Grant #DE-FG02-97ER25311

FINAL TECHNICAL REPORT

Eighth SIAM Conference on Parallel Processing for Scientific Computing

March 14-17, 1997
Hyatt Regency Hotel, Minneapolis, Minnesota

The Eighth SIAM Conference on Parallel Processing for Scientific Computing was held in downtown Minneapolis on the dates above. More than 350 people attended the meeting. This SIAM conference is the premier forum for developments in parallel numerical algorithms, a field that has seen very lively and fruitful developments over the past decade, and whose health is still robust. Other, related areas, most notably parallel software and applications, are also well represented. The strong contributed sessions and minisymposia at the meeting attest to these claims.

At this important juncture in the commercial evolution of parallel computing, the organizers of the conference decided to emphasize the past successes and future prospects of parallel processing for scientific computing. Invited speakers in the first category included Robert Bixby of Rice University, who spoke on parallelizing linear and mixed integer programming; Ridgway Scott of the University of Houston, who spoke on high performance computing in biomolecular simulation; and Paul Woodward of the University of Minnesota, who spoke on the use of parallel machines for simulations in computational fluid dynamics. Invited speakers who peered into the future included John Toole of the National Coordination Office for Computing, Information, and Communications, who spoke on the federal, academic and industrial frontiers in parallel computing for science and engineering; Daniel Reed of the University of Illinois, who spoke on patterns and policies in parallel input/output; and Larry Smarr of the National Center for Supercomputing Applications, who spoke on future trends in high performance computing architectures. These provocative and inspiring invited talks were supplemented by two invited panel discussions, one moderated by Fran Berman of UCSD on future trends in software support for parallel computation, and the other moderated by Larry Smarr on future trends in computer architecture. Both sessions featured eminent panelists and elicited substantial audience participation. A copy of the complete program accompanies this report.

The conference, sponsored by the SIAM Activity Group on Supercomputing, was organized by Co-Chairs Michael Heath of the University of Illinois and Virginia Torczon of the College of William & Mary, along with an organizing committee consisting of Greg Astfalk of Hewlett-Packard, Petter Bjorstad of the University of Bergen, Alan Karp of Hewlett-Packard, Charles Koelbel of Rice University, Vipin Kumar of the University of Minnesota, Robert Lucas of DARPA, Layne Watson of Virginia Tech, and David Womble of Sandia National Laboratories.

DISCLAIMER

SIAM conducted the conference with the partial support of the Department of Energy and the National Science Foundation.

Minisymposia, Contributed Papers, and Posters

Due to the number of interesting themes proposed, the conference featured an unusually large number of Minisymposia, twenty in all, about half of which were invited and half contributed. There were an additional twenty one regular contributed sessions, and a single poster session featuring over fifty presenters.

Proceedings

A proceedings was produced on CD-ROM and made available at the conference. The proceedings contains papers from the contributed talks, minisymposia, and poster presentations. The CD-ROM format enabled authors to use more space for their papers, allowed the use of color, and provided a format that is searchable and easily portable.

Papers in the proceedings focus on the major themes of the conference, including computational fluid dynamics, data-parallel languages, large-scale parallel applications, message-passing, molecular modeling, parallel I/O, parallel libraries, parallel software tools, parallelizing compilers, particle simulations, problem-solving environments, and sparse matrix computations.

Researchers, scholars, students, educators, scientists, engineers and managers who are using parallel computing, designing and analyzing parallel computers, or developing algorithms, tools and environments for parallel computers should find these proceedings of interest.

Tutorial

The conference was preceded by a full-day tutorial entitled ``Performance Programming in Scientific Computation'' given by Bowen Alpern of IBM T. J. Watson Research Center and Larry Carter of UCSD.

Student Support

SIAM made available a limited amount of support for students to attend the conference. The priority for funding was students, student speakers, post docs, and others, in that order. Preference was given to U.S. students. Anyone within a 50 mile radius of the conference site was not considered.

Several students were granted travel awards under this program.

Summary

This sequence of conferences continues to be enthusiastically received since those attending recognize that powerful parallel computing capabilities coupled with their creative use are essential in addressing a wide range of problems and that advances in the utilization of parallel computing will enable advances in other scientific and engineering fields.

Next Meeting

The Ninth SIAM Conference on Parallel Processing for Scientific Computing will be held in 1999. Exact dates, location, and organizers have yet to be selected.

Michael Heath and Virginia Torczon
Conference Co-Chairs

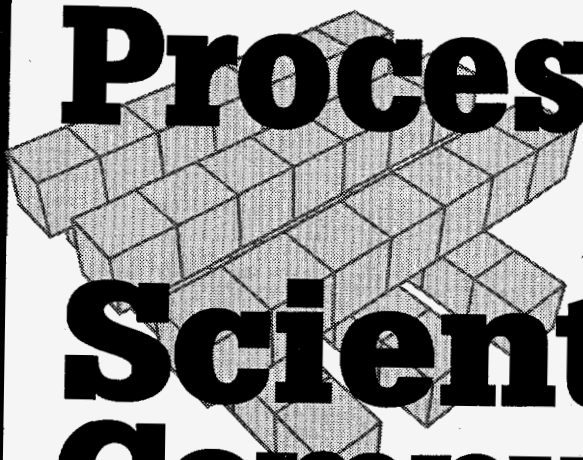
SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS
3600 University City Science Center
Philadelphia, PA 19104-2688

CONF-9703157--Abstr.

FINAL PROGRAM AND ABSTRACTS

Eighth SIAM Conference on

Parallel Processing for Scientific Computing



March 14-17, 1997

Hyatt Regency Minneapolis on Nicollet Mall Hotel
Minneapolis, Minnesota

Sponsored by SIAM Activity Group on Supercomputing

PLUS

SIAM Short Course on Performance Programming for Scientific Computation

March 13, 1997

Conference Themes

- Combinatorial optimization
- Data-parallel languages
- Large-scale parallel applications
- Message-passing
- Molecular modeling
- Parallel I/O
- Parallel libraries
- Parallel software tools
- Parallelizing compilers
- Particle simulations
- Problem-solving environments
- Sparse matrix computations

siam
Society for Industrial and Applied Mathematics

http://www.siam.org/meetings/pp97/pp97home.htm

Organizing Committee

Contents

Funding Agency

SIAM is conducting this conference with support from the Department of Energy and National Science Foundation.

Notice to All Presenters

Two standard overhead projectors and two screens will be provided in every meeting room. Speakers with special audiovisual equipment needs should inform the SIAM Conference Department of their specific requirements by February 12, 1997. If we do not hear from speakers by that date, it will be understood that a standard overhead projector is all that is needed. If a speaker sends a request for special audiovisual equipment and decides not to use the requested equipment after it has been installed, the speaker will be responsible for paying the rental fee.

Some examples of special audiovisual equipment and rental fees are:

LCD Panel	$250	26" Data Monitor	$200
35 MM Slide Projector	$38	Shure Mixer	$25
Video Projector	$500	IBM PC Computer	$225
1/2" VHS VCR	$55	Xenon 35MM Projector	$200

SIAM thanks participants for their cooperation.

Important Notice to all Meeting Participants

Times allowed for each presentation, including questions and answers:

CP = Contributed Presentations (20 minutes)

IP = Invited Plenary Presentations (60 minutes)

MS = Minisymposium (30 minutes)

The Poster Sessions will take place on Friday evening, March 14 from 6:00 PM-8:00 PM. There will be a reception during the session. Poster presenters are asked to post and setup their presentation materials at 3:00 PM on Friday. Displayed materials should be removed from the poster boards by 8:00 PM on Friday. Any materials left on the board after that time will be removed and discarded. SIAM is not responsible for any discarded poster displays that are left at the end of the session.

Note:

For papers with multiple authors, the speaker is shown in italics if known at press time.

The conference organizers expect every speaker of a scheduled presentation to register and attend the conference. If it becomes necessary for a speaker to cancel the presentation, the speaker is expected to find an alternate presenter or one of the speaker's co-authors should give the presentation. The speaker should inform the SIAM conference department of any change to his/her presentation.

A "no-show" or canceled presentation can cause serious inconvenience to the attendees and conference organizers.

For more information about poster presentations, please refer to the "Guidelines for Meeting Participants" at *http://www.siam.org/meetings/guidhome.htm*.

Get-Togethers

■ Poster Session and Reception

Friday, March 14, 1997

6:00 PM-8:00 PM

Lake Superior Room-5th Floor

Complimentary refreshments will be served. A prize will be awarded for the best poster presentation.

■ Business Meeting

SIAM Activity Group on Supercomputing

Saturday, March 15, 1997

6:00 PM-7:00 PM

Lake Superior Room-5th Floor

All are welcome to attend this meeting and reception that follows.

SIAM Short Course on Performance Programming for Scientific Computation

Thursday, March 13, 1997

Hyatt Regency Minneapolis on Nicollet Mall

Morning

8:00 Registration

8:30-9:15 Introduction
- What is performance programming?
- Challenges to attaining high performance
- The scientific method
- Visualizing computers and computation
- Extended example: seismic migration

9:15-10:00 Architecture for Performance Programmers
- The RAM and PRAM models
- Unblocked matrix multiplication
- A two-level memory model
- The memory hierarchy
- Multiple processing elements and parallelism
- Pipelines and their hazards
- The Parallel Memory Hierarchy model of computation

10:00-10:30 Coffee

10:30-12:30 General Techniques
- Localization
- Parallelization
- Pipelining
- Example: dense linear algebra
- Example: integer tallying
- Example: protein matching
- Extended example: the NAS/CG benchmark

Afternoon

12:30-2:00 Lunch

2:00-3:30 Miscellaneous Tips and Techniques
- Reading assembly code
- Timing and profiling
- A dusty-deck Cray-code example
- Inner loop considerations
- Example: the NAS/EP benchmark
- Message compression

3:30-4:00 Coffee

4:00-4:45 Portable High Performance
- The LAPACK paradigm
- Polyalgorithms and tuning parameters
- Toward a methodology for portable performance

4:45-5:30 Review
- Extended example: fast Fourier transforms

5:30 Short Course adjourns

Organizers and Instructors

Bowen Alpern, IBM T. J. Watson Research Center, and Larry Carter, University of California, San Diego and San Diego Supercomputer Center.

Description

Performance programming seeks to improve performance beyond what is achieved by programming an algorithm in the most expedient manner. The goal is to keep each processing element as busy as possible doing useful work. This entails satisfying four requirements: breaking problems into independent subproblems that can be executed concurrently, distributing these subproblems appropriately among the processing elements, making sure that the necessary data is close to its processing element, and overlapping communication with computation where possible. To attain high performance, these requirements must be satisfied whether one views "processing elements" as stages of an arithmetic or vector pipeline, functional units of a CPU, processors of a tightly coupled shared-memory multiprocessor, nodes of a distributed-memory supercomputer, or heterogeneous computers on a network. This tutorial presents general techniques for satisfying each of these requirements and illustrates their use at many different levels of application.

Level of Presentation

30% beginner; 50% intermediate; 20% advanced

The tutorial will use extended examples, including two-dimensional seismic migration, protein matching, and computational linear algebra (matrix factoring, matrix multiplication, and its degenerate cases). Seismic migration is a representative of certain partial differential equation problems, protein matching is a typical dynamic programming application, and linear algebra is ubiquitous. Other examples will be introduced to illustrate particular points. While we will survey a large number of topics and techniques, the emphasis will be on mastering conceptual structures and understanding general principles rather than on learning details.

Intended Audience

The tutorial is intended for computational scientists, application developers, and other professionals who have a need to design, implement, or tune high-performance scientific programs. It should also be of interest to computer scientists who want to develop languages, compilers, operating systems, architectures, and performance monitoring and debugging tools that can better support the needs of the performance programming community.

Instructors

Bowen Alpern received a Ph.D. in Computer Science from Cornell University in 1986. He has been a Research Staff Member in the Mathematical Sciences Department of the IBM T. J. Watson Research Center since 1986. His research interests include performance programming, visualization of computation and architecture, theoretical models of hierarchical memory and parallelism, distributed and parallel computing, message compression, computational linear algebra, and portable high-performance computing. He has published more that twenty-five technical papers in computer science. He taught a graduate-level course in performance programming for the Computer Science Department of Columbia University in 1994.

Larry Carter is a Professor in the Computer Science and Engineering Department of the University of California at San Diego, and a Senior Fellow at the San Diego Supercomputer Center. Dr. Carter received his Ph.D. from the University of California at Berkeley in 1974, and worked until 1994 at IBM's T. J. Watson Research Center in the areas of probabilistic algorithms, compilers, VLSI testing, and high-performance computation. His current research interests include scientific computation, performance programming, parallel computation, and machine and system architecture for high-performance computing.

Bowen and Larry developed the matrix multiplication package initially released with the RS/6000 and helped implement the NAS benchmarks on the IBM SP.

Important Notice

For a complete, updated description of the short course, visit: http://www.research.ibm.com/perfprog/course/pp97.html

The short course will take place on the 2nd Floor Greenway F-H; coffee breaks will be in Promenade area 2nd Floor; and lunch will be in Greenway A-B/I-J.

Program-at-a-Glance

Wednesday, March 12

6:00 PM-8:00 PM

Registration
Greenway Balcony

Thursday, March 13

7:30 AM-5:00 PM

Registration
Greenway Balcony

8:30 AM

Short Course begins
(see separate program on page 3)

5:30 PM

Short Course Adjourns

Friday, March 14

7:30 AM

Registration
Greenway Balcony

7:45 AM

Welcoming Remarks and Announcements
Michael Heath, University of Illinois, Urbana-Champaign and Virginia Torczon, College of William & Mary

8:00 AM-9:00 AM

IP1 Parallel Computing for Science and Engineering: the Federal, Academic, and Industrial Frontier
John C. Toole, National Coordination Office for Computing, Information, and Communications

Chair: Michael Heath, University of Illinois, Urbana-Champaign

Greenway C-H

9:00 AM

Coffee
Mirage

9:30 AM-11:30 AM

Concurrent Sessions

MS1 Trends in Languages and Compilers for Parallel Computing
Organizer: David A. Padua, University of Illinois, Urbana-Champaign

Nicollet D2

MS2 Graph Partitioning: Parallel Algorithms and Applications
Organizer: Anshul Gupta, IBM T. J. Watson Research Center

Nicollet D3

MS3 Interactive High-Performance Computing
Organizer: Anne E. Trefethen, Cornell Theory Center

Greenway I-J

CP1 Linear Algebra
Chair: Ahmed H. Sameh, University of Minnesota, Minneapolis

Nicollet D1

CP2 Applications I
Chair: Robert F. Lucas, DARPA/Supercomputing Research Center

Greenway A-B

11:30 AM

Lunch

12:30 PM-1:30 PM

IP2 Parallelizing Linear and Mixed Integer Programming Using Shared Memory
Robert E. Bixby, Rice University

Chair: Virginia Torczon, College of William & Mary

Greenway C-H

1:30 PM-3:30 PM

Concurrent Sessions

MS4 Recent Progress in Sparse Direct Methods
Organizer: Padma Raghavan, University of Tennessee, Knoxville

Nicollet D2

MS5 Parallel N-Body Solvers
Organizer: Robert D. Skeel, University of Illinois, Urbana-Champaign

Nicollet D3

MS6 Support for Dynamic Processor Allocation on Scalable Multiprocessors
Organizers: Vijay K. Naik and José E. Moreira, IBM T. J. Watson Research Center

Greenway I-J

CP3 Parallel Performance Evaluation
Chair: David E. Womble, Sandia National Laboratories, Albuquerque

Nicollet D1

CP4 Parallel Numerical Methods
Chair: Layne T. Watson, Virginia Polytechnic Institute and State University

Greenway A-B

3:30 PM

Coffee
Mirage Room

4:00 PM-6:00 PM

Concurrent Sessions

MS7 Software Tools for Developing Parallel Applications
Organizer: Cherri M. Pancake, Oregon State University

Nicollet D2

MS8 Parallel Linear Programming: Practical Approaches and Implementations
Organizers: Thomas F. Coleman Cornell University and Stephen J. Wright, Argonne National Laboratory

Nicollet D3

CP5 Eigenvalues
Chair: Kristyn J. Maschhoff, Rice University

Nicollet D1

CP6 Parallel Languages
Chair: Robert Schreiber, Hewlett-Packard Company

Greenway I-J

CP7 Applications II
Chair: Greg Astfalk, Convex Computer/Hewlett-Packard Company

Greenway A-B

6:00 PM-8:00 PM

Poster Session/ Reception
Lake Superior

Saturday, March 15

8:00 AM

Registration
Greenway Balcony

8:00 AM-9:00 AM

IP3 High-Performance Computing in Biomolecular Simulation
L. Ridgway Scott, University of Houston

Chair: Charles H. Koelbel, Rice University

Greenway C-H

9:00 AM-10:00 AM

IP4 High-Performance I/O Patterns and Policies
Daniel A. Reed, University of Illinois, Urbana-Champaign

Chair: Charles H. Koelbel, Rice University

Greenway C-H

10:00 AM

Coffee
Mirage

10:30 AM-12:00 PM

Panel: Future Trends in Software Support for Parallel Computation
Moderator: Francine Berman, University of California, San Diego

Greenway C-H

12:00 PM

Lunch

1:30 PM-3:30 PM

Concurrent Sessions

MS9 High-Performance and Parallel I/O Systems for Scientific Computing
Organizer: Rick L. Stevens, Argonne National Laboratory

Lake Superior

NOTE:

Nicollet rooms are on 1st Floor. Greenway rooms are on the 2nd floor. The Mirage rooms on the 2nd floor. Lake Superior rooms are on the 5th floor.

Program-at-a-Glance

MS10 Resource Scheduling/ Management for Parallel Scientific Computing
Organizer: Mark S. Squillante, IBM T. J. Watson Research Center
Greenway I-J

MS11 Biomolecular Modeling
Organizer: L. Ridgway Scott, University of Houston
Greenway C-E

CP8 Adaptive Grids
Chair: Mark T. Jones, University of Tennessee, Knoxville
Greenway F-H

CP9 Sparse Matrix Methods
Chair: Vipin Kumar, University of Minnesota, Minneapolis
Greenway A-B

3:30 PM

Coffee
Mirage

4:00 PM-6:00 PM

Concurrent Sessions

MS12 Languages and Compilers for Parallel Computing
Organizer: Charles H. Koelbel, Rice University
Lake Superior

MS13 Parallel Dense Linear Algebra Libraries
Organizer: Robert A. van de Geijn, University of Texas, Austin
Greenway C-E

MS14 Fast Parallel Orthogonal Transforms: Theory, Implementation, and Applications
Organizer: S. Lennart Johnsson, University of Houston and Harvard University
Greenway F-H

CP10 Parallel I/O
Chair: Paul C. Messina, California Institute of Technology
Greenway I-J

CP11 Molecular Dynamics
Chair: John A. Board, Duke University
Greenway A-B

6:00 PM-7:00 PM

Business Meeting
SIAM Activity Group on Supercomputing
Lake Superior

Sunday, March 16

12:30 PM

Registration
Greenway Balcony

1:30 PM-2:30 PM

IP5 The Use of Parallel Machines for CFD Simulations
Paul R. Woodward, University of Minnesota, Minneapolis
Chair: Petter E. Bjørstad, University of Bergen, Norway
Greenway C-H

2:30 PM-3:00 PM

Coffee
Promenade Area

3:00 PM-5:00 PM

Concurrent Sessions

MS15 From C++ to Java: Object-Oriented Methods in Parallel Processing
Organizer: Dennis B. Gannon, Indiana University, Bloomington
Greenway C-H

CP12 Computational Fluid Dynamics I
Chair: Lyle N. Long, Pennsylvania State University
Lake Superior A

CP13 Parallel Compilers
Chair: Piyush Mehrotra, ICASE, NASA Langley Research Center
Greenway A-B

CP14 Iterative Methods for Linear Systems I
Chair: Calvin Ribbens, Virginia Polytechnic Institute and State University
Mirage

CP15 Preconditioners
Chair: David E. Keyes, Old Dominion University
Lake Superior B

CP16 Graph Partitioning
Chair: George Karypis, University of Minnesota, Minneapolis
Nicollet A1-A2

5:00 PM-7:00 PM

Concurrent Sessions

MS16 Is Message-Passing Obsolete?
Organizer: Ewing L. (Rusty) Lusk, Argonne National Laboratory
Greenway C-H

MS17 Implementing Parallel Adaptive Procedures for PDEs: Issues and Experiences
Organizer: Peter K. Jimack, University of Leeds, United Kingdom
Lake Superior A

CP17 Computational Fluid Dynamics II
Chair: Moti Mittal, Ohio Supercomputer Center
Greenway A-B

CP18 Iterative Methods for Linear Systems II
Chair: Yousef Saad, University of Minnesota, Minneapolis
Mirage

CP19 Fast Algorithms
Chair: Alan Edelman, Massachusetts Institute of Technology
Lake Superior B

Monday, March 17

8:00 AM-10:30 AM

Registration
Greenway Balcony

8:00 AM-9:00 AM

IP6 Future Trends in High-Performance Computing Architectures
Larry Smarr, University of Illinois, Urbana-Champaign
Chair: Greg Astfalk, Convex Computer/Hewlett-Packard Company
Greenway C-H

9:00 AM-10:30 AM

Panel: Future Trends in Computer Architecture
Moderator: Larry Smarr, University of Illinois, Urbana-Champaign
Greenway C-H

10:30 AM

Coffee
Promenade Area

11:00 AM-1:00 PM

Concurrent Sessions

MS18 Problem-Solving Environments
Organizer: John R. Rice, Purdue University
Nicollet D1

MS19 Parallel Eigenvalue Methods for Scientific and Engineering Applications
Organizers: Yousef Saad and Andreas Stathopoulos, University of Minnesota, Minneapolis
Nicollet D2

MS20 Generating Efficient Parallel Scientific Code From High-Level Descriptions
Organizer: S. Lennart Johnsson, University of Houston and Harvard University
Nicollet D3

CP20 Domain Decomposition
Chair: Petter E. Bjørstad, University of Bergen, Norway
Greenway A-B

CP21 Applications III
Chair: John Shadid, Sandia National Laboratories, Albuquerque
Mirage

1:00 PM Conference Adjourns

Friday, March 14 *morning*

8:00 AM-9:00 AM

Chair: Michael Heath, University of Illinois, Urbana-Champaign

Greenway C-H

IP1

Parallel Computing for Science and Engineering: the Federal, Academic, and Industrial Frontier

One of today's challenges in science and engineering is to exploit the enormous potential of computing, communications, and information technology. High-performance computing and communications have paved the way for improving computing systems, designing and implementing fast, efficient algorithms for simulation and modeling, and developing technology for a nationwide high-speed networking infrastructure. Research in high-performance computing is leading to improved systems software and software implementing fast, efficient algorithms for simulation and modeling to solve complex problems in timely ways. Wide deployment of the products of long-term communications R&D has enabled scientists and engineers around the world to access these resources and to work more effectively with one another.

While tremendous progress has been made, long-term challenges still exist. The federal government continues to play a major role through its Computing, Information, and Communications R&D programs, built upon the core developed by the Federal High Performance Computing and Communications (HPCC) Program. After a short retrospective on the impact of high-performance computing and parallel computing in particular, current research areas, strategies and approaches of participating Federal R&D agencies will be presented.

John C. Toole
National Coordination Office for Computing, Information, and Communications

9:30 AM-11:30 AM

Nicollet D2

MS1

Trends in Languages and Compilers for Parallel Computing

This minisymposium will focus on recent accomplishments and trends in languages and compilers for parallel computing. This area is clearly of great importance in the development of applications because the language, programming paradigm, and compilation techniques have an impact on development costs as well as on the final performance of applications. The speakers in this minisymposium will discuss issues in compiling for parallel computing, including automatic parallelization and compiler techniques for explicitly parallel programs, software engineering for parallel computing and the tools necessary to support program development and maintenance, and how parallel programming languages may evolve given the increasing complexity of the parallel architectures of the future.

Organizer: David A. Padua
University of Illinois, Urbana-Champaign

9:30 Parallel Software Engineering
David J. Kuck, Kuck and Associates, Inc.

10:00 Research and Practice of Compilers for Parallel Machines
Monica Lam, Stanford University

10:30 Parallel Programming Paradigms for Future Architectures
Piyush Mehrotra, ICASE, NASA Langley Research Center

11:00 Compiling Explicitly Parallel Programs
Katherine Yelick, University of California, Berkeley

9:30 AM-11:30 AM

Nicollet D3

MS2

Graph Partitioning: Parallel Algorithms and Applications

Graph partitioning is a fundamental problem in several scientific and engineering applications, such as VLSI design, task partitioning for parallel processing, and iterative and direct sparse linear solvers. Researchers have been following several approaches that vary in terms of their scope of application, computation cost, parallelizability, and partitioning quality. In this minisymposium, the speakers will discuss parallel algorithms and parallel applications of graph partitioning.

Organizer: Anshul Gupta
IBM T. J. Watson Research Center

9:30 A Local Graph Partitioning Heuristic Meeting Bisection Bounds
Burkhard Monien and Ralf Diekmann, University of Paderborn, Germany

10:00 A Coarse-Grain Parallel Formulation of Multilevel k-way Graph Partitioning Algorithm
George Karypis, University of Minnesota, Minneapolis

10:30 MLB: Multilevel Load Balancing for Structured Grid Applications
Daniel J. Quinlan and Markus Berndt, Los Alamos National Laboratory

11:00 Spreading Metric Based Graph Partitioning Algorithms
Guy Even, University des Saarlandes, Germany; Joseph (Seffi) Naor, Technion, Israel; Satish Rao, NEC Research; and *Baruch Schieber*, IBM T. J. Watson Research Center

9:30 AM-11:30 AM

Greenway I-J

MS3

Interactive High-Performance Computing

The power of high-performance computers comes in their ability to generate answers quickly, and to enable the study of large and complex problems. Interactive simulation environments can allow for real-time manipulation of model parameters, rapid prototyping of algorithms and applications, and the use of human pattern recognition in simulation.

In this minisymposium, the speakers will consider practical means of obtaining such interative control and visualization in high-performance computing environments. The speakers will discuss a variety of techniques and technologies developed for this purpose with examples of their applications.

Organizer: Anne E. Trefethen
Cornell Theory Center

Friday, March 14 *morning*

9:30 Building Flexible Large-Scale Scientific Computing Applications with Scripting Languages
David M. Beazley, University of Utah; and Peter S. Lomdahl, Los Alamos National Laboratory

10:00 MultiMATLAB: Extending the MATLAB Problem-Solving Environment to Multi-Processors
Anne E. Trefethen, Organizer and Lloyd Trefethen, Cornell University

10:30 Web Technologies for Collaborative Visualization and Simulation
L. Beca, G. Cheng, Geoffrey C. Fox, T. Jurga, K. Oszewski, *Marek Podgorny*, and K. Walczak, Syracuse University

11:00 Drug Design in a Virtual Environment Using the WorkSpace VR Windowing Toolkit
Richard E. Gillilan, Cornell Theory Center

9:30 AM-11:30 AM

Chair: Ahmed H. Sameh, University of Minnesota, Minneapolis

Nicollet D1

CP1
Linear Algebra

9:30 A Partitioning Scheme for the Parallel Solution of Banded Linear Systems
Michael Oettli and Ahmed H. Sameh, University of Minnesota, Minneapolis

9:50 The Stable Parallel Solution of Narrow Banded Linear Systems
Peter Arbenz, Swiss Federal Institute of Technology (ETH), Switzerland, and Markus Hegland, Australian National University, Australia

10:10 A Parallel Sparse Block Matrix Solver and Climate Data Assimilations
Chris H. Q. Ding, Lawrence Berkeley National Laboratory

10:30 Linear Algebra Subprograms on Cache-Based Shared Memory Architectures
Stefano Salvini, The Numerical Algorithms Group (NAG) Ltd, United Kingdom

10:50 Analysis of Computational Balance of Parallel Numerical Algorithms
Suraj C. Kothari, and Simanta Mitra, Iowa State University

11:10 On a Distributed Design and Implementation for a Matrix Equation
Avi Purkayastha, University of Puerto Rico, Mayaguez, and Greg Henry, University of Tennessee, Knoxville

9:30-11:30 AM

Chair: Robert F. Lucas, DARPA/ Supercomputing Research Center

Greenway A-B

CP2
Applications I

9:30 Massively Parallel Processing Boosts the Solution of Industrial Electromagnetic Problems: High Performance Out-of-Core Solution of Complex Dense Systems
Guillaume Alleon, Sabine Amram, Nicolas Durante, Philippe Homsi, and Denis Pogarieloff, Aerospatiale, France; and Charbel Farhat, University of Colorado, Boulder

9:50 Parallel Algorithm for Solving Time Convolution Equations and Application to CEM
Alain Bachelot, *Pierre Charrier*, Daniel Rouart, and Agnes Pujols, University Bordeaux I, France and CEA-CESTA, France

10:10 Large-Scale Parallel Simulation of Bioelectromagnetic Interactions
Cindy Furse, *Michael Pernice*, and Adam Tinniswood, University of Utah

10:30 Parallel Performance of A Newton-Krylov-Schwarz Solver in a 3D Numerical Simulation of the Excitation Process in the Heart
Xiao-Chuan Cai and *Maria Murillo*, University of Colorado, Boulder

10:50 Parallel Computing of Semiconductor Laser Equations
Luis F. Romero, Emilio L. Zapata, and Juan I. Ramos, University of Malaga, Spain

11:10 Distribution and Scheduling Strategies for Parallel Sparse System Solvers in Device Simulation Applications
Vladimir Menkov and *Vijay K. Naik*, IBM T. J. Watson Research Center

afternoon

12:30 PM-1:30 PM

Chair: Virginia Torczon, College of William & Mary

Greenway C-H

IP2
Parallelizing Linear and Mixed Integer Programming Using Shared Memory

We will discuss parallel implementations, primarily using shared-memory models, including results for networks of workstations. Simplicity and portability will be emphasized.

Specific topics will include general mixed integer programming, parallel barrier algorithms and dual simplex methods for linear programming, and the traveling salesman problem.

Robert E. Bixby
Rice University

1:30 PM-3:30 PM

Nicollet D2

MS4
Recent Progress in Sparse Direct Methods

The robust solution of large, sparse linear systems requires efficient direct solvers which employ some form of sparse matrix factorization (Cholesky, LU, or QR). The central idea in sparse direct solution is that of limiting "fill" (zeros in the original matrix that become nonzero in the factor) by ordering the rows and columns of the matrix. The actual numeric factorization is performed in a subsequent step using the reordered matrix.

The solution of sparse linear systems is the main computation in a vast number of scientific and engineering applications. Parallel sparse direct solution is therefore of significant interest to application developers. Furthermore, the development of parallel sparse direct solvers involves new graph algorithms and data-partitioning techniques and should be of interest to those concerned with algorithm design for parallel unstructured computation.

Organizer: Padma Raghavan
University of Tennessee, Knoxville

Friday, March 14 *afternoon*

1:30 Effective Sparse Matrix Ordering: Just Around the Bend
Bruce Hendrickson, Sandia National Laboratories, Albuquerque, and *Edward Rothberg*, Silicon Graphics, Inc.

2:00 Parallel Supernodal Method for Sparse Gaussian Elimination
Xiaoye S. Li, Lawrence Berkeley National Laboratory, James W. Demmel, University of California, Berkeley, and John R. Gilbert, Xerox Palo Alto Research Center

2:30 A Parallel Sparse Direct Solver for Least Squares Problems
Chunguang Sun, Cornell University

3:00 Sparse Direct Solution on Distributed Memory Machines
Michael T. Heath, University of Illinois, Urbana-Champaign, and *Padma Raghavan*, Organizer

---

1:30 PM-3:30 PM

Nicollet D3

MS5
Parallel N-Body Solvers

The N-body calculation consumes vast amounts of computer resources in important applications like molecular dynamics, astrophysics, materials science, electromagnetics, and fluid flow. In the past decade there has been exciting progress on this problem. Remarkably faster algorithms and software have been developed for computing pairwise interactions among large numbers of particles, typically subject to forces obeying an inverse square law. At least two important categories of algorithms have emerged: tree algorithms that divide the domain into cells and represent their far fields by (for example) multipole expansions, and grid algorithms that represent the distant interactions by piecewise polynomials. The purpose of this minisymposium is to present the latest information on the design and performance of parallel algorithms and software for the N-body problem.

Organizer: Robert D. Skeel
University of Illinois, Urbana-Champaign

1:30 Ewald and Multipole Methods for Periodic N-Body Problems
John A. Board, Jr., Christophe W. Humphres, Christophe G. Lambert, William T. Rankin, and Abdulnour Y. Toukmaji, Duke University

2:00 Particle-Mesh Based Methods for Fast Ewald Summation in Molecular Dynamics Simulations
Thomas A. Darden, National Institute of Environmental Health Sciences, Lee Pedersen, University of South Carolina, Chapel Hill, Abdulnour Toukmaji, Duke University, and Mike Crowley, Pittsburgh Supercomputing Center

2:30 Parallel, Out-of-Core Methods for Fast Evaluation of Long-range Interactions
John Salmon, California Institute of Technology, and Michael S. Warren, Los Alamos National Laboratory

3:00 A Data-Parallel Adaptive N-body Method
Y. Charlie Hu, Harvard University, S. Lennart Johnsson, University of Houston and Harvard University, and Shanghua Teng, University of Minnesota, Minneapolis

---

1:30 PM-3:30 PM

Greenway I-J

MS6
Support for Dynamic Processor Allocation on Scalable Multiprocessors

Dynamic processor allocation is a fundamental component of operating environments that aim at providing efficient resource sharing on parallel and distributed systems. The need for dynamic processor allocation is felt both by the users and by the service providers of such systems. Applications need such facilities to dynamically acquire and relinquish system resources to fit the run-time varying requirements of each job. Operating systems (resource schedulers, in particular) require the dynamic resource manipulation facilities so that a high system utilization can be maintained without degrading throughout or starving any jobs.

The speakers in this minisymposium will discuss programming interfaces for application-assisted task management, programming models for the development of applications that adapt to dynamic processor allocation, run-time support for dynamic data management and redistribution, run-time infrastructure for dynamic control of processor partitions, and mechanisms for dynamic processor scheduling and allocation.

Organizers: Vijay K. Naik and José E. Moreira
IBM T. J. Watson Research Center

1:30 Task Graph Rescheduling for Parallelization of Irregular and Dynamic Computations
Apostolos Gerasoulis, Rutgers University, and *Jia Jiao*, University of Arkansas, Little Rock

2:00 Supporting Dynamic Reconfiguration of Parallel Applications on Clusters of Non-Dedicated Workstations
Sanjeev Setia, Abdur Chowdhury, Lisa Nicklas, and Elizabeth White, George Mason University

2:30 Nanothreads: Fast, Adaptive User-Level Threads for Distributed Shared Memory
Constantine D. Polychronopoulos, University of Illinois, Urbana-Champaign

3:00 Run-Time Support for Dynamic Processor Allocation in HPF Programs
Sam Midkiff, IBM T. J. Watson Research Center, José E. Moreira and Vijay K. Naik, Organizers

---

1:30-3:30 PM

Chair: David E. Womble, Sandia National Laboratories, Albuquerque

Nicollet D1

CP3
Parallel Performance Evaluation

1:30 Design and Implementation of Computational Steering for Parallel Scientific Applications
José E. Moreira and Vijay K. Naik, IBM T. J. Watson Research Center, and David W. Fan, Columbia University

1:50 High-Performance Virtual Machines (HPVM): Clusters with Supercomputing API's and Performance
Andrew Chien, *Scott Pakin*, Mario Lauria, Matt Buchanan, Kay Hane, Louis Giannini, and Jane Prusakova, University of Illinois, Urbana-Champaign

2:10 Performance Evaluation of Large-Scale Parallel Clustering in NOW Environments
D. Judd, P. K. McKinley, and A. K. Jain, Michigan State University

2:30 Runtime Incremental Parallel Scheduling with Estimated Grain Sizes
Wei Shu and Min-You Wu, State University of New York, Buffalo

Friday, March 14 *afternoon*

2:50 New Implementations and Results for the NAS Parallel Benchmarks 2
William Saphir, National Energy Research Scientific Computing Center; Rob van der Wijngaart, MRJ Technology Solutions, NASA Ames Research Center; Alex Woo, NASA Ames Research Center; and Maurice Yarrow, Sterling Software, NASA Ames Research Center

3:10 A Parallel Algorithm to Solve Continuous Time Markov Processes that Model the Performance of Parallel Computers
Raj S. Katti, North Dakota State University, and Longfei Hu, CWC, Mankato, MN

1:30-3:30 PM

Chair: Layne T. Watson, Virginia Polytechnic Institute and State University

Greenway A-B

CP4
Parallel Numerical Methods

1:30 A Comparison of Parallel Adaptive Algorithms for Multi-Dimensional Integration
T. Len Freeman and J. Mark Bull, University of Manchester, United Kingdom

1:50 Parallel Solutions of BVPs in ODEs Based on Local Matching
Hajrudin Pasic and Yu Zhang, Ohio University

2:10 Automated Time-Parallel Computation of Adjoints for a Leapfrog Scheme
Christian Bischof and *Po-Ting Wu*, Argonne National Laboratory

2:30 Parallel Evaluation of Chebyshev Series
Roberto Barrio and J. Sabadell, University of Zaragoza, Spain

2:50 An Accurate Method for Large Least Squares Problems Involving Kronecker Products on the Connection Machine 5
Charles T. Fulton and Limin Wu, Florida Institute of Technology

3:10 Computational Experience of an Interior-Point Algorithm in a Parallel Branch-and-Cut Framework
Eva K. Lee, Columbia University; and John Mitchell, Rensselaer Polytechnic University

4:00 PM-6:00 PM

Nicollet D2

MS7
Software Tools for Developing Parallel Applications

Although a considerable amount of R&D effort is being expended on software tools for parallel computers, the application developer is often the last to know when a new tool becomes available. In this minisymposium, speakers from the parallel tools community will provide an overview of what's currently available for code development, debugging, performance tuning, and runtime setup/control. The speakers will employ a compare-and-contrast approach to help attendees establish what they should look for in tool features to meet their particular needs. The speakers will also offer suggestions on how to identify sources for trying out or acquiring new parallel tools.

