin to pose a challenge to traditional file systems that impose limits on both the size of the objects, and the number of objects in the file system. Even the casual user needs more than the object name, the size and date of the creation of the object, and the limited metadata provided with classical directory systems. Some of these issues are addressed by the IEEE Mass Storage System Reference Model (MSS RM), which is seeking to provide a framework in which hardware and software from different vendors can act cooperatively and harmoniously to store, manage and distribute data. Dr. Sam Coleman of the Lawrence Livermore National Laboratory and Mr. Bob Coyne of the IBM Federal Sector Division discussed the history and current status of the Reference Model. Version 5 of the MSS RM will appear in April 1993 as a Recommended Practice instead of as a Guide. The emphasis of the Storage Systems Standards Working Group (SSS WG) is focused on decomposing storage systems into interoperable functional modules which vendors may offer as separate products, and on defining standard interfaces through which clients may be provided direct access to storage systems services. *Bob Coyne pointed out that the data management, database, and file system development and user communities are not represented in the SSS WG, and issued a plea for their active participation in the activities and deliberations of the WG.* Those interested in the SSS WG discussions may keep abreast by sending e-mail to **ieee-mss-request@nas.nasa.gov** with the request that their name and address be included in the WG reflector. General discussions on mass storage problems are also published in the USENET newsgroup **comp.arch.storage**.

Standards are essential to ensure wide availability, multi-sourcing, and interchangeability. Mr. Al Dwyer, representing the NASA-OSSA Office of Standards and Technology, spoke about the role of this office. He was followed by Mr. Jean-Paul Emard, ANSI X3 Committee Director, Mr. Sam Cheatham of the X3B5 Committee, and Mr. Ken Hallam of the X3B11 Committee who discussed the ANSI standards-making process, the work on magnetic media standards, and the status of the optical media standards, respectively.

The sheer size of the inventories makes distributed systems attractive. Bob Coyne discussed the National Storage Asset Laboratory at the National Energy Research Supercomputer Center of the Department of Energy; this will be a testbed for network-attached storage devices. In this configuration, the devices will be nodes in a network, and will provide read/write services to authorized clients on the network without the need for the data to pass through the memory of a computer controlling the devices. Experiences from the archives at NOAA, the National

Space Science Data Center at NASA, the Eros Data Center of the USGS[1] , and the National Library of Medicine were complemented by a discussion of the information management challenge posed by the Earth Observing System. Dr. Ackerman of the National Library of Medicine pointed out that while there is much discussion of gigabit networks and petabyte-sized inventories, there are still problems today in distributing much smaller files to a user community not fortunate enough to be plugged into the latest wideband network. Browsing is a significant component of the activity at large holdings, and Dr. Ken Salem described one way to handle this.

High volume holdings require high-performance storage devices. The idea of using a Redundant Array of Inexpensive Disks to provide increased bandwidth and reliability had previously been espoused by Garth Gibson, and others, and Dr. Gibson provided a simplified explanation of it in his tutorial lecture. A natural outgrowth of the RAID idea is that of RATS (Redundant Array of Tape Systems), and Ms. Ann Drapeau of the University of California at Berkeley took up this topic in her tutorial.

Professor Mark Kryder, Director of the Engineering Research Center in Data Storage Systems at Carnegie Mellon University, Pittsburgh, PA discussed the future evolution of magnetic and magneto-optic storage systems in his talk on ultra-high density recording technologies. In cooperation with the National Storage Industry Consortium, the Center has selected the goals of achieving 10 Gbit/in^2 recording density in magnetic and magneto-optic disk recording, and 1 Tbit/in^3 in magnetic tape recording.

The National Media Laboratory (NML) has been in existence since 1989, and Dr. Gary Ashton provided an overview of its structure, scope and mission and reported on NML testing results of D-1 cassettes. A different perspective, that of the system integrator, was furnished by Mr. Richard Lee in his talk on grand challenges in mass storage.

Recent magnetic and optical recording technologies were described in a number of papers. Optical recording, traditionally available on disks, is now possible on tape. ICI Imagedata, which has pioneered the concept of the digital paper, and subjected its product to one of the largest suite of tests, now has competition from the Dow Chemical Company and from Eastman Kodak. While optical storage has generally been understood to involve ablation (pit-forming), phase change, or alloy formation (respectively the modes of the ICI, Eastman Kodak and the Dow products), Optex has a medium that uses a different technique for optical data storage. This involves excitation of electrons, and trapping the excited electrons in metastable states on a receptor ion. The method is interesting and intriguing because, unlike other technologies, it exhibits a linear response and can therefore store more than just one bit per "cell." A panel discussion on the comparative merits of magnetic and optical storage, and their future, followed these papers.

Dr. Dennis Speliotis, a veteran in the field of magnetic storage, was the after-dinner speaker at the Conference Banquet. He reminisced about his experiences over more than three decades in magnetic storage and related stories of both success and failure. His parting words were significant: the way to make progress is through evolution, not revolution; the chances of failure when one attempts a dramatic change, a drastic departure from the conventional, are very high, certainly in the short term; but small, evolutionary step-changes are more likely to succeed.

Mr. Dale Lancaster of Convex Systems presented what the "state of the art" is in Mass Storage Technology. Drs. Elizabeth Williams and Tom Myers discussed the need for, and the nature of, the types of measurements and metrics of distributed and heterogeneous storage systems. Measurements were reported by Ms. Nancy Yeager of the National Center for Supercomputing Applications. Mr. Bill Collins of the Los Alamos National Laboratory presented an overview of the High Performance Data System being developed there and Dr. Milt Halem, from the NASA

[1] Although John Boyd was unable to present his paper "Interim Report on Landsat National Archive Activities," it is nevertheless included in these proceedings

Goddard Space Flight Center gave a critical and comparative analysis of three application-dependent mass storage systems being built at Goddard.

Mr. James F Berry, of the Department of Defense, chaired a panel discussion on High Performance Helical Scan Recording Systems. Representatives from Ampex, Datatape, GE, Sony and StorageTek were the participants.

The performance of the low-end helical scan tape drives was the topic of papers by Dr. Chinnaswamy, formerly of Digital Equipment Corporation, and by Mr. Gerry Schadegg of Exabyte Corporation. Exabyte now provides an on-line Technical Support Bulletin Board System (EBS). Banana Boat, as the BBS is called, can be accessed by dialing (303) 442-4323. The BBS contains information such as microcode history, technical bulletins, white papers, and articles of interest to 8 mm product users. *Mr. Schadegg advised users of 8 mm drives that those drives were not designed for 100% duty cycle, but only for 20% to 30%.* He also cautioned users that the small, handy size of the cassette should not lull them into thinking that the media does not require a controlled environment for storage, shipping and operation. Finally, tips on reducing file read latencies were discussed by Mr. R. Hugo Patterson of Carnegie Mellon University.

A number of posters were presented on the first day of the conference.

Our thanks go, in addition to the authors, to the following persons and organizations:

Dr. David Nelson, Department of Energy, the keynote speaker,
Dr. Dennis Spellotis, the after-dinner speaker,

the following session and panel discussion chairs:

Dr. Joe King, NASA/GSFC,
Dr. Mark Kryder, Carnegie Mellon University,
Dr. Milt Halem, NASA/GSFC,
Mr. James F Berry, Department of Defense,

the following members of the program committee:

Mr. Jean-Jacques Bedet, Hughes STX Corporation,
Mr. Bill Callicott, NOAA,
Dr. Sam Coleman, Lawrence Livermore National Laboratory,
Mr. Alan M Dwyer, Hughes STX Corporation,
Dr. Susan Hauser, National Library of Medicine,
Dr. Sanjay Ranade, Infotech SA, Inc.,
Dr. Elizabeth Williams, Supercomputing Research Center,

and to:

Ms. Nicki Fritz, the conference coordinator,
Westover Consulting for conference arrangements,

and Mr. Len Blasso and Ms. Ann Lipscomb for their help with the production of this document.

We are grateful to Mr. Laurence Lueck, President of Magnetic Media Information Services, for permission to reproduce the David-and-Goliath cover art from Volume XIII, Number 1, of the *Magnetic Media International Newsletter.*

Ben Kobler, NASA/GSFC
John Berbert, NASA/GSFC
P C Parthasarathan, Hughes STX Corporation

TABLE OF CONTENTS

Volume I

TABLE OF CONTENTS (Continued)

Volume I (Continued)

TABLE OF CONTENTS

Volume II

TABLE OF CONTENTS (Continued)

Volume II (Continued)

Mass Storage System Reference Model:
Version 4

Developed by the IEEE Technical Committee on Mass Storage Systems and Technology

Edited by:

Sam Coleman
Lawrence Livermore National Laboratory

Steve Miller
SRI International

Sam Coleman
Lawrence Livermore National Lab
Mail Stop L-50
P. O. Box 808
Livermore, CA 94550

1. Preface

The purpose of this reference model is to identify the high level abstractions that underlie modern storage systems. The information to generate the model was collected from major practitioners who have built and operated large storage facilities, and represents a distillation of the wisdom they have acquired over the years. The model provides a common terminology and set of concepts to allow existing systems to be examined and new systems to be discussed and built. It is intended that the model and the interfaces identified from it will allow and encourage vendors to develop mutually compatible storage components that can be combined to form integrated storage systems and services.

The reference model presents an abstract view of the concepts and organization of storage systems. From this abstraction will come the identification of the interfaces and modules that will be used in IEEE storage system standards. The model is not yet suitable as a standard; it does not contain implementation decisions, such as how abstract objects should be broken up into software modules or how software modules should be mapped to hosts; it does not give policy specifications, such as when files should be migrated; does not describe how the abstract objects should be used or connected; and does not refer to specific hardware components. In particular, it does not fully specify the interfaces.

A storage system is the portion of a computing facility responsible for the long-term storage of large amounts of information. It is usually viewed as a shared facility and has traditionally been organized around specialized hardware devices. It usually contains a variety of storage media that offer a range of tradeoffs among cost, performance, reliability, density, and power requirements. The storage system includes the hardware devices for storing information, the communication media for transferring information, and the software modules for controlling the hardware and managing the storage.

The size and complexity of this software is often overlooked, and its importance is growing as computing systems become larger and more complex. Large storage facilities tend to grow over a period of years and, as a result, must accommodate a collection of heterogeneous equipment from a variety of vendors. Modern computing facilities are putting increasing demands on their storage facilities. Often, large numbers of workstations as well as specialized computing machines such as mainframes, mini-supercomputers, and supercomputers are attached to the storage system by a communication network. These computing facilities are able to generate both large numbers of files and large files, and the requirements for transferring information to and from the storage system often overwhelms the networks.

The type of environment described above is the one that places the greatest strain on a storage system design, and the one that most needs a storage system. The abstractions in the reference model were selected to accommodate this type of environment. While they are also suitable for simpler environments, their desirability is perhaps best appreciated when viewed from the perspective of the most complicated environment.

There is a spectrum of system architectures, from storage services being supplied as single nodes specializing in long-term storage to what is referred to as "fully distributed systems". The steps in this spectrum are most easily distinguished by the transparencies that they provide, where they are provided in the site configuration, and whether they are provided by a site administrator or by system management software. The trend toward distributed systems is appealing because it allows all storage to be viewed in the same way, as part of a single large, transparent storage space that can be globally optimized. This is especially important as systems grow more complex and better use of storage is required to achieve satisfactory performance levels. Distributed systems

also tend to break the dependence on single, powerful storage processors and may increase availability by reducing reliance on single nodes.

1.1 Transparencies

Many aspects of a distributed system are irrelevant to a user of the system. As a result, it is often desirable to hide these details from the user and provide a higher-level abstraction of the system. Hiding details of system operation or behavior from users is known as providing transparency for those details. Providing transparency has the effect of reducing the complexity of interacting with the system and thereby improving the dependability, maintainability, and usability of applications. Transparency also makes it possible to change the underlying system because the hidden details will not be embedded in application programs or operating practices.

The disadvantage of using transparency is that some efficiency can be lost in resource usage or performance. This occurs because the mechanism that provides the transparency masks semantic information and causes the system to be used conservatively. High-performance data base systems, for example, may need to organize disk storage directly and schedule disk operations to gain performance, rather than depend on lower-level file systems with their own structure, scheduling, and policies for caching and migration.

There is a range of support that can be provided for distributed systems in a computer network. A system with few transparencies is often called a networked system. The simplest kind of networked system provides utilities to allow a program to be started on a specified host and information to be transferred between specified storage devices. Examples include TELNET and FTP, respectively. This type of system rarely provides support for heterogeneity. At the other end of the spectrum are fully distributed systems that provide many transparencies. An example is LOCUS. In distributed systems, a goal is for workstations to appear to have unlimited storage and processing capacities.

System and application designers must think carefully about what transparencies will be provided and whether they will be mandatory. It is possible for applications to provide certain transparencies and not others. Fundamental transparencies can be implemented by the system, saving each user from re-implementing them. A common implementation will also improve the likelihood that the transparency will be implemented efficiently

The common transparencies are:

Access
Clients do not know if objects or services are local or remote.

Concurrency
Clients are not aware that other clients are using services concurrently.

Data representation
Clients are not aware that different data representations are used in different parts of the system.

Execution
Programs can execute in any location without being changed.

Fault
Clients are not aware that certain faults have occurred.

Identity
Services do not make use of the identity of their clients.

Location
Clients do not know where objects or services are located.

Migration
Clients are not aware that services have moved.

Naming
Objects have globally unique names which are independent of resource and accessor location.

Performance
Clients see the same performance regardless of the location of objects and services (this is not always achievable

unless the user is willing to slow down local performance).

Replication
Clients do not know if objects or services are replicated, and services do not know if clients are replicated.

Semantic
The behavior of operations is independent of the location of operands and the type of failures that occur.

Syntactic
Clients use the same operations and parameters to access local and remote objects and services.

Some of the transparencies overlap or include others.

With this in mind, it is incumbent upon the Storage System Standards Working Group to identify interfaces and modules that are invariant from single storage nodes to fully distributed systems. Many sites are not likely to embrace fully distributed systems in a single step. Rather, they are likely to evolve gradually as growing system size and complexity dictate and as vendors make available products supporting fully distributed systems.

1.2 Requirements

Modern computing facilities are large and complex. They contain a diverse collection of hardware connected by communication networks, and are used by a wide variety of users with a spectrum of often-conflicting requirements. The hardware includes a range of processors from personal computers and workstations to mainframes and supercomputers, and many types of storage devices such as magnetic disks, optical disks, and magnetic tapes. This equipment is typically supplied by a variety of vendors and, as a result, is usually heterogeneous. Both the hardware characteristics and the user requirements make this type of facility extremely complicated.

To insure that the reference model applies to many computer environments, the IEEE Technical Committee on Mass Storage Systems and Technology identified the following requirements:

- The model should support both centralized and distributed hierarchical, multi-media file systems.
- The model should support the simplest randomly addressable file abstraction out of which higher level file structures can be created (e.g., a segment of bits or bytes and a header of attributes).
- Where the defined services are appropriate, the model should use national or international standard protocols and interfaces, or subsets thereof.
- The model should be modular such that it meets the following needs:
  - The modules should make sense to produce commercially.
  - It should be reasonable to integrate modules from two or more vendors.
  - The modules should integrate with each other and existing operating systems (centralized and distributed), singly or together.
  - It should be possible to build hierarchical centralized or distributed systems from the standard modules. The hierarchy might include, for example, solid state disks, rotating disks (local and remote), an on-line library of archival tape cartridges or optical disks, and an off-line, manually-operated archival vault.
  - Module interfaces should remain the same even though implementations may be replaced and upgraded over time.
  - Modules should have standardized interfaces hiding implementation details. Access to module objects should only be through these interfaces. Interfaces should be specified by the abstract object data structures visible at those interfaces.
  - Module interfaces should be media independent.

- File operations and parameters should meet the following requirements:
  - Access to local and remote resources should use the same operations and parameters.
  - Behavior of an operation should be independent of operand location.
  - Performance should be as independent of location as possible.
  - It should be possible to read and write both whole files and arbitrary-sized, randomly-accessible pieces of files.
  - The model should separate policy and mechanism such that it supports standard as well as vendor- or site-specific policy submodules and interfaces for access control, accounting, allocation, site management, security, and migration.
  - The model should provide for debugging, diagnostics, and maintenance.
  - The model should support a request/reply (transaction) oriented communication odel.
  - Request and data communication associations should be separated to support high speed direct source to destination data channels.
  - Transformation services (e.g. translation, check summing, encryption) should be supported.
- The model should meet the following naming requirements:
  - Objects should have globally unique, machine-oriented names which are independent of resource and access location.
  - Each operating system or site environment may have a different human-oriented naming system, therefore human- and machine-oriented naming should be clearly separated.
  - Globally unique, distributively generated, opaque file identifiers should be used at the client-to-storage-system interface.
- The model should support the following protection mechanism requirements:
  - System security mechanisms should assume mutual suspicion between nodes and networks.
  - Mechanism should exist to establish access rights independent of location.
  - Access list, capability or other site, vendor, or operating system specific access control should be supportable.
  - Security or privacy labels should exist for all objects.
- The model should support appropriate lock types for concurrent file access.
- Lock mechanisms for automatic migration and caching (i.e., multiple copies of the same data or files) should be provided.
- The model should provide mechanisms to aid recovery from network, client, server crashes and protection against network or interface errors. In particular, except for file locks, the file server should be stateless (e.g., no state maintained between "open" and "close" calls).
- The model should support the concept of fixed and removable logical volumes as separate abstractions from physical volumes.
- It should be possible to store one or many logical volumes on a physical volume, and one logical volume should be able to span multiple physical volumes.

2 Introduction

2.1 Background

From the early days of computers, "storage" has been used to refer to the levels of storage outside the central processor. If "memory" is differentiated to be inside the central processor and "storage" to be outside, (i.e., requiring an input-output channel to access), the first level of storage is called "primary storage" (Grossman 89). The predominant technology for this level of storage has been magnetic disk, or solid-state memory configured to emulate magnetic disks, and will remain so for the foreseeable future in virtually every size of computer system from personal computers to supercomputers. Magnetic disks connected directly to I/O channels are often called "local" disks while magnetic disks accessed through a network are referred to as "remote" or "central" disks. Sometimes a solid-state cache is interposed between the main memory and primary storage. Because networks have altered the access to primary storage we will use the terms "local storage" and "remote storage" to differentiate the different roles of disks.

The next level of data storage is often a magnetic tape library. Magnetic tape has also played several roles:

- On-line archive known as "long term storage" (e.g., less active storage than magnetic disk),
- off-line archival storage (possibly off-site),
- backup for critical files, and
- as an I/O medium (transfer to and from other systems).

Magnetic tape has been used in these roles because it has enjoyed the lowest cost-per-bit of any of the widely used technologies. As an I/O medium, magnetic tape must conform to standards such that the tape can be written on one system and read on another. This is not necessarily true for archival or backup storage roles, where nonstandard tape sizes and formats can be used, even though there are potential disadvantages if standards are not used even for these purposes.

In the early 1970s nearly every major computer vendor, a number of new companies, and vendors not otherwise in the computer business, developed some type of large peripheral storage device. Burroughs and Bryant experimented with spindles of 4-ft diameter magnetic disks. Control Data experimented with 12 in. wide magnetic tape wrapped nearly all the way around a drum with a head per track. The tape was moved to an indexed location and stopped while the drum rotated for operation. (Davis 82 presents an interesting comparison of devices that actually got to the marketplace.)

Examples of early storage systems are the Ampex Terabit Memory (TBM) (Wildmann 75), IBM 1360 Photostore (Kuehler 66), Braegan Automated Tape Library, IBM 3850 Mass Storage System (Harris 75, Johnson 75), Fujitsu M861, and the Control Data 38500. One of the earliest systems to employ these devices was the Department of Defense Tablon system (Gentile 71), which made use of both the Ampex TBM and the IBM Photostore. Much was learned about the software requirements from this installation.

The IBM 1360, first delivered in the late 1960s, used write-once, read-many (WORM) chips of photographic film. Each chip measured 1.4 x 2.8 in. and stored 5 megabits of data. "File modules" were formed of either 2250 or 4500 cells of 32 chips each. The entire process of writing a chip, photographically developing it, inserting the chip in a cell, and a cell in a file module, storing and retrieving for read, etc., was managed by a control processor similar to an IBM 1800. The complex chemical and mechanical processing required considerable maintenance expertise and, while the Photostore almost never lost data, the maintenance cost was largely responsible

for its retirement. A terabit system could retrieve a file in under 10 seconds.

The TBM, first delivered in 1971, was a magnetic tape drive that used 2-inch-wide magnetic tape in large 25,000-foot reels. Each reel of tape had a capacity of 44 gigabits and a file could be retrieved, on the average, in under 17 seconds. With two drives per module, a ten module (plus two control modules) system provided a terabit of storage. The drive was a digital re-engineering of broadcast video technology. The drive connected to a channel through a controller, and cataloging was the responsibility of the host system.

The Braegan Automated Tape Library was first delivered in the mid 1970s and consisted of special shelf storage housing several thousand half-inch magnetic tape reels, a robotic mechanism for moving reels between shelf storage and self-threading tape drives, and a control processor. This conceptually simple system was originally developed by Xytecs, sold to Calcomp, and then to Braegan. In late 1986, the production rights were acquired by Digital Storage Systems, Longmont, Colorado. Sizes vary, but up to 8,000 tape reels (9.6 terabits) and 12 tape drives per cabinet are fairly common.

The IBM 3850 (Johnson 75, Harris 75) used a cartridge with a 2.7-inch-wide, 770-in. long magnetic tape. A robotic cartridge handler moved cartridges between their physical storage location (sometimes called the honeycomb wall) and read/write devices. Data accessed by the host was staged to magnetic disk for host access. Destaging the changed pages (about 7 megabits) occurred when those pages became the least recently used pages on the staging disks. Staging disks consisted of a few real disk devices, which served as buffers to the entire tape cartridge library. The real disks were divided into pages and used to make up many virtual disk devices that could appear to be on-line at any given time.

Manufactured by Fujitsu and marketed in this country by MASSTOR and Control Data the M861 storage module uses the same data cartridge as the IBM 3850; however, it is formatted to hold 175 megabytes per cartridge. The M861 holds up to 316 cartridges and provides unit capacity of 0.44 terabits. The physical cartridges are stored on the periphery of a cylinder, where a robotic mechanism picks them for the read-write station. The unit achieves about 12-second access time and 500 mounts/dismounts per hour.

A spectrum of interconnection mechanisms was described (Howie 75) that included:

- The host being entirely responsible for special hardware characteristics of the storage system device,
- the device characteristics being translated (by emulation in the storage system) to a device known by the host operating system, and
- the storage system and host software being combined to create a general system.

This has sometimes been termed moving from tightly coupled to loosely coupled systems. Loosely coupled systems use message passing between autonomous elements.

The evolution of the architectural view of what constitutes a large storage system has been shaped by the growth in sheer size of systems, more rapid growth of interactive rather than batch processing, the growth of networks, distributed computing, and the growth of personal computers, workstations, and file servers.

Many commercial systems have tracked growth rates of 60-100% per year over many years. As systems grow, a number of things change just because of size. It becomes difficult for large numbers of people to handle tape reels, so automating the fetching and returning and the mounting and dismounting of reels becomes important. As size increases, it also becomes more difficult for humans to decide which devices to use for load balancing.

Because of this growth, early users of storage systems were forced to do much of the systems integration in their own site environments. Large portions (software and hardware) of many existing systems (Gentile 71, Penny 73, Fletcher 75, Collins

82, Coleman 84) were developed by user organizations that were faced with the problem of storing, retrieving, and managing trillions of bits and cataloging millions of files. The sheer size of such storage problems meant that only organizations such as government laboratories, which possessed sufficient systems engineering resources and talent to complete the integration, initially took on the development task. These individualized developments and integrations resulted in storage systems that were heavily intertwined with other elements of each unique site.

These systems initiated an evolution in storage products in which three stages are readily recognizable today. During the first stage, a storage system was viewed as a very large peripheral device serving a single system attached to an I/O channel on a central processor in the same manner as other peripheral devices. Tasks to catalog the files and free space of the device, manage the flow of data to and from it, take care of backup and recovery, and the many other file management tasks, were added as application programs within the systems environment. Many decisions, such as when to migrate a file, were left to the user or to a manual operator. If data was moved from the storage system to local disk, two host channels (one for each device) were required plus a significant amount of main memory space and central processing capability (Davis 82).

During this stage, the primary effort in design was machine-room automation to reduce the need to manually mount and dismount magnetic tapes.

The second stage (late 1970s to present) has been characterized by centralized shared service that takes advantage of the economies of scale and provides file server nodes to serve several, perhaps heterogeneous, systems (Svobodova 84).

This stage of the storage system evolution is the one that is most prevalent today. The storage system node entails using a control processor to perform the functions of the reference model in a storage hierarchy (O'Lear 82). The cost of the storage system makes it desirable to share these centralized facilities among several computational systems rather than provide a storage system for each computational system. This is especially true when supercomputers become a part of the site configuration.

This approach to providing storage has several advantages:

- The number of processors that have access to a file is larger than that which can share a peripheral device. (This type of access is not the same as sharing a file, which implies concurrent access.)
- Multiple read-only copies of data can be provided, circumventing the need for a large number of processors having access to common storage devices.
- Processors of different architectures can have access to common storage, and therefore to common data, if they are attached to a network and use a common protocol for bit-stream transfer.
- The independence between the levels of storage allows the inclusion of new storage devices as they become commercially available.

Some of the earliest systems in this shared service stage were at the Lawrence Berkeley National Laboratory (Penny 70) and Lawrence Livermore National Laboratory (LLNL) (Fletcher 75, Watson 80).

The Los Alamos Common File System (Collins 82, McLarty 84) and the system at the National Center for Atmospheric Research (Nelson 87, O'Lear 82) are more recent examples of shared, centralized storage system nodes.

The third stage is the emerging distributed system. An essential feature of a distributed system (Enslow 78, Watson 81a, 84, 88) is that the network nodes are autonomous, employing cooperative communication with other nodes on the network. The control processors of storage systems developed during this stage provide this capability.

The view of a storage system as a distributed storage hierarchy is neither a device nor a single service node, but is the integration of distributed computing systems and storage system architectures with the elements that provide the storage service distributed throughout the system. The distributed computing community has been very interested in the problems of providing file management services, albeit generally on smaller systems (Almes 85, Birrell 82, Brownbridge 82, Donnelley 80, Leach 82, Svobodova 84, Watson 81a). Probably the best known example at the workstation level is the SUN Microsystems "network file server" (Sandberg 85).

Several elements are necessary for a system to be classed as "distributed" (Enslow 78):

- A multiplicity of general-purpose resource elements,
- the distribution of these elements, logically and physically,
- a distributed (network) operating system,
- system transparency (service requests by name only), and
- cooperative communication among elements (nodes).

Achieving all of these elements sounds difficult and expensive. The motivations most often cited are extensibility, availability, and costly resource sharing (LeLann 81). Readily extensible systems permit the "hot wiring" necessary in large systems that can no longer afford downtime for cabling in new elements. Extensibility also means that individual elements can be upgraded without disrupting the entire system. System availability is obtained by replicating system elements in a way that permits graceful degradation. Sharing costly elements occurs through communications and networking.

The issues involved in designing distributed systems with the characteristics outlined above were discussed by Dr. Richard W. Watson of the Lawrence Livermore National Laboratory at the Eighth IEEE Mass Storage Symposium in Tucson, Arizona, May 1987. He stated (Watson 87) that the long-range goal is to design systems in which "mainframes, minicomputers, workstations, networks, multiple levels of storage, and input/output systems are viewed as elements of a logically single distributed computer whose resources are managed by and accessed through a single distributed operating system."

Individual operating systems have their own way of handling files. One reason for requiring a distributed operating system is to provide a single logical file and naming system. This distributed file system should be accessed by name only; that is, the naming and heterogeneous features of different component parts should be transparent to the user. Logically, the distributed storage system should have infinite capacity and unlimited file size. This is obtained through the use of migration among the distributed storage elements that make up the storage hierarchy. The different levels of storage probably have different storage characteristics and costs.

Other design goals include high reliability and availability, high performance (low delay and high throughput), mandatory and discretionary access control, file sharing and safe concurrent access (Lantz 85), and accounting and administrative controls (Mullender 84).

It now appears that an attainable goal is to design interconnected systems, whose subsystems can be produced by a number of vendors, such that the file service is uniform from the user's local level through all levels of the on-line hierarchy to shelf storage. Internally, the distributed hierarchical storage system will consist of multiple levels of storage such as bulk semiconductor memories, magnetic disks, magnetic tape, optical disks, automated media libraries, and manual vaults. Such a system is currently under development at Lawrence Livermore National Laboratory (Coleman 84, Foglesong 90, Gary 90, Hogan 90).

2.2 Motivation

The central architectural features of the reference model and the motivation for them can be summarized as follows:

- An object-oriented description allows the identification of a modular set of standard services and standardized client/server interfaces. The reference model servers are potentially viable commercial products and are building blocks for higher-level services and recursive integration in centralized, shared, or distributed hierarchical storage systems. This integration can be done within single-vendor systems, by third-party, value-added system integrators, or by end-user organizations. The object-oriented modularity hides implementation details, allowing many possible implementations in support of the standard abstract objects and interfaces (Booch 86).

- For the storage system to be integrated with applications and operating systems supporting many different internal file structures, the abstract object visible to storage-system clients is an uninterpreted string of bits and a set of attributes.

- For the storage system to be integrated with applications and operating systems supporting many different internal file structures, the abstract object visible to storage-system clients is an uninterpreted string of bits and a set of attributes.

- The separation of human-oriented naming from machine-oriented file identifiers allows integration with current and future operating systems and site-dependent naming systems. This implies separation of the name server as a separate module associated with the reference model (Watson 81b).

- The separation of access rights control as a site-specific module with a standard interface to the storage system accommodates the many operating systems and site-dependent access control mechanisms in existence.

- The separation of the request and data communication paths supports existing practices and the need for third-party control of transfers between two entities by direct data transfers from source to sink, as well as data transfer redirection and pipelining through such data transformations as encryption, compression, and check-summing.

- The separation of the site manager allows site-dependent policies and status to be managed. Provision is made for standard site-management interface functionality.

- Inclusion of a migration server within the file server allows each file server to be self-contained and file-migration policies for each server to be established separately. It also facilitates building a hierarchical storage system supporting automatic migration between servers. The general goal is to cache the most active data on the fastest storage servers and the least active on storage servers with the lowest cost-per-bit medium.

It is envisioned that the modules of the reference model can be integrated in various combinations to support a variety of storage needs from single storage systems to distributed, hierarchical systems supporting automatic file migration. Vendors can build and market individual standard modules or integrated systems supporting standard interfaces and functionality. Hopefully, the development of standards will increase markets and lead to modules and systems manufactured in larger numbers, thus reducing costs as a result of mass production economies.

To better understand the modularization and the requirements placed on interfaces but not to force a particular design philosophy, the discussion in this document does not restrict itself to external interfaces and services as might be expected of a reference model. The intent is not to standardize the internal structure of modules, since this is implementation- and vendor-specific, but to provide additional understanding to aid the model building, interface standardization, and implementation processes.

2.3 Reference Model Architecture

2.3.1 Abstract Objects

To follow the description of the reference model, there are several concepts that should first be established. These concepts employ the properties of abstract objects (Watson 81a), which have been succinctly listed as :

- Objects are an instance of a type (file, process, directory, account, etc.). As such, an object type is defined by:
  - An identifier.
  - A logical representation visible at an interface (e.g., a logical representation of a file is a set of attributes and a data segment of uninterpreted bits).
  - A set of operations or functions and associated parameters presented at the interface to create, destroy, or manipulate the object.
  - Specification of sequences of operations that are allowed.
- Objects are managed by servers. There can be many servers for a given type (e.g., there can be many file managers).
- Objects are of two basic classes, active and passive. To be manipulated, passive objects (such as files, directories, or accounts) must be acted on by requests from active objects presented at the server interfaces. Active objects, which are mainly processes, can directly change aspects of their own representation. Active objects can play either or both of two roles, a client role accessing a service, and a server role providing a service or managing a type of object.
- Objects are named via an identification scheme with a machine-oriented name that is unique throughout an environment. This identification scheme may be used in conjunction with protection and resource management schemes. Human-oriented naming is implemented by separate name servers that associate mnemonic, human-oriented names with the machine-oriented object identifiers. Higher-level file services might integrate the name and file services.
- Access to objects is controlled by the server through access lists, capabilities, or other techniques.

2.3.2 Client/Server Properties

The client/server model (Watson 81a) is an *object-oriented* paradigm. Simply stated, both clients and servers are active abstract objects in which the client requests services from the server through a specified interface. The word *client* is used to mean the program that accesses some service. The word *user* is reserved to mean the human at the terminal. A client is an *agent* of a user. The *server* is a provider of a service. Access to server-supported objects or services is only through defined server interfaces, thus hiding implementation details to provide transparency. Both the client and the server may be processes or collections of processes. These processes are not necessarily associated with any particular host machine. We describe the client/server interactions in terms of messages, but it is understood that local or remote procedure calls (Birrell 82) or other communications paradigms are possible.

A server may be thought of as a collection of one or more *tasks* or *processes* (concurrently executing instruction streams). A server may include request processing and other tasks supporting concurrent handling of requests from many clients. Clients may also be constructed as many cooperating, concurrent tasks.

Client and server processes interact by sending each other messages, in the form of requests and replies. A message is the smallest unit of data that can be sent and received between a pair of correspondents for a meaningful action to take place. A client process accesses a resource by sending requests containing the operation specification and appropriate parameters from one of its ports to a server port. A given process can operate in both server or client roles at different times (Watson 84). For example, during the migration of files,

a file server that manages magnetic disks can play the role of client to a file server that manages magnetic tapes. Another example is a name server that stores its catalogs in a file server.

A distinction is drawn between the words *server* and *service*. A service may include several servers (Svobodova 84). For example, a directory service might be implemented by having separate name servers for objects such as files and for other objects such as users, addresses or printers. On the other hand, one might implement a universal directory to provide the whole directory service (Lantz 85), or one might choose to implement the file service defined by the ISO-OSI Virtual Filestore (DIS 8571), where this reference model serves the unstructured file segment. Thus a complete file service will likely consist of name servers and multiple file servers.

2.3.3 Reference Model Modules

The primary reference model modules, shown in Figure 1, are:

- The *bitfile** *server*, which handles the logical aspects of bitfile storage and retrieval,
- the *storage server*, which handles the physical aspects of bitfile storage, and
- the *physical volume repository*, which provides manually or robotically retrievable shelf storage of physical media volumes.

Closely related to these modules are:

- The *bitfile client*, which is the programmatic agent of the user required to convert user desires into bitfile requests to the bitfile server and data transfer commands to the bitfile mover,
- the *bitfile mover*, which provides the components and protocols for high-speed data transfer,
- the *name server*, which provides the retention of bitfile IDs and the conversion of human-oriented names to bitfile IDs, and
- the *site manager*, who monitors operations, collects statistics, and establishes policy and exerts control over policy parameters and site operation.

These modules are not directly associated with any particular hardware or software products. The modularity of the reference model defines a virtual store for bitfiles. The storage system can be implemented with many levels of storage hierarchy, including a physical volume repository. The structure of the model permits standard interfaces and multiple instances of modules, and thus should facilitate more economical implementation of many forms of storage architectures. There may be many different instances of bitfile server and storage server combinations in which storage servers need not be of the same technology and can form a hierarchy.

The bitfile client represents the program object or agent that accesses bitfiles. This agent is not the application but acts for the application. The bitfile client can take many forms depending on how the storage system is implemented and integrated into a particular user environment; it might be one or more application programs or be functionally supported within an operating system to facilitate access to storage. The bitfile client may run on personal computers, workstations, or on larger machines. The bitfile client can also be a part of a data acquisition system needing the services of a storage system. The bitfile client can locate bitfiles by use of a name server. The user's human-oriented bitfile names are mapped to bitfile IDs and bitfile server addresses by the name server.

It is the interaction of the bitfile client with the bitfile server, the bitfile server interaction with the storage servers, and the storage server interaction with the physical volume repository that are of particular interest. There may be any number of bit-

---

*The word "bitfile" was coined by the IEEE-CS Technical Committee on Mass Storage Systems and Technology to refer to a bit string that is completely unconstrained by size and structure; it was coined to relieve those who worked on the model from being bound by any particular file management system.

file clients in the general system environment of a site. Furthermore, bitfile servers or other entities of the total storage environment, such as name servers, site managers, or migration modules, may operate in the role of bitfile clients when they need storage service themselves.

A bitfile server handles the logical aspects of the storage and retrieval of bitfiles. The bitfile server's abstract object is the bitfile, identified by a globally unique machine-oriented bitfile ID. A bitfile is a set of attributes (state fields) and an uninterpreted, logically contiguous segment of data bits.

A bitfile server may keep track of the bitfiles stored in one or more storage servers. A single bitfile server may control a hierarchy and need the services of several storage servers. As an alternative, a single bitfile server may handle the bitfiles in a single level of the storage hierarchy or a single storage technology; multiple bitfile server-storage server pairs simplify extensibility and evolution.

The bitfile server accepts requests from bitfile clients to create, destroy, store, and retrieve bitfiles, and to modify and interrogate the bitfile attributes needed for system management. The bitfile server contains a request processor to parse the requests and control the sequence of actions necessary to fulfill the requests. Before permitting access to a bitfile, the bitfile server authenticates the access rights of the requestor.

The bitfile server communicates action commands to the storage server. Each bitfile server contains a migration manager to prevent overflow of the storage space for which it is responsible. The migration manager knows which bitfile server is used to offload bitfiles, as established by the site manager and by migration and caching policies.

The storage server handles the physical aspects of bitfile storage and retrieval, and presents the image of perfect media to the bitfile server. (The *capacity* of the media, influenced by imperfections, may be visible to the bitfile server.) The storage server's abstract objects, as seen by the bitfile server, are logical volumes made up of an ordered set of bit string segments.

Physical volumes and volume serial numbers are not visible to the client. The site manager, however, may have privileged storage server commands where physical volumes and devices are treated as visible objects of the storage server. Volumes and devices are identified by volume and device IDs.

Bit stream segments are identified by segment descriptors. These segments represent how the storage server has allocated space for the bitfile data blocks. Each segment is identified by a descriptor generated by the storage server, and the ordered set is retained as a bitfile attribute by the bitfile server. The storage server must internally map bit string segments to real physical volumes (removable or not) and addresses where the bitfiles are stored, provide read/write (and some error management) of those volumes, and be able to access a bitfile mover to transmit and receive bitfile data blocks. One or more logical volumes may be mapped to a given physical volume, or one logical volume may be mapped to several physical volumes. Thus, a storage server contains storage devices, device-specific controllers that map bitfiles to bit string segments on physical volumes, and a means for handling and managing physical volumes on the storage devices.

The physical volume repository server manages a library that stores physical volumes such as tape reels, tape cartridges, or optical disk platters. Physical volumes, identified by physical volume IDs, are its abstract objects. A physical volume repository server can be used by one or more storage servers.

The site manager is a client process that can generate and send ordinary and privileged requests to the other servers to set policy parameters, install logical and physical volumes (import, export), obtain statistics and status, run diagnostics, etc.
The various clients and servers are interconnected through a communication service, which must handle all of the inter-

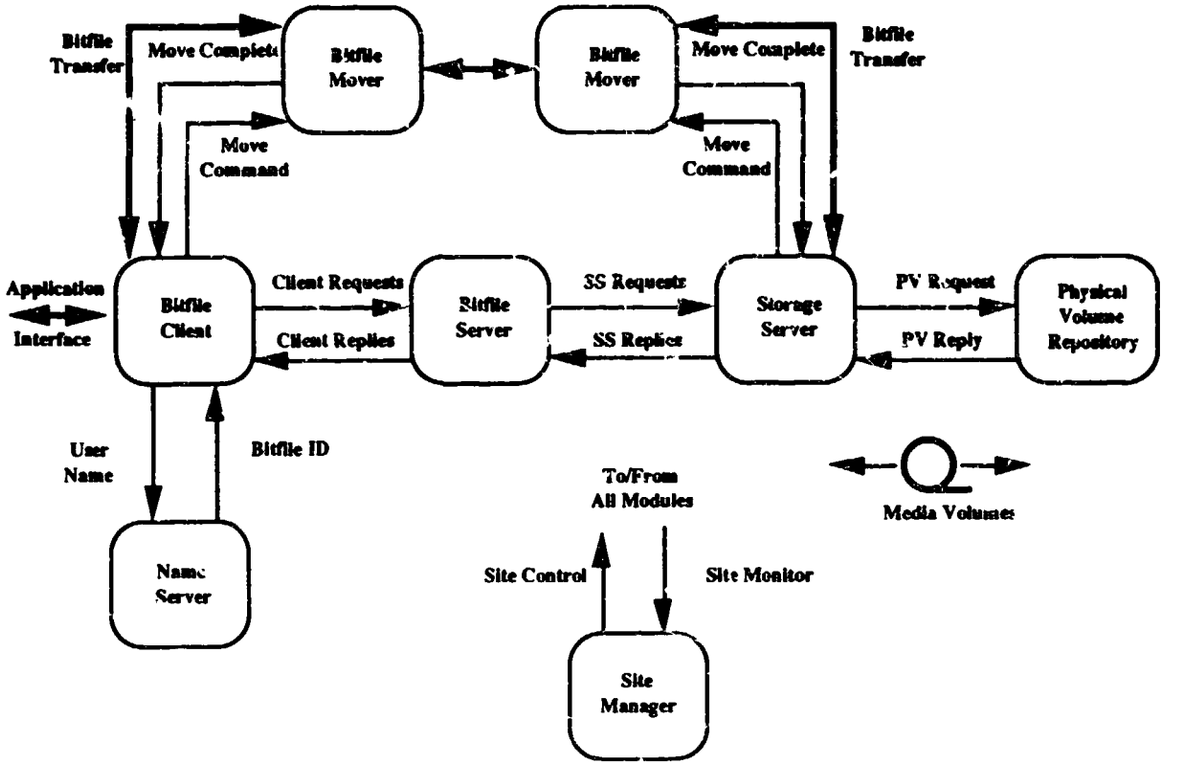


Figure 1

The Storage System Reference Model

process communications involved in requests and replies, as well as the high-speed transport of bitfiles. The elements of the communications service may be distributed through the many physical processors of the site. The reference model does not specify the details of this service but does assume its existence whether by procedure call, remote procedure call, or message passing. In particular, the model does not require homogeneity of this communication service. The communication service can include movement across I/O channels as well as across networks. The model assumes the ability to separate data movement from request movement. All that is required is that entities that must communicate are, in fact, able to do so and that standard interfaces are supported. The model has separated authentication details to allow each site to install the form appropriate to that site.

Existing storage systems often include a high-performance data path of some type, often specially designed, to handle the high-volume, high-speed data flow between the bitfile client and the storage server. In the reference model, the need for a high-speed data path is incorporated as part of the communication service referred to as the *bitfile mover*. This path has been separated from the request path to correspond to existing practice; to facilitate third-party control of transfers, to facilitate insertion of data transformations such as encryption, compression, and check-summing; and to support transfer redirection. If a general

network is used for the communication service, it has to account for this need for high transmission speed of bitfiles as well as the communication of requests and replies.

3. Detailed Description of the Reference Model

This description of the reference model discusses the entities shown in Figure 1 in more detail. In describing the functions accepted by each module, input parameters that are common to all functions are deleted for clarity. These parameters include the identification of the client making the request or other access-control information, accounting information, and a transaction identifier. Similarly deleted are the transaction identifier and the success/fail indication common to all responses. If functions fail, error information will replace the expected responses. Optional parameters that can be defaulted are enclosed in square brackets (Miller 88a, 88b).

3.1 Bitfile Client

The bitfile client is the collection of hardware and software at a user node within the site to permit that node to use the storage system. Bitfile clients are responsible for providing the storage system interface to users at the terminal or to application processes by:

- Translating user and application requests for storage services into bitfile server requests, and
- providing communication with the appropriate bitfile servers and movers as determined by the name server mapping.

The bitfile client may be library routines within the application, an interface process (local or remote), or routines within an operating system kernel, and it may combine the services provided by multiple bitfile servers, bitfile movers, and name servers to form the higher-level abstract objects of an integrated storage service. The syntax and semantics of messages that flow between the bitfile client and the bitfile server should be the same regardless of the type of bitfile client. These messages identify the bitfile to be acted upon and specify which of the basic commands and parameters are to be processed.

When a bitfile is created, the bitfile ID is generated by the bitfile server and passed back to the bitfile client for retention. When the user accesses an existing bitfile by name, the bitfile client obtains the bitfile ID from a name server or some other data base system.

Several arrangements of the bitfile client are possible. In the first arrangement, the "kernel view", all bitfiles are logically local. The bitfile client is a program in a processor with its own operating system and local storage. The storage and site-management capabilities of the local operating system are what the user sees with respect to how his bitfiles are handled. To the kernel of the operating system is added code that permits the operating system to determine if a bitfile being requested is locally stored or remote (Sandberg 85). If it is remote, this special code in the operating system kernel makes up the messages for the bitfile server and possibly for the name server. Alternatively, the local/remote distinction can be made in a library routine, and the library code can act as the bitfile client (Brownbridge 82). In either case, the bitfile is put into a local user buffer or is cached in an operating system buffer or a local bitfile and, except for a possible delay, the remote transfer is transparent to the user.

In a pure "client/server" view, all bitfiles are logically remote. Here, all references to bitfiles are translated into messages for a bitfile server, using routines in a library or other run-time support facility, such as a remote invocation system. The bitfile server might be local or remote; in a diskless workstation for example, the bitfile server is remote. Mapping human-oriented names to machine-oriented bitfile IDs is a separate, explicit step or is hidden in the run-time support facility (Svobodova 84, Watson 84).

In a third view, the systems that create and store bitfiles are separate from the systems that retrieve and process the data contained in the bitfiles. Such might be the relation-

ship between a data acquisition system that puts bitfiles into the storage system and the systems in a data processing center that use them. While there is no name server per se, some means must be provided to retain bitfile IDs when they are returned from the bitfile server and to pass them in some understandable way (e.g. using a data base management system) to the processing systems. Such systems must take care to back up their records; if the bitfile IDs are lost, the bitfiles become lost objects in the storage system.

3.2 Name Server

The development of distributed systems has caused a much more in-depth look at schemes for identifying objects. While the advent of distributed systems brought this about, the requirements recognized are not restricted to distributed systems. They apply to all systems, especially those that grow in size. Dealing properly with naming is central to achieving the location transparency needed in a distributed system. Thus, it is advantageous to look at some of the properties of identification schemes.

There are many possible ways to designate a desired object (Watson 81b):

- by an explicit name or address (object *x* or object at address *x*),
- by content (object with value or value expression *x*),
- by source (all *my* files),
- by broadcast identifier (all objects of a class)
- by group identifiers (participants in class *x*),
- by route (all objects found at the end of path *x*),
- by relationship to a given identifier (*all previous sequence numbers*), etc.

A useful informal distinction between three important classes of identifiers widely used in system design—names, addresses, and routes—is (Shoch 78):

- The name of a resource is what we seek,
- an address indicates where it is, and
- a route tells how to get there.

One should not examine such informal definitions too closely. Names, addresses, and routes can occur at all levels of the architecture. Names used in the inter-process communication layer have often been called such terms as ports, or logical or generic addresses. A human-oriented chain or path name can be thought of as a "route" through a directory. An important idea is that identifiers at different levels of the architecture referring to the same object must be bound together in contexts, statically or dynamically. Later they must be resolved using these contexts to locate the named objects.

There are important system benefits if every bitfile ID is unique (Leach 82). Less obvious are the system-wide ramifications of the total naming system, especially the choice of the particular mechanism used to create unique bitfile IDs and the mechanism to associate application-dependent, human-oriented names with them. Of the many goals and implications of identification schemes enumerated by (Watson 81b), the goals that are the most pertinent to the reference model are abstracted and discussed below.

The naming system should:

- Support at least two levels of identifiers, one convenient for people and one convenient for machines. The latter is the bitfile identifier. The former will be handled by site or operating system specification of the name servers or by imbedding a name service in a higher level file service.

  The separation of identifier levels is very important because a storage system must be integrated with many types of heterogeneous applications and operating and storage systems (centralized and distributed), each supporting its own form of human-oriented naming scheme. The reference model provides a clean separation of mechanism for these two levels of identifiers and allows their easy in-

tegration. (When the client is responsible for the use of the bitfile ID, there is the potential to create lost objects in a system and thus mechanisms must also be included to assist the system in identifying them so that the resources they use can be reclaimed.)

- Support distributed generation of machine-oriented, globally-unique bitfile identifiers. A variety of mechanisms are available to support this need (Leach 82, Mullender 84, Watson 81b). One mechanism is to include both a bitfile server ID and a time stamp in the identifier. This structure, containing node or server boundary information, is at most a hint to applications as to where to send access requests and should not restrict migration. A machine-oriented identifier is a bit pattern easily manipulated and stored by machines and may be directly useful with protection, resource management, and other mechanisms. A human-oriented identifier, on the other hand, is generally a character string that is readable by humans and that has mnemonic value. Directory path names are a common mechanism (McLarty 84).

- Provide a storage system viewed as a global space of identified objects rather than as a space of identified host computers containing locally-identified objects. Similarly, the identification mechanism should be independent of the physical connectivity or topology of the system. That is, the boundaries of physical components and the connection among them as a network, while technologically and administratively real, are invisible in object identifiers. Further, an object's name should be independent of client or server location. Users should be able to discover or influence an object's location.

- Support relocation of objects. The implication here is that there be at least two lower levels of identifiers and that the mapping and binding between them be dynamic. For example, bitfiles are expected to migrate. Therefore, the bitfile IDs should not contain storage addresses, and there must be mechanisms for updating the appropriate context (e.g. directories and tables) when objects are moved.

- Support use of multiple copies of the same object. For example, a file may be cached on disks at one or more hosts, on staging disks, or it may be stored on an archival volume. If the value of the object is only going to be read or interrogated, one set of constraints is imposed. If values are to be written or modified, tougher constraints must be imposed to achieve consistency between the contents of the copies. Policies of enforcement of such constraints are handled using the basic locking services specified by the reference model.

- Allow multiple local, user-defined (human-oriented) names for the same object by allowing multiple mappings of a given bitfile identifier within the services of one or more name servers.

- Support two or more resources sharing a single instance of a storage object without identifier conflicts.

- Minimize the number of independent identification systems needed across and within architectural levels.

3.3 Bitfile Server

A bitfile server (Falcone 88) handles the logical aspects of bitfiles that are physically stored in one or more storage servers of the storage system. As shown in Figure 2, the major components of a bitfile server are a bitfile server request processor, a bitfile descriptor manager and its descriptor table, a migration manager, a bitfile ID authenticator, and a space limit manager and its space limit table.

The bitfile server accepts requests for service on bitfiles from the bitfile client, site managers, migration manager, and other bitfile servers. A discussion of the operations that bitfile clients can request of the bitfile server regarding the bitfile follows. The function parameters are shown in Table 1.

3.3.1 Bitfile Server Commands

Abort
The client requests that a previous request be aborted.

Create
This request is used to establish a new entry in the bitfile server's descriptor table. The requestor must prove his right to do so, and when this is established, he receives a new bitfile ID from the bitfile server, which may then be saved for use when the bitfile is accessed later.

Destroy
The client requests that a bitfile descriptor be removed from the bitfile descriptor table. The space allocated to the bitfile within a storage server is

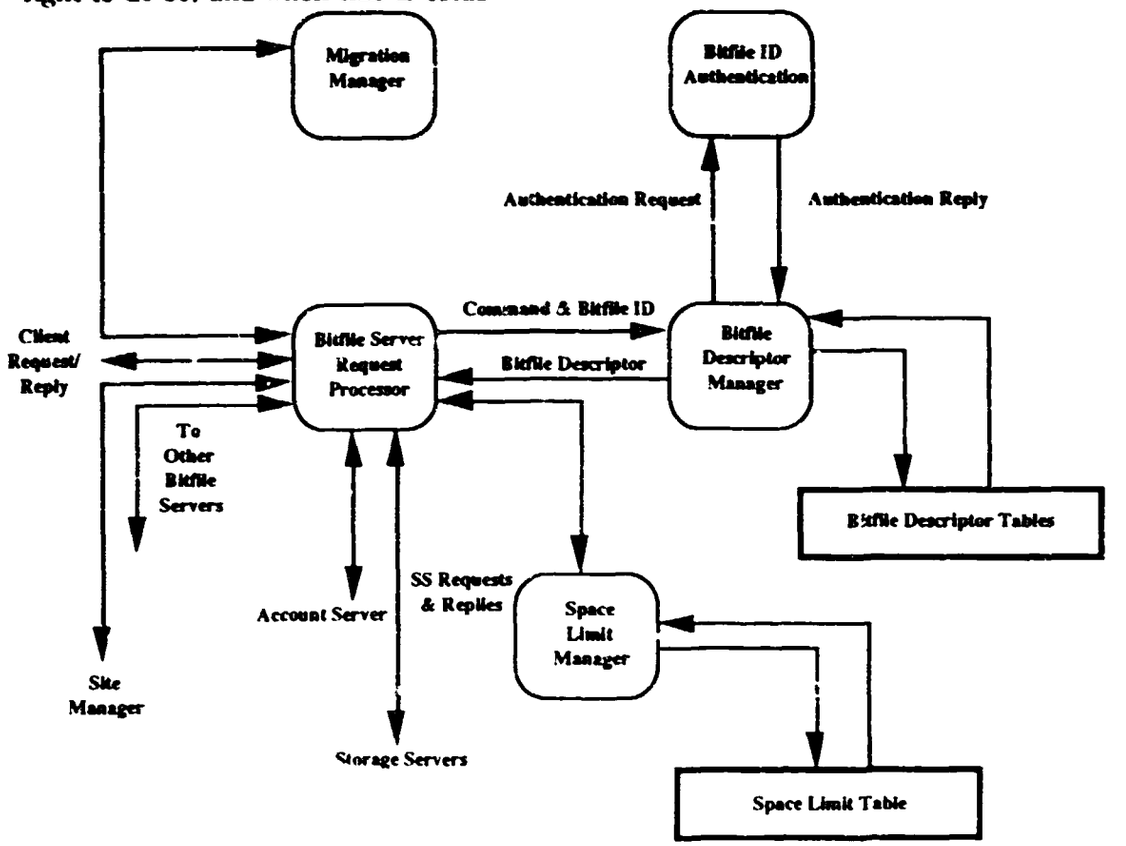


Figure 2

The Bitfile Server

deallocated and, if the medium can be rewritten, the storage server returns it to the free-space list.

Erase
This request erases data on a storage server with the specified erasure pattern. If only a segment is to be erased, then the client must specify the length of the segment and an optional offset field displacing the length into the bitfile.

Lock
The client requests that a bitfile be locked for read access or for read/write access.

Modify
The client requests a change in one or more attributes of a bitfile contained in the bitfile descriptor table.

Query
The client requests information about a bitfile or a bitfile's attributes from the bitfile descriptor table.

Retrieve
The client requests that a bitfile or a segment of a bitfile be moved from the storage server medium to the bitfile client or application buffer. If only a segment is to be moved, then the internal starting bit address (offset) and the bit string length of the segment to be transferred must be specified. (The data transfer is on a separate logical, and perhaps physical, path from the request/response path, the data block itself is not part of the response.)

Status
The client requests the status of a previous request made by the client. Although the way in which status is implemented is site dependent, generally a transaction ID must be generated to support the status request.

Store
The client requests that a bitfile data block be moved from a bitfile client or application buffer to the storage server medium. This request may include the ability to append a segment at the end of an existing bitfile or to update a physically specified segment of an existing bitfile. (The data block itself is not part of the request.)

Unlock
The client requests that any locks held be released.

The interface on which this list of commands can be sent is shown in Figure 1 as bitfile client requests and bitfile server replies.

The site manager will have additional privileged requests to control allocated space limits, examine all bitfile directory fields, set access control and migration policy parameters, etc.

3.3.2 Bitfile Request Processor

The bitfile server request processor accepts, parses, and executes request messages from:

- The bitfile clients, to store, retrieve, and manage bitfiles,
- the site manager, to provide monitoring and control,
- the migration manager, to move bitfiles to other bitfile servers, and
- other bitfile servers, to support migration.

The request processor is therefore essentially the sequencer and controller of actions within the bitfile server and the interface to the other storage system modules. Requests must be scheduled to provide the best possible response to the bitfile clients while optimizing the use of the available resources. Client-specified priority, bitfile size, and storage server availability may affect the request scheduling. These actions require a significant amount of processing. In executing a request, the request processor may interact with the bitfile descriptor manager to retrieve, create, or update bitfile attribute information, and with a storage server to allocate or release logical volume space for bitfile storage and to store and retrieve bitfiles. To select a storage server when a bitfile is created, the request processor must have information about the bitfile (bitfile size, the response desired, the protection and reliability desired, the type of storage desired, etc.) and must match this information with the characteristics of the available storage servers. Bitfile clients might be able to specify a specific storage server or logical volume as well.

Table 1
Bitfile Server Functions

Function	Parameters	Response
Abort	Transaction ID	
Create	[Initial length in bits] [Maximum length in bits] [Attribute Value/Name pair list]	Bitfile ID
Destroy	Bitfile ID	
Erase	Bitfile ID [Offset] [Length] Erasure pattern	Number of bits erased
Lock	Bitfile ID Lock type	
Modify	Bitfile ID New attribute name/value pairs	
Query	Bitfile ID Attribute name list	Attribute name/value pair list
Retrieve	Bitfile ID [Offset] [Length] Data transfer sink ID	Number of bits transferred
Status	Transaction ID	Transaction status
Store	Bitfile ID [Offset] [Length] Data transfer source ID	Number of bits transferred
Unlock	Bitfile ID	

The request processor is responsible, using the bitfile ID authenticator, for the security and integrity of the access to bitfiles, and for synchronizing the sharing of bitfiles through its locking services. The bitfile request processor collects accounting data from all affected sources regarding each transaction and sends them to the account service. The request processor also communicates with the space limit manager to determine that the resources assigned to a particular account are not overdrawn.

3.3.3 Bitfile Descriptor Manager and Descriptor Table

State and attribute information for each bitfile is kept in records in a descriptor table. Each record is called a *bitfile descriptor*. A descriptor manager provides an interface for the request processor to store, retrieve, and update bitfile descriptors. Bitfile descriptors are accessed using a bitfile ID as a key which is assigned by the descriptor manager when the bitfile descriptor is created.

A convenient way to classify bitfile descriptors is by origin and usage. Typical bitfile descriptor classes and some examples are:

- **Created and used by the bitfile client.**
  - comment
  - bitfile format
- **Created by the bitfile client and used by the bitfile server.**
  - access-control information
  - account ID
  - bitfile lifetime
  - desired level of redundancy
  - family attribute
  - maximum bitfile length
  - priority
  - security level
  - service class
  - storage class
  - type of storage desired
- **Created by the bitfile server and used by both the bitfile server and the bitfile client.**
  - access statistics
  - accounting statistics
  - bitfile allocated length
  - bitfile ID
  - bitfile length
  - creation time
- **Created and used by the bitfile server.**
  - last backup time
  - last migration time
  - location of backup copy
  - lock information
  - previous location
- **Created by the storage server and used by bitfile server.**
  - last device to write bitfile
  - location of bitfile

The importance of descriptor tables necessitates that backup and recovery be supported by the descriptor manager.

3.3.4 Bitfile ID Authenticator

The bitfile ID authenticator implements a mechanism, such as an access list or DES encryption used in a capability system, which protects the bitfile ID from being forged. It may also enforce security policy based on the security level of the bitfile, the request message, or the client. The authenticator is called by the descriptor manager when the bitfile ID is created to support the authentication mechanism and, when a request for access to the bitfile is received, to authenticate the bitfile ID presented by the client. If the access control is via an access control list, an identifier of the accessing entity (principal ID) must accompany the request and be checked against an access list that is kept, at least logically, in the bitfile descriptor. If access control is via a capability system, the bitfile ID may be encrypted along with some redundant and access-right information within the capability, and decrypted by the authenticator and compared against information in the descriptor when the bitfile is accessed. It is assumed that the authentication module can be added by a site or systems integrator since access control mechanisms and security policies are site-dependent (Jones 79b, Donnelley 80, Mullender 84).

3.3.5 Migration Manager

No single storage server now available can provide both the performance and large capacity often needed by a large storage system. A successful strategy is for a number of bitfile servers and their associated storage servers to be operated as a storage hierarchy.

A migration manager is associated with each bitfile server. The migration manager is responsible for maintaining enough free space on the logical volumes managed by its bitfile server (e.g. disk storage) to ensure that requests for new bitfiles can be honored. When the migration manager initiates a migration procedure, it first queries the bitfile descriptor manager for information about all of the bitfiles that might be migrated. This information might include the bitfile priority, size, locks, activity, idle time, and client-desired response. Bitfile clients may be given different degrees of control, by various site management policies, over the placement

of their bitfiles in the storage hierarchy. Using policy set by the site manager, the migration manager determines which bitfiles should be moved. Finally, the migration server sends requests to the bitfile server request processor to move the bitfiles to a bitfile server "lower" in the storage hierarchy. (Bitfiles move "up" in the hierarchy, toward higher-performance servers, as they are accessed; this movement is orchestrated by the bitfile server request processor.)

An alternate configuration may permit the migration manager to act as a third-party controller to initiate the request for a move. The separate request and data paths in the reference model allow data to move directly from a source storage server to a sink storage server, even though a third party initiated the transfer between the two bitfile servers. A request path may span two or more bitfile servers until the bitfile is located. To increase performance during retrieval, it may be desirable to establish a direct data transfer path, bypassing some storage servers, once the bitfile has been located. Such might be the case when bitfiles are accessed on very rare occasions and it is not economic to bring them back up the hierarchy.

3.3.6 Space Limit Manager

The space limit manager checks to see what logical volumes a given account, user, or user group is allowed to use; it controls space allocations, number of bitfiles allowed, or other policy parameters associated with space resource management that a given site may wish to enforce. The space limit table has entries for each account or principal ID for maximum and current space and bitfile limits.

3.4 Storage Server

A storage server (Savage 88) may best be visualized as an intelligent storage controller and its suite of storage devices. A storage server consists of a physical storage system (containing the physical bitfile-storage medium), a logical-to-physical volume manager, a physical device manager, a means of command authentication (unless it is a trusted component of the storage control processor), and some part of or intimate connection to the bitfile mover. A diagram of the storage server is shown in Figure 3.

The abstract objects of the storage server that are visible to the bitfile server are logical volumes and bit string segment descriptors. The descriptors of the space occupied by a bitfile form an ordered set of bit string segments identified by descriptors, each of which contains the logical volume ID, the starting point of the segment on the logical volume, and the length of the segment. The bit string segment descriptors are created by the storage server and stored in the descriptor tables of the bitfile server.

Each logical volume is considered to be a logical image of flawless media usable for storing bitfile data blocks, thus providing the separation of physical and logical space. Separation of logical and physical volumes supports segment relocation when media fail, where new storage devices are introduced, and when space utilization or transfer rate are optimized. Any media area that is unavailable for data storage because of flaw areas, formatting, control tracks, etc., is excluded from representation in the logical volume by the logical-to-physical mapping function.

The list of operations supported by the storage server is listed in Table 2. The site manager has a number of privileged operations including create, destroy, modify, and query of logical volumes, physical volumes, and physical devices.

3.4.1 Physical Storage System

A *physical storage system* consists of the devices used to read and write volumes and the drivers to control those devices (to position heads properly in relation to the media before reading or writing, etc.). The available physical storage systems cover a broad spectrum of characteristics in terms of random or sequential access, rewritable or write-once media, capacity, and performance.

3.4.2 Physical Device Manager

The *physical device manager* communicates with drivers in the physical storage system to load, unload, and position media volumes (it is the bitfile

mover that controls the actual transfer of data).

Physical device managers vary from simple modules associated with fixed-media devices, such as Winchester disks, to complex modules that deal with manually mounted volumes, as in systems with standard magnetic tape drives or automatically mounted volumes, such as in the IBM 3850 and the STK 4400 Automated Cartridge Library. In automated systems, the physical device manager communicates with a physical volume repository to request that physical volumes be mounted. The physical device manager maintains a mounted volume table to optimize mount requests. It schedules and executes requests in a manner that attempts to give the desired response to its clients while at the

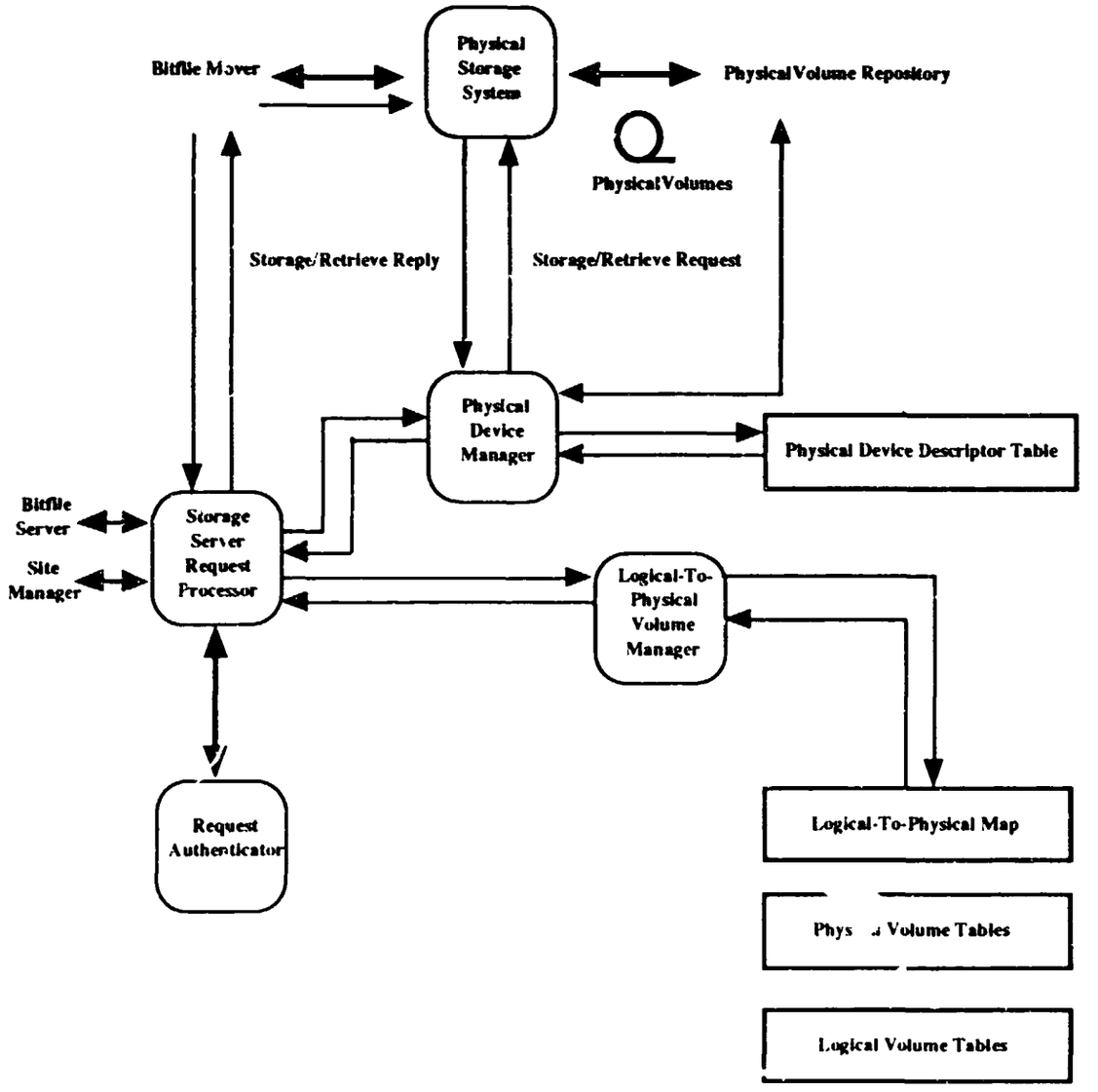


Figure 3

The Storage Server

same time making the best use of the storage system and communication resources. For example, it may be desirable to give higher priority to transfers for which volumes are already mounted or to small bitfile transfers, to limit the number of concurrent large bitfile transfers, or to use a client-specified priority.

3.4.3 Logical-to-Physical Volume Manager

The logical-to-physical volume manager maintains descriptors of attributes for each logical and physical volume and a set of tables to permit mapping the bit string segment descriptors of the logical volumes onto physical space in one or more physical volumes. The bit string descriptors include volume serial number, starting point address, and attributes for each logical and physical volume. Examples of attributes are creation time, size, security level, and physical volume attributes.

The logical-to-physical volume manager understands the characteristics of the actual physical volumes used by the storage server. Its main functions are to allocate and deallocate space and to convert logical bit string segments to physical bit string segments for that bitfile. The logical-to-physical volume manager also maintains a flaw map to map, for example, defective tracks on a magnetic disk to spare tracks (some device controllers maintain flaw maps, making duplicate maps in the volume manager unnecessary). Similarly, it maintains a map of disk tracks or magnetic tape block numbers that are used for control and formatting and that are thus unavailable for data storage. When data is moved within the storage system because errors start to occur or new physical devices or volumes are introduced, the map must be changed.

A map of the free and used space is maintained for each logical and physical volume. Space summary information for each volume may be retrieved to aid in the volume selection process. This volume information is retained in the storage tables, which must have the same reliability and performance as the directory in the bitfile server, i.e., it must be backed up and recoverable or it must be possible to build the information from other records. All of this information is available to the site manager interface.

3.5 Physical Volume Repository

The physical volume repository (Coleman 88, Savage 85), shown in Figure 4, stores physical volumes. It may be manual or mechanical.

The physical volume repository is responsible for managing the storage of media volumes and for mounting these volumes onto drives managed by the physical device manager. Volumes may be stored in an automated library that includes a robot capable of mounting the volumes or stored in a vault and mounted manually.

The architecture of the physical volume repository is that of a server that manages abstract objects called *physical volumes*. A physical volume consists of a media volume and a volume descriptor. (A physical volume is similar to a bitfile in that both include a resource and a resource descriptor.) The volume descriptor contains at least the following fields:

- The current physical location of the media volume. The volume might be in a vault, in a storage cell of an automated device, mounted on a drive, or held by a robot.
- A human-readable label by which an operator can identify the volume.
- The media type. One physical volume repository might manage both magnetic and optical media, different varieties of magnetic tape, etc.
- Information to identify the owner of the volume.
- Access-control information to validate requests. In a capability-based system, this information might be an encryption key. In other systems, this information might be a list of clients authorized to access the volume.
- Various statistics associated with the volume, such as the number of times

the volume has been mounted and the time of the last mount.

Associated with each physical volume is a *volume identifier*. This identifier, when included in a request, allows the physical volume repository to locate the descriptor for the media volume and, in a capability-based system, proves that the client is authorized to access the volume. The format of the volume identifier is not specified by the reference model. If the medium is optical disk and only one side of a physical disk can be read at a time, there may be a unique volume identifier associated with each side of a disk.

The physical volume repository maintains the volume descriptors on a storage device to which it has access. The physical volume repository cannot maintain the volume descriptor on the volume itself because:

- The reference model does not specify the format of the data on a volume. In some implementations, the physical device manager may be able to read volume labels (using a bitfile mover), but if unlabeled volumes are allowed, only the bitfile client or the ultimate application can interpret the contents of the volume.

Table 2
Storage Server Functions

Function	Parameters	Response
SS-Allocate	logical volume IDs existing segment descriptors desired length	new segment descriptors
SS-Deallocate	existing segment descriptors	
SS-Retrieve	segment descriptors starting offset bit string length sink descriptor	number of bits transferred
SS-Store	segment descriptors starting offset bit string length source descriptor	number of bits transferred

- Most types of archival media do not support "update in place", preventing the physical volume repository from maintaining dynamic information, such as the time of last access, on the volume itself. Information on WORM optical disks, once written, cannot be modified. Some volumes, such as CD-ROMs, cannot be written at all.

- One of the important pieces of information in the volume descriptor is the physical location of the volume. One can hardly access the volume to determine where it is!

The client interface consists of the operations necessary to allow the physical device manager to mount and dismount volumes and to allow the site manager to query and change the state of the repository. The operations and parameters that are unique to the physical volume repository are listed in Table 3 and described in the following paragraphs.

PVR-Dequeue

Any queued request for the specified volume with the specific write-protect mode that includes the specified drive as an acceptable drive is cancelled.

The dequeue function is routinely used by the physical device manager to remove requests for manually mounted volumes. Even though the physical volume repository maintains the queue of requested volumes, the physical device manager may be the only module able to detect that a volume has been mounted on a drive not accessible to the physical volume repository. If an operator inserts a requested volume into an automated library, the physical volume repository will mount the volume on an available drive; if the physical volume repository can identify the volume by reading an external label, and a request for this volume is queued, the physical volume repository will choose a drive acceptable for that request. Otherwise, the physical volume repository will choose any drive capable of handling the volume. In any event, the physical volume repository will not remove the request from the queue until it receives a dequeue command from the physical device manager.

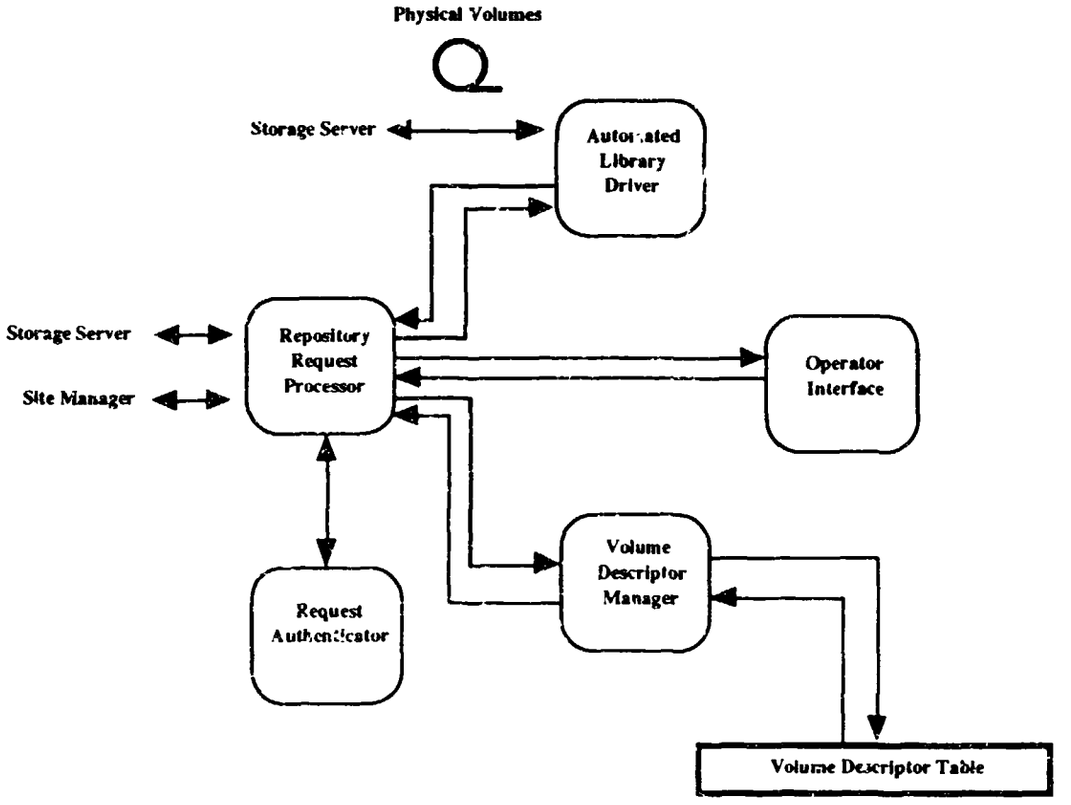


Figure 4

The Physical Volume Repository

PVR-Dismount

The media volume on the specified drive is dismounted and stored in a location selected by the physical volume repository.

The volume identifier is included in the dismount command to allow the physical volume repository to update its records; if the physical volume repository has a mechanism to identify the volume itself (by reading an external label), the volume identifier serves to confirm the physical volume repository's records and to detect anomalies.

PVR-Eject
The volume is removed from the domain of the physical volume repository. The "reason" parameter is an optional string to be sent to the operator explaining why the volume is being ejected.

In an automated system, the Eject command will probably result in the volume being moved to a port accessible by the operator.

PVR-Locate
The PVR-Locate function is used to determine volume locations when, for example, an automated library has failed and volumes are being accessed manually.

PVR-Mount
A media volume is mounted on a drive. Volumes queued for manual mounting are displayed on an operator console if the physical volume repository controls such a device (remotely controlled consoles can use the PVR-ReadQueue command described below). Some physical volume repository implementations may allow concurrent requests in which no volumes of a group are mounted until all can be mounted, or requests with a choice of volumes.

Table 3
Physical Volume Repository Functions

Function	Parameters	Response
PVR-Dequeue	volume ID write-protect mode drive ID	
PVR-Dismount	volume ID drive ID	
PVR-Eject	volume ID reason	
PVR-Locate	volume ID	current volume location
PVR-Mount	volume ID write-protect mode list of acceptable drives	drive ID or queued for manual mount
PVR-ReadQueue	queue offset maximum number of entries to send	
PVR-ReadStatus	device ID type of status desired	device status
PVR-SetStatus	device ID type of status desired value	

PVR-ReadQueue
For each request queued for a manual mount, the volume identifier, list of acceptable drives, and the write-protect mode are returned.

Providing a queue offset and a maximum number of entries to send in the PVR-ReadQueue command allows the client to receive only the number of entries that it can handle. This function also supports operator displays not under the control of the physical volume repository.

PVR-ReadStatus
The amount and type of status information is dependent upon the devices controlled by the physical volume repository and upon their configuration. Status information might include the on-line status of the device, the volume identifier of the volume mounted on the device, current or previous error information, configuration information, etc.

PVR-SetStatus
The particular status values are dependent on the devices controlled by the physical volume repository. This function is used to bring devices on-line, take them off-line, set diagnostic or manual modes, etc.

3.6 Communication Service

The communication service includes the capability to communicate request messages as well as the bitfile mover (Kitts 88) for high-speed transfer of bitfile data blocks (Allen 83).

A bitfile mover is a set of modules that move data from one source/sink to another. A storage system includes at least two bitfile movers (Figure 1), one controlled by the bitfile client and the other controlled by the storage server. Additional movers may be required for more complex routing. Figure 5 shows the control and data paths necessary to move data from source to sink.

A source or sink can be defined as:

- A memory buffer, local or remote,
- a media extent, such as on local or remote disk, or
- a channel interface connected to a device.

These definitions do not limit the methods of data transport used by the bitfile mover or the ability to transform data during the move. Because the mover's source and sink interfaces depend on the devices, network interfaces, and network protocols used by the site, the reference model does not specify them. The bitfile mover's control interface to the source or sink manager, however, can be specified.

The Move operation supported by the mover is shown in Table 4.

The source and sink descriptors may describe network interfaces, buffer addresses, or device descriptors (device addresses, block information, etc). One descriptor is sent by the bitfile client, the other is provided by the storage server.

The transformation description may specify translation, compaction, compression, encryption, and/or checksumming to be performed by the mover.

The site manager interface can through privileged commands interrogate channel status and other mover statistics.

3.7 Site Manager

Site management (Collins 88) is the collection of functions that are primarily concerned with the control, performance and utilization of the storage system. These functions are often site-dependent, involve human decision making, and span multiple servers. The functions may be implemented as stand-alone programs, may be integrated with the other storage system software, or may be policy.

Site management attempts to allocate the resources of the storage system to the best use for the overall benefit of the site Policies for the site must be set, and the manual and automatic procedures must be developed to implement those policies. The procedures must be adaptable because the requirements will change as time pro-

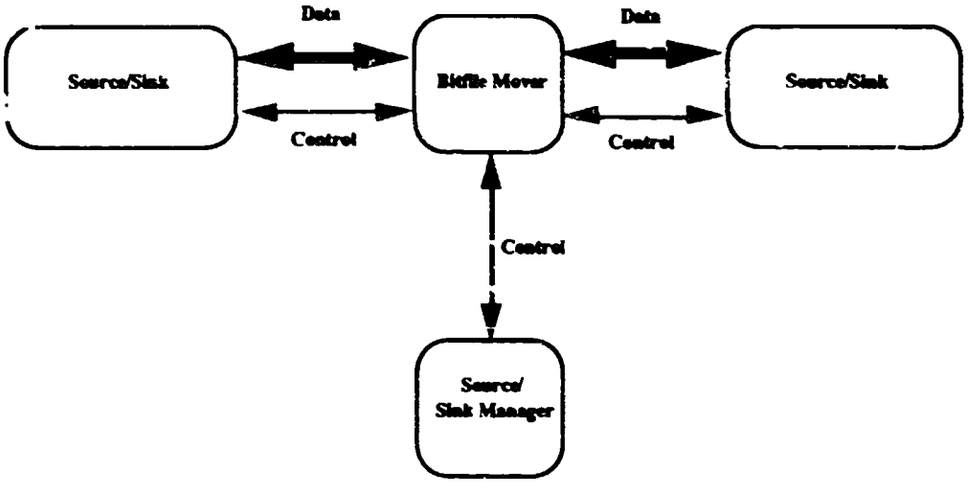


Figure 5

The Mover

Table 4
Mover Function

Function	Parameters	Response
Move	direction source descriptor sink descriptor transformation descriptor	number of source bits moved number of sink bits moved

gresses and because the same software may be run at a number of different sites.

For this discussion, the site management functions are grouped into seven areas: storage management, operations, systems maintenance, software support, hardware support, administrative control, and bitfile management. The format will be to state the requirements and then discuss the tools needed to satisfy these requirements for each area.

3.7.1 Storage Management

3.7.1.1 Requirements

The storage management function is concerned with providing good performance and reliability for user storage and access

needs, while utilizing the storage servers in an efficient and cost-effective manner. The specific goals are to optimize the overall performance of the storage servers, to control placement of bitfiles in the storage hierarchy, to maintain sufficient free space in each storage server, to control fragmentation of space on volumes, to add and delete volumes, to recover data from bad volumes, to implement data backup policies, to enforce space allocation policies, and to determine the need for equipment. Most of the activities should be automated to the extent that the task of the human system administrator is primarily one of monitoring summary reports and using reports for planning purposes.

3.7.1.2 Tools Needed

The key to storage management for any storage system is the ability to gather and utilize information about the current state of the storage servers and statistics on their transaction histories. The total space, total free space, and distribution of free space on individual volumes define the state of a storage server. Information should be extracted from the transaction log of each storage server to give the number of bitfile accesses, amount of data transmitted, average and mean response times, average and mean data transfer rates, and patterns of access by bitfile age, activity and size. Performance monitor programs are needed to provide information such as the average wait and response times, resource utilization, demand and contention, and queue lengths for storage system components.

The migration manager component of the bitfile server is the primary tool for implementing storage management policies. The migration manager uses guidelines set by the system administrator as well as historical data and current state data to determine the amount of free space to keep available for each storage server, to decide what bitfiles (by activity, size, etc.) should be stored on storage device, and to determine w files to move within the hierarchy p active bitfiles readily accessible.

The storage system should have automatic fragmentation control. Information about the amount of free space and allocated space can be used to determine when to repack volumes. This function may be performed by the migration manager or by some other storage server module.

Programs to analyze, initialize and label volumes should be provided, along with storage server commands to add and delete volumes. Two distinct types of volumes exist. Volumes like fixed magnetic disks are not demountable, and are usually defined to the operating system at system generation time. Demountable volumes such as tapes or optical disks are not subject to these constraints.

Storage servers and physical volume repositories should manage their demountable volumes largely without human intervention. The only human activities involved are occasional monitoring and revision of control parameters and supplying empty volumes to the storage server or physical volume repository when needed.

Two areas of storage management require the direct involvement of knowledgeable persons. The first is the recovery of data from bad volumes. Programs are needed to analyze, display and modify information on a volume, and to copy the entire contents of a volume to another volume without changing bitfile locations, skipping bad data which cannot be read after a number of retries. Data recovery may use the migration manager to evacuate a bad volume by moving individual bitfiles. The decision as to which type of recovery should be used in a particular case must be left to an experienced person.

System planning also requires human involvement. As new products are developed and old ones discontinued, changes in the storage servers are needed. In addition, changes in the network or user environment may require changes in the storage management policy or implementation. Statistical information may be used to decide when storage servers/devices should be enhanced, acquired and phased out. Changing patterns seen in usage information may be the best indicator that changes are needed in the policies of storage management.

3.7.2 Operations

3.7.2.1 Requirements

The operations functions are concerned with making sure that the storage system operates continuously and insuring that user requests are being satisfied in a timely and reliable manner. The system must be monitored and controlled to identify and resolve problems, to load/unload off-line volumes, and to verify that site management jobs have run correctly.

3.7.2.2 Tools Needed

An intelligent operations control center spanning the complete storage system is needed. Console displays need to show active requests for each server, requests queued for each server, volumes mounted for each storage server, space summary information (total number of volumes, number of empty volumes, free space, etc.) for each storage system in the hierarchy, resource status (processors, storage controllers, storage devices, communication links, volumes, etc.), and a special display for resources that are suspect or unavailable. Operator commands should be available for each server to restart or abort requests, and to set resources available or unavailable.

Storage system log information is needed. Messages that require action such as volume mounts and error messages should go to a display and/or hard copy console. All messages should be kept in a data base where they can be easily retrieved and displayed.

Job summary information is needed. Successful completion messages and error messages for system jobs should be written to a data base where they can be reviewed. When an important system job fails, such as backup of the bitfile descriptor tables, an operator action message should be issued.

The operational means to recover from temporary and permanent failures is needed. This includes the ability to isolate equipment which is failing or needs preventive maintenance (e.g. tape drives needing cleaning) and the ability to switch to backup equipment.

Automation of operations is needed to maximize the performance and reliability, and to minimize the manual effort. This includes automation of volume loading using a physical volume repository and automation of the decision-making process to minimize human errors and human delays.

3.7.3 Systems Maintenance

3.7.3.1 Requirements

The systems maintenance functions strive to maintain the performance, reliability, and availability of the storage system, and the integrity of the stored data. Performance is supported by monitoring the individual components and devices as well as the overall storage for failing components or out-of-balance conditions. The key to reliability and availability is the preservation of critical system information in an environment of possible hardware errors, software errors and system crashes. This information includes name server directories, bitfile descriptor tables, space limit tables, physical volume tables, physical device descriptor tables, logical-to-physical maps, logical volume tables, network configuration tables and transaction logs.

3.7.3.2 Tools Needed

System programmers must have the ability to quickly make changes in operating system or storage system parameters that affect the performance of the system. These tuning parameters may be available at execution and/or compile time. A "super-user" capability is needed so that a system programmer can execute all commands and have access to all system data.

Tools to maintain the integrity of information are needed. Programs must be available to back-up bitfiles and volumes, and to restore information from the backups. Additional tools must be available for important, dynamic tables and data bases where a backup quickly becomes out-of-date. One technique is to keep a secondary copy of dynamic information in addition to the primary copy. If either the primary or secondary fails, a new copy is immediately made of the good copy. Another technique is to keep

a journal of the important transactions. If a failure occurs, the journal is applied to a backup to restore the information to the current level. The recovery programs needed to restore a backup to the current level following a crash must be available and well tested. Several persons should be familiar with the procedures required.

A checkpoint capability is needed to restore critical storage system tables and data bases to a consistent state if a crash occurs during a transaction that makes multiple changes (such as saving a bitfile which makes a new bitfile descriptor, updates the directory that points to the descriptor, and may update the accounting data base as well). During restart following an abnormal termination, the checkpoints are used to either complete or back-off requests so that the tables and data bases will be consistent.

Verification programs are needed to check the consistency of storage system information. These programs should be designed to run in parallel with the system so that operation may continue while verification is done.

Tools to help with problem determination are needed. These include trace capability, breakpoint capabilities, selected printing of formatted and unformatted dumps of data and programs, and dump analysis programs. Tools are needed to modify and repair storage system information.

3.7.4 Software Support

3.7.4.1 Requirements

For sites that develop new storage system software, facilities must be available to develop, maintain and test that software.

For customer sites, a test facility is required to verify that a new version satisfies local security and other requirements before production use.

3.7.4.2 Tools Needed

An environment must be provided to test software changes and enhancements without disrupting the production use of the storage system. The ability to run a test version and the production version of the software simultaneously is necessary. The test software may run on the same processors as the production software or run on other processors. It may share devices such as the communication systems and the physical storage systems, but it should have its own tables, data bases, volumes, etc. Instead of running a complete test version of the storage system software, a test version of a particular component (e.g., a bitfile server) could be run using components of the production system for the rest of the system.

A regression testing capability should be available so that a comprehensive set of tests can be run at any time against the production or test system to verify security, integrity, and performance. Both the running and checking of the regression tests should be automated.

3.7.5 Hardware Support

3.7.5.1 Requirements

The hardware support functions are concerned with the display, diagnosis and correction of hardware problems, and the configuration and installation of the hardware. Hardware failures and the time needed to repair failures must be minimized especially for those failures that bring down the storage system. It must be easy to reconfigure the system hardware, and install and remove equipment.

3.7.5.2 Tools Needed

Programs to report hardware errors are needed. These programs should be able to give a detailed time history of hardware errors, and show correlated summaries of both temporary and permanent errors by error-type, device-type, specific device and volume, over specified time intervals. The ability to recognize the beginning of a problem before it becomes permanent is especially important when dealing with storage devices/volumes where permanent errors generally mean the loss of data.

Programs to exercise and diagnose all hardware components of the storage system are needed. These programs should be able to analyze the errors and recommend the corrective action. Storage devices with

mechanical parts, such as magnetic disk, optical disk, magnetic tape, and especially physical volume repositories, have a much higher error rate than strictly electronic hardware so diagnostic and exercise programs play an important role in storage systems.

The system should be redundantly configured so that components and paths can be isolated, removed for repair and upgraded with a minimal impact upon operation and performance. Dynamic reconfiguration capabilities, including the switching of the production software to a backup processor, should be available.

3.7.6 Administrative Control

3.7.6.1 Requirements

Administrative control covers the security, accounting, and management policies of the storage system. The security requirements are to implement the security policies of the site and to recognize if policy violations are being attempted. The accounting requirements are to gather usage information, to charge for use of resources, and to control the resources. The management requirements are to present summary information concerning the operation and performance of the system that can be used to justify operational and equipment expenditures and to set high-level policy.

3.7.6.2 Tools Needed

The storage system must implement the particular security policies of each site by building the policies into the programs or through the use of replaceable modules. In general, the policies involve verification that a user has access to the requested resources of a server. Access or capability information is stored with the resource and checked against similar information in the request. For some sites it is required that classification and partition levels be associated with bitfiles and requests, and that access be controlled based on certain classification and partition rules. A security log must be available that contains all security violations (as determined by a site) and all transactions above a specified security classification level. A log of all transactions should be kept to help diagnose anomalous situations.

The storage system needs a resource-charging mechanism. Charges may be incurred for the following resources and services: amount of data stored, number of bitfiles stored, data transferred, bitfiles accessed, and any of the requests made to the bitfile servers. These charges may vary for the different bitfile servers, depending on the level of performance and the class of storage used. Requests made to the storage system should contain an account code to which the charge is to be made. An account code can be stored in each bitfile descriptor along with the storage space used, the length of time stored, and a reference time for accounting purposes. An accounting program obtains the storage and bitfile charge information from the bitfile descriptors; obtain the access, data transfer and request charge information from the transaction logs; accumulate and sort the charge information; and write the charge information to an accounting file. Another accounting program has the resource charging rates and calculates the bills. Since the account codes often change, an automatic means of updating the bitfile descriptors is needed. One approach is to have a central data base of accounts from which an accounting program can update the bitfile descriptors. This data base can also be used to validate users and to show what resources they can use.

The summary information used by management to set high-level policy needs to be extracted from all the other site management reports and data bases, and presented in a useful (usually graphic) manner. A number of vendor products are available that can be used to extract, process and display information.

4. References

Allen, I. D. (1983). The role of intelligent peripheral interfaces in systems architecture. *Proc. Nat. Computer Conf.* pp. 623-630.

Almes, G. T., Black, P., Lazowska, D., and Noe, D. (1985). The Eden system: a technical review. *IEEE Trans. on Software Engineering SE-11*. (1), 43-59.

Birrell, A. D., Levin, R., Neddham, M., and Schoeder D. (1982). Grapevine: an exercise in distributed computing. *Comm. ACM*, Vol 25, No. 4, 260-274.

Booch, G. (1986). Object-oriented development, *IEEE Trans. on Software Engineering, SE-12*. (2), 211-221.

Brownbridge, D. R., Marshall, L. F., and Randell, B. (1982). The newcastle connection. *Software Practice and Experience* 12, 1147-1162.

Coleman, S. and Watson, R. (1984). Storage in the LLNL Octopus network: an overview and reflections. *Digest of Papers*, Sixth IEEE Symposium on Mass Storage Systems, Vail, Colorado, June 1984.

Coleman, S. (1988). Physical volume repository. *Digest of Papers*, Ninth IEEE Symposium on Mass Storage Systems, Monterey, California, November 1988.

Collins, B., Devaney, M., and Wilbanks, E. (1982). A network file storage system. *Digest of Papers*, Fifth IEEE Symposium on Mass Storage Systems, Boulder, Colorado, October 1982.

Collins, B. (1988). Mass storage system reference model system management. *Digest of Papers*, Ninth IEEE Symposium on Mass Storage Systems, Monterey, California, November 1988.

Davis, J. D. (1982). Mass storage systems: a current analysis. *Digest of Papers*, Fifth IEEE Symposium on Mass Storage Systems, Boulder, Colorado, October 1982.

DIS8571. *Information processing systems open systems interconnection, file transfer, access, and management* (in four parts, draft international standard ISO/DIS8571), distributed by Omicon Information Services.

Donnelley, J. E., and Fletcher, J. G. (1980). Resource access control in a network operating system. *Proc. ACM Pacific 80 Conf.*

Enslow, P. H., Jr. (1978). What is a "distributed" data processing system? *Computer*, Vol 11, No. 1, Jan, 13-21.

