NASA Technical Memorandum 4554

NASA High Performance Computing and Communications Program

Annual Report, 1992

Office of Aeronautics
High Performance Computing and Communications Office
Washington, DC

Lee Holcomb, Director
Paul Smith, Program Manager
Paul Hunter, Program Manager

National Aeronautics and Space Administration
Scientific and Technical Information Program

1993

Introduction to the High Performance Computing and Communications (HPCC) Program

Program Goal and Objectives: The National Aeronautics and Space Administration's HPCC program is part of a new Presidential initiative aimed at producing a 1000-fold increase in supercomputing speed and a 100-fold improvement in available communications capability by 1997. NASA will use these unprecedented capabilities to help maintain U.S. leadership in aeronautics and Earth and space sciences by applying them to a wide range of scientific and engineering "Grand Challenge" problems. These are fundamental problems whose solution requires significant increases in computational power and are critical to meeting national needs.

The main objective of the federal HPCC program is to extend U.S. technological leadership in high performance computing and computer communications systems. Applications of these technologies will be widely disseminated within the U.S. government, industry and academia to accelerate the pace of innovation and serve the U.S. economy, national security, education and the global environment. They also will result in greater U.S. productivity and industrial competitiveness by making HPCC technologies an integral part of the design and production process.

NASA's participation in researching such advanced tools will revolutionize the development, testing and production of advanced aerospace vehicles and reduce the time and cost associated with building them. This will be accomplished by using these technologies to develop multidisciplinary models which can simultaneously calculate changes in a vehicle's fluid dynamics, structural dynamics and controls.

This contrasts with today's limited computing resources which are forcing researchers to utilize simple, single discipline models to simulate the many aspects of advanced aerospace vehicles and Earth and space phenomenon. This is more costly and time consuming than simulating entire systems at once, but it has become standard practice due to the complexity of more complete simulations and the insufficient computing power available to perform them.

As more advanced technologies are developed under the HPCC program, they will be used to solve NASA's Grand Challenge research problems. These include improving the design and simulation of advanced aerospace vehicles, allowing people at remote locations to communicate more effectively and share information, increasing scientists' abilities to model the Earth's climate and forecast global environmental trends, and improving the development of advanced spacecraft to explore the Earth and solar system.

Strategy and Approach: The HPCC program was designed as a partnership among several federal agencies and includes the participation of industry and academia. Other participating federal agencies include the Department of Energy, the National Science Foundation, the Defense Advanced Research Projects Agency, the Department of Commerce's National Oceanic Atmospheric Administration and National Institute of Standards and Technology, the Department of Education, the Department of Health and Human Services' National Institutes of Health, and the Environmental Protection Agency.

Together government, industry and academia will endeavor to meet program goals and objectives through a four part strategy (1) to support solutions to important scientific and technical challenges through a vigorous R&D effort, (2) to reduce the uncertainties to industry for R&D and use of these technologies through increased cooperation and continued use of the government and government-funded facilities as a prototype user for early commercial HPCC products (3) to support the research network and computational infrastructure on which U.S. HPCC technologies are based and, (4) to support the U.S. human resource base to meet the needs of all participants.

To implement this strategy, the HPCC program is composed of four integrated and coordinated components that represent key areas of high performance computing and communications:

High Performance Computing Systems (HPCS)--developing technology for computers that can be scaled up to operate at a steady rate of at least 1 trillion arithmetic operations per second -- 1 teraFLOPS. Research in this area is focusing both on increasing the level of attainable computing power and on reducing the size and cost of these systems in order to make them accessible to a broader range of applications. Although these computing testbeds will not be capable of teraFLOPS performance, they can be scaled up to that level by increasing the degree of their parallelism without changing their architecture.

Advanced Software Technology and Algorithms (ASTA)--developing generic software and mathematical procedures (algorithms) to assist in solving Grand Challenges.

National Research and Education Network (NREN)--creating a national network that will allow researchers and educators to combine their computing capabilities at a sustained rate of 1 billion arithmetic operations per second -- 1 gigaFLOPS.

Basic Research and Human Resources (BRHR)--promoting long-term research in computer science and engineering and increasing the pool of trained personnel in a variety of scientific disciplines.

Under the HPCS element, NASA is establishing high performance computing testbeds and networking technologies from commercial sources for utilization at NASA field centers. The agency is advancing these technologies through its Grand Challenge applications by identifying what needs to be developed to satisfy future computing requirements. NASA also acts as a "friendly buyer" by procuring early, often immature, computer systems, evaluating them, and providing feedback to the vendors on ways to improve them.

The agency's role under the ASTA element consists of leading federal efforts to develop generic algorithms and applications software for massively parallel computing systems. This role includes developing a Software Exchange system to produce software in a modular fashion for sharing and reuse. In this manner, complex software systems will be developed at considerably reduced cost and risk.

NASA also leads the federal venture to develop a common set of systems software and tools. In this role, NASA organized and directed a workshop of experts from industry, universities and government to review current systems software and tools for high performance computing environments and identify needed developments. Attendees were organized into seven working groups to address the areas of applications, mathematical software, compilers and languages, operating systems, computing environments, software tools and visualization. Their findings were compiled into a technical report and will be used to help form a national view of what is needed in systems software.

As a participant in the National Research and Education Network, NASA pursues the development of advanced networking technologies which allow researchers and educators to carry out collaborative research and educational activities, regardless of the participants' locations or their computational resources. During fiscal year 1992, the agency entered into a cooperative agreement with the Energy Department to procure network services that operate at 45 mbps. Under the Inter-agency Interim National Research and Education Network (NREN), the two agencies selected GTE Sprint to provide access to a nationwide fiber optic network that will help meet communications needs and serve as the foundation for increasingly fast networks. Future NREN goals include developing even faster networks that operate at 155 mbps, 622 mbps and eventually at gigabit speeds.

NASA's contribution to the Basic Research and Human Resources component includes providing funding support for several research institutes and university block grants. Along these lines, NASA has already initiated graduate research programs in HPCC technologies at five NASA centers, funded several post-doctoral students, established a pilot NREN access project, increased the NASA "Spacelink" education bulletin board to boost internet service and begun exploring collaborative efforts in K-12 education.

Organization: NASA's HPCC program is organized into three projects which are unique to the agency's mission: the Computational Aerosciences (CAS) project, the Earth and Space Sciences (ESS) project and the Remote Exploration and Experimentation (REE) project . The REE project will not be active from fiscal year 1993 through fiscal 1995, but is scheduled to resume activities in FY 1996.

Each of the projects is managed by a project manager at a NASA field center while the Basic Research and Human Resources component is managed by the HPCC program office at NASA Headquarters. Ames Research Center leads the CAS project and is supported by the Langley Research Center and the Lewis Research Center. Goddard Space Flight serves as the lead center for the ESS project and receives support from the Jet Propulsion Laboratory. The REE project is lead by JPL with support from GSFC. Finally, the National Research and Education Network component, which cuts across all three projects, is managed by Ames Research Center.

Management Plan: Federal program management is provided by the White House Office of Science and Technology Policy (OSTP) through the Federal Coordinating Council on Science, Engineering and Technology (FCCSET) Committee on Physical, Mathematical and Engineering Sciences (PMES). The membership of the PMES includes senior executives of many federal agencies.

Program planning is coordinated by the PMES High Performance Computing, Communications and Information Technology (HPCCIT) Subcommittee. The HPCCIT, currently lead by DOE, meets regularly to coordinate agency HPCC programs through information exchanges, the common development of interagency programs and reviews of individual agency plans and budgets.

NASA's HPCC program is managed through the agency's Office of Aeronautics and represents an important part of the office's research and technology program. The Headquarters staff consists of the director, the HPCC program manager and the manager of the Basic Research and Human Resources component. The HPCC office is responsible for overall program management at the agency, the crosscut of NASA HPCC-related programs, coordination with other federal agencies, participation in the FCCSET, HPCCIT, its Scientific and Engineering Working Group and other relevant organizations.

Points of Contact:

Lee B. Holcomb, Director
Paul H. Smith, Program Manager
Paul Hunter, Program Manager
Office of Aeronautics
High Performance Computing and Communications Office
NASA Headquarters
(202) 358-2747

(THIS PAGE LEFT BLANK INTENTIONALLY.)

NASA Directory

Center/Institute	Acronym
Ames Research Center Moffett Field, CA 94035	ARC
Center of Excellence in Space Data and Information Sciences Goddard Space Flight Center Greenbelt, MD 20771	CESDIS
Goddard Space Flight Center Greenbelt, MD 20771	GSFC
Headquarters Washington, DC 20546	HQ
Institute for Computer Applications in Science and Engineering Langley Research Center Hampton, VA 23665	ICASE
Jet Propulsion Laboratory Pasadena, CA 91109	JPL
Langley Research Center Hampton, VA 23665	LaRC
Lewis Research Center Cleveland, OH 44125	LeRC
Research Institute for Advanced Computer Science Ames Research Center Moffett Field, CA 94035	RIACS

(THIS PAGE LEFT BLANK INTENTIONALLY.)

Table of Contents

Table of Contents (continued)

List of Figures

Computational Aerosciences (CAS) Project

Testbeds

Applications Software

Systems Software

Earth and Space Science (ESS) Project

Testbeds

Applications Software

Systems Software

List of Figures (continued)

Remote Exploration and Experimentation (REE) Project

National Research and Education Network (NREN)

Basic Research and Human Resources Project (BRHR)

Introduction to the Computational Aerosciences (CAS) Project

Goal and Objectives: The goal of the CAS Project is to develop the necessary computational technology for the numerical simulation of complete aerospace vehicles for both design optimization and analysis throughout the flight envelope. The goal is supported by four specific objectives:

□ Develop multidisciplinary computational models and methods for scalable, parallel computing systems.

□ Accelerate the development of computing system hardware and software technologies capable of sustaining a teraFLOPS performance level on computational aeroscience applications.

□ Demonstrate and evaluate computational methods and computer system technologies for selected aerospace vehicle and propulsion systems models on scalable, parallel computing systems.

□ Transfer computational methods and computer systems technologies to aerospace and computer industries.

Strategy and Approach: This research will bring together computer science and computational physics expertise to analyze the requirements for multidisciplinary computational aeroscience, evaluate extant concepts and products, and conduct the necessary research and development. The steps involved include the development of requirements and evaluation of promising systems concepts using multidisciplinary algorithms; the development of techniques to validate system concepts; the building of application prototypes to serve as proof of concept; and the establishment of scalable testbed systems which are connected by multimegabit/second networks.

Simulation of the High Speed Civil Transport (HSCT) and High Performance Aircraft (HPA) have been chosen as "Grand Challenges". Langley is the lead center for HSCT and Ames is the lead center for HPA. Lewis has the lead on modeling propulsion systems in both HSCT and HPA. Areas of interest in systems software are related to the programming environment and include user interfaces, programming languages, performance visualization and debugging tools, and advanced result analysis capabilities. Testbeds include a CM2 and iPSC/860 at ARC, smaller iPSC/860s at LaRC and LeRC, and the Touchstone Delta at Cal Tech. All three research centers plan on upgrading or adding a new testbed in 1992, and Ames plans a major testbed installation in 1993.

Organization: All of the activities at a particular center report through the Associate CAS Project Manager at the Center to the CAS Project Manager at Ames Research Center. The CAS Project Manager reports to the CAS Program Manager who is in the Office of Aeronautics (OA) at NASA Headquarters. In addition to this organizational reporting, there is a matrixed reporting across the three areas (applications, systems software, and testbeds). Ken Stevens is the primary contact for the CAS Project. Manuel Salas is the focal point at LaRC and Russell Claus is the focal point at LeRC. Other points of contact are in the organizational chart found in the next section.

Management Plan

CAS Project Manager
Ken G. Stevens, Jr.

ARC Associate Manger
Ken G. Stevens, Jr.

Application Leader
Terry Holst

Systems Software Leader
Tom Lasinski

Testbed Leader
Russell Carter

Point of Contact:
Ken G. Stevens, Jr.
ARC
(415) 604-5949

LaRC Associate Manager
Manuel Salas

Application Leader
Tom Zang

Systems Software Leader
Andrea Overman

Testbed Leader
Geoffrey Tennille

LeRC Associate Manager
Russell Claus

Application Leader
Russell Claus

Systems Software Leader
Gary Cole

Testbed Leader
Jay Horowitz

Overview of CAS Project Testbeds

Goals and Objectives: Testbeds are where the applications, systems software, and system hardware come together to be tested and evaluated. The goals of the testbeds are to provide feedback to the applications, systems software, and computer system developers as well as point the way to the computational resources necessary to solve the grand challenges in computational aerosciences. The approach is to acquire early versions of promising computer systems and map CAS applications onto these systems via to the systems software.

Strategy and Approach: The largest of the testbeds for the CAS Project will be operated by Ames Research Center. Smaller systems will be operated by Langley and Lewis Research Centers. At present, Ames has an Intel Touchstone Gamma with 128 nodes and a 32K node Thinking Machines Connection Machine 2.

Organization: Both Langley and Lewis have the commercial version of the Intel Touchstone Gamma, iPSC/860, with 32 nodes. Ames will be upgrading its two machines early in 1993. Langley will also upgrade its iPSC/860 to an Intel Paragon in early 1993. Lewis will be putting together a cluster of IBM RS6000 systems in late 1992.

Management Plan: Each center has a testbed leader (ARC--Russell Carter, LaRC--Geoff Tennille, and LeRC--Jay Horowitz). These testbed leaders form a testbed working group which coordinates use and development of the testbed systems. Further information about a testbed may be sought from the center's testbed leader.

Points of Contact:

Russel Carter	Geoff Tennille	Jay Horowitz
ARC	LaRC	LeRC
(415) 604-4999	(804) 864-5786	(216) 433-5194

Lewis Cluster Testbed

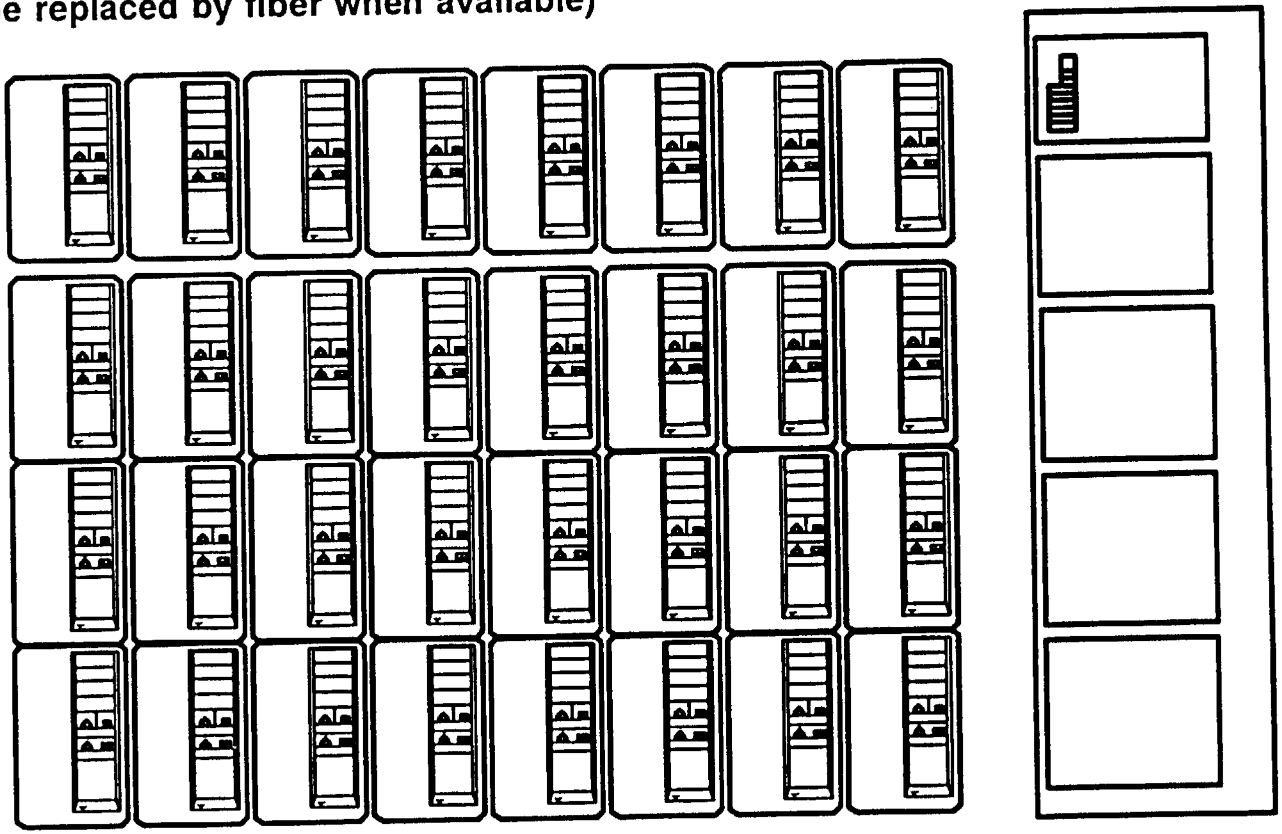
32 IBM Risc System/6000 Model 560s
- 1 GFlop aggregate performance
- 30.5 MFlops (LINPAC) / node
- Development of parallel, distributed and combined applications
- Batch applications server

Internode networking:
- Ethernet for Batch/Interactive
- 2nd Ethernet for temporary MPP network (to be replaced by fiber when available)

Model 970 Resource Manager:
- 9 GByte user file system
- Network connection to outside

8 Primary Batch Nodes:
- 6 @ 128MByte / 1 GB
- 2 @ 512Mbyte / 2 GB

24 Primary MPP Nodes:
- 64MByte / 1 GB



LeRC Parallel Processing Testbed

Objective: To establish a testbed for early evaluation of parallel architectures responsive to the computational demands of the Lewis propulsion codes.

Approach: A localized cluster of high-end IBM workstations will be assembled and configured to provide for both distributed memory, MIMD parallel processing and distributed processing applications. Internode traffic initially will be carried via ethernet but will be replaced with a fiber channel cross-bar network when available.

Accomplishments: A highly flexible configuration of clustered IBM RISC systems has been designed. The 32-node cluster will contain 32 IBM Model 560 RISC systems, each with a minimum of 64MByte memory, 1 GByte disk, and a CPU benchmarked at 30.5 MFlop (LINPAC). Some nodes will have expanded memory (4 with 128MB, 2 with 512MB). An IBM Model 970 with 6 GByte disk will act as a resource manager. The cluster will have an aggregate maximum of approximately 1 GFlop performance. Each node will have two ethernet ports, one for connection to the outside world, the other as a temporary system for internode message passing. Parallel applications will be based on PVM or locally developed parallel libraries (APPL).

Significance: The RISC Cluster will provide early evaluation of the IBM MPP environment which is intended to provide scalable TeraFLOP systems by mid-decade. In addition, the cluster is well suited to NASA Lewis' multi-disciplinary approach to aeropropulsion simulation. Different modules of the simulation (e.g. inlet, combustor, etc.) can run on different nodes of the cluster, some possibly parallelizable, others potentially requiring nodes with more memory.

Status/Plans: The system is being procured through existing contracts, with delivery expected in early September 1992. Teams of researchers and system support staff are developing the appropriate software tools and application environment.

Jay G. Horowitz
Computer Services Division
Lewis Research Center
(216) 433-5194

(THIS PAGE LEFT BLANK INTENTIONALLY.)

Overview of CAS Applications Software

Goal and Objectives: The CAS Applications Software is aimed at solving two grand challenge applications: the optimization of a High Speed Civil Transport (HSCT), and the optimization and analysis of High Performance Aircraft (HPA). The HSCT work will be performed jointly by ARC, LaRC, and LeRC. The Ames and Langley Research Centers will perform various computations associated with the airframe, and the Lewis Research Center will be in charge of the propulsion elements.

The Langley Research Center will be the overall lead center for this HSCT grand challenge. In this effort, the disciplines of aerodynamics, structural dynamics, combustion chemistry, and controls will be integrated in a series of computational simulations about a supersonic cruise commercial aircraft. Emphasis within this portion of the HPCC program generally will be placed in three areas:

(1) Accurate and efficient transonic to supersonic cruise simulation of a transport aircraft on advanced testbeds,

(2) Efficient coupling of the aerodynamic, propulsion, structures and controls disciplines on advanced testbeds, and

(3) Efficient implementation of multidisciplinary design and optimization of advanced testbeds.

Although some unsteady computations such as transonic flutter prediction will be performed, the bulk of the computations associated with this grand challenge will emphasize steady cruise conditions.

Strategy and Approach: In this effort the disciplines of aerodynamics, thermal ground plane effects, engine stability, and controls will be integrated in a series of computational simulations about a high performance aircraft undergoing a variety of maneuver conditions. Emphasis generally will be placed in two areas:

(1) Efficient simulation of low-speed, maneuver flight conditions on advanced testbeds and

(2) Efficient coupling of the aerodynamic, propulsion, and control disciplines on advanced testbeds.

Organization: The HPA grand challenge will be performed jointly by ARC and LeRC. There will be two general areas associated with this grand challenge, a powered lift application and an additional HPA simulation, which will be determined in FY 1993. ARC will perform the various computations associated with the airframe, and Lewis will be in charge of the propulsion elements. Ames Research Center will be the overall lead center for this grand challenge.

Management Plan: The application leaders report to the CAS Project Manager. There are three CAS Application team leaders, one at each of the participating NASA Centers; these applications leaders form an applications working group which coordinates the development of CAS grand challenge applications.

Points of Contact:

Terry Holst	Tom Zang	Russell Claus
ARC	LaRC	LeRC
(415) 604-6032	(804) 864-2307	(216) 433-5869

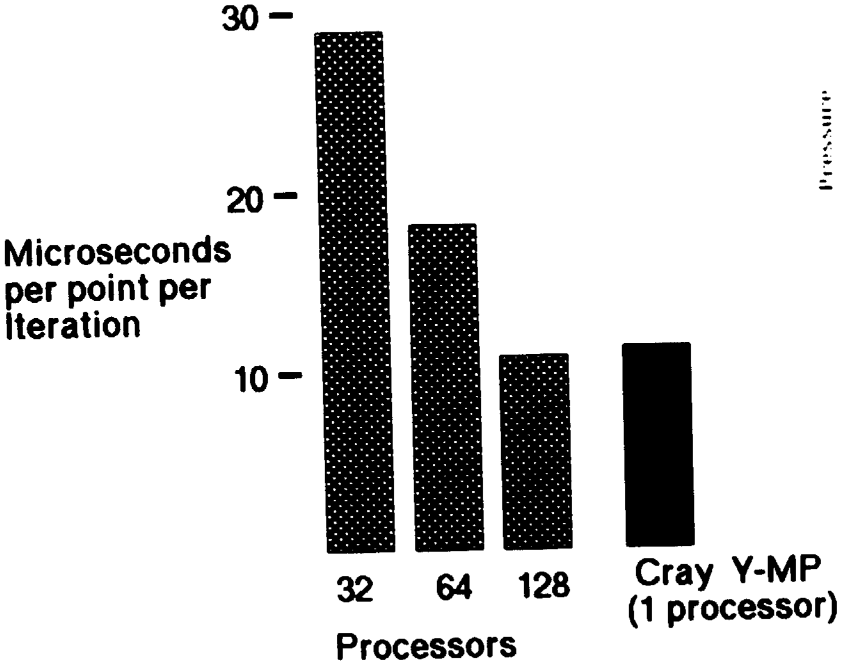
Supersonic Flow Calculations on a Parallel Computer

ARC - 3D

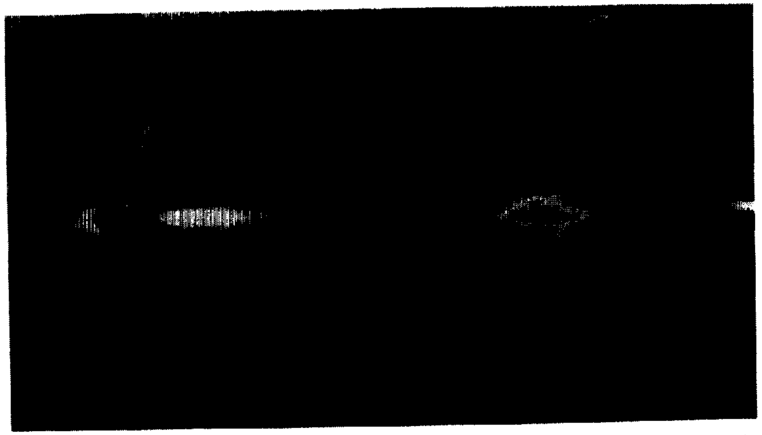
3D Euler solution

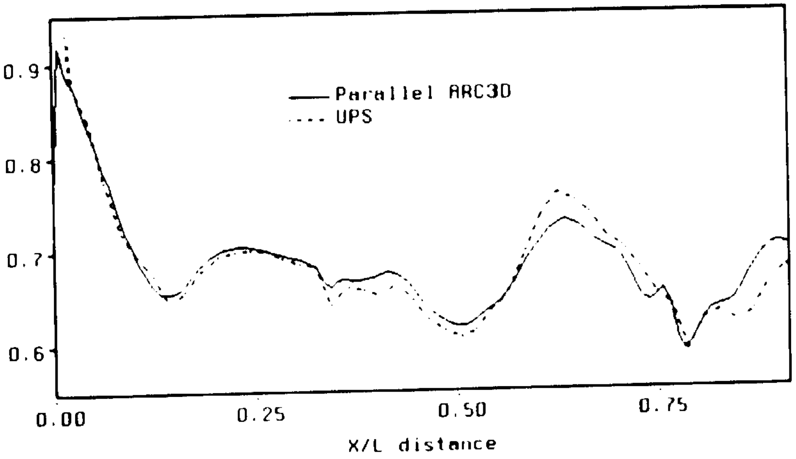
Computed on 32 processors of the Intel iPSC/860 in 3 hours, 20 minutes

Grid of 67x60x112 (450,240) points



Surface Pressure





Centerline Pressure

HSCT wing-body (upper surface)

Mach 2.1

alpha = 4.75°

Parallel Computer Optimization of a High Speed Civil Transport

Objective: The objective of this research project is to perform multidisciplinary optimization of a High Speed Civil Transport (HSCT) on a parallel computer. The optimization will consider aerodynamic efficiency, structural weight, and propulsion system performance. The multidisciplinary analysis will be performed by concurrently solving the governing equations for each discipline on a parallel computer. Developing scalable algorithms for the solution of this problem will be central to demonstrating the potential for teraFLOPS execution speed on a massively parallel computer.

Approach: The solution algorithms for each discipline will be adapted from existing, serial computer implementations to a scalable parallel computing environment. Parallelism will be pursued on all levels: fine-grained parallelism of the solution algorithm; medium-grained parallelism via domain decomposition; and coarse-grained parallelism of individual disciplines. The disciplines will be coupled to each other directly through the boundary conditions. For example, the fluid dynamic analysis will communicate aerodynamic loads to a structural analysis. The structural analysis will return surface displacements to the fluid dynamic analysis. An optimization routine will monitor the performance of the multidisciplinary system and search the design space for an optimal configuration. The discipline coverage and geometrical complexity of the test problem will be expanded as the solution methods mature and execution speed increases. Solving the complete HSCT optimization problem will require execution speeds of hundreds of MFLOPS, and will demonstrate scalability to TFLOPS.

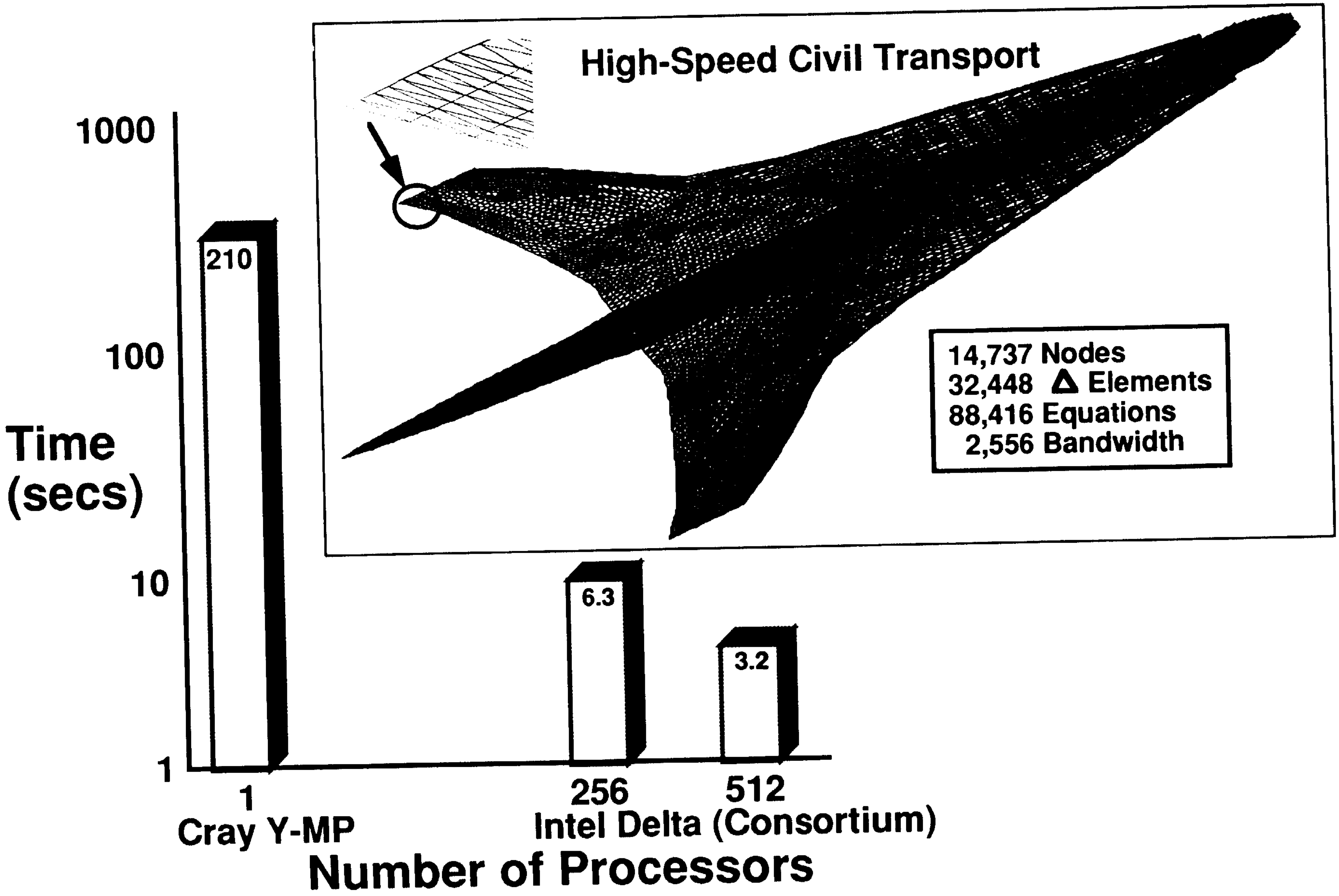
Accomplishments: The flow solution capability has been ported to the Intel iPSC/860 parallel computer. This code solves the Euler or thin-layer Reynolds-averaged Navier-Stokes equations in a zonal grid framework. Thus, geometries from simple wing-bodies to complete aircraft configurations can be treated. A set of test computations have been completed to evaluate this capability. The first case was a single-zone Mach 2.1 Euler simulation about a representative HSCTwing-body. The pressure field compares well with results from the UPS parabolized code, and execution times are comparable to the serial ARC-3D algorithm on a single Cray Y-MP processor. A second case resolves the turbulent flow about the same HSCT wing-body, at a Reynolds number of 1 million. The Baldwin-Lomax turbulence model is used.

Significance: Solving the flowfield about a complete HSCT configuration is one of the most computer-intensive aspects of this project. This capability is the cornerstone of the multidisciplinary design optimization goal. The code will also serve as a research platform for parallelization studies. These studies will guide efforts to optimize the efficiency of CFD codes on massively parallel computers.

Status/Plans: An optimization algorithm will be joined to the flow solver to establish an aerodynamic optimization capability. Techniques of parallelizing the optimization problem will be explored. Propulsion and structures modules will be coupled as they become available.

James S. Ryan
Computational Aerosciences Branch
NASA Ames Research Center
(415) 604-4496

Performance of New Node-by-Node Structural Matrix Assembly Demonstrated on Parallel Supercomputers



Fast Algorithm for Generation and Assembly of Structural Equations of Motion on Massively Parallel Computers

Objective: To develop a scalable algorithm for massively parallel computers which significantly reduces the time required to generate and assemble the structural equations of motion.

Approach: Optimization of aircraft design as well as nonlinear analysis, require many iterations involving the generation and assembly of the structural equations of motion. Since advanced complex aerospace vehicles result in large systems of matrix equations, the generation and assembly of these equations can become a significant fraction of the overall solution time. Conventional structural finite element codes generate and assemble structural matrices element by element. Parallelization of the conventional procedure, with element calculations distributed to various processors, results in poor performance because the processors attempt to simultaneously make contributions to the same matrix locations thereby creating a hardware bottleneck and synchronization problem. To circumvent this, an alternative node-by-node generation and assembly algorithm was developed which distributes nodal calculations to processors rather than element calculations.

Accomplishments: A parallel algorithm for structural finite element generation and assembly has been developed and tested on various applications. In particular, it has been applied to a Mach 3 version of an HSCT aircraft. The algorithm has been demonstrated on an Intel Delta computer to achieve nearly optimum, (i.e., linear) speed-up because it eliminates communication and synchronization bottlenecks

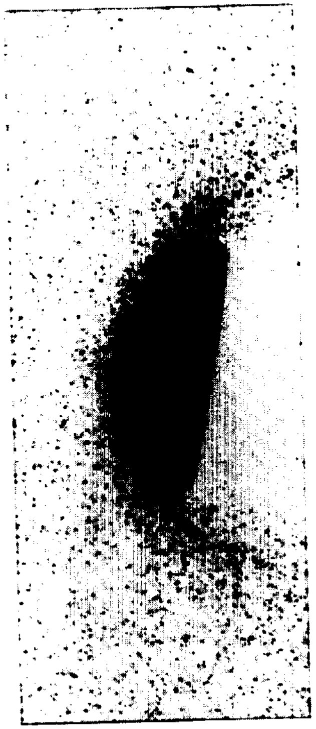
Significance: The algorithm markedly improves computation speed without loss of accuracy on distributed memory massively parallel computers. It is especially valuable in design optimization wherein there occur thousands of iterations.

Status/Plans: The algorithm is presently being applied to a Mach 2.4 version of the HSCT. It will be combined with a new faster solver, also under development, to form the underpinnings of an interdisciplinary aero/structural design system.

Ronnie E. Gillian
Computational Mechanics Branch
NASA Langley Research Center
(804) 864-2918

Particle Simulation in a Parallel Computing Environment

Flowfield Particles



Specifications:

60-deg half-angle Blunted Cone

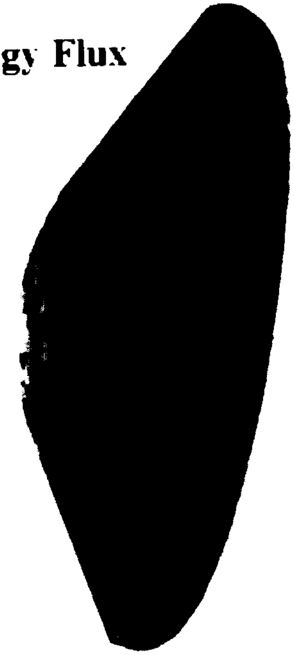
10-deg Angle of Attack

Mach Number: 24

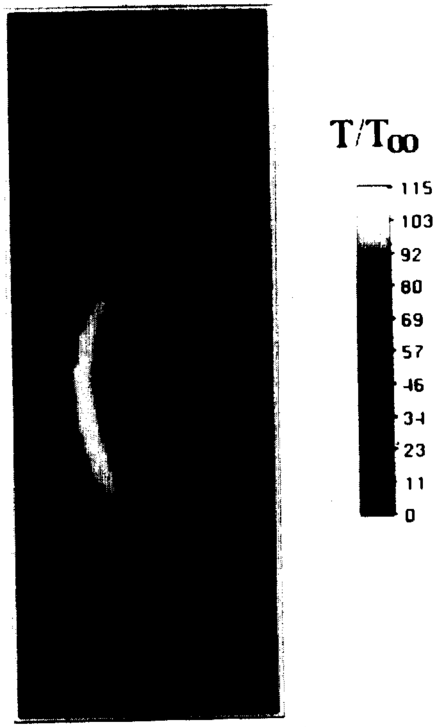
Knudsen Number: 0.04

Wall Temperature: 1500 K

Surface Energy Flux



Flowfield Temperature



Machines:

Intel Gamma (iPSC/860)
Intel Delta
Stanford DASH prototype

FALLAVOLLITA/HAAS (NASA AMES. AUG.1992)

Direct Particle Simulation of Rarefied Aerobraking Maneuvers

Objective: Direct particle simulation techniques are the only accurate means of computing highly rarefied hypersonic flowfields associated with aerobraking maneuvers. The objectives of this study are to develop and demonstrate a robust particle simulation method on multiprocessor systems, and to assess scaleup and performance of the implementation.

Approach: The flowfield is modeled by computing the motion and interaction of thousands or millions of discrete particles, thereby simulating the rarefied gas dynamics directly. The flow domain is composed of cubic cells which are grouped into blocks of fixed size (perhaps 512 cells/block). Collections of blocks can be assigned and reassigned dynamically to the available processors in a manner which evenly distributes the computational burden of the entire simulation. Blocks assigned to a given processor are typically aligned with the local flow direction, minimizing inter-processor communication requirements.

A version of the code running on the Intel Gamma machine (iPSC/860) uses a separate host computer to coordinate the processing tasks of the iPSC/860 nodes. A version running on the Intel Delta machine must use one of the processing nodes itself to serve as host for the remainder of the machine.

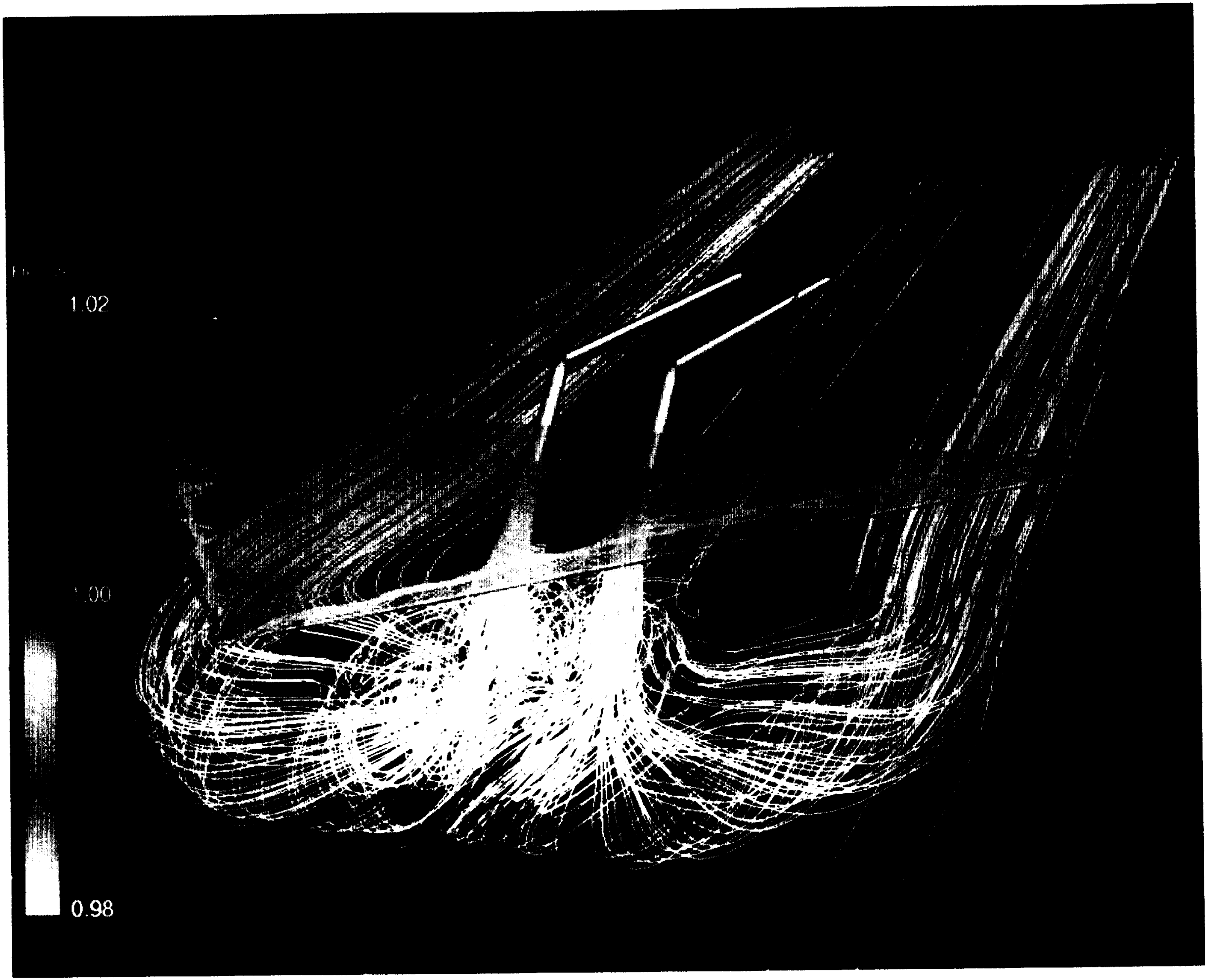
Accomplishments: A simulation code has been developed which may employ all 128 processing nodes of the Intel iPSC/860 Gamma machine. Simulations of simple channel flow, chemically-relaxing reservoirs, and complete 3-D nonequilibrium flowfields of reactive gas mixtures about blunt bodies during aerobraking indicate nearly linear scaleup with the number of nodes employed. Code performance is measured in CPU time/particle/time step. In comparison with similar code which was highly optimized for vector computers, the performance of the parallel simulation using 32 nodes on the iPSC/860 exceeds that of the Cray-2 implementation; using 64 nodes exceeds that of the Cray-YMP implementation.

The revised code running on the Intel Delta machine (without separate host) has simulated the same blunt body flows with as many as 32 processors, demonstrating computational performance nearly identical to that of the Intel Gamma code.

Significance: Particle simulation techniques are particularly well-suited to parallel computing architectures. More importantly, they impose computational burdens which are quite different from continuum CFD methods. Significant performance advantages over vector-optimized codes clearly demonstrate the tremendous utility of proper implementation in the multiprocessor environment.

Status/Plans: The codes require further testing to identify and correct unresolved errors. The code on the Intel Delta machine will be run with up to 512 processors to assess scaleup of the implementation. More sophisticated models for thermochemistry and gas-surface interaction will be added shortly.

Brian L. Haas
Eloret Institute
Aerothermodynamics Branch
NASA Ames Research Center
(415) 604-1145


1.02
1.00
0.98

High Performance Aircraft Wing Simulations on the Intel iPSC/860

Objective: To perform High Performance Aircraft wing simulations on the Intel iPSC-860 and similar parallel architectures in an effort to obtain better CPU performance.

Approach: Simulate flow past a delta wing with thrust reverser jets in ground effect on the Intel using DMMP-OVERFLOW (by Weeratunga). DMMP-OVERFLOW is a code written especially for the Intel iPSC-860 to solve multi-zone overset grid problems. To achieve optimized performance on the machine, try partitioning of grids across the nodes based on number of grid points in each direction, and based on communication intensity in each direction. Compare accuracy and performance of this optimized simulation with an optimized simulation on the Cray YMP.