Organizer: Cherri M. Pancake
Oregon State University

4:00 Performance Analysis Tools: Reaching Higher
Diane T. Rover, Michigan State University

4:30 Parallel Debugging Tools for Scientific Code Development
Jeffrey Brown, Los Alamos National Laboratory

5:00 Runtime Tools for Scientists
Al Geist, Oak Ridge National Laboratory

5:30 Development Tools for Parallel Applications
Cherri M. Pancake, Organizer

4:00 PM-6:00 PM

Nicollet D3

MS8
Parallel Linear Programming: Practical Approaches and Implementations

Linear programming problems are pervasive in science, engineering, and industry. Modern mathematical models often lead to optimization problems with many thousands of variables and with stochastic or discrete characteristics. Moreover, these modeling and optimization activities often are part of a design process that requires rapid turnaround and repeated trials. In this context, effective and robust parallel methods are essential.

Despite the enormous advances in large-scale linear programming algorithms, relatively little attention has been given to the development of practical parallel approaches.

This minisymposium will present recent work in parallel algorithms and software for large-scale linear programming: continuous, stochastic, and discrete problems. Emphasis will be on performance and implementation issues, including the important role played by parallel sparse linear algebra.

Organizers: Thomas F. Coleman, *Cornell University* **and Stephen J. Wright,** *Argonne National Laboratory*

4:00 Fully Parallel Generic Branch-and-Cut
Márta Esö, Cornell University; László Ládanyi and Theodore K. Ralphs, U.S. Air Force, Nellis Air Force Base; and *Leslie E. Trotter, Jr.*, Cornell University

4:30 Graph-Partitioning Based Sparse Matrix Orderings for Interior-Point Algorithms
Anshul Gupta, IBM T. J. Watson Research Center

5:00 Scalable Implementation of Nested Decomposition for Stochastic Programming
Alan J. King, IBM T. J. Watson Research Center

5:30 pPCx: Parallel Interior Predictor-Corrector Method for Linear Programming: Implementation and Performance
Thomas F. Coleman, Organizer; Joseph Czyzyk, Chunguang Sun, *Michael Wagner*, Cornell Theory Center, and Stephen J. Wright, Organizer

Friday, March 14 *afternoon*

4:00 PM-6:00 PM

Chair: Kristyn J. Maschhoff, Rice University

Nicollet D1

CP5
Eigenvalues

4:00 A Distributed Memory Implementation of the Nonsymmetric QR Algorithm
Jack Dongarra and *Greg Henry*, University of Tennessee, Knoxville, and *David* Watkins, Washington State University

4:20 An Efficient Parallel Implementation of a New Chasing Algorithm
Y. Deng and *Suely Oliveira*, Texas A&M University, and Z. Chen, BJ Services Company, Woodlands, TX

4:40 A Parallel Algorithm for Solving the Complex Symmetric Eigenproblem
Ilan Bar-On and *Marcin Paprzycki*, University of Texas of the Permian Basin

5:00 Parallelizing the Computation of One Eigenvalue for a Large Symmetric Tridiagonal Matrix
Jingyu Zhang and Donald Friesen, Texas A&M University

5:20 New Preconditioned Solvers for Large Sparse Eigenvalue Problems on Massively Parallel Computers
Achim Basermann and *Bernhard Steffen*, Research Centre Julich GmbH (KFA), Central Institute for Applied Mathematics (ZAM), Germany

---

4:00 PM-6:00 PM

Chair: Robert Schreiber, Hewlett-Packard Company

Greenway I-J

CP6
Parallel Languages

4:00 A Preliminary Evaluation of HPF
Scott B. Baden, University of California, San Diego, Robert Schreiber, Hewlett-Packard Company, Stephen J. Fink and Kang Su Gatlin, University of California, San Diego

4:20 Simple Parallel Extensions to Fortran 90
Robert W. Numrich and Jon L. Steidel, Cray Research Inc.

4:40 High Performance Object Oriented Scientific Programming in Fortran 90
Viktor K. Decyk, University of California, Los Angeles and Jet Propulsion Laboratory, *Charles D. Norton*, Rensselaer Polytechnic Institute and Jet Propulsion Laboratory, and Boleslaw K. Szymanski, Rensselaer Polytechnic Institute

5:00 JTpack90: A Parallel, Object-Based, Fortran 90 Linear Algebra Package
John A. Turner, Los Alamos National Laboratory, Robert C. Ferrell, Cambridge Power Computing Associates, Ltd., Brookline, MA, and Douglas B. Kothe, Los Alamos National Laboratory

5:20 Parallel Extensions to the Matrix Template Library
Andrew Lumsdaine and *Brian C. McCandless*, University of Notre Dame

5:40 ALWAN — A Coordination Language for Data Parallel Programming
Helmar Burkhart, Robert Frank, and Guido Hächler, University of Basel, Switzerland

---

4:00 PM-6:00 PM

Chair: Greg Astfalk, Convex Computer/ Hewlett-Packard Company

Greenway A-B

CP7
Applications II

4:00 Coupling Parallel Scientific and Visualization Programs to Construct a Problem Solving Environment
Peter H. Beckman, John Reynders, and Marydell Tholburn, Los Alamos National Laboratory

4:20 Automated Modeling of Parallel Algorithms for Performance Optimization and Prediction
Robert C. Durie and Adam W. Bojanczyk, *Cornell University*

4:40 Implicit Finite Element Applications: A Case for Matching the Number of Processors to the Dynamics of the Program Execution
Meenakshi A. Kandaswamy and Valerie E. Taylor, Northwestern University, and Rudolf Eigenmann and José A. B. Fortes, Purdue University

5:00 Earthquake Ground Motion Modeling in Large Heterogeneous Basins on Parallel Computers
Jacobo Bielak, Hesheng Bao, *Omar Ghattas*, Loukas Kallivokas, David O'Hallaron, Jonathan Shewchuk, and Jifeng Xu, Carnegie Mellon University

5:20 A High Resolution Finite Volume Method For Efficient Parallel Simulation of Casting Processes on Unstructured Meshes
Douglas B. Kothe and John A. Turner, Los Alamos National Laboratory, Robert C. Ferrell, Cambridge Power Computing Associates, Ltd., Brookline, MA; and S. J. Mosso, Los Alamos National Laboratory

5:40 Case Studies in Protein Structure Prediction with Real-Valued Genetic Algorithms
Charles E. Kaiser, Laurence D. Merkle, and *Gary B. Lamont*, Air Force Institute of Technology, George H. Gates, Jr., and *Ruth Pachter*, Wright Laboratory

evening

6:00 PM-8:00 PM

Lake Superior

Poster Session

The poster presentations are organized by themes to enhance interaction between presenters and attendees. There will be a reception during the session. The organizing committee will be awarding prizes for the three best poster presentations.

Simulation

Parallel Applications of a Schwarz Preconditioned Krylov Solver for Nuclear Power Plant Simulation
John Steill, *Thomas J. Downar*, and Jen-Ying Wu, Purdue University

Three Dimensional Monte Carlo Device Simulation with a Parallel Multigrid Solver
C. K. Sandalci, Ç. K. Koç, and S.M. Goodnick, Oregon State University

On the Accuracy of Anderson's Fast N-Body Algorithm
Y. Charlie Hu, Harvard University, and S. Lennart Johnsson, Harvard University and University of Houston

Parallelization and Automatic Data Distribution for Nuclear Reactor Simulations
Lorie M. Liebrock, Liebrock-Hicks Research, Calumet, MI

Partial Differential Equations

Parallel Domain Decomposition Method for Nonconforming Approximations of Elliptic Problems On Nonmatching Grids
Serguei Maliassov, University of Minnesota, Minneapolis

Parallel Processing to Solve EDP Elliptic
Xiomara Contreras, Mariela Curiel, and Angela Di Serio, Universidad Simón Bolívar, Venezuela

Parallel Implementation of a Data-Transpose Technique for the Solution of Poisson's Equation in Cylindrical Coordinates
Howard S. Cohl, Joel E. Tohline, and Xian-He Sun, Louisiana State University

Using Domain Decomposition Method and Local AMR Scheme to Solve a Class of Nonlinear Parabolic PDE
Hongyi Yu, Tulane University

Domain Decomposition Method for Solving Three-Dimensional Parabolic Differential Equations Arising in Thermal Analysis in X-Ray Lithography
Weizhong Dai, Raja Nassar, and Danqing Jiang, Louisiana Tech University

Runtime Support System for Parallel Iterative PDE Computations
Nikos P. Chrisochoides, Cornell University

Friday, March 14 *evening*

Performance and Benchmarks

A Real-Time Parallel Benchmark Suite
Brian VanVoorst, *Luiz Pires,* and Rakesh Jha, Honeywell Technology Center, Minneapolis

A Framework for Performance Prediction of Parallel Systems Based on Workload Similarity
Abdullah I. Meajil and Tarek El-Ghazawi, George Washington University

Numerical Aerodynamic Simulation Parallel Benchmarks: A Characterization Study
Abdullah I. Meajil, George Washington University

The Computational Action Norm and the Principle of Computational Least Action
Robert W. Numrich, Cray Research Inc.

An Iterative Algorithm Using Probabilistic Automata for Predicting the Performance of Parallel Computers
Raj S. Katti and Huaan Zhang, North Dakota State University

Compile-Time Partitioning of Three-Dimensional Iteration Spaces
Rizos Sakellariou, University of Manchester, United Kingdom

Iterative Methods

A Parallel Preconditioned Bi-Conjugate Gradient Algorithm for Two-Dimensional Elliptic and Parabolic Equations Using Hermite Collocation
Stephen H. Brill and George F. Pinder, University of Vermont

Preconditioned Conjugate Gradient Methods for Manufacturing Systems
Wai Ki Ching, Hong Kong Polytechnic University, Hong Kong and Chinese University of Hong Kong, Hong Kong

Construction of a Parallel GMRES(k) Solver
Elise de Doncker and Ajay Gupta, Western Michigan University

Scalability Analysis of CGLS Algorithm for Sparse Least Squares Problems on Massively Distributed Memory Computers
Tianruo Yang, Linköping University, Sweden

Data Distribution Analysis of MCGLS for Parallel Least Squares Problems
Tianruo Yang, Linköping University, Sweden

Accelerating Iterative Methods by Inducing Sparsity in the Residual Vector
Todd Goehring and Daniel Boley, University of Minnesota, Minneaspolis

Nullspace Projected Arnoldi Method for Finding Stationary Values of Constrained Quadratic Ratios and Parallel Implementation
Chang Peng and Daniel Boley, University of Minnesota, Minneaspolis

Linear Algebra and Applications

A Comparison of the Solution of the Symmetric Eigenvalue Problem with ScaLAPACK and Jacobi Methods
Domingo Giménez, University of Murcia, Spain

Tradeoffs and Performance Results of the Banded PRISM Eigensolver
Christian Bischof and William George, Argonne National Laboratory; Steven Huss-Lederman, University of Wisconsin, Madison; Xiaobai Sun, Duke University; Anna Tsao and Tom Turnbull, Center for Computing Sciences, Bowie, MD; and *Yuan-Jye Jason Wu,* Argonne National Laboratory

Parallel Threshold-Based ILU Factorization
George Karypis and Vipin Kumar, University of Minnesota, Minneapolis

Orthogonal Reduction of Dense Matrices to Bidiagonal Form on Computers with Distributed Memory Architectures
Sergey Kuznetsov, University of Minnesota, Minneapolis

New Parallel Solvers for Tridiagonal Equations
Ying Hou, XiaoMei Li, Changsha Institute of Technology, People's Republic of China

A Limited-Memory Parallel Incomplete LU Factorization
Ali Bouaricha and *Mounir Hahad,* Silvaco International, Santa Clara

Optimal Parallel Algorithms for Matrix Multiplication
Eunice E. Santos, Lehigh University

State Space Modeling and High Performance Computing of Multivariate Time Series
Celso Pascoli Bottura, *Gilmar Barreto,* and Maurício José Bordon, Universidade Estadual de Campinas, Brasil, and Jose Tarcisio Costa Filho, Universidade Federal do Maranhão, Brazil

Computational Fluid Dynamics

Interactive Visualization and Steering of Parallel CFD Methods
B. David Semeraro, Phil Papadopoulous, and James Kohl, Oak Ridge National Laboratory

Solution of Multiobjective MDO Using Parallel Genetic Algorithm
Raino A. E. Mäkinen and Jari Toivanen, University of Jyväskylä, Finland

Parallel Domain Decomposition Methods in Fluid Models with Monte Carlo Transport
Henry J. Alme and Garry H. Rodrigue, University of California, Davis and Lawrence Livermore National Laboratory, and George B. Zimmerman, Lawrence Livermore National Laboratory

Discrete Evolution of 2D Turbulence with Injection on Cray T3D
Françoise Bataille, Centre de Thermique de Lyon, INSA Lyon, France, and *Marc Daumas,* Laboratoire de l'Informatique du Parallélisme, ENS Lyon, France

Message Passing and Networking

PVMPI Provides Interoperability Between MPI Implementations
Jack J. Dongarra, Oak Ridge National Laboratory and University of Tennessee, Knoxville, Graham E. Fagg, University of Tennessee, Knoxville, and *Al Geist,* Oak Ridge National Laboratory

DATIS: A Data Transfer Interface for SPMD Programs of Scientific Computing
Xiang Yu, Princeton University, and Xiaoge Wang, Tsinghua University, People's Republic of China

Wide-Area ATM Networking for Large-Scale MPPs
Philip M. Papadopoulos and Al Geist, Oak Ridge National Laboratory

Communication Performance Modeling and Its Effect on Parallel Computation Performance of Networked Parallel Computing Environment
Meng Jie and Li Sanli, Tsinghua University, People's Republic of China

Software Tools, Environments and Libraries

CUMULVS: Collaborative Infrastructure for Developing Distributed Simulations
James Arthur Kohl, Philip M. Papadopoulos, and Al Geist, Oak Ridge National Laboratory

Step by Step Transformation of a Fortran 90 Program in HPF, using HBF Builder
Jean-Luc Dekeyser and Christian Lefebvre, University of Lille, France

OVERTURE: An Object-Oriented Software System for Solving Partial Differential Equations in Serial and Parallel Environments
David L. Brown, Geoffrey S. Chesshire, William D. Henshaw, and Daniel J. Quinlan, Los Alamos National Laboratory

A Further Proposal for a Fortran-90 Interface for LAPACK
L. Susan Blackford and Jack J. Dongarra, University of Tennessee, Knoxville; *Jeremy Du Croz and Sven* Hammarling, Numerical Algorithms Group Ltd., United Kingdom; and Jerzy Wásniewski, Technical University of Denmark, Denmark

SOLAR, a Portable Library for Scalable Out-of-Core Linear Algebra Computations
Sivan Toledo and Fred G. Gustavson, IBM T. J. Watson Research Center

evening

PINEAPL: A European Project to Develop Parallel Numerical Library Software for Industrial Applications
Stef Salvini and Sven Hammarling, The Numerical Algorithms Group Ltd, United Kingdom, and Thomas Christensen, Math-Tech ApS, Denmark

Compilers and Languages

Evaluation of High Performance Fortran Compilers in the Implementation of the Shock Tube Problem
Kevin P. Roe and Piyush Mehrotra, ICASE, NASA Langley Research Center

MC++: Parallel, Portable, Monte Carlo Neutron Transport Code in C++
Stephen R. Lee and Julian C. Cummings, Los Alamos National Laboratory, and Steven D. Nolen, Texas A&M University and Los Alamos National Laboratory

Tip C++: A Parallel Programming Model Based on The Abstract Structural Shared Memory
Tong Sun, Sanli Li, and Guilin Zhou, Tsinghua University, People's Republic of China; and Yanheng Zheng, Xianghai University, People's Republic of China

A Data-Parallel Extension to Split-C
Joan D. Lukas, Kenneth Newman, and *John P. Sullivan*, University of Massachusetts

Parallel Algorithms and Applications

Efficient Parallel FFTs for Different Computational Models
Nadia Shalaby, Harvard University

A Practical Parallel Algorithm for Delaunay Triangulation
Jeff Griffith and Mark T. Jones, University of Tennessee, Knoxville, and Paul E. Plassmann, Argonne National Laboratory

Dynamic Load Balancing With Geometric Locality
Bruce Hendrickson and Steve Plimpton, Sandia National Laboratories, Albuquerque

Parallel Computation of the Minimal Crossing Number of a Graph
Umid Tadjiev and *Frederick C. Harris, Jr.*, University of Nevada

Parallel Algorithms for Interval Polynomial Interpolation
Chenyi Hu, University of Houston-Downtown

Efficient Parallel Computing of a Longest Common Subsequence of Three Strings
Jean Frédéric Myoupo and Guillaume Luce, Université de Picardie-Jules Verne, France

Parallelization of SNePS — A Semantic Network Processing System
Chain-Wu Lee and *Min-You Wu*, State University of New York, Buffalo

Saturday, March 15

morning

8:00 AM-9:00 AM

Chair: Charles H. Koelbel, Rice University

Greenway C-H

IP3

High-Performance Computing in Biomolecular Simulation

High-performance computation offers both challenges and opportunities for improved modeling in structural biology. Numerous standard codes are now available on a wide range of platforms. These codes are being used to do simulations of systems that are an order of magnitude larger than previously possible, and for a significantly smaller cost. We describe the development of some parallel iterative techniques for solving boundary value problems for elliptic partial differential equations. We contrast implementations on distributed-memory and shared-memory scalable parallel processors. We will indicate the type of biological model that requires parallel solution on some of the largest machines available today. We also describe similar successes to parallelize existing codes for molecular dynamics and their use on biological systems. Finally, we will mention recent work parallelizing molecular imaging codes. This is allowing electron microscope data to be reconstructed to a much greater accuracy than possible before.

L. Ridgway Scott
University of Houston

9:00 AM-10:00 AM

Chair: Charles H. Koelbel, Rice University

Greenway C-H

IP4

High-Performance I/O Patterns and Policies

There is growing realization that input/output is a major impediment to high performance for an important class of grand and national challenge scientific applications. Using the Pablo environment as basis, we sketch current efforts to understand the spatial and temporal patterns of input/output present in current applications and system responses to these patterns. The results of these studies show that there are wide variations in access patterns across applications and that performance is critically sensitive to the match of application access pattern and file system policy. Building on these results, we describe the implications for parallel file systems and techniques for adaptively managing parallel input/output systems.

Daniel A. Reed
University of Illinois, Urbana-Champaign

Saturday, March 15

morning

10:30 AM-12:00 PM

Greenway C-H

Panel:

Future Trends in Software Support for Parallel Computation

The current efforts in Petaflops computing and parallel distributed computing (Metacomputing) provide new challenges for software designers and tool builders. The goal of this panel is to focus on the problems and opportunities facing software designers for platforms which support high-performance scientific applications. Prior to an interactive audience discussion, each panelist will address the following questions:

- What challenges must be addressed to develop software for Petaflops and Metacomputing platforms?

- What is needed to fill the gap between the development of software environments and tools and their wide-spread use?

- How will we know when software tools and environments meet the needs of high-performance scientific applications?

Moderator: Francine Berman
University of California, San Diego

Panelists:
Dennis Gannon, Indiana University, Bloomington

William D. Gropp, Argonne National Laboratory

David Padua, University of Illinois, Urbana-Champaign

afternoon

Cherri M. Pancake, Oregon State University

1:30 PM-3:30 PM

Lake Superior

MS9

High-Performance and Parallel I/O Systems for Scientific Computing

I/O performance has lagged significantly behind the computation and communication performance of high-performance computers. To eliminate this bottleneck, computer scientists are exploring new techniques and evaluating their effectiveness on high-performance computing testbeds nationwide. One of the most promising research developments is the design of a portable abstract-device interface for parallel I/O, which enables users to run applications portably and efficiently on a variety of high-performance systems. Also under investigation are remote I/O techniques for metacomputing systems and the use of multithreading to improve I/O performance in parallel programs. Central to these efforts is the characterization of I/O-intensive applications requirements, which are then used to guide the development of new programming language features, compiler techniques, runtime libraries, operating system support, parallel file systems, and high-performance networking software.

Organizer: Rick L. Stevens
Argonne National Laboratory

1:30 Experience with Parallel File Systems
Paul C. Messina, California Institute of Technology

2:00 Performance Modeling of a Parallel I/O System: An Application Based Approach
Evgenia Smirni, University of Illinois, Urbana-Champaign

2:30 Portable Implementation of Parallel-I/O APIs
Rajeev Thakur, Argonne National Laboratory

3:00 Where Should Collective I/O be Performed? File System or Runtime System
Alok N. Choudhary, Northwestern University

1:30 PM-3:30 PM

Greenway I-J

MS10

Resource Scheduling/ Management for Parallel Scientific Computing

Parallel processing systems make it possible to solve large, complex scientific problems that are otherwise intractable or prohibitively expensive. The scheduling policies that allocate/manage system resources among the applications submitted for execution have a significant impact on the performance of these scientific computing environments. Software support and algorithms are needed at both the system and application levels to achieve the best performance and to ease the management of system resources. The full benefits of parallel processing for scientific computing can only be realized by exploiting optimal, or near-optimal, resource scheduling methods at both the system and application levels.

The speakers in this minisymposium will present new scheduling methods and systems and novel scheduling approaches and systems that are shown to provide significant performance improvements over existing methods.

Organizer: Mark S. Squillante
IBM T. J. Watson Research Center

1:30 Modeling the Cost of Redistribution in Scheduling
Gary Shao, Rich Wolski, and *Francine Berman*, University of California, San Diego

2:00 Scheduling in a High Performance Remote-Sensing Data Server
Chialin Chang, Alan Sussman, and *Joel Saltz*, University of Maryland, College Park

2:30 Extensible Resource Scheduling for Parallel Scientific Applications
Nayeem Islam, IBM T. J. Watson Research Center, Andreas Prodromidis, Columbia University, Mark S. Squillante, Organizer, and Ajei Gopal and Liana Fong, IBM T. J. Watson Research Center

3:30 Optimal Scheduling of Coarse-Grained Parallel Scientific Applications
Mark S. Squillante, Organizer and Konstantinos Tsoukatos, University of Maryland, College Park

Saturday, March 15 *afternoon*

1:30 PM-3:30 PM

Greenway C-E

MS11

Biomolecular Modeling

This minisymposium will feature talks on parallel algorithms in the emerging field of molecular modeling, focusing specifically on biological areas. Many of the topics fall in the area commonly referred to as structural biology. Significant advances have been made in the last five years in this area. Topics at the cutting edge that have not appeared in any form before will be presented in this minisymposium. The speakers will also make an interesting introduction to some new fields of research.

Organizer: L. Ridgway Scott
University of Houston

1:30 The Parallel Fast Multipole Method for Computational Chemistry
Jürgen K. Singer, Beam Technologies Inc.

2:00 Attributes of Molecular Dynamics Calculations: Accounting for CPU Cycle
Terry W. Clark, Pacific Northwest National Laboratory

2:30 Task and Data Parallel Solution of Elliptic Systems Arising in Biological Modeling
A. Klein, University of Washington, *Mike Holst* and C. Kesselman, California Insitute of Technology

3:00 Improved 3D Reconstruction of Virus Structures Through Parallel Processing
Karen Haskell and Z.H. Zhou, University of Houston, H. Spears, Jr., Silicon Graphics, Inc., W. Chiu, Baylor College of Medicine, and L. Ridgway Scott, Organizer

---

1:30 PM-3:30 PM

Chair: Mark T. Jones, University of Tennessee, Knoxville

Greenway F-H

CP8

Adaptive Grids

1:30 The Dynamic Adaptation of Parallel Mesh-Based Computation
José G. Castaños and John E. Savage, Brown University

1:50 Load Balancing Unstructured Adaptive Grids for CFD Problems
Rupak Biswas, NASA Ames Research Center, and Leonid Oliker, RIACS, NASA Ames Research Center

2:10 Parallel Adaptive Mesh Refinement for Electronic Structure Calculations
Scott Kohn, Lawrence Livermore National Laboratory, and John Weare, Elizabeth Ong, and Scott Baden, University of California, San Diego

2:30 Parallel Object-Oriented Adaptive Mesh Refinement
Dinshaw S. Balsara, University of Illinois, Urbana, and Daniel J. Quinlan, Los Alamos National Laboratory

2:50 Worst Case Complexity of Parallel Triangular Mesh Refinement by Longest Edge Bisection
Can Özturan, Bogazici University, Turkey

3:10 A Parallel Algorithm for Mesh Smoothing
Lori A. Freitag and Paul E. Plassmann, Argonne National Laboratory, and Mark T. Jones, University of Tennessee, Knoxville

---

1:30 PM-3:30 PM

Chair: Vipin Kumar, University of Minnesota, Minneapolis

Greenway A-B

CP9

Sparse Matrix Methods

1:30 Design and Implementation of a Scalable Parallel Direct Solver for Sparse Symmetric Positive Definite Systems: Preliminary Results
Anshul Gupta and Fred Gustavson, IBM T. J. Watson Research Center, and Mahesh Joshi, George Karypis, and Vipin Kumar, University of Minnesota, Minneapolis

1:50 A Parallel Sparse Symmetric Indefinite Solver
Yogin Campbell and *Alex Pothen*, Old Dominion University

2:10 A Comparison of 1-D and 2-D Data Mapping for Sparse LU Factorization with Partial Pivoting
Cong Fu, Xiangmin Jiao, and Tao Yang, University of California, Santa Barbara

2:30 A Parallel Frontal Solver for Process Simulation
J. U. Mallya, Cray Research, Inc.; M. A. Stadtherr, University of Notre Dame; S. E. Zitney and *S. Choudhary*, Cray Research, Inc.

2:50 Sampling and Analytical Techniques for Data Distribution of Parallel Sparse Computation
Tyng-Ruey Chuang, Institute of Information Science, Academia Sinica, Taiwan, and Rong-Guey Chang and Jenq Kuen Lee, National Tsing Hua University, Taiwan

3:10 Improving Memory-System Performance of Sparse Matrix-Vector Multiplication
Sivan Toledo, IBM T. J. Watson Research Center

4:00 PM-6:00 PM

Lake Superior

MS12

Languages and Compilers for Parallel Computing

Even the most elegant parallel algorithm is useless if it cannot be efficiently implemented on real parallel hardware and applied to real applications. Parallel languages and their compilers provide the basis for these implementations. Research groups are actively developing new languages to better express parallelism, based on concepts that include data parallelism, object-oriented frameworks, and other ideas. Equally important is the development of compilers for these languages, including optimizations for efficiency and advanced runtime systems. In this minisymposium, the speakers will discuss four available languages and compare them to each other in terms of efficiency of code and ease of use.

Organizer: Charles H. Koelbel
Rice University

4:00 ZPL: Scientific Computing with Speed and Convenience
Lawrence Snyder, University of Washington

4:30 Shasta: A System for Supporting Fine-Grain Shared Memory Across Clusters
Daniel J. Scales and Kourosh Gharachorloo, Western Research Laboratory, Digital Equipment Corporation

5:00 The C* Development Environment for Scientific Programming
Michael J. Quinn, Oregon State University, and Philip J. Hatcher, University of New Hampshire

5:30 HPF/MPI: A Programming System Supporting Task and Data Parallelism
Ian Foster, David Kohr, and Rakesh Krishnaiyer, and Alok Choudhary, Argonne National Laboratory

Saturday, March 15 *afternoon*

4:00 PM-6:00 PM

Greenway C-E

MS13

Parallel Dense Linear Algebra Libraries

This minisymposium will focus on the art of high-performance parallel dense linear algebra libraries. Such libraries are used, for example, in applications involving boundary element formulations, such as electromagnetics and acoustics, as well as eigenproblems arising in computational chemistry. In addition, dense subproblems also occur in sparse linear systems. Many researchers have resorted to the development of custom implementations for individual routines. In this minisymposium, the speakers describe recent development in the area of parallel dense linear algebra libraries. They will discuss the development of general purpose parallel dense linear algebra libraries: ScaLAPACK (developed by the LAPACK project at the University of Tennessee, Oak Ridge National Laboratory, and the University of California, Berkeley) and PLAPACK (developed at the University of Texas at Austin). They will also discuss projects that target more specialized problem domains: the PRISM library and the PEigS library, both of which primarily target dense linear eigenproblems.

Organizer: Robert A. van de Geijn
University of Texas, Austin

4:00 PLAPACK: Parallel Linear Algebra Package
Robert van de Geijn, Organizer; Philip Alpatov, Greg Baker, Carter Edwards, John Gunnels, Greg Morrow, and James Overfelt, University of Texas, Austin

4:30 ScaLAPACK: A Linear Algebra Library for Message-Passing Computers
Jack Dongarra, University of Tennessee, Knoxville and Oak Ridge National Laboratory; L. S. Blackford, University of Tennessee, Knoxville; J. Choi, Soongsil University, Korea; A. Cleary, University of Tennessee, Knoxville; E. D'Azevedo, Oak Ridge National Laboratory; J. Demmel and I. Dhillon, University of California, Berkeley; S. Hammarling, The Numerical Algorithms Group, Ltd.; G. Henry, Intel; A. Petitet, University of Tennessee, Knoxville; K. Stanley, University of California, Berkeley; D. Walker, University of Wales, Cardiff; and R. C. Whaley, University of Tennessee, Knoxville

5:00 Parallel SBR: A PLAPACK Based PRISM Kernel
Yuan-Jye J. Wu and Christian H. Bischof, Argonne National Laboratory

5:30 The Performance of a New Algorithm in Eigensystem Problems For Computational Chemistry
George Fann, Pacific Northwest National Laboratory, and Inderjit Dhillon and Beresford Parlett, University of California, Berkeley

---

4:00 PM-6:00 PM

Greenway F-H

MS14

Fast Parallel Orthogonal Transforms: Theory, Implementation, and Applications

The use of Legendre functions and spherical basis functions are critical in many applications requiring computations on spherical domains. Some examples are weather, ocean, and climate modeling. Other applications can be found in antenna design and in physics. Traditionally, some of the problems have been "coerced" onto Cartesian grids because of the existence of fast computational techniques, such as the FFT, for such grids.

Until recently, algorithms for Legendre functions and spherical harmonic transforms were of arithmetic complexity O(NxN). Driscoll and Healy discovered an exact method of arithmetic complexity ($N \log_2$). The work presented in this minisymposium covers several aspects of this basic algorithm, in particular, issues of parallelization.

Organizer: S. Lennart Johnsson
University of Houston and Harvard University

4:00 Orthogonal Polynomial Transforms
David Maslen, Utrecht University, The Netherlands

4:30 Hierarchical Load-Balancing for Parallel Fast Legendre Transforms
Nadia Shalaby, Harvard University and S. Lennart Johnsson, Organizer

5:00 Applications of Fast Orthogonal Polynomial Transforms
Dan Rockmore, Dartmouth College

5:30 Parallel FFTs and FMMs in a Global Shallow Water Model
Ruediger Jakob-Chien, University of Colorado, Denver

4:00 PM-6:00 PM

Chair: Paul C. Messina, California Institute of Technology

Greenway I-J

CP10

Parallel I/O

4:00 File Access Methods on Parallel Processing
John A. Youssefi, California State University, Bakersfield

4:20 DpShell: A Data Parallel File System
Jean-Luc Dekeyser and Dominique Sueur, University of Lille, France

4:40 Experiences with the PIOUS Parallel and Distributed I/O System
Vaidy Sunderam, Emory University

5:00 Application Experience with Parallel Input/Output: Panda and the H3expresso Black Hole Simulation on the SP2
Szu-Wen Kuo, M. Winslett, Y. Chen, and Y. Cho, University of Illinois, Urbana, M. Subramaniam, Oracle Corporation, Redwood Shores, CA, and K. Seamons, Transarc Corporation, Pittsburgh, PA

5:20 I/O Optimizations for Compiling Out-of-Core Programs on Distributed Memory Machines
Mahmut T. Kandemir, Syracuse University; R. Bordawekar, California Institute of Technology, and Alok N. Choudhary, Northwestern University

5:40 An Architecture-Independent Parallel Implicit Flow Solver with Efficient I/O
Rob F. van der Wijngaart, MJR, Inc., NASA Ames Research Center, Maurice Yarrow, Sterling Software, NASA Ames Research Center, and Merritt H. Smith, Hewlett Packard Technical Consulting Laboratory

---

4:00 PM-6:00 PM

Chair: John A. Board, Duke University

Greenway A-B

CP11

Molecular Dynamics

4:00 Practical Parallelization of a Molecular Dynamics Application on Shared and Distributed Memory Machines
Masayuki Kuba, Asahi Chemical Industry, Fuji, Japan, and Marie-Christine Brunet and Constantine D. Polychronopoulos, University of Illinois, Urbana-Champaign

afternoon

4:20 Parallel MD—Simulations of Synthetic Polymers
Bernd Jung, Johannes Guttenberg Universitat, Germany, Hans-Peter Lenhof, *Peter Müller*, and Christine Rüb, Max-Planck-Institut fur Informatik, Germany

4:40 Parallel Molecular Dynamics using Truncated Octahedron Periodic Boundaries
Jeffery K. Little, AFIT/ENY

5:00 Particle—Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations
Steve Plimpton, Sandia National Laboratories, Albuquerque, Roy Pollock, Lawrence Livermore National Laboratory, and Mark Stevens, Sandia National Laboratories, Albuquerque

5:20 Parallel Particle Simulations Using the POOMA Framework
Julian C. Cummings and *William F. Humphrey*, Los Alamos National Laboratory

5:40 Automatic Differentiation of a Parallel Molecular Dynamics Application
Paul D. Hovland, University of Illinois, Urbana-Champaign, and Christian H. Bischof and Lucas J. Roh, Argonne National Laboratory

evening

6:00 PM-7:00 PM

Lake Superior

Business meeting

SIAM Activity Group on Supercomputing

Sunday, March 16

morning

Free Time for everyone.

Sunday, March 16

afternoon

1:30 PM-2:30 PM

Chair: Petter E. Bjørstad, University of Bergen, Norway

Greenway C-H

IP5

The Use of Parallel Machines for CFD Simulations

Experience with several different types of parallel machines over the last decade in performing very large simulations of turbulent compressible fluid flows will be discussed with an emphasis on the parallel computing lessons learned. Ease of use, power, and flexibility of parallel vector, distributed memory, SMP cluster, and distributed shared memory (DSM) machines and the associated programming paradigms will be contrasted. Effective techniques for "embarassingly parallel" applications are by now well understood, and sample fluid flow simulations illustrating their power in addressing fluid flow problems of scientific interest will be presented. The challenge of implementing dynamically irregular computations on parallel machines is, however, a principal focus of current work. The special potential of SMP cluster and DSM computer architectures in addressing these new challenges will be discussed.

Paul R. Woodward
University of Minnesota, Minneapolis

3:00 PM-5:00 PM

Greenway C-H

MS15

From C++ to Java: Object Oriented Methods in Parallel Processing

The speakers in this minisymposium will describe four different approaches for using object-oriented technology in parallel scientific applications. The first speaker will discuss a data parallel class library for the design of large-scale numerical simulations such as flow and particle simulations. The second speaker will describe some basic extensions to the C++ language that have been proposed as part of the High Performance C++ (HPC++) project. The third speaker will outline the design of a parallel, distributed version of the C++ standard template library and describe how scientific applications can be built using this "generic programming" style. Finally, the fourth speaker will describe new directions and opportunities in parallel scientific programming that are presented by the programming language Java and the World Wide Web.