Falcone, Joseph R. (1988). The bitfile server in the IEEE reference model for mass storage systems. *Digest of Papers*, Ninth IEEE Symposium on Mass Storage Systems, Monterey, California, November 1988.

Fletcher, J., G. (1975). Computer storage structure and utilization at a large scientific laboratory. *Proc. IEEE* 63 (8), 1104-1113.

Foglesong, Joy, et. al. (1990). The Livermore distributed storage system: implementation and experiences. *Digest of Papers*, Tenth IEEE Symposium on Mass Storage Systems, Monterey, California, May 1990.

Gary, Mark (1990). Overcoming Unix kernel deficiencies in a portable, distributed storage system. *Digest of Papers*, Tenth IEEE Symposium on Mass Storage Systems, Monterey, California, May 1990.

Gentile, R. B., and Lucas, J. R. (1971). The TABLON mass storage network. *Proc. Spring Joint Computer Conf.*, pp. 345-356.

Grossman, C.P.(1989). Evolution of the DASD storage control, *IBM Systems Journal*, Vol.28, No.2, 1989.

Harris, J. P., Rhode, R. S., and Arter, N. K. (1975). The IBM 3850 mass storage system: design aspects. *Proc. IEEE* 63 (8), 1171-1179.

Hogan, Carole, et. al. (1990). The Livermore distributed storage system: requirements and overview. *Digest of Papers*, Tenth IEEE Symposium on Mass Storage Systems, Monterey, California, May 1990.

Howie, H. R., Jr. and Salbu, E. (1975). Mass storage implementation approaches: a spectrum. AFIPS The Information Technology Series, *Memory and Storage Technology*.

Johnson, C. (1975). IBM 3850 mass storage system. *AFIPS Conf. Proc.* 44.

Jones, A. K. (1979). The object model: a conceptual tool for structuring software. "Operating Systems". Springer-Verlag, Berlin.

Kitts, D. (1988). Bitfile mover. *Digest of Papers*, Ninth IEEE Symposium on Mass Storage Systems, Monterey, California, November 1988.

Kuehler, J. D. and Kerby, H. R. (1966). A photographic mass storage system. *AFIPS FJCC Proc.* 29, 735-742.

Lantz, K. A., Edighoffer, J. L., and Hitson, B. L. (1985). *Toward a Universal Directory Service*. Report No. STAN-C5-85-1086, Stanford University.

Leach, P. J., et al (1982). UIDs as internal names in a distributed file system. *Proc. Symposium on Principles of Dist. Computing*, Ottawa, 34-41.

LeLann, G. (1981). Motivation, objectives, and characteristics of distributed systems. "Distributed system-architecture and implementation". Springer-Verlag, Berlin, 1-9.

Mclarty, T., Collins, B. and Devaney, M. (1984). A functional view of the Los Alamos central file system. *Digest of Papers*, Sixth IEEE Symposium on Mass Storage Systems, Vail, Colorado, June 1984.

Miller, S. W. and Collins, B. (1985). Toward a reference model for mass storage systems. *Computer*, Vol. 18, No. 7, July, 9-22.

Miller, S. W. (1988a). "A Reference Model for Mass Storage Systems", *Advances in Computers*, Volume 27, Academic Press.

Miller, S. W. (1988b). Mass storage reference model, special topics. *Digest of Papers*, Ninth IEEE Symposium on Mass Storage Systems, Monterey, California, November 1988.

Mullender, J., and Tannenbaum, A. S. (1984). Protection and resource control in distributed operating systems. *Computer Networks* 8, 421-432.

Nelson, M., Kitts, D. L., Merrill, J. H., and Harano, G. (1987). The NCAR mass storage system. *Digest of Papers*, Eighth IEEE Symposium on Mass Storage Systems, Tucson, Arizona, May 1987, pp. 12-20.

O'Lear, B. T. and Choy, J. H. (1982). Software considerations in mass storage systems. *Computer* 15 (7), 36-44.

Penny, S. J. and Alston-Garnjost, M. (1970). Design of a very large storage system. *Proc. Fall Joint Computer Conf.* pp. 45-51.

Sandberg, R. (1985). Design and implementation of the SUN network file system. *Proc. Tenth Usenix Conference*, 119-130.

Savage, P. (1985). Proposed guidelines for an automated cartridge repository. *Computer*, Vol 18, No. 7, July, 49-58.

Savage, P. (1988). Storage server as physical box. *Digest of Papers*, Ninth IEEE Symposium on Mass Storage Systems, Monterey, California, November 1988.

Svobodova, L. (1984). File servers for network-based distributed systems. *Computing Surveys, 16*, (4), 354-398.

Watson, R. W. (1980). Network architecture design for a back-end storage network. *Computer*, Vol 13, No. 2, Feb, 32-48.

Watson, R. W. (1981a). Distributed system architecture model. *Distributed Systems—Architecture and Implementation*, Springer-Verlag, Berlin, 10-43.

Watson, R. W. (1981b). Identifiers (naming) in distributed systems. *Distributed Systems—Architecture and Implementation*, Springer-Verlag, Berlin, 191-210.

Watson, R. W. (1984). Requirements and overview of the LINCS distributed operating system architecture. Lawrence Livermore National Laboratory, Preprint UCRL-90906.

Watson, R. W. (1987). Tutorial notes, Eighth IEEE Symposium on Mass Storage Systems, Tucson, Arizona, May 1987.

Watson, R. W. (1988). The Architecture of Future Operating Systems, UCRL Preprint 99896, presented at the Cray Users Group Meeting, Tokyo, Japan - September 1988.

Wildman, M. (1975). Terabit memory system: design history. *Proc. IEEE* 63 (8), 1160-1165.

5. Glossary

Authentication Request/Reply
The command to test the access rights of the requestor to a particular service.

Bitfile
A collection of data that can be created on, read from, written into, and deleted from a storage system. These data are treated as a string of bits without any particular structure.

Bitfile Authenticator
The process that checks the access rights of a requestor for service.

Bitfile Descriptor
The bitfile attribute information that is stored as an entry in the bitfile descriptor table.

Bitfile Descriptor Manager
The process that manages the bitfile descriptor table.

Bitfile Descriptor Table
The data store where the bitfile descriptors are stored.

Bitfile ID
A machine-oriented, globally unique identifier of a bitfile.

Bitfile Mover
The processes, including the high-level protocols, that control the movement of bitfiles.

Bitfile Server
The set of processes that control the creation, destruction, and access to the many bitfiles under its control.

Bitfile Server Request Processor
The portion of the bitfile server that acts upon requests and controls the request/reply communications with internal modules as well as other processes and servers.

Bitfile Transfer
The high-speed movement of bitfile data blocks.

Client Request/Reply
The list of permitted commands from a client to a server and the resulting responses.

Create
The bitfile client request to form a bitfile descriptor record in the bitfile descriptor table. The bitfile attributes to be contained in the bitfile descriptor are specified in the request.

Destroy
The bitfile client request to remove a bitfile descriptor from the bitfile descriptor table. The space allocated to the bitfile is deallocated and, if the media can be rewritten, returned to the free space list.

Lock
The bitfile client request to establish a lock for a bitfile in preparation for one or more stores or retrieves of the bitfile.

Modify
The bitfile client request to change one or more attributes of a bitfile as contained in the bitfile descriptor table.

Monitor Information
Status information from storage system modules used by the site manager to assist in the management and control of the storage system.

Move Command
The request to move a bitfile between specified devices.

Name Server
The server that converts between the human-oriented name for a bitfile and the machine-oriented name for the same bitfile.

Physical Volume
A bounded unit of storage media that is used to store bitfiles.

Physical Volume Move
The physical movement of a volume between the volume repository and a storage server or its return.

Physical Volume Repository
The place where physical volumes are stored when they are not at a read/write station.

Principle ID
Identification of the agent requesting service from the bitfile server.

PVR-Dismount
A request sent to the physical volume repository to remove a physical volume from a drive.

PVR-Mount
A physical volume ID sent to the physical volume repository with the request to mount it on a particular storage device in the storage server.

Query
The bitfile client request to obtain information about a bitfile or its attributes from the bitfile descriptor table.

Retrieve
The bitfile client request to move a bitfile or a segment of a bitfile from a storage server to the bitfile client. If only a segment is to be moved, then the internal starting bit address and the bit string length must be specified.

Site Control
Commands from the site manager for initial set up, operations, and management of the storage system.

SS-Allocate
The request to a storage server to make logical space available for bitfile storage.

SS-Deallocate
The request to a storage server to remove a bitfile from physical storage and return the space to the free space list.

SS-Retrieve
The request from a bitfile server to move a bitfile from a storage server to a bitfile client.

SS-Store
The request from a bitfile server to move a bitfile from bitfile client buffer to storage server media.

Status
The bitfile client request for the status of the bitfile server or of a previous request made by the bitfile client.

Store
The client request to move a bitfile data block from the bitfile client to a bitfile server medium. This request may include the ability to append a segment at the end of an existing bitfile or to update a physically specified segment of an existing bitfile.

Unlock
The client request to release the lock held on a bitfile.

The IEEE Mass Storage System Reference Model

1992 Conference on Mass Storage Systems and Technologies

September 22, 1992

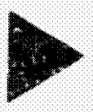
Sam Coleman
LLNL, L-60 (510) 422-4323
P. O. Box 808 (510) 423-8715 (fax)
Livermore, CA. 94550 scoleman@lini.gov

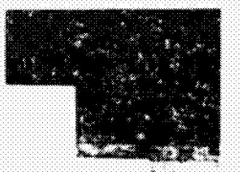
University of California
Lawrence Livermore
National Laboratory

Agenda

- Overview of the Reference Model
- Abstract Objects
- Description of the Model
  - Bitfile Client
  - Name Server
  - Bitfile Server
  - Storage Server
  - Physical Volume Repository
  - Communication Service
  - Mover
  - Site Management

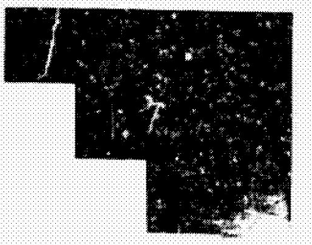
University of California
Lawrence Livermore
National Laboratory





Overview of the Reference Model

University of California
Lawrence Livermore
National Laboratory



Original Motivation for the Model

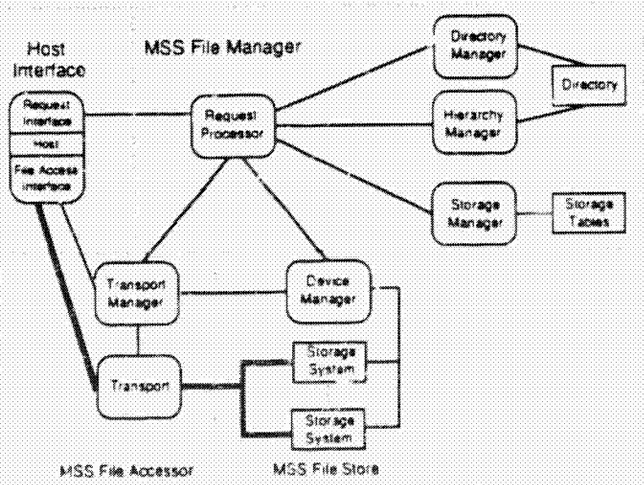
- Communications tool for discussing storage systems
  - Proper modularity
  - Transparency
  - Client interfaces
- Take advantage of existing expertise
- Reduce duplication of effort among system developers
- Encourage mutually-compatible commercial products
- Focus on distributed, high-performance storage systems

University of California
Lawrence Livermore
National Laboratory

History of the Model

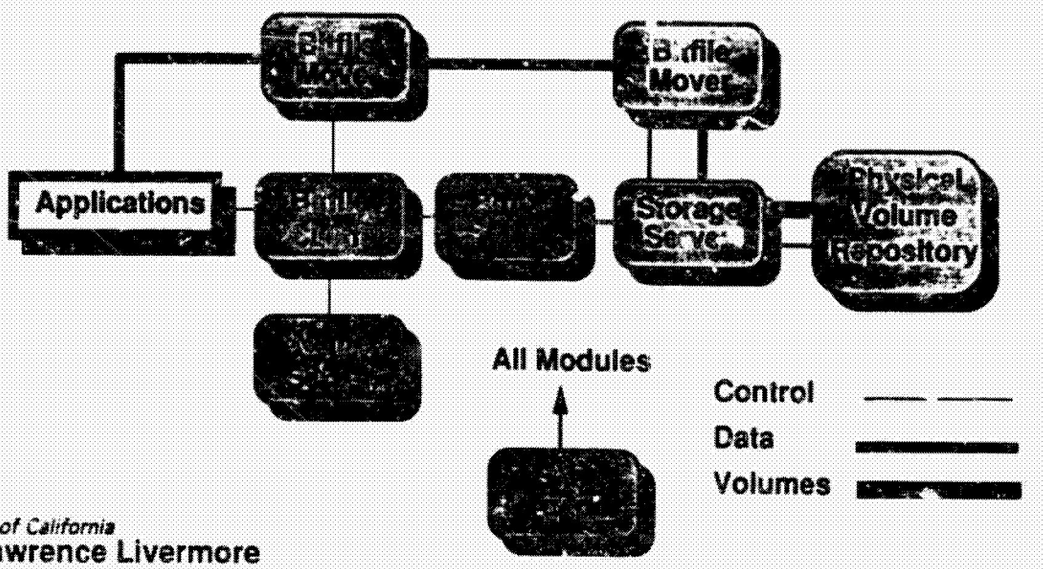
- First specialists workshop - September, 1983
- A proposed model presented at the 6th IEEE Symposium - June, 1984
- A refined model presented at the 7th Symposium - November, 1985
- Session devoted to the model at the 9th Symposium - October, 1988
- Working Group organized September, 1989
- Project 1244 kicked off at the 10th Symposium - May, 1990

The First Reference Model (June, '84)



ORIGINAL PAGE IS
OF POOR QUALITY

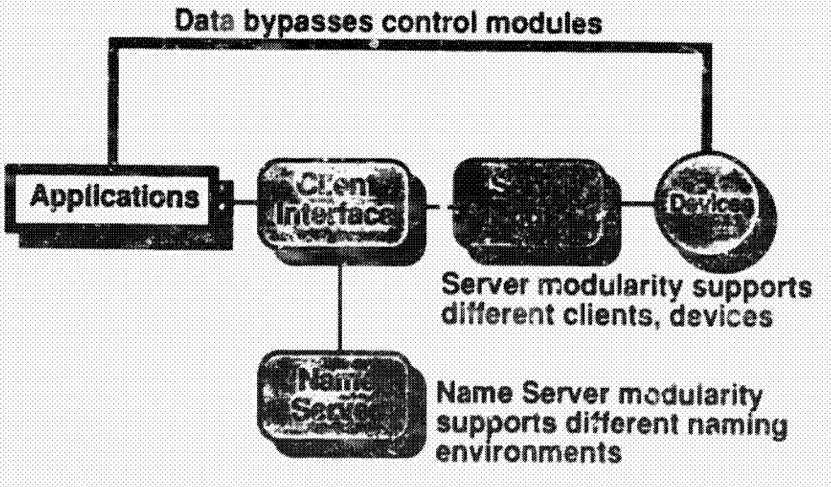
The Current Model



University of California
Lawrence Livermore
National Laboratory

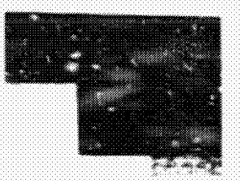
Goddard SC 9/22/92 7

The Significant Modularity



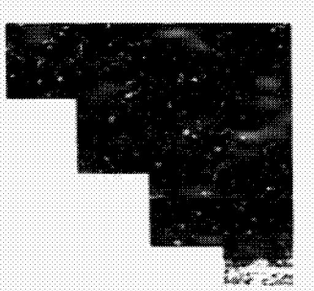
University of California
Lawrence Livermore
National Laboratory

Goddard SC 9 22 92 8



Abstract Objects

University of California
Lawrence Livermore
National Laboratory



Definitions

Abstraction:

A simplified description of a system that emphasizes some of the system's details while suppressing others.

M. Shaw

Object:

An entity whose behavior is characterized by the actions that it suffers and that it requires of other objects.

G. Booch

University of California
Lawrence Livermore
National Laboratory

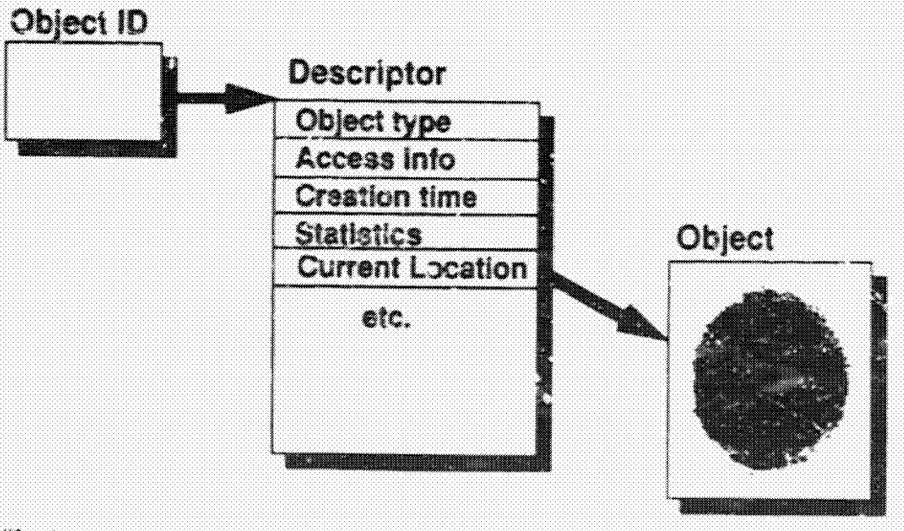
Characteristics of abstract objects

- Visible to the client
- Managed by a server that
  - Defines operations (the interface)
  - Defines legal sequences of operations
  - Provides access to objects
- May be active (e.g. processes) or inactive (e.g. magnetic tapes)
- Composed of a descriptor, an ID, and a physical object

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 1 1

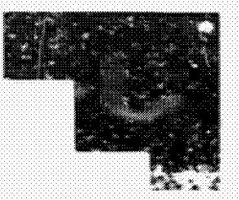
Components of an abstract object

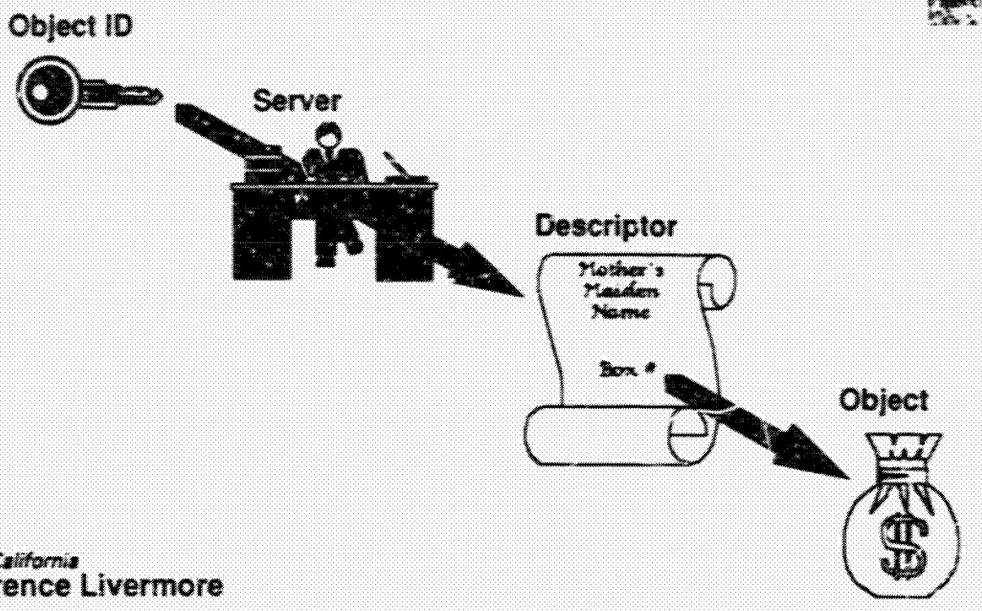


University of California
Lawrence Livermore
National Laboratory

Goddard SC 9 22 92 1 2

An Example - The Bank Safety Deposit Box Abstract Object

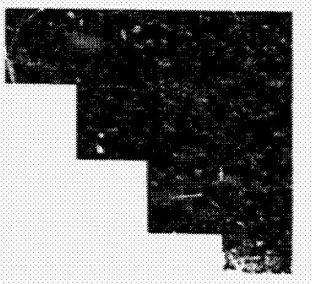


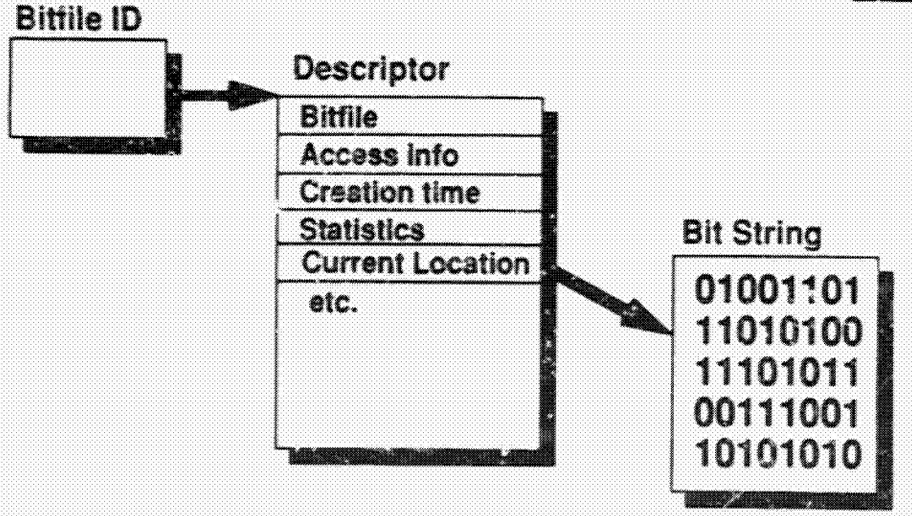


University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 13

The Fundamental Reference Model Abstract Object (the Bitfile)

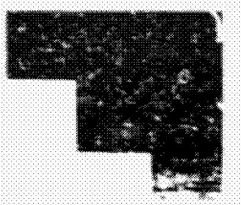




University of California
Lawrence Livermore
National Laboratory

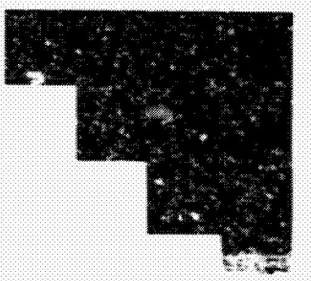
Goddard SC 9 22 92 14

Common Abstract Object Functions



- **Create**
  - Creates a descriptor (and maybe an object) and returns ID
- **Destroy**
  - Destroys the object descriptor and releases object resources
- **Read Descriptor**
  - Interrogates descriptor entries
- **Modify Descriptor**
  - Changes descriptor entries

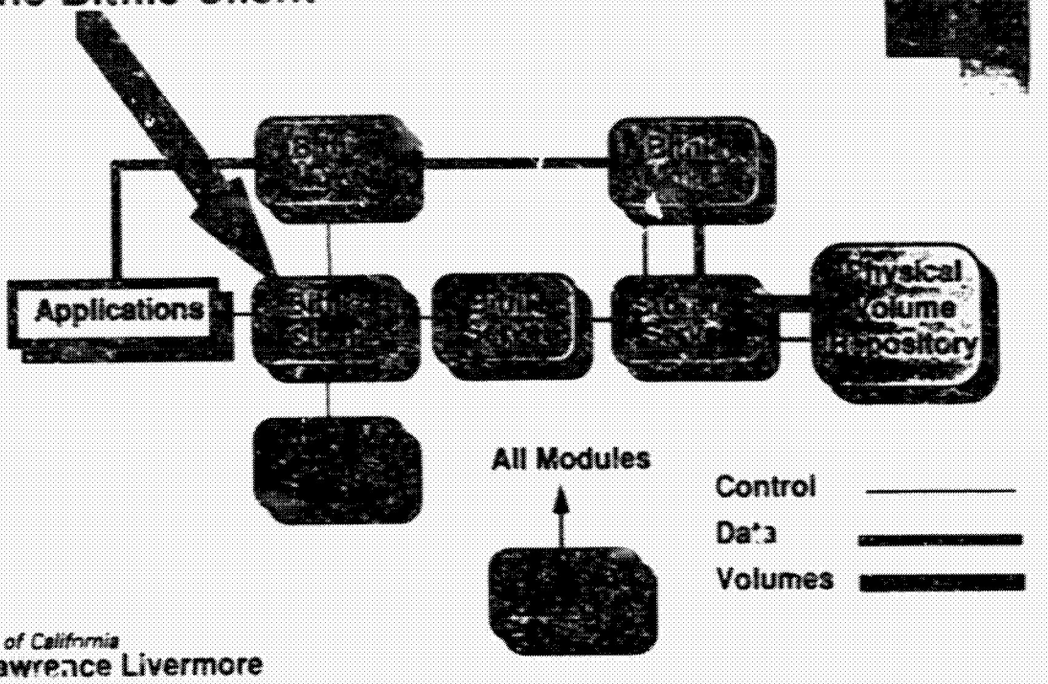
University of California
Lawrence Livermore
National Laboratory



Description of the Model

University of California
Lawrence Livermore
National Laboratory

The Bitfile Client



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9.22.92 17

The Purpose of the Bitfile Client

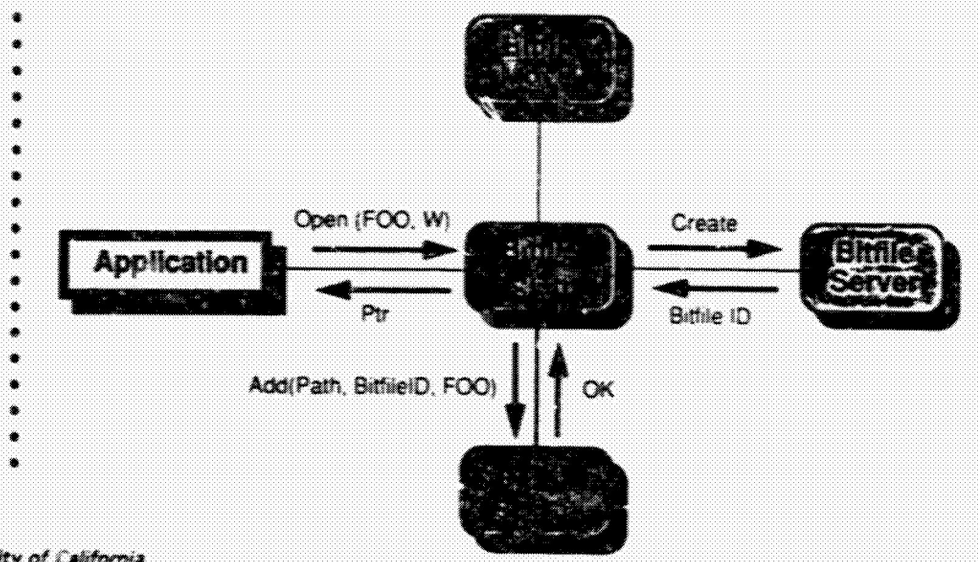
- The Bitfile Client is the programmatic agent of the user
  - Converts user desires (IPC, system, or library calls) into bitfile server, name server, and mover requests.
  - Allows the same servers to handle a variety of standard user interfaces.
  - The Bitfile Client is system-dependent!

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9.22.92 18

Creating a Bitfile

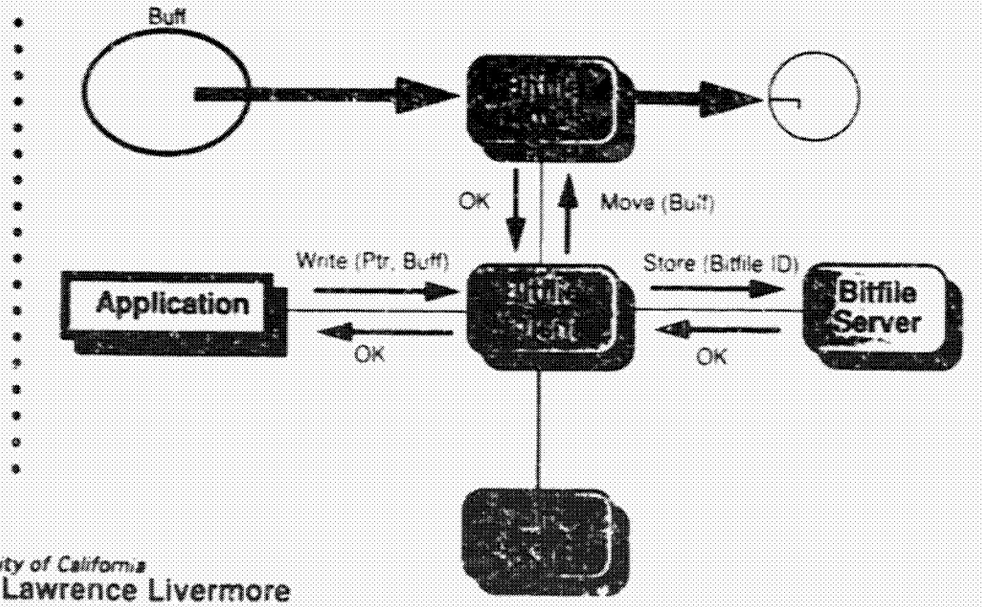
ORIGINAL PAGE IS
OF POOR QUALITY



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 19

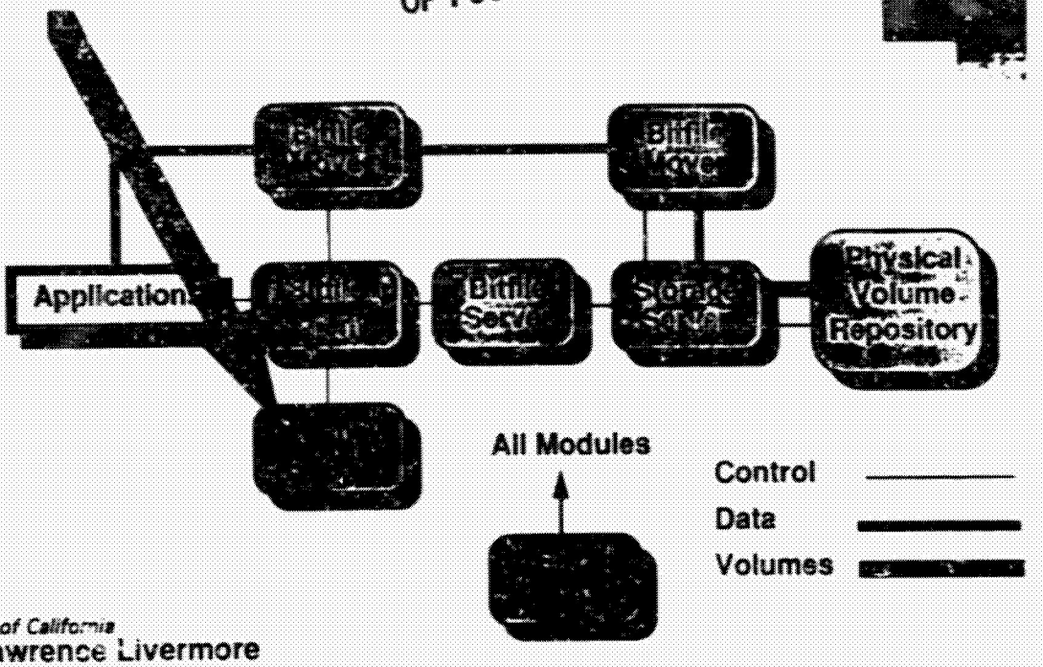
Writing a Bitfile



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 20

The Name Server



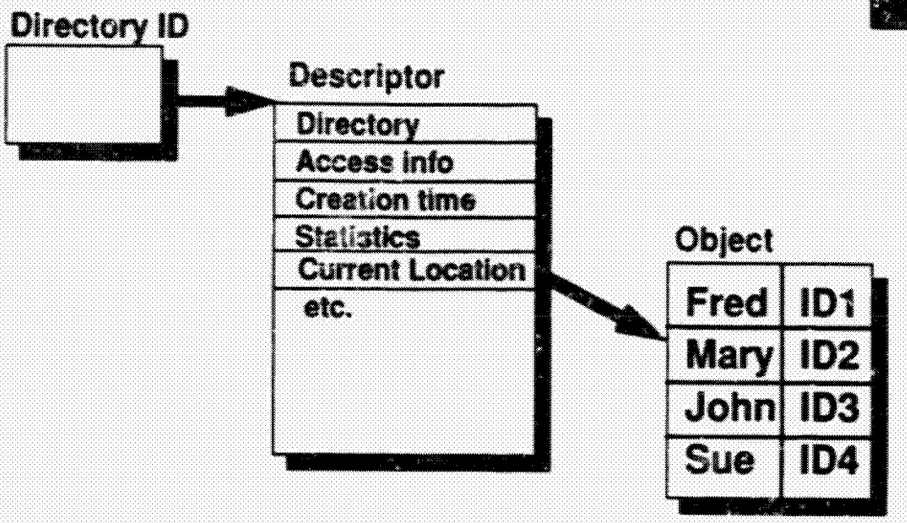
University of California
Lawrence Livermore
National Laboratory

The Purpose of the Name Server

- Provides for the retention of Bitfile IDs
- Translates human-oriented names into machine-oriented resource IDs
- Supports resource sharing
- Decouples name management from bitfile management
  - Can be used in different naming environments
  - Supports heterogeneous networks
- Manages abstract objects called "directories"

University of California
Lawrence Livermore
National Laboratory

The Directory Abstract Object



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 23

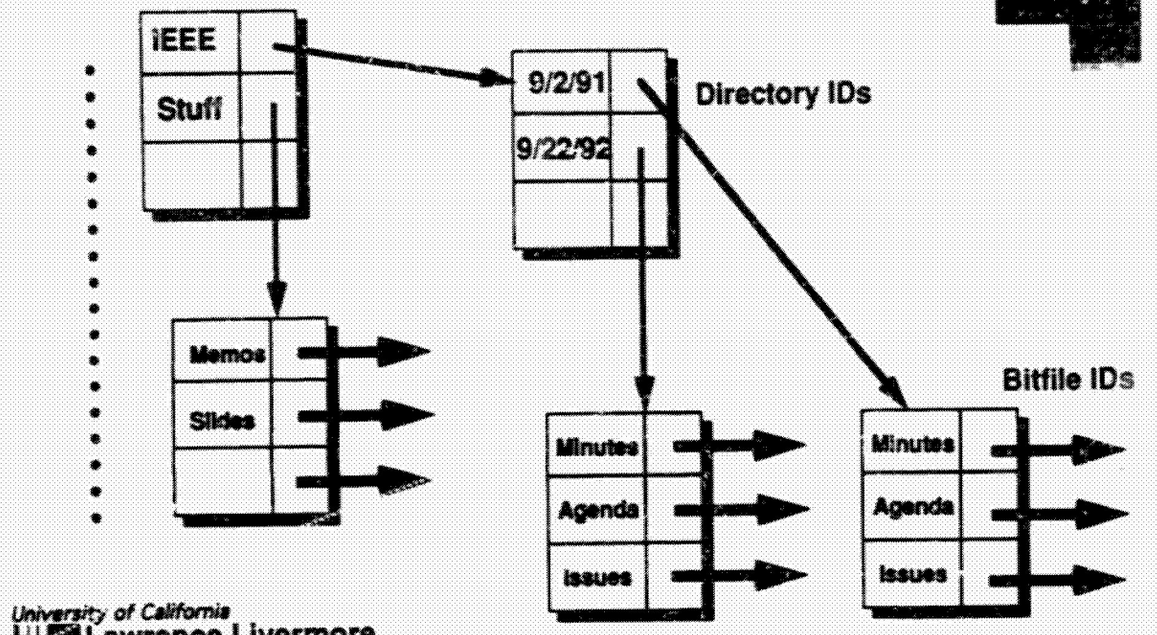
Specific Name Server Functions

- **Add Entry**
  - Inserts a name / ID pair into a directory
- **Fetch Entry**
  - Given a name, returns the ID from an entry
- **Delete Entry**
  - Removes one or more entries from a directory
- **List**
  - Returns some or all of the names in a directory

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 24

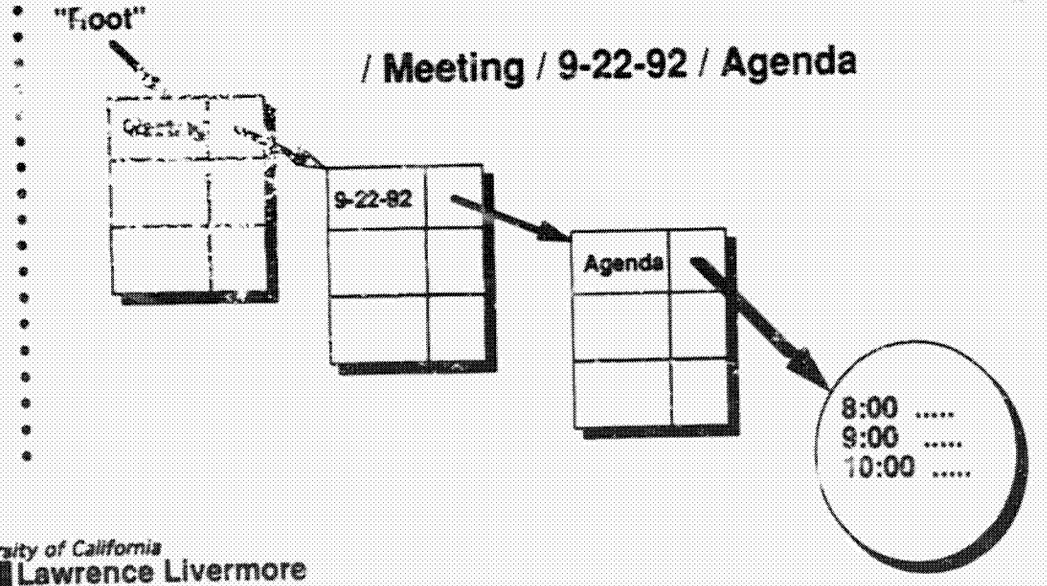
Creating a Directory "Tree"



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 25

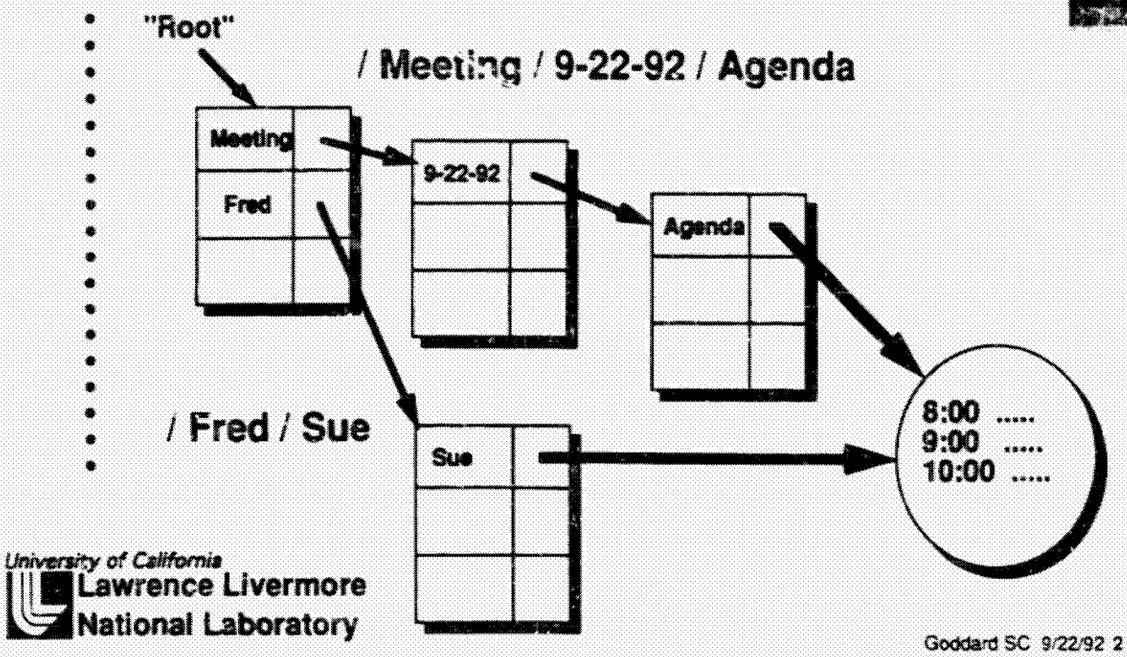
Path Names



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 26

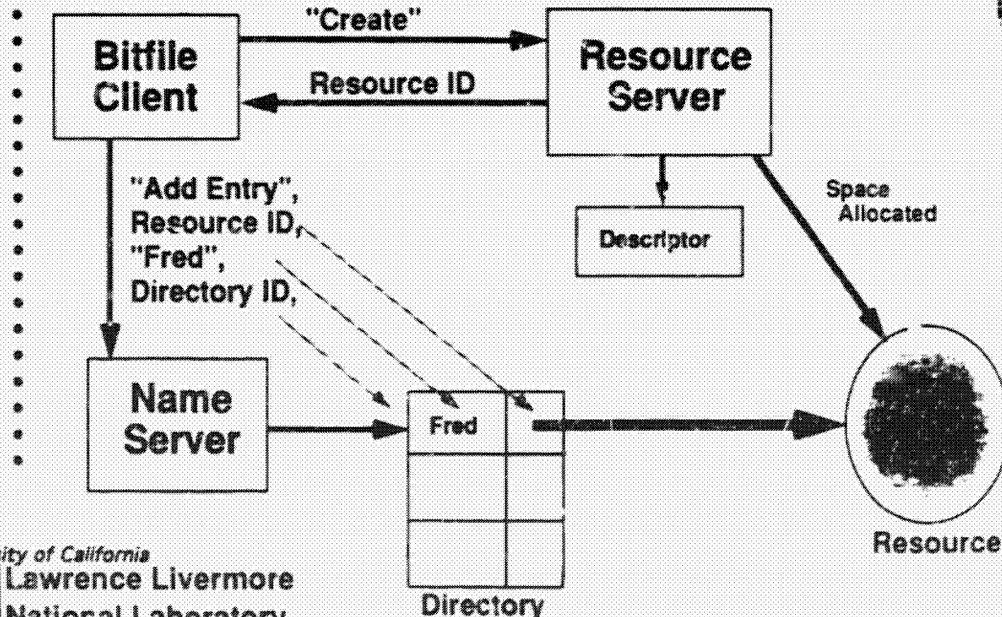
Alternate Paths



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 27

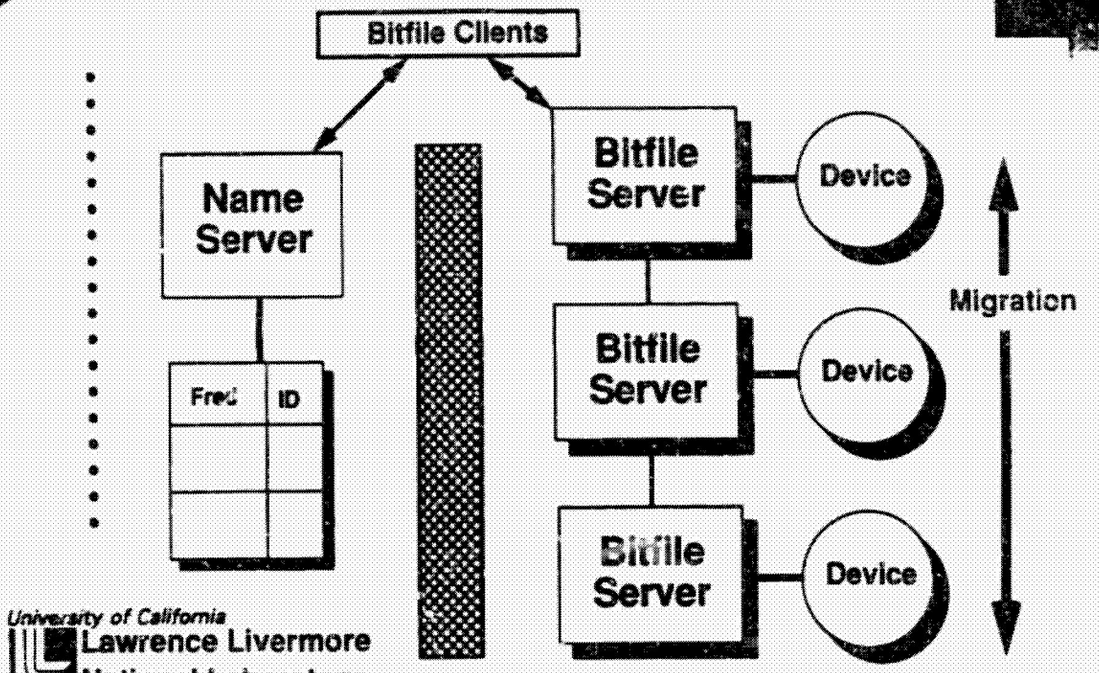
Creating and Naming a Resource



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 28

Independent Bitfile and Name Servers



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 2 9

Problems with Independent Servers

- Lost objects
- Dangling pointers
- Performance is lower when bitfile server access is necessary
- Encryption of IDs may be necessary

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9-22 92 3 0

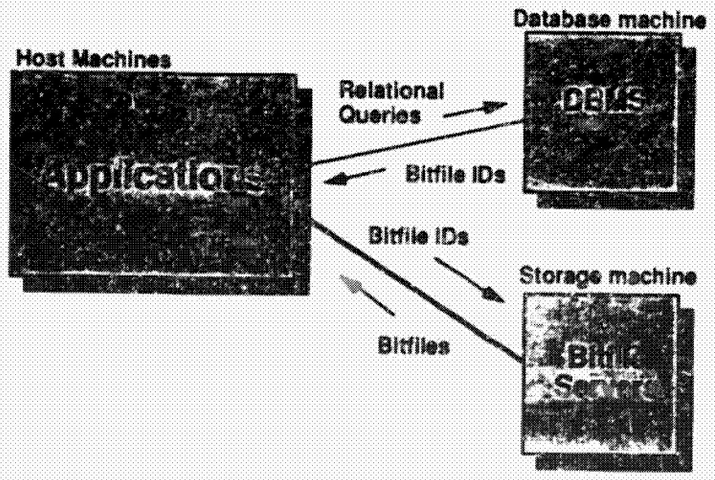
Different Naming Environments

- Unix
  - Unstructured names
- Dos
  - Name + extension
- VMS
  - Version numbers
- Etc.
- **All can use the same Bitfile Servers!**

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 31

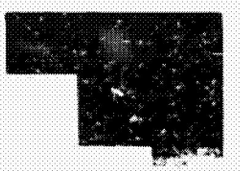
Using a Database as a Name Server

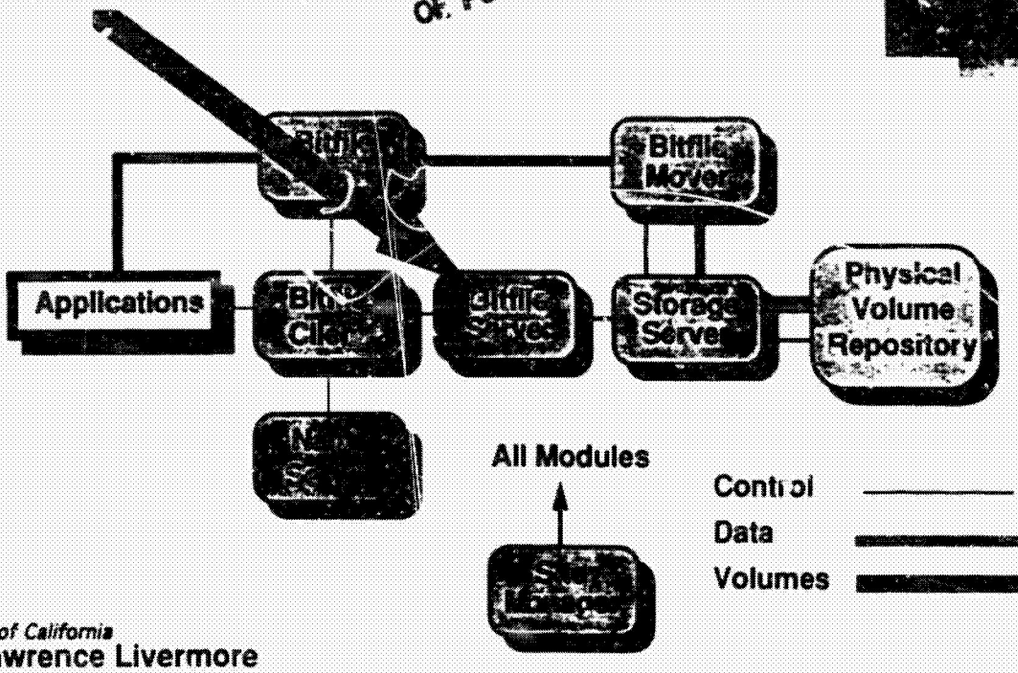


University of California
Lawrence Livermore
National Laboratory

Goddard SC 9 22 92 3 2

The Bitfile Server





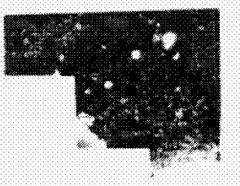
University of California
Lawrence Livermore
National Laboratory

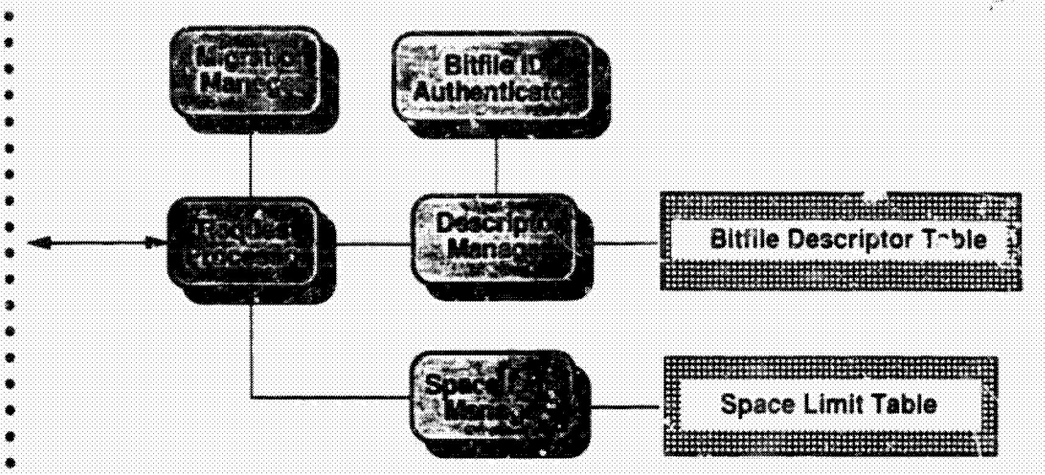
The Purpose of the Bitfile Server

- **The Bitfile Server controls the "logical" aspects of bitfile storage and retrieval**
  - **Maintains the Bitfile Descriptors**
  - **Authenticates access to bitfiles**
  - **Controls bitfile migration to/from other bitfile servers**

University of California
Lawrence Livermore
National Laboratory

Bitfile Server Internals

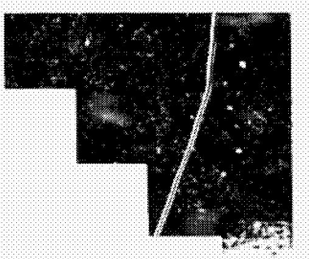




University of California
Lawrence Livermore
National Laboratory

ORIGINAL PAGE IS
OF POOR QUALITY

Goddard SC 9/22/92 3 5



Bitfile Descriptor Fields

- Created and used by Bitfile Client
  - Comment, bitfile format
- Create by Bitfile Client, used by Bitfile Server
  - Account, lifetime, security level, service desired ...
- Created by Bitfile Server, used by both
  - Statistics, bitfile length, creation time ...
- Created and used by Bitfile Server
  - Lock information, last migration time ...
- Created by Storage Server, used by Bitfile Server
  - Bitfile location, last device used

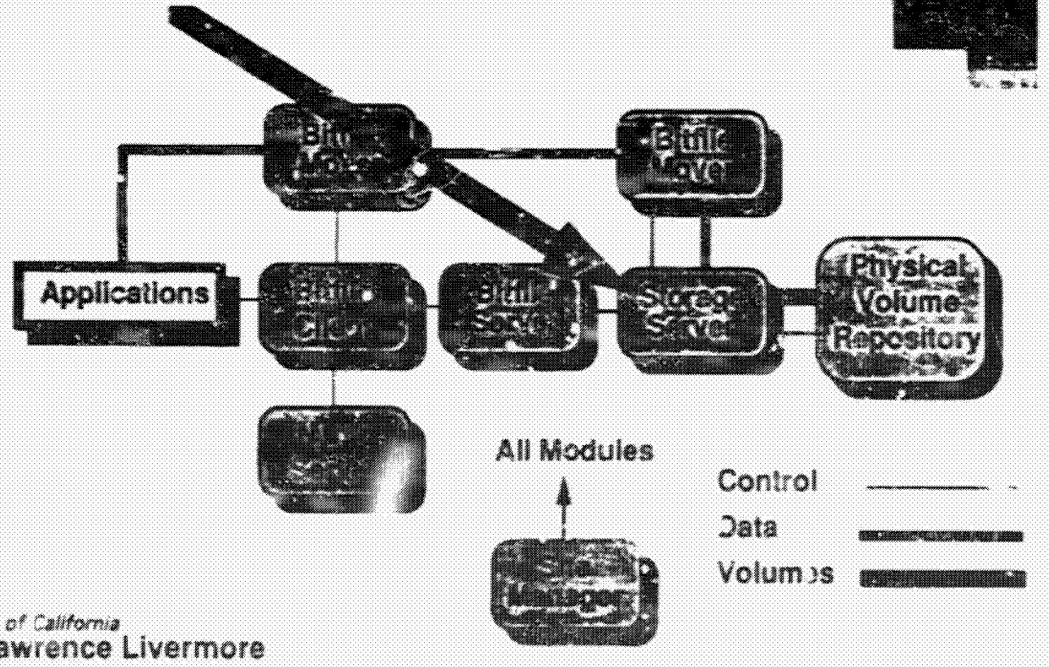
University of California
Lawrence Livermore
National Laboratory

Goddard SC 9 22 92 3 6

Bitfile Server Functions

- **Transaction Functions**
  - Abort, Status
- **Descriptor Functions**
  - Create, Destroy, Lock, Modify, Query, Unlock
- **Bitfile Access Functions**
  - Erase, Retrieve, Store

The Storage Server



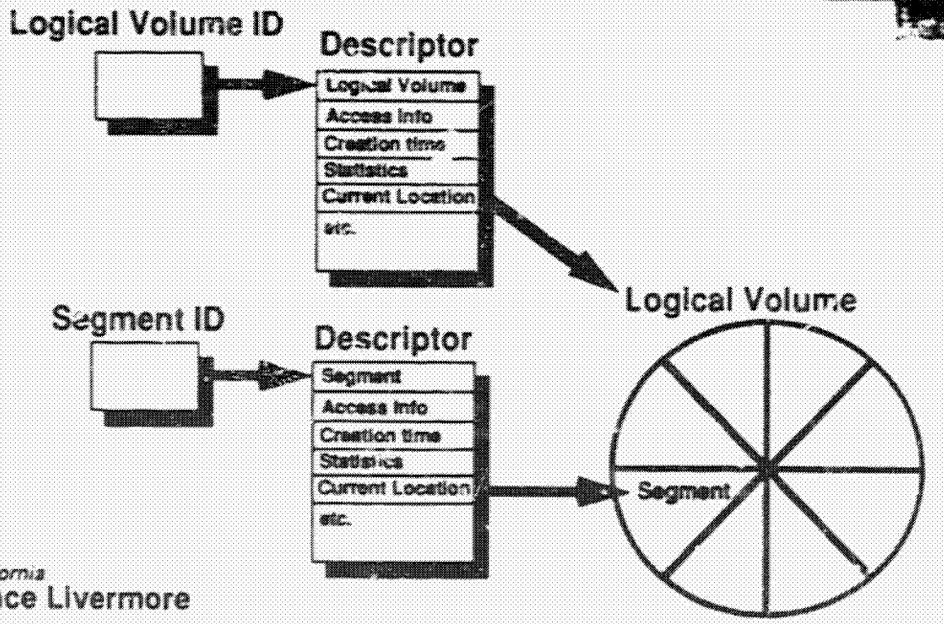
The Purpose of the Storage Server

- Controls the "physical" aspects of bitfile storage and retrieval
  - Acts like an intelligent controller
- Presents the image of perfect media to the Bitfile Server
  - (The media capacity is visible to the Bitfile Server)
- Manages abstract objects called "logical volumes" made up of sets of "bit string segments"
- "Devices" are also visible, but only to the site manager

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 39

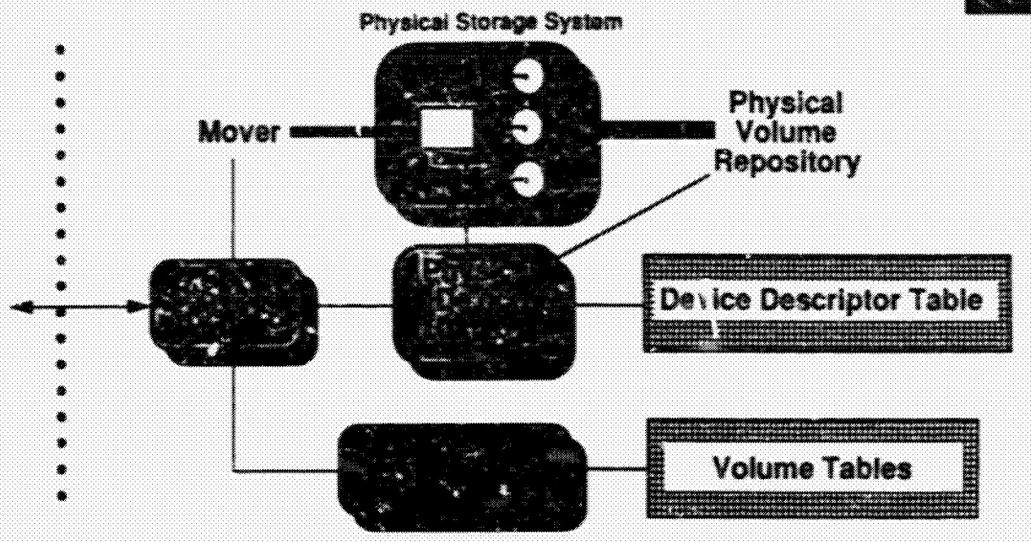
Logical Volume and Segment Abstract Objects



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 40

Storage Server Internals



University of California
Lawrence Livermore
National Laboratory

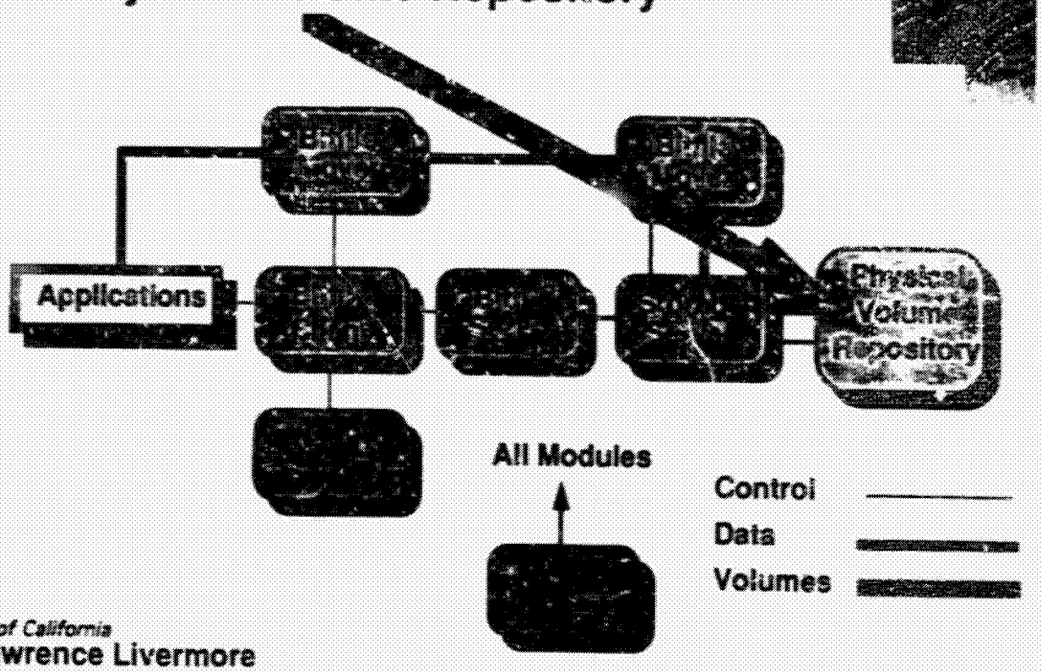
Goddard SC 9/22/92 4 1

Storage Server Functions

- **SS-Allocate**
  - Allocate space and return segment descriptors
- **SS-Deallocate**
  - Deallocate space and return it to free list
- **SS-Retrieve**
  - Transfer specified data to client
- **SS-Store**
  - Accept specified data from client

University of California
Lawrence Livermore
National Laboratory

The Physical Volume Repository



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 4 3

The Purpose of the Physical Volume Repository

- The Physical Volume Repository provides manual or robotically retrievable shelf storage of physical media volumes.
  - Tape reels, tape cartridges, optical disks, etc.
- Manages abstract objects called "physical volumes"

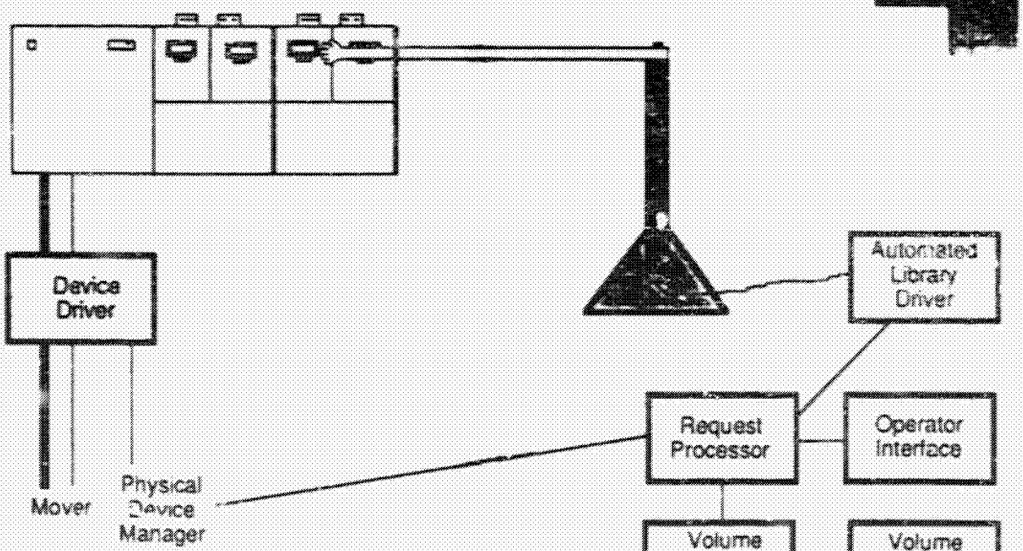
Physical Volume Descriptors

- Contains
  - Physical location
  - Human-readable label
  - Media type
  - Owner information
  - Access-control information
  - Statistics
- Descriptors are not stored on the physical volume
  - Reference model does not specify media format
  - Some media cannot be updated
  - Descriptor contains the physical location

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 4 5

Interaction Between the Physical Storage System and the Physical Volume Repository



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 4 6

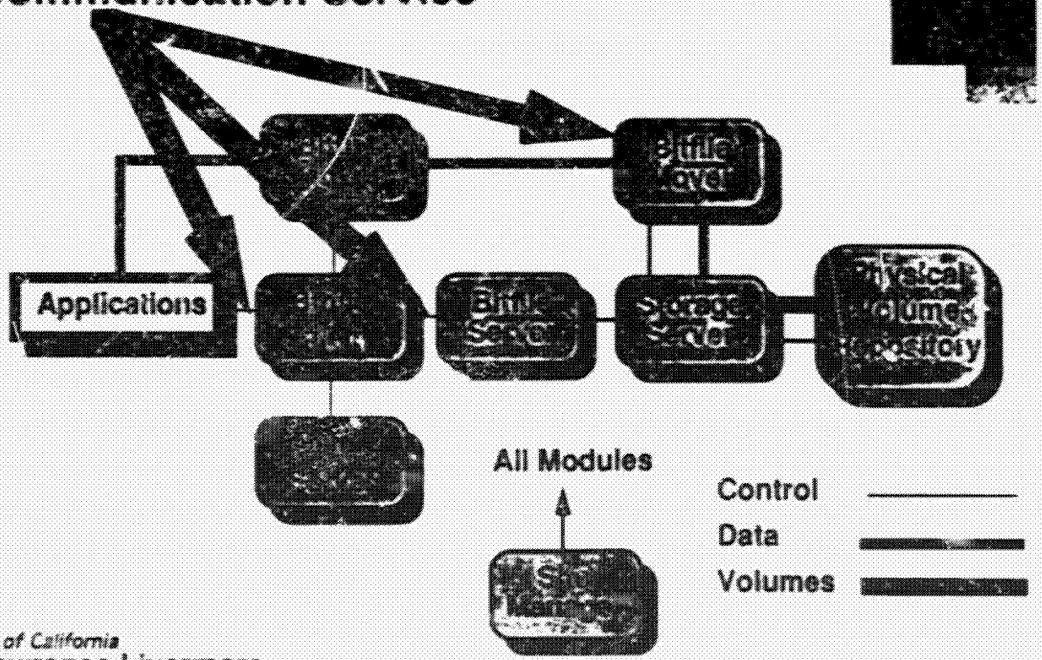
Physical Volume Repository Functions

PVR-Dequeue	Remove request from queue
PVR-Dismount	Remove volume from device
PVR-Eject	Remove volume from the PVR
PVR-Locate	Report location of volume
PVR-Mount	Mount volume on a device
PVR-ReadQueue	Return list of queued volumes
PVR-ReadStatus	Return status of specified device
PVR-SetStatus	Set status of specified device

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 4 7

Communication Service



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 4 8

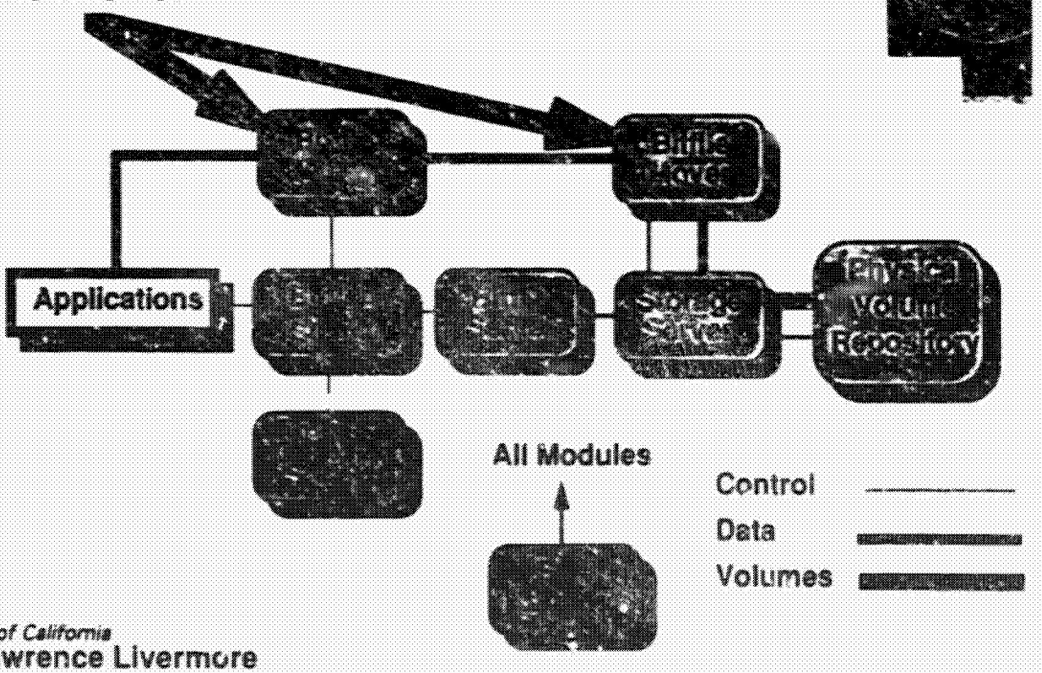
Purpose of the Communication Service

- Provides control and data communication among the modules of the model using existing protocols
- Includes the Mover

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 49

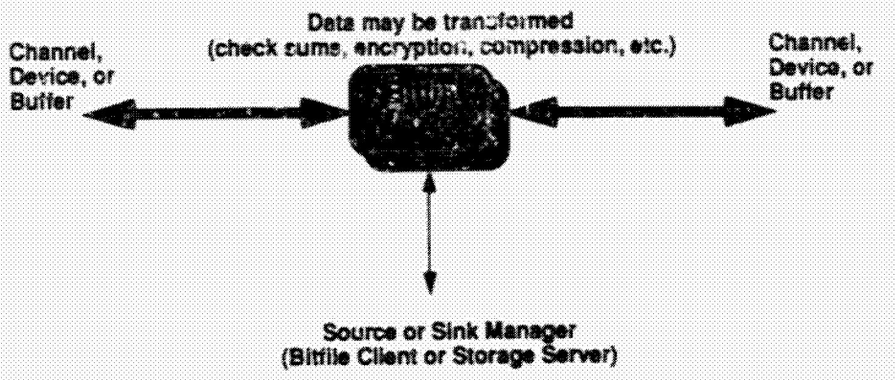
The Mover



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 50

Mover Architecture



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 5 1

Mover Function

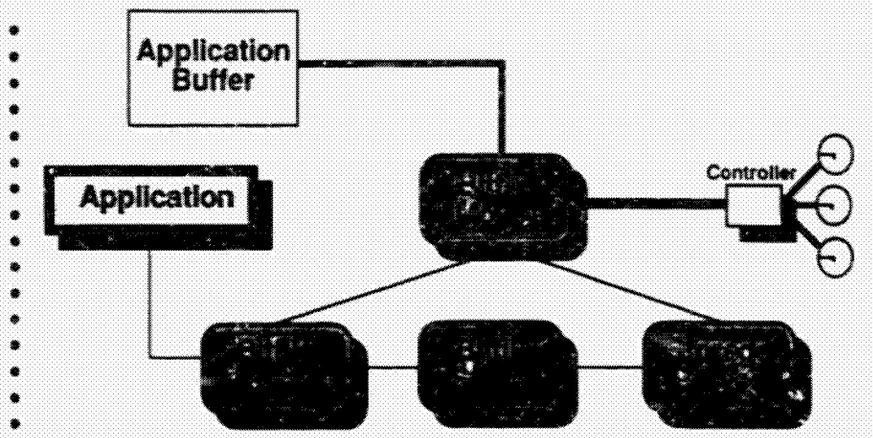
- **Move**
  - Move data from specified source to sink
  - Return the number of bits moved

ORIGINAL PAGE IS
OF POOR QUALITY

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 5 2

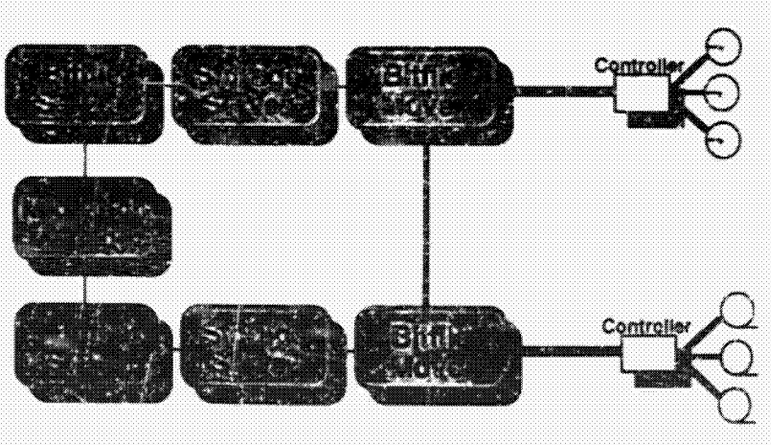
A "Local" Mover Configuration



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 53

Using Movers to Migrate Data

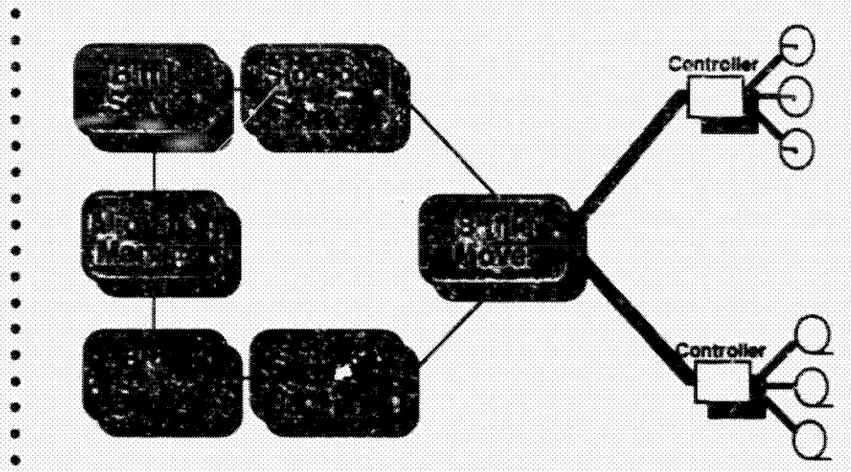


ORIGINAL PAGE IS
OF POOR QUALITY

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 54

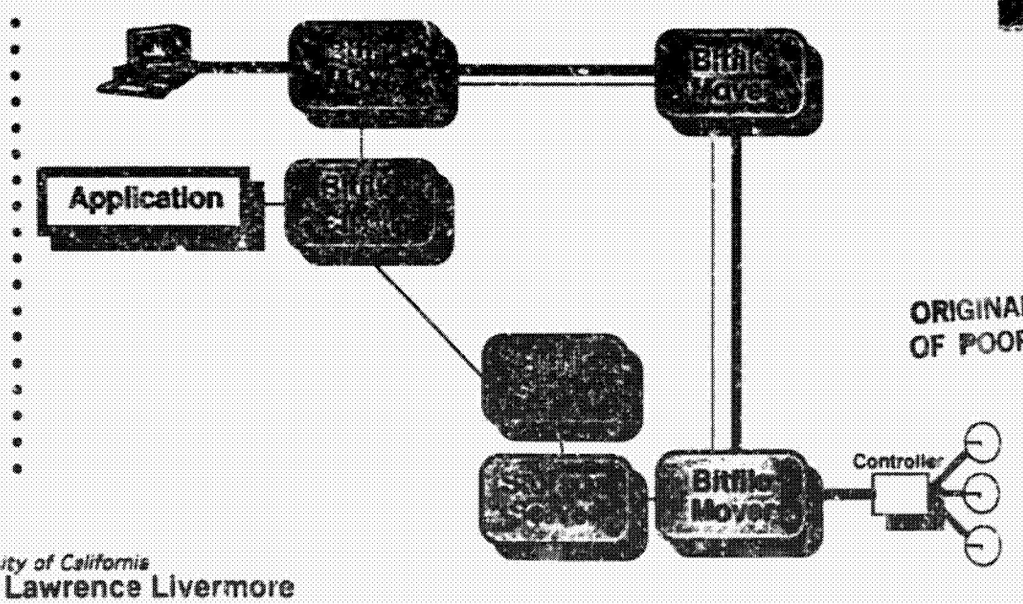
Optimizing a Local Mover



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 55

Using Movers Across a Network

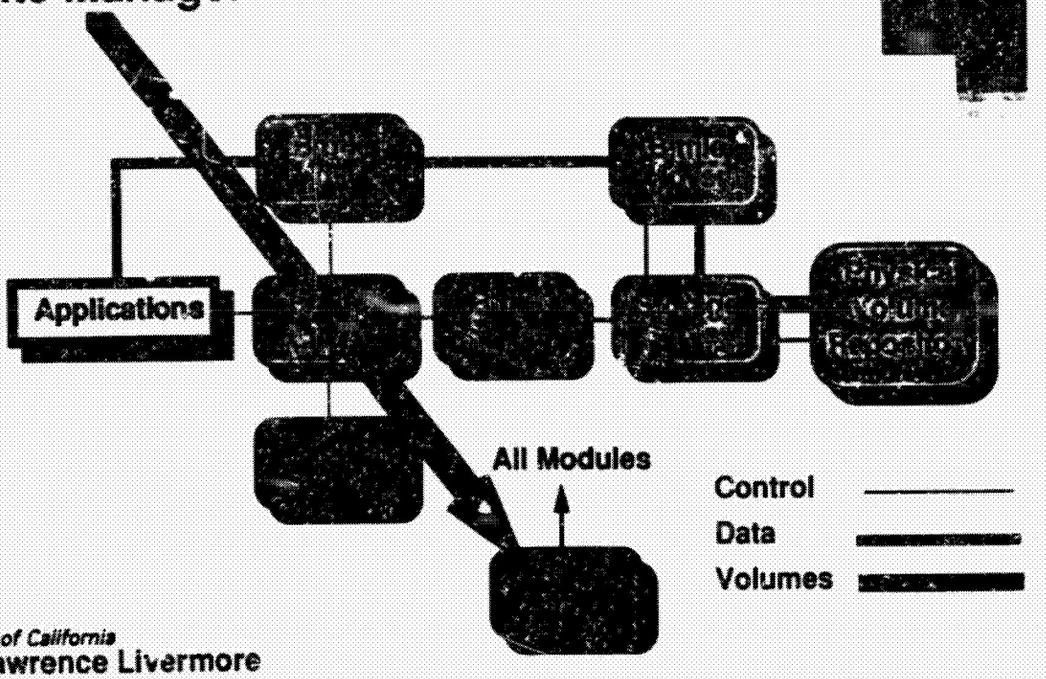


ORIGINAL PAGE IS
OF POOR QUALITY

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 56

Site Manager



University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 57

The Purpose of the Site Manager

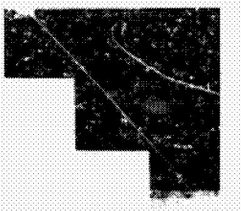
- Monitors operations
- Collects statistics
- Establishes policy
- Exerts control over policy and site operation
  - Set policy parameters
  - Installs logical and physical volumes
  - Run diagnostics
  - etc.

ORIGINAL PAGE IS
OF POOR QUALITY

University of California
Lawrence Livermore
National Laboratory

Goddard SC 9/22/92 58

Site Manager Functions



- **Storage Management**
  - Optimizes devices and performance
- **Operations**
  - Monitors systems and resolves problems
- **Systems Maintenance**
  - Maintains system component performance and reliability
- **Software Support**
  - Maintains development and test environment
- **Hardware Support**
  - Displays, diagnoses, and corrects hardware failures
- **Administrative Control**
  - Maintains security, accounting, management policies

N93-80451

Optical Media Standards for Industry

Kenneth J. Hallam
ENDL Associates
29112 Country Hills Road
San Juan Capistrano, CA 92675

Optical storage is a new and growing area of technology that can serve to meet some of the mass storage needs of the computer industry. Optical storage is characterized by information being stored and retrieved by means of diode lasers. When most people refer to optical storage, they mean rotating disk media, but there are 1 or 2 products that use lasers to read and write to tape.

Optical media also usually means removable media. Because of its removability, there is a recognized need for standardization, both of the media and of the recording method.

Industry standards can come about in one or more different ways.

1. An industry supported body can sanction and publish a formal standard. Examples of such bodies include ANSI, AIIM, ECMA and ISO.

2. A company may ship enough of a product that it so dominates an application or industry that it acquires 'standard' status without an official sanction. Such de facto standards are almost always copied by other companies with hopes of success.

3. A governmental body can issue a rule or law that requires conformance to a standard. The standard may have been created by the government, or adopted from among many proposed by industry. These are often known as de jure standards.

Standards are either open or proprietary. If approved by a government or sanctioning body, the standard is open. A de facto standard may be either open or proprietary.

Optical media is too new to have de facto standards accepted by the marketplace yet. The proliferation of non-compatible media types in the last 5 years of optical market development have convinced many of the need for recognized media standards.

There are 3 organizations presently working to establish recognized standards for optical media.

ANSI - The American National Standards Institute, committee **X3B11**
ECMA - The European Computer Manufacturers Association, committee **TC31**
ISO - The International Standards Organization, committee **SC23 and SC15**

Membership in **ANSI** is open to individuals, organizations and companies.

Membership in **ECMA** is open to companies that manufacture products in Europe.

Membership in **ISO** is open only to countries.

All work on the technical committees of all 3 organizations is accomplished by volunteers from industry. The volunteers pay their own way and spend the time necessary to formulate, draft and critique proposed standards.

As might be expected, many of the same individuals can be seen on the 3 different optical committees. The manufacturers of optical media and drives dominate the 3 committees devoted to optical media. There are a sprinkling of others, including semiconductor

manufacturers, software firms, CPU companies and user organizations. The user community is probably the most under-represented at these committees.

All 3 committees work hard to keep some level of coordination between each other. The ANSI committee X3B11 is the place that US optical media standards are developed and the recommendations for a US national position at ISO are formulated. X3B11 also nominates delegates for the US to send to the meetings of ISO-SC23.

Because ECMA is a European trade association, there are no direct, formal links between it and X3B11. However, there are formal and long established links between X3B11 and ISO/SC23. Since many of the companies that send people to the meetings of X3B11 are also members of ECMA TC31, the informal communication path is well used. ECMA is also an advisory member of ISO/SC23 and often makes direct submissions to ISO with proposed international standards.

When the members of SC23 and X3B11 can agree on all points of a proposed standard, it is possible to have a joint ANSI/ISO standard, published as a single document. However, there are often many reasons for divergence within a standard, so we wind up with both an ISO and an ANSI standard for the same thing.

The members of optical media standards committees are working under a handicap in that much of the technical data necessary to develop a comprehensive draft is still being discovered in the laboratory. The fundamental first order physics of optical recording is not as well understood as thatof traditional magnetic recording.

The true test of a proposed standard physical parameter or measurement is can several people, working in different laboratories achieve the same results consistently? It takes tremendous resources to define and develop a standard and without the time, materials and equipment generously provided by the volunteers from industry, it would not be possible.

In spite of the amount of detail present in a media standard, people often ask why the documents don't go even farther in defining what a piece of standard media looks like? A standard is not intended as a complete blueprint on how to build a product, nor as a purchase specification. Instead, it exists as a framework under which it is possible for multiple manufacturers to build products that have a good chance of true interchange.

Issues like quality and manufacturing process are best left to the market place. There must be room within a standard for companies to add value and offer their own vision of what the customer needs.

Existing standards for optical media:

CD-ROM

ISO 9660	This standard defines the file structures on a CD-ROM. Capacity = approx. 550MB
ISO 10149	Defines the physical media and the sector format of CD-ROM.

Optical WORM Computer Media:

ISO 9171	Defines a common 130mm, (5.25 inch) optical media, cartridge and sector format for 2 independent and non-compatible servo systems, (CCS and SSF). Capacity = 320MB/side
ISO 11560	Defines 130mm media that uses magneto-optic recording but is treated as WORM media. Also uses the CCS format. Capacity = approx. 320MB/side
ISO 10885	Defines 356mm, (14 inch) optical media and cartridge, as well as the sector format. Capacity = 3,400MB/side
ANSI X3.191-1991	Defines 130mm WORM optical media and cartridge that uses the SSF format and RZ recording. Capacity = approx. 650MB/side
ANSI X3.211-1992	Defines 130mm WORM optical media and cartridge that uses the CCS format and RLL 2,7 recording. Capacity = approx. 320MB/side
ANSI X3.214-1992	Defines 130mm WORM optical media and cartridge that uses the SSF format and 4/15 recording. Capacity = approx. 320MB/side
ANSI X3.200-1992	Defines 356mm WORM optical media and cartridge and sector format. Capacity = approx. 3,400MB/side
ANSI X3.220-1992	Defines 130mm media that uses magneto-optic recording but is treated as WORM media. Also uses the CCS format. Capacity = approx. 320MB/side

Optical Rewritable, (Erasable) Computer Media

ISO 10089	Defines a common 130mm Rewritable media and cartridge with 2 non-compatible servo formats, (CCS and SSF). Capacity = approx. 320MB/side
ISO 10090	Defines 90mm Rewritable/Read-Only optical media and cartridge that uses the CCS format. Already approved by ISO, this standard is being considered for submission as an ANSI standard. Capacity = approx. 128MB, (single sided).
ANSI X3.212-1992	Defines 130mm Rewritable optical media and cartridge that uses the CCS format. Capacity = approx. 320MB/side
ANSI X3.213-1992	Defines 90mm Rewritable/Read-Only optical media and cartridge that uses the DBF format. Capacity = approx. 120MB, (single sided).

Projects active within standards committees:

130mm Extended Capacity Media. A simple extension of standards 10089 and X3.212 with 650MB/side capacity and a goal of complete compatibility with first generation media at the drive level. Also known as the 2X media project. Projects active at ECMA and ISO/SC23. Est. completion 6/93

130mm Second Generation Media. Anticipated capacity is over 1GB per side. Also known as 3X media project, it is active at ISO/SC23 and X3B11. Est. completion 10/94

90mm Second Generation Media. Anticipated capacity is over 300MB on single sided media. Goal is to offer compatibility with first generation media at the drive level. Est. completion 10/94

300mm WORM Media. This is actually 2 projects at the present time. There is no agreement on a single servo method, (between CCS and SSF) and there may be separate standards for plastic and glass media as well. Projects active at ISO/SC23 and X3B11. Est completion 6/94

File and Volume Structure. This project is intended to affect all sizes of optical media. It defines the software volume and file structure that is needed to achieve true media interchange between systems. It has been developed with the needs of UNIX, MS-DOS and DEC operating systems in mind. It is possible that by the time of this conference, the draft will have been approved as a standard by ISO or ECMA. The work is essentially complete and only a review and final votes are needed at both ANSI and ISO. There are active projects at ISO/SC15, X3B11 and ECMA TC15. Est. completion 12/92

Capacity Trends

The next generation standards for 90mm and 130mm Rewritable media will not likely go beyond the stated capacities because of the lack of commercial laser diodes with short wave lengths and sufficient power for this type of application. The first generation optical media products had the benefit of a popular consumer application, (the music CDs) that caused high production volumes in 780nm wave length laser diodes. These high volumes led to low cost components and the boost in power needed for WORM or Rewritable drives was not too difficult for the diode manufacturers to make.

To obtain significant increases in bit densities on optical disks, you must write a smaller spot on the media. The only way to do that economically today is with laser diodes with a shorter wave length. Laser diodes of at least 670nm wave lengths and power outputs of 30-40mW are needed. There is no consumer application at present for laser diodes with these characteristics. This leaves prices for such diodes in the range of several hundred dollars each because of the low manufacturing volumes.

The demand for optical disk drives for computer applications is simply not high enough to justify a laser diode manufacturer investing in a lot of capital equipment to make components for these drives when other opportunities are present.

This situation may change as consumer applications open up for shorter wave length diodes. Another possibility is the use of "trick" optics that act as frequency doublers. Double the light frequency and halve the wave length. The only problem is that at present, these techniques are not very efficient. The original power level is reduced by as much as 90%.

So, given the existing limits of technology, most optical storage media can be expected to go to 2X or 3X over present capacity limits in the next few years. These capacity gains are achieved by a combination of code techniques and track density improvements.

N93-30452

53-82
15?5?3
p. 10

Technology for National Asset Storage Systems

Robert A. Coyne
Harry Hulen

IBM Federal Systems Company
9700 Bay Area Blvd., Houston, TX 77058

Richard Watson

Lawrence Livermore National Laboratory
P.O. Box 808, Livermore, CA 94550

Abstract

An industry-led collaborative project, called the National Storage Laboratory, has been organized to investigate technology for storage systems that will be the future repositories for our national information assets. Industry participants are IBM Federal Systems Company, Ampex Recording Systems Corporation, General Atomics DISCOS Division, IBM ADSTAR, Maximum Strategy Corporation, Network Systems Corporation, and Zitel Corporation. Industry members of the collaborative project are funding their own participation. Lawrence Livermore National Laboratory through its National Energy Research Supercomputer Center (NERSC) will participate in the project as the operational site and the provider of applications. The expected result is an evaluation of a high performance storage architecture assembled from commercially available hardware and software, with some software enhancements to meet the project's goals. It is anticipated that the integrated testbed system will represent a significant advance in the technology for distributed storage systems capable of handling gigabyte class files at gigabit-per-second data rates. The National Storage Laboratory was officially launched on May 27, 1992, when executives of the six founding industrial participants and John H. Nuckolls, Director of Lawrence Livermore National Laboratory, signed an agreement with Admiral James D. Watkins, Secretary, U. S. Department of Energy.

1. The High Performance Data Storage Environment

In recent years, transferring and storing information has become a major challenge in the high performance computing arena. Scientists at Livermore and other research labs now wait for hours - and even days - to retrieve their supercomputer data. While today's storage systems can move ten to twenty million bits of information each second, requirements exist today for architectures with capacity of 100 million to one billion bits per second.

Excellent research and development is taking place in the processors, communications networks, and media required to handle very large volumes of data at very high data rates.

- Several gigabit class fiber optic networks are planned that enable cross-continent access to high rate/high volume data.
- ANSI has taken the lead in providing high performance interconnect standards such as HIPPI, FDDI, IPI, SCSI, and the forthcoming Fiber Channel Standard that enable vendors to design subsystems that can work together at high data rates.
- Data striping and RAID technology have leveraged system performance an order of magnitude beyond the performance of individual devices.[1]

- Distributed storage systems, such as the Andrew File System (AFS)* that originated at Carnegie-Mellon University, offer the prospect of nationwide and worldwide storage systems which present a single file system image to the users.[2]

- The IEEE has established a Storage System Standards Working Group to bring users and vendors together to address the creation of a standards-based framework to allow subsystems and components to fit together.[3]

The national asset storage system of the near future is much more than an aggregation of high capacity components. New concepts are required to bring the available component technologies together into useful, manageable systems.

2. The Need

There are significant needs that must be addressed if the goal is to create nationwide high performance distributed storage systems.

2.1 Network-attached High Performance Storage

Most all operational storage systems at the national laboratories and supercomputer centers use general purpose computers as storage servers. These storage servers connect to storage units such as disks and tapes and serve as intermediaries in passing data to compute intensive nodes on their networks. At the Livermore Computer Center, for example, disks and tapes are connected to an Amdahl mainframe that honors requests from users on other systems, primarily Cray supercomputers, to read or write data to the storage devices.[4] As the data rates of storage devices increase, the storage server must handle correspondingly faster communications links such as HIPPI today and Fiber Channel Standard in the near future. This tends to drive the storage server computer into the supercomputer class, based on the required data bandwidth.

An alternative is to attach the storage devices (or to be precise, their control units) directly to the network. This would eliminate the need to store and forward data through a general purpose mainframe or supercomputer functioning as storage server. There is precedent for this. At the National Center for Atmospheric Research, there has existed for several years a storage system that uses an IBM mainframe computer to set up storage transfer requests but then transfers the data directly from a storage device to a Cray supercomputer. A specially designed component of the communications system built by Network Systems Corporation serves to transfer the data between the storage unit and the supercomputer. Therefore, the data does not have to flow through the IBM mainframe. Instead, the IBM mainframe functions more as storage manager than storage server in the traditional sense. More recently, Maximum Strategy Corporation and IBM have offered disk arrays capable of HIPPI connectivity that have the potential to serve as network-attached storage devices.

2.2. Multiple, Dynamic, Distributed Storage Hierarchies

Current storage systems including General Atomic's DataTree and UniTree** and IBM's System Managed Storage provide a single hierarchy of storage media. In this single hierarchy,

---

* AFS is a registered trademark of Transarc Corporation.

** DataTree and UniTree are trademarks of General Atomics

frequently used data is kept on disk, less frequently used data is kept in an automated tape library, and infrequently used data is kept in tape vaults.

With the availability of new media such as solid state disk, disk arrays, and helical scan tape, there is frequently no single hierarchy which can be applied to all data and all media.

Consider the problem of technology insertion. This may take the form of adding a new type of data storage device into an existing system, a type of tape, for example. Further assume that the existing storage devices will continue to be used. Now, what was a simple disk-to-tape hierarchy becomes as many as three or four hierarchies: disk to old tape, disk to new tape, old tape to new tape, and perhaps even new tape to old tape. Current storage systems do not have the mechanisms to handle this level of complexity.

Also, consider what happens when a system grows to regional or national scale. Multiple centers, each with disks and tapes, must cache recently accessed data to high performance media and migrate less recently accessed data to less expensive media. Usually this is done locally, but occasionally there may be requirements to migrate data from disks (or tapes) at one location to tapes at another location. In a system that recognizes only one hierarchy, extensive human supervision and intervention is required to handle inter-datacenter migration.

Multiple hierarchies are needed, based on such factors as location, data type, cost, and project affiliation. Each hierarchy must be adaptable to meet specific requirements and must be able to change over time under the control of a system administrator. The concept of multiple dynamic hierarchies is described in more detail in an IBM FSC research paper[6] and are being proposed to the IEEE Storage Systems Standards Working Group.

2.3. Layered Access to Storage System Services

In distributed systems, storage services should be presented at several layers of abstraction. At the higher levels, the services should provide a fully transparent file abstraction. At this level, concurrent access is synchronized, caching must detect conflicting read/write patterns by multiple clients, and access and modification times are tracked with timestamps. These are common characteristics taken for granted when using a file.

At a lower level, where there is less need for transparency, the user may need to work with objects such as disk or tape blocks. These objects are provided without synchronization, caching, access records, etc. At an even lower level, individual devices become visible, requiring even more intimate knowledge of the semantics of the storage media. For example, the user must take into consideration such characteristics as write-once versus read/write and sequential versus random access. It is common to build important classes of applications, such as database management systems, directly on these lower level abstractions.

