

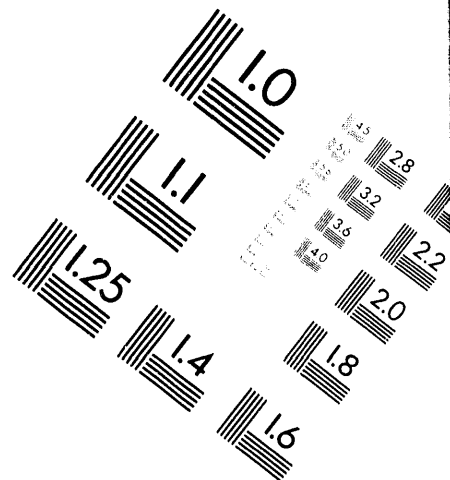
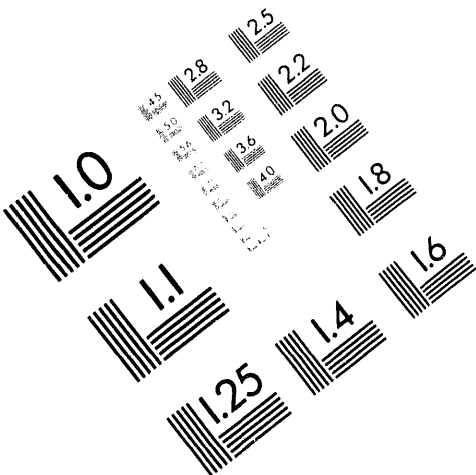


AIM

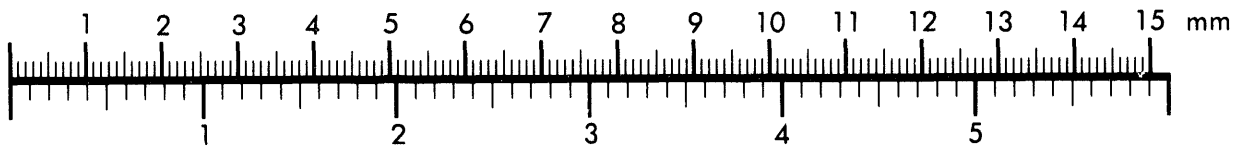
Association for Information and Image Management

1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

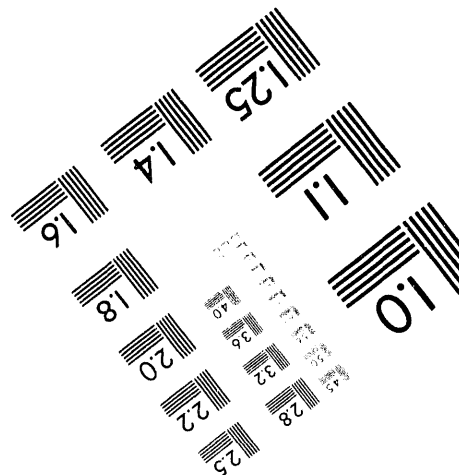
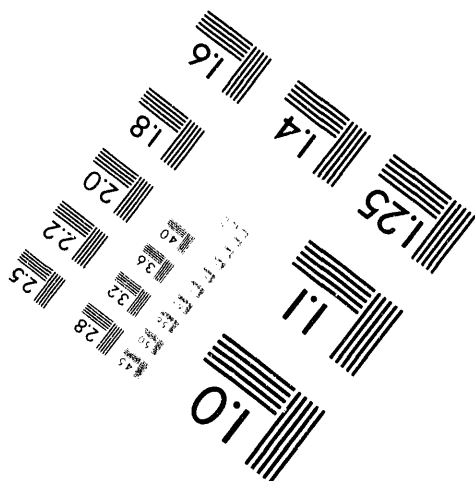
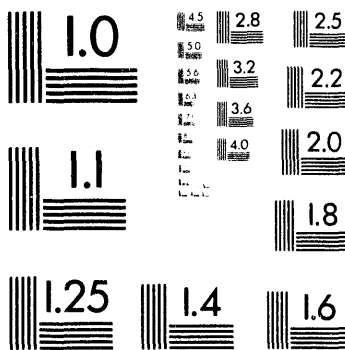
301/587-8202



Centimeter



Inches

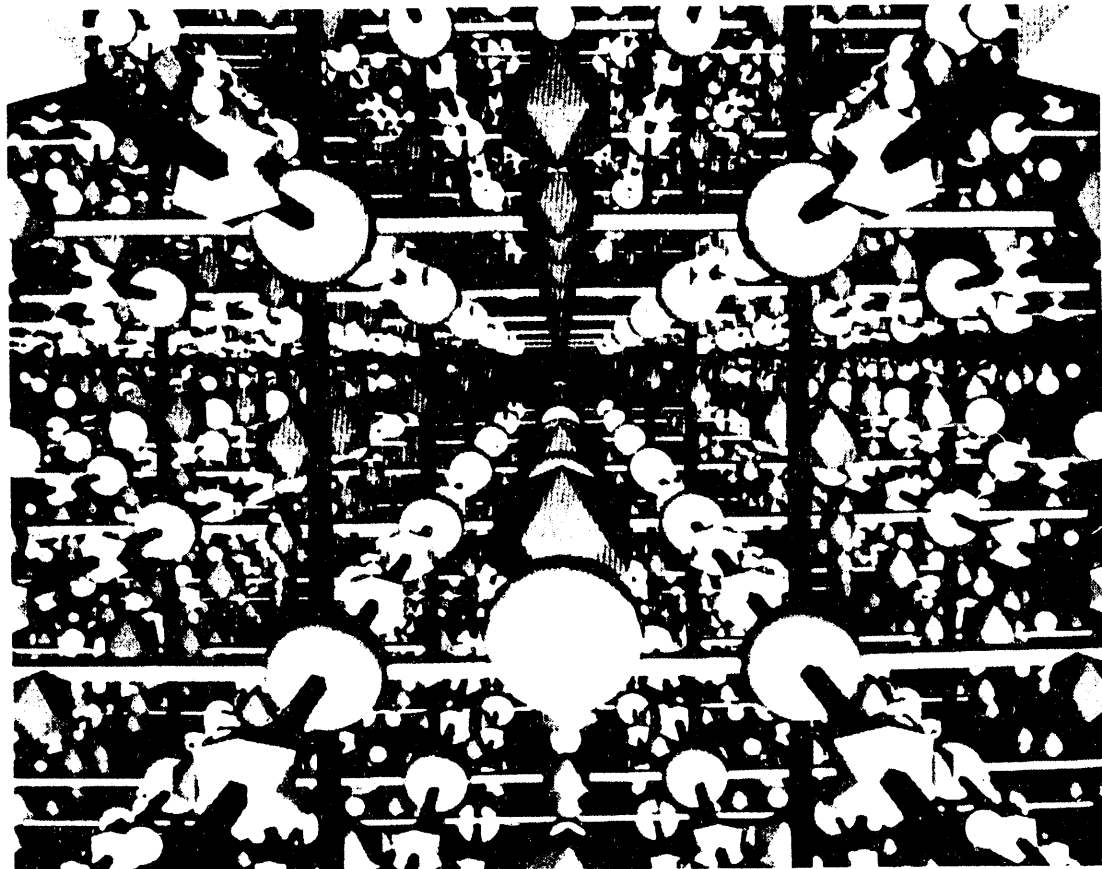


MANUFACTURED TO AIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 2

CONF-9305349--

Workshop and Conference on Grand Challenges Applications and Software Technology



MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED ^{EB}

May 4-7, 1993

Pittsburgh, Pennsylvania

Sponsoring Agencies

Advanced Research Projects Agency (ARPA)
Department of Energy (DOE)
Environmental Protection Agency (EPA)
National Aeronautics and Space Administration (NASA)
National Science Foundation (NSF)
National Institutes of Health (NIH)
National Institute of Standards and Technology (NIST)
National Oceanic and Atmospheric Administration (NOAA)
National Security Agency (NSA)

Committees

Organizing Committee

Paul H. Smith, Chair, NASA
Stephen M. Griffin, Executive Secretary, NSF
Norman Glick, NSA
Fred Johnson, NIST
Tom Kitchens, DOE
Jacob Maizel, NIH
Tom Pyke, NOAA
Joan Novak, EPA
Gil Weigand, ARPA
Robert Voigt, NSF

Program Committee

Rick Stevens, Program Chair, Argonne National Laboratory
Fran Berman, University of California - San Diego
Al Brenner, Supercomputing Research Center
Phillip Colella, University of California - Berkeley
Kevin Drogenmeir, University of Oklahoma
Ian Foster, Argonne National Laboratory
Geoffrey Fox, National Parallel Architecture Center, Syracuse
Mike Heath, University of Illinois and NCSA
Ken Kennedy, Rice University
Tom Kitchens, DOE
Robert Martino, NIH
Paul Messina, California Institute of Technology
Reagan Moore, San Diego Supercomputer Center
Dan Reed, University of Illinois
Ralph Roskies, Pittsburgh Supercomputing Center
Manny Salas, NASA Langley Research Center
Joel Saltz, University of Maryland
Ahmed Sameh, Center for Supercomputing Research and Development
Fran Sullivan, Supercomputing Research Center
Andy White, Los Alamos National Laboratory
Paul Woodward, University of Minnesota

GCW-0593

**Workshop and Conference on
Grand Challenges Applications and
Software Technology**

May 4-7, 1993

Pittsburgh, Pennsylvania

Foreword

The need for multidisciplinary and interdisciplinary collaboration to solve large-scale computational problems has been recognized since the beginning of the Federal HPCC Program. With this realization came the awareness that necessary levels of interaction have not occurred in the normal course of events in the past, and that it would be beneficial for the participating HPCC agencies to foster processes and to provide occasions specifically for the purpose of stimulating cross-disciplinary communication and interaction.

The motivation for convening the Grand Challenge Applications and Software Technology Workshop and Conference came out of the continuing belief that collaboration, shared goals, and knowledge exchange between applications scientists and software technologists are critical to meet the overall goals of the HPCC Program. The gathering was designed to bring the HPCC agencies' grand challenge research teams and leading software technologists from the academic and private sectors together with a few of the key industrial leaders, and ask them to focus their expertise on identifying high-performance computing needs and, more importantly, to anticipate those of the future.

Speaking for the Organizing Committee, we believe the workshop and conference was a valuable occasion for all concerned and hope to see a continuation of such events in the future.

Paul H. Smith, NASA
Organizing Committee Chair

Stephen M. Griffin, NSF
Organizing Committee Executive Secretary

Contents

1 Executive Summary	1
Program	4
Application Sessions	6
Working Groups	10
2 Background and Objectives	13
3 Grand Challenge Applications Panel Sessions	15
Panel A - Environment and Earth Sciences	15
Panel B - Computational Physics	17
Panel C - Comp. Biology, Chemistry, Materials Sciences	18
Panel D - Computational Fluid and Plasma Dynamics	19
Panel E - Applications of Artificial Intelligence	21
4 Grand Challenges and Industrial Applications	23
Chemical Applications: DuPont Corporation	23
Automotive and Manufacturing Applications: Ford	24
Financial Applications: Prudential Securities	24
Aeronautics Applications: Rockwell International	24
Energy Resource Extraction Applications: Western Atlas	24
Textile Applications: DuPont Fibers	25
Health Care Applications: Phillips Medical Systems	25
5 Working Groups	27
Group A - I/O, Data Systems, and File Systems	27
Status	27
Targets of Opportunity	28
Relationships	29
Findings	30
Group B - Parallel Programming Paradigms	31
Status	31
Targets of Opportunity	33
Relationships	34
Impacts	35
Findings	35
Group C - Performance Characterization and Evaluation	36

Status	36
Targets of Opportunity	37
Relationships	39
Impacts	40
Findings	41
Group D – Program Development Tools	41
Status	42
Targets of Opportunity	42
Relationships	43
Impacts	44
Findings	44
Group E – Building Multidisciplinary Applications	46
Status	46
Targets of Opportunity	47
Relationships	49
Impacts	50
Findings	50
Group F – Algorithms and Libraries I	51
Status	51
Targets of Opportunity	53
Relationships	53
Impacts	54
Findings	54
Group G – Algorithms and Libraries II	55
Status	55
Targets of Opportunity	55
Relationships	56
Impacts	57
Findings	58
Group H – Graphics and Visualization	59
Status	59
Targets of Opportunity	60
Relationships	61
Impacts	61
Findings	62
Group I – National HPCC Infrastructure	63
Status	63
Targets of Opportunity	64
Findings	67
6 Conclusions	69
7 Findings	71
A Grand Challenge Requirements Document	75
B Summary of Survey Responses	81
C Abstracts of Talks	85
D Workshop Invitees	109



SECTION 1

Executive Summary

On May 4-7, 1993, nine federal agencies sponsored a four-day meeting on Grand Challenge applications and software technology.

Over two hundred and fifty researchers from universities and national laboratories attended the meeting. The objective was to bring High-Performance Computing and Communications (HPCC) Grand Challenge applications research groups supported under the federal HPCC program together with HPCC software technologists to

- discuss multidisciplinary computational science research issues and approaches,
- identify major technology challenges facing users and providers, and
- refine software technology requirements for Grand Challenge applications research.

The first day and a half focused on applications. Presentations were given by speakers from universities, national laboratories, and government agencies actively involved in Grand Challenge research. Five areas of research were covered: environmental and earth sciences; computational physics; computational biology, chemistry, and materials sciences; computational fluid and plasma dynamics; and applications of artificial intelligence.

The next day and a half was spent in working groups in which the applications researchers were joined by software technologists. Nine breakout sessions took place: (1) I/O, Data, and File Systems; (2) Parallel Programming Paradigms; (3) Performance Characterization and Evaluation of Massively Parallel Processing (MPP) Applications; (4) Program Development Tools; (5) Building Multidisciplinary Applications; (6) Algorithm and Libraries I; (7) Algorithms and Libraries II; (8) Graphics and Visualization; and (9) National HPCC Infrastructure.

The strategy is to bring together applications researchers and software technologies to discuss high-performance computing needs.

High-performance computers are helping to solve previously intractable problems.

Programming tools are essential for dramatically increasing scientific production on parallel computers.

On the final day, a representative from each of the applications areas presented a ten-minute summary of the common themes, approaches, and obstacles in the specific Grand Challenge discipline. These were followed by thirty-minute presentations reporting the findings of the nine breakout sessions and discussing significant new opportunities for high-performance computing and communications research. In addition, representatives from several U.S. industries reported on Grand Challenge efforts under way in such areas as global finance and chemical production.

The researchers in the Grand Challenge community and in the Software Development community agreed that the federal HPCC program is already having an undeniable impact. Indeed, one might almost speak of a *high-performance revolution*, with widespread interest in and use of scalable parallel systems by industry and academia; increased focus on parallelism by workstation vendors; growing emphasis on the need for standards; and almost universal concern for developing better products (machines, languages) to make parallel computing effective.

And while the researchers agreed that it is probably too early to expect HPCC program activities to have had a major impact on science and industry, *exciting progress is being made*. In particular, high-performance computing is enabling researchers to attack previously intractable problems, such as the recent 512^3 simulation of compressible turbulence.

This increasing capability to handle complex problems has already attracted the attention of industry. *Industry involvement* was noted in numerous Grand Challenge projects, including aerospace, environmental science, biophysics, computational fluid and plasma dynamics, computational chemistry, reservoir modeling, and global computational finance.

Nevertheless, several *outstanding issues* were identified.

- **Software tools.** A clear need exists for better parallel debugging tools, tools for multidisciplinary applications, performance-monitoring tools, and language support to allow users to write programs at a higher level than currently possible. The cause for the poor software support is the fact that the Grand Challenge grants currently focus on the output of the applications rather than on the software to achieve that output. More effective mechanisms are needed for exchanging information on tool availability and accessibility.
- **Database management.** The Grand Challenge studies are generating enormous databases (up to 8 terabytes, with predictions in the petabyte range). Making this data useful will require not only high-density, high-throughput storage devices but also improved archive and directory services. Even further, specialized visualization systems will be essential for examining the vast datasets expected from Grand Challenge research.
- **Data rates.** I/O is simply not scaling fast enough to keep up with CPU speed and is threatening to become a major bottleneck for teraflops applications. Further research is needed to understand the I/O characteristics of massively parallel computers. Parallel I/O benchmarks are also needed that accurately reflect applications to requirements.
- **Standards.** Current efforts such as the evolving Message Passing Interface standard appear promising, but further work is needed, for example, to promote vendor standards for archiving large amounts of data.

*Researchers
have identified
priorities for
accomplishing
HPCC goals.*

To address these problems, the workshop participants formulated several specific suggestions:

- Support a "Grand Challenge in Software Tools" and establish a Science and Technology Center for HPCC Software Tools, to encourage collaborative development of tools for Grand Challenge applications.
- Establish or encourage "software capitalization" programs to foster the development of research ideas and prototypes into production tools and to ensure the maintenance of the software in the library.
- Promote the development of research tools into usable software aids (through an existing organization such as OSF or a new organization or industrial consortium).
- Develop a parallel software tools electronic journal/newsletter (complete with an editorial board) to publicize tool availability and tool needs.
- Require computer vendors (through explicitly worded RFPs) to provide parallel debuggers, profilers, and compilers for MPP machines.
- Have HPCC projects explicitly promote the exchange of students and staff between application and tool developers, to encourage computer scientists and computational scientists to interact more closely on Grand Challenge projects.
- Increase the current federal involvement, specifically by providing (1) stable base-level support for ongoing projects and (2) stable long-term support for new initiatives such as multidisciplinary programs and scalable I/O.

Program

Daily Schedule

Tuesday, May 4

8:00 - 9:00 Welcome
9:00 - 10:15 Application Session
 Panel A, Environmental and Earth Sciences (3 presentations)
10:15 - 10:45 Break
10:45 - 12:25 Application Session
 Panel A continued (4 presentations)
12:25 - 1:30 Lunch
1:30 - 3:35 Application Session
 Panel A continued (1 presentation)
 Panel B, Computational Physics (4 presentations)
3:35 - 4:00 Break
4:00 - 6:05 Application Session
 Panel B continued (3 presentations)
 Panel C, Computational Biology, Chemistry and Material Sciences
(3 presentations)
7:00 - 8:00 Reception
8:00 Workshop Dinner

Wednesday, May 5

8:00 - 10:05 Application Session
 Panel C continued (5 presentations)
10:05 - 10:30 Break
10:30 - 12:35 Application Session Panel C continued (2 presentations)
 Panel D, Computational Fluid and Plasma Dynamics
(3 presentations)
12:35 - 1:30 Lunch
1:30 - 3:35 Application Session
 Panel D continued (3 presentations)
 Panel E, Applications of Artificial Intelligence
(2 presentations)
3:35 - 4:00 Working Groups (Instructions)
4:00 - 5:30 Working Group Breakout Session I
7:00 - 7:30 Buffet Dinner
7:30 - 9:30 Working Group Breakout Session II

Thursday, May 6

7:30 - 8:30 Breakfast Meeting
 (Chairs and Co-chairs of Breakout Sessions)
8:30 - 12:00 Working Group Breakout Session III
12:00 - 1:00 Lunch
1:00 - 3:00 Working Group Breakout Session IV (Establish conclusions)
3:00 - 5:30 Working Group Breakout Session V (Writing)

Executive Summary

Applications (Birds-of-a-Feather)

7:00 Conference Reception

Friday, May 7

8:00 - 8:10 Welcome by Paul H. Smith,

NASA, Chairman of the Organizing Committee

8:10 - 9:10 Summary Reports from Application Panels

Moderator: David Nelson, DOE

Environment and Earth Sciences: Robin Dennis, EPA

Computational Physics: Ken Kliewer, Oak Ridge

Computational Biology, Chemistry and Material Sciences:

L. Ridgeway Scott, University of Houston

Computational Fluid Dynamics: Tom Zang, NASA Langley

Applications of Artificial Intelligence: Robert Berwick, MIT

9:10 - 10:00 Reports from Working Groups

Moderators: Lee Holcomb, NASA; Rick Stevens, Argonne

National HPCC Infrastructure: Maxine Brown, UIC

I/O, Data and File Systems: Carlos Roberto Mechoso, UCLA

10:00 - 10:25 Break

10:25 - 12:30 Reports from Working Groups Continued

Parallel Programming Paradigms: Kim Mills, Syracuse U.

Performance Characterization and Evaluation of MPP

Applications:

Robert Harrison, Pacific Northwest Laboratory

Program Development Tools: L. Ridgeway Scott, U. of Houston

Building Multidisciplinary Applications: Tom Zang, NASA

Langley Research Center

Algorithms and Libraries I: Phil Colella, UC-Berkeley

12:30 - 1:20 Lunch

1:20 - 2:10 Reports from Working Groups Continued

Algorithms and Libraries II: Bob Sugar, University of California
at Santa Barbara

Graphics and Visualization: Paul Woodward, University of Minnesota

2:15 - 3:15 Conference Address: John H. Gibbons, Science and Technology

3:15 - 3:30 Break

3:30 - 5:00 Industrial Applications Panel

Moderator: Donald Lindberg, HPCC National Coordination Office

Aeronautics: Jan Tulinius, Rockwell International Corporation

Chemical Products: David Dixon, Dupont

Automotive/Manufacturing: Howard Crabb, Ford Motor Co.

Energy Resource Extraction: James S. Nolen, Western Atlas Software

Health Care: Hartwig Blume, Phillips Medical Systems

Financial: David Audley, Prudential Securities

Textiles: James Plouffe, Dupont

5:00 Adjourn

Application Sessions

Panel A – Environmental and Earth Sciences

(Chair: Joan Novak, EPA)

Armistead G. Russell
A Distributed Computational System for Large-Scale Environmental Modeling
Carnegie Mellon University

Kenneth Galluppi and Daewon W. Byun
Comprehensive Eulerian Air Quality Modeling of Multipollutant
Species in the Atmosphere
North Carolina Supercomputer Center and EPA

Thomas O. Barnwell, Jr., and Robin L. Dennis
Linked Water and Air Quality Management
EPA

Alexander MacDonald
Future of High-Resolution Weather Prediction
NOAA/Forecast Systems Laboratory

Mary Wheeler
Reservoir Modeling for Porous Media: Partnership in Computational Science
Rice University

Kenneth Eggert and Shiyi Chen
Reservoir Modeling for Porous Media: Lattice Gas Computations
Los Alamos National Laboratory

Dave Bader
Ten- and Hundred-Year Climate Prediction
Pacific Northwest Laboratory

C. Roberto Mechoso
Development of an Earth System Model:
Atmosphere/Ocean Dynamics and Tracers Chemistry
University of California - Los Angeles

Dan Kowalski
Development of Algorithms for Climate Models Scalable to Teraflops Performance
NASA/Goddard Space Flight Center

Richard B. Rood (Presented by Peter Lyster)
High-Performance Computing and Four-Dimensional Data Assimilation:
Impact on Future and Current Problems NASA/Goddard Space Flight Center

Panel B: Computational Physics

(Chair: Tom Kitchens, DOE)

John Gardner

**Understanding Solar Activity and Heliospheric Dynamics:
Naval Research Laboratory**

George Lake (Presenter: Calvin Lin)

**Large-Scale Structure and Galaxy Formation
University of Washington**

Robert Rosner

**Convective Turbulence and Mixing in Astrophysics
The University of Chicago**

Wojciech Zurek and M. Warren

**Scalable Hierarchical Particle Algorithms for Galaxy Formation
and Accretion Astrophysics
Los Alamos National Laboratory**

Bob Sugar

**Lattice Gauge Theory
University of California - Santa Barbara**

C. Grebogi, E. Ott, and J. A. Yorke

**Controlling Chaos
University of Maryland**

Richard Crutcher and Michael Heath

**Radio Synthesis Imaging: An HPCC Application
National Center for Supercomputing Applications**

Panel C: Computational Biology, Chemistry and Material Sciences

(Chair: Bob Martino, NIH)

William Goddard

High-Capacity Atomic-Level Simulations for Design of Materials
California Institute of Technology

G. M. Stocks, B. N. Harmon, and J. W. Davenport

Computational Quantum Materials:

First Principles Simulation of Materials Properties

Oak Ridge National Laboratory, Ames Laboratory, Brookhaven National Laboratory

W. J. Camp, J. S. Nelson, S. J. Plimpton, and M. P. Sears

Computer-Aided Design of Biological and Biomimetic Materials:

Computational Design for Catalysis

Sandia National Laboratories

Jerry E. Solomon

High-Performance Computational Structural Biology

California Institute of Technology

T. H. Dunning, Jr., and A. Wagner

High-Performance Computational Chemistry:

Scalable Algorithms for Grand Challenge Applications

Pacific Northwest Laboratory and Argonne National Laboratory

Robert L. Martino

High-Performance Computing in Structural Biology and Medical Imaging

National Institutes of Health

D. Lansing Taylor and Scott E. Fahlman

High-Performance Imaging in Biological Research

Carnegie Mellon University

T. Barnes, V. Overacker, M. R. Strayer, and S. Umar

Computational Quantum Materials: Quantum Structure of Matter

Oak Ridge National Laboratory and Vanderbilt University

Mike Colvin

Computer-Aided Design of Biological and Biomimetic Materials:

Pharmaceuticals and Agrochemicals

Sandia National Laboratories

R. Glowinski, J. A. McCammon, B. M. Pettitt, and L. R. Scott

Texas Center for Advanced Molecular Computation

University of Houston

Panel D: Computational Fluid and Plasma Dynamics

(Chair: Lee Holcomb, NASA)

Carlos Felippa and Juri Toomre
High-Performance Computational Methods for Coupled Field Problems
and GAFD Turbulence
University of Colorado at Boulder

Manny D. Salas
Multidiscipline Simulation of High-Speed Civil Transport
NASA/Langley Research Center

Kalpana Chawla
Multidiscipline Simulation of High-Performance Aircraft
NASA/Ames Research Center

Russell Claus
Multidiscipline Simulation of Propulsion Systems
NASA/Lewis Research Center

Phil Colella
Computational Fluid and Combustion Dynamics
University of California - Berkeley

Dan Barnes
Numerical Tokamak Project
Los Alamos National Laboratory

Panel E – Applications of Artificial Intelligence

(Chair: Y. T. Chen, NSF)

Robert C. Berwick and Tomaso Poggio
High-Performance Computing for Learning:
Super-Human Supercomputing vs. Super-Computing for Human Computations
Massachusetts Institute of Technology

Richard Muntz (Presenter Carlo Zaniolo)
Data Analysis and Knowledge Discovery in Geophysical Databases
University of California - Los Angeles

Working Groups

Group A – I/O, Data, and File Systems

Technical Chair: Reagan Moore, San Diego Supercomputer Center

Applications Co-Chair: Carlos Roberto Mechoso, University of California, LA

This session covered issues relating to mass storage, high-speed file systems, solid state disks, parallel file systems for MPP machines, parallel I/O user libraries, mass storage management, scientific databases, and other items relating to I/O.

Group B – Parallel Programming Paradigms

Technical Chair: Joel Saltz, University of Maryland

Applications Co-Chair: Geoffrey Fox, National Parallel Architecture Center

Choosing the right programming paradigm is important. This session discussed the limitations and advantages of data parallel programming (e.g., HPP) and other programming models, such as message passing, Fortran D, and Fortran 90. Discussions focused on choosing the best programming system for the type of problem.

Group C - Performance Characterization and Evaluation of MPP Applications

Technical Chair: Dan Reed, University of Illinois

Applications Co-Chair: Robert Harrison, Pacific Northwest Laboratory

Unlike serial computers, understanding performance of MPP applications is a sometimes complex vital process involving scalability, balance, and the proper use of performance analysis tools. This session focused on the needs of applications programmers for tools to assist in understanding the performance of MPP applications.

Group D – Program Development Tools

Technical Chair: Fran Berman, University of California, San Diego

Applications Co-Chair: L. Ridgway Scott, University of Houston

A wide variety of software tools have been developed to aid in the construction of parallel programs. This session covered compilers, parallel programming libraries, and support for heterogeneous computing and distributed computing. It also covered profilers, tracers, and logical debuggers.

Group E – Building Multidisciplinary Applications

Technical Chair: Ian Foster, Argonne National Laboratory

Applications Co-Chair: Tom Zang, NASA/Langley Research Center

Many modern simulation projects could greatly benefit from the combination of simulation models that exist in many diverse computer codes. The goal of this session was to explore the needs of applications developers for tools that will aid in the construction of these multidisciplinary applications.