Organizer: Dennis B. Gannon
Indiana University, Bloomington

3:00 Parallel Object-Oriented Methods and Applications
John V. W. Reynders, Los Alamos National Laboratory

3:30 CC++: High Performance C++ Part Deux
Carl Kesselman, California Institute of Technology

4:00 Programming with the HPC++ Parallel Standard Template Library
Elizabeth Johnson, Indiana University, Bloomington; and Dennis Gannon, Organizer

4:30 Java and Web Technologies for Simulation and Modeling in Computational Science and Engineering
Geoffrey Fox and Wojtek Furmanski, Syracuse University

3:00 PM-5:00 PM

Chair: Lyle N. Long, Pennsylvania State University

Lake Superior A

CP12

Computational Fluid Dynamics I

3:00 A Fluid Flow Simulation on MPP Platforms
C. F. Bender and *Moti L. Mittal*, Ohio Supercomputer Center; and J. D. Schweickart, Ohio State University

3:20 Highly Scalable 2-D and 3-D Navier-Stokes Parallel Solver on MIMD Multiprocessors
Amir Averbuch, Tel Aviv University, Israel, and Ludmil Ioffe, Moshe Israeli, and Lev Vozovoi, Technion-Israel Institute of Technology, Israel

3:40 Parallelization of Approximate Factorization Algorithms Using 2D Processor Topologies in MPI
Joseph W. Manke and James C. Patterson, The Boeing Company

Sunday, March 16 afternoon

4:00 Parallelization of a Fourier-Chebyshev Spectral Method for CFD Applications
Roger W. Hill and Kenneth S. Ball, University of Texas, Austin

4:20 Parallel Domain Decomposition Methods for the Implicit Solution of Unsteady Compressible Navier-Stokes Flows on Moving Three-Dimensional Unstructured Meshes
Xiao-Chuan Cai, Charbel Farhat, and *Marcus Sarkis*, University of Colorado, Boulder

4:40 Large Scale Industrial Applications of Parallel Computers: Optic and Fluid Dynamic Analyses Method
Daniel C. Chan and Patrick Hu, Rockwell International Corporation

---

3:00 PM-5:00 PM

Chair: Piyush Mehrotra, ICASE, NASA Langley Research Center

Greenway A-B

CP13
Parallel Compilers

3:00 Parallelization Agent: A New Approach for Parallelizing Legacy Codes
Simanta Mitra and Suraj C. Kothari, Iowa State University

3:20 A Formal Approach to Parallelizing Compilers
Teodor Rus and *Eric Van Wyk*, University of Iowa

3:00 PM-5:00 PM

Chair: Calvin Ribbens, Virginia Polytechnic Institute and State University

Mirage

CP14
Iterative Methods for Linear Systems I

3:00 Numerical Experiments with Parallel Sparse Unsymmetric Unstructured Iterative Solvers on Network of Highend Workstations
I. Konshin, A. Nikishin, *A. Yeremin*, and M. Zatsepin, Russian Academy of Sciences, Russia and Elegant Mathematics, Inc.

3:20 The Termination of Synchronous Parallel Algorithms for Iterative Systems
Thomas J. Curnock and John Ashworth, University of Salford, United Kingdom

3:40 Parallel Iterative Solution of Finite Element Systems of Equations Employing Edge-Based Data Structures
Marcos A. D. Martins, *Alvaro L. G. A. Coutinho*, and José L. D. Alves, COPPE/UFRJ, Brazil

---

3:00 PM-5:00 PM

Chair: David E. Keyes, Old Dominion University

Lake Superior B

CP15
Preconditioners

3:00 Parallel Sparse Approximate Inverse Preconditioners
Edmond Chow and Yousef Saad, University of Minnesota, Minneapolis

3:20 Tensor Sum Approximation Preconditioners
Paul Castillo and Yousef Saad, University of Minnesota, Minneapolis

3:40 A Portable MPI Implementation of the SPAI Preconditioner in ISIS++
Stephen T. Barnard, NASA Ames Research Center, and Robert Clay, Sandia National Laboratories, Livermore

---

3:00 PM-4:00 PM

Chair: George Karypis, University of Minnesota, Minneapolis

Nicollet A1-A2

CP16
Graph Partitioning

3:00 Skewed Graph Partitioning
Bruce A. Hendrickson and Robert Leland, Sandia National Laboratories, Albuquerque; and Rafael Van Driessche, Alcatel Telecom

3:20 A Data-Parallel Implementation of the Geometric Partitioning Algorithm
Y. Charlie Hu, Harvard University, *Shang-Hua Teng*, University of Minnesota, Minneapolis, and S. Lennart Johnsson, Harvard University and University of Houston

3:40 A Scalable Geometric Partitioning Heuristic
Brian Davis and *Mark T. Jones*, University of Tennessee, Knoxville, and Paul E. Plassmann, Argonne National Laboratory

evening

5:00 PM-7:00 PM

Greenway C-H

MS16
Is Message-Passing Obsolete?

In parallel computing today, an interesting and important debate is taking place. Just as the message-passing paradigm is maturing and becoming standardized through the MPI effort, new shared-memory architectures implemented in both hardware and software are providing a new programming model. Experienced scientists disagree over whether the shared-memory model, advances in compiling techniques, and data-parallel languages like HPF will combine to make the more explicit parallel algorithm approach typified by message-passing obsolete. The speakers in this minisymposium will offer their opinions and experiences. Each speaker has experience with both approaches and can make particularly useful contributions to the debate.

Organizer: Ewing L. (Rusty) Lusk
Argonne National Laboratory

5:00 Message-Passing: Evolution and Convergence
Raja Daoud, Hewlett-Packard Company

5:30 Role of Message-Passing in Performance Oriented Parallel Programming
Vipin Kumar, George Karypis, and Ananth Y. Grama, University of Minnesota, Minneapolis

6:00 Why HPF Should be the Panacea
Thomas Haupt, Geoffrey C. Fox, and Don Leskin, Syracuse University

6:30 Message-Passing, Global Addresses, and Cache Coherence: A Perspective on Coordinating Parallel Computation
Andrew A. Chien, University of Illinois, Urbana-Champaign

5:00 PM-7:00 PM

Lake Superior A

MS17
Implementing Parallel Adaptive Procedures for PDEs: Issues and Experiences

The numerical treatment of partial differential equations whose solutions exhibit complex behavior, such as moving fronts,

Sunday, March 16 *evening*

shocks, and boundary layers is extremely computationally intensive. Two important ingredients in making such problems tractable are the use of adaptive methods and parallel computing. A number of difficult implementation issues arise when these two are combined together. This minisymposium will focus on these issues, including parallel mesh generation, parallel refinement and coarsening of meshes, dynamically adapting partitions of meshes across the processing units, and ensuring that the parallel overhead is kept to a minimum.

Organizer: Peter K. Jimack
University of Leeds, United Kingdom

5:00 Dynamic Load-Balancing for Parallel Adaptive Unstructured Meshes
Chris Walshaw, Mark Cross, and Martin Everett, University of Greenwich, United Kingdom

5:30 3D Parallel Mesh Adaptivity: Data-Structures and Algorithms
Paul M. Selwood, Martin Berzins, and Peter M. Dew, University of Leeds, United Kingdom

6:00 Dynamic Load-Balancing for Adaptive PDE Solvers with Hierarchical Meshes
Nasir Touheed, University of Leeds, United Kingdom and *Peter K. Jimack*, Organizer

6:30 An Adaptive and Parallel Framework for Partial Differential Equations
Joseph E. Flaherty, Rensselaer Polytechnic Institute

5:00 PM-7:00 PM

Chair: Moti Mittal, Ohio Supercomputer Center

Greenway A-B

CP17

Computational Fluid Dynamics II

5:00 Three-Dimensional Simulations of Compressible Turbulence on High-Performance Computing Systems
Arthur A. Mirin, R. H. Cohen, W. P. Dannevik, A. M. Dimits, and D. E. Eliason, Lawrence Livermore National Laboratory, D. H. Porter, University of Minnesota, Minneapolis, O. Schilling, Lawrence Livermore National Laboratory, and P. R. Woodward, University of Minnesota, Minneapolis

5:20 Parallel Iterative Solution of the Navier-Stokes Equations with Heat and Mass Transport
John N. Shadid, Raymond S. Tuminaro, and Scott A. Hutchinson, Sandia National Laboratories, Albuquerque

5:40 Simulations of Geophysical Flows on the IBM SP2
Anqing Cui and Robert Street, Stanford University

6:00 Asynchronous Parallel Solvers for Hyperbolic Conservation Laws
Amir Averbuch, Dganit Amitai, Tel Aviv University, Israel, and Moshe Israeli, Technion-Israel Institute of Technology, Israel

6:20 Parallel Computation of Large Bubble Cavitation
Joseph Kolibal and George Gong, University of Southern Mississippi

6:40 A Parallel Method for Predicting Supercritical Fluid Transport Properties
Lyle N. Long, Obika Nwobi, and Michael M. Micci, Pennsylvania State University

---

5:00 PM-7:00 PM

Chair: Yousef Saad, University of Minnesota, Minneapolis

Mirage

CP18

Iterative Methods for Linear Systems II

5:00 PSPARSLIB : A Portable Library of Parallel Sparse Iterative Solvers
Yousef Saad, *Sergey Kuznetsov*, and Andrew Chapman, University of Minnesota, Minneapolis

5:20 LINSOL, A Parallel Iterative Linear Solver Package of Generalized CG-Type for Sparse Matrices
Willi Schoenauer, Hartmut Haefner, and Ruediger Weiss, Rechenzentrum der Universität Karlsruhe, Germany

5:40 On the Scalability of Parallel Krylov Subspace Methods
Mark Embree, University of Oxford, United Kingdom, and *Calvin Ribbens*, Virginia Polytechnic Institute and State University

6:00 Scalability of an Adaptive GMRES Algorithm
Maria Sosonkina Driver, Donald C. S. Allison, and Layne T. Watson, Virginia Politechnic Institute and State University

6:20 Improved s-Step GMRES on Massively Parallel Processors
Guangye Li, Cray Research, Inc.

5:00 PM-7:00 PM

Chair: Alan Edelman, Massachusetts Institute of Technology

Lake Superior B

CP19

Fast Algorithms

5:00 The Future Fast Fourier Transform?
Alan Edelman and *Peter McCorquodale*, Massachusetts Institute of Technology, and Sivan Toledo, IBM T. J. Watson Research Center

5:20 Parallel Prime Edge-Length Multidimensional Symmetric FFTs
Jaime Seguel and *Johanna Ortiz*, University of Puerto Rico

5:40 A Parallel Fast Direct Solver for the Discrete Solution of Separable Elliptic Equations
Tuomo Rossi and *Jari Toivanen*, University of Jyväskylä, Finland

6:00 Fast Solvers for Parallel Adaptive hp Finite Element Methods
Abani Patra, State University of New York, Buffalo

6:20 Scalability of the Fast Multipole Method for the Helmholtz Equation
Vladimir Rokhlin, Yale University; and *Mark A. Stalzer*, Hughes Research Laboratories, Malibu

6:40 An Efficient Load Balancing Technique for Parallel FMA in Message Passing Environment
Eric Jui-Lin Lu and *Daniel I. Okunbor*, University of Missouri, Rolla

Monday, March 17 *morning*

8:00 AM-9:00 AM

Chair: Greg Astfalk, Convex Computer/Hewlett-Packard Company

Greenway C-H

IP6

Future Trends in High-Performance Computing Architectures

The last three years has seen a breathtaking restructuring of the HPC marketplace, driven by the victory of the microprocessor. In November, 1993, none of the Top 500 fastest installed computers in the world were from IBM, Hewlett-Packard, or Silicon Graphics. Instead almost all were from traditional vector SMP supercomputer manufacturers (Cray Research, Convex Computer) or MPP supercomputer companies (Intel Supercomputing, Thinking Machines Corporation) or their Japanese counterparts. Today, there is no standalone supercomputer company left and machines from SGI (having bought Cray Research), H-P (having bought Convex), and IBM (having brought out the SP-2) comprise over 50% of the Top 500 machines. Thus, we can now look forward to a market-driven battle for supremacy in the HPC market by financially stable companies with a "desktop to teraflop" product line. The Big Three (SGI, H-P, and IBM) all develop their own microprocessors, so there is also a competitive battle between MIPS, PA-RISC, and the Power Series of microprocessors. The trends we will discuss in my talk and the panel are 1) the move towards shared memory (either SMP or distributed shared memory) architectures, 2) the clustering of shared memory machines for scalability, and 3) the efficiency for the end user of writing to these machines with message passing vs. shared memory compiler paradigms. Two side issues are 1) the possible emergence of PC commodity operating systems (e.g., Windows NT) and microprocessors (e.g., Intel) as a new driving force for scalable computing and 2) whether innovative architectures and processor designs (e.g., Tera Computer) can stand against the overwhelming market forces of the Big Three.

Larry Smarr
University of Illinois, Urbana-Champaign

9:00 AM-10:30 AM

Greenway C-H

Panel:

Future Trends in Computer Architecture

Each of the panelists will offer a look into the future of their scalable systems. They will describe the evolving architectures and discuss why their companies have made the choices they have to respond to market demand. A discussion will focus on the merits of distributed shared memory vs. Clustered SMPs or other shared memory models.

Moderator: Larry Smarr
University of Illinois, Urbana-Champaign

Panelists:
Greg Astfalk, Convex Computer/ Hewlett-Packard Company

Forest Baskett, Silicon Graphics, Inc.

Tilak Agerwala, IBM Corporation

Burton Smith, Tera Computer

11:00 AM-1:00 PM

Nicollet D1

MS18

Problem-Solving Environments

Problem-solving enviroments (PSEs) are computer systems that provide all the computational facilities to solve a target class of problems. These facilities include solution methods, checking the problem formulation, visualizing the solution, selecting computers, etc. Ideal PSEs will make many decisions for users. The speakers will discuss the current possibilities for creating effective frameworks for PSEs and the use of software components to construct PSEs. PSEs within the scope of this minisymposium are targeted to scientific computing and thus can make use of the large existing inventory of libraries and software components.

Organizer: John R. Rice
Purdue University

11:00 Bond, an Environment for Heterogeneous Parallel and Distributed Computing for Structural Biology
Dan C. Marinescu, Purdue University

11:30 PSEware: A Toolkit for Building Problem-Solving Environments
James C. T. Pool, California Institute of Technology

12:00 A High-Performance Problem-Solving Environment for Optimization and Control of Chemical and Biological Processes
Joseph E. Flaherty, Rensselaer Polytechnic Institute; Philip Gill, University of California, San Diego; Ephraim P. Glinert, Rensselaer Polytechnic Institute; *Linda R. Petzold*, University of Minnesota, Minneapolis; J. Ben Rosen, University of Minnesota, Minneapolis; Boleslaw K. Szymanski, Rensselaer Polytechnic Institute; and Robert T. Tranquillo, University of Minnesota, Minneapolis

12:30 A MultiAgent Environment for MPSEs
Tzvetan T. Drashansky, Purdue University; *John R. Rice*, Organizer; Elias N. Houstis and Sanjiva Weerawarana, Purdue University; and Anupam Joshi, University of Missouri, Columbia

---

11:00 AM-1:00 PM

Nicollet D2

MS19

Parallel Eigenvalue Methods for Scientific and Engineering Applications

Many of the grand-challenge applications in sciences and engineering require the solution of large sparse eigenvalue problems. While these applications could benefit substantially from parallel processing, it can be noted that parallelism is making a slow impact relative to comparable technologies in sparse linear systems. Recent advances in parallel computing, such as massively parallel computers, cluster computing, and communication standards, as well as advances in eigenvalue methods, such as preconditioners, restarting, and improved robustness, are setting new computational challenges. Among these are the need for parallel algorithms to compute a large number of interior eigenvalues and the need for efficient parallel preconditioners. This minisymposium will focus on how new eigenvalue methods and parallel computing technologies are combined to solve the new and traditional types of matrix eigenvalue problems which arise in a few

Monday, March 17 *morning*

applications. It will also attempt to address scalability issues of current techniques, and discuss the incorporation of these new technologies into efficient parallel software.

Organizers: Yousef Saad and Andreas Stathopoulos
University of Minnesota, Minneapolis

11:00 Conjugate-Gradient Based Electronic Structure Calculations on the Cray T3E and SGI PowerChallenge
Bernd Pfrommer, Steven G. Louie, University of California, Berkeley, and Horst D. Simon, Lawrence Berkeley National Laboratory

11:30 Parallel ARPACK: Portable Software for the Solution of Large Scale Eigenvalue Problems on Distributed Memory Architectures
Kristyn J. Maschhoff, Rice University

12:00 Parallel Computation of Spectral Portrait by Davidson's Method
Bernard Philippe and Miloud Sadkane, IRISA, Campus de Beaulieu, France

12:30 Parallel Solution of Eigenvalue Problems in Electronic Structure Calculations
Andreas Stathopoulos and Yousef Saad, Organizers; and James R. Chelikowsky, University of Minnesota, Minneapolis

---

11:00 AM-1:00 PM

Nicollet D3

MS20
Generating Efficient Parallel Scientific Code From High-Level Descriptions

Programmer productivity is a major concern, in particular in the creation of codes of high performance for scalable architectures. High-level descriptions offer high productivity by having high expressitivity. But such descriptions are useful in high performance computation only if efficient code can be generated.

Domain-specific formal languages and visual programming languages are but two of the approaches to achieving productivity. Domain-specific languages, for example, allow the programmer to describe a program in a notation close to or identical to the one used in the discipline, but create a serious challenge for compiler technology.

The speakers in this minisymposium will present some current attempts to compile efficient parallel code for scientific applications and report on early accomplishments.

Organizer: S. Lennart Johnsson
University of Houston and Harvard University

11:00 A Kronecker Compiler for Fast Transform Algorithms
Nikos P. Pitsianis, University of Houston

11:30 Automatic Parallelization of Sparse Matrix Applications
Vladimir Kotlyar, *Keshav Pingali*, and Paul Stodghill, Cornell University

12:00 Efficient Data Structures for Sparse Iterative Methods
Mark T. Jones, University of Tennessee, Knoxville and *Paul E. Plassmann*, Argonne National Laboratory

12:30 Optimization of a Class of Multi-Dimensional Integrals on Parallel Machines
C. Lam, *P. Sadayappan*, and R. Wenger, Ohio State University, Columbus

---

11:00 AM-1:00 PM

Chair: Petter E. Bjørstad, University of Bergen, Norway

Greenway A-B

CP20
Domain Decomposition

11:00 A Parallel Non-Overlapping Schwarz Domain Decomposition Algorithm for Elliptic Differential Equations
Daoqi Yang, University of Minnesota, Minneapolis

11:20 Domain Decomposition Methods for Hyperbolic Problems Based on Additive Schwarz
Yunhai Wu, Old Dominion University, Xiao-Chuan Cai, University of Colorado, Boulder, and *David E. Keyes*, Old Dominion University

11:40 Application of Unstructured Grid Domain Decomposition Techniques to Overset Grids
Douglas C. Blake, Air Force Institute of Technology

12:00 The Full Domain Partition Approach for Parallel Multigrid on Adaptive Grids
William F. Mitchell, National Institute of Standards and Technology

12:20 Developing Portable, Parallel Unstructured Mesh Simulations
Robert C. Ferrell, Cambridge Power Computing Associates, Ltd., Brookline, MA, and Douglas B. Kothe and John A. Turner, Los Alamos National Laboratory

12:40 Exploring Structured Adaptive Mesh Refinement (SAMR) Methods with the Illinois Concert System
Bishwaroop Ganguly, Greg Bryan, Mike Norman, and Andrew Chien, University of Illinois, Urbana

---

11:00 AM-1:00 PM

Chair: John Shadid, Sandia National Laboratories, Albuquerque

Mirage Room

CP21
Applications III

11:00 A Massively Parallel Implementation of the Couple Acoustic Scattering Problem using MPI
Steven J. Newhouse and Ian C. Mathews, Imperial College, United Kingdom

11:20 Coupled Ocean/Atmosphere Modeling on High-Performance Computing Systems
P. G. Eltgroth, J. H. Bolstad, W.P. Dannevik, P. B. Duffy, A. A. Mirin, H. Wang, and M. F. Wehner, Lawrence Livermore National Laboratory

11:40 Parallel Implementation of Hydrostatic MM5 (Mesoscale Model)
Youngtae Kim, Zaitao Pan, Eugene S. Takle, and Suresh C. Kothari, Iowa State University

12:00 Massively Parallel Reacting Flow Calculations for Optimization of Chemical Vapor Deposition Reactors
Andrew G. Salinger, John Shadid, Harry Moffat, Scott Hutchinson, Karen Devine, Gary Hennigan, and William Hart, Sandia National Laboratories, Albuquerque

12:20 A Highly Stable Explicit Technique for Stiff Reaction-Transport PDEs
Colin J. Aro, Lawrence Livermore National Laboratory; *Al Franz*, California State University, Hayward; and Dale Stone, Lawrence Livermore National Laboratory

12:40 Parallelization of the DSN Multigroup Neutron Transport Equation on the Cray-T3D with Craft
Francoise Coulomb, CEA/Limeil-Valenton, France

1:00 PM

Conference adjourns

Speaker Index

Speaker Index

T

V

W

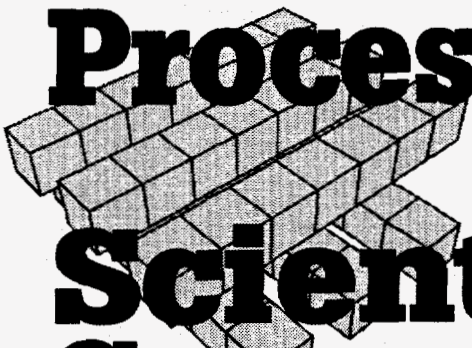
Y

Z

Abstracts

Eighth SIAM Conference on

Parallel Processing *for* Scientific Computing



March 14-17, 1997

Hyatt Regency Minneapolis on Nicollet Mall Hotel
Minneapolis, Minnesota

CP01
The Stable Parallel Solution of Narrow Banded Linear Systems

We will discuss a new method for the parallel solution of linear systems $Ax = b$ where A is a narrow banded matrix. The algorithm incorporates row interchanges for stability (pivoting) and column interchanges to enhance parallelism. The algorithm has been implemented in a portable way (Fortran, MPI) for MIMD multicomputers. Numerical experiments executed on a number of parallel computers (Intel Paragon, HP Exemplar, SGI Power Challenge) demonstrate the good scaling behavior of the algorithm.

Peter Arbenz
Swiss Federal Institute of Technology (ETH)
Institute of Scientific Computing
8092 Zurich
Switzerland
arbenz@inf.ethz.ch

Markus Hegland
Computer Sci. Lab., RSISE
Australian National University
Canberra ACT 0200
Australia
Markus.Hegland@anu.edu.au

CP01
A Parallel Sparse Block Matrix Solver and Climate Data Assimilations

The critical part of climate data assimilations is the solution of a linear equation of 10^5 unknowns with sparse coefficient matrix defined on irregular, dynamically-changing computational domains. By concurrent partitioning of data domains, we reformulate the matrix to sparse block matrix, which enables us to use efficient BLAS routines in contrast to the inefficient indirect addressing used in conventional sparse matrix techniques. The resulting highly efficient sparse matrix iterative solver achieves sustained 18 GFLOPS (single precision) performance on 512-node Paragon and 13GFLOPS (double precision) on 512-PE T3D. The entire data assimilation problem solution time is shortened by a factor of 100 on Paragon and 200 on T3D respectively over a one-head C90.

Chris H.Q. Ding
National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
Email: cding@nersc.gov

CP01
An Analytical Model to Determine Computation Distribution for Parallel Cholesky Factorization

The distribution of computations is an important factor that affects the performance of parallel algorithms. In this paper, We first present an approach to formalize the problem, within the framework of the BSP model, for SPMD algorithms in general. Then, we apply the approach to the specific case of parallel Cholesky factorization algorithm. The pattern of distribution of computations can be complex and may keep changing during the execution. This is the case with the parallel Cholesky factorization and the analytical determination is a non-trivial problem. Using the formal approach, we reformulate the problem to a number theoretic problem and solve it using the partition functions from number theory. Finally, we discuss why it is important to study the distribution of computations and how it is different from the traditional notion of load balance.

Suraj C. Kothari and Simanta Mitra
Department of Computer Science
Iowa State University, Ames, IA 50011
Email: kothari@cs.iastate.edu, mitra@cs.iastate.edu

CP01
A Partitioning Scheme for the Parallel Solution of Banded Linear Systems

A row partitioning approach for general, nonsingular narrow-banded linear systems is described which yields a naturally parallel iterative algorithm suitable for parallel computers. The system matrix is partitioned into a series of overlapping smaller matrices in which the overlaps involve only a small set of the global unknowns. The iterative solution of an implicit system for this small set of unknowns requires the solution of independent small systems in each step and yields the solution of the original system with almost perfect parallelism.

Michael Oettli
University of Minnesota
Department of Computer Science
200 Union Street SE
Minneapolis, MN 55455
Email: oettli@cs.umn.edu

Ahmed Sameh
University of Minnesota
Department of Computer Science
200 Union Street SE
Minneapolis, MN 55455

CP01
On a Distributed Design and Implementation for a Matrix Equation

We present an efficient design and implementation for the solution of the Sylvester Observer matrix equation on a distributed memory computer. It consists of two main computational intensive segments, and one of a lesser complexity: the reduction of the system matrix A to a Hessenberg form, the solution of n shifted Hessenberg systems and a number of matrix-matrix multiplies. The Hessenberg reduction is solved efficiently by using appropriate routines from the SCALAPACK package while the system solves are designed using a pipelined scheme on distributed memory computers and improved data reuse. We provide timings of our implementation on a distributed memory hierarchy computer which demonstrates both speed and efficiency of the proposed algorithm.

Avi Purkayastha
Department of Mathematics
University of Puerto Rico, Mayagüez, PR-00681-5000
E-mail: avijit@jacobi.upr.clu.edu
The work of this author was partially supported by a grant from the NSF under contract HRD-9450448.

Greg Henry
Computer Science Department
University of Tennessee, Knoxville, TN 37996-1301
E-mail: ghenry@cs.utk.edu
The work of this author was supported by the Office of

Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

CP01
Linear Algebra Subprograms on Cache-Based Shared Memory Architectures

On cache-based SMP parallel systems, the LAPACK approach, based on lower level parallelism (BLAS), can lead only to moderate levels of efficiency. In this paper I will comment on the reasons for this and show how an alternative approach which includes explicit parallelism within the top-level routines and relies on the use of single-processor BLAS-types routines achieves much greater parallel efficiency on these platforms. I will describe new subprograms for Cholesky, LU and QR factorizations, and for symmetric eigenproblems.

Stefano Salvini
Numerical Libraries Division
The Numerical Algorithms Grouip (NAG) Ltd
Wilkinson House, Jordan Hill Road
Oxford OX2 8DR, United Kingdom
E-mail: stef@nag.co.uk

CP02
Massively Parallel Processing Boosts the Solution of Industrial Electromagnetic Problems: High Performance Out-of-Core Solution of Complex Dense Systems

In this talk, we will briefly summarize the discretization method chosen for the problem of interest. Then, we will motivate the out-of-core solution of complex dense systems of equations with multiple right hand sides, and discuss the implementation and optimization of a massively parallel direct solver with checkpointing features. We will describe and comment on two real-life electromagnetic applications for which our solver has reached 50 Gigaflops on an IBM=SP2 system, which highlights the potential of massively parallel processors for electromagnetic applications.

Guillaume Alleon, Sabine Amram, Nicolas Durante,
Philippe Homsi and Denis Pogarieloff
Aerospatiale
Centre Commun de Recherches
12, rue Pasteur
92152 Suresnes Cedex, France
guillaume.alleon@siege.aerospatiale.fr

Charbel Farhat
Department of Aerospace Engineering Sciences
University of Colorado, Boulder
Boulder, CO 80309-0429
charbel@boulder.colorado.edu

CP02
Parallel Algorithm for Solving Time Convolution Equations and Application to CEM

Boundary Integral Equation Methods, when used to study transient problems, require to solve a marching-in-time scheme which is the discrete equivalent of the convolution character of the continuous integral operator. Here we are interested in new issues for solving such schemes on parallel computers. We present and analyse a parallel algorithm and consider applications to Computatioal Electromagnetics. Results of numerical simulations on the 128-processor CRAY T3D are provided.

Alain Bachelot, Pierre Charrier, Daniel Rouart and Agnes Pujols
Applied Mathematics Department
University Bordeaux l
33405 TALENCE Cedex, FRANCE
e-mail charrier@math.u-bordeaux.fr

and
Service Informatique Scientifique
CEA-CESTA 33114 LE BARP, FRANCE

CP02
Parallel Performance of A Newton-Krylov-Schwarz Solver in A 3D Numerical Simulation of the Excitation Process in the Heart

We report our numerical experiments of the Newton-Krylov-Schwarz method for solving the three dimensional, unsteady, nonlinear FitzHugh-Nagumo equation, which models the excitation process in the heart. To avoid small steps we use a fully implicit time discretization scheme, and the resulting large nonlinear algebraic system is solved with a Newton based algorithm at each time step. The performance is obtained by using PETSc of the Argonne National Laboratory on a cluster of DEC workstations.

Xiao-Chuan Cai and Maria Murillo
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309
Email: cai@cs.colorado.edu, murillo@cs.colorado.edu

CP02
Distribution and Scheduling Strategies for Parallel Sparse System Solvers in Device Simulation Applications

We present performance analysis of three partitioning and scheduling strategies that are suitable for the parallel solution of sparse triangular systems arising in the iterative solution of finite-element problems. Partitioning schemes based on RSB and RSB with vertex separators are considered. To improve parallelism, variables are reordered within each partition. Problem and architecture specific parameters are used to predict an order for the data availability on each processor. This ordering is used for prescheduling the computation and communication steps on processors. Performance results are presented from a 32-processor IBM SP2.

Vladimir Meñkov and Vijay K. Naik
IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598
s5menkov@watson.ibm.com
vkn@watson.ibm.com

CP02
Large-Scale Parallel Simulation of Bioelectromagnetic Interactions

Finite Difference Time Domain (FDTD) is the most popular method for modelling bioelectromagnetic interactions. Its large storage requirements limit its use for complex three-dimensional problems. FDTD is easy to parallelize on distributed memory multiprocessors, but important de-

tails, including I/O of large datasets, calculation of global parameters, and post-processing, remain a challenge. We discuss an MPI-based conversion of an FDTD code that addresses these issues. Performance evaluation on an IBM SP-2 and results from large-scale simulations are also presented.

Cindy Furse
Department of Electrical Engineering
University of Utah, Salt Lake City, UT 84112

Michael Pernice
Center for High Performance Computing
University of Utah, Salt Lake City, UT 84112
E-mail: usimap@sneffels.usi.utah.edu

Adam Tinniswood
Department of Electrical Engineering
University of Utah, Salt Lake City, UT 84112

CP02
Parallel Computing of Semiconductor Laser Equations

Parallel Hopscotch's and implicit, linearized θ-methods have been developed to solve the two–dimensional equations of multi–striped semiconductor lasers. For the latter, a BiCGSTAB algorithm has been used and, in both methods, an efficient technique which overlaps the communications and the computations has been developed. Such a technique also reduces the number of messages to be passed to other processors and yields an almost perfect speed–up for the Hopscotch's method which is much more efficient than implicit, linearized techniques. It is also shown that the speed–up of the latter increases almost linearly as the number of processors increases.

Luis F. Romero, Emilio L. Zapata, Juan I. Ramos
Departamento de Arquitectura de Computadores
E. T. S. Ingenieros Industriales
University of Málaga
Spain
Email: feliperg@ac.uma.es

CP03
Performance Evaluation of Large-Scale Parallel Clustering in NOW Environments

This paper presents the results of a performance study of parallel data clustering on Network of Workstations (NOW) platforms. The clustering program, P-CLUSTER, is based on the mean square-error clustering algorithm and is applied to the problem of image segmentation. Results of experiments on four NOW platforms are presented, illustrating the effects on performance of different processors, network architectures, communication packages, and latency hiding techniques.

Dan Judd, Philip K. McKinley, and Anil K. Jain
Department of Computer Science
Michigan State University
East Lansing, MI 48824
E-mail: mckinley@cps.msu.edu

CP03
A Parallel Algorithm to Solve Continuous Time Markov Processes that Model the Performance of Parallel Computers

We present a parallel algorithm that solves the following system of linear equations: $\pi Q = 0$, $\pi e = 1$, where Q is the transition rate matrix of a continuous time Markov process, π is the vector of steady state probabilities of the Markov process and e is a column vector of all 1's. The algorithm presented is useful when the Markov process has a large number of states and the matrix Q is not sparse. This is the case when modeling the performance of complex parallel computing systems. The algorithm presented converts the matrix Q into a set of sub-matrices, each representing a smaller Markov process. This set of smaller Markov processes can be solved in parallel to obtain the desired solution. The complexity of our algorithm is $O((n/m)^3 + m^3)$, where m is the number of smaller Markov processes and n the number of states in the original Markov process. Other algorithms for solving such problems have a complexity of $O(n^3)$.

Raj S. Katti
Department of Electrical Engineering
North Dakota State University, Fargo, ND 58105
Email: katti@plains.nodak.edu

Longfei Hu
CWC
1983 Premier Drive, P.O. Box 4459
Mankato, MN 56002-4459

CP03
Design and Implementation of Computational Steering for Parallel Scientific Applications

Computational steering facilities allow users to interactively monitor and control the progress of their applications. In this paper we discuss how we designed and implemented these facilities in the context of the Distributed Resource Management System (DRMS). DRMS supports computational steering of parallel applications through a programming model based on user-defined *schedulable and observable points* (SOPs). At an SOP, a previously declared set of *controllable and observable variables* can be examined and modified, data distributions can be changed, and a snapshot of the application can be taken. The user can also roll-back the application to any previously taken snapshot. *What-if* analysis of applications is well supported by this environment.

José E. Moreira and Vijay K. Naik
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
moreira@watson.ibm.com
vkn@watson.ibm.com

Wei Fan
Department of Computer Science
Columbia University
New York, NY 10027
wfan@cs.columbia.edu

CP03
High Performance Virtual Machines (HPVM's): Clusters with Supercomputing API's and Performance

The HPVM project provides software which enables high performance computing on clusters of PC's and workstations using standard supercomputing API's such as MPI,

Shmem put/get, Global Arrays, and several more are in progress. HPVM's are surprisingly competitive with MPP systems such as the IBM SP2 and Cray T3D. For example, we have demonstrated message passing latencies of 19 microseconds and 20MByte/s for small packets (< 1 kilobyte) across the cluster using MPI-FM. Elements of the HPVM project are also known as Illinois Fast Messages, MPI-FM, FM-DCS (Dynamic Coscheduling), Put/Get-FM and Global Arrays-FM. We have dozens of active user sites. More information is available from
`http://www-csag.cs.uiuc.edu/projects/cluster.html`

Andrew Chien, Scott Pakin, Mario Lauria, Matt Buchanan,
Kay Hane, Louis Giannini, and Jane Prusakova
Department of Computer Science
and National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, IL 61801
achien@cs.uiuc.edu, pakin@cs.uiuc.edu

CP03
New Implementations and Results for the NAS Parallel Benchmarks 2

The NAS Parallel Benchmarks 2 (NPB 2) provide a well-calibrated measurement of the real-world performance of parallel computers. We describe MPI C/Fortran 77 implementations of IS, EP, and FT, heretofore unreleased codes in the NPB 2 suite. Performance results for a wide range of machines and processors include first results for the new IBM SP-SC, the SGI Origin 2000, the HP Exemplar-S, the Hitachi SR2201, and the Sun Ultra Enterprise 4000.

Bill Saphir
National Energy Research Scientific Computing Center (NERSC)

Rob Van der Wijngaart
MRJ Technology Solutions, NASA Ames Research Center

Alex Woo
NASA Ames Research Center

Maurice Yarrow
Sterling Software, NASA Ames Research Center

Mail Stop T27A-1
Moffett Field, CA 94035
npb@nas.nasa.gov

CP03
Runtime Incremental Parallel Scheduling with Estimated Grain Sizes

Runtime Incremental Parallel Scheduling (RIPS) is a newly developed load-balancing algorithm. RIPS assumed equal grain sizes of tasks for simplicity. Its performance is questionable for tasks that have a large variety of grain sizes. Presented in this paper is a RIPS algorithm with estimated grain sizes. In this algorithm, each task is assigned a weight which is its estimated execution time. The estimation is obtained by taking the average of the execution time for the same type of tasks. A comparison of both algorithms is given.

Wei Shu and Min-You Wu
Department of Computer Science
State University of New York at Buffalo
Buffalo, NY 14260
shu,wu@cs.buffalo.edu

CP04
Parallel Evaluation of Chebyshev Series

We present a simple parallel algorithm for the evaluation of Chebyshev series. These series appear in some fields of Mathematics, Physics and Engineering due to their good approximation features. Comparisons of its efficiency on a CRAY T3D are shown for various numbers of processors (a maximum of 512) and using both message passing (MPI) and data parallel environment (HPF).

Roberto Barrio
GME, Department of Applied Mathematics,
Centro Politécnico Superior
University of Zaragoza,
María de Luna 3,
50015–E Zaragoza, SPAIN
E-mail: rbarrio@mcps.unizar.es

J. Sabadell
Department of Theoretical Physics,
Edificio de Matemáticas,
University of Zaragoza,
Pedro Cerbuna 12 ,
50009–E Zaragoza, SPAIN

CP04
A Comparison of Parallel Adaptive Algorithms for Multi-dimensional Integration

A central feature of adaptive algorithms for the numerical approximation of definite integrals is the list containing the sub-intervals and corresponding error estimates. Fundamentally different parallel algorithms result depending on whether the list is maintained as a single shared data structure accessible to all processors, or else as the union of non-overlapping sublists, each private to a processor. We describe several variants of these approaches, and compare numerical performances of the algorithms on multi-dimensional problems.