At the higher levels, the fully transparent file abstraction is more convenient for most end-user applications. Transparencies such as location independence can make applications much easier to write and allow portability of both application and storage objects. But these extra components of the abstraction impose significant performance penalties that some applications cannot afford. For example, a database application is intimately tied to its storage access pattern and, given the appropriate low level access, can optimize its accesses much more effectively than a generic caching file system. The database application must be given access to the lower level abstraction of disk blocks, and it must be more sophisticated to access and manage its storage at the lower level. Other clients that are more interested in high performance than in the file abstraction might include a paging system, that needs the

highest level of performance with minimum overhead from networked solid-state disks, or a satellite downlink, that receives data at such a high rate that it cannot pass through a disk buffer but must be written directly to the raw tape devices for later processing.

A national resource file system must provide levels of abstraction appropriate for diverse classes of applications. These will range from the fully abstract and highly transparent interfaces for the general user to low level services for use by sophisticated applications that are willing to obtain high performance in exchange for a more complex storage abstraction. It is essential that a storage system have the appropriate modularity to support multiple layers of storage abstractions over a range of user interfaces, file naming conventions, storage hierarchies, network connected devices, and storage management strategies.

2.4. Storage System Management

Storage system management is the collection of functions concerned with control, coordination, monitoring, performance and utilization of the storage system. These functions are often interdependent, involve human decision making, and span multiple servers. Management functions may be implemented as stand-alone programs, integrated with other storage system software, or implemented as policy. The Storage System Manager can be thought of as the collection of management processes that performs all necessary storage system management functions.

Storage system management attempts to allocate the resources of the storage system to the best use for the overall benefit of the site. Policies for the site must be set, and manual and automatic procedures must be developed to implement those policies. The procedures must be adaptable because the requirements will change as time progresses and because the same software may be run at a number of different sites.

Current storage system software packages provide tools needed to manage individual sites. As gigabit networks blur the distinction between local and remote, and as national storage systems are created, the issue of system management becomes of great concern.

3. The Collaboration

Six companies with interests in the technology for very high performance storage systems joined together with Lawrence Livermore National Laboratory to found the collaborative research project. The collaboration is defined by a set of Cooperative Research and Development Agreements (CRADAs) between the industrial participants and the Department of Energy. The National Storage Laboratory was officially launched on May 27, 1992, when executives of the six founding industrial participants and John H. Nuckolls, Director of Lawrence Livermore National Laboratory, signed an agreement with Admiral James D. Watkins, Secretary, U. S. Department of Energy.The collaboration is self-funded, with participants providing both equipment and labor. The roles of the founding participants are:

- IBM Federal Systems Company is serving as systems integrator and project coordinator. IBM Federal Systems Company is providing RISC System/6000 computers, and IBM ADSTAR is providing a 20 gigabyte HIPPI-attached high performance disk array.

- Ampex Recording Systems Corporation is contributing technology and equipment for a HIPPI-attached very high speed, high capacity cartridge tape library system.

- General Atomics DISCOS Division is providing UniTree storage system software that will serve as the framework and point of departure for the software capabilities to be developed.

- Maximum Strategy Corporation is working with IBM and Ampex to configure very high rate tape and disk control units capable of network attachment.

- Network Systems Corporation is providing expertise in network design and is supplying network switches, routers, and gateways.

- Zitel Corporation is providing a HIPPI-attached solid state memory device capable of data rates limited only by network performance.

- Lawrence Livermore National Laboratory, through its National Energy Research Supercomputer Center, is serving as the operational environment and host site for the collaborative project and the source for applications.

Four National Science Foundation laboratories have joined the NSL as members of the Executive Committee:

- Cornell Theory Center

- National Center for Supercomputing Applications

- Pittsburgh Supercomputing Center

- San Diego Supercomputer Center

Since its founding, the collaboration has been joined by three other contributing companies:

- CHI Systems, Inc. is providing HIPPI adapters for several of the computing and I/O subsystems.

- IGM-ATL, Inc. is providing a SCSI-attached 8mm tape library system.

- PsiTech Inc. is providing a HIPPI-attached high performance frame buffer.

It is expected that some additional growth in collaboration membership will occur, and indeed will be welcome. Growth in the collaboration is expected to be vertical, with new members offering a hardware, system software, or application interest that does not duplicate and is complementary to the interests of the existing membership.

4. The IEEE Storage Systems Standards Working Group

An important forum for users, developers, researchers, and suppliers to come together to work on mass storage system issues has been the IEEE Storage Systems Standards Working Group. This standards body, organized in May 1990, will provide guidelines and standards for scalable, distributed, multivendor storage systems. All of the member organizations of the joint research project are members of the IEEE Storage Systems Standards Working Group. The four basic tasks of this study, network-attached storage, multiple hierarchies, layered protocols, and storage system management, are areas of interest in the Standards Working Group.

5. Overview of the Prototype Storage System

The collaborative project will develop an operational prototype to be called the National Storage Laboratory. Figure 1 shows the prototype system as it is envisioned toward the end of

1992. This prototype will augment existing systems in the National Energy Research Supercomputer Center (NERSC) and the Open Computing Facility (OCF), both of which are located in the Livermore complex. The National Storage Laboratory will be used by LLNL scientists in their work in climatology and fusion research.

The storage system, as shown in Figure 1, is conceptually divided into four functional groups of equipment. These groups represent the computational resources, the user areas, the network storage devices, and the components that provide access control and storage system management. Various networks are shown connecting the various components to provide separation of control functions and data movement functions. These functional groups of processors, storage devices, and networks are described in the following paragraphs.

5.1. LLNL Computational Resources

Closely tied to the collaborative research project are the computational engines that are both the producers and consumers of massive quantities of data. The prototype storage system will be connected to CRAY-2 and CRAY Y-MP C-90* supercomputers at NERSC. Representing another class of computational resource will be a cluster of IBM RISC System/6000** workstations that comprise LLNL's Open Computing Facility. The computational complexes will be connected to the pool of network-attached storage resources by the Direct Data Transfer Network and to the access and control mechanisms by a Storage Access Control Network functional group. Each computational system has its own private storage and will continue to be connected to existing shared storage systems at NERSC and OCF, that are for simplicity not shown in Figure 1. Also, there are existing facility networks that provide access to users of the these computational complexes; these networks are also not shown in Figure 1.

5.2. User Areas

Users of the prototype storage system will have their own deskside UNIX workstations and will be connected via existing facility networks to the NERSC and OCF computational complexes. These workstation users will expect to use standard network file services such as NFS or AFS. Consequently, the design of the storage system prototype includes a Secondary Storage Server that provides an NFS or AFS compatible file system managed by an IBM RISC System/6000 computer. The Secondary Storage Server is in turn connected to the Network Storage Resources via the Direct Data Transfer Network and to the Access Control and Management functional group via the Storage Access Control Network. Local disk arrays on the Secondary Storage Server provide speed matching and caching between the high performance network storage resources and the workstations. Also shown in the User Area functional group is a Frame Buffer. The Frame Buffer is connected to the Direct Data Transfer (HIPPI) network and is capable of displaying movie-like sequences of high resolution images from either the storage resources or the computational resources.

Corresponding to the prototype's single User Area, as shown in Figure 1, a future full implementation of these concepts would contain many such user area functional groups, some locally attached and some remote. Each would have access to a local Secondary Storage Server and perhaps to a Frame Buffer. The Secondary Storage Server itself may be a basic workstation with a few large disks, or it may be a mainframe or a specialized storage system product.

---

* CRAY Y-MP C90 and CRAY-2 are trademarks of Cray Research, Inc.
** IBM RISC System/6000 is a trademark of International Business Machines Corporation.

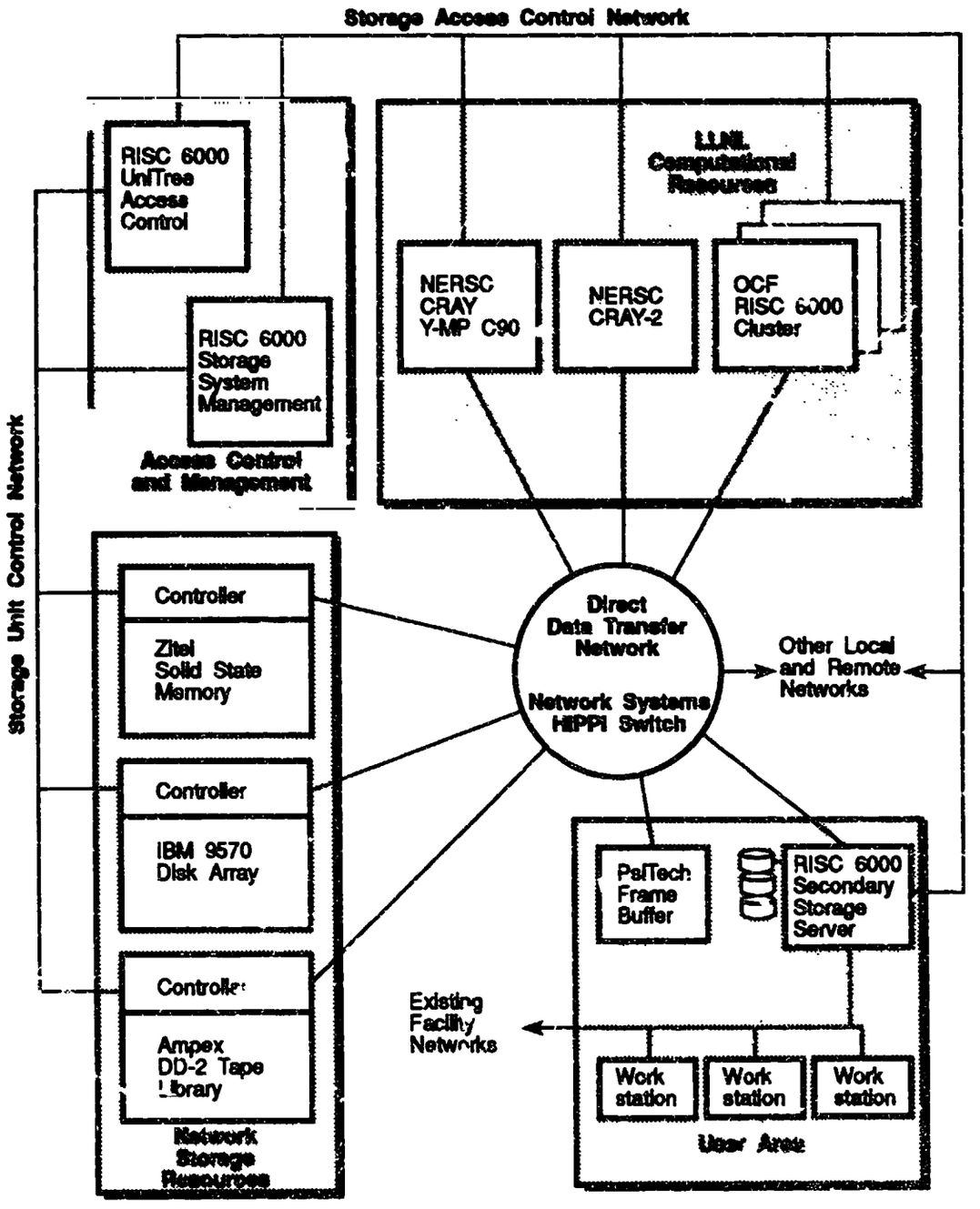


Figure 1. Conceptual Overview of the National Storage Laboratory at Lawrence Livermore National Laboratory

The Open Computing Facility's RISC System/6000 Cluster which is shown in Figure 1 as a single box without internal detail, will be in many ways like the User Area, with one of the RISC System/6000s serving as a Secondary Storage Server for the other members of the cluster. The main difference between the RISC System/6000 cluster and the User Area functional group is in the way the systems are used. The User Area computers are interactive workstations usually dedicated to an individual or a small group of people, while the workstations in the Open Computing Facility cluster are compute servers to which batch jobs are scheduled. Thus, the User Area functional group represents a building block or "generic object" that can be replicated and employed in different ways.

5.3. Network Storage Resources

One of the objectives of this research project is to gain experience with the concept of network-attached storage devices. Conceptually, network-attached storage devices can be shared by several processors while not being "owned" in a conventional sense by any of them. The functional group in Figure 1 labeled Network Storage Resources represents the high performance network-attached storage devices planned for the prototype storage system. They are a solid state memory device from Zitel, a disk array from IBM, and a robotic tape system from Ampex. The Ampex tape system will use a 19 millimeter helical scan tape format called DD-2. All three of these storage units have intelligent controllers that are connected to both the Direct Data Transfer network using a HIPPI interface and to the Storage Unit Control network via a conventional local area network interface. The controllers for the IBM disk array and the Ampex tape subsystem are from Maximum Strategy Corporation, a member of the collaborative project.

Commands to direct the Network Storage Resources components to send and receive data will be sent from system components represented by the Access Control and Management functional group in Figure 1. Data will be transferred directly to the components represented by the LLNL Computational Resources functional group and the User Area functional group via the high speed Direct Data Transfer Network. Thus, there is a separation of control and data functions. In the prototype, this separation of control and data, which is a logical concept, will be enforced by a physical separation of the control and data networks.

5.4. Access Control and Management

In most existing storage systems, both control and data pass through a storage system processor that is often a mainframe. The NERSC, for example, uses a mainframe-based storage system in which all data passes through the memory of the storage system processor.

For the storage system prototype as shown in Figure 1, the controlling entity will be a workstation class computer. This is true even though the data rates to and from the storage devices are an order of magnitude faster than in existing LLNL storage systems. Such a design is possible because the data will flow directly between the user and the device, not through the controlling entity. This is enforced in the prototype by the decision that the Access Control and Management components are not connected to the Direct Data Transfer Network.

The components of the prototype Access Control and Management System will be two IBM RISC System/6000 computers. Conceptually, this complex could grow horizontally to more workstations, or vertically to a mainframe. Also, the access control and storage system management functions could be combined into one processor.

The Access Control and Management complex is where most of the software development will take place for the collaborative project. The Access Control functions will be based on

UniTree, which is a product of General Atomics DISCOS Division, a member of the collaborative project. The storage system management functions will be new.

Operationally, the UniTree component will receive requests to store or retrieve data over the Storage Access Control Network. The requests may originate from one of the high performance computational entities shown in the LLNL Computational Resources functional group or from the Secondary Storage Server shown in the User Area. The modified UniTree software will translate the request into commands directing one of the devices in the Network Storage Resources functional group to send data to the requestor or receive data from the requestor. These commands are sent over the Storage Unit Control network.

5.5. Networks

The principal logical networks are a Direct Data Transfer Network, a Storage Control Network (shown in Figure 1 as separate Storage Unit Control and Storage Access Control networks), and various facility networks.

The initial component of the Direct Data Transfer Network is a HIPPI switch from Network Systems Corporation, a member of the collaborative project. HIPPI stands for High Performance Parallel Interface and is an ANSI standard. There will be a fiber extender to allow connection to the Open Computing Facility's RISC System/6000 cluster, which is in another building and is well beyond HIPPI distance limitations. There will be provision for future attachment to remote networks through T3 and SONET, which will be studied following the initial phase of this project.

The Storage Access Control Network and the Storage Unit Control Network are the two control networks that are part of the prototype implementation. FDDI technology is the design point for these networks. However, the networks will initially be a mixture of FDDI, Ethernet, and HYPERchannel* with a Network Systems Corporation router bridging the technologies.

Existing networks at LLNL connect components within the NSRSC and the OCF and connect these complexes to the user areas. These existing networks will form an integral part of the overall system as seen by the user. They will provide access to existing storage systems and will remain in place throughout this study. They will provide connectivity between the user workstations and the central computational resources.

6. Applications

A fundamental aspect of the National Storage Laboratory's philosophy is to use appropriate scientific applications to help set priorities and test and demonstrate the concepts embedded within the system architecture and implementation. Three application domains have been chosen by Lawrence Livermore National Laboratory to test and demonstrate the system's effect on scientific productivity:

6.1. Climatic Models

The Program for Climate Model Diagnosis and Intercomparison (PCMDI) has as its goal to understand why different climate models produce different results between each other and with actual climate measurement data. PCMDI currently needs access to very large files and

---

* HYPERchannel is a trademark of Network Systems Corporation.

multifile datasets for a variety of post-processing analyses. The NSL architecture is expected to reduce data transfer times from hours to minutes.

6.2. Magnetic Fusion Energy Models

The Magnetic Energy Fusion (MFE) modeling and experimentation involves extensive computer simulation modeling as well as experimental studies. It is common in their modelling studies to fill the supercomputer disks with intermediate and final results and not be able to proceed until this data can be transferred to shared tertiary storage. This can cause delays of minutes to hours before additional runs can proceed.

6.3. Digital Imaging

Many scientific modeling calculations generate sequential digital images which are stored, retrieved, and viewed as motion pictures, known as "movie loops." In preparing these movie loops, scientists need to edit and evaluate the effectiveness of various generated images. Currently users must wait long periods while these movies are output to slow video media. With the NSL testbed's high performance frame buffer and high resolution display, together with the high performance data storage and retrieval capability, users will be able to store these movie loops directly on digital storage and play them back in real time.

References

1. Patterson, D., G. Gibson, and R. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *ACM SIGMOD* Chicago, pp. 109-116, June 1988.
2. Daniel Nydick et al., "An AFS-based Mass Storage System at the Pittsburgh Supercomputing Center," Digest of Papers, *Proc. Eleventh IEEE Symposium on Mass Storage Systems*, pp. 117-122, October 1991.
3. IEEE Program Action Request 1244, using a base document from the IEEE Technical Conference on Mass Storage Systems and Technology, "Mass Storage Systems Reference Model," May 1990.
4. Hogan, Carole, et al., "The Livermore Distributed Storage System: Requirements and Overview," Digest of Papers, *Proc. Tenth IEEE Symposium on Mass Storage Systems*, pp. 6-17, May 1990.
5. Merrill, John and Erich Thanhardt, "Early Experience with Mass Storage on a UNIX-based Supercomputer," Digest of Papers, *Proc. Tenth IEEE Symposium on Mass Storage Systems*, pp. 117-121, May 1990.
6. Buck, A. L., and Robert A. Coyne, "Dynamic Hierarchies and Optimization in Distributed Storage Systems," Digest of Papers, *Proc. Eleventh IEEE Symposium on Mass Storage Systems*, pp. 85-91, October 1991.

The *Visible Human Project* of the National Library of Medicine: Remote access and distribution of a multi-gigabyte data set

Michael J. Ackerman, Ph.D.
Lister Hill National Center for Biomedical Communications
National Library of Medicine, Bethesda, MD 20894

As part of the 1986 Long-Range Plan for the National Library of Medicine (NLM) [1], the Planning Panel on Medical Education wrote that NLM should "...thoroughly and systematically investigate the technical requirements for and feasibility of instituting a biomedical images library." The panel noted the increasing use of images in clinical practice and biomedical research. An image library would complement NLM's existing bibliographic and factual database services and would ideally be available through the same computer networks as are these current NLM services.

Early in 1989, NLM's Board of Regents convened an ad hoc planning panel to explore possible roles for the NLM in the area of electronic image libraries. In its report to the Board of Regents [2], the NLM Planning Panel on Electronic Image Libraries recommended that "NLM should undertake a first project building a digital image library of volumetric data representing a complete, normal adult male and female. This Visible Human Project will include digitized photographic images for cryosectioning, digital images derived from computerized tomography and digital magnetic resonance images of cadavers."

The technologies needed to support digital high resolution image libraries, including rapid development; and that NLM encourage investigator-initiated research into methods for representing and linking spatial and textual information, structural informatics [3].

The first part of the Visible Human Project is the acquisition of cross-sectional CT and MRI digital images and cross-sectional cryosectional photographic images of a representative male and female cadaver at an average of one millimeter intervals. The corresponding cross-sections in each of the three modalities are to be registerable with one another. A two year contract for acquisition of this data was awarded in August 1991 to the University of Colorado at Denver. Victor M. Spitzer, Ph.D. and David G. Whitlock, M.D., Ph.D. are the principal investigators.

Under the terms of the data collection contract, the cryosectional data will be returned to NLM as photographs of the cross-sections. But the goal of the Visible Human Project is a digital image library. Towards the end of the summer of 1992, a Request for Proposals (RFP) will be issued for the digitization of the cryosectional photographs. That contract is to be awarded in the early spring of 1993. When the Visible Human Project was conceived in 1989 it was projected that the best resolution that could be expected from this digitization process would be 2,000 pixels by 2,000 pixels in 24 bit color. It is now thought that a resolution of 3,000 pixels by 3,000 pixels or even higher should be required.

Assuming a resolution of 512 pixels by 512 pixels by 12 bits of grey tone for the CT and MRI data, and 2,000 pixels by 2,000 pixels by 24 bit color for the cryosectional data, the image part of the Visible Human data set will comprise approximately 50 gigabytes of uncompressed data. This would correspond to more than 75 CD-ROMs. Increasing the resolution of the cryosectional images to 3,000 pixels to 3,000 pixels would increase the size of the image library to about 110 gigabytes.

A distribute can be used to find pictures, and pictures can be used as an index into relevant text are being experimented with. Basic research is needed in the description and representation of structures, and the connection of structural-anatomical to functional-physiological knowledge. This is the larger, long term goal of the Visible Human Project: to produce a system of knowledge structures which will transparently link visual knowledge forms to symbolic knowledge formats, so that the print library and the image library become one unified resource for medical information.

[1] National Library of Medicine (U.S.) Board of Regents. Long Range Plan: Report of the Board of Regents. U.S. Department of Health and Human Services, Public Health Service, National Institutes of Health, 1987.

[2] National Library of Medicine (U.S.) Board of Regents. Electronic Imaging: Report of the Board of Regents. U.S. Department of Health and Human Services, Public Health Service, National Institutes of Health, 1990. NIH Publication 90-2197.

[3] Brinkley, J.F. "Structural informatics and its applications in medicine and biology." Academic Medicine, 1991, 66:589-591.

N93-30454

DATA MANAGEMENT IN NOAA

55-32
159095
p-9

William M. Callicott
NOAA/NE3DIS
FB4 Room 3316 OSD/5
Suitland, MD 20233

ABSTRACT

The NOAA archives contain 150 terabytes of data in digital form, most of which are the high volume GOES satellite image data. There are 630 data bases containing 2,350 environmental variables. There are 375 million film records and 90 million paper records in addition to the digital data base. The current data accession rate is 10% per year and the number of users are increasing at a 10% annual rate. NOAA publishes 5,000 publications and distributes over one million copies to almost 41,000 paying customers. Each year, over six million records are key entered from manuscript documents and about 13,000 computer tapes and 40,000 satellite hardcopy images are entered into the archive. Early digital data were stored on punched cards and open reel computer tapes. In the late seventies, an advanced helical scan technology (AMPEX TBM) was implemented. Now, punched cards have disappeared, the TBM system was abandoned, most data stored on open reel tapes have been migrated to 3480 cartridges, many specialized data sets have been distributed on CD ROMs, special archives are being copied to 12 inch optical WORM disks, 51/4 inch magneto-optical disks have been employed for workstation applications, and 8 mm EXABYTE tapes are planned for major data collection programs. The rapid expansion of new data sets, some of which constitute large volumes of data, coupled with the need for vastly improved access mechanisms, portability, and improved longevity are factors which will influence NOAA's future systems approaches for data management.

1.0. OVERVIEW

Data is the product of investigation. In science and technology, data is all that is left after the budget is spent and the opportunity is gone. Years later, some data are "dusted off" and made a part of research resulting with substantially more impact than the original use. For example, the "ozone hole" gleaned from old NIMBUS satellite data certainly had an impact far greater than the original investigators must ever have dreamed.

Also, experiments and their individual data collections eventually become part of a much larger investigation. In certain areas of environmental research, data are combined not only from a variety of contemporary experiments and routine measurements, but also from measurements made over extended time periods.

A fact of life in environmental research is that the observed periods of measurable change for important environmental parameters are greater than the professional lives of the scientists investigating those parameters. It is only through careful retention of data records that data will be useful for future applications. Perhaps a lifetime contribution of a scientist in some critical environmental areas might be his or her data carefully and skillfully collected over a career to support high quality scientific observations and conclusions many years in the future.

This paper addresses the functions of data management, those activities needed to help ensure that the investments made in data have the maximum chance for bearing a return. Please note that data management has boundaries. It is not, for example, to be confused with the science or technology that either produces or uses data. Within the bounds of data management are those activities which involve the planning for, management and preservation of, and provision for access to data held in an archive.

2.0. ENVIRONMENTAL DATA MANAGEMENT IN NOAA

2.1. CURRENT INFRASTRUCTURE

NOAA is in the midst of acquiring information to develop requirements specifications for the modernization of its data management infrastructure. Highlighted in this evaluation is the fact that NOAA must make mission adjustments as well as invest in improved data handling resources to resurrect its data system to preserve data for expanded research.

NOAA manages six World Data Centers, three National Data Centers, and over thirty centers of data. The National Data Centers constitute the formal NOAA centers and host the six World Centers of Data. The National Centers are distributed as the climate, ocean, and geophysical science discipline centers. The NOAA Centers of Data are made up of the various laboratories, science, and major processing centers, all which use and generate data in accomplishing their mission.

The NOAA Earth Data Directory system has over 1,100 directory entries which continue to increase as new data sets are added. The directory information is in the Directory Interchange Format (DIF) for international interoperability.

The NOAA archives contain 150 terabytes of data in digital form, most of which are the high volume GOES satellite image data. There are 630 data bases containing 2,350 environmental variables. There are 375 million film records and 90 million paper records in addition to the digital data base. The current data accession rate is 10% per year and the number of users are increasing at a 10% annual rate. NOAA publishes 5,000 publications and distributes over one million copies to 41,000 paying customers. Each year, over six million records are key entered from manuscript documents and about 13,000 computer tapes and 40,000 satellite hardcopy images are entered into the archive.

2.1.1. STORAGE MEDIA IN NOAA

Applying today's conservative approach in determining a low risk media for storing satellite data may not suffice for tomorrow's high rate large volume data sources. The polar and geostationary satellite data are handled differently because of the intended use of the data. The polar satellite data, is rigidly calibrated for quantitative processing, and GOES data is used primarily in a qualitative mode for near-real-time operational support.

The 3480 magnetic tape technology offers a cost effective low risk platform for storing polar satellite data. The anticipated data volumes from the polar satellites throughout the 1990s, with future systems like NOAA KLM, will not change sufficiently to force a decision to switch to a higher risk, large capacity and high data rate media.

In 1978, a contract was made with the University of Wisconsin to archive the GOES data on 19 mm SONY Umatic Beta tapes. The data are recorded in its original spacecraft retransmitted time-stretched mode. This mode of archive will continue throughout the life of GOES-H and likely with the METEOSAT gap-filler data. The decision of how to archive the GOES-I/M data beginning in 1994 has not been made.

Those NASA EOS and European polar satellite platforms, which included instruments NOAA intended to use in its operations, have been delayed beyond the year 2000. Like the NOAA KLM program, the data rates expected from other non-NOAA satellites will not be of a magnitude to force a change in the immediate future to a higher rate and capacity media.

The Data centers receive data on a variety of media forms. Many of the low volume data sets can easily fit in floppy disks, 5 1/4 inch magneto optical (MO) disks, CD ROM, 4 mm DAT, or 8 mm EXABYTE tapes. At this level, the risk to the data can be reduced by saving multiple copies of the data. However, not all of these media forms are suitable for long term archive. The continuing collection of conventional observation data sets likewise will not force a change to a higher capacity media. The exception to this is the National Weather Service NEXRAD doppler radar program. This system when fully deployed in 1997 has the potential of accumulating about 100 terabytes a year from its 159 operational stations. Currently, the NEXRAD system integrator is proposing a PC based EXABYTE media for the high volume level II source data and magneto optical disks for the lower volume level III product data. There may be a requirement to migrate from the low-end technology employed at the station level to some more robust higher rate and density recording system for the purpose of improving future access and retrieval.

One of three events will influence NOAA to select new media alternatives. One, a proven and reliable cost effective media will offer an opportunity to mitigate obsolescence; or two, the decision to change to an alternative storage media for the GOES-I program will leverage a change; or, three, NOAA will want a higher degree of compatibility for exchange of and access to data which will be leveraged by other programs supporting global change, i.e., EOS.

2.2. ANTICIPATED GROWTH

The digital data volumes collected from the NOAA satellite programs will grow from the current rate of about three terabytes per year to about seven terabytes per year from 1994 through 2002. After then, the NOAA satellite data volumes will almost triple. The non-satellite data accession rates will increase from the current amount of about 200 gigabytes annually to upwards of 100 terabytes annually when the National Weather Service NEXRAD doppler radar program is fully deployed in 1997. The NOAA digital data holdings will grow from 150 to 1,600 terabytes in the next 15 years. If the data used by NOAA from non-NOAA sources were added, the fifteen year total would be doubled.

2.3. CONNECTIVITY

A wide area network (WAN) has been established linking the Suitland and Camp Spring computing centers. This network is an ethernet link using the TCP/IP protocol. The WAN has recently been extended to the National Climate Data Center (NCDC) in Asheville, North Carolina, and to the National Oceanographic Data Center in downtown Washington, D.C. using 1 Mbps carriers. The Suitland WAN has been extended to include the Satellite Data Services Division (SDSD) of NCDC and will soon connect the SDSD office in another Camp Springs location. Other NESDIS access to the WAN is through INTERNET. The physical links between the World Weather Building (WWB) in Camp Springs, Maryland and Federal Building 4 (FB4) in Suitland, Maryland, use spare capacity on an analog microwave link between the WWB and FB4. This microwave system will be upgraded to a digital system later in 1992. The WAN interconnections are accomplished with an array of router systems connecting thin wire and optical cable. An internal data transfer rate of 10 Mbits is available with the external data transfer limited to the carrier service procured.

The WAN permits an expanded capability to exchange data and information between NOAA centers. Currently, about 30 Gbytes per week are exchanged over the network. All of the computer to computer exchange between the FB4 NESDIS and NMC computer centers use the WAN. Remote and local workstations in NESDIS with appropriate logon permission have access to the NESDIS and NMC computer systems.

The INTERNET allows for expanded use of the WAN systems to other remote NOAA sites and to researchers who need access to NOAA data and information. NOAA under the guidance of a NOAA-wide Network Advisory Board is pursuing resources to establish and manage a NOAA INTERNET node. With these facilities, it is expected that access to NOAA data and information will increase dramatically, expanding almost exponentially as global change research activity increases.

A predominate cultural tendency is direct contact. The sophisticated scientist who is a routine user of the data knows where, what and how about the data and doesn't need search assistance and merely phones in to order direct from the inventory. The beginning scientist, who is not sure what the data is or how to get it and what is required to read it, will either phone and discuss these items with a discipline scientist or a data systems person, or sometimes will visit to personally browse through the data. Mail requests usually come from individuals looking for a specific piece of information and generally want only an answer and not the data. Planners, lawyers, builders, and general public make up this group. At the National Climate Center, 30% mail their requests, 30% fax, 33% phone the center, about 4% visit and less than 3% use electronic mail services. All requests are currently serviced with manual intervention. The low number of electronic contacts are for several reasons. One is past culture. Another is that many users do not have the means to dial in for data and information, and most who do, don't know how. The most widely used electronic contact is through facsimile. This is because non-computer types can easily use the fax, though fax is probably no more widely dispersed than personal computers (PCs) in modern offices.

2.4. ANTICIPATED USAGE CHANGES

The rapid expansion of computer technologies and the rapid increase in media capacities available for the desk top users has evolved a community of computer literate data brokers looking for data. NOAA anticipates a growing need from a new user culture who will want to perform all of their information exchange function electronically.

As the workstation performance continues to reach new heights and affordable media technologies are developed to attach billions of bytes to the workstation, this new generation of users interested in environmental data will come to NOAA expecting sophisticated access

and distribution support. In this environment, the current manual access and search methods will not be sufficient to service the unsatiable need for data.

When the global science community became aware of potential global environment problems in the last decade, a ground swell of public and eventually political interest elevated the scientific interest in determining and predicting change if, in fact, change was believed to be taking place. In the late 1980s, government funds were authorized to increase support for environmental observation and research. This led to heightened interest in the decades of environmental data accumulated in NOAA's archive centers. As a result, interest in NOAA data is expected to increase at a much greater rate than the current 10% growth.

The present NOAA Earth Data Directory is a high level information directory itemizing NOAA archived data sets. This directory is intended to initiate the unfamiliar with NOAA data and information. It is operated as a level-1 catalog where reference systems are not electronically linked for passing the user to other reference directories or inventories. Most experienced users of NOAA data already know who to contact and how to search the NOAA inventory data, and usually order direct. Ninety per cent of the access to satellite data is made within 30 days after the data have been collected.

New computing capacities and communications bandwidth available to users of environmental data will demand a greater degree of automatic search and request capabilities from the NOAA centers. NOAA plans to modernize its data services to meet the expanding requirements for more data and increased automation of services by first, developing level-2 and eventually level-3 catalog interoperability services, and, second, implementing a hierarchical storage system which would place a substantial amount of data on-line. The ultimate goal would be for NOAA to develop catalog and delivery systems which would allow a user to build a customized subset of data from the variety of data and information held by NOAA which could be automatically assembled and dispatched to the user without delay. In this perfect system, the user could search and retrieve in a seamless environment.

3.0. PLANNING FOR COLLECTION OF MEANINGFUL DATA

The mechanisms of nature and their interactions are often subtle and not well understood. As a consequence, the creation and collection of experimental data must be accomplished with care or the subtleties could be lost. This is the experimenters' province and they, alone, must bear the responsibility for success at this early stage. The penalty for failure at this stage is lost opportunity not to mention a waste of resources and, worse, possibly misleading results for future researchers.

Critical factors in data management planning may include the following:

1 - adequacy of measurement, calibration and space/time reference accuracy

2 - adherence to standards

3 - documentation of the data and its use history

4 - adequacy and preservation of recording media

5 - access and distribution

This list might be considered by some to be a gratuitous list of factors that those in the business of processing data would certainly be concerned about without the need to be reminded. Experience, though, tells us otherwise. The NOAA GOES archive exemplifies how a data set should not be managed; though admittedly, the GOES system was never intended to be used as a precise measuring tool, but as a tactical tool for supporting NOAA's forecast and warning mission. With this system, the satellite instrument responses are irreversibly altered to

normalize the aberrations resulting from nonuniform detector responses of the eight visible channels, there was no on-board infrared calibration capability, the early geo-referencing system depended on a 30 hour forecast of the spacecraft attitude in an environment where orbit adjustments were periodically applied for station-keeping, the data were archived in the spacecraft downlink format on unique 19 mm commercial video recorders, and little documentation exists. Therefore, the twelve years of data with a data volume of more than 100 trillion bytes only has marginal use for global change science experiments. Even then, it will cost millions of dollars to fix many of the known data set problems before the data set can be rendered useful even as a relative data set to the science community.

4.0. MANAGING AND PRESERVING DATA

4.1. DOCUMENTATION AND STANDARDS

The national centers will require the necessary documentation to index the data to their system and establish an information record for future applications. There are compelling reasons for standards to document and describe the data record. A most critical reason is the portability of data for the convenience of access and input. Standards in these cases are rules establishing fully described links between the data and users. Another reason is to document the record for future access, with the value of documentation geometrically increasing as the time scale extends. Without detailed documentation, the data utility will diminish as it ages.

Standards are developed for the convenience of the user and protection of the data. Standards are necessary for ensuring adequate portability, particularly for future applications, as the systems used to record the data are replaced by new and sometimes different technologies. The risks associated with fitting to unique storage or recording structures is magnified by the frequent changes in technology. An example of the benefit of using data standards is the ANSI header file used to label computer tape data. It provides minimal documentation in a uniform system generated file for portability across systems, and it reduces the chance of job failures or delays by verifying the input, thus improving system efficiency.

Some standards are set in a de facto manner where agencies who generate large numbers of high volume data sets influence users of their data to alter their systems for the convenience of using the data. In the communications world, the highly successful TCP/IP protocol is a case where a de facto standard took hold. This interface protocol was developed in the absence of network standards. Because the network was large, manufacturers and system integrators conformed to the protocol in order to market their systems in a competitive market place. Now it is probably the most widely used common communications standard.

Internationally, the CEOS Data Interchange Format (DIF) was endorsed by the members of Committee for Earth Observing Satellites (CEOS). This is a standard for structuring catalog system directory level description data to allow international exchange of high-level metadata information to facilitate access to data held in international centers.

Because of the magnitude of future data systems and the recognition that high-quality, long-term data sets are essential for research, embedded documentation and portability standards are deemed to be absolutely essential. Embedded documentation ensures that long-term data can be identified in future applications. Portability standards such as OSI protocol standards ensure that data can be interchanged in future applications.

However, over-documentation can have deleterious effects for users of data. For example, packet technologies, which are widely applied for telecommunicating data, present a data management burden when very large data sets are used, as breaking the data into many small packets creates overhead which impedes efficient handling of the data. An example of this is the GRIB data communications standard for transmitting point data values. This transmission standard grew out of the teletype era when line noise and bandwidth limitations

required a high level of data "bracketing" to reduce the loss of data through garbling and drop outs. The Global Telecommunications System is totally dependent on this standard. Data sets archived in this format are highly inefficient for modern computer processing approaches.

4.2. MEDIA

The media used to store data for future reference and, in some cases permanently, is a critical factor to consider in data management. Today, the advances made in the development of sophisticated media capabilities coupled with an ever increasing variety of media technologies have expanded the options to consider in managing data. In the past, the choices were manuscript, paper computer output, or film, all which had known life cycle and risk factor. Now a variety magnetic and optical tape and disks are among the current computer-form choices, and perhaps in the future, crystal and molecular storage systems will expand this list. The never ending development of media technologies has created a dilemma for the data manager who must now determine which media best suits the user requirements for both the present and future. Issues of cross system compatibility, future system portability, and expected media life have equal weight in the decision process.

For the data production manager, the media which can keep pace with the data flow and, at the same time, is the most cost effective, is the media of choice. For the scientist, it is the media which holds the most data on-line for convenient and efficient access. For the archivist, it is the media which contains the most data in a given unit volume, and which has the longest expected life when stored in a passive state. Often these requirements are in conflict with each other. Then, what is the solution? For some data sets, particularly small volume data sets, employing multiple types of media would be acceptable. However, other strategies are necessary for large volume data sets where redundant storage forms would either be unmanageable or too costly. These conflicting needs force the archive manager to plan to migrate between media types.

Emerging media technologies present an elevated risk to data, as these technologies often do not have a sufficient demonstrated performance record to fully predict longevity characteristics. Also, many technologies emerge without standards creating havoc with portability and system interface requirements. An example of this is the implementation of optical disk systems of which there are many form factors. After over five years of optical systems availability, only the CD-ROM has a standard (ISO 9660), and this process was leveraged primarily by the consumer audio CD market. The magneto optical (MO) recording system implementers have never developed standards and the use of these systems elevates the risk to the data because of the market place volatility and because of the virtual absence of recording standards. The higher capacity 12 and 14 inch write one, read only (WORM) optical media are relatively high cost systems, and like the MO recording systems, have no established standards, which inhibit portability and elevate the risk to the data in an archive environment.

Another concern is the mechanical performance characteristics of media. Today, probably the most stable archive media available is film. This, however, is not suitable for rapid access and deployment of data used in computer applications. The development of magnetic media continues to advance as compared to the optical media forms which so far continue to emerge. However, magnetic media technologies present some risk factors for data longevity. The earlier open reel tape technologies were long considered stable for archive purposes, but not without mechanical management to prevent material deformation and magnetic read through. The introduction of the IBM 3480 cartridge tape eliminated many of the earlier tape management problems by improving the system mechanical handling properties and error correction code applications. However, standard computer tapes are capacity limited and are not suitable for some future ultra-high volume data observation systems.

The helical scan recording technologies adopted from the video recording industry offer the necessary recording rate and capacity to meet high rate and volume requirements. These

media have physical characteristics which introduce new risk factors to data intended for long-term retention. For example, the magnetic recording surface is an unoxidized pigment which may be susceptible to corrosive pollutants. The substrate used as the backing is stretched to remove its elasticity (tensilized) for rigid tape to recording head control and to increase the run length of tape in a small cassette. The tensilized polyester substrate is predicted to have a tendency to relax or shrink over time affecting the head tracking servo and increasing the risk to successful data recovery. And, these conditions are likely to be accelerated by stress induced with excessive heat and humidity.

Until the known risk factors with magnetic media are mitigated, the data manager must develop tools to carefully monitor the media recording validity, track the media performance over time, and exercise rigid environmental storage controls to minimize these known stress factors. However, the most important element in managing data in an archive environment is to include resources to allow for periodic data migration to avoid the risk of losing data due to mechanical failures or system obsolescence, which alone is a risk common to all systems.

However, data longevity is not the only factor to consider. Data access is an equally important factor.

5.0. PROVIDING ACCESS TO ENVIRONMENTAL DATA

The quality of the data and its documentation and the preservation of these data in a lasting state have little value if access to the data and information is hindered by loss of logistics control or excessive recovery costs. A sophistical indexing scheme is necessary for locating data and information in the future. As the collection of data continues, redoubling the problem of storing and finding the data, the level of index sophistication must necessarily increase to facilitate an increasing degree of automation in the search process For example, to research a publication today in most libraries, one must know either the author or title. Some levels of "key word" searching are being made available but on a limited basis. The largest public library in this country is the New York library. It is estimated that its entire collection if digitized would represent a thousand trillion bits, or one petabit. The NASA EOS program in just 15 years will produce over 20 petabits of data. Without a very high degree of automatic indexing across many science disciplines, the utility of these diverse observation and research data sets will be greatly diminished.

However, once indexed, the researcher is faced with the problem of porting the data from an archive system for processing. There are many problems to address here. How will the data be packaged? If the data set is very large, and most will be after a long period of time, how can subsets of the data and combinations of different data be repackaged in a portable, useful, and affordable tool?

Our data problem today is akin to the comparison of the early country store and the modern supermarket [1]. The country store survived in an era when the variety of products and packaging was simple. There, the customer would order across a counter and the access service would be provided by a clerk. The clerk was the data base directory and inventory and the user's lexicon was simplified by the limited variety of available products. Imagine trying to shop in a modern supermarket using that type of access environment. Most of today's data bases put the user in that position, as there are only rare instances of browse data and the available lexicon matches are limited to high order directory services. From a researcher's point of view, modern data bases should be like the super market, where one can browse through the aisles, view the variety of products for selection, as well as read the labels containing the technical specifications of the products.

In the world of data and information, an information directory alone does not provide sufficient information for a user to find and ultimately use data. The next logical level for a user to search is the inventory which describes the physical arrangement of data. However, an ability to look at or visualize the data would increase the level of understanding of the data.

Thus, a browse data set becomes increasingly important to the user. In fact, the level of importance of browse utility is likely to increase as the data ages.

If this hierarchy of searching is complicated enough for a single data set, what about other related data sets held possibly in the same or other archive centers? This opens up another aspect of browsing through the shelves of the super market. In the super market, all of the similar products are on adjacent shelves. The catsup, mustard, mayonnaise, etc., are usually together, the baking stuff is usually grouped, etc. Another feature of food marketing is that the snacks, such as chips, are almost everywhere. The same should hold for data. The variety of data sets within a discipline should be linked at the inventory level and the ancillary in-situ data, i.e., chips of the data world, should be more widely cross linked across data set groups. Searching for data should be as simple as searching through the super market shelves where everything in the store can be felt, read and compared by the user. For the users who know exactly what they want, they can simply proceed to the appropriate aisle and select the product and leave.

But what happens when today's super market becomes a megamarket as the data world is fast experiencing? Could one afford to stroll the aisles browsing through unimaginable varieties of products? What if the products were packaged in pallet sized boxes? Could one lug the package home and shelve it there? That is what the data world is beginning to experience. In order to survive this environment, one will need to have a better sense of where to look and must be able to apportion manageable amounts of data for local consumption. In the data management world, knowledge based help tools will be necessary to assist the user, and the user must have the capability to extract pieces of data for local consumption.

Once we find the data, how do we get it home to use? In the super market scenario, if you are walking, you should be careful to purchase only what you can carry, or, if you are driving, then you are limited by the capacity of your vehicle. In either case, you should know your limitations before you buy. In the world of data, aside from the cost, your limitations are the bandwidth you can afford both in terms of electronic transfer bandwidth and the media compatibility. This is where media portability, through both mechanical and applied standards, becomes an important issue. Another important aspect not to be overlooked is the efficiency of the media for processing.

6.0. CONCLUSION

The media used for recording and storing data is only one aspect of data management. Finding and acquiring data is the other. However, the life and vitality of the data are dependent on the media capsule, and the lack of care and handling here determines the ultimate future of the data. Thus, all the efforts to develop robust data management search and select tools can only be secondary to the media technologies used to convey the data in storage and use.

REFERENCES

[1] R. Jenne, NCAR, A National Grocery Store System, April 3, 1991

INTERIM REPORT ON LANDSAT NATIONAL ARCHIVE ACTIVITIES

John E. Boyd
U.S. Geological Survey
EROS Data Center, Sioux Falls, SD 57198

ABSTRACT

The Department of the Interior (DOI) has the responsibility to preserve and to distribute most Landsat Thematic Mapper (TM) and Multispectral Scanner (MSS) data that have been acquired by the five Landsat satellites operational since July 1972. Data that are still covered by exclusive marketing rights, which were granted by the U.S. Government to the commercial Landsat operator, cannot be distributed by the DOI. As the designate national archive for Landsat data, the U.S. Geological Survey's EROS Data Center (EDC) has initiated two new programs to protect and make available any of the 625,000 MSS scenes currently archived and the 200,000 TM scenes to be archived at EDC by 1995.

A specially configured system has begun converting Landsat MSS data from obsolete high-density tapes (HDTs) to more dense digital cassette tapes. After transcription, continuous satellite swaths are (1) divided into standard scenes defined by a world reference system, (2) geographically located by latitude and longitude, and (3) assessed for overall quality. Digital browse images are created by subsampling the full-resolution swaths. Conversion of the TM HDTs will begin in the fourth quarter of 1992 and will be conducted concurrently with MSS conversion. Although the TM archive is three times larger than the entire MSS archive, conversion of data from both sensor systems and consolidation of the entire Landsat archive at EDC will be completed by the end of 1994.

Some MSS HDTs have deteriorated, primarily as a result of hydrolysis of the pigment binder. Based on a small sample of the 11 terabytes of post-1978 MSS data and the 41 terabytes of TM data to be converted, it appears that to date, less than 2 percent of the data have been lost. The data loss occurs within small portions of some scenes; few scenes are lost entirely. Approximately 10,000 pre-1979 MSS HDTs have deteriorated to such an extent, as a result of hydrolysis, that the data cannot be recovered without special treatment of the tapes. An independent consulting division of a major tape manufacturer has analyzed affected tapes and is confident that restorative procedures can be applied to the HDTs to permit one pass to reproduce the data on another recording media.

A system to distribute minimally processed Landsat data will be procured in 1992 and will be operational by mid-1994. Any TM or MSS data in the national archive that are not restricted by exclusive marketing rights will be reproduced directly from the archive media onto user-specified computer-compatible media . TM data will be produced either at a raw level (radiometrically and geometrically uncorrected) or at an intermediate level (radiometrically corrected and geometrically indexed). MSS data will be produced to an intermediate level or to a fully corrected level (radiometrically corrected and geometrically transformed to an Oblique Mercator projection). The system will be capable of providing ordered scenes within 48 hours of receipt of order.

MR-CDF: Managing Multi-Resolution Scientific Data

Kenneth Salem
Computer Science Department, University of Maryland
College Park, MD 20742
and
CESDIS, NASA Goddard Space Flight Center, Code 930.5
Greenbelt, MD 20771

Abstract

MR-CDF is a system for managing multi-resolution scientific data sets. It is an extension of the popular CDF (Common Data Format) system. MR-CDF provides a simple functional interface to client programs for storage and retrieval of data. Data is stored so that low-resolution versions of the data can be provided quickly. Higher resolutions are also available, but not as quickly. By managing data with MR-CDF, an application can be relieved of the low-level details of data management, and can easily trade data resolution for improved access time.

1 Introduction

Scientific data management libraries, such as NASA's publicly-distributed Common Data Format (CDF)[Tr90,TrGo90], implement simple data models that are tailored for scientific data. Data managed using these libraries is machine-independent, portable, and self-describing. Access to the data is performed through a set of interface functions that shield the details of storage and retrieval from application programs. The libraries provide a common interface upon which portable, application-specific tools (e.g., classifiers, analysis packages, visualization and browsing tools) can be implemented.

Because scientific data sets are often voluminous, it is desirable to make them available at several different resolutions. Preliminary examination, or *browsing*, of large amounts of data often can be performed efficiently using low-resolution data. Tentative analyses can be performed using intermediate-resolutions, and the final analysis can be performed using the data's full resolution. Lower resolution data is desirable during the preliminary stages because it allows large volumes of data to be considered in a reasonable amount of time.

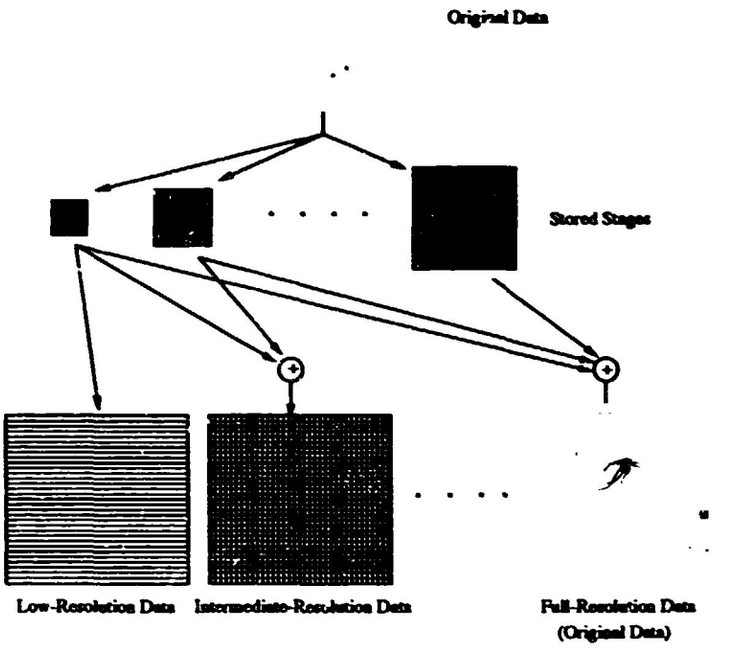


Figure 1: The Multi-Stage Representation Used by MR-CDF

This paper describes a scientific data management library called MR-CDF (Multi-Resolution Common Data Format) which permits multiple-resolution data sets to be manipulated through a simple, functional interface. Application programs that use MR-CDF see a scientific data model identical to that supported by CDF. When retrieving data, however, they are able to specify a desired resolution level. Applications requiring full-resolution data can obtain it, while those that can use lower resolutions are be able to do so simply and quickly.

MR-CDF uses a *multi-stage* representation for stored multi-resolution data. This is illustrated in Figure 1. A data set that is to be made available at R different resolutions is decomposed into R *stages*, each of which is stored. The decomposition is such that, by retrieving and combining i stages MR-CDF can produce the data at one resolution, and by retrieving $i+1$ stages it can produce the data at a higher resolution. By retrieving and combining *all* of the stages, MR-CDF can produce an exact reconstruction of the data at its original, full resolution. The process of retrieving and combining stages is completely transparent to the application that requested the data, except that lower resolution requests can be satisfied more quickly than others.

There are several difficulties involved in providing an abstract interface for multi-resolution data. The first is the wide variety of techniques that can be used to decompose data into stages

for storage. As we shall describe shortly, the decomposition process is essentially an iterative lossy compression of the data. A wide variety of compression techniques are available, and different techniques are well-suited to different types of data. Examples include various region averaging algorithms, vector quantization, and quadtree-like methods [TiMa90, Ti89]. The procedure for properly recombining the stages when data is retrieved depends on which of the many possible compression techniques was originally used to decompose the data. Tying MR-CDF to any particular compression technique would severely limit its applicability. Instead, MR-CDF must be flexible enough to accommodate a wide variety of application-specified techniques.

A second difficulty arises when applications make use of MR-CDF's simple selection facility to retrieve only a portion of the stored data set. Ideally, MR-CDF would perform the selection *before* recombining the stages to minimize the volume of data to be retrieved and recombined. This may or may not be possible, depending on which technique was used to produce the stored stages. Some compression techniques are better suited than others for producing easily-manageable data, at least within the framework of MR-CDF. Although such problems need not limit the functionality of MR-CDF, they may impact its efficiency.

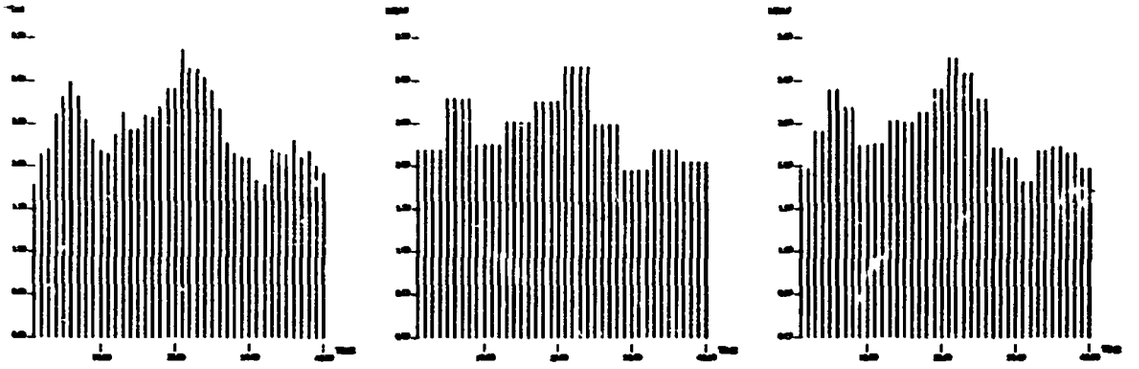
In the remainder of this paper, we describe the design, interface, and implementation of the MR-CDF library. The next section provides an overview of the features of MR-CDF. Sections 3 and 4 describe the relationship between data compression and MR-CDF, and how multi-resolution data is stored into and retrieved from an MR-CDF archive. Finally, Section 5 describes its implementation, which uses CDF's data storage and retrieval facilities.

2 What Does MR-CDF Do?

The MR-CDF library does not attempt to provide a solution to the entire scientific data management problem. An MR-CDF archive is designed to hold a set of related, similarly organized scientific data, such as a set of images or a stream of sensor data. The important task of organizing and managing multiple data sets is left to some type of meta-database, such as those described in [RoCa90, ShWa88], and is beyond the scope of MR-CDF (and CDF).

What MR-CDF *does* provide is a simple, abstract programming language interface to scientific data. MR-CDF extends the CDF interface to provide support for multi-resolution data sets. It has all of the capabilities of CDF for storage, retrieval, and organization of data, plus the following:

- MR-CDF allows selected data to be retrieved several different levels of resolution. Lower-resolution data can be retrieved more quickly than higher-resolution, allowing applications



(a) Full Resolution (b) Low Resolution (c) Medium Resolution

Figure 2: A Simple Time-Series Data Set

to trade-off retrieval time for resolution.

- Multi-resolution retrieval in MR-CDF is *progressive*. This means that once a low resolution version of the data has been retrieved, a higher resolution version of the same data can be retrieved in less time than would be required to retrieve the higher resolution data from scratch.

The multi-resolution capability of MR-CDF makes it simple for application programs to select a resolution that is suitable for the task at hand. Progressive retrieval is well-suited to applications such as data browsing. For example, an image browsing program can provide access to many low-resolution images quickly. When an interesting image is found, a progressive retrieval capability allows the browser to provide a higher-resolution version of the interesting image without retrieving the information contained in the low-resolution image a second time.

2.1 Multi-Resolution Data

Figure 2(a) shows a plot of some time-series data representing a hypothetical measured quantity "MEAS". Data of this type might be stored in an MR-CDF archive. For the purposes of this example, suppose that the MEAS variables is of the MR-CDF-defined floating-point type "REAL-4". We will use this example to describe how multi-resolution data in MR-CDF is viewed by application programs.

When a multi-resolution variable is created in an MR-CDF archive, the number of resolutions at which it can be made available is defined. Applications retrieve the values of multi-resolution variables exactly as they would a single-resolution variable, except the desired *resolution level* must be specified as well. A resolution level is specified as an integer between zero and the the

number of resolutions defined for that variable, minus one. Smaller numbers represent lower resolutions.

Suppose that "MEAS" is stored as a variable with three possible resolutions. If an application retrieved "MEAS" at resolution zero, it might receive the data plotted in Figure 2(b). Data at resolution one might look as plotted in Figure 2(c), while the data at resolution two would match the full-resolution data in Figure 2(a) exactly.

An important feature of MR-CDF is that the application will receive the same volume and type of data, regardless of the resolution level requested. In our example, the application can expect to receive ten REAL-4 values, regardless of resolution. This greatly simplifies data handling in MR-CDF applications. From the application's perspective, the advantage of lower resolution data is that MR-CDF can provide it more quickly. Although the volume of data passed to the application is independent of the resolution, MR-CDF needs to retrieve less data from its archive to produce the lower resolutions.

3 Producing Data for MR-CDF

The low resolution data in Figure 2(b) were obtained by averaging groups of four values from the original series (Figure 2(a)) and replacing the values in each group by their average. This averaging procedure is a form of *lossy data compression*, since the low resolution series can be represented using a quarter of the values required for the original series. MR-CDF is specifically designed to manage multi-resolution data that is produced by applying lossy compression to the full-resolution data. Of course, there are many more effective and sophisticated compression techniques than the averaging procedure used in the example.

Data compression is not performed by the MR-CDF library. Instead, it is assumed that the compressed data is produced externally and then stored in the MR-CDF archive. MR-CDF performs the decompression and combination of the stored data in response to application requests.

In principle, it would be possible for compression to be implemented within MR-CDF. In practice, however, compression of the data is often much more time consuming than decompression. Many compression algorithms are best performed on highly parallel machines or with special purpose hardware. [1]

[1] For example, compressing data using vector quantization involves vectorizing the input data and comparing each vector against a "codebook" of vectors to find the closest match. Using parallel hardware, the input vector can be compared against all of the codebook entries simultaneously. Decompressing the data involves much less work, since only a simple lookup in the codebook is all that is required to recreate each vector.

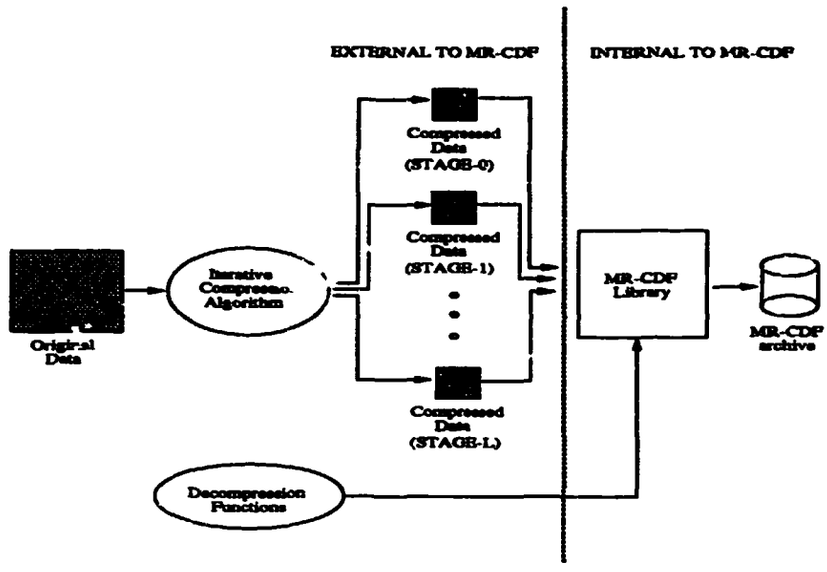


Figure 3: Creating Data for MR-CDF

Figure 3 illustrates how data suitable for progressive, multi-resolution retrieval is produced and stored in MR-CDF. An iterative compression technique (described below) is applied to the original full-resolution data, resulting in several stages of compressed data. The compressed data are stored in the MR-CDF archive. The stages are such that MR-CDF will be able to recreate the data at resolution level i by retrieving stages 0 through $i-1$ from the archive and then decompressing and recombining them. We will describe the retrieval procedure in more detail shortly.

The iterative compression algorithm shown in the figure actually represents a general class of compression procedures. During each iteration, data is compressed using *some* lossy compression technique, and then decompressed. The difference between the decompressed data and the original is computed. This difference, or error, becomes the data that is compressed during the next iteration.

The iterative compression procedure for computing three stages of compressed data is illustrated in more detail in Figure 4. As illustrated, a lossy compression function f_i is used to produce the stage-i data. Since the compression function is lossy, the decompressed data will not match the original data exactly. The difference between the original data D_0 and the decompressed data $f_i'(f_i(D_i))$ is the error (residual) data, which is used as the input to the next decompression stage. In the figure, the shaded boxes represent the compressed data stages which are actually stored in MR-CDF.

The example in Figure 4 may be somewhat misleading since it suggests that the compressed

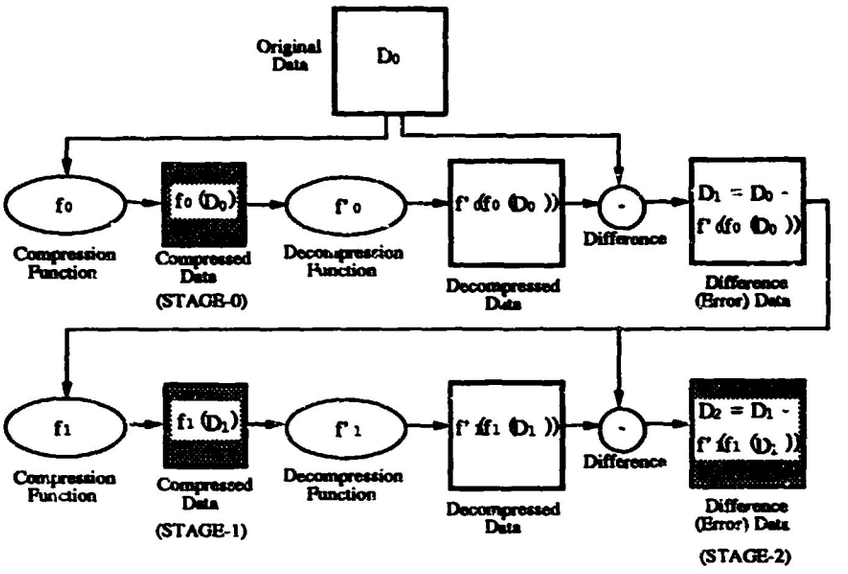


Figure 4: Iterative Compresson Procedure - Three Stages

data stages together occupy more space than does the original, full-resolution data set. In practice, this need not be the case. For more realistic compression techniques, such as those described in [TiMa90, Ti89], the stages taken together are about as voluminous as the original data. The compressed stages can be thought of as an alternative representation of the original data which makes multi-resolution retrieval more convenient.

4 Retrieving Data from MR-CDF

When an application requests data from MR-CDF, it specifies a desired resolution level. MR-CDF supplies the data at the specified resolution by retrieving one or more of the compressed data stages from the archive. To retrieve data a resolution i, stages 0 through i are retrieved. The retrieved stages are then decompressed and combined to produce the desired data.

Figure 5 illustrates how MR-CDF would handle a request to retrieve the data compressed as illustrated in Figure 4 at resolution level two. (In this case, resolution level two corresponds to the original, full-resolution data.) Since resolution level two is requested, MR-CDF retrieves the stage-0, stage-1, and stage-2 compressed data. The first two stages are decompressed, and the resulting data is additively merged into a single buffer. In this case, the buffer will contain an exact recreation of the original data D_0.

If a lower resolution level is specified, MR-CDF need only retrieve and decompress some of the stages. For example, for resolution level 0, only the stage-0 data is retrieved and decompressed.

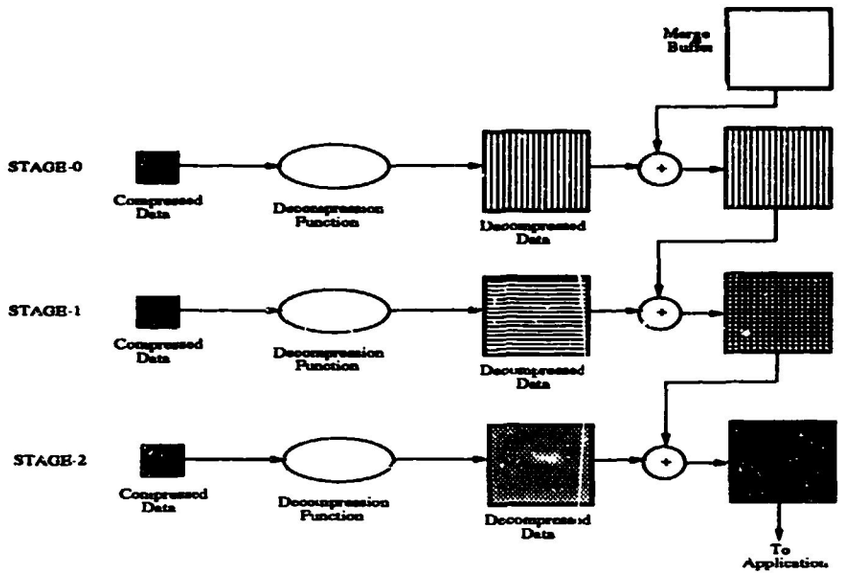


Figure 5: Decompression Procedure - Three Stages

4.1 Decompression Functions

MR-CDF's retrieval procedure requires that a set of decompression functions be available. Since an arbitrary compression function can be used to produce the stages, MR-CDF must be informed of the proper decompression function to apply at the time of retrieval. When an application stores compressed data in MR-CDF, it is required to *register* an appropriate decompression function with the library, is was illustrated in Figure 3.

A decompression function is an arbitrary procedure which accepts a set of parameters supplied by MR-CDF. These parameters include pointers to the source buffer holding the compressed data and a target buffer into which the decompressed data is to be placed. Additional information, such as the sizes of the buffers and their data types is also provided.

Each time a new multi-resolution variable is defined in MR-CDF, the names of the decompression functions to be used for each compressed data stage must also be supplied. Every decompression function is registered under a particular name. New variables that use the same decompression functions as existing variables may refer to those functions by name.

A decompression function defines a mapping from compressed data to decompressed data. In many cases, it is most convenient to implement the function as a generic piece of code, plus some additional data. For example, vector-quantized data can be decompressed by a simple function which looks-up each code word in a codebook. Changing the codebook changes the decompression function that is being implemented, but the generic code itself need not be

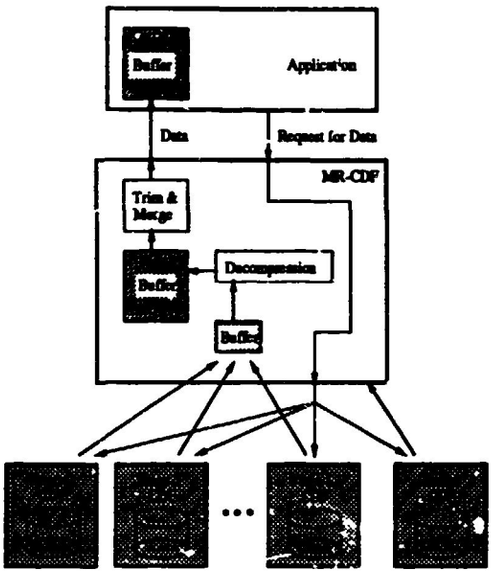


Figure 6: Implementing MR-CDF with CDF

modified.

Since this is a common occurrence, MR-CDF allows *auxiliary data* to be stored with each stage. At decompression time, both the compressed stage data and the auxiliary data are supplied to the decompression function. The advantage of auxiliary data is that common, generic decompression functions need only be registered once with MR-CDF.

5 Implementation

To an application, MR-CDF provides a superset of the services provided by CDF. MR-CDF is also *implemented* using CDF. All data storage and retrieval is performed by CDF. MR-CDF acts as a coordinator between a group of CDF archives and the application-specified decompression procedures.

Each MR-CDF archive is implemented as a set of CDF archives. Specifically, a MR-CDF archive is implemented by a single *base CDF* plus a set of *stage CDFs* for storing the compressed data stages. There is a stage CDF for each stage of every multi-resolution variable defined in the archive. This is illustrated (for an archive with a single multi-resolution variable) in Figure 6.

An MR-CDF archive may contain a mix of single-resolution and multi-resolution variables. Single-resolution variables are implemented directly in the base CDF. Requests to store and retrieve such variables are translated into appropriate CDF calls on the base CDF. In addition,

the base CDF maintains global information about the MR-CDF archive such as the number of defined variables and their names. It also maintains general information about the multi-resolution variables in the archive.

When MR-CDF receives a request to retrieve a multi-resolution variable, the following steps occur. MR-CDF first retrieves general information about the variable (such as the number of stages that are available and their sizes) from the base CDF. Using this information, MR-CDF then translates its request to a series of retrieval requests on the stage CDF's.

As data is retrieved from each stage, it is placed in a holding buffer and then passed through the appropriate decompression function. (Auxiliary decompression information is stored as attributes of the stage CDFs and is retrieved using the attribute/value manipulation facility provided by the CDF library.) The decompressed data is then trimmed and merged with data from the other stages in the merge buffer. To avoid unnecessary copying of data, the decompressed and trimmed stages are merged directly into the application's buffer.

MR-CDF runs on UNIX systems for which CDF is supported. Currently, only the C language interface is available. Since UNIX does not provide a run-time linking facility, it is not possible to define new decompression functions to the MR-CDF library without recompiling it. (New multi-resolution *variables* using existing decompression functions can be added at any time.) However, the procedure for adding new decompression functions is very simple.

6 Conclusion

MR-CDF provides an abstract interface to multi-resolution scientific data. Its program interface allows applications to define, store, select, and retrieve data. MR-CDF can make lower resolution data available quickly, allowing applications to trade off resolution for retrieval time.

MR-CDF is implemented using NASA's CDF (Common Data Format) library and runs on any UNIX system supported by CDF. Existing CDF applications can use MR-CDF with minimal modifications.

MR-CDF stores multi-resolution data as a series of compressed data stages which can be decompressed and combined to produce the data at different resolutions. Retrieval of compressed data introduces a tradeoff between I/O costs and processing costs. Compression reduces the volume of stored data, and therefore the I/O cost for its retrieval. However, the decompression and recombination of the data introduces processing overhead. Technological trends suggest such tradeoffs will become more beneficial with time. The performance of processors continues to improve rapidly, while access times for I/O devices have changed little.

Since CDF utilizes the UNIX file system, distributed operation of the MR-CDF library is possible among machines with access to a common file system, such as NFS. We are currently planning a distributed version of MR-CDF for systems which do not share files.

Acknowledgements

Thanks to M. Manohar for several helpful discussions, and for providing test data for MR-CDF.

References

[RoCa90] Roelofs, L. H., and W. J. Campbell, "Using Expert Systems to Implement a Semantic Data Model of a Large Mass Store System", *Telematics and Informatics*, 7, 3/4, 1990, pp. 361-377.

[ShWa88] Short, N., Jr., and S. L. Wattawa, "The Second Generation Intelligent User Interface for the Coastal Dynamics Data Information System", *Telematics and Informatics*, 5, 3, 1983, pp. 253-268.

[Tr90] Treinish, L., "The Role of Data Management in Discipline-Independent Data Visualization," *SPIE/SPSE Symposium on Electronic Imaging Science and Technology*, February, 1990.

[TrGo90] Treinish, L., and M. Gough, "A Software Package for the Data-Independent Management of Multidimensional Data,", *Eos*, 68, 28, July, 1987, pp. 633-635.

[TiMa90] Tilton, J. C., and M. Manohar, "Hierarchical Data Compression: Integrated Browse, Moderate Loss, and Lossless Levels of Data Compression," *Proc. International Geoscience and Remote Sensing Symposium*, May, 1990, pp. 1655-1658.

[Ti89] Tilton, J. C., "Image Segmentation by Iterative Parallel Region Growing and Splitting," *Proc. International Geoscience and Remote Sensing Symposium*, May, 1989, pp. 2420-2423.

N93-80457

High-Performance Mass Storage System for Workstations

by
T. Chiang, Y. Tang, L. Gupta, and S. Cooperman

Loral AeroSys
7373 Executive Place
Suite 101
Seabrook, Maryland 20706

ABSTRACT

Reduced Instruction Set Computer (RISC) workstations and Personnel Computers (PC) are very popular tools for office automation, command and control, scientific analysis, database management, and many other applications. However, when using Input/Output (I/O) intensive applications, the RISC workstations and PCs are often overburdened with the tasks of collecting, staging, storing and distributing data. Also, by using standard high-performance peripherals and storage devices, the I/O function can still be a common bottleneck process. Therefore, the high-performance mass storage system, developed by Loral AeroSys' Independent Research and Development (IR&D) engineers, can offload a RISC workstation of I/O related functions and provide high-performance I/O functions and external interfaces.

The high-performance mass storage system has the capabilities to ingest high-speed real-time data, perform signal or image processing, and stage, archive, and distribute the data. This mass storage system uses a hierarchical storage structure, thus reducing the total data storage cost, while maintaining high-I/O performance.

The high-performance mass storage system is a network of low-cost parallel processors and storage devices. The nodes in the network have special I/O functions such as: SCSI controller, Ethernet controller, gateway controller, RS232 controller, IEEE488 controller, and digital/analog converter. The nodes are interconnected through high-speed direct memory access links to form a network. The topology of the network is easily reconfigurable to maximize system throughput for various applications. This high-performance mass storage system takes advantage of a "busless" architecture for maximum expandability.

The mass storage system consists of magnetic disks, a WORM optical disk jukebox, and an 8-mm helical scan tape to form a hierarchical storage structure. Commonly used files are kept in the magnetic disk for fast retrieval. The optical disks are used as archive media, and the tapes are used as backup media. The storage system is managed by the IEEE mass storage reference model-based UniTree software package. UniTree software will keep track of all files in the system, will automatically migrate the lesser used files to archive media, and will stage the files when needed by the system. The user can access the files without knowledge of their physical location.

The high-performance mass storage system developed by Loral AeroSys will significantly boost the system I/O performance and reduce the overall data storage cost. This storage system provides a highly flexible and cost-effective architecture for a variety of applications (e.g., real-time data acquisition with a signal and image processing requirement, long-term data archiving and distribution, and image analysis and enhancement).

1. INTRODUCTION

RISC workstations and PCs are frequently used for office automation, command and control, scientific analysis, database management and many other applications. However, RISC workstations and PCs usually suffer the following drawbacks:

1.1 Lack of Cost Effective High-Performance Storage Capabilities

Due to the increase of add-on board applications for workstations or PCs, the demands for cost effective high-performance sto[illegible] capabilities also increase. As a result, the hierarchical storage architecture becomes n[illegible]ssary for many workstation and PC applications.

1.2 Lack of Data Acquisition Capabilities

For some applications, data is received simultaneously from multiple sources through different I/O controllers. Standalone workstations or PCs usually have limited I/O controllers, ports, and I/O bandwidth that can handle large volume composite data streams. Also, a workstation having different parallel I/O controllers can be a critical issue.

1.3 Lack of Processing Power for Computing Intensive Applications

RISC workstations and PCs used as general purpose computers usually are inefficient for such intensive computing applications as numerical analysis, image processing, and signal processing. For example, data compression, image enhancement and data formatting all require extensive computing power. Therefore, additional computing power can be useful for some workstation and PC applications.

2. OBJECTIVE

The objective of this research is to develop a cost-effective mass storage system prototype that will provide cost-effective, unlimited storage space for the workstations and PCs and offload data acquisition, storage management, and intensive computing functions from the workstations and PCs.

3. APPROACH

The high-performance mass storage system prototype consists of modular, interchangeable hardware and software building blocks. The system's building blocks are developed using Commercial-Off-the-Shelf (COTS) products where possible. This system's prototype is implemented in an open environment, using a Unix operating system and X-windows. This structure is an optimum solution for a multi-vendor environment. The hierarchical storage structure is used to provide cost-effective storage media; and the massive parallel processing system is used to perform the scalable I/O and data processing capabilities. Loral AeroSys' mass storage system design can off load I/O and intensive computing functions form workstations and PCs.

3.1 Hierarchical Storage Structure

Loral AeroSys' mass storage system prototype provides automatic migration based on user-supplied criteria. The storage management software is designed to track the physical locations of files, which is transparent to the user. The migration criteria can be adjusted to provide an efficient and cost effective solution to a specific application

3.2 Massive Parallel Processing System

The Multi-Instruction Multi-Data (MIMD) parallel system, which provides scalable processing power, is used to perform the storage management, data acquisition and other computing intensive functions for the high-performance mass storage system prototype. Loral AeroSys' system prototype consists of arrays of I/O controllers and processors; and it can receive, process, store, retrieve, and distribute data streams in parallel to achieving maximum performance.

The mass storage system can be configured to be a network server providing services to all the workstations and PCs on the network, or a dedicated I/O processor for one workstation or PC through a dedicated link to receive a faster archive rate.

4. MASS STORAGE SYSTEM PROTOTYPE

Loral AeroSys is currently building a mass storage system prototype based on the previously mentioned design approach. The high-performance mass storage system building block configuration is shown in Figure 1.

4.1 Hierarchical Storage Media Configuration

Frequently used files are stored in a disk array of three magnetic disks of 600 Megabytes each. The disk array is connected to the parallel processor through an SCSI bus. The archive files are stored in an optical jukebox with two WORM drives. The jukebox can house up to 25 (5.25") platters with a total storage capacity of 16 Gbytes. When the jukebox is full, the platter can be manually moved offline, and the prototype can still track the files. An 8-mm helical scan tape system is used to provide a system log, backup files, and distribute files. The WORM drives and tape drive are connected to a second SCSI bus.

4.2 Massive Parallel Processors

The Parsytec multicluster parallel processor system is used to host the mass storage devices and to provide parallel processing capability. Loral AeroSys' system prototype has a low-cost Inmos T800 transputer chip, rated at 25 MIPs. Four processor boards are used for the initial mass storage system configuration. The four boards are: a root processor board with 32 Mbyte memory, two SCSI controller boards with four Mbyte memory each, and Ethernet board with 2 Mbyte memory, and an RS232 daughter board. Each board has one T800 chip and four high-speed links to form a parallel processor network. The Parsytec Multicluster system runs a Unix-compatible operating system (Helios).

The Parsytec Multicluster system is also hosted by a 386 PC through an interface board. The PC provides the user interface to the Parsytec Multicluster system. The PC runs MS-DOS operating system.

UnfTree software is used to manage the files and storage media. UnfTree is implemented based on the IEEE mass storage reference model; it maintains a standard Unix-style directory structure, and provides automatic migration and network access functions. A Cygnet Jukebox Interface Management Software (JIMS) was integrated into the UnfTree to provide WORM platter mount and unmount functions.

5. PROGRESS OF THE PROTOTYPE DEVELOPMENT

The integration of storage devices to the parallel processor network, and the porting of JIMS to the parallel processor network are completed.

During porting efforts, some major problems were encountered. The most serious problem was analyzing the difference between the different Unix operating systems (e.g., System V inter-processor communication package used by JIMS, and Smile task scheduling package used by UnfTree). System V and UnfTree Smile are not supported by the Helios operating system used by parallel processing system. The solution was to implement the System V capabilities and port the UnfTree Smile package to the parallel processor. Currently, all major problems are resolved.

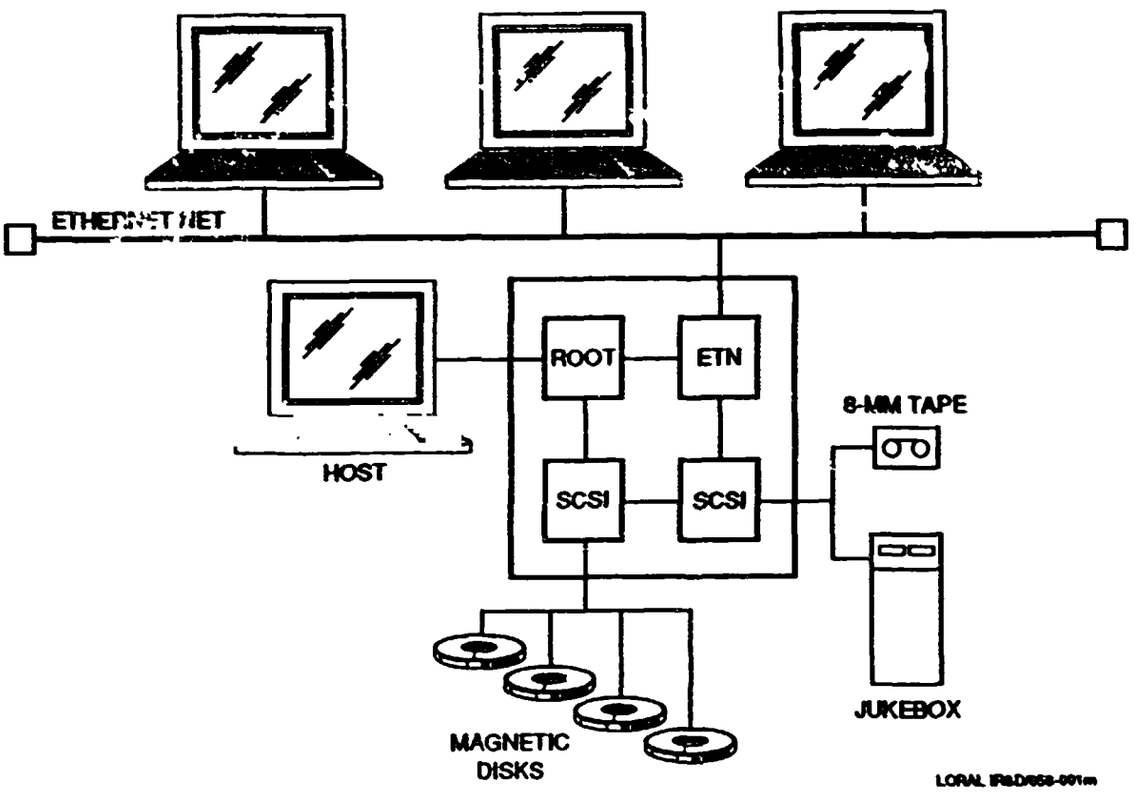


Figure 1. Mass Storage System Prototype

The success of integrating storage devices and porting of JIMS indicates that it is feasible to build a cost-effective parallel I/O and data processing mass storage system for workstations and PCs. Plans are in place to perform benchmark testings for different applications and to add rewritable optical jukeboxes and 8-mm tape library system.

6. CONCLUSION

Loral AeroSys IR&D's mass storage system prototype is a high-performance, flexible storage management system using an inexpensive implementation of the UniTree file. This mass storage system provides hierarchical file and storage management for networked, multi-vendor computer environments, and provides complete application transparency in an open environment.

N93-30458

GE Networked Mass Storage Solutions Supporting IEEE Network Mass Storage Model

Donald Herzog
GE Aerospace
Government Communications Systems Department
Camden, NJ 08102

SY-82
15-7
P-3

DuraStore Mass Storage Sub-System

The General Electric Government Communications Systems Department (GE/GCSD) has developed a near real time digital data storage and retrieval system that extends the capabilities currently available in today's marketplace. This system called DuraStore uses commercially available rotary tape drive technology with ANSI/IEEE standards for automated magnetic tape based data storage. It uses a non proprietary approach to satisfy a wide range of data rates and storage capabilities requirements and is compliant with the IEEE Network Storage Model.

Rotary Tape Drives (RTD)

The basic element of the system is the GE Rotary Magnetic Tape Drive (RTD) family of drives. The drives use 19mm helical scan technology and implement both the ANSI ID-1 standard for instrumentation data recording techniques with a BER of 10E-10, and the ANSI DD-1 standard for storing and retrieving computer compatible data with a BER of 10E-13. The drives operate in both streaming and asynchronous modes and are capable of handling ID-1 and DD-1 data streams automatically within the same drive.

Standard Interfaces

The drives are designed to be controlled using the ANSI Intelligent Peripheral Interface (IPI-3) command set. Each drive has two interfaces to the user; one interface (low speed) is for set-up/control, the other (high speed) for actual data transmission/reception and redundant command control. Currently GE has implemented the following interfaces:

A. Data Interface

1. Physical

   - HIPPI
   - SCSI

2. Control

   - IPI-3 Command Set

B. Control Interface

1. Physical
   - Ethernet

2. Network
   - TCP-IP

3. Control
   - IPI-3 Command Set

Application specific Hardware/Software

The heart of the drive controllers are the internal buffer management hardware and the IPI-3 command processing software. The drive controller has a user side that is easily modifiable at the factory to the desired data interface (HIPPI, SCSI, FDDI, IPI, RTD, etc.). All the drives in the RTD family are upward compatible and the current maximum continuous throughput of the top of the line RTD-45 is 50 MBytes/sec streaming ID-1 mode and 45 MBytes/sec asynchronous in the DD-1 mode. The drive controller can handle burst rates of 70 MBytes/sec and has a maximum buffering capacity of 4 GBytes.

Networked Automated Tape Libraries

Another element of GE's DuraStore system are GE's Data Storage System (DSS) Automated Tape Libraries (ATL). These libraries are designed to relieve the user/host file manager of the physical resource management responsibilities for the library. Each library complex is controlled by an Automated Tape Library controller. The library can be treated as a single logical device by the user/host. The Library Controller is capable of controlling up to four Automated Tape Libraries simultaneously. Users communicate to the Library Controller via Ethernet/TCP-IP using the same IPI-3 command set used to control the individual RTD drives. The Library Controller maintains all the directory information necessary to translate a file request from a user to the correct tape cassette, then locate the cassette in its appropriate bin, load and position the tape in an available drive and notify the user that his data is available to be read/written. The ATL supports mixtures of ID-1 and DD-1 volumes in the Library.

Library Administrator

The Library Administrator's interface to the ATL systems is through the Library Controller with a user friendly graphic display. In addition to the read/write functions necessary for data storage/data retrieval, the Automated Tape Libraries support the following additional functionality available to the Library Administrator and to system users:

1. Import - Enter Media and Files into Automated Tape Library
2. Export - Remove Media and Files from ATL
3. Directory - Volume/File Listings for Library/Volumes
4. Error Statistics -
    a. BER for Drives/Trend Analysis
    b. BER for Volumes/Trend Analysis
5. Full Directory Shadowing
6. Library Diagnostics
7. Resource Management

Write Protection

The Automated Tape Library also allows the user to protect his individual files/volumes. This protection is done at the volume and not the user level. Each individual user can select one of three levels of write protection:

1. Write protection on entire volume - No more data can be written to that volume

2. Write protect existing data only - Data is write protected as it is added to volume

3. No write protection - Overwriting allowed

Volume/Physical Media Linkages

The GE Automated Library supports/allows all the logical volume/physical media linkages supported by DD-1:

- 1 Volume to 1 Cassette
- Multiple Volumes to 1 Cassette
- 1 Volume to Multiple Sequential Cassettes
- 1 Volume to Multiple Parallel Cassettes (Striping)

Maximum Resource Utilization

The Library has been designed to maximize resource utilization. It monitors activity and will deallocate/reallocate assigned equipment if no activity takes place for prolonged periods of time. The Library will also automatically exercise various system diagnostics when errors tape place and automatically notify the Library Administrator of actions required to fix/further isolate problems.

The Automated Tape Libraries can support the following maximum configuration capabilities:

I 5 RTDs per Library
II 660 Medium Cassettes (40 GBytes/cassette each) per Library
III 5 million files in the data base
IV 25 TeraBytes of data/Library

The Library Controller, as a controller for 4 ATLs, will control:

a. 20 RTDs max
b. 20 million files in DB
c. 100 TeraBytes of data across 4 Libraries

N93-30459

HIGH-SPEED DATA DUPLICATION/DATA DISTRIBUTION - AN ADJUNCT TO THE MASS STORAGE EQUATION

Kevin Howard
EXABYTE
1685 38th Street
Boulder, CO 80301

Introduction:

The term "mass storage" invokes the image of large on-site disk and tape farms which contain huge quantities of low- to medium- access data. Although the cost of such bulk storage is recognized, the cost of the bulk distribution of this data rarely is given much attention. Mass data distribution becomes an even more acute problem if the bulk data is part of a national or international system. If the bulk data distribution is to travel from one large data center to another large data center then fiber-optic cables or the use of satellite channels is feasible. However, if the distribution must be disseminated from a central site to a number of much smaller, and, perhaps varying sites, then cost prohibits the use of fiber-optic cable or satellite communication. Given these cost constraints much of the bulk distribution of data will continue to be disseminated via inexpensive magnetic tape using the various next day postal service options.

For non-transmitted bulk data, our working hypotheses are that the desired duplication efficiency of the total bulk data should be established before selecting any particular data duplication system; and, that the data duplication algorithm should be determined before any bulk data duplication method is selected.

Building the Tools:

In order to compare data duplication hardware and various data duplication algorithms one must first build a suite of evaluation tools. There are several parameters required to build such a tool suite. They are:

- Burst Transfer Rate.
- Sustained Transfer Rate.
- Average Pick and Place Transport Velocity.
- Average Pick and Place Time.
- Load/Unload Time.
- Average Number of Megabytes Duplicated.
- Number of Pick and Place Mechanisms.

The *Burst Transfer Rate* is how fast data can be moved into drives on board memory. The *Sustained Transfer Rate* is how fast data can be transferred from on board memory to the media.

To compute the *Average Pick and Place Time* for a given piece of data duplication hardware we can use the following equation:

(equation 1)

$$PT = \frac{\sum_{m=1}^{X} \sum_{n=1}^{Y} (|C_m - D_n| / V_{avg})}{XYP_p}$$

where:

APPT	=	Average Pick and Place Time
C_m	=	Location of cartridge number m
D_n	=	Location of Driver number n
V_{avg}	=	Average velocity of the pick and place mechanism
X	=	Total number of cartridges
Y	=	Total number of drives
P_p	=	Total number of pick/place devices

In order to compute the *Average Load and Unload Time* one can use the following equation:

(equation 2)

$$APPT = \frac{\sum_{m=1}^{X} (L_m + U_m) / 2}{X}$$

where:

ALUT	=	Average Load and Unload Time
L_m	=	Load time for a particular drive

U_m = Unload time for a particular drive

X = Total number of drives

The Load Time is the total amount of time it takes after inserting media before it can be read from or written to. The unload time is the total amount of time it takes before the media can be removed after it is requested.

To duplicate at the maximum speed one must minimize the time not spent writing the data; that is, minimize the time loading and unloading, as well as picking and placing the cartridges. Minimizing the load/unload time can most easily be accomplished by duplicating the largest possible files. The relationship between the total duplication time, load/unload , pick and place time, and the time spent actually writing data is given below:

(equation 3)

Total time = (load/unload time) + (pick/place time) + (write time)

Assuming that the total time is fixed and that there is ample time to accomplish all items, varying the time of any one of the other terms will decrease the percentage of total time used by the other terms. Therefore, given the above constraints, increasing the time writing data will act to minimize the percentage of time spent loading/unloading and picking/placing. An example of how this helps us is given below:

Questions:

What is the total time required to give the minimum and maximum file sizes for an Exabyte EXB-120 cartridge handling system loaded with Exabyte EXB-8500 tape drives?

What are the minimum and maximum percentages of time spent writing data?

To answer these questions, the following information is needed.

Average load unload time = 50 seconds
Average pick and place time = 20 seconds
Sustained transfer rate = .5 MB/second

The minimum file size is given by the total time equaling 70 seconds:

70 = 50 + 20 + 0

Thus the file size must equal zero according to equation 2.

The maximum file size is 5,000 megabytes so the write time is 10,000 seconds

10,070 = 50 + 20 + 10,000

Thus:

The minimum time spent writing = 0%
The maximum time spent writing = 99%

Unfortunately this does not address the more important question, which is:

What is the optimum file size for the EXB-120 using EXB-8500 drives?

In order to answer the above question more information is required. The optimum file size is a function of optimum system throughput. The optimum system throughput depends upon using the drives efficiently. The first requirement to using the drives efficiently is the need to minimize the time spent not writing. The second requirement is the need to keep as many drives as possible streaming for as long as possible. In order to maximize the time spent writing, one must overlap the time spent picking and placing. Overlapping the pick/place time is a function of the number of drives and the number of pick/place devices. A way to compare the effectiveness of various duplication systems in maximizing the time spent writing is to multiply the minimum of the ratio of the pick/place devices by the average pick/place time plus the load/unload time. As was discussed in equation 2 this will maximize the writing time. This relationship is given below:

(equation 4)

$$MEDU = Min\ ((D_1 / P_1) * (APPT_1 + ALUT_1),$$

$$(D_2 / P_2) * (APPT_2 + ALUT_2),$$

$$\ldots,$$

$$(D_n / P_n) * (APPT_n + ALUT_n))$$

where:

MEDU	=	Most efficient drive usage
Min	=	The minimum function, this function selects the smallest value in a list of values
$D_{1...n}$	=	Number of drives in a particular duplication system
$P_{1...n}$	=	Number of pick/place devices in a particular duplication system
$APPT_{1...n}$	=	The average pick/place time for a particular duplication system
$ALUT_{1...n}$	=	The average load/unload time for a particular duplication system.

To keep the highest number of drives streaming simultaneously for any duplication system we must measure the sequential load/unload efficiency. That is, if there are more drives than pick/place devices the ratio between drives and pick/place devices represents the need for sequential picks and places. The efficiency of the these sequential activities can be measured as below. Please note that this is a measure of the efficiency of the pick/place algorithm and the raw speed of the hardware, whereas equation 4 is only a measure of the speed of the hardware.

The general equation for the system performance is:

(equation 5)

Performance = [1 - (TOTAL NON-WRITING TIME /
TOTAL DUPLICATION TIME)] * writing speed

Please note that this performance measure is the inverse of the percentage of the total duplication time spent not writing data. The larger this number is the better the performance.