Group F – Algorithms and Libraries I

Technical Chair: Ahmed Sameh, Center for Supercomputing Research and Development

Applications Co-Chair: Phillip Colella, University of California, Berkeley

The availability of quality numerical algorithms and libraries often limit the pace of applications development. This session covered the state of the art in algorithms and libraries for numerical PDEs, linear algebra, and finite element methods.

Group G – Algorithms and Libraries II

Technical Chair: Fran Sullivan, Supercomputing Research Center

Applications Co-Chair: Bob Sugar, University of California, Santa Barbara

Modern simulations often require new algorithms that have not been extensively used in scientific computing. This session covered areas like random number generators (including parallel random numbers), multipole methods, Monte Carlo methods, particle-in-cell algorithms, and geometric and symbolic algorithms.

Group H – Graphics and Visualization

Technical Chair: Paul Woodward, University of Minnesota

Applications Co-Chair: Tom DeFanti, University of Illinois at Chicago

Graphics and visualization are critical to modern scientific applications. This session focused on the graphics needs for Grand Challenge-type applications, including the problems of data movement, parallel rendering, portable tools like AVS, virtual reality, and other frontier issues in graphics and visualization.

Group I – National High-Performance Computing and Communications Infrastructure

Technical Chair: Charlie Catlett, University of Illinois at Urbana-Champaign

Applications Co-Chair: Greg McRae, Massachusetts Institute of Technology

Until recently the development of the national HPCC infrastructure has been developed largely by individual institutions and funding agencies. Since the creation of the HPCC initiative more coordination is possible. This session discussed the infrastructure needs and requirements of the applications teams. These include high-performance networking, integration of workstations and MPP, methods of access and allocations, support, problem reporting, and the best way to make the national computing infrastructure responsive to the needs of applications developers and users.



SECTION 2

Background and Objectives

Sharing related experiences helps identify successes and future needs.

In the past two years, the federal government has funded a number of research teams to attack Grand Challenge problems in science and engineering. While these problems include seemingly unrelated areas as materials science, biology, chemistry, earth science, physics, and artificial intelligence, many of the obstacles faced by the researchers are similar. A principal objective of the 1993 Workshop on Grand Challenges and Software Technology was to identify these common problems and to share related experiences.

Another objective of the workshop was to encourage increased communication between computational scientists and computer scientists. Often, software technologists have developed useful tools that the applications teams are not aware of; and often, the applications people need special tools that are still under development. By bringing together these two groups, the workshop sought to identify what enabling software is available and what tools are needed for successfully solving the Grand Challenge problems.

This was the second Grand Challenge workshop sponsored by the federal government. The specific agencies involved were the following:

- Advanced Research Projects Agency (ARPA)
- Department of Energy (DOE)
- Environmental Protection Agency (EPA)
- National Aeronautics and Space Administration (NASA)
- National Science Foundation (NSF)
- National Institutes of Health (NIH)

-
-
- National Institute of Standards and Technology (NIST)
 - National Oceanic and Atmospheric Administration (NOAA)
 - National Security Agency (NSA)

In the remainder of this report, we summarize the activities of the 1993 meeting on Grand Challenge Applications and Software Technology. These activities included panel discussions, working group sessions, and industrial presentations. We conclude with an evaluation of the four-day meeting and suggestions for future meetings.



SECTION 3

Grand Challenge Applications Panel Sessions

On May 4-5, 1993, representatives from universities, national laboratories, and government agencies gave presentations about their current HPCC applications. Five panel sessions were held, focusing on Environmental and Earth Sciences; Computational Physics; Computational Biology, Chemistry, and Material Sciences; Computational Fluid and Plasma Dynamics; and Applications of Artificial Intelligence.

Panel A - Environment and Earth Sciences

Panel A focused on the processes occurring in our atmosphere, oceans, and earth. Using advanced computational techniques and software tools, the researchers are developing models that can be used for environmental management and prediction. Nine topics were addressed:

*Understanding
our environment
requires
understanding
the processes
occurring in
the atmosphere,
oceans, and earth.*

- a distributed computational system for large-scale environmental modeling,
- air quality management,
- linked water and air quality management,
- high-resolution operational weather prediction,
- reservoir modeling for porous media,
- Grand Challenge in climate change,
- development of an earth system model,
- development of algorithms for climate models scalable to teraflops performance, and

- high-performance computing and four-dimensional data assimilation.

Despite the apparent diversity of topics, one can distinguish four main categories.

1. **Environmental management.** Whether the researchers are dealing with air quality, water quality, or a combination of the two, the issue is the same: How can we effectively use the models (after we have developed good models) to address practical problems such as ozone levels, acid deposition, or air pollution? The challenge is twofold. First, we must make high-performance computing understandable; this may require development of expert systems and effective interfaces to existing visualization packages for the average user, as well as design of a meta-computer for the decision maker. Second, we must learn how to use different models for a particular environmental situation; currently, major mismatches exist between the spatial and temporal systems of models.

2. **Weather forecasting.** Despite great progress in long-range forecasting, very little progress has been made in our ability to handle precipitation and cloud formation. By the mid-1990s, we expect to have a thirty-fold increase in observational data. To utilize these data effectively, researchers will need high-performance computing power of up to 30 gigaflops (and the associated standards for MPP tools and languages) to enable design of a 5-km-resolution model.

3. **Reservoir and groundwater modeling.** The need for more powerful computers and standardized tools was also emphasized in the talks on reservoir and groundwater modeling. This Grand Challenge area deals with multiphase, multi-component flows. Currently, the level of application is restricted to grid blocks of thousands of feet by thousands of feet. For practical applications, researchers must be able to deal with blocks on the order of centimeters. While parallel domain decomposition solvers appear promising, experience with running these solvers on the Intel DELTA parallel computer have underscored the need for efficient, standardized parallel I/O servers.

4. **Global and earth systems.** Finer resolution was one of several recurrent topics in the presentations on global and earth systems. Oceanic models in particular require fine resolution (say, in the tens of kilometers), which in turn depends on the availability of teraflops computers and greatly increased storage. Coupled oceanic/atmospheric models raise additional problems. Currently, researchers are experimenting with sharing computer models at different institutions; issues such as distributed heterogeneous computing and user traffic over a gigabit network must be resolved, however, before such model sharing becomes practicable.

Regardless of category, the obstacles are similar:

- domain decomposition and load balancing,
- parallel I/O,
- speedup, and
- data management.

Not surprisingly, the approaches taken to address these obstacles are also similar (numerical simulations of PDEs and ODEs, Navier-Stokes equations, and spatial discretization of the PDEs), as are the numerical methods (finite difference, finite element, spectral methods, and fast Fourier transforms). The result of this similarity is not—as one might fear—duplication of effort; rather, the result is *value*

*Teraflops
computers will
enable researchers
to consider
atmospheric
models
incorporating
more realistic
conditions.*

*Teamwork
is enabling
researchers to
tackle larger, more
complex problems.*

added: we are seeing teams of researchers tackling many facets of the science and computational science.

Joining these teams are a number of U.S. industries. The active collaboration of utilities (including the Electric Power Research Institute and the Tennessee Valley Authority), the automotive industry (including GM, Ford, and the Motor Vehicle Manufacturers Association), the American Petroleum Institute, and the Consortium for Advanced Modeling of Regional Air Quality attests to the importance of this Grand Challenge research.

And this teamwork is paying off in the ability to tackle larger, more complex problems (such as the ocean conveyor belt). Indeed, the panel concluded that "exciting progress is coming in science, environmental management, and high-performance computing."

Panel B - Computational Physics

The first panel focused on understanding and managing our planet. The second panel session addressed the even broader topic of understanding our universe. Several areas of computational physics were considered:

- understanding solar activity and heliospheric dynamics,
- large-scale structure and galaxy formation,
- convective turbulence and mixing in astrophysics,
- scalable hierarchical particle algorithms,
- quantum chromodynamics,
- controlling chaos, and
- radio synthesis imaging.

*Complex physics
problems require
sufficiently
enhanced modeling
capabilities
to simulate
complex problems
over enormous
time periods.*

The computational complexity of these problems is enormous. Phenomena of widely different length scales are intimately intertwined, vast numbers of particles and vast periods of time are involved, and thousands of multidimensional integrals must be considered.

For example, a typical astrophysical fluid dynamics problem may involve transportation of heat energy, angular momentum, magnetic fields, and elemental substances; mixing of the turbulent fluids may be driven by thermal buoyancy, convection, and gravity waves. Faced with the impracticality of doing a real calculation, the researcher typically deals with dimensionless parameters, identifies the smallest feasible scale, and regards the simulation as a type of controlled lab experiment. Yet even such a severely restricted experiment may require a thousand hours on a CRAY Y-MP.

Another typical problem focuses on the evolution of galaxies and clusters of galaxies over billions of years. Scientists must consider such questions as, Are there "Great Attractors"? Are there gases or "dark matter" obscuring our view? To answer such questions requires archiving terabytes of data.

From the evolution of galaxies to the development of the universe is another step in computational complexity. What were conditions like initially? How have they changed from the radiation-dominated era to the matter-dominated era, and what

are they like now? Solving the physical equations required to answer such questions could literally take years using traditional methods. Researchers are therefore exploring new parallel schemes, including treecodes and fast multipole methods.

Common to all these examples are six fundamental needs: (1) portable code, (2) standards for languages and message passing, (3) interactive visualization tools, (4) increased computer storage, (5) a "reasonable" debugger, and (6) improved networking. As a first step toward meeting these needs, researchers are experimenting with remotely located telescopes connected by high-speed networks to very high performance computers and online data archives that are accessed over gigabit/second networks.

Panel C - Comp. Biology, Chemistry, Materials Sciences

Panel C combined three areas of critical societal importance: biology, chemistry, and materials science. In biology, new tools and advanced computer systems developed by the HPCC program are enabling researchers to analyze data from complex molecules in living cells. In chemistry, massively parallel computers are being used to investigate outstanding problems in environmental remediation. And in materials science, high-performance computing is enabling more accurate simulation of new materials—for example, for high-temperature superconductors.

The presentations focused on six areas:

- high-capacity atomic-level simulations for design of materials,
- computational quantum materials,
- computer-aided design of biological and biomimetic materials,
- high-performance computational structural biology,
- high-performance computational chemistry, and
- high-performance computing in structural biology and medical imaging.

Since the initiation of the HPCC program two years ago, more than ten production codes have been developed in these areas, with faster throughput and increased cost-effectiveness. A major achievement was a 150,000,000-atom simulation of the biaxial failure of ceramic material. This simulation provides a macroscopic validation of the microscopic force field.

Yet much more remains to be done. Three example problems illustrate the complexity of the computations being addressed. The first example involves study of the atmospheric degradation of alternatives to chlorinated fluorocarbons. Such alternatives are being explored in accordance with federal requirements to replace current refrigerants. Typical computations involve 20 reactions, each requiring 1,000–5,000 Cray hours, for a total of 20,000 to 100,000 CRAY Y-MP node-hours. The second example involves design of ductile, ordered intermetallic (NiAl) alloys. This problem involves 20,000 CRAY Y-MP hours per dislocation pathway. Since each computation involves 3–5 pathways, the total execution time is 60,000–100,000 hours of CRAY Y-MP node-hours. The final example involves protein engineering for bioremediation. Native microbes often cannot use the contaminants for food; therefore, scientists treat the microbe's enzymes to produce a more effective set. The challenge here is that the enzymes may have hundreds of thousands of atoms. Quantum

*Realistic modeling
will be possible
only with orders
of magnitude
increases in
computational
performance.*

*U.S. companies
are actively
involved in Grand
Challenge efforts.*

effects must be included; an approach based on classical mechanics is not enough. Moreover, a huge amount of Cray time is required to compute even a very small percentage of the enzyme's activity.

Researchers are confident that their goals can be reached. As computational power increases, quantum chemistry physical models are improving; and new techniques such as the force decomposition method are proving capable of handling problems with 10,000 atoms extremely rapidly on the Intel DELTA. To fully achieve their goals, however, the researchers agree that they need portable, high-performance math libraries. In particular, they need access to libraries of FFTs, random number generators, and fully distributed dense linear algebra solvers with flexible data structures.

One very encouraging note is the broad involvement of the chemical, metallurgical, petroleum, and pharmaceutical industries in addressing these Grand Challenge problems. Industrial corporations are collaborating with computational chemists, biologists, and materials scientists in universities and national laboratories. Already, in conjunction with the semiconductor industry, these scientists have engineered band-gaps for compounds with specified light-emission properties. And the potential areas of application are enormous, including

- imaging - electron micrograph reconstruction, X-ray crystallography, and real-time visible light microscopy;
- macromolecular structure analysis and prediction - protein folding and sequence analysis; and
- simulation of materials at the atomic level - quantum mechanical methods, classical molecular dynamics, and particle-continuum models.

Panel D - Computational Fluid and Plasma Dynamics

*Grid techniques
and discretization
methods appear
especially
promising.*

Massively parallel computers and advanced software technology are enabling scientists to model fluid flow and to perform design optimization of complex systems. These new capabilities have application in such areas as automobile manufacture, aerospace vehicle design, and medicine.

The panel on computational fluid and plasma dynamics focused on four specific topics:

- geophysical and astrophysical fluid dynamics,
- multidisciplinary analysis and optimization of aerospace vehicles,
- compressible and reacting flows, and
- fusion power.

Within these topics, problem sizes vary widely. Computations involving controls are insignificant. Problems involving structures may involve 10^5 degrees of freedom and require up to 1 gigabyte of memory for direct solution. Problems involving basic fluid physics may involve more than 10^7 points with as many as 20 words per point; and more memory may be needed for calculating reacting flows. Engineering flows may involve 10^6 - 10^7 points with up to 50 words per point; from 10 to 10^3

analyses are needed for optimization, where a typical analysis run includes as many as 10^4 iterations.

For most of these problems, researchers use grid techniques or discretization. Grid approaches are principally block structured (for fluids) or unstructured (for structures). Discretization methods include second-order to fourth-order finite volume (for fluids), finite elements (for structures), PIC (for fusion), and pseudospectral and finite difference (for geophysics).

These approaches have resulted in notable success, on a wide variety of high-performance computer systems. For example, researchers have carried out a 512^3 simulation of compressible turbulence on the TMC CM-5, a time-dependent adaptive block-structured fluids computation on the CM-2, a simulation of an aircraft engine on an IBM RS/6000 network, and aeroelasticity computations on the Kendall Square KSR-1. Such successes have attracted the interest of several industries:

- The NASA computational aerosciences program has established interactions with an industry consortium focused on HPCC technology and is expanding its traditional technology transfer to the aerospace industry.
- The national Tokamak project includes General Atomics and Microelectronics Computer Corporation.
- The computational fluid dynamics and combustion project is working with burner companies on the design of software for burner simulation.

At the same time, industrial interest has also focused attention on serious needs for these Grand Challenge projects. Mathematical and algorithmic needs are particularly apparent. Interdisciplinary coupling in aerospace is not well founded, especially for ensuring stability. Aerospace design would benefit greatly from optimization methods that are more tightly coupled to the analysis and less "black box." Engineering fluids simulations need improved turbulence models. And plasma simulations would benefit from models that incorporate the essential physics more economically.

From the viewpoint of the applications environment, all projects would like to be able to run graphics code directly on the MPP—not on a graphics workstation. In addition, most projects would benefit from the availability of object-oriented scientific databases distributed across heterogeneous platforms.

A third major problem common to all these projects is dataset management. These studies generate enormous, often multidisciplinary datasets. To exploit these datasets, researchers need rapid storage, access, and perusal—preferably in real time. Many scientists believe they would also benefit from the incorporation of scientific operations in a database management system.

Finally, physicists and engineers consistently noted four outstanding problems with today's MPPs: (1) peak vs sustained performance is too low (less than 10% for production codes using implicit methods on complex grids), (2) the I/O bottleneck is severe, (3) architecture-optimized code cannot be generated rapidly, and (4) little hardware support exists for timing/performance evaluation. The solution to these problems will greatly increase the usage of massively parallel computers in computational fluid and plasma dynamics applications.

Graphics techniques can accelerate the modeling process.

Panel E - Applications of Artificial Intelligence

This last Grand Challenge area was quite different from the preceding panels. Applications in artificial intelligence do not involve a well-defined set of matrices or equations. Rather, they deal with an environment that can understand human speech and language.

Two presentations were given:

- high-performance computing for learning, and
- data analysis and knowledge discovery in geophysical databases.

Both presentations observed that Grand Challenge research is overwhelmed by a sea of data. To evaluate this data efficiently, one would like to have an intelligent system that can handle English-like queries, can "learn," and can work quickly.

Two early successes were noted. The first involves development of a library of radial basis function algorithms to adjust "templates" in an effort to train a system to evaluate "normalized" patterns. The second success involves construction of a database system that can be used to examine the evolution of cyclones over time and space.

The researchers questioned whether a specially designed architecture will be needed for more extensive AI applications. A special VLSI architecture has been developed, and there are special parallel machines with fewer than 1000 nodes that use a database foundation. Serious questions about parallelism, scalability, and the I/O bottleneck remain to be answered, however.

*"Intelligent"
systems may
help evaluate
data quickly.*



SECTION 4

Grand Challenges and Industrial Applications

One of the stated goals of the federal High Performance Computing and Communications program is to spur gains in U.S. productivity and industrial competitiveness. Many of the Grand Challenge applications presented at the workshop included collaborative projects with industry.

On Friday, May 7, representatives from various U.S. corporations discussed examples of industrial involvement in high-performance computing. Corporations represented included DuPont, Ford Motor Company, Prudential Securities, Rockwell International, and Western Atlas International.

Chemical Applications: DuPont Corporation

*High-performance
computing
benefits major
industrial projects.*

Experimentalists at DuPont Corporation rely on high-performance numerical simulations for accurately calculating the properties of molecules. Such information is used to predict binding energies, tertiary structure, enzymatic activity, operation of diffusion-controlled processes, and inhibition and promotion of DNA expression—information necessary in the design of environmentally safe new enzymes. Researchers are also interested in using accurate numerical predictions for the design of chemical plants. Here the Grand Challenges span several sciences, including chemistry (thermodynamics, kinetics), chemical engineering (fluid dynamics), engineering (structural design), mathematics (optimization), and environmental control (waste minimization). DuPont is also involved in ecosphere/geophysical modeling, for example, to demonstrate compliance with environmental regulatory agency regulations or to identify favorable locations to drill for oil and gas.

Automotive and Manufacturing Applications: Ford

Ford Motor Company uses high-performance computers in diverse computer-aided engineering (CAE) and computer-aided design (CAD). By displaying CAE/CAD data as realistic three-dimensional objects, researchers hope to dramatically reduce the time required to construct, modify, and evaluate design concepts. In one case (a styled wheel), Ford scientists have cut the process from 24 days to 16; in another (a control arm), they reduced the process from 33 to 15 days. In both cases, the use of high-performance computers and advanced software permitted the scientists to triple the number of elements analyzed, thus providing a more accurate simulation.

Financial Applications: Prudential Securities

With an unpredictable number of diverse users throughout the world, and strict time constraints imposed by global markets, Prudential Securities relies heavily on high-performance computers (including IBM RS/6000 X-stations, a 72-node Intel Paragon, and a 32-node Intel i860 hypercube). The company views scalable parallel computers as a "computational opportunity" for gaining a competitive advantage in selected strategic markets. Prudential is particularly interested in improved software and interfaces that will enable interactive, real-time computing on a globally distributed network of heterogeneous parallel computers.

Aeronautics Applications: Rockwell International

Rockwell International scientists have turned to high-performance computing to provide what they call "affordable system optimization." The objective is to develop an optimization process that best satisfies customer needs, at a given cost, through all stages—from specification of requirements, through design and manufacturing, to operations and support. Rockwell is investigating Taguchi methods and concurrent engineering techniques to improve design and production process.

Industrial applications include reservoir modeling, design optimization, textile processing, automotive engineering, global financing, molecular modeling, and simulation of the brain.

Energy Resource Extraction Applications: Western Atlas

The huge computational demands of reservoir simulation and seismic processing have made oil companies some of the earliest purchasers of advanced computers. Western Atlas International believes that numerical reservoir models are becoming even larger, with more components, more unknowns, and a proportionate increase in the number of runs required. Such applications will require high-capacity parallel computers. Effective use of these computers will, in turn, require significant advances in systems software—new algorithms for solving stiff systems of linear equations, improved load-balancing methods, better interprocedural analysis tools to handle very large applications (500,000 lines of code), interactive debuggers for each processor, and emulators capable of simulating parallel operation on a single-processor workstation.

Textile Applications: DuPont Fibers

The textile industry and the DOE national laboratories have joined in a partnership called AMTEX. The objective is to develop the information technology necessary for "a demand-activated production system." According to DuPont Fibers, such a system will require advanced systems integration and multidisciplinary design capabilities. The result, it is expected, will be a step-change productivity increase.

Health Care Applications: Phillips Medical Systems

Phillips Medical Systems scientists are addressing three Grand Challenge applications: (1) generation of and access to very large databases with fundamental clinical information, (2) performance modeling of imaging and information management systems, and (3) high-speed computations and communications for intelligent information management. For example, to make effective use of an anatomical, neuro-functional, physical-parameter four-dimensional atlas of the brain (covering a scale from 1 micron to 20 centimeters and the entire human life span), researchers need very fast, expert-driven database management tools.



SECTION 5

Working Groups

Following the presentations by the applications teams, the workshop broke up into working groups comprising both computational scientists and software technologists. The focus was on the software technology requirements for Grand Challenge applications research.

Group A – I/O, Data Systems, and File Systems

Chair: Reagan W. Moore; Co-chair: C. Roberto Mechoso

The current trends in computer hardware have resulted in CPU speeds increasingly outstripping I/O capability for supporting both dataset sizes and dataset access rates. Many applications are limited by the I/O capacity of current hardware, and compromise by working with reduced problem sizes. Research efforts are needed for the development of high-performance file systems that integrate local disk cache with archival storage, as well as for the development of database management systems supporting scientific datasets.

Increasingly larger datasets have imposed enormous demands for efficient input/output systems.

Status

The working group focused on two particular issues: data size and data rates.

Data Size of Applications. Typical current and near-term HPCC applications involve data traffic of from 1 GB to 4 TB per run. Several applications envision writing out 10-GB files in the near term and 100-GB or larger files in the longer term per run. This projection has implications for staging areas. In some applications, the amount of output data produced by one run depends only on how long the

simulation is run. In other words, the output size is directly influenced by the CPU cycles available.

With increasingly larger data runs possible on HPCC platforms, archival volume requirements are increasing at least as fast as processor performance. Current archival sizes per Grand Challenge group typically range from 500 MB to 500 GB, with a peak of 10 TB. Anticipated requirements go up to 10 PB per group.

This enormous amount of data imposes more than just storage requirements. Several simulation-type HPCC applications generate large numbers of files (10^5) and corresponding numbers of job submissions, requiring substantial human management time. Some groups are attempting to alleviate this problem by increasing staging area storage and developing better data-management software.

Data Rates. Current and near-term HPCC applications involve I/O rates of 1-40 MB/s to disk and 0.5-6 MB/s to archive. Users indicate that 1 GB/s to disk and 100 MB/s to archive will be required in the near future. This disk rate is achievable today on appropriately configured CRAY C-90's, but is not achievable on most MPP platforms. The archival rate is not possible in any cost-effective way today.

The choice of algorithm strategy and the necessity for the recomputation of data are strongly influenced by the ratio of I/O to CPU power. For instance, current limited I/O rates are forcing more codes to rely on "in-core" solution algorithms.

Targets of Opportunity

Fast and convenient access to data—whether remote or local—is needed.

This section is organized top-down through the storage hierarchy: from primary memory through secondary and tertiary storage. Requisite network communication is implicit in all levels of the system. The main theme of recent I/O research is greater performance by increasing parallelism in all levels of the system, that is, beyond the massively parallel processors.

Memory-Level I/O Research. Research is under way in compiler and run-time mechanisms to utilize the memory hierarchy more efficiently, for example, by data blocking, data placement, prefetching, file caching, and run-time analysis of I/O access patterns. The goal is to give the illusion of a giant main memory by static and dynamic optimizations that improve locality of accesses. One example is memory servers, which expand the primary memory available to a given processor by using memory in idle nodes within an MPP.

Disk-Level I/O Research. At the disk level the two major areas of research are high performance and schemes for storing information found in MPPs. High performance includes I/O benchmarks, high-performance file systems for MPPs, and parallel disk organizations for MPPs. I/O benchmarks are needed to give realistic metrics for evaluating alternatives in trying to accelerate Grand Challenge applications. The lack of benchmarks has inhibited progress in many fields, while good benchmarks have led to rapid innovation.

Research in storage schemes includes parallel file systems for MPPs, persistent object stores, and scientific databases. The goal of parallel file systems is to remove sequential bottlenecks between parallel processors and parallel disks. Persistent

object stores and scientific databases are new paradigms to add type information and structure, enabling researchers to access information more intelligently and more efficiently.

An important consideration for all the topics at this level is convenient and fast access over local and wide-area networks for distributed data. Grand Challenge applications have huge data repositories that must be made available to researchers across the organization and the country.

"Tape"-Level I/O Research. "Tape"-level I/O includes all storage media accessed via robots: tape archives, optical jukeboxes, and network-attached peripherals. At this level, migration strategies for network-attached peripherals are still being developed. A key topic is extending automatic migration from local disk to the next two levels of the storage hierarchy. Data management issues include databases for large and numerous objects and data organization for different access patterns. New latency-tolerant schemes must be invented to hide the inherent latencies of tape robots, measured in tens of seconds, so that this massive capacity is efficiently used. A final challenge for this area is offering high-bandwidth data transfers to the bottom of the memory hierarchy.

Relationships

The research topics identified in the preceding section raise pointed questions about existing I/O policy and relationships between applications and software developers.

Software Sharing. Technological software-sharing mechanisms include anonymous FTP, Gopher servers, and the hypermedia MOSAIC software. These mechanisms are heavily used by the community. The near-term establishment of distributed file systems such as AFS and DFS will further facilitate software sharing. The NASA HPCC software-sharing mechanism should be more strongly promoted, and its use by all the Grand Challenge applications groups encouraged.

*Parallel I/O
benchmarks must
accurately reflect
applications
requirements.*

Balance. The HPCC program has emphasized teraflops applications. This focus has exacerbated the divergence of CPU speed and I/O capability: I/O is not scaling as fast and is becoming a major bottleneck for teraflops applications. Storage of data associated with real-time, I/O-intensive applications will require stronger funding support to create easily used distributed systems. For many applications, both the amount of data that must be stored and the rate at which the data needs to be accessed will soon increase beyond the capability of current systems.

Interdisciplinary Interactions. The interdisciplinary Grand Challenge teams are helping break down sociological barriers between computer scientists, applications scientists, and system integrators. Improved collaboration between these groups can be facilitated through development of parallel I/O benchmarks that accurately reflect applications requirements. These interactions can be further enhanced by considering improvements in I/O and storage to be part of the deliverables for these projects.

Vendor Software. Vendors are moving in the right direction for supporting Grand Challenge application I/O requirements—but too slowly. The HPCC community needs to promote vendor standards for archiving large amounts of data. An example is an industry effort to define standard interfaces for databases to make use of tertiary storage. Commercial support for scientific databases also needs to be encouraged.

Industrial Applications. The very large amounts of data generated by Grand Challenge applications is forcing systems software development for efficiently handling large objects. Scientific application projects have increased the understanding of the importance of I/O among computer scientists and between them and application scientists. High-performance scientific databases can have a significant effect on areas outside the current HPCC focus (e.g., medical image database, geographical information systems).

Findings

Effective I/O requires robust file systems and database management systems.

The findings of the I/O working group are organized into two parts: (1) development of secure, robust, high-performance, high-capacity, transparent file systems and (2) development of database management systems supporting scientific datasets. We feel that each of these areas is important in its own right, and we therefore rank activities within each area rather than across both areas. We also recognize that a great number of interdependencies exist between areas. In particular, for many applications there is a need for layered access to storage, which will require careful integration of many of the components discussed below.

The performance parameters in the following discussions are based on requirements for the next few years.