J. Mark Bull
Centre for Novel Computing
Department of Computer Science
University of Manchester
Manchester, M13 9PL, United Kingdom

T. Len Freeman
Department of Mathematics
University of Manchester
Manchester, M13 9PL, United Kingdom
E-mail: freeman@ma.man.ac.uk

CP04
Parallel Computations for Kronecker Product Least Squares Problems

Parallel implementations involving QR, LU, URV and SVD methods for solving the Kronecker Product least squares problem on the Connection machine 5 are compared with respect to timing and stability. The QR, LU and URV methods are considerably cheaper than the SVD method, but only the QR and URV methods have acceptable stability properties to justify their use on large problems. Dense linear systems involving 6.5 million variables

can be solved with the QR method using no more than 800 Megabytes of memory.

Charles T. Fulton and Limin Wu
Applied Mathematics Program
Florida Institute of Technology
Melbourne, Fl 32901
E-mail: fulton@math.fit.edu

CP04
Computational Experience of an Interior-Point Algorithm in a Parallel Branch-and-Cut Framework

We use an interior-point algorithm in a branch-and-cut framework for solving nonlinear mixed integer programs. In contrast to solving the relaxation to optimality at each tree node, the relaxation is only solved to near-optimality, and the obtained near-optimal point is then used as a reference for cutting planes as well as for guidelines on node selection. The code is run on a network of UNIX-based workstations. Computational results on various classes of nonlinear mixed integer programs will be presented.

Eva K. Lee
Industrial Engineering & Operations Research
Columbia University
New York, NY 10027
E-mail: evakylee@columbia.edu

John Mitchell
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
E-mail: mitchj@rpi.edu

CP04
Parallel Solutions of BVPs in ODEs Based on Local Matching

Presented is a domain-decomposition-based iterative method for solving boundary value problems (BVPs)in ordinary differential equations (ODEs) as multi-point BVPs in which both local (subdomain)solutions and their matching may be performed in parallel. Several second-order equations are solved for illustration.

Hajrudin Pasic and Yu Zhang
Ohio University, Athens, OH 45701
pasic@bobcat.ent.ohiou.edu

PP97 Abst# 129
CP04
Automated Time-Parallel Computation of Adjoints for a Leapfrog Scheme

The leapfrog scheme is a commonly used second-order method for solving differential equations. We show how the associativitiy of the chain rule of differential calculus can be used to (1) expose and exploit intermediate derivative sparsity arising from the typical localized nature of the timestepping operator and (2) compute adjoints in an automated and time-parallel fashion. Experimental results with a 2-D shallow water equation model on IBM SP and Sparcstation networks demonstrate the considerable algorithmic improvement *and* speedup that can thus be obtained.

Christian Bischof and Po-Ting Wu
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
{bischof,pwu}@mcs.anl.gov

CP05
A Distributed Memory Implementation of the Nonsymmetric QR Algorithm

The QR algorithm is the crux of the serial nonsymmetric eigenvalue problem. Recent efforts to parallelize this algorithm have made significant advances towards solving the parallel *nonsymmetric* eigenvalue problem. Most methods to date suffer a scalability problem. In this talk we discuss an approach for parallelizing QR which overcomes many of the disadvantages to date. We also give insights into what is necessary for a parallel algorithm to work using these strategies. Performance of a parallel implementation on the Intel Paragon^{TM} system is reported.

Jack Dongarra and and Greg Henry
Computer Science Dept., 111 Ayres Hall
University of Tennessee
Knoxville, TN 37996-1301
dongarra@cs.utk.edu
ghenry@cs.utk.edu

David Watkins
Pure and Applied Mathematics Dept.
Washington State University
Pullman, WA 99164
watkins@delta.math.wsu.edu

CP05
An Efficient Parallel Implementation of a New Chasing Algorithm

Chasing algorithms are commonly used to find eigenvalues of tridiagonal matrices. All of the old methods use a *standard chasing step*. Using the graph representation of matrices we can show that they correspond to chasing edges of the triangles representing the current entries of the matrix. We have developed a new algorithm for tridiagonalizing arrowhead matrices based on a *new chasing step*, which chases nodes instead. The new algorithm is more pipelined than the old one. We will show why, and also compare performances between the Paragon and Ncube, for the new algorithm.

Zhongming Chen
BJ Services Company
8701 New Trail Drive
Woodlands, TX 77381

Y. Deng and Suely Oliveira
Texas A&M University
College Station, TX 77843-3112
Email: suely@cs.tamu.edu

CP05
A Parallel Algorithm for Solving the Complex Symmetric Eigenproblem

Recently the problem of computing the spectrum of large complex symmetric (non-Hermitian) matrices attracted researchers working in the field of quantum reaction dynamics. A new sequential algorithm that takes advantages of the symmetry (in much of the same way as the algorithm for the Hermitian case) has been proposed by V. Ryaboy and I. Bar-On. The code has proven to be by much faster

than the general solver, and sometimes even faster than the Hermitian solver. In addition, the computed results were as reliable as for the general algorithm. The aim of the presentation is to discuss the parallel performance of the new algorithm on a Cray J-916 and a Silicon Graphics Power Challenge computers. The comparison will be made with the parallel LAPACK kernels for both the general and the Hermitian eigenproblems.

Ilan Bar-On and Marcin Paprzycki
Department of Mathematics and Computer Science
University of Texas of the Permian Basin
Odessa, TX 79762
E-mail: baron_i@utpb.edu, paprzycki_m@utpb.edu

CP05
New Preconditioned Solvers for Large Sparse Eigenvalue Problems on Massively Parallel Computers

We present preconditioned solvers to find a few extreme eigenvalues and eigenvectors of large sparse symmetric or nonsymmetric matrices based on the Jacobi-Davidson (JD) method by G. L. G. Sleijpen and H. A. van der Vorst. For preconditioning, we apply a new adaptive approach using the QMR iteration. To parallelize the solvers, we divide the considered scope of the spectrum into a few overlapping intervals and asynchronously exchange eigenvector approximations from neighbouring intervals to keep the solutions separated. Per interval, matrix-vector and vector-vector operations of the JD iteration are parallelized by determining a data distribution and a communication scheme from an automatic analysis of the sparsity pattern of the matrix. We demonstrate the efficiency of this hierarchical parallelization strategy by timings on an Intel Paragon and a Cray T3E system with matrices from real applications.

Achim Basermann and Bernhard Steffen
Research Centre Jülich GmbH (KFA)
Central Institute for Applied Mathematics (ZAM)
52425 Jülich, Germany
E-mail: A.Basermann@kfa-juelich.de
E-mail: B.Steffen@kfa-juelich.de

CP05
Parallelizing the Computation of One Eigenvalue for A Large Symmetric Tridiagonal Matrix

Based on the parallel scheme [3] by Chung and Yan, this paper presents a modified parallel algorithm to speed up the computation of one eigenvalue for a large symmetric tridiagonal matrix with the bisection method. A *scaling method* is presented to overcome the inherent underflow and overflow problems in Chung and Yan's parallel scheme. With the proper assumption, this paper presents a theorem which guarantees that underflow and overflow problems can be avoided by using the proposed scaling method. The experimental results on testing large symmetric tridiagonal matrices are included.

[3] Chung, Kuo-Liang and Wen-Ming Yan, "Solving the Symmetric Tridiagonal Eigenvalue Problem on Hypercubes", *Computers Math. Applic.*, Vol. 25, No. 9, pp. 91-96, 1993.

Jingyu Zhang and Donald Friesen
Computer Science Department
Texas A&M University
College Station, TX 77843
E-mail: jzhang@cs.tamu.edu

CP06
A Preliminary Evaluation of HPF

HPF is a data parallel Fortran dialect currently implemented on diverse commercial platforms ranging from workstations to Massively Parallel Processors. To date, performance data are sparse. We will present preliminary measurements of selected benchmarks, comparing different implementations of HPF running on the same hardware, and also equivalent explicit SPMD implementations. We will discuss optimization issues that can dramatically affect performance.

Scott B. Baden
Department of Computer Science & Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093-0114
Email: baden@cs.ucsd.edu

Robert Schreiber
HP Labs 3L-5, Hewlett-Packard Company
1501 Page Mill Rd, Palo Alto, CA 94304-1126
Email: schreiber@hpl.hp.com

Stephen J. Fink and Kang Su Gatlin
Department of Computer Science & Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093-0114
Email: sfink@ucsd.edu, kgatlin@cs.ucsd.edu

CP06
ALWAN — A Coordination Language for Data Parallel Programming

ALWAN is a parallel coordination language and programming environment. Design goals have been: improved programmability of parallel applications, performance portability, reuse of software components, and mixed-language programming. ALWAN provides a set of high-level constructs for the development of parallel program skeletons and supports the linkage to sequential building blocks written in C and FORTRAN 77. The ALWAN compiler is implemented on various UNIX platforms that offer PVM or MPI message passing support. Target machines include IBM SP2, INTEL PARAGON, CRAY T3D, and workstation clusters. Performance is close to corresponding message passing programs.

Helmar Burkhart, Robert Frank, and Guido Hächler
Informatics Department
University of Basel
Mittlere Strasse 142
CH-4056 Basel
Switzerland
email: burkhart@ifi.unibas.ch

CP06
Parallel Extensions to the Matrix Template Library

We present a template library for C++ to enable compositional construction of matrix classes suitable for high performance numerical linear algebra computations. The library based on our interface definition — the Matrix Template Library (MTL) — is written in C++ and consists of a small number of template classes that can be composed to represent commonly used matrix formats (both sparse and dense) and parallel data distributions. A comprehensive set of generic algorithms provide high performance for

all MTL matrix objects without requiring specific functionality for particular types of matrices. We present performance data to demonstrate that there is little or no performance penalty caused by the powerful MTL abstractions and show examples of how MTL can be used effectively in parallel scientific computing applications.

Andrew Lumsdaine and Brian C. McCandless
Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556
E-mail: (lumsdaine.1, mccandless.1)@nd.edu

CP06
High Performance Object Oriented Scientific Programming in Fortran 90

We illustrate how Fortran 90 supports object oriented concepts by example of plasma particle computations on the IBM SP. Our experience shows that Fortran 90 and object oriented methodology give high performance while providing a bridge from Fortran 77 legacy codes to modern programming principles. All of our object oriented Fortran 90 codes execute more quickly than the equivalent C++ versions, yet the abstraction modeling capabilities used for scientific programming are comparably powerful.

Viktor K. Decyk, (decyk@physics.ucla.edu)
Physics Deparment
University of California, Los Angeles
405 Hilgard Ave., Los Angeles, CA 90095-1547
and
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109-8099

Charles D. Norton, (nortonc@olympic.jpl.nasa.gov)
Jet Propulsion Laboratory
California Institute of Technology
High Performance Computing Systems and Applications Group
MS 168-522, 4800 Oak Grove Drive, Pasadena, California 91109-8099

Boleslaw K. Szymanski, (szymansk@cs.rpi.edu)
Department of Computer Science
Rensselaer Polytechnic Institute
110 Eighth Street, Troy, New York 12180-3590

CP06
F^{--} : Simple Parallel Extensions to Fortran 90

We describe a parallel extension to Fortran 90 that adds exactly one extension to the language, a set of square brackets containing processor coordinates. Just as local data coordinates are contained within parentheses, global processor coordinates are contained within brackets. On a distributed memory machine, a statement such as x(i,j) = y(i,j)[p,q] generates a load from remote address y(i,j) in the memory of processor [p,q] followed by a store to local address x(i,j) in the memory of the local processor. Our model is a Single-Program-Multiple-Data (SPMD) programming model. It assumes that good performance requires the programmer to understand data locality and to communicate that locality explicitly to the compiler. It provides a Fortran-like syntax to support these requirements that is simple for the programmer to understand and simple for the compiler developer to implement.

Robert W. Numrich and Jon L. Steidel
Cray Research
655 Lone Oak Drive, Eagan, MN 55121
rwn@cray.com

CP06
JTpack90: A Parallel, Object-Based, Fortran 90 Linear Algebra Package

We have developed an object-based linear algebra package, currently with emphasis on sparse Krylov methods, driven primarily by needs of the Los Alamos National Laboratory parallel unstructured-mesh casting simulation tool TELLURIDE. We describe our object-based Fortran 90 approach, which enhances maintainability, performance, and extensibility, our parallelization approach using a new portable gather/scatter library (PGSLib), current capabilities and future plans, and present performance results on a variety of platforms.

John A. Turner
Los Alamos National Laboratory, Group XTM
Box 1663, MS B226, Los Alamos, NM 87545
E-mail: turner@lanl.gov

Robert C. Ferrell
Cambridge Power Computing Associates, Ltd.
2 Still St., Brookline, MA 02146

D. B. Kothe
Los Alamos National Laboratory, Group T-3
Box 1663, MS B216, Los Alamos, NM 87545

CP07
Coupling Parallel Scientific and Visualization Programs to Construct a Problem Solving Environment

Scientists often utilize several well established parallel programs to solve problems. However, many are written as complete, monolithic solutions for a given problem domain. Our toolset for creating domain-specific problem solving environments is a Java/CORBA "workspace", and a set of scientific applications that have been slightly modified, so they can be coupled. Coupled programs require two channels of interaction, one to signal *events*, such as the completion of an iteration, and one for pushing and pulling the data, often in different formats, between the programs. We will give an exmaple of a plasma simulation interacting with spectral analysis and visualization engines.

Peter H. Beckman, John Reynders, and Marydell Tholburn
Los Alamos National Laboratory
Los Alamos, NM 87544
Email: beckman@lanl.gov

CP07
Automated Modeling of Parallel Algorithms for Performance Optimization and Prediction

We describe a tool for building parallel applications on distributed memory machines. The central component of this tool is the Parallel Program Factory (PPF). The input to the PPF is a dataflow graph describing the application on the level of functional blocks. Given the graph and a col-

lection of parallel libraries, the PPF automatically selects an efficient parallel implementation of the application. A Space Time Adaptive Processing problem developed by the signal processing community is used to demonstrate the concept.

Robert C. Durie and Adam W. Bojanczyk
Department of Electrical Engineering
Cornell University, Ithaca, NY 14853
E-mail: bobd@ee.cornell.edu

CP07
Earthquake Ground Motion Modeling in Large Heterogeneous Basins on Parallel Computers

We discuss the development of parallel algorithms and software for predicting, by computer simulation, the ground motion of large basins during strong earthquakes. The challenges include the development of numerical methods for simulating large-scale seismic wave propagation through highly heterogeneous media, absorbing boundaries that limit the computational domain, techniques for incorporating seismic excitations, mesh generation and mesh partitioning methods for very large and highly unstructured meshes, and parallel code generators that help automate the task of writing efficient portable unstructured mesh solvers for distributed memory parallel supercomputers. We have used this capability to model the seismic response of the San Fernando Valley in Southern California.

Jacobo Bielak, Hesheng Bao, Omar Ghattas
Loukas Kallivokas, David O'Hallaron, Jonathan Shewchuk and Jifeng Xu
Carnegie Mellon University
Pittsburgh, PA 15213
Email: oghattas@cs.cmu.edu

CP07
Implicit Finite Element Applications: A Case for Matching the Number of Processors to the Dynamics of the Program Execution

Generally, parallel scientific applications are executed on a fixed number of processors determined by an efficiency analysis of the application's computational kernel. In this paper, we present the results of an in-depth study *quantifying* the advantages of matching the number of processors to the parallelism profile for finite element analysis. The results indicate that the effectiveness of varying processor allocation is several orders of magnitude better than that of fixed processor allocation for problems with 10^5 to 10^9 nodes.

Meenakshi A. Kandaswamy
CIS Dept., Syracuse University
Syracuse, NY 13244
E-mail: meena@tana.ece.nwu.edu

Valerie E. Taylor
ECE Dept., Northwestern University
2145, Sheridan Road, Evanston, IL 60208

Rudolf Eigenmann and Jose' A. B. Fortes
School of ECE, 336 EE Bldg.
Purdue University, West Lafayette, IN 47907

CP07
A High Resolution Finite Volume Method for Efficient Parallel Simulation of Casting Processes on Unstructured Meshes

We discuss selected aspects of a new parallel three-dimensional (3-D) computational tool for the unstructured mesh simulation of Los Alamos National Laboratory (LANL) casting processes. This tool, known as **Telluride**, draws upon on robust, high resolution finite volume solutions of metal alloy mass, momentum, and enthalpy conservation equations to model the filling, cooling, and solidification of LANL castings. We briefly describe the current **Telluride** physical models and solution methods, then detail our parallelization strategy as implemented with Fortran 90 (F90). This strategy has yielded straightforward and efficient parallelization on distributed *and* shared memory architectures, aided in large part by new parallel libraries **JTpack90** for Krylov-subspace iterative solution methods and **PGSLib** for efficient gather/scatter operations. We illustrate our methodology and current capabilities with source code examples and parallel efficiency results for a LANL casting simulation.

Douglas B. Kothe
Los Alamos National Laboratory, Group T-3
MS B216, Los Alamos, NM 87545 (*dbk@lanl.gov*)

J. A. Turner
Los Alamos National Laboratory, Group XTM
MS B226, Los Alamos, NM 87545 (*turner@lanl.gov*)

R. C. Ferrell
Cambridge Power Computing Associates, Ltd.
2 Still St., Brookline, MA 02146 (*ferrell@cpca.com*)

S. J. Mosso
Los Alamos National Laboratory, Group XHM
MS F663, Los Alamos, NM 87545 (*sjm@lanl.gov*)

CP07
Case Studies in Protein Structure Prediction with Real-valued Genetic Algorithms

Accurate and reliable prediction of macromolecular structures has eluded researchers for nearly 40 years, primarily because exhaustive search for the minimum energy native conformer is computationally intractable. Our research in evolutionary and parallel computation for structure prediction continues with analysis of real-valued genetic algorithms using specialized operators. Effectiveness is increased by incorporating biochemistry knowledge to limit the search space. Parallel GAs featuring MPI "Farming" and "Island" models on large dimensional NOW and MPP distributed machines are presented.

Charles E. Kaiser, Laurence D. Merkle, Gary B. Lamont
Department of Electrical and Computer Engineering
Graduate School of Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433
(lmerkle, ckaiser, lamont)@afit.af.mil

George H. Gates, Jr., Ruth Pachter
Wright Laboratory
3005 P St., Ste. 1
Wright-Patterson AFB, OH 45433-7702
(gatesgh, pachterr)@ml.wpafb.af.mil

CP08
Parallel Object Oriented Adaptive Mesh Refine-

ment

In this paper we study adaptive mesh refinement for elliptic and hyperbolic systems. We use the AFAC variant of multigrid as a test case of an elliptic system. For our hyperbolic system example we use higher order Godunov methods for solving the Euler equations of gas dynamics. The object is to do this in a load balanced way in a parallel environment. Adaptive mesh refinement poses several problems for expressing the self-adaptive algorithm. We show that object oriented programming simplifies the task of expressing the self-adaptive algorithm. Parallel adaptive mesh refinement poses the further difficulty of expressing the basic solver, whether elliptic or hyperbolic, in a fashion that parallelizes seamlessly. We show that use of an array class library, P++, helps express this parallelism in a transparent way. The further challenge is to load balance the application. This has been achieved by writing a load balancer for logically rectangular mesh problems. Performance figures will be presented and discussed.

Dinshaw S. Balsara
National Center for Supercomputing Applications
Astronomy, University of Illinois, Urbana
152 C.A.B., 605 E. Springfield Ave.,
Champaign, IL 61820
Email: u10956@ncsa.uiuc.edu

Daniel J. Quinlan
Los Alamos National Laboratory
MS B265, Los Alamos, NM 87544

CP08
Load Balancing Unstructured Adaptive Grids for CFD Problems

Mesh adaption is a powerful tool for efficient unstructured-grid computations but causes load imbalance among processors on a parallel machine. A global load balancing method is presented that, after each parallel tetrahedral mesh adaption, determines whether repartitioning will be beneficial. If so, a remapping algorithm assigns new partitions to processors to minimize the redistribution cost. The new partitions are accepted only if the remapping cost is compensated by the improved load balance. Results indicate that this strategy is effective for large-scale scientific computations on distributed-memory multiprocessors.

Rupak Biswas
Mail Stop T27A-1
NASA Ames Research Center, Moffett Field, CA 94035
E-mail: rbiswas@nas.nasa.gov

Leonid Oliker
RIACS, Mail Stop T27A-1
NASA Ames Research Center, Moffett Field, CA 94035

CP08
The Dynamic Adaptation of Parallel Mesh-Based Computation

We present efficient algorithms and data structures for dynamic refinement/coarsening (adaptation) of unstructured FE meshes on loosely coupled parallel processors. We describe a) a parallel adaptation algorithm, b) an online parallel repartitioning algorithm based on mesh adaptation histories, c) an algorithm for the migration of mesh elements between processors, and d) an integrated object-oriented framework for the adaptation, repartitioning and migration of the mesh at runtime. A two-dimensional triangle-based prototype demonstrates the feasibility of these ideas.

José G. Castaños and John E. Savage
Department of Computer Science
Brown University, Providence, RI 02912-1910
(jes, jgc)@cs.brown.edu

CP08
A Parallel Algorithm for Mesh Smoothing

Maintaining good mesh quality during the generation and refinement of unstructured meshes in finite-element applications is an important aspect in obtaining accurate discretizations and well-conditioned linear systems. One cost-effective approach is mesh smoothing (changing the geometric locations of vertices in the computational domain's surface and volume) to locally optimize a relevant mesh quality measure. In this talk, we present an efficient mesh smoothing algorithm based on nonsmooth optimization techniques and a scalable implementation of this algorithm. We present experimental results for two- and three-dimensional simplicial meshes demonstrating the effectiveness of this approach for a number of popular mesh quality measures.

Lori A. Freitagand Paul E. Plassmann
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue, Argonne, IL 60439-4844
plassman@mcs.anl.gov

Mark T. Jones
Department of Computer Science
University of Tennessee
Knoxville, TN 37996-1301
jones@cs.utk.edu

CP08
Parallel Adaptive Mesh Refinement for Electronic Structure Calculations

We have applied adaptive mesh refinement techniques to the solution of the LDA equations for electronic structure calculations. Local spatial refinement concentrates memory resources and numerical effort where it is most needed, near the atomic centers and in regions of rapidly varying charge density. The structured grid representation enables us to employ efficient iterative solver techniques such as conjugate gradients with multigrid preconditioning. We have parallelized our solver using an object-oriented adaptive mesh refinement framework.

Scott Kohn
Center for Applied Scientific Computing
Lawrence Livermore National Laboratories, Livermore, CA
Email: skohn@chem.ucsd.edu

John Weare
Department of Chemistry and Biochemistry
University of California at San Diego, La Jolla, CA

Elizabeth Ong
Department of Mathematics
University of California at San Diego, La Jolla, CA

Scott Baden
Department of Computer Science

University of California at San Diego, La Jolla, CA

CP08
Worst Case Complexity of Parallel Triangular Mesh Refinement by Longest Edge Bisection

We present a logarithmic algorithm for performing parallel refinement of triangular meshes by the widely used longest edge bisection procedure. We show that the refinement propagation forms a data dependency which can be expressed as a forest of directed trees. We solve a parallel Euler Tour problem on the trees to propagate the refinement. After propagation, we apply the refinement templates. Our algorithm improves earlier reported results which had linear worst case complexity.

Can Özturan
Computer Engineering Department
Bogazici University
80815 Bebek, Istanbul, Turkey
ozturan@icase.edu, ozturanc@cs.rpi.edu

CP09
A Parallel Frontal Solver for Process Simulation

We present a parallel frontal solver for problems arising in Chemical Process Engineering. The plant model is formulated by using internal variables that correspond to each process unit and interface variables which correspond to the connections between them. This, or a separate reordering phase, gives rise to a Bordered Block Diagonal Form (BBDF). The diagonal blocks are solved in parallel by using a frontal solver and inner loop parallelism is exploited for the interface.

Jayarama U. Mallya
Cray Research, Inc.
655-E, Lone Oak Drive, Eagan, MN 55121
E-mail: mallya@cray.com

Mark A. Stadtherr
Department of Chemical Engineering
University of Notre Dame
Notre Dame, IN 46556
E-mail: stadtherr.1@nd.edu

Stephen E. Zitney and Samar Choudhary
Cray Research, Inc.
655E, Lone Oak Drive, Eagan, MN 55121
E-mail: choudh@cray.com

CP09
Sampling and Analytical Techniques for Data Distribution of Parallel Sparse Computation

We present a compile–time method to transform Fortran 90 programs where arrays can be declared sparse into equivalent Fortran 90 programs where sparse arrays are further annotated with compression and distribution schemes to facilitate parallel execution. The method is based on sampling input sparse matrices to determine their sparsity structures, and on partitioning sparse matrices according to cost functions measured from the target machine. Similar techniques are also applicable at run–time under the inspector–executor framework.

Tyng–Ruey Chuang
Institute of Information Science
Academia Sinica
Nankang, Taipei 11529, Taiwan
trc@iis.sinica.edu.tw

Rong–Guey Chang and Jenq Kuen Lee
Department of Computer Science
National Tsing Hua University
Hsinchu 300, Taiwan
dr848305@cs.nthu.edu.tw
jklee@cs.nthu.edu.tw

CP09
A Comparison of 1-D and 2-D Data Mapping for Sparse LU Factorization with Partial Pivoting

This paper presents a comparative study of two data mapping schemes for parallel sparse LU factorization with partial pivoting on distributed memory machines. Our previous work has developed an approach that incorporates static symbolic factorization, nonsymmetric L/U supernode partitioning and graph scheduling for this problem with 1-D column-block mapping. The 2-D mapping is commonly considered more scalable for general matrix computation but is difficult to be efficiently incorporated with sparse LU because partial pivoting and row interchanges require frequent synchronized inter-processor communication. We have developed an asynchronous sparse LU algorithm using 2-D block mapping, and obtained competitive performance on Cray-T3D. We report our preliminary studies on speedups, scalability, communication cost and memory requirement of this algorithm and compare it with the 1-D approach.

Cong Fu, Xiangmin Jiao and Tao Yang
Department of Computer Science
University of California at Santa Barbara
Santa Barbara, CA 93106
Email: {cfu, jiao, tyang}@cs.ucsb.edu

CP09
Design and Implementation of a Scalable Parallel Direct Solver for Sparse Symmetric Positive Definite Systems

Solving large sparse systems of linear equations is at the core of many problems in engineering and scientific computing. It has long been a challenge to develop parallel formulations of sparse direct solvers due to several different complex steps involved in the process. In this talk, we will describe one of the first efficient, practical, and robust parallel solver for sparse symmetric positive definite linear systems that we have developed and will discuss the algorithmic and implementation issues involved in its development. Experimental results of this MPI based solver on IBM SP2 and Cray T3D will be presented.

Anshul Gupta and Fred Gustavson
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
Email: (anshul, gustav)@watson.ibm.com

Mahesh Joshi, George Karypis
and Vipin Kumar
Department of Computer Science
University of Minnesota, Minneapolis, MN 55455
Email: (karypis, kumar)@cs.umn.edu

CP09
A Parallel Sparse Symmetric Indefinite Solver

We describe a parallel solver for sparse, symmetric indefinite, linear systems of equations. The solver uses MPI protocols for portability across distributed-memory and cluster computers. This is the first parallel sparse indefinite solver reported in the literature known to us.
A critical issue here is the design of dynamic data structures that can accommodate the swapping and communication of columns and rows in parallel within a symmetric pivoting scheme. We discuss the various design choices that we have made in our parallel implementation; we report results on an IBM SP-2 for problems from structural analysis and linear programming.

Yogin Campbell and Alex Pothen
Computer Science Department
Old Dominion University, Norfolk, VA 23529-0162
({yogin, pothen}@cs.odu.edu)
The second author is also affiliated with ICASE, MS 132 C, NASA
Langley Research Center, Hampton, VA 23681 (pothen@icase.edu).

CP09
Improving Instruction-Level Parallelism in Sparse Matrix-Vector Multiplication using Reordering, Blocking, and Prefetching

Sparse Matrix-Vector Multiplication is an important kernel that often runs inefficiently on superscalar RISC processors. This paper describe techniques increase instruction-level parallelism and improve performance. The techniques include reordering to reduce cache misses (originally due to Jalby and Temam), blocking to reduce load instructions, and prefetching to prevent multiple load-store units from stalling simulteneously. The techniques improve performnance from about 40 Mflops (on a well-ordered matrix) to over 100 Mflops on a 266 Mflops machine.

Sivan Toledo
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304
Email: toledo@parc.xerox.com

CP10
DpShell: A Data Parallel File System

Most parallel I/O systems allow parallel accesses to sequential files. The DpShell is a multidimentionnal file system. Each file is distributed onto a grid of I/O processors as `template` onto `HPF` processors grids. Data-parallel tasks running on different parallel computers with distincts topologies can interact efficiently by reading or writing Data Parallel file. Object migrations from one parallel computer to another lead us to define efficient dynamic algorithms with communication and computation overlapping

Jean-Luc Dekeyser and Dominique sueur
University of Lille, France
Laboratoire d'Informatique Fondamentale de Lille
Villeneuve d'Ascq, France
dekeyser@lifl.fr, sueur@lifl.fr

CP10
I/O Optimizations for Compiling Out-of-Core Programs on Distributed-Memory Machines

Since many of large scale computational problems usually deal with large quantities of data, optimizing the performance of I/O subsystems of massively parallel machines is an important challenge for system designers. We describe data access reorganization strategies for efficient compilation of out-of-core data-parallel programs on distributed memory machines. Our analytical approach and experimental results indicate that the optimizations introduced in this paper can reduce the amount of time spent in I/O by as much as an order of magnitude on both uniprocessors and multicomputers.

Mahmut Kandemir
CIS Department
Syracuse University, Syracuse, NY 13244
E-mail: mtk@top.cis.syr.edu

Rajesh Bordawekar
CACR, Caltech, Pasadena, CA 91125

Alok Choudhary
Dept. of Electrical and Computer Engineering
Northwestern University, Evanston, IL 60208-3118

CP10
Application Experience with Parallel Input/Output: Panda and the H3expresso Black Hole Simulation on the SP2

This paper summarizes our experiences using the Panda parallel I/O library with the H3expresso numerical relativity code on the Cornell SP2. Three performance issues are described: providing high performance despite the SP2's heterogeneity, providing efficient off-loading of output data, and satisfying users' desire to dedicate fewer nodes to I/O. We explore the tradeoffs between potential solutions, and present performance results for our approaches. We also show that Panda's array-level I/O interface matches H3expresso's needs.

S. Kuo, M. Winslett, Y. Chen, and Y. Cho
Computer Science Department
University of Illinois, Urbana IL 61801

M. Subramaniam
Oracle Corporation, Redwood Shores, CA 94065

K. Seamons
Transarc Corporation, Pittsburgh, PA 15219 **CP10**

Experiences with the PIOUS Parallel and Distributed I/O System

The PIOUS system supports parallel and distributed I/O in general purpose networked environments. PIOUS permits increased throughput by declustering files, and provides concurrency control mechanisms for consistent access. We report on our experiences with a version of PIOUS implemented for PVM. Overheads associated with parallel I/O over networks are measured, and the different concurrency control modes are evaluated. We describe the porting of the NASA NHT-1 benchmarks to PIOUS, and report on observed performance behaviour. The use of PIOUS in conjunction with PVM or MPI systems for parallel computing on heterogeneous networks is discussed

Vaidy Sunderam
vss@mathcs.emory.edu Emory University, Atlanta, GA

30322

CP10
An Architecture-Independent Parallel Implicit Flow Solver with Efficient I/O

Design and performance of implicit flow solver RANS-MP are discussed. The efficient bi-directional multi-partition solver algorithm, in conjunction with the MPI and newly developed MPI I/O libraries, leads to truly portable program text AND performance. Superior computational results are presented for cache-based distributed- memory systems, an MPP (IBM SP2) and a LAN (SGI cluster), and a vector machine (Cray C90). We also report excellent parallel I/O on the SP2.

Rob F. Van der Wijngaart
MRJ, Inc, NASA Ames Research Center
Mail Stop T27A-1, Moffett Field, CA 94035
(415) 604-3983, wijngaar@nas.nasa.gov

Maurice Yarrow
Sterling Software, NASA Ames Research Center
Moffett Field, CA 94035
(415) 604-5708, yarrow@nas.nasa.gov

Merritt H. Smith
Hewlett Packard Technical Consulting Laboratory
Mail Stop 37, 3404 East Harmony Road
Ft. Collins, CO 80525
mhsmith@fc.hp.com

CP10
File Access Methods on Parallel Processing

Data storage is frequently ignored whenever evaluating algorithms. It is very impotant to consider the performance of storing and retrieving data whenever evaluation parallel algorithms for two major reasons. First, in most parallel architectures processors are assigned to a single process until completion of that process; thus the assigned processors are unavailable for use by other processes until all processing, including I/O and stoarge, is compete. Second, the manner in which parallel processors perform disk storage can have a profound effect on overall system performance; choosing the wrong algorithm can overload the file-server processor for long period of time, degrading the overall performance of all other currently-running processes. We compare four different methods of file access on an Intel Paragon. In method A, the data is stored into separately-patitioned files, one for each processor. In method B, each processor performs its own access to the global file with no coordination. Method C is a modification of method B in which the nodes are coordinated such that they perform their disk access one at a time. In method D, all file access is performed by a single processor (node 0), which communicates the partitions to/from all other processors.

John A. Youssefi
California State University
Bakersfield, CA 93311
abbas@ultrix6.cs.csubak.edu

CP11
Automatic Differentiation of a Parallel Molecular Dynamics Application

The ADIC and ADIFOR automatic differentiation tools have proven extremely useful for obtaining the derivatives needed in many sequential scientific applications. But many new scientific programs are written in or ported to parallel systems to achieve maximal performance. We provide an overview of our approach to the complex task of applying automatic differentiation techniques to parallel programming environments, in the context of a parallel molecular dynamics application written in C++ with PVM/MPI message passing.

Paul D. Hovland
Department of Computer Science
University of Illinois
Urbana, IL 61801
Email: hovland@cs.uiuc.edu

Christian H. Bischof and Lucas J. Roh
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

CP11
Parallel Particle Simulations Using the POOMA Framework

The POOMA Framework is an object-oriented library and application suite implemented in C++ which enables the parallel simulation of the dynamics of interacting particles and fields. Using templates, POOMA provides a flexible and intuitive syntax to express global field and particle operations and creates applications with near hand-coded performance. POOMA is employed at present in multi-material hydrodynamics calculations, gyrokinetic plasma dynamics, and Monte Carlo neutron transport simulations. Currently under development within the POOMA Framework are interactive visualization tools for use in all applications developed using POOMA.

Julian C. Cummings and William F. Humphrey
Advanced Computing Laboratory
Los Alamos National Laboratory
CIC/ACL, MS B287, Los Alamos, NM 87545
E-mail: billh@acl.lanl.gov

CP11
Practical Parallelization of a Molecular Dynamics Application on Shared and Distributed Memory Machines

This paper present the comprehensive performance profiling, optimization, and benchmarking of a Molecular Dynamics code on both shared and distributed multiprocessor systems. In our analysis, we consider language, compiler, architecture, and algorithmic changes for each parallelizable portion and discuss the respective bottlenecks. We give examples of the existence of alternatives for parallelization and compare the performance of each by discussing their advantages and disadvantages. With the above techniques we achieved a speedup of 42 on a 128 processor Cray-T3D.

Masayuki Kuba
Asahi Chemical Industry, Fuji, 416, JAPAN
Email: kuba@cat.fuji.asahi-kasei.co.jp

Marie-Christine Brunet
Center for Supercomputing Research and Development
University of Illinois at Urbana-Champaign, IL 61801, U.S.A.

Email: maiki@csrd.uiuc.edu

Constantine D. Polychronopoulos
Center for Supercomputing Research and Development
University of Illinois at Urbana-Champaign, IL 61801, U.S.A.
Email: cdp@csrd.uiuc.edu

CP11
Parallel Molecular Dynamics using Truncated Octahedron Periodic Boundaries

Supercritical evaporation is a highly favorable process in advanced combustion devices. Droplets exposed to supercritical environments evaporate rapidly due to the reduction of surface tension when exposed to the high density and high energy surroundings. In pursuit of a simulation of supercritical droplet evaporation, a large scale molecular dynamic model with minimal periodicity effects was developed. This incorporated parallel molecular dynamic decomposition strategies, linked-cell and Verlet neighbor list filtering enhancements, and a truncated octahedron periodic boundary. The shape of this boundary provides the closest match with the initial spherical geometry of the evaporating droplet. This minimizes the imbalanced density and thermal gradients produced by periodicity effects. This is important since the low energy surface of the droplet would readily respond to such gradients. Applying the molecular filtering and parallel techniques resulted in a fairly efficient program for modeling the large systems required for the supercritical studies. The simulations were performed on an IBM SP2 using message passing library calls. Parallel performance measures will be presented as well as visualizations showing the success of the minimized periodicity effects.