To bound this performance measure we first give the equation which defines the worst relative performance. The true worst case is no data being written; however this is unreasonable. Therefore, we will define the worst case to be the combination of all sequential drives being stopped and then sequentially reloaded, that is:

(equation 6)

$$CYC = \left(\sum_{m=1}^{X} 2 * APPT_m \right) + ALUT + P$$

$$\#\ of\ Cycles = \frac{Total\ data\ written}{Data\ set\ size * \#\ of\ sequential\ drives * \#\ pick/place\ devices}$$

TOTAL DUPLICATION TIME =

(Total data written / Sustained transfer rate) +

(# of Cycles * (APPT + ALUT + P))

The worst case performance measure uses the following template:

Performance = total writing time / total duplication time * duplication speed

= [1 - (Non-writing time / total duplication time)] *
duplication speed

= [1 - ((total reoccurring non-writing time +
initial non-writing time) / total duplication time)] * duplication speed

Therefore, the worst case performance measure is given by:

$$INIT = \sum_{o=1}^{X} APPT_o$$

$$Performance_w = [\, 1 - ((\sum_{n=1}^{\#\ of\ Cycles} CYC_n + INIT) / TOTAL\ DUPLICATION\ TIME\ )\,] * DS * PA$$

Where:

DS	=	Sustained drive transfer rate
PA	=	# of drives able to write in parallel
CYC	=	Amount of non-writing time used per cycle
INIT	=	Initial pick and place time
$Performance_w$	=	Sustained duplication percentage of non-writing time worst case
X	=	The number of drives
# of Cycles	=	The number of pick/place cycles required to complete the duplication
P	=	Pause between each pick/place cycle

Next we give the equation which defines the best possible relative performance:

(equation 7)

$$INIT = \sum_{o=1}^{X} APPT_o$$

$$\text{Performance}_b = \left[1 - \left(\left(\left(\sum_{n=1}^{\#\text{ of Cycles}} (\text{APPT}_n + \text{ALUT}_n + P_n)\right) + \text{INIT}\right) / \text{TOTAL DUPLICATION TIME}\right)\right] * DS * PA$$

where:

INIT	=	Initial start-up time
X	=	Number of drives
Performance_b	=	Sustained duplication percentage of non-writing time best case

The pause term of equations 5 and 6 can be non-zero under a variety of circumstances. One such circumstance is when the duplicated data set size is too small to keep all sequential drives streaming.

To compute the best fixed data set size for duplication purposes, we use the following equation:

(equation 8)

$$\text{Data set size} = \left(\sum_{m=1}^{X-1} (\text{APPT}_m + (2 * \text{APPT}))\right) * \text{Min}(s, b)$$

Where:

X = The total number of Drives

s = Sustained transfer rate

b = Burst transfer rate

Equation 8 represents a good first approximation of the data set size needed to maximize the data duplication effort. As the data set size increases from equation 8 the duplication effort performance approaches the limit set by equation 7.

The final tool allows us to understand the relative cost and benefits of a duplication system. First we define the total system cost.

(equation 9)

$$TC = \left(\sum_{m=1}^{\text{total duplications}} D_m + (W_m * C_m) + PER_m + MED_m\right) + (S * F) + P + E + CO$$

Where:

TC	=	Total duplication system cost for the life of the duplication equipment
D_m	=	Amortization of duplication equipment per duplication
W_m	=	Media weight per duplication
C	=	Cost of transportation per duplication
PER_m	=	Personnel cost per duplication
MED_m	=	Media cost per duplication
S	=	Size of the foot print of the duplication equipment
F	=	Floor space cost
P	=	Total price of the duplication equipment
E	=	Total electrical cost
CO	=	Total cooling cost for the duplication equipment

The system benefits where defined in equation eight using the best case rating. Therefore the cost benefit ratio for a duplication system is given below:

(equation 10)

$$CBR = TC / Performance_b$$

Where:

CBR = Cost benefit ratio

Using the Tools:

We now have the tools to answer the question posed earlier; that is, what is the optimum file size for the EXB-120 using EXB-8500 tape drives? First we restate the question as two equivalent questions:

1 What is the minimum data set size needed to keep an EXB-120 (with EXB-8500 drives) streaming the greatest amount of the time?

2 What is the largest file size which can fit on an EXB-8500 tape?

The answer to the second question is the easiest, approximately 5 gigabytes. The answer to the first question is computed as follows:

APPT = 20 seconds

Number of sequential drives = 4

Internal drive buffer size = 1 megabyte

Burst rate = 1.5 MB/second

Sustained rate = .5 MB/second

Minimum Data set size = 50 megabytes

Therefore:

The best duplication data set sizes are greater than or equal to 50 megabytes and less than or equal to 5,000 megabytes.

We can also answer these questions:

- What is the best performance for an EXB-120 with 8500 drives in the data duplication application?

- What is the worst reasonable performance for an EXB-120 with 8500 drives in the data duplication application?

To answer the first question we need the following information:

APPT	=	20 seconds
ALUT	=	50 seconds
Pause	=	0 seconds
INIT	=	80 seconds
Total data written	=	4000 MB
Total duplication time	=	9400 seconds

DS = .5 MB/second

PA = 4 drives

# of Cycles = 20 cycles

Therefore:

$Performance_b$ = 1.68 MB/second duplication rate

To answer the second question we need the following information:

Cyc = 210 seconds

Therefore:

$Performance_w$ = 1.09 MB/second duplication rate

If we replace the EXB-8500 drives with their equivalent half height versions what would be the best performance? The correct data set size should be 120 MB; however, we kept the data set size at 50 MB so that we can compare equivalent situations.

APPT = 20 seconds

ALUT = 50 seconds

Pause = 0 seconds

INIT = 160 seconds

Total data written = 4000 MB

Total duplication time = 1700 seconds

# of Cycles = 10 cycles

DS = .5 MB/second

PA = 8 drives

Therefore:

$Performance_b$ = 1.98 MB/second

In order for us to be able to compare devices with the same number of drives, we will now compute the performance of two EXB-120s with EXB-8500 drives. This data set size should be 50 MB

APPT = 20 seconds

ALUT = 50 seconds

Pause = 0 seconds

INIT	=	80 seconds
Total data written	=	4000 MB
Total duplication time	=	1600 seconds
# of Cycles	=	10 cycles
DS	=	.5 MB/second
PA	=	8 drives

Therefore:

$$\text{Performance}_b = 2.3 \text{ MB/second}$$

We see that two EXB-120 with EXB-8500 drives perform the data duplication task better than one EXB-120 with the equivalent half height drives. The natural next question would be, which data duplication equipment gives me the best cost/benefit ratio. This question is answered below.

Because of the similar nature of the equipment being compared, we can make several simplifying assumptions before calculating the total cost. These assumptions are given below:

- The amortization cost would be approximately the same and therefore does not need to be included
- The media weight and cost of transportation would be the same and therefore do not need to be included
- Personnel costs would be the same and therefore do not need to be included
- Floor space size difference is negligible and therefore does not need to be included
- Energy cost differences are negligible and therefore do not need to be included
- Cooling cost differences are negligible and therefore do not need to be included

$P_{2 \text{ EXB-120 with EXB-8500 Drives}} = 200{,}000$ dollars

CBR = 200000 / 2.3

= 86,956

$P_{\text{EXB-120 with half height drives}} = 100{,}000$ dollars

CBR = 100000 / 1.98

= 50,505

As can be seen given the above assumptions and the amount of data to be duplicated the better value would be the single EXB-120 with eight half height drives versus two EXB-120s with full height drives.

N93-30460

The Fundamentals and Futures of Removable Mass Storage Alternatives

by
Linda Kempster, President
Strategic Management Resources, Ltd.
6503 Lisa Lane, Bowie, Maryland 20720-4706

INTRODUCTION

This article reflects my view of how the storage products have been introduced into the marketplace, where they came from, and where others will continue to come from in the future. My corporate goal is to be a resource for those searching for removable solutions to mass storage problems.

My introduction to optical storage occurred a few months before signing a non-disclosure agreement with FileNet on August 8, 1983. By 87 or 88, as the optical craze was getting more popular, I started looking for similar or complementary storage technologies. I am still looking and my research is constantly turning up new entrants into this field. Due to the scope of the coverage in this field, this article does not dwell on any single technology. The goal is to provide information that is not compiled in any other single source and focus on facts that are not commonly known.

I have provided a few baseline assumptions to ensure the mathematical calculations remain consistent. 1) Hard-copy 8.5" x 11" documents which are scanned at 200 dots per inch (dpi) and compressed at a ratio of 10:1 result in a document image which requires an average of 50 Kilobytes (KB) of storage. 2) An average ASCII page requires 2 KB of storage. 3) An average file cabinet drawer can hold 2500 pieces of paper. 4) One GB of storage can hold an average of 20,000 document images. A reel of 6250 tape holds 180 Megabytes (MB).

HELICAL SCAN TAPE CASSETTES

Cassettes have become very familiar. Using helical scan technologies, Metrum Information Storage has developed a drive that can write 14.5 GB on a T-120 Super VHS cassette providing a storage media price of $.002 per MB. In storage equivalents, that roughly equates to :

- 290,000 document images
- 29 four-drawer file cabinets
- 80 reels of 6250 tape
- 7,250,000 ASCII documents

An autochanger with a footprint of 21 square feet can hold 600 cassettes and provide an automated storage capacity of 8,700 GB or 8.7 Terabytes (TB) at a system cost of $0.06 per MB. Access to any file on a mounted tape is 45 seconds.

Sony, Hitachi and Ampex have developed three basic sizes of digital cassettes: small, medium and large. In the video world, there are two different ways of recording data on these media. Basically, D-1 technology refers to a video signal that is divided into three separate components, digitized then recorded onto tape. During playback, the three streams are output as three analog signals. This tape format meets instrumentation standard ANSI X3B.6. The D-2 technology is different in that after the video signal is divided into three component signals, they are combined before digitization and recording onto tape. The output during playback is a combined analog signal. The D-2 video tapes are made by Hitachi, Sony Ampex, Fuji, Maxell, TDK and 3M. Taking advantage of digital recording market opportunities and the availability of standardized media, RCA, E Systems, Ampex, and Sony have developed helical scan digital recorders which provide the storage formats and data capacities listed below:

- D-1S 16 GB
- D-1M 44 GB
- D-1L 100 GB
- D-2S 25 GB
- D-2M 75 GB
- D-2L 165 GB

Ampex currently has an autochanger available that will hold 256 D-2S cassettes for a total automated capacity of 6.4 TB. E-Systems offers a 220-unit data tower that can provide automated access to 5.5 TB. Each of these units require less than 21 square feet. Multiple libraries can be linked together to provide even more robotically-addressable storage capacity. Library systems are in use today. In a standalone mode, RCA can simultaneously write to 1, 2, 3, or 4 large D-1 cassettes at 400 Megabits per second (Mbps) each and reach a maximum of 1 600 Gigabits per second (Gbps).

Helical scan technology has also been applied to smaller tape formats. Several companies, including Hewlett Packard, Sony, Hitachi, GigaTrend and Archive, have introduced a 4 millimeter (mm) cassette that holds 2 GB of uncompressed data on 60 meters of tape. There are organizations using this computer-grade tape to replace COM (computer output microfilm) and master CD-ROMs. For automated applications, library units are available that manage up to 60 cassettes. Another member of the small cassette group is being offered by Exabyte. The 8 mm format holds 5 GB of uncompressed data. Vendors such as Bull, IBM, Sun, NCR and Wang support this technology. Mass storage libraries have been developed that will hold up to 432 cassettes providing up to 2,160 GB of robotically-addressable storage.

The mainframe environment is served by two helical scan technologies from MASSTOR. Their tape cartridge is 6.5 inches square and 3 inches wide. The capacity is almost 32 GB. The library unit for this format holds 32 cartridges and provides 1 TB. They also provide a tube-shaped unit, 1.8 inches in diameter and 3.4 inches long. The library unit which holds these 350 MB units can accommodate 316 "tubes". The honey-comb shaped interior of the library uses gravity to move the tape units into the readers.

MICROFICHE

A laser-based microfiche technology has been introduced by iBASE Systems Corporation. This printing device allows a user to select images from an optical or magnetic storage unit and print them at either 200 or 300 dpi on microfiche . Using approved film-development methods can produce archivable images and provide acceptable back-up optical storage.

LONGITUDINAL RECORDING TAPE CARTRIDGES

Carlisle Memory Products and 3M jointly developed the 1/4" cartridge format. The cartridge is 4" x 6" and can hold 2.1 GB. The use of barium ferrite media in the quarter inch cartridge (QIC) may support the storage of 35 GB by 1995. There are currently over 7 million QIC drives in use today.

Digital Equipment Company has over 300,000 tape drives using their standardized .5 inch cartridge for storage. They have introduced an external drive unit that uses the tapes for removable storage. The 4.1 inch square tape cartridges hold 2.6 GB. The next generation will hold 5.2 GB and by 1994, the capacity could reach 50 GB. A small library unity holds 7 tape cartridges.

Storage Technology has developed and sold over 4,000 library units which house 6000 of the 200-MB IBM 3480 cartridges and provide 1.2 TB of robotically-addressable storage. The modular libraries have a footprint of 121 square feet and are large enough to allow a technician

to actually enter the unit to provide any service necessary. The libraries serve IBM and over 15 non-IBM platforms. Up to 16 libraries can be linked to offer 19.2 TB of data storage. The next generation of cartridge, the 3490, offers a capacity of 400 MB with 2:1 compression. After that, 36 track tapes will be introduced and the native capacity will reach 800 MB. Future developments will support helical scan recording resulting in 20 GB per tape. The strategic direction of the company is to introduce systems more in line with an office environment. Using this same cartridge, Memorex/Telex has the capacity to manage over 1.3 million tapes using a combination of libraries.

LaserTape Systems has used the 3480 cartridge in conjunction with non-erasable digital paper, often called optical tape. By cutting digital paper into .5 inch wide strips 541 feet long, the company can store up to 100 GB in a single unit. Using the math presented in the previous paragraph, the library capacity expands to 600 TB in 121 square feet. Sixteen libraries would provide 9,600 TB or 9.6 Petabytes of robotically-addressable storage. At $250 per cartridge, the media price would be $.0025 per MB.

CARD-BASED TECHNOLOGIES

Storage systems that need to operate in mobile environments may use the low-cost option offered by numerous vendors supporting the 2.86 MB optical card. This credit-card sized optical storage media can store 1,430 ASCII text documents or 57 document images using WORM technology on a media carrying a 10-year life expectancy. These cards are being used for personalized medical record storage. Within the year, a multi-layer phase change card may be introduced that has the projected capacity of 1 GB. The entertainment industry could use this media to record up to 10 audio CDs or a full length movie. This revolutionary commercial introduction would truly inspire the techofan!

Chip cards can provide 8 KB of storage. The 8 bit microprocessor may support CPU, RAM, ROM or EEPROM functions. Memory-only chip cards can provide 2 KB of nonvolatile data storage.

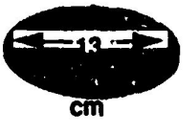
Magnetic strip cards can be used as coin replacement units for mass transit operations or telephone services. The reusable cards are being used by USPS, NYNEX, GTE and Canada Post.

Memory cards are coming in numerous formats complete with a variety of storage capacities ranging from 2 MB to 64 MB. These plug-in memory formats are geared to serve the notebook or laptop industry

COMPACT DISC READ ONLY MEMORY

One of the least expensive media for mass distribution of reference-type database information is the 12 cm Compact Disc Read Only Memory (CD-ROM). A 650 MB disc can hold 13,000 document images or 325,000 pages of ASCII text. An autochanger is on the market that can hold 250 discs and provide access to a networked library of published material. One vendor has introduced a single unit which houses 64 drives in a single cabinet.

There are emerging applications for this inexpensive media. Write once (W1) drives are available to those who wish to store non-erasable information on inexpensive CDs. The drives to read these discs are much less expensive than other W1 drives. The CD Recordable media is being used to capture mainframe data and play it back on inexpensive drives. The Photo CDs will soon be available in consumer photo development stores. The 250 million cameras in place today will provide the capture devices. The images will be displayable on most TV screens. A technology has been reintroduced with the announcement of the 3.5 inch read only memory (O-ROM) disc. The capacity is 122 MB and because they are smaller and lighter, they spin 2 - 9 times faster and provide faster access and seek times. Unlike the first introduction of CD media this size, it is in a protective case, similar to a floppy cartridge.



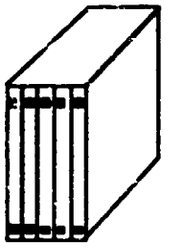
REWRITABLE DISKS

One of the reasons that WORM disks were not readily accepted into the marketplace is that data processing professionals did not like the permanence associated with the media. According to Dr. Robert Freese, magneto-optic (m/o) technology uses some principles of magnetic storage to store data on 5.25" (13 cm) optical disks. Storage capacities range from 400-1500 MB These disks have found homes beside high capacity workstations and in

jukebox environments. Multi-media drives have been introduced by numerous vendors which will write to either 5.25" WORM or erasable platters.

The emerging format of erasable technology is the 3.5" m/o with the standardized capacity of 128 MB. There are disks available with 256 MB capacity and projected capacities reach 520 MB. These disks can spin faster and provide faster seek times. There are currently 18 vendors announcing this format and it is expected that these disks will be more popular for desktops and notebooks, than the 5.25".

Other 3.5 inch media include the barium ferrite disks with a capacity of 21 MB. At $25 per disk and under $300 for a drive, this is becoming an attractive media for smaller applications.



12 & 14 INCH OPTICAL MEDIA

LMSI markets a revolutionary 12" WORM drive. The drive operates with a dual head so that each side of the disk is available to the user simultaneously. The user is provided with 5.6 GB of storage without the need to flip the disk in a standalone environment. To compliment this drive, the company offers a 5-platter magazine that fits as a single unit into a slightly different version of the new drive providing 28 GB of storage with a disk swap time of less than 5 seconds. For security, the magazine can easily be vaulted when necessary. The next generation of this system will provide 5 - 6 GB per side of each optical platter. Other 12" WORM drives have been introduced that will store up to 9 GB per platter.

Kodak's latest 14" optical platter has the capacity of 10.2 GB and incorporates a non-erasable form of phase-change technology. The 100-platter jukebox provides 1.02 TB of storage.

Rewritable video disks offer dense storage for analog images. Access time between frames is less than 1 second. The disks hold up to 108,000 black and white or full color images (vs 13,000 on a CD-ROM) and with the right adapter, the disk appears to the system as a "large" CD.

DIGITAL PAPER

The last technology is one of the most exciting and versatile. Imagine a roll of aluminum foil that is silver on one side, golden on the other, and very durable feeling. This is what ICI Imagedata calls Digital Paper. Cut into long strips and wrapped around a reel it has been referred to as optical tape. A 12.5 inch reel of this magic media can hold 1 TB of data and someday may hold up to 40 TB. According to figures from The Sierra Club, the storage of 1 TB of ASCII data on this media instead of paper would save 42,500 trees. Metrum currently sells the CREO drive in the US. Canada now has three of these units running to store satellite data. Australia has two systems doing the same thing. Other sales are pending throughout the world to support a variety of applications including supercomputing sites, oil data storage and medical imaging.

N93-30461

THE NT DIGITAL MICRO TAPE RECORDER

Toshikazu Sasaki,
John Alstad
Mike Younker
Sony Magnetic Products Inc.
10833 Valleyview Boulevard
Cypress, CA 90630

S/2-35
15972
P. 15

INTRODUCTION

The description of an audio recorder may at first glance seem out of place in a conference which has been dedicated to the discussion of the technology and requirements of mass data storage. However there are several advanced features of the NT system which will be of interest to the mass storage technologist. Moreover, there are a sufficient number of data storage formats in current use which have evolved from their audio counterparts to recommend a close attention to major innovative introductions of audio storage formats.

While the existing analog micro-cassette recorder has been (and will continue to be) adequate for various uses, there are significant benefits to be gained through the application of digital technology. The elimination of background tape hiss and the availability of two relatively wideband channels (for stereo recording), for example, would greatly enhance listenability and speech intelligibility. And with the use of advanced high-density recording and LSI circuit technologies, a digital micro recorder can realize unprecedented compactness with excellent energy efficiency.

This is what has been accomplished with the NT-1 Digital Micro Recorder. Its remarkably compact size contributes to its portability. The high-density NT format enables up to two hours of low-noise digital stereo recording on a cassette the size of a postage stamp. Its highly energy-efficient mechanical and electrical design results in low power consumption; the unit can be operated up to 7 hours (for continuous recording) on a single AA alkaline battery. Advanced user conveniences include a multifunction LCD readout. The unit's compactness and energy-efficiency, in particular, are attributes that cannot be matched by existing analog and digital audio formats. The size, performance, and features of the NT format are of benefit primarily to those who desire improved portability and audio quality in a personal memo product.

The NT Recorder is the result of over ten years of intensive, multi-disciplinary research and development. What follows is a discussion of the technologies that have made the NT possible:

(1) NT format mechanics
(2) NT media
(3) NT circuitry and board

NT MECHANICS

In order to achieve the required high areal recording density, the NT format employs the now-familiar rotary head double-azimuth helical-scanning system. The technique used in the NT format, however, represents a significant departure from the rotary-head designs used in VCRs and DAT recorders. Specifically, the small size of the NT system has made it necessary to take entirely new approaches to loading and tracking.

With conventional rotary-head systems, the transport must include a loading mechanism that withdraws the tape from the cassette shell and wraps it around a portion of the head drum. Beta, VHS, 8mm video, and DAT mechanisms all employ some variation of this technique, using either a "U" or an "M" loading pattern. Such mechanisms are necessarily complex as the

tape must be handled with great precision and care. These designs are also not space-efficient because a significant volume in front of the cassette must always be set aside to permit the tape wrap. With conventional rotary-head systems, therefore, it is impossible to reduce size, weight, and cost beyond a certain point. Moreover, the nature of these tape-wrapping mechanisms is such that cassettes cannot be loaded or ejected while the power is off.

The non-loading system employed in the NT format is a novel solution to these problems. As shown in Figure 1, there is no need to withdraw the tape from the shell. Tape wrap is instead accomplished by inserting the head drum assembly into the front opening of the cassette. Built into the cassette shell are molded tape guides which serve the same function as the inclined and vertical guides in conventional rotary-head systems. Pressure rollers, too, are an integral part of the shell, making the head drum and capstan the only external elements that need be engaged with the cassette. Since tape travel is fully contained within the cassette, the non-loading system provides the high-density recording benefits of a rotary-head design while preserving much of the simplicity and space-efficiency of conventional fixed-head mechanisms.

At the heart of the NT format is the non-tracking system technology. (NT is derived from Non-Tracking.) Achieving higher recording density entails shorter wavelengths, thinner tape, and narrower tracks. While advanced magnetic head, metal-evaporated tape, and signal modulation technologies (discussed below) all contribute to the attaining of the NT format's very high recording density, the narrow track width requirement creates certain problems. With such narrow tracks, read/write performance (data integrity) would be severely compromised by the slightest imprecision in tracking. Unit-to-unit compatibility would be difficult to ensure. These problems are further compounded by the limited practical tolerances in the NT cassette's built-in tape guides and by the extremely short length of exposed tape with which the non-loading system must work. In fact, these circumstances make it impossible to use conventional tracking schemes.

These tracking issues were only able to be addressed by abandoning the traditional approach based on high-precision servo control. The non-tracking playback method employs double-scanning combined with high-speed memory to accurately read all of the recorded data. Because the NT does not rely on tracking precision, it eliminates the need for fixed control heads and automatic track finding signals and circuits, making the entire system considerably simpler and smaller.

In conventional rotary-head systems, there must be a one-to-one tracking correspondence between record and playback. That is, since two heads with opposite azimuths alternately lay down successive tracks during record, each track must be traced at the same angle by the corresponding head during playback. If this is not done precisely, mistracking occurs and data are lost. With the non-tracking playback method, the one-to-one tracking correspondence with the recorded tracks is intentionally altered. The speed of the head drum rotation is doubled, resulting in a double-density scan. Moreover, because the tape speed remains the same, the actual head trace path during playback is, in effect, further inclined. This is shown in Figure 2.

This figure also shows how the non-tracking method can read all of the recorded data despite the skewed head trace. Consider the output from head A as it makes four scans at double speed. Demodulating the RF output to digital signals results in data strips 1 through 4, corresponding to the four traces. (Note that although only 32 data blocks are shown in this illustration for simplicity, each track actually comprises 104 data blocks.) Focusing our attention on the data contained on track A , we see that each data strip 1 through 4 contains part of the information on the track with a certain amount of overlap. Interspersed among track A's blocks are data from other tracks. Error blocks occur whenever head A tries to read data from a track recorded by head B because of the azimuth discrepancy. The four traces contain all the data blocks necessary to fully reconstruct track A. The information at this point, however, is out of sequence.

The out-of-sequence data are fed to a buffer memory. By using random access sequential reading, the data are compiled into the correct order. During recording, the data should be written with sequential memory. The output of the memory is clocked by a quartz reference oscillator, then error-corrected prior to D/A conversion. The system recognizes the correct position in memory for each data block. (Figure 4). Each data block actually consists of smaller sub-blocks, one of which contains an address. Since each block, or piece of the jigsaw puzzle, is represented by a unique address that corresponds to unique position in the buffer memory, the process is quite simple.

The non-tracking method described above depends for its operation on the speed of the readback head being greater than that of the recording head; in this way each read track intersected several written tracks. But changing drum rotation obviously does not allow for a smooth transition between record and playback. Also the servo would require some special function in order to track during ramp-up and down. However the same non-tracking operation can be achieved using a constant-velocity drum if the read/write track width ratio is adjusted so that two read passes are made for each write track. The readback redundancy thus obtained can be used to reconstruct the written data in an analogous, albeit less intuitive, fashion as can be seen from study of figure 3.

Figure 5 is a block diagram that shows the data flow and signal processing involved during NT format playback.

Unlike conventional rotary-head systems, the NT format eliminates the need for tracking servo control. Incorrect tape speed, nevertheless, can cause a discrepancy between the rate at which data are written to memory and the rate at which they are read from memory, the latter being determined by the quartz reference clock. Such discrepancies can cause memory overflow or underflow. Therefore, the NT format requires servo control to regulate tape speed. Figure 6 is a block diagram of the NT playback servo system.

The address values from the playback data are read and compared to reference address values generated by the reference clock. The difference between these address values must be kept reasonably constant in order to prevent memory overflow or underflow. To ensure this, a phase error is derived by subtracting an offset value from the address difference value. A digital low pass filter averages the phase error values over time, the gain is adjusted , and the signal summed with the speed error component obtained via the motor tachometer. The resulting servo data are converted into a PWM (pulse with modulation) signal.

When the carrier component of the PWM signal is removed by a low-pass filter, a motor drive voltage results. The entire servo loop works to keep the amount of data in the non-tracking memory buffer approximately constant.

In the non-loading system, approximately one-third of the head drum diameter is inserted into the cassette shell. To make this possible, the diameter of the drum must be small, and the side of the assembly that is inserted into the cassette must be extremely thin. Furthermore, because the rotary head assembly is inserted at an angle, the top and bottom of the drum must be constructed with slanted cuts. These requirements have been met through the use of an ultra-high-precision miniature three-layered drum. The diameter of the drum is 14.8mm as specified by the NT format. The side of the assembly that is inserted into the cassette is only 4mm thick.

A highly sensitive head is necessary to ensure playback RF output because the track is narrow, the recording wavelength is short, and the relative speed is slow due to the small size of the drum. It is also difficult to ensure head contact because the tape width is narrow and the tape tension is low. This problem can be resolved by using the MIG (Metal In Gap) head. Known as a double azimuth type, it maximizes extremity shape and alignment, and is positioned on an ultra-small platform.

Because of considerations of RF characteristics, efficiency and induced noise, the rotary transformer had to be placed inside the rotating head. To achieve better winding, an extra thin tape was employed on the small diameter core.

The combined effects of the rotary head transformer, adjacent cross talk and saturation recording places limits on the minimum frequency and/or maximum flux reversal length. Furthermore, in order to adequately suppress crosstalk between adjacent tracks through alternate-azimuth recording, a low frequency component should not exist. This means that a DC-free modulation code which has low maximum-to-minimum frequency ratio is required.

This led to the development of the LDM modulation. Based on the MFM used generally for floppy disks, LDM-2 (Low Deviation Modulation) is free of any DC component and suppresses low frequencies. The minimal flux reversal interval is 1 T, and the maximum flux reversal interval is 2.5 T (1 T is the equivalent of a 1-bit interval before modulation). (see figure 7).

Although the NT format utilizes a rotary-head system, the cassette is similar to most fixed-head cassette formats in that it has two sides. At the end of one side, the cassette can be turned over to continue record/play on the other side. A 120-minute NT cassette, therefore, provides 60 minutes per side. Figure 8 diagrams the location of the forward and reverse tracks on the tape, illustrating how the rotary-head helical-scan system can be implemented in a bi-directional design.

To accommodate bi-directional operation, the cassette lid is hinged symmetrically, enabling it to open upward or downward. The bi-directional design also dictates the inclusion of two pressure rollers, one at each forward corner of the cassette. The action of inserting the cassette automatically opens the lid; the capstan is then pressed against the take-up-side pressure roller to initiate tape drive.

The Aramid base film is characterized by a high Young's modulus, and it has enabled the development and refinement of a manufacturing process which ensures firm adhesion of the deposited metal layers. It also results in low friction loss at each reel, enabling low-tension tape drive. Consequently, the NT tape, while only 4.8 microns in thickness, is highly reliable and durable.

NT CIRCUITRY

While the design of the NT format in itself enables a high degree of miniaturization, the fact that it is a digital audio recording system makes complex electrical circuitry inescapable. Therefore, in order to realize the design goals of extreme compactness and low power consumption, the NT-1 Digital Micro Recorder incorporates the latest circuit miniaturization technologies.

Six new LSI chips, in particular, were developed expressly for NT format applications. Of these, five are CMOS devices. These six chips are the equivalent of 1.8 million transistors.

1. DSP LSI contains digital over-sampling filters for the A/D and D/A converters, error correction and concealment code encoder, decoder, modulator, demodulator, and non-tracking processing circuitry; it is used in conjunction with an external 1 megabit dynamic RAM chip, which provides the necessary non-tracking and servo buffers.

2. ADA LSI contains the A/D and D/A converters plus all ancillary analog and digital circuits.

3. DET LSI contains digital circuits for playback RF equalization, PLL and associated functions.

4. DRV LSI contains a high performance DC-to-DC converter, power supply regulators, and motor driver circuitry.

5. Micro-CTL LSI contains a microprocessor that performs calculations for motor servo and system control and controls LCD readout.

6. R/P IC contains the RF record and playback amplifiers.

The LSI chips in the Digital Micro Recorder are interconnected via the SSB (simple serial bus). This unique architecture enables the exchange of large volumes of data between the central microprocessor and the individual LSI circuits. At the same time, it reduces the number of required pins on the LSI chips and the number of signal paths on the circuit board. SSB thus facilitates real-time control of numerous functions while recording circuit complexity. As an added benefit, the simplified signal paths decrease the generation and induction of transient noise.

NT TAPE MEDIA

The NT Digital Micro Tape Cassette is ultra-compact about the size of a postage stamp. It is 30mm wide, 21.5mm deep, and 5mm thick, making its volume approximately 1/4 that of a microcassette and 1/25 that of a compact cassette. Figure 9 depicts relative cassette sizes. Notwithstanding these diminutive dimensions, the NT cassette provides a maximum record/play time of 120 minutes.

As explained earlier, the cassette shell incorporates self-aligning tape guides and pressure roller elements that are external to the cassette in conventional rotary-head recording systems. These components are, of course, central to the NT format's non-loading system. The lid mechanism is a relatively simple construction that assures a perfect seal, effectively keeping contaminants out of the cassette.

NT format development from the outset has been based on the use of metal-evaporated tapes because the medium is in several ways ideal for digital recording:

1. In metal-evaporated tapes, only the active magnetic component is deposited onto the base film. This is in contrast to metal particle, oxide, or ferrite formulations, which require an inert binder material. Thus, the magnetic layer of a metal-evaporated tape can be extremely thin. As a result, the recording field can penetrate the entire thickness of the magnetic layer. This makes it possible to overwrite data with 100 per cent erasure, thereby eliminating the need for a separate erase head. The absence of inert binder material in metal-evaporated tapes results in higher magnetic material density, and higher output levels and C/N (carrier-to-noise) ratio.

2. When recording on metal-evaporated tape,the unit magnetizing length (one half the shortest wavelength) is greater than the thickness of the magnetic layer. Under these conditions, self-demagnetization is reduced as compared to that obtained in particulate media. Because of this reduced self-demagnetization, a lower coercivity is needed than in metal particulate tape in order to sustain similar high recording densities. This means that record head pole-tip saturation is less likely to occur.

The dual-layer metal-evaporated formulation used in the NT Digital Micro tape has been optimized for bi-directional record/play. This necessitated a double metal evaporated layer. The direction of tilt of the columnar structure of the evaporated film is set by the incident angle of the metal vapor. The highest playback output obtains when the columns tilt in the direction of head motion. This is illustrated in figure 10. An abrasion-free backcoating prevents wear of the built-in plastic tape guides. A low-abrasion protective coating on the magnetic layer ensures tape stability and prevents premature head wear.

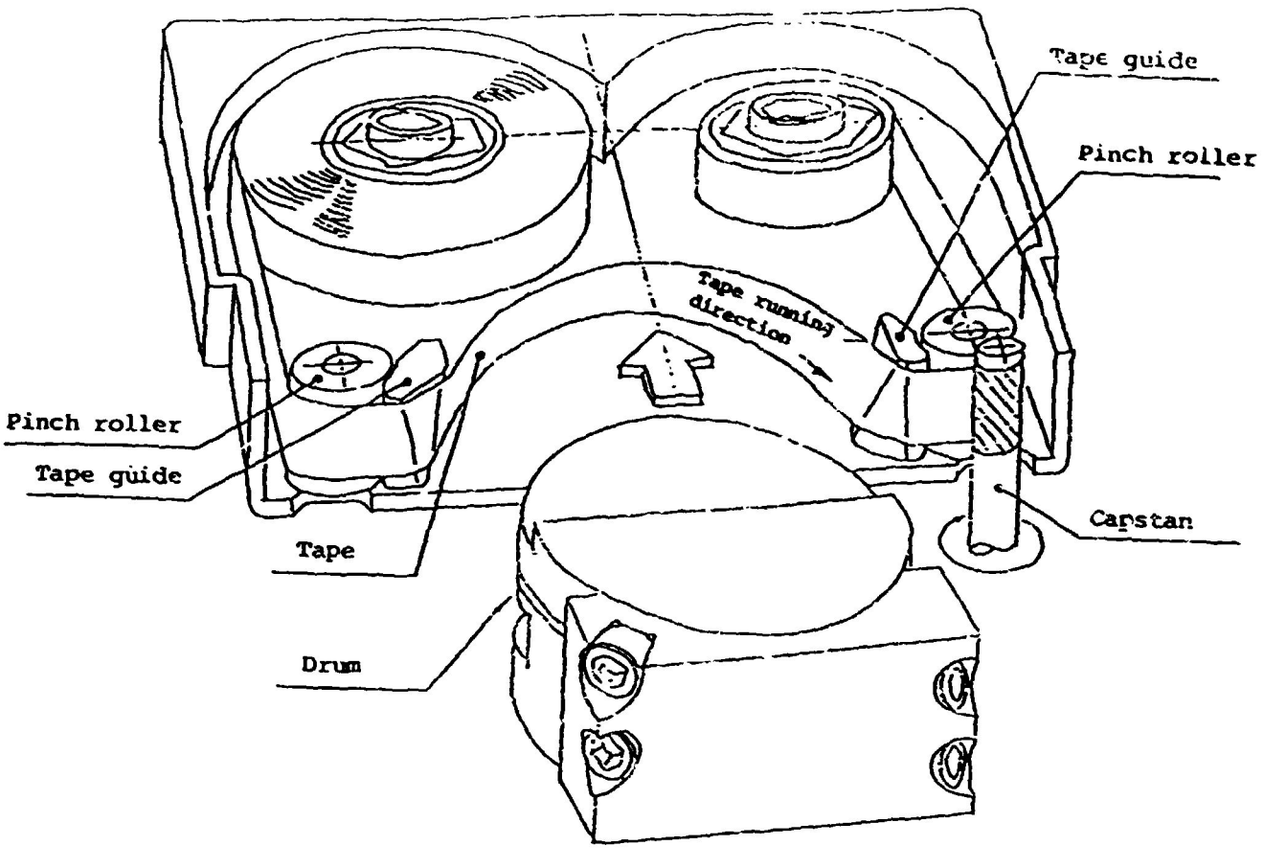
The use of microprocessor control has eliminated the need for trimpots and other similar devices. All control data are held in a non-volatile random access memory so that calibrations and adjustments are retained in the absence of power. Without mechanical trimmers, circuit boards can be smaller, and problems stemming from contact failure or drift are eliminated.

All electrical components-including the LSI chips, resistors, capacitors, and inductors-are mounted on the surface of a highly flexible circuit board. Because the board can be folded, parts that are usually remotely located-such as the headphone and microphone jacks, switches, and LCD-can now be mounted directly on the board as well. This design not only aids miniaturization but also reduces the number of components that can lead to noise or reliability problems.

CONCLUSION

This report covers the introduction of the prototype ultra small NT recorder. The future should bring even greater recorder miniaturization and reduced power consumption concurrent with progress in the semiconductor industry. In fact, some expect to see volume and power consumption reduced to approximately 1/10 of their current state. Furthermore, the possibility of reducing cassettes to 1/25th the size of compact cassettes is very appealing in this age of environmental concerns, which values energy-savings and limited use of natural resources. These concerns, which took root in the past decade, are likely to continue into future generations.

Figure 1 Non-loading Method



Cross-section of Cassette and Non-loading Format Tape Passage

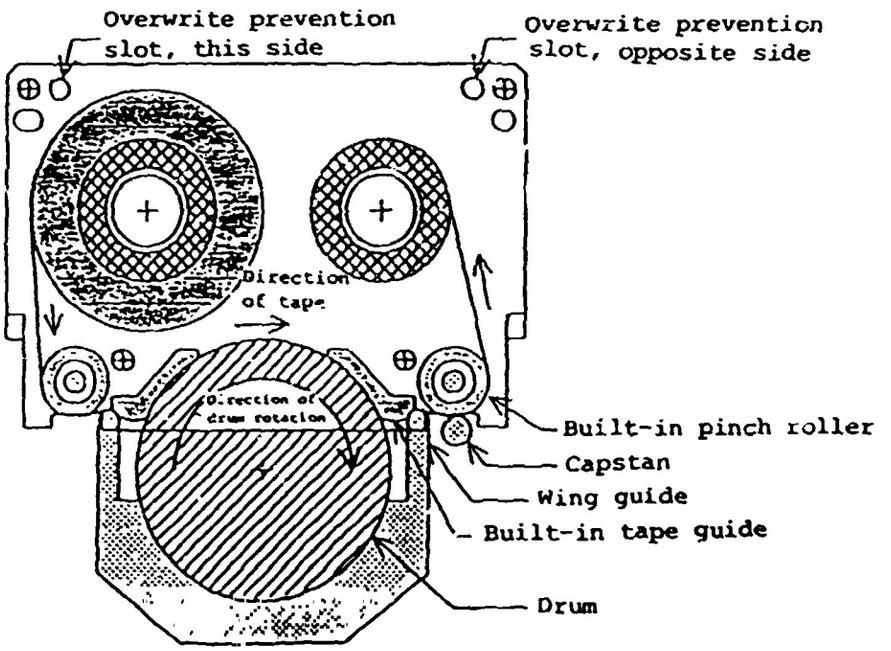


Figure 2

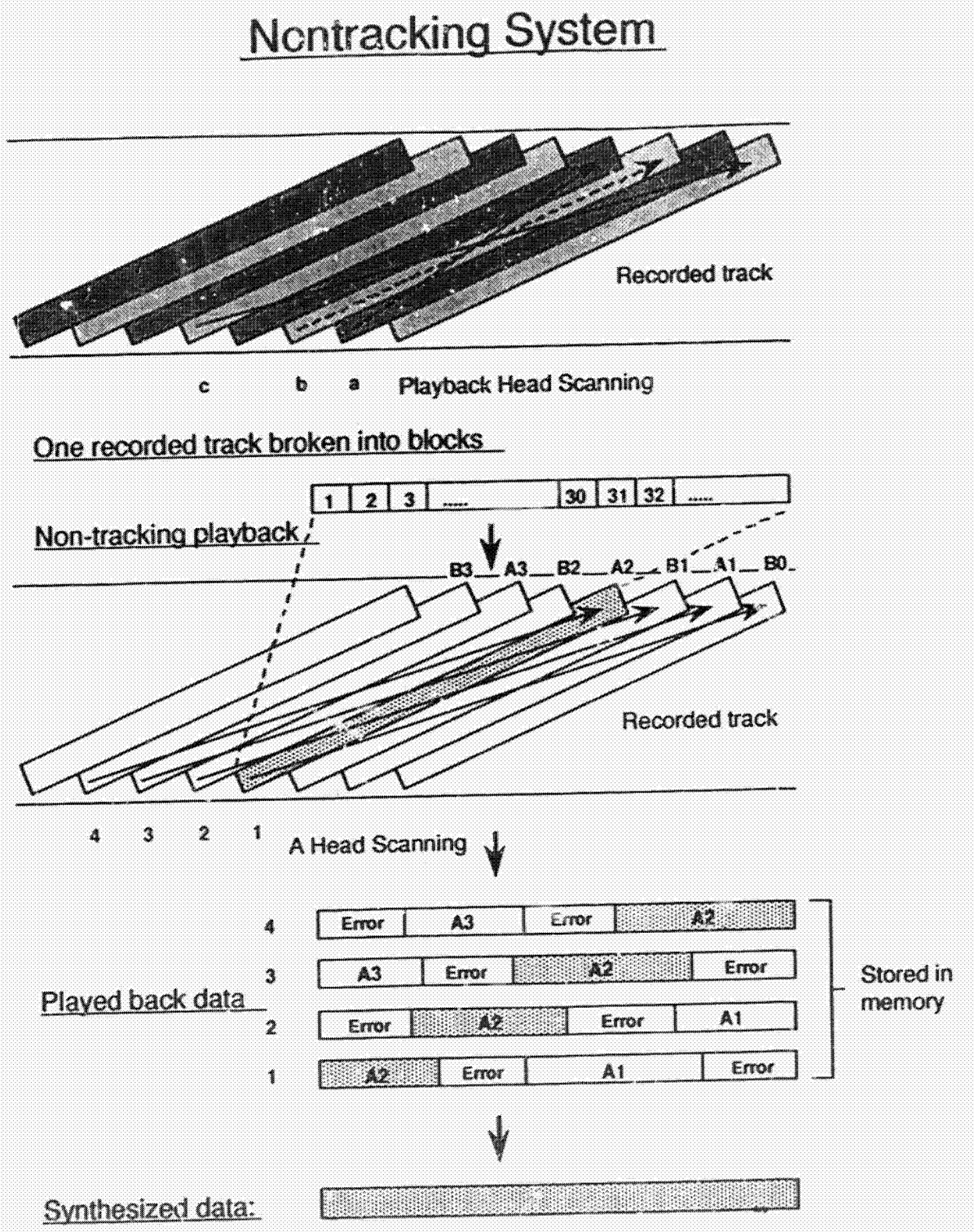

Nontracking System
Recorded track
c
b
a
Playback Head Scanning
One recorded track broken into blocks
1
2
3
.....
30
31
32
.....
Non-tracking playback
B3
A3
B2
A2
B1
A1
B0
Recorded track
4
3
2
1
A Head Scanning
Played back data
4
Error
A3
Error
A2
3
A3
Error
A2
Error
2
Error
A2
Error
A1
1
A2
Error
A1
Error
Stored in memory
Synthesized data:

Figure 3

Nontracking System

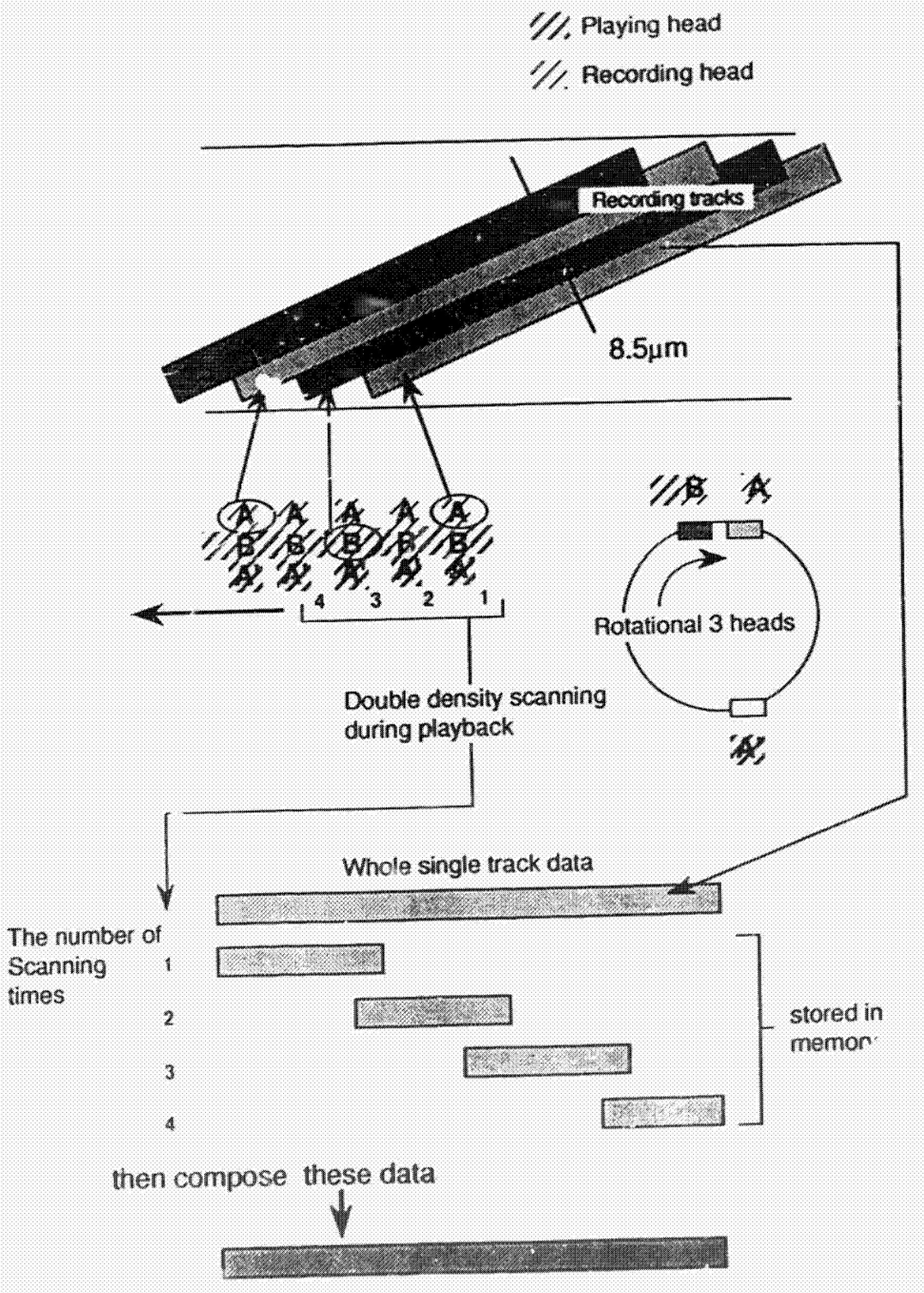


Figure 4

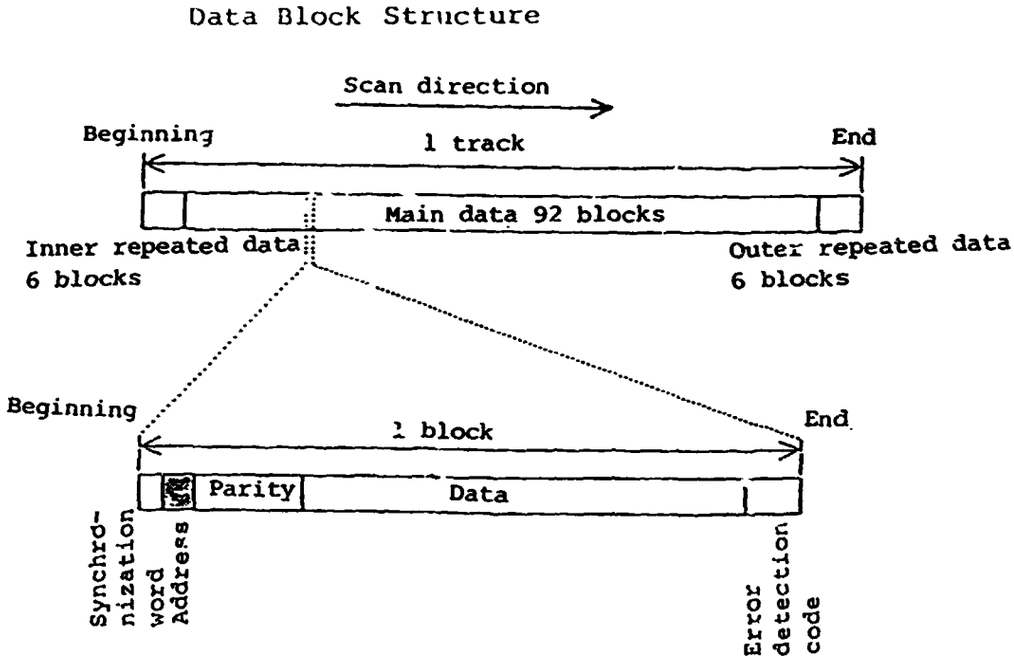

Data Block Structure
Scan direction
Beginning
1 track
End
Main data 92 blocks
Inner repeated data
6 blocks
Outer repeated data
6 blocks
Beginning
1 block
End
Parity
Data
Synchro-
nization
word
Address
Error
detection
code

Figure 5

Block Diagram for Signal Processing of N[illegible]tracking Playback

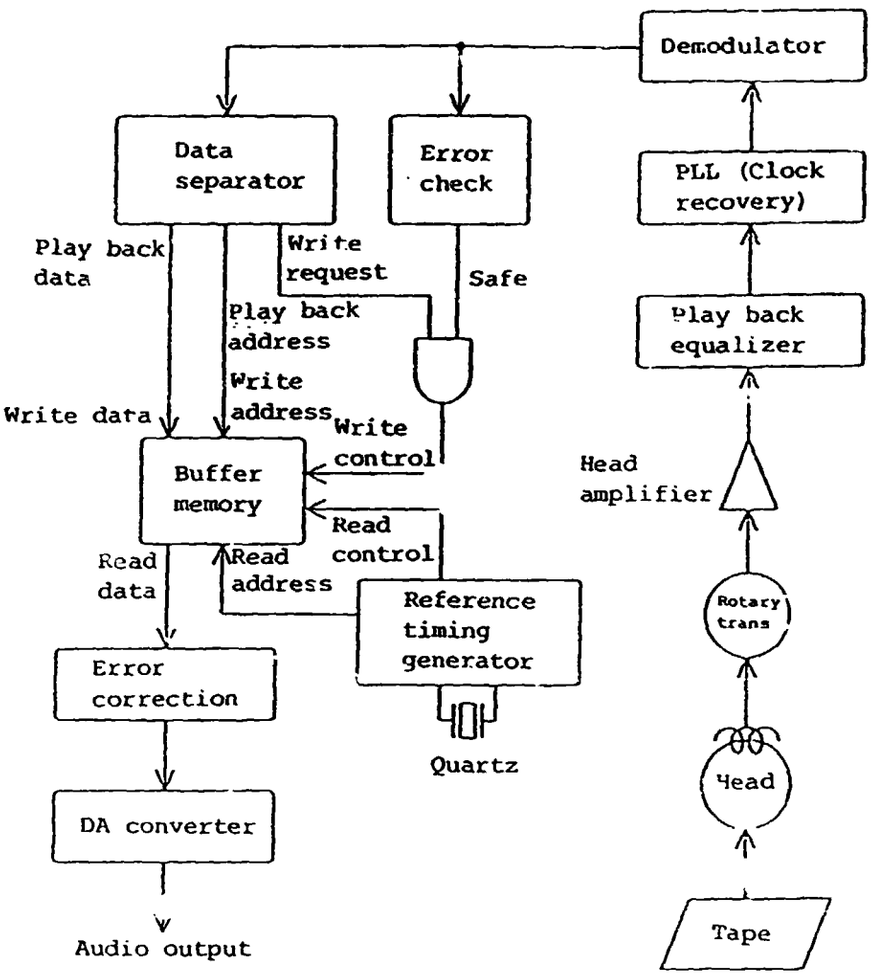


Figure 6 Block Diagram of Non-tracking Playback Servo

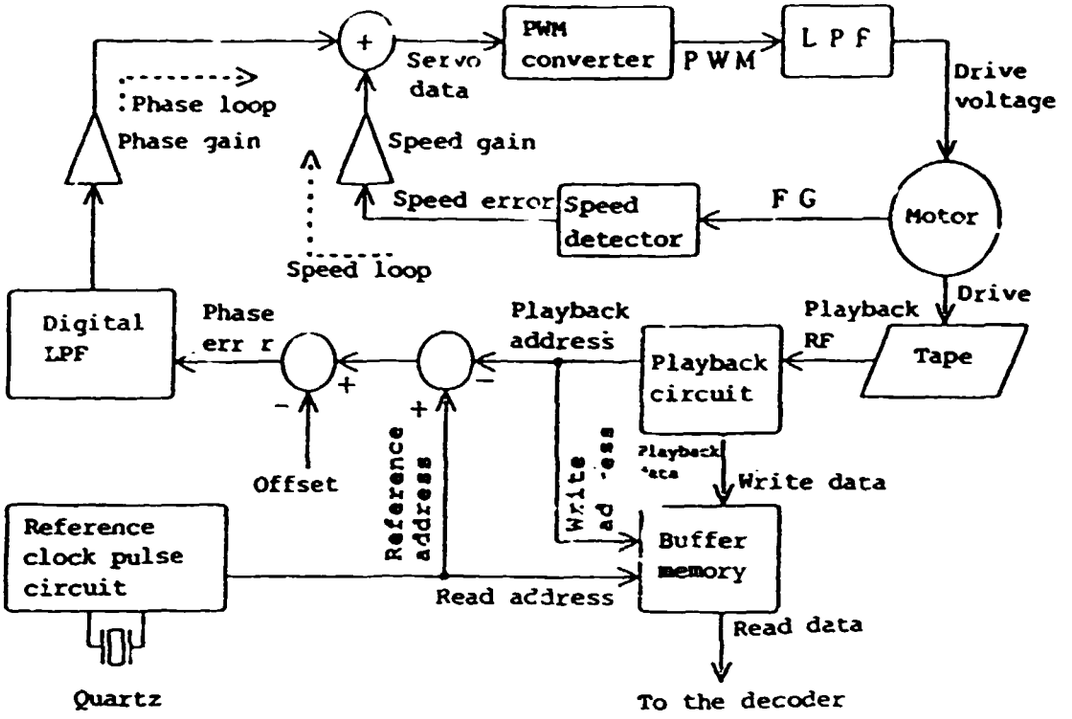


Figure 7 Examples of LDM-2 Modulated Waveform

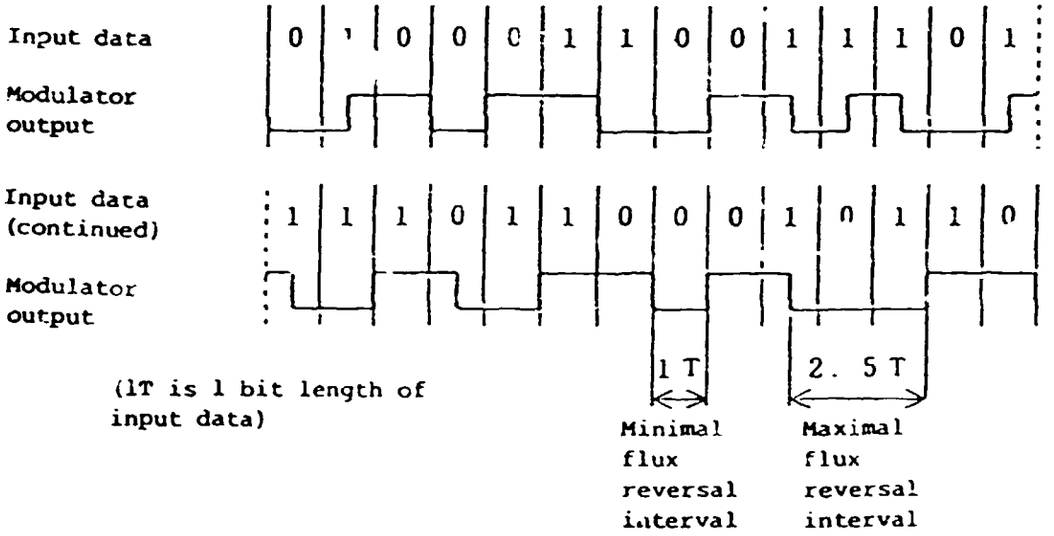


Figure 8

Tape Pattern

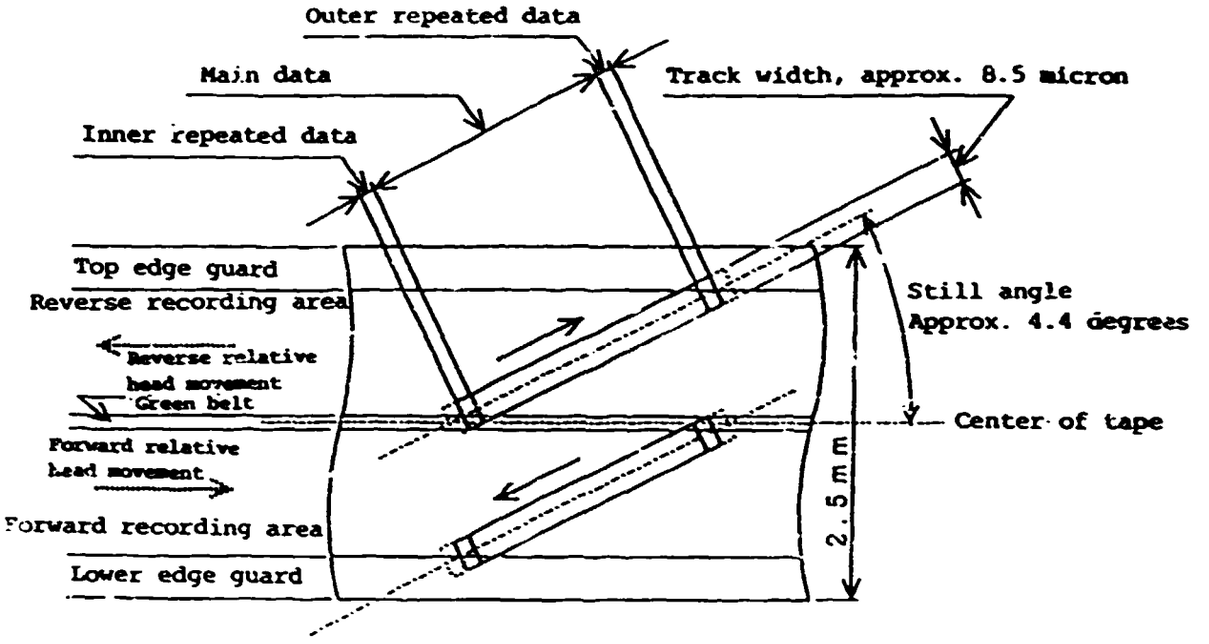
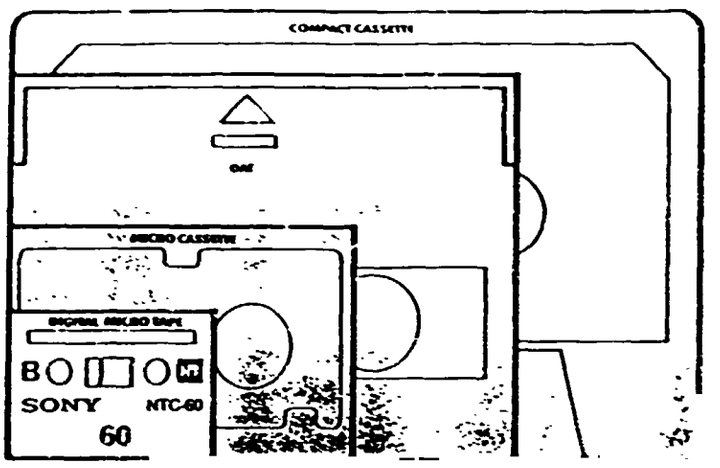


Figure 9



CASSETTE SIZE COMPARISON (FRONTAL AREA ONLY)
NT™ DIGITAL MICRO TAPE, *SHOWN ACTUAL SIZE.*

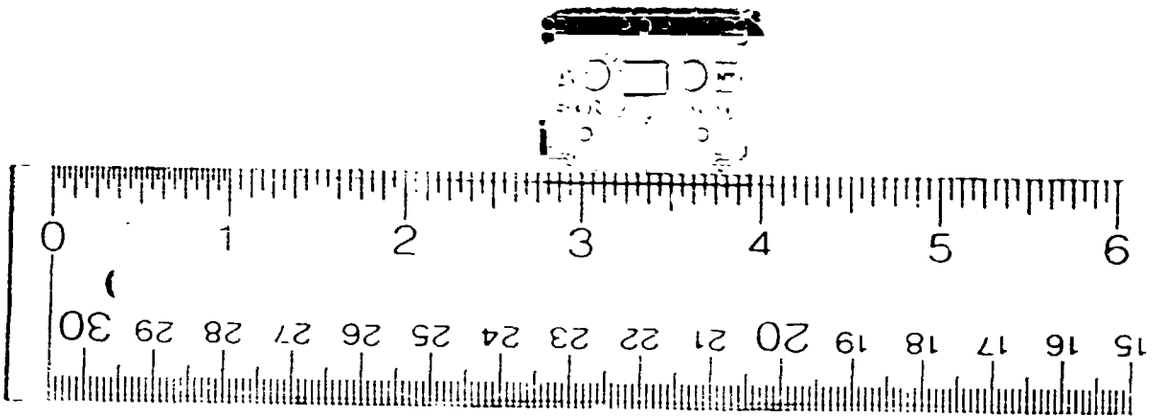
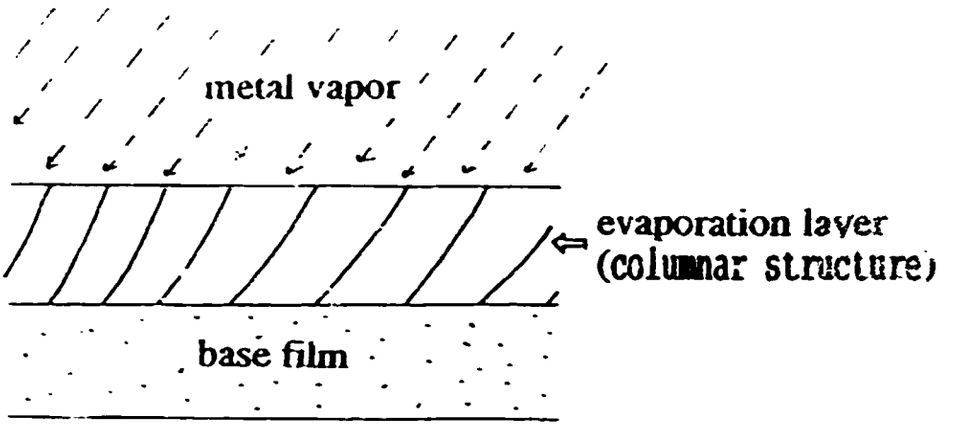
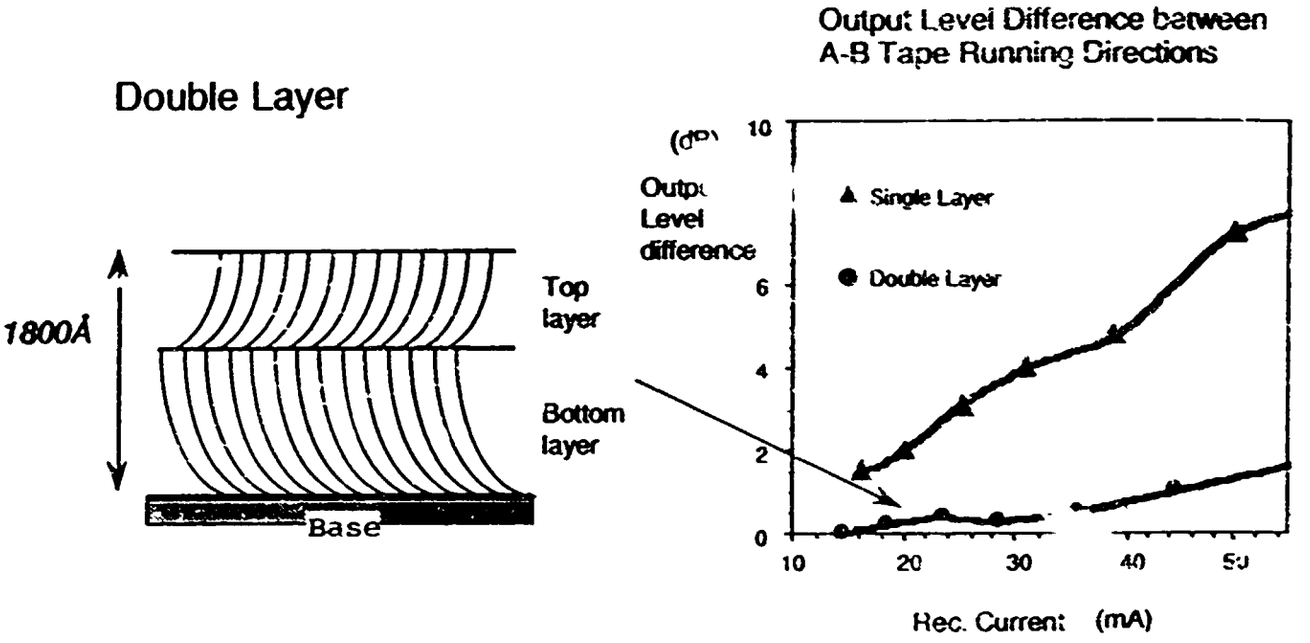


Figure 10


metal vapor
evaporation layer
(columnar structure)
base film


Output Level Difference between
A-B Tape Running Directions
Double Layer
1800Å
Top
layer
Bottom
layer
Base
Output
Level
difference
Single Layer
Double Layer
10
6
4
2
0
10
20
30
40
Rec. Current (mA)

N93-80462

RAID 7 Disk Array

Lloyd Stout
AC Technology Systems and Storage Technology
8201 Greensboro Drive
Suite 220
McLean, VA 22102

ABSTRACT

Each RAID level reflects a different design architecture. Associated with each is a backdrop of imposed limitations, as well as possibilities which may be exploited within the architecturai constraints of that level. There are three (3) unique features that differentiate RAID 7 from all other levels.

(1) RAID 7 is asynchronous with respect to usage of I/O data paths. Each I/O drive (includes all data and one parity drives) as well as each host interface (there may be multiple host interfaces) has independent control and data paths. This means that each can be accessed completely, independently, of the other. This is facilitated by a separate device cache for each device/interface as well.

(2) RAID 7 is asynchronous with respect to device hierarchy and data bus utilization. Each drive and each interface is connected to a high speed data bus controlled by the embedded operating system to make independent transfers to and from central cache.

(3) RAID 7 is asynchronous with respect to the operation of an embedded real time process oriented operating system. This means that exclusive and independent of the host, or multiple host paths, the embedded OS manages all I/O transfers asynchronously across the data and parity drives.

A key factor to consider is that of the RAID 7's ability to anticipate and match host I/O usage patterns. This yields the following benefits over RAID's built around micro-code based architectures.

RAID 7 appears to the host as a normally connected Big Fast Disk (BFD).

RAID 7 appears, from the perspective of the individual disk devices, to minimize the total number of accesses and optimize read/write transfer requests.

RAID 7 smoothly integrates the random demands of independent users with the principles of spatial and temporal locality. This optimizes small, large, and time sequenced I/O requests which results in users having an I/O performance which approaches performance to that of main memory

Sustained Host I/O Transfer Rates

The real issue as far as RAID I/O performance is concerned is the sustained transfer rate to the host. In the RAID 7 device the data drives represent the available bandwidth to store data. If the number of data drives were to be five (5) and those drives were capable of a sustained transfer rate of 200 Kbytes/sec, then RAID 7 could offer the host a 5 X .95 X 200 Kbytes/sec or 950 Kbytes/sec sustained transfer rate. It is significant that unlike other RAID levels, RAID 7 offers a linear increase in sustained transfer capacity as the number of drives increases.

Single Spindle Read/Writes

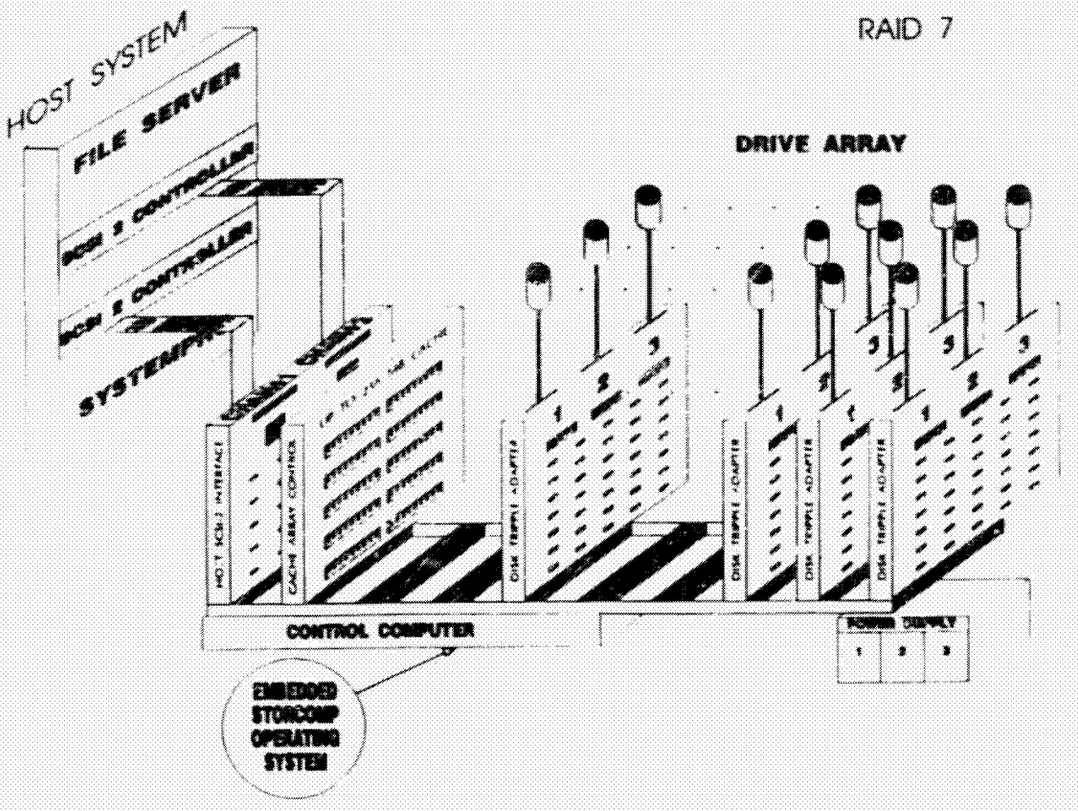
One simple measure of a RAID device ought to be how it answers the following two questions: 1) Can the RAID perform small reads and writes better than a single spindle?; 2) Can the RAID perform large reads and writes better than a single spindle? RAID 5 for example cannot match single spindle performance for large writes, and for some small writes can muster only 1/20th of single spindle performance. RAID 7, however, exceeds single spindle performance in all cases.

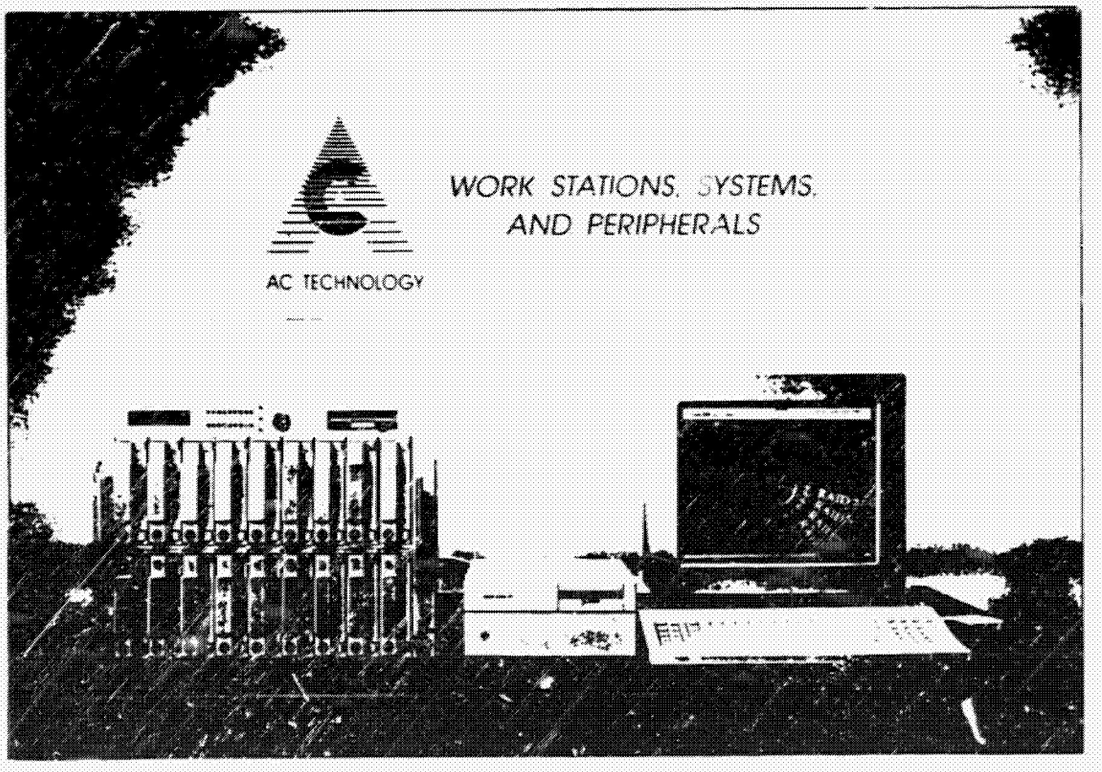
Most of the published "White Papers" on RAID performance compare different measures with different architectures. For example, Mbytes/sec are used to evaluate RAID 3 while I/O's per second are used to measure RAID 5. The problem with such comparisons is two fold: (1) they do not match real world systems which most always have a continuous mix of small and large requests, (2) they mask the performance of the untested measure.

System Configurations

Series A	Series B	Series C	Rackmount
8 Logic Slots	14 Logic Slots	20 Logic Slots	8 Logic Slots
8 Drive Slots 3.5" Only	16 Drive Slots 5.25" and 3.5"	24 Drive Slots 5.25" and 3.5"	9 Drive Slots 3.5" Only
600 Watt FT	1200 Watt FT	1800 Watt FT	600 Watt FT
3 Device I/Os	6 Device I/Os	12 Device I/Os	3 Device I/Os

.. uses industry standard SCSI disks
.. upgradable to 256 Mbytes cache
.. multiple host scalability
.. completely transparent-load and go
.. requires no special software or drivers


RAID 7
HOST SYSTEM
FILE SERVER
SCSI 2 CONTROLLER
SCSI 2 CONTROLLER
SYSTEMPRO
DRIVE ARRAY
CONTROL COMPUTER
EMBEDDED
STORCOMP
OPERATING
SYSTEM


AC TECHNOLOGY
WORK STATIONS, SYSTEMS,
AND PERIPHERALS

N93-30463

Tutorial: Performance and Reliability in Redundant Disk Arrays[1]

Garth A. Gibson
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsbugh PA 15213-3890

S14-61
ISP104
P - 40

A *disk array* is a collection of physically small magnetic disks that is packaged as a single unit but operates in parallel. Disk arrays capitalize on the availability of small-diameter disks from a price-competitive market to provide the cost, volume, and capacity of current disk systems but many times their performance. Unfortunately, relative to current disk systems, the larger number of components in disk arrays leads to higher rates of failure. To tolerate failures, redundant disk arrays devote a fraction of their capacity to an encoding of their information. This redundant information enables the contents of a failed disk to be recovered from the contents of non-failed disks. In this tutorial I will highlight the simplest and least expensive encoding for this redundancy, known as *N+1 parity*. In addition to compensating for the higher failure rates of disk arrays, redundancy allows highly reliable secondary storage systems to be built much more cost-effectively than is now achieved in conventional duplicated disks.

Disk arrays that combine redundancy with the parallelism of many small-diameter disks are often called Redundant Arrays of Inexpensive Disks (RAID). This combination promises improvements to both the performance and the reliability of secondary storage. For example, Table 1 compares IBM's premier disk product, the IBM 3390, to a redundant disk array constructed of 84 IBM 0661 3½-inch disks. The redundant disk array has comparable or superior values for each of the metrics given in Table 1 and appears likely to cost less.

In the first section of this tutorial I explain how disk arrays exploit the emergence of high-performance, small magnetic disks to provide cost-effective disk parallelism that combats the access and transfer gap problems. The flexibility of disk-array configurations benefits manufacturer and consumer alike. In contrast, I describe in this tutorial's second half how parallelism, achieved through increasing numbers of components, causes overall failure rates to rise. Redundant disk arrays overcome this threat to data reliability by ensuring that data remains available during and after component failures.

As far as the organization of redundant data in a disk array is concerned, it can be treated as a coding problem. The redundancy internal to a disk corrects non-catastrophic failures and identifies catastrophic failures, whereas redundancy at the disk-array level corrects catastrophic disk failures. Codes as simple as parity, which is not a single error-correcting code, can provide single-failure protection because of this internal redundancy and its ability to identify failed disks. Mirroring, the traditional mechanism for single-erasure correction in disk subsystems, has high overhead costs that can be reduced with N+1-parity codes. The characteristics of these N+1-parity codes depend on the organization of user data in the array. Although some self-

[1] This material describes a tutorial, whose slides are included, largely derived from my University of California at Berkeley dissertation, *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*, to be published by MIT Press. This research was funded by NSF grant MIP-8715235, NASA/DARPA grant NAG 2-591, a Computer Measurement Group fellowship, and an IBM predoctoral fellowship.

Metric	IBM 3390	Redundant Disk Array
Disk Units	1	70+7+7
Formatted User Data Capacity (MB)	22,700	22,400
Number of Useful Actuators	12	77
Avg. Access Time (msec)	19.7	19.8
Max. Read I/Os/sec/Box	609	3,889
Max. Write I/Os/sec/Box	609	≥ 972
Max. Transfer Rate (MB/sec)	15	130
Disk Power Consumption (W)	2,900	1,000
Volume for Disks (cubic feet)	97	11
Mean Time To Data Loss (1,000 hours)	50-250	6,600
Component Disk Costs ($1,000)	?	67
Customer Price ($1,000)	156-260	?

Table 1: Comparison of a Strawman Redundant Disk Array to an IBM 3390. *A "strawman" redundant disk array constructed with 84 IBM 0661 model 370 (3½-inch) disks has many advantages over IBM's top-end disk product, the IBM 3390. It has the user capacity of 70 disks; its overhead is 7 disks (10%) for redundant data and 7 disks (10%) for on-line spares. Because parity data is distributed among 77 of the disks and because user data is not stored on spare disks, only 77 disks contribute to its performance. For the maximum I/O accesses per second calculation, the transfer unit is a single sector. For the maximum transfer rate calculation, the transfer unit is a track from every disk that contains user data (77 disks). Most metrics apply to disk components only and may be degraded when controller and host effects are included. The IBM 3390 mean time to failure is not publicly known but can be expected to be better than IBM's previous top-end product, which is reported to have had a mean time to failure of 53,000 hours. To compare costs (based on 1990-1991 data), I show the price a disk array manufacturer would pay for comparable 3½-inch disks from Seagate and the price range that IBM's best customers pay for a maximally configured IBM 3390 and half of an IBM 3990 (disk controller).*

tuning database applications prefer not to automatically stripe data, most disk arrays rely on striping to improve performance by balancing the load across disks and enabling the parallel transfer of large requests. Byte-interleaved striping provides increased transfer bandwidth without increasing access bandwidth in a manner analogous to, but more flexibly than, the way that parallel transfer disks increase transfer but not access bandwidth. In contrast, block-interleaved striping provides both high-transfer and high-access bandwidth at the cost of greater software complexity.

More complex and expensive codes can be used to provide multiple-failure correction in very large or very reliable disk arrays, but these will not be addressed here.

In this tutorial's second section, I review the performance expectations for non-redundant disk arrays. Disk arrays derive their performance advantages by "striping" the data across multiple disks. The greatest benefit of striping is that it decreases transfer times for large requests. In addition, striping automatically distributes independent accesses to balance the workload across disks. Because each disk access involves substantial overhead, the unit of striping must be carefully chosen to avoid a mismatch with the array's workload. A striping unit size with wide success in the absence of workload knowledge is about the capacity of one track. For workloads that emphasize large sequential transfers, byte-interleaved striping with synchronized rotations and seeks offer the largest decreases in response time. However, byte-interleaved organizations have

a much lower throughput for small random accesses. A block-interleaved striping organization provides nearly as low response times and much higher access throughputs as do byte-interleaved organizations.

Redundant data reduces some of the performance benefits of data striping, however, because this redundant data must be updated as user data is updated. In this tutorial's third section I address the performance penalties associated with maintaining redundant data encodings. Without assistance from file system or application software, the main penalty to performance is as little as one and as much as three extra accesses that must be performed with every small, random access. In contrast, with a file system that groups small write accesses into large write accesses, an N+1-parity redundant disk array with block-interleaved striping can provide nearly all of the performance of its disks as well as inexpensive, high reliability. Other, less complete solutions to the performance penalty associated with small random accesses include caching, applications hints, and floating parity organizations. With the performance expectations outlined in these sections and the much lower cost for redundant data, an N+1-parity disk array with block-interleaved striping is the best organization for a single-erasure-correcting redundant disk array.

Finally, before turning to disk array reliability, I discuss the characteristics of disk lifetimes. Although anecdotes of disk failure models abound, little concrete data has been widely published, and there is no consensus among the many vigorously pressed opinions. Yet the distribution of magnetic disk lifetimes is critical to the proper design of failure-tolerant disk systems. To set the stage for an examination of disk array reliability, I offer an analysis of two particular populations of 5¼-inch disks observed over 18 months beginning in 1987. These two populations, totaling 1350 disks, have significantly different lifetime distributions, probably derived from the greater maturity of the manufacturing process for the older of these two disk models. For example, assuming an exponential distribution for lifetimes and ignoring failures during an initial "breaking in" period, the mean time to failure (MTTF) of the newer product is 115,000 hours while the older product has a 368,000 hour MTTF. The appropriateness of an exponential model for disk lifetime distributions is important to the final section's disk array reliability results. The data I present indicates that there is reasonable evidence to indicate that the lifetimes of the more mature of these products can be modeled by an exponential distribution with a mean lifetime of over 200,000 hours. For the less mature of these products, there is evidence that an exponential random variable is too simplistic a model, although it cannot be ruled out.

In the last section of this tutorial, I seek to facilitate the cost-effective design of reliable secondary storage by presenting analytic models of the reliability of redundant disk arrays. The models include a wide spectrum of disk array designs so that individual designers will be able to characterize the reliability of the system they want to build. The most fundamental model considers the effect of independent, random disk failures on an array's data lifetime; it is based on a well-studied Markov model and yields a simple expression for reliability. Another model yields an analytic expression for reliability by solving separate submodels for data loss derived from multiple-disk failure causes such as those induced by sharing interconnect, controller, cooling, and power-supply hardware and concurrent, independent disk-failures. Although N+1-parity protection only insures the correction of a single disk in a parity group, disk arrays can be organized so that each disk in a support-hardware group is contained in a distinct parity group. In this way, dependent disk failures are tolerable because they affect at most one disk per parity group. These models have been validated against a detailed disk-array lifetime simulator for a wide variety of parameter selections. Agreement in most cases is within the simulator's 95% confidence interval.

The models I present in this chapter show that a redundant disk array can easily be designed to provide higher reliability than a single disk. Moreover, with a small overhead for parity and spare disks, a redundant disk array can achieve very high reliability. For some configurations including my strawman configuration, a N+1-parity disk array with on-line spare disks achieves

higher reliability than the more expensive mirrored disk array. As more and more reliability is required of more and more general purpose computer systems, reliability-cost tradeoffs will become critical. The models and design implications discussed in this tutorial will enable secondary storage system designers to achieve reliability goals with cost-effective redundant disk array solutions.

For more information, see:

(1) Peter M. Chen, Garth A. Gibson, Randy H. Katz, David A. Patterson, "An Evaluation of Redundant Arrays of Disks Using an Amdahl 5890," *Proceedings of the 1990 ACM Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Boulder CO, May 1990.

(2) Peter M. Chen, David A. Patterson, "Maximizing Performance in a Striped Disk Array," *Proceedings of the 17th Annual International Symposium of Computer Architecture (SIGARCH)*, Seattle WA, May 1990, pp 322-331.

(3) Ann L. Chervenak, Randy H. Katz, "Performance of a RAID Prototype," *Proceedings of the 1991 ACM Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, May 1991.

(4) Garth A. Gibson, Lisa Hellerstein, Richard M. Karp, Randy H. Katz, David A. Patterson, "Coding Techniques for Handling Failures in Large Disk Arrays," *Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS III)*, Boston MA, April 1989, pp 123-132.

(5) Garth A. Gibson, *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*, PhD dissertation, University of California at Berkeley, UCB/CSD 91/613, 1991. To be published by MIT Press.

(6) Garth A. Gibson, David A. Patterson, "Designing Disk Arrays for High Data Reliability," *Journal of Parallel and Distributed Computing*, to appear, January 1993.

(7) Jim Gray, Bob Horst, Mark Walker, "Parity Striping of Disc Arrays: Low-Cost Reliable Storage with Acceptable Throughput," *Proceedings of the 16th International Conference on Very Large Data Bases (VLDB)*, August 1990, pp 148-161.

(8) Mark Holland, Garth A. Gibson, "Parity Declustering for Continuous Operations in Redundant Disk Arrays," *Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS V)*, October 1992.

(9) Randy H. Katz, G. A. Gibson, D. A. Patterson, "Disk System Architectures for High Performance Computing," *Proceedings of the IEEE*, Volume 77 (12), December 1989, pp 1842-1858.

(10) Michelle Y. Kim, "Synchronized Disk Interleaving," *IEEE Transactions on Computers*, Volume C-35 (11), November 1986.

(11) Edward K. Lee, Randy H. Katz, "Performance Consequences of Parity Placement in Disk Arrays," *Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IV)*, Palo Alto CA, April 1991.

(12) M. Livny, S. Khoshafian, H. Boral, "Multi-disk Management Algorithms," *Proceedings of the 1987 ACM Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, May 1987.

(13) Jai Menon, Jim Kasson, "Methods for Improved Update Performance of Disk Arrays," *Proceedings of the Hawaii International Conference on System Sciences*, 1992.

(14) Richard R. Muntz, John C. S. Lui, "Performance Analysis of Disk Arrays Under Failure," *Proceedings of the 16th International Conference on Very Large Data Bases (VLDB)*, Dennis McLeod, Ron Sacks-Davis, Hans Schek (Eds.), Morgan Kaufmann Publishers, August 1990, pp 162-173.

(15) John K. Ousterhout, Fred Douglis, "Beating the I/O Bottleneck: A Case for Log-Structured File Systems," *ACM Operating Systems Review*, Volume 23 (1), January 1989, pp 11-28.

(16) David A. Patterson, Garth A. Gibson, Randy H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *Proceedings of the 1988 ACM Conference on Management of Data (SIGMOD)*, Chicago IL, June 1988, pp 109-116.

(17) A. L. Narasimha Reddy, Prithviraj Banerjee, "Evaluation of Multiple-Disk Systems," *IEEE Transactions on Computers*, December 1989.

(18) Mendel Rosenblum, John K. Ousterhout, "The Design and Implementation of a Log-Structured File System," *Proceedings of the 13th ACM Symposium on Operating System Principles*, 1991.

Redundant Disk Arrays

Performance and Reliability

Sept 22-24, 1992

Garth A. Gibson

School of Computer Science

Carnegie Mellon University

5000 Forbes Avenue

Pittsburgh, PA 15213-3890

garth.gibson@cs.cmu.edu

412-268-5890

FAX: 412-681-5739

Outline

Motivation

- Technology, performance, design leverage, cost
- Market activity

Non-Redundant Disk Array Performance

- Striping for concurrency or parallel transfer
- Selecting stripe unit size

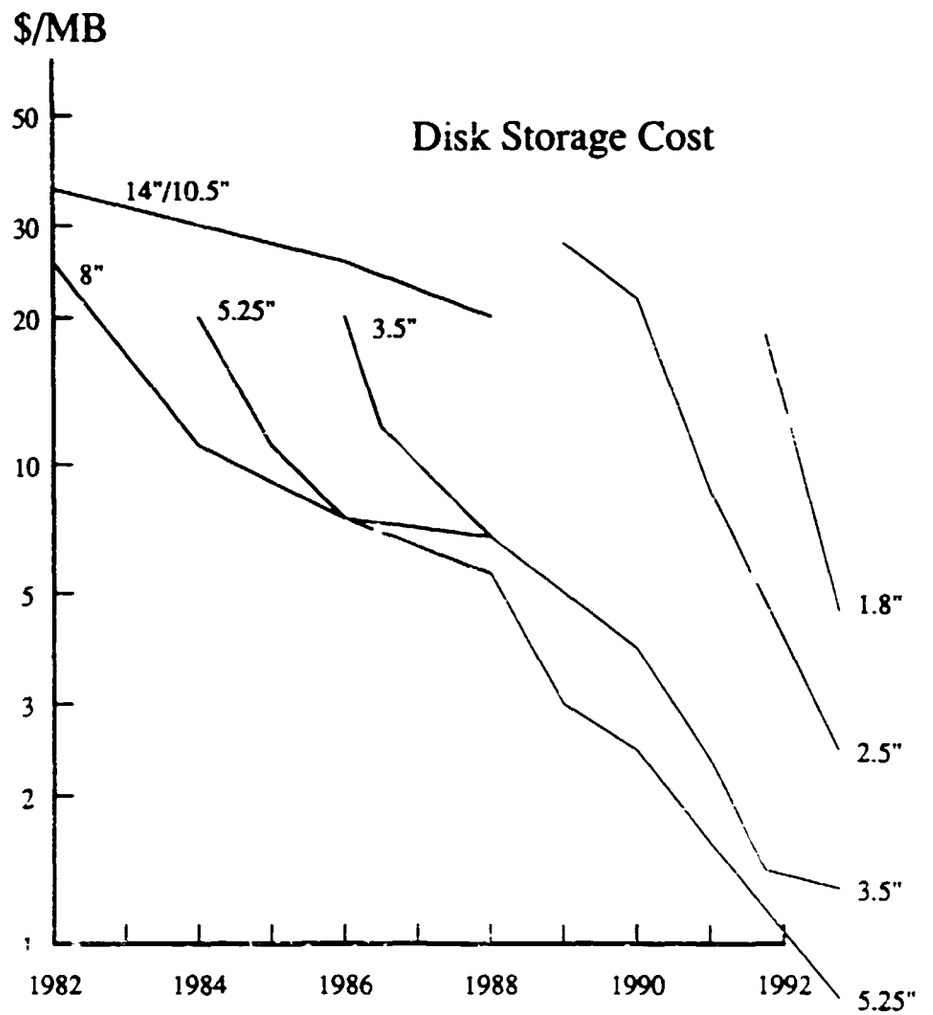
Redundant Disk Array Performance

- Taxonomy and fundamental performance
- Small write problem and solutions
- Online reconstruction performance

Redundant Disk Array Reliability

- Disk failures: lifetime data
- Independent failures models
- Dependent failures models
- Simulation results

Trends in Disk Capacity Cost



extended from A. Vasudeva, Fujitsu, SDNC, Apr 88.