Accomplishments: Time accurate Navier-Stokeson computations were performed on the Intel iPSC-860 to simulate flow past a delta wing with thrust reverser jets in ground effect. The four overset grids were loaded into four different cubes. Various partitionings were explored across the processors in a cube based on grid density and communication intensity considerations. It was determined that the grid density based partitioning optimizes overall efficiency. Excellent load balancing was obtained when 112 nodes were used in the manner tabulated below:

Grid. #	Grid Size	Cube Size
1	(70,56,70)	(4,2,4)=32 nodes
2	(83,81,47)	(4,4,2)=32 nodes
3	(60,71,52)	(4,4,2)=32 nodes

Increasing the number of nodes for grid 4 adversely affected the ioad balancing and did not result in improved performance. I/O performance of the machine was deemed inadequate for unsteady problems requiring frequent output of solution. Performance comparisons with Cray YMP are listed below:

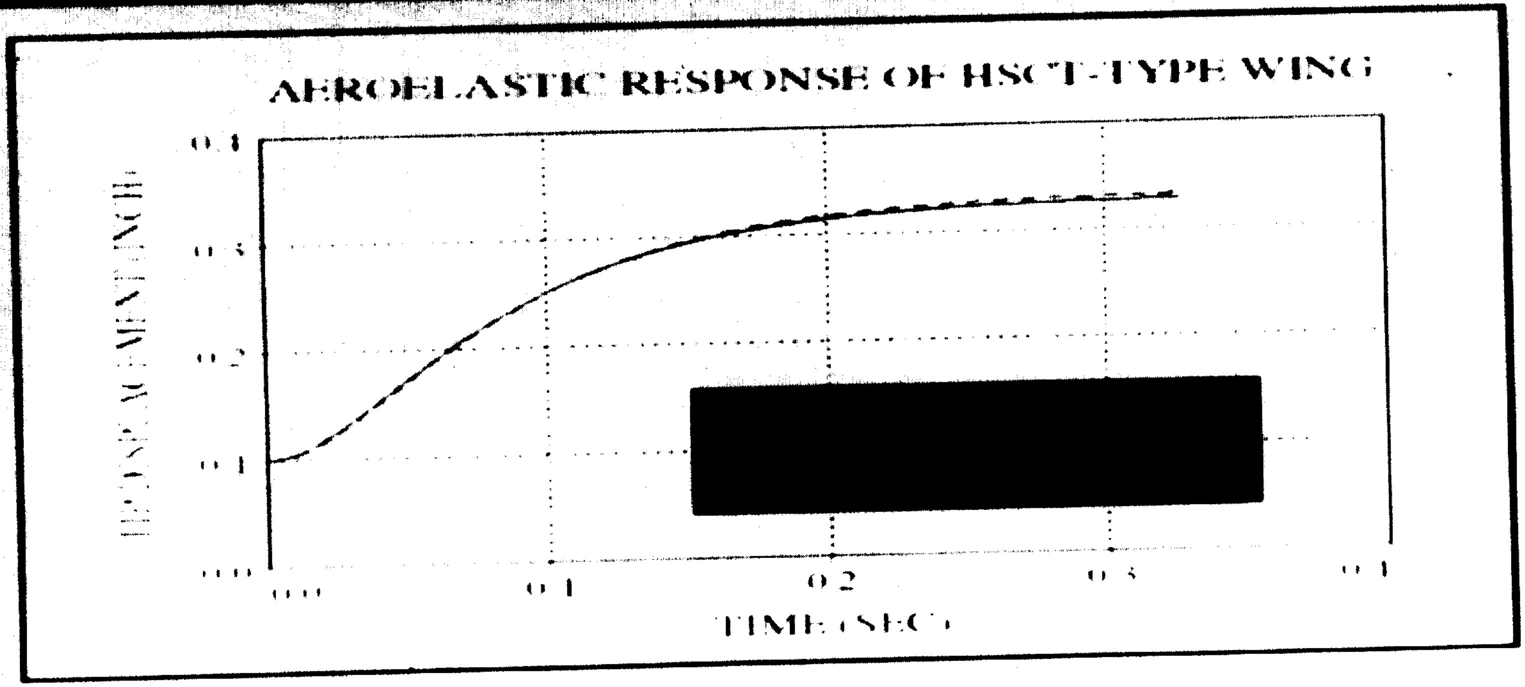
Section	YMP 1 Proc.	iPSC/860 112 Proc.
time/pt./step	14 microsecs	7microsecs.
grids solved	sequentially	in parallel
memory	12 MW	112 MW
Accuracy	64 bit	64 bit

Significance: For the first time, overset multi-zone Navier-Stokes computations were performed on the Intel. Performance of the Intel was compared with the Cray YMP, and it was shown that a 112 node Intel performs about twice as fast as one YMP processor for the problem under study.

Status/Plans: Computations will be carried out on the Intel iPSC-860 to simulate high lift devices on the FLAC (Flight Lift and Control) wing.

Kalpana Chawla
MCAT Institute
Computational Aerosciences Branch
Ames Research Center
(415) 604-4981

Sisira Weeratunga
Computer Sciences Corporation
NAS Applied Research Branch
Ames Research Center
(415) 604-3963


AEROELASTIC RESPONSE OF HSCT-TYPE WING

Aeroelastic Computations on the Intel iPSC/860 Computer Using Navier-Stokes Code ENSAERO

Objective: The High Speed Civil Transport (HSCT), the next generation supersonic civil transport aircraft, poses several challenges to designers, particularly in aeroelasticity. Because of the multidisciplinary nature, the aeroelastic computations are orders of magnitude more expansive than single discipline computations. Computational speed on the current serial computers has almost reached maximum limit. An alternate is to use parallel computers. The objective is to conduct aeroelastic computations on configurations using parallel computers.

Approach: The 3D Navier-Stokes equations of motion coupled with the finite-element structural equations of motion are solved using a time accurate numerical integration scheme with configuration adaptive dynamic grids. Within each domain (cube), computations are also performed in parallel. The information between the fluids and structures is communicated at the boundary interfaces. The figure on the facing page illustrates the advantages of using the parallel computers.

Accomplishment: Based on the above approach, the computer code ENSAERO was developed on the Cray Y-MP serial computer. Version 2.0 of the code that solves the Navier-Stokes/Euler equations simultaneously with the modal structural equations of motion has been mapped onto the Intel iPSC/860 computer. The fluids part of ENSAERO, including the moving grid, are computed in a cube of 32 processors, and the modal structures is computed in a cube with one processor. Although the structures model is simplified using a modal approach, it is a first-of-a-kind effort to solve fluid/structural interaction problem on a parallel computer. Using ENSAERO 2.0 on the Intel, computations are successfully made on a HSCT-type wing at M = 0.90. Figure 2 shows the tip response of a HSCT-type strake-wing configuration computed on Intel iPSC/860 computer.

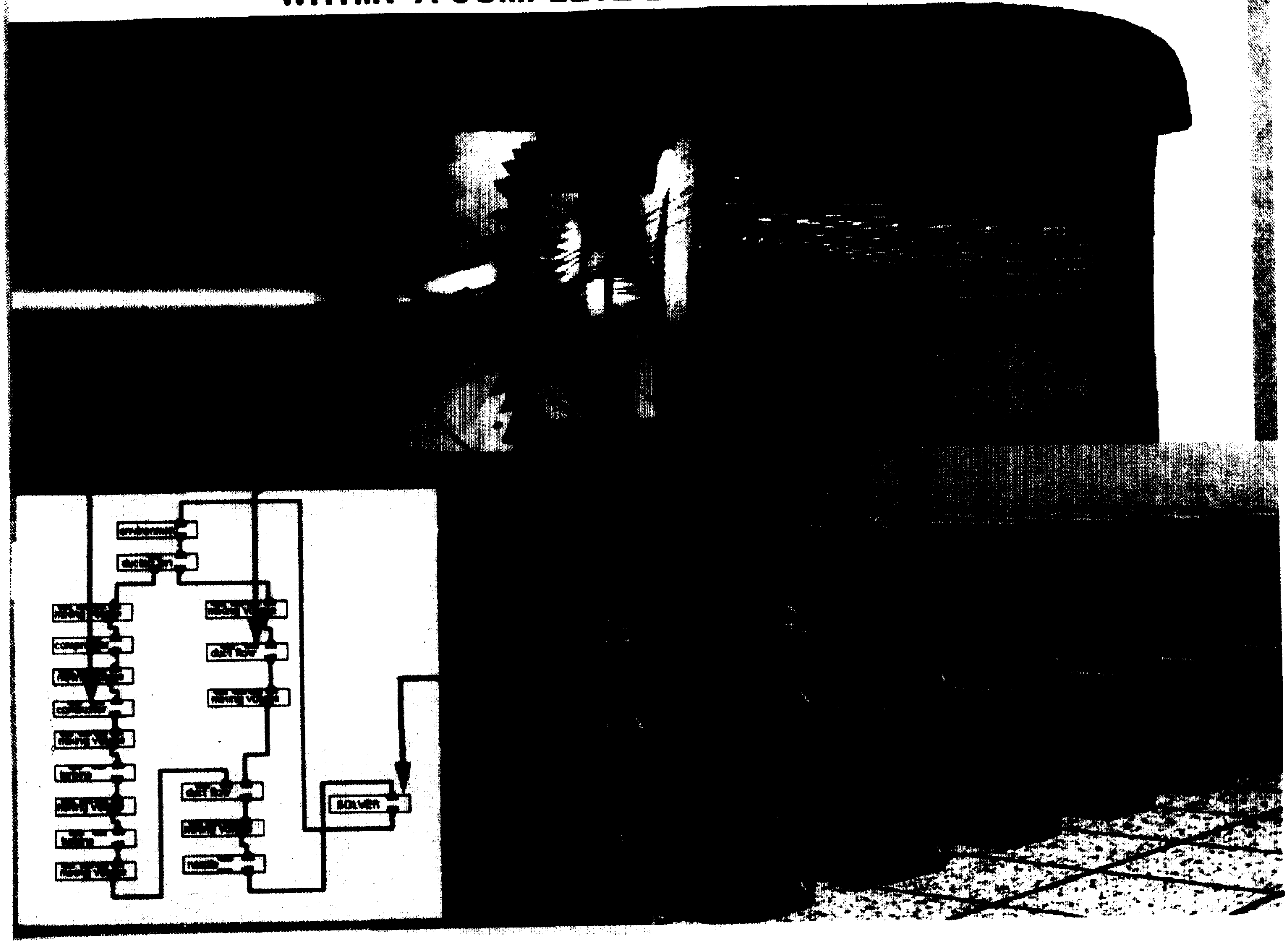
Significance: The successful implementation of ENSAERO on the Intel is a major stepping stone in the development of general purpose computer codes to solve fluid/structures interaction problems on parallel computers.

Status/Plans: Work will be continued to replace the modal structures with finite-element structures using advanced elements such as shell elements. The computational efficiency will be increased using more processors for both fluids and structures. The ENSAERO capability will be extended to model full-aircraft using the zonal grids for fluids and the sub-structures for structures. In the immediate future, aeroelastic results will be computed for the HSCT wing-body configurations. In the long run, controls and optimization disciplines will be implemented.

Guru Guruswamy
and Chansup Byun
Computational Aerosciences Branch

Eddy Pramono
and Sisira Weeratunga
NAS Applied Research Branch
(415) 604 6329/6416

TESTBED SIMULATION OF MULTIPLE COMPONENTS WITHIN A COMPLETE ENGINE CALCULATION



AVS Engine Simulation

Objective: To develop a prototype NPSS Numerical Propulsion System Simulator (NPSS) executive that will allow the dynamic selection of engine components through a visual programming interface in which the engine components are independent of hardware platforms. The NPSS executive will make use of an environment that provides for collecting engine component codes of differing fidelity across a heterogeneous computing system.

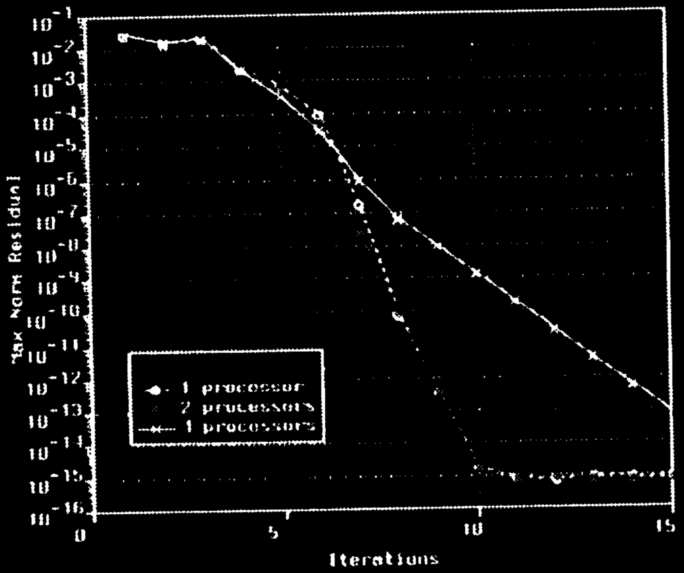
Approach: A NPSS executive will be created from recent advances in object oriented programming techniques combined with AVS (Application Visualization System) and with recent developments in communication software such as PVM (Oakridge's Parallel Virtual Machine) and APPL (Lewis's Application Portable Parallel Library).

Accomplishments: A one-dimensional, full engine simulation code has been reworked into an object oriented structure by GE Lynn, Mass and integrated into AVS locally at Lewis. An engine component object within the full engine simulation (e.g., a compressor) has been remotely executed through a PVM connection to the Lewis's Cray YMP and also to a SUN workstation. In addition, IBM has developed an object oriented front end to AVS to dynamically setup, build and control a full engine simulation.

Significance: The prototype executive eases the process of dynamically connecting engine component codes of different analysis levels across various machine architectures.

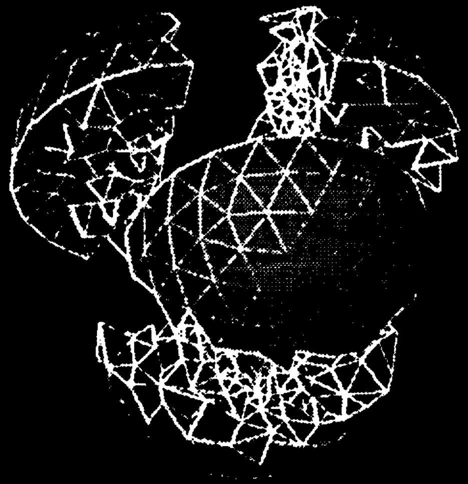
Status/Plans: The AVS implementation of a full engine simulation is now structured to allow the use of new system solvers. Work is underway to integrate a robust, stiff-equation solver. Future plans will provide for a suite of solvers. Work will continue on integrating PVM and APPL into AVS. The current version of our full engine simulation is written in Fortran. A full object oriented implementation is planned

Gregory J. Follen
Interdisciplinary Technology
Office
NASA Lewis Research Center
(216) 433-5193



[illegible]

[illegible]



[illegible] View of Four Processor Mesh

3-D Implicit Unstructured Euler Solver for the Intel Gamma Computer

Objective: To implement a 3D finite-volume unstructured grid algorithm for solving the Euler equations on a highly parallel computer. To investigate performance and resource tradeoffs for various implicit solution strategies.

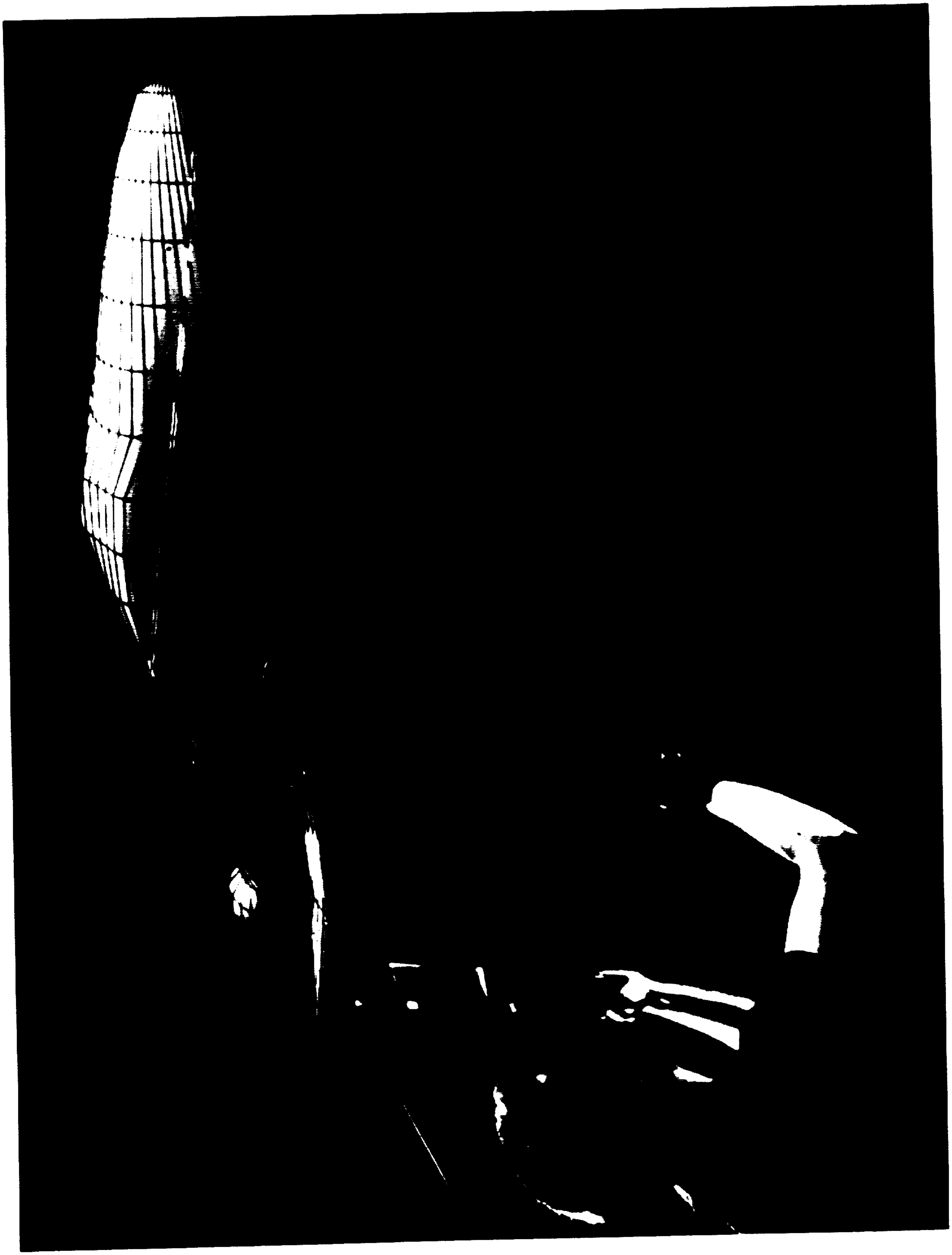
Approach: The Euler equations of gas dynamics are solved on unstruct-ured grids comprised of tetrahedral cells. An upwind finite-volume formulation with linear reconstruction is used to spatially discretize the Euler equations. A backward Euler time integration with Newton linearization produces a large sparse linear system of algebraic equations. This linear system is solved using a generalized minimum residual method (GMRES) with preconditioning (a modified incomplete L-U factorization technique). The implementation on the Intel Gamma machine assumes an apriori partitioning of the tetrahedral mesh amongst multiple processors. The discretization of the equations and GMRES solvers are implemented in the multi-processor environment using interprocessor communication so that results are identical to the single-processor implementation. For purposes of the GMRES preconditioning, interprocessor communication has been eliminated which is a departure from the single-processor implementation but does not appear to degrade the performance of the method seriously.

Accomplishments: The implicit code with linear reconstruction was tested on subsonic, transonic, and supersonic flow problems. Currently, a single 8Mb processor on the Intel Gamma machine can accommodate approximately 500 vertices of the mesh using double-precision (64 bit) storage. The memory requirement is dominated by the storage/ of the sparse linear system. Using single-precision storage of the linear system, the upper limit for mesh size is approximately 100,000 vertices for the Ames Intel Gamma computer. The code has been bench-marked at 200MFlops on 128 processors. This is only slightly better than a single processor version of the code on a CRAY Y-MP (150MFlops). The performance on the Intel is expected to improve as the memory per processor is increased and communication/computation ratio is reduced.

Significance: Results indicate that implicit unstructured grid calculations can be efficiently carried out in a highly parallel computing environment. The finite-volume discretization and GMRES solution strategies are easily implemented on highly parallel machines using a message passing paradigm.

Status/Plans: With anticipated up-grades in memory and the number of processors on the Ames Intel supercomputer, the code will be tested on larger problems. The code will also be modified to include the Navier-Stokes terms with turbulence modeling.

Timothy Barth
Timothy Tomaich (U. of Mich.)
Samuel Linton (MCAT Institute)
Fluid Dynamics Division
Ames Research Center
(415)604-6740



The Virtual Wind Tunnel

Objective: To provide a real-time interactive virtual environment for the exploration and visualization of simulated, unsteady three-dimensional fluid flows.

Approach: Virtual reality interface techniques are combined with fluid flow visualization techniques to develop an intuitive system for exploration of fluid flows. For display, the virtual reality interface uses the Fake Space Labs BOOM2C, a head-tracked, wide-field, high-resolution, two-color channel stereoscopic CRT system. For user interaction, a variety of technologies are employed including hand tracking and hand gesture recognition. The user's hand gesture is used to indicate a command to the visualization system; position is used to determine where that action should take place. There are two computational architectures currently implemented: local and distributed. In the local architecture, all computation and graphics associated with the visualization take place on the user's workstation. In the distributed architecture, the computations associated with the visualizations are performed on a remote supercomputer, currently a Convex C3240, and the results of the computations are sent to the workstation for rendering via the UltraNet gigabit network. The distributed architecture allows the higher computational power and larger memory of a supercomputer to be used to investigate larger flow solutions. The distributed architecture also allows two or more virtual reality stations to interact with the same flow data, so two or more researchers can collaboratively investigate a solution.

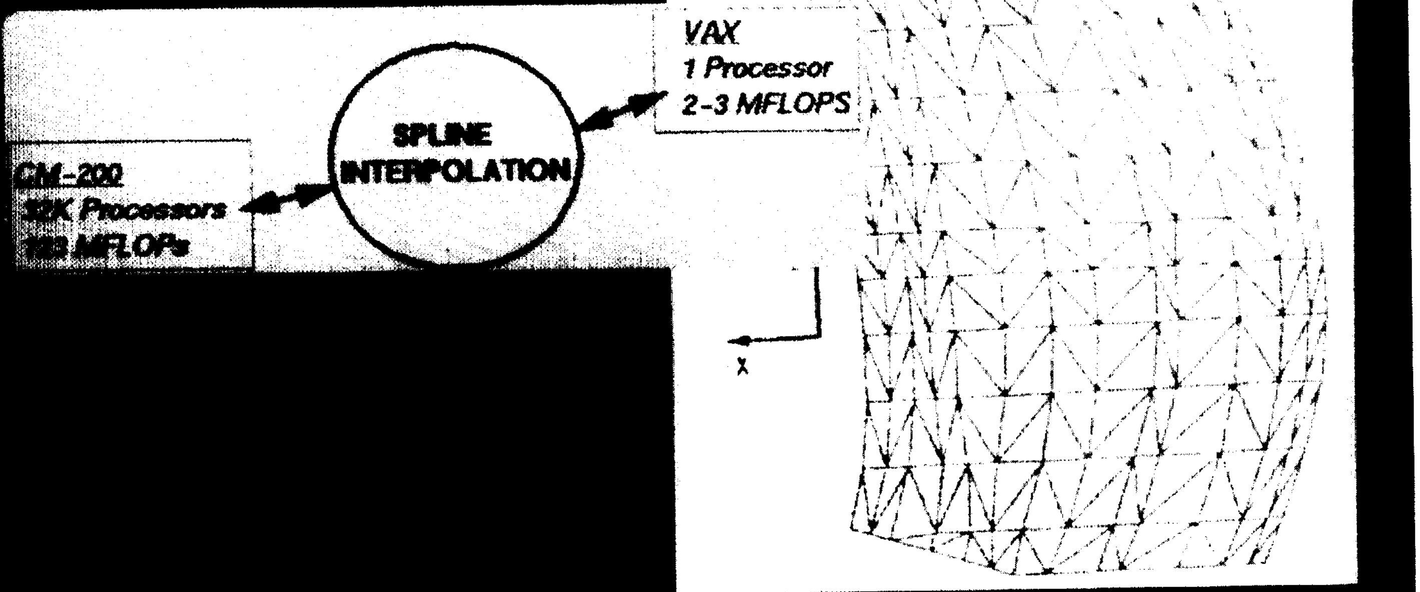
Accomplishments: The virtual wind tunnel system currently works in both local and distributed modes. Multiple zone unsteady data sets are supported for visualization. The visualization techniques currently work on the velocity vector field part of the flow solution only. The velocity vector field is visualized via simulated particle advection. Collections of particles can be manipulated directly by the user and their evolution can be observed in real time (~10 frames/second). A basic interface based on three-dimensional menus and sliders is used to control the visualization and other aspects of the virtual environment. To sustain the three-dimensional aspects of the virtual wind tunnel, a frame rate of at least 10 frames/second must be maintained. This frame rate constraint requires that the entire flow solution be resident in physical memory. The total size of data that can be viewed is limited by the memory of the computer system. This limitation is 256 megabytes in the local architecture and one gigabyte in the distributed architecture.

Significance: Virtual reality interface technology has shown great promise in enhancing the ability of researchers to explore extremely complex flows. The virtual wind tunnel project will test this enhancement by attempting to produce a working research tool.

Status/Plans: The priority of the virtual wind tunnel project will be transition from a research effort to a tool that can be used by flow researchers. Several features will be implemented as part of this process. Scalar visualization capability will be added. The control interface will be enhanced and voice recognition control will be added. Enhancement of the current display to full color is being investigated. The virtual wind tunnel will be tailored to the research problems of selected flow researchers.

Steve Bryson
NASA Ames Research Center
bryson@nas.nasa.gov
415/604-4874

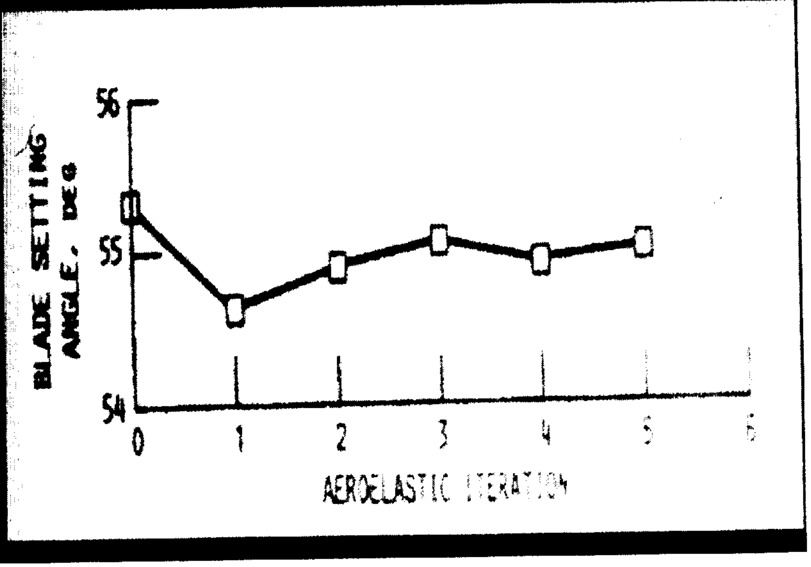
HETEROGENEOUS SIMULATION OF AERO/STRUCTURAL COUPLING



CM-200
32K Processors
[illegible] MFLOPs

SPLINE INTERPOLATION

VAX
1 Processor
2-3 MFLOPS



An Aerodynamic/Structural Analysis for Turbomachinery Blade Rows Using a Parallel/Serial Computer

Objective: Develop and demonstrate a multidisciplinary aerodynamic/ structural prediction method for application to steady-state turbomachinery design on a heterogeneous, massively parallel/ serial computer system. Apply this integrated analysis to assess and understand the effects of steady fluid/structure interaction on blade performance.

Approach: Perform an initial "weak" serial coupling of the UTRC, single blade-row, massively parallel version of the VSTAGE flow code with the NASA serial version of the MHOST structural code on a Connection Machine CM-200 computer system. Investigate methods of improving the coupling of the analyses and the parallel efficiency of the flow solver. Apply the coupled analyses to predict the steady-state aerodynamic performance and structural behavior of a turbomachinery blade design.

Accomplishments: The successful installation of the structural analysis code was followed by the development of an initial aerodynamic/structural model for the serial coupling. During this model development, a trim-to-power coupling process demonstrated the heterogeneous use of a simple trim algorithm, run on the serial front-end (VAX) computer of the CM-200, and the massively parallel flow solver, run on the SIMD processors of the CM-200. Investigations to improve the fluid/structural coupling process were initiated using a two-dimensional model problem. Improvements to the parallel efficiency of the flow solver on a 16K CM-200 were investigated, resulting in computational performance near parity with a single Cray YMP processor.

Significance: The integrated coupling of a turbomachinery fluid flow analysis and a structural analysis will demonstrate the viability of multi-disciplinary turbomachinery design application on a heterogeneous computer network and provide a framework for NASA to learn from and build upon in support of long range NPSS goals.

Status/Plans: All of the component codes and data models are functioning and the development of the interface software was initiated. An initial serial coupling is anticipated during this fiscal year, with refinements and application to follow during the 1993 fiscal year.

Russel W. Claus
Interdisciplinary Technology Office
NASA Lewis Research Center
(216) 433-5869

Parallel Performance of CDNS Kernel

Touchstone Delta Results

Grid Size	Number	MFLOPS				
	of Nodes	$\partial/\partial x$	$\partial/\partial y$	$\partial/\partial z$	Mean	Per Node
128^3	16	137	129	205	150	9.4
	32	204	252	414	266	8.3
	64	331	330	822	413	6.4
	128	252	464	1,533	442	3.5
	256	260	325	2,683	411	1.6
256^3	128	646	855	1,672	905	7.1
	256	970	1,064	3,223	1,315	5.1
	512	620	1,474	6,143	1,222	2.4
384^3	288	1,324	1,473	3,802	1,768	6.1
	384	1,374	1,751	5,007	2,002	5.2
432^3	432	1,385	2,107	5,640	2,184	5.1
512^3	512	1,636	2,635	6,700	2,632	4.6

Compressible Turbulence Simulation on the Touchstone DELTA

Objective: Implement a production code for simulations of compressible, homogeneous turbulence on the Touchstone DELTA, and perform computations with unprecedented resolution.

Approach: The compressible direct numerical simulation code (CDNS) which has been previously adapted to the Intel i860 Hypercube and used on moderately sized problems (128^3 grids), was implemented on the 512-node Touchstone DELTA, and a production run was performed on a 384^3 grid.

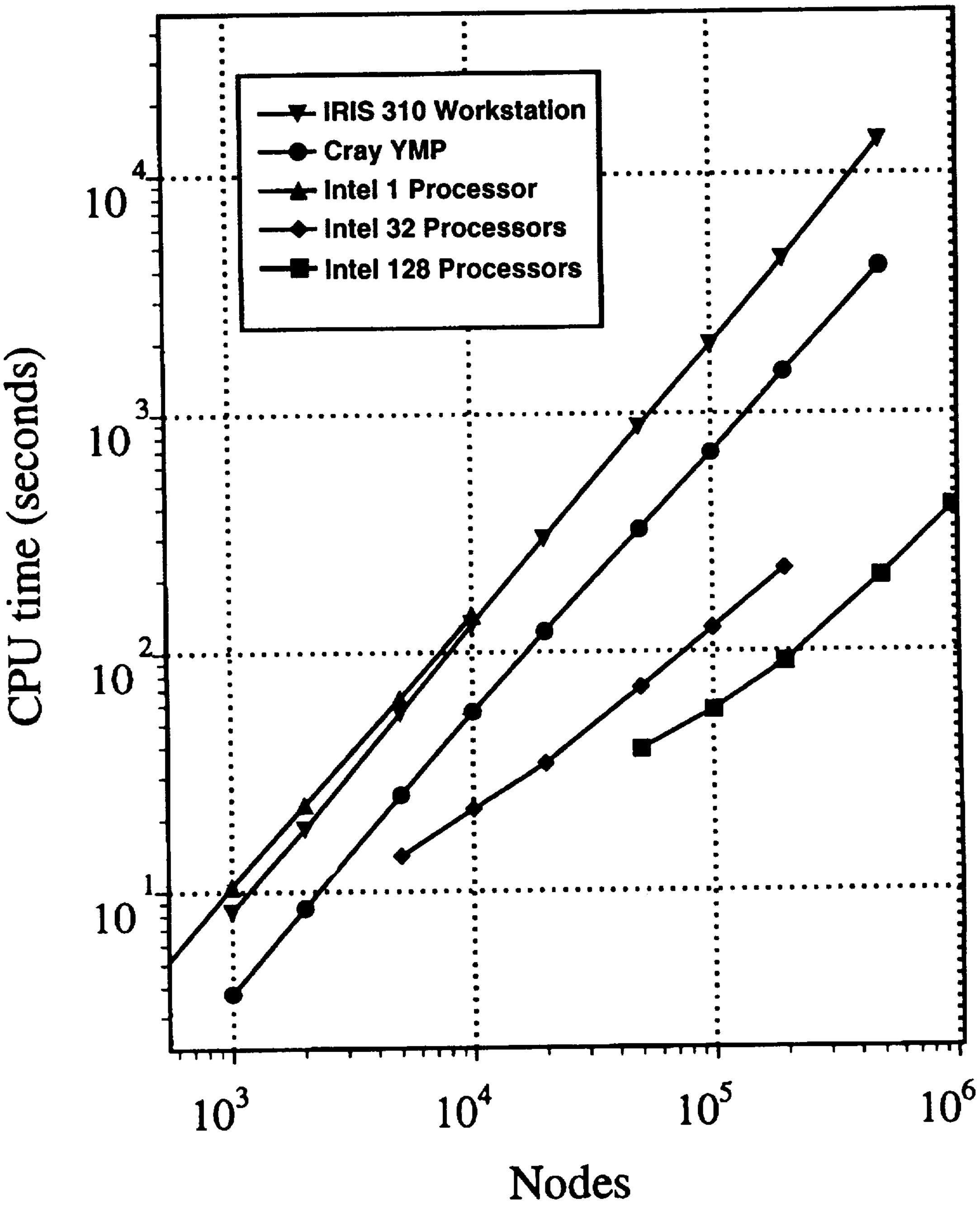
Accomplishments: The 3-D turbulence simulation code CDNS (and its variants) are heavily used at NASA Langley for basic research on the physics of compressible transition and turbulence. Most of the arithmetic work (and all of the inter-node communication) in the time-stepping algorithm of CDNS resides in the solution of scalar tridiagonal equations. These implicit equations are solved in the DELTA version of CDNS by a balanced Gauss elimination algorithm, which operates on data distributed over multiple nodes. The table documents the performance of a CDNS kernel--the three subroutines for computing derivatives in each coordinate direction. For the larger problems, sustained speeds in excess of 2 Gflops are achieved on the CDNS kernel. The full CDNS code achieves comparable speeds, e.g., 2.1 Gflops on a 384^3 grid distributed on 512 nodes. A complete simulation has been performed on the 384^3 grid and the results are currently being analyzed.

Significance: The CDNS code and its kernel are written in standard Fortran (with the sole addition of Intel message passing calls). The implementation strategy (especially that adopted for the implicit equations) is readily extendible to such production CFD codes as the single-block versions of CFL3D and ARC3D.

Status/Plans: The production run on the 384^3 grid will be thoroughly analyzed, and additional simulations on even larger grids (up to 450^3) will be conducted for compressible turbulence in the presence of strong eddy shocklets. The capability of simulating homogeneous turbulence in uniform shear flow will be added to the DELTA version of CDNS.

Thomas M. Eidson,
Gordon Erlebacher
Fluid Mechanics Division/ICASE
NASA Langley Research Center
(804) 864-2180

3D Delaunay Triangulation



Marshal L. Merriam NASA Ames

Parallel Implementation of an Algorithm for Delaunay Triangulation

Objective: To implement Tanemura's algorithm for 3D Delaunay triangulation on Intel's Gamma prototype. Tanemura's algorithm does not vectorize to any significant degree and requires indirect addressing. Efficient implementation on a conventional, vector processing supercomputer is problematic. However, efficient implementation on a parallel architecture is possible.

Approach: The 3D Delaunay triangul-ation algorithm due to Tanemura is mapped onto the Intel Hypercube (Gamma). It uses a unique partitioning strategy to take advantage of the MIMD parallelism available through Intel's Gamma prototype. Under the MIMD paradigm, each processor has a separate copy of the program and a spatially contiguous portion of the data. Under this domain decomposition approach, each processor is responsible for many range queries. These need not be (and generally are not) synchronized. The fact that range queries in different processors require different amounts of time is not a problem, since they don't interact. Processors finishing early simply proceed to the next query. Similarly, the differences in the number of queries required to form a given tetrahedron also does not affect efficiency. The remaining problem is controlling the interactions between tetrahedra, especially those on different processors. This is done by appropriate partitioning of the domain and interprocess communication.

Accomplishments: The 3D Tanemura Algorithm for Delaunay triangulation has been successfully mapped onto the Intel Hypercube Gamma and benchmarked against both a serial (on a workstation) and vectorized (Cray YMP) version of the algorithm. On an IRIS 310/VGX using large amounts of virtual memory, speeds of 40-120 nodes per second were measured (performance on large problems was degraded by page faults). Speeds on the Cray YMP (1 processor) were only 2-3 times faster (on the order of 7 MFLOPS was obtained), and the memory available was considerably less. This particular algorithm has few vectorizable operations: indirect addressing, conditional execution, and considerable integer arithmetic further degrade performance. On the Intel, single processor speed was on the order of the workstation speeds (but memory limits one to a small numbers of nodes). For 128 processors, up to 1,750,000 nodes can be accommodated. On a 1 million node case (approximately 6 million tetrahedron), execution times was about 7 minutes, which is 20 times the Cray YMP speed. The results are summarized in the accompanying figure.

Significance: This implementation of Tanemura's algorithm has produced a practical and efficient way of triangulating very large numbers of points. It is of special interest because it utilizes a true MIMD paradigm. As such, it is fundamentally different than many other parallel implementations, which can be (and often are) efficiently implemented on SIMD machines.

Status/Plans: The algorithm and code will be further improved to obtain peak efficiency on the Intel Gamma and the Delta machines. The 3D triangulation capability will be used to generate large grid systems for computations using an unstructured mesh code. Documentation and a user friendly form of the code are also being generated.

Marshal L. Merriam
Computational Technology Branch
Ames Research Center
(415) 604-4737

(THIS PAGE LEFT BLANK INTENTIONALLY.)

Overview of CAS Systems Software

Goal and Objectives: The CAS Systems Software activity is targeting key areas of systems software that are important to the development of CAS applications but which are receiving insufficient attention from the computer industry and others. The goal is to have a suite of systems software which is efficient to both the computers time and the applications developers time.

Strategy and Approach: The approach of the CAS Systems Software activity is to target key areas. These areas will tend to be more related to the end user, e.g., programming languages and environments, and less related to the details of the hardware, e.g., device drivers. Areas currently under investigation include programming languages, e.g., Fortran D, HP Fortran, and Fortran 90; distributed programming environments, e.g., PVM and APPL; performance analysis and visualization tools, e.g. AIMS; visualization and virtual reality tools, e.g., virtual wind tunnel; and object oriented environments for coupling disciplines and aircraft components. When prototype software is developed, it is used to aid the development and/or execution of grand challenge applications and evaluated as to its efficient use of the testbed and the application developer.

In addition to prototype development, there is extensive evaluation of testbed vendor supplied systems software and, in select cases, cooperative development or enhancement of the systems software. CAS does not plan on developing and supporting commercial grade systems software but plans on developing systems software technology which can become a non proprietary standard which can be commercialized by the private sector.

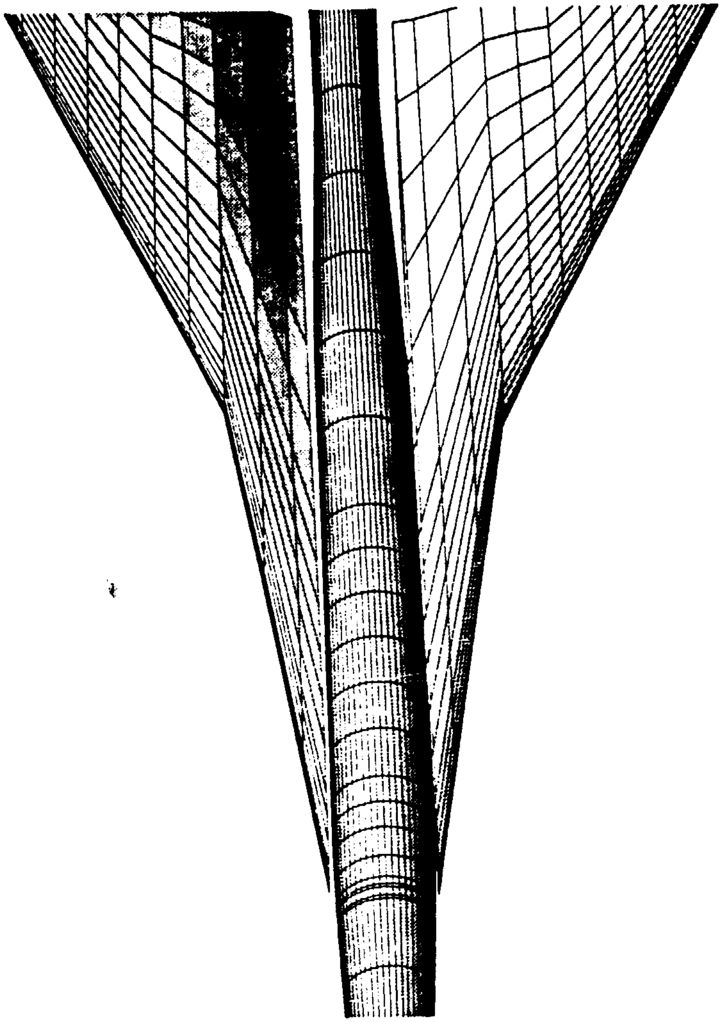
Organization: The systems software work is done at the NASA research centers, ICASE, RIACS, and by grantees. ARC, LaRC, and LeRC share resources within the CAS Project as well as with the Earth and Space Sciences (ESS) Project.

Management Plan: Each center has a systems software leader: ARC--Tom Woodrow: LaRC -- Andrea Overman: and LeRC -- Greg Follen. These leaders coordinate activities within the CAS project and work with the ESS Project to coordinate all of the NASA systems software work under HPCC.

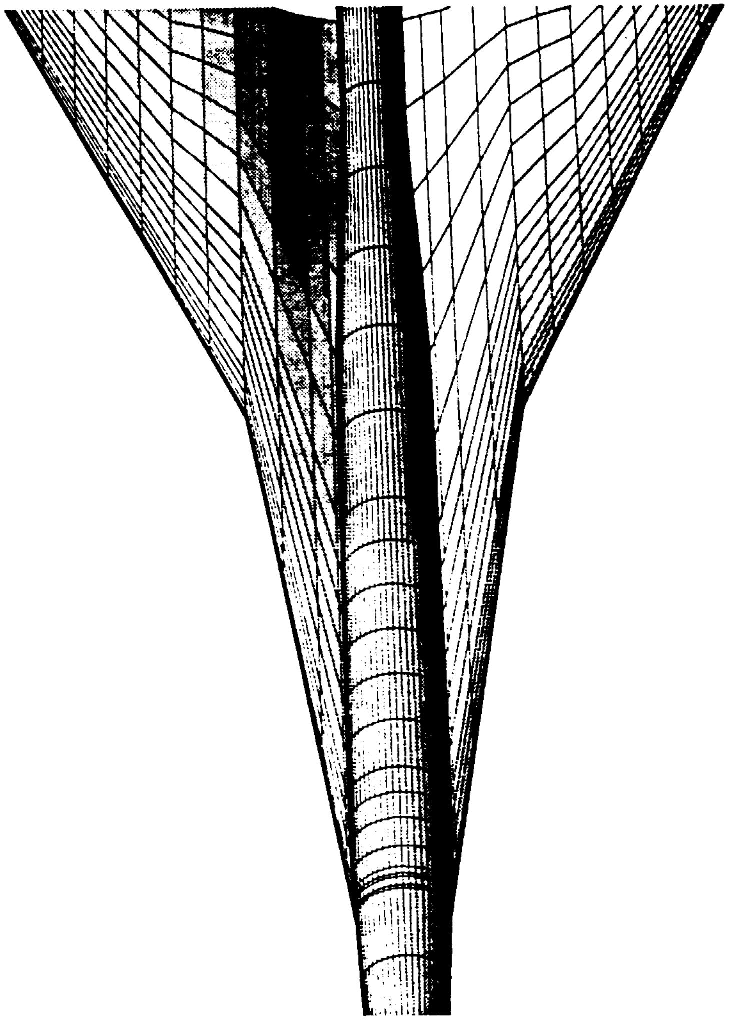
Points of Contact:

Tom Woodrow	Andrea Overman	Greg Follen
ARC	LaRC	LeRC
(415) 604-5949	(804) 864-5790	(216) 433-6193

WING FUSELAGE INTERSECTION EXAMPLE



Original Wave Drag Data



Automatically Relofted Geometry

Surface Definition Tool for HSCT Design & Optimization

Objective: Develop tools for automatically generating a smooth surface definition for a High-Speed Civil Transport (HSCT) vehicle as design variables are changed.