Jeffery K. Little
AFIT/ENY, Bldg 640, 2950 P Street
Wright-Patterson AFB, OH 45433-7765
Email: jlittle@afit.af.mil

CP11
Parallel MD–Simulations of Synthetic Polymers

A parallel algorithm is developed that allows efficient Langevin–dynamics simulations of single macromolecular coils which are the typical structure of synthetic polymers in solution and in bulk. Contrary to the usual spatial decomposition algorithms, we map the one–dimensional topology of the chain molecule on the parallel computer. Only short–range interactions between atoms are considered. The necessary neighbourhood search is done by a master–slave method.
This work is supported by the DFG–Research Cluster "Efficient Algorithms For Discrete Problems And Their Applications", grant LE 952/1–2

Bernd Jung
Editorial office "Macromolecular Chemistry and Physics"
Hegelstr. 45, 55122 Mainz, Germany

Hans–Peter Lenhof, Peter Müller, Christine Rüb
Max-Planck-Institut für Informatik
Im Stadtwald, 66123 Saarbrücken, Germany
E-mail: pmueller@mpi-sb.mpg.de

CP11
Particle–Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations

The parallel implementation of a molecular dynamics code suitable for simulation of general molecular systems using spatial–decomposition methods is discussed. Long-range Coulombic forces are computed using a particle-mesh Ewald (PPPM) technique. A multiple–timescale integration method known as rRESPA is also used to improve the computational efficiency. The load–balancing and interprocessor communication issues that arise from implementing PPPM and rRESPA within a spatial–decomposition framework are highlighted. Some performance numbers for large–scale biomembrane simulations on a Intel Paragon are given.

Steve Plimpton
MS 1111
Sandia National Labs
Albuquerque, NM 87185
E-mail: sjplimp@cs.sandia.gov

Roy Pollock
MS L-50
Lawrence Livermore National Labs
Livermore, CA 94550

Mark Stevens
MS 1111
Sandia National Labs
Albuquerque, NM 87185

CP12
Highly Scalable 2-D and 3-D Navier-Stokes Parallel Solver on MIMD Multiprocessors

We present a new parallel multidomain algorithm for the solution of the incompressible two and three-dimensional Navier-Stokes equations. To achieve this goal we apply the multidomain local Fourier (MDLF) method which is a spectral method for the parallel solution of 2-D and 3-D time dependent nonlinear PDEs, to the parallel solution of 2-D and 3-D Navier-Stokes equations. The parallelization is achieved by domain decomposition (DD). The important advantage of the MDLF method is that it enables to perform the matching of each harmonic separately, and thus to eliminate the global coupling of the interface unknowns. We show that In each time step we solve the Helmholtz and Poisson equations. We show that for elliptic equation of Poisson type only the lowest harmonics of the pressure are treated and matched globally. Therefore, most of the communication that is required for parallelization of the Navier-Stokes equations is mainly local between adjacent subdomains (processors). Therefore,the parallel algorithm is highly scalable. The 2-D and 3-D Navier-Stokes equations are implemented on MIMD message-passing multiprocessors (60 processors SP2, 20 processors MOSIX (network of i586) and network of 10 ALPHA workstations) and it achieves above 85% efficiency. The same code written with PVM was executed on all the above distinct computational platforms.

Amir Averbuch
School of Mathematical Sciences, Tel Aviv University
Tel Aviv 69978, Israel
E-mail: amir@math.tau.ac.il

Ludmila Ioffe, Moshe Israeli, Lev Vozovoi
Faculty of Computer Science

Technion-Israel Institute of Technoloty
Haifa 3200, Israel

CP12
Large Scale Industrial Applications of Parallel Computers: Optic and Fluid Dynamic Analyses Method

We report our experience in converting two computing intensive analysis codes to a message passing shared memory computing environment which consists of a IBM SP2 computer and a network of workstation connected by ethernet, respectively. These codes solve partial differential equations using Fourier transform and Legendre Gauss quadrature method. A domain decomposition approach was adpoted, in which all the basic computational kernels remain intact and only the portion of code that uses these kernels is modified. With a fixed problem size involving 5 million degrees of freedom, we demonstrated a 80 percent parallel efficiency for up to 32 processors. and shortened the analysis turnaround time from 21 days to 1 day.

Daniel C. Chan
CFD Technology Center, Mail Stop IB 39
Rocketdyne Division, Rockwell International Corporation
6633 Canoga Avenue, Canoga Park, CA 91303
dchan@rdyne.rockwell.com

Patrick Hu
Optics Group, Mail Stop IA 04
Rocketdyne Division, Rockwell International Corporation
6633 Canoga Avenue, Canoga Park, CA 91303

CP12
Parallelization of a Fourier-Chebyshev Spectral Method for CFD Applications

A Fourier-Chebyshev collocation spectral method is used to simulate the 3-d unsteady flow inside a cylindrical enclosure with rotating ends and sidewalls. The solution approach, which combines an influence matrix technique for mass conservation and matrix diagonalization for solving the pressure Poisson equation and Helmholtz equations for velocity components, is parallelized for distributed memory applications using MPI and for shared memory applications using Cray directives. The parallelization approach is described and scaling results are presented for both platforms.

Roger W. Hill and Kenneth S. Ball
Department of Mechanical Engineering
University of Texas at Austin
Austin, TX 78712
Roger.Hill@SEMATECH.Org
kball@burst.me.utexas.edu

CP12
Parallelization of Approximate Factorization Algorithms Using 2D Processor Topologies in MPI

We present a technique for parallelization on distributed memory computers of the Approximate Factorization (AF) algorithms used in fully implicit computational fluid dynamics (CFD) programs. We employ a 2D processor topology to decompose the 3D solution data for each stage of the AF algorithm. This enables an efficient scalable communication strategy to transpose the solution data between AF stages. We use the Message Passing Standard (MPI) for communication, taking advantage of MPI communicators, topologies and general data types. We present performance results on the NAS IBM-SP2 for the NAS BT Benchmark and the Full Potential Rotor (FPR) code from the NASA Ames AFDD.

Joseph W. Manke and James C. Patterson
The Boeing Company
P.O.Box 3707, #MS 7L-22
Seattle, WA 98124-2207
joseph.w.manke@boeing.com
pattersn@redwood.rt.cs.boeing.com

CP12
A Fluid Flow Simulation on MPP Platforms

The full time dependent Navier-Stokes equations for an incompressible fluid are solved in an axisymmetric sudden expansion geometry for Reynolds number (Re) = 2500 to simulate fluid flow. A vorticity stream function formulation, based on global conservation of mass and momentum, is used for flow simulation. The vorticity equation is solved using an alternating-direction fully implicit (ADI) algorithm which results in a tridiagonal system of linear equations. This tridiagonal system of linear equations is solved by using GMRES iterative solver. The stream function equation is a Poisson equation with variable coefficients and it results in a block tridiagonal system of linear equations. The block tridiagonal system is solved by a parallel variant of the Gauss-Seidel iterative method. MPP platforms with Message-Passing Interface (MPI) are used in these simulations. The scalability and performance of the code on CRAY T3D is discussed.

C.F. Bender and M. L. Mittal
Ohio Supercomputer Center
Columbus, OH 43212

J.D. Schweickart
The Ohio State University
Columbus, OH 43210

CP12
Parallel Domain Decomposition Methods for the Implicit Solution of Unsteady Compressible Navier-Stokes Flows on Moving Three-Dimensional Unstructured Meshes

We report our experiences using the Variable Degree Schwarz preconditioned FGMRES methods, on distributed memory parallel computers, in the implicit solution of the three-dimensional unsteady compressible Navier-Stokes equations discretized on moving unstructured meshes at both subsonic and transonic regimes. Two levels of preconditioned FGMRES iterative methods are used in the calculations. Our implementation is based on MPI and we discuss its performance on a cluster of workstations and also on an IBM SP2.

Xiao-Chuan Cai
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309
Email: cai@cs.colorado.edu

Charbel Farhat
Department of Aerospace Engineering
University of Colorado at Boulder

Boulder, CO 80309
Email: charbel@alexandra.colorado.edu

Marcus Sarkis
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309
Email: msarkis@cs.colorado.edu

CP13
Parallelization Agent: A New Approach for Parallelizing Legacy Codes

Over the years, scientists and engineers have built a large software base of numerical modeling codes for scientific computing. These legacy codes, are very valuable because of the solid pedigree associated with them. We are developing an experimental system called Parallelization Agent to facilitate parallelization of legacy codes. The physics, mathematics, and numerical methods together create a complexity which makes automatic parallelization difficult. By creating powerful abstractions for important classes of scientific and engineering codes and by developing a structured parallelization method based on these abstractions, the Parallelization Agent research promises to show new ways to solve an otherwise intractable problem. We have developed a prototype for parallelizing a subclass of finite difference codes. We have applied the prototype tool to parallelize the Penn State/NCAR MM5, the fifth generation Mesoscale Meteorology model. In this paper we will discuss the new approach and our experience with the prototype system.

Simanta Mitra and Suraj C. Kothari
Department of Computer Science
Iowa State University, Ames, IA 50011
Email: (mitra, kothari)@cs.iastate.edu

CP13
A Formal Approach to Parallelizing Compilers

This paper describes parallelizing compilers which allow programmers to tune parallel program performance through an interactive dialog. Programmers specify language constructs that define sequential processes, such as assignment or for-loops, to be used as units of computation, while the compiler discovers the parallelism existent in the source program in terms of these units. Programmers may provide target machine architectural features used by compilers to coalesce sequential processes, controlling process granularity and ensuring process load balance.
This work was partially supported by the grant NGT-51321 from the NASA Jet Propulsion Laboratory.

Teodor Rus and Eric Van Wyk
Department of Computer Science
The University of Iowa
Iowa City, Iowa 52242
rus,vanwyk@herky.cs.uiowa.edu

CP14
Parallel Iterative Solution of Finite Element Systems of Equations Employing Edge-Based Data Structures

We present several optimization techniques for the sparse matrix-vector multiplication needed on the conjugate gradient solution of finite element system of equations on unstructured grids composed by triangles and tetrahedral elements. The optimization techniques are based on the transition from a conventional element-by-element scheme to an edge-by-edge data structure. Improvments in the conventional element-by-element scheme are also obtained exploring the natural rank deficiency of the element matrices. Although this study is restricted to problems governed by the quasi-harmonic equation, the results apply equally for more involved problems. The resulting algorithms are implemented considering the vectorization and parallelization capabilities of current supercomputers. The edge-by-edge techniques are applied in the solution of large scale industrial configurations, showing considerable improvements in comparison with the standard element-by-element scheme.!

Marcos A. D. Martins, Alvaro L. G. A. Coutinho
and Jose L.D. Alves
COPPE/UFRJ - Civil Engineering Department
Caixa Postal 68506
21945-970 Rio de Janeiro, Brazil
E-mail: alvaro@coc.ufrj.br

CP14
The Termination of Synchronous Parallel Algorithms for Iterative Systems

We describe a method of decentralised convergence testing, resulting in a significant speed up due to reduced message-passing in the system; for a widely used class of linear iterative systems, we adapt it to translate a global convergence criterion into a set of (different) processor-based criteria which ensures consistency between the solution obtained and that obtained using central co-ordination, but with much improved performance. Finally, we present numerical results demonstrating the effectiveness of the technique.

Thomas J. Curnock and John Ashworth
Dept. of Computer and Mathematical Science
University of Salford, M5 4WT, UK
Email: J.Ashworth@mcs.salford.ac.uk

CP14
Numerical Experiments with Parallel Sparse Unsymmetric Unstructured Iterative Solvers on Network of Highend Workstations

We present results of numerical experiments with iterative solution methods for large sparse unstructured unsymmetric linear systems on networks of highend workstations. We show that incomplete triangular factorization preconditioned GMRES algorithms do allow efficient parallel implementations on such high latency low data transfer rate computer systems with the NFS mounted common disk space. We demonstrate using sample problems from the oil industry that one can really exloit networks of 8 workstations connected only via Ethernet.

A. Yeremin
Center For SuperComputer and Massively Parallel Applications
of the Russian Academy of Sciences
Moscow 117333, Russia
and Elegant Mathematics, Inc. (USA)
E-mail: yeremin@ext1.ccas.ru

CP15
A Portable MPI Implementation of the SPAI Pre-

conditioner in ISIS++

A parallel MPI implementation of the Sparse Approximate Inverse (SPAI) preconditioner is described. SPAI has proven to be a highly effective preconditioner, and is inherently parallel because it computes columns (or rows) of the preconditioning matrix independently. However, there are several problems that must be addressed for an efficient MPI implementation: load balance, latency hiding, and the need for one-sided communication. The effectiveness, efficiency, and scaling behavior of our implementation will be shown for different platforms.

Stephen T. Barnard
NASA Ames Research Center
T027-A, Moffett Field, CA 94035-1000
Email: barnard@nas.nasa.gov

Robert Clay
Sandia National Laboratory
Livermore, CA 94550
Email: rlclay@ca.sandia.gov

CP15
Tensor Sum Approximation Preconditioners

Preconditioning techniques based on incomplete LU factorizations offer a limited degree of parallelism and are often difficult to implement. This talk will present an ADI-like alternative which consists of approximating an arbitrary matrix A by a tensor sum of matrices. Recall that the tensor product $X \otimes Y$ of the matrices X by Y is a block matrix whose generic block in block-entry i, j is the matrix $x_{ij}Y$. The tensor sum $X \oplus Y$ is the matrix $X \oplus Y = X \otimes I + I \otimes Y$. Such matrices arise naturally when discretizing an elliptic PDE with constant coefficients on a rectangular domain. In more general cases, it is still possible to obtain the least-squares approximation of the original matrix in the form of a tensor sum. The resulting preconditioning operation consists of solving alternatively with each of the two terms of the tensor sum. Each of these solves amounts to solving linear systems with multiple right-hand sides (in parallel), similarly to the Alternating Direction Implicit (ADI) procedure. We will discuss a method based on this approach and illustrate its implementation and performance on workstation clusters.

Paul Castillo, and Yousef Saad
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455
Email: castillo@cs.umn.edu

CP15
Parallel Sparse Approximate Inverse Preconditioners

There has been much excitement recently over the use of approximate inverses for parallel preconditioning. The preconditioning operation is simply a matrix-vector product, and in the most popular formulations, the construction of the approximate inverse seems embarassingly parallel. However, difficulties arise in practical parallel implementations. This talk will survey approximate inverse preconditioners, and discuss the wide variety of options, such as for sparsity pattern selection. We address the pros and cons of each method, and put the methods into perspective.

Edmond Chow
Department of Computer Science, and
Minnesota Supercomputer Institute
University of Minnesota
Minneapolis, MN 55455
E-mail: chow@cs.umn.edu

Yousef Saad
Department of Computer Science, and
Minnesota Supercomputer Institute
University of Minnesota
Minneapolis, MN 55455

CP16
A Scalable Geometric Partitioning Heuristic

The partitioning of a unstructured, computational mesh for parallel computation is a difficult (but essential) task. The aspects of the partitioning that are computationally important, however, depend heavily on how the mesh is used. In this talk we motivate a computational model specifying how a partitioning heuristic should be scalable, should ensure that the induced partition graph has bounded degree (independent of the number of partitions), and should allow for an efficient repartitioning (after small changes to the mesh or changes in the computational loading of the processors). We present a geometric partitioning heuristic that attempts to satisfy these properties, an efficient parallel implementation of this heuristic, and computational results for this heuristic used within several applications.

Brian Davis and Mark T. Jones
Department of Computer Science
University of Tennessee
Knoxville, TN 37996-1301
jones@cs.utk.edu

Paul E. Plassmann
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue, Argonne, IL 60439-4844
plassman@mcs.anl.gov

CP16
Skewed Graph Partitioning

Graph partitioning is an important abstraction used in solving many scientific computing problems. Unfortunately, the standard partitioning model does not incorporate considerations that are important in many settings. We address this by describing a generalized partitioning model which incorporates the notion of partition *skew* and is applicable to a variety of problems. We then develop enhancements to several important partitioning algorithms necessary to solve the generalized partitioning problem. Finally we demonstrate the benefit of employing several of these generalized methods to static decomposition of parallel computing problems.

Bruce A. Hendricksonand Robert Leland
Sandia National Laboratories
Albuquerque, NM 87185
Email: bahendr@cs.sandia.gov

Rafael Van Driessche
Alcatel Telecom

CP16
A Data-Parallel Implementation of the Geometric

Partitioning Algorithm

We present a data-parallel implementation of the geometric partitioning algorithm in High Performance Fortran (HPF). The geometric partitioning algorithm is the one of the only two algorithms (the other one is the spectral partitioning algorithm) that mathematically guarantee the quality of the partitions. We formulate the geometric partitioning algorithm in terms of segmented prefix sums and parallel selections, and provide a data-parallel procedure for geometric sampling. Experiments on partitioning unstructured meshes have shown that our data-parallel formulation of the geometric partitioning algorithm is efficient. Moreover, it generates good separators that are competitive with those generated by other partitioning algorithms such as the spectral bisection. To our knowledge, this is the first parallel implementation of the algorithm.

Y. Charlie Hu
Harvard University, Cambridge, MA
E-mail: hu@das.harvard.edu

Shang-Hua Teng
University of Minnesota, Minneapolis, MN
E-mail: steng@cs.umn.edu

S. Lennart Johnsson
Harvard University, Cambridge, MA
University of Houston, Houston, TX
E-mail: johnsson@cs.uh.edu

CP17
Asynchronous Parallel Solvers for Hyperbolic Conservation Laws

This paper presents parallel asynchronous finite difference solvers for the solution of hyperbolic conversation laws. These solvers are specifically designed to minimize synchronization overhead on multi-processos systems. This is achieved by allowing each processor to proceed with the calculation of the next time step as soon as it finishes the current one, regardless of whether its neighbors have also finished. Unlike existing synchronous (lock-step) methods which utilize external explicit synchronization mechanism to assure accurate results, the proposed methods avoid synchronization as they are constructed to implicitly contain the numerical constraints that assure accurate results. Following the line of our previous works on parabolic equations we first develop asynchronous, corrected-asynchronous, and adaptive-asynchronous versions for known synchronous finite-difference schemes of the scalar converation law. Their consistency, stability and convergence are analyzed in detail. Those are also applied as proofs for the Independent Time Step Method (ITSM) which turns out to be a special case of the proposed method. Our methods are demonstrated for Euler upwind, Lax-Wendroff, Beam-Warming and Roe-Sweby TVD schemes. However, they can be extended to other schemes and other problems. Then, for the multi-dimensional conversation law we develop asynchronous versions of known characteristics methods. It is also demonstrated for an asynchronous solution of the shallow water equations used for weather prediction.

Amir Averbuch, Dganit Amitai
School of Mathematical Sciences, Tel Aviv University
Tel Aviv 69978, Israel
E-mail: amir@math.tau.ac.il

Moshe Israeli
Faculty of Computer Science
Technion-Israel Institute of Technology
Haifa 3200, Israel

CP17
Simulations of Geophysical Flows on the IBM SP2

A simulation code for the three-dimensional, unsteady, turbulent, variable density, and rotation-influenced motions of an incompressible fluid has now been implemented on the IBM SP2 using message passing. The Navier-Stokes code was successfully validated for a variety test cases. Simulations of a laboratory model of coastal upwelling will be presented. A finite volume method is used to discretize the governing Navier Stokes equations in general curvilinear coordinates on a single non-staggered grid. A semi-implicit time-advancement scheme, with the Adams-Bashforth method for the explicit terms and Crank-Nicolson for the implicit terms, has been used. All the spatial derivatives are approximated with second-order central differences with the exception of the convective terms, which are handled with special upwind-difference schemes. A multigrid method, with ZEBRA as
smoother, is used to solve the pressure Poisson equation, while the momentum equations are solved with an approximate factorization technique. The algorithm is second-order accurate both in space and in time.

Anqing Cui and Robert Street
Environmental Fluid Mechanics Lab
Stanford University, Stanford, CA 94305
Email: cui@leland.Stanford.EDU

CP17
Parallel Computation of Large Bubble Cavitation

A parallel solver for inviscid, incompressible flow based on Chorin's method of artificial compressibility along with an adaptive free-boundary tracking algorithm is used to model steady-state, cavitating flow. The parallel algorithms are constructed to account for the complexities introduced by the front-tracking algorithm and are designed to improve the parallel computational throughput of the adaptive, unstructured mesh algorithm. Simulations demonstrating the solution approach are done on the PSC Cray-T3D.

Joseph Kolibal
Department of Mathematics
University of Southern Mississippi
Hattiesburg, MS 39406-5045
Email: kolibal@eos.st.usm.edu

George Gong
Program in Scientific Computing
University of Southern Mississippi
Hattiesburg, MS 39406-5045
E-mail: gong@eos.st.usm.edu

CP17
A Parallel Method for Predicting Supercritical Fluid Transport Properties

Supercritical droplet evaporation and combustion are some of the least understood processes that occur in advanced combustors, especially in diesel engines, gas turbine engines, and rocket motors. These systems all operate at supercritical or near supercritical conditions. In this regime the complex physics do not allow one to make the same

simplifying assumptions that are often made for gases or liquids. Continuum mechanics is of limited usefulness. We are especially interested in studying surface tension, thermodynamic transport phenomenon, and equations of state. In these extremely high presssure and high temperature environments, experimental data is often not available since experiments are very difficult at these conditions. In this paper, the determination of transport coefficients of supercritical fluids using a molecular dynamics method and the Green-Kubo formulae is described. Preliminary results for bulk liquid argon and liquid oxygen are compared with experiment. Results are presented for sub-critical, near-critical, and super-critical conditions. The computer code is written in Fortran-77 and uses the message passing interface (MPI). Most of the computer runs are made on an IBM SP-2 computer. Parallel efficiency and other performance measures will be presented.

Lyle N. Long, Obika Nwobi and Michael M. Micci
233 Hammond Building
Pennsylvania State University
University Park, PA 16802
Email: lnl@psu.edu

CP17
Three-Dimensional Simulations of Compressible Turbulence on High-Performance Computing Systems

A three-dimensional PPM code is used to examine compressible fluid turbulence in three dimensions. The code runs on a number of parallel architectures, including MPPs and SMP clusters. We consider problems of current interest, such as Rayleigh-Taylor and Richtmyer-Meshkov instability and turbulent mixing, and interactions of a shock with pre-existing turbulence. We present performance results on leading-edge platforms, including those supported under the Accelerated Strategic Computing Initiative (ASCI).

A. A. Mirin, R.H. Cohen, W.P. Dannevik, A.M. Dimits, and D.E. Eliason
Lawrence Livermore National Laboratory
P.O. Box 808, Livermore, CA 94550
mirin@natali.llnl.gov

D.H. Porter
University of Minnesota, Minneapolis
Laboratory for Computational Science and Engineering
200 Union Street SE, Minneapolis, MN 55455

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

P.R. Woodward
University of Minnesota, Minneapolis
Laboratory for Computational Science and Engineering
200 Union Street SE, Minneapolis, MN 55455

CP17
Parallel Iterative Solution of the Navier-Stokes Equations with Heat and Mass Transport

The iterative solution of the governing transport equations for momentum, heat and mass transfer in flowing fluids can be very difficult. These difficulties arise from the nonlinear, coupled, nonsymmetric nature of the system of algebraic equations resulting from numerical approximation of the PDEs. This talk focuses on evaluating a number of preconditioned Krylov methods for solution of the Navier-Stokes equations with heat and mass transport. Our discussion considers efficiency, robustness and implementation issues for fully-coupled solutions methods.

John N. Shadid, Raymond S. Tuminaro, and Scott A. Hutchinson
Sandia National Laboratories
Albuquerque, NM 87185
jnshadi@cs.sandia.gov, tuminaro@jemez.cs.sandia.gov, sahutch@jemez.cs.sandia.gov

CP18
Scalability of an Adaptive GMRES(k) Method

Applications involving large sparse nonsymmetric linear systems encourage parallel implementations of popular iterative solution techniques, such as GMRES(k). One variation of GMRES(k) is to adapt the restart value k for any given problem and use Householder reflections in the orthogonalization phase to achieve high accuracy. This paper compares the abilities of this adaptive procedure and the standard GMRES(k) to maintain constant efficiency with increase in problem size and number of processors. The Householder transformations can be performed without global communications and modified to utilize an arbitrary row distribution of the coefficient matrix. The impact on the GMRES(k) performance of this modification is studied here along with preconditioning techniques to improve convergence. Several scalable parallel preconditioners are tested on a wide range of matrices arising in homotopy methods for structural mechanics and circuit simulation problems.

Maria Sosonkina Driver, Donald C. S. Allison, and Layne T. Watson
Computer Science Department
Virginia Politecnic Institute and State University
Blacksburg, VA 24061-0106
Email: masha@csgrad.cs.vt.edu

CP18
PSPARSLIB : A Portable Library of Parallel Sparse Iterative Solvers

PSPARSLIB is a library of FORTRAN subroutines for solving large sparse linear systems in parallel. Its primary goal is to provide modules which can be be used to simplify the development and implementation of sparse iterative solvers for distributed memory computers. The linear system is first partitioned, then split according to the partitioning, a data structure for this distributed sparse matrix is constructed and, finally, a preconditioned Krylov solver is invoked for its solution. Results will be presented for various platforms including an cluster of SGI Power Challenge, a cluster of IBM RS600 workstations an IBM SP-2 and Cray T-3D.

Yousef Saad, Sergey Kuznetsov, Andrew Chapman
University of Minnesota
Department of Computer Science
Minneapolis, MN 55455
saad@cs.umn.edu, kuznesov@msi.umn.edu, chapman@msi.umn.edu

CP18
Improved s-Step GMRES on Massively Parallel

Processors

This paper presents an improved s-step GMRES algorithm for solving general unsymmetric linear systems. This algorithm uses about the same number of operations and memory as required by GMRES. However, it generates a transformed Hessenberg matrix by using block matrix operations and block data communications. The numerical results demonstrate that this algorithm can be more efficient than the standard GMRES on a cache based single CPU computer with optimized BLAS kernels. Furthermore, the gain in efficiency is more significant on MPPs due to both efficient block operations and efficient block data communications. Numerical experiments on some real-world problems also show that the stability of this algorithm is somewhat in-between the standard GMRES and the GMRES with classical (unmodified) Gram-Schmidt implementation depending on s, the size of the block.

Guangye Li
Cray Research, Inc. - a Silicon Graphics company
655E Lone Oak Drive Eagan, MN 55121
E-mail: gli@cray.com

CP18
On the Scalability of Parallel Krylov Subspace Methods

We consider the parallel performance and scalability of several well-known Krylov subspace iterative methods for the solution of linear systems of equations. Different understandings of parallelism are compared and evaluated for our particular setting—the numerical solution of elliptic partial differential equations. Our interest is in scalable parallel mathematical software for such problems. For this work the computational environment consists of Parallel ELLPACK, the Parallel Iterative Methods package (PIM), and MPI. We report experimental results on an Intel Paragon.

Mark Embree
University of Oxford
Mathematical Institute
Oxford OX1 3LB, United Kingdom

Calvin Ribbens
Department of Computer Science
Virginia Polytechnic Institute
and State University
Blacksburg, VA 24061-0106
Email: ribbens@vt.edu

CP18
LINSOL, a Parallel Iterative Linear Solver Package of Generalized CG-Type for Sparse Matrices

The matrix-vector multiplication (MVM) is the key operation for iterative solvers. Efficient parallelization starts with single-processor optimization. The extremely large and sparse matrix is subdivided into 9 different data structures to support pipelining. Cache-reuse is possible by section striping. For the parallelization row blocks are distributed to the p processors. Overlapped processing and sending supports latency hiding. Included are 8 generalized CG methods and 2 polyalgorithms. Results are presented for CRAY T90 and T3E, Fujitsu VPP, NEC SX4 and IBM SP.

Willi Schoenauer, Hartmut Haefner, Ruediger Weiss
Rechenzentrum der Universitaet Karlsruhe
Postfach 6980, D-76128 Karlsruhe, Germany
E-mails: schonauer@, haefner@, weiss@rz.uni-karlsruhe.de

CP19
The Future Fast Fourier Transform?

We expect improvements in arithmetic speed to outpace advances in communications bandwidth. Therefore we propose that an inexact DFT such as an approximate matrix-vector approach or the Dutt-Rokhlin algorithm may outperform any exact parallel FFT under certain conditions. This idea arose from a study of the singular values of the discrete Fourier matrix. The speedup may be as large as a factor of three in situations where FFT run time is dominated by communication.

Alan Edelman and Peter McCorquodale
Department of Mathematics
Massachusetts Institute of Technology, 2-380
77 Mass. Ave., Cambridge, MA 02139-4307
edelman@math.mit.edu, petermc@math.mit.edu

Sivan Toledo
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304
toledo@parc.xerox.com

CP19
An Efficient Load Balancing Technique for Parallel FMA in Message Passing Environment

A tremendous amount of work had been devoted to parallelization of Fast Multipole Algorithm (FMA). Most parallel implementations of FMA perform poorly on irregular particle systems. Partitioning techniques that have been developed result in large communication overhead due to the tight relationship in the translations of multipole and local expansions (TMLE). We present another partitioning technique called *weighted subtrees*. The weighted subtrees scheme partitions the FMA tree into subtrees. These subtrees are assigned to processors one after the other until the workload, measured in terms of the weight of the subtrees, within a processor is close to desired workload. This technique minimized the communication overhead of TMLE and indicated an improved load balancing.

Eric Jui-Lin Lu and Daniel I. Okunbor
University of Missouri-Rolla
Computer Science Department, MCS325
Rolla, MO 65401
Email: jlu@umr.edu

CP19
Parallel Prime Edge-length Multidimensional Symmetric FFTs

We present a fast algorithm for computing the discrete Fourier transform of a symmetric, multidimensional data sample array of prime edge-length. The algorithm, which is built from well-established mathematical specifications, computes the whole multidimensional discrete Fourier transform from a subarray of non-repeated (i.e. fundamental) data avoiding redundant arithmetic and I/O operations but preserving at the same time, the good computational properties of traditional (non-symmetric) FFT. A parallel implementation of our method based on MPI standards is also presented. Our algorithm can be used as a core procedure in x-ray crystal structure determination where several three-dimensional, large scale symmetric dis-

crete Fourier transform are normally required.

Jaime Seguel and Johanna Ortiz
Department of Mathematics
University of Puerto Rico
Mayaguez, PR
Email: jseguel@jacobi.upr.clu.edu

CP19
Fast Solvers for Parallel Adaptive *hp* Finite Element Methods

In this paper we report on the development of a new class of parallel solvers for adaptive *hp* finite element methods. Adaptive *hp* finite element methods,in which grid size and local polynomial order of approximation are both independently adapted, are capable of delivering super-algebraic and even exponential rates of convergence, as seen in the work of Babuska, Oden, Demkowicz and others. In recent work targeting such grids we have developed domain decomposition solvers using two level iterative substucturing with coarse grid type preconditioning. Both theoretical estimates on the condition number showing polylogarithmic bounds of the type $(C(1+\log p)(1+\log(Hp/h)))$. Extensions required for the use of these solvers as inner solvers for either inexact Newton methods for non-linear problems and/or as part of an implicit time integrator for transient problems has been done. Numerical studies also substantiate the performance of these solvers.

Abani Patra
Department of Mechanical Engineering
SUNY-Buffalo
Buffalo, NY 14260
Email: abani@eng.buffalo.edu

CP19
Scalability of the Fast Multipole Method for the Helmholtz Equation

We present scalability results for a single stage Fast Multipole Method for the Helmholtz equation (electromagnetic simulations) running on an SP2 using MPI. Excellent scalability is demonstrated on problems up to $N = 250,000$ unknowns with 1 to 64 nodes. The scalability is due to the fact that the pairwise interactions between N points requires only the sharing of $O(N)$ information. We also show how to reduce the space to $O(N^{5/4})$ which alleviates the need for out-of-core techniques.

Vladimir Rokhlin
Department of Computer Science
Yale University
P.O. Box 2158, New Haven, CT 06520-2158

Mark A. Stalzer
Hughes Research Laboratories
Computational Physics Department RL65
3011 Malibu Canyon Rd., Malibu, CA 90265
Email: stalzer@hrl.com

CP19
A Parallel Fast Direct Solver for the Discrete Solution of Separable Elliptic Equations

A parallel fast direct solver for linear systems with separable coefficient matrices is considered. For example, such systems are obtained by discretizing the two–dimensional Poisson equation posed on rectangular domains with the five–point difference scheme on nonuniform rectangular grids. Although the computational procedure is rather different, the method resembles the cyclic reduction method implemented using the partial fraction expansions. The parallel implementation using the MPI standard is described. In the numerical experiments, the considered method is shown to be, in the sequential case, more than two times faster than the well–known BLKTRI subroutine in the Fishpak library. Also, a good parallel scalability is demonstrated on an IBM SP2 parallel computer.

Tuomo Rossi and Jari Toivanen
Department of Mathematics
University of Jyväskylä
P.O. Box 35, FIN-40351 Jyvaskyla, Finland
Email: toivanen@jyu.fi

CP20
Application of Unstructured Grid Domain Decomposition Techniques to Overset Grids

Overset grids provide a flexible means of generating space-filling grids necessary for several classes of partial differential equations solvers. A new technique for partitioning these grids on distributed architectures is presented. Previous implementations partitioned each of the structured grids separately. This method, on the other hand, partitions the communications graph for the global grid set. Parallel performance of this new technique has proven to be superior to previously-used approaches.

Douglas C. Blake
Air Force Institute of Technology
Wright-Patterson Air Force Base, OH 45433
Email: dblake@afit.af.mil

CP20
Developing Portable, Parallel Unstructured Mesh Simulations

We have developed a programming model and a library of parallel routines (PGSLib) to support parallel unstructured mesh simulations. The model and library was developed for DOE's ASCI project. Current applications using PGSLib include: JTpack90 (a parallel linear algebra package), TELLURIDE (metal casting simulation) and CHAD (high-speed reacting flow simulation). This talk will describe our implementation strategy, show some results and discuss how we are preparing for new parallel computers such as clusters of SMPs.

Robert C. Ferrell
Cambridge Power Computing Associates, Ltd.
2 Still St., Brookline, MA 02146
E-mail: robert@ferrell.com
D. B. Kothe
Los Alamos National Laboratory, Group T-3
Box 1663, MS B216, Los Alamos, NM 87545

J. A. Turner
Los Alamos National Laboratory, Group XTM
Box 1663, MS B226, Los Alamos, NM 87545

CP20
Exploring Structured Adaptive Mesh Refinement (SAMR) Methods with the Illinois Concert System

Structured Adaptive Mesh Refinement (SAMR) simula-

tion methods are attractive because they can increase computational efficiency dramatically. However, their irregular and less predictable computational and data structure makes them challenging to parallelize efficiently on large-scale parallel machines. We are using the Illinois Concert C++ system (which supports dynamic, object-based parallelism) to build a flexible SAMR code for the Cosmology NSF Grand Challenge. The Concert System provides language support to ease the expression of the dynamic parallelism, as well as compiler and runtime support for efficient execution. Our current analysis, performed for the Cray T3D and SGI Power Challenge architectures, indicates that good performance should be feasible with modest programming effort. The paper and presentation will include a performance evaluation of the code's achieved performance, and an evaluation of using the Illinois Concert system (and parallel object systems in general) for this type of computation.

Bishwaroop Ganguly, Greg Bryan, Mike Norman, and Andrew Chien
Department of Computer Science
and National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
ganguly@students.uiuc.edu, gbryan@ncsa.uiuc.edu
and achien@cs.uiuc.edu

CP20
Domain Decomposition Methods for Hyperbolic Problems Based on Additive Schwarz

Additive Schwarz is a preconditioned iterative method of domain decomposition type that has been theoretically proved and experimentally demonstrated to yield asymptotically optimal convergence rates (independent of mesh density and number of subdomains) for elliptic problems. A major practical virtue of additive Schwarz is its scalable data parallelism, with a parallel granularity that grows with the nuumber of subdomains. We extend the Schwarz theory to linear time-dependent hyperbolic problems and prove a convergence rate asymptotically independent of time and space mesh parameters and the number of subdomains, provided that the timestep and the coarse mesh are fine enough.

Yunhai Wu
Computer Science Department
Old Dominion University
Norfolk, VA 23529-0162 USA
wu@cs.odu.edu,

Xiao-Chuan Cai
Computer Science Department
University of Colorado-Boulder
Boulder, CO 80309-0430 USA
cai@cs.colorado.edu

David E. Keyes
Computer Science Department
Old Dominion University
Norfolk, VA 23529-0162 USA &
ICASE, M/S 403 NASA Langley Res. Ctr.
Hampton, VA 23681-0001 USA
keyes@icase.edu

CP20
The Full Domain Partition Approach for Parallel Multigrid on Adaptive Grids

The combination of adaptive refinement, multigrid and parallel computing for solving partial differential equations is considered. In the full domain partition each processor contains a partition of the grid plus the minimum number of additional coarse elements required to cover the whole domain. A parallel multigrid algorithm using the full domain partition is presented. The method obtains multigrid rates of convergence while communicating between processors only twice per V-cycle.

William F. Mitchell
Applied and Computational Mathematics Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001
Email: mitchell@cam.nist.gov

CP20
A Parallel Non-Overlapping Schwarz Domain Decomposition Algorithm for Elliptic Differential Equations

A parallel *nonoverlapping* Schwarz algorithm based on a finite element discretization is applied to convection diffusion problems with fronts and layers. At inter-domain boundaries, a Robin type boundary condition is weakly imposed into the finite element variational form. Grid refinement is made on subdomains that contain fronts and layers and coarse grids are applied on subdomains in which the solution changes slowly and smoothly. Message passing parallel implementations on an nCUBE2 machine are conducted to show the efficiency and accuracy of the method.