File System Development. The group identified four findings in this section:

- Provide high-performance local file systems to keep jobs CPU bound (rate 10–500 MB/s, size 300 GB–1000 GB). This effort comprises support for research and development of parallel file systems (including the goal of standardisation), standard parallel file system interfaces from programming languages, parallel run-time library interfaces, and serialization methods for files stored in parallel file systems.
- Develop high-performance archival storage systems (rate 1–50 MB/s, size 10 TB to 1 PB). No archival systems are available with high-speed parallel support. This should be an important priority. It will require support for staging data before MPP execution begins, support for subsetting of large datasets, development and standardization of archival storage interfaces to local parallel file systems, and support for third-party transfer to network attached peripherals and systems.
- Ensure ubiquitous access to conventional distributed file systems (rate 1–10 MB/s). Such access includes client-level support of distributed file systems (AFS and DFS) from within MPPs, interfaces and migration schemes for integration of distributed file systems with archival storage systems, large file

capability in distributed file systems, and scaling for support of a very large number of files.

- Ensure ubiquitous access to high-performance distributed parallel file systems (rate 10–500 MB/s). Research and development are needed in distributed parallel file systems. Also needed is support for third-party access in a distributed parallel file system.

Database System Development. Five specific areas were suggested for database system development:

- Support for large objects
- Fast access to subsets of large datasets
- New indexing and access methods for scientific data
- Interface to tertiary storage systems
- Language interfaces to parallel database systems

The group further suggested that a set of Grand Challenge benchmarks and sample data be created for all levels of computer systems. This set should be made easily available to computer systems researchers so that they can evaluate new ideas to improve performance of Grand Challenge applications.

Group B – Parallel Programming Paradigms

Chair: Joel Saltz; Co-chair: Geoffrey Fox

Choosing the right programming paradigm is important. The Working Group on Programming Paradigms considered the advantages and limitations of data parallel programming and other programming models such as message passing, shared memory, and data flow. The discussion focused on the fit between the various paradigms and the HPC applications.

Status

The working group reviewed results of the written survey of the Grand Challenge teams and discussed the general opinions of data parallel programming with applications researchers in the group. To supplement this information, the working group constructed a set of questions with the intent of getting workshop attendees to clarify their positions on some specific issues. Three other working groups with a high concentration of application researchers were approached with the following questions:

- Which paradigms do you prefer: shared-memory (name space) process parallelism, message passing, data parallel, or none of the above (give counterexamples)?
- Suppose that there is a "good data-parallel (Fortran)" compiler to handle general unstructured and block-structured problems. Generalize this with large-grain-sized functional parallelism support for such data-parallel modules. Would this model be useful? sufficient?

Researchers must know how to choose the right program paradigm for each given problem.

- How much efficiency would you be willing to give up in exchange for a major improvement in productivity (e.g., automatic parallelism)?
- The programming paradigm supported by KSR is an example of one that supports quick ports (with poor initial performance) and incremental code optimization and parallelization. How important to you is this sort of capability?
- How important is C++ functionality? Why are you using C++ (if you are)? What particular features of C++ do you find useful? In particular, do you make use of inheritance?

Based on all the input from the surveys and discussions, the working group identified the following items as summarizing the current status of use of programming paradigms.

No single paradigm is the answer.

1. No single paradigm is "the answer," even within a single program. The current software provided by vendors represents different programming paradigms: data parallel, basic shared memory, and message passing. There is no consensus among Grand Challenge applications people on the best model to use. To some extent the choice is based on taste, but it is also based on specific application domain considerations, on software availability, and on different types of functionality found within a given program. The working group noted that there is a hierarchy in the choice of programming paradigms. Users express a definite preference for general, portable, high-level languages such as C++ for portions of programs associated with preprocessing, functional control, and postprocessing. When it comes to computationally intensive parts of programs, however, applications researchers are more likely to use a language for which the compilers are more efficient, such as Fortran. There is a further tendency among researchers working on heavily used algorithms (e.g., FFT or linear equation solvers) to focus even more closely on efficiency with explicit message passing and even assembly language.
2. Currently, each vendor tends to promote only a single paradigm (matching its architecture) and lacks the resources to give substantial support to multiple paradigms. This situation is in contrast to item 1 above. Furthermore, the hardware technology cycle is faster than the robust tool development cycle, leading to constant pressure on paradigm development.
3. Portability of applications is an important consideration. It is not so much the case that users expect to be moving their codes daily, but rather that the underlying hardware is still in a state of flux. Users are uncomfortable with committing substantial optimization efforts to vendor-specific language extensions, message-passing systems, or assembly languages when they expect hardware changes.
4. Informal standards are emerging. The need for portability across architectures has been noted, not only by the users, but also by the vendors. As a result, informal volunteer efforts have emerged during the past year to address this issue. The High Performance Fortran Forum (HPFF) has drafted a data parallel programming interface which will be common across multiple vendor

Portability is vital, in view of the diversity of computer architectures.

architectures. This effort is expected to continue to extend the class of algorithms and data structures addressed. Following the HPFF model, a similar effort is under way to provide a standard message-passing interface (MPI). These efforts have fairly wide participation and user interest.

5. Rich data and operation abstractions are getting increasing attention because of the resulting code simplification and reuse. C++ is being used by several projects, and there are reports of substantial productivity gains due to the higher level of abstraction.

Targets of Opportunity

The working group identified several targets for new research directions.

Conducting Empirical Studies of Programming Paradigms. Many paradigms suffer from the lack of critical evaluation. The studies proposed here should involve the porting of realistic applications and should be carried out in close collaboration with applications researchers. The results would answer many questions regarding the suitability of programming paradigms for both general use and for specific topics, and could be used to drive the development of the next generation of computer software.

It is important to note that some of these studies may fail to produce usable codes; from the viewpoint of developing programming paradigms, such negative results are as important as producing good codes. In order to come to a broad understanding of the applicability of programming paradigms, the studies should also include industrial and commercial applications.

Achieving Better Performance with Data Abstraction. A wide range of Grand Challenge researchers expressed strong interest in the use of data abstraction. The need for data abstraction is particularly pressing to researchers who solve problems with complex hierarchically defined data structures such as structured adaptive grids or irregular tree-type data structures found in fast multipole algorithms. Most Grand Challenge applications scientists who employ data abstraction use C++. Unfortunately, current compilers typically do not generate efficient code when confronted with abstract data types and inheritance. (Fortran 90 also supports many aspects of data abstraction, but there is little experience with this language in the Grand Challenge community, because of the poor availability of full Fortran 90 compilers. Moreover, some of the applications researchers are exploiting the C++ feature of inheritance, which is not supported in Fortran 90.)

Supporting Incremental Optimizations. The ability to port a code to a new architecture quickly and optimize for performance later is very attractive to users. For example, this is possible in shared-memory programming paradigms, such as those supported by Kendall Square Research, software implementations of virtual shared memory, or (future) compilation methods. For this approach to be successful, users need to be able to run with realistic data sets and obtain feedback that they can use to determine how to optimize performance. Tools and techniques that facilitate incremental performance optimization are potentially important.

*Paradigm studies
must include
realistic industrial
and commercial
applications.*

Providing Language, Compiler, and Programming Paradigm Support for Irregular, Block-Structured, and Adaptive Problems. A large fraction of Grand Challenge applications surveyed make use of sparse data structures or linked lists (16 out of 28 teams). In addition, 12 teams employ block-structured grids, and 4 teams make use of nonuniform tree-type data structures for fast multipole algorithms. Also, dynamic adaptivity is growing in importance. In order to prove attractive to a broad range of Grand Challenge researchers, programming paradigms and associated compilers and run-time support will have to be able to effectively handle irregular problems. Continued research on programming paradigms, compiler techniques, and run-time support aimed at such problems is needed.

Relationships

Many working group members from computer science emphasized the positive influence work with applications people has had on their research. This included supplying practical problems and providing valuable feedback. The feeling of the Working Group is that computer science and applications should be tightly intertwined. However, there is also a significant feeling that much of the current interaction is only token discussion. Instead of this, real interaction through day-to-day contact is needed. For this to happen, both computer science researchers and applications scientists must adapt. Computer science people must be willing to rapidly prototype relatively prosaic (at least not leading-edge) systems, and applications scientists must be willing to tolerate the occasional system error.

A fundamental question raised in the working group was, *What is the purpose of Grand Challenge grants?* The Grand Challenge applications described at the workshop focused on the output of the applications rather than on the software used to obtain that output. In these cases, the computer science component was often regarded as peripheral to the main goal of the grant. Applications specialists, who are judged on the quality of their science, are understandably reluctant to wait for computer scientists to develop general, robust systems. In HPCC projects where computer science is more central, the opposite problem may occur—scientific results may become peripheral. This reflects a more general problem: computer scientists and applications experts have different professional goals. Applications scientists must develop new scientific results, while computer scientists must focus on general computational principles. However, the funding agencies might improve collaboration by clearly specifying the equal importance of computer science output and applications science output in evaluating grant results. Also, new incentives are needed to encourage interdisciplinary collaboration.

Another way to improve the effectiveness of computer science researchers in the programming paradigm area would be to increase the visibility of programming language research outside the computer science community. This could lead to more use of particular programming paradigms to express advanced algorithms, resulting in better evaluation of new languages and systems. It would also help ensure that new language developments would be relevant to the needs of the applications scientists.

It is worth noting that computational scientists developing numerical algorithms face similar problems. One working group member reported that his group minimized this problem by having a "captive" applications expert for doing evaluations. This enabled them to attack hard problems all the way from the problem definition

New incentives are needed to encourage interdisciplinary collaborations.

The federal HPCC program has heightened interest in parallel computing.

to its implementation. This may provide a model for work on parallel programming paradigms.

Impacts

Perhaps the overriding sense of this working group was that, at this point, it is really too early to expect significant impact from the HPCC program outside of the program itself. That said, however, it is important to make a few observations about the potential impact of the HPCC program.

It is clear that the HPCC work has heightened interest in parallel computing, and, as a result, it has increased the pressure on hardware and software vendors and the computer science research community to produce the products (machines, languages, tools, etc.) that will make parallel computation effective. It has also led to increased interest in standards, which will permit application developers to build programs based on paradigms and software infrastructure that will be supported on a large number of different parallel computing environments. Examples include High Performance Fortran (supporting the data parallel paradigm) and the evolving Message Passing Interface (supporting a process parallel paradigm) standards. In the future, this sort of technology should have a major beneficial impact on non-HPCC computing.

While the working group has no direct information concerning the amount of technology transfer to date, it is clearly important to transfer the HPCC community's experience with parallel paradigms to those performing high-end computation in industries such as aerospace, petroleum, and pharmaceuticals. Many of these industries depend heavily on solving computational problems that are quite similar to those attacked by HPCC researchers, and they can benefit from knowledge of what has worked well for HPCC researchers, even if they are not in a position to make direct use of application software developed in the HPCC program.

Findings

The working group made the following suggestions:

1. Methods should be developed to allow mixing of different programming paradigms and languages in a single framework.

Examples of such mixing might include standard ways of calling message-passing Fortran from C++, invoking message-passing code from HPF, or coupling a heterogeneous set of data parallel programs. At the current time, vendors do not view the support of mixed-programming paradigms as a particularly high priority. Researchers interested in multidisciplinary problems also expressed a strong interest in large-grained, process-level parallelism with subordinate data-parallel modules.

2. "Templates," which capture key computational, communication, and I/O characteristics of important application codes, should be written and made widely available.

Templates are well-structured, documented algorithms to solve key computational problems. These templates could be used to carry out research to

Experience in program paradigms must be transferred to aerospace, petroleum, and pharmaceutical industries.

Mixed programming paradigms should be integrated into a unified framework.

develop and evaluate programming paradigms and for multiprocessor performance studies. Several efforts similar in spirit to this are already under way, including the PBWG group collecting parallel benchmarks, but additional efforts should be encouraged.

3. Widely available, modular components for software tools should be developed to speed the implementation of new programming paradigms.

Measures are required to reduce the time and the costs associated with tool and compiler development because of the rapid pace of hardware development. Examples of such projects include the ARPA initiatives to develop a common set of parallel compiler run-time support routines and to develop a compiler infrastructure.

4. New incentives should be developed to encourage computer scientists and application scientists to interact more closely.

These incentives could include new awards, sabbatical-in-place arrangements to allow interaction with applications groups, or specific funding of software development for particular applications. The model of having an "outside" member within the group should also be considered. Most important, grants must recognize that computer science and application science results are equally important (and the problems equally challenging).

Group C - Performance Characterization and Evaluation

Chair: Dan Reed; **Co-chair:** Robert Harrison

What sets performance tools apart from other software tools in the HPCC arena? The answer is simply that performance is the ultimate reason for using massively parallel architectures in the first place. Either absolute performance (solving larger problems faster than previously possible) or price performance (solving the same problems less expensively) is ultimately the driving rationale for using MPP systems. Thus, achieving a reasonable fraction of the potential peak performance of such a system is an essential part of the code development process. The development of stable performance analysis tools represents an enabling link in the evolution of an application from a crude parallel model to an efficient research or industrial code. However, in order for such tools to be used with confidence by a broad user community, they must move beyond research prototypes to become robust, commercial-quality products.

Status

Many tools are available for evaluating performance on a single processor, a few tools for a moderate number of processors, and almost no tools for a large number of processors. The problems plaguing Grand Challenge applications developers also hinder the rapid development of robust performance tools:

- unreliable hardware,
- immature system software, and

To understand the performance of massively parallel processing applications requires tools focusing on scalability, load balance, and program performance.

- incremental delivery of software.

Many researchers are investigating research aspects of performance evaluation for massively parallel systems. However, not all research groups have the expertise or interest in producing distributable software, and few of the groups that do have the interest and expertise have the necessary financial support and personnel to produce complete, robust tools.

Vendors are announcing new architectures so rapidly that their internal performance tools group have difficulty producing robust, effective tools—both from rapidity of development and from difficulty in securing the necessary resources. Some of the vendors realize this problem and are reaching out to the research community, but there have been few successful technology transfers of high-quality software from academic or national laboratory research groups.

The result is a paucity of robust performance tools for popular parallel systems. The perception of the (potential) user community is that this problem is even worse than it is, with little awareness of what tools are available. For example, there are several tools for balancing loads across processors, yet several users at the workshop mentioned the need for such tools. This situation is exacerbated by difficulties in disseminating performance analysis software:

1. No mechanism exists for exchanging information on tool availability and accessibility.
2. Significant restrictions (stemming from the current legal and funding environment) do inhibit the dissemination of software.
3. Even the research groups with the interest and staff to develop near-production-quality tools may not have the interest and staff to continue maintaining and developing these tools, handling user comments and complaints, etc.
4. Many application groups and computer centers do not have the resources/staff to investigate what tools would be useful for the local user community, to install tools, and to support local users.

Information about new tools must be dispersed quickly, to enable researchers to exploit new architectures.

Targets of Opportunity

The working group identified four areas meriting special attention.

Support for High-Level Languages. High-level programming languages present special needs for performance measurement tools to support source code optimization by applications programmers. Also, the user interface to performance measurement tools should be accessible through high-level languages. An opportunity exists for an enhanced relationship between measurement and languages to provide the user with a more powerful program development environment. Requirements include

- measurements correlated with high-level semantic constructs of emerging high-performance computing languages,
- compiler optimizations traceable to source code and performance implications provided to the programmer.

- automated instrumentation for new languages,
- languages/directives for explicit control of measurements, and
- hardware and software support for access to hardware performance instrumentation.

Examples of these facilities exist, but they should become standard within HPCC languages. The goal is to make measurement information on MPP performance as accessible as symbolic debugging information is on conventional workstations.

Needed are performance analysis tools that consider the effects of distributed resources.

I/O and Distributed Systems. The performance of HPCC systems is increasingly affected by disk I/O and network I/O to external systems such as mass storage, visualization devices, workstation clusters, and other heterogeneous components. Conventionally, consideration of these I/O capabilities has been segregated from performance evaluation of computer systems. The large datasets of Grand Challenge problems and the trend towards application of distributed systems require evaluation of external support systems in combination with primary MPP computers. Application programmers must be provided the means for measuring performance parameters affecting their program execution, even when operation is on a remote subsystem. Both processes and data may be distributed among heterogeneous processing platforms. The resulting data accesses or the process communication can limit the overall performance. Access to I/O and remote subsystem measurement tools is essential for application programmers to determine the causes of performance degradation. Additionally, distributed resources tend to be shared, and delays caused by the resulting contention must be included.

Open Performance Tools and Infrastructure. One possibility for increasing the impact of current performance tools is to make them easier to use and more readily interoperable within an open performance system infrastructure. Higher-level languages, more complex component modules, and heterogeneous systems are being used in the construction of application codes; current performance tool technology must be applicable to these new target areas. Furthermore, access to performance data should be provided by language, module, and system developers for integration into open performance measurement systems.

Good tools are better than standard tools.

In contrast to performance tool standardization, we advocate an investment in performance tool infrastructure development with an emphasis on open technology. It is better to have good tools than standard tools; but if tools are made interoperable through open interfaces, then good tools could be more easily applied to new target areas. Specifically, we suggest

- interfaces to performance measurements (developed by application and system software module developers);
- performance instrumentation and database components, integrated with compilation systems;
- development of scalable performance presentation technology, accessible to performance tool builders; and

- specification of a default level of basic performance measurement and analysis functionality for parallel systems, providing a basis for transportable performance tools.

Performance Characterization, Modeling, and Experimentation. Performance analysis has both short-term and long-term goals. Typical short-term goals are to help write and tune programs that run fast for current problems on current computer systems, and to establish procurement criteria. One long-term goal is to produce programs that will also perform well in the future (e.g., for larger or longer problems on computer systems with more processors or different computation/communication tradeoffs). A complementary long-term goal is to help design computer systems that will effectively support future applications.

Current tools are usually specialized.

All of these goals require determining the performance characteristics of application programs and computer systems. However, currently available tools and techniques provide only a subset of the capabilities needed to meet these goals. These tools are generally restricted to empirical measurements of a particular combination of code, system, and sample problems. Little or no support is provided for developing predictive models of application performance, or even for acquiring the information needed to construct such models. Additional work is needed characterize massively parallel systems and extract application requirements, as well as to construct and validate models.

These gaps can be filled, in part by applying existing techniques to a wider range of applications and systems, and in part by further research collaborations between computer science researchers, application developers, and vendors. Our major suggestion is that

- each major MPP class and application community should be supported by at least one research group chartered specifically to characterize the performance of important applications on that MPP class, both now and with an eye toward future scalability, and to develop the tools and techniques necessary to support such characterization.

Relationships

Are computer science people developing the right tools? The answer to this depends on whom one asks. Performance tools such as PICL and Paragraph from Oak Ridge National Laboratory, PRISM from Thinking Machines Corp., or ATEXPERT from Cray Research have many enthusiastic users in the HPC community. On the other hand, some respondents to the workshop survey believe that the state of programming tools is dismal. Possible reasons for this attitude might be that

User-friendly tools are essential to increased productivity.

- parallel tools, like parallel programs, tend to be quite complex;
- tools are often geared to system developers rather than applications developers;
- attention spans for learning or relearning tool use are notoriously short;
- scalability of performance analysis tools to large numbers of processors is an open research question;

- it is very hard to make a tool friendly to all users;
- users are quick to blame the tool (or tool maker) for any problem they encounter; and
- new tools usually lack adequate documentation.

The working group did suggest several new ways to improve collaborations:

- Funding from government agencies is needed to enable applications users, tool developers, and representatives from the vendor community to work together to develop and refine tools to make applications developers more productive.
- Technology transfer programs such as that fostered by the Software Capitalization Program from NSF to enable developers to transfer tools from a research environment to the market place, and therefore to the applications developer.
- Funding, such as ARPA fellowships, that would enable computer science students to spend 6-8 weeks with applications developers, or students in a scientific field to work with tool developers.
- Education projects for users, such as a catalogue that summarizes available tools along with extensive help in the uses of the tools, possibly accompanied by a videotape containing software demonstrations.
- Education projects for tool developers and performance tool users to determine real user needs that would drive development.

Impacts

The HPCC program has already had a dramatic impact on computer science research in general, and performance tools in particular. One measure of this impact is the widespread interest in MPP systems among academic computer scientists, which is in marked contrast to their general lack of interest in the previous generation of vector supercomputers. Applications code developers and users, on the other hand, are also interested in exploring MPP systems, but many remain to be convinced of their practical utility. One benefit of this renewed interest in leading-edge, high-performance architectures is that for the first time in more than a generation, computer scientists and applications scientists have a common interest—namely, MPP systems—that could serve as a basis for substantial collaboration and cross fertilization. Performance analysts and tool developers can play an especially critical role in broadening the acceptance and effectiveness of MPP systems for large-scale applications. Because of the immaturity of algorithms and systems software for MPP platforms, current users need all the help they can get in realizing the high performance offered by such architectures. Moreover, many potential “converts” to MPP in industry are reluctant to take the plunge until they know that a mature environment (including development tools and/or third-party applications codes) will be available to enable them to use these systems effectively.

Most potential industrial users are not inclined to invest time or financial resources in a parallel machine without some proof of principle that they will receive

For the first time in more than a generation, computer scientists and applications scientists have a common interest—MPP systems.

adequate return through enhanced performance. One approach might be a collaboration among a tool developer, a scientific research group, and industrial partners in the co-development of both an application of interest and an emerging performance analysis tool. Such a collaboration would provide the tool developer with vital feedback on the efficacy of the tool, while the application developers benefit from the performance tuning that the tool enables. This symbiosis would address one of the main problems noted in our discussions, namely, that applications developers simply either do not know about performance tools or do not know how to use them. Remedying this situation could have a strong impact on applications developers within HPCC and on the competitiveness of U.S. industry as more companies become convinced that large-scale applications codes can be addressed effectively on massively parallel machines. Performance characterization can also play a critical role in the procurement process, particularly in facilitating an appropriate match between applications requirements and machine characteristics.

Findings

We summarize the suggestions of the working group participants as follows:

Technology transfer is a key aspect of developing programming tools for massively parallel computing.

- HPCC projects should explicitly fund the exchange of students and staff between application and tool developers. Such exchanges will provide training for both groups, influence the development and use of performance analysis tools, and establish long-term research collaborations.
- The collection, maintenance, and distribution of a catalogue of performance tools and demonstrations should be a separate and explicitly funded activity. This includes tutorials on basic concepts of performance evaluation, as well as identification of actual tools for specific purposes.
- A permanent funding mechanism (similar to the NSF Software Capitalization Program) should be established to bring software tools to the point where technology transfer is both feasible and economically attractive.
- Collaborative development of tools to construct predictive performance models for scalability of Grand Challenge applications should be supported. Such models are vital in reducing risk and development cost in achieving high performance on current and future architectures.
- In tool R&D, more emphasis should be placed on integrating support for I/O and networked services, in anticipation of these activities becoming more critical to applications.

Group D – Program Development Tools

Chair: Fran Berman; Co-chair: L. Ridgway Scott

A vast array of tools exists to support parallel computing. These tools come from three different sources—computer vendors, the research community, and commercial software companies. The existing tools fall into a number of categories:

- node compilers (e.g. native compilers, AC);

- parallelization support tools, including data decomposition and mapping tools (e.g., FORGE90, KAP, PYRROS, HeNCE);
- debuggers (node and parallel debuggers);
- parallel programming languages (CM-Fortran, Fortran-M, Fortran-Linda, HPP, PC N, pFortran, Strand, etc.);
- communication libraries (active messages, CMMD, Express, NX/2, PICL, PVM, p4, TCGMSG, etc.);
- program/performance analysis tools (e.g. IPS2, PABLO, Paragraph, Upshot, etc.); and
- math libraries (e.g., ScaLAPACK).

Status

The tools frequently do not address the problems of interest to application programmers, do not function as advertised, and/or do not deliver a significant fraction of the performance available from the computer. The availability, performance, and spectrum of tools for parallel computers must be improved across the board: existing tools must work and be robust; new tools must be developed to support parallelization, MIMD programming, etc. The poor quality or lack of these tools inhibits the growth and extent of the high-performance computing community.

A uniform complaint voiced by the group was that the software tools provided by the MPP computer vendors are grossly inadequate. The core set of tools available on MPPs—operating systems, compilers, debuggers, performance analysis tools—is not up to the standards of robustness and performance expected for commercial computers. Tools are nonexistent or, even worse, do not work. Certain classes of tools were singled out as being in a particularly poor state. Parallel debuggers received the most criticism. Parallel math libraries were thought to be almost as bad, though math libraries are essential for building a mature application software base for parallel computing.

Commercial and public domain sources have tried to fill the gaps left by the computer vendors, but have had varying levels of success. Many active research projects are also producing potentially useful tools, but these efforts are available only in prototype, or fragmented and buried inside various application efforts. A key issue for this working group was how to develop these technologies into effective tools.

Targets of Opportunity

Many tools for parallel software development have been created; however, application programmers often do not know which tools might be applicable, how to get the tools, or how good they are. We must develop and strengthen mechanisms to turn research prototypes into well-developed tools with a broad user base.

We need enhanced mechanisms to turn prototype tools into well-developed tools with a broad user base.

Basic Tools and Libraries. The primary target of opportunity is to develop robust, effective versions of basic tools, many of which purportedly exist but are not usable. This suggestion is in contrast to an approach that looks for sophisticated "magic bullets" such as HPF—an approach greeted with considerable skepticism by application programmers within this group.

Infrastructure. A mechanism for the exchange of high-performance software is critical. This mechanism must be aggressively publicized, nurtured, and used. Support and maintenance for tools in this exchange *must* be addressed to make this an effective mechanism for software sharing.

Distributed, Network, and Heterogeneous Tools and Libraries. One area in which new tools are seen as necessary by both users and computer scientists is the burgeoning distributed, network, and heterogeneous (DNH) arena for high-performance computing applications. A clear target of opportunity is to develop software tools based on experimental efforts in DNH. Homogeneous systems and very heterogeneous systems represent distinct targets of opportunity because of the fundamentally distinct research issues they raise.

Distributed Data Management Tools. A critical area of need voiced by almost every HPCC applications group is tools to support distributed data management. Significant attention must be devoted to the development of these tools.

Relationships

Applications programmers need parallel software tools that are robust, portable, and easy to use. In addition, the tools must be very well supported, with clear value apparent to the users within a short evaluation time. This level of sophistication requires a substantial development effort.

Unfortunately, it has fallen to the computer science research community to develop these tools. We say "unfortunately" because academic computer scientists are rewarded for basic research, not for software development. Furthermore, computer scientists are not always aware of the problems of interest to application programmers. As a result, a large gap exists between the invention and utility of applicable software tools.

We believe that the current infrastructure for parallel software tool distribution is fundamentally flawed. Computer science researchers do not get credit or financial support for providing robust well-supported tools. The computer vendors are driven by high-margin hardware sales, with software lost in the shuffle. There are rewards for delivering the latest and greatest massively parallel computer, but not necessarily in having the software to make the machine really useful. Furthermore, the small software vendors are limited by having only a small HPCC market on which to focus.

The result is a gap between the research arena and the marketplace. Therefore, one of our key suggestions is to facilitate technology transfer by providing or strengthening a supporting infrastructure. Several ways exist to do this. The current approach supported by the HPCC program is to encourage interdisciplinary teams; this activity should be expanded. In addition, there must be a mechanism for

Computer scientists must be rewarded for delivering the software that makes state-of-the-art computers usable.

making tools widely available. The National Software Exchange is correct in spirit, but it must be an active agent rather than a passive one. Moreover, it should provide adequate mechanisms for storing codes that have to be supported and updated (tools) as well as codes that do not.

Impacts

Researchers have begun to think in terms of very large problem sizes and datasets because of HPCC. In software tools development, researchers are attacking problems inherent to large concurrent systems.

Near- and Long-Term Impact. Currently, individual application groups are maintaining their own communications libraries and have been layering their own environment-specific tools on top of them. Duplication of effort abounds. Effective basic software tools will allow higher-level software to be developed and will expedite applications research.

In the long term, effective software tools will facilitate the development of libraries serving as a solid base for technology transfer. Although not sufficient, this is necessary for technology transfer outside of the HPCC program.