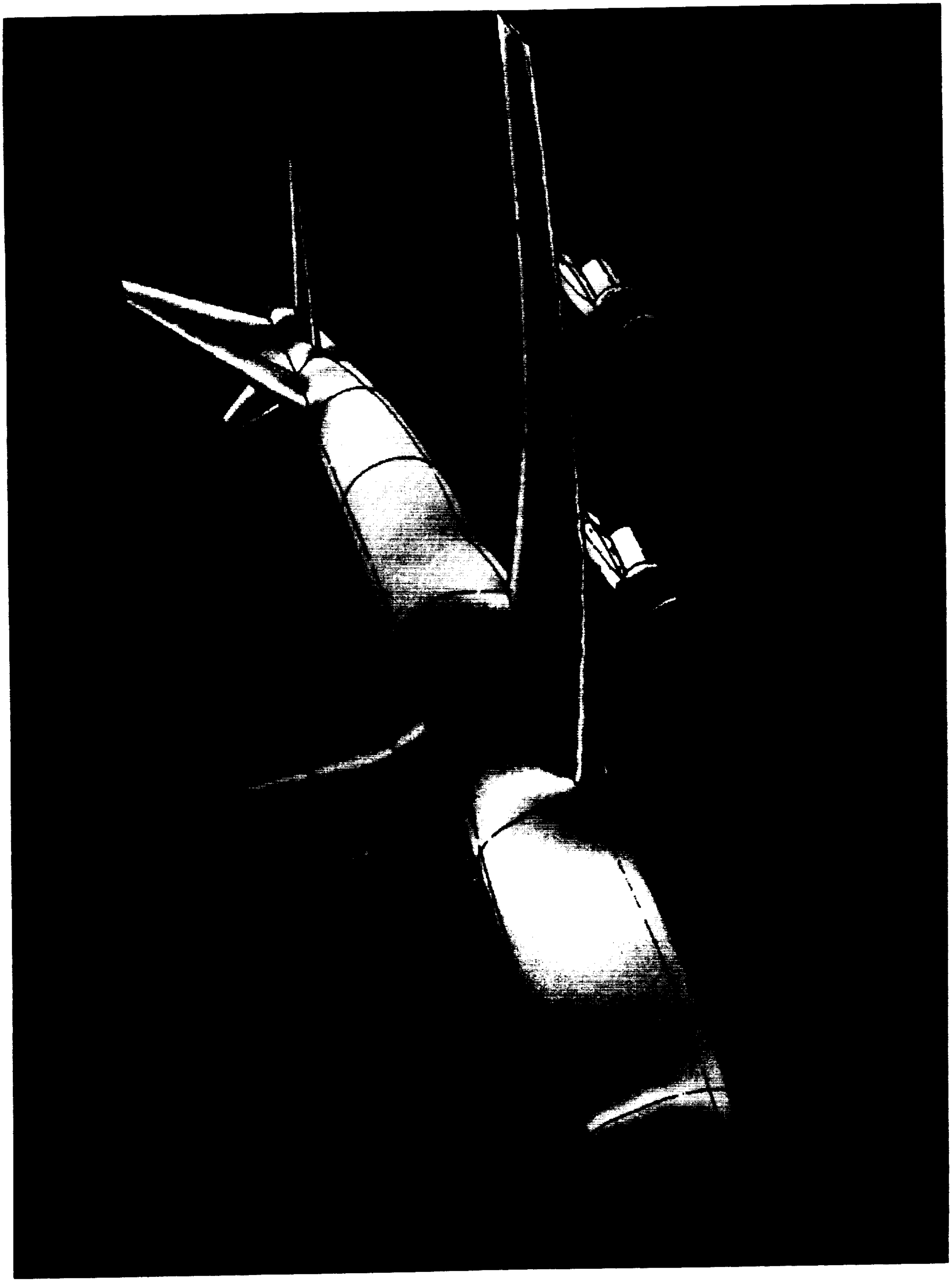
Approach: Develop semi-analytic tools for resolving the wing-fuselage intersection and adding an appropriate fillet. The tools start with the basic geometry description of the linear aerodynamics analysis codes employed in preliminary design and generate an analytic surface definition suitable for nonlinear aerodynamics analysis codes.

Accomplishments: The semi-analytic surface definition tool has been successfully applied to a variety of supersonic transport configurations. The input to the tool is a "wave drag deck", which is the standard geometry description used in preliminary design for this vehicle class. The output is the analytic description of the wing and fuselage, with proper intersections and fillets. The surface definition may be converted to a variety of standard descriptions, such as Hess or PLOT3D formats. The figure illustrates the results for a Mach 1.7 low sonic boom configuration. In the initial wave drag deck description, there is an obvious gap between the wing and the fuselage. This has been eliminated in the final description. The surface definition tool has been applied to several other configurations including the Mach 2.4 baseline, which is the focal point for the Langley HSCT Grand Challenge effort.

Significance: The design and optimization using nonlinear aerodynamics codes requires the ability to generate new surface definitions (and volume grids) automatically as the design variables are changed. This tool provides the capability for imbedding an automatic surface definition module in a design and optimization system for an HSCT.

Status/Plans: The tool is being augmented to meet the needs of computational structural mechanics codes. In addition, volume grids for computational fluid mechanics codes will be added. Several strategies for including the effects of design variable changes are being considered.

Raymond L. Barger, Mary S. Adams
Fluid Mechanics Division
Langley Research Center
(804) 864-2315



Research on Unstructured CFD tools for Efficient Use of MIMD Machines

Objectives: To modify a current explicit, adaptive, unstructured CFD code to run efficiently on MIMD machines using "off-the-shelf" software; to develop parallel load balancing algorithms to distribute efficiently the computational work among processors; to demonstrate on a practical commercial aircraft configuration the methodology developed; and to develop graphical tools to display the results from parallel computations.

Approach: The unstructured, adaptive, ALE, explicit edge-based Euler code FEFLO92 was modified and ported to the Intel MIMD machine environment. The PARTI-package, developed at ICASE, was selected and used as the "off-the-shelf" software for inter processor transfer. A new load balancing algorithm that is based on giving and taking elements at inter processor boundaries was proposed, studied and implemented.

Accomplishments: The FEFLO92 code was successfully ported to the Intel MIMD machine environment using the PARTI software for communication. A commercial aircraft configuration, the B-747, was computed using 32 processors. Some representative results are displayed in the figure on the facing page. The speeds achieved are on the order of 6Mflops per processor, which is to be expected. A new load balancing algorithm was implemented and tested on 2-D and 3-D configurations. Unlike recursive subdivision algorithms, this new algorithm is based on load deficit difference functions. Elements are exchanged along the boundaries of sub domains until a good work balance among the processors is achieved. It was found that the algorithm converges extremely fast, usually in less than 10 passes over the mesh. This implies a faster algorithm for machines with more than 2^{10} (= 1024 bytes) processors and makes it very attractive for applications that require dynamic load balancing (e.g., transient problems with adaptive h-refinement or remeshing, or particle-in-cell codes). A new set of graphical tools to display the results from parallel computations was developed. These tools allow for checking the partition of a domain among processors, the global or local results from simulations, and the grids employed.

Significance: This is the first time the PARTI software was used outside the developer's circle. Shortcomings and possible extensions were reported. The new load balancing algorithm should prove effective for applications that require dynamic load balancing (e.g., transient problems with adaptive h-refinement or remeshing, or particle-in-cell codes).

Status/Plans: Work continues on the implementation of the new load balancing algorithm on a parallel machine and on parallel adaptive h-refinement. Extensions of the parallel visualization tools are also planned.

Dorothee Martin and Rainald Lohner
CMEE/SEAS
The George Washington University
Washington, DC 20052
(202) 994-5945

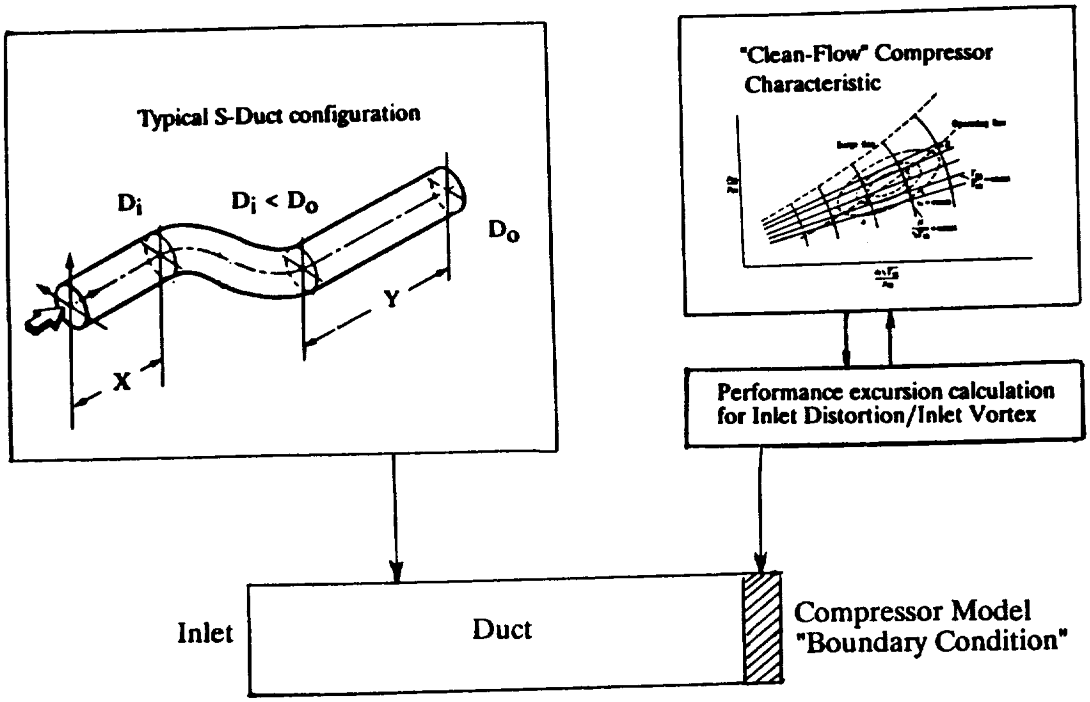

Typical S-Duct configuration
D_i
$D_i < D_o$
D_o
X
Y
"Clean-Flow" Compressor Characteristic
Performance excursion calculation for Inlet Distortion/Inlet Vortex
Inlet
Duct
Compressor Model "Boundary Condition"

Figure A. Relationship between key research elements to support integrated compressor-inlet module development.

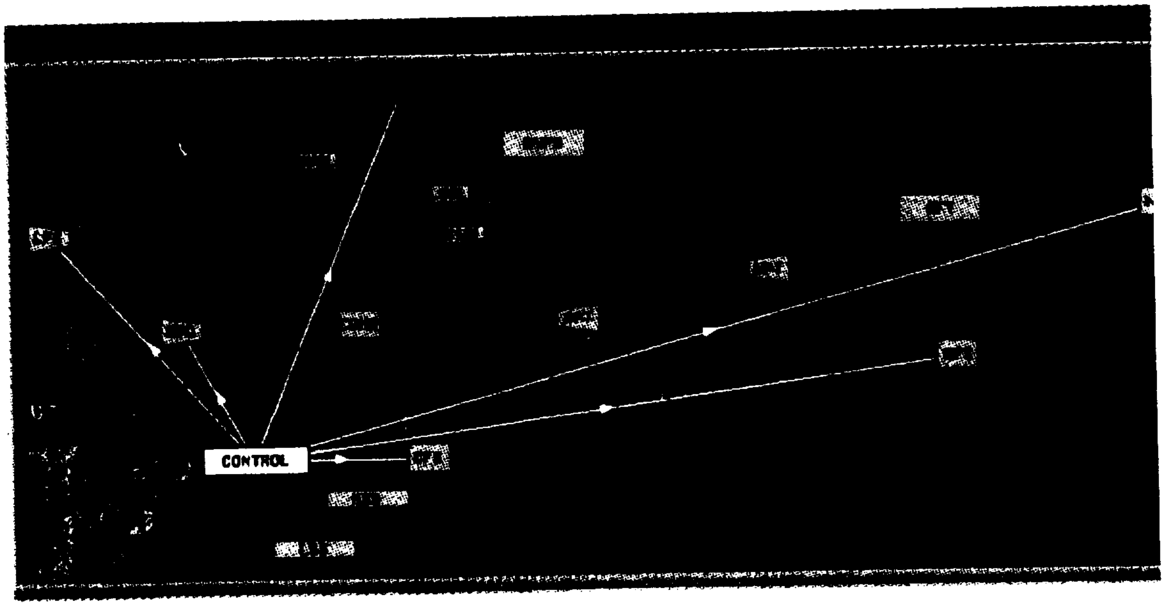


Figure B. Graphical user interface for prototype two-spool gas turbine object-based simulation.

Compression System Stability Analysis Methods for High Performance Aircraft Propulsion Systems

Objective: Develop three-dimensional models for gas turbine system stability characteristics and combine with an advanced compressor performance prediction method within an object-based simulation framework. Demonstrate an advancement in unified compression system simulation capability by predictive computation and stability assessment of a High Performance Aircraft compression system subject to distorted three-dimensional inlet flows. Perform research on compression system stability modeling, inlet modeling, and object-based simulation environments.

Approach: Diverse length and time scales associated with three-dimensional dynamic propulsion system compressor characteristics requires a research approach in three areas:

1) A mean-flow distortion transfer prediction for the compression system flow field based on new models for performance excursion calculations, which rely on Navier-Stokes solvers to predict the (undistorted) average-passage compressor characteristics.

2) Inlet flow field calculation technique incorporating the coupling of the compressor with the upstream flow.

3) An object-based real-time simulation environment to resolve miss-matched fidelity among the system simulation CFD, stability analysis algorithms, and (coupled) system solver algorithms.

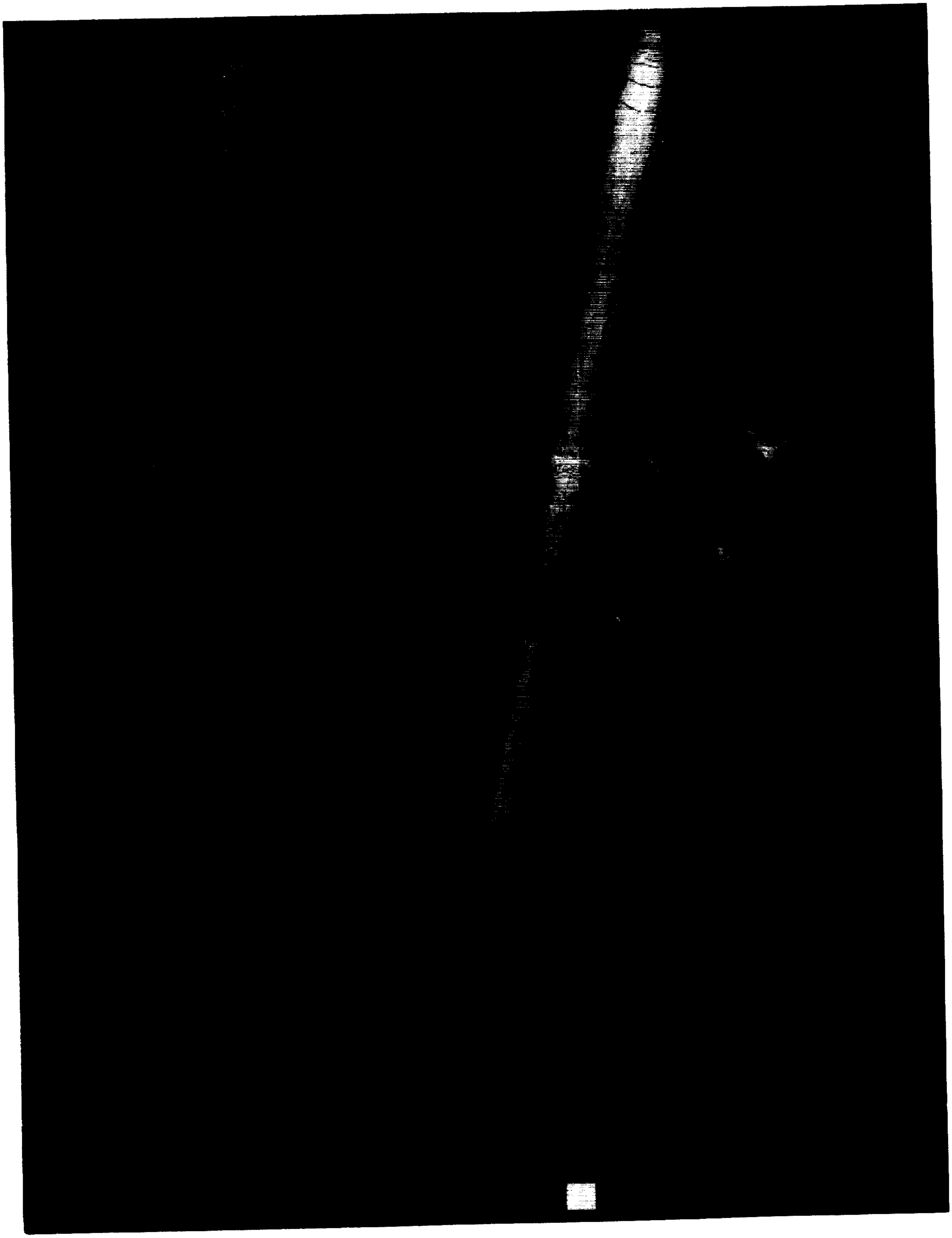
Accomplishments: In FY92 a prototype two-dimensional compressor stability code, based on the Moore-Greitzer-Longley approach, was ported to the Cray-YMP, and preliminary interface definitions were established for the clean-flow compressor characteristic and for the inlet model. These interface definitions are essential for integrating the compressor stability analysis modules with the prototype object-based simulation framework. Figure A illustrates the relationship between the key research elements of the compression system. Figure B illustrates the graphical user interface for the prototype object-based simulation recently completed; preliminary validation exercises have been completed. Key simulation features are the development of an object-based hierarchy, the definition of connector objects, and the successful integration of existing FORTRAN code into the Lisp language framework.

Several CFD codes were identified as candidates for inlet flow field predictions as part of the coupled compressor-inlet flow field calculation. An iterative approach to the coupled compressor-inlet flow field calculation was identified for preliminary system analysis technique development. A new research thrust was initiated to develop inlet modeling techniques that capture the essential fluid-dynamic flow characteristics without the CPU associated with a Navier-Stokes solver.

Significance: Simulation exercises with the prototype object-based simulation are the first-of-a-kind for a gas turbine application; successful (accurate) runs with the benchmark case constituted a critical step in demonstrating the viability of an object-based approach for gas turbine simulation. The simulation framework developed in this study can now be used to combine clean-flow compressor CFD codes, innovative inlet models, and the three-dimensional compressor stability analysis routines.

Status/Plans: Detailed object definitions for the stability analysis routines are being developed. Inlet modeling continues in conjunction with plans for completion of coding to implement the coupled compressor-inlet flow field calculation technique.

Colin K. Drummond
NASA Lewis Research Center
(216) 433-3956



The Design and Implementation of a Parallel Unstructured Euler Solver Using Software Primitives

Objective: The development of a set of optimized software primitives to facilitate the implementation of unstructured mesh calculations on massively parallel architectures.

Approach: A three-dimensional unstructured Euler solver was implemented on the Intel iPSC/860 and the Touchstone Delta machine using software primitives. To minimize solution time, we developed a variety of ways to reduce communication costs and to increase the processors' effective computational rate. The communication optimizations were encapsulated into the software primitives to ensure portability.

Accomplishments: The communication optimizations that were developed caused a reduction in the volume of data to be transmitted between processors and a reduction in the number of message exchanges. The single processor computational rate was increased two-fold by reordering the order of the computation, resulting in improved cache utilization and reduction in memory management overheads. We carried out a detailed study to evaluate the effects of our optimizations. The combined effect of these optimizations was a three-fold reduction in overall time. We ran a variety of test cases. The largest test case was of a highly resolved flow over a three-dimensional aircraft configuration. The free stream Mach number was 0.768 and the incidence was 1.116 degrees. We ran both an explicit solver and a V cycle multigrid solver. A sequence of four meshes was used for the multigrid calculations, with the finest mesh containing 804K mesh points. The same 804K mesh was also employed in the explicit solver. The explicit solver for this case achieved 778 Mflops on 256 processors on the Delta and 1.5 Gflops on 512 processors. The multigrid case achieved 1.2 Gflops on 512 processors. By comparison, the same code runs at about 100 Mflops on a single processor of a CRAY Y-MP and 250 Mflops on a single processor of a CRAY C90.

Significance: A set of highly optimized software tools was created which can be used to implement irregular computations on massively parallel machines. These tools can be used manually by both users and distributed memory compilers to automatically parallelize irregular codes.

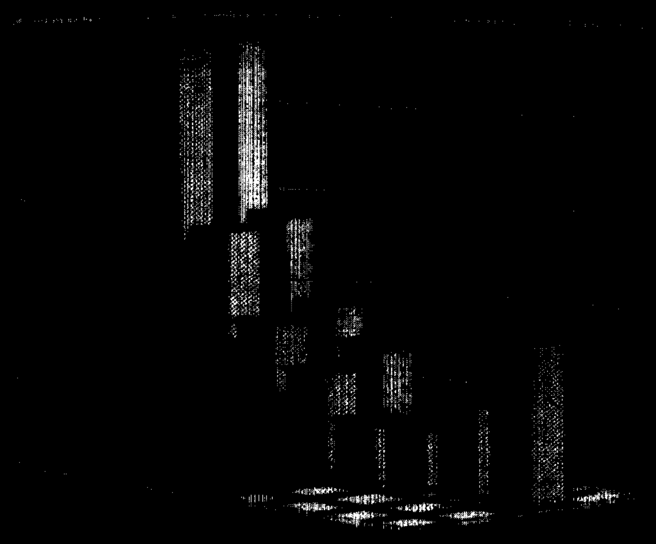
Status/Plans: To build a set of tools that can be used to parallelize adaptive irregular problems efficiently . These tools will be needed to develop an efficient unstructured multigrid based code to solve the Navier Stokes equations in three dimensions.

Joel Saltz
ICASE
NASA Langley Research Center
(804) 864-2210

Simulation Studies of Architectural Scalability

WBS 2.2 System Design Tools

- Accomplishments
  - Modeled a parallel version of ARC2D on ANL/BDE Simulator
  - Simulation results within a [illegible] of actual [illegible] measured on [illegible] Gamma



- [illegible] Program Focus
  - [illegible]
  - [illegible]

System Design Tools: Simulation Studies for Architectural Scalability

Objective: To understand the architectural approaches suitable for future generations of computer systems; to study the scalability of parallel CFD applications; and to develop/evaluate multiprocessor prototypes suitable for meeting future NASA requirements.

Approach: In order to understand the architectural app'oaches suitable for future generations of computing systems (of which teraFLOPS systems are a part), and to develop/evaluate multiprocessor prototypes suitable for NASA HPCC, the issue of scalability was addressed using AXE — a rapid prototyping/modeling environment developed at ARC. In FY92, the issue of whether representative HPCC applications can be implemented in a scalable fashion on highly parallel systems was studied based on simulations and calibrated by actual runs on testbed systems.

Accomplishments: In FY92, ARC2D, a representative application for HPCC/CAS, was modeled using AXE. We were able to predict its execution time on the Intel "Gamma" to within 7% in most cases.

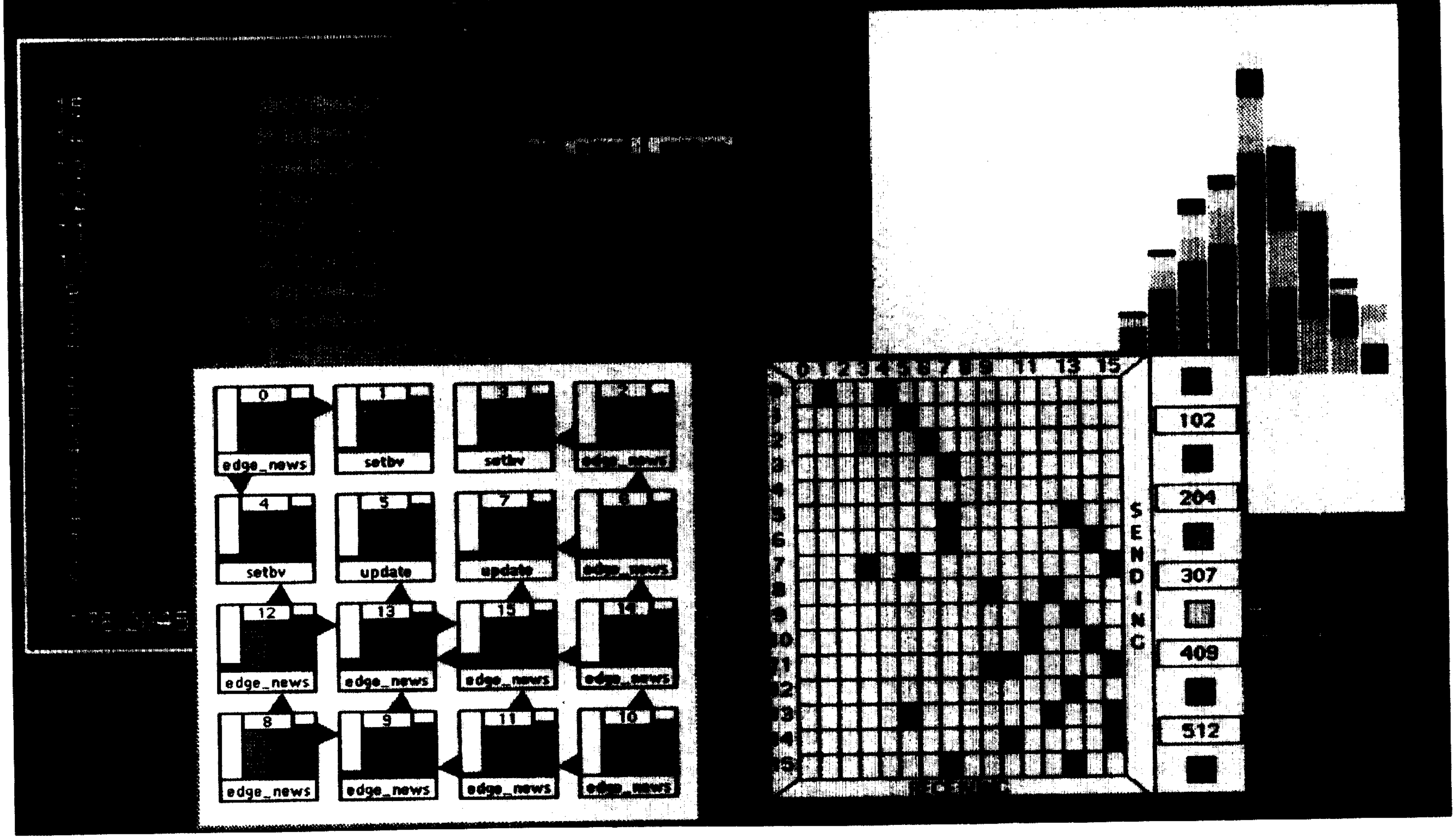
Significance: This simulation capability enables us to predict the performance of these applications on systems with different machine parameters (such as those having a larger number of nodes and faster routing systems) and to evaluate the scalability of both machines and applications

Status/Plan: We intend to project the performance of our ARC2D model on thousands of processors (connected as hyper cubes as well as 2D-meshes). Based on our simulation results, we hope to be able to determine software and hardware bottlenecks and vary hardware parameters to see how they affect performance.

Jerry C. Yan
Ames Research Center
(415) 604-4381

The Ames InstruMentation System

WBS 3.1 Software Components



Software Components and Tools: Instrumentation, Performance Evaluation and Visualization Tools

Objective: Investigate new techniques for instrumenting, monitoring and presenting the state of parallel program execution in a coherent and user-friendly manner; develop prototypes of software tools and incorporate them into the run-time environments of HPCC testbeds to evaluate their impact on user productivity.

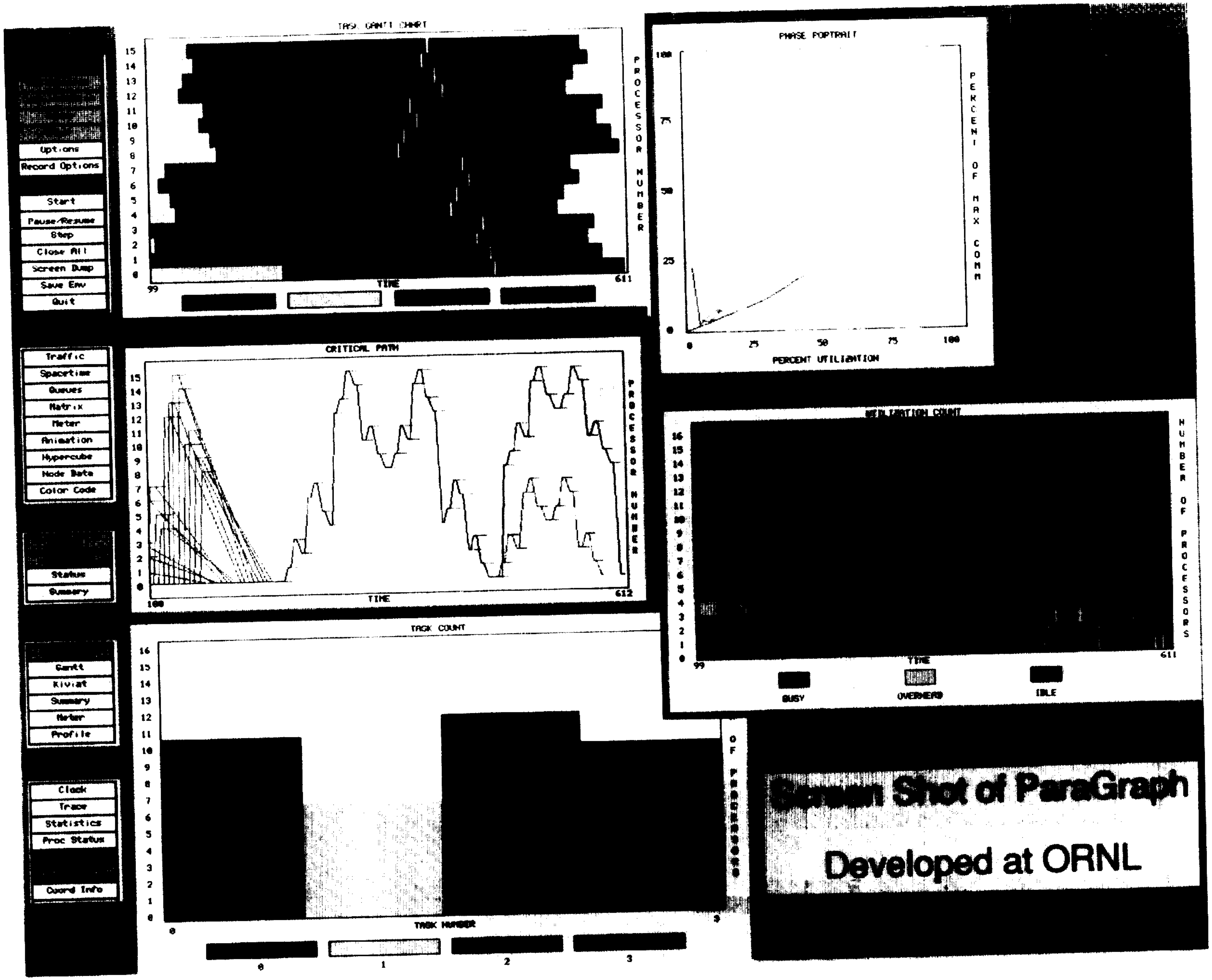
Approach: Debugging program performance on highly parallel and distributed systems is significantly more difficult than on more traditional systems because of multiple concurrent control strands and the asynchronous nature of execution. Therefore, research was undertaken to develop new techniques for monitoring and presenting the state of concurrent program execution in a coherent and user-friendly manner. Prototypes of software tools for monitoring execution were also developed to illustrate these concepts. These debugging and instrumentation facilities were incorporated into run-time environments to aid user productivity. Resource utilization monitoring techniques were also studied in order to help spot load imbalances and facilitate the development of dynamic resource management strategies.

Accomplishments: The Ames InstruMentation System (AIMS) was distributed to LaRC/ICASE, LeRC, ARC/RND; and U. of Michigan (Prof. D. Rover) and U. of Illinois (Prof. D. Reed) for teaching/evaluation. A Beta-version of AIMS currently accepts C programs and runs on the 512-node iPSC/Delta at Cal Tech. An intrusion compensation algorithm was also developed for AIMS to compensate for the overhead caused by instrumentation software.

Significance: This instrumentation capability provides users with detailed information about the program execution process to enable the tuning of their parallel applications on HPCC Testbeds.

Status/Plan: AIMS will be extended to permit CAS testbed users to visualize and tune the parallel execution of their applications. We plan to work closely with CAS application specialists at various NASA centers and identify tool features most useful for their work. Instead of taking the (unscalable) approach of "using 5 minutes to examine 5 milliseconds of execution", we will also develop monitoring and visualization methodologies/systems that will be more selective/intelligent about what data are to be collected and displayed. Although software instrumentation is much more flexible than hardware monitors, they perturb the application they are trying to monitor. Therefore, we will be exploring usage of hardware monitors available on future HPCC Testbeds.

Jerry C. Yan
NASA Ames Research Center
(415) 604-4381



Screen Shot of ParaGraph

Developed at ORNL

Parallel Programming Tools

Objective: To provide tools which will aid users in their development of efficient programs on parallel computers. Specifically, tools will be provided to assist program construction, debugging, and performance optimization.

Approach: Keep abreast of commercial tools and tools developed in national laboratories and universities, evaluate tools which appear likely to be useful to CAS users, and acquire and install the tools that are useful. Develop tools whose functionality is needed by the users but is not already provided by existing tools.

Accomplishments: Four communication libraries, APPL, PARTI, PICL, PVM, and Express were acquired and installed on the parallel systems. APPL provides communication primitives that are portable to shared-memory and message-passing computers. PARTI handles array references for irregular problems on a message-passing machine to give an appearance of shared memory. PICL generates execution traces for performance optimization. PVM supports distributed computing using heterogeneous computers. Express also provides tools for debugging and performance visualization. ParaGraph was made available to CAS users to visualize the trace data generated by PICL. ParaGraph's graphic animation and presentation of the events, the time spent in computation and communication, and the resource utilization can help users optimize program performance. An evaluation of the performance tools AIMS and PAT (supplied by Intel) lead us to undertake the development of APTview. APTview will provide summary information as well as the detailed history of individual processors. It will supply links between the visualization of the parallel activities and the source code which produces the visualized events.

Significance: Bringing tools in from outside significantly reduces the possibility of duplication in effort, enables CAS users to have tools at the earliest possible time, and enables tool developers to focus their attention on developing necessary tools that are not currently available.

Status/Plan: APPL, PICL, PARTI, PVM, and ParaGraph were made available to general CAS users. Express and AIMS will be available shortly; plans were made to acquire Forge 90 and Fortran Linda. Progress was made in designing APTview.

Duane Carbon
NAS Division, RND
NASA Ames Research Center
(415) 604-4413

Computation Time for Five Sensitivity Derivatives

NACA 1406 Airfoil; Transonic Inviscid Flow

Sensitivity Derivative Solution Procedure	Number Solutions	Time (Sec)*
Flow Equation, Divided Differences, AF Method	6 Nonlinear	36
Standard Form, Quasi-Analytical, Direct Method	1 Nonlinear, 5 Linear	38†
Flow Equation, Automatic Differentiation, AF Method	1 Nonlinear, 5 Linear	25

*All calculations performed on a Cray Y-MP computer.

†Average time for several matrix solution methods.

Automatic Differentiation of CFD Codes for Multidisciplinary Design

Objective: To apply emerging automatic differentiation (AD) methods and tools to existing high fidelity (advanced) computational fluid dynamics (CFD) codes in order to efficiently obtain aerodynamic sensitivity derivatives for use in multidisciplinary design-optimization methodologies.

Approach: Automatic differentiation exploits the fact that exact derivatives can be computed easily for all elementary arithmetic operations and intrinsic functions. The various forms of AD can be understood as particular methods of applying chain rule differentiation to the large number of operations defining the computation. The ADIFOR automatic differentiation tool, being developed at Argonne National Laboratories (ANL) and Rice University, augments a Fortran 77 code with statements for the computation of derivatives. In this particular implementation, derivatives are computed more or less in lockstep with the computation of the output function. Use of ADIFOR to obtain derivative information from existing CFD codes is being jointly investigated by NASA and ANL.

Accomplishments: Successful application of the ADIFOR tool by ANL personnel to an existing 2-D transonic small disturbance equation (TSDE) code provided by NASA demonstrated the versatility of this approach and resolved initial NASA concerns about "differentiating" the iterative implicit solution algorithms (with type-dependent operators) typically found in CFD codes. The original TSDE code, developed under NASA grant at Texas A&M University, also computed sensitivity derivatives using two other independent procedures: divided (finite) differences and quasi-analytical (hand-differentiated discrete) equations. All three procedures agree at subsonic and transonic flow conditions. At supersonic free-stream conditions, the quasi-analytical results disagree with those from the other two procedures. The table shows a comparison of the solution computation times for the above three solution procedures at high-subsonic flow conditions. The AD method is seen to be somewhat faster than the others.

Significance: In regard to computer science issues, this was the first application of the ADIFOR tool to obtain derivatives of functions determined by an iterative implicit solution algorithm (i.e., solutions of discretized, nonlinear, partial differential equations). In order to obtain reasonable computational times for derivatives from the AD processed code with such iterative implicit algorithms, a new paradigm for execution was required. In regard to using CFD in multidisciplinary design, the development time to obtain a code producing correct derivative information from an existing CFD code was significantly reduced from a man-year to a man-month.

Status/Plans: Application of ADIFOR to an advanced CFD 3-D configuration code was initiated. Use of ADIFOR generated derivative code in an incremental iterative form of the sensitivity derivative equation will be investigated.

Perry A. Newman, Lawrence L. Green, and
Kara J. Haigler
Fluid Mechanics Division
Langley Research Center
(804) 864-2247

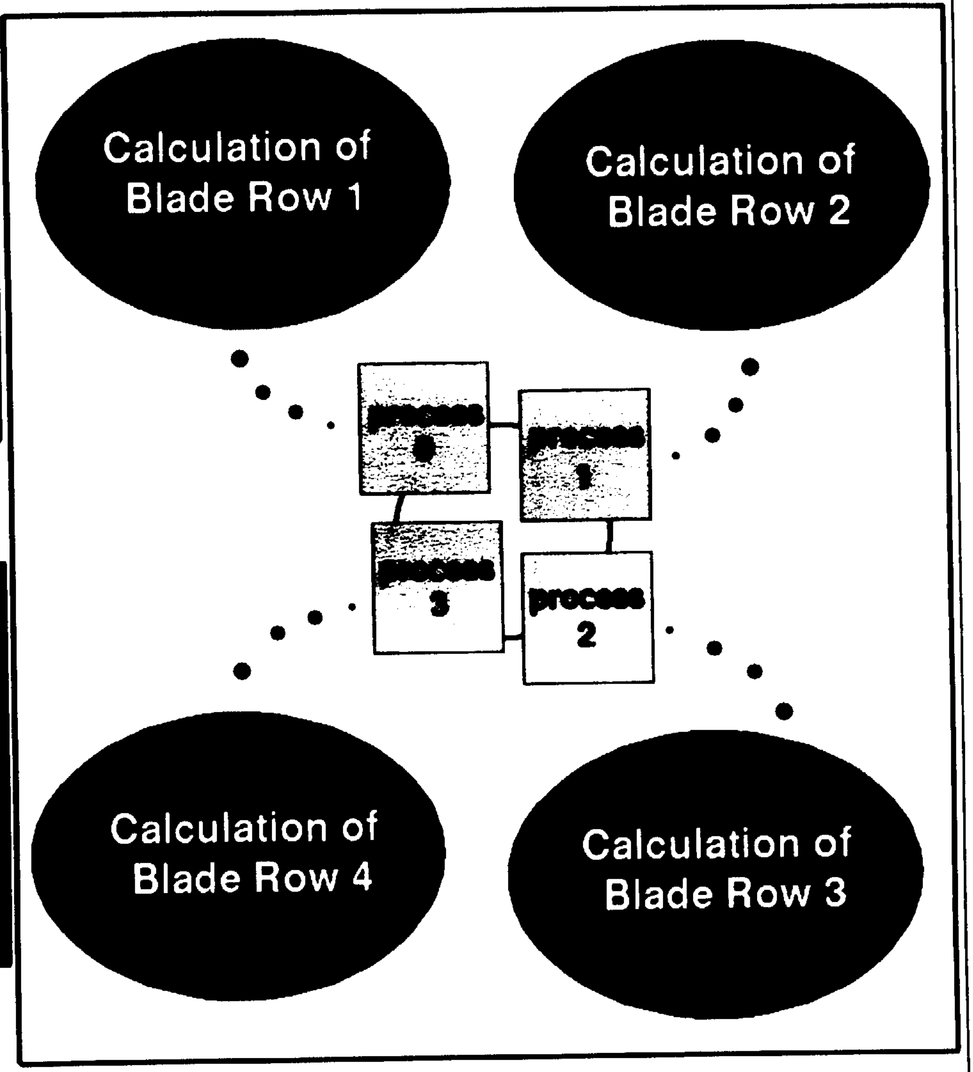
APPL Results for MSTAGE

IBM RS6000 Performance*		
Procs	Seconds	Speedup
1	20660	–
2	10912	1.89
4	6038	3.42

* 45 cycles

Hypercluster Performance **		
Procs	Seconds	Speedup
1	40185	–
2	20231	1.99
4	10272	3.91

** 15 cycles


Calculation of Blade Row 1
Calculation of Blade Row 2
process 0
process 1
process 3
process 2
Calculation of Blade Row 4
Calculation of Blade Row 3

Demonstration of a Portable Parallel Turbomachinery Code

Objective: To demonstrate the utility of the Application Portable Parallel Library (APPL) for porting large application codes to various multiprocessor systems and networks of homogeneous machines.

Approach: The Application Portable Parallel Library (APPL) was developed at LeRC as a tool to minimize the effort required to move parallel applications from one machine to another, or to a network of homogeneous machines. APPL was targeted to a number of distributed and shared memory, MIMD machines, and networked homogeneous workstations.

John Adamczyk's Average Passage Turbomachinery Code was chosen to demonstrate APPL's utility across a variety of multiprocessor systems and a network of workstations. The inviscid version of the code, ISTAGE, was demonstrated first on the Intel iPSC/860 and Delta machines, the Alliant FX/80, and the Hypercluster (a LeRC multiarchitecture testbed, using Motorola 88000 RISC processors). Investigation of the viscous version of the code, MSTAGE for multiple blade rows, was initiated. MSTAGE is particularly suited to run across a network of workstations because it does not require a large amount of communication between processes.

A shared memory version of MSTAGE was previously run on a Cray Y-MP. A message-passing version is being developed using APPL. Each processor is assigned a particular "blade row" to solve. This implementation is limited to running N blade rows on 1 to N processors. The current problem set-up uses a 218x31x31 mesh for each of the 4 blade rows. Restart files are used for communicating between blade row calculations. Processors are synchronized between iterations, so the correct versions of the restart files are read/ written.

The code was then moved to a network of IBM RS6000 workstations. A minor modification to the code was required because of a difference between the M88000 and RS6000 FORTRAN compilers.

Accomplishments: The MSTAGE code was run successfully across a network of IBM RS6000 workstations using APPL. The speedup achieved by running the four blade row case across four IBM RS6000 workstations was 3.42.

Significance: Using APPL across a network of workstations is a viable environment for developing and running applications. Also, the portability of an application code using APPL was demonstrated. A message-passing version of the MSTAGE code was developed on the Hypercluster and easily moved to a network of IBM RS6000 workstations. This demonstration has sparked an interest in using a distributed computing environment with other problem set-ups and other application codes.

Status/Plans: APPL has been developed for the Intel iPSC/860, the Intel Delta, the IBM RS6000 series of workstations, the SGI Iris4D series of workstations, the Sun Sparcstation series of workstations, the Alliant FX/80, and the Hypercluster. In the future, APPL may be ported to additional target machines and/or extended to allow computation across a network of heterogeneous machines.

Work also continues on refinement of the message-passing version of MSTAGE. Increasing speedup is under investigation.

Angela Quealy
Sverdrup Technology, Inc.
Lewis Research Center
(216) 826-6642

(THIS PAGE LEFT BLANK INTENTIONALLY.)

Introduction to the Earth and Space Science (ESS) Project

Goal and Objectives: The goal of the ESS Project is to accelerate the development and application of high performance computing technologies to meet the Grand Challenge needs of the U.S. Earth and space science community.

Many NASA Grand Challenges address the integration and execution of multiple advanced disciplinary models into single multidisciplinary applications. Examples of these include coupled oceanic atmospheric biospheric interactions, 3-D simulations of the chemically perturbed atmosphere, solid earth modeling, solar flare modeling, and 3-D compressible magnetohydrodynamics. Others are concerned with analysis and assimilation into models of massive data sets taken by orbiting sensors. These problems have significant because they have both social and political implications. The science requirements inherent in the NASA Grand Challenge applications necessitate computing performance into the teraFLOPS range.

The project is driven by three specific objectives:

□ Development of algorithms and architecture testbeds capable of fully utilizing massively-parallel concepts and scalable to sustained teraFLOPS performance;

□ Creation of a generalized software environment for massively parallel computing applications; and

□ Demonstration of the impact of these technologies on NASA research in Earth and space sciences physical phenomena.

Strategy and Approach: The ESS strategy is to invest the first four years of the project (FY92-95) in formulation of specifications for complete and balanced teraFLOPS computing systems to support Earth and space science applications, and the next two years (FY96-97) in acquisition and augmentation of such a system into a stable and operational capability, suitable for migration into Code S computing facilities. The ESS approach involves three principal components:

1) Use a NASA Research Announcement (NRA) to select Grand Challenge Applications and Principal Investigator Teams that require teraFLOPS computing for NASA science problems. Between four and six collaborative multidisciplinary Principal Investigator Teams, including physical and computational scientists, software and systems engineers, and algorithm designers, will address the Grand Challenges. In addition, 20 (10 initially in FY93) Guest Computational Invest–igators will develop specific scalable algorithmic techniques.