3.5": $/MB down 26% per year, 61% in 3 years

Little Disks Are Better

Driven by personal computer and laptop market

Much larger market -- lower profit margin

- more R&D amortized and required

Inherent advantages

- lower mass to spin, to seek
- cooler operation
- higher reasonant frequencies
- ⇒ tighter design tolerances
- shorter stroke to seek

Trends Aggravate I/O Effects

VLSI and multiprocessing trends

⇒ 50 - 100 % / year for processors

Gordon Bell (CACM) predicts

⇒ 150 % / year for supercomputers

but magnetic disk performance lags

< 5 % / year access rate

< 4 % / year data rate

by Amdahl's law for unequal speedups:

processor utilization decreases

⇒ I/O bottlenecks performance

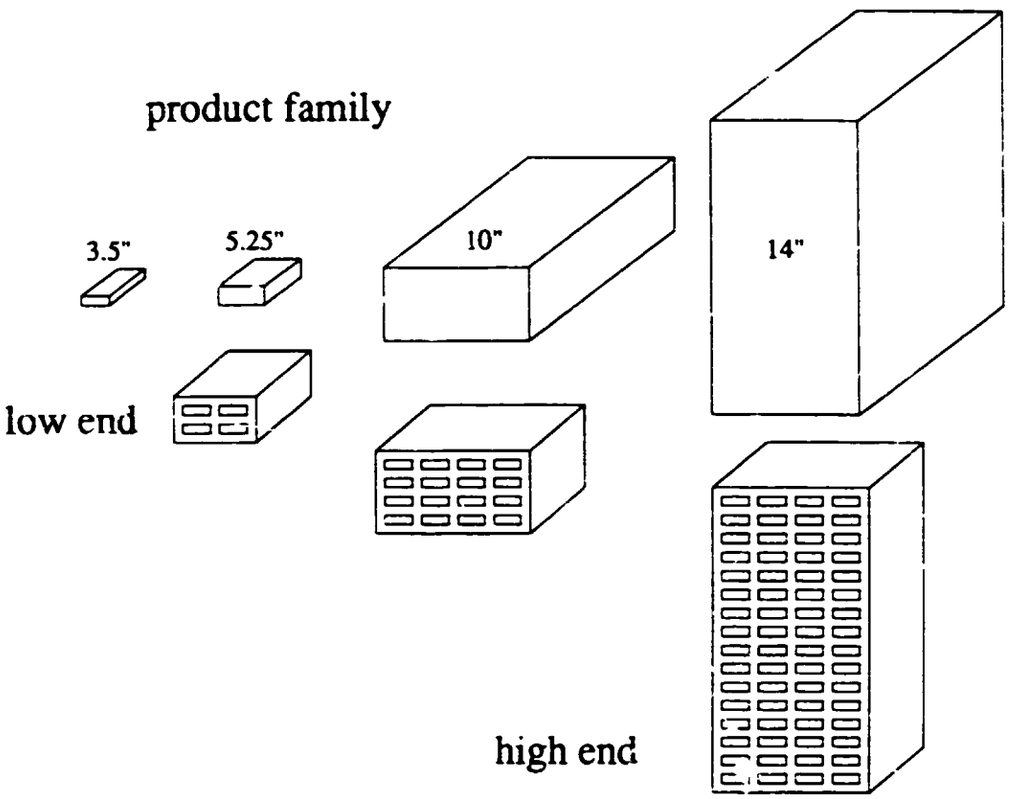
Parallelism via Arrays

analogue to multi-microprocessors

	Single IBM 3380 K (14")	*3390 (11")*	Disk Array 50 IBM 0661s (3.5")	*70*	
Capacity (GB)	7.5	*22.7*	16	*22.4*	2X *1X*
Actuators	4	*12*	50	*70*	12X *6X*
Peak IO/s	200	*600*	2500	*3500*	12X *6X*
Peak MB/s	12	*16.8*	85	*140*	7X *8X*

⇒ *order of magnitude gains possible*

Manufacturing Advantages



conventional: 4 disk design teams

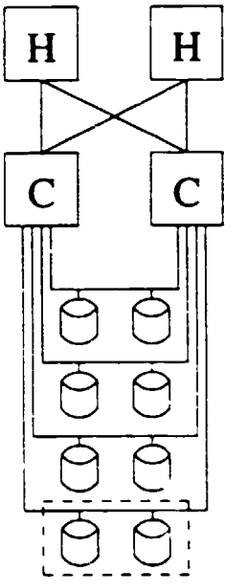
disk array: 1 disk design team

Less Expensive

High Reliability

parity protected

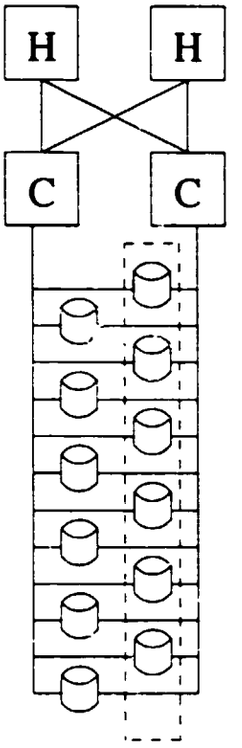
33% overhead



parity disks

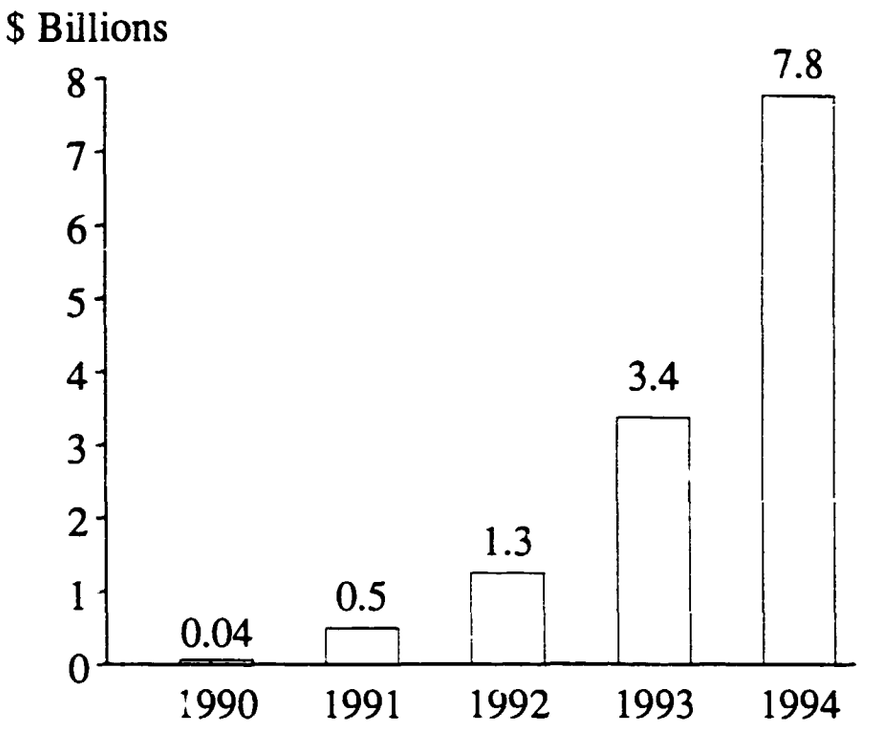
shadowing or mirroring

100% overhead



shadow disks

Worldwide Array Market Estimates



	1990	1991	1992	1993	1994
IBM Mainframe	0	0.115	0.240	1.100	3.380
IBM AS/400	0	0.055	0.224	0.620	1.085
DEC VAX	0	0.002	0.070	0.120	0.780
Other Minis	0	0.013	0.130	0.330	0.560
Scientific	0.010	0.030	0.075	0.250	0.450
PC/LAN server	0.015	0.227	0.410	0.655	0.980
Unix/Net Server	0.015	0.037	0.120	0.275	0.530
Total	0.04	0.480	1.269	3.350	7.765

Source: Montgomery Securities, Dec 91

Recap

CPU performance trends

disk technology trends

(redundant) disk arrays

low cost reliability

broad product family

rapid market, product, and research growth

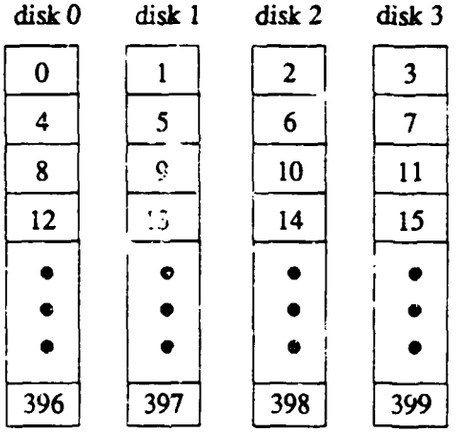
Basic Performance

Many Actuators

⇒ many IO/s if disk load is balanced

⇒ many MB/s if transfer is in parallel

Data Striping



small random access

⇒ uniform disk load

large sequential access

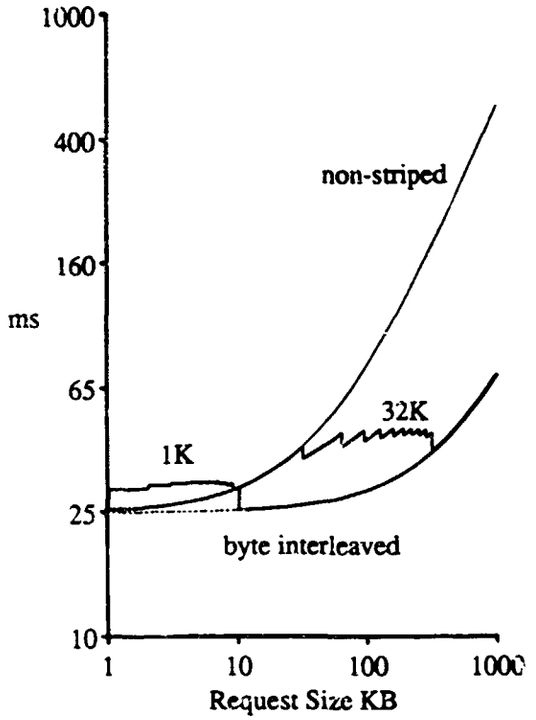
⇒ parallel transfer

Livny, Sigmetrics 87

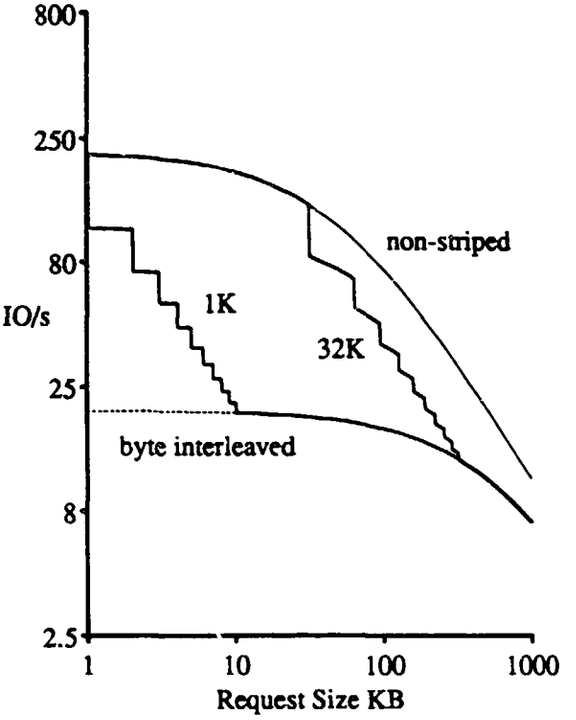
Striped Disk Array Performance

10 perfectly evenly loaded, synch disks

Response Time
at low loads



Throughput
at 50% utilization



derived from Jim Gray, VLDB'90

unit of striping is important!

Striping Unit Size

Benefit

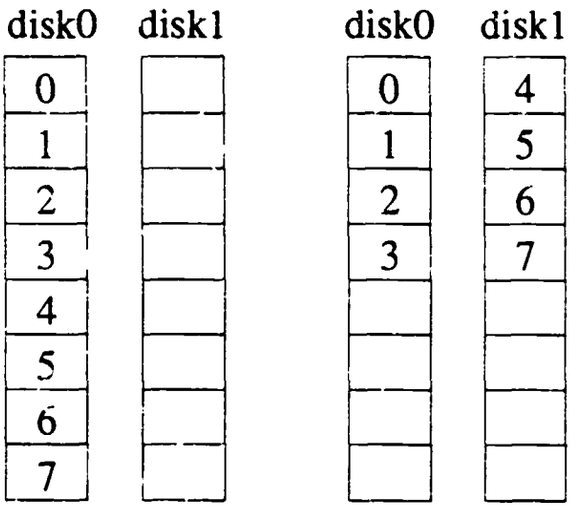
decreased transfer time

(stripe unit / transfer rate)

Penalty

additional seek + rotate

(average seek + rotate)



Goal: rules of thumb

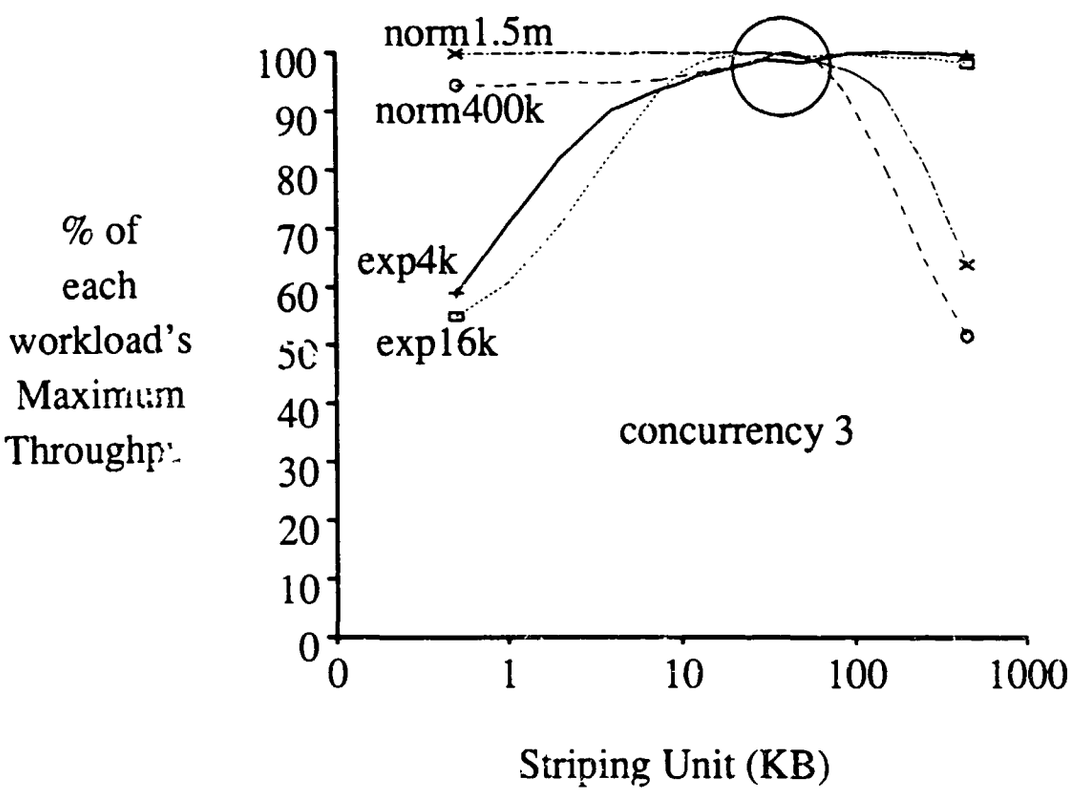
Metric: throughput

Experiment (Peter Chen, SigArch90)

16 synch disk simulator

stochastic workload

Known Concurrency Workload

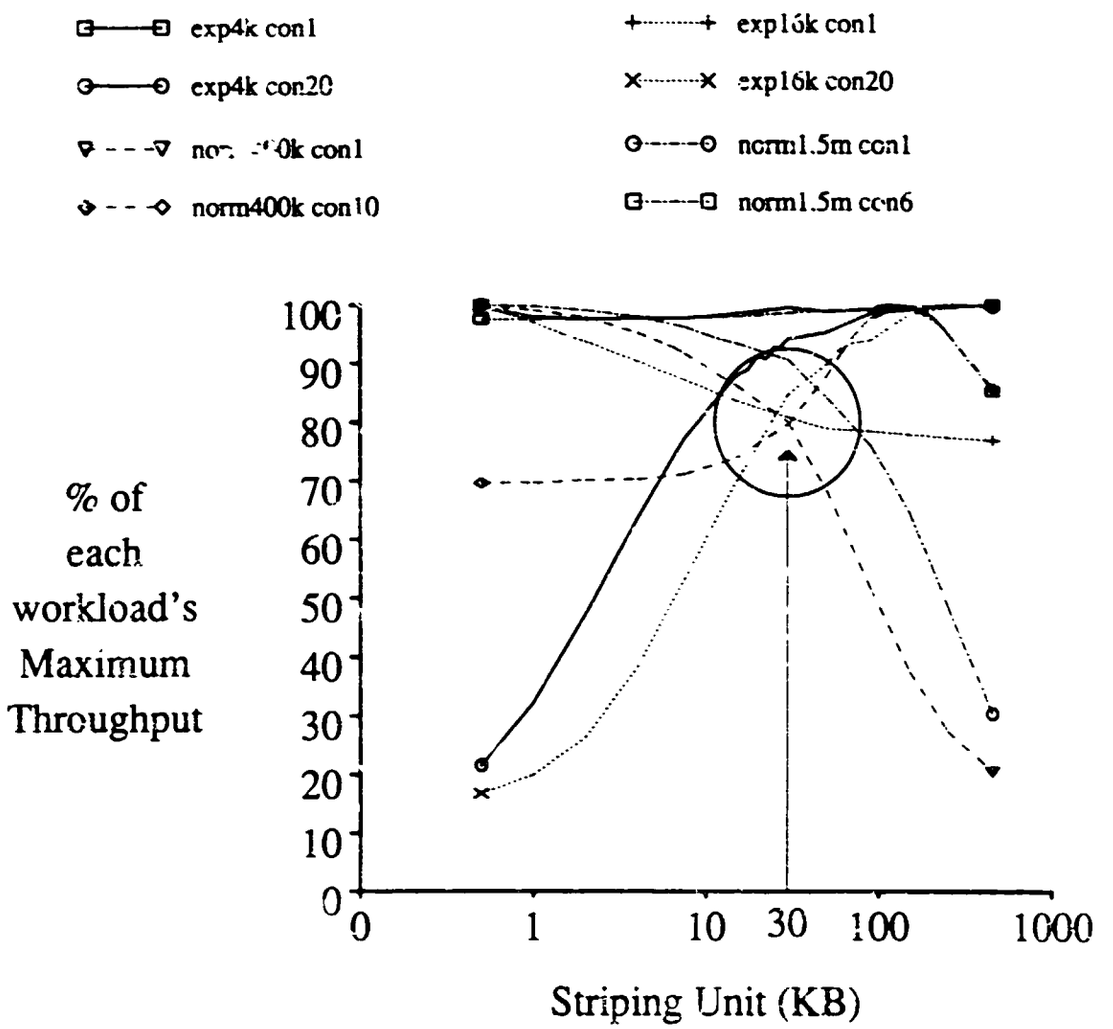


Striping Unit = Slope x (concurrency - 1) + 1 sector

Slope = S x Positioning Time x Transfer Rate

(S = 1/4 for 16 disks)

"Zero" Workload Knowledge



Striping Unit =

2/3 x Positioning Time x Transfer Rate

Striping Performance Recap

disk array has many actuators

striping utilizes parallelism

balances disk loads

provides parallel transfer

striping unit sensitive to workload

workload concurrency is most important

rules of thumb depend on simple disk specs

The Catch is Data Losses

more parallelism

⇒ more components

⇒ more frequent failures

70 disks, each 150 Khour MTTF

exponential disk lifetimes

mean time to data loss falls

17 years to 89 days

⇓

Redundant Arrays of

Inexpensive Disks (RAIDs)

Organization Taxonomy

Patterson, Gibson, Katz, Sigmod 88

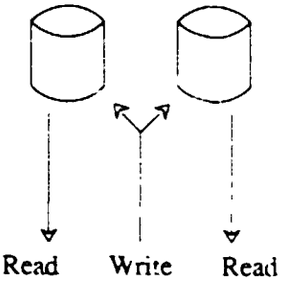
- redundancy organization ? effect on performance ?
- using simple deterministic approximations

  based on average access times

RAID 1. Mirroring - Tandem

- replicate all data
- 100 % overhead
- groups of 2
- write to both

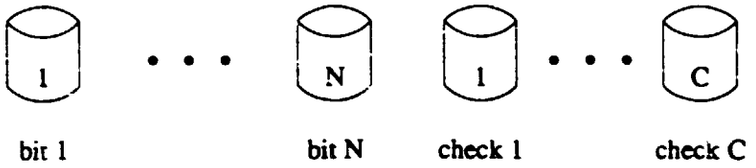
  read from either



\+ uses all bandwidth for reads

\- but only half bandwidth on writes

RAID 2. Bit Interleaved with Hamming Code
Connection Machine's DataVault



- check 1..C is Hamming code of bit 1..N

word size N	check bits C	overhead
8	4	50 %
16	5	31 %
32	6	19 %

\+ lower overhead

\+ better large write bandwidth

\+ soft error correction on the fly

\- only 1 IO at a time across N+C disks

\- unit of access is N times bigger

$\Rightarrow$ small writes must preread, merge,

then overwrite all N+C disks

RAID 3. Bit Interleaved with Parity
Maximum Strategy's Strategy 2

- parity (C=1) is a single error detect code

  but disk controller identifies failed disk

⇒ parity allows single error correction

after a disk failure, bitwise test:

parity(good disks) = stored parity ?

if so, lost bit is 0, else 1

\+ still lower overhead

\- same small access problems

\- same reduced parallelism

(so RAID 2 can do double error correction)

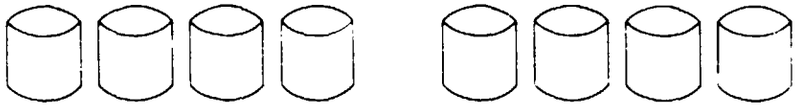
RAID 5. Block Interleaved with Rotated Parity Array Technology's RAID+

- remap bits to disks so logical sector is all on a single disk

⇒ small reads require only 1 disk

⇒ small writes require 4 IOs since each data bit toggled requires corresponding parity be toggled

⇒ parallel small writes block on parity disk so spread parity across all disks



d0	d1	d2	**p0**
d3	d4	d5	**p1**
d6	d7	d8	**p2**
d9	d10	d11	**p3**

RAID 4.

d0	d1	d2	**p0**
d4	d5	**p1**	d3
d8	**p2**	d6	d7
p3	d9	d10	d11

RAID 5.

\+ same low overhead

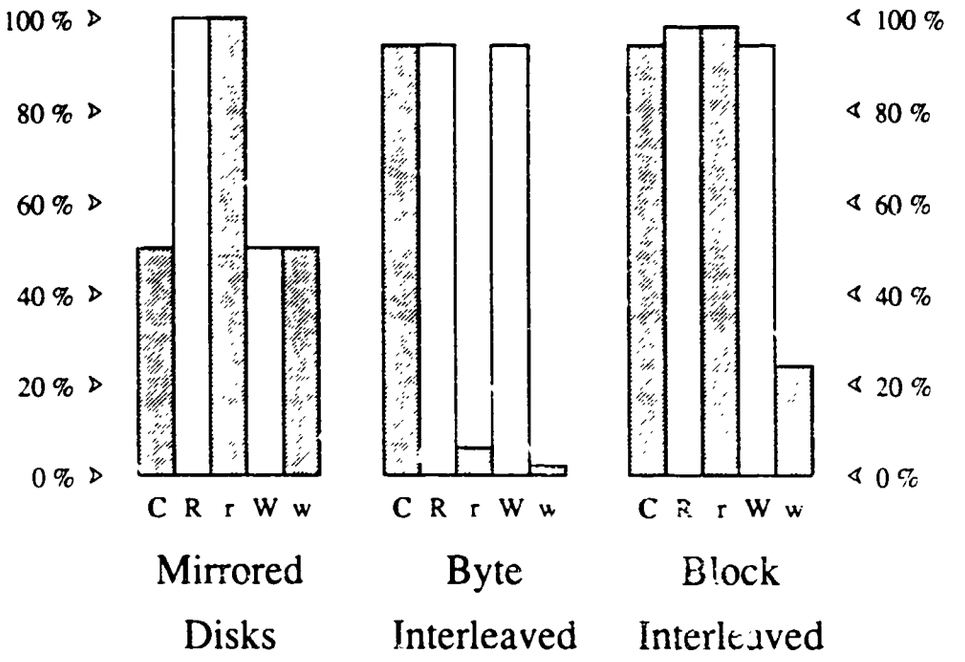
\+ almost full large access bandwidth

\+ full small read bandwidth

\- small write bandwidth is half mirroring

Back of the Envelope
Maximum Throughput

relative to 16 disk non-redundant array



C : User Capacity

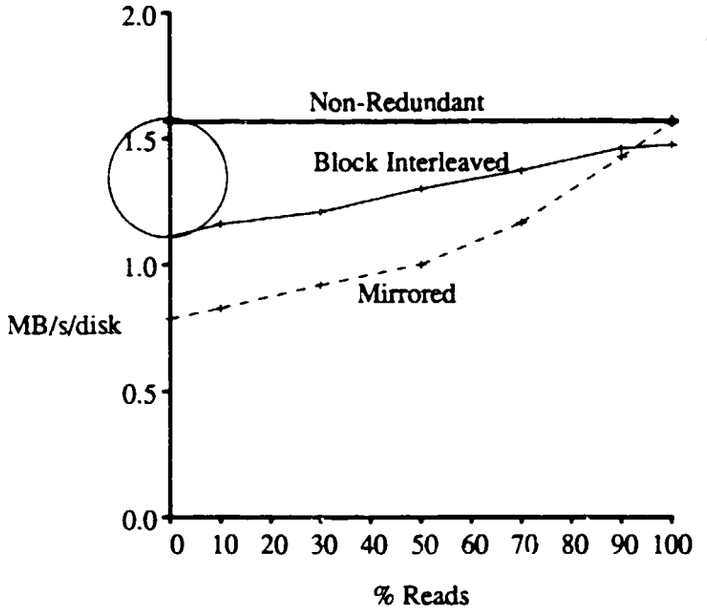
R,W : large reads, writes

r,w : individual reads, writes

Amdahl Measurements

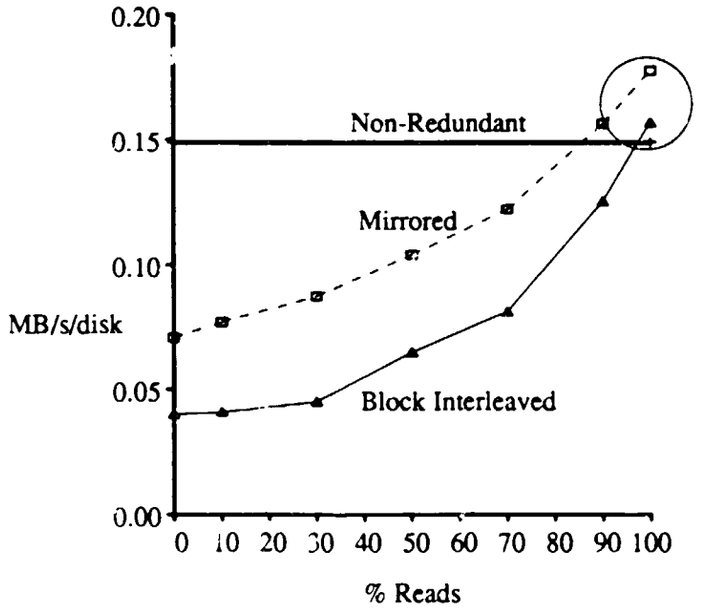
P Chen, Sigmetrics'90

20 Amdahl 6380s



Large Accesses

1.5 MB average



Small Accesses

6 KB average

Floating Parity Allocation

Menon, Hawaii Syst Sci 92

Small write problem is 4 accesses per write

preread and overwrite of data and parity

Dynamically reallocate parity each overwrite

overwrite takes 10%-20% rev vs 100% rev

⇒ preread and overwrite of parity in "1" access time

With 1 free track per cylinder (15 tracks) of parity

average distance to overwrite block is 1.6 blocks

Transaction processing: data preread hits in cache

⇒ small writes take 2 accesses: equal to mirroring !!

Log-Structured File System

Rosenblum, SOSP 91

Large, writeback file caches

⇒ dominant traffic is writes

Treat disk (array) as log; write only end of log

⇒ no seeks during writeback of many files

Delayed writeback

⇒ Group small writes into large writes

Small writes only when very idle

Log wrap around requires compaction

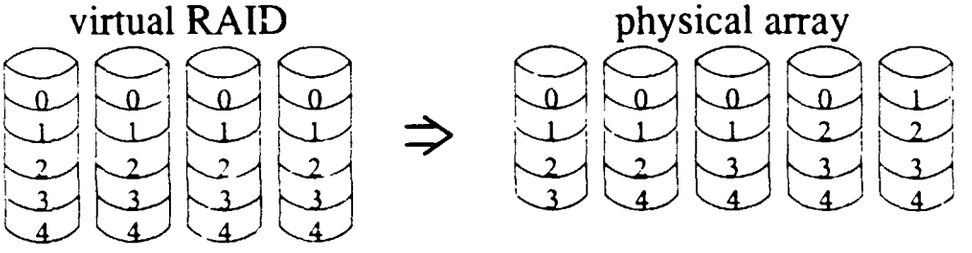
cost-benefit selection of region to compact

⇒ Sprite implementation experience

50% - 85% compacts on empty region

Parity Declustering for Reconstruction

Muntz, VLDB 90, and Holland, CMU TR 92



reconstruction reads 100% of remaining 3 disks

reconstruction reads 75% of remaining 4 disks

mapping uses balanced incomplete block designs

⇒ faster reconstruction and/or
faster user access during reconstruction

smart (work reducing) algorithms lose
they cause excess seeks on replacement disk

Recap

rapidly improving compute speeds

⇓

smaller but not faster disks

⇓

striped disk arrays for performance

⇓

increased failure rates

⇙ ⇘

mirroring

+ small IO/s

N+1 parity = RAID 5

+ large IO/s
+ low cost

⇓

floating parity and

log-structured file systems

⇓

parity declustering

Disk Lifetime Data

collected Jan 89 through June 90

two populations of 5.25" disks

1) 859 disks, 350 MB
2) 523 disks, 200 MB

DOAs, customer burn-in failures, field failures

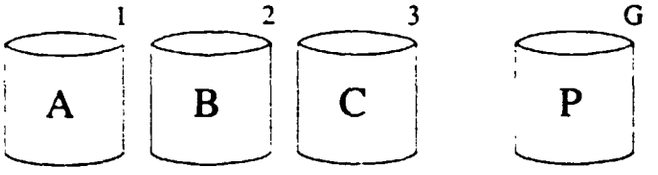
fit to Weibull lifetime distribution model

if shape is 1.0, lifetime is exponential

	95% conf. int. on Weibull shape	exponential mle MTTF-disk
1)	0.59 - 1.04	80,000 hr
2)	0.62 - 1.30	338,000 hr

use exponential lifetime distribution model

Reliability Modeling

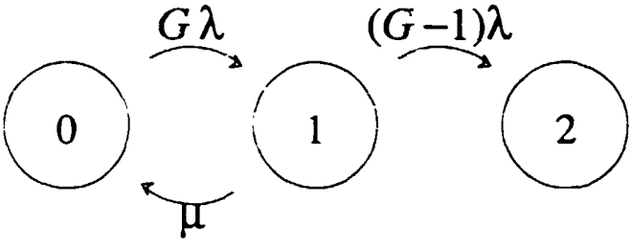


P = XOR(A,B,C)

C = XOR(A,B,P)

disk failure rate: λ = 1/MTTF-disk

disk repair rate: μ = 1/MTTR-disk



MTTF-disk >> MTTR-disk

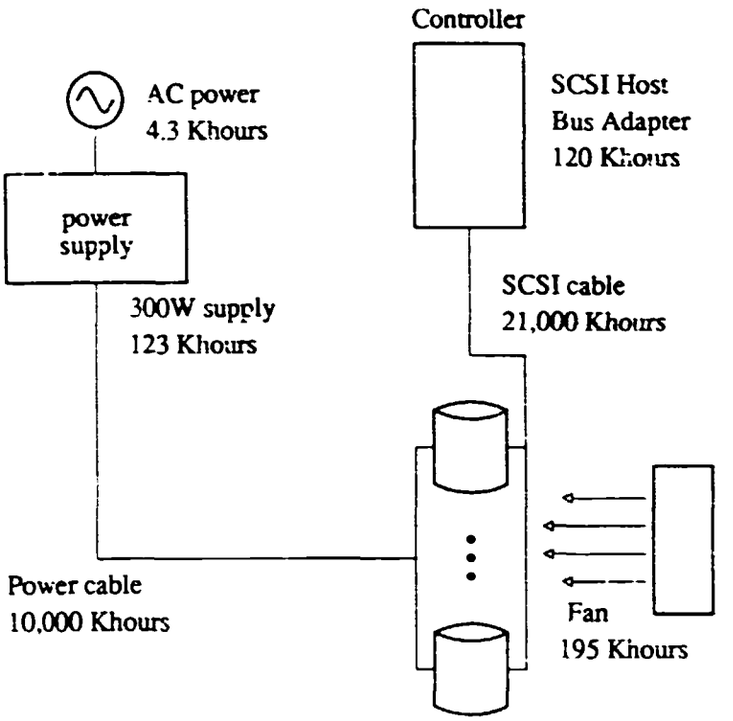
$$\text{MTTDL-RAID} = \frac{\text{MTTF-disk}^2}{\text{N G (G-1) MTTR-disk}}$$

7 groups (N) of 10+1 (G) disks

MTTF-disk = 150 Khr

MTTR-disk	2 week	3 day	1 day	4 hr	1 hr
MTTDL_RAID (Mhr)	0.087	0.406	1.2	7.3	29

Disk Support Hardware



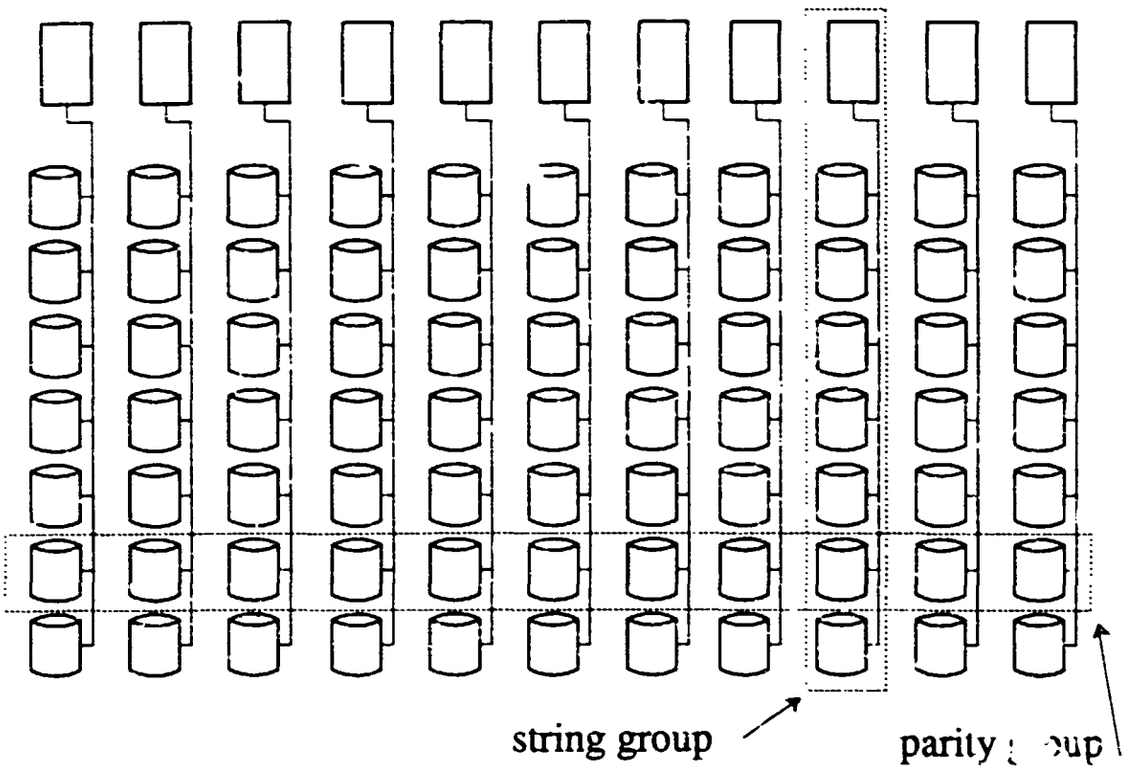
power source critical

non-disk, non-AC

46 Khour MTTF-string

⇒ MTTDL-RAID < MTTF-disk ?

Orthogonal Parity Groups



strings have separate power, cooling, cabling

string failure is one disk per parity group

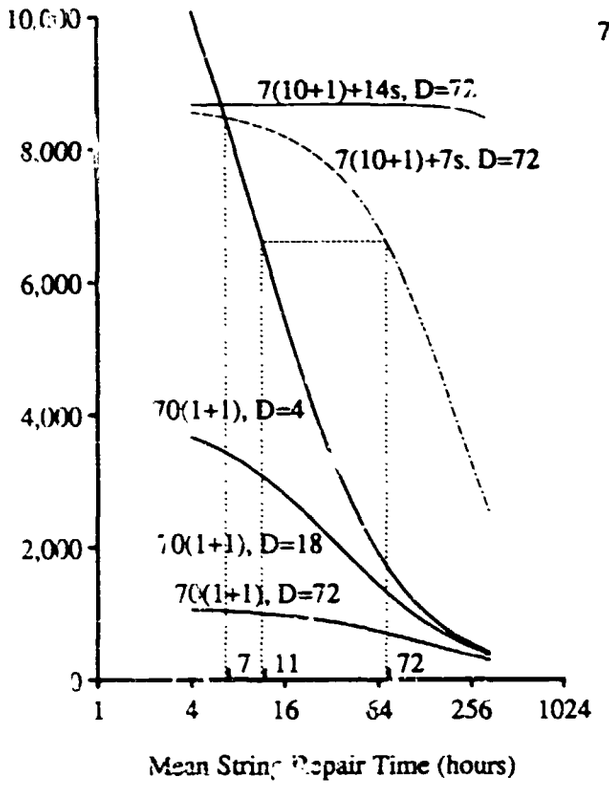
150 Khr MTTF-string & 3 day MTTR-string

MTTR-disk	1 hr	4 hr	3 days
MTTDL-RAID	356 Khr	331 Khr	132 Khr

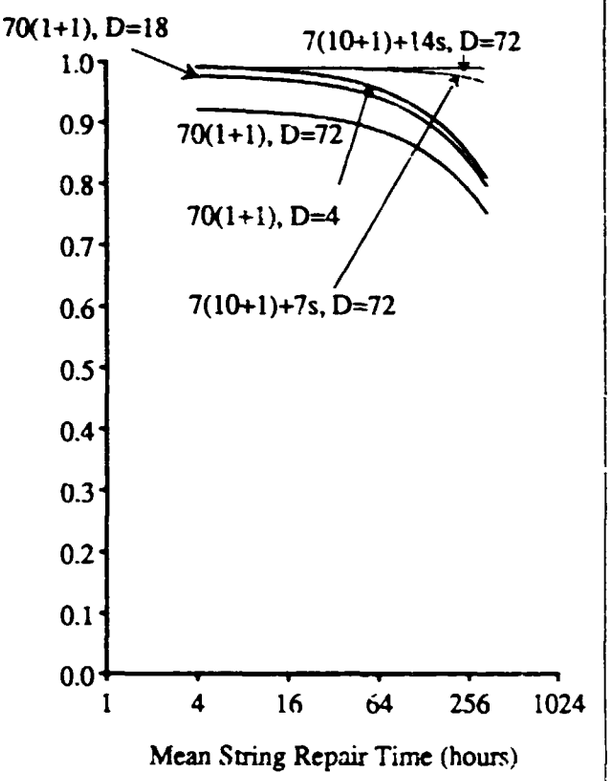
Reliability Comparison

Mirror Disks vs N+1+Spares Array

Mean Data Lifetime
(1000 hours)



10yr Reliability



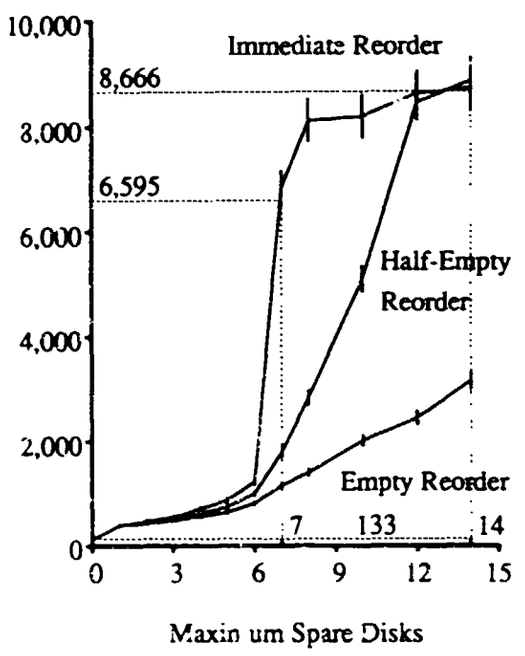
mean disk and string lifetimes = 150,000 hrs

mean disk recovery time = 1 hr, immediate reorder

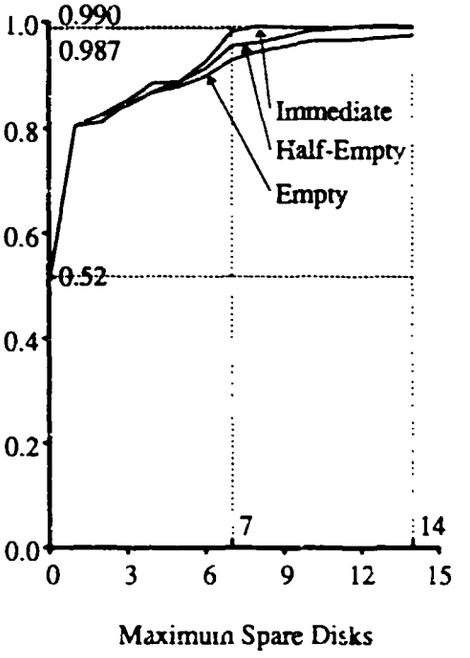
Delayed Reorder and Partial Strings of Spares

Mean Data Lifetime

(1000 hours)



10yr Reliability



1 spare is effective, 12 spares are very effective

mean disk and string lifetime = 150,000 hrs

disk recovery time = 1 hr, disk delivery time = 72 hrs

mean string repair time = 72 hrs

Recap Reliability

disk lifetime distribution approx exponential

⇓

independent disk failures greatly overcome

⇓

faster repair using spare disks

⇓

support hardware failures devastating

⇓

orthogonal RAID against string failures

⇓

fast repair and spares very cost effective

⇓

RAIDs achieve high data reliability

but, watch out for poor reconstruction coverage

Summary

technology pushing disk arrays

striping utilizes parallelism for performance

redundancy required for reliability goals

N+1 parity is cost effective RAID

orthogonal RAID w/ spares highly reliable

Topics not covered

workstation/network architecture for arrays

on-line data compression

double+ failure correction

N93-80464

Striped Tertiary Storage Arrays

Ann L. Drapeau
Computer Science Department
University of California
571 Evans Hall
Berkeley, CA 94720

515-82
157.05
P-11

1 Introduction to Striping

Data striping is a technique for increasing the throughput and reducing the response time of large accesses to a storage system [12], [7], [8], [4]. In striped magnetic or optical disk arrays, a single file is striped or interleaved across several disks; in a striped tape system, files are interleaved across tape cartridges. Because a striped file can be accessed by several disk drives or tape readers in parallel, the sustained bandwidth to the file is greater than in non-striped systems, where accesses to the file are restricted to a single device.

Gibson [4] gives an excellent discussion of striping in magnetic disk arrays, much of which can be generalized to optical disk and magnetic tape arrays. Two methods of striping data are byte-interleaved and block-interleaved striping. In a byte-interleaved system, files are interleaved a byte at a time across the collection of disks or cartridges that make up a "stripe". In such a system, each device or cartridge in the stripe will be involved in every access. This makes synchronization of the devices easy, but does not allow any parallelism among the drives or readers in the stripe.

In a block-interleaved system, data interleaving is done in larger increments. The size of the interleaved block may be chosen to optimize sustained bandwidth (as done by Chen and Patterson for disk arrays [2]) or to minimize response time. In a block-interleaved system, several accesses to a stripe may occur in parallel if the individual accesses are small enough that they don't involve all the disks or cartridges in the stripe. This potential parallelism is an advantage of block-interleaved systems over byte-interleaved systems. This advantage may be offset, however, by increased latency penalties; drives acting independently will become unsynchronized, and subsequent large accesses involving several drives or cartridges will have to wait for the completion of the slowest device. Unless the devices in a stripe are kept strictly synchronized, a striped system will have longer positioning latencies than a non-striped system.

Failures are more frequent in systems with many components. In large storage arrays, potential failures include transient media errors, media wear, head failure, other mechanical problems with the device, and breakdown of

controllers, power supplies or cables [13]. To ensure adequate reliability of storage arrays, some form or error correction encoding must be maintained in the array. Although it is not necessary to perform striping to include such redundancy information [6], it is convenient to calculate error correction codes over a stripe.

The choice of an error correcting code for a storage array is based on its ability to protect the data against likely errors and on minimizing the impact of the code on the performance and capacity of the array [4]. Performance of write operations is affected by the addition of ECC, since extra redundancy calculations and extra write operations to store the error correction information must be performed. Also, the choice of ECC will affect performance when data is being reconstructed after a disk or cartridge failure. The ECC chosen will also affect the amount of useful data storage on the array, since redundancy information must be stored in place of other data.

Gibson [4] showed that for disk arrays, single bit parity provides good data reliability as long as sufficient empty or "spare" space is left in the array for reconstructing data in the event of disk failures. We will perform a similar reliability analysis for tape arrays. Our intuition is that tape arrays will require more redundancy than simple parity. As will be discussed in detail in Section 3, magnetic tape systems face more difficult reliability challenges than disk drives. Media and head wear problems as well as the occurrence of errors uncorrectable by ECC make it likely that errors will occur more frequently in large tape systems than in disk arrays. It is likely that a more powerful error correcting scheme than simple parity will be needed to protect against these errors.

In the sections that follow, we argue that applying striping to tertiary storage systems will provide needed performance and reliability benefits. Section 2 will discuss the performance benefits of striping for applications using large tertiary storage systems. It will introduce commonly available tape drives and libraries, and discuss their performance limitations, especially focusing on the long latency of tape accesses. This section will also describe an event-driven tertiary storage array simulator that we are using to understand the best ways of configuring these storage arrays. Section 3 will discuss the reliability problems of magnetic tape devices, and describe our plans for modeling the overall reliability of striped tertiary storage arrays to identify the amount of error correction required. Finally, Section 4 will discuss work being done by other members of the Sequoia group to address latency of accesses, optimizing tertiary storage arrays that perform mostly writes, and compression.

2 Striping for Performance

In this section, we argue that striping is needed in large tertiary storage systems because a growing number of applications require tertiary storage systems with high sustained throughput. Striped systems will provide this throughput better than currently available devices and libraries can. Examples of applications requiring high sustained throughput (up to hundreds of Megabytes per second) include those that use traditional archival systems to store results

of large calculations, satellite and seismic data, and records of financial institutions. They also include applications using large amounts of video, and library applications that try to give a user acceptable response time on queries of large data sets.

2.1 Tertiary Storage Devices

Currently available tertiary storage devices don't offer high sustained throughput. Figure 1 shows some of the magnetic tape drives and one magneto-optical disk currently on the market. It compares their capacity, bandwidth and approximate drive cost. The magnetic tape drives can be divided into helical scan recording and linear recording devices. Of the helical scan devices, the 4mm DAT and 8mm technologies are low cost and high capacity. However, they have quite low bandwidth (0.5 MBytes/sec or less). In addition, like all the magnetic tape devices, access time (that is, time to position the tape and read or write a particular bit on the tape) is long. Access time will be discussed further below. In the mid-range of cost for helical scan drives is the Metrum VLDS technology. This device has good capacity (15 GBytes/cartridge) but its bandwidth (1.2 MBytes/sec) is only a small improvement over the inexpensive drives. The graph also shows the 19mm DD2 technology, which is very expensive, but provides the best capacity and bandwidth (125 GBytes/cartridge and 15 MBytes/sec). It should be noted that even this device is incapable of supporting the high bandwidth (100 MBytes/sec or more) required by many applications without striping. Of the linear recording technologies, the 1/4" is inexpensive and high capacity but suffers from low bandwidth. The mid-priced 1/2" IBM 3490 technology has low capacity (480 MBytes/cartridge) but moderate bandwidth (6 MBytes/sec). Finally, the graph shows a 5.25" magneto-optical disk that is fairly low cost. The disk is lower in capacity than the tape drives and transfers at a fairly low rate (1.25 MBytes/sec). However, the access time on the magneto-optical drive is shorter than that of the magnetic tape drives by several orders of magnitude.

These drives offer a wide range of performance and capacity. However, none of the drives can sustain bandwidth in the range of hundreds of Megabytes per second. In traditional archival systems, it is possible to get near the specification of sustained bandwidth for a particular device, since large files are written in their entirety and seldom re-read. In library applications, where accesses to the tertiary storage system are likely to be fairly random, maintaining high sustained throughput is more difficult, since tapes will be switched often. As will be described in the next section, access times on the drives are quite long: a minute or more for ejecting an old tape, loading a new tape and positioning in preparation for data transfer. In systems (like libraries) where random accesses occur, sustained throughput will be lower than the specified maximum for the drives. Striping will be particularly important in these systems to sustain reasonable throughput.

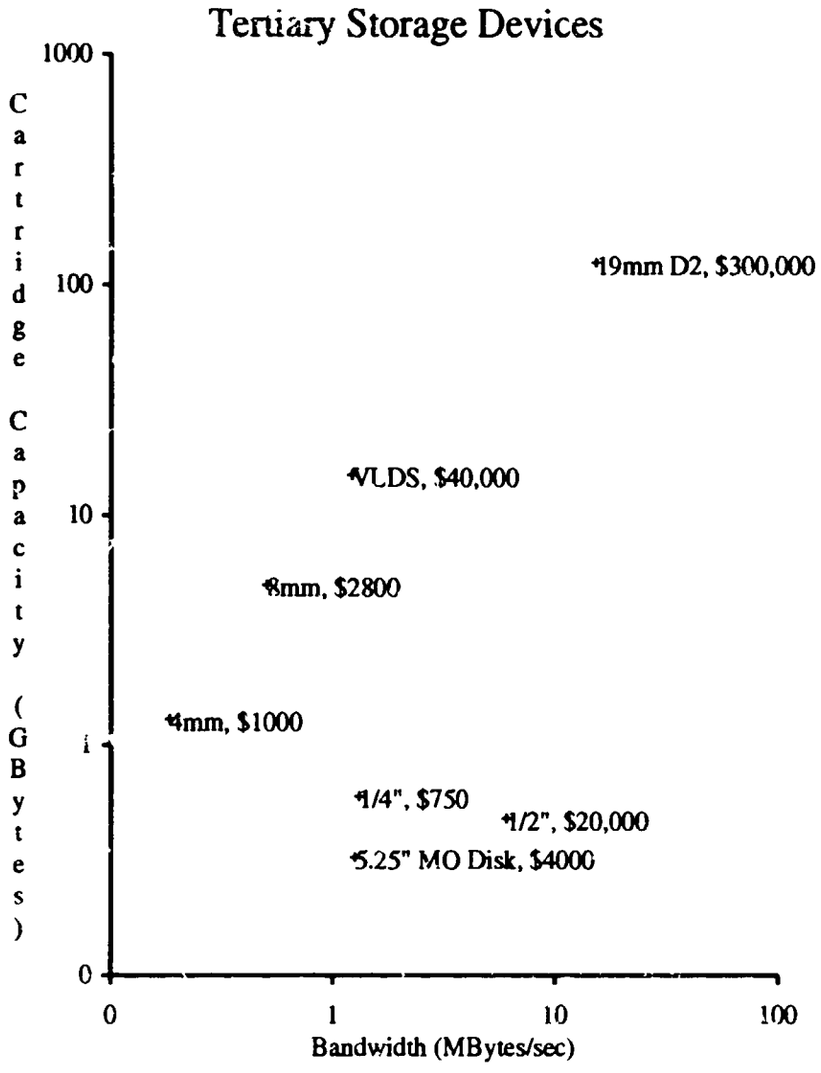


Figure 1: *Drive capacity and bandwidth for magnetic tape and magneto-optical disk products.*

Operation	4mm DAT	8mm Exabyte	0.5" Metrum
Mean load time (sec)	16	35.4	28.3
Mean eject time (sec)	17.3	16.5	3.8
pRewind startup time (sec)	15.5	23	15
Rewind rate (MB/sec)	23.1	42.0	350
Search startup time (sec)	8	12.5	28
Search rate (MB/sec)	23.7	36.2	115
Read transfer rate (MB/sec)	0.17	0.47	1.2
Write transfer rate (MB/sec)	0.17	0.48	1.2

Table 1: *Measurements of 4mm, 8mm and 0.5" helical scan magnetic tape drives.*

2.2 Tape Drive Measurements

In order to better understand tape devices and robots, we performed detailed measurements of their operation. Measurements were made for three devices: an 8mm Exabyte drive, a 4mm WangDAT drive and a 0.5" VLDS Metrum drive. All three are helical scan magnetic tape drives.

Table 1 summarizes the performance measurements made on the individual tape drives The first two operations are mechanical: loading a tape into a drive and ejecting a tape from drive. These operations are quite slow; part of the reason these operations for this is the fairly complex mechanical manipulation of the tape in a helical scan system. On a load operation, additional time is spent reading format information from the start of the tape. On each of the three devices, the combination of a load and an eject operation takes at least 30 seconds.

Rewind and forward search behavior on each of the devices can be characterized as fairly linear after an initial startup time. Table 1 shows the startup time and rewind and search rates for each device. Measurements of search and rewind rates were made for tapes written entirely with 10 MByte files.

Finally, Table 1 shows the sustained read and write rates to a user process measured for each of the drives. In each case, the read and write bandwidth obtained are close to specifications, but the drive can easily perform much worse than this optimum. The devices must be kept streaming in order to achieve good bandwidth. Otherwise, in order to avoid tape and head wear, the drive will initially pull the head away from the tape and loosen tape tension, and then after a few more seconds of no activity, the drum will stop spinning. Later accesses will require spinning up the drum and reapplying tape tension.

Table 2 shows an important access time parameter, the "average seek" time, or the time to search over 1/3 of the volume of the tape. Such an average seek time may correspond poorly to actual workloads, but we will use it as

Device	1/3 tape volume (GBytes)	Time (sec)
4mm DAT	.400	25
8mm EXB8500	1.5	54
Metrum VLDS	5	70

Table 2: *Average seek times for each 4mm DAT, 8mm EXB8500 and Metrum VLDS drives, where the average seek is defined as being the time to search over 1/3 the volume of the tape cartridge.*

an initial basis for comparison.

It should be noted from the tape drive measurements presented above that the access times on these devices are very long. An access requiring a tape switch will include a rewind of some portion of the old cartridge, an eject operation, two robot operations to remove one cartridge and grab the new cartridge, a load of the new cartridge and a search to position the new cartridge for data transfer. (The timing of robot operations will be discussed in the next section.) Given the long mechanical delays and search times for the tape devices measured here, an access that includes a cartridge switch can have a latency of several minutes. Tertiary storage arrays for applications that will perform random accessing of data must be carefully configured to attempt to overcome these serious latency problems. Section 2.4 will discuss our efforts to understand how best to configure tertiary storage arrays.

2.3 Automatic Handling of Cartridges

To provide higher bandwidth and capacity than can be supplied by a single device, several companies have built automated library systems that hold tens or thousands of cartridges that can be loaded by robots into some number of magnetic tape or optical disk drives. Using several drives in a library increases the aggregate bandwidth; however, the bandwidth to or from an individual cartridge does not change. If files are restricted to a single data cartridge, the bandwidth to a particular file is limited to the bandwidth supported by a single device. Striping within or between these automated libraries removes this limitation on bandwidth to a single file by spreading accesses to the file across several devices.

Table 3 shows a classification of some of the robots available for handling magnetic tape cartridges and optical disk platters automatically. Large libraries generally contain many cartridges, several drives and one or two robot arms for picking and placing cartridges. The cartridges are often arranged in a rectangular array. Other "large library" configurations include a hexagonal "silo" with cartridges and readers along the walls, and a library consisting of several cylindrical columns holding cartridges that rotate to position them. Usually these large libraries are quite expensive ($500,000 or more), but they often have low cost per MByte compared to less expensive robotic devices. Carousel devices are moderately priced (around $40,000) and hold around 50 cartridges. The carousel rotates to

Type	No. Cartridges	No. Drives	No. Robot Arms	Cost
Large Library	10 to 1000	several	one or two	high
Carousel	around 50	one or two	one (carousel)	moderate
Stacker	around 10	one	one (magazine or arm)	low

Table 3: *Devices for handling tertiary storage cartridges automatically.*

Time to grab cartridge from drive	19.2 sec
Time to push cartridge into drive	21.4 sec

Table 4: *Times for robot to grab a cartridge from a drive and push a cartridge into a drive for the EXB-120 robot system.*

position the cartridge over a drive, and a robot arm pushes the cartridge into the drive. In most cases, there are one or two drives per carousel. Finally, the least expensive device ($10,000 or less) is a stacker, which holds around 10 cartridges in a magazine and loads a single reader. The magazine may move vertically or horizontally to position a tape in front of the drive slot, or the stacker may have a robot arm which moves across the magazine to pick a cartridge.

In order to develop a model for robot access time that could be included in our performance simulations, which will be described in the next section, we measured an Exabyte EXB-120 robot. This robot is a simple rectangular array of 116 cartridges and four tape drives. We measured robot arm movement time from various positions in the array. We found that robot arm movement varied between 1 and 2 seconds in the array. Since this time is so small compared to the latencies of tape accesses, we are modeling this as a constant value. We also measured the time to grab a cartridge from a drive as well as the time to push a cartridge into a drive. Table 4 shows these latter two measurements, both around 20 seconds.

2.4 Performance Simulations

To better understand how to configure striped tertiary storage arrays, we have written an event-driven array simulator. This simulator uses performance models for tape devices, optical disk drives and robots that are based on the device and robot measurements described in the previous sections. The simulator takes as input a set of parameters that are applied to general performance models to simulate the behavior of particular devices. In response to an input workload that includes request arrival, size and position distributions, the simulator calculates the mean response time and queueing delay for requests and specifies the sustained bandwidth and request rates provided for a particular configuration and workload. Preliminary simulation results will be discussed in my talk.

Our performance simulations have two goals. First, we want to understand how best to configure striped tertiary storage arrays. This analysis is geared toward identifying performance bottlenecks in a system, and trying to overcome them. This might, for example, lead to the discovery that adding more readers to a system would dramatically improve performance. Depending on the applications to be run, an array might be designed for maximum sustained throughput or to minimize average latency, so the simulator notes both these metrics.

The second goal of our simulations is to identify desirable properties of future drives and robots. For example, we can use the simulator to determine what the effect on performance would be if a particular drive's mechanical (load or eject) or fast search operations were twice as fast, or if its sustained bandwidth were doubled. We hope to identify a list of desirable properties that may influence companies building devices and robots to design components that are better-suited to perform well in tertiary storage arrays.

3 Striping for Reliability

Besides offering higher bandwidth than non-striped systems, striped systems that include redundancy also offer the potential for much-needed reliability improvements in large tertiary storage systems. Reliability issues for tape arrays are more complex than for disk arrays systems. Issues of particular concern for magnetic tape systems are uncorrectable bit error rates, tape wear and head wear.

3.0.1 Tape Media Reliability

The rate of raw errors (i.e., errors before any error correction has been performed) is quite high on magnetic tape media. Most of these errors are caused by "dropouts," in which the signal being sensed by the tape head drops below a readable value. Dropouts are most commonly caused by protrusions on the tape surface that temporarily increase the separation between the head and the tape, causing a loss in signal intensity [9]. The debris that becomes embedded in the tape and causes dropouts may come from loose pieces of substrate left on the surface when the tape is sliced, from the atmosphere or may accumulate from wear caused by contact between the head and media. Dropouts can also be caused by inhomogeneities in the tape's magnetic coating.

Because of the high raw bit error rates on magnetic tape devices, all drives incorporate large amounts of internal error correction code. However, some errors will occur that the ECC cannot correct. The rate of such errors is called the Uncorrectable Bit Error Rate, and for current products, is around one uncorrectable bit error in every Terabyte of data read. When such an error is encountered, the entire data block on which ECC is performed is lost.

Uncorrectable bit errors in the range of one per Terabyte are of particular concern in multi-Terabyte tertiary storage systems, since such systems WILL contain uncorrectable errors. In addition, if the system has a sustained

read rate of 10 MBytes/sec, then an uncorrectable error will be encountered every 28 hours, on average. If data reliability is important in such systems, then the addition of redundancy information to the system is essential.

Magnetic tapes that are frequently overwritten eventually wear out. In a traditional archival system, where data is written and probably never read again, this is not a serious concern. However, in library applications there is no limit on the number of times a tape may be read. Tapes last on average several hundred passes [1]. However, they wear out even sooner if a particular segment of the tape is accessed repeatedly [5]. Linearly recorded tapes do not suffer so quickly from tape wear-out as tapes written by helical scan methods because the interface with the head is less abrasive, but wear is still a concern. In large tape libraries, algorithms must be developed to track the number of passes to a tape cartridge and replace it before wearout occurs.

In an interactive library application, wear due to stops and starts on the tape is likely to be severe, since we will be performing random accesses to the tapes. Severe wear is manifested by large portions of the magnetic binding material flaking away from the tape backing. Such problems make large sections of a tape unreadable.

3.0.2 Tape Head Wear

Tape heads undergo considerable wear in all tape systems. They last for a few thousand hours of actual contact between the head and medium. Some tape wear is necessary in order to keep the heads in optimum condition [10]. Tape wear helps remove from the head particles that may have been transferred there from the tape surface or the atmosphere, or which came from the tape coating under conditions of friction or extremely high or low humidity. All tape drive manufacturers recommend periodic use of a cleaning cartridge to remove debris from the tape head. Eventually, the head wear becomes extreme. We are exploring algorithms for scheduling both cleaning and replacement of the heads to assure adequate reliability.

3.0.3 Modeling Tape Array Reliability

Head failure is the main cause of tape drive failure; however, the drive may also have other mechanical or electrical failures. Also, as mentioned in Section 1, reliability modeling of arrays must include modeling the failure rates of controllers, power supplies, cables, etc.

We plan to analyze the reliability of tertiary storage arrays with the aim of determining how much error correction (single-bit parity or some form of Reed-Solomon coding) is necessary to ensure adequate reliability of the array. This work will make use of the RELI reliability simulator written by Garth Gibson [4], which estimates the mean time to data loss for particular disk array configurations. We plan to modify the simulator to perform a similar analysis for tertiary storage arrays.

4 Other Research Issues

There are a number of other research topics that are being pursued in the Sequoia group at U.C. Berkeley.

A number of issues having to do with the best ways of configuring tertiary storage arrays have not been touched upon in the previous discussion. These include the decision of the best interleave unit for laying out data, and the best unit of data transfer for optimizing performance for either sustained bandwidth or number of accesses performed per minute. An additional issue is that of allocating buffer space needed to perform synchronization.

Joel Fine is addressing the long latency of accesses to the tape system by looking at abstracts [3]. An abstract is a small subset of a data set that may be able to answer queries to the data set. Because the abstract is small, it can be stored on disk or retrieved fairly quickly from tape. If an abstract can provide a high enough "hit rate" (i.e., can satisfy a reasonable number of queries), then it is worthwhile to build the abstract, a process that can computationally intensive and time-consuming.

Carl Staelin and John Kohl are looking at applying the Log- Structured File System (LFS) ideas of Mendel Rosenblum's work [11] to tertiary storage arrays. LFS systems are write-optimized. The tape array system would be treated as a log. Therefore, writes would be performed sequentially. This is an attractive idea in a tape array system where data is seldom re-read, since it allows cartridges to be written sequentially and minimizes the number of time-consuming switches. Re-reading the data is less efficient in a log-structured system, since there is no guarantee that an entire file is written sequentially.

Finally, we are looking at using compression in striped tertiary storage systems. Compression is appealing because effective bandwidth and capacity are increased when fewer bits are moved and stored. Although many magnetic tape drive manufacturers are now putting compression at device level, we are looking at compression in higher levels of the system. At a higher level, more is known about the nature of data produced by an application, and a compression algorithm appropriate to the data can be chosen.

5 Summary

Striping in tertiary storage array systems is a good idea, both for performance and reliability reasons. Striping offers the potential for higher bandwidth to a single file than can be achieved without striping. And, the additional redundancy available in a striped tape system can offset the reliability problems caused by tape and head wear and uncorrectable bit errors. We are performing simulations to understand the best ways to configure tape arrays composed of currently available devices and robots, and to understand desirable properties for future devices and robots. We are also modeling the reliability of tape array systems to understand how much ECC is needed to maintain adequate reliability, and are developing algorithms for maintaining and replacing tape heads and cartridges.

References

[1] Bharat Bhushan. *Tribology and Mechanics of Magnetic Storage Devices*. Springer-Verlag, New York, 1990.

[2] Peter M. Chen and David A. Patterson. Maximizing performance in a striped disk array. In *Proceedings International Symposium on Computer Architecture*, May 1990.

[3] Joel A. Fine, Thomas E. Anderson, Michael D. Dahlin, James Frew, Michael Olson, and David A. Patterson. Abstracts: A latency-hiding technique for high-capacity massstorage systems. Submitted to ASPLOS-V, March 1992.

[4] Garth Alan Gibson. *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*. PhD thesis, U. C. Berkeley, April 1991. Technical Report No. UCB/CSD 91/613.

[5] H. Goto, A. Asada, H. Chiba, T. Sampei, T. Noguchi, and M. Arakawa. A new concept of data/DAT system. *IEEE Transactions on Consumer Electronics*, 35(3), August 1989.

[6] Jim Gray, Bob Horst, and Mark Walker. Parity striping of disc arrays: Low-cost reliable storage with acceptable throughput. In *Proceedings Very Large Data Bases*, pages 148–161, 1990.

[7] M. Y. Kim. Synchronized disk interleaving. *IEEE Transactions on Computers*, C-35:978–988, November 1986.

[8] M. Livny, S. Khoshafian, and H. Boral. Multi-disk management algorithms. In *Proceedings SIGMETRICS*, pages 69–77, May 1987.

[9] C. Denis Mee and Eric D. Daniel, editors. *Magnetic Recording. Volume II: Computer Data Storage*. McGraw-Hill, New York, 1988.

[10] C. Denis Mee and Eric D. Daniel, editors. *Magnetic Recording, Volume III: Video, Audio, and Instrumentation Recording*. McGraw-Hill, New York, 1988.

[11] Mendel Rosenblum and John K. Ousterhout. The design and implementation of a log-structured file system. In *Proceedings of the 13th ACM Symposium on Operating Systems Principles*, October 1991.

[12] K. Salem and H. Garcia-Molina. Disk striping. In *Proceedings IEEE Data Engineering*, pages 336–342, February 1986.

[13] Martin Schulze, Garth Gibson, Randy H. Katz, and David A. Patterson. How reliable is a RAID? In *Proceedings IEEE COMPCON*, pages 118–123, Spring 1989.

N93-30465

NATIONAL MEDIA LABORATORY MEDIA TESTING RESULTS

William Mularie
National Media Laboratory
P. O. Box 33015
St. Paul, MN 55133-3015

Government Concerns

The government faces a crisis in data storage, analysis, archive and communication. The sheer quantity of data being poured into the government systems on a daily basis is overwhelming systems ability to capture, analyze, disseminate and store critical information. Future systems requirements are even more formidable: with single government platforms having data rate of over 1 Gbit/sec, >Terabyte/day storage requirements, and with expected data archive lifetimes of over 10 years. The charter of the National Media Laboratory (NML) is to focus the resources of industry, government and academia on government needs in the evaluation, development and field support of advanced recording systems.

The Model

The National Lab concept was created in response to the government awareness that various aspects of critical systems acquisition and support were not being met by the traditional government/contractor relationships. It was recognized that the perspective and access to highly-leveraged resources gained from a closer relationship with a consortium of **commercially-focused**, global corporations could benefit many aspects of the government system procurement and support cycle.

NML Continuing Tasks

A key responsibility of the NML is to provide sustaining user support for government recording systems and archive of data. This involves field support to sites to: solve current systems and media problems; give assistance in defining media handling, shipping and storage methodologies; advice and assistance in maintaining and recovery of data in current archives; and to provide assistance in determining the direction of system upgrades.

NML, based upon our ongoing advanced tape evaluation tasks, is also involved in assisting Program Offices and defining recording media requirements necessary for reliable, next generation data recording and archive. NML has been responsible for raising issues relating to reliability and performance as international industry concerns. One example of this involves the archival stability of various types of magnetic pigments; as a result, manufacturers are finally focusing on providing archivally stable media for critical data applications.

Industry/Government Cooperation

The government needs in advanced recording and storage lead commercial markets requirements by 3 to 5 years, both in performance and archival data storage requirements. Joint government/industry participation in the National Media Lab benefits the government by providing highly-leveraged access to the vast resources of the supporting industry and university laboratories to help meet current and future government recording systems evaluation and support.

The domestic recording industry (through NML technical reviews open to domestic industry participation) benefits from the focus on leading-edge requirements. This may assist in building competitiveness of the domestic recording industry in future global markets. Unless the US. manufacturers of advanced storage systems are provided with both a common goals, and a mechanism for focus and cooperation in designing a future system, the US. Government faces the real possibility of either a) having no acceptable method of capturing the vast amounts of data being collected or b) relying on an offshore source.

N93-30466

Evaluation of D-1 Tape and Cassette Characteristics: Moisture Content of Sony and Ampex D-1 Tapes When Delivered

S17-35

15-07

p. 22

Gary Ashton
National Media Laboratory
P. O. Box 33015
St. Paul, MN 55133-3015

Commercial D-1 cassette tapes and their associated recorders were designed to operate in broadcast studios and record in accordance with the International Radio Consultative Committee (CCIR) 607 digital video standards. The D-1 recorder resulted in the Society of Motion Picture and Television Engineers (SMPTE) standards 224 to 228 and is the first digital video recorder to be standardized for the broadcast industry. The D-1 cassette and associated media are currently marketed for broadcast use. The recorder was redesigned for data applications and is in the early stages of being evaluated. The digital data formats used are specified in MIL-STD-2179 and the American National Standards Institute (ANSI) X3.175-190 standard.

In early 1990, the National Media laboratory (NML) was asked to study the effects of time, temperature, and relative humidity on commercial D-1 cassettes. The environmental range to be studied was the one selected for the Advanced Tactical Air Reconnaissance System (ATARS) program. Several discussions between NML personnel, ATARS representatives, recorder contractors, and other interested parties were held to decide upon the experimental plan to be implemented. Review meetings were held periodically during the course of the experiment.

The experiments were designed to determine the dimensional stability of the media and cassette since this is one of the major limiting factors of helical recorders when the media or recorders are subjected to non-broadcasting environments. Measurements were also made to characterize each sample of cassettes to give preliminary information on which purchase specifications could be developed.

The actual tests performed on the cassettes and media before and after aging fall into the general categories listed on the following page.

Tests Before Aging:

- Bulk magnetics
- Surface roughness
- Mechanical properties
- Surface electrical resistivity
- Thickness of the overall tape and each coating
- Tape stiffness
- Tape shrinkage at elevated temperatures for various times
- Quality of tape edges (width, width variation, and weave values)
- Test of commercial and custom packaging
- Magnetic print-through

Tests Before and After Aging:

- Static and dynamic friction
- Cassette operation and dimensions
- D-1 recorder:
  - Signal (RF) output at 40 Mhz
    Noise output at 39 Mhz
    Bit and burst error rates
    Tape tensions

Tests Only After Aging:

- D-1 recorder signal (RF) and noise output after 10 cycles

Test Reports and Data on Diskettes Available:

Reports were written detailing the technical background, methods, equipment used, and results of experiments performed by the National Media Laboratory to evaluate commercial D-1 cassettes for use in a wide range of temperatures and humidities. The variables evaluated include manufacturer's lot, time, temperature, and humidity. Cassettes from two manufacturers, Sony and Ampex, were evaluated. These reports are listed below and are available by contacting:

National Media Laboratory
P.O. Box 33015
Saint Paul, MN 55133-3015

Phone: (612) 736-6183
Fax: (612) 736-4430

Test Reports:

Ashton, Gary R. May 1992. *Coating and Substrate Thickness of Sony and Ampex D-1 Tape.* NML Test Report TR-0013.

Ashton, Gary R. May 1992. *Friction Characteristics of Ampex and Sony D-1 Tapes.* NML Test Report TR-0009.

Ashton, Gary R. February 1992. *Initial Evaluation of D-i Tape and Cassette Characteristics.* NML Technical Report RE-0003.

Ashton, Gary R. May 1992. *M-H Meter Tests on Sony and Ampex D-1 Tape.* NML Test Report TR-0011.

Ashton, Gary R. May 1992. *Magnetic Print-Through Effects in Sony and Ampex D-1 Tapes.* NML Test Report TR-0015.

Ashton, Gary R. May 1992. *Modulus (Stress-Strain Curves) of Sony and Ampex D-1 Tape.* NML Test Report TR-0006.

Ashton, Gary R. May 1992. *Packaging Plan for D-1 Cassettes.* NML Test Report TR-0001.

Ashton, Gary R. May 1992. *Packaging Tests of Commercial D-1 Cassettes and Cases.* NML Test Report TR-0002.

Ashton, Gary R. May 1992. *Relative Humidity of Sony and Ampex D-1 Tapes when Delivered.* NML Test Report TR-0004.

Ashton, Gary R. May 1992. *Resistivity Characteristics of Ampex and Sony D-1 Tape.* NML Test Report TR-0005.

Ashton, Gary R. May 1992. *Shrinkage of Sony and Ampex D-1 Tapes.* NML Test Report TR-0008.

Ashton, Gary R. May 1992. *Stiffness of Sony and Ampex D-1 Tape..* NML Test Report TR-0014.

Ashton, Gary R. May 1992. *Surface Roughness of Sony and Ampex D-1 Tapes.* NML Test Report TR-0012.

Ashton, Gary R. May 1992. *Thermal and Hygroscopic Time Constants of Sony and Ampex D-1 Tape Cassettes.* NML Test Report TR-0016.

Ashton, Gary R. May 1992. *Vibrating Sample Magnetometer (VSM) Tests on Sony and Ampex D-1 Tape.* NML Test Report TR-0010.

Ashton, Gary R. May 1992. *Width and Weave Characteristics of Sony and Ampex D-1 Tape.* NML Test Report TR-0007.

Data Available on Diskettes:

Commercial D-1 Cassettes and Media Test Data: 1990-1991 Data.

Commercial D-1 Cassettes, Media, and Packaging Test Data: 1991-1992 Data.

As an example of the reports generated, the report, *Relative Humidity of Sony and Ampex D-1 Tapes when Delivered*, has been included. The technique used to determine the relative humidity or moisture content of the cassettes as received from the manufacturer was developed by NML specifically for the ATARS program needs. The technique outlined in the attached report example is applicable to the problem of determining the moisture content of any flexible magnetic media for incoming inspection and quality control of archive conditions. The technique also clearly indicates the amount of time needed for the media to respond to changes in the relative humidity of the storage environment.

Introduction

1 Purpose of the Test

The purpose of this test was to determine the moisture content of the tapes when delivered from the manufacturer. This is accomplished by determining the equilibrium relative humidity (RH) of the tapes at delivery. The equilibrium RH is the RH at which the tapes remain constant in weight over time. This is also an indicator of the RH of the environment in which the tapes were packaged.

2 I a Tested

This test examined D-1 digital video tapes from two manufacturers:

Mfg	Model	Mfg Lot #	Test Lot
Ampex	219-M034 (medium)	11571	X
		11961	Y
		12201	Z
	219-L076 (large)	88134/88135	J
Sony	D1M-34 (medium)	NA32112A	T
		NA22113A	U
		NA40114A	V
	D1L-76 (large)	NA92113A	L

In all cases, the cassettes were production items with unknown manufacturing dates.

1 3 Test Requirements

This test is required to understand the moisture content of the cassettes and magnetic tapes as they are received from the supplier. This information is important in determining the conditioning that is required before the cassettes can be used in a recorder, since there is a humidity or moisture content range for recorder operation. Tapes received with the moisture content required for operation can be used with little or no preconditioning. Tapes out of the required moisture content range may require conditioning in a controlled environment before use.

2 Summary

As shown in the following tables, all eight lots of tape tested were packaged at a moisture level corresponding to 45 to 55% relative humidity at 72°F. The variation between lots was smaller in the Ampex tapes than in the Sony tapes.

Ampex D-1 Delivered RH (%)

Size	Lot	Relative Humidity (%)	
		Average	Std. Dev.
Large	J	52.50	1.8
Medium	X	52.63	0.6
Medium	Y	55.14	1.4
Medium	Z	51.10	1.3

Sony D-1 Delivered RH (%)

Size	Lot	Relative Humidity (%)	
		Average	Std. Dev.
Large	L	54.05	1.8
Medium	T	45.90	2.7
Medium	U	45.48	4.0
Medium	V	52.6	1.0

3 References

For background information on the effect of relative humidity in magnetic tapes, see:

Cuddihy, Edward F. 1976. Hygroscopic properties of magnetic recording tape. *IEEE Transactions on Magnetics* 12:2 (March) 126-35.

Test records are maintained by the 3M Records Storage Department, 3M Center Buildings 223 and 224.

Report

4.1 Test Equipment

The following equipment was used in this test:

Mettler Precision Balance, model PM1200, serial number K8412, calibrated 4/3/91.

Mettler Precision Balance, model PM4000, serial number K40517, calibrated 10/25/91 (denoted HOP4000 in Exhibit 1).

Mettler Precision Balance, model PM4000, serial number L58924, calibrated 4/3/91.

Mettler Precision Balance, model PM6100, serial number L03873, calibrated 4/3/91.

Environmentally-controlled rooms in Building 235 at 3M Center used were:

72°F (22°C), 80% RH Room 3B-355
72°F (22°C), 50% RH Room 3C-346
72°F (22°C), 20% RH Room 3B-359

4.2 Test Facility Installation and Set-up

Three equivalent environmental rooms were used, each containing a Mettler Precision Balance. Equivalent means the same rate of air flow and the same temperature, ±2°C. The relative humidity of the rooms was different.

4.3 Test Procedures

1. Use two temperature and humidity-controlled rooms at the same temperature (20 to 25°C) and at two different relative humidities, H_1 and H_2. By using rooms with a difference of about 60% RH, there will be enough mass change to measure with good certainty.

2. Measure and record the initial weight of each of the samples.

3. Place half of the samples in the H_1 chamber and half in the H_2 chamber.

4. Measure and record the weight of each of the samples for at least 7 days; preferably longer.

4.4 Test Results and Analysis

4.4.1 Recorded Data

At first two cassettes were placed in each of the three environmentally-controlled rooms. The Sony T and U lots were measured using these three environments. As experience was gained with the technique and it was realized that the cassettes were delivered at close to 50% RH moisture content, the 72°F, 50% RH test condition was eliminated from the test procedure. The 72°F, 50% RH test condition data was analyzed in exactly the same way as the 72°F, 20% RH and 72°F, 80% RH data as shown below.

In general, before measurements, the room temperature and relative humidity of the three environmentally-controlled rooms were measured. The RH measurements were generally lower than the nominal 20%, 50%, and 80% values. In calculations, the nominal values were used.

Exhibit 1 is the actual recorded data collected during this test.

4.4.2 Test Results

Relative humidity as delivered was calculated as follows. Given the following definitions:

W_0 Original weight of sample
W_1 Final weight of sample in chamber 1
W_2 Final weight of sample in chamber 2
H_0 Original value of humidity
H_1 Relative humidity in chamber 1
H_2 Relative humidity in chamber 2

And assuming the weights change proportional to H_1 and H_2, the following ratio holds:

$$\frac{W_0 - W_1}{W_0 - W_2} = \frac{H_0 - H_1}{H_0 - H_2}$$

Solving for H_0 results in the following:

$$H_0 = \frac{H_1 (W_2 - W_0) + H_2 (W_1 - W_0)}{W_2 - W_1}$$

Given the following definitions:

$h_0(t)$ Calculated value of humidity at some time
$w_1(t)$ Weight of sample in chamber 1 at some time before the final time
$w_2(t)$ Weight of sample in chamber 2 at some time before the final time

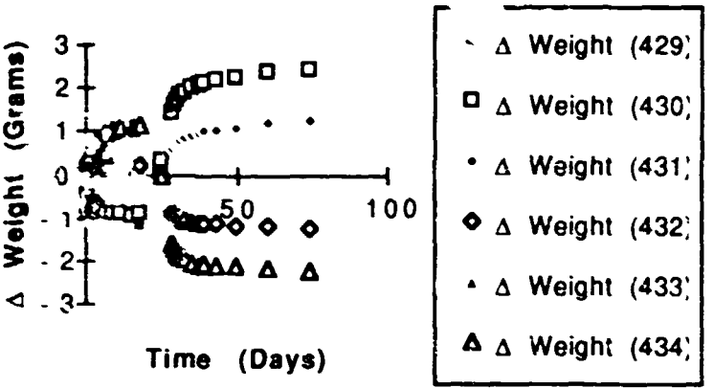
And assuming the time constant for weight change is the same for both chambers (same temperature, etc.), then $w_1(t)$ and $w_2(t)$ can be substituted for W_1 and W_2 in the above formula for H_0 to result in:

$$h_0(t) = \frac{H_1 (w_2(t) - W_0) + H_2 (w_1(t) - W_0)}{w_2(t) - w_1(t)}$$

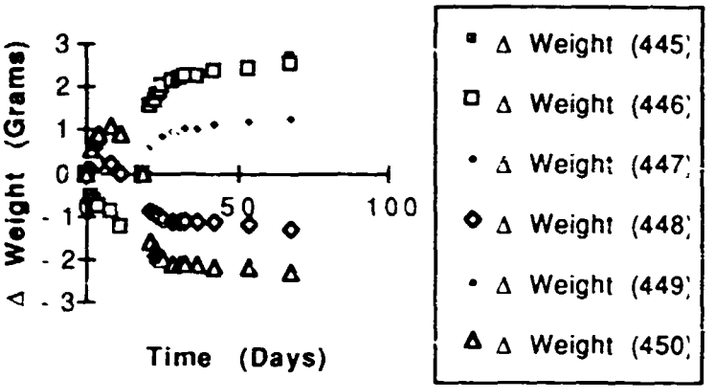
The weights of all samples in the environment at H_1 were averaged to obtain a value of $w_1(t)$. Similarly, the weights of all samples in the environment at H_2 were averaged to obtain a value of $w_2(t)$. Values reported for each lot are averages over all $h_0(t)$ calculated.

The following eight graphs show the change in weight of the cassettes plotted as a function of time. The first two graphs with lot T and U data were measured differently from the other six lots. Lots T and U cassettes were placed in 80, 50, and 20% RH environments while the other lots were placed in only 80 and 20% RH environemnts. Lot T tapes in the 80% RH room were swapped with tapes in the 20% RH room after 25 days. A similar swap was performed with the lot U tapes at 18 days time.

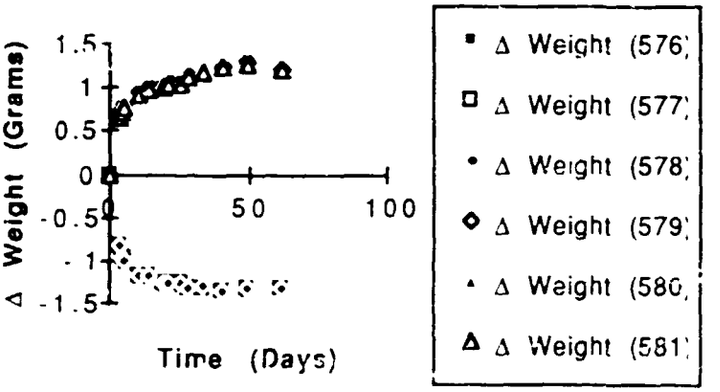
RH Data for Sony Lot T



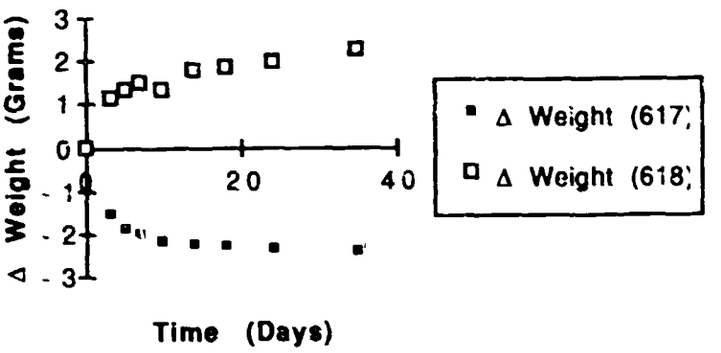
RH Data for Sony Lot U



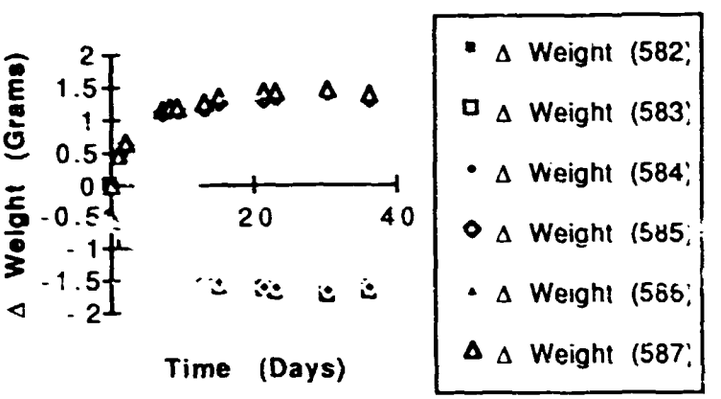
RH Data for Sony Lot V



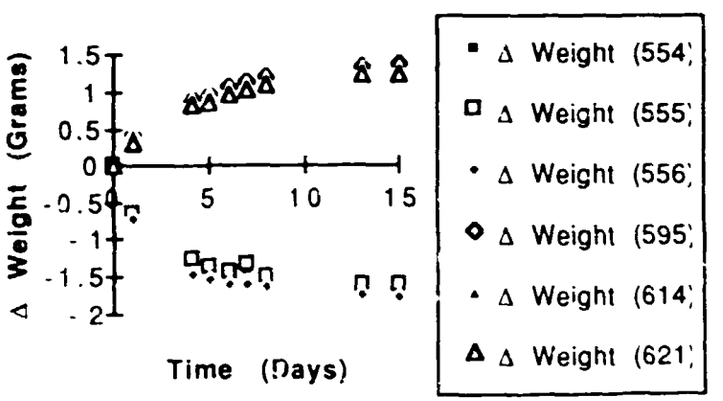
RH Data for Sony Lot L



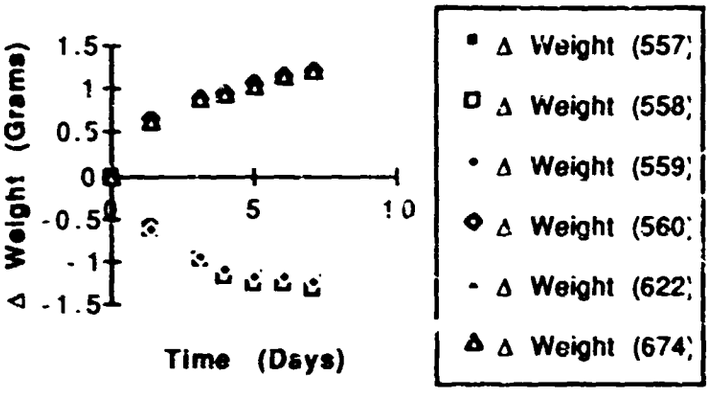
RH Data for Ampex Lot X



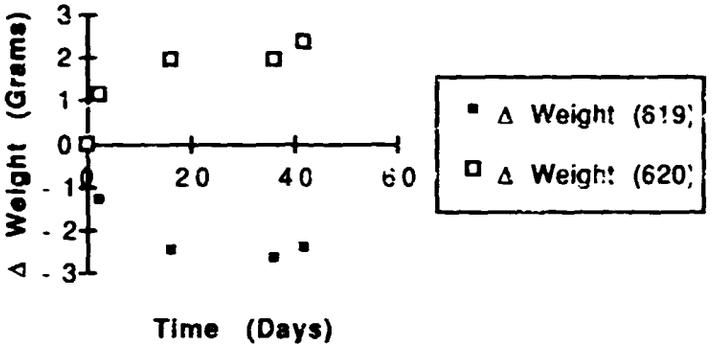
RH Data for Ampex Lot Y



RH Data for Ampex Lot Z



RH Data for Ampex Lot J



5 Conclusions

All eight lots of tape tested were packaged at a moisture level corresponding to 45 to 55% relative humidity at 72°F

6 Recommendations

If the range of cassette moisture is outside of the acceptable 45 to 55% relative humidity range at 72°F, the tapes must be conditioned before use.

7 Appendix

Exhibit 1 is attached.