Applications' Impact on Computer Science. The computer science community believes that there has been a shift in the emphasis of research funding. Fundamental research has lost ground to directly applicable research in today's zero-sum funding environment. The question arises as to whether we really understand the long-range impact of an implicit reduction of funding for basic research.

Industry's Impact on Applications. Vendors are driven to deliver hardware before the software is usable. This situation results in unstable platforms, which in turn have a negative impact on applications. Tool and applications developers spend an inordinate amount of time and effort overcoming the hurdles presented by an immature system.

*A stronger effort
is needed to
transfer research
results to both
computer vendors
and users.*

Applications' Impact on Industry. Industry will be interested in high-performance computers only when interesting applications are possible on them. Vendors have already used mature applications to market vector supercomputers. It is reasonable to assume that applications being developed now will be used in a similar and effective manner.

Findings

Our suggestions for improving HPCC software tools fall into two broad groups. The first group is concerned with infrastructure and technology transfer mechanisms; the second group consists of suggestions for specific tools.

Technology Transfer Infrastructure and Mechanisms. A theme throughout this meeting was the need to enhance the infrastructure and mechanisms to transfer technology from the computer scientist to the applications programmer. A key component of this is software sharing. For a truly effective software sharing system, it must have the following features:

Working Groups

- A software-sharing library must be funded to include people who maintain and support the software in the library. Note that the HPCC Software Exchange as proposed does not provide such a solution.
- A parallel software tools electronic journal/newsletter (complete with editorial board) should be developed to include codes, tool announcements, and publicity. The newsletter needs to be widely distributed within the community.
- Canned demonstrations of tools that are ready for use outside of their development community should be available in software-sharing libraries, so that application users can evaluate tools quickly.

A great deal of discussion focused on the need for a smoother technology-transfer from research to the computer vendors as well as to the users. A "bridge" process is needed and should be funded by HPCC. It is critical that some organization take appropriate research community tools and develop them into robust prototypes, providing support, maintenance, and documentation. Two possible options for the "bridge organization" are to

- fund and expand the charge of an existing organization such as OSF, the national laboratories, or the NSF supercomputer centers; or
- set up a new independent organization or industrial consortium.

The vendors must be strongly encouraged to build and support tools—especially stable and adequate tools which would serve as a basis for effective high-level tools. We present two specific suggestions to facilitate this increased support:

Vendors must be strongly encouraged to provide adequate tools.

- Word RFPs to require computer vendors to provide high-performance compilers, and usable tools for MPP machines. Good tools should be made a requirement for the vendors, not an option.
- Encourage the commercialization of tools that have had success in the research community.

Other ideas to improve the availability and quality of high-performance software tools include the following:

- Establish or fund existing "software capitalization" programs to foster the development of research ideas and prototypes into production tools.
- Fund a "Grand Challenge in Software Tools." This would recognize that basic tools for MPPs are needed across all applications. The whole life cycle of tools is a difficult process and should be recognized as a Grand Challenge in and of itself.
- Fund a Science and Technology Center for HPCC Software Tools.
- Change the reward system for tools. For example, HPCC could sponsor a tools competition that would have prestigious awards analogous to the Gordon Bell prize. Applications users would be the judges, and the tool builders would have an additional incentive to work on robust, usable tools.

Group E - Building Multidisciplinary Applications

Software Tool Development. The utility of advanced software tools is seriously undermined by the poor state of the basic software tools on MPP computers. This includes node compilers, operating systems, and basic communication routines. Any suggestions in this section are moot unless these issues are addressed.

We have identified both short-term and long-term needs for HPCC software tools. Therefore, we make the following suggestions:

- Elevate the quality and performance of the basic HPCC tools such as node compilers, debuggers, and profilers.
- Encourage standards efforts where appropriate. For example, the efforts of the Message Passing Interface forum will address an important and immediate need for enhanced software portability.
- Fully fund heterogeneous computing in all of its aspects including application heterogeneity, resource management, data management, network management, and matching of tasks to machines. Some of this research needs to be treated as "big science," with mechanisms for large projects that include software, applications, and network research.
- Fully fund parallel debugger efforts. Such efforts must be funded for all phases of the life cycle—fundamental research, prototyping, and development of robust tools and standard debugger interfaces. MIMD and metacomputer debuggers need to be emphasized.
- Develop a compiler infrastructure for making visible the compiler's internal representations and transformations to other software tools.

Finally, further research is required to develop distributed data management tools; parallel application libraries; data abstractions for parallel architectures; easily layered, object-oriented data abstractions for parallel architectures; multimedia tools for parallel systems; and operating systems for parallel architectures.

Group E - Building Multidisciplinary Applications

Chair: Ian Foster; **Co-chair:** Tom Zang

This working group addressed the systems integrations issues associated with "multidisciplinary applications," in which either several different "disciplines" are coupled in a single modeling system (e.g., fluids and structures in an aeronautics code) or several different "levels" of analysis are combined within a single "discipline" (e.g., linear, Euler, and Navier-Stokes aerodynamics). Issues concerning the internals of individual discipline modules were not considered germane to the group.

Status

Various applications were represented in the working group, including aeronautics, geophysics (environmental and global climate modeling), biological systems, and plasma physics. Chemistry, materials, astrophysics, and quantum chromodynamics were not represented.

There was a surprising commonality among the different fields in terms of both status and obstacles to progress. In both aeronautics and geophysics, integrated

Many modern simulation projects could benefit greatly from the use of integrated codes such as the earth system models.

codes have been operational on Cray supercomputers for several years. Typically, these are limited to relatively simplified linkages among a few components, with a strong fluids component. Codes have also been developed in heterogeneous, networked environments. Examples of such integrated codes include the following:

- Simulation of aircraft, both a high-performance aircraft and a high-speed civil transport together with their propulsion systems, incorporating fluid dynamics, structural mechanics, surface heating, and controls models.
- Environmental models incorporating atmospheric dynamics, surface water, and groundwater models.
- Earth system models incorporating atmospheric dynamics, atmospheric chemistry, biosphere, and ocean dynamics models.
- Predictive computational models of cellular organelles, coupling fluid dynamics, rigid body mechanics, molecular dynamics models, etc.

A common data model would capture data from individual disciplines in a unified framework.

In all these areas, there is strong interest in developing increasingly sophisticated systems that couple ever more advanced simulations of more diverse physical systems. However, progress is limited by two factors: computational limitations of current supercomputers, and limited understanding of the physics and numerics of the interfaces between components, especially with respect to stability. MFP computers are seen as a potential solution to the first problem, but progress in this area will necessarily increase the pressure on applications scientists to improve understanding of coupling issues.

Multidisciplinary applications on MPP computers are for the most part underdeveloped, primarily because of the difficulties inherent in adapting not just one but a suite of existing codes for efficient parallel execution. This is a severe problem. Indeed, many of the group members felt that this may well be the pacing item in demonstrations of HPCCP technology for real-world Grand Challenge problems.

Targets of Opportunity

The consensus was that opportunities exist for enhanced activities in four areas: common data models and database systems, languages and tools for developing multidisciplinary applications, application design optimization, and education.

Data Models and Database Systems. Data management is very much an enabling technology for multidisciplinary design and analysis. The current state of the art involves a series of custom-designed data "translators" that map information between various applications. These are expensive to generate and require extensive validation. The opportunity for errors entering the multidisciplinary design process are greatly amplified across each application interface.

A preferred data management approach would involve a "common data model" representation. This common model would capture individual discipline data fields within a unified representation that provides consistent information at the boundaries of each discipline. Disciplinary data is "owned" by each discipline and available as limited subsets of the unified representation. This information can be distributed across a variety of computing platforms. Data access requirements, as determined

by the "physics" of discipline interfaces, define data protocols. Where it is appropriate, object-oriented data hierarchies are used to organize and minimize data storage while providing linkages into discipline applications. This approach is facilitated by the use of geometry modeling standards and data standards. For example, in aeronautics, the Non-Uniform Rational B-Splines standard (as embodied in the imminent NASA-IGES formal standard) enables linkages between CAD systems, grid generators, visualization tools, and application codes. Finally, the database manager must organize information across a variety of hardware platforms while maintaining "virtual" ownership on perhaps a single workstation. Efficient access to a heterogeneous mixture of computing platforms (MPP to workstation) is needed.

In summary, key points in this area are

- use of a common data model among disciplines,
- use of an object-oriented data hierarchy to organize data,
- standard geometry and data representations, and
- platform independence.

Languages and Tools. Better languages and tools are urgently required to support development of multidisciplinary applications. Currently, the vast majority of component programs are written in Fortran 77. Coupling is achieved in a variety of ad hoc ways, including (a) creation of a single executable, (b) files, (c) specialized shared-memory database routines, (d) message-passing in a network, (e) AVS, and (f) intercube communication on a hypercube. In the future, an increasing number of components will be written in C, C++, and Fortran 90.

There was a strong consensus within the working group that languages were urgently required that addressed issues of encapsulation, modularity, and portability in MPP and network environments. Current message-passing systems are seen as too clumsy, and either specialized languages or graphical programming systems are required. The need to support integration of both task- and data-parallel modules was emphasized, as was the need for tools that permit MPPs to be partitioned among disciplines, with simple and efficient communication between partitions.

There was considerable debate within the working group as to the importance of object-oriented programming techniques for multidisciplinary applications. Several argued that the abstraction capabilities of C++ were an important technology that could be used to structure complex multidisciplinary applications codes and encourage code reuse; others disagreed. One member argued for object-oriented extensions to Fortran 90 that supported inheritance, by means of a preprocessor.

Two major topics of discussion were language interoperability and standards. For example, it is increasingly common for software developers to use C++ wrappers to call Fortran 77 or Fortran 90 routines. A standard procedural C++/Fortran interface would be very helpful. The reverse interface—from Fortran 90 to C++—would also be useful for database access and graphics libraries. Standard message-passing systems were also seen as important.

In summary, key points in this area are the need for

- languages and tools for systems integration aspects of multidisciplinary applications on MPP and networked computers,

*Task-parallel
and data-parallel
modules should
be integrated.*

- languages and tools for integrating task and data parallelism, and
- standard interfaces between different languages.

Input from industry is needed about ways to improve design optimization methodologies.

Applications Design Optimization. Multidisciplinary applications in both Grand Challenge projects and industry possess significant complexity. They may involve hundreds or thousands of design variables and tens or hundreds of modules. In manufacturing, the move towards concurrent engineering introduces many other considerations, since manufacturing or marketing considerations may not be describable by continuum models.

A strong need exists for improved optimization methodologies and software support for such complex problems. The working group did not feel that its members represented a sufficiently wide range of expertise to make detailed suggestions in this area. Input from industry, universities, and government labs is urgently required to define goals in this area, and this input should be taken into account in defining HPCC program directions.

Relationships

Multidisciplinary approaches are likely to become dominant in more fields of science and engineering.

Communication difficulties, both between applications scientists and computer scientists and between application scientists working in different disciplines, were seen as hindering progress in multidisciplinary applications. In addition, several speakers alluded to lack of experience in modern software engineering techniques among scientists developing components of multidisciplinary models. It was also felt that new graduates, while often more exposed to new techniques, lack experience in multidisciplinary research and industrial software engineering problems. Several solutions were discussed, including (a) an increased emphasis on multidisciplinary degree programs (Georgia Tech and Virginia Polytechnic Institute were cited as examples of successful programs), (b) improving training in software engineering for science majors, and (c) programs that involve students in software development projects at labs and companies.

The working group expressed concern about the apparent lack of interest among computer scientists in the problems facing developers of multidisciplinary applications. The current focus on getting single-discipline modules running on MPP computers is understandable and justified. However, multidisciplinary approaches are likely to become dominant in most fields of science and engineering, and better tools are urgently required. This situation in turn requires increased dialogue between computer scientists and applications developers. The working group emphasizes that good tools will be developed only in a multidisciplinary technical environment, in which groups having highly diverse combinations of computer scientists, applications developers, etc., tackle the needed research in this area.

Templates are the key to reuse of parallel code.

The working group was also concerned with the difficulties facing multidisciplinary applications developers who wish to learn about and evaluate new languages and tools. Currently, information is primarily disseminated by word of mouth; better mechanisms are urgently needed. One approach would be to prepare a set of "template" codes representative of multidisciplinary applications concerns and to encourage software tool developers to use these templates when evaluating their systems.

Impacts

The working group agreed that it is too early to expect the activities of the HPCC program to have had significant impact on science and industry. However, there was broad consensus that by allowing simulation of entire systems rather than individual components, the combination of high-performance MPP computers and improved techniques for coupling different disciplines can be expected to enable significant progress in many areas of science and engineering. We give just three examples of where such "whole system" simulations are planned; each is dependent on availability of the greatly increased computational capabilities that are being developed in the HPCC program.

- The aerospace industry is implementing integrated product/process development (IPPD). A key element of IPPD is the development of total system simulations that allow the substitution of computational prototypes for physical prototypes. This approach could substantially reduce design cycle time and development cost, and improve product quality.
- In the environmental sciences, integrated earth system models could greatly improve the timeliness and scientific foundation of data, motivating important environmental decisions such as mitigation strategies for global change.
- In biophysics, models that integrate multiple levels of knowledge about protein structure would allow new approaches to the problem of predicting the three-dimensional shapes adopted by complex proteins. Solution of this protein-folding problem would have great significance, both for science and for drug design and manufacture.

The availability of high-quality numerical algorithms and libraries can determine the pace of applications development.

Findings

The working group reached consensus on the following suggestions, which are given equal priority.

1. Steps should be taken to strengthen connections between multidisciplinary applications and database communities. New research in this area should be initiated if necessary technology is not already available.
2. The HPCC program should encourage both research on languages and tools for the systems integration aspects of multidisciplinary applications, and the timely dissemination of the results of this research to the multidisciplinary applications community. In particular,
 - languages and tools for coordinating execution of sequential, parallel, and data-parallel components on MPP and networked computers should be made a research priority;
 - a framework should be established for the evaluation of such tools in multidisciplinary applications; and
 - standard interfaces should be defined between programming languages commonly used for development of multidisciplinary components (e.g., Fortran 90 and C++).

3. Industry and university input should be sought to define HPCC program directions in applications design optimization.
4. A program should be established for retraining current scientists and engineers in critical HPCC multidisciplinary technologies, and the university community should be encouraged to establish or strengthen multidisciplinary programs and curricula.

In addition, several group members suggested that researchers working on load-balancing techniques, visualization tools, and parallel debuggers be encouraged to consider the particular requirements of multidisciplinary applications.

Group F – Algorithms and Libraries I

Chair: Ahmed Sameh; Co-chair: Phillip Colella

The lack of high-quality numerical and symbolic algorithms and libraries often limits the pace of applications development. This situation is particularly true on massively parallel machines. Group F considered the state of the art in algorithms, libraries, and software for the numerical solution of partial differential equations that arise in various applications, and the associated linear algebra algorithms and kernels. The majority of the members of this break-out session were application scientists who are not only experts in their applications domains but also seasoned computational scientists quite familiar with computer architecture and system software, as well as with the design and implementation of numerical and nonnumerical algorithms. Furthermore, almost all of them are associated with interdisciplinary activities that involve computer scientists and applied mathematicians. In spite of this, our deliberations considered the needs of applications scientists who do not have the benefit of such expertise or the benefit of the involvement of the computer science and the applied mathematics communities.

Flexible parallel languages will allow users to match their algorithms to a specific architecture.

Status

The working group focused on three areas: research activities, software-sharing mechanisms, and federal support.

Major Research Activities. Grand Challenge teams assembled at national laboratories, universities, and a few industries represent a small number of computational scientists who are sophisticated users of advanced computer architectures and who are willing to work hard at implementations that result in high performance. While the HPCC program is not limited to parallel computing, the design of parallel algorithms for a variety of applications is a vital component of the program.

For some application domains, parallel algorithms can be provided by generic mathematical software libraries that are finely tuned for a few architectures and that contain some of the basic algorithms for FFTs, linear algebra kernels, and algorithms for dense and sparse matrix computations. While such libraries may be sufficient for some applications scientists, however, they are far from being satisfactory for sophisticated users or for users seeking high performance. Factors that limit the usability of such generic libraries include limited functionality, inappropriate data structures, and lack of algorithm scalability. This situation shows

the need for parallel languages that allow sophisticated users to match their algorithms to the architecture at hand, namely, by selecting appropriate data structures and by matching data movements to architectures. For less sophisticated users, it is advantageous to incorporate such features in the compiler as well. In addition to the need for parallel languages with the above flexibility, the group pointed out the need for (i) application-specific libraries that may include mesh generation tools, discretization tools, solution and update tools, and mesh refinement and load-rebalancing tools, and (ii) application-specific languages such as FIDIL for computational fluid dynamics. Applications scientists who are reluctant or unable to make the investment required to create large-scale applications codes that achieve the highest performance possible for each new architecture that comes along must have available generic or application-specific libraries of value for rapidly producing prototype codes that test or verify computational models. This rapid prototyping may then be followed by incremental refinement to enhance performance for a particular architecture. Such refinements should entail further algorithm/architecture matching, including parallel communication utilities for changing data structures and implementing other common communication tasks.

Software-Sharing Mechanisms. Software sharing occurs mainly between co-operating groups. Application scientists are reluctant to exchange software for two major reasons:

- The software is experimental and often poorly documented. If such software is not used properly by others, it may reflect badly on the investigator.
- Distribution and maintenance of large-scale application codes is a very expensive activity. Researchers at universities and national laboratories are not funded explicitly for providing high-quality sharable software that can be used by others or for providing consultation on its use.

One approach is to fund activities at some institutions where application scientists are working closely enough with systems software groups that are capable of supporting and maintaining the resulting software for a much wider community. Software that has a large enough market should be transferred to an interested industrial partner.

Sharing mechanisms may not be limited only to complete software packages; templates (or pseudo-code representations) of general algorithms that are language- and architecture-independent may be of great value as well.

Federal Support. The federal HFCC program has concentrated mainly on Grand Challenge problems and teams to solve them. Concern has been expressed that support for the underlying software infrastructure may be inadequate. By software infrastructure, the group means (i) system software for MOP's (i.e., languages, compilers, and communication primitives); (ii) research in new application-specific algorithms and the resulting software on a variety of architectures; and (iii) application-specific languages. Also, as mentioned above, additional funding is needed for supporting the maintenance, distribution, and consulting activities for applications software of value to a large group of users in the national laboratories and universities. Industry involvement is highly desirable in the distribution of higher-quality versions of such software.

*Software with
a large market
should be
transferred to an
industrial partner.*

Targets of Opportunity

Several areas of research need to be strengthened, while others need to be initiated:

- Algorithm and software developments in application-specific domains, including parallel computation and communication algorithms.
- System software that allows easier and more efficient use of parallel architectures. Such software should include language notation for data mapping, good compilers, and effective debugging tools.
- Application-specific languages that increase the level of abstraction sufficiently so as to respond to architectural changes quickly and, if this language can be compiled efficiently, without loss of performance.

Computer vendors can help interdisciplinary teams.

Relationships

The group believes that the HPCC initiative deserves credit in fostering interaction between the computer science community and the applications scientists. While most members of this break-out session belong to both communities, roughly half expressed the need for more cooperation between these two communities in their own institutions. The HPCC program has started to change the culture in several science and engineering disciplines by demonstrating that computation is just as important as theory and experiments in advancing these fields.

The most important barrier to interaction between computer and applications scientists is the cost of entry. Most computer scientists have no background in any of the classical science and engineering disciplines, and the majority of engineers and scientists are not aware of the most advanced parallel computer architectures or state-of-the-art algorithms for these computers. In fact, it has been demonstrated that application experts can more successfully bridge this gap than computer scientists. One approach is to prepare graduate students in the interdisciplinary field of computational science and engineering, which is the interface between computer science and the classical disciplines of science and engineering. Resistance to establishing such new programs in various universities may be overcome by federal funding of the most promising programs. Also, computational scientists (who are amply represented in the Grand Challenge teams at national laboratories) need to gain the confidence of their experimental colleagues by seeking experimental validation of their computational results.

Interaction with industry (including computer manufacturers) is vital to the goals of the HPCC initiative. It is through industrial interaction that the knowledge from the Grand Challenge teams can be disseminated. Members of the group also pointed that computer manufacturers can be of help to interdisciplinary teams involved in the research activities outlined above by providing access to dedicated machines and assistance in implementing Grand Challenge software on their machines. Access to dedicated machines is essential, as the development of finely tuned algorithms and whole applications codes require performance-monitoring and measurement tools that are intrusive in multiuser environments. Expansion of existing approaches, or development of new ones, for working with industry is vital if the HPCC program is to have an impact on industrial competitiveness. With computer manufacturers, these approaches could facilitate more effective computer architectures and systems software, as well as the development of applications software and

Staff exchange programs should be endorsed.

underlying software libraries on new machines. Collaborations with other industries will facilitate technology transfer to those industries that can substantially benefit from the technology produced in the HPCC program. Such technology transfer can be significantly enhanced through exchange of researchers among universities, national laboratories, computer manufacturers, and industry, in general.

Impacts

The group identified several ways in which the HPCC program has had a major impact on algorithm and library development:

- The HPCC program has provided a mechanism for funding state-of-the-art applications software. Industry involvement in this program needs to be strengthened.
- The program has dramatically increased the number of scientists and computational scientists with hands-on experience with parallel computers. This represents an advantage to U.S. research and development similar to that provided by early experience with compilers and operating systems several decades ago. It is imperative, however, that continued and increased support be provided for educational programs in computational science and engineering so as to ensure a permanent supply of researchers who combine both application and computer science expertise.
- Parallel algorithms have been beneficial to most of the Grand Challenge applications represented in this session. In fact, several application scientists observed that developing parallel algorithms for their applications often leads to faster execution on uniprocessors. Since massively parallel systems are not mature, however, more time and resources are needed before the impact of parallel algorithms on applications is fully realized on such architectures.
- Grand Challenge problems have led to increased understanding and use of broad-band communication structures and large distributed databases.

Findings

With respect to technical issues, we suggest that application-specific systems software and libraries be developed so as to provide a stable platform on which Grand Challenge applications can be built. Such platforms should address

- the development and implementation of algorithms in support of applications, including parallel communication algorithms;
- high-level language support for data mapping and data communication; and
- increasing programming productivity, for example, by application-specific languages.

With respect to institutional issues, we make four suggestions:

- enhance and expand technology transfer through scientists exchange and extended collaborative efforts on specific end-user applications;

Applications-specific libraries must provide a stable platform for Grand Challenge applications.

- promote interagency and intermission activities in application-specific algorithms (e.g., PDE algorithms across several disciplines);
- redress the HPCC program imbalance: specifically, fund manufacturer, applications, and systems collaborations for developing software infrastructure; and
- support educational programs in computational science and engineering or in scientific computing.

Group G – Algorithms and Libraries II

Chair: Fran Sullivan; **Co-chair:** Bob Sugar

The development and implementation of algorithms for massively parallel computers are important components of the HPCC program and should receive greater emphasis than they currently do. This is an area in which collaborations between computer scientists and applications scientists can yield major dividends. Such dividends have already begun to be realized within the Grand Challenge applications groups. As indicated below, however, we believe that there are a variety of problems on which new collaborative efforts are needed. New approaches are also needed for sharing algorithmic ideas and software. In addition, vendors must pay greater attention to the needs of the applications teams if the use of massively parallel machines is to progress.

Status

Grand Challenge applications groups are developing algorithms for specific machines and writing optimized code to implement them. A number of groups are working on closely related problems and might benefit from greater communication.

Existing software-sharing mechanisms are not working well and not rapidly enough to be of real use to the individual Grand Challenge participants. There was a strong consensus that conventional libraries were not at all appropriate to the rapidly evolving needs of the Grand Challenge community. Codes are developed by the teams themselves, and the teams cannot find time to produce well-documented, distribution-quality programs. The usual vendors such as IMSL cannot contemplate entry into a very thin market with fast-evolving algorithms. The strong emphasis on code optimization for individual machines makes portable libraries impractical at this time.

Modern simulations often require new algorithms that have not been extensively used in scientific computing.

Targets of Opportunity

In the context of the HPCC program and its associated Grand Challenge applications, the following specific areas of research were identified:

- Algorithms for decomposition of regular and irregular data structures arising in PDE discretization to obtain efficient load balancing, and fast algorithms for solving elliptic PDE on these types of structures. Of particular need are efficient methods that can be used with adaptive methods where these types of load-balancing issues must be performed dynamically as the discretization adapts to the solution structure.

- Parallel random number generators with appropriate speed and correlation properties. Correlation of random numbers across processors has also been a major source of concern. Software to test these programs is also needed. Randomized algorithms are a major technique in algorithms research and often are the only effective way to overcome NP-hard problems. For such algorithms, the availability of parallel random number generators is fundamental.
- Symbolic and combinatorial (as opposed to floating point) algorithms. Symbolic algorithms are currently useful in a variety of local computations within Grand Challenge problems. We expect combinatorial algorithms, numeric (but not necessarily floating-point) computations, and other symbolic methods to gain in importance in such fields as biotechnology in the coming years.
- New algorithms for irregular problems. Sparse systems and irregular patterns of data are not being addressed by the algorithms that are currently available. In order for these systems to be useful, good parallel algorithms are needed to handle these structures.
- General multidimensional pattern recognition and pattern-matching algorithms. These will provide informational structures to analyze informational structures in higher-dimensional spaces, which are useful, for example, in three-dimensional molecular structure databases.

Robust, correct implementations of these algorithms would be extremely useful and, on the whole, are not available. However, work on the algorithms themselves is as important as work on the implementations.

The HPCC should provide strong encouragement and incentives to vendors to develop robust software environments for their MPP hardware (concurrent with hardware delivery). Lack of this support has significantly hampered algorithm development and implementation on these machines.

Relationships

Although most researchers have the feeling that the Grand Challenge community (GCC) is not reinventing techniques already known to the computer science community (CSC), the true situation is the opposite: the GCC is, in fact, duplicating work several times over. Two factors lead to this situation. First, applications scientists believe that they require specialized algorithms for each application, so that general methods cannot be used. Second, and more distressing, large segments of the CSC have decided that their discipline can and should be isolated from the messy, real-world problems of Grand Challenges. While the GCC is making an effort to learn what is already known, parochial issues from both sides hinder such efforts. The recent NRC report "Computing the Future" strongly suggests that computer science should, to a much greater extent than is now the case, derive inspiration from real-world applications such as the Grand Challenges. We repeat that suggestion and strongly support the collaborative interdisciplinary requirements of the HPCC Grand Challenge program. These have the salutary effects of allowing the GCC to become aware of the work of computer scientists and of bringing that additional talent in computation to bear on Grand Challenge problems.

*It is imperative
that the Grand
Challenge
community not
reinvent computer
science techniques.*

We note with caution that there are significant problems in the universities for junior faculty in computer science who choose to work on applications. It is not always possible for them to be rewarded for doing "computer science" for such work. Part of this is due to the departmental structure of the universities—a problem not generally shared by the laboratories. Part is due to a tendency of algorithmists in computer science to work on theoretically elegant rather than real-world problems. We approve of current efforts to remediate the problem. There have been several multidisciplinary programs recently begun in computational science at several universities, and there is a small but growing fraction of the CSC working with Grand Challenge-like problems. We encourage increased interactions between the national and industrial labs and the faculty and graduate students in computer science. Another useful activity would be to organize open research conferences on topics related to Grand Challenge problems.

Impacts

A short-term impact of the HPCC program has been in providing massively parallel computers to Grand Challenge research teams. This has had the effect of promoting parallel computing as an industry. In addition, the Grand Challenge teams are quite sophisticated and have been able to affect the usability of these computer systems. By striving for performance on immature systems, they have uncovered software and hardware inadequacies in these systems, and vendors have used this information to improve their systems. Thus the research teams benefit the entire community by informing the vendors of mistakes and pitfalls at early stages of the development cycle. One hopes that the teams will also encourage the vendors to provide a stable and understandable set of development and optimization tools. Such tools, in turn, would smooth the way for the rest of the scientific computing community as they try to enter the massively parallel computing field.

Another important impact has been on high-technology industries. It is interesting to note that workstation vendors are designing multiprocessor machines. IBM, Sun, HP, and SGI have all started to provide high-end multiprocessor workstations. In addition, some other industries have begun to use parallel computing in their own research and development. These include the oil industry, where multiprocessors are used for exploration and analysis of seismic data; the aerospace industry, in collaboration with NASA, for the design of new aircraft; and the materials industry, in collaboration with DOE, for the computational analysis of materials properties. It is hoped that the continued success of these efforts will lead to the desired long-term impact of HPCC: improved industrial competitiveness.