The Investigators provide a means to rapidly evaluate and guide the maturation process for scalable massively parallel algorithms and system software and to thereby reduce the risks assumed by later ESS Grand Challenge researchers when adopting massively parallel computing technologies.

2) Provide successive generations of scalable computing systems as testbeds for the Grand Challenge Applications; interconnect the Investigators and the testbeds through high speed network links (coordinated through the National Research & Education Network); and provide a software development environment and computational techniques support to the investigators.

3) In collaboration with the Investigators Teams, conduct evaluations of the testbeds across applications and architectures leading to downselect to the next generation scalable teraFLOPS testbed.

Organization: The Goddard Space Flight Center serves as the lead center for the ESS Project and collaborates with the Jet Propulsion Laboratory. The HPCC/ESS Inter-center Technical Committee, chaired by the ESS Project Manager, coordinates the Goddard/JPL roles. The ESS Applications Steering Group, composed of representatives from each science discipline office at NASA Headquarters and from the High Performance Computing Office in Code R, as well as representatives from Goddard and JPL, provides ongoing oversight and guidance to the project.

The Office of Aeronautics, jointly with the Office of Space Science and Applications, select is the ESS Investigators through the peer reviewed NASA Research Announcement process. The ESS Science Working Group, composed of the Principal Investigators chosen through the ESS NRA and chaired by the ESS Project Scientist, organizes and carries out periodic workshops for the investigator teams and coordinates the computational experiments of the Investigations. The ESS Evaluation Coordinator focuses activities of the Science Working Group leading to development of ESS computational and throughput benchmarks. A staff of computational scientists supports the Investigators by developing scalable computational techniques.

The ESS Project Manager serves as a member of the NASA wide High Performance Computing Working Group, and representatives from each Center serve on the NASA-wide Technical Coordinating Committees for Applications, Testbeds, and System Software Research.

Management Plan: The project is managed in accordance with the formally approved ESS Project Plan. The ESS Project Manager at GSFC and the JPL Task Leader together oversee coordinated development of Grand Challenge applications, high performance computing testbeds, and advanced system software for the benefit of the ESS Investigators. Monthly, quarterly, and annual reports are provided to the High Performance Computing Office in Code R. ESS and its Investigators contribute annual software submissions to the High Performance Computing Software Exchange.

Points of Contact:

Jim Fischer GSFC (301) 286-3465	Robert Ferraro JPL (818) 354-1340

Overview of ESS Testbeds

Goal and Objectives: The goal of the ESS testbeds activity is to assure that the development of high end scalable computer systems evolve in a direction leading to sustainable teraFLOPS for ESS applications. The objectives are to

□ Develop metrics for comparing and evaluating scalable parallel systems which measure their completeness and balance in light of ESS applications and which can be used to specify systems to meet NASA requirements through competitive procurement.

□ Provide useful feedback to the system vendors regarding the effectiveness and limitations of their products for performing ESS Grand Challenge applications in such a form that will help them to improve subsequent generations.

Strategy and Approach: Access to a wide variety of large scalable testbeds is required to stimulate the ESS Investigators to develop top notch Grand Challenge applications. These applications will serve as the source of a representative mix of parallel computational techniques and implementations. As these problems are formulated on particular parallel architectures and begin to stabilize into useful tools for the Investigators, they will be examined by project personnel to identify key computational kernel and data movement components. These key components will be recast in ways that make them portable to other scalable systems and instrumented so as to report important values during execution. They will be selected to cover and link the features of the architecture which make a significant contribution to end-to-end speed of execution. In this form, the key components will be run on different scalable systems as a suite of ESS parallel benchmarks which measure the performance envelope of each system. Access to preproduction and early serial number machines enables this activity to perform a pathfinder function.

Organization: Both GSFC and JPL manage and operate ESS owned testbeds onsite. JPL provides support to ESS Investigators for the Intel Delta at Caltech. In addition, GSFC has entered into a variety of arrangements with institutions which own, or are in the process of acquiring, large scalable testbeds. Most of these arrangements involve the exchange of NASA funds for machine access and user support. GSFC has established the position of Evaluation Coordinator, whose job it is to identify and develop the ESS parallel benchmarks.

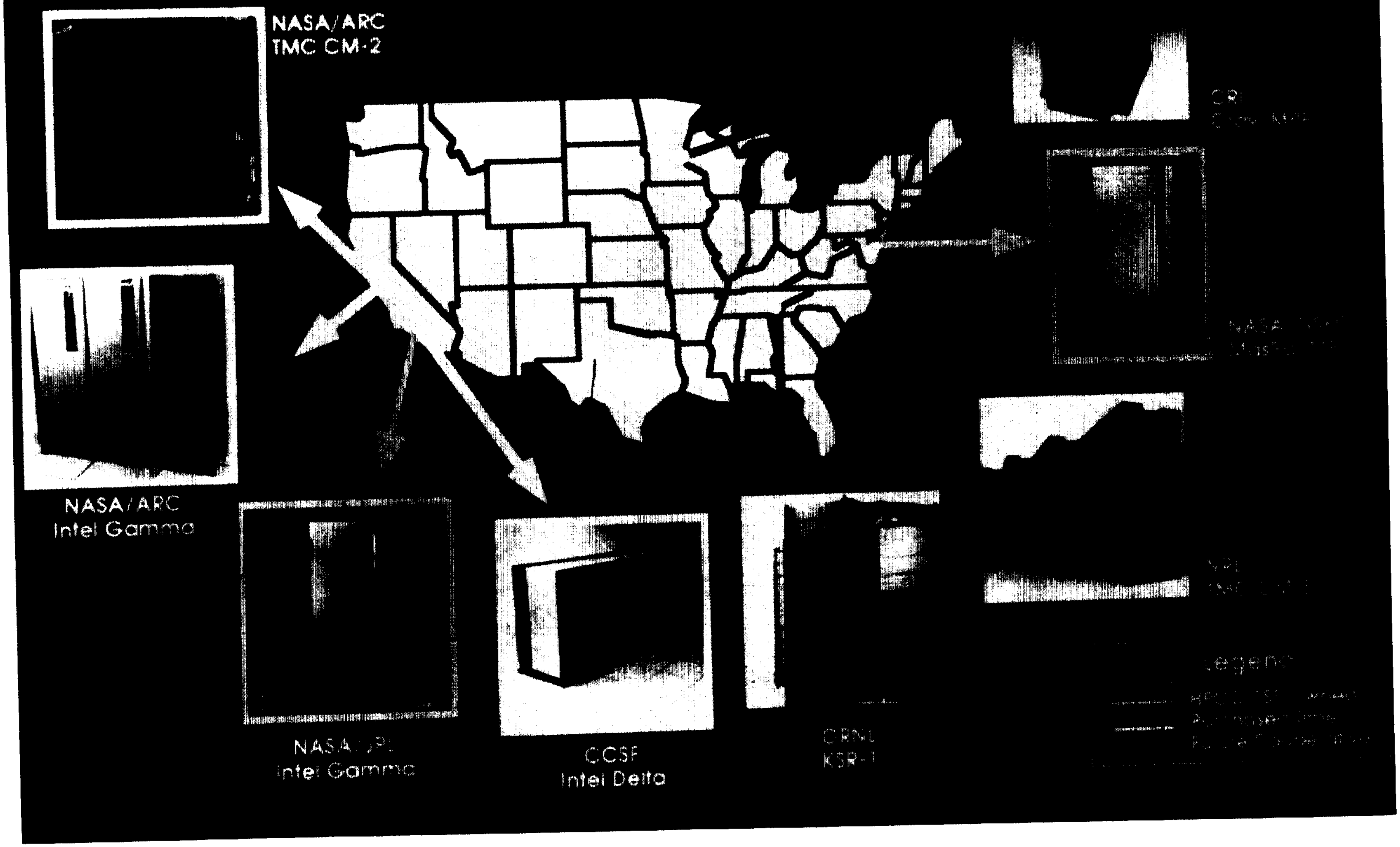
Management Plan: At GSFC, a Deputy Project Manager for Testbeds directs the in-house testbed activities and coordinates arrangements with other institutions for testbed access. At JPL, a Deputy Task Leader directs the in-house testbed activity and access to the Intel Delta. The Evaluation Coordinator reports to the ESS Project Manager.

Points of Contact:

Lisa Hamet
NASA Goddard Space Flight Center
(301) 286-9417

Jean Patterson
Jet Propulsion Laboratory
(818) 354-8332

HPCC/ESS Testbeds



Arranged Investigator Access to Large Scalable Testbed Systems

Objective: To obtain access for ESS NRA Investigators to large scalable testbed systems which have the potential to scale to teraFLOPS performance.

Approach: In order to acquire machine time, ESS establishes agreements with CAS centers and non-NASA research labs which own, or plan to own, large scalable systems. This approach leverages on the substantial capital investments already made by other organizations and also provides Investigators with the use of larger machines than NASA could afford to purchase. The acquisitions are selected to broaden the variety of systems available to ESS Investigations, over and above the ESS-owned testbeds, the MasPar MP-1 at GSFC, and the Intel Touchstone Gamma at JPL. As computer cycles on remote systems are obtained, ESS testbed managers work with the remote systems' administrators to establish working arrangements and to facilitate system access for ESS Investigators.

Accomplishments:

- ESS received and began to use a percentage of NASA's allotment of time on the Intel Touchstone Delta at Cal Tech.

- ESS received an allotment of computing time on the CAS-owned Intel Touchstone Gamma and Thinking Machines Corporation (TMC) CM-2, both located at ARC.

- ESS negotiated with Cray Research, Inc. for time on their Phase-1 Cray/MPP located in Eagan, MN, scheduled to be available in early FY93. In preparation, ESS purchased a Cray Y-MP/EL system and installed it at GSFC. On this system investigators will develop code and run the Cray/MPP cross compiler in preparation for remote execution.

- ESS reached agreement in principle with NRL to establish an Interagency Agreement to acquire time on their TMC CM-5.

- ESS reached agreement in principle with the Department of Energy to establish an Interagency Agreement to acquire time on the Kendall Square Research KSR-1 located at Oak Ridge National Lab.

Status/Plans: In FY93, relationships will be further developed with organizations supplying ESS with testbed cycles, and access to new systems will be actively pursued. The findings of the Investigators will assist the project toward acquisition of a GSFC resident technology refreshment testbed to be delivered in early FY94.

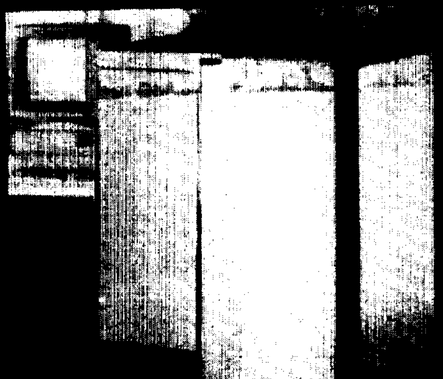
Significance: Exposing the Investigations to a wide variety of scalable systems assists the project to rule out weak contenders for objective reasons consistent with ESS requirements. It also enhances the Investigators chances of success in solving their Grand Challenge problems by increasing the likelihood that they can find an architecture well-matched to their problem. The larger sizes of these shared machines allow larger problems to be run, a factor which aids the system vendors by allowing the Investigations to test closer to the maximum potential of the machines and uncover weaknesses there, thereby spurring further development of the largest systems and hastening the ultimate construction of a teraFLOPS system.

Lisa Hamet
Goddard Space Flight Center
Code 930.7
(301) 286-9417

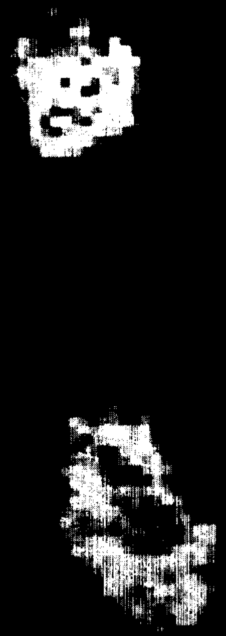
MasPar MP-1 Upgrade

The MasPar MP-1 at GSFC was upgraded from 8K processors with 128 MB memory to 16K processors with 1 GB memory in FY 92.

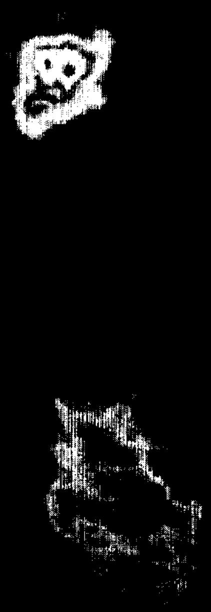
The upgrade allowed a Maximum Entropy Method deblurring algorithm to deblur a distorted HST image to twice the resolution previously possible by use of double precision floating point.



MasPar MP-1



HST Design Resolution



2X HST Design Resolution

Upgraded MasPar MP-I

Objective: To enable computation of larger sized problems and test code scalability, improve throughput, and allow true timesharing on the GSFC MasPar MP- 1.

Approach: ESS doubled the number of processors and raised the amount of aggregate memory in the MP-1 by a factor of eight to allow ESS scientists to expand problem sizes, examine the linearity of scale in performance, and more rigorously test MP-1 potential. Upgrade of the operating system facilitated true system timesharing.

Accomplishments: The GSFC MasPar MP-I was upgraded from 8,192 processors with 128 Mbytes of memory to 16,384 processors with 1 Gbyte of memory in FY92. This upgrade increased peak performance to 1.2 gigaFLOPS. The MP-1 host computer performance was increased by a factor of 9 and the parallel array disk speed by a factor of 5. GSFC also became a BETA test site for version 3.0 of the MasPar operating system. The most significant enhancement in version 3.0 is the capability for true timesharing. This is accomplished by automatically swapping long-running programs out to disk (on the MasPar parallel disk array) to allow waiting jobs to run, then swapping them back in.

Significance: These upgrades are critical in the preparatory effort for startup work by the ESS Investigators. A larger number of processors is always desirable; however, increasing the memory size by a factor of eight is a more vital improvement. ESS anticipates the Investigators' codes and data sets to be quite large; more memory will allow clearer and more quickly obtained results. The operating system enhancement appears to be significant. No longer does the MP-1 require enforcement of rules for maximum job length or a scheduling mechanism for long runs. It now operates in traditional multi-user fashion.

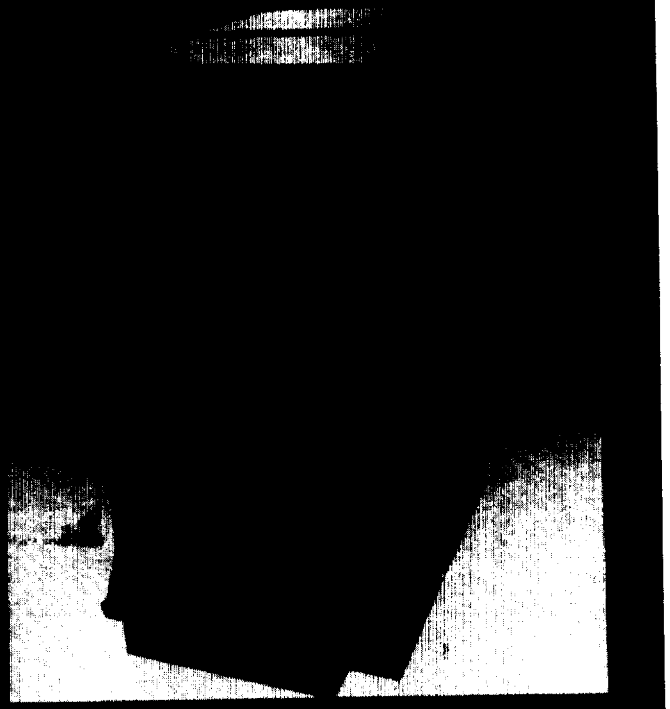
Status/Plans: ESS has received and installed version 3.0 PRODUCTION of the MasPar operating system, which will correct bugs in the BETA release. ESS scientists will continue ramping up on MasPar knowledge and experience in preparation for quickly bringing Investigators up to speed on the machine when they begin work in FY93. Also, in early FY93, ESS will be installing FDDI on the MP-1 to increase communication speed for remote Investigators.

Lisa Hamet
Goddard Space Flight Center
Code 930.7
hamet@nibbles.gsfc.nasa.gov
(301) 286-9417

Remote Access to Pre-production Cray/MPP In FY93

[illegible]

- 128 Processors
  - 19.2 GFLOPS peak
  - 256 Mwords memory



Cray/MPP in Minnesota

- Cray MPP software set
  - Cray Fortran and C

Established Remote Access to Pre-production Cray/MPP

Objective: To obtain access for ESS NRA Investigators to the Phase 1 Cray/MPP hardware as it is developed by the vendor.

Approach: ESS procured a 2 CPU Cray Y-MP EL with 256 Mbytes of memory and 10 Gbytes of disk space to serve as a development system for code to be run on the *First Phase* Cray Research, Inc. Cray/MPP, to be located in Eagan, MN, in FY93. ESS is the first group to be granted this early access. Codes will be compiled at GSFC on the EL, then run remotely via the on-sight CRI applications engineer. Initially, the codes will be run on the Cray/MPP emulator located in Eagan, MN, until a prototype system is available. Performance analysis tools will be available locally for users to run on the output received from the remote runs.

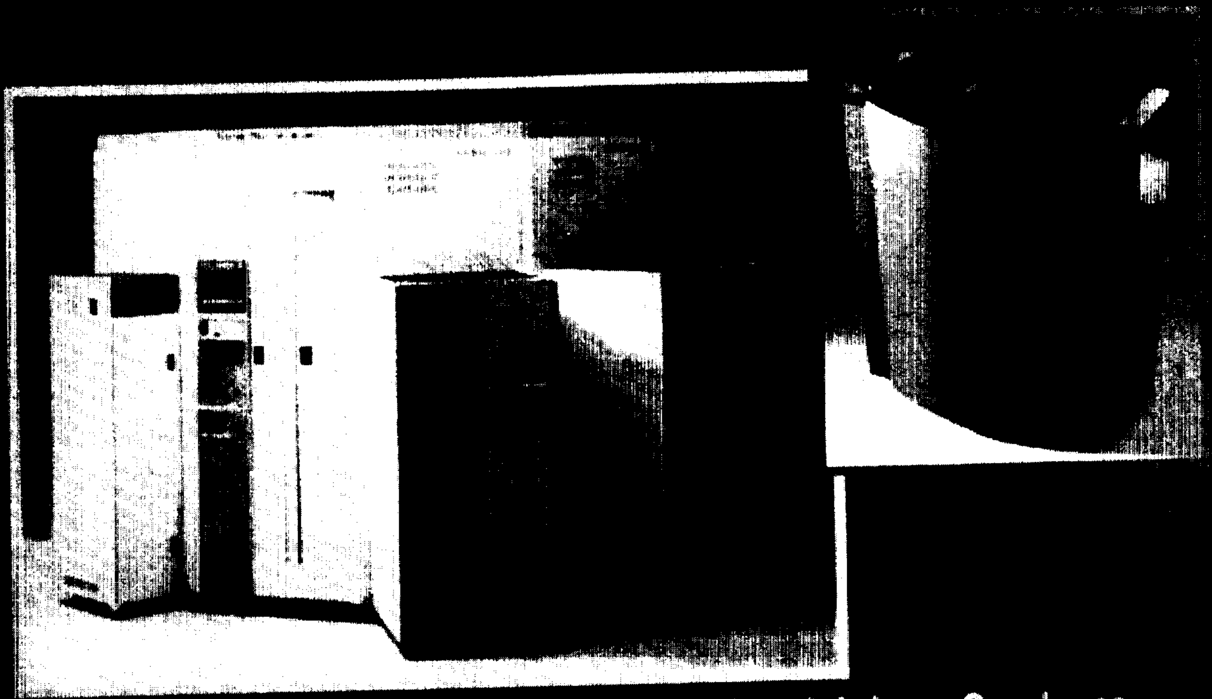
Accomplishments: The Cray Y-MP EL was installed at GSFC and has entered the acceptance period.

Significance: ESS Investigators and support staff will be the first group given access to the newest massively parallel, potentially teraFLOPS-scalable testbed currently in development. There are two equally important benefits. First, Investigators will be exposed to yet another architecture on which to tackle their Grand Challenge problems, and from which to give input into future ESS testbed procurements. Second, the Investigators will serve as a true test audience for Cray/MPP system software and software development tools. This scenario presents the greatest opportunity for HPCC input to a hardware vendor as the system is in development, so that lead time of implementing system enhancements and bug fixes may be near real-time. Thus, there is greater potential for a degree of software maturity, even in the initial release of the Cray MPP.

Status/Plans: ESS is considering trading in the Cray Y-MP EL for a small Cray/MPP system in early FY94 (scheduled release date for the product), provided that there is positive feedback from the ESS community during this remote access testing period. The initial Cray/MPP will not be standalone; a Cray Y architecture machine more advanced than the Y-MP EL will serve as the front end.

Lisa Hamet
Goddard Space Flight Center/Code 930.7
hamet@nibbles.gsfc.nasa.gov
(301) 286-9417

High Performance Data Management Testbed



[illegible] 2-256

StorageTek 4400 Automatic Cartridge System

- 1.2 terabytes — 6,000 longitudinally written 3240 tapes.
- in support of experiments in high speed data assimilation, data browsing and visualization.

Established High Performance Data Management Testbed

Objectives: Assure the availability of scalable high performance mass storage subsystems to meet the requirements of teraFLOPS computing. Demonstrate mass storage capacity, sustainable data rates, low latency, high reliability, and commercial availability which follow a line leading to the requirements of NASA flight missions.

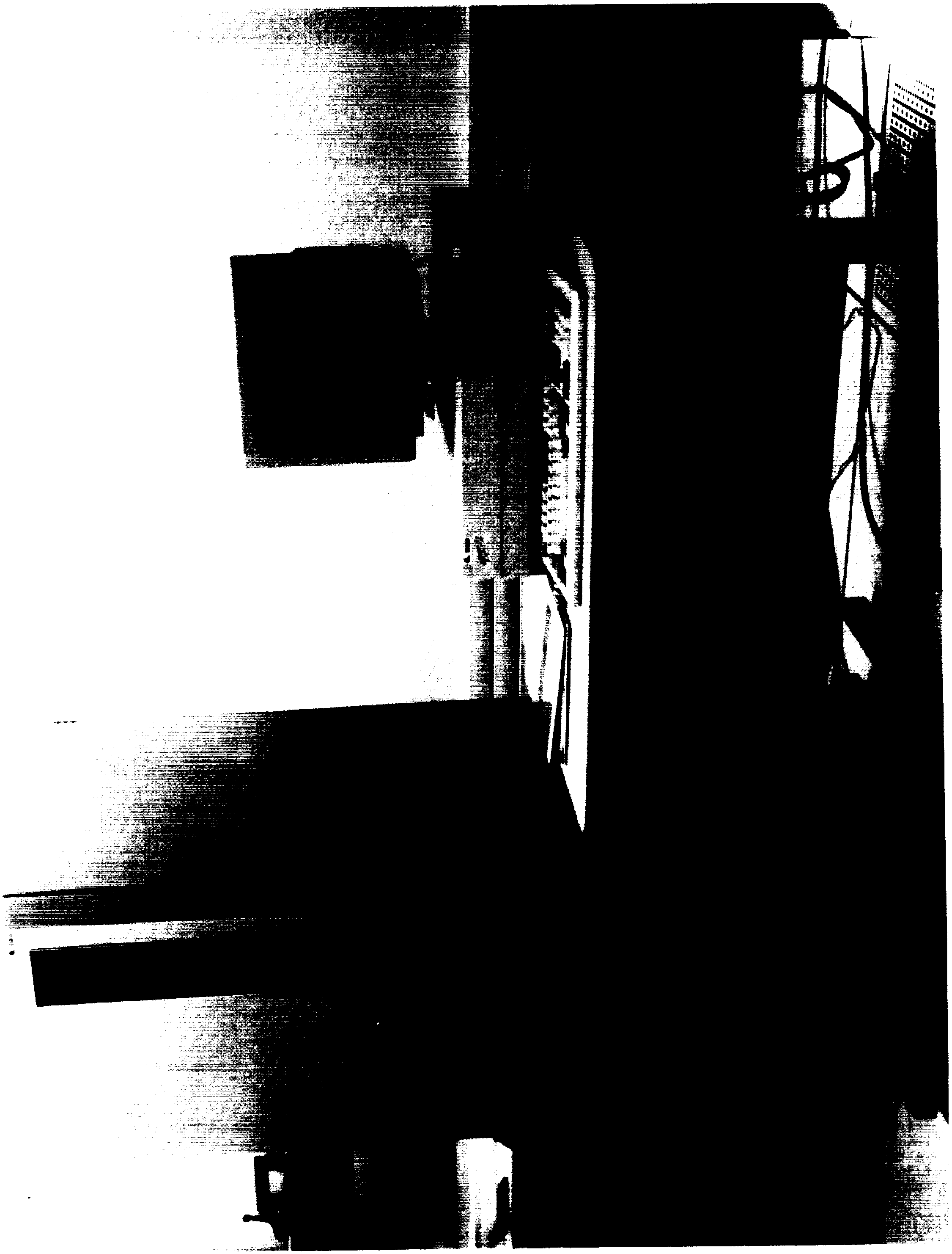
Approach: Establish an evolving testbed to host and drive experimental configurations for high performance mass storage and data management software tightly coupled to a scalable high performance computing system.

Accomplishments: ESS has established a high performance data management testbed at GSFC. A Storage Tek 4400 mass storage silo was acquired and connected to the Cray Y-MP EL to serve as an initial system in FY93. It has entered the acceptance period.

Significance: Assimilation of massive volumes of acquired data into running numerical models is essential to guide their accurate integrations. It is also a driving requirement of the ESS scalable system architecture, outside of CAS requirements. At regular intervals, acquired data is brought in from mass storage, formatted, superimposed on the model data in the processor memory, and assimilated into the model. Increases in the speed at which models run, multiplied by increases in the number of data sets from new complementary instruments which must be simultaneously loaded, project requirements for mass storage performance which dramatically exceed any current solutions. One recent study by Halem projected requirements for sustained rates in the gigabits per second range by 1998.

Status/Plans: The FY94 enhanced system will substitute the Cray/MPP system for the EL. Storage Tek is projecting 120 Terabytes per silo and continuous streaming at 15 megabytes per second within 2 years. ESS will work with the vendors to ensure continuing increases in the effective rates and capacities. ESS experiments envisioned to make use of this capability include high speed browsing and real-time reduction of data sets to allow visualization.

Lisa Hamet
Goddard Space Flight Center/Code 930.7
hamet@nibbles.gsfc.nasa.gov
(301) 286-9417



JPL Earth and Space Science Testbed

Objective: To establish and maintain a distributed memory Multiple Instruction - Multiple Data (MIMD) testbed for the Earth and Space Science (ESS) project. The function of the testbed is to provide early access to MIMD architecture computers, to the ESS Grand Challenge Investigators, and to ESS System Software researchers. The testbed will also function as a beta test-site for software products from industry and university research groups.

Approach: The testbed, located at JPL, is to provide a development environment for ESS Grand Challenge PI Teams and Guest Computational Investigators, including a Concurrent File System for parallel I/O research and development. Code developed on the testbed will be portable to the Concurrent Supercomputing Consortium's Intel Delta machine, located at the California Institute of Technology. Access to the testbed is via Internet connection.

Accomplishments: The JPL ESS testbed was established in February of 1992. It currently consists of an Intel iPSC/860 Gamma computer, with 16 processor nodes, 3 I/O nodes, and cross-compiling platforms. Each processor has 16 Megabytes of main memory. The processors are connected via a hypercube topology. Each I/O node has 1.2 Gigabytes of disk space attached. The 3.6 Gigabytes of disk space may be accessed in parallel by applications running on the testbed via Intel's Concurrent File System (CFS) software. The operating system is Intel's NX message passing OS. Beta test-site agreements have been established for several Intel software products: DGL, a distributed graphics programming library; C++; and Prosolver DES, an out-of-core dense matrix solver. Access to this software for testing and evaluation is available to all testbed users.

Significance: This testbed provides an environment for ESS researchers to develop new computational methods for MIMD architectures, test and evaluate pre-release versions of software products, and develop codes to address ESS Grand Challenge computing problems.

Status/Plans: The testbed will be upgraded over the life of the ESS project as new technology becomes available, so that ESS researchers will continually have access to the latest advances in MIMD hardware and software.

Robert Ferraro
Observational Systems Division
JPL California Institute of Technology
(818) 354-1340

Intel Touchstone Delta System Testbed

Objective: To provide NASA HPCC researchers access to one of the most advanced high performance computing platforms available for research and development of codes to address NASA Grand Challenge computing problems in Computational Aeroscience (CAS), Earth and Space Sciences (ESS), and Remote Explorations and Experimentation (REE).

Approach: NASA is a member of the Concurrent Supercomputing Consortium (CSCC), with an 11.9% share of the Intel Touchstone Delta. The Delta is the largest Multiple Instruction-Multiple Data parallel computer available today, with 520 processor nodes, 32 I/O server nodes with a total of 90 Gigabytes of Concurrent File System storage, 2 User Service nodes, a HIPPI node, and a hierarchical file storage system. The processor, I/O, service, and HIPPI nodes are organized into a 16 by 36 mesh. Sustained computational rates in excess of 10 Gigaflops have been demonstrated on some applications. As a consortium member, NASA HPCC researchers have access to the Delta for code development and production runs. Delta time is allocated through the NASA HPCC projects.

Accomplishments: Researchers at the five NASA centers that are participating in the HPCC Program have access to the Delta System. A number of NASA's Grand Challenge problems are currently being run. These problems include structural modeling for the High Speed Civil Transport, turbulence simulations, numerical propulsion simulations, planetary imaging of Venus, planetary rover stereo vision computation, helioseismology studies, compressible magnetohydrodynamics convection, particle simulations of the solar wind termination shock, and electromagnetic scattering and radiation analysis.

Significance: NASA HPCC scientists have demonstrated the usefulness of this architecture on CAS, ESS, and REE science applications, obtaining some of the highest performance numbers for scientific codes on any general purpose computer in existence. NASA's share of the Delta is fully subscribed, with HPCC researchers eagerly awaiting the next generation CSCC machine.

Status/Plans: The CSCC is in its second year of existence and anticipates the acquisition of a new generation of high performance computer this year.

Jean Patterson
NASA Delta Coordinator

Julie Murphy
NASA Delta Administrator

JPL/California Institute of Technology
(818) 354-8332

(THIS PAGE LEFT BLANK INTENTIONALLY.)

Overview of ESS Applications Software

Project Goal and Objectives: The goal of the ESS applications software activity is to enable the development of NASA Grand Challenge applications on those computing platforms which are evolving towards sustained teraFLOPS performance. The objectives are to

□ Identify computational techniques which are essential to the success of NASA Grand Challenge problems,

□ Formulate embodiments of these techniques which are adapted to and perform well on highly parallel systems, and

□ Capture the successes in some reusable form.

Strategy and Approach: The strategy is to select NASA Grand Challenges from a vast array of candidate NASA science problems, to select teams of aggressive scientific Investigators to attempt to implement the Grand Challenge problems on scalable testbeds, and to provide institutionalized computational technique development support to the Investigations in order to accelerate their progress and to capture the results. The approach involves use of the peer-reviewed NASA Research Announcement as the mechanism to select the Grand Challenge Investigations and their Investigator teams. In-house teams of computational scientists are developed at GSFC and JPL to support the Investigations.

Organization: The Office of Aeronautics, jointly with the Office of Space Science and Applications, select the ESS Investigators through the peer reviewed NASA Research Announcement process. The ESS Science Working Group, composed of the Principal Investigators chosen through the ESS NRA, and chaired by the ESS Project Scientist, organizes and carries out periodic workshops for the investigator teams and coordinates the computational experiments of the Investigations. The ESS Evaluation Coordinator focuses activities of the Science Working Group leading to development of ESS computational and throughput benchmarks. A staff of computational scientists supports the Investigations by developing scalable computational techniques.

Management Plan: At GSFC, a Deputy Project Manager for Applications directs the in-house team of computational scientists. At JPL, a Deputy Task Leader performs the same function. ESS and its Investigators contribute annual software submissions to the High Performance Computing Software Exchange.

Points of Contact:

Steve Zalesak
Goddard Space Flight Center/Code 930.7
zalesak@gondor.gsfc.nasa.gov
(301) 286-8935

Robert Ferraro
Jet Propulsion Laboratory
ferraro@zion.jpl.nasa.gov
(818) 354-1340

Selection of ESS Grand Challenge Investigations

Using the NASA Research Announcement Process

4-6 Principal Investigator Teams
Develop Grand Challenge Applications
10-20 Guest Computational Investigators
Develop Scalable Algorithm Techniques

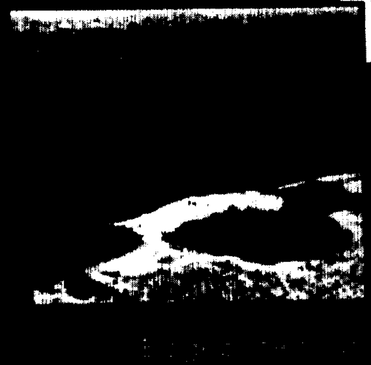
NRA release
February

Proposals due
May

Peer Review
July

Technical
Review
Committee
August

Announcement
of Selections
September



Selected ESS Investigations Using the Peer Reviewed NASA Research Announcement

Objective: Select NASA Grand Challenge scientific Investigators who will provide a means to rapidly evaluate and guide the maturation process for scalable parallel algorithms and system software and to thereby reduce the risks assumed by later ESS Grand Challenge researchers when adopting similar technologies.

Approach: Issue a NASA Research Announcement internationally requesting proposals for Grand Challenge Investigations. The breadth of the NRA includes all NASA science. Select between four and six collaborative multidisciplinary Principal Investigator Teams including physical and computational scientists, software and systems engineers, and algorithm designers. In addition, select between ten and twenty Guest Computational Investigators to develop specific scalable algorithmic techniques. Bring the selected teams under award and form them into a Science Working Group to organize computational experiments to be jointly carried out.

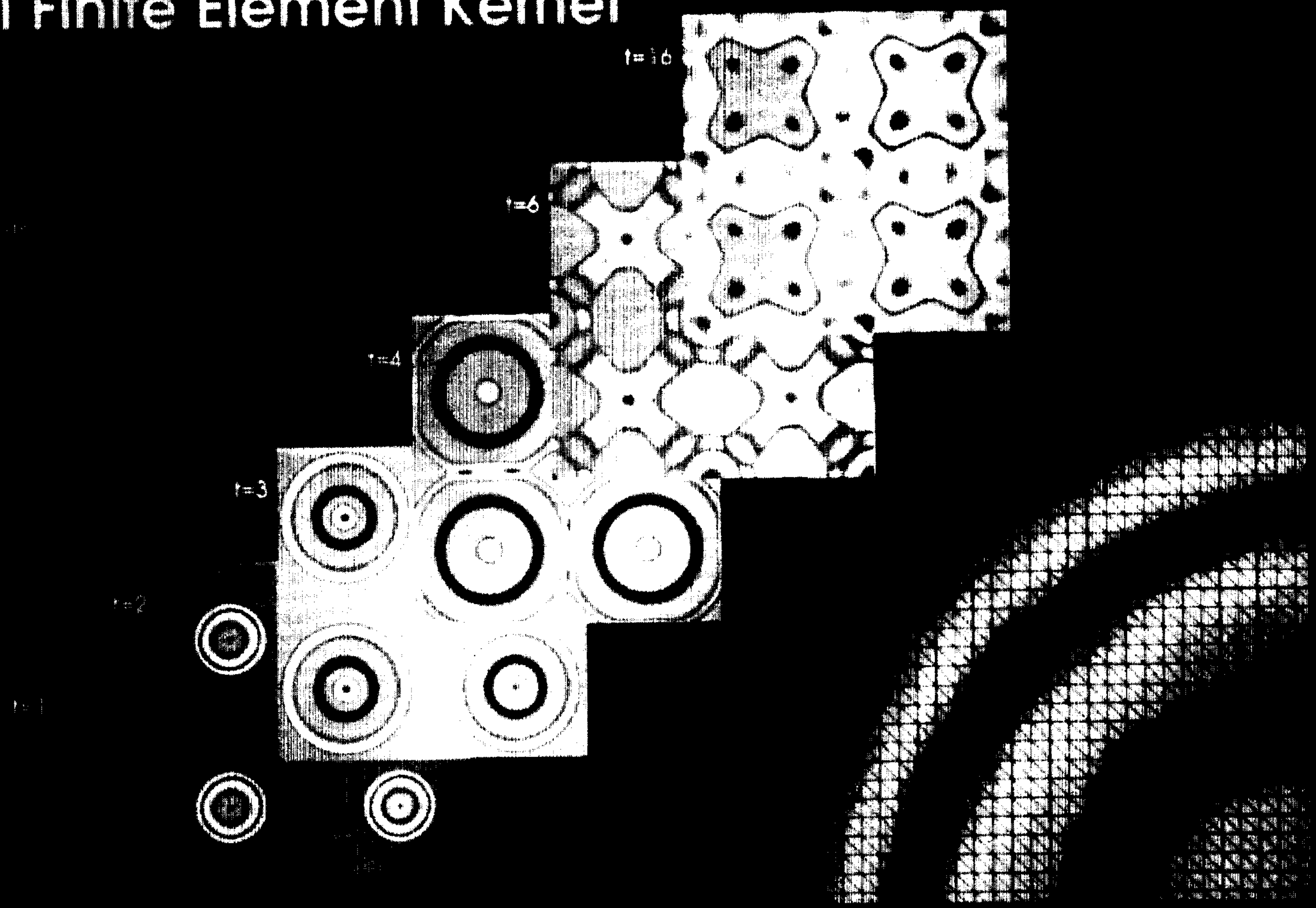
Accomplishments: The NASA Research Announcement was written, and formal approval for its release was secured jointly from the Associate Administrators of the Office of Space Science and Applications (OSSA) and The Office of Aeronautics (OA). The NRA was released in February, a preproposal conference was held in March, 208 proposals were received in May, 608 peer reviews were received in July, the 31 member Technical Review Panel ranked the proposals in August, and the Selection Committee made its recommendations to the Selection Official in September.

Significance: The collaboration between OSSA and OAST that has developed through the entire NRA process has greatly strengthened the entire ESS Project and holds the promise of keeping the Code R ESS activity highly relevant to the OSSA science community.

Status/Plans: Funding for Year-1 of the selected awardees will be established early in FY93. User support, training, and technique development support will be provided for the Investigators during the year allowing them to complete their first annual evaluations and reports at the end of FY93. The first two Science Working Group workshops will be held in FY93.

Jim Fischer
Goddard Space Flight Center
Code 930.7
fischer@nibbles.gsfc.nasa.gov
(301) 286-3465

Parallel Finite Element Kernel



Developed Parallel Finite Element Kernel

Objective: Create an environment in which to examine all aspects of the parallelization of unstructured grid finite element codes without the overhead and inertia associated with development of a production code.

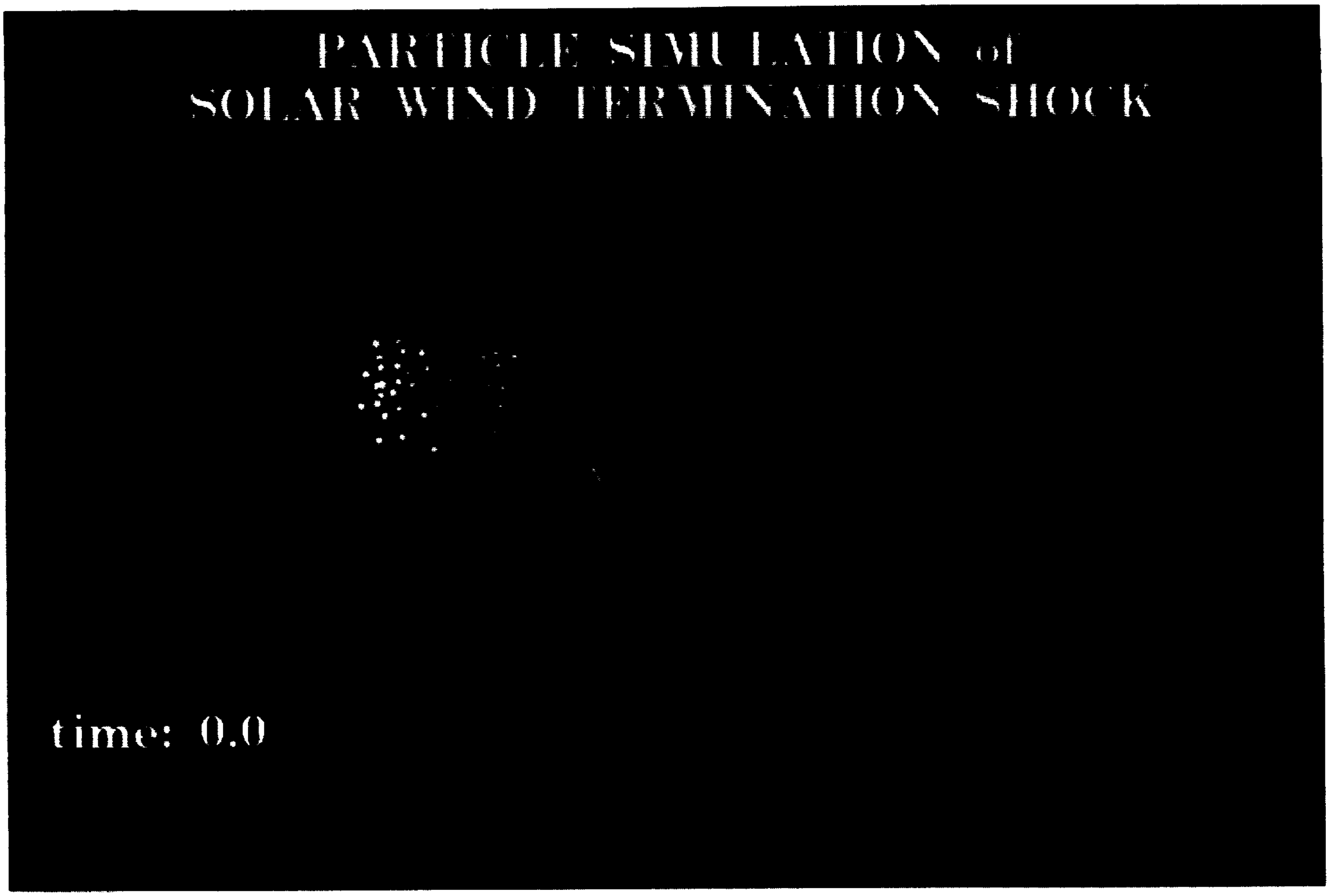
Approach: Rewrite an unstructured grid unsteady compressible gasdynamics code from scratch. This approach 1) takes advantage of the experience and knowledge of parallel algorithm specialists that has been accumulated so far to structure and optimize the code and data layout for speed, and 2) structures the code in such a way that the unknown factors are easy to explore, creating a learning tool for the computational scientist and for students such as those in the NASA Summer School in High Performance Computational Sciences. Extensive efforts were made to keep the code simple so that it can be easily rewritten from scratch on a variety of systems and with a variety of languages.