Daoqi Yang
Institute for Mathematics and its Applications
University of Minnesota
514 Vincent Hall, 206 Church Street SE
Minneapolis, MN 55455-0436
Email: yang@ima.umn.edu

CP21
Parallelization of the DSN Multigroup Neutron Transport Equation on the Cray-T3D with Craft

Solving the DSN multigroup transport equation is typically sequential, and its parallelization is difficult.We explain how we changed the usual algorithm into a parallel iterative method, adapted to the Cray-t3d ; the new iterative method will be described in details. We will then show some numerical results obtained on the Cray-t3d.

Francoise Coulomb
CEA/Limeil-Valenton
94195 Villeneuve-Saint-Georges cedex
France
E-mail : coulomb@limeil.cea.fr

CP17
Coupled Ocean/Atmosphere Modeling on High-Performance Computing Systems

We investigate performance of a coupled ocean/atmosphere general circulation model on high-performance computing systems. Our programming paradigm has been domain decomposition with message-passing, though with the emergence of SMP clusters we are investigating how to best support that architecture as well. We consider how to assign processors to the major model components so as to obtain

optimal load balance. We examine throughput on contemporary parallel architectures, such as the Cray-T3D/T3E, IBM-SP family, and Intel Paragon/Tflops.

A. A. Mirin, J.H. Bolstad, P.B. Duffy,
P.G. Eltgroth, H. Wang, and M.F. Wehner
Lawrence Livermore National Laboratory
P.O. Box 808, Livermore, CA 94550
mirin@natali.llnl.gov

CP21
A Highly Stable Explicit Technique for Stiff Reaction-Transport PDEs

The numerical simulation of chemically reacting flows is a topic that has attracted a great deal of current research. At the heart of numerical reactive flow simulations are large sets of coupled, nonlinear Partial Differential Equations (PDEs). These PDE systems are almost always stiff. Thus, explicit time differencing schemes must generally be abandoned despite their simplicity and efficiency on parallel and vector computers. Implicit time stepping schemes, while possessing excellent stability characteristics, are not as efficient on parallel machines and introduce a large amount of computational overhead devoted to solving the resulting simultaneous system of algebraic equations. We present an algorithm which is explicit and, at the same time, permits a very large stable timestep. The algorithm is derived from a preconditioned time differencing (or dynamic iteration) scheme and will be analyzed on a small system of reactive flow PDEs.

Colin J. Aro
Scientific Computing Applications Division
Lawrence Livermore National Lab
Livermore, CA 94551
E-mail: aro@llnl.gov

Al Franz
Department of Mathematics and Computer Science
California State University, Hayward
Hayward, CA 94542

Dale Slone
Scientific Computing Applications Division
Lawrence Livermore National Lab
Livermore, CA 94551

CP21
Parallel Implementation of Hydrostatic MM5 (Mesoscale Model)

This paper describes the parallel implementation of the hydrostatic MM5 (Mesoscale Model), which has been widely used for climate studies. The parallel model was tested on the IBM SP1, a distributed memory parallel computer, at Argonne National Laboratory. We parallelized the model in machine-independent manner using the RSL (Runtime System Library), a library for handling message-passing and index transformation. The paper discusses validation of the parallel implementation and presents performance results.

Youngtae Kim
Department of Computer Science
Iowa State University
Ames, IA 50011
kim@cs.iastate.edu

Zaitao Pan and Eugene S. Takle
Department of Atmospheric Science
Iowa State University
Ames, IA 50011
panz@iastate.edu, gstakle@iastate.edu

Suresh C. Kothari
Department of Computer Science
Iowa State University
Ames, IA 50011
kothari@cs.iastate.edu

CP21
Parallel Implementation of the Coupled Fluid-Structure Acoustic Scattering Problem

The solution of the three dimensional acoustic scattering problem in an infinite fluid domain requires significant computational work. The sparse symmetric dynamic stiffness matrix is factorised and substituted into a fully populated fluid matrix to solve for the object's exterior surface pressure. The paper will briefly describe the acoustic theory and discuss the efficient parallel implementation of this problem. Results will be presented for large coupled problems for 1000's fluid degrees of freedom and 10,000's of structural degrees of freedom.

Steven J. Newhouse and Ian C. Mathews
Structural Acoustics Group
Department of Aeronautics
Imperial College
Prince Consort Road
London, SW7 2BY, United Kingdom
E-mail:s.newhouse@ic.ac.uk

CP21
Massively Parallel Reacting Flow Calculations for Optimization of Chemical Vapor Deposition Reactors

Efficient and robust massively parallel algorithms are used to solve 3D reacting flow applications. An unstructured grid finite element method is implemented in the code MP-Salsa to solve incompressible flow coupled with heat and mass transport. A fully-coupled inexact Newton method with backtracking in conjuntion with a parallel preconditioned Krylov linear solver (**Aztec**) allow the direct calculation of steady-states. The quick solution time enables the optimization of a Chemical Vapor Deposition reactor model.

Andrew G. Salinger, John Shadid, Harry Moffat,
Scott Hutchinson
Karen Devine, Gary Hennigan, William Hart
Sandia National Labs, Albuquerque, NM 87109
Email: agsalin@cs.sandia.gov

MS01
Parallel Software Engineering

The proliferation of new parallel programming models and languages has led users to demand: "No more programming paradigm shifts." This talk will focus on certain aspects of paradigm shift-free parallel programming. The new KAP/Pro Toolset will be used to illustrate a parallel software engineering discipline that is designed to minimize new-paradigm thinking. The use of directives with automatic restructuring, parallel profiling, performance vi-

sualization, parallelism validation, and run time portability will be discussed.

David J. Kuck
Kuck and Associates, Inc.

MS01
Research and Practice of Compilers for Parallel Machines

This talk presents an overview of some recent results and promising directions in research on compilers for parallel machines. Research in compilers produces general and powerful abstractions which is directly useful for managing the complexity of compiler development. This talk also discusses the transfer of technology from research to practice, and describes a new effort to develop a royalty-free common compiler infrastructure for both researchers and practitioners.

Monica Lam
Stanford University

MS01
Parallel Programming Paradigms for Future Architectures

High performance architectures are becoming increasingly complex, providing a heterogeneous mixture of shared and distributed memory models. On the other hand, applications requiring such computing power are also becoming more complex exhibiting multiple types of parallelism. This talk will first discuss the adequacy of the current languages and compilers to effectively exploit this parallelism and then will discuss possible programming paradigms for future computing environments.

Piyush Mehrotra
ICASE NASA Langley Research Center

MS01
Compiling Explicitly Parallel Programs

Two of the most popular programming models, message passing SPMD and shared memory threads, have received relatively little attention from the compiler research community. These explicitly parallel models offer new challenges in program analysis related to shared variables, synchronization, and communication. Meanwhile, increasingly popular clusters of SMPs offer new opportunities for optimization within the deep and often semantically complex memory hierarchy. In this talk I will present some of these challenges and recent results in the area.

Katherine Yelick
University of California, Berkeley

MS02
Parallel Graph Partitioning

Recently, a number of graph partitioning algorithms have been developed that are based on the multilevel graph partitioning that have moderate computational complexity. Some of these multilevel schemes provide excellent partitions for a wide variety of graphs. Even though multilevel graph partitioning algorithms produce high quality partitions in relatively small amount of time, the ability to perform partitioning in parallel is important particularly in the context of adaptive grid computations that dynamically adjust the discretization of the physical domain. We provide an overview of the number of issues involved in developing parallel formulations of multilevel graph partitioning algorithms, and discuss some possible solutions.

George Karypis
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455

MS02
A Local Graph Partitioning Heuristic Meeting Bisection Bounds

The problem of graph partitioning is of major importance for a broad range of applications. VLSI-Layout and parallel programming are only two examples where large graphs have to be cut into pieces. The partitioning of a graph into several clusters is often done by recursive bisection. Bisection heuristics are usually divided into global and local methods. Popular global methods are inertial or spectral partitioning. For a long time, the Kernighan-Lin algorithm (KL) was the only efficient local heuristic and is still widely used in several applications. Based on the idea of so called *Helpful Sets*, we recently proposed a new local graph bisection heuristic. Helpful Sets can be used to prove upper bounds on the bisection width of regular graphs. Thus, we are able to give upper bounds on the quality of the bisections found by HS. In this paper, we present the HS-idea, the bisection heuristic, ideas of the proofs of upper bounds, and, finally, discuss the performance of the method compared to other heuristics.

Burkhard Monien and Ralf Diekmann
Department of Mathematics and Computer Science
University of Paderborn, Fuerstenallee 11,
D-33102 Paderborn, Germany
e-mail: {bm, diek}@uni-paderborn.de

MS02
MLB: Multilevel Load Balancing for Structured Grid Applications

The Multilevel Load Balancing algorithm (MLB) is a parallel algorithm that determines the communication schedule necessary to balance a distributed discrete multidimensional load function. MLB focuses upon structured grid computations and is used for the load balancing of structured grid computations and particularly structured grid adaptive mesh refinement computations. Though the implementation is involved the computation associated with MLB is relatively simple and the communication complexity of the algorithm and the communication schedule it produces is bounded by the log of the number of processors.

Daniel J. Quinlan and Markus Berndt
Los Alamos National Laboratories
Scientific Computing Group
Mail Stop B256
Los Alamos, NM 87545

MS02
Spreading Metric Based Graph Partitioning Algorithms

A fuller version of this paper appears in the Proc. 8th ACM-SIAM Symp. on Discrete Algorithms. Supported by

GSF grants ABC123, DEF456, and GHI789.
We present a new and simple $O(\log n)$ approximation algorithm for finding minimum capacity ρ-separators. A ρ-separator is a subset of edges whose removal partitions the vertex set into connected components such that the sum of the vertex weights in each component is at most ρ times the total vertex weight. This yields $O(\log n)$ approximation algorithms for finding b-balanced cuts and k-multiway separators (for most values of b and k). The algorithm is based on a powerful technique called spreading metrics that enables us to formulate the minimum capacity ρ-separator problem as an integer program, and use the fractional solution of the respective linear program to achieve the approximation.

Guy Even
Univ. des Saarlandes,
FB 14 Informatik, LS Prof. Paul,
66123 Saarbrücken, Germany.
guy@cs.uni-sb.de

Joseph (Seffi) Naor
Computer Science Dept., Technion,
Haifa 32000, Israel.
naor@cs.technion.ac.il
Research supported in part by Grant No. 92-00225 from the United States-Israel Binational Science Foundation (BSF), Jerusalem, Israel.

Satish Rao
NEC Research Institute
4 Independence Way
Princeton, NJ 08540.
satish@research.nj.nec.com

Baruch Schieber
IBM T.J. Watson Research Center,
P.O. Box 218, Yorktown Heights, NY 10598.
sbar@watson.ibm.com

MS03
Building Flexible Large-Scale Scientific Computing Applications with Scripting Languages

We describe how we have used scripting languages such as Python, Tcl, and Perl to build highly modular, extensible, and interactive scientific applications for massively parallel supercomputers and workstations. Scripting languages provide an extremely flexible mechanism for supporting interactivity, batch processing, rapid prototyping, and debugging. We also describe SWIG, a tool that automatically generates scripting language interfaces, a parallelized version of the Python language we have developed for use on MPPs, and how we have applied these tools to a large-scale molecular dynamics application.

David M. Beazley
University of Utah, Salt Lake City, UT

Peter S. Lomdahl
Los Alamos National Laboratory
Los Alamos, NM

MS03
Drug Design in a Virtual Environment Using the WorkSpace VR Windowing Toolkit

Biomolecules are complex three-dimensional geometric structures built from thousands of atoms. By simulating the forces between atoms, researchers are beginning to be able to predict how well small candidate drug molecules can bind to these giant structures. Successful prediction of a new lead for drug development not only involves reasonably accurate modeling of molecular forces but also an understanding of the shape of the active site pocket and a good deal of chemical intuition not easily codified within a computer algorithm. Live interaction with a simulation while it is happening brings human pattern recognition skills, expert knowledge and adaptability into the picture. Our molecular simulation of a drug molecule binding at the active site of an enzyme is an example of true interactive steering in which hand-eye coordination is preserved. The application is build on the WorkSpace Toolkit, a C++ class library developed at the Cornell Theory Center for virtual reality application development. WorkSpace is a true Java(TM)-like windowing system that handles parent-child events and allows the user to launch and manage multiple applications from within the virtual environment. Video examples captured from the Cornell Theory Centers virtual reality facility, the Visual Insight Zone, will be shown.

Richard E. Gillilan
Cornell Theory Center,
Cornell University,
Ithaca, NY

MS03
Web Technology for Collaboration and Interactive Visualization

Explosive growth of the Web engulfs many of the computer technology areas in unanticipated ways. After reducing barriers to information access in an unprecedented fashion, Web begins to be populated by tools promoting Direct, real time collaboration between remote users. In our paper we present design, implementation, and efficiency analysis of Tango, a collaboratory system created in Norheast Parallel Architectures Center at Syracuse University. Tango, written entirely in Java, consists of two parts: the runtime system, supporting passing of arbitrary events and data between multiple instances of Web applications and providing complete session management, and the API that allows to convert any application, either a Java applet, Java application, or an application written in any other language, into a collaboratory tool. We will discuss system design priciples, briefly overview its implementation, and present a number of applications ported or interfaced to Tango.

Marek Podgorny
Syracuse University , Syracuse, NY

Geoffrey C. Fox
Syracuse University
Syracuse, NY

MS03
MultiMATLAB: Extending the MATLAB Problem-Solving Environment to Multi-Processors

MATLAB is a high-level language, and a problem-solving environment, for mathematical and scientific calculations. In many of the scientific and engineering communities it has become the dominant language for desktop numerical computing. MultiMATLAB is a system that enables one to run MATLAB conveniently on multiple processors. We will describe the system and applications built using the

tool on the IBM SP2.

Anne E. Trefethen
Cornell Theory Center,
Cornell University
Ithaca, NY

Lloyd N. Trefethen
Computer Science Department,
Cornell University
Ithaca, NY

MS04
Parallel Supernodal Method for Sparse Gaussian Elimination

We discuss the development of SuperLU, a new sparse Gaussian elimination routine for workstations and SMPs. A variety of innovations, including some closely tied to cache and superscalar architectures in modern workstations, lets serial SuperLU attain up to 40the sparse matrix. Parallel SuperLU attains up to 7-fold speedup on an 8-CPU AlphaServer 8400, up to 6-fold speedup on an 8-CPU C90, and up to 6-fold speedup on an 8-CPU SGI Power Challenge.

Xiaoye S. Li
Lawrence Berkeley National Lab
Berkeley, CA

James W. Demmel
University of California
Berkeley, CA

John R. Gilbert
Xerox Palo Alto Research Center
Palo Alto, CA

MS04
Sparse Direct Solution on Distributed Memory Machines

We consider the direct solution of sparse systems where the matrix is large and symmetric positive definite. We examine several issues related to the overall efficiency of a fully parallel solver and its use within a parallel application. The relative proportion of time taken by each of the four major steps in the solution process is quite different from that for serial solution and has implications for overall efficiency. We consider instances when the ordering is performed once and the system is factored and solved several times, and when the ordering and factorization are performed once and the system is solved for several right-hand-side vectors. We also consider different initial data distribution schemes before ordering, and the effect of locality, such as that available within a parallel application where the solver is the main computation kernel. With parallel nested dissection, data can be redistributed after identifying as many submatrices as the total number of processors. This is crucial for the efficiency of the remaining part of ordering and the next symbolic factorization step. However, after symbolic factorization, data can be redistributed yet again for the numeric steps using a measure based on operation counts or nonzeroes in the factor. We consider the trade-offs between improved performance and the cost of redistribution. We also consider the effects of machine characteristics (such as the communication bandwidth and latency compared to the execution rate) on each step in the four-step solution process. We give performance results for our solver on Intel and IBM multiprocessors. The solver uses improved versions of "Cartesian nested dissection" and "parallel contracted ordering" and a multifrontal factorization which can use one- or two-dimensional mapping of dense submatrices in the distributed phase. The triangular solution step can be performed using distributed substitution schemes. Alternatively, when solving for several right-hand-side vectors, submatrices obtained from "selective inversion" are used to replace distributed substitution schemes by efficient matrix-vector multiplication kernels.

Michael T. Heath
Department of Computer Science and NCSA
University of Illinois
1304 West Springfield Ave., Urbana, IL 61801
e-mail: heath@cs.uiuc.edu

Padma Raghavan
107 Ayres Hall
Department of Computer Science
University of Tennessee, Knoxville, TN 37996-1301
e-mail: padma@cs.utk.edu

MS04
Effective Sparse Matrix Ordering: Just Around the BEND

When performing sparse matrix factorization, the ordering of matrix rows and columns has a dramatic impact on the factorization time. This paper describes an approach to the reordering problem that produces significantly better orderings than previous methods. The algorithm is a hybrid of nested dissection and minimum degree ordering, and combines an assortment of algorithmic techniques.

Bruce Hendrickson
Sandia National Laboratories

Ed Rothberg
Silicon Graphics, Inc.

MS04
A Parallel Sparse Direct Solver for Least Squares Problems

We describe the design and performance of an efficient parallel sparse matrix package for solving large-scale linear least squares problems on message-passing machines. This package implements both the QR factorization method and the method of corrected semi-normal equations for least squares problems. We discuss three aspects of our parallel sparse solver: the design of the underlying parallel sparse matrix algorithms, the design of the user-friendly interface that is simple, flexible and portable, and the performance results obtained on an IBM Scalable POWERparallel System SP2. Effective parallel strategy for handling dense rows in a sparse least squares problem will be presented.

Chunguang Sun
Advanced Computing Research Institute
Cornell Theory Center
Cornell University
Ithaca, NY 14853-3801

MS05
Ewald and multipole methods for periodic N-body

problems

Many realistic biomolecular simulations require use of periodic boundary conditions to create a surface-free environment for the molecule of interest and associated solvent molecules to interact. Electrostatic interactions are the principal computational cost of such simulations. We have implemented two codes: a parallel variant of an Ewald summation method which computes the effect of infinite periodic bounary conditions, and a parallel variant of a multipole algorithm which explicity computes the interactions within a large but finite periodic system. Each has a regime of applicability, with Ewald favoring smaller systems and fewer processors, and the multlipole methods favoring larger systems and more processors.

John A. Board, Jr.
Christopher W. Humphres
Christophe G. Lambert
William T. Rankin
Abdulnour Y. Toukmaji
Duke University, Durham, NC

MS05
Particle-Mesh Based Methods for Fast Ewald Summation in Molecular Dynamics Simulations

The development of the smooth Particle Mesh Ewald (SPME) method will be briefly reviewed. Some recent performance enhancements will then be reviewed, including the implementation of PME on massively parallel computers. Next, SPME will be compared to the original Hockney-Eastwood PPPM method. Finally, we derive a class of optimal PME methods, which match the accuracy of the PPPM using however, much simpler "influence" functions, which can be computed on the fly.

Tom Darden
NIEHS/NIH, RTP NC

Lee Pedersen
University of North Carolina
Chapel Hill, NC

Abdulnour Toukmaji
Duke University, Durham, NC

Mike Crowley
Pittsburgh SuperComputing Center
Pittsburgh, PA

MS05
A Data–Parallel Adaptive N–body Method

We present a data–parallel formulation of the 3–D Anderson's method for adaptive N–body simulations. Our formulation consists of an efficient array–based representation for the adaptive hierarchy that models the dynamically moving particles, and a data–parallel implementation (in HPF) of various partitioning methods. These partitioning methods balance nodal weights (for computation) but GEO and RSB further guarantee, with some additional cost, that the size of edge–cut for nonuniform particles is as low as that for uniform particles. Load balancing for nonuniform particle distributions requires uneven partitioning of the arrays – a feature not currently supported in HPF–I. We present methods for handling this problem in HPF–I using extrinsic procedures and suggest some new language features desired in the revision of HPF for supporting irregular computation. Experimental results about these partitioning schemes and the complete code in adaptive particle simulations will be presented.

Y. Charlie Hu
Harvard University, Cambridge, MA

S. Lennart Johnsson
Harvard University, Cambridge, MA
University of Houston, Houston, TX

Shanghua Teng
University of Minnesota, Minneapolis, MN

MS05
Parallel, Out-of-core Methods for Fast Evaluation of Long-range Interactions

Hierarchical treecodes have, to a large extent, converted the compute-bound N-body problem into a memory-bound problem. The large ratio of DRAM to disk pricing suggests use of out-of-core techniques to overcome memory capacity limitations. We will describe a parallel, out-of-core treecode library, targeted at machines with independent secondary storage associated with each processor. Borrowing the space-filling curve techniques from our in-core library, and "manually" paging results in excellent spatial and temporal locality and very good performance.

John Salmon
California Institute of Technology
Pasadena, CA

Michael S. Warren
Los Alamos National Laboratory
Los Alamos, NM

MS06
Task Graph Rescheduling for Parallelization of Irregular and Dynamic Computations

Many real world scientific computing problems are irregular and dynamic, requiring sophisticated software support systems. In this paper we discuss the approach of applying task graph rescheduling method to the efficient parallelization of such applications. We study several representative dynamic applications, such as the N-body simulation and related problems in CFD. We start with an initial global compile-time scheduling, and apply new rescheduling algorithms to improve performance when this schedule degenerates over the iterative process. Two new rescheduling algorithms are developed: One for task graph weight variation, the other for dynamic spawning of subgraphs. These algorithms are localized and incremental, with very low complexity, yet they yield schedules that are competitive to global rescheduling from scratch. The algorithms are tested on random and real applications such as the N-body and Vortex Sheet simulation in CFD with considerable success. Our experiments show that global scheduling using sophisticated methods can reduce the overall parallel time, and our fast rescheduling algorithms can correct run-time imbalance with very low cost.

Apostolos Gerasoulis
Rutgers University
Piscataway, NJ 08854

Jia Jiao
University of Arkansas

Little Rock, AR 72204

MS06
Run-time Support for Dynamic Processor Allocation in HPF Programs

To allow parallel programs to adjust to internal processor requirements and/or to adapt to changes in the external run-time environment, it is desirable to execute parallel applications on time-variant processor partitions – partitions that can change their size over time. HPF programs have a well-defined sequential semantics and data distribution annotations, which combined can support this type of execution. HPF, however, does not provide abstractions for the dynamic manipulation of processors by the application. In this paper, we propose additions to HPF annotations to help the compiler in automatically generating a parallel program to execute on a time-variant processor partition. Implementation of such execution mode requires extensive cooperation between the compiler, the run-time system, and the operating environment. The compiler should be able to generate code that executes correctly on a time-variant partition and that allows the run-time system to perform data redistributions whenever there is a change in the processor configuration. Similarly, the operating environment should provide facilities for acquiring and releasing processors dynamically, and migrate execution of the program from one set of processors to a different set. In this paper, we describe the DRMS run-time system that allows implementation of these operations. We also describe the interactions of the run-time system with the compiler and operating environment. Our discussion is centered on the IBM SP2 and HPF Compiler products, but the concepts are applicable to many other systems.

Sam Midkiff
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

José E. Moreira
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

Vijay K. Naik
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

MS06
Nanothreads: Fast, Adaptive User-Level Threads for Distributed Shared Memory

We approach multithreading based on user-level threads not only as an effective means of parallelism exploitation and hiding memory latency on DSM systems, but also as a powerful space- and time-sharing mechanism. In order to take full advantage of software-based multithreading, a concerted effort at the architecture, compiler, and operating system levels is necessary. Such is the approach we adopted in the design and implementation of Nanothreads, which implements ultra-light-weight threads in user-space, and incorporates sophisticated but fully distributed thread management and scheduling techniques. We present the architecture of the Nanothreads runtime system and its implementation on the Intel Quad P6 and the SGI Origin DSM systems.

Constantine D. Polychronopoulos
Center for Supercomputing Research and Development
University of Illinois at Urbana-Champaign
1308 W. Main St.
Urbana, IL 61801
cdp@csrd.uiuc.edu

MS06
Supporting Dynamic Reconfiguration of Parallel Applications on Clusters of Non-dedicated Workstations

Parallel applications executing on workstation clusters must be fault-tolerant in order to survive network and workstation failures, and must have the ability to withdraw from a workstation on detecting owner activity. In addition to these properties, *reconfigurable* parallel applications have the ability to dynamically shrink or expand their degree of parallelism. Preliminary research indicates that this property leads to significant performance advantages on clusters of non-dedicated workstations because of the unpredictable nature of the available processing capacity in these environments. In this paper, we describe an approach that uses application-level checkpointing and data repartitioning for supporting dynamically reconfigurable loosely synchronous parallel applications.

Sanjeev Setia
Department of Computer Science
George Mason University
Fairfax, Virginia 22030

Abdur Chowdhury
Department of Computer Science
George Mason University
Fairfax, Virginia 22030

Lisa Nicklas
Department of Computer Science
George Mason University
Fairfax, Virginia 22030

Elizabeth White
Department of Computer Science
George Mason University
Fairfax, Virginia 22030

MS07
Parallel Debugging Tools for Scientific Code Development

A fully functioning parallel debugger is an essential tool for the development and analysis of parallel scientific programs. The complexity and validation of scientific programs is limited by the sophistication of the tools available to use in the development process. Debugging codes in the presence of increasing algorithm and hardware complexity requires a new generation of development tools. Advanced features such as distributed data visualization and computational steering integrated with the debugger needs to become commonplace. This presentation will describe user debugging needs for high performance parallel code development as determined in the ASCI program, survey the current state of parallel debugging tools including Dolphin TotalView, TMC Prism, and the NASA Ames P2D2 debuggers, and suggest areas where further research and development is required.

Jeff Brown
Los Alamos National Lab

MS07
Runtime Tools for Scientists

Parallel runtime tools perform tasks such as scheduling, checkpointing, interactive steering and visualization. Because these tools have the potential of being used on every run rather than to initially debug or tune an application, it is important that the tools be designed with scientists in mind. This presentation will address three issues regarding parallel runtime tools: What are the scientist's needs? Why don't they use existing tools? And what runtime tools are out there both research and commercial?

Al Geist
Oak Ridge National Lab

MS07
Development Tools for Parallel Applications

The process of developing the code for a parallel application involves a number of repetitive, error-prone tasks that could be facilitated greatly through tool support. This presentation examines what kinds of program analysis and restructuring activities are supported by currently available tools; tools for building executables from complex code hierarchies are also discussed. Existing tool are limited to certain computers, certain programming languages, and certain situations. The tradeoffs of more generalized approaches — tools that can span multiple architectures and programming paradigms — will also be addressed.

Cherri Pancake
Oregon State University

MS07
Performance Analysis Tools: Reaching Higher

Performance analysis tools provide users with information about system and application behavior during execution. The information comes in diverse forms depending on what is being monitored and how it is being monitored. One of the most challenging aspects of performance analysis is presenting the information at a level of abstraction that is meaningful to the scientist. Moreover, tools should provide the scientist with some insights on tuning an application. The role of performance analysis tools is complicated by emerging computing and problem-solving environments. This presentation will consider the capabilities and limitations of current performance analysis tools with respect to the needs of scientists in several scenarios.

Diane T. Rover
Michigan State University

MS08
Graph-Partitioning Based Sparse Matrix Orderings for Interior-Point Algorithms

A key computational step in interior-point (IP) methods for solving linear programming (LP) problems is the solution of a sparse symmetric system of linear equations. The time and memory requirements and the parallelizability of this step depend on the initial ordering of rows and columns in the sparse matrix. Traditionally, a heuristic known as *multiple minimum degree* (or one of its variants) has been almost universally employed by the LP community for ordering sparse matrices arising in an IP code. In this talk, we show that, with suitable modifications and improvements, the long neglected graph-partitioning based approach to sparse matrix ordering can be much more suitable for both serial and parallel IP implementations.

Anshul Gupta
P. O. Box 218
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

MS08
Fully Parallel Generic Branch-and-Cut

We describe a fully parallel implementation of the branch-and-cut algorithm for integer programming. Parallelism is exploited through simultaneous investigation of search tree nodes and through interleaving LP solution and cut generation at each node. The implementation is generic in that for a specific application, the user need only supply preprocessing and separation functions; the remaining components of the implementation, including inter-process communication, search tree management, LP solution, and cut pool maintenance, are entirely internal. Additional distinctive features of the implementation include the capability of branching on cuts, the use of multiple cut pools servicing different parts of the search tree, and a graphical user interface designed to enable interactive cut generation. This code has been used for computational investigation of three distinct combinatorial optimization applications: traveling salesman, vehicle routing, and set partitioning (crew scheduling) models.

This work is supported by NSF through grant DMS-9527124 funding the Advanced Computational Optimization Laboratory (ACOL) of the Cornell Theory Center.
Márta Eső and László Ladányi
School of OR&IE, Cornell University
Ithaca, NY 14853

Theodore K. Ralphs
U.S. Air Force, Nellis Air Force Base
Las Vegas, NV 89030

Leslie E. Trotter, Jr.
School of OR&IE, Cornell University
Ithaca, NY 14853

MS08
pPCx: A Parallel Interior Predictor-Corrector Method for Linear Programming: Implementation and Performance

We present a parallel implementation of a primal–dual infeasible interior point method for linear programming. The underlying algorithm is based on Mehrotra's predictor–corrector idea, a serial implementation was released at Argonne National Labs as PCx. Most of the work in an interior point method lies in solving a symmetric sparse positive (semi–) definite system of linear equations: we present a new parallel multifrontal Cholesky factorization that efficiently handles dense rows and columns as well as near-degeneracies. The constraint matrix is stored in a distributed form, thus enabling the solution of very large problems that cannot be solved on a single processor. We will discuss implementation details and report performance results obtained at the Cornell Theory Center on an IBM SP2.

Thomas F. Coleman
Computer Science & Center for Applied Mathematics
Cornell University

Chunguang Sun
Cornell Theory Center (ACRI)
Cornell University

Michael Wagner
Operations Research and Cornell Theory Center (ACRI)
Cornell University
Stephen Wright
Mathematics and Computer Science Division
Argonne National Laboratories

MS09
Where should Collective I/O be Performed? File System or Runtime System

Collective I/O is a mechanism to permit reorganization of I/O requests to improve I/O performance in parallel computers. Originally generalized collective I/O was proposed (e.g., PASSION) at the runtime system level, whereas some specific patterns were implemented at the file system level (e.g., M-Record mode in Intel PFS). There are tradeoffs in performance, interface, flexibility and ease of use in performing collective I/O at different levels. In this talk we will discuss the tradeoffs in implementing collective I/O at different levels, in particular at the file system level and the runtime system level.

Alok Choudhary
Northwestern University
Evanston, IL 60208

MS09
Performance Modeling of a Parallel I/O System: An Application Based Approach

The broadening disparity in the performance of I/O devices and the performance of processors and communication links on parallel platforms is the primary obstacle to achieving high performance in many parallel application domains. To bridge this gap, parallel file systems must offer scalable I/O performance. In this paper, we investigate how the I/O configuration of the parallel file system may influence the application execution time. We claim that understanding the interaction of the application access pattern and parallel I/O software and hardware is necessary to select the disk parallelism (i.e., the number of disks the file is striped upon) that is most beneficial to application performance. To validate this claim, we construct a series of I/O benchmarks that encapsulate the behavior of a class of I/O intensive access patterns. Experimental measurements on the Intel Paragon demonstrate the conditions under which preferred striping configurations exist. Based on this experience, we propose a simple, product form queuing network model that effectively predicts both benchmark and I/O nsive scientific application execution times across various configurations.

Evgenia Smirni
University of Illinois, Urbana-Champaign
Computer Science Department
1304 W. Springfield Avenue
Urbana, IL 61801-2987

MS09
Portable Implementation of Parallel-I/O APIs

The lack of a standard, portable application-programming interface (API) for parallel I/O has been a major impediment to writing truly portable parallel programs. However, the MPI-IO standard, being defined by the MPI Forum, may be a solution to this problem. In this talk, I will describe a framework I have developed, called ADIO, for implementing parallel-I/O APIs, such as MPI-IO, portably and efficiently on multiple file systems. I am currently using ADIO to implement MPI-IO on all parallel computers and networks of workstations. I will describe my experience in implementing MPI-IO and discuss associated performance issues.

Rajeev Thakur
Mathematics and Computer Science Division
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439

MS10
Modeling the Cost of Redistribution in Scheduling

Distributed heterogeneous resources coordinated by fast networks provide a new platform for resource-intensive parallel applications. However, careful scheduling of such applications is required to achieve their performance potential. Since resource load varies dynamically, the scheduling mechanism may consider redistribution of application tasks to improve performance. In this paper, we focus on the development of dynamically parametrizable models to determine the cost (in terms of execution delay) of redistribution. When such models are combined with predictive models for application execution, a scheduler can determine whether redistribution is profitable at a given point in time.

Gary Shao, Rich Wolski, and Fran Berman
Department of Computer Science and Engineering
University of California, San Diego

MS10
Extensible Resource Scheduling for Parallel Scientific Applications

A large/important class of parallel scientific/engineering applications requires the simultaneous scheduling of its tasks on computing nodes due to the synchronization/communication among these tasks. In this paper, we present a new approach for scheduling such applications in parallel processing systems, and we describe an implementation that is portable, fault-tolerant (it can recover from resource failures), and extensible (new scheduling paradigms/techniques are easily incorporated). Results from the IBM SP2 and a cluster of workstations, under a variety of different scientific workloads, demonstrate that our scheduling software is efficient and provides significant performance improvements over existing scheduling approaches in a controllable manner. Our scheduling software also provides fault-tolerance services to applications so that they can recover from resource failures.

Nayeem Islam
IBM T. J. Watson Research Center, Yorktown Heights, NY

Andreas Prodromidis
Columbia University, New York, NY

Mark Squillante
IBM T. J. Watson Research Center, Yorktown Heights, NY

Ajei Gopal
IBM T. J. Watson Research Center, Yorktown Heights, NY

Liana Fong
IBM T. J. Watson Research Center, Yorktown Heights, NY

MS10
Scheduling in a High Performance Remote-Sensing Data Server

Many applications access large volumes of multi-dimensional data, but often generate output images whose sizes are much smaller than that of the input data. This makes it imperative, from a performance perspective, to process the input data on the same machine that the data is stored on. Careful scheduling from the database server that manages the data is required to overlap computation, communication and I/O effectively from multiple queries to maximize overall performance in a multi-user environment.

Chialin Chang, Alan Sussman, and Joel Saltz
Department of Computer Science and UMIACS
University of Maryland, College Park 20742

MS10
Optimal Scheduling of Coarse-Grained Parallel Scientific Applications

A large/important class of parallel scientific/engineering applications consists of coarse-grained independent parallel tasks that do not require simultaneous resource allocation. In this paper, we present an optimal method for scheduling system resources among such applications in parallel processing systems. We then use validated analytic models to compare the performance provided by the optimal policy with that provided by existing scheduling methods. Our results demonstrate the significant performance benefits of this optimal policy which increase sharply as the number of nodes in the system grows. We also describe our early experiences with implementing the optimal policy in a scheduling software system on UNIX.

Mark S. Squillante
IBM T. J. Watson Research Center, Yorktown Heights, NY

Konstantinos Tsoukatos
University of Maryland, College Park, MD

MS11
Attributes of Molecular Dynamics Calculations: Accounting for CPU Cycle

To shed light on computational issues relevant to molecular dynamics (MD) performance, we overview features in programs that simulate molecular systems of increasing complexity. We also investigate algorithm and parameter effects on workload and program structure. For example, we have found in our study using the Cray C90 that the percentage of branch statements executed in a simulation of a protein in water can range from 5.8% to 9.9% by varying the pairlist update frequency.

Terry W. Clark
Pacific Northwest National Laboratory
P.O. Box 999, Mailstop K1-90
Richland, WA 99352

MS11
Improved 3D Reconstruction of Virus Structures through Parallel Processing

Three-dimensional reconstruction from electron cryomicroscopy images is a technique for determining macromolecular structures of complexes such as spherical viruses with icosahedral symmetry. To calculate higher resolution structures, it is necessary to combine hundreds of extremely noisy images. Complexity increases quadratically with the number of images, making reconstruction from hundreds of particles impractical on conventional workstations. We discuss a parallel shared-memory implementation of the most intensive calculation. Speed-ups obtained lead to reconstructions with measurably improved resolution.

K. H. Haskell
University of Houston, Houston, TX 77204-3476
Z. H. Zhou
University of Houston, Houston, TX 77204-3476
H. Spears, Jr.
Silicon Graphics, Inc., Houston, TX 77077
W. Chiu
Baylor College of Medicine, Houston, TX 77030
L. R. Scott
University of Houston, Houston, TX 77204-3476

MS11
Task and Data Parallel Solution of Elliptic Systems Arising in Biological Modeling

An elastic-electrostatic model of biological membranes and other biostructures is derived from a generalized Debye-Hückel electrostatic model and large deformation elasticity theory. This model leads to a coupling of the nonlinear Poisson-Boltzmann equation and nonlinear elasticity equations. Discretizations and data-parallel multilevel methods are briefly discussed. Several algorithms are then presented for handling the coupled system, some of which have interesting task-parallel features. Parallel implementations in C++ and CC++ are discussed.