Another objective of HPCC is to foster collaboration between the computer and application science communities. To date, increased collaboration has occurred among members of Grand Challenge teams, but the impact of HPCC directly on computer science has been small. There does seem to be a trend in computer science towards the experimental, and the theoretical community is considering more realistic models of computation than the PRAM. Nevertheless, the computer science community is still perceived as too theoretical for the immediate needs of HPCC.

Finally, while most of the participants believe that MPPs are the future trend, they noted that some short-term goals of the Grand Challenge projects still require the more mature technology of vector machines. Some researchers feel that it is too

HPCC research provides valuable feedback to vendors early in the design cycle.

The trend is clearly toward more realistic models of computation.

soon to assume that the continued health of the computer industry will come only from massive parallelism.

Findings

To facilitate work on software development, vendors should

1. balance memory bandwidth with CPU speed;
2. balance communication speed with CPU speed;
3. provide good node compilers to allow programmers to make more effective use of machines without resorting to assembly language coding;
4. provide fast multiple-precision arithmetic (the predominant techniques are still numeric and greater size of problems demands greater precision. It should be noted that some participants disagreed on the importance of this suggestion, while others find multiple-precision arithmetic essential to doing serious work on these machines);
5. pay greater attention to symbolic and combinatorial aspects of computing, such as fixed-point arithmetic;
6. provide language constructs that allow the easy definition of complex data structures (C++ is an example of a high level language that allows this);
7. provide high-level standardized message passing routines; and
8. provide a random number generator with their machines.

*Informal libraries
should be set up
at the national
supercomputer
centers.*

The problem of sharing of information among scientists is very important. However, we do not think that the massively parallel field is mature enough yet to warrant portable standardized libraries. Portable libraries would not be efficient enough to be used widely and so are not reasonable at the moment. However, we strongly suggest establishing some system of informal libraries with a different set for each machine class. These informal libraries would include algorithms, templates, and code. The library servers could be set up at different servers in different places (perhaps for different machines). The national supercomputer centers would be logical choices for locating such servers.

Finally, we suggest that the current practice of requiring computer scientists on Grand Challenge teams be continued. This practice has stimulated interactions between these groups. Efforts are also needed to increase interactions among scientists in the academic, laboratory, and industrial communities. Although the HPCC has helped to bring the first two closer together, greater efforts are needed to increase the participation of the industrial community. This is essential if the program is to succeed in its important goal of improving U.S. competitiveness.

Group H - Graphics and Visualization

Chair: Paul Woodward; Co-chair: Tom DeFanti

Status

Graphics and other visualizations are crucial to modern scientific applications.

The scientific visualization working group reviewed the current status of software and hardware development activities designed to support the high-performance computing community. One of the principal trends in this area is the development of specialized systems and environments for the interactive exploration of very large datasets. Grand Challenge computations are often characterized by either the use of or the generation of datasets of a terabyte or more. These datasets are so large that visual representations are often the only means available to examine the data. Even with image representations, the datasets are still so large that many views must be generated. Several groups are engaged in developing tools that enable a researcher to explore such vast datasets, bringing into the computer or workstation memory only small subsets of the data, which are specified interactively by the researcher, and then rendering them in any of a number of ways in real time. Developing such data exploration hardware and software systems places extreme demands upon the computer network, I/O subsystems, image-rendering engines, and even the display subsystem if the data is to be displayed at full resolution.

The working group noted efforts at several centers to develop *image-rendering software* that can be executed on massively parallel processing systems. The need for this software is driven by the immense size of the datasets to be visualized. Often, the dataset can be mounted and dealt with only on the MPP system that created it. Even when other systems can access the dataset over the network, rendering of images on the MPP may be required if only this system has sufficient rendering speed to produce an image of the data in a reasonable time.

The development of software to support scientific visualization in *distributed, heterogeneous computing environments* is proceeding rapidly. This activity is vital if visualization requirements of researchers remote from the Grand Challenge computing hardware are to be met. However, the working group noted that much of this effort is currently aimed at visualization of significantly much smaller datasets than are common in the Grand Challenge applications community. Special efforts of this sort targeted at support for Grand Challenge applications are needed in the future.

Visualization can produce comprehensible views of real systems such as automobiles and airplanes.

The working group also discussed research activities in developing *virtual reality* environments in relation to the Grand Challenge applications. Both immersive displays and large flat-screen displays were discussed. The principal advantage of these efforts for Grand Challenge applications was agreed to be their potential help in dealing with the visualizations of extremely complex systems. Both approaches offer high-resolution displays that can, if the accompanying I/O and data-rendering systems are large and fast enough, put onto a single display much more than the small window of the workstation screen to which researchers have become accustomed. In either case a familiar mechanism is supported for navigating through the dataset, that of walking for the immersive systems, and that of driving (or flying) for the large-screen systems.

A brief discussion of fundamental research in visualization techniques noted that the NSF Science and Technology Center for Computer Graphics and Scientific

Visualization is developing new rendering techniques that are more soundly based upon the physics of *radiative transfer and scattering*. These new techniques should be capable of producing much more photorealistic images of real systems, such as automobiles and airplanes. The usefulness of these techniques in enhancing the perception of physical systems that cannot normally be seen by the eye, such as complex molecules or the structure of the earth's magnetosphere, remains to be seen. Presumably, success will depend upon the extent to which familiar visual cues, such as surface textures, can be employed to convey significant scientific data in these unfamiliar settings.

Use of scientific visualization for the *remote steering* of instruments such as microscopes and telescopes was also noted as an important current activity. Remote steering of supercomputer simulations, although possible using visualization techniques to develop the user interface, is not yet common in this community because of the enormous amount of computer time required for a single Grand Challenge run.

Finally, the working group noted the tremendous usefulness of scientific visualization in *education, training, and technology transfer*. Not only are visualizations of complex calculations indispensable in explaining the science involved to nonexperts, but also visual user interfaces to the associated simulation software serve to reduce barriers to use of the software by nonexperts.

Software-Sharing Mechanisms. Present software-sharing mechanisms consist of FTP servers at various centers and of informal software sharing among collaborators. These mechanisms are effective but inadequate. Clearly, some mechanism is needed for locating software that is available over the network. In addition, more incentives to share software are needed. There is a reluctance to share software among competing research groups and centers, which is in part a result of the system that sets these groups up as competitors rather than as collaborators. Direct and visible rewards for the sharing of software could change this situation. This problem is not universal in the community, but it needs to be addressed by the funding agencies, who are the only members of the community with a means to change the present system of incentives.

Balance of HPCC Funding. The working group concluded that present HPCC funding is too strongly skewed toward the purchase of MPP hardware and that more support is needed for the disk and I/O subsystems, networks, software tool development, and people who are required to make these MPP systems useful in addressing Grand Challenge applications. As developments in compilers and application programs allow us to extract more and more of the potential performance from these MPP systems, this support issue becomes increasingly critical.

*Innovative
graphics include
volume rendering
and virtual reality.*

Targets of Opportunity

The primary system needs of HPCC scientific visualization that computer system vendors should address are as follows:

- higher I/O bandwidth,
- effective interactive operating systems,

- high-speed network interfaces, and
- better compiler and debugging tools.

The working group identified several areas that need enhanced federal funding support:

- development of scalable parallel visualization algorithms,
- tools and environments for interactive exploration of massive datasets,
- development of new visualization/analysis paradigms,
- graduate student support for work in this area, and
- conferences and workshops that bring this interdisciplinary community together.

Relationships

The working group felt that not enough computer science research in scientific visualization is addressing Grand Challenge applications. These applications require visualization tools with

- embedded quantitative analysis capabilities, such as the generation of individual numbers, various statistical measures of the data, spectra, and other operations that may be application specific;
- support for complex data types and structures, such as those that arise from overlapping, adaptively refined, or unstructured grids, staggered grids, and binary tree structures;
- high efficiency for use with large datasets.

To improve collaborations between computational and computer scientists in this area, the working group felt that programs aimed at interdisciplinary collaborations and workshops could be enhanced. These collaborations would also benefit greatly from better network connectivity and associated network-based collaborative software tools.

Impacts

The impact of scientific visualization upon the HPCC program is chiefly based upon the role of visualization as an enabling technology, which allows a researcher to understand extremely complex physical systems. Not only does visualization enable the primary researcher's understanding of his simulation data, but it also enables him to communicate that understanding to other members of an interdisciplinary community of experts, and, equally important, to the public that ultimately is called upon to support the research. Secondary impacts of scientific visualization on the HPCC program include its role in driving requirements for network bandwidth and for the interactive use of MPP computer systems.

The impact of the HPCC program on scientific visualization research is chiefly to drive the development of scalable visualization algorithms. Related to this is the development of multidisciplinary tools for the visualization of massive datasets.

An easy-to-use, interactive tool to guide large-scale computations at run time will dramatically improve scientists' productivity.

The impact of all these developments on industry is to reduce the barriers to the use of HPCC application codes in industry. Graphical user interfaces are perhaps the most effective means of promoting the use of new and complex software packages by scientists not trained in the details of the inner workings of those codes. A common difference between commercial and research code packages is the lack of flexible and powerful graphical user interfaces in the latter codes, for handling both data fed into a simulation model and data produced as code output. With accelerated development of modular visualization software targeted at Grand Challenge applications, the development of such user interfaces will represent a much smaller fraction of the overall code development effort and therefore make technology transfer to industry both easier and more common.

Findings

The visualization working group focused on the following findings:

- **Implementation of visualization techniques on massively parallel computers.** Most Grand Challenge computations will be performed on powerful parallel computing platforms. In many cases, only these systems will have enough memory, computing power, and peripheral storage to visualize the simulation data. Therefore, visualization algorithms must be developed that run efficiently on such systems. These visualization programs must be able to run in a highly interactive mode, in order to permit researchers to explore datasets dynamically. Visualization techniques of particular interest to the Grand Challenge applications community for implementation on MPP systems include three-dimensional perspective volume rendering, polygonal surface rendering, and combinations of both these techniques. The tremendous computing power offered by MPP systems will also enable more sophisticated, physically based rendering methods that can capture more of the information inherent in the simulation. Such advanced rendering techniques might include, for example, a detailed treatment of light scattering from textured surfaces.
- **Very large datasets.** Visualization packages like AVS and Explorer, which allow many groups to write their own specialized modules and then to incorporate them into a larger, unified whole, are an excellent means of software sharing and reuse. However, a number of flaws in such packages were noted that relate to the essential Grand Challenge need to visualize very large datasets. If visualization software sharing and reuse are to become the rule in the HPCC community, research is needed on modular, high-performance visualization environments that are scalable to handle very large Grand Challenge applications and that are capable of utilizing powerful MPP systems. These environments should allow quantitative information to be extracted from a dataset using the visualization tools, and they should allow a variety of application-specific analysis modules to be incorporated into the visualization environment. These environments should also support a broad range of data types and structures. In particular, effective means of visualizing vector fields in three dimensions are needed, as are mechanisms to handle data specified on overlapping or unstructured meshes or on staggered or adaptively refined grids. Many of the Grand Challenge projects involve interdisciplinary

Many Grand Challenge-type applications are too enormous to be viewed in real time.

collaborations, and the visualization environment must be able to handle simultaneously the preferred data structures for each discipline.

*Complex
computations
require a highly
interactive
response.*

- **Highly interactive response.** The Grand Challenge applications, because they often involve computations on extremely fine grids, demand large data volumes (and very high image resolution). The size of the datasets from these applications can exceed a terabyte, making it impossible to render all the data into any single image. Consequently, researchers often wish to explore their datasets interactively, with only a fraction of the data displayed at any single instant. This drives a need to get as much data onto the display as possible in each image, and hence very high resolution display is essential. In addition, the researcher needs an effective means of navigating through his dataset, keeping at all times a sense of where he is with respect to the whole. Virtual reality systems offer one means of creating a natural sense of this navigation through the data. Flat-screen-based displays allow one to navigate through the data as if in a virtual vehicle. There are tradeoffs in these approaches involving resolution and display speed, as well as other factors, and research in this area is needed. Any such approach, however, has a fundamental need for very rapid image rendering and display under interactive user control, and with a special emphasis on very large datasets. A high degree of parallelism in both the rendering engine and in the I/O subsystem is likely to be a common feature of these interactive visualization systems.
- **Massive data storage, retrieval, and management in a heterogeneous networking environment.** These data storage, I/O, and networking issues have been separated out for detailed consideration by other working groups. Nevertheless, our summary of the needs of high-performance scientific visualization could not be complete without stressing the fundamental dependence of this activity upon the infrastructure, both hardware and software, for data storage and for data movement from one device to another.
- **Increased support of visualization infrastructure.** This infrastructure includes visualization hardware and software both at high-performance computing centers and at the researcher's home institution, both wide area and local networks, and support people both at centers and working locally with researchers. Development of this visualization infrastructure for high-performance computing is a natural follow-on to the nation's successful program of installation of high-performance computing infrastructure. The new HPC computing systems have unprecedented power to perform scientific computations, but unless the capability to visualize the results of these computations is adequately supported, many of the potential scientific discoveries of this program may elude us.

Group I - National HPC Infrastructure

Chair: Charlie Catlett; Co-chair: Maxine Brown

The global economy is rapidly shifting toward information technologies. Advanced information technologies (including supercomputers, new paradigms for immersive data analysis, digital libraries, concurrent engineering, and the ability of

workers to use these technologies) form the basis for economic competitiveness during the next fifty years. The present U.S. infrastructure provides access to goods and services, natural resources, manufacturing centers, and other components of an industrial society. The HPCC infrastructure must provide access to information technologies, goods, and services that will be necessary in order to compete in the new, information-based global economy.

Status

The temptation exists to think of the HPCC infrastructure in terms of capital investments such as supercomputers, databases, and networks. These are the most tangible components of the infrastructure; however, they are only the building blocks from which an infrastructure can be constructed. We define the national HPCC infrastructure in terms of two distinct but interdependent areas: technology infrastructure and human infrastructure. The HPCC technology infrastructure consists of three layers:

The federal HPCC program requires coordinated efforts between individual institutions and funding agencies.

- **Hardware and networks**, the bottom layer, includes computing hardware (supercomputers, workstations), networks, and storage archives. Here, we focus on systems that are available to the national community, including national networks and computing and storage systems available to the national community (as opposed to private computing resources).
- **Resources and services**, the middle layer, include such resources as national HPCC centers (providing computing services) and digital libraries (providing access to information) as well as support services for mobile and wireless hosts and workstation-based video conferencing. The resources and services provide a bridge between data and information.
- **The user environment**, the top layer, includes software systems that allow the user to locate, access, and effectively utilize services and resources and to collaborate with other users. This is the environment in which the user "navigates" through information and through the lower two layers of the technology infrastructure. The user environment is a bridge between information and knowledge.

Each layer builds on the layer below and provides infrastructure for the layer above.

Cutting across these three layers is a human infrastructure component consisting of teams of professionals, centers, and various communities. These human structures provide services and are themselves resources in terms of consulting, training, and development and advancement of the infrastructure itself.

Targets of Opportunity

The working group identified targets of opportunity for each of the layers outlined above.

Technology Infrastructure: Hardware. The need for teraflops computing power is well established for Grand Challenges. Today a number of supercomputers are available that have ideal peak processing power of over 100 Gflops. The current set of large, scalable machines will rapidly be outgrown by the current Grand

*Peak performance
is not important—
teraflops
application
performance is.*

Challenge teams, and so the infrastructure of large-scale systems needs to continue to expand both in terms of the number of moderately sized machines available for training and code development and in terms of the size of the major Grand Challenge systems. At least as important as the amount of memory are the processor-processor and processor-memory I/O bandwidth and latency and the external I/O bandwidth. We stress that teraflops ideal peak or even "benchmark" rates are not important to Grand Challenge applications—teraflops application performance is.

The Grand Challenge teams are projecting single computational experiments that create 10^{11} to 10^{12} bytes and archives (storing results from multiple experiments) of 10^{13} to 10^{15} bytes for a single Grand Challenge team archive. With teraflops computers, these experiments will create 10^9 to 10^{12} -byte data sets in a matter of hours. Revolutionary approaches to data storage are essential. Moreover, data must be readily accessible. Today's high-performance computing environments can routinely handle datasets up to several gigabytes with effective transfer rates at between 1 and 10 Mbytes/s. A balanced environment, according to the Grand Challenge team surveys, should allow single data files to be transmitted in 1 to 10 seconds and datasets (of a thousand such files) in 15 minutes to 2 hours. Coupling this with the data file and dataset sizes above results in the need for effective data transmission rates of 10^6 to 10^9 bytes per second—the higher rate being ten times the peak throughput of high-end local area networks in 1993 and twenty times the highest observed throughput of the fastest tertiary storage devices available in 1993.

To meet these increasingly higher data rates will require networks that can support multiple gigabits per second. In addition, the network capabilities requirements of both the high-end and the average user community continue to grow.

Technology Infrastructure: Resources and Services Current network-accessible resources and services are divided into three classes: those that are experimental (MOSIS, WhitePages), those supported as unique national facilities (supercomputer centers, NIC, library search capabilities), and those supported by community and/or volunteer efforts (on-line FTP archives, WAIS, World Wide Web, Gopher servers, USENET news). New services are outlined below, with an analysis of their projected impact on the infrastructure.

*The volume of
scientific data
being generated
is growing
exponentially.*

- **Information services.** These are becoming increasingly richer, with digital libraries, yellow-pages, white-pages, and data archives providing more intuitive user interfaces and thus growing in popularity.
- **Data archives.** The amount of data being generated is growing at an exponential rate. Instruments are becoming more commonplace on the network, and remote control of these instruments as well as real-time archiving of data will produce data streams between 10 and 100 MBytes/s. Making this data useful will require far more than high-density, high-throughput storage devices. The traditional directory services based on file names will yield to archive services coupled with digital libraries that can search based on content.
- **Special-purpose services.** Special-purpose services such as networked instruments, production facilities for image/video/CD-ROM output, and access to "turnkey" computing services will allow novice users to take advantage of popular and proven computational science codes. The community will need to

focus on providing high performance per application, as well as on maximizing access for training and software development. In addition, tele-classrooms and workshop facilities will allow groups of users to congregate in one or several locations for conferences, training events, symposia, etc. These tele-centers will provide for the distribution of meetings among major tele-centers as well as participation by individuals from their home workstations.

Emphasis must be placed on the security of computer data.

- **Access services.** Access services are emerging that will have a profound impact on the national infrastructure. In particular, packet video capabilities will be bundled in workstations over the next several years, with an estimated $O(10^4)$ systems capable of generating data streams of roughly 1 MB/s and $O(10^2)$ systems capable of generating up to tens of megabytes per second at high resolution. These will produce an aggregate national network requirement that may become the driving force of the network infrastructure. Additionally, within three years, $O(10^3)$ laptop systems will be capable of generating 1 MB/s data streams, and $O(10^4)$ systems will be generating 10 kbits/s streams; both sets will take advantage of wireless connections. The infrastructure must be able to accommodate these mobile systems in terms of both capacity and straightforward access.

All these resources and services must be demonstrable secure and trustworthy enough to be heavily used by U.S. industry, government entities, and even many educational institutions. Strong emphasis must be placed not only on security of sites and data but on robust authentication of communication and access. Advanced capabilities such as policy-based and type-of-service data routing and access will be required. Research is needed on how to instrument the HPCC infrastructure to support security (e.g., auditing technology) and on how to better understand (and minimize) the corresponding tradeoffs in speed and ease of use.

Technology Infrastructure: Software and User Environment. Even for very advanced scientific researchers such as those involved in Grand Challenge teams, the user interface to the infrastructure must allow them to focus on the scientific problems and not on the logistics of using the infrastructure. Research is therefore needed in several key areas:

A fundamental final step toward integration of the infrastructure is to provide a national file system.

- **National file system.** A fundamental first step toward integration of the infrastructure is to provide a national file system—allowing users to access their data, regardless of location, with consistent identifiers, or “pathnames.” The implementation of such an infrastructure will require considerable coordination.
- **Better operating systems, compilers, and code development, optimization, and execution environments for scalable systems.** The Grand Challenge teams are spending much effort on porting and optimizing applications for scalable systems. This experience should be captured through software sharing and through involving outside computer scientists and software developers in the projects.
- **Intelligent interfaces.** The many separate display standards, image and file formats, directory/indexing mechanisms, and other information storage techniques in use today must be hidden from the user by intelligent interfaces.

At the same time, coordination among information providers is necessary to ensure that the legion of information sources within the national infrastructure does not become the digital Tower of Babel.

- **Support for loading and selecting computer resources.** Services that allow the user to select the optimum computing resource at a particular moment, as well as services for allowing experiments to span several noncollocated processing resources, will be required.
- **Data analysis tools with intuitive user interfaces.** Visualization and other data analysis systems designed over the past several years for today's datasets will not scale to the data sets being envisioned by the Grand Challenge teams. The sheer quantity of data to be produced by these Grand Challenge experiments will require an entirely new data analysis paradigm.
- **Virtual environments.** Immersive technologies such as head-mounted or "walk-in" displays have the potential for both data analysis and collaboration, perhaps enabling a new generation of teleconferencing. The impact on the infrastructure of multiple, synchronized high-definition images will be significant compared even with single high-definition image streams, much less with today's relatively low-resolution image streams.
- **"Team" software.** Systems that will allow multiple team members to cooperatively analyze large data sets will be needed in order to allow these teams to work together effectively despite their geographic separation.

*Teraflops
computers require
a well-balanced
technological
and human
infrastructure.*

Human Infrastructure. The HPCC human infrastructure includes supercomputer centers, science and technology centers, Grand Challenge teams, the gigabit network testbed community, and the Internet community. These groups provide more than supercomputer access, also furnishing the expertise needed to "harness" scalable computers, effective peer-review mechanisms for computational science, extensive training and documentation, software, information, and data repositories, and development activities in user interfaces and computing environments themselves. These organizations also serve as intellectual centers, providing environments with long track records of catalyzing interdisciplinary computational science teams.

Findings

The working group proffered several suggestions to support and enhance both the technology infrastructure and the human infrastructure.

Technology Infrastructure. It is important to focus not only on the visible components, such as teraflops computer systems, but also on a well-balanced structure of networks, storage systems, and software environments that allow users to effectively access and utilize these teraflops systems. To maintain this balance, we make the following suggestions:

- Interconnect major national resources such as supercomputers, archives, and data analysis environments with network rates of at least 622 Mbits/s by

FY95 and at least 2.4 Gbits/s by FY96. Major universities and regional and corporate laboratories should have network access mechanisms that allow for peak rates of several hundred megabits per second by FY95 and up to one gigabit per second by FY96. The typical university sites should have peak capabilities of no less than tens of megabits per second by FY96.

- Fund coordinated efforts among major data and information providers in areas such as transparent interconnection of digital libraries and standards for information indexing, storage, and display.
- Integrate capabilities such as desktop video conferencing, virtual environments, and mobile and wireless hosts, and coordinate these capabilities with major national testbeds.

Human Infrastructure. The human infrastructure for HPCC provides fundamental reservoirs of expertise. These have a gestation period of at least two generations of technology, and their continuity assures smoother transitions from one generation of technology to another. We suggest that the federal HPCC program provide stable and long-term funding for new initiatives as well as stable base-level funding for ongoing operational support.



SECTION 6

Conclusions

The increasing capability to handle complex problems has already attracted industry involvement.

By bringing together applications scientists and software developers—groups that ordinarily simply “pass in the night”—the 1993 Grand Challenge workshop encouraged the sharing of ideas and problems in high-performance computing.

In addition to providing a status report on current Grand Challenge research and on recent software developments, the workshop provided a forum for identifying new mechanisms for improved software sharing and for exchanging information about software and hardware limitations impeding the High-Performance Computing and Communications program.

The researchers in the Grand Challenge community and in the Software Development community agreed that the federal HPCC program is already having an undeniable impact. Indeed, one might almost speak of a *high-performance revolution*, with widespread interest in and use of MPP systems by industry and academia; increased focus on parallelism by workstation vendors; growing emphasis on the need for standards; and almost universal concern for developing better products (machines, languages) to make parallel computing effective.

And while the researchers agreed that it is probably too early to expect HPCC program activities to have had a major impact on science and industry, *exciting progress is being made*. In particular, high-performance computing is enabling researchers to attack previously intractable problems, such as the recent 512³ simulation of compressible turbulence.

This increasing capability to handle complex problems has already attracted the attention of industry. *Industry involvement* was noted in numerous Grand Challenge projects, including aerospace, environmental science, biophysics, computational fluid and plasma dynamics, computational chemistry, reservoir modeling, and global computational finance.

Nevertheless, several *outstanding issues* were identified. Foremost was dissatis-

*Full use of
massively
parallel machines
will require
advances in tool
design, dataset
management,
and visualization
systems.*

faction with the current state of software tools. A frequent lament was that computer science researchers are not meeting the needs of the applications researchers, that those software tools that have been developed for high-performance systems usually are left in the prototyping stage. Indeed, some called the state of software tools "dismal." A clear need exists for better parallel debugging tools, tools for multidisciplinary applications, performance-monitoring tools, and language support to allow users to write programs at a higher level than currently possible. The cause for the poor software support is clear: a lack of incentive. The Grand Challenge grants currently focus on the output of the applications rather than on the software to achieve that output. Academia generally does not encourage computer scientists simply to "help out on a project"; nor do research institutions fund long-term projects to bring a software package up to commercial quality. Commercial vendors also are not rewarded for providing robust software tools: the emphasis of the HPCC program has been on more powerful machines, even when the software is not available to make the machines truly usable. The resulting paucity of tools is made worse by difficulties in sharing those tools that do exist. More effective mechanisms are needed for exchanging information on tool availability and accessibility.

A second major problem is dataset management. The Grand Challenge studies are generating enormous datasets (up to 8 terabytes, with predictions in the petabyte range). Making this data useful will require not only high-density, high-throughput storage devices but also improved archive and directory services. Entirely new data analysis paradigms will also be needed to add type information and structure, enabling researchers to access information more intelligently and more efficiently.

Even further, specialized visualization systems will be essential for examining the vast datasets expected from Grand Challenge research. Currently, several efforts are under way to develop image-rendering software that can be executed directly on massively parallel processing systems. Others are developing tools for graphics display offline. Moving the data to workstations results in high storage and I/O bandwidth requirements; moreover, if the data is to be displayed at full resolution, new techniques are needed to address the extreme demands placed on the image-rendering engines and the display subsystem.

Related to the growing data size of applications is the question of data rates. Current HPCC applications involve I/O rates of up to 6 MB/s to archive; this rate will increase to 100 MB/s in the near future. I/O is simply not scaling fast enough to keep up with CPU speed, and is threatening to become a major bottleneck for teraflops applications. Further research is needed to understand the I/O characteristics of massively parallel computers. Parallel I/O benchmarks are also needed that accurately reflect application I/O requirements. The Scalable I/O Initiative sponsored by the Concurrent Supercomputing Consortium will provide a first step toward meeting these needs, if it is funded by the various federal agencies.

Finally, a significant number of researchers felt that increased emphasis should be placed on standards. Current standards efforts include High Performance Fortran (which met with mixed reviews from the researchers, some of whom were extremely skeptical about its value), and the evolving Message Passing Interface standard; both focus on ensuring that programs are supported on a large number of different parallel computing environments. Further efforts are needed to promote vendor standards for archiving large amounts of data. An example is an industry effort to define standard interfaces for databases to make use of tertiary storage.



SECTION 7

Findings

The workshop participants suggested several new mechanisms for improving the effectiveness of the HPCC program.

There is nothing quite so frustrating for the avid do-it-yourselfer than to begin a project, suddenly need a particular tool, but have no idea where in the house to look for it.

— *Practical Homeowner's 1987 Do-It-Yourself Annual*

Many of the researchers at the Grand Challenge workshop expressed this feeling of frustration in discussing their attempts to find a suitable tool (whether a debugger or a performance evaluation tool), to use a (not-well-documented) tool efficiently, and to reuse it (i.e., have access to it through a software library) in other applications. Indeed, a majority of the suggestions at the workshop focused on the problem of development, maintenance, and distribution of tools. Here we summarize three of the principal suggestions and—more important—propose a mechanism for implementing each suggestion.