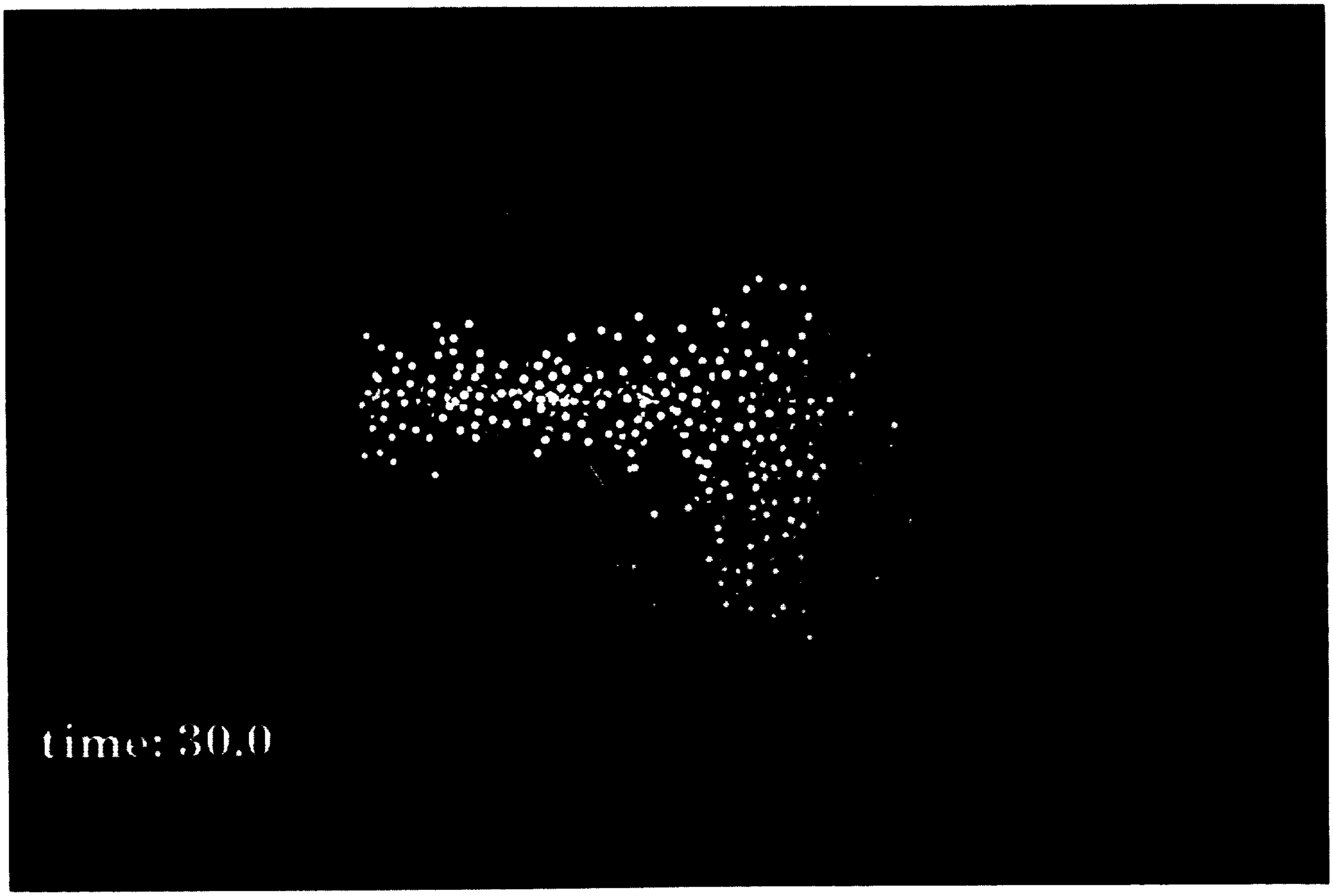
Accomplishments: Implemented a simple two dimensional finite element code on the 8,192 processor MP-1, consisting of 16,384 elements and 8,192 nodes. It ran at about 10 iterations per second using only MasPar's MPL 'C' language. Then it was speeded up to 69 iterations per second by coding the communications routines in assembler. This is 1.5 times faster than the same code running on a single Cray YMP processor. A completely parallel version of the unstructured grid finite element code was produced and runs on the MP-1. Simple numerical dissipation terms were also added in preparation for upwinding based on characteristic decomposition. A remote pipe was set up from the MP-1 to SGI workstations, enabling visualization in real-time using X-Windows.

Significance: In early FY92, a 2-D version of an unstructured mesh finite element hydrodynamics code (developed by R.Lohner/GWU and currently operating in 3-D on a Cray) was ported to the MasPar MP-1 by rewriting the key computational kernels in the MasPar MPL language. It ran 25 times slower than the same code running on a single Cray YMP processor. This process showed that taking code and trying to port it to the scalable parallel system was not only an arduous exercise, but resulted in poor performance. This process punctuated the need to totally rewrite the application from scratch with the considerations of the parallel architectures in mind.

Status/Plans: In FY93, the kernel will be written for additional machines of interest such as the Touchstone series, the KSR-1, the CM-5, and the Cray/MPP. The simple structure of the kernel will allow the code to be rewritten in an unconstrained manner, taking advantage of the varying features of the programming environments available on each of these systems.

Steve Zalesak
Goddard Space Flight Center
Code 930.7
zalesak@gondor.gsfc.nasa.gov
(301) 286-8935


PARTICLE SIMULATION of
SOLAR WIND TERMINATION SHOCK
time: 0.0


time: 30.0

Plasma Particle Simulations of the Solar Wind Termination Shock

Objective: Through implementation of a hybrid plasma particle-in-cell (PIC) simulation code, to study the effect of energetic ions on the structure of the solar wind termination shock (where the solar wind flow is reduced from supersonic to sub-sonic as a result of its interaction with the interstellar medium) and its potential for accelerating the energetic ions to cosmic ray energies.

Approach: In plasma hybrid PIC codes, the orbits of thousands to millions of plasma ions are followed in self-consistently computed electromagnetic fields. The electrons are treated as a conducting fluid. The ions can be anywhere, but the field equations are solved on a discrete grid. At each time step, first, the position and velocities of the particles are updated by calculating the forces on them by interpolating the field values at the grid points; second, the updated fields and electron variables are found by solving the field and fluid equations on the grid using the ion density and the fluid velocity. A hybrid PIC code has been implemented on the DELTA using the General Concurrent Particle-In-Cell Algorithm (GCPIC) (Liewer and Decyk 1989). With GCPIC, the physical domain of the particle simulation is partitioned into sub-domains and each assigned to a processor. The partitioning leaves each sub-domain with roughly equal numbers of particles.

When particle densities are non-uniform, these sub-domains will have unequal physical sizes. Each processor maintains the list of assigned particles and carries out necessary calculations associated with these particles. When particles move to new sub-domains, they are passed to the appropriate processor.

Good load balancing is maintained as long as the number of particles per sub-domain is approximately equal. When this condition becomes false, the load is dynamically re-balanced by repartitioning of the simulation domain. To study the effect of energetic solar wind pickup ions on the shock structure, two separate ions components are followed in the simulations.

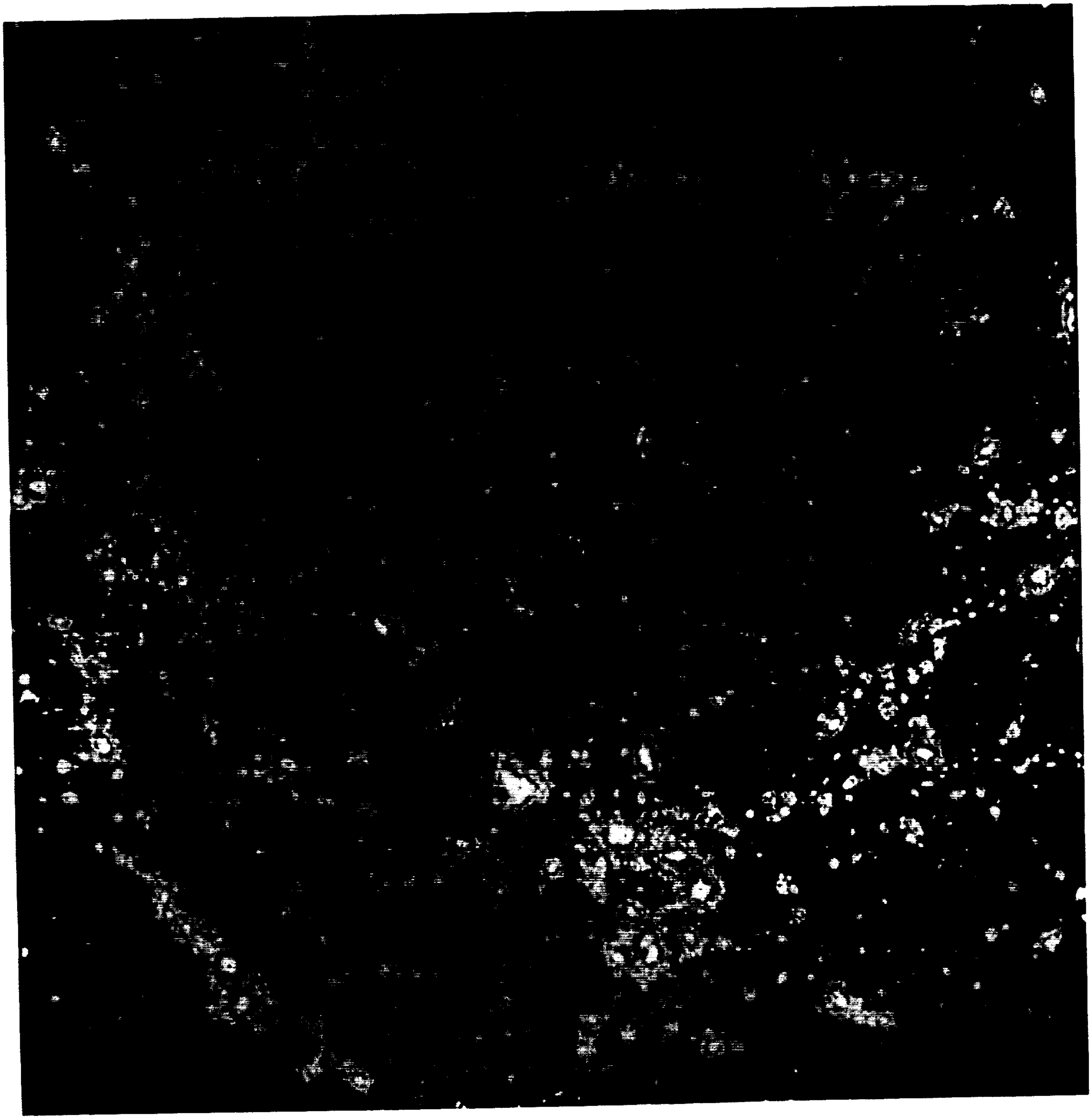
Accomplishments: Preliminary simulations of the termination shock with a population of energetic pickup ions show that these ions, even at a 10% level, have a dramatic effect on the structure of the termination shock. Under certain conditions, a large fraction of the energetic ions are "reflected" by the shock and travel back upstream where they excite large amplitude waves. The waves are convected back towards the shock by the solar wind flow.

Significance: These results may help scientists interpret data from deep space spacecraft (Voyager 1 & 2, Pioneer 10 & 11) which may encounter the termination shock in the near future. Large amplitude waves generated by the reflected ions may give scientists and ground systems personnel advanced warning of an encounter with the termination shock.

Status and Plans: Future simulations will be used to determine if the upstream waves and the shock lead to the generation of very energetic ions through a first order Fermi effect, wherein the ions are energized by bouncing back and forth between the converging waves and shock. This way the pickup ions may be shown to be the source of the anomalous cosmic rays.

Paulett C. Liewer
Earth and Space Sciences Division
Jet Propulsion Laboratory/
California Institute of Technology
(818) 354-6538

Nick Omidi
Electrical Engineering Department
University of California, San Diego
(619) 534-7304



N-Body Simulation on the Touchstone Delta

Objective: Astrophysicists typically simulate the behavior of dark matter using N-body methods. In an astrophysical N-body simulation, the phrase-space density distribution is represented by a large collection of "bodies" (labeled by the indices i, j which evolve in time according to Newtonian laws of motion and Universal Gravitation. The objective is to prepare a faster N-body simulation.

Approach: Rather than implement Newtonian laws directly, consortium researchers use an approximate method that employs an adaptive tree data structure. The time required to obtain an approximate answer for N bodies is proportional to $N\ logN$, which allows for simulation of much larger systems. On the 512-node Delta, the scientists achieve speedups in excess or 400 over the single processor speed.

Accomplishments: In March 1992, the researcher team ran a simulation with 8,783,848 bodies on 512 processors for 780 timesteps. The simulation was of a spherical region of space 10 megaparasecs on a side which is large enough to contain several hundred typical galaxies. Their simulation ran continuously for 16.7 hours, and carried our 3.24 x 10 ^{14} floating point operations, for a sustained rate of 5.4 gigaflops per second. Had they attempted the same calculation with a conventional $O(N^2)$ algorithm, it would have taken almost 3,000 tines as long to obtain an equivalent answer. The scientists created 15 checkpoint files totaling 4.21 gigabits. The Delta allowed them to evolve several hundred large halos simultaneously, in a realistic environment, providing the researchers with much needed statistics, as well as information concerning environmental effects on evolution which cannot be obtained from isolated halo models.

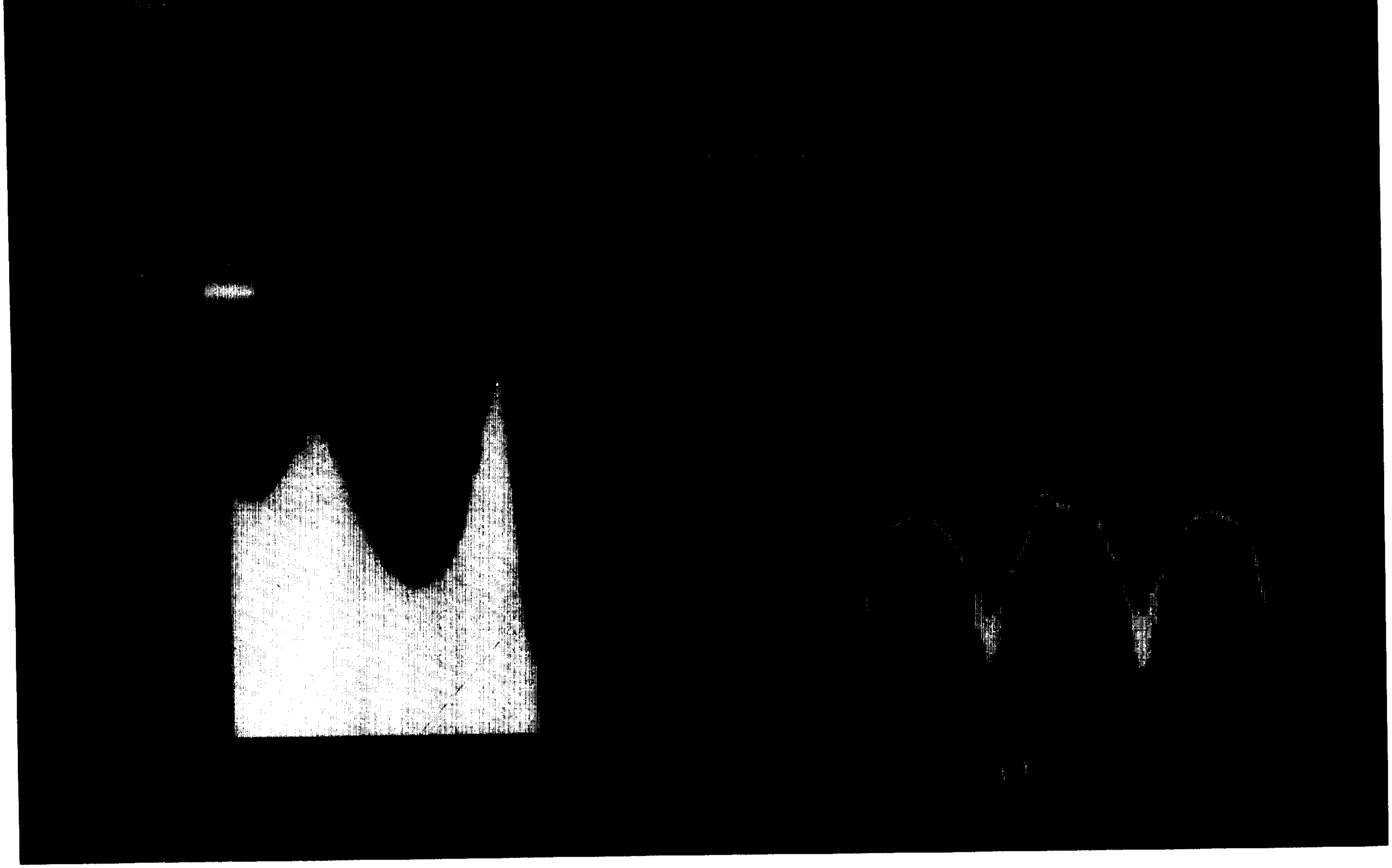
In June 1992, in response to the recently announced measurement of the microwave background anisotropy by the COBE satellite, the team ran two large simulations of the Cold Dark Matter model of the Universe. The COBE measurement has constrained the last remaining free parameters left in this popular theory, and allows the scientist to completely specify the statistical properties of the initial conditions. These are the largest N-body simulations ever run.

Significance: The simulations represented regions with diameters of 250 and 100 megaparsecs and had 17,158,608 and 17,154,598 bodies, respectively. The individual particles each represented about $3.3\text{x}10^{10}$ Msun and 2.0 x 10^9 Msun, respectively, so that the galaxy-size halos are expected to form with tens of thousand of individual particles, enough to obtain reasonable statistics concerning the distribution and correlation of sizes. The spatial evolution was 20 kiloparsecs in both cases. The simulations ran for 597 and 667 timesteps in 23.5 and 28.6 hours, respectively, and wrote 21 and 27 data files for a total of 11.53 and 13.72 gigabits. They respectively performed 4.33 X 10^{14} and 5.32 x 10^{14} floating point operations, sustaining rates of 5.2 and 5.1 gigaflops per second.

Michael Warren
Los Alamos/UCSB

John Salmon
CRPC/Caltech

H – Field Magnitude



Electromagnetic Scattering Calculations on the Intel Touchstone Delta

Objective: To develop analysis tools and numerical techniques that use massively parallel processing systems, such as the Intel Touchstone Delta System, for the solution of large-scale electromagnetic scattering and radiation problems. The analysis codes are used to design and analyze a range of electromagnetic systems such as reflector antennae, scattering objects such as airplanes, and waveguide regions such as multi-component millimeter waveguides.

Approach: A parallel electric field integral equations code, which we have developed to run on several distributed memory parallel processing systems, has been ported to the Delta. The code is used to analyze fields scattered from arbitrarily shaped objects. This code, which uses a moment method to solve the integral equations, results in a dense system of equations with corresponding matrix order proportional to the component's electrical size. Its solution yields the induced current and secondary observational quantities.

To fully realize the Delta's resources, an out-of-core dense matrix solution algorithm that uses some or all of the 90 gigabytes of Concurrent File System (CFS) has been used. Because the limiting part of the simulation is the amount of storage space available, making efficient use of the large CFS is essential.

Accomplishments. The largest calculation completed to date computes the fields scattered from a perfectly conducting sphere modeled by 48,672 unknown functions, resulting in a complex valued dense matrix needing 37.9 gigabytes of storage. The out-of-core LU matrix factorization algorithm was executed in 8.25 hours at a rate of 10.35 gigaflops. The total time to complete the calculation was 19.7 hours; the added time was used to compute the 48,672 x 48,672 matrix entries, solve the system for a given excitation, and compute the observable quantities. The test case of simulating fields scattered from a sphere was chosen for this calculation because an analytical solution is available to compare with computed solution. The Delta-computed fields demonstrated excellent agreement with this exact solution.

Significance: The above computation is significant for several reasons: 1) problems of this size have not been previously reported, 2) the time to complete calculations in this extended regime is short enough that it can be used for engineering calculations, 3) this calculation demonstrates that very large amounts of data can be operated on concurrently, and 4) this calculation extends understanding of the accuracy and stability of large integral equations.

Status/Plans: We are planning even larger runs on the Delta system in the near future. The Delta, with its large CFS, will permit solution of systems with greater than 70,000 unknowns. Currently we are modeling a 1 - 2 gigahertz reflector antennae.

Tom Cwik
Telecommunications Science and
Engineering Division
(818) 354-4386

Jean Patterson
Observational Systems Division
Jet Propulsion Laboratory/
California Institute of Technology

David Scott
Intel Supercomputer Systems Division
Beaverton, Oregon

Visualization of Planetary Data

Objective: To maximize the understanding of planetary data, scientific visualization techniques are being developed for parallel processing environments. These types of visualizations enable researchers to perceive relationships that exist within the data even though the relationships may not be immediately observable.

Approach: One of the most significant of these techniques is terrain rendering. It is used to create three-dimensional perspective views of planetary imagery, simulating what an observer close to the planet's surface would see. Massively parallel computers' such as the Intel Touchstone Delta System, provide a computational platform where rendering very large datasets can be performed in significantly less time than on conventional computers. Terrain rendering software developed at JPL has been used to produce animations based on data from many of the planets and their satellites. JPL's Digital Image Animation Lab (DIAL) production rendering software has been ported to the Delta as an investigation of the applicability of massively parallel computers to the problem. The software employs a ray-casting algorithm and was modified to run in parallel by assigning each processor to produce a portion of each output image. Much of the software technology used to accomplish the port to the Delta was supplied by JPL's Image Analysis Systems Group under the Concurrent Image Processing Testbed Project.

Accomplishments: The application of parallel computers to terrain rendering has provided impressive performance improvements over the production version. It has allowed animation sequences to be produced overnight, instead of requiring several weeks of computer time. It also has allowed generation of large format, very high-resolution, high quality single images in a matter of seconds instead of hours. For example, a 3000x4000 pixel rendered image of the surface of Venus was produced on the Delta on a 64 processor sub-mesh in less than one minute.

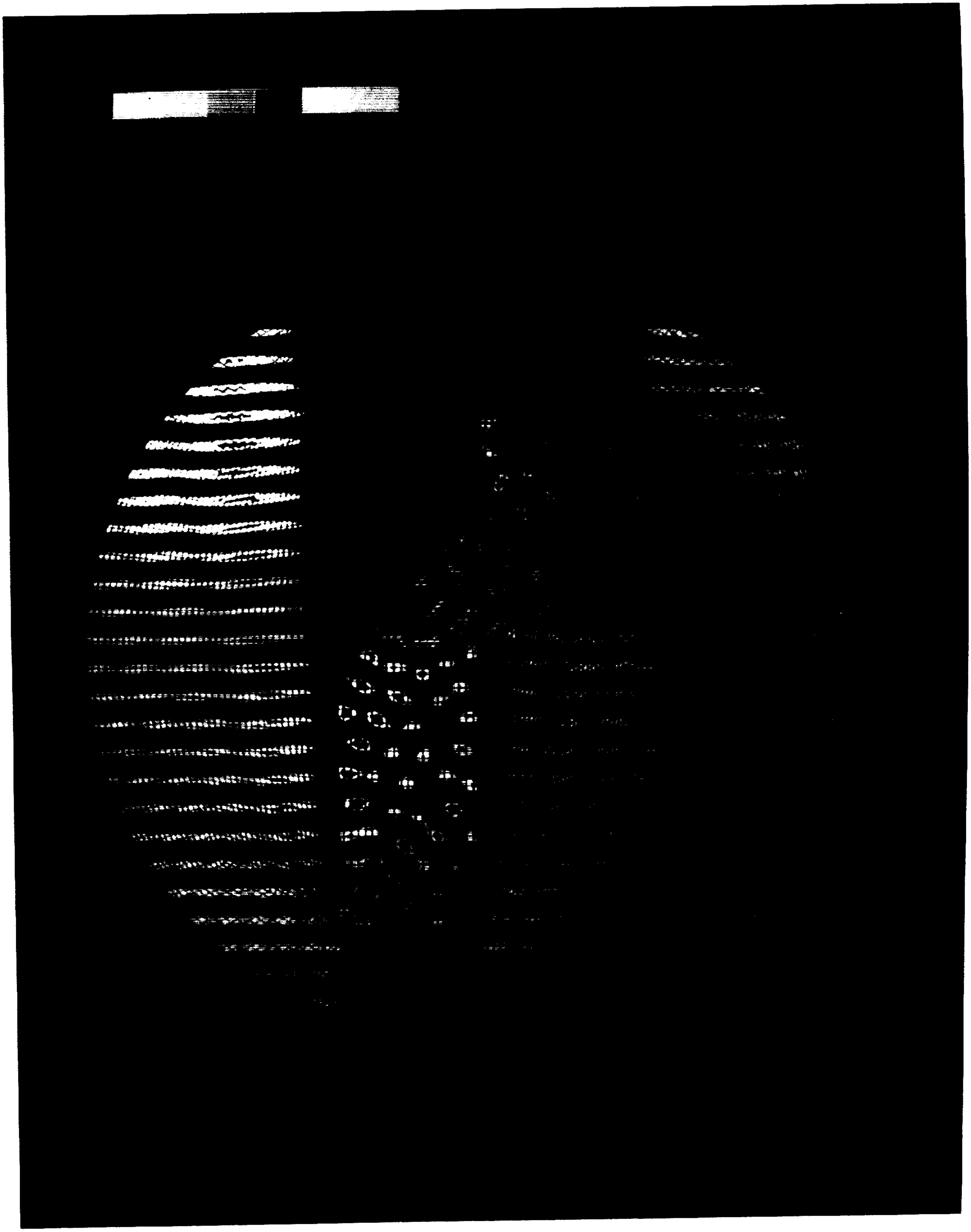
Significance: The Magellan spacecraft, with its Synthetic Aperture Radar (SAR), has returned to earth more data than all other planetary exploration missions combined. Application of massively parallel computers to these datasets has permitted processing of information that would not have been possible on conventional computers. Future missions currently being planned will return even greater volumes of data. Current computing capabilities would not be sufficient to adequately support visualization activities on this data.

Status/Plans: Some preliminary animations using the Delta have been produced for test and analysis purposes, and further visualizations are planned. Using techniques now under development at JPL, scientists will be able to interactively navigate through their data, as if they were on the surface of a planet. Research in these areas will be applied to the parallelization of other image processing applications in the near future.

Steven L. Groom, Stephen H. Watson
Observational Systems Division

Eric M. DeJong
Earth and Space Sciences Division

Jet Propulsion Laboratory/
California Institute of Technology
(818) 354-4055



Finite Element Modeling of EM Scattering Problems

Objective: To develop parallel and otherwise advanced numerical algorithms to model electromagnetic scattering from composite-material objects of large electrical size, based on explicit discretization of the volume for partial differential equation solutions in the frequency domain.

Approach: In order to correctly model the curl-curl operator of electromagnetic wave propagation, a variety of finite element types and boundary conditions have been proposed and implemented. These include node-based elements with tangential and normal component boundary enforcement conditions, and edge-based elements with only tangential component enforcement. In either case, the bulk of the finite element computation time is the solution to a sparse matrix system. We have implemented both major varieties in parallel code on the Delta, and also use direct factorization, as well as iterative matrix solvers for the sparse system. While the methods we use are extensible to the full Delta machine, to date, we have used the Delta system as a set of smaller parallel computers of size 4 to 64 processors. In this way we have been able to make evaluations of different finite element types and techniques concurrently.

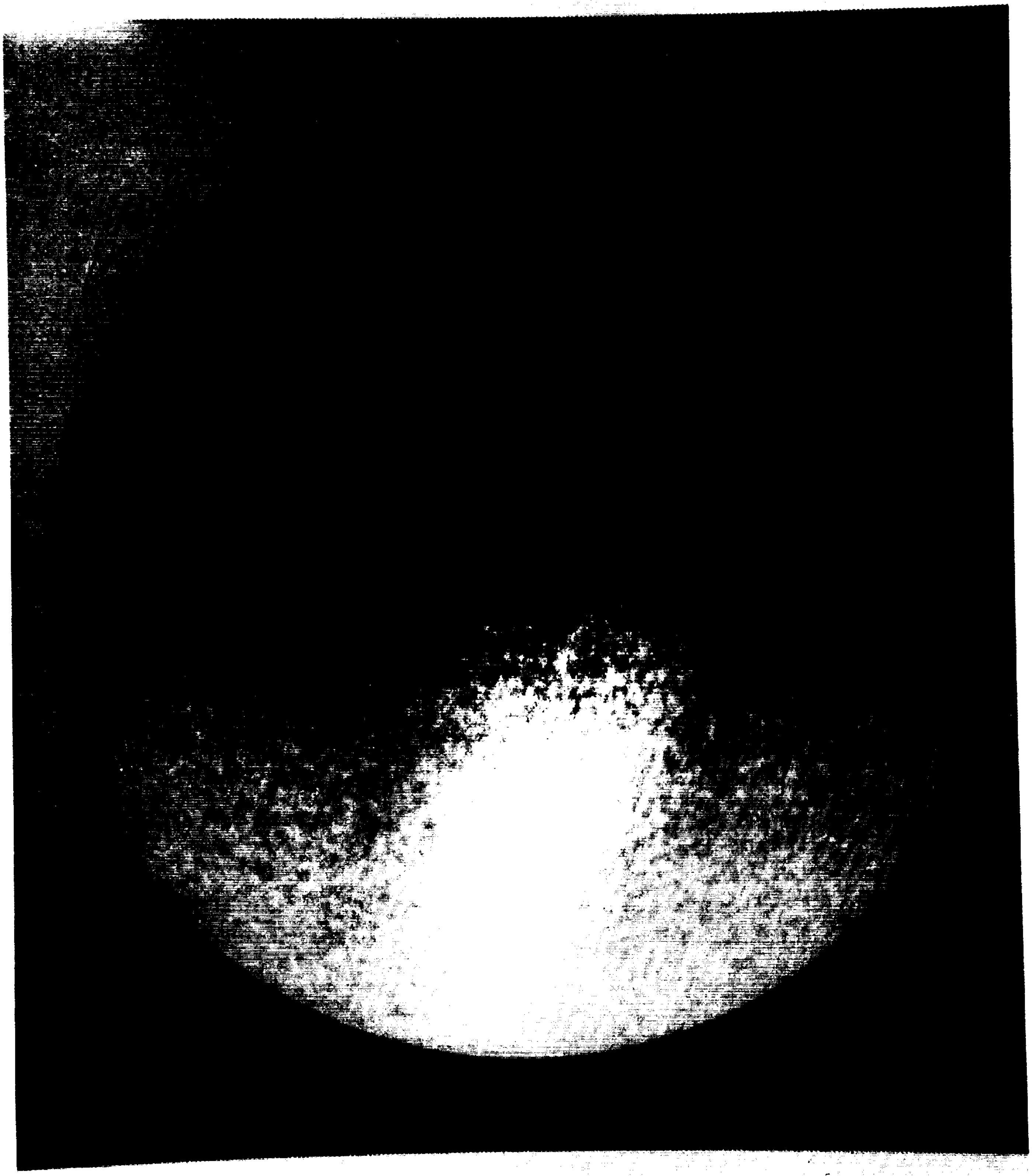
Accomplishments: The implemented varieties of finite elements and solvers enable performance characterizations of larger cases than on alternative computers. We have solved problems in excess of 120,000 unknowns to date, showing exceptional promise for using simple edge elements with iterative solvers. Solution for scattering for a 1.5 wavelength conducting cube shows excellent agreements with test-range measurements and integral equation solutions. Scattering from a 20 wavelength 2-D conducting bent duct with approxi-mately 70,000 unknowns also agrees very well with an integral equation technique solution.

Significance: These computations demonstrate the feasibility of 2-D and 3-D computation by iterative solution on parallel computers, which solution technique scales extremely well for existing and future large parallel computers. Projecting these methods, we estimate current Delta resources are adequate for solving problems involving millions of unknowns and occupying hundreds of cubic wavelengths.

Status/Plans: We are planning modeling computation of larger objects, up to the capacity of the 8 Gigabytes of in-core memory. We also are continuing to examine performance issues, particularly of higher-order finite elements, conforming wave-absorbing boundary conditions, and many-material composite objects.

Jay Parker
Robert Ferraro
James McComb
Suguru Araki

Observational Systems Division
Jet Propulsion Laboratory/
California Institute of Technology
(818) 354-6790



Helioseismology Studies of Solar Structure and Dynamics

Objectives: Accomplish fast processing of extremely large image data files of the sun using distributed memory parallel computers. This research project in the young field of helioseismology will help obtain a greater understanding of the three-dimensional thermodynamic structure and dynamical motions of the unseen interior regions of the sun.

Approach: The interior and surface layers of the sun are constantly in vibration in very small amounts in the inward and outward directions. This enables the application of geophysical observational and theoretical techniques to study of the sun's interior. A solar telescope acts much like a seismograph, measuring the speed and direction of movement of each portion of the sun's surface. The pictures, called 'filtergrams', are taken 11 hours a day, 200 days per year, from Mt. Wilson Observatory's 60-Foot Solar Tower Telescope. These filtergrams are transferred to the Intel Touchstone Delta for processing.

The uniqueness of this particular project is the product of the relatively high spatial resolution of solar images (with one million information elements available in each image), with the enormous total number of these images obtained since observations began in 1987. Just the sheer volume of the enormous database (which now totals roughly one terabyte in size) makes the availability of a massively-parallel computer such as the Delta so attractive. Up to 2.5 gigabytes of data are generated in a single observing day, and the processing of a 10-day time series of raw filtergrams occupies 25 gigabytes of disk space on the file system. The Delta offers enormous amounts of temporary disk space for temporary storage of both input and output images. No other supercomputer center in the United States has been able to provide anything close to this much disk space - even temporarily. As part of the image processing, the raw solar filtergrams are converted into maps of the line-of-sight velocity, which cover the entire visible hemisphere of the sun.

Accomplishments: Through the use of massively parallel computers, analysis of a backlog of images was made possible, allowing insight into many more images than would be possible using conventional computers and supercomputers. The independence of images allows the use of coarse-grained parallelism on the Delta in order to speed up effective data processing throughput by a factor of 30 to 50 over con-ventional single-processor supercomputers.

Significance: This work will result in a better understanding of the future behavior of the sun; specifically, the refinement of the picture of solar internal structure to a such a level as to be able to predict with a higher degree of confidence how and when the sun will evolve its structure in the future. The software that is being developed for the Delta under this project will also be used in the analysis of data that will be returned by the NASA-sponsored Solar Oscillation Investigation, which is being developed to fly on the Solar and Heliospheric Observatory (SOHO) spacecraft.

Status/Plans: When we have been able to extend our power spectral analysis to include the so-called "tesseral" harmonic modes, we will be able to study the internal angular velocities of the sun better. We also hope to be able to make better short-term predictions about future changes in the level of solar activity during upcoming sunspot cycles.

Ed Rhodes, Natasha Johnson
Jet Propulsion Laboratory/
University of Southern California
(818) 354-3467

Dennis Smith
University of Southern California
Sylvain Korzennik
Harvard-Smithsonian Center for Astrophysics

(THIS PAGE LEFT BLANK INTENTIONALLY.)

Overview of ESS Systems Software

Project Goal and Objectives: The goal of the ESS systems software research activities is to make future full-up teraFLOPS computing systems significantly easier to use than 1991 vintage conventional vector processors for ESS Grand Challenge applications. The objectives are to ildentify and remove system software weaknesses which are obstacles to NASA's eventual integration of scalable systems into production computing operations, and to identify and develop system software components which make scalable systems easier to use than current vector processors

Strategy and Approach: A modest number of focused high payoff projects are being supported in important topic areas where in-house expertise is available to provide technical direction. At GSFC, these areas are 1) the achievement of effective and efficient architecture-independent parallel programming; 2) development of data management strategies and tools for management of petabytes of data; 3) implementation of advanced visualization techniques matched to teraFLOPS system requirements; and 4) the development of the Software Sharing Experiment which is the first two year phase of the Federal High Performance Computing Software Exchange. At JPL, these areas are 1) tools for the numerical solution of partial differential equations on parallel computers, including parallel unstructured mesh generation; 2) investigation of new parallel programming paradigms applied to science applications; 3) skeleton parallel implementations of popular numerical methods; 4) investigation of automated dynamic load balancing mechanisms; and 5) systolic data flow tools for high throughput data processing applications.

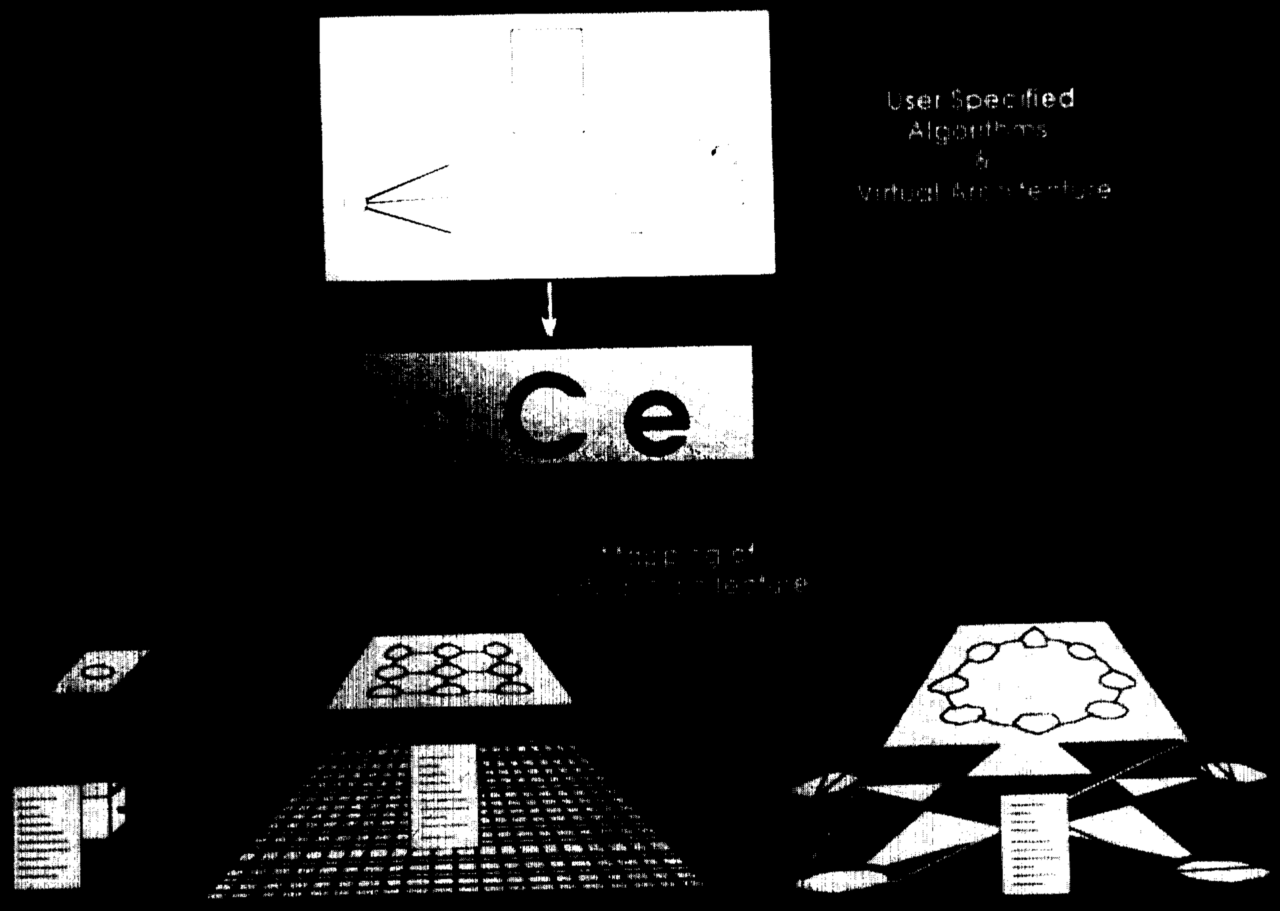
Organization: Extensive collaborations with the academic and vendor community are anticipated to assist the architecture-independent programming work. The visualization activity is operating in collaboration with ARC. The Software Sharing Experiment enjoys a collaboration with DARPA, DoE, EPA, NIST, NOAA, NSA, and NSF, and receives interagency oversight from the FCCSET Working Group on Science and Engineering Computing. The parallel programming paradigm investigation is being done in collaboration with the University of Virginia, with other academic research institution collaborations anticipated. JPL also supports basic research at the California Institute of Technology on architecture features that pertain to data movement.

Management Plan: At GSFC, a Deputy Project Manager for Systems Software Research directs subelement activities. At JPL, a Deputy Task Leader performs the same function.

Points of Contact:

John E. Dorband Robert Ferraro
Goddard Space Flight Center / Code 930.7 Jet Propulsion Laboratory
dorband@nibbles.gsfc.nasa.gov ferraro@zion.jpl.nasa.gov
(301) 286-9419 (818) 354-1340

Architecture Independent C programming Environment



User Specified
Algorithms
&
Virtual Architecture

Mapping of
Virtual Architecture

Defined Architecture Independent C Programming Environment (aCe)

Objective: To create a programming environment independent of physical computer architecture which will promote simplified development of efficient parallel code.

Approach: First, design a language that allows specification of programs in terms of the parallel architecture best suited to the algorithm rather than the architecture on which the algorithm will actually be executed. Then, map the language to a variety of physical architectures and compare the same algorithm run on various architectures. Finally, distribute a program across multiple architectures, fitting various components to the architectures where they are best suited.

Accomplishments:

□ Version 1.0 language specification complete

□ Version 1.0 compiler in alpha test

□ Version 1.0 code generator near completion for SGI workstation

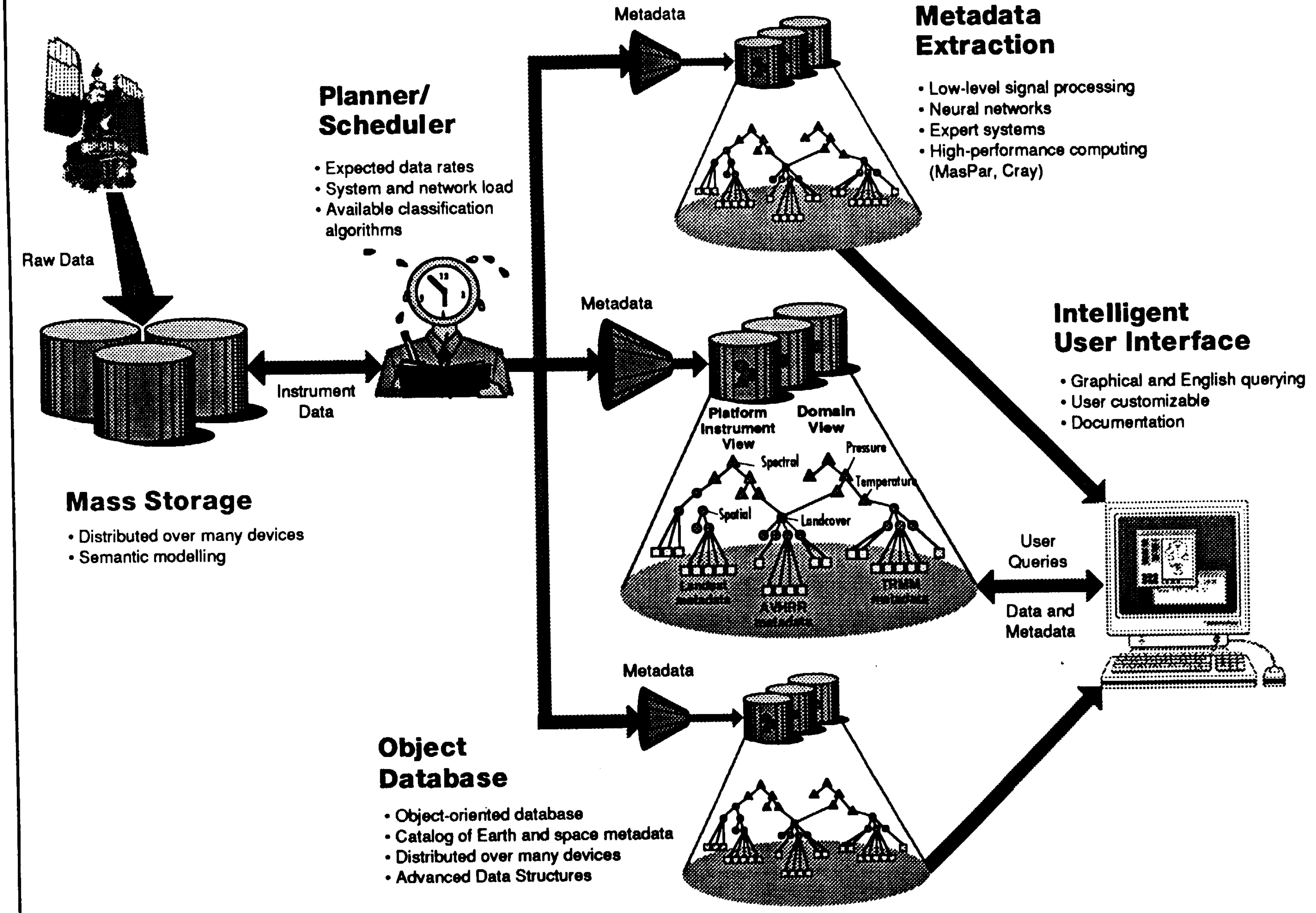
Significance: It is inherently difficult to implement on, port to, or port applications between parallel architectures with currently available languages. This is due to 1) lack of identical implementations of languages across multiple architectures, and 2) lack of architecture independent expressiveness for describing algorithms that are easily implemented across different architectures. The architecture independent C programming environment (aCe) addresses these issues by 1) providing a syntax that allows simple expression of algorithms in terms of the virtual architecture that best suits the algorithm, 2) providing a semantic that can be mapped to most commonly available architectures, and 3) providing a flexible implementation, so that new architectures may be supported with as little effort as possible. The availability of aCe should result in the development of better algorithms, longer lasting code, improved architectures, and greater flexibility to improve architecture design with minimal software migration cost.

Status/Plans: In FY93, version 1.0 of aCe for the SGI and MasPar MP-1 will be completed and tested. In the future, architectures like the Cray MPP, Intel Paragon, and Cray C90 will be supported. The intent is to do comparative studies of the performance of aCe on various architectures. If aCe proves to be an effective programming tool, mapping to heterogeneous computing environments will be addressed.

John E. Dorband
GoddardSpaceFlightCenter
Code 930.7
dorband@nibbles.gsfc.nasa.gv
(301) 286-9419

HPCC/ESS Data Management

OAST



Mass Storage

- Distributed over many devices
- Semantic modelling

Planner/ Scheduler

- Expected data rates
- System and network load
- Available classification algorithms

Metadata Extraction

- Low-level signal processing
- Neural networks
- Expert systems
- High-performance computing (MasPar, Cray)

Intelligent User Interface

- Graphical and English querying
- User customizable
- Documentation

Object Database

- Object-oriented database
- Catalog of Earth and space metadata
- Distributed over many devices
- Advanced Data Structures

Developed Data Management Software

Objective: Develop efficient algorithms for SIMD and MIMD machines that automatically extract image content, organize and manage databases, and that enable efficient methods of browsing large, complex spatial databases through faster querying methods.

Approach: Develop algorithms for automatic georegistration of spatial data sets, techniques for spatially organizing satellite observations, techniques for automatic extraction of metadata from imagery, and rapid access indices for searching large, complex spatial databases.

Accomplishments: Completed the first draft of a "white paper" covering the state-of-the-art in data management on high performance machines.

Designed and implemented a backpropagation neural network on a SIMD machine (MasPar) for satellite image characterization and published the results. Designed and implemented a decision tree algorithm on a massively parallel machine (MasPar).

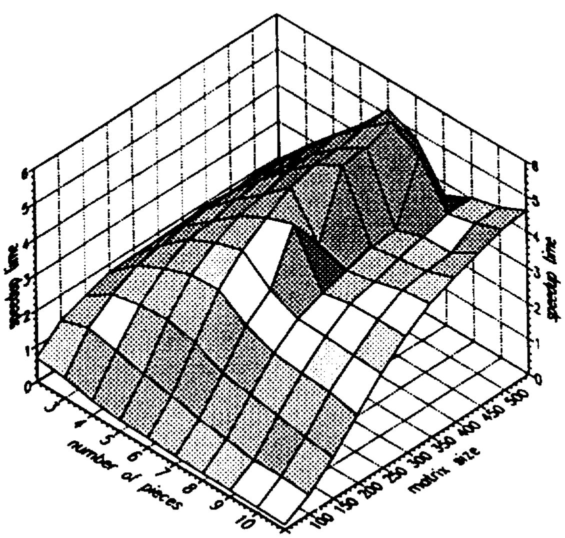
Significance: "White paper" serves as a living document on performance results of evolving technologies and will serve as documentation on future research directions to avoid duplication of efforts. Neural network performed several orders of magnitude faster than serial machines. Decision trees were used to validate, classify and optimize data characterization (machine learning).

Status/Plans: Implement algorithms for extracting image features, organizing spatial data for browse using spherical quadtrees, and analyze performance of decision trees and neural networks.

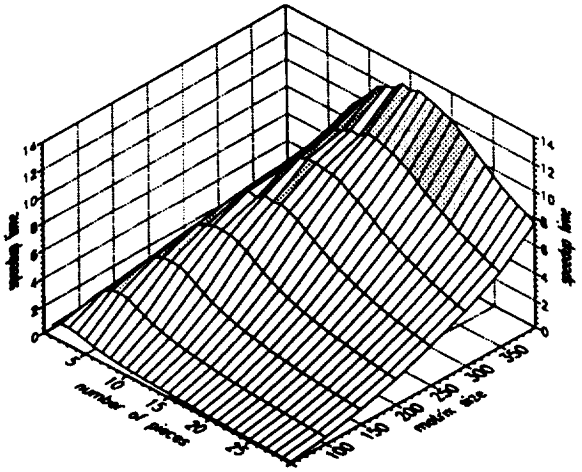
William J. Campbell
Goddard Space Flight Center
Code 930.1
campbell@nssdca.gsfc.nasa.gv
(301) 286-8785

Mentat Performance

Gaussian Elimination with Partial Pivoting

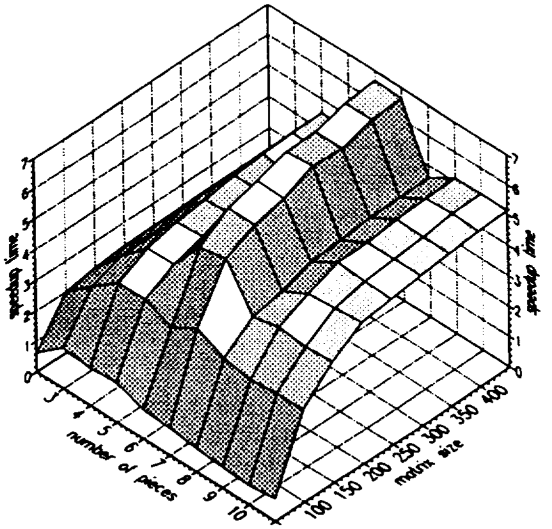


8-processor Sun 3/60 network

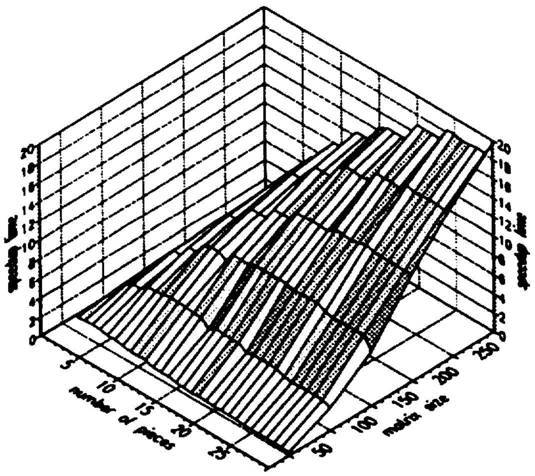


32-processor Intel iPSC/2

Matrix Multiplication



8-processor Sun 3/60 network



32-processor Intel iPSC/2

Evaluation of New Parallel Programming Paradigms: Mentat

Objective: Our objective is to port Mentat to the Intel i860 Gamma machines at JPL and to the Caltech Delta machine, and to investigate the feasibility of applying object-oriented parallel techniques to real scientific application codes. We have chosen two application codes now running on the Caltech/JPLMark IIIfp Hypercube and the Intel iPSC/860 Gamma machines to be used in this investigation. Our goals are 1) to determine the utility of the Mentat system for scientific applications; 2) to evaluate the performance of an object-oriented system versus hand-coded; and 3) to determine what enhancements to a system such as Mentat would make it more useful in a scientific environment.

Approach: Two problems plague programming parallel MIMD architectures. First, writing parallel programs by hand is very difficult. The programmer must manage communication, synchronization, and scheduling of tens to thousands of independent processes. The burden of correctly managing the environment often overwhelms programmers. Second, once implemented on a particular MIMD architecture, the resulting codes are ,usually not usable on other MIMD architectures; the tools, techniques, and library facilities used to parallelize the application are specific to a particular platform. Thus, considerable effort must be re-invested to port the application to a new architecture. Given the plethora of new architectures and the rapid obsolescence of existing architectures, this represents a continuing time investment.

Mentat has been developed to directly address the difficulty of programming MIMD architectures and the portability of applications. The three primary design objectives are to provide 1) easy-to-use parallelism, 2) high performance via parallel execution, and 3) applications portability across a wide range of platforms. Mentat combines a medium-grain, data-driven computation model with the object-oriented programming paradigm and provides automatic detection and management of data dependencies. The data-driven computation model supports high degrees of parallelism and a simple decentralized control, while the use of the object-oriented paradigm permits the hiding of much of the parallel environment from the programmer.

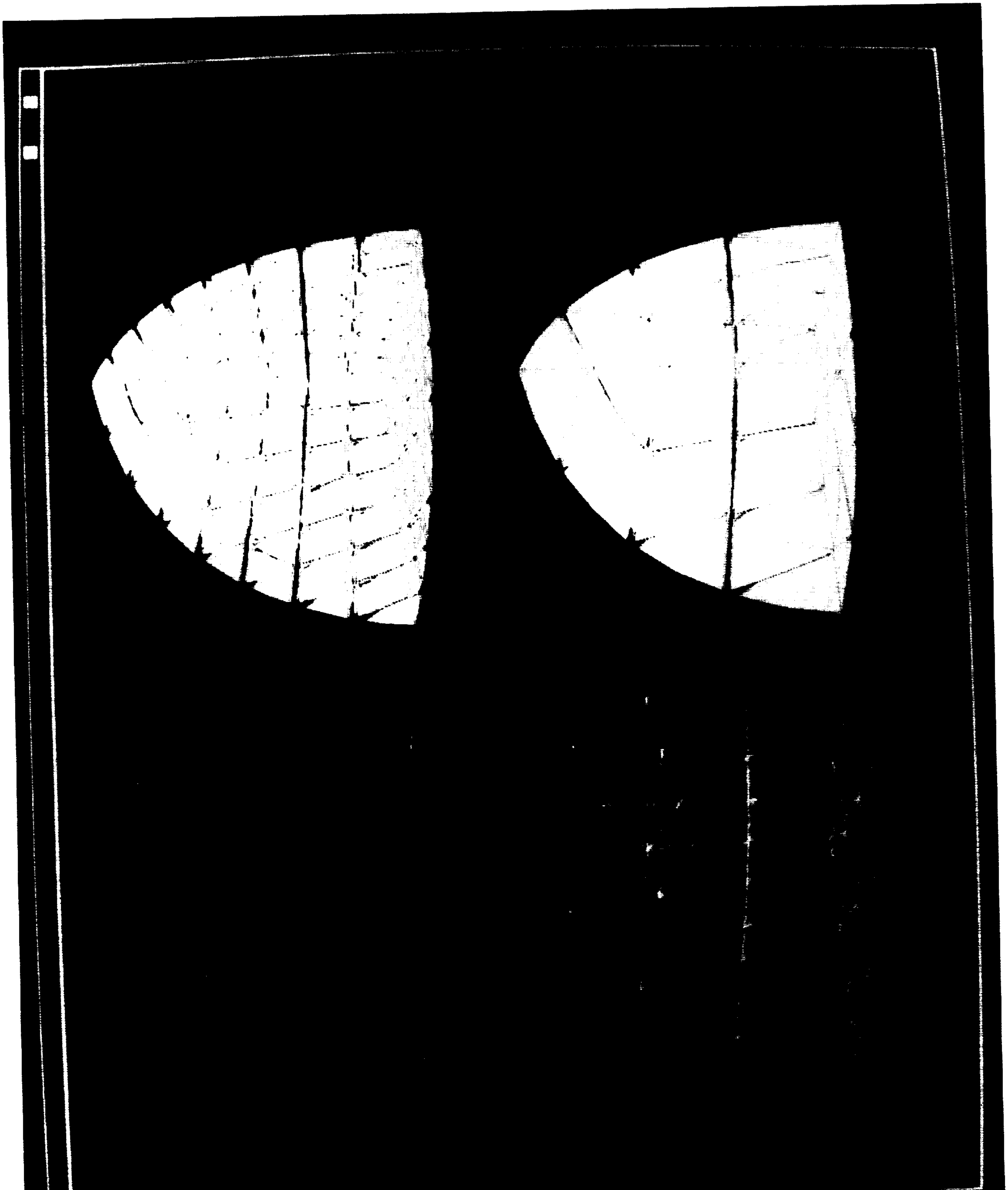
Accomplishments: An alpha release of Mentat for Sun 3's, Sun 4's, the Intel iPSC/2, and the Silicon Graphics Iris is available. Performance results on a range of applications are available and quite encouraging.

Significance: The premise underlying Mentat is that writing programs for parallel machines does not have to be hard. Instead, it is the lack of appropriate abstractions that has kept parallel architectures difficult to program, and hence, inaccessible to mainstream, production scientific program-mers. If successful, this project will demonstrate that object-oriented parallel processing techniques can significantly reduce the complexity of writing parallel software.

Status/Plans: Mentat has been ported to the Intel iPSC/860 Gammas at JPL and at Oak Ridge. Performance testing has recently begun. When the Paragon OS is available, Mentat will be ported to Paragon. We have just entered phase two of the effort, implementing the two chosen applications in C++/MPL. We expect completion of one of the two by spring of 1993.

Andrew Grimshaw
Department of Computer Science
University of Virginia
(804) 982-2204

Robert Ferraro
Observational Systems Division
Jet Propulsion Laboratory/
California Institute of Technology
(818) 354-1340



C-2

Parallel Refinement and Partitioning of Finite Element Meshes

Objective: Construct, using parallel computers, surface and volume finite element meshes partitioned appropriately for use on distributed memory parallel computers. This type of mesh is used in solving a variety of partial differential equations.

Approach: Generation of finite element meshes, which properly capture complex geometry and have the mesh density required to perform calculations, is a computationally intensive task. Geometric constraints can often be encapsulated in a mesh which is too coarse to perform accurate calculations. Using a coarse mesh as the geometry description, automated mesh refinement is applied to each existing element, splitting it into a sufficient number of new elements to satisfy the local mesh density requirement. The origin of new elements, with respect to the input mesh, is retained, so that efficient removal of duplicate information is possible. Since refinement is done on each element individually, without reference to other elements, the algorithm is expected to parallelize efficiently. The refined mesh is then partitioned in a manner appropriate for use on distributed memory parallel computers.

Accomplishments: A set of codes which take, as input, a coarse finite element mesh and produce a dense, partitioned mesh of the same character have been implemented on workstations. The mesh partitioner has also been implemented on an Intel iPSC/860 hypercube. An example of a coarse input mesh and the results at each stage of refinement showing in the figure. The mesh, taken from an electromagnet scattering problem, represents an octant of a dielectric sphere. The full mesh for the scattering problem would require the region surrounding the sphere to be meshed as well. This technique allows the generation of a mesh of arbitrary density which retains the curvature of the original coarse mesh.

Significance: The sequential implementation of the mesh refinement algorithm provides the starting point for a parallel implementation. The parallel implementation of the mesh partitioner significantly reduces the time required to partition finite element meshes for use on distributed memory parallel computers.

Status/Plans: The parallel implementation of a mesh refinement code coupled to the partitioner has begun. Future improvements to the current algorithm are being considered, which will greatly reduce or eliminate the communication between processors. Future work will provide dynamic repartitioning of meshes in situations where the mesh evolves through localized refinement.

Koz Tembekjian
Suguru Araki
Robert Ferraro
Observational Systems Division
Jet Propulsion Laboratory/
California Institute of Technology
(818) 354-1340

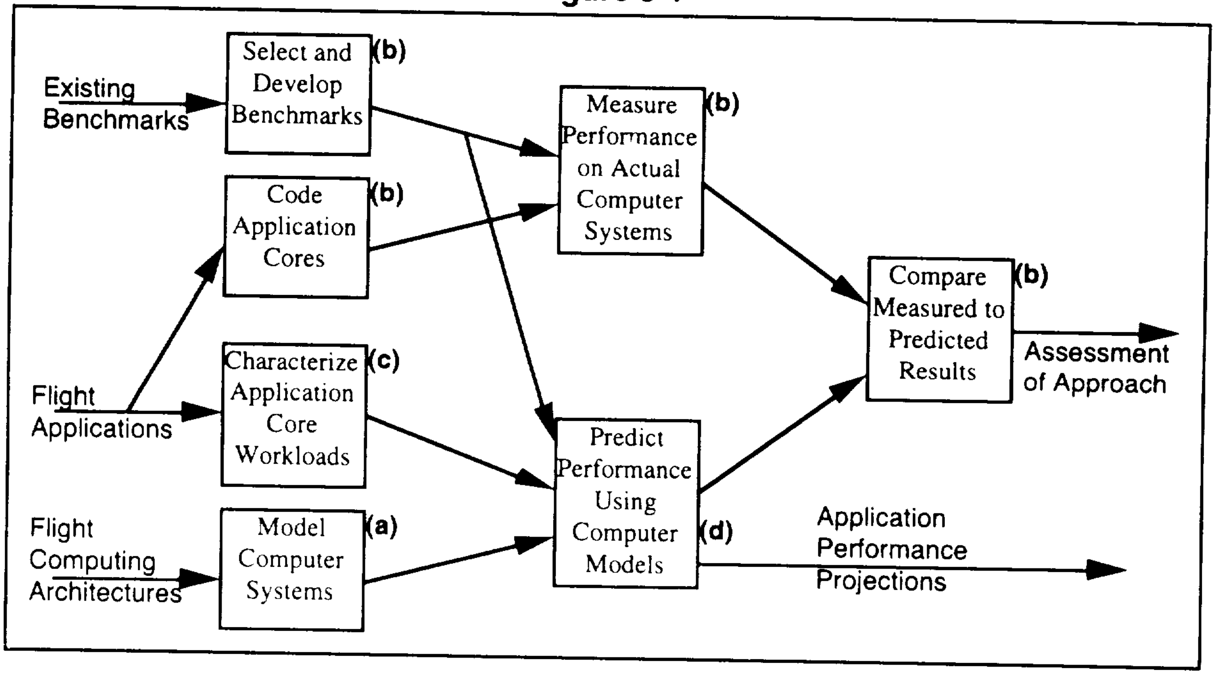
(THIS PAGE LEFT BLANK INTENTIONALLY.)

Introduction to the Remote Exploration and Experimentation (REE) Project

Goal and Objectives: Space exploration and scientific investigation have opened new windows to address difficult and far-reaching questions about the Universe. Because of the complexity of these activities and the large amount of data they produce, they are wholly dependent on computers for their success. For past and current missions, the bulk of this computation has been done on the Earth, with a small number of essential functions (e.g., spacecraft attitude control) being managed by on-board computers of relatively modest capability.

The goal of the Remote Exploration and Experimentation (REE) Project is to develop spaceborne computing technology which will enable high-performance, fault-tolerant, adaptive space systems for a new generation of missions to explore the Earth and Solar System. The focus of the Project in FY92 is the development and demonstration of a modeling and evaluation methodology which can be used to project performance of high-performance spaceborne computing systems (see Figure 3-1.) This is supported by four specific objectives (1) develop performance models for space qualifiable single and multiprocessor system architectures; (2) validate these models against actual hardware using standard benchmarks; (3) develop workload characterizations of representative space applications; and (4) predict performance for the characterized applications executed on the modeled architectures.

Figure 3-1



Existing Benchmarks
Select and Develop Benchmarks
(b)
Code Application Cores
(b)
Flight Applications
Characterize Application Core Workloads
(c)
Flight Computing Architectures
Model Computer Systems
(a)
Measure Performance on Actual Computer Systems
(b)
Predict Performance Using Computer Models
(d)
Compare Measured to Predicted Results
(b)
Assessment of Approach
Application Performance Projections

Strategy and Approach: Future plans call for scientific instruments whose data is increasingly voluminous, far exceeding the constraints set by telemetry rates. Moreover, ambitious mission scenarios will require real-time systems capable of precision soft landing (and hazard avoidance) on the Moon and Mars, aerocapture into orbit at Mars, rendezvous and docking between spacecraft, navigation and control of robotic vehicles and manipulator arms, and precision pointing and tracking of scientific instruments. Other scenarios call for data and control systems capable of adaptive response to scientific targets of opportunity. These plans and scenarios dictate mission requirements in which spaceborne computing plays a role of increased importance.

The long-range strategy of the REE Project is to develop an architectural design for a space-qualifiable gigaFLOPS flight computing system by means of a series of limited-scope experiments carried out on scalable testbeds using selected test cases. Projections of performance for the full-scale flight system will be validated by means of these experiments.

In FY92, this process began with the development of a modeling and evaluation capability for space computing systems. Researchers developed models of three computing systems using the SES/Workbench tool. These models were validated by comparing modeled performance to actual performance obtained by running selected benchmarks on these machines.

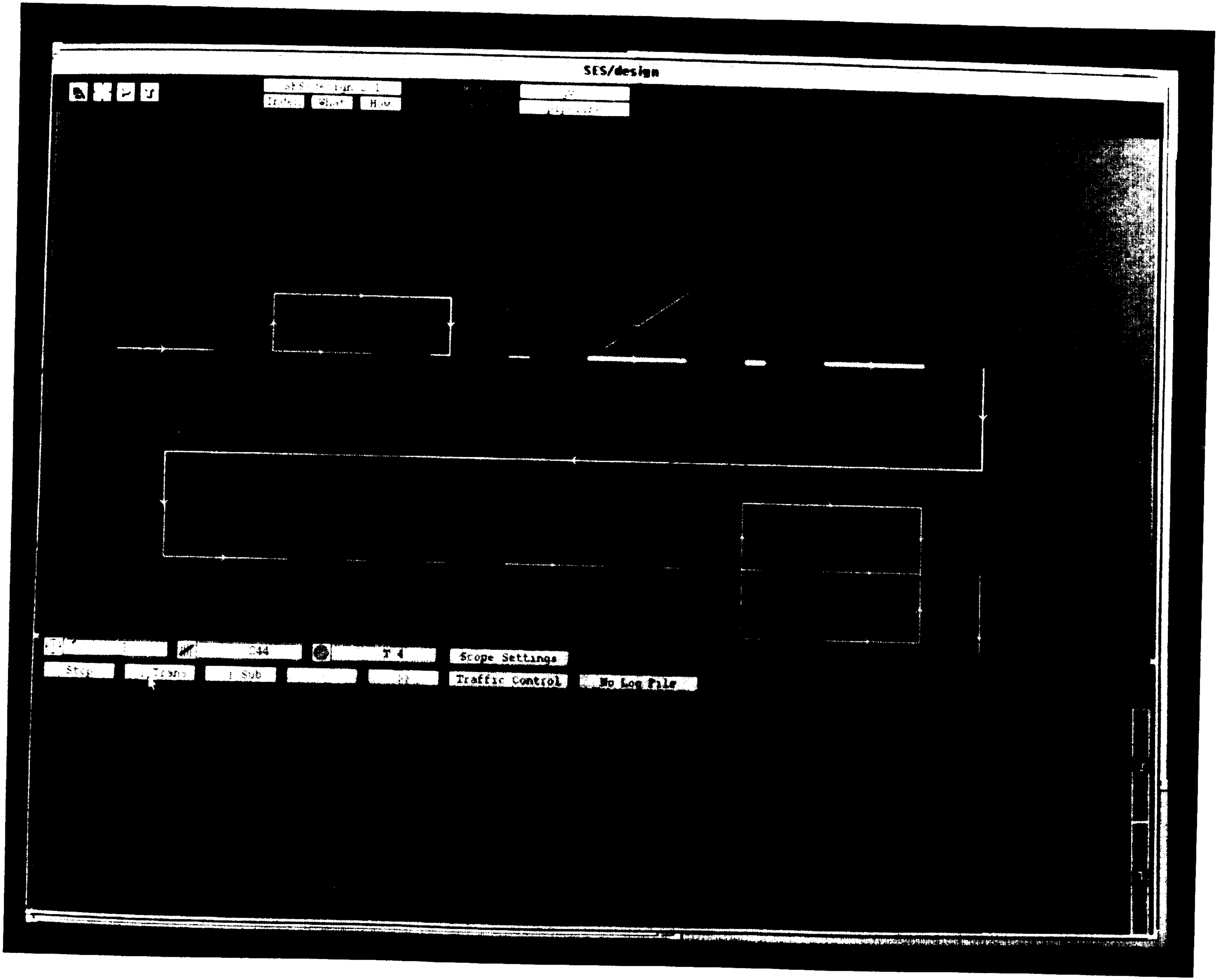
Application specialists developed algorithms, code, and workload characterizations for applications involving analysis of atmospheric line spectra, stereo vision for planetary rovers, real-time motion planning for robotic manipulators, and the combination of upper triangular Square Root Information Filter (SRIF) matrices. Three of these applications were implemented on the Intel Gamma machine at JPL. Their performance on this machine will be compared to the performance predicted by the i860 model for the same applications, based on their workload characterizations.

The REE Project is sponsoring a workshop entitled "Computing in Space: User Requirements and Technology Challenges" in December 1992 in collaboration with the University of Illinois. Members of key user groups have been invited. This workshop will produce a white paper which will help guide and focus future NASA developments in spaceborne computing.

Organization: The REE Project is part of the NASA High Performance Computing and Communication Program; all of the activities at a particular center report through the REE Project Manager, the Task Manager for Grand Challenge Applications, and the Task Manager for Modeling and System Design for the Jet Propulsion Laboratory.

Management Plan: The project is managed in accordance with the formally approved REE Project Plan. All activities report through the REE Project Manager, John Davidson, the Task Manager for Grand Challenge Applications, Jean Patterson, and the Task Manager for Modeling and System Design for the Jet Propulsion Laboratory, Edwin Upchurch. Monthly, quarterly, and annual reports are provided to the High Performance Computing Office in Code R.

Point of Contact:
John Davidson
JPL
(818) 354-7508



PRECEDING PAGE BLANK NOT FILMED

(THIS PAGE LEFT BLANK INTENTIONALLY.)

REE Accomplishments: Benchmarking and Performance Modeling

Objective: To develop and demonstrate a modeling and evaluation methodology which can be used to project the performance of high-performance spaceborne computing systems.

Approach: Mission designers and spacecraft designers need the capability to determine what functions on-board computing systems can perform and to select computing architectures for required spaceborne processing. If applications can be characterized and computing systems modeled with sufficient accuracy, these models can be used to make early design decisions. The REE project is evaluating the effectiveness of this approach by modeling several computing systems for which hardware systems are available, selecting and developing space related benchmarks, using the models to predict the performance of the benchmarks, and comparing the predictions against the measured performance when executing on the hardware.

Accomplishments: Engineers developed workload characterizations for three benchmarks related to space applications in addition to the standard Whetstone and Drystone benchmarks. The first two represent major spacecraft engineering subsystems, the Attitude and Articulation Control System and the Command and Data System. The third models some of the characteristics of robotic control.

Modeling experts used the SES/Workbench tool to develop performance models of three processor systems: the Honeywell GVSC, the JPL-developed Space-16, and the Intel i860. (The accompanying graphic shows the computer screen during the execution of one of these models.) The Generic VHSIC Spacecraft Computer (GVSC) implements the Mil-Std 1750A instruction set, is pipelined for instructions, runs Ada, and is actually space qualified. Space-16 is a CISC processor based on space qualifiable National Semiconductor chips utilizing several special purpose chips designed at JPL, including a Direct Memory Access Coprocessor (DMAC). It is pipelined for instructions and supports "C". Lastly, the Intel i860 is a RISC processor, pipelined for floating point, and is not expected to be space qualified. It is, however, the node processor for the Intel Gamma multiprocessor and allows for parallel processing evaluation and modeling. A multi-processor model will be developed for the i860 as time permits.

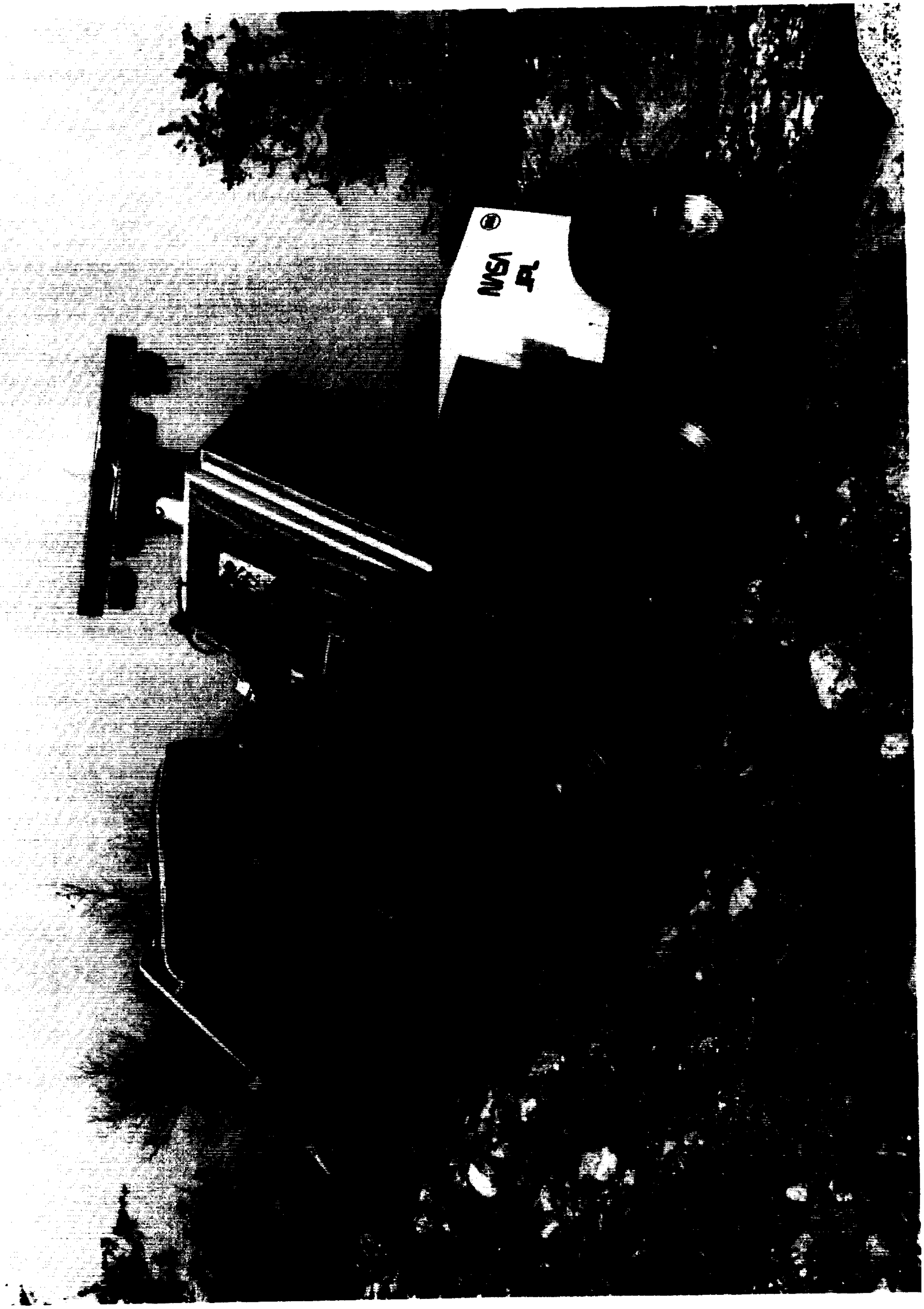
Benchmarks were run on the GVSC and Space-16 processors to yield timing data. Comparison with the modeling predictions will validate the models.

Significance: Performance modeling and benchmarking will assist design teams in assessing the suitability of candidate processor/language combinations for new projects. Workload characterizations may also be used to highlight program characteristics for instruction set and processor system designers in both the public and private sectors. This methodology can be used to perform system-level architectural tradeoffs.

Status/Plans: The Project will publish a report describing the methodology and the evaluation results near the end of FY92. The "Computing in Space" workshop in December and its resulting white paper will conclude the year's activities. REE is then slated to be temporarily suspended until FY96. The project will culminate in demonstrating and evaluating a recommended scalable architecture for high-performance on-board computing.

E. Upchurch
Information Systems Division
(818) 356-6172

J. Davidson
Electronics and Control Division
(818) 354-7508



NASA
JPL

Monte Carlo Simulations for Performance Evaluation of Stereo Vision Systems

Objective: (1) To optimize the measurement sensitivity of stereo vision algorithms by using Monte Carlo simulations to evaluate algorithm design alternatives: (2) To enhance the speed of stereo vision algorithms by exploiting parallel computation.

Background: "Stereo vision" is the method for estimating "range images" (arrays of 3-D coordinates) by matching corresponding features in stereo pairs of intensity images. This is a primary approach to 3-D sensing for a number of tasks, including remote sensing for mapping, obstacle detection for semi-autonomous planetary rovers, and work-space modeling for telerobotic servicing of orbital spacecraft. Two important problems in the development of algorithms for stereo vision are (1) optimizing the accuracy of range estimates, given the presence of noise in the images, and (2) achieving sufficient speed for real-time applications, given the high level of computation required by stereo vision algorithms.

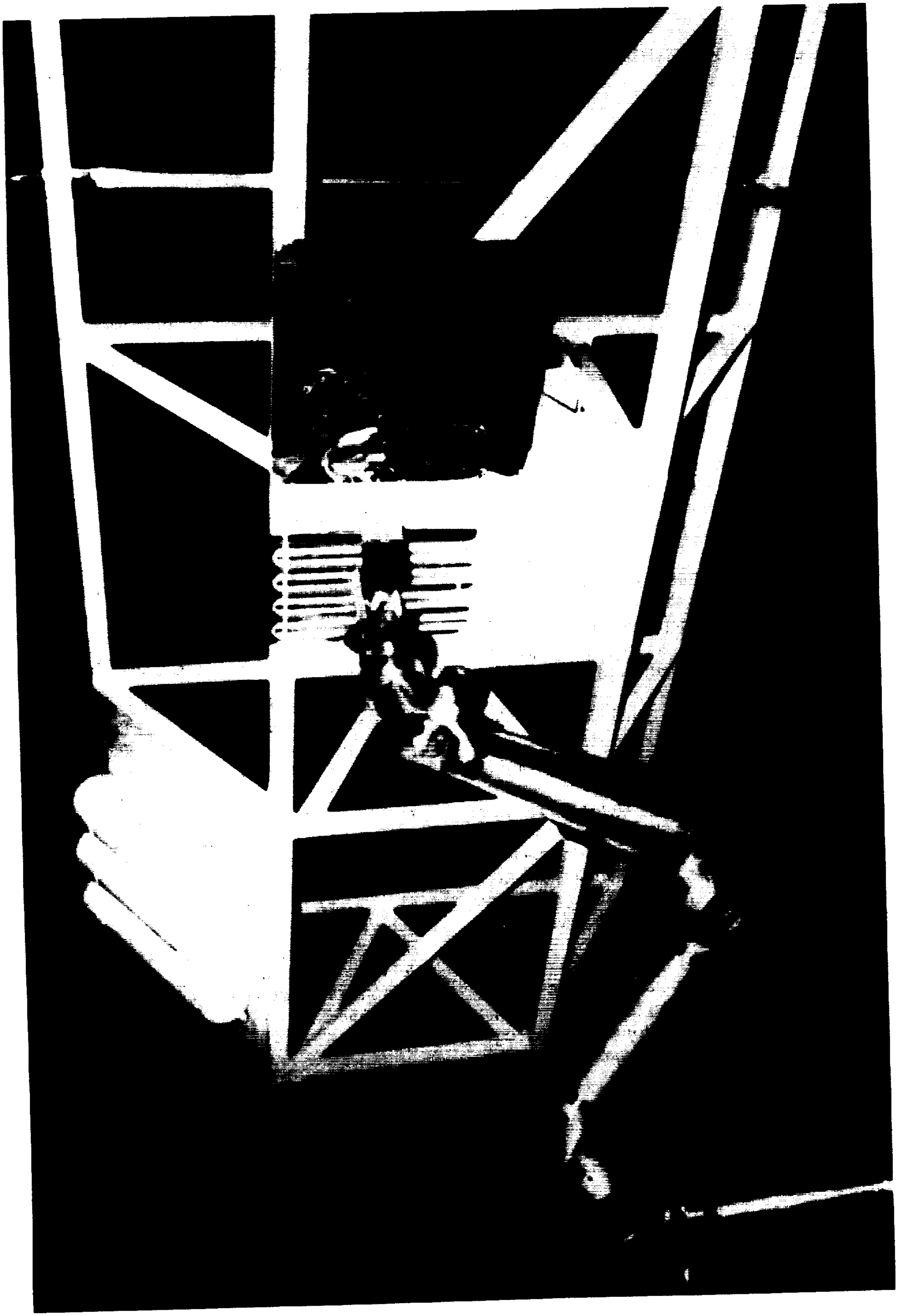
Approach: Accuracy depends on a number of factors that are difficult to model analytically. Therefore, empirical testing with large data sets is an important part of evaluating the performance of algorithm design alternatives. Simulations over large numbers of trials are used to obtain statistically valid results. Parallel computing is employed to perform the entire set of trials in a practical amount of time. For real time applications, individual stereo image pairs must be processed in a very short time. In this case, parallel computing is used to process each image pair within the time limit. Stereo vision algorithms are easily decomposed into a pipeline of operations. Furthermore, several stages of the pipeline can be decomposed into independent processes applied in parallel to separate subsets of the images. Parallel super-computers of the Delta class are used to study possible decompositions and the resulting speed-ups. Later, the algorithms are transferred to embedded processors for fine-tuning for the eventual application.

Accomplishments: For the simulation study, a performance evaluation methodology was designed, and an initial data set was collected. Important components of the stereo vision algorithms were ported to the Gamma and Delta computers, and an initial parallel decomposition of these components was implemented. Results to date with the Gamma machine show a nearly linear speed-up with the number of processors employed. This conforms to expectations, because the algorithm is dominated by computation, not by inter-process communication. The same stereo vision algorithms are being used to drive workload models for selecting or designing computer architectures for future space flight applications.

Significance: Prior to this research, JPL developed and demonstrated the first stereo vision system to be successfully used for real-time, semi-autonomous navigation of robotic vehicles. The current research will lead to sensor systems with higher accuracy and speed, enabling new applications of this sensor technology.

Status/Plans: Studies are continuing of accuracy and speed as a function of several algorithm design parameters. Results of these studies will be transferred to on-going robotic vehicle projects for NASA and other agencies. A number of extensions to the scope of the study are possible, including joint estimation of range and other scene properties, as well as the design of custom architectures for these computations.

Larry Matthies
Electronics and Control Division
Jet Propulsion Laboratory/
California Institute of Technology
(818) 354-3722



Parallelization of a Global Robot Motion Planning Algorithm

Objective: Perform Global Path Planning (DOF) for the high degree of freedom robot manipulators on Space Station Freedom.

Approach: There are two general Global Path Planning algorithms currently in use: cell decomposition (or configuration space) methods, and potential field methods. The computational requirements for configuration space methods grow exponentially with the number of degrees of freedom of the manipulator. This has led to the popularity of potential field methods due to their lower computational requirements. However, the potential field methods suffer from being local path planners, and are susceptible to not finding paths by getting stuck in local minima. With the advent of massively parallel machines, the computational requirements for configuration space methods may now actually be attainable.

A parallel algorithm is developed for an existing approximate cell decomposition method specifically designed for robot manipulators. The method requires generating a tree which represents the configuration space of the robot manipulator. Each subtree at any level can be computed independently of any other subtree. However, the amount of computation for each subtree is not known in advance. The parallel algorithm uses a coarse grain approach of allocating subtrees to processors. The algorithm uses a "card-playing" type scheduler, distributing subtrees to processors as they become available.

The Intel iPSC operating system allows having the scheduler be part of a node that is doing subtree calculations. Thus, there is no need to have a single processor dedicated to doing scheduling.

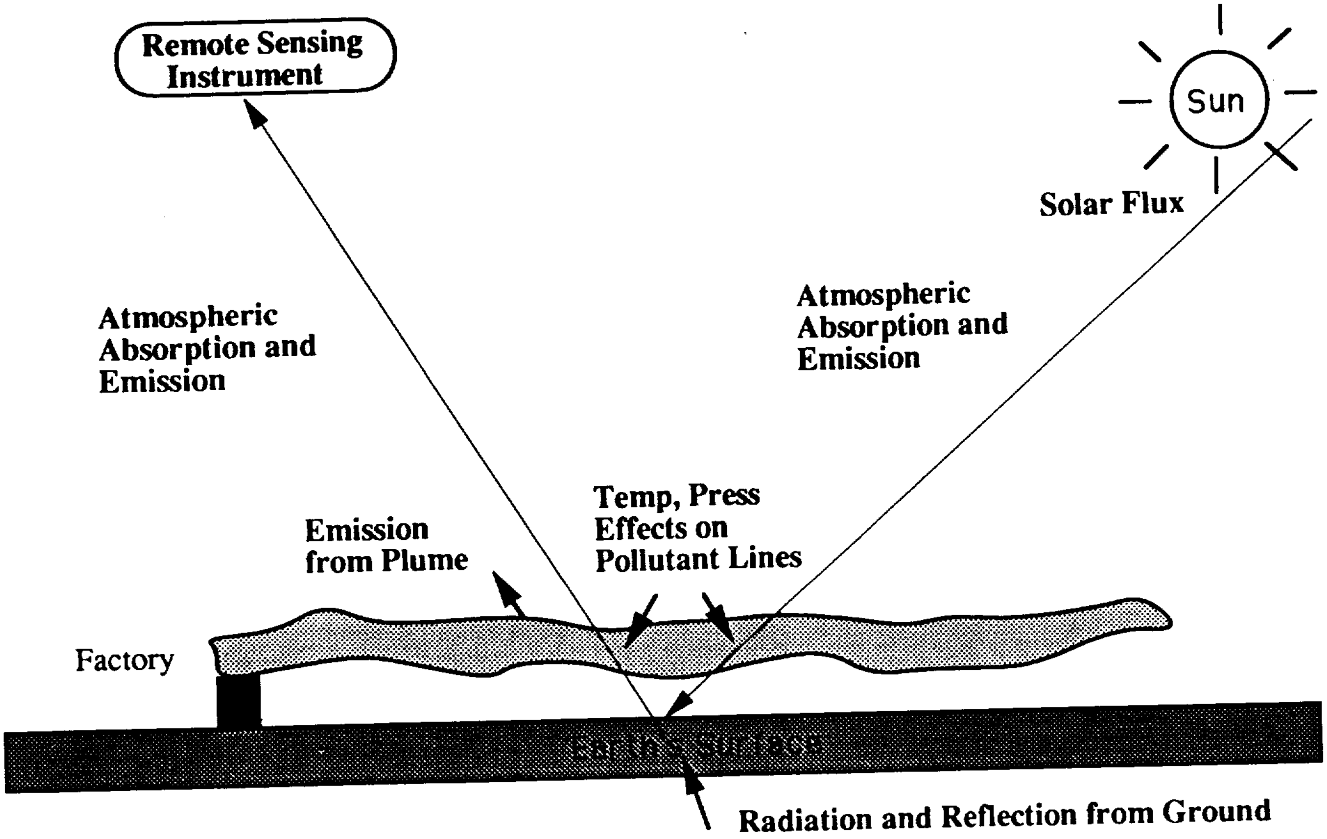
Accomplishments: The sequential version of the path planning algorithm was ported to the Intel iPSC/860 Gamma machine. The parallel algorithm is currently being implemented.

Significance: Space Station Freedom will be equipped with the Special Purpose Dexterous Manipulator (SPDM) consisting of a 4 DOF torso and two 7 DOF arms. In order to have the ability of specifying high level commands, a Path planner running in near real-time is required. Current path planners are usually limited to small DOF manipulators due to computational require-ments. Parallelizing the algorithm to run on massively parallel machines will allow the path planner to run in near real-time.

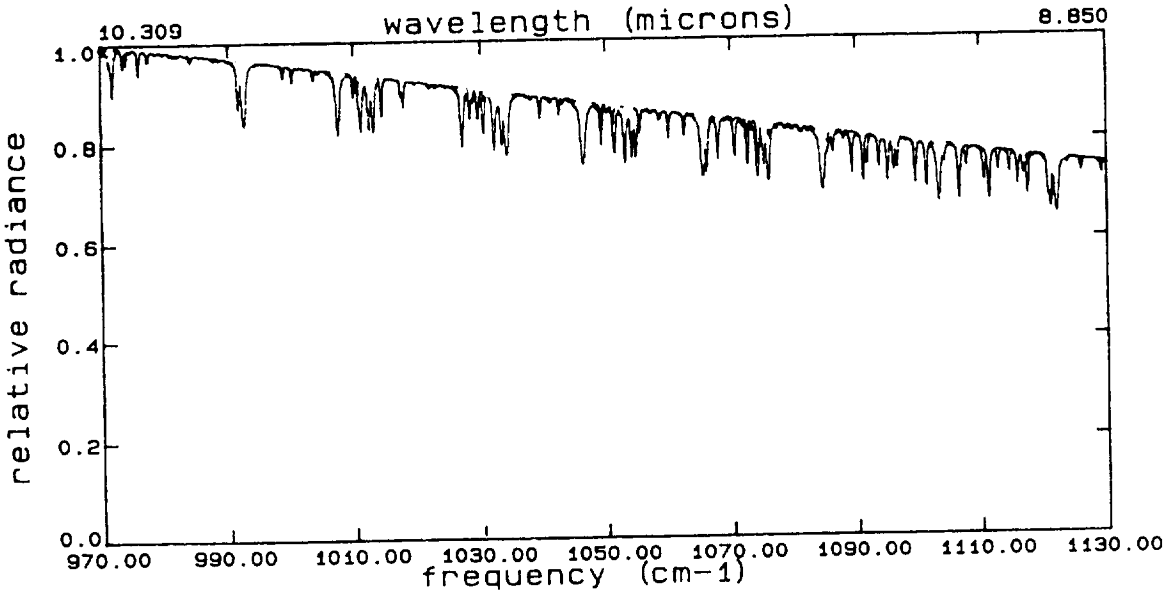
Status/Plans: Implementation of the parallel algorithm is continuing. After implementation, the algorithm will be benchmarked on the Intel Touchstone Delta Machine.

David Lim
Electronics and Control Division
Jet Propulsion Laboratory/
California Institute of Technology
(818) 354-3571

Radiative Transfer Process


Remote Sensing Instrument
Sun
Solar Flux
Atmospheric Absorption and Emission
Atmospheric Absorption and Emission
Temp, Press Effects on Pollutant Lines
Emission from Plume
Factory
Earth's Surface
Radiation and Reflection from Ground

Synthesized Spectrum



The Retrieval of Atmospheric Parameter Profiles from Remote Sensing Data

Objective: To develop a high performance algorithm for the analysis of remote sensing data by numerically iverting the radiative transfer equation governing the absorption and emission of IR radiation by the atmosphere. The results are used to characterize concentration, temperature, and pressure profiles in the Earth's atmosphere.

Approach: Spectral data is compared with a forward model of the atmospheric absorption and emission profile via a Square Root Information Filter (SRIF) algorithm. Parallelization of the code requires that a parallel version of a Two Triangular Matrix Householder Reduction (TTHH) be implemented. The broadening of spectral lines in the atmospheric spectra is modeled by use of a Voigt function algorithm.

Accomplishments: A parallel version of the TTHH algorithm was implemented and tested in the C programming language. Code for the TTHH and Voigt algorithms was input into a model of the i860 processor architecture by the JPL architectures modeling team. Preliminary studies of the execution time speed up and the efficiency of parallelization were performed for TTHH on the Gamma machine using eight i860 processors and on a sequential Sun Sparc2 workstation for the Voigt algorithm. Work load assessments of both TTHH and Voigt are in progress.

Significance: The very large volumes of remote sensing data gathered by the Earth Observing System and other future generations of instruments will render present methods of data analysis impractical and obsolete. The implementation of parallelized on-board processing techniques will greatly ease present constraints imposed by the bandwidth of the downlinking system. The capabilities added by on-board processing also will allow greater flexibility for remote sensing instruments to respond to episodic events which might otherwise go unobserved due to downlinking restrictions. Our work allows the tradeoffs between different on-board processing and data downlinking schemes to be assessed before decisions on flight hardware configurations need to be made.

Status/Plans: A report on the workload assessment and performance of the TTHH and Voigt algorithms is being prepared.

Larry Sparks
Suguru Araki
James McComb
Observational Systems Division

John Davidson
Electronics and Control Division
Jet Propulsion Laboratory/
California Institute of Technology
(818) 354-6194

(THIS PAGE LEFT BLANK INTENTIONALLY.)

Introduction to the National Research and Education Network (NREN)

Goal and Objectives: The NREN is both a goal of the HPCC Program and a key enabling technology for the success in the other components. The NREN is the future realization of an interconnected gigabit computer network system supporting HPCC. The NREN is intended to revolutionize the ability of U.S. researches and educators to carry out collaborative research and educational activities, regardless of the physical location of the participants or the computational resources to be used. As the name implies, NREN is a network for research and education, not a general purpose communication. Nonetheless, its use as a testbed for new communications technologies is vital. A fundamental goal of the HPCC is to develop and transfer advanced computing and communications technologies to the private sector of the U.S. as rapidly as possible and to enhance the nation's research and education enterprise.

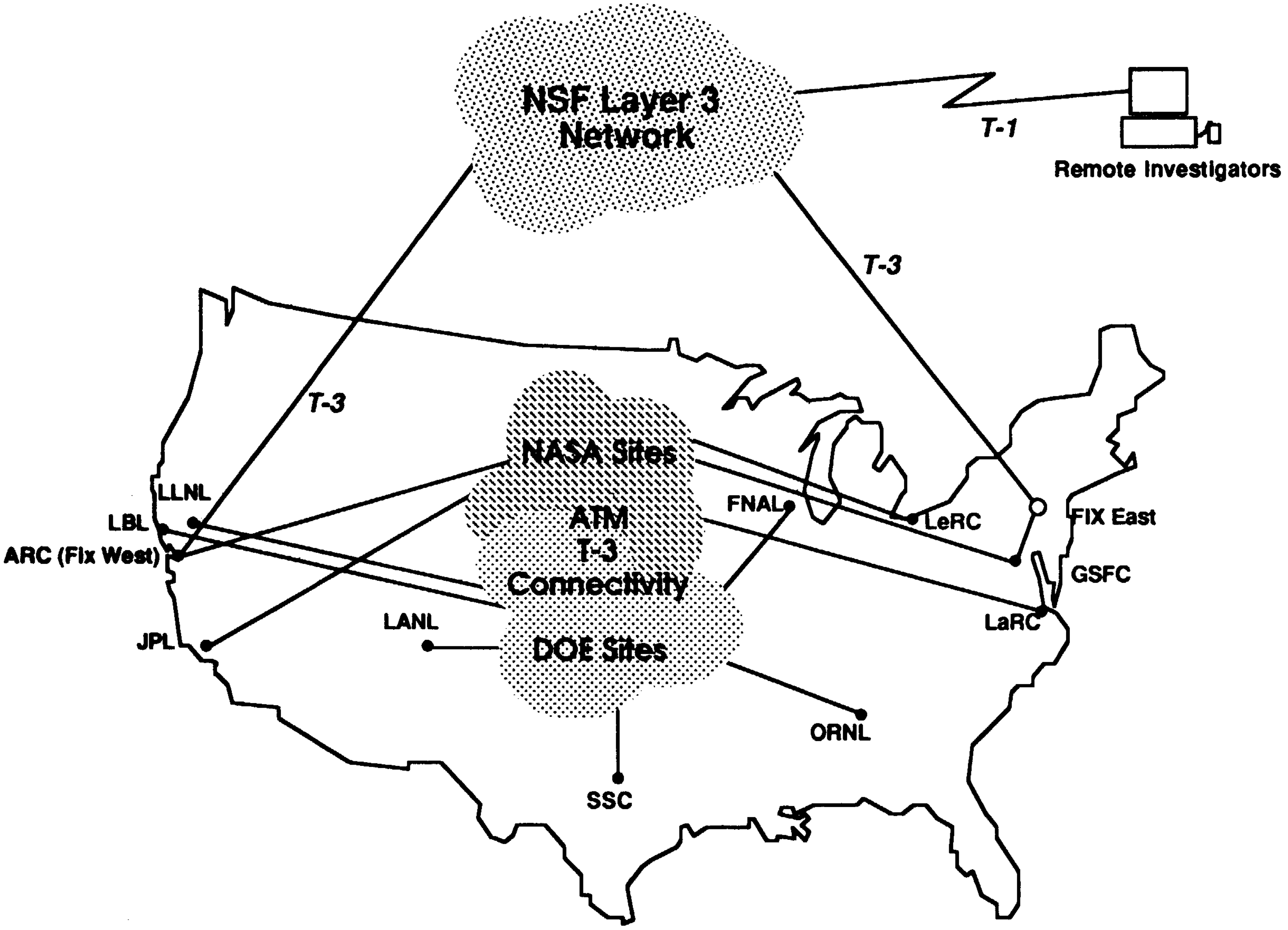
Strategy and Approach: The NREN represents a joint agency development effort that permits every participating agency to satisfy mission-critical requirements while also supporting an infrastructure that brings access to high performance systems at varying capability levels. The NASA effort will be directed at defining the engineering and management parameters to guide the architecture of the Interagency Interim NREN and at participating directly in the development and deployment of gigabit network technologies and architectures. In addition to addressing the science needs for very high bandwidth communications, the NREN Project will establish pilot programs with the K-12 educational community in order to discover the best mechanisms for distributing NASA information and science to the educational communities in the United States and provide practical models for the use of sophisticated computational and networking resources in these educational communities.

Organization: Specifically, each NASA Grand Challenge project wants high band–width connectivity to computational facilities at the NASA Centers, as well as to a widely dispersed number of researchers and principal investigators located at federal, university, and corporate sites. These requirements are compiled from surveys of the Computational Aeroscience (CAS) and Earth and Space Science (ESS) projects that are part of the NASA HPCC Program. These requirements will be met through a joint effort with Department of Energy in the establishment of a fast-packet network.

Besides providing an early move to fast-packet technology, NASA will use the joint DOE/NASA procurement to progress up the technology curve to higher data rate capabilities and eventually to gigabit per second speeds. This will involve the NASA NREN Project in a contributor or lead role in a wide variety of gigabit pilot efforts over the life of the Program.

Management Plan: The NREN Project is conducted or in collaboration with support from the Grand Challenge scientists from the CAS and ESS Projects, achieving direct involvement between NASA scientists and the K-12 educational community.

Point of Contact:
James Hart
JPL
(415) 604-6251


NSF Layer 3
Network
T-1
Remote Investigators
T-3
T-3
NASA Sites
ATM
T-3
Connectivity
DOE Sites
LLNL
LBL
ARC (Fix West)
JPL
LANL
FNAL
LeRC
FIX East
GSFC
LaRC
ORNL
SSC

Gigabit Network Testbeds

Objective: Involve the NASA NREN Project in a wide spectrum of gigabit pilot activities throughout the United States. Apply the knowledge gained from these networking technologies to the NASA Grand Challenges and science missions as quickly as possible.

Approach: Establish an understanding of all existing gigabit network pilot activities and contribute to those in which NASA has some directly relevant information, technical expertise, or geographic influence. Use the gigabit efforts to directly influence the direction of the IINREN implementation with Sprint and meld the two efforts whenever possible.

Accomplishments: The NASA NREN effort has an active role in the following gigabit testbed activities and is closely monitoring the five existing gigabit pilot activities sponsored by ARPA.

ACTS--we are exploring the use of satellite technology in the gigabit networking arena, and we are investigating the extension of the Sprint ATM fast-packet network via the ACTS.

ATDNet--a multi-agency network pilot in the Washington D.C. area which will connect to Goddard.

BAGnet--a Bay Area gigabit networking effort focused on multimedia seminar sharing with significant involvement from a wide spectrum of corporate, federal and academic organizations.

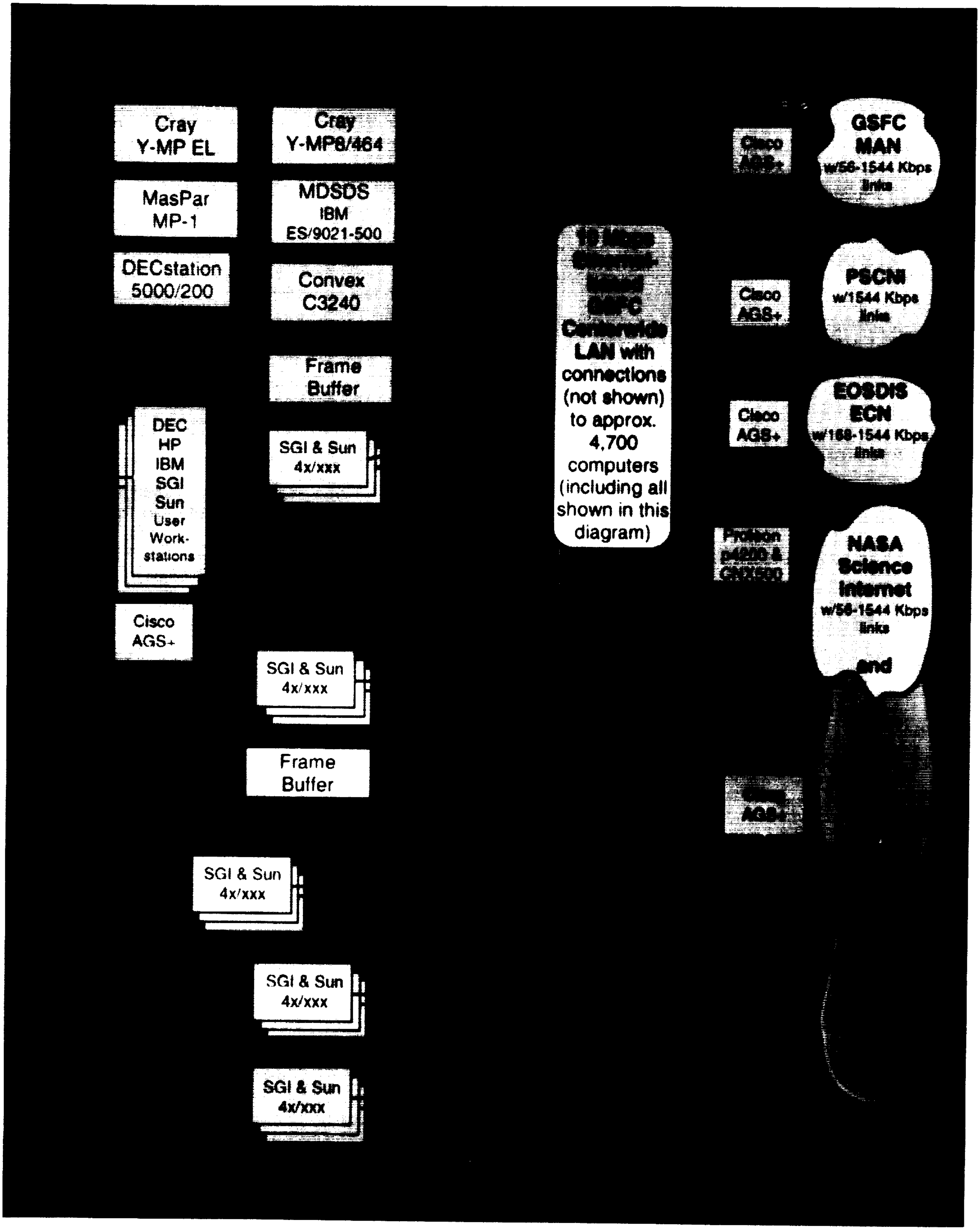
MAGIC--a Sprint-based network stretching from Minnesota to Kansas with connectivity to federal government sites and the Sprint testbeds.

NASA-LLNL-SprintTestbed--Sprinttestbedaccess granted as part of the DOE/NASA procurement that will give early access to advanced technologies at the Ames Research Center and expected connectivity to MAGIC and other high bandwidth test networks.

Significance: The NASA NREN Project is involved and monitoring all major gigabit activities, and will leverage this involvement into enhanced connectivity for NASA scientists and a greater understanding of future telecommunications technologies.

Status/Plans: Aggressively implement the NASA-LLNL-Sprint testbed at Ames so that the expected networking advances, and their significance, can be demonstrated for the NASA management and scientific staff. Complete a plan for ACTS involvement in the Sprint ATM network and continue interaction with all other gigabit testbed activities.

James Hart
Information and Communications Systems Division
Ames Research Center
(415) 604-6251


Cray
Y-MP EL
Cray
Y-MP8/464
MasPar
MP-1
MDSDS
IBM
ES/9021-500
DECstation
5000/200
Convex
C3240
Frame
Buffer
DEC
HP
IBM
SGI
Sun
User
Work-
stations
SGI & Sun
4x/xxx
Cisco
AGS+
SGI & Sun
4x/xxx
Frame
Buffer
SGI & Sun
4x/xxx
SGI & Sun
4x/xxx
SGI & Sun
4x/xxx
10 Mbps
Ethernet-
based
GSFC
Centerwide
LAN with
connections
(not shown)
to approx.
4,700
computers
(including all
shown in this
diagram)
Cisco
AGS+
GSFC
MAN
w/56-1544 Kbps
links
Cisco
AGS+
PSCNI
w/1544 Kbps
links
Cisco
AGS+
EOSDIS
ECN
w/168-1544 Kbps
links
Proteon
p4200 &
CNX500
NASA
Science
Internet
w/56-1544 Kbps
links
and
Cisco
AGS+

Readied 100 Mbps Intra-Center Network at GSFC for Initial NREN Connection

Objective: Provide high performance connectivity between GSFC-based ESS Testbed Systems/Investigators and the NASA NREN connection to GSFC.

Approach: Plan, with the NREN Project, an end-to-end national architecture and a phased implementation which meets ESS requirements for high performance connectivity and interoperability among its Testbed Systems and the workstations of its Investigators. Co-operate with and leverage use of the GSFC institutional networking activities (CNE Project/Code 500) to develop an intra-Center high performance network architecture which meets the leading edge requirements of ESS, and over time can be easily extended to accommodate additional needs at GSFC. Assess the readiness of new technologies to meet the higher network bandwidth and easier use requirements of ESS through participation in advanced networking testbeds.

Accomplishments:

□ Developed an architectural plan for extending and interfacing GSFC 100 Mbps FDDI and 1000 Mbps UltraNet networks to the ESS MasPar MP-1 and Cray Y-MP/EL Testbed Systems in building 28 and to the demarc in building 1 identified for the initial NASA NREN connection of 45 Mbps.

□ Designed, installed, and assessed a FDDI network within building 28. Connected it to GSFC's new inter-building FDDI network which extends to building 1, and connected it to the UltraNet through a successfully beta-tested interface.

□ Involved ESS in planning for the ATDnet as an advanced technology demo of a very high speed metropolitan area network (MAN) using SONET/ATM technology to interconnect DARPA, DIA, DISA, NRL, NSA, and GSFC.

Significance: This activity is a key component in the evolution toward shared use of remote network resident resources by the science community. Whether GSFC Investigators use remote testbeds, remote Investigators use GSFC testbeds, Investigations move data between distributed archives and testbeds or link various combinations of distributed testbeds, or high performance computing is used in some other scenario, an evolving high performance transparent GSFC network is an essential component.

Status/Plans: In FY93, complete installation and initiate operational use of the ESS Testbed System connections with FDDI and the 45 Mbps link between GSFC and the NSFnet. Also, complete interfacing and initiate operational use of the initial NASA NREN connection to GSFC of 45 Mbps. Throughout FY93, develop designs with the HPCC NREN Project on technically feasible local area interfaces to NASA NREN connections at 155 Mbps, which are expected to be available in early 1994. Also in FY93, depending on DoD funding, initiate 155 Mbps SONET/ATM connections to the ADTnet MAN to gain early experience in local area interfacing, managing, and utilization of this NREN-related technology.

Pat Gary
Goddard Space Flight Center
Code 930.6
pgary@dftnic.gsfc.nasa.gov
(301) 286-9539

(THIS PAGE LEFT BLANK INTENTIONALLY.)

Introduction to Basic Research and Human Resources Project (BRHR)

Goal and Objectives: The goal of the BRHR project is to support the long term objectives of the HPCC Program with substantial basic research contributions and by providing individuals fully trained to utilize HPCC technology in their professional careers. This program is pursued through basic research support targeted at NASA's Computational Aerosciences Project and the Earth and Space Sciences Project.

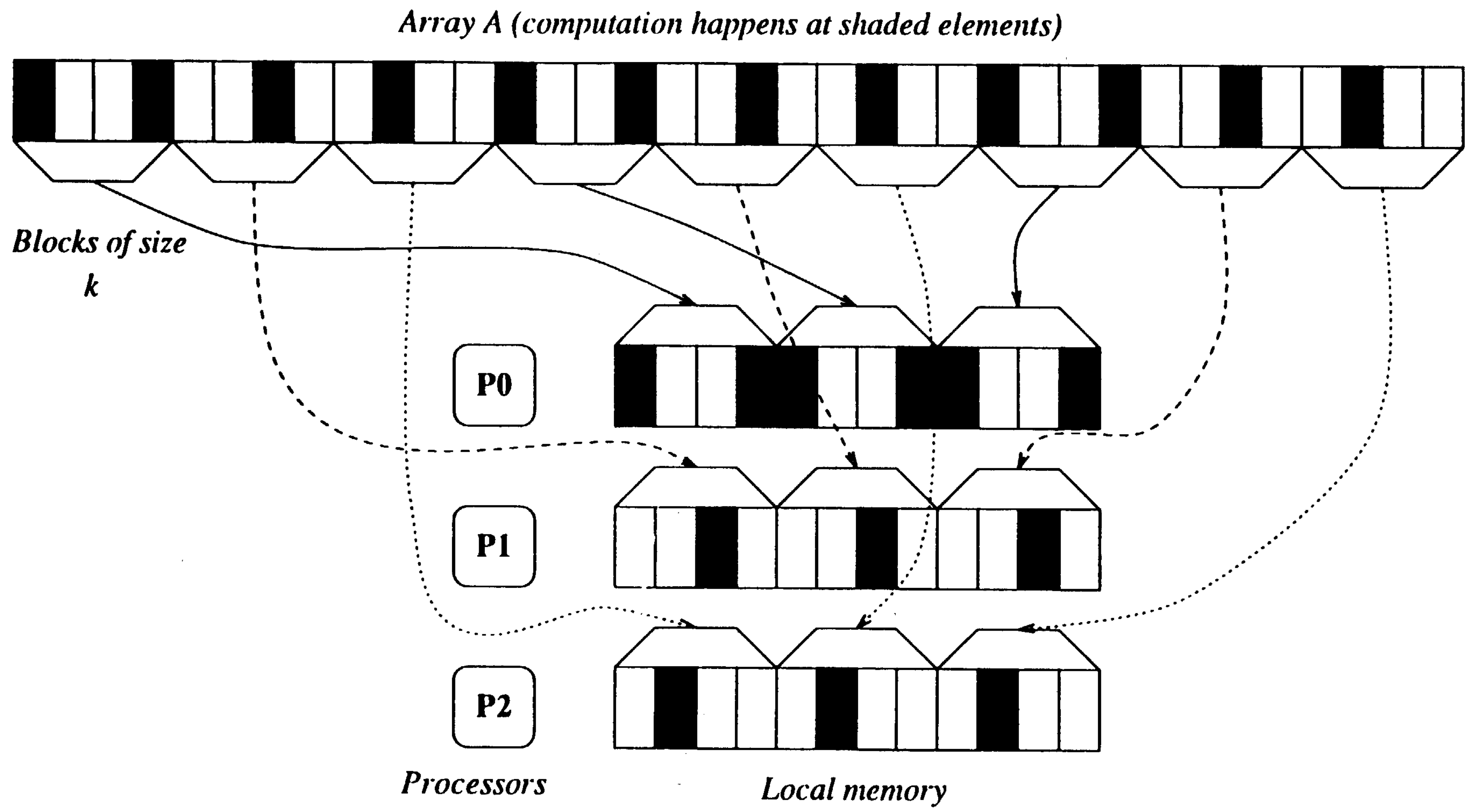
Strategy and Approach: BRHR promotes long-term research in computer science and engineering, while increasing the pool of trained personnel in a variety of scientific disciplines. BRHR encompasses a diversity of research activities at NASA centers and US universities and spans a broad educational spectrum: kindergarten through secondary education (K-12); graduate student research opportunities; post-doctoral study programs; and basic research conducted by experience professionals. This project encourages diverse approaches and technologies, with a focus on software technology and algorithms, and leverages the ongoing NASA research base.

In Fiscal Year 1992, the BRHR project initiated programs for graduate and post doctoral student research. The project also seeded K-12 efforts/pilot projects through several NASA research centers. BRHR is planned to further expand in these efforts while being integrated into the Computational Aerosciences and Earth and Space Science projects. In addition, BRHR produced fundamental research results as reported on in this section of the HPCC annual report.

Organization: NASA Headquarters serves as the lead for the BRHR Project with support from the following NASA Centers: Ames Research Center (ARC); Langley Research Center (LaRC); Goddard Space Flight Center (GSFC); Marshall Space Flight Center (MSFC); Lewis Research Center (LeRC); and, the Jet Propulsion Laboratory (JPL). BRHR has research projects at the following NASA supported research institutes: the Institute for Computer Applications in Science and Engineering (ICASE) at LaRC, the Research Institute for Advanced Computer Science (RIACS) at ARC and the Center of Excellence in Space Data and Information Sciences (CESDIS) at GSFC.

Management Plan: The project is managed by the BRHR Project Manager at NASA Headquarters who coordinates developments with NASA centers, the HPCC projects, and other Federal agencies and departments participating in the national HPCC program.

Point of Contact:
Paul Hunter
NASA Headquarters
Washington, DC
(202) 358-4618



Problem: Determine the sequence of local memory accesses for each processor.

Compiler and Runtime Optimization for Data-Parallel Languages

Objective: To develop compiler optimization for automatically distributing array data on a distributed-memory MIMD and SIMD parallel machine to minimize communication due to misaligned operands of operations and to develop routine techniques for addressing generation and efficient collective communication in support of the compile-time optimizations using the data-parallel language High Performance Fortran (HPF).

Approach: The communication patterns of an HPF program on a distributed-memory parallel machine are captured in a metric space abstraction, where the metric models the communication characteristics of the machine. A recurrence equation on communication costs naturally leads to a dynamic programming algorithm for determining optimal, or near-optimal, data layouts for variables and intermediate results that minimize communication cost. For most practical communication metrics of interest (discrete, grid, ring, fat-tree), the dynamic programming algorithm can be sped up further by exploiting the structure of the metric, resulting in a class of compact dynamic programming (CDP) algorithms whose running times are low-order polynomials in the size of the data flow graph of the program, but are independent of the number of processors in the machine.

On the runtime side, a technique based on finite-state machines was developed for generating local addresses and structured communication patterns for the general two-level mappings and block-cyclic distributions in HPF.

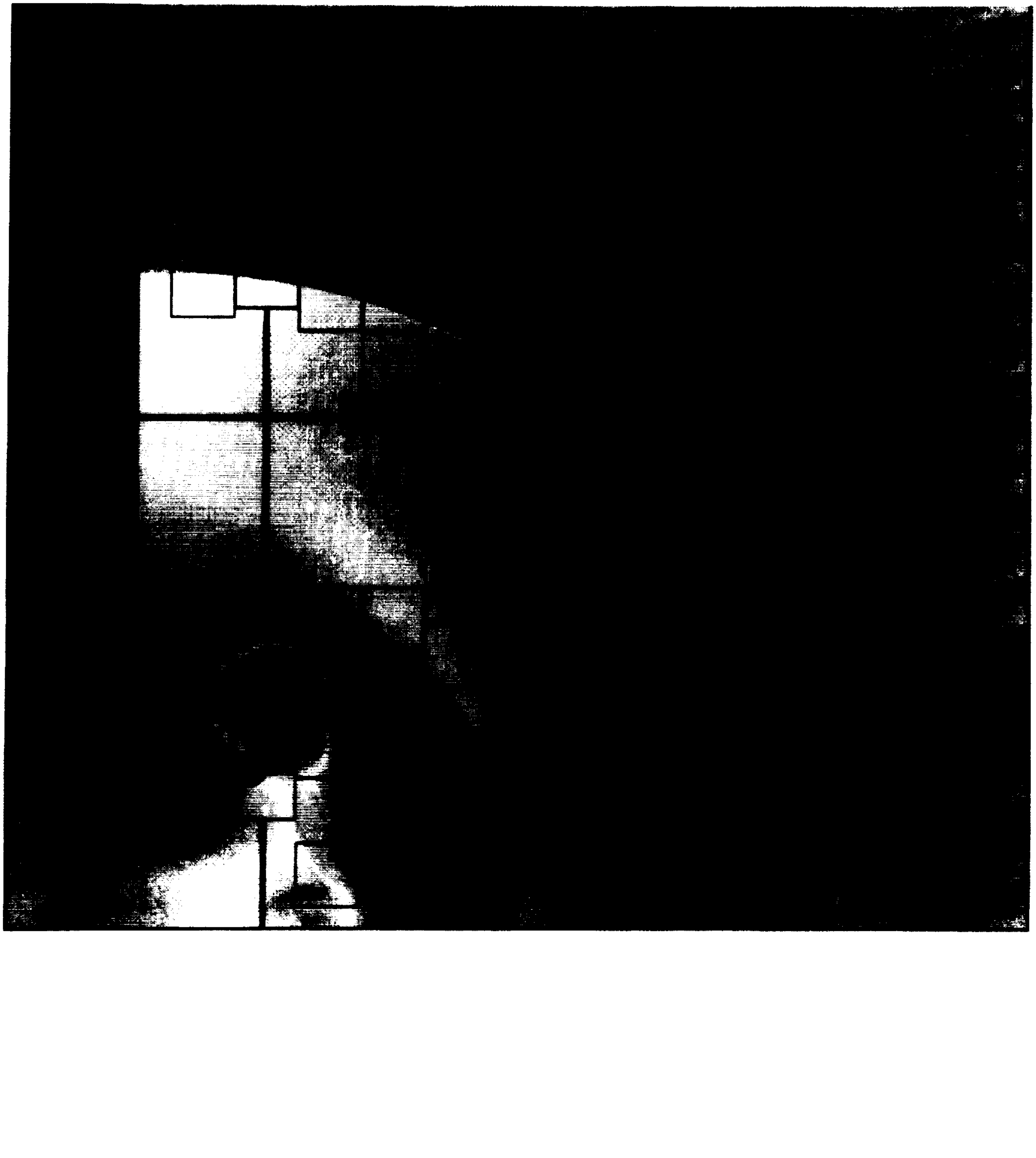
Accomplishments: A comprehensive theory of CDP algorithms was developed. The theory handles a variety of communication metrics. It applies uniformly to single program statements, basic blocks or code, and control flow. It handles transformation array operations such as reductions, spreads, transpositions and shifts. It extends naturally to dynamic layouts that are infinite functions of loop induction variables, thereby accounting for replication, privatization, and realignment of arrays. A prototype implementation of some of the algorithms demonstrated the validity of the approach and confirmed that they can be implemented efficiently. The runtime technique was prototyped on the iPSC/860 and was demonstrated to be extremely efficient.

Significance: The current state of compiler technology requires programmers to annotate programs with alignment and distribution directives indicating how the programmer feels arrays ought to be distributed for minimum communication. Our theory shifts the task from the programmer to the compiler, provides a rigorous framework for automatically determining optimal alignments and promotes portability.

Our runtime technique for address and communication generation is the first to handle the full mapping schemes in HPF in a general and efficient manner. While solutions to various special cases have been known and implemented, ours is the first general solution of the problem. Our solution is being studied by the compiler groups at several major US vendors of parallel machines for possible inclusion in their extra space systems.

Status/Plans: A prototype compiler is being developed and implemented to demonstrate the capabilities of the optimization. To promote software reuse, the first version of the compiler is structured as a directives generator for HPF compilers. We expect future versions of the compiler to be integrated more smoothly into HPF compilers. Further algorithmic research will focus on inter procedural analysis, automatic determination of block sizes, and code generation issues.

Robert Schreiber
Siddartha Chatterjee
RIACS
Ames Research Center
(415) 604-3965



Parallel Implementation of an Adaptive Mesh Refinement (AMR) Method on the CM-2

Objective: To devise a method to efficiently implement AMR methods, with their complicated data structures and communication patterns, on massively parallel SIMD machines.

Approach: To devise a mapping which stores points on the various grids that need to communicate in the same (or nearby) processor, and eliminate distant routing.

Accomplishments: The mapping problem was solved, a parallel implementation completed and near linear speedup obtained.

Significance: AMR methods with much more efficient approximation of the solution can be used on parallel machines.

Status/Plans: To use the structure of AMR for coarse grain parallelism in a MIMD mode, coupled with massive parallelism as used in the SIMD mode used here, to achieve near optimal load balancing over the entire program (possible on CM-5 but not on CM-2).

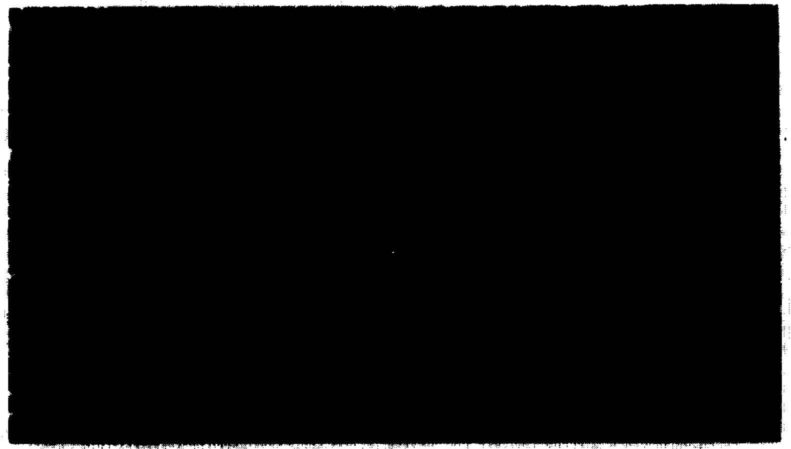
Joseph Oliger
RIACS
Ames Research Center
(415) 604-4992

EXAMPLE: 3-D ADAPTIVE GRID REFINEMENT AND COARSENING

FSMACH = 0.4 ALPHA = 0.0 deg

SYMMETRIC NACA 0012 WING

INITIAL MESH: 16481 EDGES



FIRST REFINEMENT: 24718 EDGES

SECOND REFINEMENT AND COARSENING: 23185 EDGES

Rupak Biswas - RIACS
Roger Strawn - US Army AFDD

Dynamic Mesh Adaptation Methods for 3-D Unstructured Grids

Objective: To develop efficient techniques for the fast refinement and coarsening of three-dimensional unstructured grids. These methods will be implemented on parallel computers and used to solve realistic problems in helicopter aerodynamics.

Approach: The unsteady 3-D Euler equations are solved for helicopter flow fields using an unstructured grid. Error indicators are used to identify regions of the mesh that require additional resolution. Similarly, regions with low errors are targeted for coarsening. The object is to optimize the distribution of mesh points so that the flow field is accurately modeled with a minimum of computational resources. The mesh coarsening and refinement algorithm is the key to the success of this procedure. The data structure for this algorithm is implemented in the C programming language and consists of a series of linked lists. These linked lists allow the mesh connectivity to be rapidly reconstructed when individual mesh points are added or deleted. It also allows for anisotropic refinement of the mesh.

Accomplishments: A preliminary version of the mesh coarsening ad refinement algorithm had been developed and tested for sample problems in 3-D. One such result is shown in the figure on the facing page. In this case, an initial solution was obtained for the Euler equations on a coarse mesh. Regions with high errors were then targeted for refinement by estimating the density gradients along each edge of the mesh. New grid points were added in regions where the error is high, and the calculation was continued on the new mesh. In the final step, the mesh was simultaneously coarsened and refined. Error indicators based on density gradients were again used to target the points to be added and deleted. This results in a more optimal distribution of points for the mesh. Note that the total mesh size is similar for the last two computations. The final mesh size is similar for the last two computations. The final mesh yields a much more accurate solution, however.

Significance: Aerodynamics calculations performed on structured-grids have difficulties resolving localized flow field features such as shocks, vortices, and aerodynamic waves. Unstructured models can make use of localized mesh refinements to resolve these flow features. However, this mesh refinement is only effective if it can be performed efficiently in three dimensions. This new procedure for dynamic mesh adaptation directly addresses this problem by using an innovative data structure that is well suited for large-scale computations. When coupled with a 3-D unstructured grid Euler solver, the mesh adaptation scheme will provide accurate solutions for complex aerodynamic flow fields.

Status/Plans: The dynamic mesh adaptation scheme is currently being tested for large problems on a Cray Y-MP computer. Particular attention is focused on computer CPU time and memory requirements. Future work will test the performance of the mesh adaptation scheme and 3-D Euler solver on a massively parallel computer system.

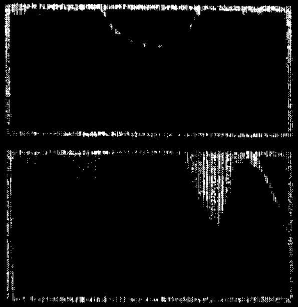
Rupak Biswas
RIACS
NASA Ames Research Center, MS T045-1
Moffett Field, CA 94035-1000
(415) 604-4411

Roger Strawn
US Army AFDD, ATCOM
NASA Ames Research Center, MS 258-1
Moffett Field CA 94035-1000
(415) 604-4510

NASA Summer School in
High Performance Computational Sciences

Techniques & Methods

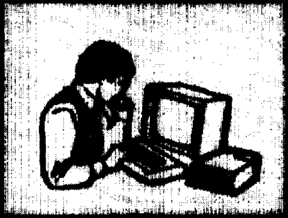
- Vortex Methods
- Unstructured Grid Methods
- Free Lagrangian Techniques
- Particle-in-Cell Techniques
- Adaptive Mesh Refinement
- Shock Capturing Methods



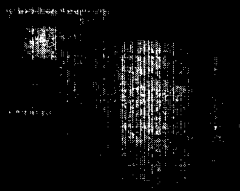
Summer School Goal

Train the next generation of physicists in massively parallel techniques and algorithms to enable them to exploit High Performance Computing and Communications (HPCC)

[illegible]



[illegible]



Lab sessions in code development given by the vendors
- MasPar Computer Corporation
- Thinking Machines Corporation

Established NASA Summer School for High Performance Computational Sciences

Objective: Train the next generation of physicists in massively parallel techniques and algorithm development to support the goals of the HPCC Program.

Approach: The NASA Summer School for High Performance Computational Sciences is an intensive three-week program of lectures and lab sessions held at the NASA Goddard Space Flight Center in the early summer. It is jointly sponsored by HPCC/ESS and Goddard's Space Data and Computing Division. Sixteen students are selected through a national solicitation performed by USRA. They must be working toward PhDs in a physical science discipline and have an interest in utilizing massively parallel computer architectures to solve problems within their respective fields. The students are brought to Goddard from their home institutions and housed at the University of Maryland during the program. A number of the world's leading computational scientists have served as instructors (H. Trease of LANL, R. Lohner of GWU, S. Zalesak of GSFC, S. Baden of Univ. of CA, A. Mankofsky of SAIC, P.Colella of UC Berkeley). Lectures focus on advanced techniques in computational science, with special emphasis on computational fluid dynamics and on algorithms for scalable parallel computer architectures. Access to scalable computer systems is provided to serve as teaching platforms. The vendors of the selected systems are brought in to give lectures and hands-on workshops on code development for their product.

Accomplishments:

- □ Students became functional in the art of "thinking parallel".
- □ They became knowledgeable in advanced computational techniques.
- □ They became hooked on the power of emerging scalable parallel systems.
- □ Most students requested continued access to Goddard's parallel testbed computers.

Significance: The two sessions that the school has operated have assisted the ESS Project to understand the formal training requirements of intelligent but novice users of scalable parallel systems, and to evolve a suitable curriculum to meet these needs.

Status/Plans: Based on the success of the program and the positive reactions of the students, ESS plans to continue the Summer School in FY93 and is considering its expansion to provide training for members of the Investigator teams selected through the ESS NRA.

Dan Spicer
Goddard Space Flight Center Code 930
spicer@vlasov.gsfc.nasa.gov
(301) 286-8541

FY 1992 HPCC GRADUATE STUDENTS

STUDENT	UNIVERSITY	PROPOSAL	CENTER	SUPPORT
S. BERRYMAN	YALE	UNSTRUCTURED MESH AND SPARSE MATRIX PROBLEMS ON PARALLEL COMPUTERS	LARC	HPCC
P. HOMER	ARIZONA	CONSTRUCTING SCIENTIFIC APPLICATIONS AS HETEROGENEOUS, DISTRIBUTED PROGRAMS	LERC	HPCC
J. ROBICHAUX	ILLINOIS	NUMERICAL SIMULATION OF BLUFF BODY WAKE FLOWS USING A SPECTRAL DECOMPOSITION METHOD	LERC	UMF*
D. SMITH	CARNEGIE MELLON	VIRTUAL ARCHITECTURE FOR HIGH PERFORMANCE IMAGE PROCESSING	GSFC	HPCC
J. TRUJILLO	PRINCETON	PARALLEL SPECTRAL ELEMENT ALGORITHM FOR SIMULATION OF FLOWS IN TIME-DEPENDENT DOMAINS	LERC	UMF*
J. WEISSMAN	VIRGINIA	APPLYING PARALLEL OBJECT-ORIENTED COMPUTATION TO SCIENTIFIC COMPUTATION ON MIMD ARCH.	JPL	HPCC
A. ZIMMERMAN	STANFORD	MULTIPROCESSOR ARCHITECTURES	ARC	HPCC

*UMF = UNDERREPRESENTED MINORITY FOCUS

High Performance Computing and Communications-Graduate Student Research Program

Objective: The HPCC-sponsored Graduate Student Research Program (HPCC-GSRP) is an effort to involve the next generation of entry level practitioners of HPCC techniques and methods in NASA specific applications, in an attempt to secure their interest early in their professional careers, to NASA's long- term benefit. It is anticipated that the exposure to NASA's research needs, and the collaboration with NASA scientists and engineers through this program, will produce long lasting relationships between this next generation of computational scientists and NASA researchers.

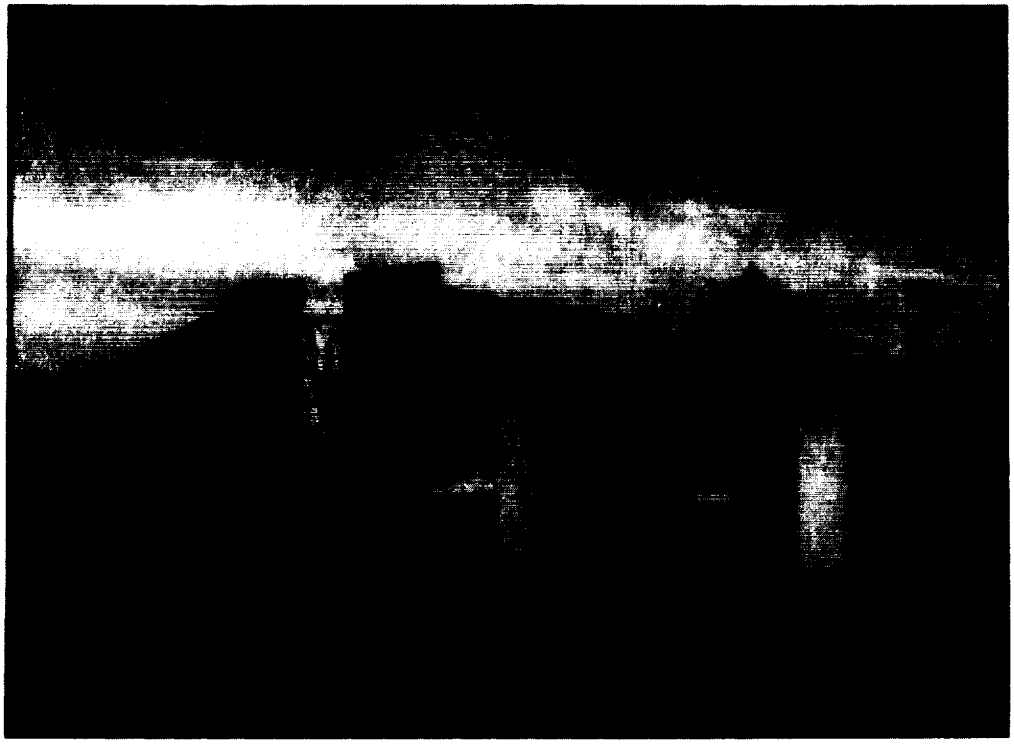
Approach: To maximize the payoff of the funding available, the HPCC-GSRP has been established as an HPCC specific add-on to the existing NASA Graduate Student Research Program. This enables the HPCC program to reach a much broader community than would be possible for a more stand-alone effort.

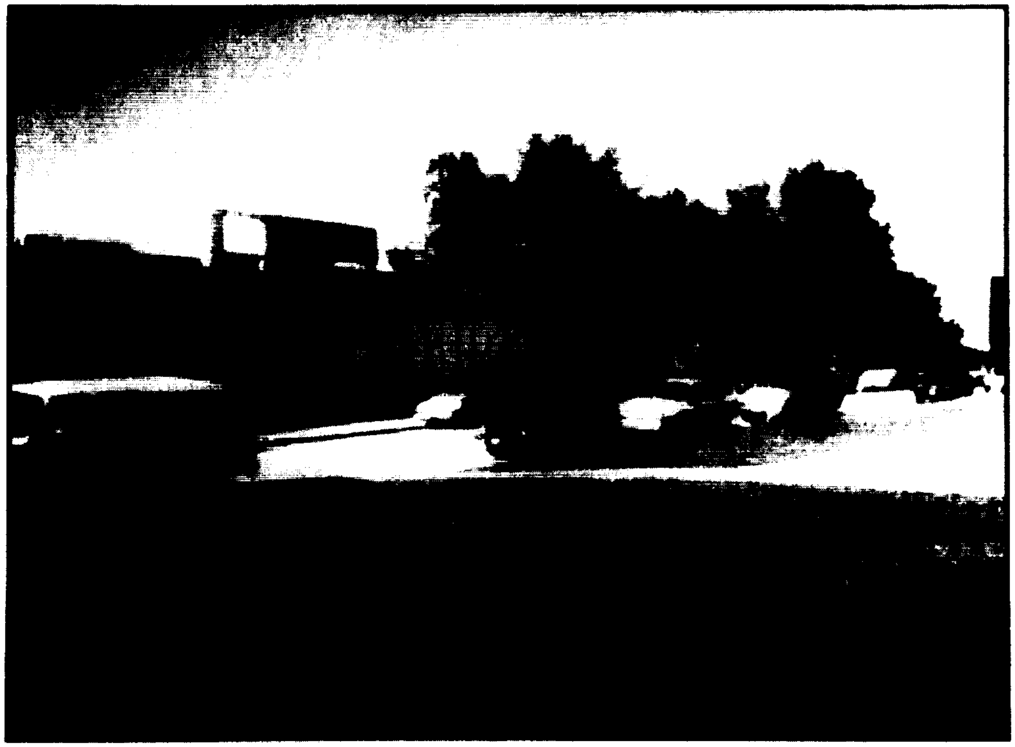
Accomplishments: In FY92, seven students were selected from U.S. universities for funding on HPCC research projects. The selected research topics represented a diversity of interest, indicating the breadth the of HPCC applications.