A. Klein
University of Washington, Seattle, WA

M. Holst
Caltech, Pasadena, CA

C. Kesselman
Caltech, Pasadena, CA

MS11
The Parallel Fast Multipole Method for Computational Chemistry

A scalable shared memory implementation of the fast multipole method (FMP) for computational chemistry is presented. Restricting attention to salt melts allows to concentrate on the properties of the FMP. The performed computer experiments imply that for accurate results much higher multipole expansion depths than in astrophysical applications are required. Expansion depth requirements for accurate time stepping appear especially stringent. Some simple computer experiments can aide in the selection of a sufficient expansion depth.

Jürgen K. Singer
Beam Technologies, Inc.
Ithaca, NY
jurgen@acm.org

MS12
HPF/MPI: A Programming System Supporting Task and Data Parallelism

We describe the design, implementation, and application of a programming system that supports a combined task and data parallel approach to programming. This system provides a High Performance Fortran (HPF) binding for the Message Passing Interface (MPI), hence allowing MPI to be used as a coordination framework for HPF programs. Applications that can benefit from this mixed model include pipelines and multidisciplinary applications. The HPF/MPI implementation incorporates techniques for negotiating data distribution information and computing efficient communication schedules for intertask data transfers. We present performance results that demonstrate the utility of the techniques in a range of applications.

Ian Foster, David Kohr, Rakesh Krishnaiyer
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439

MS12
The C* Development Environment for Scientific Programming

C* is a data-parallel extension of the C programming language, in which parallelism is expressed through the simultaneous application of a single operation to a data set. In this paper we develop a C* program to solve a simple problem in scientific computing, making use of the C* Development Environment (CSDE), a Motif-based graphical user interface developed at Oregon State University. The C* Development Environment supports the editing, compilation, debugging, and performance tuning of C* programs.

Michael J. Quinn
Oregon State University
Corvallis, OR

Philip J. Hatcher
University of New Hampshire
Durham, NH

MS12
Shasta: a System for Supporting Fine-Grain Shared Memory across Clusters

Shasta is a system that supports shared memory in software on a cluster of computers. Two interesting aspects of Shasta are that shared data can be kept coherent at a fine granularity, and the coherence granularity can vary across different data structures in an application. In this paper, we describe the design and programming interface of Shasta and report parallel performance results on our cluster, including the effect of varying the coherence granularity of individual data structures.

Daniel J. Scales and Kourosh Gharachorloo
Digital Equipment Corporation
Western Research Laboratory
250 University Ave.
Palo Alto, CA 94301.

MS12
ZPL: Scientific Computing with Speed and Convenience

The convenience of writing parallel programs in Fortran has motivated many Fortran compatible approaches to parallel programming: Fortran parallelization, directives such as HPF, Fortran plus message passing libraries. Using two simple computations as the basis for comparison, these Fortran approaches are compared with the array language ZPL. The results challenge the assumption that such approaches will suffice. The ZPL programs are both simple and fast.

Lawrence Snyder
University of Washington, Seattle, WA

MS13
ScaLAPACK: A Linear Algebra Library for Message-Passing Computers

This talk will describe the ScaLAPACK software library that extends the LAPACK library to run scalably on MIMD, distributed memory, concurrent computers as well as heterogeneous clusters of processors. For such machines the memory hierarchy includes the off-processor memory of other processors, in addition to the hierarchy of registers, cache, and local memory on each processor. Like LAPACK, the ScaLAPACK routines are based on block-partitioned algorithms in order to minimize the frequency of data movement between different levels of the memory hierarchy, providing for a latency tolerant approach. In ScaLAPACK, matrices are distributed with a two-dimensional block-cyclic data distribution which is well-suited for use in block-partitioned algorithms.

Jack Dongarra
University of Tennessee, Knoxville and Oak Ridge National Laboratory

L. S. Blackford, University of Tennessee, Knoxville;
J. Choi, Soongsil University, Korea;
A. Cleary, University of Tennessee, Knoxville;
E. D'Azevedo, Oak Ridge National Laboratory;
J. Demmel and I. Dhillon, University of California, Berkeley;
S. Hammarling, The Numerical Algorithms Group, Ltd.;
G. Henry, Intel
A. Petitet, University of Tennessee, Knoxville;
K. Stanley, University of California, Berkeley;
D. Walker, University of Wales, Cardiff;
R. C. Whaley, University of Tennessee-Knoxville

MS13
The Performance of a New Algorithm in Eigensystem Problems For Computational Chemistry

We present a new parallel algorithm for solving the real symmetric tridiagonal eigenproblem and describe its performance on the IBM SP, SGI-Cray T3D and SGI PowerChallenge. Parallel performance and and accuracy results of the eigensolver on different types of electronics structures calculations will be presented. In addition, performance statistics of the eigensolver relative to the total chemistry application will be presented. For example, for an n = 996 Moller-Plesset second order perturbation theory calculation we obtain a speedup of more than 3 over the existing methods for solving the dense symmetric eigensystem problem in the parallel eigensolver PeIGS using 100 processors of the IBM SP-2. This example has the characteristic that

it has a degenerate eigenvalue cluster of size 939 using LAPACK DSTEIN's definition. Other examples from density functional theory and self-consistent field theory will also be presented. The new algorithm is still under research and further improvements are expected.

George Fann
Pacific Northwest National Laboratory
Inderjit Dhillon and Beresford Parlett
University of California, Berkeley

MS13
PLAPACK: Parallel Linear Algebra Package

The PLAPACK project represents an effort to provide an infrastructure for implementing application friendly high performance linear algebra algorithms. The package uses a more application-centric data distribution, which we call Physically Based Matrix Distribution, as well as an object based (MPI-like) style of programming. It is this style of programming that allows for highly compact codes, written in C but usable from FORTRAN, that more closely reflect the underlying blocked algorithms, while retaining high performance.

Robert van de Geijn
(with Philip Alpatov, Greg Baker, Carter Edwards,
John Gunnels, Greg Morrow, James Overfelt, The University of Texas at Austin

MS13
Parallel SBR: a PLAPACK Based PRISM Kernel

Reducing a full dense symmetric matrix to tridiagonal form is one of the key steps in computing eigenvalues and eigenvectors of a symmetric matrix. In contrast to the traditional tridiagonalization, a multi-step reduction called Successive Band Reduction (SBR) was introduced and served as a part of computation kernel in the PRISM eigensolver. We present our experiences in implementing SBR based on PLAPACK infrastructure. We give a comparison of the programming styles between using PLAPACK and traditional parallel linear algebra libraries. We show that the simplicity of the resulting code make it very easy to document, maintain, and extend.

Yuan-Jye J. Wu and Christian H. Bischof
Argonne National Laboratory

MS14
Parallel FFTs and FMMs in a Global Shallow Water Model

The design and parallelization strategy of an algorithm for a high resolution numerical weather prediction model in spherical coordinates is described. The algorithm replaces the spherical harmonics transform with a combination of Fast Fourier Transform and Fast Multipole Method, while retaining the uniform representation qualities of the spherical harmonics. The presentation focuses on the data dependence analysis for vectorization & parallelization of the code.

Ruediger Jakob-Chien
University of Colorado
P.O. Box 173364
Denver, CO 80217
phone: (303) 556-4016
fax: (303) 556-8369
email:rudy@cse.cudenver.edu

MS14
Orthogonal Polynomial Transforms

Fast orthogonal polynomial transforms are efficient algorithms for expanding functions in orthogonal polynomial sequences. They occur throughout the natural sciences, especially in those disciplines, e.g., earth sciences, which involve calculations in spherical coordinates. The huge amount of data involved (and time constraints for predicitive applications) mandates efficient parallelizable algorithms. We present an approach based on polynomial arithmetic, to the Driscoll Healy algorithm for orthogonal polynomial transforms, and discuss aspects of its parallel implementation.

David Maslen
Utrecht University
Mathematics Department
Budapestlaan 6
3584 CD, Utrecht
NETHERLANDS
Telephone: 31-30-253-4557
Fax: 31-30-251-8394

MS14
Applications of Fast Orthogonal Polynomial Transforms

Various tasks in signal processing and data analysis can be characterized as orthogonal polynomial transforms. In this talk we outline several such, ranging from numerical climate modeling to the analysis of data on distance transitive graphs.

Dan Rockmore
Department of Mathematics
Dartmouth College
Address: Hanover NH 03755
Telephone: (603) 646-3260

MS14
Hierarchical Load balancing for Parallel Fast Legendre Transforms

We present a parallel Fast Legendre Transform (FLT) based on the Driscol–Healy algorithm with computation complexity $O(N \log^2 N)$. The parallel FLT is load–balanced in a hierarchical fashion. We use a load–balanced FFT to deduce a load–balanced parallel fast cosine transform, which in turn serves as a building block for the Legendre transform engine, from which the parallel FLT is constructed. We demonstrate how the arithmetic, memory and communication complexities of the parallel FLT are hierarchically derived via the complexity of its modular blocks.

Nadia Shalaby
Harvard University
Cambridge, MA 02138-2901

S. Lennart Johnsson
University of Houston and Harvard University

MS15
Java as a High Performance and Parallel Language

for Scientific Computation

We discuss the possibility that Java – rather than variants of Fortran or C++ – will be the preferred science and engineering language in the future. After surveying the technical and societal reasons suggesting this trend, we describe approaches to achieving high performance in both sequential and parallel Java. The discussion will be placed in the context of a general use of the Web for coarse grain software integration of large scale applications but our main thrust will be use of Java in data parallel mode on tightly coupled computers.

Geoffrey Fox
NPAC, Syracuse University
111 College Place , Syracuse NY 13244-4100

MS15
Programming with the HPC++ Parallel Standard Template Library

We present an overview of the HPC++ Parallel Standard Template Library (PSTL), a parallel version of the C++ Standard Template Library (STL). The PSTL is part of HPC++, a C++ library and language extension framework being developed by the HPC++ consortium as a standard model for portable parallel programming in C++. The PSTL includes distributed versions of the seven STL containers (vector, list, deque, set, map, multiset, multimap), as well as parallel versions of the STL algorithms. A key component of the PSTL is the parallel iterator, which provides global access to all elements in the distributed containers and faciliates generic parallel programming.

Elizabeth Johnson
Indiana University, Bloomington, IN

Dennis Gannon
Indiana University
Bloomington, IN

MS15
Parallel Object-Oriented Methods and Applications

We present a templated, object-oriented FrameWork for parallel application development. Design and evolution of the FrameWork will be discussed. Techniques for the encaspulation of parallelism will be compared and contrasted. Specific focus will be placed on template based polymorphism, code acceleration with expression templates, and utilization of the STL container/iterator/algorithm paradigm. Portable parallelism and application component interfaces will be discussed in the context of current Los Alamos applications.

John V. W. Reynders
Advanced Computing Laboratory - Los Alamos, NM

John V.W. Reynders
Los Alamos National Laboratory
Los Alamos, NM

MS16
Message Passing, Global Addresses, and Cache Coherence: A Perspective on Coordinating Parallel Computation

The message passing programming model (embodied in NX, MPL, MPI, etc.) has long be the *de facto* standard for portable parallel computation. However recent product scalable parallel systems from HP/Convex and Silicon Graphics support cache coherent distributed shared memories with the promise of a simpler programming model. We will describe recent performance trends in systems with explicit and implicit data movement (particularly comparing cluster-based High Performance Virtual Machines and DSM's), and their implications for portable high performance scientific programming.

Andrew A. Chien
Department of Computer Science
University of Illinois, Urbana, IL

MS16
Message-Passing: Evolution and Convergence

In the high-performance arena, end-users and independent software vendors are demanding message-passing, MPI in particular, as a parallel programming model. The high cost of developing parallel software promotes the use of this universal, scalable, and standard model, resulting in its long-term viability. In its evolution, MPI is embracing threads and aspects of shared-memory programming, providing parallel I/O, and a degree of inter-operability between vendor implementations, without sacrificing scalability. This will place message-passing at the point of convergence, bridging the gap between the various parallel programming models, and presenting a solid and flexible layer on which future models may be built.

Raja Daoud
Hewlett-Packard Company, Convex Division
Richardson, TX

MS16
Why HPF Should be the Panacea

We discuss the trade-offs between message passing and data parallel programming models from both in principle and experience points of view. We discuss the class of applications which can use HPF effectively and current successes with existing compilers. We also see how the corresponding HPC++ and HPJava address some of the difficulties in deploying HPF. We expect explicit message passing to continue as dominant parallel computing paradigm

Tom Haupt
Geoffrey C. Fox
Don Leskiw
NPAC, Syracuse University
111 College Place
Syracuse NY 13244-4100

MS16
Role of Message Passing in Performance Oriented Parallel Programming

With the availability of scalable shared-address space parallel computers it might appear that the message-passing style of parallel programming is no longer useful. A shared-address space programming style (e.g., thread based) can be potentially much simpler to work with, as it relieves the programmer of the burden of data partitioning/placement and of having to perform explicit communication. However, in doing so, it also hides many of the performance costs from the user. These costs arise from communication hot-spots, waiting on mutexes, and coherence protocols; all

of which are difficult to model. On the other hand, message passing makes communication costs explicit to the programmer. It also allows efficient implementation of communication operations in which multiple processors participate; thus alleviating hot-spots. Message passing also facilitates explicit data partitioning for locality. This potentially enhances the performance significantly. Whereas shared address space programming is suited for ease of programming, on conventional systems, messape passing potentially offers significantly higher performance.

Vipin Kumar
George Karypis
Ananth Y. Grama
University of Minnesota, Minneapolis, MN 55455

MS17
An Adaptive and Parallel Framework for Partial Differential Equations

We describe a framework for adaptive and parallel computation that is capable of (i) generating three-dimensional unstructured meshes, (ii) automatically refining and coarsening these meshes, (iii) partitioning the computation into subdomains that may be processed in parallel, and (iv) maintaining a balanced parallel computation. Mesh generation is supported by a hierarchical database that is connected to geometrical modeling procedures and to a parallel library having capabilities for processor scheduling and reassignment.

Joseph E. Flaherty
Department of Computer Science
and
Scientific Computation Research Center
Rensselaer Polytechnic Institute
Troy, NY 12180
USA

MS17
Dynamic Load-Balancing for Adaptive PDE Solvers with Hierarchical Meshes

Many adaptive techniques for the solution of both steady and time-dependent PDEs rely on a hierarchy of meshes: starting with a coarse global mesh and refining this by different amounts in different regions. Since this refinement (or de-refinement) is part of the solution process it is necessary to dynamically alter the way in which the mesh is partitioned if a parallel implementation of such an algorithm is to remain load-balanced. We describe a technique for achieving this, based upon consideration only of the coarse mesh.

Nasir Touheed and Peter K. Jimack
School of Computer Studies
University of Leeds
Leeds, LS2 9JT
UK

MS17
3D Parallel Mesh Adaptivity: Data-Structures and Algorithms

It is clear that parallel adaptive meshing is required for efficient solution of transient CFD problems on distributed memory computers. We will discuss issues arising in the parallelization of a general-purpose, unstructured, tetrahedral adaptivity code. In particular, we will consider data-structure issues such as communications links and the parallelization of complex hierarchical data-structures. We will also discuss algorithmic aspects of the code such as the parallelization of a depth limited recursive search.

Paul M. Selwood, Martin Berzins and Peter M. Dew
School of Computer Studies
University of Leeds
Leeds, LS2 9JT
UK

MS17
Dynamic Load-Balancing for Parallel Adaptive Unstructured Meshes

A parallel method for dynamic partitioning of unstructured meshes is described. It employs a new iterative optimisation technique which both balances the workload and minimises the interprocessor communications overhead. Experiments on a series of adaptively refined meshes indicate that the algorithm provides partitions of an equivalent or higher quality to static partitioners and much more quickly. Perhaps more importantly, the algorithm results in only a small fraction of the amount of data migration in comparison.

Chris Walshaw, Mark Cross and Martin Everett
Centre for Numerical Modelling and Process Analysis
University of Greenwich
Wellington St, Woolwich
London, SE18 6PF, UK
Email: C.Walshaw@gre.ac.uk

MS18
Bond, an Environment for Heterogeneous Parallel and Distributed Computing for Structural Biology

Structural Biology requires vast amounts of storage and computing cycles that can only be provided by heterogeneous parallel and distributed systems. X-ray diffraction and electron microscopy are the methods of choice to gather experimental information about the structure of large macromolecules like viruses. Experiments using intense X-ray beams and CCD detectors often produce thousands of raw data file scattered over many sites. The raw data is then subject to indexing, structure factor refinement, and analysis for solving the phase problems. Electron microscopy, can also produce large volumes of raw data and requires automatic particle identification on micrographs as well as 3D image reconstruction. Last but not least, model building involves access to protein data bases and data bases of previously solved structures. Bond is an environment for parallel and distributed computing targeted at structural biology. It uses a network of intelligent agents to carry out a variety of tasks grouped in two basic categories. Level 1 tasks are quite general and can be used by any application using a heterogeneous system. They cover functions related to the computing infrastructure namely data and program migration, task scheduling and task mapping to various platforms, exporting and importing different objects (data and programs), data mining and the collaborative support. Level 2 tasks to be developed at a later stage of the project cover the domain specific knowledge processing. Built on top of Level 1, Level 2 tasks provide support for model building and learning.

Dan C. Marinescu.
Computer Sciences Department

Purdue University
West Lafayette, In 47907, USA

MS18
A High-Performance Problem-Solving Environment for Optimization and Control of Chemical and Biological Processes

Our project is the development of a high-performance problem-solving environment (PSE) for the optimization and control of chemical and biological processes. The initial focus is on tissue-engineering applications such as control of wound contraction and engineering design of a bioartificial artery. Development of the PSE presents significant computational and computer science challenges. The problems involved are described by time-dependent partial differential equations (PDEs) in two or three spatial dimensions. These are discretized with adaptive finite element methods, giving a large-scale system of differential-algebraic equations (DAEs) in time. The computational problem is to optimize a given functional subject to a system of PDEs and other equality and inequality constraints. An effective interactive environment requires the use of massively parallel computers and state-of-the-art methods and software for large-scale optimization, sensitivity analysis for DAEs, adaptive finite-element methods for PDEs, parallel computing and computing environments. We will describe the current status of our project, emphasizing the rewards and challenges of working in an interdisciplinary team, and show some preliminary results.

Joseph E. Flaherty
Rensselaer Polytechnic Institute

Philip Gill
University of California San Diego

Ephraim P. Glinert
Rensselaer Polytechnic Institute

Linda R. Petzold
University of Minnesota

J. Ben Rosen
University of Minnesota

Boleslaw K. Szymanski
Rensselaer Polytechnic Institute

Robert T. Tranquillo
University of Minnesota

MS18
PSEware: A Toolkit for Building Problem-Solving Environments

Scientific and engineering problems are increasingly attacked by geographically dispersed, multi-disciplinary teams utilizing heterogeneous computing systems ranging from desktop systems to the most powerful parallel system accessible to the team. These teams, for example, the Grand Challenge Application Groups, also now frequently construct their own domain specific programming environment to facilitate their investigations. The PSEware Project is a research project exploring the provision of a toolkit for such users to construct their own Problem Solving Environments (PSEs) for parallel object-oriented programming. The PSEware (pronounced SEAware) project is an application driven effort addressing the following major issues: a) PSEs that support the transformation of symbolic problem definitions to parallel object-oriented programs that can be executed efficiently on a variety of sequential and parallel architectures; b) object-oriented libraries of parallel program templates or archetypes that can be refined to obtain specific applications by using ideas such as inheritance; c) user-interface archetypes for scientific and engineering PSEs that can be refined to construct a PSE for specific problem domain; and d) technologies for collaboration and ubiquitous distributed computing focused on the Internet, the Web, and Java, with the goal of applying these technologies to distributed collaborative PSEs.

James C. T. Pool
California Institute of Technology, Pasadena, CA

MS18
A MultiAgent Environment for MPSEs

We present a paradigm for simulating complex systems which may involve multiple physical phenomena and complicated geometries. The computational structure is that of cooperating agents and much of the proposed methodology is widely applicable to scientific computing, but the focus here is on phenomena modeled by partial differential equations (PDE). The computational process is to subdivide the physical object as components of simple geometric shapes modeled by a single problem solving environment (PSE). PSEs are viewed as agents which solve the PDE on each component independently. The interfaces between the components must have physical interface conditions satisfied; mediator agents use relaxation techniques for this. The architecture of an agent-based environment for building systems to implement this paradigm is described, using PSEs which are encapsulated into solver agents. This approach is naturally parallel and highly scalable; it is suitable for a wide variety of parallel and distributed high performance computing (HPC) architectures; it allows for the reuse and evolution of existing HPC software, and for a convenient abstraction of the solution process for non-expert users. *SciAgents*, a system which implements the proposed architecture, is presented and used to solve an example problem illustrative of real world, multidisciplinary problems.

Tzvetan T. Drashansky, John R. Rice, Elias N. Houstis, Sanjiva Weerawarana
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907-1398
email: {ttd,jrr,enh,saw}@cs.purdue.edu

Anupam Joshi
Department of Computer Engineering & Computer Science
University of Missouri
Columbia, MO 65211
email: joshi@csdeca.cs.missouri.edu

MS19
Parallel ARPACK: Portable Software for the Solution of Large Scale Eigenvalue Problems on Distributed Memory Architectures

ARPACK is a Fortran 77 implementation of the Implicitly Restarted Arnoldi Method used for solving large sparse eigenvalue problems. A parallel implementation of ARPACK (ARPACK-P) will be described which is tar-

geted toward distributed memory platforms. The existence of ARPACK-P is an illustrative example of how recent advances in parallel computing and new techniques for solving eigenvalue problems can be combined to produce efficient and reliable parallel software. A further extension of ARPACK-P, a parallel out-of-core eigen-solver, will also be presented. Here we will show how this solver can be used to determine a significant percentage (up to 50large matrices.

Kristyn J. Maschhoff
Rice University,
Department of Computational and Applied Mathematics

MS19
Parallel Computation of Spectral Portrait by Davidson's Method

The eigenanalysis of matrices or operators based only on the knowledge of the spectrum may be misleading in the non-normal case. Instead of the spectrum, one may fully characterize the spectral behavior of a non-normal matrix by analyzing its spectral portrait, i.e. the set of its resolvent norm. In this talk, we propose a parallel version of the generalized Davidson method for analyzing and plotting the spectral portrait of large non-normal matrices. We report the performance results obtained on the machine Paragon.

Bernard Philippe
Miloud Sadkane,
IRISA, Campus de Beaulieu
FRANCE

MS19
Lanczos and Conjugate-Gradient Minimization Based Eigenvalue Calculations on the Cray T3E

In this talk we will report about the implementation of a block shifted Lanczos algorithm on the Cray T3E. We will discuss various issues in implementing the block Lanczos on parallel machines, in particular the trade-offs which can be made between the various levels of parallelism in the algorithm (task parallelism for different shift intervals, data parallelism in the matrix operations). We will also present an analysis of the performance of the T3E, a next generation parallel supercomputer based on the 600 Mflop/s Alpha EV-5 microprocessor and the torus interconnect from the T3D. Finally we will compare the Lanczos approach with a conjugate gradient minimization approach for eigenvalue calculation, and present actual applications results. We are applying these eigenvalue techniques to find the smallest eigenpairs of large matrices arising from a Fourier space discretization of a Schroedinger-type equation in the context of Density Functional Theory calculations in Solid State Physics. Our results demonstrate the potential for scientific discovery in material science, using scalable scientific computing technology. Our efficient algorithms, combined with the power of the T3E, allow us to determine structural and electronic properties of complex new materials and defects in solids, and permit the simulation of amorphous materials and liquids.

Bernd Pfrommer, Steven G. Louie
Dept. of Physics, University of California, Berkeley
Horst D. Simon
NERSC Division, Lawrence Berkeley National Laboratory

MS19
Parallel Solution of Eigenvalue Problems in Electronic Structure Calculations

Predicting the electronic structure of complex systems is an outstanding problem of condensed matter physics, and a source of computationally challenging eigenvalue problems. These eigenvalue problems are very large and sparse, and hundreds, or even thousands of eigenvalues are required. To reduce the matrix size, we consider irregular domains, and data structures similar to the ones for unstructured meshes. The Davidson eigenvalue solver is adopted to allow for the benefits of preconditioning. We present a parallel implementation of the electronic structure code that runs both on clusters of workstations and on massively parallel processors. Several issues are addressed such as synchronization, block Davidson algorithm, subdomain partitioning, and the efficient communication of the boundary information needed for matrix-vector multiplication and preconditioning operations. On the Cray T3D, a combination of PVM and MPI communication libraries is used to achieve optimal performance. The good scalability of the resulting code enables efficient use of powerful parallel platforms and allows to solve much bigger problems than has been possible before.

Andreas Stathopoulos,
Yousef Saad
Department of Computer Science, University of Minnesota
James R. Chelikowsky
Department of Chemical Engineering, University of Minnesota

MS20
Automatic Parallelization of Sparse Matrix Applications

Over the past ten years, there has been a lot of progress in the automatic parallelization of dense matrix computations. These advances are being incorporated into a new generation of commercial compilers for languages like HPF. However, the majority of scientific applications, such as finite element methods, require sparse matrices, which cannot be handled effectively by this technology. The Bernoulli project is addressing this problem by synthesizing sparse matrix codes from a high level algorithmic specification of the computation together with user directives for sparse data layouts. In this talk, we describe this approach and present preliminary results in synthesizing parallel programs for iterative solvers for large systems of linear equations on the IBM SP-2.

Vladimir Kotlyar
Department of Computer Science,
Cornell University
Ithaca, NY

Keshav Pingali
Department of Computer Science,
Cornell University, Ithaca, NY

Paul Stodghill
Department of Computer Science,
Cornell University
Ithaca, NY

MS20
A Kronecker Compiler for Fast Transform Algorithms

We present a source-to-source compiler that processes matrix formulae in the form of Kronecker product factorizations into High Performance Fortran. The Kronecker product notation allows for simple expressions of algorithms such as Walsh-Hadamard, Haar, Slant, Hartley, and FFTs as well as wavelet transforms. The compiler is based on a set of term rewriting rules that translate high level matrix descriptions into parallel and sequential loops and assignment statements. We also provide back-end translators for FORTRAN, FORTRAN-90, C and MATLAB.

Nikos P. Pitsianis
Department of Computer Science
University of Houston
Houston TX 77204-3475

MS20
Efficient Data Structures for Sparse Iterative Methods

Obtaining good processor performance on high-performance architectures for sparse iterative methods is difficult. Indirect addressing, lack of data reuse, and the small size of the matrices that can take advantage of the BLAS routines contribute to the inefficiencies of codes for these methods. In this talk, we discuss the graph reductions and corresponding layered data structures used to achieve efficiency in the software package BlockSolve. This software is designed for high-performance, parallel computers and uses preconditioners based on incomplete matrix factorizations. We present computational results from current parallel machines and discuss the effectiveness of this approach for problems from a number of scientific application areas.

Mark T. Jones
Department of Computer Science
University of Tennessee
Knoxville, TN

Paul E. Plassmann
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439

MS20
Optimization of a Class of Multi-Dimensional Integrals on Parallel Machines

Multi-dimensional summations involving products of arrays arise in certain kinds of computational physics calculations that model electronic structure. Besides the issue of optimal distribution of the arrays among the processors, there is also scope for reordering of the operations using algebraic properties (commutativity, associativity, distributive law) to significantly reduce the number of operations executed. A framework for optimization of computational cost, communication cost and temporary storage requirements has been developed, that can be used to synthesize efficient code.

C. Lam, P. Sadayappan and R. Wenger
Ohio State University
2015 Neil Avenue
Columbus, OH 43210

PS
Parallel Domain Decomposition Methods in Fluid Models with Monte Carlo Transport

To examine the domain decomposition of a coupled Monte Carlo - finite element calculation, it is important to use a domain decomposition that is suitable for the individual models. We have developed a code that simulates a Monte Carlo calculation running on a massively parallel processor. This code is used to examine the load balancing behavior of three domain decomposition strategies for a Monte Carlo calculation. The results are presented.

Henry J. Alme and Garry H. Rodrigue
Department of Applied Science
University of California@Davis
Livermore, CA, 94550
E-mail: almehj@kristen.llnl.gov

George B. Zimmerman
X-Division
Lawrence Livermore National Laboratory
Livermore, CA, 94550

PS
State Space Modeling and High Performance Computing of Multivariate Time Series

State space realizations have been increasingly studied and applied to multivariate time series modeling. In this work we present a methodology for high performance computation for an algorithm developed by Masanao Aoki (*State Space Modeling of Time Series - Springer-Verlag, 1990*) for state space modeling of time series. The proposed algorithm implementation allows the obtention of the triple (**A**, **B** and **C**) of system's matrices for the state space model for multivariable sytems. The computational implementation of such algorithm is done via the utilization of parallel and distributed processing; with the objective of real time state space modeling of time series.

Celso Pascoli Bottura, Gilmar Barreto, Maurício José Bordon
Faculdade de Engenharia Elétrica
Universidade Estadual de Campinas, Brasil

José Tarcisio Costa Filho
Departamento de Engenharia Elétrica
Universidade Federal do Maranhão, Brasil

PS
A Parallel Preconditioned Bi-Conjugate Gradient Algorithm for Two-Dimensional Elliptic and Parabolic Equations Using Hermite Collocation

A fast and parallelizable method to solve sparse matrix equations arising from the Hermite collocation discretization of elliptic and parabolic partial differential equations is obtained using a bi-conjugate gradient algorithm. A Red-Black Gauss-Seidel matrix preconditioner provides a structure which makes the algorithm amenable to parallel processing. Increased efficiency is achieved through optimally locating the "collocation points". Results concerning eigenvalues and the efficacy of the method when applied to model problems can be demonstrated.

Stephen H. Brill
Department of Mathematics and Statistics
University of Vermont, Burlington, VT 05405
E-mail: brill@uvm-gen.emba.uvm.edu

George F. Pinder
Departments of Civil and Environmental Engineering
and of Mathematics and Statistics
University of Vermont, Burlington, VT 05405

PS
OVERTURE: An Object-Oriented Software System for Solving Partial Differential Equations in Serial and Parallel Environments

The OVERTURE Framework is an object-oriented environment for solving PDEs. It is a collection of C++ libraries that enables the use of finite difference and finite volume methods at a level that hides the details of the associated data structures, as well as the details of the parallel implementation. It is based on the A++/P++ array class library, and uses curvilinear grids and the composite overlapping grid method to represent problems with complex moving geometry.

David L. Brown, Geoffrey Chesshire, William Henshaw, and Daniel Quinlan
Los Alamos National Laboratory
Los Alamos, NM 87545
Email: dlb@lanl.gov, geoff@c3serve.c3.lanl.gov, henshaw@c3serve.c3.lanl.gov, dquinlan@c3serve.c3.lanl.gov

PS
Preconditioned Conjugate Gradient Methods for Manufacturing Systems

This paper studies the application of Preconditioned Conjugate Gradient method in solving the steady state probability distribution of failure-prone manufacturing systems under hedging policies. Preconditioner is constructed by taking circulant approximation of the generator matrix. The preconditioned system is proved to has singular values clustered around one when the number of inventory levels tends to infinity. The preconditioner can be inverted easily in parallel. Numerical examples are given to demonstrate the fast convergence rate of our method.

Wai Ki Ching
Department of Applied Mathematics,
Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong,
and
Department of Systems Engineering and Engineering Management,
Chinese University of Hong Kong,
Shatin, N.T., Hong Kong
Email: wkching@se.cuhk.hk

PS
Runtime Support System for Parallel Iterative PDE Computations

For parallel iterative PDE computations, one of the main difficulties in attaining performances close to peak performance of the target multicomputer is the memory latency (i.e., network latency and communication overhead). In this paper we show that the communication overhead can be eliminated or masked through algorithm and code restructuring. The underline idea of our approach is to integrate computation and communication at the instruction level.

Nikos P. Chrisochoides
Advanced Computing Research Institute
Computer Science Department and Theory Center
Cornell University, Ithaca, NY 14853
nikosc@cs.cornell.edu

PS
Parallel Implementation of a Data-Transpose Technique for the Solution of Poisson's Equation in Cylindrical Coordinates

We present a parallel finite-difference algorithm for the solution of the 3D cylindrical Poisson equation. The algorithm is based on a data-transpose technique, in which all computations are performed independently on each node, and all communications are restricted to global 3D data-transposition between nodes. The data-transpose technique aids us in implementing two sequential algorithms for the solution of the cylindrical Poisson equation. The first algorithm is based on the alternating direction implicit method and the second is based on Fourier Analysis. Execution times are measured and compared on a 8K-node massively parallel MasPar MP-1 and on a single processor Cray C90 vector machine.

Howard S. Cohl and Joel E. Tohline
Department of Physics and Astronomy
Louisiana State University, Baton Rouge, LA 70803
E-mail: hcohl@rouge.phys.lsu.edu

Xian-He Sun
Department of Computer Science
Louisiana State University, Baton Rouge, LA 70803

PS
Parallel Processing to Solve EDP Elliptic

This paper proposes a different approach to paralelize a new iterative projection method, proposed by U. García Palomares and X. Contreras, that finds an approximate solution to an elliptic partial differential equation. The method generates as a solution a function which lies in the space spanned by a given finite basis. The strategy used in this alternative is to assign more work load to the processors. In that sense, the method is applied in two times. First time: Each processor uses the method to find a direction over the subspace generated by a given orthogonal subset of the basis. After this, the result is sent to the master processor. Second time: The master processor applies the method to find a new direction for the next iterate. As opposed to a previous work of X. Contreras, M. Curiel y A. Di Serio which each processor calculates the step along a given direction of the established orthogonal set and the master processor uses this data and updates the new iterate.

Xiomara Contreras, Mariela Curiel and Angela Di Serio
Simón Bolívar University , Venezuela
Email: xiomara@ldc.usb.ve

PS
Domain Decomposition Method for Solving Three-Dimensional Parabolic Differential Equations Arising in Thermal Analysis in X-Ray Lithography

X-ray lithography is an important technique in microfabrication used to obtain microstructures and devices with high

aspect ratios. The process consists of x-ray irradiation of a photoresist, such as polymethylmethacrylate (PMMA), deposited on a silicon substrate. Prediction of the temperature distribution in the resist and substrate is very useful in studying the effect of high flux x-ray exposure on distortions in the resist as well as on bonding between resist and substrate. In this study, we develop a three-dimensional numerical algorithm with high inherent parallelism to solve the parabolic differential equations on multilayers based on the domain decomposition method for thermal analysis of x-ray irradiated photoresists. The alternating direction implicit (ADI) method is employed in the algorithm. To overcome the problem with unknown values at the interface, the generalized 'Divide and Conquer' procedure for solving tridiagonal linear systems is applied. Thus, the computational procedure is simple and efficient. Numerical example is computed by using a NCSA supercomputer(Power Challenge).

This research is supported by a Louisiana Educational Quality Support Fund grant, Contract No: 72-6000792, and a NCSA grant No:DMS960004N.

Weizhong Dai, Raja Nassar and Danqing Jiang
Mathematics and Statistics
College of Engineering and Sciences
Louisiana Tech University
Ruston, LA 71272
e-mail: dai@math.latech.edu

PS
Discrete Evolution of 2D Turbulence with Injection on Cray T3D

We present in this paper the simulation of an incompressible turbulent fluid in a channel with blowing through the floor plate. The fluid velocity is used to define a conservative discrete vector (u_x, u_y). The system evolves from the initial state to converge to an asymptotic behavior according to the Navier-Stoke equations and using a k-ϵ model of turbulence. Furthermore, we introduce a new method to take into account the blowing. The computed result is compared to experimental results and solutions from other non-parallel codes to validate the discrete model and the parallel implementation.

Françoise Bataille
Centre de Thermique de Lyon
INSA Lyon - France Bataille@CETHIL.INSA-Lyon.Fr

Marc Daumas
Laboratoire de l'Informatique du Parallélisme
ENS Lyon - France Marc.Daumas@ens-lyon.fr

PS
Construction of a Parallel GMRES(k) Solver

Given a large, sparse, non-symmetric system $Ax = b$, we outline an iterative solver which uses the well-known GMRES(k) method (Saad and Schultz, 1986; Barrett et al., 1994). We discuss our implementation using MPI and analyze the parallel characteristics of the algorithm. We also consider the parallel properties of an ILU preconditioned version, and eigenvalue translations to improve the convergence of the GMRES(k) iterations (Kharchenko and Yeremin, 1992).

1. R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, 1994.
2. S. Karchenko and A. Yeremin, Eigenvalue Translation Based Preconditioners for the GMRES(k) Method. Elegant Mathematics, Inc. EM-RR 2/92, 1992.
3. Y. Saad and M. Schultz, GMRES: A Generalized Minimal Residual Algorithm for solving Non-Symmetric Linear Systems. SIAM J. Sci. Stat. Comput., 7, pp. 856-869

Elise de Doncker and Ajay Gupta
Western Michigan University, Kalamazoo, MI 49008
Email: elise@cs.wmich.edu

PS
Step by Step Transformation of a FORTRAN 90 Program to HPF, using HPF-Builder

HPF-Builder graphical environment goal is to free the HPF programmers of all the syntactic constraints due to the data mapping. All the data distribution and alignment are insured in an interactive and visual way. HPF TEMPLATES and PROCESSORS become the visual support for alignments of arrays and distributions on grids of processors. HPF-Builder automatically generates the corresponding HPF directives and inserts them in the FORTRAN 90 source code. It results in a HPF code.