- *Suggestion:* Collaborative development of tools for Grand Challenge applications should be supported. Special attention should be given to parallel debuggers, performance evaluation tools, and interactive visualization software. Such efforts must be supported for all phases of the life cycle—fundamental research, prototyping, and development.

Mechanism: Support a “Grand Challenge in Software Tools,” and establish a Science and Technology Center for HPCC Software Tools.

- *Suggestion:* Tools that are ready for use outside of their development community should be made available in software-sharing libraries.

Mechanism: Establish or fund existing “software capitalization” programs to foster the development of research ideas and prototypes into production tools and to ensure the maintenance of the software in the library.

-
-
- *Suggestion:* Tools that have had success in the research community should be commercialized.

Mechanism: Fund an existing organization such as OSF, the national laboratories, or the NSF supercomputer centers; or set up a new independent organization or industrial consortium.

- *Suggestion:* Researchers need to be kept informed of the numerous tools that are available. At the same time, computer scientists must be made aware of the needs of applications scientists for specific tools.

Mechanism: Develop a parallel software tools electronic journal/newsletter (complete with an editorial board) to publicize tool availability and tool needs.

That the computer scientist is caught between Charybdis and Scylla—encouraged to create prototype tools as research projects, but given little credit for developing production-quality tools for widespread use—was clearly recognized by the conference participants. Many felt that the vendors must assume more responsibility in providing useful tools.

- *Suggestion:* The quality and performance of the basic HPCC tools should be improved. In particular, vendors should provide good node compilers to allow programmers to make more effective use of machines without resorting to assembly language coding; provide fast multiple-precision arithmetic; pay greater attention to symbolic and combinatorial aspects of computing, such as fixed-point arithmetic; provide language constructs that allow the easy definition of complex data structures; provide high-level, standardized message-passing routines; provide a random number generator with their machines; and develop a compiler infrastructure for making visible the compiler's internal representations and transformations to other software tools.

Mechanism: Require computer vendors (through explicitly worded RFPs) to provide good, usable tools for MPP machines. Have federal agencies support commercial development of software tools.

Of course, providing appropriate tools (whether from vendors or from computer scientists) requires an awareness of what is needed. While some applications projects explicitly made provision for "computer science work," most do not. The workshop participants agreed that much more interaction is needed.

- *Suggestion:* New incentives should be developed to encourage computer scientists and application scientists to interact more closely on Grand Challenge projects.

Mechanism: Have HPCC projects explicitly fund the exchange of students and staff between application and tool developers.

Software technology requirements certainly extend beyond tools for Grand Challenge applications research. Several areas were identified at the workshop as needing increased attention. Here we first list the various suggestions; a single mechanism is proposed to address these suggestions.

- *Suggestion:* High-performance file systems must be developed to ensure ubiquitous access to both conventional and high-performance distributed file systems.

Findings

- *Suggestion:* Database management systems must be developed that (1) support large objects, (2) provide fast access to subsets of large datasets, (3) interface to tertiary storage systems, and (4) incorporate new indexing and access methods for scientific data.
 - *Suggestion:* Methods should be developed to allow mixing of different languages in a single framework.
 - *Suggestion:* "Templates" that capture key computational, communication, and I/O characteristics of important application codes should be written and made widely available.
 - *Suggestion:* More emphasis should be placed on integrating support for I/O and networked services.
 - *Suggestion:* Standards efforts such as the Message Passing Interface forum should be encouraged.
 - *Suggestion:* Heterogeneous computing must be fully funded through large projects that include software, applications, and network research.
 - *Suggestion:* Engineers and scientists must be retrained in multidisciplinary technologies, and new researchers must be educated in this critical HPCC area through the establishment of multidisciplinary curricula at universities.
- Mechanism:* Increase the current federal involvement, specifically by providing (1) stable and long-term funding for new initiatives and (2) stable base-level funding for ongoing projects.



SECTION A

Grand Challenge Requirements Document

Prior to the three-day Grand Challenge meeting in Pittsburgh, the Program Committee prepared a detailed list of questions that were mailed out to the participants several weeks before the meeting. The four-part survey included questions on applications; Grand Challenge requirements (resources, algorithms, tools, I/O); the current status of tools, libraries, graphics facilities, and networks; and the most important needs in high-performance computing.

The results of the survey, which were circulated to the conference participants prior to the meeting, proved an invaluable resource for focusing the discussion groups.

In this appendix, we reproduce the survey and then summarized the survey responses. The individual responses are available electronically via anonymous ftp to the address `info.mcs.anl.gov` in the directory `/pub/tech_reports/hpcc`. Two files are included: `survey-merged.ps` and `survey.merged.enscript`.

Part A: Application Overview Issues

Please cover the following issues in the summary applications presentation sessions.

1. Brief Description of Your Application Problem.
 - description of the "scientific problem"
 - algorithms used
 - codes (size and number)
2. Characterization of Typical Computations Planned by GC Team.
 - describe "typical" computations

-
-
- current/planned compute resources
 - I/O requirements
3. Use of Visualization and Networking.
- graphics workstations
 - framebuffers
 - remote computing via networks
 - distributed computing via networks
4. Languages and Programming Tools Currently Used.
- Fortran, C, C++, others
 - PVM, p4, Express, PCN, others
5. Most Outstanding Problem You Wish Addressed at This Workshop.

Detailed Grand Challenge Requirements

Part B: Problem, Algorithm Description and Resource Requirements

1. Please list and briefly describe the major codes you plan to employ in the next 3-5 years for each of your Grand Challenge applications. How long do you estimate that each code would take to execute (please specify some current platform or estimated computational rate)?
2. Which of the following mathematical problems are solved in the course of running the code? For example: N-body force calculation, Navier-Stokes equations, Magnetohydrodynamics, Schrodinger equation, Maxwell's equations, Eigenvalue/eigenvector calculation, Direct simulation of physical phenomena, Sequence comparison, Fourier transform, Other (please specify)
3. Which algorithms do you currently use to solve your problem? For example:
 - Treatment of N-body problems- Fast Multipole Method (Rokhlin-Greengard, Barnes-Hut, etc.), Explicit calculation of N-body interaction requiring $O(N^2)$ operations, Explicit calculation of N-body interaction using cut-off.
 - Spatial discretization of PDE- Finite element, Finite difference, Spectral, Finite volume.
 - Temporal integration of PDE: Explicit solution of time-dependent PDEs, Operator splitting methods for implicit solution of time dependent PDEs, Fully implicit time dependent PDE solvers.
 - Monte Carlo algorithms
 - PIC methods
 - Geometric and symbolic algorithms
 - Discrete event simulations
 - Fast Poisson solvers

- Eigenvector/eigenvalue calculations: QR, Jacobi, Bisection methods.
 - Linear system solving: Direct methods, Iterative method (Preconditioner: none, diagonal, SOR, SSOR, ILU, Block ILU, other - please specify)
 - Fast Fourier transform
 - Other (please specify)
4. Estimate the percentage of time associated with each of the above algorithms.
 5. How much computational work (time on a specified platform) is associated with each algorithm?
 6. How much memory is needed for typical computations?
 7. What type of data structures do you employ?
 8. If your problem uses a grid or mesh, which of the following do you use? Single structured grid, Block structured grid (problem solved on multiple, interacting meshes), Unstructured grid.
 9. If your problem is formulated using matrices, which of the following do you use? Dense matrix representation; Banded, envelope, or skyline representation; Sparse matrix representation.
 10. What are (or will be) your I/O requirements with regard to the following? Data (in bytes) moved to and from secondary or tertiary storage during program execution; Minimum I/O bandwidths; Desired I/O bandwidths.
 11. Must you execute several different programs to solve your problems? If so, do these execute as a single application? as a pipeline? on the same computer? on different computers?

Part C: Needs Assessments and Opinions

I/O and Data Systems

1. What features would you most like to see in new I/O and data systems?
2. How large is the archive (something other than flat files) for data management?

Parallel Programming Paradigms and Program Development Tools

3. What parallel tools or languages have members of the group used?
4. Has the use has been minimal, moderate, or extensive?

Ex. mple software systems for next four questions: Parallel Languages/Compilers (CM Fortran, C*, *Lisp, PCN, Strand, Forge90, C with message-passing calls, Fortran with message-passing calls, etc.); Parallel Programming Libraries or Tools (Express, PVM, p4, Linda, Parti); Performance-Monitoring Tools (e.g., Gauge, Pablo, ParaGraph, Upshot); Parallel Debuggers (specify machine if the tool is supplied by the vendor); Other software tools that pertain to parallel machines (name the tools)

-
-
5. Which tools do you find are most useful?
 6. Which tools are conceptually poor but are well implemented (i.e., the tool does not do anything you need to have done, but does its tasks reliably)?
 7. Which tools are conceptually good but poorly implemented?
 8. What is the source of your parallel programming support software?

Performance Evaluation of MPP Applications

9. How do you characterize the performance of your application (MFlops, size of problem attempted, CPU time per model timestep, other)?
10. How do you know that your code is running at a reasonable performance level?
11. Which is more important: faster turnaround on existing problem sizes, or larger problem sizes with same turnaround time as today?
12. Do you expect to be able to use the same numerical methods on serial machines and MPPs for your application areas?

Discussion of Libraries

13. Do you use numerical libraries or solvers (e.g., LINPACK, LAPACK, EISPACK, PCGPACK, SPARSEKIT)? If you do use such codes, please specify which.
14. Do you use commercial or publicly distributed applications codes (e.g., GASP, NASTRAN)?

Graphics and Visualization

15. How do you (or will you) view your results? Small set of fixed or floating point numbers, Set of 2-D pictures, Set of 3-D pictures, Animations
16. What fraction of your group's time is spent analyzing the results of runs vs running them vs developing code?
17. What is the biggest problem in your postprocessing environment?
18. What capabilities do you want in a visualization system?

19. Is it important to be able to generate graphics directly from an MPP system?

National High-Performance Computing and Communications Infrastructure

20. How often do you use networks of workstations on a single application? If you do use networks of workstations on a single application, please answer the following:
 21. What performance do you achieve?
 22. Do you wish to use heterogeneous networks of machines, or are you using them because you do not have access to true MPP systems of sufficient capacity?
 23. What software tools do you use for distributed computing?

24. Would you prefer to compute on a network of workstations, on a tightly coupled centralized MPP, or on a network of MPPs?

Multidisciplinary Applications

25. If your problem involves execution of several different programs, which do you expect to parallelize? How will you exchange data between programs (e.g., files, sockets, message-passing)?

Part D: Most Important Needs

Indicate the areas where progress on your research can be accelerated if you (and your colleagues) could be assisted by other professional communities, including those of mathematics, computer science, and other computational science disciplines, and also by the computer industry. In particular, are there factors that are currently limiting your productivity (you may use the items on the following list as examples, adding to them where appropriate)? In your response, be as specific as possible concerning the character of the problem(s).

- Mathematical description of the problem(s) - theoretical or practical limitations
- Numerical limitations - word size; fraction or exponent limitation
- Algorithmic limitations - require new approaches, more generality, higher performance, etc.
- Library inadequacies
- Programming model inadequacies
- Language/compiler limitations - inadequate languages: what do you need? - inadequate compilers: in what way?
- Tool limitations - programming, debugging, performance monitoring, graphics, data management
- Turnkey packages that you would like to see developed
- Programming environment limitations - too cumbersome, not responsive, inadequate procedures, etc.
- Hardware limitations - performance, memory, I/O, connectivity

If you had one "software magic bullet," what would you use it for? Please specify one of the items listed above, if appropriate.



SECTION B

Summary of Survey Responses

Al Brenner (Supercomputing Research Center); Alok Choudhary (Syracuse); Tom Morgan, Gail Pieper, and Rick Stevens (Argonne National Laboratory); Joel Saltz (Maryland)

1. Overview

We received 28 completed surveys. This report provides a summary of some of the issues that have a particularly strong potential bearing on directions that need to be taken in systems software and hardware.

2. Issues

The teams identified a wide range of issues that they believed important:

- development of capabilities to allow asynchronous data transfer while processors are engaged in computations;
- faster parallel I/O;
- development of robust compilers for parallel architectures that abstract away low-level details of machines and algorithms, without catastrophic performance losses;
- extension of HPF to cover nonstandard data structures (block-structured data, unstructured data, etc.);
- compilers that take better advantage of RISC or vector processor architectures;
- decreasing communication latency;

-
-
- support for collective communications;
 - tools and techniques for heterogeneous environments; and
 - methods for load balancing.

3. Languages, Programming Tools, and Libraries Currently Used

Although there are almost as many different languages and tools being used by GC teams as the number of teams, the following briefly describes the most commonly used languages and tools. Fortran and C are both widely used: many teams use both (20 teams use Fortran, and 20 use C); 13 use C++, although no team reported using C++ as their only programming language. Message-passing libraries are widely used (22 teams reported using some kind of message-passing library, including libraries supplied by vendors, PVM, and PCL). Ten teams reported using a vendor-supplied parallel Fortran (typically CM Fortran); 5 teams reported using a vendor-supplied parallel C (typically C*); 5 teams reported using extensions for programming shared-memory machines.

Other languages and tools used by some teams include PCN, CC++, and pFortran.

Other tools used include p4, Linda, DTM AVS, PARTI, and Smalltalk (one or two users each).

CMSSL is extensively used by a number of teams. A few teams use BLAS, LINPACK, LAPACK, and EISPACK. A number of groups have developed their own C++ class libraries.

4. Data Structures Employed

The directions in Grand Challenge-related systems software research must be motivated by carefully considering the nature of the algorithms being tackled by the teams. Out of the 28 Grand Challenge teams responding to the survey, 26 reported that part of their work involves solving well-structured problems, including structured mesh and dense matrix calculations. However, 16 of the teams also report that some of their calculations involve problems with irregular indirection patterns including unstructured mesh and general sparse matrix calculations, traversal of linked lists, and calculations involving explicitly enumerated atom lists (in molecular dynamics problems). Approximately 12 teams reported that some of their calculations involved block-structured grids. In some cases the intergrid data transfer patterns are static; in at least 4 cases, the patterns change during the course of the problem. There were 4 teams that reported using tree-structured algorithms such as fast multipole algorithms.

5. Software Support Requirements

In general, most teams felt that most of the available tools are primitive and inadequate. There is very substantial interest in portable compilers able to give "reasonable" performance. There is a need to provide language support to allow users to write programs at a higher level than what is currently supported on parallel architectures. Many users want to be able to specify parallelism and data and task decompositions at a high level. Debuggers and performance tools were requested by almost all the teams. A large majority felt that software and architectural support for high-performance I/O is inadequate.

6. Compiler, Programming Tool, and Library Requirements

Portability of compilers across various machines was the most frequently requested feature. Fine tuning codes for different architectures was a major problem faced by most of the teams. Many researchers complained of buggy eta compilers. Quite a few of the responses criticized the quality of code produced by the compilers. Several responses asked for better node compilers.

Most of the responses pointed out the need for a much larger set of tools. Debugging tools and performance-monitoring tools were frequently given the highest priority. Quite a few responses mentioned the need to standardize tools across different architectures. Graphics tools were also considered highly inadequate.

I/O libraries were requested by many teams. Image processing and display libraries were also requested by a few teams.

7. Networking and Visualization Requirements

Most teams access MPP through what are perceived to be inadequate networks. Many plan to move to faster local networks. Graphics and visualization for examining output create particularly troublesome network problems. Most researchers report that they carry out visualization off-line; this results in high storage and I/O bandwidth requirements. Dataset sizes can range from 1 MB to 8 TB.

8. Algorithmic Requirements

A number of researchers felt that new algorithms need to be developed, for example:

- efficient algorithms to find eigenvalues in parallel,
- fast Poisson solvers for massively parallel machines'
- methods for efficiently solving sparse linear systems,
- efficient domain decomposition strategies,
- algorithms to minimize communication time for traversal of distributed data structures, and
- static and dynamic load-balancing methods.

What appear needed are truly parallel algorithms that are not directly connected with the earlier sequential formulations.

9. Hardware Requirements

Lack of sufficient I/O bandwidth on parallel machines was cited as the biggest single hardware limitation. Many users believe that the I/O bandwidth at each processor must at least equal that of sequential machines. Some users do not currently face I/O performance bottlenecks but predict that the I/O capabilities will at some point become a bottleneck. A number of users felt that parallel machines should offer a single I/O interface. Users felt that machines currently have a wide variety of I/O interfaces and that this lack of standardization causes problems in porting programs onto different machines.

Many users feel that the machines they are using should have a higher communications bandwidth and lower communication latency. Increased message latency

was cited as a particular bottleneck. Another frequently cited limitation was a scarcity of usable memory at each processing node.

Some users felt the need for better floating-point performance and larger cache size for the processors. Other users were unhappy that individual processors could achieve only a fraction of the possible peak performance on most real problems.

10. Mathematical Problems Solved

The Grand Challenge teams are solving a very wide variety of problems drawn from various scientific disciplines. In this brief summary of the survey results, we do not attempt to give a comprehensive picture of what physical or biological problems each team is attempting to solve and the mathematical formulations and numerical algorithms they employ. We do present a partial list of the mathematical problems being attacked by the Grand Challenge teams:

- N-body force calculation
- Solutions of the Schroedinger equation
- Solutions of the Dirac equations
- Flow solver: Navier-Stokes equations
- Hydrodynamics
- Magnetohydrodynamics
- Maxwell's equations
- Stochastic PDE (Brownian dynamics)
- Poisson solution of implicit radiation transfer equations
- Radiative transfer through absorbing and scattering media
- Conductive solver: conduction equation
- Integration of well-behaved, stiff, and beyond stiff ODEs
- Controls: 4th-order system of equations
- Shapiro filter
- Numerical integration of 2nd-order differential equations
- Integro-differential equations (aerosol physics)
- Eigenvalue/eigenvector problems
- Solution of linear systems
- Nonlinear parameter optimization
- Fourier transforms
- Moving body dynamics: 6 degrees of freedom
- Convolutions
- Radiative transfer equations
- Sequence comparison
- Direct simulation of physical phenomena
- Multidimensional pattern recognition
- 3D ray tracing through diffuse media
- Coupled structural dynamics and Navier-Stokes equations
- Coupled control, structural dynamics, and system identification
- Coupled Maxwell and Ginsburg-Landau equations



SECTION C

Abstracts of Talks

Panel A – Environmental and Earth Sciences

Chair: *Joan Novak, EPA*

A DISTRIBUTED COMPUTATIONAL SYSTEM FOR LARGE-SCALE ENVIRONMENTAL MODELING

Armistead G. Russell
Carnegie Mellon University

Environmental issues are of great societal concern. In the United States alone, more than \$30 billion per year is spent on air pollution control. A key premise of this proposal is that high-performance computing environments and numerical experimentation can play vital roles not only in unraveling the complex physics and chemistry occurring in the environment but also in shaping the design of cost-effective public policies. Most advanced environmental models are very computationally and data intensive. Distributed computer systems offer the potential to revolutionize the current approach to environmental modeling.

The proposed enabling technologies—giga-performance heterogeneous computing systems, advanced software environments, parallel architectures, and Gbits/s networks—offer the possibility not only of increasing the resolution but also of providing the power needed to perform systematic sensitivity/uncertainty analyses, assimilate observational data, and incorporate more realistic treatment of the underlying physical and chemical processes.

An initial application of a distributed computing environment is proposed that exploits high-speed communications networks and special-purpose architectures. Research will be carried out to improve the models and their mapping onto advanced architectures and to create a flexible software environment that will enable more widespread use of the tools. Extensive collaboration with industry, government, and other universities is planned.

COMPREHENSIVE EULERIAN AIR QUALITY MODELING OF MULTIPOLLUTANT SPECIES IN THE ATMOSPHERE

Kenneth Galluppi and Daewon W. Byun
North Carolina Supercomputer Center and
Environmental Protection Agency

Current EPA comprehensive air quality models simulate effects of atmospheric dynamics (transport, diffusion, and deposition), cloud processes (cloud mixing, aqueous chemistry, and wet scavenging), photochemical and thermal reactions (gas phase reactions of NO, NO₂, CH₄, CO₂, formaldehydes, Volatile Organic Compounds, nitrates, SO₂, sulfates, and oxidants), reactions of atmospheric particulates, and anthropogenic and biogenic emissions on air quality. Many of the inputs to the air quality models are provided by meteorological numerical models or objective analyses of observations. Because of strong dependency of air quality models on meteorological models, both modeling activities are coordinated.

One of the objectives of the Models-3 project is to develop next-generation air quality models with upcoming HPCC technology. The models will be used to enhance progress in related science and algorithms as well as to provide easy-to-use tools for environmental policy decision-making. EPA's Grand Challenge team is assigned to design a modeling system that is generalized to address multiple air quality issues from urban to regional scales and that is modular to enable continuous upgrading of science components. At a later stage, the models will be expanded to handle multimedia pollutant issues.

LINKED WATER AND AIR QUALITY MANAGEMENT

Thomas O. Barnwell, Jr., and Robin L. Dennis
U.S. EPA

The objectives of this project are to assist th the EPA's Chesapeake Bay Liaison Office in developing nitrogen control strategies for mitigating eutrophication in the Chesapeake Bay and to study the development of multimedia pollutant control strategies for air and water emissions. The project will link the MM4 Mesoscale Meteorological Model, the RADM Regional Acid Deposition Model, the HSPF Hydrological Simulation Model - FORTRAN, a 3-Dimensional Hydrodynamic Model of the Bay, and a 3-Dimensional Water Quality Model of the Bay.

Initial emphasis will be on linkage of RADM and HSPF. Improvements in RADM will focus on development of a finer grid scale to better resolve deposition processes

and emission sources, making RADM very compute intensive and prompting its movement to a massively parallel architecture. Additional work in HSPF will improve its ability to track nitrogen deposition in forests through the development of process descriptions for nitrogen transport and transformation in forested ecosystems. These improvements are necessary to provide the proper linkage between atmospheric deposition estimates and nitrogen inputs to the Bay throughout the terrestrial pathway. These improved codes will be tested in the Chesapeake Bay Watershed.

We will be exploring the use of the linked models in distributed and heterogeneous computing environments. Issues investigated will include transferring information between models both through simple data linkages and through more dynamic coupling. Also being developed are more intuitive model interfaces and decision support systems for generating control scenarios and analyzing model output, as well as exploring visualization techniques for model calibration, validation, and assessment of control strategies.

FUTURE OF HIGH-RESOLUTION WEATHER PREDICTION

Alexander MacDonald
NOAA/Forecast Systems Laboratory

A review of the past 30 years of weather prediction shows that progress in large-scale fields such as the 36-hour 500-millibar height has been quite remarkable. In the same period, improvements of precipitation forecasts, such as the 0- to 24-hour heavy-precipitation prediction issued by the National Meteorological Center, has been surprisingly modest. It is well known in the meteorological community that accurate prediction of cloud and precipitation is a difficult problem, both in the short term (will it rain tomorrow?) and the long term (will the central United States become much drier in the year 2050 as a result of greenhouse effects?) predictions.

NOAA's Forecast Systems Laboratory is taking a three-step approach to improving weather prediction. These steps are to greatly increase observations by using (1) heterogeneous data, (2) four-dimensional assimilation, and (3) massively parallel computers for high-resolution mesoscale models. Each step will be reviewed, with current results from small and regional scales presented, and implications for global models given.

High-performance computing and communications will improve weather prediction if a balanced program of increasing observations, meteorological models knowledge, and computing speed is pursued nationally.

RESERVOIR MODELING FOR POROUS MEDIA: PARTNERSHIP IN COMPUTATIONAL SCIENCE

Mary Wheeler
Rice University

Many of our groundwater resources are threatened by sources of pollution. The challenge is to develop efficient and accurate computer models for predicting the

distribution and movement of subsurface fluids in underground geological structures, and to predict contaminant transport through underground heterogeneous porous media. Important aspects of the problem include the interaction between subsurface and surface water, the spatial locations of contaminant sources, chemical reactions that occur in the subsurface (e.g., sorption, precipitation, dissolution, complexation, biological transformation), and the lack of measured data to adequately characterize the aquifer. Successful models should aid in the development and monitoring of remediation processes.

The first main problem is the numerical computation of fluid flow and transport in porous media, which involves solution of a coupled system of nearly hyperbolic and nearly elliptic parabolic equations for the fluid pressures and concentrations of dissolved substances. Emphasis is on efficient, high-accuracy solutions, especially in flows characterized by discontinuous or nearly discontinuous fluid and/or porous material behavior. The second main problem is to quantify the uncertainty associated with simulation results as a function of the uncertainty of medium and fluid parameters, especially when important measurement data are unavailable.

We plan a two-phased approach. A typical first-phase computation will involve a computation of single phase (i.e., in the saturated zone below the water table) fluid motion involving possibly several fluid components which react and diffuse. DOE sites suffering from groundwater contamination will be targeted for computation. Second-phase computations will take into account multiphase flow both in the vadose zone above the water table, and in the saturated zone. These problems typically require the solution of hundreds of thousands to millions of nonlinearly coupled unknowns, for simulation times from days to hundreds or even thousands of years.

This is a collaborative effort with Oak Ridge National Laboratory, Brookhaven National Laboratory, Rice University, SUNY Stony Brook, Texas A&M University, and the University of South Carolina.

RESERVOIR MODELING FOR POROUS MEDIA: LATTICE GAS COMPUTATIONS

Kenneth Eggert and Shiyi Chen
Los Alamos National Laboratory

We are developing enhanced oil recovery strategies. Our focus is on lattice gas computations. We use two major codes to model pore-scale processes. The first code, the lattice Boltzmann multiphase porous media model, has the capability of modeling multiple phases of fluids flowing through complex porous media. It includes variable contact angle and variable surface interface tension capabilities. The second code, the percolation-network model, bases the description of both the medium and the displacement process at a somewhat larger scale. Both codes are run on the CM-200.

TEN- AND HUNDRED-YEAR CLIMATE PREDICTION

Dave Bader
Pacific Northwest Laboratory

The goal of our project is to predict climate change on ten- to one-hundred-year time scales. To this end, we use models capable of estimating the range of possible future climate variables on spatial scales that are relevant to human activities to guide planning and policy development. The current approach is to couple models describing the atmospheric circulation, the ocean circulation, land surface material and energy exchange, and sea ice dynamics and thermodynamics. Algorithms and codes for each of these climate subsystems are different. The largest and most complex codes are the general circulation models (GCMs) of the atmosphere and ocean. The atmospheric codes typically include submodels for the land surface, and the ocean models can include sea-ice submodels. A variety of algorithms and approaches is used for GCMs, which must be capable of simulating geophysical fluid dynamics on appropriate scales. The approaches include finite-difference and spectral formulations of the geophysical fluid systems in spherical coordinates. The model codes typically run 10,000 to over 50,000 lines of Fortran code.

We are planning three types of computation: finite-difference atmospheric GCM, spectral-formulation atmospheric GCM, and finite-difference ocean GCM. The enormity of these computations is illustrated by our effort in developing a finite difference atmospheric GCM. Currently, our model requires about 8 single-CPU-hours per simulated year on a CRAY C90. The simulation we are planning—which will involve a ten-year simulation of the global atmosphere and land-surface, with a three-dimensional grid of about 10^5 cells, and prescribed ocean and sea-ice distributions—will require 140 single-CPU-hours.

DEVELOPMENT OF AN EARTH SYSTEM MODEL: ATMOSPHERE/OCEAN DYNAMICS AND TRACERS CHEMISTRY

C. Roberto Mechoso
University of California - Los Angeles

The goal of this project is to develop a coupled earth system model (ESM) of atmospheric and oceanic circulations and chemical tracers. The model development uses the following major components: (1) the UCLA general circulation model of the atmosphere (AGCM), (2) the GFDL/Princeton University general circulation model of the ocean (OGCM), (3) the NASA Ames/UCLA chemical/aerosol tracer model (CATM), and (4) the Lawrence Livermore National Laboratory atmospheric chemistry model (LACM). Versions of these component models are operational, and preliminary coupling of the components has been carried out.