Significance: FY92 was the first year of the HPCC-GSRP, and it was established too late to be incorporated in the mailing of the "Call for Proposals" for the previously existing NASA GSRP. None-the-less, more high quality proposals were received than could be supported in FY92, indicating a healthy depth of research talent NASA can tap into given sufficient resources.

Status/Plans: The HPCC-GSRP is an ongoing program, and the students selected in FY92 will be supported in FY93, assuming sufficient progress has been made toward their objectives. In addition, a similar number of new awards will be made in FY93. The HPCC-GSRP in now incorporated as a special chapter in the NASA-wide Graduate Student Research Program announcement that is distributed nationwide biannually.

Paul Hunter
NASA Headquarters, Code RC
Washington, DC 20546
(202) 358-4618





High Performance Computing and Communications K-12 Outreach Program

Objective: The HPCC K-12 outreach program encompasses NASA's efforts to bring the benefit of HPCC technology to the classrooms of America at the earlier stages of students' education to capture their interest in the exciting possibilities of careers in computational science and engineering in order to attract high caliber students at an early age to NASA and aerospace careers.

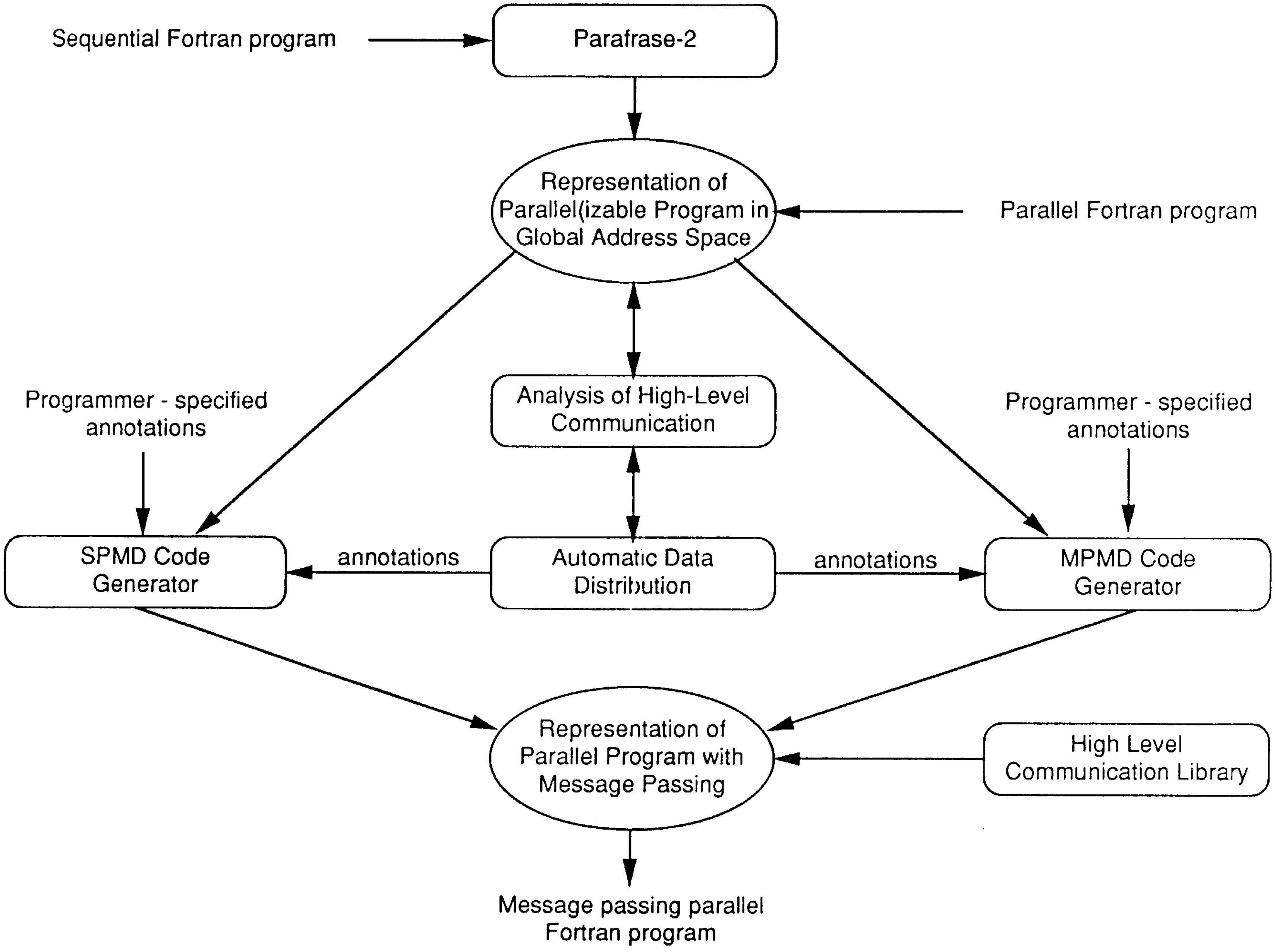
Approach: The approach to starting this effort has been to build upon existing NASA resources and programs, such as Spacelink (a NASA supported database of educational materials related to NASA's science and aeronautics activities), the Teacher Resource Centers at each NASA center, and the Aerospace Education Specialist Program.

Accomplishments: The first steps in this effort were taken in FY92 by selecting two pilot schools (Monta Vista High School, California and Thomas Jefferson High School, Virginia) to begin the development of networking supported research and education at the high school level. In addition, Spacelink was upgraded to 800 dial in service to facilitate access, and NREN training was established for the NASA Aerospace Education Specialist program.

Significance: The improvement in the access to Spacelink will greatly enhance the educational community's ability to access this NASA resource by eliminating the most significant practical barrier today: cost of connection. However, because of the limited capacity of the 800 number established (32 lines at ARC and 16 at GSFC), near term access will be granted to teachers only (controlled by a password protection mechanism). The pilot school program is the baseline for developing future high school programs within NASA's HPCC Program.

Status/Plans: Having started several efforts in FY92, the next few years will build upon and expand these efforts. Specifically, in FY 93, NASA will identify NASA owned or controlled assets to deploy on the NREN for education, conduct several workshops in K-12 asset and curricula development, integrate the HPCC effort with NASA and Federal plans for education in general, and assist other educational personnel such as the Aerospace Education Specialists in utilizing HPCC technology and products to accomplish their mission.

Paul Hunter
NASA Headquarters, Code RC
Washington, DC 20546
(202) 358-4618


Sequential Fortran program
Parafrase-2
Representation of Parallel(izable Program in Global Address Space
Parallel Fortran program
Analysis of High-Level Communication
Programmer - specified annotations
Programmer - specified annotations
SPMD Code Generator
annotations
Automatic Data Distribution
annotations
MPMD Code Generator
Representation of Parallel Program with Message Passing
High Level Communication Library
Message passing parallel Fortran program

Overview of PARADIGM compiler project

Portable Parallel Algorithm Design Framework

Objective: Develop an automated means to convert explicitly parallel programs written for shared multiprocessor models for execution on distributed memory multicomputers.

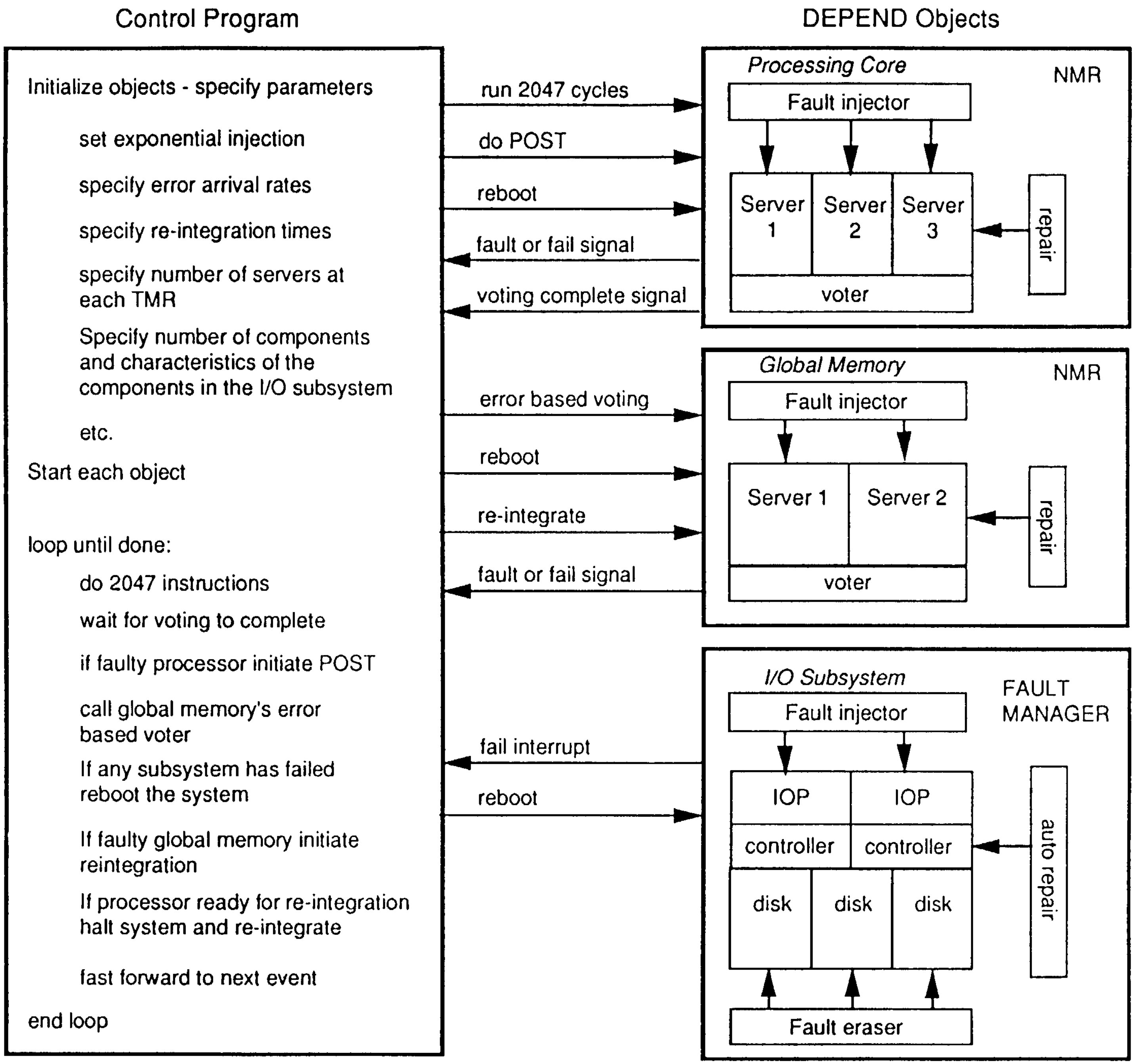
Approach: Distributed memory multiprocessors are increasingly being used for providing high levels of performance for scientific applications by connecting several thousand off-the-shelf, powerful microprocessors through simple low-cost interconnection networks. These distributed memory machines offer significant advantages over the shared memory multiprocessors in terms of cost and scalability, but they are much more difficult to program than shared memory machines since the programmer has to distribute code and data on the processors manually, and manage communication tasks explicitly. The approach to make the programming of distributed memory machines easier is to explore compiler technology in an effort to automate the translation process from code written for shared memory multiprocessors to code that will run on distributed memory machines.

Accomplishment: Significant progress has been made in several areas: 1) Development of a basic compiler that takes a shared memory program written in FORTRAN and, given user directives for data distribution, generates a message passing program; 2) Development of automated code partitioning techniques for functional parallel multiple instruction multiple data parallelism (MIMD); 3) Development of automated data partitioning techniques for both MIMD and single instruction multiple data (SIMD) programs; and 4) Development of automated compiler techniques to generate high level communication functions from shared memory parallel programs.

Significance: These techniques will lead to the support of efficient, portable, scalable, parallel programming of the massively parallel distributed memory machines of the future.

Status/Plans: Future plans are to integrate the above mentioned technologies into the PARADIGM (PARAllelizing compiler for Distributed memory General purpose MIMD processors) compiler that will generate portable code for a wide variety of distributed memory multicomputers. Subsequently, the performance of the PARADIGM compiler will be evaluated and benchmarked on large scientific FORTRAN applications.

Dr. Prithviraj Banarjee
Center for Reliable and High Performance Computing
University of Illinois, Urbana-Champaign, IL
(217) 333-6564


Control Program
DEPEND Objects
Initialize objects - specify parameters
set exponential injection
specify error arrival rates
specify re-integration times
specify number of servers at each TMR
Specify number of components and characteristics of the components in the I/O subsystem
etc.
Start each object
loop until done:
do 2047 instructions
wait for voting to complete
if faulty processor initiate POST
call global memory's error based voter
If any subsystem has failed reboot the system
If faulty global memory initiate reintegration
If processor ready for re-integration halt system and re-integrate
fast forward to next event
end loop
run 2047 cycles
do POST
reboot
fault or fail signal
voting complete signal
error based voting
reboot
re-integrate
fault or fail signal
fail interrupt
reboot
Processing Core
NMR
Fault injector
Server 1
Server 2
Server 3
repair
voter
Global Memory
NMR
Fault injector
Server 1
Server 2
repair
voter
I/O Subsystem
FAULT MANAGER
Fault injector
IOP
IOP
controller
controller
auto repair
disk
disk
disk
Fault eraser

The DEPEND simulation model of the Tandem Integrity S2 architecture

Design Environment for Dependable High Performance Systems

Objective: Develop a design environment for the development of high-performance computing systems that can analyze both reliability and performance issues from the conceptual design stage through the simulation and prototype testing stage.

Approach: The approach was to develop a Computer-Aided Design (CAD) tool based on object-oriented techniques. The simulation-based tool would provide facilities to model components typically found in fault-tolerant systems.

Accomplishment: A simulation-based CAD tool called DEPEND was developed which allows a designer to study a complete computer system in detail. DEPEND provides an object-oriented framework for system modeling. Facilities are provided to rapidly model components found in fault-tolerant systems. DEPEND provides an extensive, automated fault injection facility which can simulate realistic fault scenarios. For example, the tool can inject correlated and latent errors, and it can vary the injection rate based on the workload on the system. New methods, based on time acceleration and hybrid simulation, were developed which allow simulation of extended periods of operation, before not possible. The time acceleration technique simulates the time region around a fault in great detail and then leaps forward to the next fault. DEPEND was used to evaluate the reliability of the Tandem Integrity S2 (see figure) under near-coincident errors.

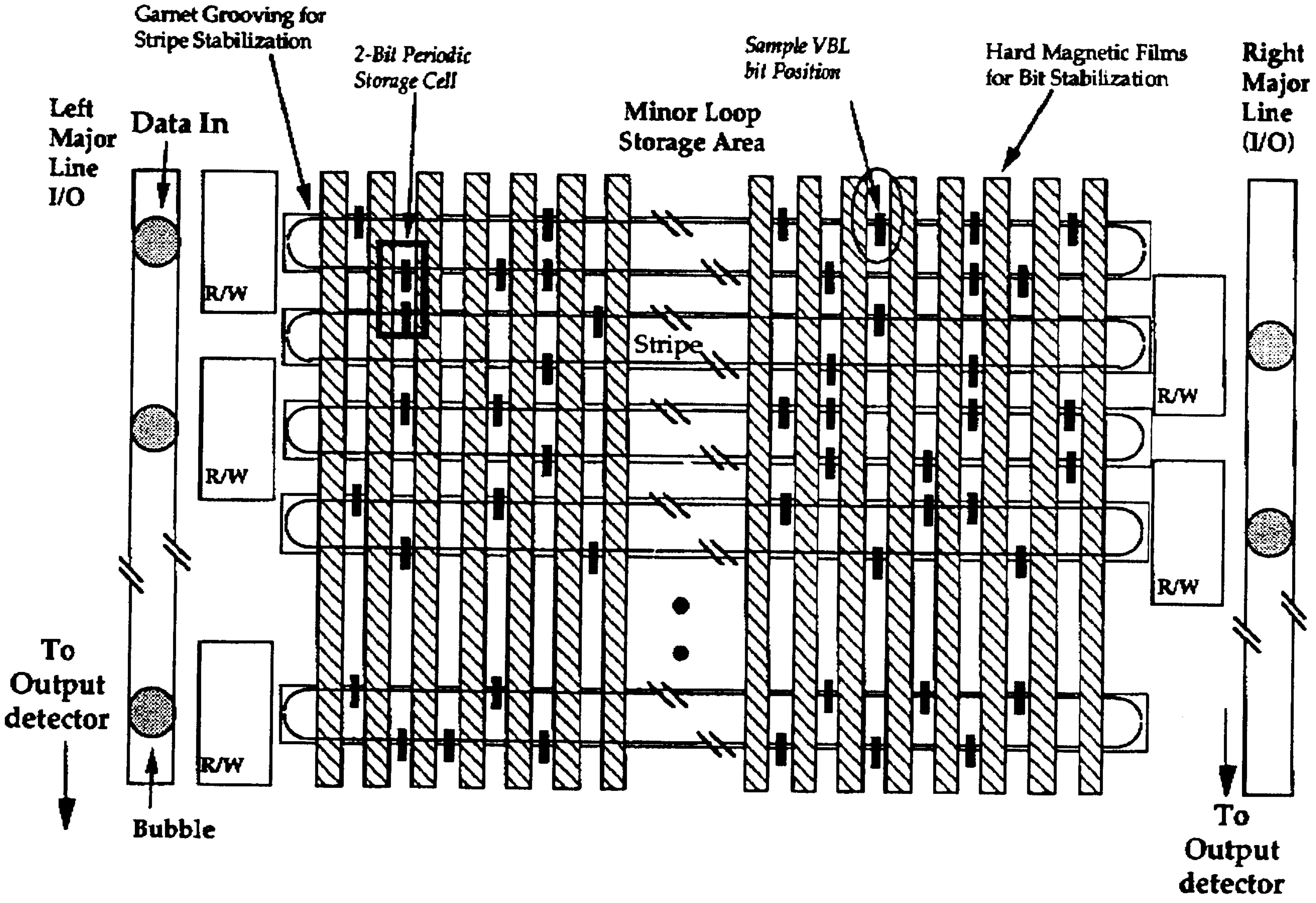
In cooperation with Ball Aerospace and the University of Arizona, DEPEND was used to study various architectures of the electronic camera and spectrometer unit of the Hubble Telescope.

Significance: This approach makes it possible to model a large system for hundreds of years. Hybrid simulation consists of modeling parts of the system in detail, extracting key distributions that capture its behavior, and then using these distributions to drive simpler continuous-time Markov chain or Monte Carlo simulation models. This marriage of detailed functional simulation with analytical or simple simulation models makes it possible to evaluate a detailed model of a system in minutes. Currently, there are no other general-purpose, simulation-based tool that provides such an extensive automated environment to make the analysis of complex fault-tolerant systems feasible.

Status/Plans: Both the government and industry have expressed a strong interest in the tool. NASA Ames is supporting the development of DEPEND for use in analyzing the Space Station testbed. NASA Langley has ported DEPEND to evaluate its use in studying candidate large-scale computing system designs for future space-based computing platforms.

Dr. Ravi K. Iyer
Center for Reliable and High Performance
Computing
University of Illinois, Urbana-Champaign, IL
(217) 333-9732

VBL Memory Layout



R. Katti
#44/vbl40gpssys.drw2
12/07/92

Micromagnetic Simulations on the Touchstone Delta for High Performance Computing

Objective: To develop a high spatial resolution and high temporal resolution micro magnetic model to simulate the statics and time-evolutionary dynamics of micro magnetic structures for Vertical Bloch line (VBL) memories. Improving the computational algorithms and code, and studying the physics, material sensitivities, and design sensitivities of the computed structures is an integral part of an effort relating to ongoing experimental work on VBL memories.

Approach: The phenomenological Landau-Lifschitz-Gilbert (LLG) equation, which models the dynamic behavior of magnetic moments subject to acting magnetic fields, is solved with respect to the applied fields, demagnetizing fields, exchange fields, and anisotropy fields and their boundary conditions. A computer coded version of this simulation runs on a CM-2 Connection Machine, so the code and the algorithms are being modified to run effectively on a Touchstone Delta.

Accomplishments: The demagnetizing field calculation was successfully implemented on the Touchstone Delta. The performance of the implementation was also benchmarked. The benchmarking indicated that significant improvements in run times can be obtained by distributing integrations within two-dimensional Fourier transforms across multiple nodes.

Significance: The ability to perform the demagnetizing field calculation indicates that the other magnetic field components can be implemented to solve the LLG equation meaningfully. However, since the demagnetizing field calculation is the most computationally intensive portion of the simulation, any performance gains made with respect to it significantly reduces total computation time

Status/Plans: The demagnetizing field calculation has been implemented on the Touchstone Delta. Improvements to the demagnetizing field calculation will be made, initially, by improving the integration in the two-dimensional Fourier calculation by distributing the calculation across multiple nodes. Subsequently, the applied, exchange, and anisotropy fields will be added, and the LLG solving algorithm will be implemented, so that test cases can be run and computational performance can be benchmarked.

Dr. Romney R. Katti
Space Microelectronic Device Technology Section
Jet Propulsion Laboratory
(818)354-3054

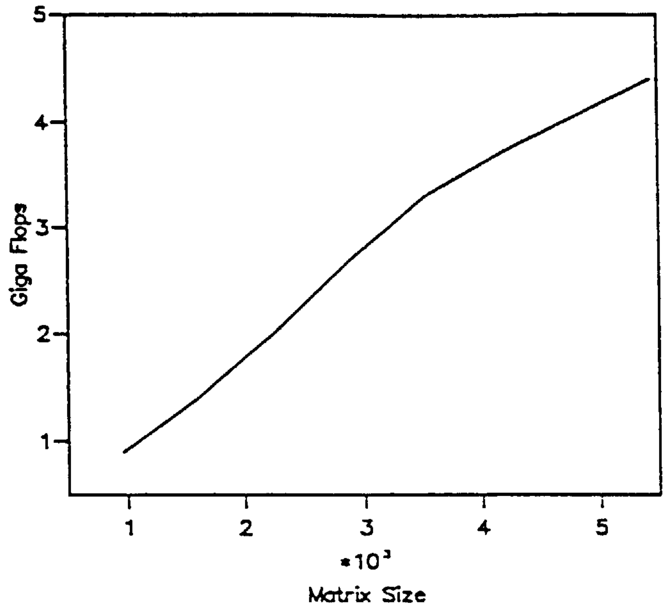


Figure 1: Performance of matrix multiplication problem as a function of matrix size for single precision data on a 128-node INTEL iPSC/860 machine.

Nodes	Matrix order	m	n	k	Total Mflops	Overlap fraction	Mflop / node	Mflop' / node
1 x 2	700	700	350	350	73	0.96	36.5	38.0
2 x 2	1000	500	500	500	157	0.94	39.3	41.8
2 x 4	1400	700	350	350	295	0.94	36.9	39.3
4 x 4	2000	500	500	500	607	0.95	37.9	39.9
4 x 8	2800	700	350	350	1132	0.94	35.4	37.7
8 x 8	4000	500	500	500	2382	0.94	37.2	39.6
8 x 16	5440	680	340	340	4356	0.92	34.0	37.0

Table 1: Dependency of peak performance of matrix multiplication on the number of nodes and matrix size.

The Design and Implementation of Basic Computational Kernels (Templates) on Parallel Machines for Computational Fluid Dynamics and Structures Applications

Objective: To design and implement basic computational kernels (templates) to facilitate porting of NASA codes on parallel machines.

Approach: A number of very powerful parallel distribution computers are commercially available. One reason that these machines are not yet widely used is that it is difficult for the ordinary scientist and engineer to use them efficiently. Thus, we feel that there is a need to design and implement basic computation kernels for parallel scalable distributed system in different application areas. One issue which is central to the design of basic computational kernels is the data distribution on a parallel machine. It is possible that a certain data distribution is very efficient for one of the isolated kernels, but could be inappropriate if incorporated into a larger application. We are investigating two approaches to address this problem. The first one is to design a computational kernel for a specific data distribution which gives optimal performance, along with a set of basic redistribution routines to convert a user data distribution to the specific data distribution. The second approach is to design a computational kernel which works well for a set of commonly occurring data distributions and, thus, requires no explicit redistribution of data.

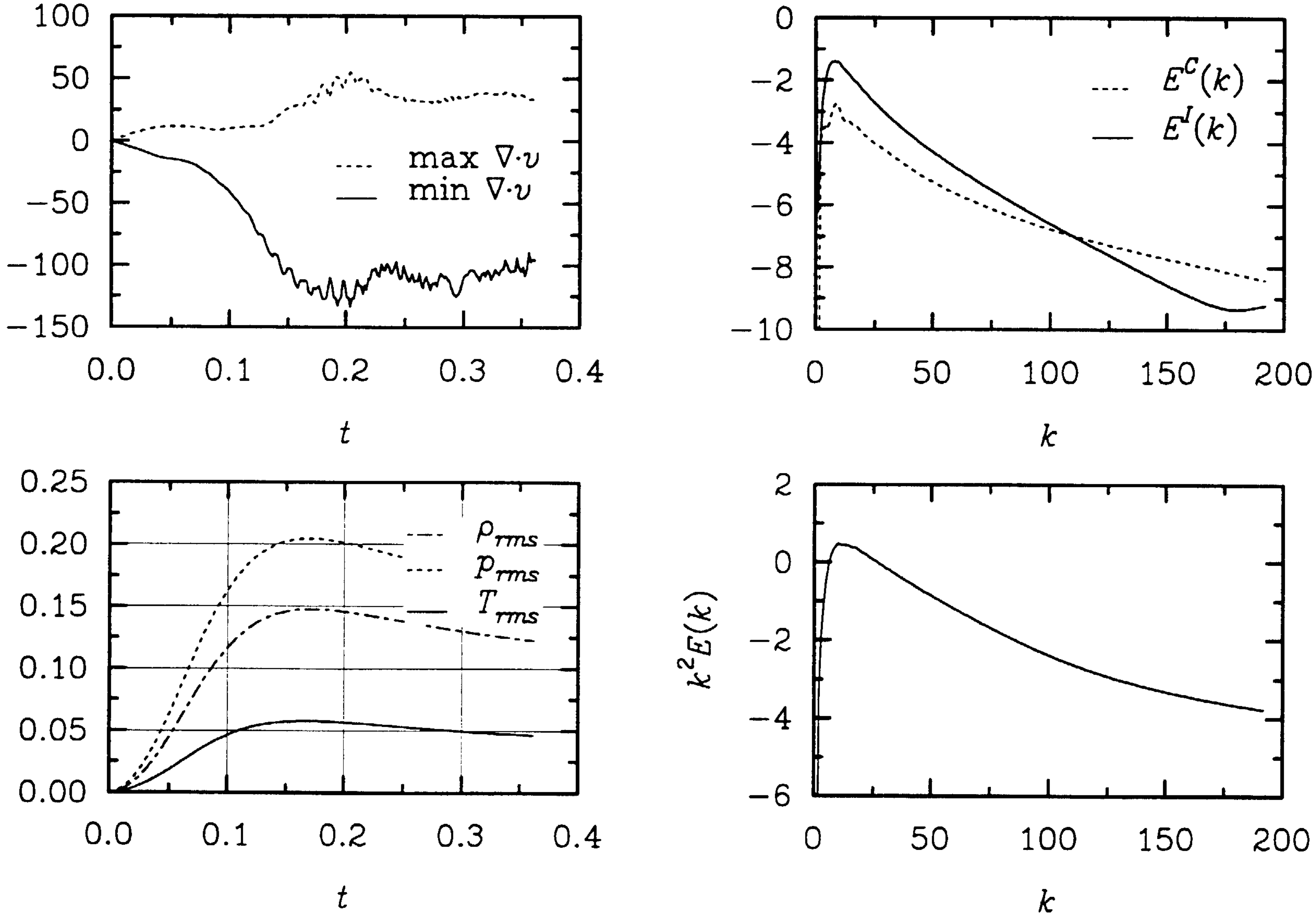
Accomplishments: We have designed and implemented a few of the basic computational kernels namely (1) matrix-matrix multiplication, (2) 1-d FT, and (3) 3-d FFT. The Matrix multiplication kernel which we designed and tested works for a set of commonly occurring data distributions. We tested the kernels on the INTEL iPSC/860. We obtained a peak performance of 4.4 Gflops on a 128-node machine which is around 34-Mflops/node for a 5440x5440 matrix (This work was done in collaboration with Fred Gustavson of IBM T.J. Watson Research Center). For this kernel, we also evaluated the dependency of peak performance on the number of modes and matrix size (see Figure 1 and Table 1). We have also designed and tested 1-d FFT and 3-d FFT kernels which have potential application in 3-d simulation of compressible turbulence. For a 3-d FFT implementation on a 32-node INTEL iPSC/860, we were able to obtain 227 Mflops, which is around 14 Mflops/node. We are in the process of designing a sparse direct solver on a distributed memory parallel machine. We have done a preliminary study to evaluate the impact of scheduling and partitioning on the performance of a parallel direct solver.

Significance: The availability of basic computational kernels will help in porting existing, or implementing new, NASA applications on parallel machines. These kernels, which will be hand turned to a particular architecture, will result in an efficient implementation of the complete application. Thus, an ordinary scientist or engineer need not be aware of various hardware/software features of a parallel machine which are critical for obtaining good performance.

Status/Plans: We are planning to design and implement other basic computational kernels, such as sparse direct and iterative solvers, tridiagonal and block tridiagonal solvers, pentagonal and block pentagonal solvers, etc., relevant to NASA applications. Initially, we will design and implement these kernels for distributed memory parallel machines, such as INTEL iPSC/860 and DELTA, and later we plan to do the same on other available parallel machines.

M. Zubair
ICASE, ASA Langley Research Center
Hampton, VA
Old Dominion University, Norfolk, VA
(804) 864-2174

384^3, 512 nodes
Touchstone Delta



Simulation of Compressible Turbulence on the Touchstone Delta

Objective: To simulate compressible, isotropic, Navier-Stokes turbulence on the Intel Touchstone Delta parallel computer with unprecedented resolution.

Approach: The structured, vectorized grid code, CDNS, was adapted to the Intel Hypercube/Touchstone Delta computers. The code has been used for production runs on the Touchstone Delta in simulations of compressible turbulence and at higher Reynolds numbers that were feasible previously. The Computational domain is periodic in all directions. Spatial derivatives are 6th order, and the time advancement is 32rd order. A host/client paradigm allows the scalar portion of the code to run a Sun workstation, and the Delta to be used as an array processor. This increases the efficiency of the code.

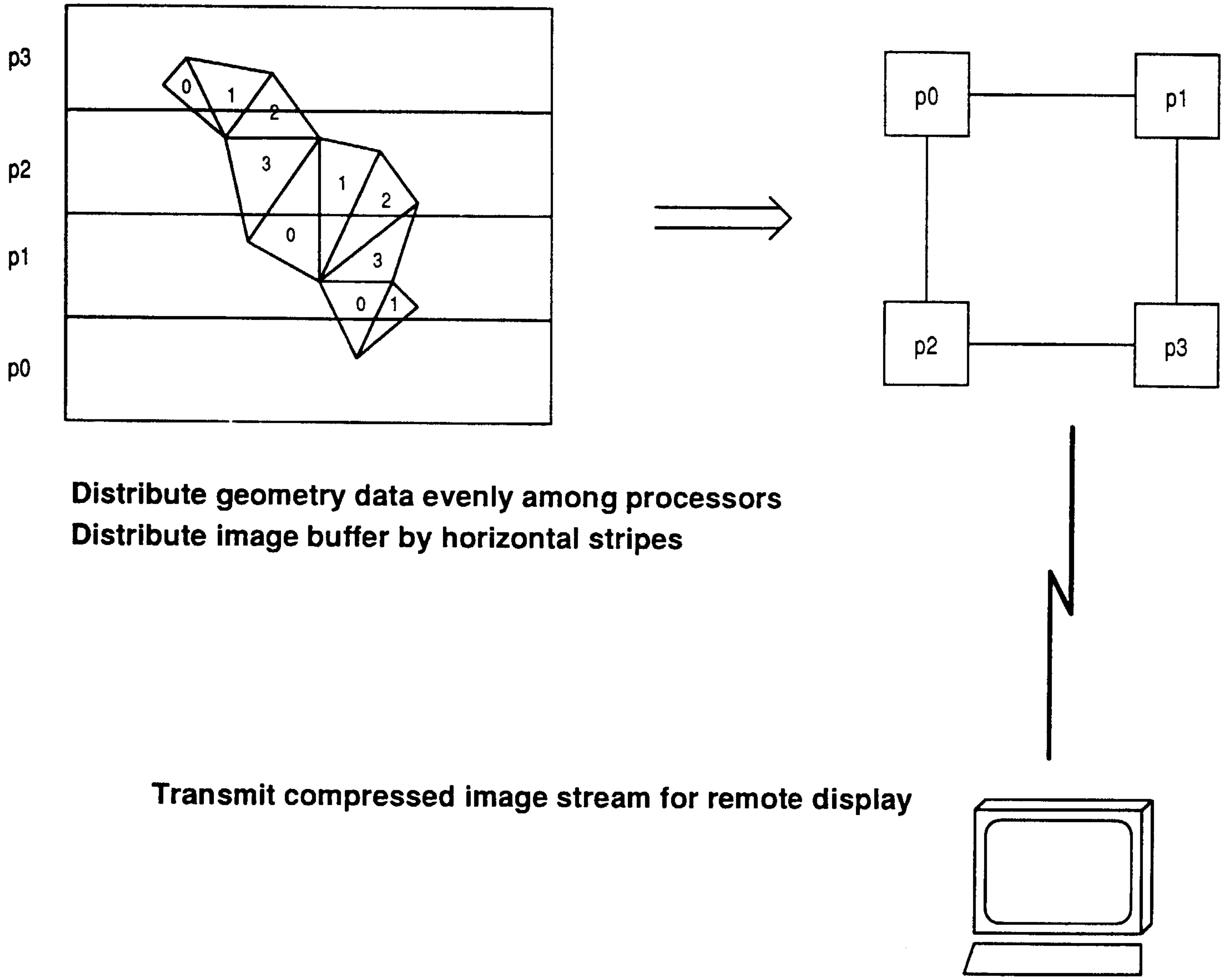
Accomplishments: The 3-D turbulence simulation code CDNS (and its variants) are heavily used at NASA Langley for basic research on the physics of compressible transition and turbulence. The spatial derivatives are based on a 6th order compact scheme to discretize the derivatives in the 3 coordinating direction. The resulting implicit periodic scalar tridiagonal system is efficiently solved by a completely balanced Gauss elimination algorithm which operates on data distributed over multiple nodes. Sustained speeds of over 2 Gflops are achieved on 384^3 grids size problems. A new storage strategy allows us to solve 450^3 problems, on which we expect a sustained rate of 3 Gflops. Turbulent simulations on 384^3 grids were conducted for isotropic turbulence to study the effect of an isotropic distribution of strong shocks on the turbulent statistics. The data is now being processed. The initial turbulent Mach number is 0.7, the spectrum peaks at $k_0 = 8$ and the initial $Re_{l1} = 49$. These conditions will lead to a strong shocklet distribution, evidenced by the time history of maximum and minimum dilatation. Note that the solenoidal and irrotational energy spectra are well resolved, decreasing by more than 6 orders of magnitude over two decades of wave numbers. The enstrophy spectrum is also well resolved.

Significance: The CDNS code and its kernel are written in standard Fortran (with the sole addition of Intel message passing calls). The implementation strategy (especially that adopted for the implicit equations) is readily extendible to such production CFD codes as the single block versions of CFL3D and ARC3D.

Status/Plans: This Delta version of CDNS will permit 3-D compressible turbulence simulations to be conducted for turbulence Reynolds numbers a factor 5 larger than what we have achieved on 128^3 simulations in the Cray 2. Large-eddy simulations of grids of 256^3 will be initiated over the next year. Performance studies on the Delta will continue.

T. Eidson and G. Erlebacher
ICASE
Langley Research Center
(804) 864-2174

Parallel Rendering on Distributed Memory Supercomputers



Distribute geometry data evenly among processors
Distribute image buffer by horizontal stripes

Transmit compressed image stream for remote display

Parallel Rendering Algorithms for Distributed Memory Supercomputers

Objective: To develop algorithms and methodologies which allow large-scale supercomputers to efficiently generate visual output from the computations which are running them.

Approach: In order to produce visual output, the result of large-scale supercomputers have traditionally been post-processed off-line on specialized graphics workshops. Techniques are being developed which will allow the graphics operation to be integrated directly with the applications software on distributed memory parallel rendering supercomputers such as the iPSC/860, Delta, Paragon and CM-5. This approach requires both efficient parallel rendering algorithms and a practical method for transferring the resulting image stream to the user's workstation.

Accomplishments: A prototype parallel renderer has been developed which distributes the large-scale graphics data structures (geometric description and image buffers) evenly among the processors, and exploits parallelism in both the transformation and rasterization phase of the rendering pipeline. The performance of the rendering algorithm has been examined in detail, both analytically and experimentally, on the iPSC/860. The rendering algorithm has characteristics which makes it especially practical for embedding within complete applications. The prototype rendering also exploits parallelism to compress the image stream both spatially and temporally, allowing it to be transmitted over conventional networks (Ethernet) and displayed on users' workstations at a few frames per second.

Significance: By rendering complex datasets in place as they are created, the need to transfer huge volumes or raw data across the network for post-processing is eliminated. Instead, we transmit a compressed image stream, which is at most a few megabytes per frame under worst-case assumptions. This approach is potentially useful for debugging, execution monitoring, and interactive steering of super computer applications in real time.

Status/Plans: Work is underway to incorporate the current rendering algorithms and image compression techniques into a parallel library which can be embedded within applications. Significant improvements in performance and functionality area anticipated relative to the prototype implementation. Algorithms issues remain relating to load balancing and scalability to large numbers of processors (hundreds or more). Work will continue on these topics.

Thomas W. Crockett
ICASE
NASA Langley Research Center
tom@icase.edu
(804) 864-2182

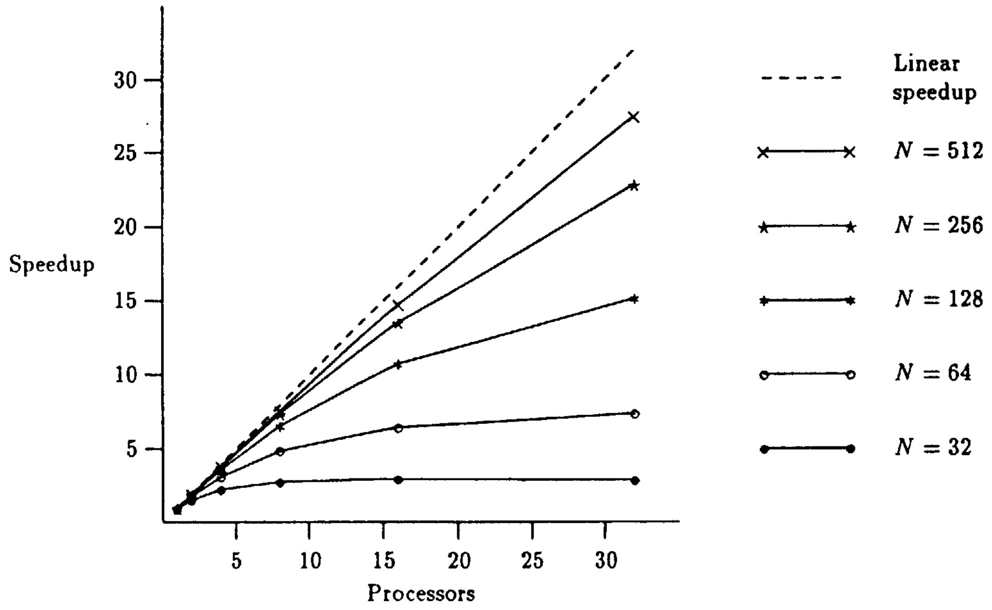


Figure 1: Speedup of Gaussian elimination program

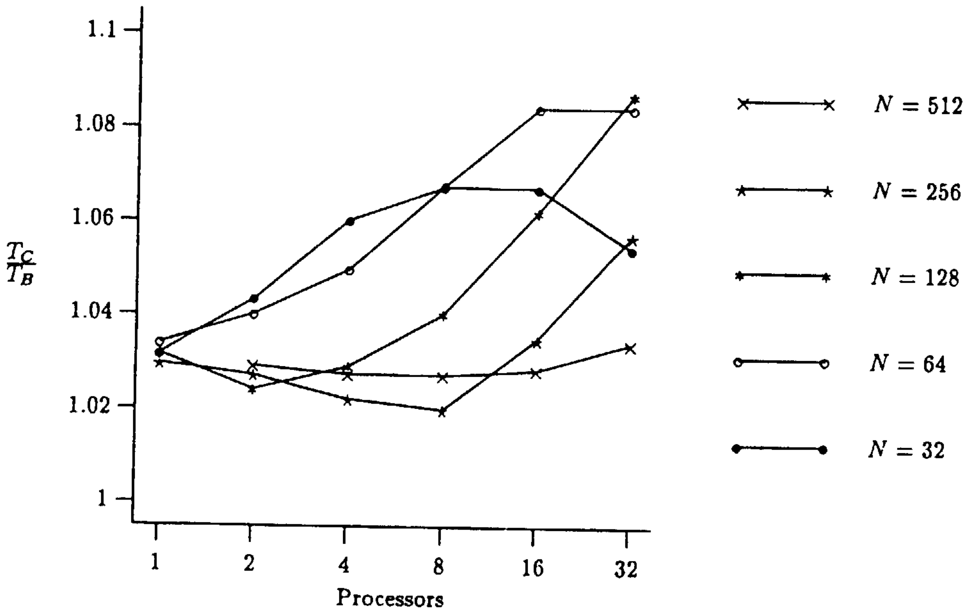


Figure 2: Ratio of times for compiler generated code (T_C) and best parallel times (T_B) for Gaussian elimination

Low Latency Software Interface for Distributed Memory Multiprocessors

Objective: To develop a software interface for use on distributed memory multiprocessors that will support fine grained, communication intensive applications typically found at NASA.

Approach: The operating system running on each node of the Intel iPSC/860 (NX) was modified to allow direct control of the hardware. The original version of NX supported a very general purpose message handler that is the source of much of the inefficiency found in data transmission. By circumventing this layer of software, it is possible to write efficient, application-specific software interfaces.

Accomplishments: In a demonstration program that is a typical kernel found in NASA fluid dynamics codes (a pipelined tridiagonal solve), the time to send a one word message was reduced by more than a factor of 100. This ability to pipeline communications at a very fine level will make it possible to take advantage of finer levels of parallelism in many codes.

Significance: A very important characteristic of all parallel machines is the time it takes to transfer data between processors. As the time decreases, the number of applications that effectively can be run on a machine increases. Hardware designers have recognized this and have reduced the startup time to send a message from approximately 100us to approximately 100ns. However, the software associated with this operation is still around 50us. By balancing the time to send a single word of data with the time to perform a floating point operation, it will be possible to implement applications that send a large number of very small messages more easily and efficiently.

Status/Plans: A set of routines will be developed to support an existing library package that is used in unstructured dynamic applications (PARTI). We also are encouraging hardware vendors, where possible, to support this work in their machines. Finally, we are interested in developing an interface that is efficient and portable across a wide range of parallel machines.

Matt Rosing
ICASE
Langley Research Center
Hampton, VA 23665
(804) 864-2174

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188
1. AGENCY USE ONLY (leave blank)	2. REPORT DATE December 1993	3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE NASA High Performance Computing and Communications Program: Annual Report, 1992		5. FUNDING NUMBERS
6. AUTHOR(S) Lee Holcomb, Paul Smith and Paul Hunter		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA-TM-4554
11. SUPPLEMENTARY NOTES		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category-82		12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

The National Aeronautics and Space Administration's HPCC program is part of a new Presidential initiative aimed at producing a 1000-fold increase in supercomputing speed and a 100-fold improvement in available communications capability by 1997. As more advanced technologies are developed under the HPCC program, they will be used to solve NASA's 'Grand Challenge' problems, which include improving the design and simulation of advanced aerospace vehicles, allowing people at remote locations to communicate more effectively and share information, increasing scientist's abilities to model the Earth's climate and forecast global environmental trends, and improving the development of advanced spacecraft. NASA's HPCC program is organized into three projects which are unique to the agency's mission: the Computational Aerosciences (CAS) project, the Earth and Space Sciences (ESS) project, and the Remote Exploration and Experimentation (REE) project. An additional project, the Basic Research and Human Resources (BRHR) project, exists to promote long term research in computer science and engineering and to increase the pool of trained personnel in a variety of scientific disciplines. This document presents an overview of the objectives and organization of these projects as well as summaries of individual research and development programs within each project.

14. SUBJECT TERMS NASA programs, research and development, parallel programming, software engineering, computerized simulation, computational fluid dynamics, aerospace engineering, space exploration, earth sciences			15. NUMBER OF PAGES 139 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclass	18. SECURITY CLASSIFICATION OF THIS PAGE Unclass	19.SECURITY CLASSIFICATION OF ABSTRACT Unclass	20. LIMITATION OF ABSTRACT Unlimited

Available from NASA Center for AeroSpace Information
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
(301) 621-0390