Exhibit 1 Recorded Data

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
7/24/91	1:30 PM	619	Ampex	J	71	17	1345.76	HOP4000
7/26/91	3:30 PM	619	Ampex	J	71	18	1344.47	HOP4000
8/9/91	4:30 PM	619	Ampex	J	71	16	1343.25	HOP4000
8/29/91	5:30 PM	619	Ampex	J	65	16	1343.11	HOP4000
9/4/91	9:30 AM	619	Ampex	J	72	16.5	1343.33	PM4000
7/24/91	1:30 PM	620	Ampex	J	74	77	1351	HOP4000
7/26/91	3:30 PM	620	Ampex	J	74	77	1352.13	HOP4000
8/9/91	4:30 PM	620	Ampex	J	75	77	1352.99	HOP4000
8/29/91	5:30 PM	620	Ampex	J	71	62	1352.95	HOP4000
9/4/91	9:30 AM	620	Ampex	J	74	78.5	1353.37	PM6100
9/5/91	11:00 AM	620	Ampex	J	74	78	1353.39	PM6100
6/7/91	5:00 PM	617	Sony	L	72	20	1380.515	
6/10/91	4:00 PM	617	Sony	L	72	20	1378.97	
6/12/91	4:00 PM	617	Sony	L	72	20	1378.63	
6/14/91	4:45 PM	617	Sony	L	72	20	1378.52	
6/17/91	3:00 PM	617	Sony	L	72	20	1378.32	
6/21/91	3:00 PM	617	Sony	L	72	20	1378.28	
6/25/91	3:30 PM	617	Sony	L	72	20	1378.19	
7/1/91	3:15 PM	617	Sony	L	72	20	1378.12	
7/12/91	2:00 PM	617	Sony	L	72	20	1378.08	
6/7/91	5:00 PM	618	Sony	L	72	80	1369.64	
6/10/91	4:00 PM	618	Sony	L	72	80	1370.77	
6/12/91	4:00 PM	618	Sony	L	72	80	1370.98	
6/14/91	4:45 PM	618	Sony	L	72	80	1371.14	
6/17/91	3:00 PM	618	Sony	L	72	80	1370.95	
6/21/91	3:00 PM	618	Sony	L	72	80	1371.43	
6/25/91	3:30 PM	618	Sony	L	72	80	1371.51	
7/1/91	3:15 PM	618	Sony	L	72	80	1371.61	
7/12/91	2:00 PM	618	Sony	L	72	80	1371.87	
5/13/91	2:30 PM	429	Sony	T			669.475	PM4000
5/14/91	3:30 PM	429	Sony	T	71.5	18	668.98	PM4000
5/15/91	3:00 PM	429	Sony	T	71.5	18	668.81	PM4000
5/16/91	4:00 PM	429	Sony	T	71	17	668.74	PM4000
5/17/91	3:00 PM	429	Sony	T	71	19	668.7	PM4000
5/20/91	8:00 AM	429	Sony	T	71	18	668.57	PM4000
5/22/91	4:15 PM	429	Sony	T	71	18	668.552373	PM6100
5/24/91	4:00 PM	429	Sony	T	71	19	668.537367	PM6100
5/28/91	11:00 AM	429	Sony	T	71	20	668.527362	PM6100
5/31/91	3:30 PM	429	Sony	T	71	19	668.26	PM4000
6/7/91	4:45 PM	429	Sony	T			668.47	PM4000
5/13/91	2:30 PM	430	Sony	T			668.75	PM4000
5/14/91	3:30 PM	430	Sony	T	71.5	18	668.25	PM4000
5/15/91	3:00 PM	430	Sony	T	71.5	18	668.09	PM4000
5/16/91	4:00 PM	430	Sony	T	71	17	668.02	PM4000
5/17/91	3:00 PM	430	Sony	T	71	19	667.97	PM4000
5/20/91	8:00 AM	430	Sony	T	71	18	667.84	PM4000
5/22/91	4:15 PM	430	Sony	T	71	18	667.837055	PM6100
5/28/91	11:00 AM	430	Sony	T	71	20	667.807042	PM6100

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
5/31/91	3:30 PM	430	Sony	T	71	20	667.817046	PM6100
6/7/91	4:45 PM	430	Sony	T	71	20	668.077162	PM6100
5/13/91	2:30 PM	431	Sony	T			670.86	PM4000
5/14/91	3:30 PM	431	Sony	T			670.79	PM4000
5/15/91	3:00 PM	431	Sony	T	70	49	671.02	PM4000
5/16/91	4:00 PM	431	Sony	T	70	48.5	671.04	PM4000
5/17/91	3:00 PM	431	Sony	T	70	48.5	671.03	PM4000
5/20/91	8:00 AM	431	Sony	T	70	48	670.99	PM4000
5/22/91	4:15 PM	431	Sony	T	70	48	671.053486	PM6100
5/24/91	4:00 PM	431	Sony	T	70	48	671.058488	PM6100
5/28/91	11:00 AM	431	Sony	T	70	48	671.028475	PM6100
5/31/91	3:30 PM	431	Sony	T	70	48	671.073495	PM6100
6/7/91	4:45 PM	431	Sony	T			671.1	PM4000
5/13/91	2:30 PM	432	Sony	T			668.67	PM4000
5/14/91	3:30 PM	432	Sony	T			668.79	PM4000
5/15/91	3:00 PM	432	Sony	T	70	49	668.83	PM4000
5/16/91	4:00 PM	432	Sony	T	70	48.5	668.85	PM4000
5/17/91	3:00 PM	432	Sony	T	70	48.5	668.85	PM4000
5/20/91	8:00 AM	432	Sony	T	70	48	668.8	PM4000
5/22/91	4:15 PM	432	Sony	T	70	48	668.867514	PM6100
5/24/91	4:00 PM	432	Sony	T	70	48	668.862511	PM6100
5/28/91	11:00 AM	432	Sony	T	70	48	668.827496	PM6100
5/31/91	3:30 PM	432	Sony	T	70	48	668.88252	PM6100
6/7/91	4:45 PM	432	Sony	T			668.9	PM4000
5/13/91	2:30 PM	433	Sony	T			672.71	PM4000
5/14/91	3:30 PM	433	Sony	T			673.18	PM4000
5/15/91	3:00 PM	433	Sony	T	75	72	673.39	PM4000
5/16/91	4:00 PM	433	Sony	T	75	72	673.53	PM4000
5/17/91	3:00 PM	433	Sony	T	74	73	673.61	PM4000
5/20/91	8:00 AM	433	Sony	T	74	74	673.75	PM4000
5/22/91	4:15 PM	433	Sony	T	75	73	673.80471	PM6100
5/24/91	4:00 PM	433	Sony	T	75	73	673.824719	PM6100
5/28/91	11:00 AM	433	Sony	T	75	72	673.829721	PM6100
5/31/91	3:30 PM	433	Sony	T	75	73	673.919761	PM6100
6/7/91	4:45 PM	433	Sony	T			674.04	PM4000
5/13/91	2:30 PM	434	Sony	T			670.69	PM4000
5/14/91	3:30 PM	434	Sony	T			671.13	PM4000
5/15/91	3:00 PM	434	Sony	T	75	72	671.32	PM4000
5/16/91	4:00 PM	434	Sony	T	75	72	671.435	PM4000
5/17/91	3:00 PM	434	Sony	T	74	73	671.52	PM4000
5/20/91	8:00 AM	434	Sony	T	74	74	671.66	PM4000
5/22/91	4:15 PM	434	Sony	T	75	73	671.718782	PM6100
5/24/91	4:00 PM	434	Sony	T	75	73	671.7588	PM6100
5/28/91	11:00 AM	434	Sony	T	75	72	671.778809	PM6100
5/31/91	3:30 PM	434	Sony	T	75	73	671.843838	PM6100
6/7/91	4:45 PM	434	Sony	T			671.98	PM4000
6/7/91	4:00 PM	429	Sony swap	T			668.496	
6/10/91	5:00 PM	429	Sony swap	T	75	73	669.995	
6/11/91	2:45 PM	429	Sony swap	T	74	73	670.093	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
6/12/91	3:15 PM	429	Sony swap	T	75	73	670.232	
6/14/91	5:00 PM	429	Sony swap	T	75	74	670.393	
6/17/91	2:30 PM	429	Sony swap	T	75	74	670.527	
6/19/91	5:15 PM	429	Sony swap	T	75	74	670.606	
6/21/91	2:30 PM	429	Sony swap	T	75	75	670.638	
6/25/91	3:00 PM	429	Sony swap	T	75	75	670.702	
7/1/91	2:45 PM	429	Sony swap	T	75	75	670.785	
7/12/91	1:30 PM	429	Sony swap	T	75	77	670.868	
7/26/91	2:00 PM	429	Sony swap	T	74	77	670.938	
6/7/91	4:00 PM	430	Sony swap	T			667.766	
6/10/91	5:00 PM	430	Sony swap	T	75	73	669.228	
6/11/91	2:45 PM	430	Sony swap	T	74	73	669.378	
6/12/91	3:15 PM	430	Sony swap	T	75	73	669.5	
6/13/91	2:45 PM	430	Sony swap	T	75	73	669.592	
6/14/91	5:00 PM	430	Sony swap	T	75	74	669.662	
6/17/91	2:30 PM	430	Sony swap	T	75	74	669.808	
6/19/91	5:15 PM	430	Sony swap	T	75	74	669.871	
6/21/91	2:30 PM	430	Sony swap	T	75	75	669.915	
6/25/91	3:00 PM	430	Sony swap	T	75	75	669.96	
7/1/91	2:45 PM	430	Sony swap	T	75	75	670.048	
7/12/91	1:30 PM	430	Sony swap	T	75	77	670.134	
7/26/91	2:00 PM	430	Sony swap	T	74	77	370.201	
6/7/91	4:00 PM	431	Sony swap	T			671.107	
6/10/91	5:00 PM	431	Sony swap	T	75	73	671.745	
6/11/91	2:45 PM	431	Sony swap	T	74	73	671.813	
6/12/91	3:15 PM	431	Sony swap	T	75	73	671.871	
6/13/91	2:45 PM	431	Sony swap	T	75	73	671.917	
6/14/91	5:00 PM	431	Sony swap	T	75	74	671.959	
6/17/91	2:30 PM	431	Sony swap	T	75	74	672.037	
6/19/91	5:15 PM	431	Sony swap	T	75	74	672.087	
6/21/91	2:30 PM	431	Sony swap	T	75	75	672.104	
6/25/91	3:00 PM	431	Sony swap	T	75	75	672.129	
7/1/91	2:45 PM	431	Sony swap	T	75	75	672.21	
7/12/91	1:30 PM	431	Sony swap	T	75	77	672.288	
7/26/91	2:00 PM	431	Sony swap	T	74	77	672.354	
6/7/91	4:00 PM	432	Sony swap	T			668.922	
6/10/91	5:00 PM	432	Sony swap	T	71	18	668.019	
6/11/91	2:45 PM	432	Sony swap	T	71	18	667.942	
6/12/91	3:15 PM	432	Sony swap	T	71	18	667.886	
6/13/91	2:45 PM	432	Sony swap	T	71	19	667.859	
6/14/91	5:00 PM	432	Sony swap	T	71	19	667.843	
6/17/91	2:30 PM	432	Sony swap	T	71	18	667.771	
6/19/91	5:15 PM	432	Sony swap	T	71	18	667.754	
6/21/91	2:30 PM	432	Sony swap	T	71	20	667.796	
6/25/91	3:00 PM	432	Sony swap	T	71	20	667.772	
7/1/91	2:45 PM	432	Sony swap	T	71	18	667.753	
7/12/91	1:30 PM	432	Sony swap	T	71	20	667.713	
7/26/91	2:00 PM	432	Sony swap	T	71	18	667.654	
6/7/91	4:00 PM	433	Sony swap	T			674.048	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
6/10/91	5:00 PM	433	Sony swap	T	71	18	672.462	
6/11/91	2:45 PM	433	Sony swap	T	71	18	672.32	
6/12/91	3:15 PM	433	Sony swap	T	71	18	672.209	
6/13/91	2:45 PM	433	Sony swap	T	71	19	672.145	
6/14/91	5:00 PM	433	Sony swap	T	71	19	672.103	
6/17/91	2:30 PM	433	Sony swap	T	71	18	671.99	
6/19/91	5:15 PM	433	Sony swap	T	71	18	671.944	
6/21/91	2:30 PM	433	Sony swap	T	71	20	671.984	
6/25/91	3:00 PM	433	Sony swap	T	71	20	671.939	
7/1/91	2:45 PM	433	Sony swap	T	71	18	671.903	
7/12/91	1:30 PM	433	Sony swap	T	71	20	671.859	
7/26/91	2:00 PM	433	Sony swap	T	71	18	671.774	
6/7/91	4:00 PM	434	Sony swap	T			671.994	
6/10/91	5:00 PM	434	Sony swap	T	71	18	670.393	
6/11/91	2:45 PM	434	Sony swap	T	71	18	670.25	
6/12/91	3:15 PM	434	Sony swap	T	71	18	670.143	
6/13/91	2:45 PM	434	Sony swap	T	71	19	670.075	
6/14/91	5:00 PM	434	Sony swap	T	71	19	670.041	
6/17/91	2:30 PM	434	Sony swap	T	71	18	669.918	
6/19/91	5:15 PM	434	Sony swap	T	71	18	669.883	
6/21/91	2:30 PM	434	Sony swap	T	71	20	669.922	
6/25/91	3:00 PM	434	Sony swap	T	71	20	669.88	
7/1/91	2:45 PM	434	Sony swap	T	71	18	669.856	
7/12/91	1:30 PM	434	Sony swap	T	71	20	669.802	
7/26/91	2:00 PM	434	Sony swap	T	71	18	669.72	
5/20/91	10:15 AM	445	Sony	U	71	18	669.84	PM6100
5/21/91	4:45 PM	445	Sony	U	71	18	669.357732	PM6100
5/22/91	4:15 PM	445	Sony	U	71	18	669.217669	PM6100
5/23/91	6:00 PM	445	Sony	U	71	18	669.10762	
5/24/91	4:00 PM	445	Sony	U	71	19	669.067603	
5/28/91	11:00 AM	445	Sony	U	71	20	668.957554	
5/31/91	4:00 PM	445	Sony	U	71	19	668.66	
5/20/91	10:15 AM	446	Sony	U	71	18	674.67	PM6100
5/21/91	4:45 PM	446	Sony	U	71	18	674.129854	PM6100
5/22/91	4:15 PM	446	Sony	U	71	18	674.019805	PM6100
5/23/91	6:00 PM	446	Sony	U	71	18	673.889748	
5/24/91	4:00 PM	446	Sony	U	71	19	673.854732	
5/28/91	11:00 AM	446	Sony	U	71	20	673.739681	
5/31/91	4:00 PM	446	Sony	U	71	19	673.44	
5/20/91	10:15 AM	447	Sony	U	70	48	676.52	PM6100
5/21/91	4:45 PM	447	Sony	U	70	48	676.620962	PM6100
5/22/91	4:15 PM	447	Sony	U	70	48	676.680989	PM6100
5/23/91	6:00 PM	447	Sony	U	70	48.5	676.690994	
5/24/91	4:00 PM	447	Sony	U	70	48	676.695996	
5/28/91	11:00 AM	447	Sony	U	70	48	676.700998	
5/31/91	4:00 PM	447	Sony	U	70	48	676.44	
5/20/91	10:15 AM	448	Sony	U	70	48	675.19	PM6100
5/21/91	4:45 PM	448	Sony	U	70	48	675.340393	PM6100
5/22/91	4:15 PM	448	Sony	U	70	48	675.395417	PM6100

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
5/23/91	6:00 PM	448	Sony	U	70	48.5	675.410424	
5/24/91	4:00 PM	448	Sony	U	70	48	675.40042	
5/28/91	11:00 AM	448	Sony	U	70	48	675.420428	
5/31/91	4:00 PM	448	Sony	U	70	48	675.15	
5/20/91	10:15 AM	449	Sony	U	74	75	671.06	PM6100
5/21/91	4:45 PM	449	Sony	U	75	74	671.578672	PM6100
5/22/91	4:15 PM	449	Sony	U	75	73	671.778809	PM6100
5/23/91	6:00 PM	449	Sony	U	75	74	671.878853	
5/24/91	4:00 PM	449	Sony	U	75	73	671.958889	
5/28/91	11:00 AM	449	Sony	U	75	72	672.098951	
5/31/91	4:00 PM	449	Sony	U	75	73	671.9	
5/20/91	10:15 AM	450	Sony	U	74	75	675.07	PM6100
5/21/91	4:45 PM	450	Sony	U	75	74	675.640526	PM6100
5/22/91	4:15 PM	450	Sony	U	75	73	675.830611	PM6100
5/23/91	6:00 PM	450	Sony	U	75	74	675.925653	
5/24/91	4:00 PM	450	Sony	U	75	73	676.000687	
5/28/91	11:00 AM	450	Sony	U	75	72	676.140749	
5/31/91	4:00 PM	450	Sony	U	75	73	675.95	
6/7/91	4:00 PM	445	Sony swap	U			668.898	
6/10/91	5:00 PM	445	Sony swap	U	75	73	670.495	
6/11/91	2:45 PM	445	Sony swap	U	74	73	670.648	
6/12/91	3:15 PM	445	Sony swap	U	75	73	670.746	
6/13/91	2:45 PM	445	Sony swap	U	75	73	670.84	
6/14/91	5:00 PM	445	Sony swap	U	75	73	670.902	
6/17/91	2:30 PM	445	Sony swap	U	75	74	671.051	
6/19/91	5:15 PM	445	Sony swap	U	75	74	671.093	
6/21/91	2:30 PM	445	Sony swap	U	75	75	671.131	
6/25/91	3:00 PM	445	Sony swap	U	75	75	671.168	
7/1/91	2:45 PM	445	Sony swap	U	75	75	671.231	
7/12/91	1:30 PM	445	Sony swap	U	75	77	671.307	
7/26/91	2:00 PM	445	Sony swap	U	74	77	671.639	
6/7/91	4:00 PM	446	Sony swap	U			673.668	
6/10/91	5:00 PM	446	Sony swap	U	75	73	675.206	
6/11/91	2:45 PM	446	Sony swap	U	74	73	675.361	
6/12/91	3:15 PM	446	Sony swap	U	75	73	675.492	
6/13/91	2:45 PM	446	Sony swap	U	75	73	675.582	
6/14/91	5:00 PM	446	Sony swap	U	75	73	675.666	
6/17/91	2:30 PM	446	Sony swap	U	75	74	675.801	
6/19/91	5:15 PM	446	Sony swap	U	75	74	675.871	
6/21/91	2:30 PM	446	Sony swap	U	75	75	675.9	
6/25/91	3:00 PM	446	Sony swap	U	75	75	675.951	
7/1/91	2:45 PM	446	Sony swap	U	75	75	676.038	
7/12/91	1:30 PM	446	Sony swap	U	75	77	676.121	
7/26/91	2:00 PM	446	Sony swap	U	74	77	676.189	
6/7/91	4:00 PM	447	Sony swap	U			676.81	
6/10/91	5:00 PM	447	Sony swap	U	75	73	677.453	
6/11/91	2:45 PM	447	Sony swap	U	74	73	677.528	
6/12/91	3:15 PM	447	Sony swap	U	75	73	677.584	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
6/13/91	2:45 PM	447	Sony swap	U	75	73	677.624	
6/14/91	5:00 PM	447	Sony swap	U	75	73	677.666	
6/17/91	2:30 PM	447	Sony swap	U	75	74	677.765	
6/19/91	5:15 PM	447	Sony swap	U	75	74	677.802	
6/21/91	2:30 PM	447	Sony swap	U	75	75	677.827	
6/25/91	3:00 PM	447	Sony swap	U	75	75	677.86	
7/1/91	2:45 PM	447	Sony swap	U	75	75	677.935	
7/12/91	1:30 PM	447	Sony swap	U	75	77	678.006	
7/26/91	2:00 PM	447	Sony swap	U	74	77	678.074	
6/7/91	4:00 PM	448	Sony swap	U			675.507	
6/10/91	5:00 PM	448	Sony swap	U	71	18	674.622	
6/11/91	2:45 PM	448	Sony swap	U	71	18	674.544	
6/12/91	3:15 PM	448	Sony swap	U	71	18	674.487	
6/13/91	2:45 PM	448	Sony swap	U	71	19	674.457	
6/14/91	5:00 PM	448	Sony swap	U	71	19	674.444	
6/17/91	2:30 PM	448	Sony swap	U	71	18	674.362	
6/19/91	5:15 PM	448	Sony swap	U	71	18	674.341	
6/21/91	2:30 PM	448	Sony swap	U	71	20	674.388	
6/25/91	3:00 PM	448	Sony swap	U	71	20	674.35	
7/1/91	2:45 PM	448	Sony swap	U	71	18	674.339	
7/12/91	1:30 PM	448	Sony swap	U	71	20	674 313	
7/26/91	2:00 PM	448	Sony swap	U	71	18	674.216	
6/7/91	4:00 PM	449	Sony swap	U			672.411	
6/10/91	5:00 PM	449	Sony swap	U	71	18	670.747	
6/11/91	2:45 PM	449	Sony swap	U	71	18	670.613	
6/12/91	3:15 PM	449	Sony swap	U	71	18	670.497	
6/13/91	2:45 PM	449	Sony swap	U	71	19	670.43	
6/14/91	5:00 PM	449	Sony swap	U	71	19	670.391	
6/17/91	2:30 PM	449	Sony swap	U	71	18	670.278	
6/19/91	5:15 PM	449	Sony swap	U	71	18	670.245	
6/21/91	2:30 PM	449	Sony swap	U	71	20	670.287	
6/25/91	3:00 PM	449	Sony swap	U	71	20	670.248	
7/1/91	2:45 PM	449	Sony swap	U	71	18	670.223	
7/12/91	1:30 PM	449	Sony swap	U	71	20	670.202	
7/26/91	2:00 PM	449	Sony swap	U	71	18	670.0826	
6/7/91	4:00 PM	450	Sony swap	U			676.446	
6/10/91	5:00 PM	450	Sony swap	U	71	18	674.812	
6/11/91	2:45 PM	450	Sony swap	U	71	18	674.67	
6/12/91	3:15 PM	450	Sony swap	U	71	18	674.56	
6/13/91	2:45 PM	450	Sony swap	U	71	19	674.511	
6/14/91	5:00 PM	450	Sony swap	U	71	19	674.457	
6/17/91	2:30 PM	450	Sony swap	U	71	18	674.332	
6/19/91	5:15 PM	450	Sony swap	U	71	18	674.297	
6/21/91	2:30 PM	450	Sony swap	U	71	20	674.336	
6/25/91	3:00 PM	450	Sony swap	U	71	20	674.299	
7/1/91	2:45 PM	450	Sony swap	U	71	18	674.267	
7/12/91	1:30 PM	450	Sony swap	U	71	20	674.237	
7/26/91	2:00 PM	450	Sony swap	U	71	18	674.133	
6/28/91	2:30 PM	576	Sony	V	71	18	673.645	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
7/1/91	2:45 PM	576	Sony	V	71	18	672.732	
7/2/91	3:30 PM	576	Sony	V	71	18	672.619	
7/3/91	1:30 PM	576	Sony	V	71	19	672.57	
7/8/91	5:30 PM	576	Sony	V	71	18	672.416	
7/12/91	1:30 PM	576	Sony	V	71	20	672.404	
7/17/91	4:45 PM	576	Sony	V	71	19	672.329	
7/19/91	3:15 PM	576	Sony	V	71	18	672.313	
7/23/91	4:30 PM	576	Sony	V	71	18	672.325	
7/26/91	1:45 PM	576	Sony	V	71	18	672.28	
7/31/91	1:45 PM	576	Sony	V	71	18	672.279	
8/7/91	5:00 PM	576	Sony	V	71	17	672.249	
8/16/91	3:45 PM	576	Sony	V	71	17	672.269	
8/29/91	4:30 PM	576	Sony	V	65	16	672.274	
6/28/91	2:30 PM	577	Sony	V	71	18	672.411	
7/1/91	2:45 PM	577	Sony	V	71	18	671.591	
7/2/91	3:30 PM	577	Sony	V	71	18	671.464	
7/3/91	1:30 PM	577	Sony	V	71	19	671.43	
7/8/91	5:30 PM	577	Sony	V	71	18	671.239	
7/12/91	1:30 PM	577	Sony	V	71	20	671.223	
7/17/91	4:45 PM	577	Sony	V	71	19	671.142	
7/19/91	3:15 PM	577	Sony	V	71	18	671.141	
7/23/91	4:30 PM	577	Sony	V	71	18	671.126	
7/26/91	1:45 PM	577	Sony	V	71	18	671.088	
7/31/91	1:45 PM	577	Sony	V	71	18	671.089	
8/7/91	5:00 PM	577	Sony	V	71	17	671.058	
8/16/91	3:45 PM	577	Sony	V	71	17	671.07	
8/29/91	4:30 PM	577	Sony	V	65	16	671.076	
6/28/91	2:30 PM	578	Sony	V	71	18	673.42	
7/1/91	2:45 PM	578	Sony	V	71	18	672.586	
7/2/91	3:30 PM	578	Sony	V	71	18	672.455	
7/3/91	1:30 PM	578	Sony	V	71	19	672.414	
7/8/91	5:30 PM	578	Sony	V	71	18	672.244	
7/12/91	1:30 PM	578	Sony	V	71	20	672.246	
7/17/91	4:45 PM	578	Sony	V	71	19	672.156	
7/19/91	3:15 PM	578	Sony	V	71	18	672.135	
7/23/91	4:30 PM	578	Sony	V	71	18	672.145	
7/26/91	1:45 PM	578	Sony	V	71	18	672.103	
7/31/91	1:45 PM	578	Sony	V	71	18	672.107	
8/7/91	5:00 PM	578	Sony	V	71	17	672.065	
8/16/91	3:45 PM	578	Sony	V	71	17	672.076	
8/29/91	4:30 PM	578	Sony	V	65	16	672.087	
6/28/91	2:30 PM	579	Sony	V	75	74	670.963	
7/1/91	2:45 PM	579	Sony	V	75	75	671.64	
7/2/91	3:30 PM	579	Sony	V	75	75	671.718	
7/3/91	1:30 PM	579	Sony	V	75	74	671.772	
7/8/91	5:30 PM	579	Sony	V	75	74	671.921	
7/12/91	1:30 PM	579	Sony	V	75	77	671.977	
7/17/91	4:45 PM	579	Sony	V	75	76	672.013	
7/19/91	3:15 PM	579	Sony	V	75	75	672.028	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
7/23/91	4:30 PM	579	Sony	V	74	76	672.034	
7/26/91	1:45 PM	579	Sony	V	74	77	672.097	
7/31/91	1:45 PM	579	Sony	V	74	77	672.149	
8/7/91	5:00 PM	579	Sony	V	74	78	672.22	
8/16/91	3:45 PM	579	Sony	V	75	77	672.27	
8/29/91	4:30 PM	579	Sony	V	71	62	672.187	
6/28/91	2:30 PM	580	Sony	V	75	74	666.127	
7/1/91	2:45 PM	580	Sony	V	75	75	666.815	
7/2/91	3:30 PM	580	Sony	V	75	75	666.888	
7/3/91	1:30 PM	580	Sony	V	75	74	666.948	
7/8/91	5:30 PM	580	Sony	V	75	74	667.09	
7/12/91	1:30 PM	580	Sony	V	75	77	667.147	
7/17/91	4:45 PM	580	Sony	V	75	76	667.185	
7/19/91	3:15 PM	580	Sony	V	75	75	667.201	
7/23/91	4:30 PM	580	Sony	V	74	76	667.207	
7/26/91	1:45 PM	580	Sony	V	74	77	667.263	
7/31/91	1:45 PM	580	Sony	V	74	77	667.361	
8/7/91	5:00 PM	580	Sony	V	74	78	667.389	
8/16/91	3:45 PM	580	Sony	V	75	77	667.43	
8/29/91	4:30 PM	580	Sony	V	71	62	667.354	
6/28/91	2:30 PM	581	Sony	V	75	74	673.536	
7/1/91	2:45 PM	581	Sony	V	75	75	674.186	
7/2/91	3:30 PM	581	Sony	V	75	75	674.259	
7/3/91	1:30 PM	581	Sony	V	75	74	674.315	
7/8/91	5:30 PM	581	Sony	V	75	74	674.466	
7/12/91	1:30 PM	581	Sony	V	75	77	674.524	
7/17/91	4:45 PM	581	Sony	V	75	76	674.544	
7/19/91	3:15 PM	581	Sony	V	75	75	674.573	
7/23/91	4:30 PM	581	Sony	V	74	76	674.588	
7/26/91	1:45 PM	581	Sony	V	74	77	674.672	
7/31/91	1:45 PM	581	Sony	V	74	77	674.709	
8/7/91	5:00 PM	581	Sony	V	74	78	674.774	
8/16/91	3:45 PM	581	Sony	V	75	77	674.812	
8/29/91	4:30 PM	581	Sony	V	71	62	674.737	
7/24/91	1:15 PM	582	Ampex	X	71	17	682.356	
7/25/91	4:00 PM	582	Ampex	X	71	17	681.761	
7/26/91	1:45 PM	582	Ampex	X	71	18	681.586	
7/31/91	1:45 PM	582	Ampex	X	71	18	681.023	
8/1/91	4:00 PM	582	Ampex	X	71	18	680.963	
8/2/91	3:30 PM	582	Ampex	X	71	18	680.956	
8/6/91	5:15 PM	582	Ampex	X	71	18	680.822	
8/8/91	3:00 PM	582	Ampex	X	71	17	680.823	
8/14/91	4:30 PM	582	Ampex	X	71	16	680.763	
8/16/91	3:45 PM	582	Ampex	X	71	17	680.751	
8/23/91	3:45 PM	582	Ampex	X	72	18	680.713	
8/29/91	4:30 PM	582	Ampex	X	65	6	680.759	
7/24/91	1:15 PM	583	Ampex	X	71	17	683.553	
7/25/91	4:00 PM	583	Ampex	X	71	17	682.946	
7/26/91	1:45 PM	583	Ampex	X	71	18	682.652	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
7/31/91	1:45 PM	583	Ampex	X	71	18	682.178	
8/1/91	4:00 PM	583	Ampex	X	71	18	682.095	
8/2/91	3:30 PM	583	Ampex	X	7.	18	682.08	
8/6/91	5:15 PM	583	Ampex	X	71	18	681.925	
8/8/91	3:00 PM	583	Ampex	X	71	17	681.93	
8/14/91	4:30 PM	583	Ampex	X	71	16	681.891	
8/16/91	3:45 PM	583	Ampex	X	71	17	681.867	
8/23/91	3:45 PM	583	Ampex	X	72	18	681.817	
8/29/91	4:30 PM	583	Ampex	X	65	16	681.856	
7/24/91	1:15 PM	584	Ampex	X	71	17	681.779	
7/25/91	4:00 PM	584	Ampex	X	71	17	681.244	
26/91	1:45 PM	584	Ampex	X	71	18	680.965	
./31/91	1:45 PM	584	Ampex	X	71	18	680.422	
8/1/91	4:00 PM	584	Ampex	X	71	18	680.425	
8/2/91	3:30 PM	584	Ampex	X	71	18	680.354	
8/6/91	5:15 PM	584	Ampex	X	71	18	680.199	
8/8/91	3:00 PM	584	Ampex	X	71	17	680.254	
8/14/91	4:30 PM	584	Ampex	X	71	16	680.164	
8/16/91	3:45 PM	584	Ampex	X	71	17	680.161	
8/23/91	3:45 PM	584	Ampex	X	72	18	680.111	
8/29/91	4:30 PM	584	Ampex	X	65	16	680.149	
7/24/91	1:15 PM	585	Ampex	X	74	77	681.283	
7/25/91	4:00 PM	585	Ampex	X	75	77	681.732	
7/26/91	1:45 PM	585	Ampex	X	74	77	631.992	
7/31/91	1:45 PM	585	Ampex	X	74	77	682.407	
8/1/91	4:00 PM	585	Ampex	X	75	77	682.447	
8/2/91	3:30 PM	585	Ampex	X	74	77	682.436	
8/6/91	5:15 PM	585	Ampex	X	74	77	682.464	
8/8/91	3:00 PM	585	Ampex	X	75	78	682.562	
8/14/91	4:30 PM	585	Ampex	X	75	77	682.609	
8/16/91	3:45 PM	585	Ampex	X	75	77	682.648	
8/23/91	3:45 PM	58	Ampex	X	74	77	682.704	
8/29/91	4:30 PM	585	Ampex	X	71	62	682.605	
7/24/91	1:15 PM	586	Ampex	X	74	77	683.727	
7/25/91	4:00 PM	586	Ampex	X	75	77	684.206	
7/26/91	1:45 PM	586	Ampex	X	74	77	684.45	
7/31/91	1:45 PM	586	Ampex	X	74	77	684.918	
8/1/91	4:00 PM	586	Ampex	X	75	77	684.965	
8/2/91	3:30 PM	586	Ampex	X	74	77	684.972	
8/6/91	5:15 PM	586	Ampex	X	74	77	685.025	
8/8/91	3:00 PM	586	Ampex	X	75	78	685.116	
8/14/91	4:30 PM	586	Ampex	X	75	77	685.179	
8/16/91	3:45 PM	586	Ampex	X	75	77	685.207	
8/23/91	3:45 PM	586	Ampex	X	74	77	685.251	
8/29/91	4:30 PM	586	Ampex	X	71	62	685.167	
7/24/91	1:15 PM	587	Ampex	X	74	77	682.72	
7/25/91	4:00 PM	587	Ampex	X	75	77	683.164	
7/26/91	1:45 PM	587	Ampex	X	74	77	683.384	
7/31/91	1:45 PM	587	Ampex	X	74	77	683.861	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
8/1/91	4:00 PM	587	Ampex	X	75	77	683.929	
8/2/91	3:30 PM	587	Ampex	X	74	77	683.922	
8/6/91	5:15 PM	587	Ampex	X	74	77	683.987	
8/8/91	3:00 PM	587	Ampex	X	75	78	684.069	
8/14/91	4:30 PM	587	Ampex	X	75	77	684.137	
8/16/91	3:45 PM	587	Ampex	X	75	77	684.159	
8/23/91	3:45 PM	587	Ampex	X	74	77	684.189	
8/29/91	4:30 PM	587	Ampex	X	71	62	684.12	
8/1/91	3:45 PM	554	Ampex	Y	71	18	681.33	
8/2/91	3:30 PM	554	Ampex	Y	71	18	680.715	
8/5/91	6:15 PM	554	Ampex	Y	71	17	680.033	
8/6/91	5:15 PM	554	Ampex	Y	71	17	679.938	
8/7/91	5:00 PM	554	Ampex	Y	71	17	679.886	
8/8/91	3:00 PM	554	Ampex	Y	71	17	679.949	
8/9/91	4:15 PM	554	Ampex	Y	71	16	679.83	
8/14/91	4:30 PM	554	Ampex	Y	71	16	679.735	
8/16/91	3:45 PM	554	Ampex	Y	71	17	679.762	
8/1/91	3:45 PM	555	Ampex	Y	71	18	684.485	
8/2/91	3:30 PM	555	Ampex	Y	71	18	683.846	
8/5/91	6:15 PM	555	Ampex	Y	71	17	683.244	
8/6/91	5:15 PM	555	Ampex	Y	71	17	683.14	
8/7/91	5:00 PM	555	Ampex	Y	71	17	683.051	
8/8/91	3:00 PM	555	Ampex	Y	71	17	683.156	
8/9/91	4:15 PM	555	Ampex	Y	71	16	683.006	
8/14/91	4:30 PM	555	Ampex	Y	71	16	682.876	
8/16/91	3:45 PM	555	Ampex	Y	71	17	682.885	
8/1/91	3:45 PM	556	Ampex	Y	71	18	684.906	
8/2/91	3:30 PM	556	Ampex	Y	71	18	684.193	
8/5/91	6:15 PM	556	Ampex	Y	71	17	683.449	
8/6/91	5:15 PM	556	Ampex	Y	71	17	683.36	
8/7/91	5:00 PM	556	Ampex	Y	71	17	683.312	
8/8/91	3:00 PM	556	Ampex	Y	71	17	683.299	
8/9/91	4:15 PM	556	Ampex	Y	71	16	683.28	
8/14/91	4:20 PM	556	Ampex	Y	71	16	683.158	
8/16/91	3:45 PM	556	Ampex	Y	71	17	683.145	
8/1/91	3:45 PM	595	Ampex	Y	75	77	684.943	
8/2/91	3:30 PM	595	Ampex	Y	?	77	685.366	
8/5/91	6:15 PM	595	Ampex	Y	74	76	685.841	
8/6/91	5:15 PM	595	Ampex	Y	74	77	685.919	
8/7/91	5:00 PM	595	Ampex	Y	74	78	685.998	
8/8/91	3:00 PM	595	Ampex	Y	75	78	686.06	
8/9/91	4:15 PM	595	Ampex	Y	75	77	686.135	
8/14/91	4:30 PM	595	Ampex	Y	75	77	686.273	
8/16/91	3:45 PM	595	Ampex	Y	75	77	686.302	
8/29/91	4:30 PM	595	Ampex	Y	71	62	686.284	
8/1/91	3:45 PM	614	Ampex	Y	75	77	682.098	
8/2/91	3:30 PM	614	Ampex	Y	?	77	682.55	
8/5/91	6:15 PM	614	Ampex	Y	74	76	682.987	
8/6/91	5:15 PM	614	Ampex	Y	74	77	683.055	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
8/7/91	5:00 PM	614	Ampex	Y	74	78	683.138	
8/8/91	3:00 PM	614	Ampex	Y	75	78	683.188	
8/9/91	4:15 PM	614	Ampex	Y	75	77	683.258	
8/14/91	4:30 PM	614	Ampex	Y	75	77	683.392	
8/16/91	3:45 PM	614	Ampex	Y	75	77	683.411	
8/29/91	4:30 PM	614	Ampex	Y	71	62	683.376	
8/1/91	3:45 PM	621	Ampex	Y	75	77	682.395	
8/2/91	3:30 PM	621	Ampex	Y	?	77	682.72	
8/5/91	6:15 PM	621	Ampex	Y	74	76	683.215	
8/6/91	5:15 PM	621	Ampex	Y	74	77	683.271	
8/7/91	5:00 PM	621	Ampex	Y	74	78	683.37	
8/8/91	3:00 PM	621	Ampex	Y	75	78	683.418	
8/9/91	4:15 PM	621	Ampex	Y	75	77	683.493	
8/14/91	4:30 PM	621	Ampex	Y	75	77	683.636	
8/16/91	3:45 PM	621	Ampex	Y	75	77	683.643	
8/29/91	4:30 PM	621	Ampex	Y	71	62	683.579	
9/6/91	9:30 AM	621	Ampex	Y	74	78.5	683.86	PM6100
9/6/91	2:45 PM	557	Ampex	Z	71	17	680.82	
9/7/91	11:30 PM	557	Ampex	Z	72	18	680.24	
9/9/91	3:45 PM	557	Ampex	Z	72	18	679.87	
9/10/91	2:00 PM	557	Ampex	Z	72	16	679.7	
9/11/91	3:00 PM	557	Ampex	Z	72	18	679.63	
9/12/91	2:00 PM	557	Ampex	Z	71	17	679.6	
9/13/91	4:00 PM	557	Ampex	Z	71	17	679.58	
9/6/91	2:45 PM	558	Ampex	Z	71	17	683.68	
9/7/91	11:30 PM	558	Ampex	Z	72	18	683.065	
9/9/91	3:45 PM	558	Ampex	Z	72	18	682.69	
9/10/91	2:00 PM	558	Ampex	Z	72	16	682.5	
9/11/91	3:00 PM	558	Ampex	Z	72	18	682.4	
9/12/91	2:00 PM	558	Ampex	Z	71	17	682.41	
9/13/91	4:00 PM	558	Ampex	Z	71	17	682.36	
9/6/91	2:45 PM	559	Ampex	Z	71	17	682.32	
9/7/91	11:30 PM	559	Ampex	Z	72	18	681.7	
9/9/91	3:45 PM	559	Ampex	Z	72	18	681.38	
9/10/91	2:00 PM	559	Ampex	Z	72	16	681.21	
9/11/91	3:00 PM	559	Ampex	Z	72	18	681.15	
9/12/91	2:00 PM	559	Ampex	Z	71	17	681.15	
9/13/91	4:00 PM	559	Ampex	Z	71	17	681.09	
9/6/91	2:45 PM	560	Ampex	Z	74	79	682.63	
9/7/91	11:30 PM	560	Ampex	Z	74.5	78	683.29	
9/9/91	3:45 PM	560	Ampex	Z	74	78	683.52	
9/10/91	2:00 PM	560	Ampex	Z	75	79	683.58	
9/11/91	3:00 PM	560	Ampex	Z	74	80	683.69	
9/12/91	2:00 PM	560	Ampex	Z	74	80	683.79	
9/13/91	4:00 PM	560	Ampex	Z	75	79	683.84	
9/6/91	2:45 PM	622	Ampex	Z	74	79	680.83	
9/7/91	11:30 PM	622	Ampex	Z	74.5	78	681.48	
9/9/91	3:45 PM	622	Ampex	Z	74	78	681.71	
9/10/91	2:00 PM	622	Ampex	Z	75	79	681.77	

Exhibit 1 Recorded Data (continued)

Date	Time	Tape #	Mfg	Lot	Temp (°F)	RH (%)	Weight (g)	Scale
9/11/91	3:00 PM	622	Ampex	Z	74	80	681.88	
9/12/91	2:00 PM	622	Ampex	Z	74	80	681.98	
9/13/91	4:00 PM	622	Ampex	Z	75	79	682.03	
9/6/91	2:45 PM	674	Ampex	Z	74	79	687.26	
9/7/91	11:30 PM	674	Ampex	Z	74.5	78	687.9	
9/9/91	3:45 PM	674	Ampex	Z	74	78	688.14	
9/10/91	2:00 PM	674	Ampex	Z	75	79	688.2	
9/11/91	3:00 PM	674	Ampex	Z	74	80	688.31	
9/12/91	2:00 PM	674	Ampex	Z	74	80	688.41	
9/13/91	4:00 PM	674	Ampex	Z	75	79	688.47	

N93-80467

Grand Challenges in Mass Storage - A Systems Integrators Perspective

Richard K. Lee
Data Storage Technologies, Inc.
579 Franklin Turnpike, P. O. Box 1293
Ridgewood, NJ 07450

Daniel G. Mintz, WJ Culver Consulting, Inc.
8500 Leesburg Pike, Suite 402, Vienna, VA 22182

S/3-82
IEEE 58
p. 5

Within today's much ballyhooed supercomputing environment, with its GFLOPS of CPU power, and Gigabit networks, there exists a major roadblock to computing success; that of Mass Storage.

We consider the solution to this mass storage problem to be one of the "Grand Challenges" facing the computer industry today, as well as long into the future. It has become obvious to us, as well as many others in the industry, that there is no clear single solution in sight.

The Systems Integrator today is faced with a myriad of quandaries in approaching this challenge. He must first be innovative in approach, second choose hardware solutions that are volumetric efficient; high in signal bandwidth; available from multiple sources; competitively priced; and have forward growth extendibility. In addition he must also comply with a variety of mandated, and often conflicting software standards (GOSIP, POSIX, IEEE, MSRM 4.0 and others), and finally he must deliver a systems solution with the "most bang for the buck" in terms of cost vs. performance factors. These quandaries challenge the Systems Integrator to "push the envelope" in terms of his or her ingenuity and innovation on an almost daily basis.

Within our presentation we will explore this dynamic further, and attempt to acquaint the audience with rational approaches to this "Grand Challenge".

Introduction:

WJ Culver Consulting and Data Storage Technologies are collaborating together on this presentation based on our individual efforts in supporting EOSDIS ECS Phase C/D Proposal teams, and in our joint preparation of a winning solution for the NASA LaRC EOSDIS "Version 0" DAAC[1] , whose contract was recently awarded to the WJ Culver Consulting team and will be installed on site at LaRC during July and August of this year (1992).

Our two organizations have been intimately involved in many facets of mass storage system design and integration, and we feel that we have special insights into the problems facing this segment of the computer industry. We will explore this subject from the perspective of having to design and field systems today, with vision towards what the future holds.

Definitions:

Mass Storage has become a widely and many times improperly used term today. It can be found as a reference to a simple disk or tape drive or in referring to PedaByte level systems. For sake of consistency we will define Mass Storage as any type of storage system exceeding .1TB in total size (on-line), under control of a centralized File Management scheme. (Authors Note: We hope that this definition does not confuse the reader any further than he might already be confused on this subject!)

---

[1] The significance of the "Version 0" prototype has been heightened in recent months based on reports to Congress by the GAO, and to NASA by the NRC, emphasizing the importance of this prototyping effort prior to the fielding of the EOSDIS ECS systems.

On-line refers to a storage device; DMA, Network or Peripherally connected, which responds to file requests in 0 to 15 seconds (approximately).

Off-line refers to a storage device, Network or Peripherally connected, which responds to file requests in several minutes to several days or weeks (approximately).

An **Automated Library** is a physical volume repository (PVR) which houses bitfile data contained in volumes of robotically handled cartridges or cylinders. These systems are Network or Peripherally connected and respond to file requests in 45 seconds to tens of minutes depending upon the size and physical architecture of the repository.

Today's Supercomputing Environment:

With very few exceptions, today's supercomputing center has become a hodge podge of many different types of CPU's (vector, scalar, parallel/massively parallel, Visualization, RISC, CISC, etc., etc.). Each of these units is in competition with the others for dominance of system resources, and all are interconnected by elaborate networking schemes (HyperChannel, FDDI, HIPPI, kluge, and others). For many years now the high performance computing industry has been focused only on how to achieve the highest level of CPU performance (many times by networking heterogeneous CPU's together) without paying any attention to the "crisis in storage management" these systems have created. This blind pursuit of computing horsepower has created an acute crisis in today's data center; that of how to manage the huge volumes of input and output data required/produced by these machines.

These advanced processors produce volumes of bitfile data well beyond most systems managers wildest hallucinations. Local and network disk and tape systems are overwhelmed by the growing demand for bitfile data and their ability to store and archive vast numbers of exponentially increasing bitfiles is critically inadequate. Current disk farms can only store files for a period of 12-36 hours before being overwritten to make way for new bitfiles[2]. This has created an often untenable situation for both the systems manager and the end-user.

To better manage this critical task, dedicated file managers and intricate software schemes have been developed by many. These systems attempt to keep ahead of user needs by staging and re-staging bitfile data sets to the most appropriate media for the level of activity encountered. This is usually done over relatively low-bandwidth channels on low efficiency and high cost medias (magnetic disk and square tape). Traditional off-line round/square tape drives have been augmented by tape libraries which behave like "slow-moving" freight trains of bitfiles; "The information gets there eventually, but it's a bumpy ride along the way".

These band-aid approaches have gone a long way to help alleviate the problem for the short term, but are woefully inadequate for the long haul. The need for wide bandwidth, volumetric efficient storage systems is paramount to solve these problems.

Another factor exacerbating this crisis further is the impact of scientific visualization on the supercomputer center. This new science in computing has brought about a great many breakthroughs in terms of solutions to problems that were previously dealt with as great streams of numbers on print-out paper, but not without a cost. Visualization files are on the order of 1GB[3,4] in size each and when animated together produce a major drain on bitfile storage resources. In many centers it is less expensive to re-run the simulation on the supercomputer, than to store the visualization data. This is further compounded by the types of

---

[2] Results from a privately sponsored survey of 18 leading supercomputing centers in the US during 1990 by Data Storage Technologies and CIRRUS Aerospace

[3] Physics Today, October 1987, "A Numerical Laboratory", Karl-Heinz Winkler et al.

[4] AIAA/NASA Second International Symposium on Space Information Systems, September 1990, "High Rate Science Data Handling on Space Station Freedom", T. Handley et al.

hardware required to store many of these images i.e. wide-bandwidth RAID systems optimized for image transfer

The icing on the cake in terms of this entire situation is the new fiscal realities that everyone in the government and private sector are now facing. The days of well funded initiatives and large departmental budgets are gone and will never return. Today, all decisions are made in terms of cost as the first priority (the COTS mind set), with all other requirements a distant second at best. Some additional new requirements that now must be met include adherence to federally mandated software standards, such as POSIX, GOSIP, and the ever often cited IEEE Mass Storage Reference Model V4.0.

All of these factors add up to an extremely difficult set of orders that the Systems Integrator must march to. Within the following section we will come to grips with many of these problems.

Solving the Quandaries facing the Systems Integrator today:

It is our concerted opinion that "Innovation in Approach" is the key to meeting the challenge at hand. Adherence to tried and true solutions of the not too distant past just aren't acceptable any longer. Each systems requirement must be met as an entirely new challenge with no preconceived mind sets dragged along as excess baggage. This philosophy however must be tempered against the tendency to become romantically attached to the newest latest greatest technology and mistakenly use it to try and solve a problem for which it was never intended (as was the case when optical disk was first introduced).

The traditional tools of data storage have been solid state memory (CPU based), rotating magnetic disk (CPU and network attached), and magnetic tape (on and off-line). For the most part these tools have suffered from a very conservative design approach in order to achieve high reliability, most times at the expense of performance and volumetric efficiency and high unit costs.

The basic technologies supporting these tools have seen dramatic improvements in respect to performance and volumetric efficiency over the past five years, but these benefits have been primarily passed on to the consumer and PC/workstation markets. In order to solve the storage dilemmas we find today, these technologies must be applied in a broader sense to the high performance computing environment. Some instances of this can be seen in the advent of SSDD's (DRAM based), low-cost RAID systems, and the use of television broadcast helical scan recording technology for data storage (19mm DD-1 and DD-2, and 1/2" D-3)[5][2]. These technologies offer a high level of performance in terms of greater signal bandwidth and data capacity and are highly volumetrically efficient and reliable (99.00+% availability), but have yet to enter the mainstream in great volume. Tools of this ilk are the saviors of the future in our opinion.

Other storage technologies that are entering the mainstream are enhanced optical disk and the first generation of optical tape drives. Optical disk storage has had a very difficult time in penetrating the high performance computing marketplace because of its low bandwidth, long latency, and high cost in respect to other technologies. This trend is slowly changing and optical disk is expected to have its place in the hierarchy of mass storage in the years to come. The interesting new optical technology is that of tape. CREO and ICI have collaborated on an early entry with this technology (open reel based) and new offerings are in the works from LaserTape, Newell and STK (cartridge based). These systems offer very high capacities in a small form factor with modest data rates (3-4.5 MB/s currently).

---

[5] For further information see:
a - THIC, March 1990, "Interfacing 19mm Helical Scan Recording Systems to Computing Environments".
b - 10th IEEE Symposium on Mass Storage Systems (vendor paper), May 1990, "19mm Helical Scan Recording Technology for Data Intensive Computing Environments"
c - THIC, October 1990, "19mm Data Storage Applications" Richard Lee et al.

It is clear to us that in order to meet the varied needs of the end-user today you must choose from all of the available storage tools a hierarchy of devices to solve the problem at hand. Systems of today and into the future will be a hybridization of all of these technologies. Each device will be used in the hierarchy where best suited (an open systems hardware approach). As time goes by each component can be upgraded or replaced with the latest-greatest device to continue the value of the hybrid approach, without scrapping the entire system.

"Innovation in approach" is not strictly limited to hardware or systems architecture. The choice of software tools is equally as important. There are many approaches to file management in use today. Some are proprietary single manufacturer approaches and others are collaborative amongst end-users, and manufacturers. All claim to be open architected and compliant with the emerging IEEE Mass Storage Reference Model (whatever that means[6]). These file management systems must also be compatible with government mandated standards such as GOSIP and POSIX[7]. This does tend to limit the field at this point in time, but everyone will have to be compliant at some point in the future in order to survive.

Amongst the myriad of commercially available file management software packages available (Andrew, E-Mass, Mesa, Unitree, UNICOS FMS, and others), all approach the management of bitfiles on a hierarchical basis. Files are mounted, dis-mounted, and migrated through a hierarchy of storage devices based on frequency of use and relative priority, with access security threaded throughout. These systems are all adept at their task and differ only in philosophy and approach. Choosing one of these systems is a much more difficult task than architecting and configuring the hardware portion of the mass storage system. Attendant with the need to innovate in terms of approach is the need to reduce storage costs incrementally. Storage related costs in the supercomputer center are now approaching 50+% of the entire capital budget in most facilities. The size of the capital budget in the future will diminish, but the amount of storage required will continue to increase. This mandates the use of low cost (relative) storage devices and attendant media. Only by aligning the requirements of the supercomputing data center with emerging mass produced storage technologies can this dilemma be solved. This points towards technologies that have applications in other, more commodity driven markets, such as consumer electronics, PC/Workstations and broadcast television. As mentioned earlier both RAID and helical scan magnetic tape come from these backgrounds and bring not only higher levels of performance and volumetric efficiency but substantially lowers costs as well (RAID disk = $1-3 pr/MB, HS Tape = $1.00 pr/GB).

The architecture of most mass storage systems today is comprised of a dedicated File Server CPU, interconnected on one side to a network or the supercomputer (via a wide bandwidth peripheral channel) and on the other to a myriad of storage devices/systems. The management of activity within the dedicate file server is handled by the file management software which behaves like a large disk drive to the host supercomputer or network. This approach has proven to be the benchmark today, but is very expensive and wasteful. Many facilities require a supercomputer similar in capabilities to its host to act as a file server in order to have enough available high speed peripheral interconnects available (the file server acts as a "governor" to the entire computing facility as it controls the flow and speed of all devices connected to it.

This approach has worked for some time now, but will not survive in its present capacity into the future. The use of FDDI and HiPPI fabrics with intricate switching networks will soon obsolete this approach. The elimination of an expensive CPU will be a great cost and time savings to the supercomputer center. The use of these wide bandwidth "fabrics" will also allow the interface of new HiPPI/IPI peripherals directly to the host supercomputers. This will speed up system performance by orders of magnitude.

---

[6] After two years of work, the IEEE Storage Systems Working Group would just as soon have no one mandate that a storage system be compliant to the Mass Storage Reference Model (V4.0 or earlier) as the newest thinking is quite different as to when these "models" were conceived in the minds of the IEEE MSRM executive committee.

[7] The reconciliation of TCP/IP protocols against the OSI FTAM's has created a wide rift in both the end-user and manufacturer's communities.

We see the future as one where simplicity in approach will be the winning solution. The ultra intricate, cobbled together, dedicated file servers of today will be replaced by wide bandwidth, direct connected, volumetric efficient, peripherals in the not too distant future. This approach is the only one which will allow the supercomputer CPU to ever achieve its full potential and pay back to the end-user and his sponsors.

Conclusions:

Within our brief overview of the Mass Storage marketplace and the "Grand Challenges" that it presents to the Systems Integrator we have attempted to show that a new order must emerge in order to meet the end-users requirements and yet be affordable in terms of procurement, and flexible in terms of future growth. Only by accepting a new paradigm in terms of architecture and approach will the supercomputing industry ever be able to harness the ever growing "Crisis in Mass Storage".

THE MODERN HIGH RATE DIGITAL CASSETTE RECORDER

Martin Clemow
Penny & Giles Data Systems Ltd
The Mill, Wookey Hole,
Wells, Somerset BA5 1BB
England

INTRODUCTION

The magnetic tape recorder has played an essential role in the capture and storage of instrumentation data for more than thirty years. During this time, data recording technology has steadily progressed to meet user demands for more channels, wider bandwidths and longer recording durations. When acquisition and processing moved from analogue to digital techniques, so recorder design followed suit. Milestones marking the evolution of the data recorder through these various stages - multi-track analogue, high density longitudinal digital and more recently rotary digital - have often represented important breakthroughs in the handling of ever-greater quantities of data.

Throughout this period there has been a very clear line of demarcation between data storage methods in the "instrumentation world" on the one hand and the "computer peripheral world" on the other. This is despite the fact that instrumentation data, whether analogue or digital at the point of acquisition, is now likely to be processed on a digital computer at some stage. Regardless of whether the processing device is a small personal computer, a work-station or the largest supercomputer, system integrators have traditionally been faced with the same basic problem - how to interface what is essentially a manually controlled, continuously running device (the tape recorder) into the fast start/stop computer environment without resorting to an excessive amount of complex custom interfacing and performance compromise.

The increasing availability of affordable high power processing equipment throughout the scientific world is forcing recorder manufacturers to make their latest and perhaps most important breakthrough - the computer-friendly data recorder.

This paper discusses the operating characteristics of such recorders and considers the resultant impact on both data acquisition and data analysis elements of system configuration.

BRIDGING THE GAP

Traditional multi-track recorders (both analogue and high density digital) take the timebase of the information to be recorded for granted. The tape runs continuously at an appropriate speed and data is applied to the input for the duration of the experiment or process. Just like the trace on a paper chart recorder, the record is in a simple Y-T form, with "Y" being represented by the magnetic flux pattern on tape while the "T" information is contained in the tape motion itself. If a recorded tape is re-wound and replayed at the same speed and in the same direction, the output is expected to be a close representation of the original input data, including its timebase. Timebase compression or expansion can be achieved by increasing or decreasing the tape speed. Time inversion is also possible by reversing the direction of tape movement. The important point is that an indication of the passage of time is inherent in the operation of the classical data recorder.

Until now, this feature has been both a strength and a weakness. A strength in terms of the ability to manipulate the passage and direction of time on a recorded experiment during the analysis process, but a weakness when it is necessary to input the data to a computer in anything but the simplest free-running mode. Given that most computers require data to be input to disk or memory in chunks at a fixed rate, it is not a simple matter to control the data flow from a constant speed system efficiently without recourse to time-consuming stop-

reverse-restart routines. In contrast, computer peripherals start and stop rapidly in order to control the flow of data. This latter attribute would, therefore, appear to be a necessary characteristic for a data recorder to be considered as computer-friendly.

In addition to fast start/stop of the tape itself, some high rate digital cassette recorders incorporate input and output data buffering to allow the tape transport to start and stop during data transfer as necessary. The buffer capacity will be determined by the need to ensure that all possible sequences of tape movement (ramp up, ramp down, etc.) can be accommodated without loss of data.

The use of buffered data input/output, while greatly simplifying the actual transfer of data, introduces more wide-ranging implications than might at first be obvious. For a user to gain the maximum benefit from the closer integration of the recorder into the computer environment, it becomes necessary to consider the whole data acquisition and analysis process rather than just the recorder itself.

If we accept the fundamental principle that computers need to clock data into memory in bursts by starting and stopping the tape, how are we going to retain the important timebase information which was so conveniently available by the very movement of the tape on a continuously running system? This consideration leads naturally on to the actual control of data. On the command to start, traditional data recorders ramp gently up to speed, lock in and then data is available on the correct timebase. When told to stop, they ramp gracefully down again to rest. If "good" data has been recorded on the tape at these ramping points, it is effectively lost or at least corrupted due to the slewing of the tape speed.

This is clearly unacceptable for reliable data transfer so a subtle change of emphasis is needed. It is important now to think in terms of controlling the flow of data - not the movement of the tape itself.

Computer friendliness also implies reliable and convenient data management. It is relatively easy to append housekeeping data during recording, but what type of data will be most useful, and how can it be used to best effect? For example, if the user intends to search his records by date, time or event, it is critical that he develop an overall strategy for the creation, logging and management of this type of auxiliary information.

DATA FORMATTING

Intuitively, it would seem desirable to establish a common data format throughout the data capture and processing path if only to avoid the complexity and cost of unnecessary format conversions. This philosophy requires an analysis not only of the way data is to be recorded, but of the whole network (both current and planned future expansion) to establish, for example, the best word width to use (for example: 8, 16 or 32 bits). Some recorders support only 8-bit formats while others can be user configured for all three formats. If a common interface format can be used throughout, the total system can be greatly simplified.

If the source data is serial in nature, it is important to decide carefully when to convert from serial to parallel. In general, high rate serial recording channels are difficult and expensive, so it is often best to perform the conversion before recording. The use of standardizing on a common data interface format will generally reduce overall system complexity and cost, with the added benefit of increased flexibility and equipment utilization.

BUFFERED DATA TRANSFER

It is most unlikely that the clock rate of the acquisition process (e.g. analogue-to-digital conversion) will be identical to that of the analyzing computer. This means that a change of timebase is almost certain to be required somewhere within the data path. Looking at the complete system, several important points should be considered. In any recording system, if the tape is to be used efficiently data should be recorded on tape at the maximum rated density.

In the case of a continuously running system (longitudinal or rotary), this has traditionally meant adjusting the tape speed (and scanner speed, if appropriate) to match the input or output data rate. However, when the recorder incorporates a read/write buffer, it is usually arranged so that data is written to or read from tape at a single, fixed rate and tape speed.

Input/Output rates below the recorder's specified maximum will result in its buffer filling or emptying at a slower rate. The recorder accommodates this by automatically stopping the tape until such a time that the level of data in the buffer reaches a pre-determined level. The rate at which data is written to, and read from tape is, therefore, completely independent of user data transfer rate. This severance of the traditional direct link between user data transfer rates and tape read/write rates means that a buffered system can also accommodate data which is not continuous (i.e. intermittent or burst data) and be able to operate at any user controlled transfer rate (continuously variable) within its rated range.

Clearly, the buffered approach would appear to have important advantages for computer based applications, particularly if the tape drive is specifically designed for very fast start/stop operation - thereby necessitating only a relatively small data buffer.

An interesting additional benefit, which should not be overlooked, is that buffered systems do not have to actually be in the normal recording mode (with tape running) in order to capture, say, an unexpected transient event. They can wait in standby mode until the event commences and then data can be written to tape from the buffer as previously described. This reduces wear and tear not only on the recorder itself, but also on heads and media in the case of fixed-head systems where nothing is in motion until data is transferred from the buffer on to tape.

Similarly, when reading data at a low transfer rate, tape motion only occurs as necessary to maintain a level of data in the buffer commensurate with the user transfer rate.

AUXILIARY DATA

While we have seen that the buffered approach has much to commend it with regard to the handling of different (and perhaps variable) input/output transfer rates and computer entry, there remains the problem of the consequent loss of relationship between timebase and tape motion since, as we have already discussed, the tape only moves when data is passing between tape and buffer. If timing is already intrinsic in the user's data stream - for example, where the input clock is synchronous with the analogue-to-digital sampling process - only periodic updates may be necessary in order to keep everything under control. Alternatively, more precise timing information may be required. Some high rate digital cassette recorders incorporate an internal clock which is written to a separate (auxiliary) track in the form of a date/time code. This timing information may subsequently be used to support high speed search during replay.

Another useful method of providing reference information is by using event markers. On some recorders, the controlling computer can write unique event markers along with event ID character strings to the auxiliary track. These can be scanned at high speed in order to locate selected records and also to provide an event log or directory of all events on a tape. With buffered systems, users should expect this information to be recorded in synchronism with its associated user data in order to maintain the necessary precise relationship between the location of the event marker and the data to which it refers.

COMMAND AND CONTROL

Clearly, significant improvements over traditional methods of l of data recorders are needed if systems are to be integrated successfully into the computer environment. Typically, commands and status requests pass between the recorder and controller via a conventional communications interface such as IEEE488 or RS-449.

DATA FLOW

The control of data flow in continuously running systems is relatively simple since it is only necessary to start the tape running (at the correct speed) and allow data to flow in or out of the recorder. With buffered systems, however, the movement of the tape itself is a secondary issue as this process is automatically controlled by the action of the recorder attempting to empty or fill its buffer. One advantage of a recorder which has been designed with an integral buffer is that it should not be possible to either overfill or empty its buffer during data transfer operations.

With continuous inputs, this may simply mean ensuring that the input clock rate does not exceed the rated maximum for the recorder. If the input is in the form of burst data - blocks of finite length with gaps in-between - it is generally permissible to exceed the maximum continuous rate for short periods. In the case of such "burst" data, it is advisable to implement a "hand-shaking" protocol so that the recorder can control the flow of data within the capacity limits of its buffer.

On replay, the situation is slightly different since it should be possible for the computer to control the transfer of data in accordance with its own needs and activities. Here, a hand-shaking protocol is essential since the mere fact that the computer may have requested data does not in every case mean that data will be immediately available. Consider the situation where a new cassette has been loaded into the transport and placed at the beginning-of-tape but no other tape movement has yet taken place. The computer may request data and offer an output clock, but the recorder's buffer as yet contains no data. Instead, the recorder will acknowledge the request for data and immediately start to move tape in order to fill the buffer. At a certain point, there will be sufficient data within the buffer for an output transfer to commence. As long as the computer continues to demand data, the recorder will maintain an appropriate level of data in its buffer, starting and stopping the tape as necessary. At some point in the transfer process, the computer may decide that it has sufficient data and cease to request further data. Recognizing this, the recorder will discontinue the reading process although some valid data may remain in the buffer ready for transfer later.

A convenient method of achieving this is to use a common, bidirectional data input/output interface including hand-shaking lines which control the flow of data to and from the recorder. For example, a DATA READY signal may be asserted by the recorder to indicate that it is ready to receive data and a USER DATA ENABLE may be asserted by the user to indicate that applied data is valid. When reproducing, a DATA READY signal asserted by the recorder means that valid data is available, while USER DATA ENABLE is asserted by the user to indicate that he is ready to accept outgoing data.

MEDIA

Our discussion hitherto has dealt with the general issues involved in integrating the attribute "computer-friendliness" to data recorders and is basically independent of the choice of media. The trend throughout all classes of recording is towards the use of standard cassettes. There is actually a paradox here since modern open-reel tapes can contain an enormous amount of data and represent the most efficient method of storage by volume. (Remember that every cassette in effect contains an empty reel of similar volume to the media itself.) Open reels may not be convenient to load or keep free from contamination and are therefore considered "unfriendly" by some users. Conversely, cassettes are convenient to load, both manually and automatically, and their acceptance is now almost universal.

Although a full discussion on the range of cassette media is beyond the scope of this particular presentation, many equipment designers now elect to use commercially available multi-sourced cassettes rather than to develop custom-designed media for reasons of economics and availability.

TOWARDS A 1000 TRACKS DIGITAL TAPE RECORDER

J. M. Contellier, J. P. Castera, J. Colineau, J. C. Lehureau
Thomson CSF, Laboratoire Central de Recherches
Domaine de Corbeville
Orsay Cedex F-91404
France

F. Maurice, C. Hanna
Thomson Consumer Electronics, R&D France
Ilkirch F-67403
France

INTRODUCTION

As the demand for high data rate (up to 1 Gb/s), high density (down to 1 μm^2/bit) tape recorder increases, the main investigation trend is an improvement of the well known helical scan concept. The drawbacks of this technology are also well known; sophisticated mechanics, head to tape contact and wear problems. In our fixed head approach, the recorder mechanics is made much more simple, but the complexity is turned towards the integrated magnetic components, which have to record and reproduce hundreds of tracks in parallel. Our multiplexed write inductive head and magneto-optical readout head will be described, and the global system performances evaluated.

RECORDING HEAD

To avoid the impractical number of connections necessary for addressing individually a large number of tracks, the heads have been arranged on a matrix array of rows and columns. A conventional addressing technique is used to multiplex the recording process. Each head is located at the crossing of two coils, the row wire being used to feed the data to be recorded, and the column wire to select the desired elements. The present multiplexed write component is composed of 32 (data) x 12 (selection) = 284 heads.

A planar head technology has been developed for the thin film pole realization. The top part of the head is then a flat surface, about 7 x 3 mm large, designed to record on a 8 mm ME tape. The bottom part of the head is a mix of conventional ferrite grooving, coil winding, glass fusion and polishing. Recorded track width is 18 µm.

A two bean interference method, using a monochromatic light has been used to characterize the head to tape contact, with about 10nm resolution. The interference pattern takes place between the tape and a dummy glass component where the protuberant magnetic pole shape of the multitrack head has been reproduced. In this experiment, the tape can be static or running. It has been shown that temporal and spatial homogeneous close contact can be achieved between a moving magnetic tape and a large active area head. The head to tape average spacing is directly correlated to the tape roughness (about 50 nm, measured by atomic force microscopy) and does not vary significantly with the applied pressure (typically around 2.10^4 PA). Therefore, the head to tape contact is as good as it is for a rotating head.

Signals recorded with a multiplexed head have been compared to signals recorded with a state of the art 8 mm MIG single track head. The output/current curves show a similar maximum output level for both heads. For optimized currents, output/frequency curves are identical.

READOUT HEAD

The head to tape speed is very low in our system: 2.6 cm/s. Only an active readout device can then be considered.

A simple transducer has been realized to pickup the magnetic flux from the full width of the tape: on a GGG substrate, 2 magnetic layers are separated by a non-magnetic gap layer. The tape is running on the edge of this 3 layer assembly. The magnetization change in one of the layers, due to the recorded tape proximity, is analyzed using the well known Kerr effect. The full magneto-optical device is then made of a laser diode, the magnetic sensor, a few lenses or mirrors, a polarizer and a linear CCD. The laser spot is focused on the full sensor width, in such a way that each track magnetization will be imaged on a different CCD pixel. No laser bean deflection is needed.

The signal over noise ratio of this head is proportional to the laser diode power, the magnetic efficiency and the figure of merit of the transducer. The use of Sendust for the sensor magnetic layers, and the optimization of the layer thicknesses has led to a 4-% magnetic efficiency. The figure of merit of the transducer has reached 4.10^{-4}. With a 50 MW laser diode, a good enough 26dB peak to rms, full band signal over noise ratio has been obtained to reproduce 20Mb/s on our present demonstrator. The recorded bit length is 0.5µm.

DIGITAL PROCESSING

A conventional 8-10 modulation code has been used to adjust the channel to the magneto-optical head characteristics. The output signal has to be equalized and the clock has to be recovered for each independent track. It has been done at a reasonable cost by multiplexing all tracks in a pipelined architecture. Signal is digitized right at the CCD output.

SYSTEM PERFORMANCE

The raw bit error rate measured for the overall system is in the range 10^{-5}. A Reed Solomon error correcting has also been implemented, and the system interfaced with a video codec. A digital video demonstration is now settled in our laboratory.

CONCLUSION

A new concept of fixed head recording has been demonstrated, with state of the art performances. The advantages of such a system over conventional rotating heads are numerous. The simple and reduced mechanics involved will lower the price of the recorder. The low head to tape speed decreases tremendously head wear and tape damage. For space application, the absence of gyroscopic effect due to the high speed rotating drum, the possibility of backward readout may be essential.

N93-30470

EVOLUTION OF A HIGH-PERFORMANCE STORAGE SYSTEM BASED ON MAGNETIC TAPE INSTRUMENTATION RECORDERS

Bruce Peters
Datatape Incorporated
360 Sierra Madre Villa
Pasadena, CA 91109

S21-35
159111
P. 4

INTRODUCTION

In order to provide transparent access to data in network computing environments, high-performance storage systems are getting smarter as well as faster. Magnetic tape instrumentation recorders contain an increasing amount of intelligence in the form of software and firmware that manages the processes of capturing input signals and data, putting them on media and then reproducing or playing them back. Such intelligence makes them better recorders, ideally suited for applications requiring the high-speed capture and playback of large streams of signals or data.

In order to make recorders better storage systems, intelligence is also being added to provide appropriate computer and network interfaces along with services that enable them to interoperate with host computers or network client and server entities. Thus, recorders are evolving into high-performance storage systems that become an integral part of a shared information system.

Datatape has embarked on a program with the Caltech sponsored Concurrent Supercomputer Consortium to develop a smart mass storage system. Working within the framework of the emerging IEEE Mass Storage System Reference Model, we are building a high-performance storage system that works with the STX File Server to provide storage services for the Intel Touchstone Delta Supercomputer. Our objective is to provide the required high storage capacity and transfer rate to support grand challenge applications, such as global climate modeling.

REQUIREMENTS

Reliable, high-performance storage is a basic requirement of emerging network computing systems used for analytical problem solving applications. With the advent of mixed media data types, including computational digital movies, storage must accommodate bitfiles in excess of one gigabyte. In order to move these bitfiles in and out of storage without bottlenecks, transfer rates must exceed ten megabytes per second. Access time must be predictable within reasonable human-interaction parameters, which is normally in seconds or minutes. Access must be provided with high data integrity on the order of one error in 10E12 bits or better. Data security must be provided through controlled access.

The Concurrent Supercomputer Consortium has these requirements. In order to support the n-dimensional , nonlinear modeling of the grand challenge applications, large bitfiles up to 100 gigabytes and high transfer rates at HIPPI speed (up to 50 megabytes per second) are needed. Data integrity must reach the order of one error in 10E15 bits. Standard interfaces (e.g., HIPPI) and protocols (e.g., the IPI-3 command set for high-performance virtual disk) are needed. Compartmentation of data, such as storing separate bitfiles or classes of bitfiles on separate removable media and providing controlled access to the media, is an acceptable approach to data security.

Thus, storage must be provided as a subsystem characterized by the full range of systemic parameters, such as performance, functionality, security, and reliability, maintainability, and availability.

BASIC STORAGE SYSTEM

Initially, Datatape will supply the Consortium with two DCTR-LP400 Digital Cassette Tape Recorder/Reproducers (LP400s) capable of sustained transfer rates up to 400 Mbps, two Variable Rate Buffers (VRBs) capable of buffering 384MB of data with burst transfer rates up to 480 Mbps, and two HIPPI interface modules. Protocols and commands are being developed to manage the control and data paths.

The LP400 is a high-performance 19mm magnetic tape recorder that is capable of recording and reproducing digital data rates from 50 to 400 Mbps on the small, medium or large commercially available D-1 tape cassette. The large tape cassette stores nearly one terabit of data. The LP400 handles wideband data via 8- or 16-bit parallel I/O and complies with the ANSI-ID-1 format with a bit error rate of 1 error in 10E10 bits. Each set of four tracks (a track set) is addressable by the corresponding track-set identification, recorded in the control track. Local control is provided via a remotable control panel. Remote command and status operation is provided via IEEE-488 or RS-422 interfaces.

The Variable Rate Buffers (VRBs) are used to extend the recorders from being instrumentation recorders to being computer peripherals. A VRB and an LP400 recorder make up a peripheral storage unit. The VRB transfers data to and from the host computer via a HIPPI interface in bursts determined by the specific characteristics of the host interface, and it transfers data to and from the recorder in the continuous streams that the recorder uses. The VRB features automatic rewrite, whereby bad or marginal areas of tape are skipped over. This feature enhances data integrity to better than 1 error in 10E12 bits. In addition to HIPPI, the VRB can accommodate other host interfaces such as SCSI, SCSI-II and FDDI.

The HIPPI interface module has separate data and control interfaces, supporting peer-to-peer data transfers. The HIPPI data interface is fully compatible with the relevant HIPPI standards. It is a dual simplex configuration; either the receiver or the transmitter can function separately, but not both at the same time. The data path is 32 bits wide and transfers data at a rate of 800 Mbps. The HIPPI control interface is an Ethernet port. The command/status set is modeled after the Maximum-Strategy version of the IPI-3 command set for disk arrays.

STORAGE SYSTEM EXTENSIONS

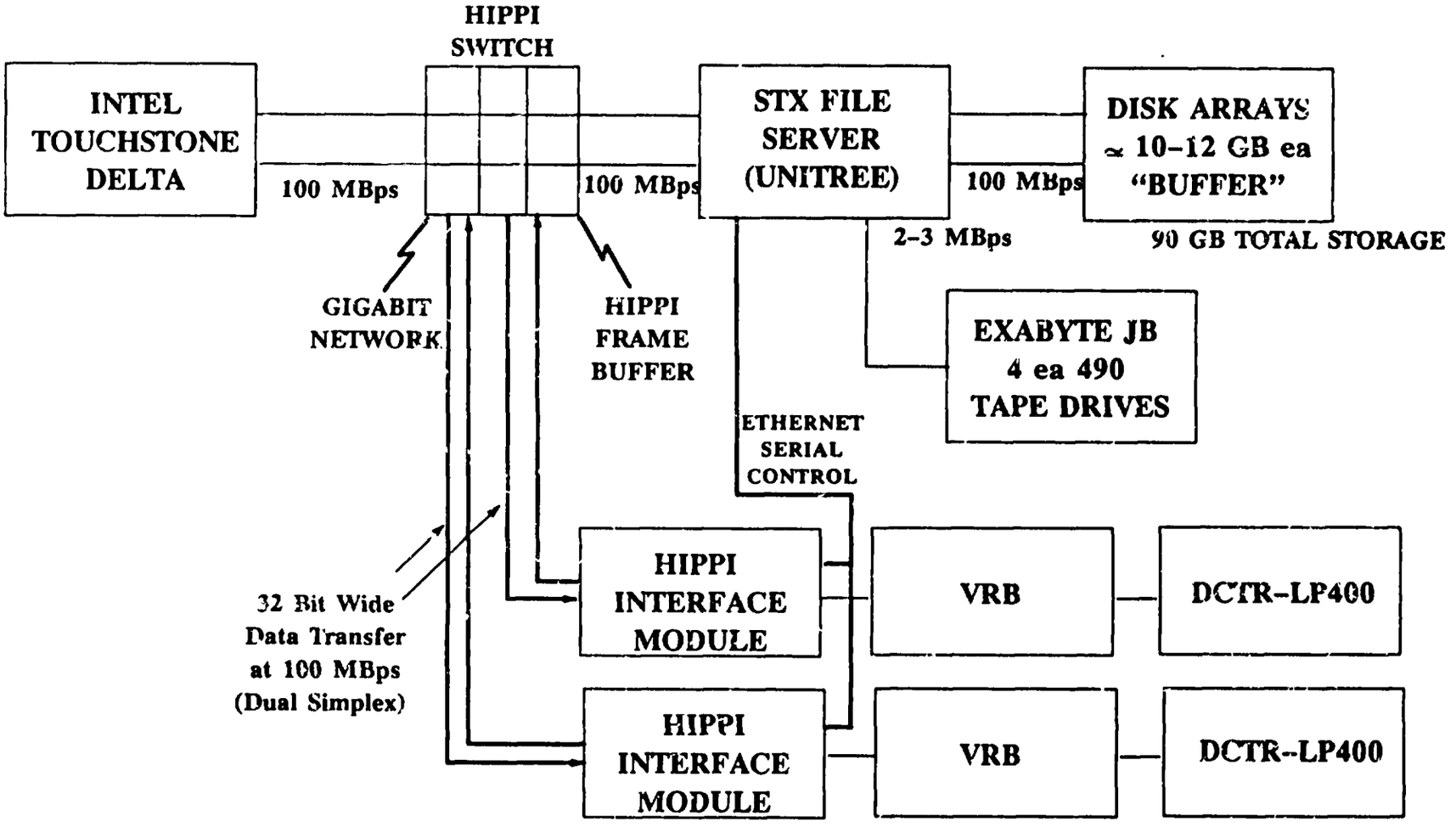
In parallel to the development of the wide-bandwidth interface, Datatape is developing additional functionality to improve and extend the storage systems' performance and scalability. For example, third-loop EDAC will be added to the VRB to improve the corrected bit error rate to 1 error in 10E15 bits.

Robotic library capabilities are being developed to evolve this magnetic tape peripheral storage system to a hybrid, hierarchical mass storage system. Our plan is to provide capabilities as a Physical Volume Repository or Physical Volume Library, interfacing in the Storage Server to support data transfers using either physical or logical file names. A carousel-and-picker module is being designed to handle multiple sizes of cassettes. A feature will be provided to enable multiple, independent accesses to a single carousel, which virtually guarantees access to any cassette. It also extends the storage-capacity growth potential of the system. The control framework is being structured to accommodate heterogeneous storage drives and media, such as magnetic and optical disk, as well as magnetic tape.

FUTURE SMART STORAGE

The ultimate goal is a coherent, balanced high-performance storage system that can be configured and adapted to specific operational, technical and economic requirements. Such a system will meet the basic goals of the IEEE Mass Storage System Reference Model: open architecture for general purpose storage systems; applicable to distributed systems as well as centralized and standalone systems; and scalability. In addition, such a system will provide services to facilitate the collection, processing, analysis and dissemination of data. Examples include signal processing, data reduction and enrichment, compression and encryption.

CALTECH SUPERCOMPUTER ARCHITECTURE



93-80471

MASS OPTICAL STORAGE - TAPE (MOST)

William S. Oakley
554 Greenmeadow Way
San Jose, CA 95134

BACKGROUND

In today's large mainframe and supercomputer environment there exists a continuous demand for increased performance in digital storage systems. The user need for near-line storage capacity is currently doubling every four years. In addition to higher capacity, a desire exists for higher data transfer rates, and longer term database archivability, at lower, and lower, cost. Each component of this quartet of demands appears to be insatiable. Magnetic tape technology presently dominates the digital mass storage markets, but the continuous growth of requirements is drawing attention to the limitations of the technology as an archival mass storage medium. The lowest cost option currently available for long term data storage is to use magnetic tape, although it is not well suited to meeting the need for many tens of years of reliable storage. Today, the majority of magnetic tape mass storage systems are based on the IBM 3480/3490 (or compatible) tape drives. These drives offer only moderate transfer rates and relatively small increments of storage, both of which create a logistics problem due to the large numbers of cartridges necessary in a typical system and the time taken to transfer data. Both higher cartridge capacity and data transfer rates are available in some helical scan magnetic tape systems; however, these command a substantially higher price, exacerbating the cost problem, and are not compatible with most installed systems or tape databases.

To speed data access a variety of IBM 3480 compatible robotic tape cartridge servers have been implemented with capacities up to 6,000 cartridges. These provide more acceptable access times, but individual cartridges continue to provide only small capacity increments. Although this is the industry preferred solution at present, it results in massive, expensive robotics, with slow access to only a few terabytes of storage, and does not address the needs of very long term archivability or higher data transfer rates.

A surge of interest in optical storage has recently been created by the advent of optical discs. These have been seen as a potential solution to the storage capacity problem and few large systems have been implemented which employ up to 14 inch diameter discs storing several gigabytes of data per side. At present, the data transfer rates of existing drives is very low, typically under a megabyte per second, and their cost remains high. Regardless of this, optical disc technology has found some interest in the storage community, and sales of optical disc systems are expected to reach over $2 Billion by 1996. Both write once and erasable technologies are available, with most current interest going to the smaller erasable magneto-optic systems.

For larger systems, the few gigabytes of storage offered by a single optical disc is much too low, and robotic mechanical "Jukeboxes" have been implemented containing hundreds or perhaps thousands of discs. These systems are generally also inadequate, offering only a temporary and incomplete solution, due to limited data transfer rates, slow disc access, large physical size, and substantial cost. Reliability and maintainability problems also exist due to the mechanical nature of the approach and the need for many disc interchanges. A much better solution is required for large system mass storage.

It is inevitable that the tape storage market will soon see the introduction of very competitive products based on Digital Optical Tape (DOT™) technology. This will be particularly true in those market segments requiring large databases, due to the markedly superior archival properties and storage capacities of optical tape. Several manufacturers are introducing various types of write once (WORM) archival optical tape media. Little interest appears to

exist at this time for erasable tape, although both magneto-optic and dye polymer erasable tape have been demonstrated . This lack of interest in erasable tape stems from the current industry orientation toward archival mass storage systems, and could change in the future if digital TV video recorders adopt optical tape technology. In the near term, the emergence of optical tape products, with their price/performance benefits over magnetic systems, will probably greatly expand both the capabilities, and the market volume, of high end tape based systems.

LaserTape Systems Incorporated has been researching a new digital storage product based on laser writing onto optical tape. This Digital Optical Tape System (known as DOTS™), is targeted at the large computer mass storage market. The DOT™ system utilizes the IBM 3480 removable cartridge, which contains about 160 meters of one mil (0.001inches) thick, half inch wide tape. This length of tape can provide up to 50 GigaBytes of user data when allowance is made for error correction (ECC), track spacing, headers, etc. Half mil thick tape is also available, and provides nominally 100 GigaBytes of user data per cartridge. The initial systems use laser diodes of 830 nanometer wavelength as the write/read source, and provide a one micron recorded bit size. Thus the optical tape areal storage density is the same as for optical discs.

The input/output (I/O) data transfer rates of the LaserTape drive can be between 6 and 15 MegaBytes per second, depending on the particular optical tape media used. Several different types of optical tape media exist both from U.S. and foreign sources, and all are compatible with the LaserTape system. The archival lifetime of all media types is expected to substantially exceed 25 years, and tape vendors are working towards establishing 100 year archivability. Bit error rates, after correction, are expected to be better than 10^{-13} for the LaserTape drive.

Rapid increases in both cartridge capacity and system data rate are anticipated in the future. Short wavelength lasers operating in the green, blue, and U.V. regions of the spectrum are in development, and these sources will enable approximately an order of magnitude increase in both tape areal storage density and data transfer rate. By 1995 the DOTS™ technology should be capable of storing about half a Terabyte in a single 3480 cartridge, with I/O data transfer rates of over 100 Megabytes per second.

SYSTEM CONFIGURATION CONSIDERATIONS

Today's large systems are configured using three level memory systems. The central processing unit (CPU) interacts with local fast random access (RAM) primary memory, which itself interchanges data with on-line secondary (disc) storage. The disc storage is loaded on demand with the desired files from tertiary storage, which is invariably either off-line operator or robotically accessed tape. Future system architectures will probably retain this basic heirarchy with changes occuring at each functional level as storage technology advances. CPU's are migrating to multiprocessor systems, such as Thinking Machines Corp.'s new CM-5, a massively parallel computer which uses up to 2,000 of Sun Microsystems SPARC microprocessors. To support these and similar systems RAM is becoming larger and faster. Distributed systems are rapidly becoming the order of the day as networks effectively become the system backplane.

Today's rotating disc systems will someday be replaced by larger capacity, much faster systems using different technology such as, perhaps, optical holographic storage. This type of system is already in development as evidenced by the Holostore system being developed by MCC in Austin, Texas. This approach potentially offers several gigabytes of storage with microsecond access times. The Holostore technology uses volume holographic storage in optical crystals. It is a page oriented device that writes and reads data in a two dimensional optical form using a laser source. The system is physically small, has no moving parts, and is a parallel access device capable of very high transfer rates.

Development of high data rate mass storage systems are not, of course, dependent on the success of the Holostore technology. Magnetic disc systems currently available substantially meet the needs of such a system, except for some I/O rate nd latency limitations in the RAM to disc interface. Current wisdom has it that each MegaFlop (a Million Floating point arithmetic operations per second) of processing power requires about five megabytes/second of I/O. This means that today's 2 MegaFlop (about 10 MIP, RISC microprocessors require I/O rates of 10 Megabytes per second from RAM. It follows that the input data rates from either the Holostore, or from disc to RAM, should be similar. Even given a certain amount of reuse of data in RAM or disc for a particular computation, the data I/O rates required from tertiary storage are not significantly lower. This is particularly true if the average file size substantially approaches the secondary storage capacity, requiring frequent file transfers.

For supercomputer and mainframe systems, the CPU rates are in the hundreds of MegaFlops, and file sizes are often in many hundreds of megabytes. This results in a need for tertiary storage I/O rates in the range of a hundred megabytes per second. Access time to a required data set is also a factor of considerable importance. To support this compute intensive environment, secondary storage using either discs or a Holostore system will be required with a storage capacity of one or two gigabytes. Downloading bitfile data sets to such a system will require frequent transfers of hundred megabyte size files. File transfers of this size can be required every few tens of seconds, i.e. on a continuous basis . To support systems of this nature, a tertiary storage system of hundreds, or perhaps thousands of gigabytes is required, with continuous transfer rates in the range of a hundred megabytes per second, and access times of about ten seconds.

One possible means of addressing this need is the use of disc arrays. The number of independent disc drives is determined by either the cumulative size of the memory desired, or the cumulative I/O rate desired. Data transfer rates of magnetic disc drives permit high system data rates with only a few drives. However, a multi terabyte memory requires so many disc drives as to be impractical. Disc arrays only offer modest memory sizes therefore do not fit the mass storage need.

The introduction of optical tape drives with a user capacity of a hundred gigabytes per 3480 cartridge and with a data transfer rate of 15 MBytes/second potentially provides a better solution. A system comprising only a single tape drive, with an autoloader containing ten 100 Gigabyte tapes, provides access to a terabyte of data in just a few tens of seconds. This is not a vision for the far distant future. The 3480 cartridge autoloaders, the basic tape moving system, and the optical head fabrication technology already exist. All that remains is to combine the available technology assets into an optical tape drive.

THE TECHNOLOGY

The basic technology to be implemented in producing a high performance tape drive is mostly a combination of two standard technologies. The tape drive mechanism is essentially a standard IBM 3480 compatible magnetic system modified to use optical tape. Virtually all of the standard tape control electronics, including the tape velocity servo, is utilized, and virtually all of the tape movement mechanics is preserved. The standard magnetic head is, of course, removed and replaced by an optical write/read head. The optical head mostly uses technology which has been proven in the optical disc industry. Standard, although improved, optical focus and tracking techniques are used to maintain track following and beam focus on the moving tape. The basic system read/write scheme is shown in *Figure 1*.

The linear stop/start tape system of the IBM 3480 compatible is fully preserved and the system operates at 3 meters per second, between the standard tape speeds of 2 and 4 meters per second. The high data rate is achieved by optically writing a transverse column array of one micron diameter optical bits, with all bits in the array being written simultaneously. Having written a column array of perhaps 48 or 64 bits, the normal tape advance allows a subsequent array to be written in less than a microsecond. In this manner data rates of the order of 100

to 150 Megabits per second are achievable. *Figure 2*, shows a transverse multibit column located between two servo tracks. Servo and data bits are written simultaneously. Data recording is achieved by individual modulation of each bit in the array at about a two megahertz rate. As all transverse bits in an array are recorded simultaneously, an array of 64 data bits, for example, would give a data rate of about 128 Megabits per second.

With a 1.5 micron longitudinal (down tape) spacing, a 3 meter per second tape speed gives a 2 MHz bit rate for each bit in the transverse array. A bit center to center transverse spacing of 1.7 microns is used and arrays of 32 and 80 bits per track are planned, corresponding to user data rates of 6 and 15 Megabytes per second. For the 80 bits wide column the written swath width is about 0.15 millimeters, which permits up to approximately 80 separate swaths to be written across the 12.5 mm tape width. This allows about 6,400 bits to be stored across the tape width, and is the primary reason the system has such a high storage capacity per cartridge.

The multi track format implemented permits quasi random access to data in that it is not necessary to read the entire tape in a sequential manner. A transverse motion of the optical head assembly allows each of the 80 individual tracks to be directly accessed thus providing some aspect of parallel access. Each of the approximately 80 tracks contains 1.5 gigabytes of data and is individually identified by coding within the track format, as is the track position along the tape. Ey this means rapid access to any known file location can be achieved. The location of data within a tape is identified by placing both swath number identity and 'down tape' position data in the servo tracks. This allows the system to continually validate its location on the tape.

The average access time to data on any tape, once loaded, is [illegible] of the end to end tape time of 110 seconds. For a 100 gigabyte capacity tape this provides an average access time of about 37 seconds. If an autoloader of ten cartridges were to be used, the time to exchange cartridges will be about 8 seconds, thus providing an average access time of nominally 45 seconds to any data. Use of shorter tape lengths in a cartridge can obviously reduce average access times within a cartridge.

BIT ARRAY GENERATION

The basic technology of recording onto optical tape has been successfully demonstrated by LaserTape and the key system feature is the means of bit array generation. A variety of methods can be implemented to generate the desired array of modulatable diffraction limited bits. One approach fabricated and tested at Las Tape was based on acousto-optic multifrequency diffraction[1,2]. In this technique a number of radio frequency acoustic waves are input to an optically transmissive crystal by means of a piezoelectric transducer, and form a corresponding set of travelling optical diffraction gratings in the crystal. When illuminated by a coherent optical source, each RF frequency and the resulting diffraction grating in the crystal forms an optical beam at a specific diffraction angle, and thereby a corresponding spot position in the tape plane. Binary modulation of the input RF driving voltage intensity modulates the optical spot, resulting in data recording on the tape.

This technique was implemented in a system fabricated and tested at LaserTape in mid 1991, and successfully demonstrated writing and reading to and from optical tape at data rates equivalent to 6 Megabytes per second. Only 8 of the designs 22 frequency channels were electronically supported, but the system validated the technology of writing/reading to/from rapidly moving optical tape with diffraction limited spots sizes. A limitation of the system was the complexity resulting from implementing the digital modulation and multiple beam generation in the same device. As described in reference 2, several cross modulation effects occur due to using the acousto-optic device for the dual purposes of modulation and beam steering. Techniques were designed which mostly compensated for these effects at the cost of increased electronic complexity. However the basic parameters of acousto-optics and the

steering. Techniques were designed which mostly compensated for these effects at the cost of increased electronic complexity. However the basic parameters of acousto-optics and the complexities arising from the cross modulation limit the usefulness of acousto-optic systems to user data rates below 6 Megabytes per second. An inherently better approach is one in which the multiple optical beam generation and beam modulation occur in separate devices. Designs of this nature are now being pursued, based on the numbers and modulation rates described above.

TECHNOLOGY GROWTH OPTIONS

The initial product planned by LaserTape provides up to 100 Gigabytes of user storage with data transfer rates of up to 15 Megabytes per second. This system is based on available laser diodes operating at a wavelength of 0.83 microns. Use of a shorter wavelength laser source enables a proportionately smaller written spot size, which in turn provides greater storage densities and possibly also higher data rates.

One proposed future implementation employs a frequency doubled Neodymium crystal laser outputing in excess of 100 milliwatts at a wavelength of 0.53 microns (green). The current state of the art in this technology is 140 milliwatts cw, with rapid power increases anticipated in the near future. The amount of optical power available will determine the data transfer rate for a given optical system and tape media sensitivity. For the most sensitive of the available media, a nominal data transfer rate of 100 Megabytes per second should be achieved with 120 milliwatts of average input optical power, allowing for system transmission factors. This 800 megabit per second transfer rate matches the needs of the emerging fibre channel data nets as planned under the U.S. Government HPCC program.

The use of the shorter wavelength will allow recording of 0.5 micron spots with 0.64 micron spacing. For a system using a tape velocity of 4.0 meters per second, a bit spacing of 0.64 microns implies a data rate of 6.25 megabits per second for a single writing point. A data rate of 100 Megabytes (800 Megabits) per second therefore requires that 128 column array data bits be written simultaneously across each swath, giving a swath width of about 80 microns. This in turn will permit a storage capacity of up to 500 gigabytes per cartridge.

Increased laser power will enable a greater number of bits to be written in parallel with correspondingly higher data rates. The present optical systems are estimated to be capable of track widths of 200 microns, potentially providing up to 250 megabytes per second write rates if sufficient laser power were to be available. A source laser power of 1/4 watt would be sufficient to provide a data rate of 200 megabytes per second. This data rate matches the full transfer rate of the HIPPI data communications protocol. It is clear that a significant portion of any program to produce an ultra high performance tape drive should be expended in optimizing the laser source.

Further into the future, laser sources in the ultra violet region of the spectrum may be anticipated. These may provide bit sizes of 0.2 microns with 0.25 micron spacing, allowing even higher data rates and storage capacities of well over a terabyte per cartridge. A bit spacing of 0.25 microns corresponds to an areal density of 10,000 megabits per square inch, which may well be achieved during this decade. Eventually, spectrally selective media and write/read techniques will potentially increase both of these parameters by another two or three orders of magnitude

CONCLUSIONS

It can be concluded that the burgeoning demand for mass storage, quick accessibility, and high I/O data rates, is probably best met by optical tape systems. These are the only systems that can provide the universal quartet of requirements (capacity, I/O rate, archivability, and cost effectiveness) at acceptable levels in the foreseeable future.

1. A. Korpel. Acousto-optics. A Review of Fundamentals.
IEEE Proceedings, Vol 69 No.1. Pgs 48 to 53, January 1981.

2. D. Hecht. Multifrequency Acoustoptic Diffraction.
IEEE Transactions on Sonics and Ultrasonics, VOL.1. SU-24, No.1.
January 1977.

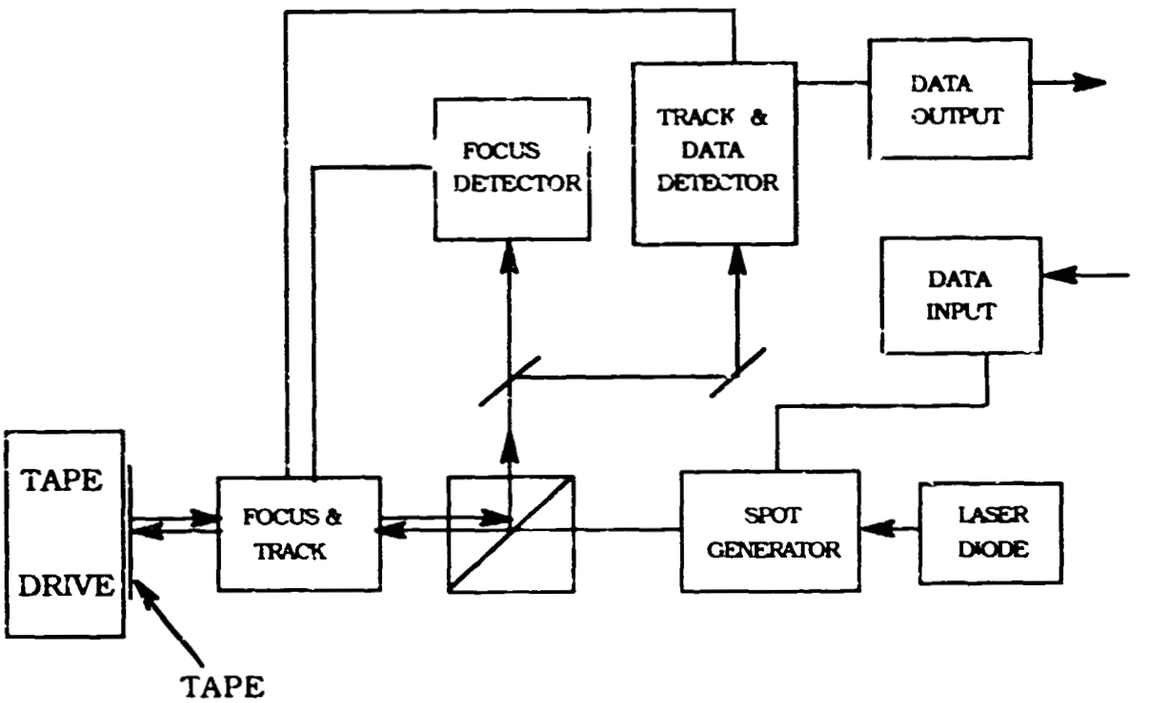


FIGURE 1 BASIC SYSTEM FUNCTIONS

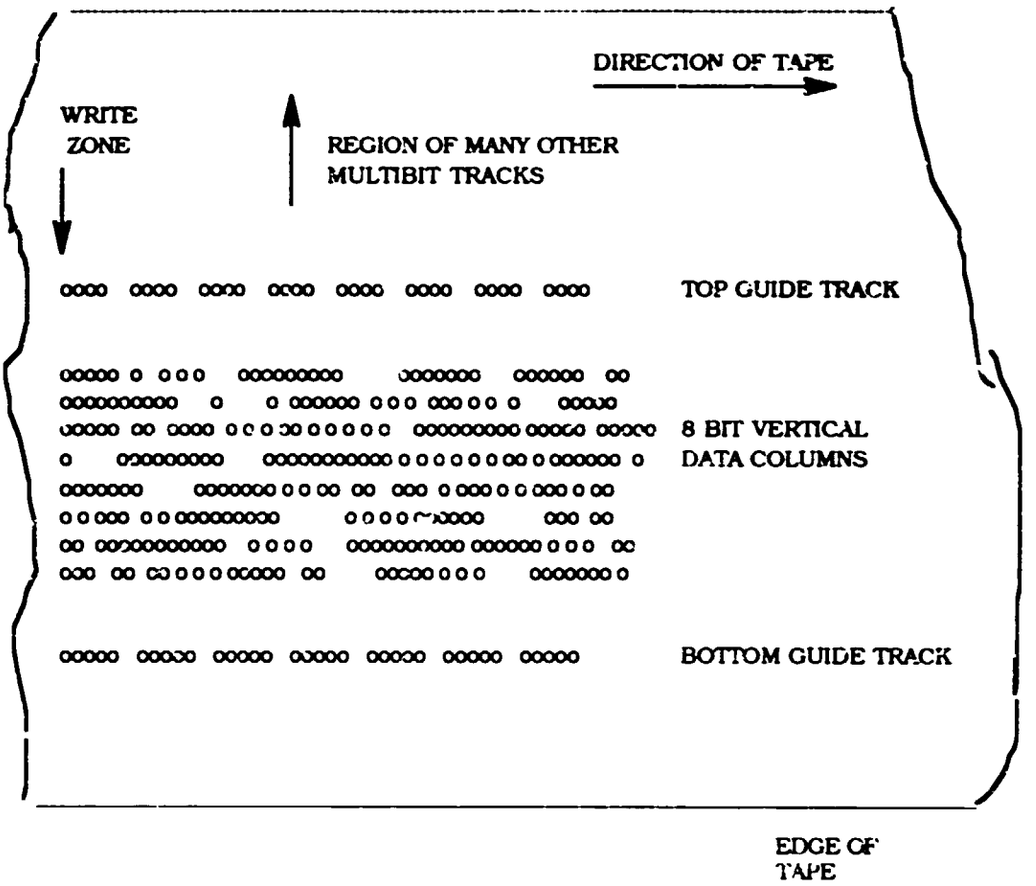


FIGURE 2 LINEAR RECORDING FORMAT
(ONE EIGHT BIT DATA COLUMN SHOWN)

N93-30472

ICI OPTICAL DATA STORAGE TAPE - AN ARCHIVAL MASS STORAGE MEDIA

S23-82

15911 3
P. 9

Andrew J. Ruddick
ICI Imagedata
Brantham
Manningtree Essex CO11 1NL U. K.