Jean-Luc Dekeyser and Christian Lefebvre
Laboratoire d'Informatique Fondamentale de Lille
Villeneuve d'Ascq, France
dekeyser@lifl.fr, lefebvre@lifl.fr

PS
Parallel Applications of a Schwarz Preconditioned Krylov Solver for Nuclear Power Plant Simulation

A nested FGMRES linear solver was developed and implemented in the EPRI nuclear reactor simulation code RETRAN-03. The method was accelerated using Schwarz preconditioning and mapped onto a distributed memory parallel computer. Significant performance improvements were achieved and full physics, three dimensional modeling is now possible on engineering workstations for practical size reactor models and applications. Research Sponsoredby the Electric Power Research Institute

John Steill, Thomas J. Downar and Jen-Ying Wu
School of Nuclear Engineering
Purdue University
W. Lafatyette, IN 47907-1290
E-mail: (steill, downar)@ecn.purdue.edu

PS
PVMPI Provides Interoperability between MPI Implementations

Presently, different MPI implementations cannot interoperate with each other. In order to do distributed computing across different vendors' machines now requires that a single MPI implementation, such as MPICH, be used rather than the vendors' own optimized MPI implementations. This talk describes a software package called *PVMPI* we are developing that allows interoperability of vendors' optimized MPI versions. Our approach builds on the proven and widely ported Parallel Virtual Machine. The use of PVMPI is transparent to MPI applications and allows

intercommunication via all the MPI point-to-point calls. PVMPI allows more flexible control over MPI applications than is currently indicated by the MPI-2 forum by providing access to all the process control and resource control functions available in the PVM virtual machine.

Graham E. Fagg
University of Tennessee
Knoxville TN, 37996

Jack Dongarra
Oak Ridge National Lab and University of Tennessee
Knoxville TN, 37996

Al Geist
Oak Ridge National Lab
Knoxville TN, 37996
geist@bigblu0.EPM.ORNL.GOV

PS

Comparing the Solution of the Symmetric Eigenvalue Problem with ScaLAPACK and Jacobi methods

The routine PDSYEVX of ScaLAPACK and a blocked Jacobi method are compared. PDSYEVX works by reducing the initial matrix to tridiagonal form, thus having a cost of $\frac{4n^3}{3p} + O\left(n^2\right)$ flops when computing eigenvalues. The blocked Jacobi method has a cost of $\frac{4n^3}{p} + O\left(n^2\right)$ flops per sweep and needs approximatelly $\log n$ sweeps. But the Jacobi method has a higher inherent parallelism, and this is why it could be competitive with the routine PDSYEVX with a large number of processors.

Domingo Giménez
University of Murcia
Departamento de Informatica y Sistemas
Aptdo 4021, 30001 Murcia, Spain
Email: domingo@dif.um.es

PS

Accelerating Iterative Methods by Inducing Sparsity in the Residual Vector

Iterative methods have been developed which exploit the sparsity in the coefficient matrix of large systems of linear equations to reduce the cost per iteration. However, sparsity in the residual vector is usually not considered or preserved. Here we describe how sparsity may be induced and maintained in the residual, how this affects the cost per iteration and convergence rate for iterative methods, and how to exploit it to yield greater parallelism in the computation.

Todd Goehring and Daniel Boley
Department of Computer Science
University of Minnesota
200 Union Street SE, rm 4-192
Minneaspolis, MN 55455
goehring@cs.umn.edu, boley@cs.umn.edu

PS

A Practical Parallel Algorithm for Delaunay Triangulation

The generation of quality meshes is of fundamental importance in certain areas of scientific computation. These meshes can be quite large and their creation can require significant computational resources. An important tool in several mesh generation schemes is the construction of Delaunay triangulations. We present a new parallel algorithm for the construction of Delaunay triangulations on distributed memory computers. This algorithm allows for the efficient insertion of vertices into an existing Delaunay triangulation. We give experimental results for an mpi-based implementation of this algorithm.

Jeff Griffith and Mark T. Jones
Department of Computer Science
University of Tennessee
Knoxville, TN 37996-1301
jgriffit@cs.utk.edu, jones@cs.utk.edu

Paul E. Plassmann
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue, Argonne, IL 60439-4844
plassman@mcs.anl.gov

PS

A Limited-memory Parallel Incomplete LU Factorization

This paper presents a new strategy for improving the standard incomplete LU factorization for use in preconditioned Krylov linear solvers. This strategy is based on a limited fill-in approach where the nonzero structure of the factored matrix depends on both the magnitude of the nonzero elements and the initial number of nonzeros in the matrix. An efficient implementation of our limited-memory incomplete LU factorization using outer loop parallelization with events-based synchronizations on both shared and distributed shared memory computers is also presented. Our experimental results, obtained with large sparse problems from semiconductor simulation applications, show that our strategy is more robust and more efficient in terms of linear iterations and execution times than the standard method.

Ali Bouaricha and Mounir Hahad
Silvaco International, Santa Clara, CA 95054
ali@silvaco.com mounirh@silvaco.com

PS

Parallel Computation of the Minimal Crossing Number of a Graph

Determining the crossing number of a graph is an important problem with applications in areas such as circuit design and network configuration. In this paper we present the first parallel algorithm for solving this combinatorial optimization problem. This branch-and-bound algorithm, which adds and deletes crossings in an organized fashion, presents us with the opportunity to verify many conjectures which are decades old, as well as pursuing future work in efficient circuit design.

Umid Tadjiev and Frederick C. Harris, Jr.
Department of Computer Science
University of Nevada
Reno, NV 89557
tadjiev@cs.unr.edu, fredh@cs.unr.edu

PS

Dynamic Load Balancing With Geometric Locality

The fundamental computation in a wide range of simulations is to calculate interactions between near-by *objects*.

On a parallel machine, if the objects move or their computational requirements change, a dynamic load-balancing problem ensues. Examples include such diverse applications as adaptive grid calculations, molecular dynamics, contact detection in Lagrangian grids, particle-in-cell electromagnetics, smoothed particle hydrodynamics and graphics rendering. In addition to the usual considerations, a dynamic load balancing algorithm for these simulations should
(1) preserve geometric locality,
(2) evolve incrementally as the calculation proceeds, and
(3) enable efficient detection of objects which extend into other processor's domains.
These constraints eliminate most sophisticated dynamic load-balancing algorithms from consideration. However, the simple algorithm known as recursive coordinate bisectioning (RCB) has all the necessary properties. While it does not produce optimal partitionings in the graph-theoretic sense, we believe it has been under-appreciated as a robust dynamic load balancing technique for a wide range of problems. We have used RCB in several of the listed applications and present results attesting to its efficacy.

Bruce Hendrickson and Steve Plimpton
Sandia National Laboratories
Albuquerque, NM 87185-1110
Email: bahendr@cs.sandia.gov

PS
A New Parallel Solvers for Tridiagonal Equations

Based on minimal rank decoupling idea and matrix modification formula, a new algorithm is presented, designed to solve tridiagonal equations efficiently on multicomputers. The algorithm keeps the structure property of the orginal matrix with some parameters choosen,then it can get stable solution with local LU factorization without pivoting.Both computation and communication complexities of the algorithm are presented; All algorithms are designed based on PVM and tested on network of workstations.

Ying Hou, XiaoMei Li
Department of Computer Science
Changsha Institute of Technology
P.R. China, 410073

PS
Parallel Algorithms for Interval Polynomial Interpolation

Interval algorithms have been developed recently to reliably approximate unknown functions with taking data measurement errors into considerations. The algorithms involve very intensive computations to determine an appropriate polynomial to model the unknown function, and to minimize the overestimation due to interval arithmetic. Parallel algorithms are desirable. In this presentation, we will briefly review interval polynomial interpolation algorithm first. Then, we report several parallel formulations of the algorithms to significantly reduce computation time complexity. Comparisons will be presented both analytically and numerically.
This research was partially supported by NSF grants CCR-9503757, CDA-9522157; and DoD/ARO grant DAAH-0495-1-0250.

Chenyi Hu
Center for Computational Science
and Advanced Distributed Simulation
University of Houston-Downtown
Houston, Texas 77002
E-mail: hu@happy.dt.uh.edu

PS
On the Accuracy of Anderson's Fast N–body Algorithm

We present an empirical study of the accuracy–cost tradeoffs of Anderson's method. The various parameters that control the degree of approximation of the computational elements and the separateness of interacting computational elements, govern both the arithmetic complexity and the accuracy of the methods. We present models for predicting the running time and the accuracy of the potential and force evaluations for uniform particle distributions, and a scheme for choosing optimal parameters that give the best running time for a prescribed error. The scheme is numerically verified via simulations of up to one million particles on the CM–5E. Our experiment shows that for a given error requirement, using near–field containing only nearest neighbor boxes and optimal hierarchy depth which minimizes the number of arithmetic operations minimizes the total number of arithmetic operations, which often minimize the total running time for implementations with balanced efficiencies at different stages of the method.
We also investigate the choice of sphere radii in Anderson's method and their impact on the error for some small variations of the uniform particle distribution.

Y. Charlie Hu
Harvard University, Cambridge, MA
hu@das.harvard.edu

S. Lennart Johnsson
Harvard University, Cambridge, MA
University of Houston, Houston, TX
johnsson@cs.uh.edu

PS
Communication Performance Modeling and Its Effect on Parallel Computation Performance of Networked Parallel Computing Environment

This paper studies the performance of communication and parallel computation in the MPI-based networked parallel computing environment, based on Pentium PC n 100M bps fast-switch Ethernet interconnection. We use timing experiments in such environment to develop a systematic method for evaluating communication performa (including point-to-point communication, collective communication and collective computation) . Besides, by testing a typical application, we have analyzed the e of communication ability on the performance of parallel computation(such as spe efficiency).

Meng Jie and Li Sanli
Department of Computer Science & Technology
Tsingha University
Beijing, 100084, P.R. China
E-mail:meng@tirc.cs.tsinghua.edu.cn

PS
Parallel Threshold-based ILU Factorization

Factorization algorithms based on threshold incomplete LU factorization have been found to be quite effective in pre-

conditioning iterative system solvers. However, their parallel formulations have not been well understood and they have been considered to be unsuitable for distributed memory parallel computers. In this paper we present a highly parallel formulation of such factorization algorithms. Our algorithm utilizes our parallel multilevel k-way partitioning and independent set computation algorithms to effectively parallelize both the factorization as well as the solution of the resulting triangular systems, used in the application of the preconditioner. Our experiments on Cray T3D show that significant speedup can be achieved in both operations; thus, allowing threshold incomplete factorizations to be successfully used as preconditioners in parallel iterative solvers.

George Karypis and Vipin Kumar
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455
Email: karypis@cs.umn.edu

PS

An Iterative Algorithm Using Probabilistic Automata for Predicting the Performance of Parallel Computers

We present an iterative algorithm that solves the following system of linear equations: $\pi M = 0$, $\pi e = 1$, where M is the transition rate matrix of a continuous time Markov process, π is the vector of steady state probabilities of the Markov process and e is a column vector of all 1's. In models of parallel systems the matrix M can be expressed as $M = P \oplus Q$, where $\oplus$ is the generalized Kronecker sum, and P and Q are two inter-dependent automata. Such a system can be solved iteratively by assuming a solution for the automaton described by P and solving for the steady state probabilities of Q and then using this solution to solve P. This process is repeated until the solutions for P and Q are the same for two iterations. We have also developed a method to estimate the error in the above algorithm. This allows us to correct the solution obtained by our algorithm to any desired accuracy. Our algorithm converges in 3 or 4 iterations. If P is an $r \times r$ matrix and Q is $s \times s$, then M is an $rs \times rs$ matrix. The complexity of our algorithm is $O(r^3 + s^3)$ and other algorithms have a complexity of $O((rs)^3)$. The complexity of the error estimation algorithm is $O((rs)^2)$.

Raj S. Katti, and Huaan Zhang
Department of Electrical Engineering
North Dakota State University, Fargo, ND 58105
katti@plains.nodak.edu

PS

CUMULVS: Collaborative Infrastructure for Developing Distributed Simulations

The CUMULVS software environment provides remote collaboration among scientists by allowing them to dynamically attach to, view, and "steer" a running simulation. Users can interactively examine intermediate results on demand, saving effort for long-running applications gone awry. In addition, it provides fault tolerance to distributed applications via user-directed checkpointing, heterogeneous task migration and automatic restart. This talk describes CUMULVS and how this tool benefits scientists developing large distributed applications.

James Arthur Kohl, Philip M. Papadopoulos,
and G. A. Geist, II
Oak Ridge National Laboratory
P.O. Box 2008, Oak Ridge, TN 37831-6367
Email: kohl@msr.epm.ornl.gov

PS

Orthogonal Reduction of Dense Matrices to Bidiagonal Form on Computers with Distributed Memory Architectures

We present a method for the blocked reduction of dense matrices to bidiagonal form by Householder transformations. The method is based on the idea of using un-normed reflector vector. We consider a representation of Householder reflections that permits us to work directly with rows and columns of a matrix to be operated. The numerical tests show that compared with the ScaLapack bidiagonalization the algorithm has more good performance characteristics due to using more long messages.

Sergey Kuznetsov
University of Minnesota
Department of Computer Science
Minneapolis, MN 55455
kuznesov@msi.umn.edu

PS

MC++: Parallel, Portable, Monte Carlo Neutron Transport Code in C++

We have developed an implicit Monte Carlo neutron transport code in C++ using the Parallel Object-Oriented Methods and Applications (POOMA) class library. MC++ runs in parallel on and is portable to a wide variety of platforms, including MPPs, clustered SMPs, and individual workstations. It contains appropriate classes and abstractions for particle transport and parallelism. Current capabilities of MC++ are discussed, along with future plans and physics and performance results on many different platforms.

Stephen R. Lee
Los Alamos National Laboratory, Group XTM
Box 1663, MS B226, Los Alamos, NM 87545
Email: srlee@xdiv.lanl.gov

Julian C. Cummings
Los Alamos National Laboratory, Group CIC-ACL
Box 1663, MS B287, Los Alamos, NM 87545

Steven D. Nolen
Texas A&M University/Los Alamos National Laboratory, Group XTM
Box 1663, MS B226, Los Alamos, NM 87545

PS

Parallelization and Automatic Data Distribution for Nuclear Reactor Simulations

Detailed nuclear reactor simulations take many times real time on high performance workstations. Even supercomputers can not run these simulations fast enough to prevent minor malfunctions from becoming major catastrophes. These simulations must be parallelized. Parallel architectures and models will be considered for parallelization of nuclear reactor simulation. The data parallel model will be emphasized. Data parallel programming tools for multiple component applications and special parallelization

considerations will also be discussed.

Lorie M. Liebrock
Liebrock-Hicks Research
2315 County Rd., Calumet, MI 49913
Email: liebrock@mtu.edu

PS
Solution of Multiobjective MDO Using Parallel Genetic Algorithm

A multiobjective multidisciplinary optimization (MDO) of two- dimensional airfoil design is presented. In multiobjective problems, one interest is to find the set of Pareto optimal solutions. In this paper an approximation for the Pareto set is obtained by using a genetic algorithm (GA). Since GAs are naturally parallel, it is convenient to use parallel computers on the solution of this very laborious problem. The first objective function is the drag coefficient. As a constraint, it is required that the lift coefficient is above a given value. The CFD analysis solver is based on inviscid Euler equations. The discretization is done using a finite volume method and the steady state solution is obtained by an implicit pseudo-time integration with a multigrid acceleration. The second objective function is equivalent to the integral of the transverse magnetic radar cross section over a given sector. The computational electromagnetics wave field analysis solver is based on the time-harmonic two-dimensional Maxwell equations. In this case these equations can be reduced to the Helmholtz equation, which is solved using a fictitious domain method. Numerical experiments illustrate the above evolutionary methodology on a IBM SP2 parallel computer.

Raino A. E. Mäkinen and Jari Toivanen
University of Jyväskylä
Laboratory of Scientfic Computing
P.O. Box 35, FIN-40351 Jyvaskyla, Finland
E-mail: rainom@math.jyu.fi

PS
Parallel Domain Decomposition Method for Nonconforming Approximations of Elliptic Problems On Nonmatching Grids

An iterative method for solving systems of linear algebraic equations arising from nonconforming finite element discretizations with nonmatching grids for second order elliptic boundary value problems is considered. The method is based on decomposition of the original domain into nonoverlapping subdomains. The elliptic problem is presented in the macro-hybrid form with Lagrange multipliers at the interfaces between subdomains. A block diagonal preconditioner is proposed which is spectrally equivalent to the original saddle point matrix.

Serguei Maliassov
Institute for Mathematics and Its Applications
514 Vincent Hall
206 Church Str. S.E.
University of Minnesota
Minneapolis, MN 55455
malyasov@ima.umn.edu

PS
An Efficient NAS Parallel Benchmarks Characterization for Supercomputer Design

Experimental design of supercomputers calls for quantifiable methods to evaluate the requirements of different workloads within an application domain. Such methods can help establish the basis for scientific design of supercomputers driven by application needs, to optimize performance to cost. The selection of which features to include in a new computer system depends on the needs of the workloads the system willxecut. This paper investigates the supercomputer system needs for a class of contemporary benchmarks taken from NASA/NAS Parallel Benchmarks (NPB) suite. The NPB represents an implementation independent problem set, representative of Computational Aeroscience workload computations. The NPB workloads have been implemented on nearly every supercomputer platform and results have been reported by the vendors. Also, it is now widely cited in the high performance computing as a reliable measure of sustained performance. Data is presented for NPB requirements of these resources. The requirements suggest upper limits on the resources needed for efficient processors. In this study, we also examine nonuniformities in the distribution of instruction-level parallelism. Several nonuniformities in instruction-level parallelism are investigated including variation between benchmark class and by instruction class within benchmark. In addition, the average basic block length for NPB and instruction class distribution as well as the shortest path a workload would be executed on a supercomputer will be shown. The results confirm that workloads in NPB represent a wide range of non-redundant applications with different characteristics. Our results indicate that the parallelism obtained for NPB has a relatively smooth temporal profile which exhibits a high degree of uniformity in the parallelism.

Abdullah I. Meajil
The George Washington University
Washington, DC 20052
Email: abdullah@cesdis.gsfc.nasa.gov

PS
Performance Prediction of Parallel Systems Based on Large-Scale Parallel Workload Similarity

Performance prediction of workloads on parallel systems use a priori information to estimate performance when the input data size, or the machine parameters change. This work fills in an important gab. Given an application that has never been implemented on a target machine, we propose a methodology to predict the performance of such an application on that machine. This allows application developers make intelligent choices before committing to a specific machine, directly without having their own benchmarking activity. This is accomplished by representing the workloads using the parallel instruction centroid, which is a metric that embodies parallelism, critical path length, and instruction mixes properties. The difference between these centroids is measured as a representation of similarity. The most similar workload to ours is used for prediction, after compensating for the difference in communication requirements. In addition to filling the previously described gab, it will be shown that this method provides higher prediction accuracy in the majority of the cases, and accounts for dynamic code behaviors.

Abdullah I. Meajil
The George Washington University
Washington, DC 20052
Email: abdullah@cesdis.gsfc.nasa.gov

PS
Efficient Parallel Computing of a Longest Commom Subsequence of Three Strings

The longest Common Subsequence Problem (LCS Problem for short) of two, three or more strings, frequently arises in a number of areas such as genetic engineering, data compression, text processing and syntactic pattern recognition. This paper presents a parallel algorithm for the LCS problem of three strings. It generalizes the work of Li (*ParallelComput.* vol 20, 1994, pp. 1323-1334) on the LCS problem of two strings. Thank to an invariant we obtained by the study of the data structure of a sequential algorithm of the LCS problem of three strings: it yields a linear time parallel algorithm. This performance is remarkable since Hakata and Imai (LNCS n0 650, 1992, pp.469-478) addressed the following: "there has been no efficient algorithm for the LCS of three strings which outperforms the naive $O(n^3)$ algorithm.

Jean Frédéric Myoupo and Guillaume Luce
Université de Picardie-Jules Verne
LaRIA, Faculte de Mathematiques et d'Informatique
33, rue Saint Leu, 80039 Amiens, France
E-mail: myoupo@laria.u-picardie.fr

PS
The Computational Action Norm and the Principle of Computational Least Action

This paper defines a computer as a collection of instructions represented as computational momenta moving through computational phase space subject to appropriate boundary conditions. A program is a sequence of these momenta that performs a specified amount of computational work in some amount of computational time. The computational action norm of a program is the total computational action accumulated by the momenta during execution of the program and the distance between two programs is the distance defined by the action norm. The paper defines a metric space of programs based on the action norm and postulates the Principle of Computational Least Action to characterize program optimization as a variational principle. It illustrates the principle by computing the computational distance between a SAXPY kernel running on a computer with fast memory and the same kernel running on a computer with slow memory.

Robert W. Numrich
Cray Research
655 Lone Oak Drive, Eagan, MN 55121
E-mail: rwn@cray.com

PS
Wide-Area ATM Networking for Large-Scale MPPs

Oak Ridge and Sandia National Laboratories are involved in a project to network their two large Paragon MPPs using native ATM OC-12 (68 Mbytes/sec) interfaces. This will create a production 290 Gigaflop, 2800 node parallel computing resource. We will describe the project and focus on the implementation of efficient long-distance message passing over ATM. Performance results from several large-scale production codes will be presented.

Philip M. Papadopoulos
G.A. Geist II
Oak Ridge National Laboratory
Mathematical Sciences Section
P.O. Box 2008, Oak Ridge, TN 37831-6367
Email: phil@msr.epm.ornl.gov

PS
Nullspace Projected Arnoldi Method for Finding Stationary Values of Constrained Quadratic Ratios and Parallel Implementation

We will present an iterative nullspace projected Arnoldi method for the stationary values of the quadratic ratio under linear constraint $\rho(x) =_{C^T x=0} \frac{(x,Ax)}{(x,Bx)}$, where $A \in \mathbf{C}^{(n\times n)}$, $B \in \mathbf{R}^{(n\times n)}$, and $C \in \mathbf{C}^{(n\times k)}$ are all large sparse matrices. So far no efficient algorithms for this problem have been seen. We will derive our iterative algorithms based on Arnoldi method, discuss the convergence property, and provide the comparison of implementation on single workstation and on clustered workstations with MPI. Our test problems come from dielectirc waveguide analysis.

Chang Peng and Daniel Boley
Department of Computer Science
University of Minnesota
200 Union Street SE, rm 4-192
Minneaspolis, MN 55455
Email: cpeng@cs.umn.edu

PS
Introduction to the Real-time Parallel Benchmark Suite

Real-time computing on parallel machines has not received much attention, although many real-time applications can benefit from high performance computing. The Real Time Parallel Benchmark Suite is designed to help evaluate the suitability of parallel machines for real-time computations. The suite contains application level, kernel level, and low level benchmarks. The purpose of this paper is to introduce the benchmarks, report results to date, and encourage others to participate in benchmarking parallel machines.

Brian VanVoorst, Luiz Pires, Rakesh Jha
(vanvoors, pires, jha) @ htc.honeywell.com
Honeywell Technology Center
3660 Technology Drive
Minneapolis, MN 55418

PS
Evaluation of High Performance Fortran Compilers in the Implementation of the Shock Tube Problem

We evaluate three commercial HPF compilers using an explicit CFD code. The code implements a TVD scheme in the Riemann or shock tube problem using HPF. The access pattern in the code causes problems for the compilers; we examine two versions of the code so as to ease the burden on the compilers. We compare the performance of the codes produced by the HPF compilers with an equivalent MPI based code on the IBM SP2.

Kevin P. Roe and Piyush Mehrotra
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center, Mail Stop 403
Hampton, VA 23681-0001
E-mail: kproe@icase.edu

PS
Compile-Time Partitioning of Three-Dimensional Iteration Spaces

This paper presents a novel approach for compile-time partitioning of loop nests comprising two inner nested loops both of which have bounds linearly dependent on the index of the outermost, parallel loop; a simple example of such a loop nest is triangular matrix multiplication. Our approach is analysed using symbolic analysis techniques for enumerating loop iterations, which provide cost estimates ideal for parallelising compilers. Experimental results on a virtual shared memory parallel computer demonstrate the efficiency of the method.

Rizos Sakellariou
Department of Computer Science
University of Manchester
Oxford Road, Manchester M13 9PL, U.K.
e-mail: rizos@cs.man.ac.uk

PS
PINEAPL: A European Project to Develop Parallel Numerical Library Software for Industrial Applications

PINEAPL (Parallel Industrial NumErical Applications and Portable Libraries) is a collaborative project, funded by the European Commission, with partners in Denmark, France, Italy and the UK from industry and academic and research establishments. The aim of the PINEAPL project is to produce a library of parallel numerical software that is relevant to a wide range of industries. The library is intended to be portable and efficient across a wide range of parallel machines, and to utilise existing software wherever possible. Naturally, the numerical algorithms should be accurate, stable and robust. In this presentation we shall describe the project structure and report on progress so far.

Stef Salvini and Sven Hammarling
The Numerical Algorithms Group Ltd
Wilkinson House, Jordan Hill Road
Oxford OX2 8DR, United Kingdom
stef@nag.co.uk, sven@nag.co.uk

Thomas Christensen
Math-Tech ApS
Admiralgade 22, 1066 Copenhagen, Denmark
E-mail: thomas@math-tech.dk

PS
Three Dimensional Monte Carlo Device Simulation with Parallel Multigrid Solver

We present the results in embedding a multigrid solver for Poisson's equation into the parallel 3D Monte Carlo device simulator, PMC-3D, which was previously using the sequential successive overrelaxation solver. We have obtained significant speedups in the sequential case ranging from 5 to 15 on a single HP 712/80 workstation. We have also implemented the parallel multigrid solver, where the speedups range from 3 to 9 based on timing results on a 32-node nCUBE multiprocessor.

C.K. Sandalcı, Ç.K. Koç, and S.M. Goodnick
Department of Electrical and Computer Engineering
Oregon State University, Corvallis, OR 97331
Email: sandalci@ECE.ORST.EDU

PS
Optimal Parallel Algorithms for Matrix Multiplication

To achieve tight asymptotic bounds on running time for standard matrix multiplication of two (not necessarily square) matrices, we present what types of data layouts are needed and show that no one data layout can achieve optimal running times for all cases, instead optimal layouts depend on dimensions of each matrix and on the number of processors. To ensure results are applicable to a wide-spectrum of machines, we work within the well-known LogP model. Research partially supported by an NSF CAREER grant.

Eunice E. Santos
Department of Electrical Engineering and Computer Science
Lehigh University
Bethlehem, PA 18015
santos@eecs.lehigh.edu

PS
Interactive Visualization and Steering of Parallel CFD Methods

We present a method for interactively visualizing and steering a distributed Computational Fluid Dynamics application. We demonstrate the utility of being able to interactively adjust code parameters and visualize the result on the computed flow field as a distributed CFD application is running. This approach allows the user to adjust parameters which affect convergence rate such as time step size, or others such as inlet back pressure and to visualize the result as the application is running.

B. David Semeraro, Phil Papadopoulous
and James Kohl
Oak Ridge National Laboratory
P.O. Box 2008, Oak Ridge, TN 37831-6367
Email: semeraro@concorde.epm.ornl.gov,
phil@msr.epm.ornl.gov,
kohl@msr.epm.ornl.gov

PS
Efficient Parallel FFTs for Different Computational Models

We select the Fast Fourier Transfrom (FFT) to demonstrate a methodology for deriving the optimal parallel algorithm according to predetermined performance metrics, within a computational model. Following the vector space framework for parallel permutations, we provide a specification language to capture the algorithm, derive the optimal parallel FFT specification, compute the arithmetic, memory, communication and load–balance complexity metrics, apply the analytical performance evaluation to PRAM, LPRAM, BSP and LogP computational models, and compare with actual performance results.

Nadia Shalaby
Harvard University
Aiken Computation Laboratory
33 Oxford Street, Cambridge, MA 02138
E-mail: nadia@deas.harvard.edu

PS
A Data-Parallel Extension to Split-C

Split-C is an extension of C for parallel processing that allows arrays to be spread across processors. However, data parallel programming is not supported. We have developed an extension to Split-C for the CM-5 that includes parallel array operations as in Fortran 90/HPF and integrates these with the Split-C forms of assignment. We also introduce compiler algorithms for efficient handling of these parallel array operations so that their run-time perfomance approximates that of hand-written loops.

Joan D. Lukas, Kenneth Newman, and John P. Sullivan
Department of Mathematics and Computer Science
University of Massachusetts
100 Morrissey Blvd.
Boston, Boston, MA 02125-3393

PS
Tip C++: A Parallel Programming Model Based on The Abstract Structural Shared Memory

This paper studies parallel programming models in Clusters of Workstations for the purpose of finding a more efficient model to support a virtual global shared memory and multiple styles of parallelism description in a programming language. Firstly, we presents the Abstract Shared Memory Model. Then on this basis, we propose the parallel programming model Tip C++ supporting task parallel, data parallel and object parallel styles. The primitives of performance optimization, compile-time optimizations and task scheduling related with the model are also discussed in this paper. Keyword: Virtual shared memory (VSM), Distributed Shared Memory(DSM), Object-Oriented, Parallel Programming, Cluster of Workstations, task parallel, Data parallel, Object Parallel.

Tong Sun, Sanli Li, Guilin Zhou
and Yanheng Zheng
Xianghai University, Beijing, China

PS
SOLAR, a Portable Library for Scalable Out-of-Core Linear Algebra Computations

SOLAR is a portable high-performance library for out-of-core dense matrix computations. SOLAR works on parallel computers and on workstations. It combines portability with high performance by using existing high-performance in-core subroutine libraries and by using an optimized matrix input-output library. It supports in-core computations on both shared-memory and distributed-memory machines using the BLAS, LAPACK, and ScaLAPACK. SOLAR's matrix input-output module supports both conventional I/O interfaces and several flavors of parallel I/O interfaces.

Sivan Toledo and Fred G. Gustavson
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304
Email: toledo@parc.xerox.com

PS
A Further Proposal for a Fortran 90 Interface for LAPACK

The purpose of this paper is to discuss the design of a Fortran 90 interface to LAPACK. Our emphasis at this stage is on the design of an improved user-interface to the package, taking advantage of the considerable simplifications which Fortran 90 allows. The proposed design makes use of assumed-shape arrays, optional arguments, and generic interfaces. In this paper we implement interfaces to the subset of LAPACK routines for solving systems of linear equations $AX = B$ with a general matrix A, and for symmetric and Hermitian eigenproblems.
This research is partially supported by the Danish Natural Science Research Council through a grant for the EPOS project (Efficient Parallel Algorithms for Optimization and Simulation).

L. Susan Blackford, (formerly L. S. Ostrouchov)
Department of Computer Science, University of Tennessee
107 Ayres Hall, Knoxville, TN 37996-1301
Email: susan@cs.utk.edu

Jack J. Dongarra
Department of Computer Science, University of Tennessee
107 Ayres Hall, Knoxville, TN 37996-1301, and
Mathematical Sciences Section, Oak Ridge National Laboratory
P.O.Box 2008, Bldg. 6012, Oak Ridge, TN 37831-6367
Email: dongarra@cs.utk.edu

Jeremy Du Croz and Sven Hammarling
Numerical Algorithms Group Ltd
Wilkinson House, Jordan Hill Road, Oxford OX2 8DR, UK
Email: jeremy@nag.co.uk, sven@nag.co.uk

Jerzy Waśniewski
UNI•C, The DanishComputing Centre for
Research and Education, DTU, Bldg. 304
DK-2800 Lyngby, Denmark
Email: jerzy.wasniewski@uni-c.dk

PS
Tradeoffs and Performance Results of the Banded PRISM Eigensolver

The PRISM project has been investigating the performance of an implementation of its symmetric eigensolver which internally uses banded matrices. This variant raises tradeoffs that are unusual in an eigensolver, such as bandwidth in intermediate reductions, 1D to/from 2D data re-distributions, and bandgrowth control. This paper presents studies of these tradeoffs, highlights their interactions and suggests approaches for reconciling the various objectives. Performance comparisons to currently available parallel eigensolvers show the competitiveness of this approach.

Christian Bischof and William George
Argonne National Laboratory
9700 S. Cass Ave., Argonne, IL 60439

Steven Huss-Lederman
University of Wisconsin, Madison
Computer Sciences Department
Madison, WI 53706
lederman@cs.wisc.edu

Xiaobai Sun
Department of Computer Science
Duke University, Durham, NC 27708

Anna Tsao and Tom Turnbull
Center for Computing Sciences
17100 Science Drive, Bowie, MD 20715

Yuan-Jye Jason Wu
Argonne National Laboratory, Argonne, IL
9700 S. Cass Ave., Argonne, IL 60439
jwu@mcs.anl.gov

PS
Parallelization of SNePS — A Semantic Network Processing System

The ultimate goal of artificial intelligence is to build a system with human intelligence. With millions of concepts to be a system of human intelligence, almost all the existing systems cannot deliver their results within feasible time. With the emergence of parallel computers, massive parallel artificial intelligence opens an exciting realm of realizing the grand challenges.
Semantic network, where knowledge entities are represented by nodes and edges, provides a way to represent the underlying structure which other artificial intelligence systems lack of. By tracing the edges and nodes, new knowledge can be derived. In this paper, we propose a parallel semantic network system based on the syntax of SNePS. The objectives of this paper are to design a system which can (1) accommodate large amount of knowledge so that it can approach human intelligence; (2) achieve automatic deduction within tolerable period.

Chain-Wu Lee and Min-You Wu
Department of Computer Science
State University of New York at Buffalo
Buffalo, NY 14260
lee-d,wu@cs.buffalo.edu

PS
Data Distribution Analysis of MCGLS for Parallel Least Squares Problems

In this paper we mainly study different data distribution of MCGLS, a modified algorithm of CGLS which is a basic iterative method to organize the computation of conjugate gradient method applied to normal equations, for solving least squares problems on massively parallel distributed memory computers. The performance of CGLS on this kind of architecture is always limited because of the global communication required for the inner products. MCGLS can achieve some improvements on parallel performance by creating situations where communication can be overlapped with computation. Here a theoretical data distribution analysis is presented which allows us to decide best data decomposition for sparse and large least squares problems. It is shown that the row-block decomposition is much more efficient than other decomposition models. Several numerical experiments on Parsytec GC/PowerPlus are described as well.

Tianruo Yang
Department of Computer Science
Linköping University
S-581 83, Linköping, Sweden
E-mail: tiaya@ida.liu.se

Jin-yun Yuan
Departamento de Matemática
Universidade Federal do Paraná
Centro Politécnico, CP: 19.081
CEP: 81531-990, Curitiba, Brazil

PS
Isoefficiency Analysis of Krylov Subspace Methods on Parallel Distributed Memory Computers

For the solution of linear systems of equations, where the matrix is large and sparse, Krylov subspace methods are among the most frequently and most successfully used iterative solution methods. In this paper a performance model called isoefficiency concept is used to analyze the behavior of these methods implemented on massively parallel distributed memory computers. Two different mappings of data to processors are compared by putting these communication times into the isoefficiency concept that tries to model scalability aspects. Using the mapping strategy, performance can be improved by designing parallel algorithms such that the global communication time of inner product is reduced.

Tianruo Yang
Department of Computer Science
Linköping University
S-581 83, Linköping, Sweden

Jin-yun Yuan
Departamento de Matemática
Universidade Federal do Paraná
Centro Politécnico, CP: 19.081
CEP: 81531-990, Curitiba, Brazil

PS
Using Domain Decomposition Multiplicative Schwarz Method and Adaptive Local Refinement Strategy to Solve a Class of Nonlinear Parabolic PDE

For the purpose of parallel computing, domain decomposition methods have been used in solving a class of nonlinear parabolic pdes. Multiplicative Schwarz algorithm is applied to both on- and two- level overlapping subdomains. Local adaptive procedure is used to recursively solve the problem on the regions where higher resolution is needed. Extrapolation method with automatic joint order and stepsize control is used to solve the resulted differential algebraic equations. Efficiency is achieved by breaking up the whole large system into independent very small subsystems.

Hongyi Yu
Department of Mathematics
Tulane University
New Orleans, La 70118
E-mail: hoy@math.tulane.edu

PS
DATIS: A Data Transfer Interface for SPMD Programs of Scientific Computing

DATIS is a software interface for SPMD programs in parallel scientific computing to accomplish data transfer. It reduces the complexity of parallel programming. DATIS provides a simple user interface, grouped-data-transfer functions and a mechanism to overlap computation and communication. By using DATIS, the deadlock, which may occur when several messages are transfered among processors, can be avoided. In DATIS, the data in grouped-data-transfer can be packed and the communication overhead

can be reduced.

Xiang Yu
Department of Computer Science
Princeton University
Princeton, New Jersey 08544

Xiaoge Wang
Department of Computer Science
Tsinghua University
Beijing, 100084, P.R. China
wang@tirc.dcs.tsinghua.edu.cn