The ESM will have a highly modular structure, in which components can be interchanged or added with a minimum of computational difficulty. The code will be highly optimized, including local parallelization within each module as well as between modular components. Finally, the coupled model will be configured to work

in a distributed (heterogeneous) computer environment across high-speed networks. The AGCM, OGCM, CATM, and LACM are all grid-point models, which is an advantage in parallel architectures. An important issue to be addressed in model development is load balancing among processors, which may become more critical as the parameterizations of physical and chemical processes grow in sophistication and the computational load between model regions becomes more variable.

The ESM will be used to study the climate system, climate change, and climate/chemistry interactions. This work is divided into studies of the global chemistry of the atmosphere and the fundamental coupled dynamics of the global atmosphere and ocean system. The interannual variability of the coupled atmosphere and oceans, including the seasonal cycle and transients up to decadal time scales, will be simulated. We plan to couple the AGCM and CATM to study global ozone depletion, including the effects of atmospheric aerosols on the chemistry of the upper atmosphere. The LACM will be used to develop and test photochemistry, and heterogeneous chemistry algorithms will be used in the coupled AGCM/CATM. The LACM will also be employed for tropospheric chemistry simulations using dynamical fields from the coupled AGCM/OGCM.

DEVELOPMENT OF ALGORITHMS FOR CLIMATE MODELS SCALABLE TO TERAFLUPS PERFORMANCE

Dan Kowalski
NASA/Goddard Space Flight Center

Our project involves long-term climate modeling using a coupled ocean, atmosphere, land-surface, and sea-ice model. The atmosphere and ocean models use the Arakawa C and B grid finite-difference schemes, respectively, for calculating fluid dynamic evolution. The atmospheric model includes multiband short- and long-wave radiation parameterizations; a stability-dependent, turbulent boundary layer scheme; the Relaxed Arakawa Schubert (RAS) scheme for penetrative convection; and a mosaic land surface model. The ocean model is a reduced-gravity, layered ocean model and includes a surface mixing parameterization at the ocean's surface.

Currently, a typical coupled ocean/atmosphere run of 30 simulated years requires about 225 single-node CRAY Y-MP hours. We plan an initial implementation of several prototype problems from our coupled climate model on several testbed massively parallel processors, including a six-node CRAY Y-MP, MasPar MP-1, Intel DELTA, and CRAY C-90. We will then conduct a full implementation of an improved coupled model. Some of the prototypes are atmospheric dynamical core, ocean dynamical core, shortwave radiation parameterization, cumulus parameterization, mosaic land surface model, and model diagnostic I/O.

**HIGH-PERFORMANCE COMPUTING AND 4D DATA ASSIMILATION:
IMPACT ON FUTURE AND CURRENT PROBLEMS**

Richard B. Rood; Presenter Peter Lyster
NASA/Goddard Space Flight Center

Our proposal focuses on the Grand Challenge of four-dimensional data assimilation to produce research-quality datasets for earth science studies. This involves the collection of diverse earth-observational data sets and the incorporation of these data into models of the ocean, land surface, and atmosphere, including chemical processes. Ultimately the goal of data assimilation is the calculation of consistent, uniform spatial and temporal representations of the earth's environment that can be used for scientific analysis and synthesis.

The proposal has two major thrusts. First, we propose to apply leading research in parallel software and algorithms to a number of existing techniques for assimilation that can benefit from greatly increased computer power. The operational global atmospheric data assimilation system of the Data Assimilation Office at the Goddard Space Flight Center, along with techniques for tracer advection and chemical analysis, will be implemented on parallel software/hardware systems and assessed both for performance and for development effort required. The product of our work will be a new generation of integrated, operational models running on heterogeneous parallel systems at orders of magnitude greater performance than the current operational system. The assimilation of reactive constituents is a high priority in NASA environmental observation programs, including the Upper Atmosphere Research Satellite and the Earth Observing System, and this will be an important step in making the assimilation of reactive constituents computationally viable.

Second, we propose to explore new data assimilation methodologies that can be implemented only with computers that far exceed the capacity of today's machines. We will emphasize the development of the first three-dimensional full Kalman filter scheme. We expect to exploit existing parallel system software as well as to develop techniques that arise out of the special requirements of data assimilation. Three-dimensional Kalman filtering schemes will challenge the limits of teraflops machines for the foreseeable future.

The data assimilation effort at NASA requires the extension of atmospheric data assimilation techniques to many disciplines of Earth Science. In order to develop a successful assimilation system for Earth Science, there will be a continual need to process and reprocess data sets with ever-improving and more complete assimilation systems. There will also be a requirement to diagnose the quality of the data sets. Data assimilation provides the most computer-intensive undertaking in NASA Earth Science research. It is essential that assimilation algorithms and models be able to exploit the future computing environment.

Panel B – Computational Physics

Chair: *Tom Kitchens, DOE*

UNDERSTANDING SOLAR ACTIVITY AND HELIOSPHERIC DYNAMICS

John Gardner
Naval Research Laboratory

This project seeks to develop and implement advanced numerical algorithms on parallel computers in order to solve Grand Challenge problems in solar and heliospheric physics. The effort will augment and be leveraged by existing programs at the Naval Research Laboratory in both solar theory and computation. We have proposed that NASA's HPCC initiative should have as one of its goals to understand the solar driving engine: the mechanisms of solar activity and the dynamics of the heliosphere.

Solar activity is the underlying driver for many important phenomena in space physics. UV and X-ray radiation from the sun play dominant roles in the formation of the earth's ionosphere and are thought to be partially responsible for the observed variations in the south polar ozone hole. High-energy radiation from solar flares strongly perturbs the structure of the earth's upper atmosphere and thus greatly affects the lifetimes of NASA satellites in low earth orbit. Planetary magnetospheres form and evolve as a result of the interaction between each planetary magnetic field and the solar wind originating in the corona. Solar coronal mass ejections drive geomagnetic storms and are responsible for electric power disruptions on earth. The ejecta are also the origin of interplanetary shocks and energetic particles.

During the next decade, solar activity will be the focus of a number of NASA space physics missions. The Gamma Ray Observatory is making observations of γ -ray emission from solar flares. YOHKOH is returning exciting data on X-ray emission from flares and active regions for the first time giving us detailed information on the three-dimensional structure and magnetic topology of solar flares and associated active regions, Ulysses is beginning its study of the solar wind at high latitudes, promising to provide new insights into the wind's acceleration mechanism and the three-dimensional structure of the heliosphere. In the next few years, the Solar and Heliospheric Observatory will begin study of the sun's inner and outer corona and the heliosphere.

To realize fully the scientific return from these missions, we urgently need to expand our dedicated effort in numerical and theoretical modeling into the massively parallel generation of computation hardware. We propose a program of research to develop new methods for numerical simulations on teraflops-range parallel computers and to apply this powerful new technology to outstanding problems in solar and heliospheric physics.

LARGE-SCALE STRUCTURE AND GALAXY FORMATION

George Lake; Presenter Calvin Lin
University of Washington

When we observe the universe, we use the dynamics and clustering of the luminous galaxies to infer the properties of the dark matter. Yet, even in our largest simulations of large-scale structure ($N = 10^{6.3}$), a galaxy is represented by ~ 100 particles. As a result, it dissolves by two-body relaxation in $\sim 10^8$ years. Many scientific problems can be addressed only by following the luminous tracers for billions of years. Testing models of the initial fluctuations by simulating their evolution into present-day structures is paramount now that COBE has provided a normalization of the power spectrum on large scales, and we have the correlation functions of clusters and galaxies on smaller scales. In addition, we have increasing data on the structure of the universe at earlier epochs. The Hubble Space Telescope has enabled researchers to follow QSO absorption line systems to small enough redshifts that galaxy association is unambiguous. Researchers are imaging the "anomalous population of blue galaxies" in clusters at modest redshift, and recent investigations promise to show us how clusters of galaxies have evolved. To understand these new discoveries, we must follow the evolution of the luminous tracers in the general sea of cosmic structure.

Common problems emerge in the simulation of large-scale structure, the formation of a cluster of galaxies, and the formation of a galaxy itself. For example, the rate of the dissolution of substructure on all three scales has become a popular way of estimating the present value of the cosmological density parameter Ω . Carlberg used the merger rate of galaxies to argue for a value of Ω near unity. Richstone, Loeb, and Turner came to a similar conclusion using the substructure present in clusters of galaxies. Toth and Ostriker concluded that $\Omega \ll 1$ by considering the accretion of satellites by disks. All of these ideas are exciting, but none are useful without careful calibration against simulations with 10^8 or 10^9 particles.

Our goals are threefold: (1) to simulate the formation of a cluster in an appropriate cosmological context ($\Omega = 1$ only), total mass = 10^{16} solar masses; (2) to simulate the formation of a galaxy in an appropriate cosmological context, total mass = 10^{13} solar masses (3-5 times the mass of the local group); and (3) to simulate large-scale structure ($\Omega = 1$) and a Cold Dark Matter power spectrum, total mass = 10^{18} solar masses. These simulations will each have follow 3×10^7 particles for 3,000 iterations. We will run some of these simulations on the KSR (where we are willing to perform calculations that take weeks) and also intend to do at least one production run on the Intel Touchstone DELTA (where each simulation will take roughly 1 day on the 512-node machine).

CONVECTIVE TURBULENCE AND MIXING IN ASTROPHYSICS

Robert Rosner
The University of Chicago

Turbulent convection plays a central role in many astrophysical circumstances,

ranging from the transport of energy from stellar and planetary interiors to their surfaces, to the mixing of specific angular momentum and dynamo activity in stars, planets, and possibly accretion disks, to finally the highly dynamic mixing encountered in novae and supernovae. We propose to develop the next generation of multidimensional hydrodynamic codes to attack these problems. Specifically, we plan to use massively parallel machines, establishing testbed and scalable codes from several approaches. These simulations are unique in their simultaneous demands for computing power, data handling capabilities, and scientific visualization. To address these demands, we have organized a team of researchers from the University of Chicago, Argonne National Laboratory, the University of Colorado at Boulder, Michigan State University, and the National Center for Atmospheric Research/High Altitude Observatory.

SCALABLE HIERARCHICAL PARTICLE ALGORITHMS FOR GALAXY FORMATION AND ACCRETION ASTROPHYSICS

Wojciech Zurek and M. Warren
Los Alamos National Laboratory

Particle codes can be applied to a variety of astrophysical (and other) problems. We are using a cosmological N -body treecode to study galaxy halo formation caused by gravitational clustering. We are also developing a smooth particle hydrodynamic code to study tidal disruption of stars in the vicinity of a black hole. These codes have been implemented on parallel supercomputers (especially on the Intel Touchstone DELTA). A major challenge is the development of techniques for analyzing the tremendous amount of data.

This project is carried out in collaboration with researchers at the California Institute of Technology, Syracuse University, the Pennsylvania State University, and Mount Stromlo Observatory.

LATTICE GAUGE THEORY

Bob Sugar
University of California at Santa Barbara

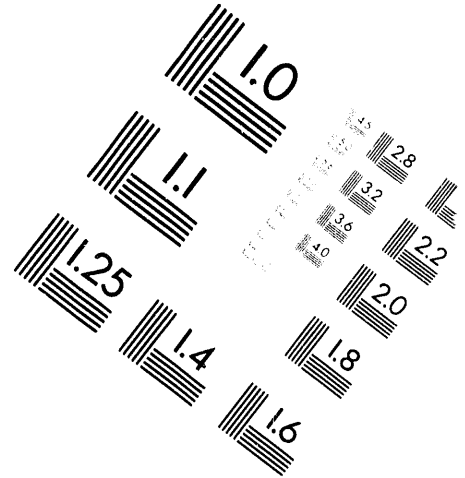
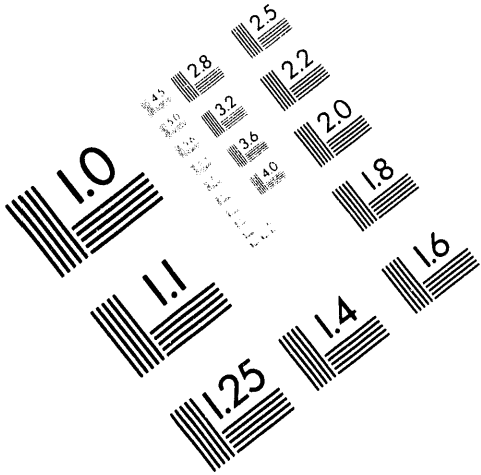
We plan to carry out a broad program of research in lattice gauge theory. Lattice gauge theory addresses some of the most fundamental theoretical problems in high energy physics and is directly relevant to the Department of Energy's experimental programs in high energy and nuclear physics. Our past work has included studies of the mass spectrum of the strongly interacting particles, the behavior of nuclear matter at high temperatures, the weak interactions of strongly interacting particles, and the Higgs sector of the electroweak interactions. A controlled calculation of the spectrum would provide a major test of quantum chromodynamics, the theory of the strong interactions. It would also demonstrate that we are in a position to make reliable calculations of physical quantities that are not as well determined experimentally as the masses. An understanding of the transition between the state of ordinary matter and the high-temperature quark-gluon plasma



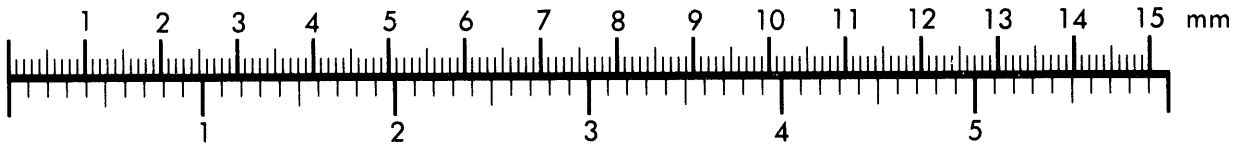
AIM

Association for Information and Image Management

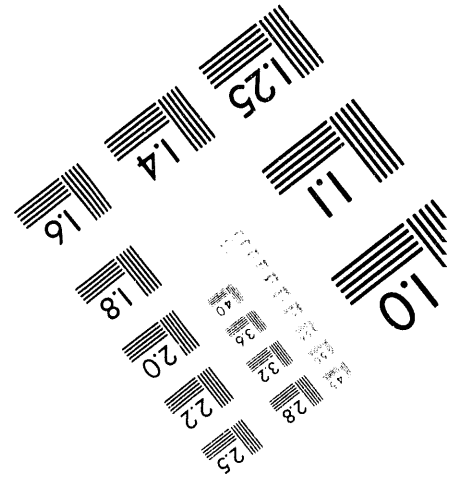
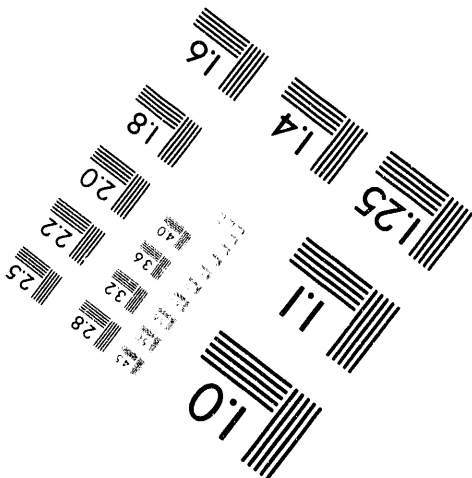
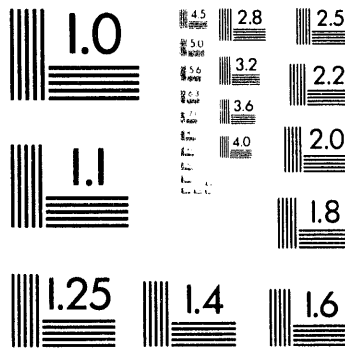
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910
301/587-8202



Centimeter



Inches



MANUFACTURED TO AIM STANDARDS
BY APPLIED IMAGE, INC.

2 of 2

and of the properties of the plasma is important for the interpretation of heavy-ion collision experiments planned by nuclear physicists at facilities such as the Relativistic Heavy Ion Collider. It is also important for cosmological models. The study of nonperturbative effects in the standard model of high energy physics is likely to be a particularly fruitful area for lattice gauge theory, as it will bring the theory into direct contact with ongoing experiments at the major DOE high energy physics laboratories and with experiments planned at the Superconducting SuperCollider Laboratory. In particular, lattice gauge theory provides the only known avenue for making first-principle calculations of the effects of strong interactions on weak processes. Therefore, it holds the promise of enabling us to make crucial tests of the standard model. We propose to continue our work in all of these areas.

The study of lattice gauge theory requires enormous computing resources and can benefit greatly from the new generation of massively parallel computers. In anticipation of these high-performance computers, we have developed parallel code that runs on a wide variety of platforms. At least initially, we plan to use the Intel Paragon at Oak Ridge National Laboratory, for which the code has been specifically tuned.

CONTROLLING CHAOS

C. Grebogi, E. Ott, and J. A. Yorke
University of Maryland

Our research is concerned with using small controlling perturbations to improve the performance of a system that is in a chaotic state. Two closely related general problems are being investigated: (1) control of long-term system behavior by location and stabilization of unstable periodic and steady-state orbits naturally embedded in the chaotic attractor, and (2) the rapid direction of an orbit to some target in phase space. Our technique for the solution of the first problem can be implemented without knowledge of a mathematical model of the system. The technique, which involves the use of embedding methods to analyze experimental data, has already been carried out in simple experiments by other groups, including applications to the control of a magnetoelastic buckling beam, arrhythmically beating heart tissue, lasers, spin-wave systems, and thermally convecting fluid. Our research currently concentrates on development of basic techniques for the solution of problems (1) and (2) for low-dimensional attractors. In this case computational requirements are not severe. However, we anticipate that the computational requirements will rapidly become limiting as we try to move to higher-dimensional chaotic processes. In particular, we foresee a need for large memory and rapid search capabilities.

RADIO SYNTHESIS IMAGING: AN HPCC APPLICATION

Richard Crutcher and Michael Heath
National Center for Supercomputing Applications

This project involves the direct implementation of three computing recommendations of the Astronomy and Astrophysics Survey Committee (chartered each decade

by the National Academy of Sciences and the National Research Council to survey needs of astronomy). The project will implement a prototype of the next generation of astronomical telescope systems—remotely located telescopes connected by high-speed networks to very high performance, scalable architecture computers and on-line data archives, which are accessed by astronomers over Gbit/sec networks. Specifically, a data link will be installed between the millimeter-wave synthesis array at Hat Creek, California, operated by the Berkeley-Illinois-Maryland Association (BIMA) and the National Center for Supercomputing Applications (NCSA) at Urbana, Illinois, for real-time transmission of data to the high-performance, massively parallel Thinking Machines Corporation CM-5. The very computationally intensive algorithms for calibration and imaging of radio synthesis array observations will be optimized, and new algorithms that use the massively parallel architecture will be developed. The project will experiment with and develop the techniques for (1) connecting via HiPPI the MPP CM5 and a high-performance visualization workstation (the SGI ONYX) at NCSA for distributed high-performance computing and visualization, and (2) extending this distributed computing and visualization environment by connecting a high performance visualization workstation at the University of Wisconsin to the NCSA CM5 over the BLANCA gigabit/sec network. Finally, the project will implement an on-line, digital archive of the BIMA radio telescope database and a digital library of processed images.

This research and development project will be carried out by a team of experts in radio astronomy, algorithm development for massively parallel architectures, high-speed networking, and database management in conjunction with Thinking Machines Corporation personnel. The development of this complete software, distributed computing, and data archive and library solution to the radio astronomy computing problem will advance our expertise in high-performance computing and communications technology and our knowledge of the astronomical universe.

Panel C – Computational Biology, Chemistry, and Material Sciences

Chair: *Bob Martino, NIH*

HIGH-CAPACITY ATOMIC-LEVEL SIMULATIONS FOR DESIGN OF MATERIALS

William Goddard
Caltech

We are working on two projects: one in quantum chemistry, the other in molecular dynamics. In the quantum chemistry project, we are concerned with systems of hundreds to thousands of atoms. The pseudo-spectral approach is being used, as well as Hartree-Fock. The goal is to build an algorithm that will scale as N^2 . In the molecular dynamics project, we are dealing with 200,000 to 1 million atoms per cell. Our approach is to divide the system into a hierarchy of cells. On a 30-node KSR machine, this approach turns out to be both faster and more accurate than standard approaches. We are now converting to the CM-5 and studying ways to exploit distributed memory.

**COMPUTATIONAL QUANTUM MATERIALS:
FIRST PRINCIPLES SIMULATION OF MATERIALS PROPERTIES**

G. M. Stocks, B. N. Harmon, and J. W. Davenport
Oak Ridge National Laboratory, Ames Laboratory, and
Brookhaven National Laboratory

We combine developments in the first principles theory of the electronic structure and energetics of materials, atomistic molecular dynamics simulations, and parallel processing to solve materials science problems of scientific and technological importance. Specifically, we are developing a range of methods for materials properties simulation that are based on a quantum mechanical description of the interactions. The methods include tight binding molecular dynamics, Car-Parrinello molecular dynamics, and locally self-consistent multiple scattering theory. These methods bridge the gap between traditional molecular dynamics and Monte Carlo simulation methods based on classical interactions and modern first principles electronic structure methods based on density functional theory. The new methods will enable us to perform quantum-statistical simulations on systems containing hundreds to thousands of atoms. We use parallel processing techniques in order to develop efficient scalable techniques that will allow us to solve materials problems currently out of reach.

**COMPUTER-AIDED DESIGN OF BIOLOGICAL
AND BIOMIMETIC MATERIALS:
COMPUTATIONAL DESIGN FOR CATALYSIS**

W. J. Camp, J. S. Nelson, S. J. Plimpton, and M. P. Sears
Sandia National Laboratories

Our Grand Challenge project is the creation of massively parallel computational tools enabling the study of catalytic and biocatalytic reactions from quantum mechanical and atomistic viewpoints. The application areas include microbiology (enzymatic reactions in proteins) and energy (gasoline cracking via polymeric reactions, catalytic reactions on crystal and cluster surfaces). We are developing simulation codes that perform quantum mechanical electronic structure calculations via local density functional, Hartree-Fock, and many-body formulations and molecular dynamic and Monte Carlo computations using empirical force fields. Most of the work is with codes we are writing ourselves; in other cases we are parallelizing existing commercial codes (DMOL and Discover from Biosym).

Typical computations are diffusion of small polymers in zeolites, metalloporphyrin trapping and reduction of carbon dioxide, active site analysis of metal clusters, and charge density shifts in enzyme and cofactor reactions. More algorithm development is needed to carry out these computations effectively. For example, longer timescale molecular dynamics simulations will need different algorithms that allow for much longer timesteps. Currently, performance is limited by the speed of individual processors. We are exploring implicit methods and bond-constrained techniques to overcome this barrier. Similarly, new serial and parallel algorithms are

needed to improve the scaling characteristics of electronic structure calculations, to enable larger atomic systems to be studied. We are exploring two promising linear scaling methods: one new algorithm is being explored that exhibits explicitly linear scaling (i.e., the computational work to simulate N atoms is $A \cdot N$, with A being a modest-sized number). In another project we are developing a massively parallel version of Yang's divide-and-conquer algorithm which also scales linearly.

In conjunction with the Grand Challenge, an optimized parallel library of dense-matrix linear algebra and finite-difference decomposition routines has been developed. The library includes MIMD routines with C and F77 interfaces for finite-difference operations, FFTs, many BLAS Level 1 and 2 operations, matrix transpose and multiply, Cholesky and LU decompositions, and Hermitian eigenvalue problems, as well as parallel graphics.

HIGH-PERFORMANCE COMPUTATIONAL STRUCTURAL BIOLOGY

Jerry E. Solomon
California Institute of Technology

The Computational Structural Biology project is focused on two specific science problems: (1) de novo protein three-dimensional structure prediction from primary amino acid sequence; and (2) predictive computational models of cellular organelles. Both of these problems, in addition to being computationally intensive, share the necessity of building new programming and user tools for exploring the full details of the problem.

We have developed a hierarchical computational approach to the de novo prediction problem which utilizes a novel static Monte Carlo method for generating large ensembles of candidate protein backbone structures on an fcc lattice as the top-level (coarse-grain) step. Our lattice model is capable of naturally incorporating known protein folding constraints into the chain generation process, and the algorithm is easily parallelized for use on large-scale massively parallel processors (MPPs). The lattice model, although a drastic simplification of the real conformational space of proteins, is able to capture much of the fundamental physics and chemistry of the folding process, including solvent accessibility effects. We illustrate the algorithm and some early results used for validation on serial workstations. The predictive computational model problem for cellular organelles focuses on building accurate (testable) models of the flagellar axoneme and the tRNA transcription operations of the ribosome. These models utilize the so-called teleological approach, wherein the problem is abstracted in terms of the system behavior rather than explicitly in terms of the problem constraints.

On the HPCC technology development side, we are concentrating on building sophisticated tools that enable both applications programmers and science users to take full advantage of the computational power afforded by large-scale MPPs to explore alternative approaches to problem solutions. In particular, we are attempting to build what we term a "numerical laboratory," which allows users to literally conduct experiments much in the same way as an experimental biologist would carry out experiments in a wet lab. These tools, implemented in CC++ (a parallel, object-oriented language) developed at Caltech, are being used to construct a protein-folding workbench and a predictive computational modeling workbench.

Both of these workbenches are capable of operating at three distinct levels, depending upon the computer literacy of the user, and are designed to be easily extensible in functional capability.

**HIGH-PERFORMANCE COMPUTATIONAL CHEMISTRY:
SCALABLE ALGORITHMS FOR GRAND CHALLENGE APPLICATIONS**

T. H. Dunning, Jr., and A. Wagner
Pacific Northwest Laboratory and Argonne National Laboratory

The High-Performance Computational Chemistry project is a multilaboratory collaboration to develop the computational technology required to solve Grand Challenge-class problems in computational chemistry. This focus of this project is on the development of the new algorithms, software, and diagnostics that will be required to fully exploit massively parallel computers capable of teraflops performance. In particular, efforts center on the development of algorithms that scale to larger numbers of processors, use hierarchical memory much more effectively, and make much smaller I/O demands than current methods. The goal is to develop modeling systems that are capable of solving problems orders of magnitude larger than currently possible. The work is focused on a suite of computational chemistry modeling methods appropriate to three specific problem areas of urgent interest to the U.S. Department of Energy and the chemical and petroleum industries: the incineration of halohydrocarbons, the interaction of soil minerals with pollutants, and the redesign of biodegradative enzymes.

Work is under way on the development of performance models and code for a fully distributed Hartree-Fock program for computing the electronic structure of molecules, scalable methods for dense eigensystem solution, and improved data sharing and data communications paradigms for computational chemistry codes. In the future this work will be extended to include scalable techniques for treating electron correlation (both perturbation theory and configuration interaction techniques) and for coupled semi-empirical electronic structure- molecular dynamics.

The High Performance Computational Chemistry project is a collaboration involving the Pacific Northwest Laboratory and Argonne National Laboratory, with participation by five industrial collaborators (Allied Signal, Amoco, DuPont, Exxon and Phillips).

**HIGH-PERFORMANCE COMPUTING IN STRUCTURAL BIOLOGY
AND MEDICAL IMAGING**

Robert L. Martino
National Institutes of Health

Scalable parallel computer architectures provide the computational rates demanded by advanced biomedical computing problems. NIH has developed a number of parallel algorithms and techniques useful in determining biological structure and function. These applications include processing of electron micrographs to

determine the three-dimensional structure of viruses, using ab initio quantum mechanical methods to investigate the energetics of how anti-cancer drugs affect the DNA in cancer cells, applying molecular dynamics to study the effects of hydration on protein structure, calculating the solvent accessible surface area of proteins to predict the three-dimensional conformation of these molecules from their primary structure, and aligning two three-dimensional positron emission tomography images for brain function studies. Timing results demonstrate substantial performance improvements with parallel implementations compared with conventional sequential systems.

HIGH-PERFORMANCE IMAGING IN BIOLOGICAL RESEARCH

D. Lansing Taylor and Scott E. Fahlman
Carnegie Mellon University

The goal of the Automated Interactive Microscope project is to bring high-performance computing to bear on the image processing, analysis, storage, and display problems associated with advanced fluorescence microscopy in the biological sciences. We will investigate a wide variety of image processing, display, and recognition algorithms. For example, we will perform nearest neighbor and minimum entropy deconvolutions to extract a clear three-dimensional image from a set of image planes at different depths in the specimen. Part of our research will focus on determining which algorithms are most useful. We plan to use graphics workstations (e.g., the SGI Onyx/2 with VTX graphics attached to the microscope) and, via a high-speed link, run on the CRAY C90, T3D, and CM-2 at the Pittsburgh Supercomputing Center.