1. Introduction

At the 1991 Conference on Mass Storage Systems And Technologies ICI Imagedata presented a paper which introduced ICI Optical Data Storage Tape. This paper placed specific emphasis on the media characteristics and initial data was presented which illustrated the archival stability of the media.

This paper covers more exhaustive analysis that has been carried out on the chemical stability of the media. Equally important, it also addresses archive management issues associated with, for example, the benefits of reduced rewind requirements to accommodate tape lubrication effects that result from careful tribology control in ICI Optical Tape media.

ICI Optical Tape media has been designed to meet the most demanding requirements of archival mass storage. It is envisaged that the volumetric data capacity,long term stability and low maintenance characteristics demonstrated in this paper will have major benefits in increasing reliability and reducing the costs associated with archival storage of large data volumes.

2. Summary Of ICI Optical Tape Media Characteristics

The general characteristics of ICI Optical Tape media have been discussed in many other conferences and presentations (eg. reference 1) and will not be presented in detail here. The features are summarized in the table below and inspection of these indicates the suitability of optical tape for mass storage. The remainder of this paper focuses specifically on the media characteristics that relate to the reliability of the media for archival applications.

Table 1. Cost/ Performance Features of ICI Optical

Low on-line cost	10¢/MB - 40¢/MB depending on format
Low media cost	0.5¢/MB -1¢/MB falling with time
Rapid access	2GB - 20GB per sec. depending on format
High data rate	>3MB/sec
Volumetric efficiency	Factor 10 higher than advanced helical magnetic
Indelible media	
Unlimited read	>40,000 rewind cycles
Long media life	> 30 years

3. Archive Life - Analysis Of Media Stability

3.1 Extended Battelle Testing

Previous published data on ICI Optical Tape has discussed results from accelerated ageing performed at the Battelle Institute in the Battelle Class II test (reference 2). This historical data has clearly demonstrated media lifetimes in excess of 15 years for the product. More recent evaluation has extended the period of testing and lifetimes in excess of 30 years are now predicted. The analysis is discussed in detail below.

3.1.1 Testing Regime

The accelerated ageing test was carried out on full length reels of ICI 1012 Optical Tape packaged in a glass flanged reel.

Prior to the test blank and written areas of the tape were characterized with a map of BER using the Creo 1003 Optical Tape Recorder (OTR). This was done on 3 metre long sections of the tape at three points along the tape length corresponding to inner diameter, mid - diameter, and outer wraps of the wound flanged tape.

The tape was aged in an environment of mixed corrosive gases for a period of 60 days as defined in the Battelle Class II test. Previous work by Battelle Institute has generated a correlation factor which shows this to be equivalent to 30 years in a "typical" office environment.

Following accelerated ageing the data at each section of the tape was then re-read on the OTR and the BER compared with initial maps from the unaged sample. In addition the blank areas were re-mapped, and data was then written in these areas and the BER determined.

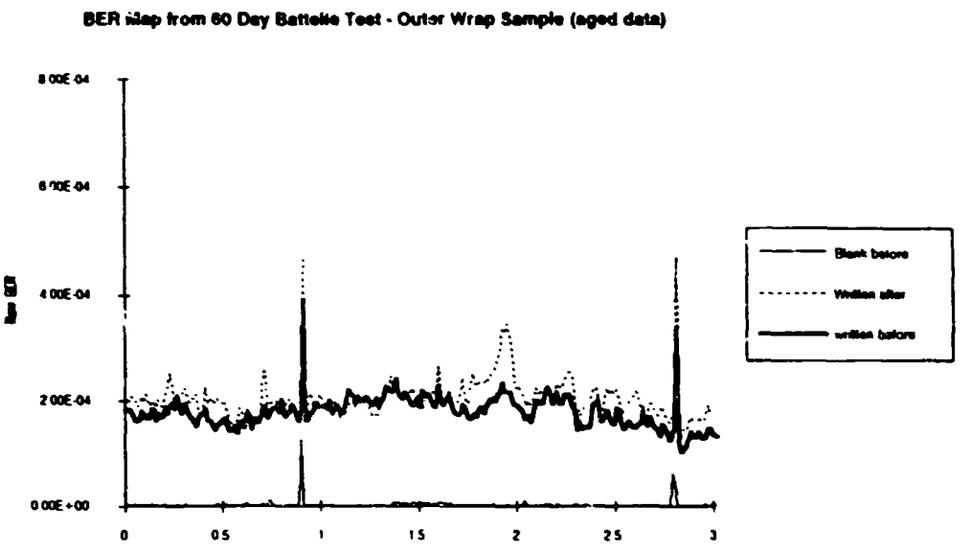
3.1.2 The Results

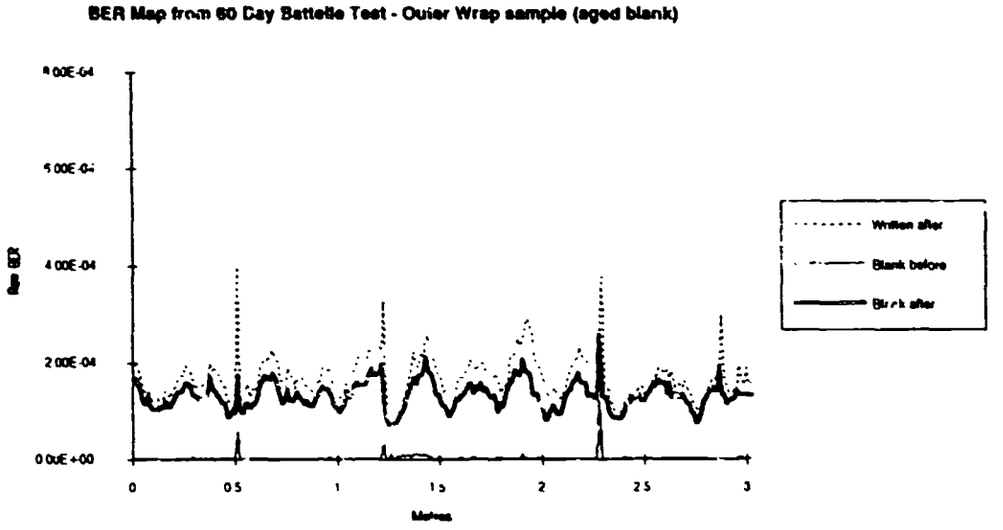
The BER maps obtained are given in figure 1. For simplicity only data from the outer wrap section of the tape is shown. Our previous experience from the Battelle test indicates that this is where degradation is most rapid and this data represents, therefore, the worst case. In these maps the BER measured for each record is plotted against position along the 3 metre sample section.

In summary, inspection of the data shows that the BER of written data does not increase significantly on ageing. A small increase is observed consistently down the length of the sample. The cause of this has not yet been identified and will be further investigated At all points of the sample, however, the data remains fully correctable and well within the limits of ECC (raw BER of 8x 10-4). The blank regions of the tape indicate a detectable increase in BER during ageing. However data subsequently written in these areas is also fully correctable.

In summary, this result indicates that ICI Optical Tape, both blank and with written data has a lifetime well in excess of 30 years

Figure 1. BER Maps From Battelle Test





3.2 Arrhenius Tests

3.2.1 Test Method

Previously reported Arrhenius testing of ICI Optical Tape media has used the measured change in reflectivity and CNR to obtain a lifetime prediction in excess of 300 years at 20°C (68°F) and 60% RH (reference 3). More detailed testing reported here has used a measured degradation in BER as the definition of failure. This is believed to be a more sensitive test of media deterioration.

One difficulty in carrying out accelerated ageing on tape samples is that exposure to elevated temperature and humidity required for rapid test results can warp, and ultimately embrittle, the polyester base film to an extent that the media can no longer be wound onto the optical tape recorder for read/write testing. To overcome this experimental problem, by inspection we have found that the major cause of failure in our media after rapid ageing is the corrosive growth of pinholes in the alloy reflector layer. Consequently, a microscope image analysis technique was developed in order to quantify the growth of defects in the reflector. This technique was not affected by mechanical degradation of the base On unaged samples a correlation was then developed between the pinhole count (quantified by area fraction of the inspected sample) and corrected BER as measured on the Creo 1003 Optical Tape recorder. From this correlation a pinhole count "failure point" could be defined. This was equal to an area fraction of 1×10^{-4}. This point was then used to define the failure of aged samples inspected via image analysis.

The accelerated ageing was carried out by exposing short strips of tape media to a range of temperatures (95°C (203°F), 90°C(194°F), 80°C(176°F)) at a fixed RH of 70% The tape samples contained 2GB of written data split equally at either end of the sample in order to assess any difference between ageing of written and unwritten areas. For each temperature condition the sample was removed from the chambers and inspected by microscopic image analysis every 3 days. The pinhole count at ten points on the strip was taken including areas of written data. This was done avoiding areas of the sample obviously affected by handling damage. The average of these ten readings was used as measure of the sample degradation.

3.2.2 The Results

Typical data obtained is given in figure 2. This is data from exposure at 90°C (194°F), 70%RH. It can be seen that failure occurs catastrophically after exposure for an extended period allowing ready definition of the time at which the failure point was exceeded - in this case 42 days. Media tested at other temperatures gave plots of a similar characteristic (for simplicity data plots not shown).

From these results the failure data is plotted in figure 3 in accordance with analysis via Arrhenius kinetics.

Inspection of this graph allows estimation of an activation energy associated with the failure mechanism. This is calculated as 1.36 eV.

Although the data is not presented explicitly in this report it was noted that comparison of pinhole growth between written and unwritten areas of the sample revealed no significant difference. This is entirely consistent with the results from Battelle testing which also indicate written and unwritten media ages at same rate and via the same mechanisms.

Figure 2. Results From Arrhenius Tests

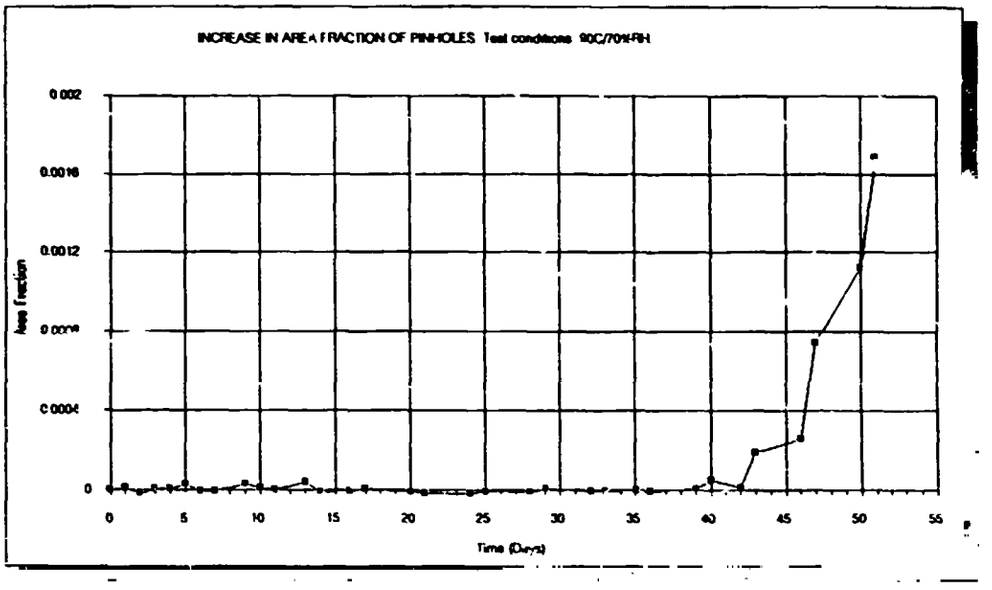
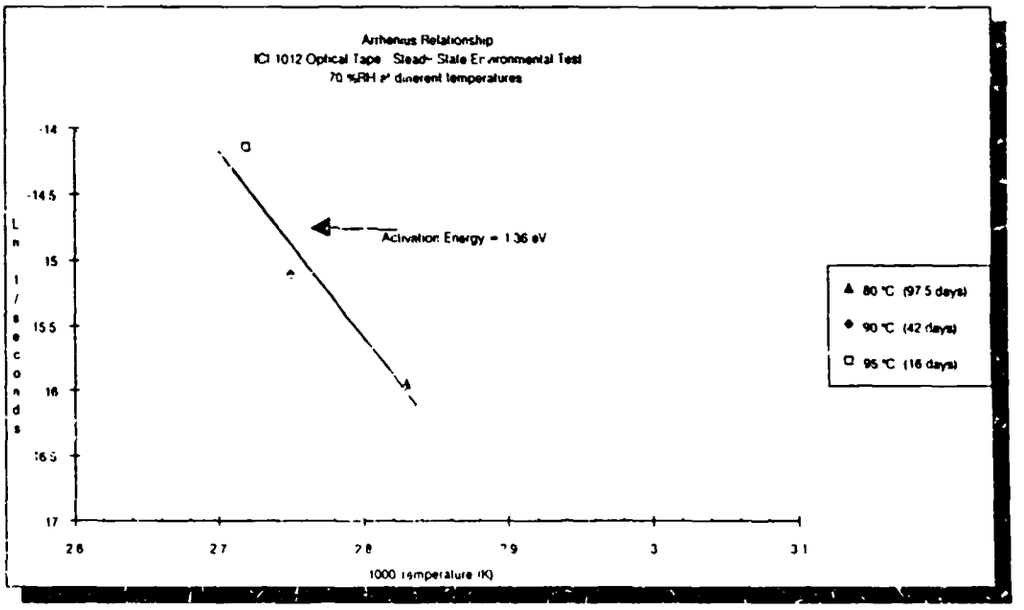


Figure 3. Presentation Of Arrhenius Data



The activation energy of 1.36 eV compares very favourably with 1.5 eV quoted by Sony for their rigid optical disc "Century Media". Based on this activation energy media lifetimes of 500 years can be predicted for tapes stored under controlled conditions of 18°C (65°F)and 70%RH. Taking into account the large errors associated with calculation of activation energies and extrapolation of such predictions it is, perhaps, more reasonable to conclude that **the data fully supports lifetime claims in excess of 100 years.**

4. ICI Optical Tape - Predicted Rewind Requirements

The data in section 3 addresses issues of chemical stability in the media structure. From this we can conclude that the media is intrinsically very robust in both written and unwritten forms and we can predict extremely long lifetime for the ICI Optical Tape product.

Equally important to the lifetime of the data, however, are tribological effects which, if not properly controlled, can cause damage to tape media in archive due to pack distortion, and in the worst case creasing and cinching of the tape reel destroying its functionality and the data stored within it. This is a very well known phenomena to archivists of magnetic media and can result in high costs associated with good archive management.

This section of the paper addresses analysis and modelling work carried out within ICI Imagedata which builds on the magnetic media experience and illustrates how the tribology of our media has been designed to overcome these detrimental effects.

4.1 Background

The purpose of the programme is to assess the length of time over which tapes may be safely stored prior to suffering distortion caused by tension relaxation. It is well known that over time tension relaxes due to the phenomenon of creep. As the tension relaxes, so the interlayer pressure decreases which in turn impairs the ability of a tape reel to withstand the stresses associated with normal drive operation. In practice this would be evident as longitudinal interlayer slippage during acceleration or deceleration.

Other failure mechanisms may also occur, particularly in regions where the cumulative effects of interlayer pressure set up tangential compressive stresses within the tape. In such regions layers of tape may actually separate to form voids (cinching). All forms of loss of pack integrity are to be avoided as the end result is likely to be localized degradation at best and complete loss of data at worst.

4.2 Test Method

Due to the long term nature of the effects described above it is necessary to use predictive techniques to assess and develop media characteristics. Once the product is defined it is then possible to commit to a lengthy assessment for true archival performance.

The predictive technique which we have adopted is based on accelerating the rate of creep in order to cause failure. An independent measure of true creep rate can then be used to estimate the time which would have been taken under normal storage conditions. This is the approach developed by Eschel and Bertram (reference 4) in their study of the archival properties of magnetic media. The results of this work are widely recognized as providing the best guidelines for maintaining magnetic tapes in long term storage.

The procedure is as follows:-

i) A long term measure of the true rate of creep of the media must be made. This requires a high resolution measurement of tape extension (a few microns per week) which must be isolated from variations in temperature and humidity for the duration of the test.

ii) The prediction of lifetime to failure requires not only creep data, but also a measure of the relaxation in interlayer pressure from initial winding to the point of failure. This is best estimated by accelerating the rate of creep and determining the change in pressure corresponding to the point at which the reel is no longer able to withstand normal handling on the drive. In practice this is assessed by braking a spinning reel at aggressive decelerations (580 rads/s≤ compared with 20rads/s≤ on the drive), any indication of interlayer slip (illustrated by a chalk line along the radius) is taken as the failure point.

The method of measuring interlayer pressure is also that discussed by Eschel and Bertram. Thin (25µm, 0.001") stainless steel tabs are inserted into the reel during initial winding. The force required to pull the tabs from the wound reel, together with the measured friction between tape and tab, provide an estimate of the interlayer pressure.

iii) The measures of percentage relaxation to failure and true creep rate are used to predict the lifetime of a tape in storage. The estimation relies on an extrapolated fit for the creep extension of the substrate. Based on creep data generated by Bogy et al (reference 5) the total strain can be shown to behave with time as $\epsilon \sim t^{0.0722}$. Eschel and Bertram use this empirical approximation to derive an expression for the predicted time to failure, Equation 1:

$$t_{failure} = t_1 \left[\frac{\frac{1}{E} + \frac{\%P}{E}}{\frac{1}{E} + \frac{\Delta\epsilon}{\sigma}} \right]^{\frac{1}{0.0722}}$$

Where:

	$t_{failure}$	is the estimated time to failure.
	E	is Young's Modulus (3531 N/mm^2, 5*10^5 psi).
	%P	is the percentage pressure relaxation at failure and
	$\Delta\epsilon/\sigma$	is creep strain per unit stress at time t_1.

4.2 Results

4.2.1 Creep Rate

The rate of creep of fourteen samples of Optical Tape media has been measured over a period of 208 days. The tapes have extended at an average creep strain per unit load of 5.0*10^{-7} /psi.

These figures can be compared with an average of 2.4 *10^{-7}/psi and a maximum of 6.5*10^{-7}/psi for various polyester substrates measured under similar conditions in Reference 5.

4.2.2 Pressure Relaxation

Accelerated creep tests have been carried out by maintaining the reels at an elevated temperature of 45°C (113°F)/50%RH. Tapes are periodically removed, conditioned to ambient and tested for slippage during deceleration at 580rads/s^2

For the full reel, failure occurred after the pressure had relaxed by 80%-85%. This level of reduction is much greater than the 50% level predicted in the literature for magnetic media. The interlayer pressure at failure is remarkably low given the large inertias experienced by the

full length reels. **It is clear that the tribology developments of ICI Optical Tape media which were initially directed to give give excellent tape handling and wear, and to allow easy transportation will also give good pack integrity at low interlayer pressures that can develop in archive.**

4.2.3 Predicted Rewind Interval

Extrapolations for a full length reel, based on equation 1, are shown in Table 2 below. The creep figures are based on average creep plus two standard deviations. The result for the tapes studied in Ref 4 are shown for comparative purposes. Estimates are given for percentage relaxations of 50% and 80%. **Clearly, being able to maintain reel integrity at low interlayer pressures has a dramatic effect on extending the rewind interval.**

The predicted rewind interval of 39 yrs should not be taken literally but rather as an indication of the relative advantage of pack integrity at low interlayer pressures.

Table 2. Estimated Rewind Interval

	Creep Strain per Unit Stress (/psi)	Young's Modulus (psi)	$t_{failure}$ (yrs)
Full Reel 50%	$6.5 * 10^{-7}$	$5 * 10^{5}$	3.1
Full Reel 80%	$6.5 * 10^{-7}$	$5 * 10^{5}$	39
Ampex 50% (Ref 4)	$5 * 10^{-7}$	$5 * 10^{5}$	3.5

Given the many assumptions and aggressive levels of deceleration which have been used to derive the storage lifetime we believe that further work is required to establish reliably an upper limit. Based on the results generated so far it is nevertheless possible to say that a period of 5 years represents a safe, conservative interval for tapes stored in a well maintained archive. It is fully expected that further analysis will extend this prediction to 10 years or more. Since creep is very sensitive to temperature, there will be severe implications where storage is under elevated temperature conditions.

4.3 Rewind Period in Archive - Conclusions

The conclusion from this work is that rewind intervals of 5 years can be safely assumed for ICI Optical Tape full length reels where storage conditions are maintained at 18°C (65°F), 50%RH. This is currently believed to be a very conservative analysis. However further work is required for more accurate predictions.

This length of time, together with the ease and speed of retensioning, represents a relatively low level of maintenance. The critical factors in determining this are the rate at which the media creeps and the robustness of the pack at low interlayer pressures which is due to carefully controlled surface chemistry between overcoat and backcoat layers in the wound pack specific to the ICI media structure.

5. Summary

Data presented in this paper shows that the chemistry and tribology of ICI Optical Tape media has been carefully designed to create a media ideally suited for the requirements of a low cost, low maintenance, high reliability data archive. In summary, using industry standard tests the following characteristics have been demonstrated:

i. A media structure stable well in excess of 100 years under ideal storage conditions

ii. A media lifetime in excess of 30 years in the presence of corrosive gases, typical of the standard office environment.(testing will continue in order to identify the failure point)

iii. Tape rewind periods that are well in excess of magnetic media requirements, allowing for reduced archive management costs.

Combined with the unsurpassed volumetric capacity and low cost that can be achieved with optical tape, we believe these archival performance characteristics make it an ideal medium for many mass storage applications.

6. References

1. J. F.Duffy, OPTICAL DATA STORAGE TAPE: A NEW HIGH DENSITY DATA STORAGE AND ARCHIVE MEDIUM, Presented at THIC, 1/2 October 1991.

2. W. H. Abbott, THE DEVELOPMENT AND PERFORMANCE CHARACTERISTICS OF MIXED FLOWING GAS TEST ENVIRONMENT, IEEE Trans on Components, Hybrids and Manufacturing Technology, 11 pp22-35 (1988)

3. R. A. Maclean, J. F. Duffy, ICI Optical Data Storage Tape, NASA Conf. On Mass Storage Systems and Technologies, 1991

4. N. Bertram and A. Eshel, RECORDING MEDIA ARCHIVAL ATTRIBUTES, Final Report to RADC, TR-80-123, April,1980.

5. D. B. Bogy, N. Bugdyaci and F. E. Talke, EXPERIMENTAL DETERMINATION OF CREEP FUNCTION FOR THIN ORTHOTROPIC POLYMER FILMS, IBM J. Res. Develop., 23, pp. 450-458 (1979).

N93-30473

Flexible Storage Medium For Write-Once Optical Tape

Andrew J. G. Strandjord, Steven P. Webb, Donald J. Perettie, and Robert A. Cipriano

The Dow Chemical Company
Central Research
1702 Building
Midland, Michigan 48674

Abstract

A write-once data storage media has been developed which is suitable for optical tape applications. The media is manufactured using a continuous film process to deposit a ternary alloy of tin, bismuth, and copper. This laser sensitive layer is sputter deposited onto commercial plastic web as a single-layer thin film. A second layer is sequentially deposited on top of the alloy to enhance the media performance and act as an abrasion resistant hard overcoat.

The media was observed to have laser write sensitivities of less than 2.0 njoules/bit, carrier-to-noise levels of greater than 50dB's, modulation depths of ~100%, read-margins of greater than 35, uniform grain sizes of less than 200 Ångstroms, and a media lifetime that exceeds 10 years.

Prototype tape media was produced for use in the CREO drive system. The active and overcoat materials are first sputter deposited onto three mil PET film in a single pass through the vacuum coating system, and then converted down into multiple reels of 35mm x 880m tape. One mil PET film was also coated in this manner and then slit and packaged into 3480 tape cartridges.

1. Introduction

Optical data storage is quickly becoming a viable and often preferred option to magnetic storage. The promise of high data densities and archival stability has initiated the development of thin-film optical medias which can be substituted into applications where magnetic technology is inadequate. For example, optical tape offers storage densities of over one terabyte on a single 12 inch diameter reel of 3mil film (35mm x 880m). This is equivalent to ~5,000 reels of standard magnetic tape[1]. CREO Electronics Corporation[2] has developed a commercial drive for optical tape and is currently using the write-once media developed by ICI (Digital Paper)[3-4] for developmental evaluation. LaserTape Systems, and others, are in the process of developing new drive technology for the use of optical tape packaged in 3480-type cartridges (1/2" x 570ft). This would offer multi-gigabyte storage capacities to the broader consumer market.

Years of developmental work at The Dow Chemical Company has produced a flexible optical media which can be used in tape applications. The media has been shown to have many superior properties with respect to the other medias being developed for optical tape. This paper reports on some of the results which have been collected recently.

2. Optical Tape Requirements

Tape is considered a non-rigid media and must meet several special handling requirements which are not major issues for rigid media, such as discs and cards. Tape media must be sufficiently flexible to accommodate motion around the small hubs and rollers associated with tape handling without degradation of the layered structure[2]. Polyester films which range in thickness from about 1/2 to 3 mil's have been shown to have the necessary physical and optical properties required by current drive technologies. Active coatings which are sputter deposited onto these substrates must also be

sufficiently flexible to withstand the mechanical handling associated with winding and unwinding of the tape. Coatings which are either thick or rigid are susceptible to cracking and delaminating, and are therefore not suitable for the production of optical tape.

The cleanliness and smoothness of the substrate materials are critical issues which can affect performance of the media. Misinterpretation of debris and surface non-uniformities can lead to error rates which exceed the level which can be handled by the correction code of the drive. Bit-error-rates (BER) of up to $\sim 10^{-4}$ can currently be accommodated by the CREO drive, but lower levels are preferred. These concerns can be addressed by pre-cleaning the substrate and/or subbing the base PET with an organic layer in a controlled environment.

Another critical requirement, which is relevant for all storage medias, is that the data must remain environmentally stable for long periods of time. This is especially true for those media which are being targeted for archival storage applications. For example, the optical tape standard set by CREO requires a 15 year media lifetime in an office environment (25°C/50%RH)[1].

Premature aging of many thin film media has been shown to occur by several different mechanisms: 1) the active layer can degrade by such processes as oxidation or phase segregation, to make the media less sensitive to laser writing and/or increase the bit-error-rate; 2, the written data spots can change with time to cause edge deformation or phase reversal, thus reducing the playback signals; or 3) the media could mechanically degrade due to wear and abrasion during media handling. The first two forms of environmental degradation are strongly dependent on the composition and structure of the active layer within the media, though some stabilization can sometimes be achieved by overcoating this layer with a topcoat. Protection from frictional wear, due to film contact with itself and the roller mechanisms, is best afforded by a hard thin-film overcoat. However, most write-once media that have a hard overcoat in direct contact with the active layer are insensitive toward laser writing.[5-7]

The current laser write sensitivity requirements for optical tape are less than 2 nanojoules per bit for 35mm tape (CREO) and less than 1 njoule per bit for the 3480 tape format (Laser Tape).

3. Dow Optical Media

The new write-once optical data storage media developed at The Dow Chemical Company exceeds these requirements for tape applications. The recording layer[8-14] is a ternary metal alloy system containing tin, bismuth and copper in a weight percent ratio of 70/25/5. This layer is preferably deposited by plasma sputtering from a cast target.

The morphology of the sputtering target is a complex distribution of metallic and inter-metallic phases which are present in varying amounts. Figure 1 is a back-scattering electron image of the machined sputtering target. The predominant phase is a tin-rich matrix with 4-5 wt% incorporated bismuth. The bright phase seen in the figure consists of large areas of segregated bismuth (<1%Sn) which predominates at the grain boundaries of the Sn-rich phase. The needle-like structures are the copper containing phases. They consist of a core of Cu_3Sn (ε-CuSn) surrounded by Cu_6Sn_5 (η-CuSn)[15].

The relative proportions and distribution of these different phases are a function of the temperature history of the sputtering target during casting and machining. Fast cooling of the target results in a fine phase structure, while slow cooling allows the phases to grow into larger crystallites.

The heating behavior of the SnBiCu alloy is depicted in the calorimetric trace (DSC) shown in Figure 2. The relatively low temperature endotherm at 143°C is near the SnBi eutectic temperature[15]. The alloy becomes pasty in consistency at this point and can be characterized as a soft, mobile state which contains many of the properties of both a solid and a liquid.

The low temperature mobility associated with the SnBiCu alloy (especially with regard to the bismuth phase) implies that small amounts of heat at the surface of the alloy during deposition can have a pronounced influence on

the composition of the final thin film media. If the target temperature approaches 150°C, the phases have an opportunity to both segregate and migrate along the thermal gradients within the alloy. Compositional analysis of a sputtering target after a series of extended high power depositions, in which the temperature of the target surface was well above 143°C, showed evidence of this type of elemental migration. The Cu containing phases tended to migrate from the surface into the bulk, while the Sn concentration was found to increase at the surface. Films made during coating runs with this target were found to be rich in bismuth, relative to the expected initial target composition. By routine process monitoring and proper bonding of the target to a cooled backing plate, temperature-dependent deposition can be uniformly controlled.

4. Manufacturing

The medium is manufactured using a continuous film process where the alloy is sputter deposited directly onto thin polymeric web. A simplified schematic of the web coating system is shown in Figure 3. This coater is configured to sputter coat up to three layers at a time using three separate DC magnetron cathodes. Also located in the system is a pre-glow station for ionized gas cleaning of the substrate before coating. Each of these four stations (mini-chambers) is isolated from each other in space, thereby producing a local environment for the containment of the plasma gasses. This allows separate processes to be carried out simultaneously at each station without cross contamination between the four sputtering sources, or alternatively allows for the incorporation of multiple targets of one material to increase the production rate.

All of the critical process parameters are continually monitored during the coating process to ensure consistent and uniform coatings. Down web reflectance and transmission spectra are collected as the primary quality control feedback loop of the system. Film properties are easily maintained to within 1% of their targeted value for the duration of the coating run.

The equipment described above was used to refine the coating parameters and to produce samples for market testing before moving the process over to a much larger, production machine.

5. Laser-Write Sensitivity

The laser-written data bits, in this write-once system, are in the form of non-reflective spots on a reflective background. The mechanism for spot creation was found to be different than a purely ablative and/or evaporative process. In these films, the alloy melts and flows out of the laser irradiated area, thereby opening up a dark pit or hole[16-17]. The medium is sensitive towards laser writing over a broad wavelength range and will therefore be compatible with lower wavelength lasers as that technology develops (see Figure 4-5).

Laser write sensitivity measurements were made using several different techniques. Figure 6 shows the relative laser-write sensitivities of the SnBiCu films as a function of reflectivity. These measurements were made using a static test apparatus in which a 10 milliwatt diode laser is focussed onto the surface of the media (see Figure 7). To determine the laser-writing sensitivity at constant power, the pulse width is decreased until the data spot can no longer be visualized. Digital spots were produced on the Dow Media with energy levels of less than 2 nanojoules per bit.

Tape drives are presently being developed to use media which have reflectivity levels between 35 and 55%. The laser-write sensitivities shown in Figure 6 are observed to be very similar for all of the media samples made with reflectivity levels between 30 and 60%. This invariance makes the SnBiCu media very versatile towards fulfilling new media requirements which may occur as drive technology evolves and can also allow this media to be used in totally new, film-based media formats, i.e. optical floppies, laminated cards, and discs[1].

Dynamic sensitivity measurements were made on an APEX Optical Media test system[18]. Under normal operating conditions, the APEX system automatically focuses and tracks on compact disc type media. The 3mil films produced for optical tape were converted into a form which was compatible with the APEX test system. Specifically, the film was cut into discs which were ~4 1/2" OD x 1 1/2" ID and

then sandwiched between two mirror-flat polycarbonate discs (~50 mil thick) to create a sample in which the optical path was similar to a compact disc. Additionally, this 3mil film media does not contain tracking information. Therefore, the auto-tracking feature of the equipment is unusable and continuous reading of data relies on the inherent stability of the spin stand to keep the laser head over the data during subsequent revolutions of the disc/film after writing. Table vibrations, and the like, eventually cause the data to drift out of the field of view of the read laser beam. The APEX system will, however, consistently focus on the media, as long as the media have been carefully sandwiched to form a "flat", wobble free surface.

The Dow Optical Media has been successfully evaluated using the APEX test system. Typical performance evaluations have shown modulation depths of ~100% and carrier-to-noise levels of greater than 50dB's at 10milliwatt/250nsec laser settings, 30 ft/sec media speeds, and 30 kHz spectrum analyzer bandwidths (see Figures 8-9).

Threshold values were obtained for the media by changing the laser energy and measuring both the modulation depth and carrier-to-noise level at each new setting (see Figures 10 & 11). Both curves show a drop off in the performance as the laser write energy is decreased beyond the 2 nanojoule level. The spot shapes are observed to be very uniform in size with clearly defined borders (see Figure 12).

Several reels of the Dow Optical Tape were sent to CREO for further read-write evaluations. They were able to write successfully on the media using laser pulse widths of less than 115nsec. An optical image of the laser written data is shown in Figure 13. The read margin for this media is shown in Figure 14. The width of the curve at the bottom indicates what range of threshold values (focus, laser power, etc.) at which the data can be read with low error rates. Margin widths of 35 and greater are judged as superior by CREO. A value of between 36 and 39 is estimated for the Dow Media.

6. Environmental Stability

Media stability was evaluated by subjecting the films to various environments of high temperature and high humidity. The goal of these accelerated aging experiments is to extract a lifetime for the media at room temperature and room humidity. Lifetime is defined as the time period in which the media remains usable relative to a set of media standards. For the SnBiCu films, it was assumed that the media are acceptable as long as they remain within 10% of their original reflectivity specifications. Therefore, determining the time it takes a sample to degrade to 90% of its original value, as a function of water and temperature, will allow for the calculation of the lifetime at room temperature and humidity (20°C and 50% RH). Experiments run under isobaric (constant water) aging conditions have shown that no significant temperature effect can be discerned between 20°C and 100°C (see Figure 15). This simplifies the calculation of the lifetime to include only the effect of water.

The data plotted in Figure 16 show the smooth relationship between the environmental water concentration and the lifetime of the media. The open square represents the extrapolated lifetime at ambient conditions. Lifetimes of greater than 10 years have been predicted for the media.

7. Prototype Development

In addition to tape, other types of media have been envisioned for the metallized web. Optical cards and discs could be produced easily by using an embossed film as the substrate (containing pre-formatting and tracking information), coating this web with the active layer, and then laminating to a base substrate. Schematic representations of these prototypes are shown in Figure 17.

8. Conclusions

A write-once data storage medium has been developed which is suitable for optical tape applications. Typical performance values for the medium are as follows:

a) reflectivity levels between 35 and 55%,

b) laser write sensitivities of less than 2 nanojoules/bit,

c) modulation depths of ~100% @ 250nsec and 10 mwatt laser settings.

d) carrier-to-noise levels which are >50dB's @ 30kHz bandwidth, 1 MHz/250nsec/ 10mwatts laser settings, and media translation speeds of 30fps,

e) media lifetimes >10years, and

f) read margins of >35.

Work is continuing in an effort to refine the manufacturing process parameters and qualify for the CREO drive system.

9. References

1. D. Fountain, "Digital Paper", Byte, McGraw-Hill Inc., NY, February 1989.
2. D. Gelbart, "An Optical Tape Recorder Using Linear Scanning", Conference Digest Topical Meeting on Optical Data Storage, Vancouver, Canada, pp.34-37, March 5-7, 1990.
3. A. Ruddick and J. Duffy, "ICI's Optical Tape Offers Flexible Alternative to Rigid Optical Media", Optical Memory News, Rothchild Consultants, pp. 10-13, February 1991.
4. P. Vogelgesang and J. Hartmann, "Erasable Optical Tape Feasibility Study", Proc. SPIE Optical Data Storage Technology and Applications, Los Angeles, vol. 899, pp.172-177, 1988.
5. M. Terao, S. Horigome, K. Shigematsu, Y. Miyauchi, and M. Nakazawa, "Resistance to Oxidation of Te-Se Optical Recording Films", Proc. SPIE Optical Data Storage, Incline Village, vol. 382, pp.276-81, 1983.
6. S. Chao, Y. Huang, Y. Chen, and L.Yan, "Materials for Multiple Stages of Archival Optical Recording", Proc. SPIE Optical Storage Technology and Applications, Los Angeles, vol. 899, pp.240-43, 1988.
7. A. Gotoh, S. Nakamichi, and S. Horigome, SPIE 1989 Technical Digest Series on Optical Data Storage, pp.24-27, 1989.
8. V. Kurfman and R. Gransden, US Patent 4,115,619 (1978).
9. V. Kurfman and R. Gransden, US Patent 4,211,822 (1980).
10. H. Marton and V. Kurfman, US Patent 4,241,129 (1980).
11. V. Kurfman, US Patent 4,501,208 (1985).
12. V. Kurfman, US Patent 4,998,239 (1986).
13. A.J.G. Strandjord, R.L. Yates, and D.J. Perettie, US Patent 4,998,239 (1991).
14. A.J.G. Strandjord, D.J. Perettie, and R.L. Yates, US Patent 5,016,240 (1991).
15. M. Hansen, "Constitution of Binary Alloys", McGraw-Hill, N.Y., 1958.
16. H. Haskal, "Dynamics of Pit Formation in Ablative Optical Recording", Proc. SPIE Optical Data Storage, Incline Village, vol. 382, pp.174-181, 1983.
17. A.J.G. Strandjord, S.P.Webb, D.R. Beaman, and S.L.B. Carroll, "Thin Film Coatings for Flexible Optical Data Storage", Proc. SPIE Optical Thin Films III: New Developments, San Diego, CA, pp. 127-131, vol. 1323, July 9-11, 1990.
18. OHMT-300 WORM Test System, Apex Systems Inc., Boulder, CO, 80301.

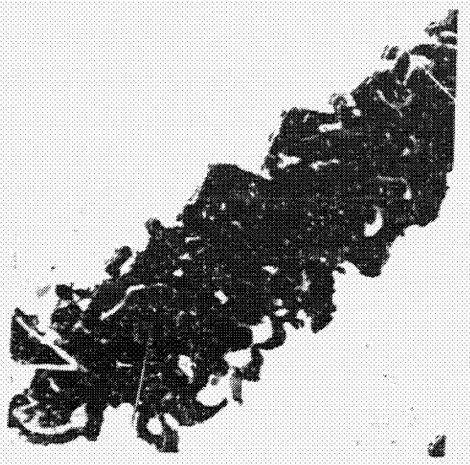


Figure 1. Back-scattered electron image (800x) of SnBiCu target (Cameca Camebax Electron Microprobe). Light Grey = Sn (4-5%Bi), White = Bi (<1%Sn), Dark Grey = Cu_6Sn_5, & Black = Cu_3Sn.

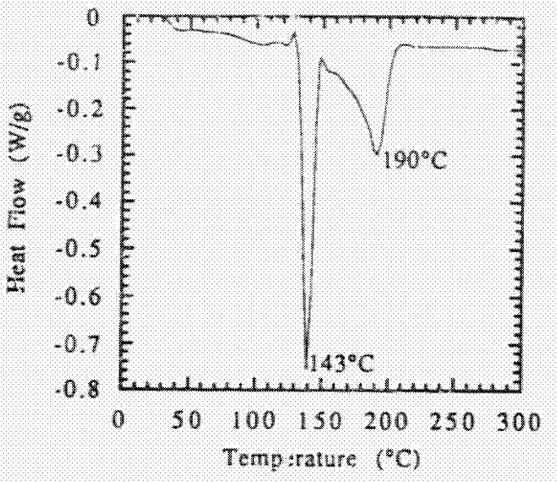


Figure 2. Differential scanning calorimetric (DSC) trace of the SnBi alloy, 10°C/min scan rate, (Dupont 9900).

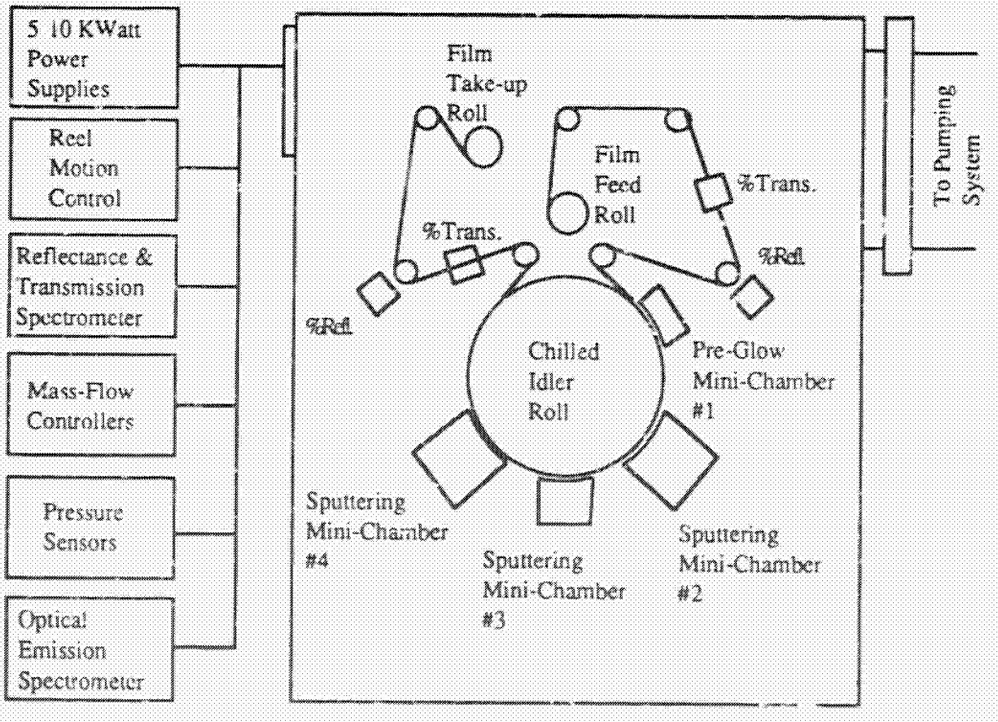


Figure 3. Schematic diagram of the vacuum coating equipment used to sputter deposit the active alloy and protective overcoat.

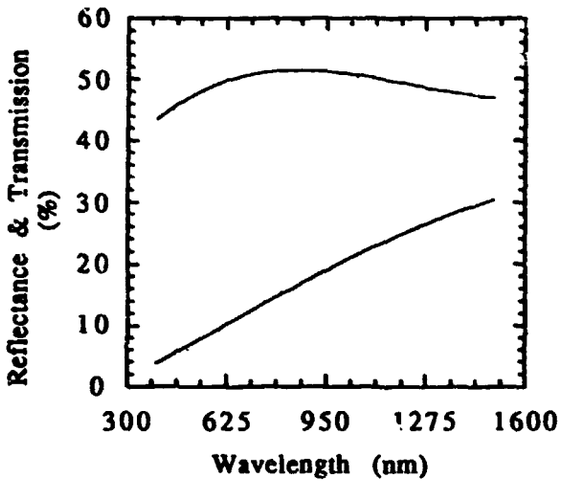


Figure 4. Plot of reflectivity (upper) and transmission (lower) of SnBiCu film.

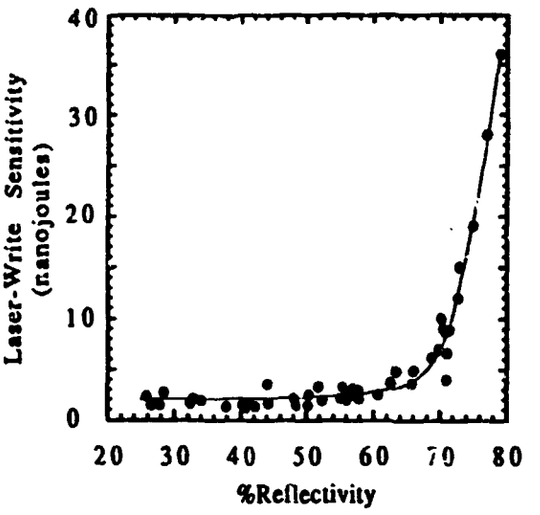


Figure 6. Laser write threshold of SnBiCu films as a function of reflectivity.

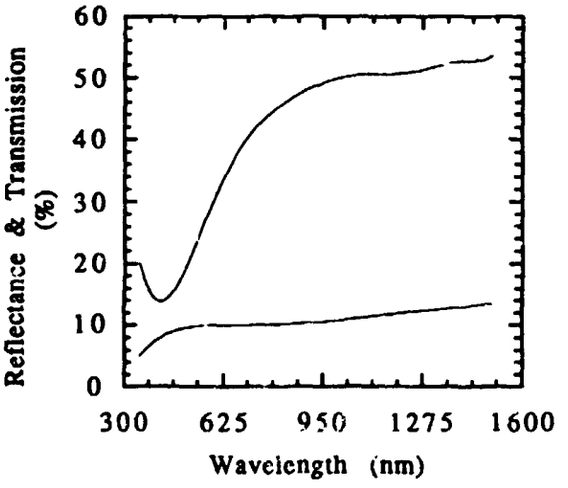


Figure 5. Plot of reflectivity (upper) and transmission (lower) of SnBiCu/overcoat media.

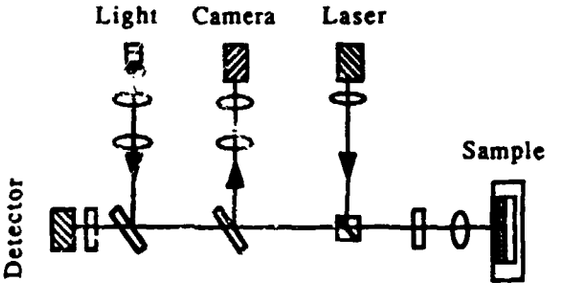


Figure 7. Static laser-write test system.

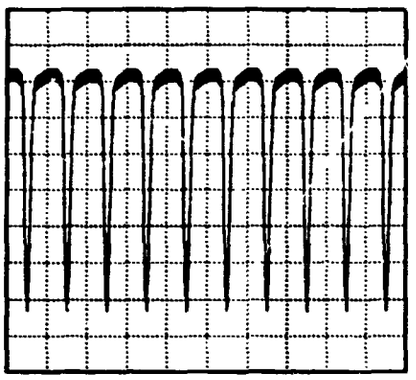


Figure 8. Digitized oscilloscope trace of the playback signal from the APEX test System

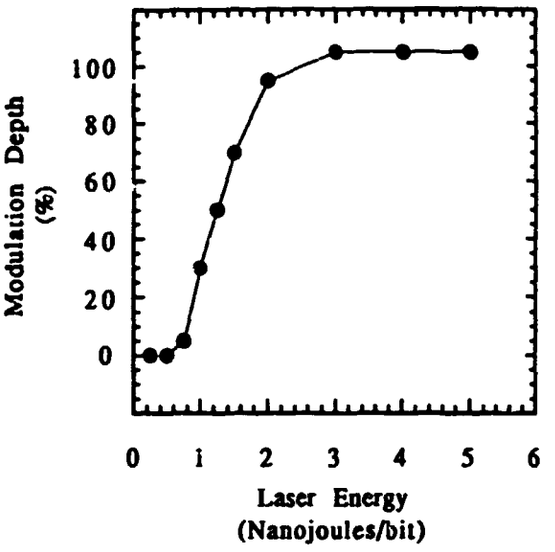


Figure 10. Plot of modulation depth as a function of laser-write energy.

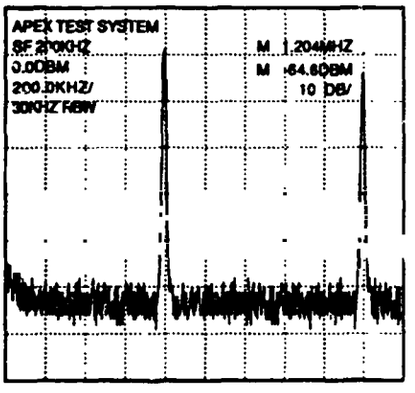


Figure 9. Digitized frequency spectrum from the APEX test system.

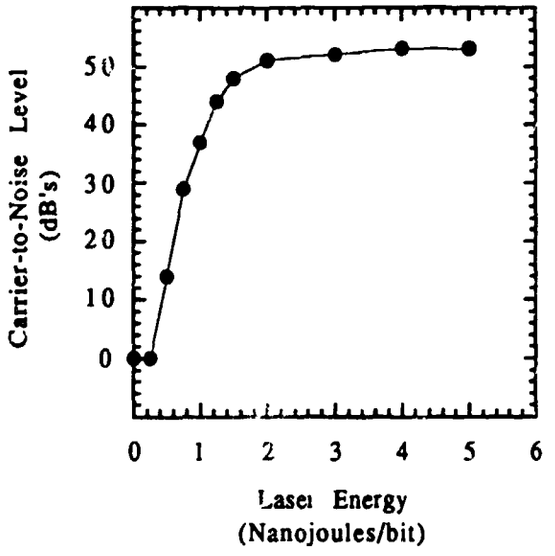


Figure 11. Plot of carrier-to-noise level as a function of laser-write energy.

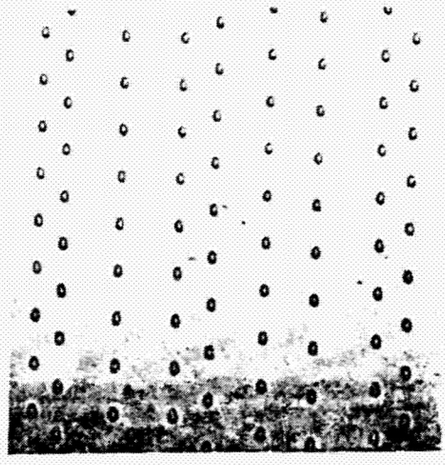


Figure 12. Optical photograph of data written on the SnBiCu media using the APEX system.

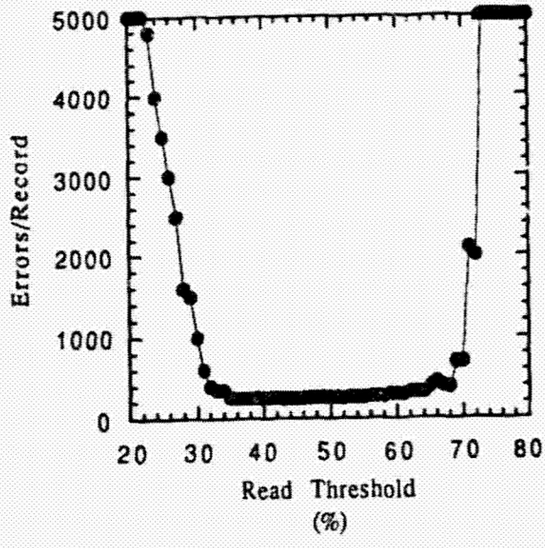


Figure 13. Read margin data from the CREO tape drive system.

Figure 14. Optical photograph of data written by the CREO tape drive system.

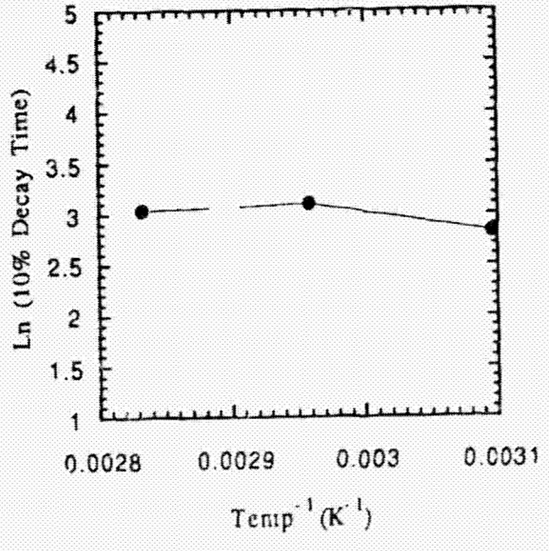


Figure 15. Environmental stability of the SnBiCu Media. Arrhenius plot at constant water level (78.6 Torr).

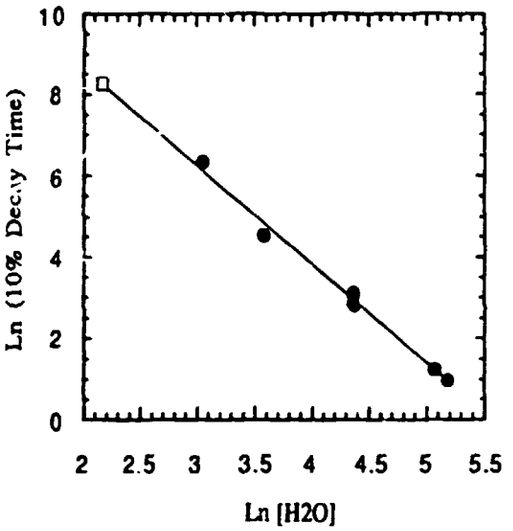


Figure 16. Environmental stability of the SnBiCu media. Humidity dependence.

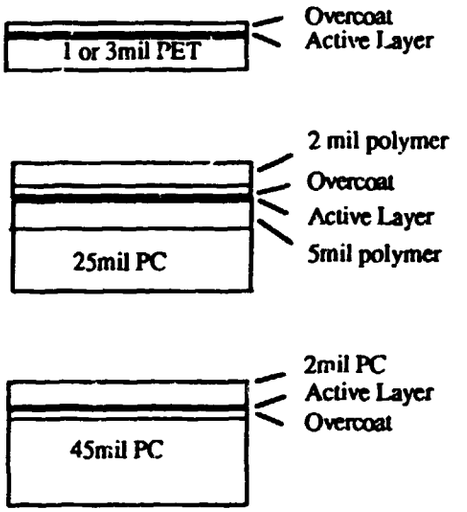


Figure 17. Schematic representations of new media formats.

N93-30474

ELECTRON TRAPPING DATA STORAGE SYSTEM AND APPLICATIONS

Daniel Brower, Allen Earman and M. H. Chaffin
Optex Corporation, 2 Research Court
Rockville, MD 20850

S25-82
15715 ABS ONLY
p. 1

The advent of digital information storage and retrieval has led to explosive growth in data transmission techniques, data compression alternatives, and the need for high capacity random access data storage. Advances in data storage technologies are limiting the utilization of digitally based systems. New storage technologies will be required which can provide higher data capacities and faster transfer rates in a more compact format. Magnetic disk/tape and current optical data storage technologies do not provide these higher performance requirements for all digital data applications.

A new technology developed at the Optex Corporation out-performs all other existing data storage technologies. The Electron Trapping Optical Memory (ETOM) media is capable of storing as much as 14 gigabytes of uncompressed data on a single, double-sided $5\frac{1}{4}$ inch disk with a data transfer rate of up to 12 megabits per second. The disk is removable, compact, lightweight, environmentally stable, and robust. Since the Write/Read/Erase (W/R/E) processes are carried out 100% photonically, no heating of the recording media is required. Therefore, the storage media suffers no deleterious effects from repeated Write/Read/Erase cycling.

ETOM media are novel erasable data storage media which utilize the phenomenon of electron trapping common in a class of luminescent materials known as IR stimulable phosphors. They are composed of an alkaline-earth sulfide host lattice and two rare earth dopants (the luminescent and trapping centers). Data storage is a fully photonic process which involves the interaction of light with the dopant ions and their electrons within the media. Also, due to their exceptionally wide dynamic range, these materials are capable of multilevel or non-binary recording. This coding technique can provide up to four times the data transfer rate using four discrete amplitude levels.

The media uses two laser wavelengths to accomplish the W/R/E processes. The transfer of data is based on a quantum effect which involves exciting a luminescent ion and passing its excited electron to a nearby trapping ion. Once bound to the new ion, the electron falls to the ground state of the ion; this traps the electron. This is the stored state and is a stable configuration for the electron. It will remain trapped until a photon of the read light source excites it from the ground state to the excited state. From here it can migrate back to a luminescent ion and fall to the ground state. The transfer back to the ground state is accompanied by the emission of a photon which is detected by the disk drive and indicates stored data.

Optex Corporation has developed this rewritable data storage technology for use as a basis for numerous data storage products. Industries that can benefit from the ETOM data storage technologies include: telecommunications, entertainment, video imagery, and data/image acquisition and storage. Products developed for these industries are well suited for the demanding store-and-forward buffer systems and archival storage systems needed for these applications. For example, a digital video recording system based on 4x subcarrier sampling of standard NTSC composite color video (i.e., the D-2 standard) requires approximately 1 gigabyte per minute of digitized video frames, and a transfer rate of 120 megabits per second. A 130 mm ETOM disk can store up to 14 minutes with less than 50 ms access time to any frame. If a data compression techniques such as the current MPEG standard is employed, the same ETOM disk can store up to 18 hours of compressed digital video programming.

The "State" of "The State of The Art" In Mass Storage Technology

Dale Lancaster, Convex Computer Corporation
3000 Waterview Parkway, Richardson, TX 75083

Introduction

In the last couple years, there has been an abnormal amount of interest and activity in the automated mass storage application area. At Convex we have been heavily involved in some of these efforts. This paper will describe some of our experiences and also discuss the trends that are occurring in this industry.

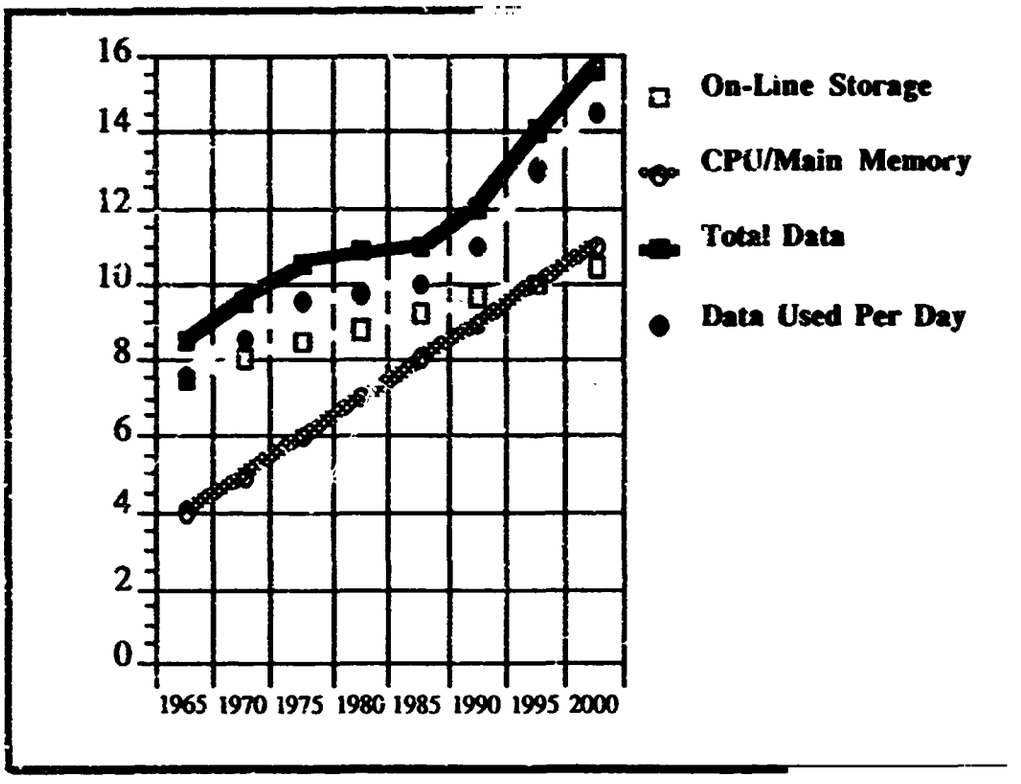
The Tortoise and the Hare; Or Mass Storage and Very Fast Computers

Looking back on the history of computing, it is obvious that mass storage technology has grossly fallen short of keeping up with processor technology. It was observed that as many as 25 years ago the ratio between the amount of affordable on-line disk storage and the amount of physical memory on a computer was about 3000 to 1. In other words, a typical departmental computer would have maybe 16k bytes of processor physical memory and about 50 MBytes of "affordable" disk storage. Now in 1992, that ratio has dropped to about 20 to 1 or less. This is best illustrated graphically using the chart below. The amount of storage in bytes is represented on a log base 2 scale over time in years. These are my own home grown numbers but I believe represent a fairly true picture. The really curious question is what happens when the total amount of physical memory on a computer is more than the amount of "affordable disk" that will be connected to it? It will be possible to buy computers with over 1 Terabyte of memory by the end of the decade for what is considered a reasonable price. However the equivalent amount of disk storage will be quite expensive. Maybe one option is to store all active data in memory and spool inactive data out to very high speed tape directly and avoid the use of disks. I doubt this will happen, but it may one day lead to this architecture.

Certainly disk storage has achieved some impressive densities and speeds, but processor speeds and memory capacities have done much better. Because the computers are running much faster, they are producing much more data than in the past and at a rate that mass storage technology cannot typically handle. Disk storage today costs about $2-5 per MByte and that cost is dropping as higher density drives are produced. The problem is that disk storage will always remain much more expensive than people want because their mass storage needs are growing at a rate that is faster than drive technology can handle. Without new technology, it is possible that the single largest cost for a data center will be for the storage and management of data.

Technology now exists that allows a computer to have on-line access to virtually all the data that exists in the computer center. Until recently only the most prestigious and "richest" computer centers could afford this type of technology. Now even a modest computer center can afford it and as a result everybody is getting on the bandwagon of "on-lining" all their data and automating their data storage and management.

In the past, the solutions to on-lining data were jukeboxes and the ever popular STK Silo using 4480 tapes. Optical jukeboxes have yet to really catch on and I think mainly because the drives are still very slow and the total amount of "affordable" storage is not very high. The 3480/4480 tapes have done quite well in this area, but because of the expensive technology to automate it, it has not been applicable to the smaller data centers. The other problem has been that the Unix operating system has never been able to handle the virtual disk concept (the ability to store more data on a filesystem than what actually is available on the disk).



The software problem with Unix has now been solved in many ways with many vendors. The most popular of these solutions is the UnfTree Central File Manager (UnfTree) from DISCOS. Convex has also produced its own virtual disk product called the Convex Storage Manager (CSM). These two products are actually complementary in that CSM handles native ConvexOS filesystem full conditions and UnfTree handles network based client access to the large archives that it manages. We have plans at Convex to merge these two products.

The Realities of Tape Technology

The hardware problem has also been solved with the emergence of helical scan tape technology and high speed interfaces to these tapes. The two most recent additions are VHS and D2. Metrum Inc has taken VHS video tape technology and adapted it for digital storage and Ampex and E-systems have done likewise for the D2 tape technology. With helical scan, data is stored by writing tracks of data at an angle across the tape rather than as longitudinal tracks of data such as you find on 9-track and 3480. By doing this, you can archive much higher tape density and throughput. A single T-120 cassette can store 14 GBytes of data with access rates of 2-3 Mbytes/second and the D2 can store 25 GB on a small cassette and 165 GBytes on a large cassette with access rates of 15 MBytes/sec. With these new tape technologies it is possible for most data centers to cost effectively store all its data in a small 20 sq foot tape robot. As well, with UnfTree and CSM, all these data (between 6 and 8 TBytes) can appear to the user as being completely on-line.

Our first experience with truly massive storage of data has been with the STK Silo. It is easy to say this technology has been extremely reliable and easy to use. But the reality is that it has

rapidly fallen behind in performance and capacity. A single STK Silo can hold about 2.4 TBytes of data on a good day. With newer 36-track 4490 tape drives, this capacity could double to 4.8 Terabytes. The read/write rate still hovers around 3 Mbytes/sec. The footprint for this storage is in the 150 sq ft range at a cost of about $600,000. With newer technology, it is possible to buy an 8 Terabyte system for the same price and a footprint of about 20 sq ft. However, STK is the incumbent and has plans to adopt the helical scan technology and integrate into the existing Silos. The table below gives you a comparison of the various tape technologies available today (all of which connect to a Convex computer as well).

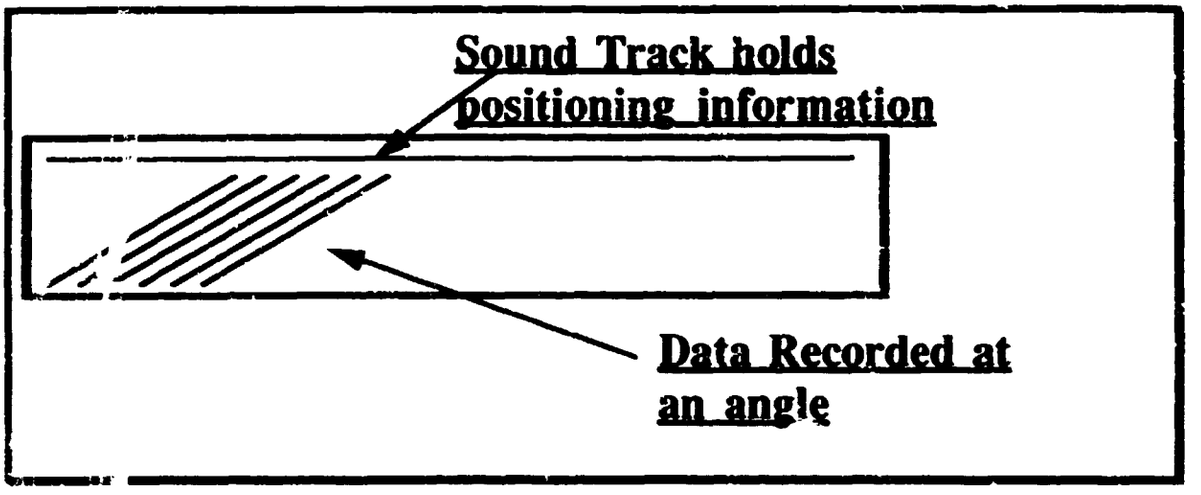
Tape Technology Comparison

DRIVE	CAPACITY	SPEED	COST
4mm	1 GB	.2 MB/s	$10k
34/4480	200 MB	3 MB/s	$30k
VHS	14 GB	3 MB/s	$40k
D2	25-165 GB	15 MB/s	$200k

One of the tape technologies clearly not shown is 8mm. It has been our experience that this tape technology is very unreliable in terms of re-reading tapes that have been written. We have also heard of the tape cartridges themselves breaking after only a few hundred robotic mounts and unmounts. Because of the relatively high rate of non-readability of the tapes, I believe anyone would be taking undo chances for using this technology for daily use of mass storage and data archiving. It may be this situation will change and then 8mm will become a very competitive product for this application.

Helical scan is an interesting technology that was specifically designed for the video industry. However, when taking this technology to store data, problems can occur. First of all, when recording video, the user of this technology does not care if the recording is absolutely without error. Since the human eye cannot perceive a single or double bit error in a million bits, it's not a problem. However, if this is digital data, it could easily represent the data for your bank account. Now the level of concern goes up a few notches. The *Bit Error Rate* (**BER**) for helical scan without extensive error detection and correction is easily below 1 bit in a few gigabytes of data. A couple of D2 and D1 vendors have produced drives of this quality. However, most users find this BER unacceptable. The drives produced by Metrum and Ampex have BERs of 1 bit in about 1 Terabyte of data. To achieve this level of BER, two additions were made to the recorders. The first is a read-after-write of the data that is being recorded. The recorder will continuously re-try the write of a block data until it has read back the entire block error free. For reading of data, 3 levels of error detection and correction are used to recover from tape errors due to bad heads or deteriorated tape. These drives are brand new and have just recently been put into production. Much factory testing has been done successfully and with extensive field testing. I believe these drives will prove to be mainstay products in mass storage.

Helical Scan Tape Technology



One of the interesting features of helical scan tape, especially in the D2 recorder from Ampex and VHS recorder from Metrum, is the sound track is used to store block and file positioning information. By doing this, the tape can be "searched" at almost rewind and fast-forward speeds. This alleviates the need to read the data on the tape to find EOFs between files to do tape positioning. I believe this feature will help these tape drives to be even more popular for the virtual disk application. However on the down side, these drives are designed to use large blocks of data to achieve the high data rates. So in a fileserving type of application where there are many small files stored on the tape, the performance will drop dramatically. Also, if this fast search capability is not used by the software or not supported by the drive, then the file retrieval and tape search operations bring the performance to a crawl. These two *features* have to be worked around by the software that controls these drives. For small file storage, the virtual disk software could consider doing clustering of these small files so that many files are written at once in very large blocks (and retrieved in like manner). For tape search, it is imperative that both the software and the hardware support the use of the sound track for block and file positioning on tape.

The last issue with the helical scan drives are the head wear. The average lifetime of the heads (there are typically 4 in a single drive), is *projected* to be about 500 hours of actual read/write time. At first, this seems pretty low, but the reality is that the tape heads are never constantly in use, much of the time is used to do tape search (if using the sound track) and rewind. Assuming there are at least two of these drives used in a virtual disk application, it would be safe to assume a head-use duty cycle of about 20-30% of wall clock time. This means that the heads have to be replaced about every 3 or 4 months. The heads are quite expensive, especially on the D2 tape. As these drives and heads go into mass production, the costs and durability of the heads should get much better.

A closing comment on the use of these new high speed drives. This new technology demands that you use a computer capable of keeping the drive busy, otherwise, why buy a drive that can run 15 MB/sec and your workstation can only push it to 5-10 MB/sec. Also, you will typically have more than one of these drives and this would further saturate any typical workstation today. At Convex our architecture is designed to support many such drives and also allow for simultaneous network activity. As people begin to benchmark such performance, it will become clear why the investment in the computer is as important (or more so) as the mass storage robotics and drives.

The Realities of Fileserver and Mass Storage Software

As mentioned earlier, the clear trend for most computer centers is to on-line all or most of their data. The recent availability of software to do under Unix has made it both possible and extremely desirable. Another effort to make this more available is the effort of the IEEE Mass Storage Working Committee. This Committee is tasked with coming up with a model or *standard* for the design and implementation of a software based mass storage system. At this point, the model is very high level and thus, almost any data management software system is compliant. As these folks make the model more detailed, it will cause proprietary systems and products to be revised or replaced.

The first application that most people are interested in includes that of the virtual disk concept. A typical computer center actively uses about *10-20%* of the data on the disk. The other 90% would be considered old. Many vendors, including Convex have implemented software to handle this capability. We have modified ConvexOS, a Unix based OS, to be able to recognize and generate *file-faults*; the ability to access a block of data that does not currently exist on disk and should be *paged* into the disk from tape by the operating system. We detect this fault at the read or write level, not the open. By detecting the fault at the read/write level, this "file-fault" feature can be used by our native NFS implementation. In this manner, client computers using Convex filesystems via NFS can store data on the NFS mounted filesystem and have it migrated to tape automatically. If the file fault was detected only on the open of a file, NFS could not be used, since NFS opens are not propagated back to the NFS server. In a nutshell, we have treated the disk space much like we treat physical memory. We can page in individual blocks of a file on demand, with read ahead, just like we page in pages of virtual memory to physical memory. It is obvious though that this demand paged virtual disk must be tuned very carefully, otherwise you could easily have a thrashing and resource allocation problem due to the demand placed on it by the fast processor. In general, the *paging* feature should only be used on very large files; otherwise, it is best to have the entire file read in when a fault occurs.

By having the file-fault feature in the kernel, the migration of the file from disk to tape and back is totally transparent to the applications running on the system. All read, write, open and close system calls can be used without change. The migration of files is done until a disk-full condition or periodically as needed to keep the disk at a low-water mark for free space. One of the main problems that will occur when using the native Unix filesystem, is that you could eventually run out of inodes (file handles) for your filesystem. If a virtual filesystem is terabytes in size, you could assume this could easily represent hundreds of millions of files. I would guess this is one of the major problems that will have to be solved in the future for all native Unix filesystems.

Another application that is very popular is a centralized network archive/fileserver system. At Convex we have embraced UniTree as the primary software package to handle this. UniTree creates and exports its own filesystem to client computers on a network. The client computers access the files in UniTree via FTP and/or NFS. The client computers simply store data to this filesystem using familiar interfaces and the host or server computer running UniTree ensures that the data is migrated to tape as needed. With UniTree and the new tape technology, it is possible to have a network fileserver/archiver that can store on order of several terabytes that can be transparently accessed by client computers and it does not have the limited number of inodes problem that occurs with normal Unix filesystems.

One of the major weaknesses with network based archiving and fileserving using this new technology is that NFS (Network File System) has not a clue about file migration and long access times due to tape mounts and such. In the end, something will have to be done to solve these problems. The timeout problem with NFS is fixed by simply tuning all your NFS clients to have a longer timeout period for those filesystems that are known to be under file migration control. The other problem of identifying files that are migrated cannot be easily fixed with NFS. When doing a long listing of the files on that filesystem, there is no way to tell which files

are migrated and which ones are not. Also, there is no easy signalling mechanism to inform the user that the file being accessed is currently being staged into disk. This is something that will eventually have to be taken care of by some means.

At Convex, we have installed several UniTree systems around the world. We found quickly that UniTree out of the box from DISCOS (the providers of UniTree) was far from being a production level product. Working with Titan Corporation, we have produced the world's first production quality UniTree system. We found many problems related to data integrity, disk full conditions and in general where the UniTree system is stressed by hundreds of requests. DISCOS continues to improve the quality of the software and believe that as other vendors bring the product to market that it will become the dominant file management product.

One of the critical things we have learned is that when brining up any mass storage solution, it should be done slowly. Trying to move a Terabyte of data into a *on-line* state in one or two days is not ideal since the system may indeed work fine, but will be swamped with all this new data that in reality only 10% of it will ever really be used. It will take several days or maybe even weeks for a mass storage system to become stable. Stable, in this context, would mean that most of the data that will be used on a regular basis will be in the disk cache and the data not used often or at all will be in the tape system.

Related to network fileserving is that of the Distributed Computing Environment (DCE) from OSF. The Distributed File System (DFS) portion of DCE (known as the Andrew filesystem) will allow for the creation of a single monolithic filesystem over an unlimited number of computers. Many of the companies I have talked to are very interested in using this technology. Currently DFS is not in production and only used experimentally mainly at Universities.

One of the weaknesses of DFS already is that it does not have the ability to migrate files to and from tape. So when a DFS filesystem gets full, you have the same problems as before with the normal Unix filesystem. However, there are efforts under way to integrate DFS with UniTree to allow for a virtual disk that serves a whole network of computers. I believe this will create a "perfect" world for most people as long as Andrew and UniTree live up to their billings.

Another technology that is based on Convex hardware and software is the *EMASS* storage system from E-Systems. It is an integrated data management solution consisting of D2 tape drives and robots and the Fileserv software. One of the basic concepts of EMASS is that of scalable/growable data storage. It is a fact that as people on-line their data, they will continue to need more and more data storage. With the EMASS DataLibrary, you can grow from about 27 Terabytes for the first module, up to 10 PETABYTES. There are many sites in the world that could use this capability and capacity today. The average amount of storage used by centers today is about 1 Terabyte. By the year 2000, this number will grow to about 100 Terabytes or more. Given this to be the case, EMASS is positioned well to handle these requirements.

The Fileserv software is very extensive in its ability to track and store data. It has a very rich accounting system and the interface to the system is via the normal ConvexOS filesystem using our file-fault interface. One of the most interesting features about Fileserv is that it supports what I call *tape sniffing*. This is the ability to automatically track the BER of all the tapes in the system so that as tapes begin to degrade in readability, the data is copied to another tape and the bad tape ejected from the system. Even those tapes that are never read through normal demands on the system are tracked by simply reading those tapes. This activity is tunable and is a background process that does not add a significant load to the system. The nice thing about this feature is that it puts a stop to the question of how long a storage media lasts. In this case, it really doesn't matter as long as it's reasonable (say 5-10 years).

Summary

In summary, I can say that fundamentally every data center in the world is or will shortly be very interested in solving their data management problem through far more efficient and effective means than what they have today. I also believe that helical scan tape technology will be the mainstay storage technology to accomplish this, coupled with an IEEE Mass Storage Reference Model based software system. I think that it will be commonplace to have DCE/Andrew on most computers with at least one Andrew server running UnfTree as the virtual disk manager.

I also believe that as more and more computer centers on-line most of their data, that they will want the ability to understand what data they have to better utilize it. This is generally known as the *meta-data problem*. The intent is that if there are several million files, how do you know if you processed/read all the data on a given item or topic? By integrating expert systems and object-oriented databases with the file management software, users can have an extremely productive tool that in some cases would give companies a competitive edge for their particular applications. There are some people in the world just starting to really work this issue and I believe within a couple years there will be substantial prototype systems available to help manage this problem. An interesting side effect of the meta-data problem is that companies will need to generate more and more meta-data by "massaging" their data. This will require both more processing power and more storage, so it is imperative that companies invest properly in their computers so they can expand both the processing capability and their mass storage options easily.

There is also a large portion of the computer population that is not well informed on the state of mass storage technology. I believe as people learn about this leading edge technology, demands for totally integrated mass storage solutions will increase dramatically.

Measurements Over Distributed High Performance Computing And Storage Systems

Elizabeth Williams
Supercomputing Research Center
17100 Science Drive
Bowie, Maryland 20715-4300

Tom Myers
Department of Defense
9800 Savage Road
Ft. Meade, Maryland 20755-6000

1.0 Introduction

Requirements are carefully described in descriptions of systems to be acquired but often there is no requirement to provide measurements and performance monitoring to ensure that requirements are met over the long term after acceptance. A set of measurements for various Unix-based systems will be available at the 1992 Goddard Conference on Mass Storage Systems and Technologies. The authors invite others to contribute to the set of measurements. This abstract gives the framework for presenting the measurements of supercomputers, workstations, file servers, mass storage systems, and the networks that interconnect them. Production control and database systems are also included. Though other applications and third party software systems are not addressed, it is important to measure them as well.

The capability to integrate measurements from all these components from different vendors, and from the third party software systems has been recognized and there are efforts to standardize a framework to do this. The measurement activity falls into the domain of management standards. Standards work is ongoing for Open Systems Interconnection (OSI) systems management; AT&T, Digital and Hewlett-Packard are developing management systems based on this architecture even though it is not finished. Another effort is in the UNIX International Performance Management Working Group [1]. In addition, there are the Open Systems Foundation's Distributed Management Environment and the Object Management Group. A paper comparing the OSI systems management model and the Object Management Group model has been written [2].

The IBM world has had a capability for measurement for various IBM systems since the 1970's and different vendors have been able to develop tools for analyzing and viewing these measurements. Since IBM was the only vendor, the user groups were able to lobby IBM for the kinds of measurements needed. In the UNIX world of multiple vendors, a common set of measurements will not be as easy to get. It is hoped that this paper will strengthen the effort to describe a minimum set of measurements.

2.0 Uses for Measurements

Seven types of uses have been identified. These are:

(1) distributed computing system scheduling

(2) fire-fighting - solve immediate problems to provide acceptable response time and resource allocation to all processes

(3) tuning systems for current workloads

(4) capacity planning

(5) allocating resources

(6) looking for trends and characterizing workloads

(7) verifying system strategies are working or assumptions about workloads are valid, e.g. locality of reference

The following two points are very important. (1) For fire-fighting and tuning, a systems administrator must be able to link a particular "event" to a set of user commands.The systems administrator should be able to know when a resource is responding slowly and which process is causing the problem. We stress that it is important to be able to link particular events of interest back to user commands though we know that it is sometimes difficult. (2) Process as well as system-wide measurements are needed.

It is also understood that taking measurements and collecting them are overhead and may in extreme cases affect the performance of the systems measured; this is not specifically addressed in this paper. However, data can be collected at various levels of detail depending on how much overhead is involved. The most complete level of measurement is a log or trace of each transaction or event. The next level of measurement is a set of counters that produce a histogram, which is an approximation to the distribution, of the metric of interest. The least detailed level of measurement is a simple counter from which the average and variance of the metric of interest can be derived. The level of measurement for any component depends on the overhead associated with the workload.

3.0 Model of Distributed High Performance Computing Systems

In Figure 1 we present a model of the components of a distributed high performance computing system. This model includes input sources to indicate the collection of data for processing in the system. The distributed characteristics of this model are not depicted specifically but one can think of NASA's EOS system as the basis for this model.

The components in the model are supercomputers, workstations, mass storage system, file servers, networks, input machines, database systems and production control systems. The model represents both hardware and distributed software aspects of the components. Each circle represents a hardware component. Each square represents a software component that may be implemented on some subset of the hardware components. The network is represented by arrows indicating interconnections. The dots indicate a set of distributed components.

Below the system component level are lower level resources that are also necessary to measure. These are the hardware resources such as CPU, memory, disks, channel, external I/O, paging and caches, and the software resources such as buffers and queues.

Measurements at both system component and hardware/software resources levels are desired.

4.0 References

(1) Leon Traister and Terry Flynn, "A Measurement Architecture for Unix-Based Systems", CMG Transactions, Winter, 1991, pp. 69-77.

(2) Peggy Quinn and George Preoteasa, "Reconciling Object Models for Systems and Network Management", Technical Report, UNIX System Laboratories, Inc.

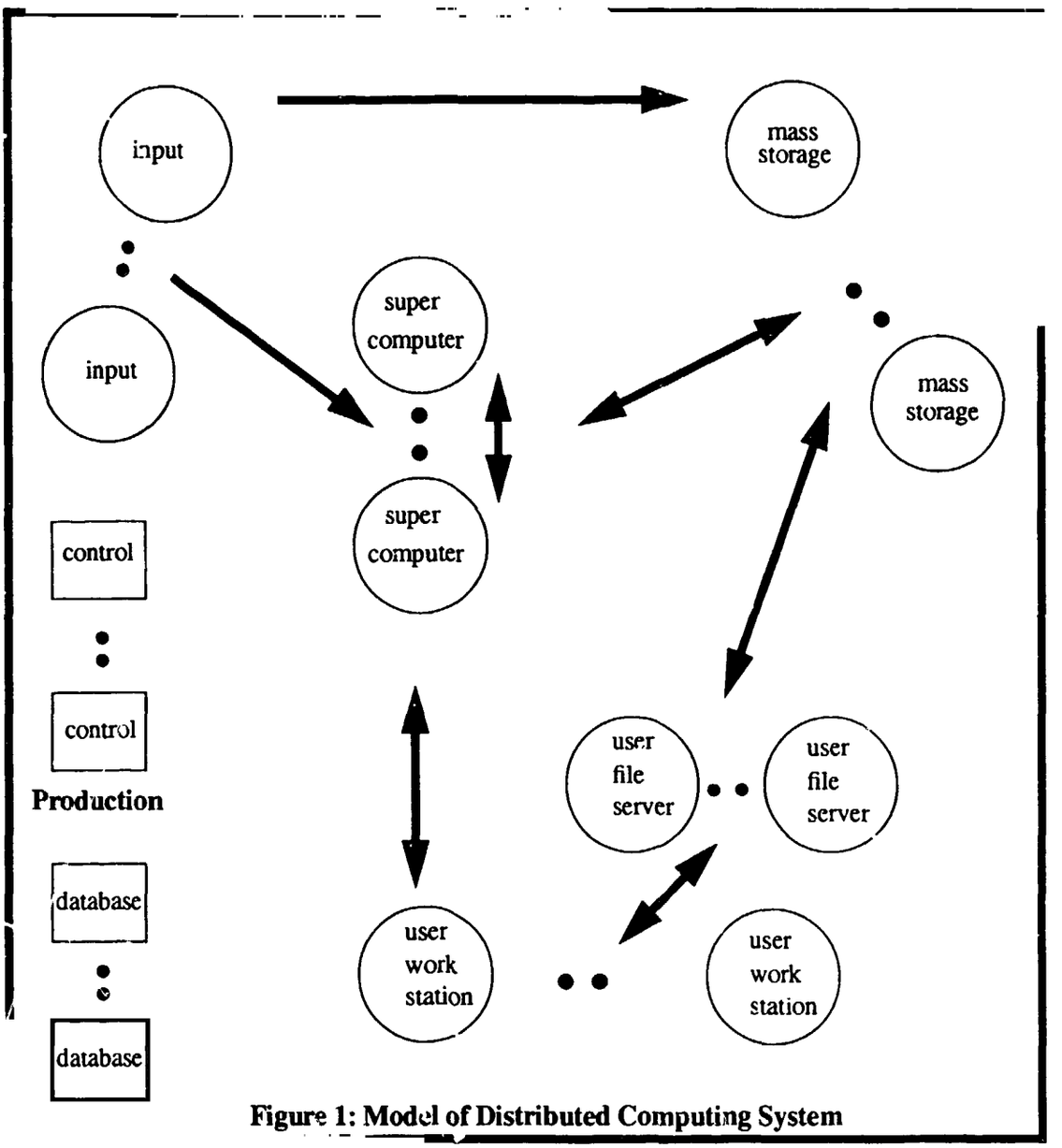


Figure 1: Model of Distributed Computing System

N93-30477

Analysis of Cache for Streaming Tape Drive

V. Chinnaswamy
8 Quail Hollow Road
Westboro, MA 01581

S28-35
15511 8
P. 12

1 Introduction

A tape subsystem consists of a controller and a tape drive. Tapes are used for backup, data interchange and software distribution. This paper is concerned only with the backup operation. During a backup operation, data is read from disk, processed in CPU and then sent to tape. The processing speeds of a disk subsystem, CPU and a tape subsystem are likely to be different. A powerful CPU can read data from a fast disk, process it and supply the data to the tape subsystem at a faster rate than the tape subsystem can handle. On the other hand, a slow disk drive and a slow CPU may not be able to supply data fast enough to keep a tape drive busy all the time. The backup process may supply data to tape drive in bursts. Each burst may be followed by an idle period. Depending on the nature of the file distribution in the disk, the input stream to the tape subsystem may vary significantly during backup. To compensate for these differences and optimize the utilization of a tape subsystem, a cache or buffer is introduced in the tape controller.

Most of the tape drives today are streaming tape drives. A streaming tape drive goes into reposition when there is no data from the controller. Once the drive goes into reposition, the controller can receive data, but it cannot supply data to the tape drive until the drive completes its reposition. This reposition time may vary from several milliseconds to a few seconds depending on the technology of the drive. A controller can also receive data from the host and send data to the tape drive at the same time.

This paper investigates the relationship of cache size, host transfer rate, drive transfer rate, reposition and ramp up times for optimal performance of the tape subsystem. Formulas developed here will also show the advantages of cache watermarks to increase the streaming time of the tape drive, maximum loss due to insufficient cache, trade offs between cache and reposition times and the effectiveness of cache on a streaming tape drive due to idle times or interruptions due in host transfers.

In Section 2, several mathematical formulas are developed to predict the performance of the tape drive. Some examples are given in Section 3 illustrating the usefulness of these formulas. Finally, a summary and some conclusions are provided in Section 4.

2 Mathematical Analysis

The performance of a tape subsystem depends on several variables and their relationships. In this section, several formulas are developed for the throughput of the tape drive.

Let

- λ denote the host transfer rate,
- μ, the drive transfer rate,
- C, the cache size.

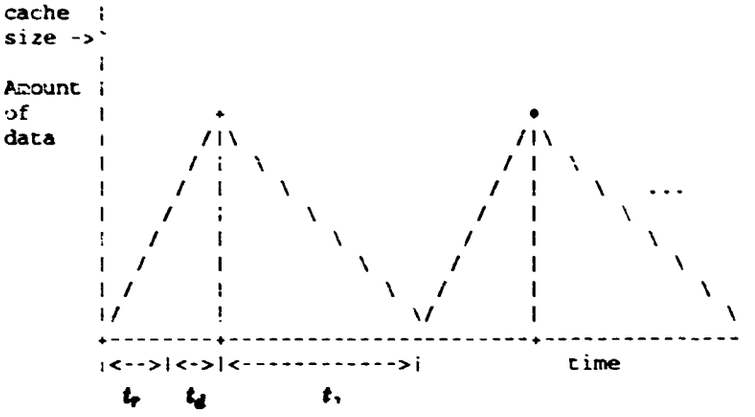
- t_r, the reposition time,
- t_d, the ramp up time delay to request,
- t_s, the streaming time before next reposition.

Any other variables of interest will be defined as needed. For now, refer to $P = t_r + t_d + t_s$ as a period. All the throughput numbers will be in kilobytes per second. All times will be in seconds.

2.1 Host transfer rate < the drive transfer rate

case i: $\lambda < \mu, \lambda(t_r + t_d) < C$, no idle time in host transfer,

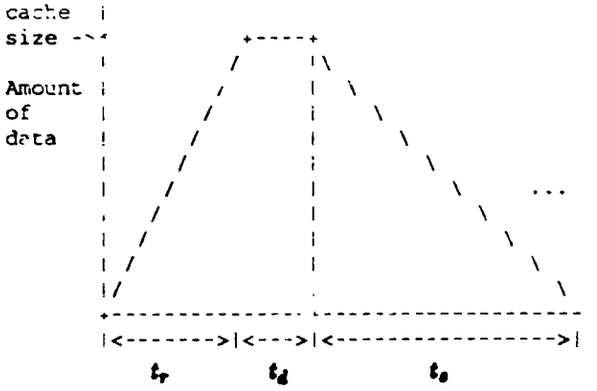
i.e., cache does not get filled up during reposition and ramp up time.



t_s denotes the drive streaming time. Each period repeats itself until the whole backup operation is over. So throughput can be calculated from just one period.

case ii: $\lambda < \mu, \lambda(t_r + t_d) > C$, no idle time in host transfer,

i.e., cache gets filled up during reposition and ramp up time.



When cache gets full and the drive is in reposition, host transfer gets blocked. When this happens bandwidth is lost. There is no idle time in host transfer except during this blocking time. The above two cases will be analyzed first before getting into several other cases.

Analyzing the figures for one period, we get the following relationships:

$$\lambda(t_r + t_d + t_e) = \mu t_e \quad if\ \lambda(t_r + t_d) \leq C$$

$$C + \lambda t_e = \mu t_e \quad if\ \lambda(t_r + t_d) > C$$

From these two equations, we can get the value of t_e

$$t_e = \begin{cases} \frac{\lambda(t_r+t_d)}{(\mu-\lambda)} & if\ \lambda(t_r + t_d) \leq C \\ \frac{C}{(\mu-\lambda)} & if\ \lambda(t_r + t_d) > C \end{cases}$$

Since the process repeats itself for each period, the effective throughput, T, of the tape subsystem can be calculated from one period.

$$T = \frac{Total\ Data\ Transferred\ by\ Drive}{Total\ Time} = \frac{\mu t_e}{(t_r + t_d + t_e)} = \frac{\mu}{1 + \frac{(t_r+t_d)}{t_e}}$$

We may often refer to T as an approximate throughput since we are neglecting the initial time due to label checking, track turn around time, etc. However, these times would become negligible when we are considering several hours of backup time.

Using the conditions above, we get

$$T = \begin{cases} \lambda & if\ \lambda(t_r + t_d) \leq C \\ \frac{\mu}{1 + \frac{(t_r+t_d)(\mu-\lambda)}{C}} & if\ \lambda(t_r + t_d) > C \end{cases}$$

2.1.1 Maximum loss in effective throughput

Wher $\lambda < \mu$ and $\lambda(t_r + t_d) > C$, the host transfer is blocked when the drive is in reposition. In this case, there is a loss in throughput due to insufficient cache.

The loss in throughput due to insufficient cache is given by

$$L = \lambda - \frac{\mu}{1 + \frac{(t_r+t_d)(\mu-\lambda)}{C}}$$

Differentiating with respect to λ, we can prove that the maximum loss occurs when

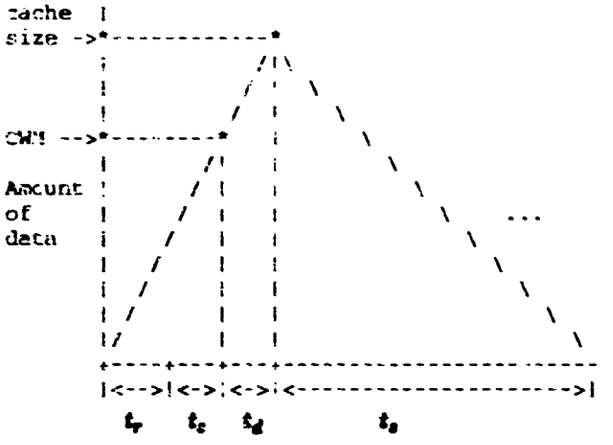
$$\lambda = (\frac{C}{t} + \mu) - \sqrt{\frac{C * \mu}{t}}$$

where $t = t_r + t_d$. For C = 512, t = 1.35, and μ = 800, the maximum loss occurs when λ = 628. When λ = 628 KB/sec, we get only a throughput of 550 KB/sec, a loss of 78 KB/sec.

2.1.2 Cache Watermarks

When $\lambda(t_r + t_d) < C$, we might think of introducing a watermark level at $C - \lambda t_d$ such that we fill up the cache before the tape drive starts transferring data.

When the controller tells the drive to start writing data, the drive does not start writing data immediately. There is a ramp up delay in its response time. This time is not negligible for some drives. Suppose the ramp time is .5 seconds. If the drive is told to transfer data when the cache is 100 percent full, the host transfer will be blocked for 500 milliseconds.



In this case, we have

$$\lambda(t_r + t_c + t_d + t_s) = \mu t_s \text{ if } \lambda(t_r + t_d) < C$$

where t_c is the additional wait time to bring the data in cache to the watermark level.

There is no point in setting a watermark if $\lambda(t_r + t_d) \geq C$.