We believe that this research will change the practice of microscopy in at least four important ways: (1) we will be able to run current image-processing and display algorithms in real time, providing timely feedback for the experimenter; (2) we will be able to employ much more ambitious processing and display algorithms than are used currently, allowing the experimenter to see the data in new ways; (3) we will be able to automate some of the recognition tasks associated with biological experiments, making it possible for the system itself to detect and record interesting events, where "interesting" is defined by the experimenter; and (4) once the system can recognize interesting events, it can perform certain experimental interventions automatically.

COMPUTATIONAL QUANTUM MATERIALS: QUANTUM STRUCTURE OF MATTER

T. Barnes, V. Overacker, M. R. Strayer, S. Umar
Oak Ridge National Laboratory, Vanderbilt University

This proposal centers on three computational methods to apply classical and quantum field theory to atomic, nuclear, and condensed matter physics. These

are (1) the time-dependent Hartree-Fock method applied to collisions and structure of atoms, molecules, and nuclei; (2) Monte Carlo perturbation theory applied to processes in accelerator and detector physics, notably RHIC; and (3) hydrodynamics applied to relativistic nuclear collisions and plasmas. Our objective is to advance the study of matter under strongly interacting conditions, as encountered in many branches of energy research, including fusion plasmas, lasers, and accelerators. While many of the proposed methods are transferable into applied areas, our emphasis is on understanding systems simple enough that results of high reliability are possible. Part of our proposed research deals with matter at energies below one MeV—an area that is also relevant to quantum chemistry and materials research. The remainder of our investigations deal with matter at higher energies—an area unique to the Grand Challenges.

**COMPUTER-AIDED DESIGN OF BIOLOGICAL
AND BIOMIMETIC MATERIALS:
PHARMACEUTICALS AND AGROCHEMICALS**

Mike Colvin
Sandia National Laboratories

The U.S. pharmaceutical and agricultural chemical industries are unequalled in the world. Not only do U.S. firms hold the largest share of the worldwide market, they are the source of most innovative new products in the biomedical and agrochemical markets. Both industries differ from most manufacturing industries in that a much larger fraction of their revenue goes into research and development of products for which they have only a few years of exclusive license protection. For example, a typical pharmaceutical drug takes 12 years to develop and get approved, leaving only 5 years of patent protection in which to recoup hundreds of millions of dollars in development costs. Hence, there is an enormous competitive advantage in shortening this development time. Moreover, traditional, exhaustive screening methods of designing new drugs and pesticides are proving inadequate. Instead, "rational" methods are being sought to design pharmaceuticals and pesticides with predictable potencies.

The Center for Computational Engineering at Sandia is collaborating with U.S. industry and universities to use new computational tools to solve problems in biomedical and agricultural chemistry. Sandia has unique experience in using highly accurate *ab initio* quantum chemical methods for large chemical systems. These very methods are far more computationally intensive than traditional semi-empirical methods and require extensive development of new algorithms and implementation on massively parallel supercomputers. In addition, Sandia has extensive capabilities in computational modeling of materials and molecular dynamics simulations.

Sandia is collaborating with researchers at a major U.S. pharmaceutical and several universities and medical schools to apply computational chemistry to solve problems in pharmaceutical and agrochemical design. The collaboration with the pharmaceutical company has involved computer modeling of numerous pharmaceuticals including anti-viral drugs and diagnostic imaging agents. Sandia is also working with the Johns Hopkins Medical School to develop more potent analogs

of the widely used anticancer drug cyclophosphamide. Sandia performed a series of ab initio quantum chemical calculations of cyclophosphamide metabolites bound to DNA, which constituted some of the largest quantum chemical calculations ever performed. Another collaboration is with the Department of Entomology at the University of California at Davis to help develop a new type of highly potent and environmentally save insecticide. Sandia is also collaborating with New York University and Lawrence Livermore National Laboratory to study how environmental carcinogens bind to DNA.

TEXAS CENTER FOR ADVANCED MOLECULAR COMPUTATION

Roland Glowinski, J. Andrew McCammon, B. Montgomery Pettitt,
and L. Ridgway Scott
University of Houston

High-performance computing can solve critical problems in biomolecular design, once several issues are resolved. The most obvious problem relates to the difficulty in using advanced computer architectures. In addition, new algorithms (mathematical and computer science-based) and methods (physical science based) are needed to solve problems with reasonable efficiency. Finally, to guide and provoke the development of both algorithms and software, scientists need a detailed understanding of what types of problems can be treated with vastly improved computing resources. We address all of these research areas in this proposal.

We will utilize and enhance parallel programming languages developed both by our group and by others to facilitate development and maintenance of complex algorithms for diverse high-performance architectures. One goal is to achieve supercomputer performance (and better) on low-cost parallel computers for commercial-grade codes with minimal programmer effort.

Examples of algorithms that we will study include fast summation methods for nonbonded forces in molecular dynamics, more efficient (implicit) time stepping schemes in molecular dynamics (which preserve energy), the use of multigrid and domain-decomposition techniques to resolve the electrostatic force distributions in the Brownian dynamics model, and parallel algorithms and data structures with an emphasis on load balancing.

The following Grand Challenge problems will be studied to improve our understanding of what can be resolved computationally (and what cannot): (1) thermodynamic cycle perturbation theory applied to ligand-binding proteins, (2) Poisson-Boltzmann/Brownian dynamics applied to protein design, (3) non-Boltzmann sampling and transition-state trajectories of ligand binding, (4) quenched high-temperature dynamic peptide mimetic design, and (5) grand canonical dynamics applied to salt solutions of nucleic acid triple helices.

Panel D – Computational Fluid and Plasma Dynamics

Chair: *Lee Holcomb, NASA*

HIGH-PERFORMANCE COMPUTATIONAL METHODS FOR COUPLED FIELDS AND GAFD TURBULENCE

Carlos Felippa and Juri Toomre
University of Colorado at Boulder

This project represents a merger of two HPCC Grand Challenge proposals: “High-Performance Computational Methods for Coupled Field Problems” and “Coherent Structures and Dynamics of Geophysical and Astrophysical Turbulent Flows.” The merged project involves a team of researchers at the University of Colorado at Boulder, the University of Colorado at Denver, the University of Minnesota, and the National Center for Atmospheric Research. This team will (1) develop and implement algorithms and software on massively parallel computers for solving coupled field problems in structural and fluid dynamics and (2) study turbulent flows that arise in geophysical and astrophysical fluid dynamics (GAFD).

The first component considers three problems: three-dimensional aeroelasticity, massively actuated “adaptive” structures, and electromagnetic-thermomechanical interaction. The emphasis in this component is on investigating benefits of finite-element-based partitioned analysis methods and heterogeneous parallel computation.

The second component considers four problems: geostrophic turbulence, oceanic convection, deep convection in planetary atmospheres, and stellar compressible convection, all constrained by effects of rotation and stratification. The emphasis in this component is on exploiting massively parallel architectures to increase the resolution of in three-dimensional simulations employing (variously) pseudo-spectral, finite-difference, multigrid, and PPM approaches in studying the intense turbulence encountered in planetary and stellar settings, including the large-scale coherent structures and mean flows that can coexist with such turbulence.

Computer science challenges paced by these problems include performance modeling and analysis, performance evaluation tools, program transformation and automatic data distribution, heterogeneous computation, very large scientific databases, and high-performance visualization.

MULTIDISCIPLINE SIMULATION OF HIGH-SPEED CIVIL TRANSPORT

M. D. Salas
NASA/Langley Research Center

In the past two decades, a foundation has been established for the advanced design of aerospace vehicles in mathematics, computer software and hardware, and engineering practices, leading to a new emerging technology called multidisciplinary design and optimization (MDO). This technology requires the repetitive analysis of

the vehicle structure, flow field, propulsion system, and controls as design variables are changed in the search for an optimum. The number of analyses required is proportional to the number of design variables that are allowed to change. For a typical high-speed civil transport configuration, the number of design variables can reach several hundreds. Since each analysis could take several hours of CPU time on a CRAY Y-MP computer and could require a Gigaword or more of memory, this Grand Challenge imposes severe requirements on computer resources. The difference in requirements imposed by the various disciplines lends itself to a heterogeneous computer implementation. In addition, the problem requires access to large, dynamic databases from different computer architectures and the rapid interrogation and visualization of this data. We explore the computer requirements both in terms of hardware and software needed to develop a framework for MDO.

MULTIDISCIPLINE SIMULATION OF HIGH-PERFORMANCE AIRCRAFT

K. Chawla
NASA/Ames Research Center

This project involves comparing the performance of high-performance aircraft computational fluid dynamics simulations on the parallel machines Intel iPSC-860 and TMC CM-2 and on the vector machine CRAY Y-MP. The problems under consideration require the use of multizone overset grid topology and solution of the Navier-Stokes equations. Our focus is on issues such as performance degradation on the Intel resulting from heavy message passing while solving all grids in parallel, strategy to map grids to processors, and disadvantages arising because of the SIMD nature of the CM-2.

We have compared the performance of various matrix solvers on different parallel machines, focusing on a single discipline application—specifically, the simulation of flow past a delta wing with thrust reverser jets in ground effect. We are also exploring strategies to map multidiscipline problems to the parallel machines, in particular, the Intel.

MULTIDISCIPLINE SIMULATION OF PROPULSION SYSTEMS

Russell W. Claus
NASA Lewis Research Center

Implementing new technology in aerospace propulsion systems is becoming prohibitively expensive. One of the major contributors to the high cost is the need to perform many large-scale system tests. The traditional design analysis procedure decomposes the engine into isolated components and focuses attention on each single physical discipline (e.g., fluid or structural dynamics). Consequently, the interactions that naturally occur between components and disciplines can be masked by the limited interactions that occur between individuals or teams doing the design and must be uncovered during expensive engine testing. This situation can pose serious problems for a highly integrated propulsion system, where tight coupling can produce unforeseen interactions having adverse effects on system performance.

For example, the installation of an improved efficiency turbine into an existing engine can result in a rebalancing of the engine cycle, with the consequence that the compressor is driven at a less efficient operating point, thereby reducing the total system performance as measured by specific fuel consumption or some other system parameter. As a result, the introduction of advanced technology into propulsion systems takes many years and is prohibitively expensive. New design techniques that combine concurrent engineering and numerical simulation may provide a system design solution.

We discuss a cooperative effort of NASA, industry, and universities to integrate disciplines, components, and high-performance computing into a Numerical Propulsion System Simulator (NPSS). NPSS is involved in the development of propulsion models for the computational aerosciences Grand Challenges. Development of propulsion modules requires advances in the following technologies: (1) interdisciplinary analysis to couple the relevant disciplines such as aerodynamics, structures, heat transfer, chemistry, materials, controls; (2) integrated system analysis to couple d subsystems, and components at an appropriate level of detail; (3) a high-performance computing platform composed of a variety of architectures, including massively parallel processors, to provide the required computing speed and memory; and (4) a simulation environment that provides a user-friendly interface between the analyst and the multitude of complex codes and computing systems that will be required to perform the simulations.

COMPUTATIONAL FLUID AND COMBUSTION DYNAMICS

Phil Colella
University of California - Berkeley

We are studying computational fluid dynamics as applied to combustion problems. Specifically, our focus is on three-dimensional, time-dependent, turbulent fluid-flow problems, in a variety of Mach number regimes ranging from extremely subsonic to hypersonic, and in complex physical geometries. Our numerical methods are based on finite difference methods on block—structured, dynamically adaptive grids in which the total number of grid points and their distribution in space change as function of time and the solution. We also use two techniques for representing complicated boundary geometries: composite overlapping meshes (e.g., CMPGRD, Chimera), and Cartesian grid/volume-of-fluid representation of geometry. We exploit visualization—in particular, animations generated directly from a massively parallel processing system—to gain new insights into the phenomena. To date, our principal limiting factor has been the capability to run very large simulations. MPP systems promise to provide the necessary resources to address truly Grand Challenge-scale problems.

NUMERICAL TOKAMAK PROJECT

Dan Barnes
Los Alamos National Laboratory

The Numerical Tokamak Project (NTP) is applying high-performance computing and communication to the most elusive aspect of tokamak design and operation—the problem of losses of plasma particles and energy. These losses always exceed the rates calculated for a completely static plasma because of fluctuations in which a large number of the confined particles move together (collectively). Such collective fluctuations enhance transport of heat in much the same way that fluid fluctuations (turbulence) lead to enhanced heat transport rates, for example, because of waves in the ocean or eddies in the atmosphere. The understanding of this process and its scaling as machine parameters are increased is critical to tokamak development. While impressive progress has been made, a key difficulty that remains is the detailed calculation of the solution of plasma transport models. The recent increase in the computing power available from massively parallel computers has made feasible the direct numerical solution of these models.

Unique physical features of this problem include the representation of the toroidal geometry, in which magnetic field lines travel on nested tori with a varying winding number, nonlocal response associated with the almost free motion of plasma particles along this magnetic field, nonlocal particle interactions via the electromagnetic field, and removal of fast time scales and/or small space scales by analytic or numerical approaches. The particular numerical difficulties that are introduced include use of general coordinates with nonstandard periodicity conditions, nonlocal dependencies associated with the solution of nonconstant coefficient elliptic problems, and nonlocal mapping of particle lists onto grid lists.

The NTP has developed both fluid and particle (kinetic) approaches for the study of tokamak turbulent transport. The former offers the advantages of no numerical noise and relatively few degrees of freedom per physical mesh point. The latter faithfully represents the long-mean-free path effects in a much more computationally intensive approach, and introduces the additional complication of sampling noise. Typical fluid calculations include about half a million degrees of freedom, while typical particle calculations use one to several million particles.

Unique numerical techniques developed to deal with the NTP problem include domain decomposition as a means of partial localization of particle information, fast parallel FFTs, convolutions, and linear system solvers, and conjugate gradient elliptic solvers. These approaches are described in terms of their numerical implementations on various (massively) parallel platforms (CRAY, TMC, and Intel). Network and visualization requirements to support these algorithms and their applications are also described.

Panel E – Applications of Artificial Intelligence

Chair: *Bob Voigt, NSF*

HIGH-PERFORMANCE COMPUTING FOR LEARNING: SUPER-HUMAN SUPERCOMPUTING vs SUPER-COMPUTING FOR HUMAN COMPUTATIONS

Robert C. Berwick and Tomaso Poggio
Massachusetts Institute of Technology

Many supercomputer problems involve important scientific calculations carried out at superhuman speeds: physical simulation of molecular structure through quantum mathematics, hydrodynamics, weather and atmospheric prediction, and many others. Another set of Grand Challenge problems underscores a rather different kind of high-performance calculations: those carried out by the human brain, especially involving learning and perception. From this perspective, these human-like problems stretch the limits of sample and memory complexity, rather than the computational complexity traditionally linked to supercomputing. This distinction illuminates once again the difference between super-human supercomputing and super-computing for human computations.

By envisioning a simple conversation with two people, we illustrate the pervasiveness of memory-intensive learning and brain-like supercomputing in everyday human tasks, what computational progress has been made and what new computational demands are entailed, and how this work points the way to new supercomputer architectures suited to nonscientific, human-oriented Grand Challenge problems.

DATA ANALYSIS AND KNOWLEDGE DISCOVERY IN GEOPHYSICAL DATABASES

Richard Muntz; Presenter Carlo Zaniolo
University of California - Los Angeles

This project will demonstrate the technical feasibility of building sophisticated information systems for geophysical databases and the benefits of such systems in promoting both theoretical and experimental research in the earth sciences. The testbed includes atmospheric model data, satellite stratospheric data, and climate data. Characteristic of these applications are the identification and monitoring of complex patterns and their evolution over space and time. We will study novel indexing and abstraction techniques for efficient search and monitoring of massive data sets, and the optimization of complex spatial-temporal queries and rules. An implementation based on a mass-storage system and an intelligent front-end is planned. Supercomputer testbeds will be used for parallel search and computation intensive functions.



SECTION D

Workshop Invitees

Robert Allen, AT&T
Lou Auslander, CUNY
Dave Bader, DOE Office of Health and Environmental Research
Mike Bailey, San Diego Supercomputer Center
Ron Bailey, NASA Ames Research Center
Raymond Bair, Pacific Northwest Laboratory
Dan Barnes, Los Alamos National Laboratory
Thomas Barnwell, Environmental Protection Agency
Alan Barr, Caltech
Forest Baskett, Silicon Graphics Corp.
Isabel Beichl, National Institute of Standards & Technology
John Bell, DOE
Marsha Berger, Courant Institute
Donna Bergmark, Cornell University
Fran Berman, University of California - San Diego
Robert C. Berwick, Massachusetts Institute of Technology
Kenneth Birman, Cornell University
Alan Blatecky, MCNC
Marjorie Blumenthal, National Academy of Sciences/Nat. Research Council
Ronald Boisvert, NIST
Jay Boris, Naval Research Laboratory
Lou Branscomb, John F. Kennedy School for Government
Joseph Bredekamp, NASA Headquarters
Alfred Brenner, Supercomputing Research Center
Frederick Brooks, Jr., University of North Carolina
Robert Brown, Silicon Graphics, Inc.

J. C. Browne, University of Texas
Duncan Buell, Supercomputing Research Center
Henry Burkhardt, Kendall Square Research Corp.
Ralph Burnstein, Electrical Power Research Institute
James Burrows, National Institute of Standards & Technology
DaeWon Byun, Environmental Protection Agency
Bill Camp, Sandia National Laboratories - Albuquerque)
David Cargo, National Security Agency
John Carlson, Cray Research, Inc.
William Carlson, Supercomputing Research Center
Pat Carson, National Institutes of Health
T. L. Casavant, University of Iowa/Parallel Processing Lab.
Charlie Catlett, University of Illinois NCSA
John Cavallini, Department of Energy
Kalpana Chawla, NASA Ames Research Center
Y. T. Chien, National Science Foundation
Alok Choudhary, Syracuse University
Mel Ciment, National Science Foundation
Charles Clark, NIST
Russel Claus, NASA Lewis Research Center
Carlie Coats, North Carolina Supercomputing Center
Tom Cole, U.S. Army Engineer Waterways Experiment Station
Phillip Colella, University of California - Berkeley
Stephen Colley, Ncube
Michael Colvin, DOE
George Cotter, National Security Agency
Thomas Crocket, NASA Langley Research Center
Richard Crutcher, University of Illinois
Ernie Daddio, Environmental Information Service
James Davenport, Brookhaven National Laboratory
Anil Deane, NASA Goddard Space Flight Center
Viktor Decyk, Jet Propulsion Laboratory
Tom DeFanti, UIC/EECS
Robin Dennis, Environmental Protection Agency
Carleton DeTar, University of Tennessee
John Dennis, Rice University
Dave DeWitt, University of Wisconsin
Charles Dickens, OSTP
Pete Dillingham, Cray Research Inc.
Jack Dongarra, University of Tennessee
John Dorband, NASA Goddard Space Flight Center
Shawn Downing, NOAA
Richard Draper, Supercomputing Research Center
Kevin Drogenmeir, University of Oklahoma
Thom Dunning, Pacific Northwest Laboratory
Doug Dwoyer, NASA Langley Research Center
Roger Dyson, NASA Lewis Research Center
Todd Elvins, San Diego Supercomputer Center
Robert Ewald, Cray Research, Inc.

Workshop Invitees

Richard Ewing, Texas A&M University
Carlos Felippa, University of Colorado at Boulder
Jack Fellows, DOE Office of Management and Budget
Jeanne Ferrante, University of Colorado
Robert Ferraro, Jet Propulsion Laboratory
James Fischer, NASA Goddard Space Flight Center
Jon Flower, ParaSoft Corp.
Greg Follen, NASA Lewis Research Center
Ian Foster, Argonne National Laboratory
Geoffrey Fox, Syracuse University
Joan Francioni, University of Southwestern Louisiana
Richard Freund, NCCOSC RDTE
Richard Friesner, Columbia University
Kenneth Galluppi, North Carolina Supercomputer Center
Dennis Gannon, Indiana University
John H. Gardner, Naval Research Laboratory
Al Geist, Oak Ridge National Laboratory
Louis Gerstner, IBM Corporation
John H. Gibbons, Office of Science and Technology Policy
Norman Glick, National Security Agency
William A. Goddard, California Institute of Technology
Robert Goldberg, Government Operations
Mel Goldstein, NASA Goddard Space Flight Center
Gene Golub, Stanford University
Howard Gordon, Supercomputing Research Center
Al Gore, White House
David Gottlieb, Brown University
Steven Gottlieb, Indiana University
Leslie Greengard, Courant Institute, New York University
Steve Griffin, National Science Foundation
Nico Habermann, National Science Foundation
Chuck Hansen, Los Alamos National Laboratory ACL
Anthony Hearn, RAND
Robert Harrison, Battelle-PNL
Michael Heath, University of Illinois
Kristin Hessenius, NASA Headquarters
Danny Hillis, Thinking Machines Corp.
Dan Hitchcock, Department of Energy
Jeff Holand, U.S. Army Engineer Waterways Experiment Station
Lee Holcomb, NASA Headquarters
Terry Holst, NASA Ames Research Center
Sally Howe, National Institutes of Health
Fred Howes, DOE
Paul Hunter, NASA Headquarters
Mack Hyman, Los Alamos National Laboratory
Steve Isakowitz, DOE Office of Management and Budget
Barry Jacobs, NASA Goddard Space Flight Center
Van Jacobson, Lawrence Berkeley Laboratory
Calvin Johnson, DCRT, NIH

Frederick Johnson, National Institute of Standards & Technology
Gary Johnson, Department of Energy
Lennart Johnsson, Thinking Machine Corp.
Anita Jones, University of Virginia
Bob Kahn, CNRI
Jeffrey Kalb, MasPar Computer Corp.
Charles Kalina, National Institutes of Health
Malvin Kalos, Cornell University
Ken Kennedy, Rice University
Christopher Kerr, Princeton University
Bill Kirchhoff, DOE
Thomas Kitchens, DOE
Kenneth Kliewer, DOE
Chuck Koelbel, Rice University
Sylvain G. Korzennik, Smithsonian Astrophysical Observatory
Norman Kreisman, Department of Energy
George Lake, University of Washington
Monica Lam, Stanford University
Meemong Lee, Jet Propulsion Laboratory
Creon Levitt, NASA Ames Research Center
Kai Li, Princeton University
Donald Lindberg, National Institutes of Health
Jerry Linn, National Institute of Standards & Technology
Rik Littlefield, Battelle-PNL
Fred Long, National Oceanic and Atmospheric Administration
John Lou, Jet Propulsion Laboratory
Steve Louis, Lawrence Livermore National Laboratory
Sandy MacDonald, National Oceanic and Atmospheric Administration
Jacob Maizel, National Institutes of Health
Allen Malony, University of Oregon
Vincent Marier, National Security Agency
George Marsaglia, Florida State University
Robert Martino, National Institutes of Health
Michael Mascagni, Supercomputing Research Center
Edward Masi, Intel Scientific Computers
Daniel Masys, National Institutes of Health
Edward McCracken, Silicon Graphics Computer Systems
Greg McRae, Lawrence Livermore National Laboratory
Scott McNealy, Sun Microsystems, Inc.
Carlos Roberto Mechoso, University of California, Los Angeles
Piyush Mehrotra, ICASE (USRA)
John Mellor-Crummey, Rice University
Paul Messina, California Institute of Technology
Barton Miller, University of Wisconsin
Ray Miller, National Security Agency
Clark Mobarry, NASA Goddard Space Flight Center
Reagan Moore, San Diego Supercomputer Center
Robert Morris, National Security Agency
Richard Robert Muntz, University of California, Los Angeles

Workshop Invitees

Dave Nelson, Department of Energy
Mike Nelson, Office of Science and Technology Policy
Les Nichols, NASA Lewis Research Center
Mike Norman, University of Illinois NCSA
Joan Novak, Environmental Protection Agency
Ed Ott, University of Maryland
James Ousley, Control Data Systems, Inc.
Ross Overbeek, Argonne National Laboratory
Robert Palmer, Digital Equipment Corp.
V. Pankoni, National Science Foundation
Janice Patrick, IBM Federal Systems Co.
Merrel Patrick, National Science Foundation
David Patterson, University of California
Michael Pavloff, Silicon Graphics
Larry Peterson, University of Arizona
Eckhard Pfeiffer, Compaq Computer Corporation
Lawrence Picha, NASA
Lewis Platt, Hewlett-Packard Company
Gary Pope, University of Texas at Austin
Terrence Pratt, NASA Goddard Space Flight Center
Tom Pyke, National Oceanic and Atmospheric Administration
Irene Qualters, Cray Research, Inc.
Cal Ramos, National Institutes of Health
Daniel Reed, University of Illinois
David Robard, DCRT, NIH
John Rollwagen, U.S. Dept of Commerce
Richard Rood, NASA Goddard Space Flight Center
Joan R. Rosenblatt, National Institute of Standards & Technology
Ralph Roskies, Pittsburgh Supercomputer Center
Robert Rosner, University of Chicago
Bruce Ross, Geophysical Fluid Dynamics Laboratory
Armistead G. Russel, Carnegie Mellon University
Manuel Salas, NASA Langley Research Center
Joel Saltz, University of Maryland
Ahmed Sameh, University of Illinois CSRD
William Scherlis, DARPA/SISTO
Klaus Schulten, Beckman Institute, University of Illinois at Urbana
L. Ridgway Scott, University of Houston at University Park
Mary Anne Scott, DOE
John Sculley, Apple Computer, Inc.
Kamy Sepehrnoori, University of Texas at Austin
James Sethian, University of California
Peter Shames, Jet Propulsion Laboratory
Bruce Shapiro, FCRDC
Margaret Simmons, Los Alamos National Laboratory
Horst Simon, NASA Ames Research Center
Ronald Skates, Data General Corp.
Burton Smith, Tera Computer Co.
Lauren Smith, Supercomputing Research Center

Paul H. Smith, NASA Headquarters
David Smitley, Supercomputing Research Center
Marc Snir, IBM T.J. Watson Research Center
Jerry Solomon, California Institute of Technology
Stephen Squires, Defense Advanced Research Projects Agency
Guy Steele, Thinking Machines Corp.
Thomas Sterling, NASA Goddard Space Flight Center
Kenneth Stevens, NASA Ames Research Center
Rick Stevens, Argonne National Laboratory
Walter Stevens, NIST
Malcom Stocks, Oak Ridge National Laboratory
James M. Stone, University of Maryland
Michael Strayer, DOE
Max Suarez, NASA Goddard Space Flight Center
Bob Sugar, University of California, Santa Barbara
Francis Sullivan, Supercomputing Research Center
D. Lansing Taylor, Carnegie Mellon University
Frank Tower, National Oceanic and Atmospheric Administration
J. G. Treybig, Tandem Computers
Lew Tucker, Thinking Machine Corp.
Sait Umar, Vanderbilt University
James Unruh, Unisys Corporation
Anthony Villasenor, NASA
Robert Voigt, National Science Foundation
Al Wagner, Argonne National Laboratory
David Walker, Oak Ridge National Laboratory
Steven Wallach, Convex
Michael Warren, Los Alamos National Laboratory
Harvey Wasserman, Los Alamos National Laboratory
Richard Watson, Lawrence Livermore National Laboratory
Howard Watts, Cray Computer
Gilbert Weigand, DARPA/CSTO
Jack Wells, Oak Ridge National Laboratory
Mary Wheeler, Rice University
Andy White, Los Alamos National Laboratory ACL
Elizabeth Williams, Supercomputing Research Center
Irving Wladawsky-Berger, IBM Corporation
Stephen Wolff, National Science Foundation
Paul Woodward, University of Minnesota AHPCRC
John Wooley, DOE
Patrick Worley, Oak Ridge National Laboratory
Chee Yap, New York University
Jeffrey Young, Environmental Protection Agency
Steven Zalesak, NASA Goddard Space Flight Center
Thomas Zang, NASA Langley Research Center
Mary Zosel, Lawrence Livermore National Laboratory
Wojciech H. Zurek, Los Alamos National Laboratory

DATE

FILMED

6/17/94

END