Solving for t_s, we get

$$t_s = \frac{\lambda(t_r + t_c + t_d)}{(\mu - \lambda)}$$

Throughput = $T = \frac{\mu t_s}{(t_r + t_c + t_d + t_s)} = \frac{\mu}{1 + \frac{(t_r + t_c + t_d)}{t_s}}$

Using the value of t_s, we get

$$T = \frac{\mu}{1 + \frac{(\mu - \lambda)}{\lambda}} = \lambda$$

Introducing a cache watermark has not changed the throughput. But the streaming time has increased (and consequently the number of repositions during a given time has decreased) since

$$t_s = \frac{\lambda(t_r + t_c + t_d)}{(\mu - \lambda)} > \frac{\lambda(t_r + t_d)}{(\mu - \lambda)}$$

Given the total time or total amount of data, we can easily calculate the number of repositions saved by using the cache watermark. If increased number of repositions causes any reliability concerns, it is worth considering introducing cache watermarks when $\lambda(t_r + t_d) < C$. When $\lambda(t_r + t_d) > C$, there is no point in introducing a cache watermark. It does not change the throughput.

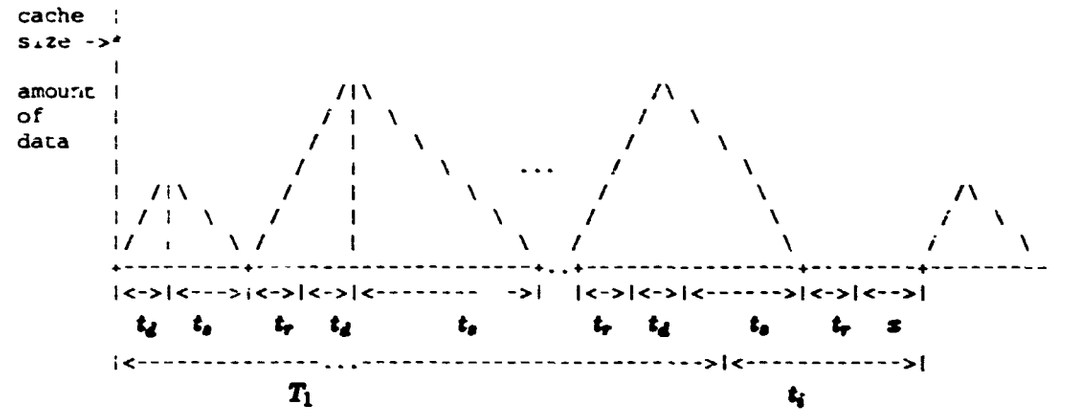
2.1.3 Host transmission has idle periods

Let

- T_1 be the continuous host transfer time
- t_i be the idle period before the next transmission.

We will assume that these times are contants and do not vary from period to period.

case iii: $\lambda < \mu, \lambda(t_r + t_d) < C, t_i > \frac{C}{\mu} + t_r$

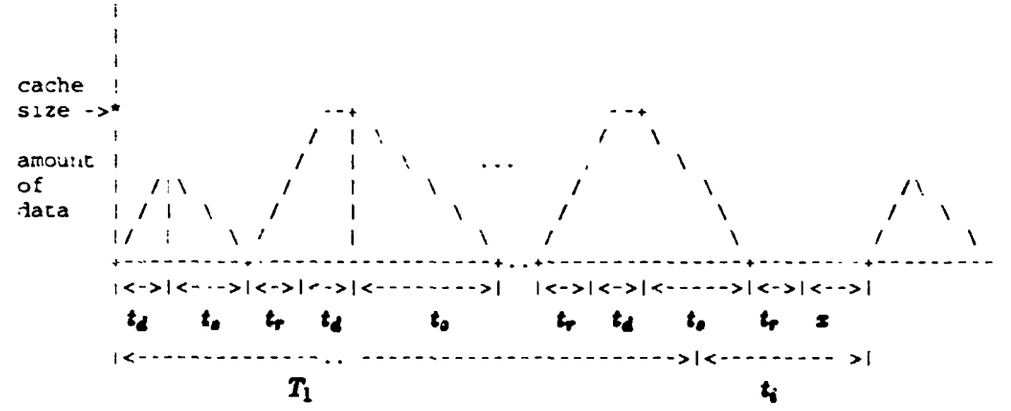


The approximate effective throughput is given by

$$T \approx \frac{T_1 \lambda}{T_1 + t_i}$$

The results are also true for the case $0 < t_i < \frac{C}{\mu} + t_r$

case iv: $\lambda < \mu, \lambda(t_r + t_d) > C, t_i > \frac{C}{\mu} + t_r$



In this case, the host transfer is blocked when the cache gets full. The approximate effective throughput is given by

$$T = \frac{[T_1 - n(t_r + t_d - \frac{C}{\lambda})]\lambda}{T_1 + t_i}$$

where n is given by

$$n = \lfloor \frac{T_1 - \frac{\lambda t_d}{\mu - \lambda}}{t_r + t_d + \frac{C}{\mu - \lambda}} \rfloor$$

If $[T_1 - n(t_r + t_d + \frac{C}{\mu - \lambda})] > (2 + \frac{\lambda}{\mu - \lambda})t_d + t_r$

then n = n + 1.

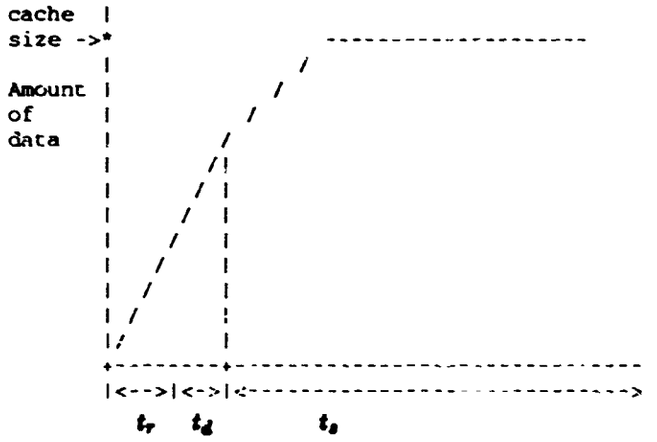
The results are also true for the case $0 < t_i < \frac{C}{\mu} + t_r$

2.2 Host Transfer Rate > Drive Transfer Rate

In this section, we will analyze all cases arising from the condition when host transfer rate exceeds the transfer rate of the drive.

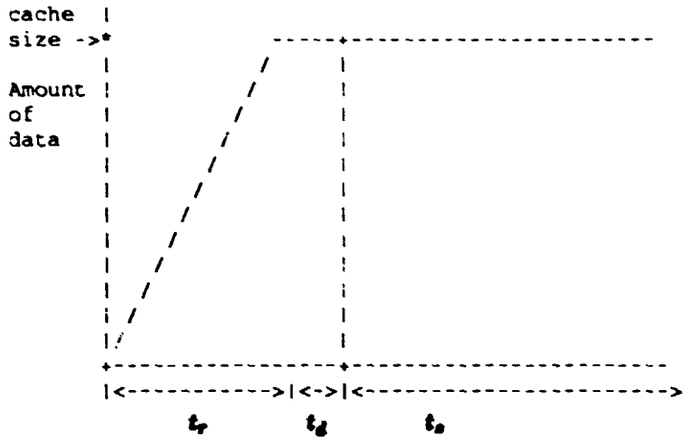
2.2.1 No Idle Time in Host Transfer

case v: $\lambda > \mu$, $\lambda(t_r + t_d) < C$, no idle time in host transfer.



In this case, input is blocked as soon as cache is full. Input rate will be limited to the output rate. Effective throughput of the tape drive is the maximum throughput capacity of the tape drive. Cache has no significant impact.

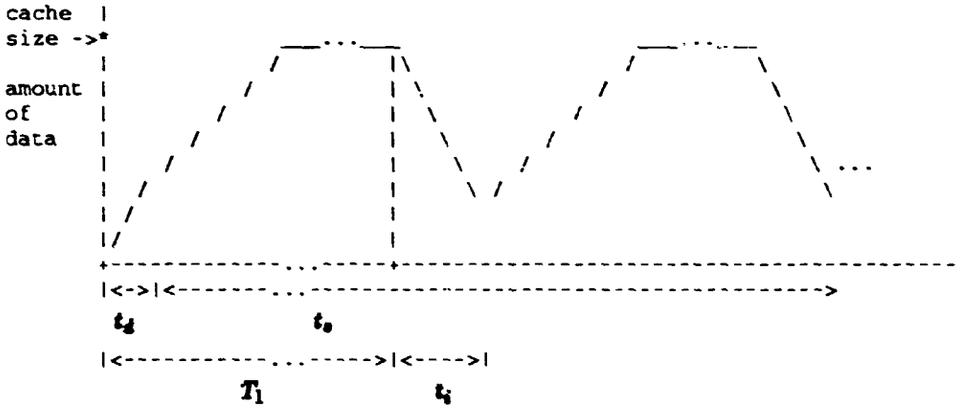
case vi: $\lambda > \mu$, $\lambda(t_r + t_d) > C$, no idle time in host transfer.



In this case, input is blocked as soon as cache is full. There is no input or output transmission for some period. This is a lost bandwidth. After this no transmission period, input rate will be limited to the output rate. Cache again has no impact.

2.2.2 Host transmission has idle periods

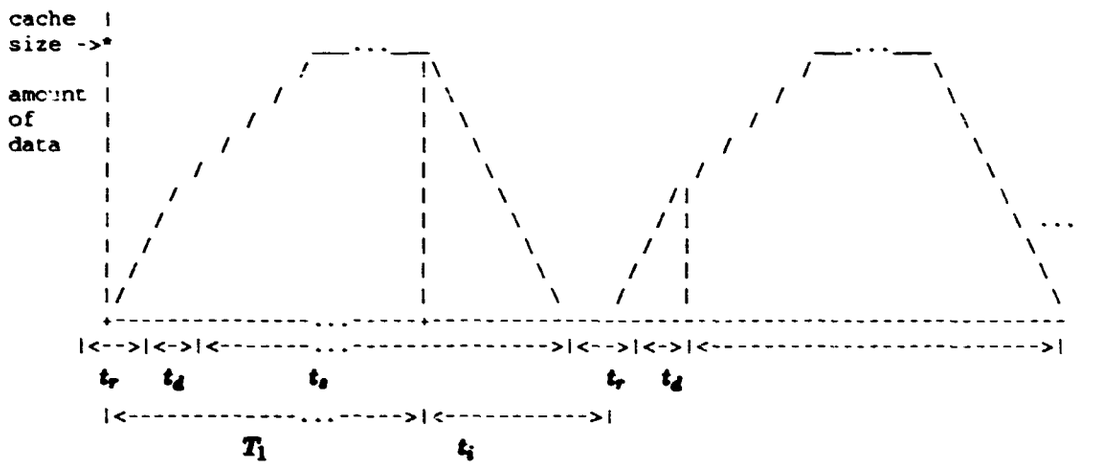
case vii: $\lambda > \mu, \lambda(t_r + t_d) < C, t_i < \frac{C}{\mu}$ or $\lambda > \mu, \lambda(t_r + t_d) > C, t_i < \frac{C}{\mu}$



In this case, the input transmission begins before the drive empties the cache. The drive streams all the time. The effective throughput is approximately the same as the drive transfer rate.

case viii: $\lambda > \mu, \lambda(t_r + t_d) < C, \frac{C}{\mu} < t_i < \frac{C}{\mu} + t_r$ or $\lambda > \mu, \lambda(t_r + t_d) > C, \frac{C}{\mu} < t_i < \frac{C}{\mu} + t_r$

i.e., the input transmission begins after the drive empties the cache, but before the drive completes its reposition.



$$T = \frac{(T_1 + t_i - \frac{C}{\mu} - t_r - t_d)\mu + C}{(T_1 + t_i)}$$

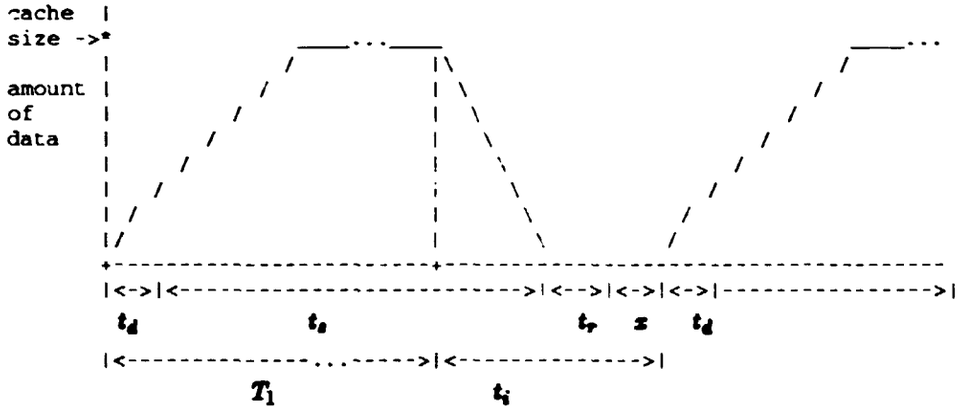
$$= \frac{(T_1 + t_i - t_r - t_d)\mu}{(T_1 + t_i)}$$

$$= (1 - \frac{t_r + t_d}{T_1 + t_i})\mu$$

The approximate effective throughput is less than the drive transfer rate. How much less will depend on the reposition time and idle time.

case ix: $\lambda > \mu, \lambda(t_r + t_d) < C, t_i > \frac{C}{\mu} + t_r$ or $\lambda > \mu, \lambda(t_r + t_d) > C, t_i > \frac{C}{\mu} + t_r$

The idle period is longer. The drive empties cache, completes reposition and then waits for data.



The effective throughput for one period is given by

$$T = \frac{(T_1 - t_d)\mu + C}{T_1 + t_i}$$

$$= \frac{(T_1 + t_i)\mu - t_i\mu + C - \mu t_d}{T_1 + t_i}$$

$$= (1 - \frac{t_i + t_d - \frac{C}{\mu}}{T_1 + t_i})\mu$$

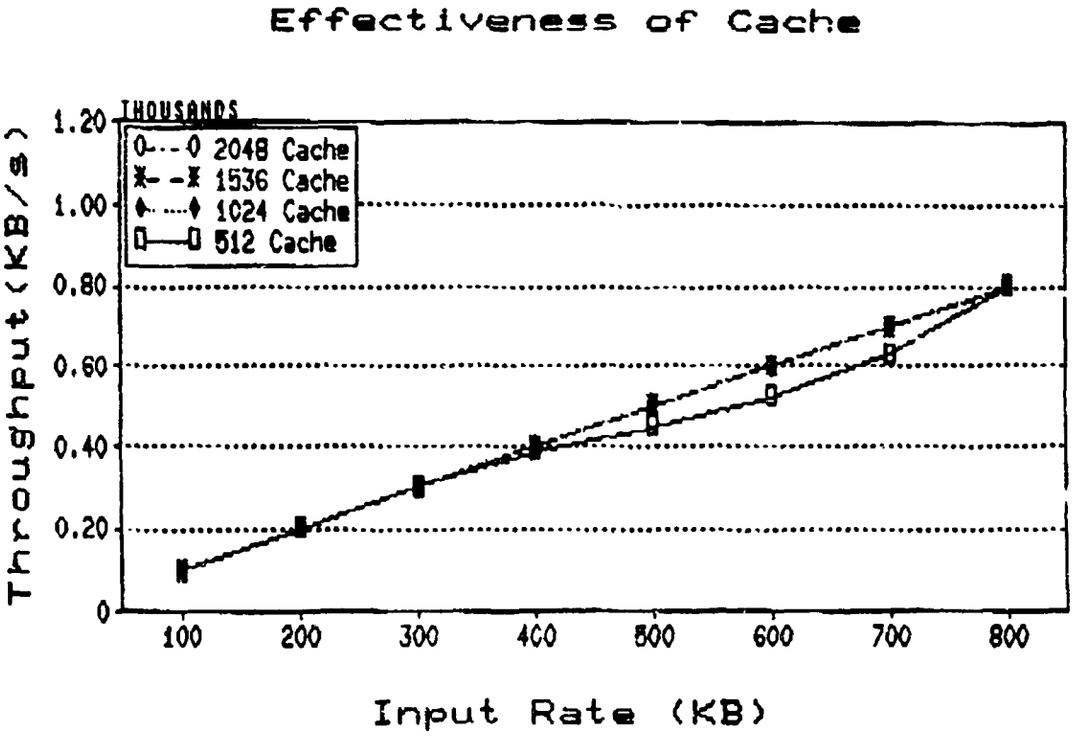
$$= (1 - \frac{t_r + t_d + z}{T_1 + t_i})\mu$$

The approximate effective throughput is less than the drive transfer rate. How much less will depend on the reposition time, wait time and idle time.

3 Some Illustrative Examples

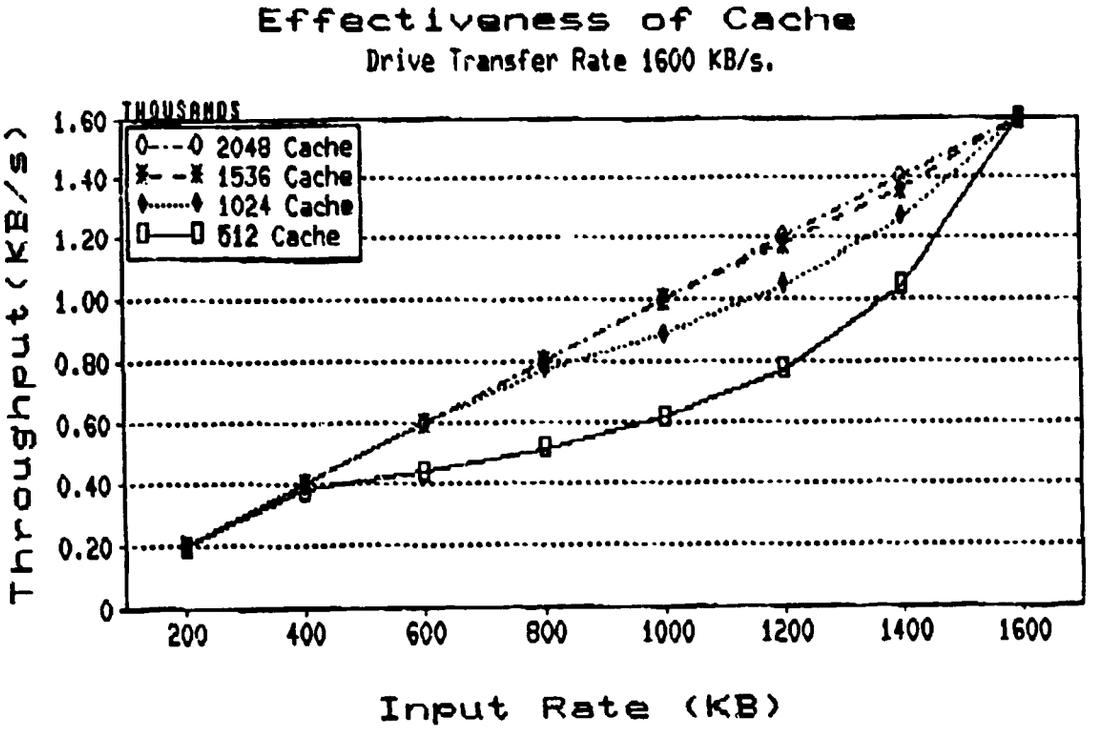
Suppose we have a drive with μ = 800, t_r = 1.0 second, t_d = 0.350 second, C = 512 KB. For continuous host transfer and for all $\lambda < \mu$, the graph in Figure 1 gives the throughput for different cache sizes. We lose some throughput with 512 KB cache. 1024 KB cache gives better performance than 512 KB cache. More than 1 MB cache seems to be a waste.

Figure 1: Cache Sizes and Throughput



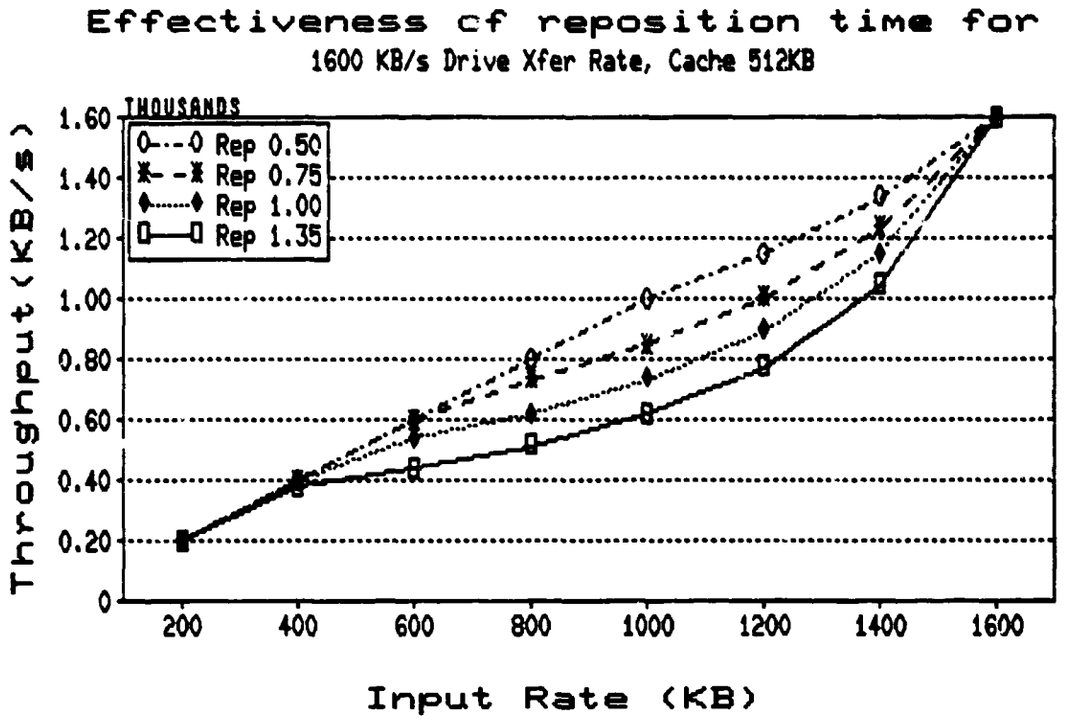
Let us consider the effect of increasing only the tape drive speed, i.e., μ = 1600, t_r = 1.0 second, t_d = 0.350 second. Figure 2 shows the performance for various cache sizes. For all $\lambda < \mu$, increasing the drive transfer rate will decrease the performance of the system unless there is an increase in cache size. A cache size of 2 MB is needed when the drive transfer rate is increased to 1600 KB/sec.

Figure 2: Performance for Different Cache Sizes



A comparison of Figure 1 and Figure 2 shows that increasing the transfer rate of the tape drive without a comparable increase in cache size and/or decrease in reposition time has a negative impact in the performance for certain range of input values. The throughput can be increased by reducing the reposition and ramp up time instead of increasing the cache size.

Figure 3: Performance for different reposition times



4 Summary of Results and Conclusions

case i: $\lambda < \mu, \lambda(t_r + t_d) < C$, no idle time in host transfer $\Longrightarrow T \approx \lambda$

case ii: $\lambda < \mu, \lambda(t_r + t_d) > C$, no idle time in host transfer $\Longrightarrow T \approx \frac{\mu}{1+\frac{(t_r+t_d)(\mu-\lambda)}{C}}$

case iii: $\lambda < \mu, \lambda(t_r + t_d) < C, t_i > \frac{C}{\mu} + t_r \Longrightarrow T \approx \frac{T_1 \lambda}{T_1 + t_i}$

case iv: $\lambda < \mu, \lambda(t_r + t_d) > C, t_i > \frac{C}{\mu} + t_r \Longrightarrow T \approx \frac{[T_1 - n(t_r + t_d - \frac{C}{\mu})]\lambda}{T_1 + t_i}$

case v: $\lambda > \mu, \lambda(t_r + t_d) < C$, no idle time in host transfer $\Longrightarrow T \approx \mu$

case vi: $\lambda > \mu, \lambda(t_r + t_d) > C$, no idle time in host transfer $\Longrightarrow T \approx \mu$

case vii: $\lambda > \mu, \lambda(t_r + t_d) < C, t_i < \frac{C}{\mu} \Longrightarrow T \approx \mu$

case viii: $\lambda > \mu, \lambda(t_r + t_d) < C, \frac{C}{\mu} < t_i < \frac{C}{\mu} + t_r \Longrightarrow T \approx (1 - \frac{t_r + t_d}{T_1 + t_i})\mu$

case ix: $\lambda > \mu, \lambda(t_r + t_d) < C, t_i > \frac{C}{\mu} + t_r \Longrightarrow T \approx (1 - \frac{t_r + t_d + \pi}{T_1 + t_i})\mu$

case x: $\lambda > \mu, \lambda(t_r + t_d) > C, t_i < \frac{C}{\mu} \Longrightarrow T \approx \mu$

case xi: $\lambda > \mu, \lambda(t_r + t_d) > C, \frac{C}{\mu} < t_i < \frac{C}{\mu} + t_r \Longrightarrow T \approx (1 - \frac{t_r + t_d}{T_1 + t_i})\mu$

case xii: $\lambda > \mu, \lambda(t_r + t_d) > C, t_i > \frac{C}{\mu} + t_r \Longrightarrow T \approx (1 - \frac{t_r + t_d + \pi}{T_1 + t_i})\mu$

When the host transfer rate is less than the drive transfer rate and if cache doesn't get filled up during reposition, the throughput rate would be the same as the host transfer rate. When the host transfer rate exceeds the drive transfer rate and either the host transfer has no idle time or the idle time is less than the time to empty cache, the throughput would be the same as the drive transfer rate. In all other cases, we lose throughput. The amount of loss would depend on the parameter values and their relationships.

In case ii, we lose throughput either because we have insufficient cache or the reposition time is high.

In cases iii, viii, ix, xi, and xii, we lose throughput because of idle time from host transfer. When there is an idle period, t_i, the tape drive

- will stream if $t_i < \frac{C}{\mu}$.
- will not stream if $t_i \geq \frac{C}{\mu}$.

In case iv, we lose throughput due to both idle time and insufficient cache.

These formulas are helpful to understand the behavior of the new tape subsystems when there are changes to any of the parameter values. They also predict the backup throughputs for any specified parameter values.

N93-30478

S29-82

159119

LANL High-Performance Data System (HPDS)

M. William Collins, Danny Cook, Lynn Jones, Lynn Kluegel, and Cheryl Ramsey
Los Alamos National Laboratory
Computer Systems Group MS B294
Los Alamos, New Mexico 87545

p. 6

Abstract

The Los Alamos High-Performance Data System (HPDS) is being developed to meet the very large data storage and data handling requirements of a high-performance computing environment. The HPDS will consist of fast, large-capacity storage devices that are directly connected to a high-speed network and managed by software distributed in workstations. This paper will present the HPDS model, the HPDS implementation approach, and experiences with a prototype disk array storage system.

Introduction

Advances in massively parallel, large-memory computers and high-speed cooperative processing networks have created a high-performance computing environment that allows researchers to execute large-scale codes that generate massive amounts of data. A large problem will generate from tens of gigabytes up to several terabytes of data. These requirements are one to two orders of magnitude greater than what the best supercomputing data storage systems are now able to handle and will require a new generation of data storage systems. As the massively parallel machines become more powerful, the data handling and data storage requirements will likewise increase, requiring even more powerful data storage systems.

To meet the data storage and especially the data handling requirements of this high-performance computing environment, a data storage system model, in which storage devices are directly connected to a high-performance network and data is transferred directly between the storage devices and the client machines, is needed.

HPDS Model

The High-Performance Data System (HPDS) model is based on storage devices that are connected directly to a high-performance network, such as a HIPPI-based (High-Performance Parallel Interface) network, so that data can be transferred directly between the storage devices and client machines, instead of the traditional method requiring an intermediary mainframe computer.

The HPDS model is shown in Figure 1. Disk devices are used to meet high-speed and fast-access requirements, and tape devices are used to meet high-speed and high-capacity requirements. By connecting the disk and tape devices directly to a high-speed network, higher data transfer rates and reduced hardware costs are realized. The model uses separation of control and data to provide increased flexibility and performance.

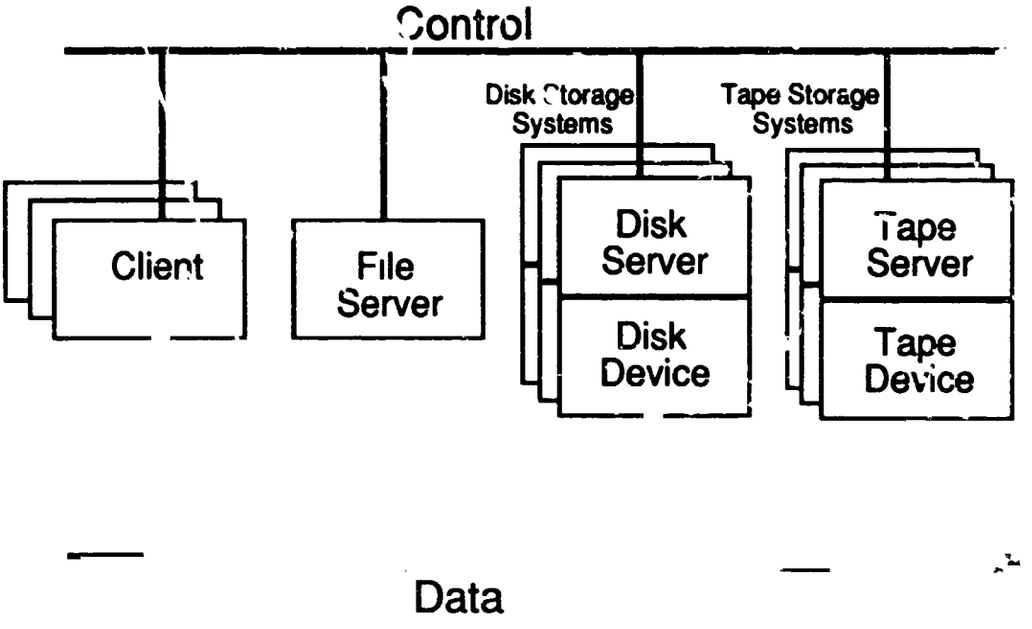


Figure 1. High-Performance Data System Model.

The recent availability of HIPPI-attached disk arrays allows implementation of a disk array storage system based on the HPDS model. Other computing installations have connected HIPPI-attached disk arrays to clients and have operated them in a master-slave mode with the client sending read/write commands to the disk array using a HIPPI command-data port. This mode necessitates implementing a device driver for each client and creates integrity and security problems because each client can read or write anywhere in the disk array.

A more secure approach, and one that allows a peer-to-peer data transfer between the disk array and the client machine, is to associate a workstation with the disk array to implement a disk array storage system. In the HPDS model, this workstation is referred to as the disk server. All requests to store and retrieve data are made to the disk server, which then issues the read/write commands to the disk array through an Ethernet "command-only" port using TCP sockets. The read/write commands specify that the disk array is to transfer the data to/from the client machines using the HIPPI "data-only" port. The disk array will not accept commands on its HIPPI data-only port, so access can only be through the disk server. The disk server will provide device management and storage management capabilities for the disk array and will implement a data transfer protocol with the client machine.

The same approach will be used for HIPPI-attached tape devices when they become available. A workstation-based tape server will be associated with a HIPPI-attached tape device to implement the tape storage systems shown in the HPDS model

The file server component of the HPDS model will implement user interface and file management capabilities that are distributed on multiple workstations.

HPDS Implementation

Implementation of the HPDS, as shown in Figure 2, is underway. The approach is to implement a series of prototypes that will provide improving capabilities for client machines in a timely manner. This approach will better allow new technologies to be used as they become available and for experience to be gained and used more effectively. Work has started on the file server and disk server prototypes. The various server components of the HPDS will be distributed on multiple workstations and will employ message communication using TCP sockets.

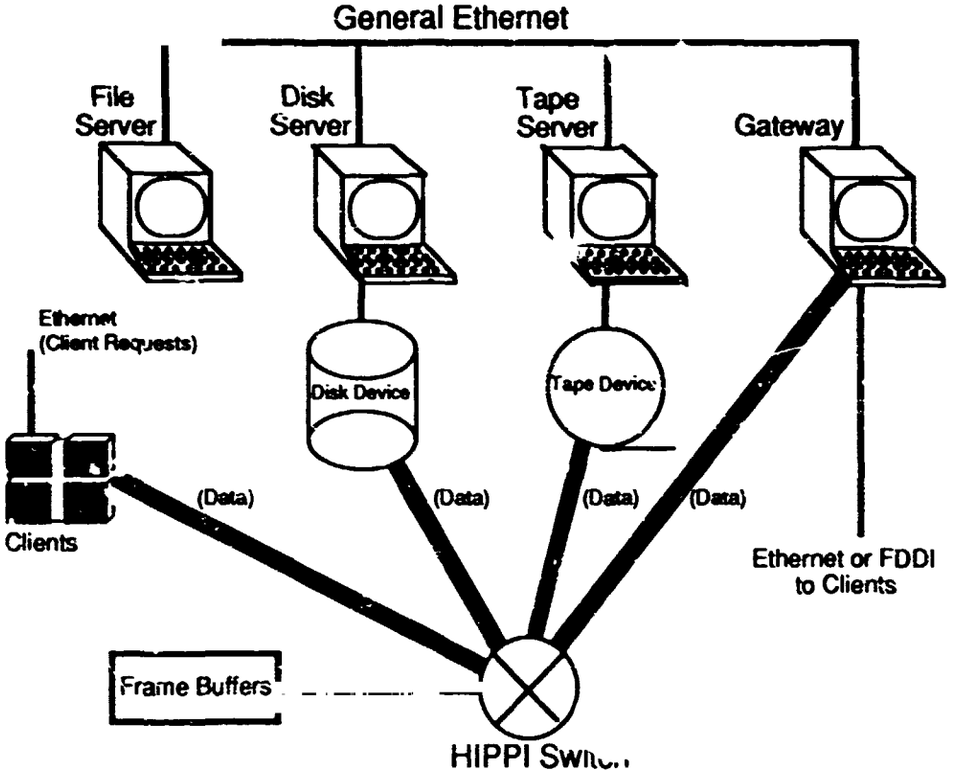


Figure 2. High-Performance Data System Implementation.

The file server will consist of user interface servers, name servers, and a location server. In the initial prototype, an interface called the Data Transfer Tool (DT Tool) will be implemented to transfer files or parts of files between a client machine and the HPDS. DT Tool will be implemented as a command interface on the client machines and as a DT Tool server on a file server workstation. DT Tool functionality, operation, and syntax will be similar to FTP. The function of the name servers is to map the user path names for files to a file identifier that is used to access the data from a storage system. The initial name server will provide a UNIX file structure and UNIX file management capabilities. The location server will map the file identifier to the storage system(s) that currently stores the file. The location server provides for the initial placement of files and for the subsequent migration/caching of files between different storage systems. Future user interfaces might include an implementation of an NFS-like transparent interface to HPDS files and a Metadata Tool that would allow users to build metadata files that describe and provide structured access to the data.

The principal functions of the disk server are to provide storage and device management for the disk array and to provide control for the data transfer process. A dedicated workstation will be used for the disk server, which will provide the view of logical storage spaces with requests to create/delete storage spaces, store/retrieve data in the storage spaces, query/modify attribute information, and status/abort requests. The disk server maps the logical storage spaces to the physical storage of the disk array.

Once the disk array storage system has been implemented and evaluated, a tape storage system will be implemented. HIPPI-attached tape devices are not available now but may become available before the end of 1992. An Ampex DD-2 helical scan recorder may be acquired for evaluation purposes and would be equipped with a HIPPI attachment when it became available. Possible use of HIPPI-attached DD-1 helical scan recorders and HIPPI-attached IBM 3490 tape devices is also being examined.

For a client machine to use the HPDS directly, the machine must have a HIPPI connection and must install special user interface and data transfer software. A "Gateway Machine" will be implemented to allow HPDS data store and data retrieval for machines that do not have a HIPPI connection or for machines where it is not practical/desirable to install the special software. The Gateway Machine will cache HPDS data and allow it to be accessed using standard protocols (i.e., FTP, NFS, AFS) over Ethernet and FDDI networks.

The IEEE Mass Storage System Reference Model and the emerging standards for the IEEE Model were used in the design of the HPDS to take advantage of the IEEE Model knowledge base, to make the HPDS more understandable to others, and to allow future hardware/software systems based on IEEE Model standards to be used. The HPDS file server implements the IEEE Model name server, location server, bitfile server, and migration functionality, while the HPDS disk server implements the IEEE Model storage server and bitfile mover functionality. The IEEE Model system management functions of storage management, operations, systems maintenance, and administrative control will be implemented.

Early Experiences

An early prototype disk array storage system was implemented by connecting an IBM RS/6000 workstation to the Ethernet command-only port of an IBM 9375 disk array. The IBM disk array consists of 16 data disks that provide a storage capacity of 23.3 gigabytes and a maximum data transfer rate of 65 megabytes per second. Storage management and device management software was implemented on the workstation.

As shown in Figure 3, the prototype disk array storage system was connected to the Los Alamos Advanced Computing Laboratory HIPPI network, which allowed the disk array storage system to have HIPPI connections with a Thinking Machines CM-2, a CRAY Y-MP, an IBM 3090, and a high-resolution HIPPI frame buffer.

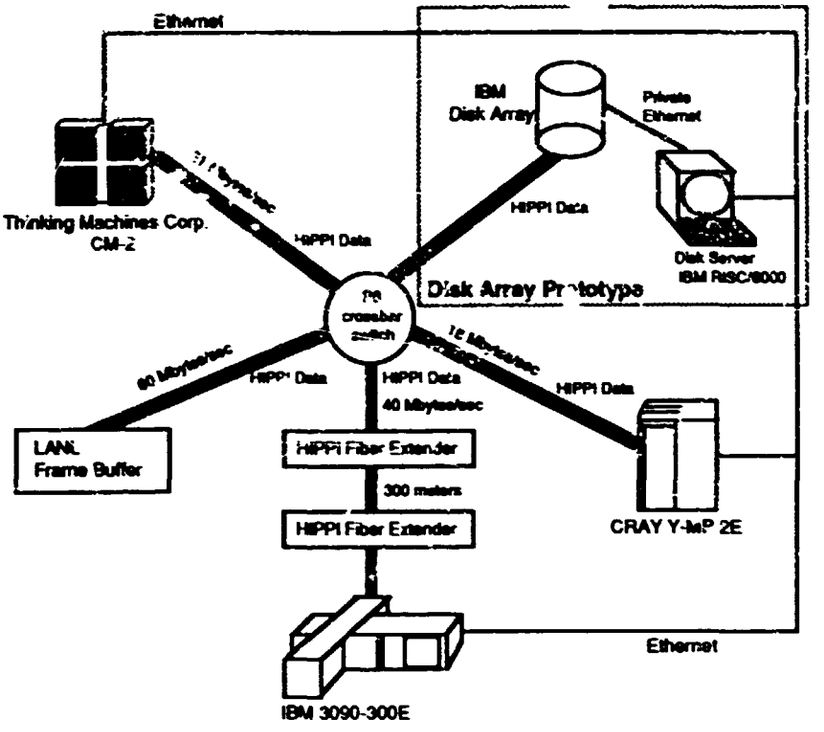


Figure 3. Prototype Disk Array Storage System.

A high-speed Data Transfer Protocol (DTP) that allows UNIX-based systems to transfer data over a HIPPI connection was implemented. DTP is based on the separation of control and data where control uses TCP socket connections, and "raw" data (data without headers) is transmitted over a HIPPI connection. This separation allows for reliable delivery of control messages, while simultaneously allowing large blocks of data to be transferred over the HIPPI with minimum overhead. DTP assumes that HIPPI error checking will detect essentially all data errors and that the error rate is low, so large data blocks (i.e., megabytes) can be used. The protocol provides flow control, block-level retransmission, and timeouts. Data transfer can consist of whole files, parts of files, or appending to the end of a file and can be initiated by either the sender or receiver. DTP is viewed as a temporary solution because the goal is to use TCP sockets for the HIPPI data connections eventually.

DTP protocol was implemented on the disk server of the prototype disk array storage system and on the client machines. Files can be transferred between the disk array storage system and the Connection Machine (CM-2) Data Vault at 21 megabytes per second (limited by the speed of the DataVault), the CRAY Y-MP disk at 16 megabytes per second (limited by the speed of the CRAY disk), and the IBM 3090 expanded memory at 40 megabytes per second.

To transmit visualization data from the disk array to the HIPPI frame buffer, the workstation issues a command to the disk array to write to the frame buffer. Files are transferred from the disk array to the frame buffer at 60 megabytes per second, which is approaching the maximum transfer rate of the IBM disk array. This drives the frame buffer at 12 frames per second.

At these transfer rates, it is possible to transfer a two-gigabyte visualization file from the CM-2 or CRAY Y-MP to the disk array in less than two minutes and then display the file on the frame buffer in 30 seconds.

Conclusions

The HPDS is aimed specifically at meeting the requirements of a high-performance computing environment of massively parallel machines, large-memory supercomputers, cooperative processing networks, high-performance visualization systems, and high-speed networks. With the current availability of networking components and disk array devices that operate at 100 megabytes per second and the expected availability of high-speed, large capacity tape devices, it is now feasible to implement a HPDS for production use.

The implementation of a prototype disk array storage system has demonstrated that workstations can be used to control the high-speed transmission of data over a HIPPI network between client machines and HIPPI-attached storage devices.

N93-30479

OPTIMIZING DIGITAL 8MM DRIVE PERFORMANCE

Gerry Schadegg, Exabyte Corporation
1685 38th Street, Boulder, CO 80301

S30-35
159120
P. 12

Overview

The experience of attaching over 350,000 digital 8mm drives to 85-plus system platforms has uncovered many factors which can reduce cartridge capacity or drive throughput, reduce reliability, affect cartridge archivability and actually shorten drive life. Some are unique to an installation. Others result from how the system is set up to talk to the drive. Many stem from how applications use the drive, the work load that's present, the kind of media used and, very important, the kind of cleaning program in place.

Digital 8mm drives record data at densities that rival those of disk technology. Even with technology this advanced, they are extremely robust and, given proper usage, care and media, should reward the user with a long productive life. The 8mm drive will give its best performance using high-quality "data grade" media. Even though it costs more, good "data grade" media can sustain the reliability and rigorous needs of a data storage environment and, with proper care, give users an archival life of 30 years or more.

Various factors, taken individually, may not necessarily produce performance or reliability problems. Taken in combination, their effects can compound, resulting in rapid reductions in a drive's serviceable life, cartridge capacity or drive performance. The key to managing media is determining the importance one places upon their recorded data and, subsequently, setting media usage guidelines that can deliver data reliability. This paper explores various options one can implement to optimize digital 8mm drive performance.

A Digital 8mm End User Perspective

We generally can classify a majority of user problems to just one of two areas — either *reliability* or *performance*.

The first, reliability, relates to the mechanical failure rate of a particular tape drive. It may surprise some people but a significant majority of the drives received in repair are not really broken. They are suffering from a lack of proper care and/or poor media management. The good news is that these types of failures can be reduced. Also, with proper evaluation and tracking, system integrators can plan adequate service loads, costs and charges.

From a performance standpoint, users either have drives that are not running at transfer speed or writing as much data per cartridge as expected. Here too, both can be addressed and drive performance optimized.

Reliability

Typical of the industry, a reliability specification for computer hardware is based on a statistically distributed mean-time-between-failure (MTBF). Half the product population is expected to exceed the design specification while the other half will not. The entire population is normally represented by a bell curve. Some small percentage is expected to fail very early whereas an almost equal number will work forever. It's simply a rule of statistics.

All equipment manufacturers measure product reliability by MTBF -- the average time between hardware failures that require some form of repair. When a product is new, the useful lives of each of its components, both electrical and mechanical, are added by formula to reach a design goal MTBF a number that usually becomes the product specification. As the product is improved, this number increases. Early 8mm drives were shipped with only a 20,000-hour MTBF and current drives are shipping with a 40,000-hour MTBF.

In the tape industry, MTBF is expressed as total power-on hours of operation. Total power-on hours (POH) can be calculated as follows: a drive powered on 24 hours per day (7 days per week) is powered on 720 hours each month (24 hours times 30 days). Population trend MTBF is calculated as follows:

$$\frac{\text{total population} * \text{POH/mo} * \text{n months}}{\text{total returns for n months}} = \text{MTBF}$$

But, why do you need to understand MTBF? Population MTBF based on returns can be used to confirm whether or not a product is meeting its design specification, living up to the user's expectations as defined by the manufacturer.

Population MTBF is calculated as follows:

Total 8mm drives shipped	287,903 units	100 percent
Active U.S. population	183,980 units	64 percent
Total U.S. depot returns	2,467 per month	7,401 per quarter
Return rate per month	2,467 / 183,980	= 1.34 percent
Assumed POH/mo	600	
MTBF	183,980 * 600 / 2,467	= 44,846 hours

Given our in-house repair activity, we track our population MTBF on a quarterly basis (see chart). Results have shown that we have exceeded 40,000 hours since the beginning of 1990 — a respectable track record for tape technology.

These numbers are affected by duty cycle because, although powered on, a tape drive is rarely in motion 100 percent of the time — reading and/or writing to tape. Duty cycle is the percentage of time that the drive is in mechanical motion.

Because individual applications vary widely, a typical application is assumed for specification purposes. Based on customer input regarding average application use in a cross-section of each customer's user-base environments, Exabyte was able to define a standard application as 600 power-on hours per month (24 hours per day, 20 days per month) with an accumulated 60 tape-motion hours per month (a 10-percent duty cycle).

What does this mean to the user? A user can roughly estimate how many units will be returned for service if the drives are performing as designed (their specified MTBF).

$$\frac{\text{avg. POH/mo/unit}}{\text{MTBF}} \times 100 = \text{percent failures per unit/mo}$$

Planning must take duty cycle into account. A drive's life is impacted by the percentage of its power-on time that it's actually in motion reading and/or writing. If a drive is powered on 24 hours a day but only reads and writes 2.4 hours per day, its duty cycle is 10 percent (24 hours divided by 2.4 hours) the standard application. But if it's reading and writing 4.8 hours per day, it's operating at twice as many hours (24 hours divided by 4.8 hours) or a 20-percent duty cycle. Most likely it will need repair within half the amount of time.

$$\frac{\text{avg. no. POH/mo/unit}}{\text{avg. no. of tape-motion hours/mo/unit}} = \text{avg. percent duty cycle}$$

The "8mm Drive Return Rate" chart shows the average percentage of tape drives that may require repair on a monthly basis given various duty cycles and an MTBF of 40,000 hours.

Performance

There are a variety of application factors that impact an 8mm drive's operation. To some extent, some of these factors are unavoidable. The goal or objective is to minimize their impact. In brief, the factors that most impact performance are the application, the type of media being used, how the media's being used, the operating environment and whether or not the drive is being kept clean. Changing some factors will affect changes in both performance and reliability.

Application and System Factors

It's fairly obvious that, to improve performance and reliability, one of the first areas to investigate is unnecessary duty cycle. If the 8mm drive can sustain a 500 kilobyte-per-second data transfer rate and, as such, take about 33 minutes to store 1,000 megabytes of data, it's streaming continuously and operating at peak throughput. If however the drive takes over an hour to perform the same job, it's taking twice as long and has doubled the duty cycle — maybe unnecessarily.

The first place to investigate is whether the system is maintaining sufficient data flow to the drive. For example, if the drive is attached to a local area network, is the drive being utilized during those time periods when network load is normally reduced? Keeping track of how much time it takes for the drive to operate on a test case data file is an excellent indication of network load impact.

Is the tape drive sharing the bus with very busy disk drives? Heavy disk demand for bus access can dramatically affect the host's ability to keep the tape drive streaming.

Is the system transmitting really small blocks of data? If it is, just the amount of interface overhead itself is going to reduce streaming performance. It would be like depositing $100, one penny at a time. For the 2.5-gigabyte 8mm drive, the EXB-8200, block sizes smaller than or not multiples of 1 K have an impact on performance and capacity. This is not as significant a problem for the 5-gigabyte drive, the EXB-8500, which is capable of packing variable block sizes. The rule of thumb is to select the largest practical block size to improve bus utilization and drive performance.

Was the system software driver originally designed for a start/stop tape device? If it was, it may not be optimized to stream the multiple-gigabyte 8mm drives and may be forcing the drive into excessive start/stop motion at an expense to throughput.

There are many legitimate reasons for adapting existing system software tape drivers to an 8mm device. Time to market, installed base, service systems, training and a host of other reasons come into play. However, doing so may result in real inefficiencies in drive performance and reduced media reliability.

For example, a software driver for a small-capacity serpentine-type might return to the beginning of tape to update a directory after each file or file sub-directory has been written. This would cause the 8mm drive to shuttle hundreds of passes up and down the tape due to its very large storage capacity and the requirement of full serial access along the tape for each update. This activity adds unnecessary passes to the tape and more head/tape contact time than would be required in streaming mode. Only a small percentage of time is spent in useful data transfer. From the user's perspective, this could appear as an apparent early tape life failure. In addition, the drive's read/write head life may be reduced to less than it otherwise should be.

Adapted drivers should be evaluated for long-term viability. Where appropriate, drivers should be modified to make better use of available 8mm capabilities. In cases like this example, simply keeping the tape directory on the system until the tape data transfer operation is complete can result in a marked improvement in media and drive life. Total application run time will also be reduced. The opportunity to use high-speed search features may also be lost.

Media

Use of the right kind of media and proper media care can affect 8mm drive performance just as much as other factors. Some media types also affect data reliability and cartridge capacity.

Several grades of 8mm media are being supplied to the data processing industry by media vendors. Their one point of commonality stems from the fact that the magnetic tape is eight millimeters wide. Beyond that, the media can vary widely in formulation (their composition and structure), film thickness (the media substrate), length of media per cartridge (expressed as meters or minutes) and, lastly, the physical construction of the cartridge (material, how it's made, its resistance to contaminants and differences in recognition hole size and location). The bottom line is that video grade (generic) media is optimized for video recording purposes and data grade media is optimized for data processing.

The digital 8mm drive can read and write most generic 8mm metal-particle tapes although using low-quality media may cause a loss in tape capacity, data throughput, long-term archivability and data integrity. The drive was not designed to write or read some of the newer video tapes.

Loss in capacity and data throughput are caused by a high rate of dropouts in poor-quality media. While recording data, the 8mm drive performs an immediate read-after-write data verification and rewrites every block of unreadable data, making sure that all data is correctly written and readable somewhere on the tape. Of course, this process degrades performance throughput and eats up capacity when the drive encounters a significant number of media errors. The drive does have a cut-off point where, after too many rewrites, it will return an uncorrectable media error and terminate the recording process. This is done for user protection. If the media cartridge is bad enough to warrant this type of termination, users should not use it to store data.

Recent video introduction include tapes whose lengths exceed the industry standard 112 meters. Their capacity is expressed as time in a variety of lengths including 135, 140 and 150 minutes. Users will not gain capacity as a result of the additional length as digital 8mm drives

assume that the tape is 112 meters which is the maximum. Furthermore, these tapes can have cartridge recognition hole patterns which the digital 8mm drives may or may not recognize.

Hi-8 media formulations are available as enhanced metal particle (hi-8MP), metal evaporated (Hi-8ME) and barium ferrite (BaFe). These were developed for video applications and do not yet lend themselves well for use in 8mm data storage devices. Although they can be used in the digital 8mm drive, they most likely will produce higher error rates because they produce magnetic signal amplitudes that are different from standard metal particle.

Exabyte has a data grade media designed specifically to lessen the degradation of the magnetic qualities (i.e., metal particles) of the tape after prolonged storage. Newly developed "powders" encapsulate and protect the metal particles used in the magnetic coating and slow degradation. This results in a uniform recording surface which helps ensure dependable recording and preservation of the stored data along with extending the tape's archival/shelf life. According to accelerated test data, the data grade tape's archival shelf life is estimated to exceed 30 years when stored under recommended environmental conditions.

The improved formulation also has a new binder and lubricant which house the metal particles, greatly improving the durability and, in turn, the reliability of the recording process. Tests measuring dwell performance and repeated passes in streaming mode indicate that the improved data grade media can withstand up to 1,500 passes under recommended environmental conditions.

A pass occurs when *any given section of tape passes through the tape path under tension.* A back-space operation, a read, a write and a forward-space operation all constitute a *pass* on that section of tape. For example, a start/stop operation involves three passes. The tape comes to a complete stop when data is discontinued (1st pass); because this stopping point is beyond the point where data was discontinued, the drive must reverse and back up to the point where data stopped (2nd pass); and finally, the drive proceeds to write when data becomes available (3rd pass). Applications that require multiple searches to the same or very nearby locations (such as directory or label areas) quickly accumulate passes in a localized area.

A newly developed backcoating helps prevent frictional changes associated with repeated usage by protecting the tape. It maintains stable performance even when the tape is operated in complex start/stop motions. The combined backcoating and improved media formulation result in a tape surface which improves head performance of 8mm data storage subsystems.

Digital 8mm drives record data at densities that rival those of disk technology. Even with technology this advanced, the drives are extremely robust and, given proper usage, care and media, will reward the user with a long productive life. To reiterate, the drives perform best with a high-quality "data grade" media which can prolong drive head life; provide up to a 30-year archival life; be used for up to 1,500 passes; and deliver a cartridge shell specifically designed for data processing which offers as many as 10,000 lid opening and closings.

Media Care

A major problem here is *over use* of tapes. If an end user is not familiar with the detailed motion characteristics of an application, excessive passes can accumulate. Mechanical loader and library-type applications are particularly prone to this problem. When tapes begin to break down from overuse, they begin to generate tape debris. This causes unnecessary wear and data integrity problems which , in turn, lead to degraded read/write performance because the drive is forced to perform excessive error recovery routines.

Media acclimation is also important. Before using an 8mm data cartridge, allow it to acclimate to the operating environment for twenty-four (24) hours, or for the amount of time it has been exposed to dissimilar conditions — whichever is less.

Proper storage is a must. Always store the 8mm tapes on edge. *Do not stack flat.* Constant environment control is more important than absolute temperature values, so *keep the environment constant.* The closer it is kept to *ideal* , the better data recovery results will be. It's also a good idea to keep a storage log on all tapes, locations, contents and history.

Tapes should be exercised on an average of once every twelve (12) months by running them from beginning to end and back to the beginning at normal speed (not rewind or high speed). This operation is best performed by reading to end-of-tape and rewinding at normal speed. It will remove any stress which can build up in the tape pack during the storage interval. Tapes stored at higher temperatures should be exercised more frequently.

Data Archival

When data is to be archived for extended periods of time (several years), optional steps can be taken that further insure data integrity and recovery over and beyond digital 8mm's normal and extensive built-in data recovery margins. The tape unit used for recording archival data should be exceptionally clean. It is also suggested that brand new cartridges be exercised from end-to-end up to four times at normal (not high) speed. This proving process will remove any potential debris that could have been generated during the tape manufacturing process. This last step is especially critical whenever the tape being used is not of high-quality, recommended "data grade" material.

Archival data should always be recorded using the read-after-write check and rewrite features of the 8mm drive. When recording an archival tape, the tape should be completely recorded from end to end *without stopping.* This means that the system must be set up to constantly stream data to the tape drive. When the end of tape is reached, rewind it at normal speed, not at high speed. To store, clearly label each cartridge. Include all pertinent information such as the model and serial number of the recording tape unit, the date, the density, any error statistics and a log number. Always store the tape data cartridge in an 8mm cartridge storage container. An excellent practice is to further seal the tape cartridge container in a polyethylene bag for long-term storage.

Environmental Factors

If high humidity exists (greater than 45 percent), increased tape coating abrasivity occurs which causes increased tape drive head wear. The same applies to tape wear although to a lesser degree. If low humidity exists, the combination of friction, low humidity and organic material (contained in the magnetic coating) can cause the formation of what is called *friction polymers.* These brown or bluish stain deposits appear on the head surface. They are very hard and, as they build up, increase the effective distance between tape and head, reducing signal strength. A desirable range for humidity is 35 to 45 percent.

As the temperature increases, the maximum allowable relative humidity to optimize drive performance decreases. For instance, with an increase from 22 degrees Celsius (72° F) to 32 degrees Celsius (90° F), the *maximum* allowed relative humidity *decreases* from 80 percent to approximately 50 percent. Thus, temperature has an indirect effect on performance by crowding the humidity limits as temperature increases. It is an important factor to be considered when integrating (or operating) drives into a system configuration.

Airflow and Location

While there are many requirements for successful integration of an 8mm drive into a system, proper airflow and cooling are key to maximizing drive and media life. High temperatures reduce the humidity tolerance of tapes and can cause higher drive failure rates. To maximize drive life, a minimum temperature rise over ambient is desirable. This is achieved by drawing sufficient airflow through the drive in order to maintain tape path temperature at or near room temperature. Too much airflow can cause excessive amounts of dirt and dust to be ingested by the drive leading to performance problems. When a balanced airflow exits, particulate contamination is generally not a concern in the average office environment.

In addition to balancing airflow, consideration should be given to mounting location. The preference is to keep the drive away from a floor level or other areas where dirt can collect. Conversely, locating a drive on or in the top of a systems cabinet may expose it to elevated temperatures.

Cleanliness

Running drives without cleaning will result in media deposit build-up in the tape path and on the heads which, in turn, will increase error rates and ultimately result in drive failure. The rate of deposit build up varies widely and is dependent on tape quality, tape usage (new, worn or ready to be retired), tape motion (streaming versus start/stop), tape path condition (aligned, clean, new or used), and the effects of temperature and humidity on the media.

Operation without cleaning will eventually result in a significant accumulation of deposits that can become burnished into the tape path, resulting in drive failures and/or unacceptable drive performance. When this happens, single or even multiple cleaning passes of Exabyte cleaning tape will NOT remove the deposits. Factory cleaning is necessary. Drives have been returned to the factory with suspected early life head failures, only to find that excessive deposits were the cause of failure.

Regular cleaning with an Exabyte Cleaning Cartridge will prevent this deposit build up, as well as, maintain the tape path and read/write heads in a clean condition. Based on test findings and allowing for the wide variety of applications and media, a preventive maintenance specification was established for 8mm data recording drives. It stipulates that one cleaning pass be performed at least once per month or after the transfer of approximately 30 hours of tape motion. In any worse-than-normal environment, cleaning should be more frequent. This specification has been validated by testing and, where properly applied, has proven successful throughout the field population.

Abrasive cleaning tapes can destroy heads. Use of other than Exabyte-approved cleaning tapes can result in much-degraded head life. Instances have occurred in which all useful head life has been removed from drives in as few as five cleaning passes. Also, some types of abrasive tapes, that are typical of video cleaning cartridges, can deposit material on the read/write heads, rendering them immediately useless. To avoid this problem, Exabyte Cleaning Cartridges should be made readily available to end users, along with adequate training, strong warnings and counseling. **Use of unauthorized cleaning procedures can void warranty.**

When applications do not monitor drive usage and prompt users for needed cleaning activity, users can estimate the amounts of data transferred over operating periods and establish regular cleaning intervals as appropriate. Cleaning frequency could be based on number of tapes processed, number of jobs run, shifts, hours, days or weeks, etc., up to a maximum interval of once per month.

Check List

Exabyte has developed an "Integration Check List" for its 8mm tape drives. It may help predict the potential for drive failures that can be caused by integration and application characteristics. All of the factors in the entire list should be evaluated for each application because, for the most part, a single questionable variable may have no adverse affect on performance and reliability. However, multiple questionable variables can have a compounded detrimental effect. When unacceptable conditions exist, serious reliability and performance problems are likely to occur. Guidelines are as follows: **U = Unacceptable, ? = Questionable, O = Optimal Condition**.

Integration Factors

Tape Path Temperature	> 4 C Rise	U
	< 1 C Rise	?
	1 to 4 C Rise	O
Tape Drive Location	Contamination High (Near Floor or Dirt)	?
	Away from Floor	O

Application and System Factors

Average Transfer Rate (2.5 GB drive)	<= 123 KB/s	?,
	> 123 KB/s but < 246 KB/s	?
	= 246 KB/s (streaming)	O
Average Transfer Rate (5 GB drive)	<= 250 KB/s	??
	> 250 KB/s but < 500 KB/s	?
	= 500 KB/s (streaming)	O
Average Block Size (2.5 GB drive)	< 1 K	U
	1 K or multiples of 1 K	O
Average Block Size (5 GB drive)	< 1 K	?
	1e	
Number of Blocks Transmitted (per SCSI Command)	Low	?
	High	O
Number of Tape Passes per Use (representative of start/stop operations)	Unnecessary repetitious positioning	U
	>6	?
	<= 6	O
Directory/Label Updating	Frequent	U
	Once per session	O
Monitor and/or Prompt for Cleaning Interval (on operator console)	None	?
	Yes	O
Monitor/Prompt for Number of Tape Passes (on operator console)	No	?
	Yes	O
Monitor/Prompt for Soft Error Rates (to prompt media replacement decision)	No	?
	Yes	O

Media

Media	Use "generic" or "video grade"	??
	Use "Data Grade"	?
	Use EXATAPE	O
Require Cleaning as a Condition of Warranty	No	U
	Yes	O
Use Approved Cleaning Cartridges	No	?
	Yes	O

Suppc · and Training

Provide Media Usage and Handling Guidance	No	?
	Yes	O
Provide Cleaning Practices Guidance, Documentation and Training	No	?
	Yes	O
Provide Clear, Strong Warn ing against Using Alternate Cleaning Tapes	No	U
	Yes	O

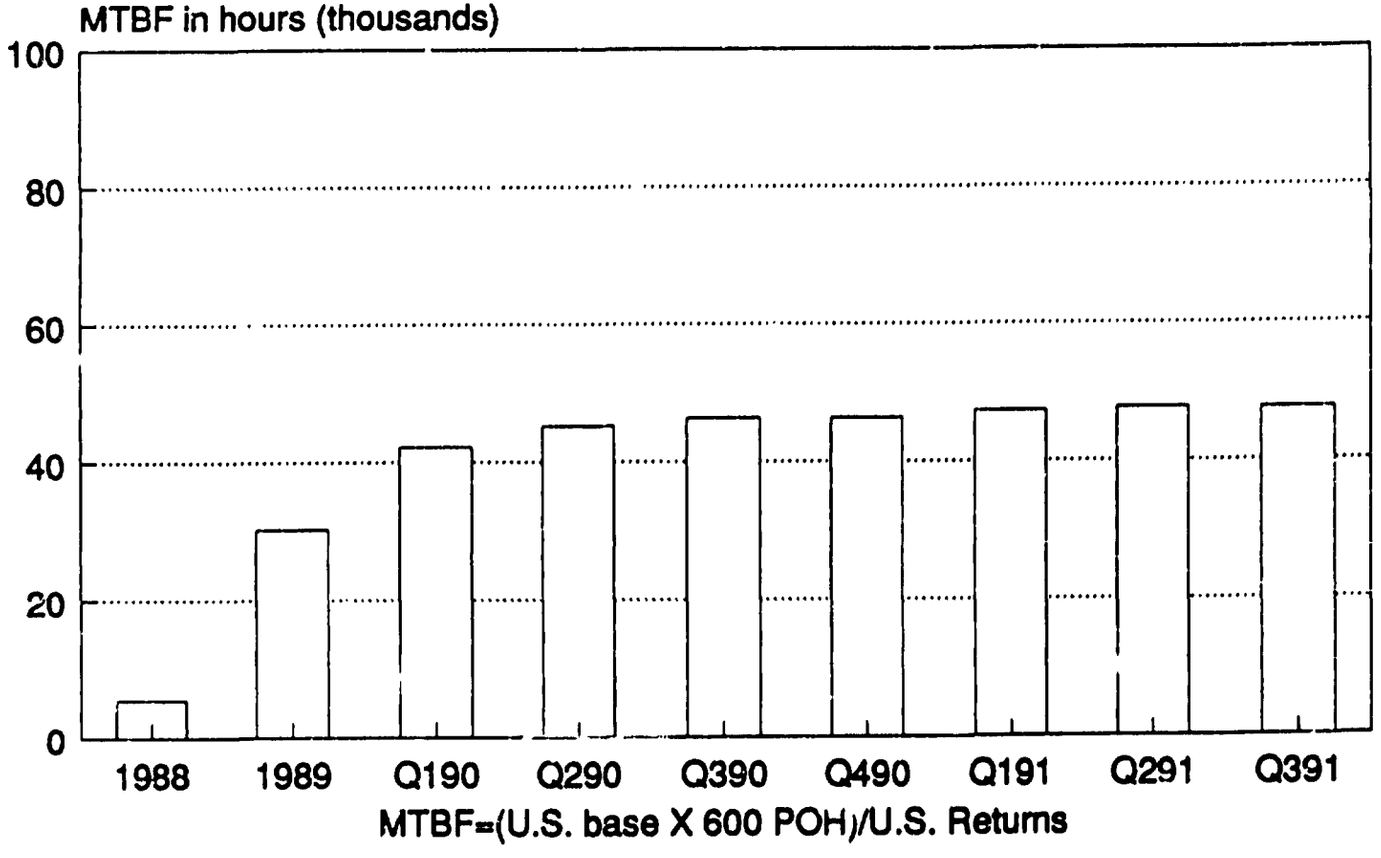
Installation Specific

Temperature (Long Term Avg)	<5 C or > 40 C	U
	5 C to < 16 C or > 24 C to 40 C	?
	16 to 24 C	O
Humidity Non-Condensing (Long Term Avg)	< 20% or > 80%	U
	20% to < 30% or > 45% to 80%	?
	30% to 45%	O
Air Conditioning Absent or Set Back at Night during Drive Operation	Yes	?
	No	O
Operation in Excessively Contaminated Environment	Yes	U
	No	O
Recommended Cleaning with Approved Cleaning Cartridge	No	J
	Yes	O
Use Cleaning Cartridge beyond Specified Useful Life	Yes	U
	No	O
Tape Acclimatization	No	U
	Yes	O
Onsite and offsite media storage	Uncontrolled or unknown	?
	Controlled environment within storage spec	O
Duty Cycle (approximate tape motion hours per day)	> 3	?
	<= 3	O

Tape Mounted in Powered-On Drive	Entire day	?
(exposure to contaminants)	Only during operation	O
Reliable Service Practices		
Track Units	No	?
(by serial number)	Yes	O
Track Failures	No	U
(by type, customer and serial number)	Yes	O
Track Repair Reports	No	?
(by serial number)	Yes	O
Supply All Information to Service Provider	No	?
(dumps, tapes, units)	Yes	O
Centralized Support Organization	No	?
	Yes	O

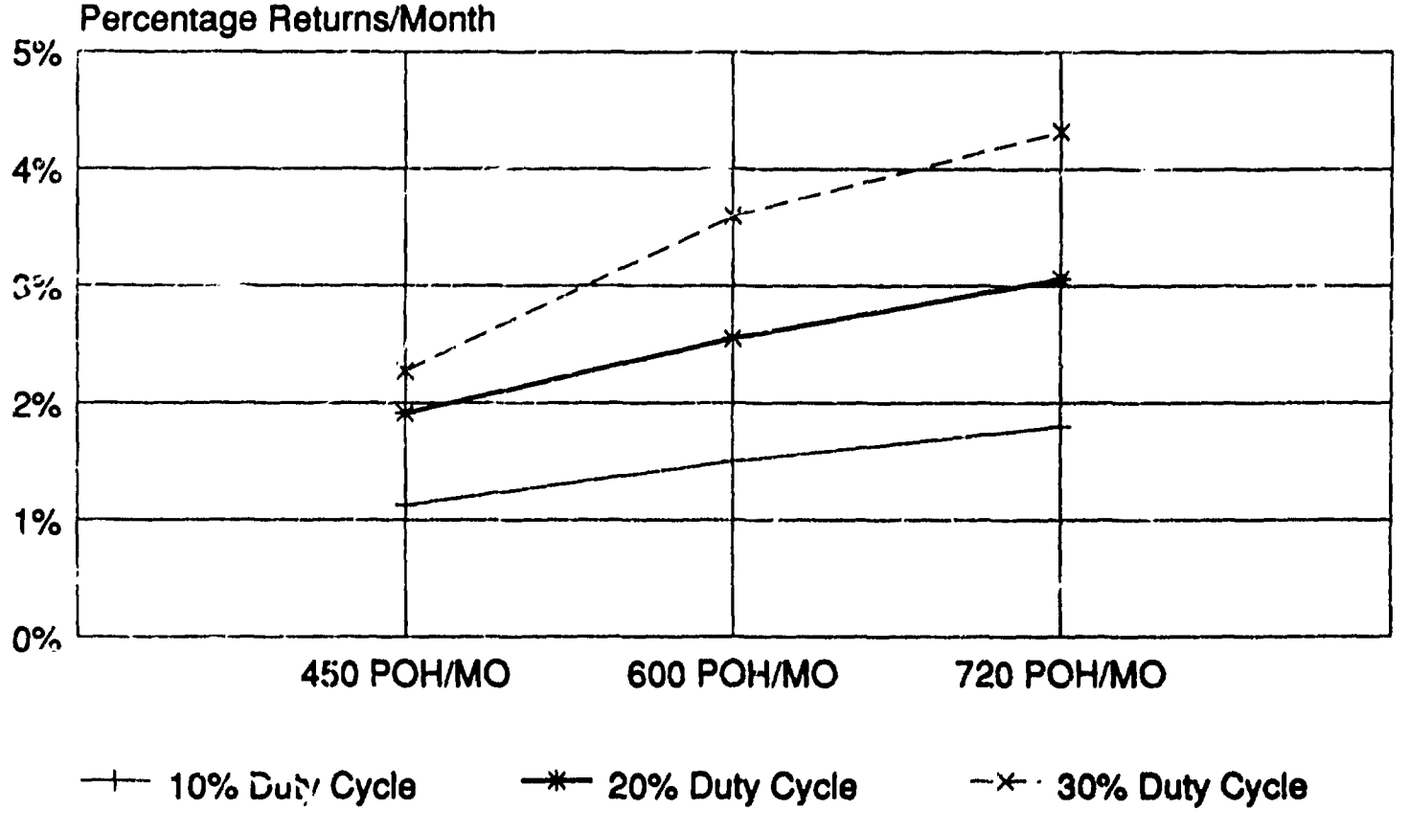
This may be an expansive list but it is the simplest way to summarize all of the factors that can affect digital 8mm drive performance. As you may have noticed, performance and reliability are closely tied together. Digital 8mm drives are extremely robust and, by having these various factors optimized, will reward the user with a long productive life. Furthermore, high-quality 8mm "data grade" media, even though it costs more, can sustain the reliability and rigorous needs of a data storage environment and, with proper care, give users an archival life of 30 years or more.

2.5GB 8mm Drive MTBF
Field Ongoing Reliability



MTBF based on U.S. Population

8mm Drive Return Rate
40,000-Hour MTBF



1/1/92

Using Transparent Informed Prefetching (TIP) to Reduce File Read Latency

R.H. Patterson, G.A. Gibson, M. Satyanarayanan
Carnegie Mellon University

Outline

I/O performance is lagging

No current solution fully addresses read latency

TIP to reduce latency

- **exploits high-level hints that don't violate modularity**
- **converts throughput to latency**

Preliminary TIP test results

As processor performance gains continue to outstrip Input/Output gains, I/O performance is becoming critical to overall system performance. File read latency is the most significant bottleneck for high performance I/O. Other aspects of I/O performance benefit from recent advances in disk bandwidth and throughput resulting from disk arrays [Patterson88], and in write performance derived from buffered write-behind and the Log-structured File System [Rosenblum91]. The access gap problem limiting improvements in read latency is exacerbated by distributed file systems operating over networks with diverse bandwidth [Spector89, Satyanarayanan85]. In this paper, we focus on extending the power of caching and prefetching to reduce file read latencies by exploiting hints from high-levels of a system. We describe such Transparent Informed Prefetching, TIP, and its benefits. We argue that hints that disclose high level knowledge are a means for transferring optimization information across, without violating, module boundaries. We discuss how TIP can be used to convert the high throughput of new technologies such as disk arrays and log-structured file systems into low latency for applications. Our preliminary experiments show reductions in wall-clock execution time of 13% and 20% for a multiple module compilation tool (make) accessing data on a local disk and remote Coda file server, respectively, and a reduction of 30% for a text search (grep) remotely accessing many small files.

Solutions to I/O Bottleneck

	Latency	Throughput
Read	demand caching prefetching	disk arrays
Write	buffered writes	disk arrays buffered writes LFS

But, cache effectiveness is declining

This table shows the mechanisms most heavily used to combat the growing I/O bottleneck. Written data benefits from write-behind buffering and log-structured files systems, while I/O throughput is directly increased by parallelism in disk arrays. Read latency, however, is only reduced by caching and prefetching. As will be shown next, caches will not, by themselves, be able to relieve the I/O bottleneck, and prefetching will emerge as a critical approach to the problem.

Effective I/O Performance with Caching

$$T_{I/O} = MC_M + (1-M)\,C_H \approx MC_M$$

$$T_E = T_C + N_A T_{I/O} \approx T_C + N_A MC_M$$

$T_{I/O}$ = I/O time
M = cache miss ratio
C_M = cost of a miss
C_H = cost of a hit
T_E = execution time
T_C = computation time
N_A = number of I/Os

Miss ratio for effective I/O performance to scale with CPU performance

CPU/IO Perf.	Current=1	10	100
Miss Ratio	40%	4%	0.4%

Caches reduce the average I/O service time by reducing number of I/O requests that must be serviced by slow peripheral devices. The ratio of requests thus serviced to the total number of requests is the miss ratio. For caches to compensate for the growing gap between CPU performance and I/O peripheral performance, they must reduce their miss ratios. This simple model quantifies this relationship.

The average I/O service time, $T_{I/O}$, is the weighted sum of the service times for requests that miss in the cache and must be serviced by the I/O subsystem, C_M, and for requests that hit in the cache, C_H. The cache miss ratio, M, weights the sum. Since $C_H << C_M$, the average I/O service time is roughly MC_M. The execution time for a program, T_E, is the sum of the time spent on computation, T_C, and the total time spent on I/O. Time spent on I/O is, in turn, the product of the number of I/O requests, N_A, and the average time to service a request. As processor improvements reduce T_C relative to C_M, the miss ratio, M, must be reduced to achieve corresponding reductions in the time spent on I/O. The table shows the improvement needed in the cache miss ratio for the effective I/O performance to keep pace with processor gains. A cache that currently has a 40% miss ratio must improve to 4% to match a ten-fold increase in processor performance and to 0.4% to match the 100 fold increase expected in the next ten to fifteen years. As the next slide shows, such miss ratios are most unlikely.

Cache Miss Ratios

	1985 BSD Study				1991 Study
Cache Size	390KB	4MB	8MB	16MB	7MB (avg)
Miss Ratio	49.2%	28.0%	26.2%	25.0%	41.4%

- **Diminishing returns from larger caches**
- **Disappointing performance over time**
  - **> growing file sizes**

Clearly, caching alone cannot provide the needed performance improvements

The numbers in this table are drawn from [Ousterhout85] and [Baker91]. The 1985 tracing study of the UNIX 4.2 BSD file system predicted cache performance for a range of cache sizes assuming a 30 second flush back policy for writes. The 1991 study measured cache performance on a number workstations running Sprite. The Sprite cache size varied dynamically, but averaged 7MBytes. The diminishing returns from increasing cache size are evident in the 1985 results. Also striking is the difference between the predicted and measured performance of a large cache. The large cache was not nearly as effective as expected. The authors of the study concluded that growing file sizes were to blame for the disappointing cache performance. This result is strong evidence that we cannot rely on increased cache sizes to give us the extremely low miss ratios needed to improve effective I/O performance. This leaves us with prefetching as a tool for improving I/O read latency.

Transparent Informed Prefetching (TIP)

1) **Encapsulate programmer knowledge about future I/O requests in a hint**

2) **Transfer hint to file system**

3) **File system uses hints to transparently prefetch data and manage resources**

Prefetching can pre-load the cache to reduce the cache miss ratio, or, at least reduce the cost of a cache miss by starting the I/O early and thereby improve effective I/O performance. While there have been a number of approaches to prefetching [Kotz91, Smith85, McKusick84, Feiertag7], it is often difficult to know what to prefetch, and prefetching incorrectly can end up hurting performance [Smith85].

To be most successful, prefetching should be based on *knowledge* of future I/O accesses, not inferences. We claim that such knowledge is often available at high levels of the system. Programmers could give hints about their programs' accesses to the file system. Thus informed, the file system could transparently prefetch needed data and optimize resource utilization. We call this Transparent Informed Prefetching (TIP).

Obtaining Hints

Early knowledge of serial file access

Access patterns part of code algorithm

- **large matrix supercomputing: read by row, read by column**

Hints generated by: programmer, compiler, profiler

Critical to the success of informed prefetching is the availability of accurate and timely hints. An important part of our research will be to expose such hints in important, I/O-dependent applications. However, we don't think this will be as hard as it might seem. After all, the success of sequential readahead is largely the product of "discovering" that an application is sequentially accessing its files; this is really known a priori because a programmer has chosen to do so. Often, it is known well in advance that many files will be thus accessed. It is a simple step to have programmers notify the I/O system, through a hint, of sequential access patterns.

In addition to the simplest hints about sequential accesses, programmers could give hints about more complex, non-sequential access patterns. An important beneficiary of this approach will be the large scientific programs that execute alternating row and column access patterns on huge matrix data files [Miller91]. At least one of these access patterns will not be sequential in the file's linear storage, yet the pattern is easily and obviously specified by a programmer.

In addition to programmer-generated hints, compilers could automatically generate hints, or a profiler could be used to generate hints for future runs of a program.

Application Examples

grep foo *

- **Shell expands '*' to a list of filenames.**
- **Grep searches for a string, 'foo,' in all of the files in the list.**
- **From invocation, it is known that all of the files on the list will be read sequentially.**
- **Give a hint about all of the files at once.**

make

- **makefile specifies all files to be touched from the start**
- **make generates hints for binaries it will invoke and the files they will touch.**

While we believe that scientific applications will be major beneficiaries of TIP, common Unix applications can also benefit. Her are two examples.

Given the command 'grep foo *,' the shell expands the '*' into a list of all files in the current directory and invokes the 'grep' program which searches for the string 'foo' in all the files. Grep, or even the shell if it knows a little about grep from a command registry, can issue a hint notifying a TIP system that all the files in the list will soon be read. If the system has stored these files on an underutilized disk array, many or all will be fetched concurrently.

We expect programs issuing hints on behalf of other programs, such as the shell on behalf of grep, to be a common occurrence. Another example is the 'make' program which orchestrates the compilation of program modules and their linking with standard libraries. 'Make' determines its actions according to a 'makefile' of instructions. After parsing a 'makefile' and checking the status of all modules to be built, 'make' constructs a set of command sequences that it will pass to a shell for execution. These commands or the shell itself can issue hints about their I/O accesses. Pursuing a TIP approach more aggressively, 'make' can use the same command registry as the shell to issue hints even before it issues the commands.

TIP Converts High Throughput to Low Latency

Use excess storage bandwidth to pre-load caches with future accesses and overlap I/O with computation

Expose concurrency to pack low-priority queue with prefetch requests

- **Optimize seek scheduling**
- **High-throughput disk arrays simultaneously service multiple requests**
- **Multiple network requests may be batched together**

Cache management superior to LRU

Armed with knowledge of future file accesses, a system employing TIP can improve performance in three important ways.

1) At the most basic level, TIP, as for all prefetching, can overlap slow I/O accesses with other useful work so that applications spend less time idly waiting for these accesses to complete. But, because TIP systems know what to prefetch, they can prefetch more aggressively to pre-load the cache with future accesses and further reduce cache misses.

2) Using TIP, normally short I/O queues can be filled with low-priority prefetch requests giving more opportunities for low-level I/O optimizations. For an individual disk, deeper queues allow better arm and rotation scheduling [Seltzer90]. For a disk array, deeper queues mean more requests are available for concurrent servicing by independent disks. On a network, prefetch requests can be batched together, reducing network and protocol processing overhead.

3) TIP improves cache management to further reduce cache miss ratios. If it is known what data will be needed in the future, it may be possible to outperform an LRU page replacement algorithm, even without prefetching. Unneeded blocks can be released early, and needed blocks can be held longer.

The first two benefits make TIP an excellent mechanism for exploiting the high throughput of emerging storage technology to provide the low latency that these technologies cannot provide. Combined with improved cache management, these three benefits make TIP a powerful tool for overcoming the widening access gap.

Hints are Disclosure not Advice

Hints that disclose	Hints that advise
I will read file F sequentially with stride S I will read these 50 files serially & sequentially	cache file F reserve B buffers & do not read-ahead

- **Users not qualified to give advice**
- **Advice not portable, disclosure is**
- **Disclosure allows more flexibility**
- **Disclosure supports global optimizations**
- **Disclosure hints consistent with sound SWE principles**

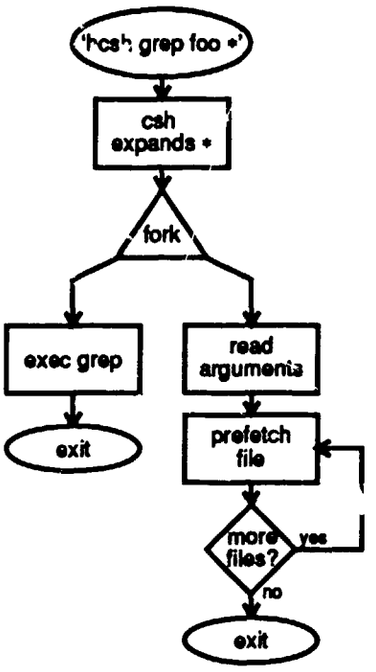
As the previous slide showed, TIP is much more than simple prefetching; it is a strategy for optimizing I/O. For a number of reasons, such powerful optimizations depend on having hints that disclose knowledge of future I/O operations instead of hints that give advice about I/O subsystem operation.

Advice about low-level operations depends on detailed system-specific knowledge. Even if a user had such knowledge of a system's static configuration, they could not know about the system's dynamic state. Thus, the user is not qualified to give advice on how to optimize the dynamic operation of the system. Furthermore, such system-specific knowledge would not apply to other systems, and so, advice that exploits it would not be portable to other systems.

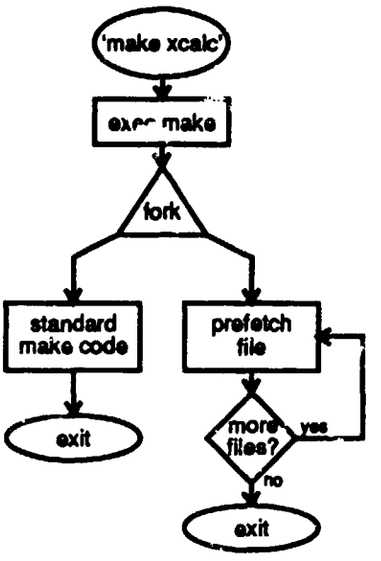
Additionally, hints that advise, such as, 'cache this file,' do not give much usable knowledge to the TIP system. What should the TIP system do if it cannot cache the whole thing? Should it cache a part of the file? Which part? If, instead, the application discloses how it will access the file, the TIP system has the flexibility to respond appropriately. This flexibility is crucial for balancing competing demands for global resources.

Good hints that disclose are specified using the same semantics that an application later uses to demand access to its files, whereas bad hints which advise concern themselves with a system's implementation. It is not a coincidence that good hints are compatible with modular software design. They are a means for transferring optimization information across module boundaries without violating those boundaries.

Preliminary Test



Prefetching for shell expansion.



Prefetching for 'make.'

Our research into a TIP approach began with simple, controlled experiments demonstrating the potential benefits and obstacles of informed prefetching. Our goals with these experiments were to validate TIP as a tool for reducing read latency, determine if more than a simple, user-level mechanism is needed, uncover implementation problems, and develop experience incorporating hints into applications.

We used two hardware platforms for our tests. The local disk tests were conducted on a Sun Sparcstation 2 running Mach 2.5/BSD Unix 4.3. The remote tests were run on two Decstation 5000/200 running Mach 2.5, one of them the client, and the other the server for the Coda distributed file system [Satyanarayanan90].

We tested the two applications previously mentioned, shell expansion of '*' for 'grep,' and 'make' building a program called 'xcalc.' Flow charts for the two test programs are above. The chart on the left shows the configuration for exploiting shell expansion of '*.' A fork operation splits the program into two processes. The command runs down the left side of the fork, while an independent prefetch process runs down the right side of the fork. The prefetch process uses the expanded list of filenames to determine what to prefetch. The right-hand chart shows the configuration for the 'make' example. It is similar to the previous example except that a tracing facility [Mummert92] is used to determine in advance the files to prefetch.

To prefetch from the local disk, the prefetch process simply read the appropriate files, indirectly causing the data to be moved into the cache. To prefetch remotely from Coda, the prefetch process used a special prefetch ioctl to explicitly and asynchronously transfer the file to the local machine.

Test Results

Application	Local Disk				Distributed File System (Coda)			
	hot cache	cold cache	cold cache w/prefetch	% reduction	hot cache	cold cache	cold cache w/prefetch	% reduction
make xcalc	9.17 (0.03)	14.19 (0.13)	12.40 (0.07)	12.6	18.29 (2.00)	40.41 (3.63)	32.20 (2.74)	20.3
grep foobar *	1.22 (<0.01)	3.29 (0.13)	3.30 (0.04)	0	1.85 (0.01)	7.86 (0.77)	5.55 (0.68)	29.4

- **make xcalc: compile & link X window calculator**
- **grep foo *: 58 files, 1 MB**
- **Results limited by lack of parallelism in I/O subsystem**

This table compares the elapsed times to run two applications with and without prefetching on both the local disk and the Coda distributed file system. The first application, 'make xcalc,' compiles and builds the X window calculator tool. The second, 'grep foobar *,' searches 58 files containing a total of 1 MByte all stored in (the cache of) a remote Coda file server.

The numbers in parentheses are the standard deviations for the measurements. Since the local tests were performed on a Sun Sparcstation 2 whereas the Coda tests were performed on Decstation 5000/200, the numbers are not directly comparable. In the 'hot cache' runs, all data read throughout the job were in the local buffer cache, so the job never blocked for the disk. These numbers represent a lower bound on the elapsed time. At the start of the 'cold cache' runs, there was no data in the buffer cache or client disk cache, though, in the distributed case, the server's buffer cache was not cleared between runs. The 'cold cache w/ prefetching' runs were started just like the 'cold cache' runs, but they used prefetching to speed access to the files. The '% reduction' represents the benefits of prefetching.

TIP systems will only be able to approach the lower bound represented by the 'hot cache' numbers when combined with high-throughput I/O subsystems unavailable for these tests. In the grep test on the local disk, the execution time is dominated by I/O. The disk is constantly running flat out, so there is no time for prefetching. Grep with a disk array would still keep one disk busy and would run in about the same amount of time, but grep with TIP and a disk array would keep many disks busy. The total time spent on I/O would drop and performance approaching the 'hot cache' lower bound should be possible.

Lessons from Tests

- **Independent prefetch process overhead too high**
- **Single prefetch process ⇒ no deep prefetch queues**
- **Coda ioctl allowed too much prefetching**
  - **> thread starvation - need low-priority prefetching**
  - **> premature cache flushing - need to track consumption**
- **Poor cache buffer replacement performance**
- **Disk write scheduling often very inefficient**

Although our experiments were preliminary, they served their purpose of demonstrating the benefits of informed prefetching and educating us about implementation pitfalls.

Using independent prefetch processes incurred a lot of extra overhead, especially in the local disk tests. Context switching, process scheduling inefficiencies, system call cost, and, on the local disk, data copy costs all reduced the performance of the prefetch tests. But, the most serious hindrance to prefetching from the local disk was that, because the read system calls used are blocking, there was never more than one prefetch request in the queue at a time. Thus, we did not benefit from the scheduling advantages offered by deeper queues

The coda tests avoided this problem with the asynchronous prefetch ioctl. They suffered instead from over-prefetching. Until we reduced the priority of the prefetches, they interfered with demand fetches, reducing performance. Also, prefetches sometimes got ahead of the actual job and caused prefetched data that had not yet been used to be replaced in the cache by newly prefetched data. Clearly, a real system will need to track data consumption to avoid this problem. This was an extreme example of the cache manager making uninformed decisions. The cache held onto data that had just been used in preference to prefetched data that was about to be used. Integrating TIP with the cache manager should greatly improve performance. In the tests, we avoided this problem by using a very large cache that could all of the data.

Writes of whole blocks were not buffered and thus were interleaved with both prefetch and demand reads which led to very poor disk scheduling. This highlighted the importance of buffered writes.

Summary

TIP uses hints to convert high throughput storage to low latency where caching fails

Hints that disclose, not advise, provide the best information and are consistent with sound SWE principles.

Applicable to local disk and network file servers

Immediate Plans

- **modify Coda/BSD/Mach to accept and exploit correct hints**
- **find & instrument applications**
  - **> make, search, visualization, simulation**

Transparent Informed Prefetching, TIP, extends the power of caching and prefetching to reduce both local and remote file read latency by exploiting application-level knowledge of future access patterns. TIP systems can cooperate with resource management policies to increase the utilization and efficiency of high-throughput network and storage systems. Many future accesses become current accesses that can exploit the parallelism of disk arrays or may batched to reduce network overheads. Disk accesses and buffer allocation may be improved with foreknowledge of future accesses. TIP effectively converts the high throughput of new peripheral technology into low read latency for application programs.

Informed prefetching depends on hints from applications that disclose their future I/O accesses in terms of operations on files. Hints should not give advice about I/O subsystem operation nor be expressed in terms of resource management policy options. This distinction is important for hint portability and consistency with software engineering principles of modularity, and for the TIP system to be able to effectively manage global resources.

Preliminary tests have confirmed the potential benefits of informed prefetching and highlighted some of the potential pitfalls of implementation.

Our next step is to implement TIP in a Coda/BSD/Mach operating system. Then we will identify and instrument applications to provide the required hints to the system.

References

[Baker91] Baker, M.G., Hartman, J.H., Kupfer, M.D., Shirriff, K.W., and Ousterhout, J.K., "Measurements of a Distributed File System," *Proc. of the 13th Symp. on Operating System Principles*, Pacific Grove, CA, October 1991, pp. 198-212.

[Feiertag71] Feiertag, R. J., Organisk, E. I., "The Multics Input/Output System," *Proc. of the 3rd Symp. on Operating System Principles*, 1971, pp 35-41.

[Kotz91] Kotz, D., Ellis, C.S., "Practical Prefetching Techniques for Parallel File Systems," *Proc. First Int'l Conf. on Parallel and Distributed Information Systems*, Miami Beach, Florida, Dec. 4-6, 1991, pp. 182-189.

[McKusick84] McKusick, M. K., Joy, W. J., Leffler, S. J., Fabry, R. S., "A Fast File System for UNIX," *ACM Trans. on Computer Systems*, V 2 (3), August 1984, pp. 181-197.

[Miller91] Miller, E., "Input/Output Behavior of Supercomputing Applications," University of California Technical Report UCB/CSD 91/616, January 1991, Master's Thesis.

[Miller91b] Miller, Ethan, private communication.

[Mummert92] Mummert, L., Satyanarayanan, M., "Efficient and Portable File Reference Tracing in a Distributed Workstation Environment," Carnegie Mellon University, manuscript in preparation.

[Ousterhout85] Ousterhout, J.K., Da Costa [illegible]rrison, D., Kunze, J.A., Kupfer, M., and Thompson, J.G., "A Trace-Dr[illegible]nalysis of the UNIX 4.2 BSD File System," *Proc. of the 10th Symp. on Operating System Principles*, Orcas Island, WA, December 1985, pp. 15-24.

[Patterson88] Patterson, D., Gibson, G., Katz, R., A, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *Proc. of the 1988 ACM Conf. on Management of Data* (SIGMOD), Chicago, IL, June 1988, pp. 109-116.

[Rosenblum91] Rosenblum, M., Ousterhout, J.K., "The Design and Implementation of a Log-Structured File System," *Operating Systems Review (Proceedings of the 13th SOSP)*, Volume 25 (5), October 1991, pp 1-15.

[Satyanarayanan85] Satyanarayanan, M., Howard, J. Nichols, D., Sidebotha[illegible], Spector, A., West, M., "The ITC Distributed File System: Principles and [illegible]sign," *Proceedings of the Tenth Symposium on Operating Systems Principles*, ACM, December 1985, pp. 35-50.

[Satyanarayanan90] Satyanarayanan, M., Kistler, J. J., Kumar, P., Okasaki, M. E., Siegel, E. H., Steere, D. C., "Coda: A Highly Available File System for a Distributed Workstation Environment," *IEEE Transactions on Computers*, V C-39 (4), April 1990.

[Seltzer90] Seltzer, M. I., Chen, P. M., Ousterhout, J. K., "Disk Scheduling Revisited," *Proc. of the Winter 1990 USENIX Technical Conf.*, Washington DC, January 1990.

[Smith85] Smith, A.J., "Disk Cache--Miss Ratio Analysis and Design Considerations," *ACM Trans. on Computer Systems*, V 3 (3), August 1985, pp. 161-203.

[Spector89] Spector, A.Z., Kazar, M.L., "Wide Area File Service and The AFS Experimental System," *Unix Review*, V 7 (3), March 1989.

*U.S. GOVERNMENT PRINTING OFFICE: 1993-728-150/60025

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, September 22-24, 1992

4. TITLE AND SUBTITLE
Goddard Conference on Mass Storage Systems and Technologies
Volume I

5. FUNDING NUMBERS
902

6. AUTHOR(S)
Ben 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
93B00038
Code 902

9. SPONSORING / MONITORING AGGENCY NAME(S) AND ADDRESS (ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA CP-3198, Vol. I

11. SUPPLEMENTARY NOTES
Kobler: Goddard Space Flight Center, Greenbelt, MD;
Hariharan: STX Corporation, Lanham, MD.

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 September 1992. Similar to last year's conference, this year's gathering served as an informational exchange forum for topics primarily relating to the ingestion and management of massive amounts of data and the attendant problems (data ingestion rates now approach the order of terabytes per day). Discussion topics include the IEEE Mass Storage System Reference Model, data archiving standards, high-performance storage devices, magnetic and magneto-optic storage systems, magnetic and optical recording technologies, high-performance helical scan recording systems, and low end helical scan tape drives. Additional discussion topics addressed the evolution of the identifiable unit for processing purposes (file, granule, data set or some similar object) as data ingestion rates increase dramatically, and the present state of the art in mass storage technology.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Magnetic tape, magnetic disk, optical disk, mass storage, software storage	356
	16. PRICE CODE A16

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL