SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS

Final Program

Third SIAM Conference on Parallel Processing for Scientific Computing

DECEMBER 1–4, 1987

The Westin Bonaventure Hotel
Los Angeles, CA

CONFERENCE THEMES

- High-Speed Computer Architectures
- Computer Performance Evaluation
- High-Speed Computer Environments
- Numerical Domain Decomposition Methods
- Scientific Programming Languages
- Numerical Linear Algebra
- Distributed Computing
- Computational Complexity
- Adaptive Numerical Methods
- Numerical Particle Methods

Sponsored by the SIAM Activity Group on Supercomputing

TABLE OF CONTENTS

ORGANIZING COMMITTEE

Garry Rodrigue, Chair
Lawrence Livermore National Laboratory and University of California, Davis

Joseph Oliger
Stanford University

John Van Rosendale
Argonne National Laboratory

Burton Smith
Supercomputing Research Center

Robert G. Voigt
ICASE
NASA Langley Research Center

FUNDING AGENCIES

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

MEETING HIGHLIGHTS

INVITED PRESENTATIONS

Tuesday, December 1, 8:30 AM
Invited Presentation 1
Parallel Computing Via Multidomain Techniques for Spectral Methods

Multidomain techniques have become a popular way for the application of spectral methods to simulating flow on complex geometries. It turns out that these techniques suggest a natural way of doing the computations in parallel, namely each subdomain is treated in a different processor. The way of transferring the information from one processor to another becomes crucial, especially for hyperbolic problems that arise from simulating compressible flows.

The author will review spectral multidomain techniques in the context of parallel computations. Ways of improving boundary conditions were explored that will speed up convergence. In particular, ways of composing matching conditions were tested for hyperbolic equations —in particular the Euler equations of gas dynamics.

The plan is to use the Flex machine at ICASE to test the validity and efficiency of the multidomain technique. This research attempts to give an answer to the question whether it is preferable to parallelize the algorithm without physically subdividing the domain of computations, or whether it is more gainful to divide the domain and to treat the problem in different subdomain.

In both cases, the connection between boundary conditions and transfer of information between processor will be discussed and clarified.

David Gottlieb
Brown University

Tuesday, December 1, 9:15 AM
Invited Presentation 2
Adaptive Mesh Refinement for Parallel Processors

Adaptive algorithms can be difficult to parallelize efficiently. This is because:
(1) adaptive methods use more complicated data structures than arrays, making parallel access to these data structures difficult;
(2) adaptive methods concentrate the work in localized regions, making load balancing on a multiprocessor difficult.

In this work, an adaptive mesh refinement algorithm for solving the Euler equations is implemented on a Cray XMP with four processors. This method of adaptive refinement is based on the use of logically rectangular fine grid patches embedded in a coarse, underlying grid. This method of nesting multiple grids allows the bulk of the computational work to be done on regular grids, although the complete grid hierarchy is irregular. We discuss both a fine grained and coarse grained approach to parallelizing the algorithm. A binary decomposition strategy is used to load balance the computation. The binary decomposition algorithm is easy to implement, but does not take into account the communication overhead that would occur on a non-shared memory machine. We show results, and discuss the overhead of this method. Surprisingly, the localization of work, combined with the adaptive data structures, make implementation of the binary decomposition an easy task.

Marsha J. Berger
Courant Institute of Mathematical Sciences
New York University

Tuesday, December 1, 1:30 PM
Invited Presentation 3
Neural Computing

The questions to be addressed in this talk are two-fold: what is the appropriate set of tools for the study of the highly parallel computing networks in the human brain; and what are the strategies for building highly parallel computers suitable for programming with "intelligence." The thrust of this paper is that there overall architectural principles that unite both sides of this study: namely, that a computer no longer be thought of as a unitary system, but rather as a network of more specialized devices; and that many of these devices be structured as highly parallel arrays of interacting neuron like components. We shall illustrate this with a brief discussion of the following topics: the high level architecture of the frog's brain as revealed in studies of the mechanisms *visual motor coordination;* the use of supercomputing in modelling the interacting levels of the retina; stability analysis in the study of neural networks; the implications of associative memory techniques for intelligence computer design.

Michael A. Arbib
University of Southern California, Los Angeles

Tuesday, December 1, 2:15 PM
Invited Presentation 4
Optical Computing Systems

There are several computing problems that require much more computing power than now is available. Optics can offer potentially greater speed and parallelism. In addition, in several cases, the data are in optical form. Examples of the latter are optical fiber data systems where the electronic control now slows operation and image processing systems where the data are naturally in optical form. Several optical computing systems being built within the Center for Optoelectronic Computing Systems and elsewhere will be discussed.

Examples are a bit-serial machine to explore architectures and the speed limits; an optical cellular array for signal processing; a connection intensive system for pattern recognition and processing; optical symbolic computing for optical artificial intelligence machines; and optical associative memory that uses four-wave mixing and holography.

W. Thomas Cathey
University of Colorado, Boulder

INVITED PRESENTATIONS

Wednesday, December 2, 8:30 AM
Invited Presentation 5
Automatic Decomposition of Fortran Programs for Execution on Multiprocessors

It is clear that future generations of scientific supercomputers will employ multiple independent processors. What form of programming support software should be provided with such machines? Existing Fortran programs, written for sequential machines, are not well suited to parallel execution. If these programs are to run efficiently on a multiprocessor system, they must be decomposed into subproblems that can be executed in parallel.

The usual approach to decomposition is to provide language primitives or system calls that permit concurrent programming in Fortran. The programmer is thus responsible for handling all the synchronization. Unfortunately, concurrent programming is unnatural for many scientific programmers. It would be better to permit the programmer to design parallelism into his programs at an abstract level and have the programming environment automatically construct synchronization primitives.

In his talk, the author explores automatic techniques for uncovering parallelism in Fortran programs and for translating them to run efficiently on multiprocessor supercomputers. These techniques rely upon a sophisticated theory of dependence in programs developed by Kuck at Illinois and extended for our previous work on automatic vectorization. In addition, the methods rely extensively on intraprocedural and interprocedural data flow analysis.

In order to achieve enough parallelism to make the translation profitable, a number of very ambitious transformations are required, including loop distribution, loop interchange, loop alignment, and loop fusion. The problem of generating optimal code using these transformations has been shown to be intractable, but heuristic techniques have proved effective in practice. The talk will conclude with a discussion of our experience with a prototype translation system, called PFC Plus, developed at Rice.

Ken Kennedy
Rice University

Wednesday, December 2, 9:15 AM
Invited Presentation 6
Schwarz Splittings and Template Operators

Schwarz' alternating method (SAM) is an old mathematical technique dating from 1860. The method was commonly employed as a tool for theoretical analysis, but it is still not yet widely used for large scale scientific computations. The earlier experiences showed that SAM converged slowly. Our analyses show that some new methods based on SAM are competitive with other solution techniques. Some generalizations of SAM, namely Schwarz splittings (SS), are presented here. Several acceleration schemes are also discussed in this talk. In particular, when these techniques are applied to the solution of the model problem, an optimal complexity can be achieved. For many important applications, such as performing parallel computations in a non-shared memory environment, using composite grids and also applying fast solvers in an irregular region, SS's are found to be useful techniques.

In order to identify the types of problems for which these new techniques are most suitable, a new structure for the linear operators called Template operators has been developed. Some decay results for the elements of the inverses of sparse operators are given. These results provide a theoretical basis for determining when these SS techniques can be used successfully.

Wei Pai Tang
Stanford University

Wednesday, December 2, 1:30 PM
Invited Presentation 7
Parallel Systems: Performance Modeling and Numerical Algorithms

Historically, there have been two major techniques for modeling parallel systems: discrete event simulation and mathematical analysis. Although simulation models can mimic a real-world system as closely as understanding permits and needs require, highly detailed simulation models can be computationally taxing. In contrast, analytic techniques often can quickly provide mathematical insight into the behavior of systems over a broad range of parameter values. Their major limitation is the number of restrictive assumptions that often must be satisfied to insure tractability and accuracy. For many parallel systems, evaluation of realistic analytic models is as complex or even more complex than equivalent simulations.

Transient behavior, complex access patterns, and large state spaces all make analysis difficult. Eliding unnecessary detail while retaining important features is the essence of the modeler's art. The author presents a set of techniques for analyzing the performance of parallel architectures. These techniques permit the comparative evaluation of interconnection networks for message-based processors (e.g., hypercubes) as a function of communication pattern, including communication locality. In addition, the author shows how simple analytic techniques can capture the salient details of network traffic and memory conflicts in shared memory processors.

Daniel A. Reed
University of Illinois, Urbana

Wednesday, December 2, 2:15 PM
Invited Presentation 8
Asymptotics and Domain Decomposition in Parallel Computing

Mathematical models for scientific phenomena usually involve multiple scales, with the consequence of rapid temporal or spatial variations in the solution. While the details of these rapid variations may not be of interest, the numerical computations must somehow incorporate any effects on those aspects of the solution one is interested to determine.

Where the local behavior of the solution is known, analytic patches may be used, while in other regions special local grids will have to be employed in order to resolve the special features of the solution.

The sub-domains together with rules governing their interactions form a domain decomposition method. Domain decomposition derived from asymptotic analysis can take advantage of parallel vector architecture and represents, indeed, a truly "new" class of parallel algorithms.

In his talk, the author surveys the asymptotic methods employed to identify rapid varying regions and to analyze their local behaviors. He will discuss the incorporation of the local asymptotic solutions and the partition of subdomains into a numerical scheme via a domain decomposition algorithm. An analysis of the domain decomposition algorithm within the frame-work of asymptotic preconditioning and deferred correction will be presented and the mapping of an asymptotics-induced domain decomposition method onto two parallel processing machines will be discussed.

Raymond C. Y. Chin
Lawrence Livermore National Laboratory

Thursday, December 3, 8:30 AM
Invited Presentation 9
A Fast Multipole Algorithm for Large-Scale Particle Simulations

The evaluation of Coulombic or gravitational interactions in large-scale ensembles of particles is an integral part of the numerical simulation of a large number of physical processes. Examples include celestial mechanics, plasma physics, the vortex method in fluid dynamics, molecular dynamics, and the solution of the Laplace equation via potential theory. In a typical application, a numerical model follows the trajectories of a number of particles moving in accordance with Newton's second law of motion in a field generated by the whole ensemble. In many situations, in order to be of physical interest, the simulation has to involve thousands of particles (or more), and the fields have to be evaluated for a large number of configurations. Unfortunately, an amount of work of the order $O(N^2)$ has traditionally been required to evaluate all pairwise interactions in a system of N particles, unless some approximation or truncation method is used. As a result, large-scale simulations requiring high accuracy (as encountered in highly correlated systems) have been extremely expensive in some cases, and prohibitive in others.

The author presents an algorithm for the rapid evaluation of the potential and force fields in large-scale systems of particles. In order to evaluate all pairwise Coulombic interactions of N particles to within round-off error, the algorithm requires an amount of work proportional to N, and this estimate does not depend on the statistics of the distribution. Both two and three dimensional versions of the algorithm have been constructed, as we will discuss their applications to several problems in physics, chemistry, biology, and numerical complex analysis.

The author concludes with a discussion of the implementation of the algorithm on a variety of parallel architectures and observes that almost all communications required by the procedure are local, making it well-suited for such machines.

Leslie Greengard
Yale University

MEETING HIGHLIGHTS

INVITED PRESENTATIONS

Thursday, December 3, 9:15 AM
Invited Presentation 10
Lattice Gas Methods for Solving Partial Differential Equations

Lattice gas methods are logical, discrete techniques for efficiently solving partial differential equations. These methods have been shown to solve 2D compressible Navier Stokes, reaction-diffusion, 2D MHD, and Burger's equations. They can exactly conserve mass, energy, and momentum; they are fast (about one billion cell updates per second on a CRAY XMP48); they are memory efficient (one CRAY word defines 10.6 cells). The codes are robust, efficient, and short (approximately 50 FORTRAN lines); complicated boundaries are easily incorporated. They are fully parallel (N processors run N times faster). Estimates indicate that a $(512)^3$ processor computer could cost $300,000. Small simulators (256×514 cells) have been built for around $1000. Present lattice gas methods are constrained by Mach numbers that cannot exceed 1.4, density-dependent viscosity, velocity-dependent equations of state, and cell-averaging requirements.

The author will review recent developments, presenting details of 2D and 3D Navier Stokes solvers along with movies of complicated flows and concluding with a discussion of future challenges.

Gary Doolen
Los Alamos National Laboratory

Thursday, December 3, 1:30 PM
Invited Presentation 11
Strategies for the Mapping and Scheduling of Scientific Computations

Work underway at Yale and at ICASE on automated problem mapping, runtime scheduling of computations and load balancing will be described. Through the analysis of a variety of problems in scientific computing, the following tradeoffs are explored: (1) communication costs and load imbalance in multiprocessors with preferential access to local memories, and (2) synchronization costs and load imbalance in machines where the access time to all memory is approximately equal.

Parameterized mapping schemes are developed that allow one to adjust the granularity of parallelism in several different ways. The chosen mapping influences the balance of load as well as the observed communication and synchronization overheads. Both analytic modeling and empirical studies on the Intel iPSC hypercube and the Encore Multimax are utilized in this work. The prospects for incorporating what has been learned in these studies into the Crystal/ACRE parallel programming environment will be discussed.

Joel Saltz
Yale University

Thursday, December 3, 2:15 PM
Invited Presentation 12
Graph Theory plus Group Therory equals Answers to Communication Problems

A primary obstacle in obtaining significant speedup of highly parallel computer programs on massively parallel distributed processors is global communication. For this reason, we have been investigating theories which enable us to exhibit provably optimal global communication schemes on a wide class of interconnection networks.

One such class for which a theoretical treatment is successful are those networks which can be represented as the directed graph of a group. The elements of the group represent processors (the nodes of the graph) while the generators correspond to communication lines (the arcs of the graph). The author has shown that if the group processes a certain sequencing property P then various global communications tasks (e.g., universal broadcast, accumulation or exchange) can be performed in minimal time.

The hypercube is a prime example of a network which can be represented as the graph of a group (an elementary abelian 2-group) and the author has been able to show that this group does in fact have property P. At this time no groups are known to the author that do not have property P.

Plans are to extend these notions to networks which cannot be represented as the graph of the group, but may still possess optimal global communications schemes. Ultimately, one should expect to be able to measure the communication performance of an interconnection network in these terms, in order to identify those networks that are in some sense "optimal" in terms of global communication.

Vance Faber
Los Alamos National Laboratory

Friday, December 4, 8:30 AM
Invited Presentation 13
Parallel Processing Research at IBM

Two highly parallel processor prototyping efforts are currently underway at the IBM T.J. Watson Research Center: GF11, a Single Instruction Multiple Data Stream (SIMD) array of processors which can achieve close to 10 billion floating point operations per second on certain computations. The Research Parallel Processor Prototype (RP3), which is a Multiple Instruction Stream Multiple Data Stream (MIMD) system consisting of conventional microprocessors. A high bandwidth, low latency switch together with a flexible memory structure makes this machine potentially suitable for a variety of applications. The talk will briefly describe these two projects and report on their current status. Motivations, goals, and technical plans will be discussed to provide an overall perspective for these efforts.

Tilak Agerwala
IBM T.J. Watson Research Center

Friday, December 4, 9:15 AM
Invited Presentation 14
Programming Environments for Multiprocessor Architectures

Programming multiprocessor architectures is a complex task which requires the user to have thorough knowledge of the machine being targeted. Several different approaches are being explored so as to make this task easier. These range from efforts aimed at automatically transforming sequential programs for parallel execution to designing new languages which will allow the explicit control of the parallelism. The primary goal of all these efforts is to provide an environment which allows the programmer to express the algorithm in a manner which not only leads to efficient execution but also provides some measure of portability across architectures. In his talk the author will survey the different approaches that are currently being explored.

Piyush Mehrotra
Purdue University

SPECIAL FUNCTIONS

Welcoming Reception
Monday, November 30, 8:00 PM – 10 PM
Pasadena Exhibit Hall
Westin Bonaventure
Cash Bar

Beef and Beer Party
Tuesday, December 1, 7:30 PM – 9:30 PM
Pasadena Exhibit Hall
Westin Bonaventure
$15.00

Meeting of the SIAM Activity Group on Supercomputing
Wednesday, December 2, 7:00 PM
Pasadena Exhibit Hall
Cash Bar

FINAL PROGRAM

Monday, November 30/PM

4:00 PM/California Foyer
Registration Opens

8:00 PM/California Foyer
Registration Closes

Tuesday, December 1/AM

7:00 AM/California Foyer
Registration Opens

8:15 AM/San Diego Room
Welcoming Remarks

8:30 AM/San Diego Room
Invited Presentations 1 and 2
Chair: Garry Rodrigue
Lawrence Livermore National Laboratory

8:30
Parallel Computing Via Multidomain Techniques for Special Methods
David Gottlieb
Brown University

9:15
Adaptive Mesh Refinement for Parallel Processors
Marsha J. Berger
Courant Institute of Mathematical Sciences
New York University

10:00 AM/Pasadena Exhibit Hall
Coffee

10:40 AM
CONCURRENT SESSIONS

Tuesday, December 1, 10:40 AM–12:00 Noon
Contributed Presentations 1/San Diego Room

Chair: Robert G. Babb, II, Oregon Graduate Center

10:40/46/A1
The Design and Implementation of Parallel Algorithms in Ada
Richard F. Sincovec
University of Colorado, Colorado Springs

11:00/123/A1
DINO—A New Language for Numerical Computation on Local Memory Multiprocessors
Matthew Rosing and Robert B. Schnabel
University of Colorado, Boulder

11:20/160/A1
Scientific Parallel Processing with LGDF2
Robert G. Babb, II and David C. DiNucci
Oregon Graduate Center

11:40/157/A2
The BF (Boundary-Fitted) Coordinate Transformation Technique of DEQSOL (Differential EQuation SOlver Language)
Chisato Konno, Michiru Yamabe, Miyuki Saji and Yukio Umetani
Hitachi Ltd.

Tuesday, December 1, 10:40 AM–12:00 Noon
Contributed Presentations 2/Sacramento Room

Chair: Kai Hwang, University of Southern California

10:40/105/A2
Communication Complexity of Matrix Multiplication
David C. Fisher
Harvey Mudd College

11:00/19/A2
Orderings for Parallel Sparse Symmetric Factorization
Charles E. Leiserson, Massachusetts Institute of Technology and Thinking Machines Corporation and John G. Lewis, Boeing Computer Services and Massachusetts Institute of Technology Center for Advanced Engineering Study

11:20/162/A3
An Efficient Fixed Size Array for Solving Large Scale Toeplitz Systems
Afshin Daghi, V. K. Prasanna Kumar and Ali Safavi
University of Southern California

11:40/188/A3
Architecture and Operation of a Systolic Sparse Matrix Engine
Robert J. Dunki-Jacobs, Steven W. Hammond, Robert M. Hardy and Terry M. Topka, GE Corporate Research and Development Center

Tuesday, December 1, 10:40 AM–12:00 Noon
Contributed Presentations 3/San Gabriel A–C

Chair: Ahmed Sameh, University of Illinois, U-C

10:40/170/A3
Accelerating with Rank-One Updates
Timo Eirola and Olavi Nevanlinna
Helsinki University of Technology, Finland

11:00/91/A3
A Parallel, Hybrid Algorithm for the Generalized Eigenproblem
Shing Ma, Merrell Patrick and Daniel Szyld
Duke University

11:20/143/A4
Multiprocessor Algorithm for the Symmetric Generalized Eigenvalue Problem
Bill Harrod, Ahmed Sameh and Mark Schaefer
University of Illinois, Urbana-Champaign

11:40/182/A4
Parallel QR Factorization on a Hypercube Multiprocessor
Eleanor Chu, University of Waterloo, Alan George, University of Tennessee and Oak Ridge National Laboratory

Tuesday, December 1, 10:40 AM–12:00 Noon
Contributed Presentations 4/Santa Barbara A–C

Chair: G.W. Hedstrom, Lawrence Livermore National Laboratory

10:40/14/A4
Shock Calculations on Multi Processors
Bjorn Sjogreen
Uppsala University, Sweden

11:00/126/A5
Parallel Processing of a Domain Decomposition Method
R. C. Y. Chin, G. W. Hedstrom, Lawrence Livermore National Laboratory, J. S. Scroggs, University of Illinois, Urbana-Champaign and D. C. Sorensen, Argonne National Laboratory

11:20/18/A5
Moving Point and Particle Methods for Convection-Diffusion Problems
Michael Rees
Numerical Analysis Group, United Kingdom

11:40/177/A5
Random-Walk Simulation of Diffusion-Reaction-Convection
D. J. Hebert
University of Pittsburgh

12:00 Noon
Lunch

Tuesday, December 1/PM

1:30 PM/San Diego Room
Invited Presentations 3 and 4
Chair: Burton Smith
Supercomputing Research Center

1:30
Neural Computing
Michael A. Arbib
University of Southern California, Los Angeles

2:15
Optical Computing Systems
W. Thomas Cathey
University of Colorado, Boulder

3:00 PM/Pasadena Exhibit Hall
Coffee

Tuesday, December 1, 3:40–5:00 PM
Contributed Presentations 5/San Diego Room

Chair: Edward K. Blum, University of Southern California

3:40/3/A5
Parallel Elliptic Solvers
E. Gallopoulos and Y. Saad
University of Illinois, Urbana-Champaign

4:00/17/A6
The Parallel Waveform Relaxation Multigrid Method
Vandewalle Stefan
Katholieke Universiteit Leuven, Belgium

4:20/31/A6
A Pipelined Block QR Algorithm for a Ring of Vector Processors
Christian H. Bischof
Cornell University

4:40/32/A6
Conjugate Gradient Preconditioners for Parallel Computers
Gerard Meurant
Departement de Mathematiques Appliques, France

Note: The numbers that follow the individual times for each presentation represent the numerical order and page number of the abstract.

FINAL PROGRAM

Tuesday, December 1/PM
Continued

Tuesday, December 1, 5:00–7:00 PM
Poster Session 1/Pasadena Exhibit Hall/Cash Bar

1/A22
Scheduling of Precedence-Constrained Tasks on Multiprocessors
C. C. Price, Stephen F. Austin State University and M. A. Salama, California Institute of Technology

2/A22
Iterative Suboptimal Strategies for Improving the Transients in Adaptive Control Systems
M. De La Sen and M. J. Gonzalez-Gomez, Universidad Del Pais Vasco, Spain

5/A22
A Comparison of Hydrodynamics Multitasking Algorithms
David A. Mandell, Los Alamos National Laboratory

6/A23
Efficient Parallel Algorithms for Controlability and Eigenvalue Assignment Problems
B. N. Datta and Karabi Datta, Northern Illinois University

7/A23
Use of Matrix Determinant of Kronecker Product in Parallel Matrix Computations
Karabi Datta, Northern Illinois University

11/A23
Task Assignment in a Multiprocessor System
C. Siva Ram Murthy and V. Rajaraman, Indian Institute of Science, India

12/A23
Algorithm and Performance Notes Concerning Block LU Factorization
Jim Armstrong, CONVEX Computer Corporation

13/A24
Use of Mathematical Switches to Solve Differential Equation Problems
Yi-ling F. Chiang, New Jersey Institute of Technology

16/A24
Block Truncated-Newton Methods for Unconstrained Minimization
Stephen G. Nash, George Mason University

22/A24
Calculating Good Initial Guesses for Iterative Solutions of Parabolic Partial Differential Equations in Parallel
A. Louise Perkins, University of California at Davis in Livermore

23/A25
Systematic Method of Condition/Event Nets Design
Boleslaw Mikolajczak, Southeastern Massachusetts University

26/A25
Neural Nets and Sparse Matrices
Horst D. Simon, Boeing Computer Services

27/A25
Large Memory Applications on the SCS-40
R. E. Anderson, R. G. Grimes, A. C. Mong and H. D. Simon, Boeing Computer Services

29/A25
A Parallel Matrix Inversion Algorithm for Triangular Toeplitz Systems
You Zhaoyong and Li Lei, Xian Jiaotong University, China

30/A26
Two Supernodal Implementations of General Sparse Factorization for Vector Computers
C. Cleve Ashcraft, John G. Lewis and Barry W. Peyton
Boeing Computer Services

33/A26
Recursive Doubling Algorithm for Solution of Tridiagonal Systems on Hypercube Multiprocessors
Omer Egecioglu, Alan J. Laub and Cetin K. Koc, University of California, Santa Barbara

37/A26
Matrix Processing Computers — The Logical Step into the Future
David E. T. Foulser, Saxpy Computer Corporation

39/A26
Parallel Implementation of a Neumann-Dirichlet Preconditioner in Domain Decomposition
Wlodek Proskurowski and Majid Haghoo, University of Southern California

40/A27
The Network Emulation Problem
James Abello, University of California, Santa Barbara, Donna J. Brown, University of Illinois, Michael R. Fellows, University of New Mexico and Michael A. Langston, Washington State University

44/A27
Performance Evaluation for Stochastic Dynamic Programming on Vector Multiprocessor
Floyd B. Hanson, Argonne National Laboratory and University of Illinois at Chicago

45/A27
Parallel Runge-Kutta Methods for Ordinary Differential Equations
Syvert P. Norsett and Ivar Lie, University of Trondheim, Norway

47/A27
Scientific Computation in FORTRAN 8X
Alan Wilson, Active Memory Technology

50/A28
Simulation of Larger Dimensions on the Intel iPSC Hypercube
Johnny Petersen and Jan Lindheim, Chr. Michelsen Institute, Norway

51/A28
Parallel Implementation of a New Linear Matrix Solver
Lisette de Pillis, Johnny Petersen, Chr. Michelsen Institute and John de Pillis, Bergen Scientific Centre, Norway

53/A28
Parallel Implementation of Fourier Pseudospectral Methods on a Loosely Coupled Array of Processors
Zaphiris Christidis and Vijay Sonnad, IBM Corporation

54/A28
Optimal Algorithms for Parallel Matrix Multiplication on Vector Computer
You Zhaoyong and Li Lei, Xian Jiaotong University, China

55/A29
Exploiting Multilevel Parallelism in PDE Software on the CRAY X-MP
Michael Bieterman, Boeing Computer Services

57/A29
An Extension of Ryckaer's Algorithm
Sheng-Bai Zhu, Texas Tech University

58/A29
Solving Very Large Dense Systems of Linear Equations
Michael Pernice, National Center for Atmospheric Research

59/A29
Low-Accuracy, High Speed Methods in the Numerical Solution of Ordinary Differential Equations
Chris R. Coray, Utah State University and Jean Michel Favre, Tektronix Inc.

60/A29
Application of Supercomputers in MD Study of Liquid Phase Isomerizations
Sheng-Bai Zhu, Texas Tech University

61/A30
Fortran 9y: Extensions to FORTRAN for Supporting Network Connected Computers
D. E. Stevenson, Clemson University

62/A30
Parallel Algorithms for Market Equilibrium Problems
Ana Nagurney, University of Massachusetts

63/A30
Concurrent I/O System for the Hypercube Mutliprocessor
Andrew Witkowski, California Institute of Technology

193/A30
A Pseudo-Spectral Method Implemented on a SIMD Machine
J. V. W. Reynders, Naval Research Laboratory, and Princeton University

L-1/A31
Parallel Multigrid Algorithms in Computational Fluid Dynamics
Eric Barszcz, NASA Ames Research Center; Tony F. Chan, University of California, Los Angeles, and RIACS NASA Ames Research Center; Dennis Jespersen, NASA Ames Research Center; and Ray S. Tuminaro, Stanford University and RIACS NASA Ames Research Center

L-2/A31
Matrix Decomposition on Message Passing Architectures
M. Cosnard and B. Tourancheau, Institut National Polytechnique de Grenoble, France

L-3/A31
A Linear Processor Network for the Knapsack Problem
M. Cosnard and A. G. Ferreira, Institut National Polytechnique de Grenoble, France

7:30 PM/California Foyer
Registration Closes

7:30 PM/Pasadena Exhibit Hall
Beef and Beer Party
$15.00

Note: The numbers that follow the individual times for each presentation represent the numerical order and page number of the abstract.

FINAL PROGRAM

Wednesday, December 2/AM

8:30 AM/San Diego Room
Invited Presentations 5 and 6
Chair: Joseph Oliger
Stanford University

8:30
Automatic Decomposition of Fortran Programs for Execution on Multiprocessors
Ken Kennedy
Rice University

9:15
Schwarz Splittings and Template Operators
Wei Pai Tang
Stanford University

10:00 AM/Pasadena Exhibit Hall
Coffee

10:40 AM
CONCURRENT SESSIONS

Wednesday, December 2, 10:40 AM–12:00 Noon
Contributed Presentations 6/Santa Barbara A–C

Chair: A. Louise Perkins, University of California, Davis

10:40/129/A6
Concurrent Management of Priority Queues for Adaptive Algorithms
John Kapenga and Elise deDoncker-Kapenga
Western Michigan University

11:00/83/A7
Programming Abstractions for Run-Time Partitioning of Scientific Continuum Calculations Running on Multiprocessors
Scott B. Baden
University of California, Berkeley

11:20/64/A7
Implementation of the Acceptance-Rejection Method on Parallel Processors: A Case Study in Scheduling
William Celmaster
Bolt Beranek and Newman

11:40/74/A7
On the Placement of Parallel Processes
Michael R. Leuze and Stephen R. Schach
Vanderbilt University

Wednesday, December 2, 10:40 AM–12:00 Noon
Contributed Presentations 7/San Gabriel A–C

Chair: Gregory A. Darmohray, Lawrence Livermore National Laboratory

10:40/104/A7
Mapping Large Scale Computational Problems on a Highly Parallel SIMD Computer
H. M. Liddell, Queen Mary College
D. Parkinson, University of London, United Kingdom

11:00/24/A8
Multiprogramming and the Performance of Parallel Programs
Muhammad S. Benten and Harry F. Jordan
University of Colorado, Boulder

11:20/122/A8
Using Mathematical Modeling to Aid in Parallel Algorithm Development
Elizabeth Eskow and Robert B. Schnabel
University of Colorado, Boulder

11:40/4/A8
The Cerberus Multiprocessor Simulator
Eugene D. Brooks, III, Timothy S. Axelrod and Gregory A. Darmohray
Lawrence Livermore National Laboratory

Wednesday, December 2, 10:40 AM–12:00 Noon
Contributed Presentations 8/San Diego Room

Chair: Youcef Saad, University of Illinois, U-C

10:40/109/A8
A New Parallel Algorithm for Linear Triangular Systems
Avi Lin and He Zhang
Temple University

11:00/142/A9
A Parallel Algorithm for the Singular Value Decomposition of Rectangular Matrices
William D. Shoaff and David C. Chan
Florida Institute of Technology

11:20/125/A9
A Parallel Algorithm for the Singular Value Decomposition
E. R. Jessup, Yale University and D. C. Sorensen, Argonne National Laboratory

11:40/89/A9
A Multiprocessor Scheme for Singular Value Decomposition
Michael Berry and Ahmed Sameh
University of Illinois, Urbana-Champaign

Wednesday, December 2, 10:40 AM–12:00 Noon
Contributed Presentations 9/Sacramento Room

Chair: Jeffrey Scroggs, University of Illinois, Urbana

10:40/35/A9
A Parallel Homotopy Method for Solving a System of Polynomials
Lionel M. Ni, Shui-Nee Chow and Yun-Qiu Shen
Michigan State University

11:00/94/A10
The Granularity of Homotopy Algorithms for Polynomial Systems of Equations
L. T. Watson, J. Harimoto and J. P. Bixler
Virginia Polytechnic Institute and State University

11:20/167/A10
A Parallel Nonlinear Integer Programming Algorithm Based on Branch and Bound and Simulated Annealing
Ken S. Bosworth and G. S. Stiles
Utah State University

11:40/131/A10
Recursive Binary Partitions
George Cybenko, Tufts University and Tom Allen, Alphatech, Inc.

12:00 Noon
Lunch

Wednesday, December 2/PM

1:30 PM/San Diego Room
Invited Presentations 7 and 8
Chair: Jack Dongarra
Argonne National Laboratory

1:30
Parallel Systems: Performance Modeling and Numerical Algorithms
Daniel A. Reed
University of Illinois, Urbana

2:15
Asymptotics and Domain Decomposition in Parallel Computing
Raymond C. Y. Chin
Lawrence Livermore National Laboratory, and National Science Foundation

3:00 PM/Pasadena Exhibit Hall
Coffee

Wednesday, December 2, 3:40–5:00 PM
Contributed Presentations 10/San Diego Room

Chair: Peter Linz, University of California, Davis

3:40/49/A11
Parallel Architectures for Optical Computing
Kai Hwang and Ahmed Louri
University of Southern California

4:00/76/A11
Synchronization of Nonhomogeneous Parallel Computations
Dan C. Marinescu and John R. Rice
Purdue University

4:20/120/A11
Intrinsically Parallel Algorithms
Oliver A. McBryan
University of Colorado, Boulder

4:40/124/A11
A Software Tool for Building Supercomputer Applications
Dennis Gannon, Daya Atapattu, Mann Ho Lee and Bruce Shei
Indiana University, Bloomington

Special Notice to Contributed Presentation Authors and Chairmen of Contributed Presentation Sessions:

Twenty minutes are allowed for each contributed presentation. Presenters are requested to spend a maximum of 17 minutes for their presentation, and 3 minutes for questions and answers.

Please note:

For presentations with more than one author, an underlineation is used to denote the author who will present the paper.

Note: The numbers that follow the individual times for each presentation represent the numerical order and page number of the abstract.

Wednesday, December 2/PM
Continued

Wednesday, December 2, 5:00–7:00 PM
Poster Session 2/Pasadena Exhibit Hall/Cash Bar

65/A32
Fast Fourier Transforms on the BBN Butterfly Parallel Processor
William Celmaster, Bolt Baranek and Newman

66/A32
Scientific Computer Performance Evaluation: A Guide for the Perplexed
M. Edward Borasky, Floating Point Systems, Inc.

67/A32
Fast Banded Matrix Multiplication on Parallel Computers
Richard Bertram and Jerry F. Magnan, Florida State University

68/A32
Parallel and Vector Processing of the Incremental Dynamic Programming Algorithm
Daan Sandee, Control Data Corp. and Dennis Morrow, and Mohan Ramamurthy, Supercomputer Computations Research Institute

69/A33
Simulation of Transonic Flow on a Concurrent Computer
Mark E. Bassett and Christopher J. Catherasoo, AMETEK Computer Research Division

71/A33
Distributed Orthogonal Factorization: Givens and Householder Algorithms
Alex Pothen and Padma Raghavan, The Pennsylvania State University

72/A33
A Constraint Algorithm for Maintaining Rigid Bonds in Molecular Dynamics Simulations of Large Molecules
S. G. Lambrakos, E. S. Oran, J. P. Boris, and M. Nagumo
Naval Research Laboratory

75/A33
Pattern Recognition by Neural Networks on Hypercube
Alex W. Ho and Wojtek Furmanski, California Institute of Technology

77/A34
A Parallel Tridiagonal Equations Solver for the Butterfly Multiprocessor
Swarn P. Kumar, Boeing Computer Services

78/A34
Electromagnetic Scattering Analysis on the Hypercube
Jean E. Patterson, Ruel H. Calalo, Farzin Manshadi, Paulett C. Liewer, William A. Imbriale and James R. Lyons
Jet Propulsion Laboratory

79/A34
A Performance Comparison of Three Mini Supercomputers: The Alliant FX/8, The SCS-40, and The Convex C-1
Harvey J. Wasserman, Margaret L. Simmons, and Olaf M. Lubeck, Los Alamos National Laboratory

80/A34
Multitasking a Sparse Matrix Code on the Alliant FX/8
Iain S. Duff, Harwell Laboratory, England

81/A35
Parallel Triangular Matrix Inversion
David C. Fisher, Harvey Mudd College and Linda Sattler, The Claremont Graduate School

82/A35
Parallel Processing Bibliography (in Machine-Readable Form)
E. N. Miya, NASA Ames Research Center

85/A35
Analysis of Vectorisable Preconditioners for 3D Problems
Victor Eijkhout, University of Nijmegen, The Netherlands

86/A35
Decomposition and Solution of Transient Finite Element Problems on Hypercubes
Gregory A. Lyzenga, Arthur Raefsky, California Institute of Technology and Bahram Nour-Omid, Lockheed Palo Alto Research Laboratory

87/A36
Serial-to-Parallel Conversion of Real Time Continuous System Simulations
Alan M. Baum and Donald J. McMillan, General Motors Research Laboratories

88/A36
Design of a Real Time, Interactive Simulation Computer with a Parallel Processing Organization
Yukoh Kobayashi, California Research and Development

92/A36
Parallel Subspace Iteration for Solving the Generalized Eigenvalue Problem
Umesh Mahajan and Merrell Patrick, Duke University

93/A36
Performance of the NCUBE Hypercube
Geoffrey C. Fox, David W. Walker, California Institute of Technology and Gary R. Montry, Sandia National Laboratories

95/A36
Convergence of Certain Parallel Source Iteration Schemes in Particle Transport
Paul Nelson, Air Force Weapons Laboratory, and Texas A&M University, C. P. Katti, Jawaharlal Nehru University, India and Beny Neta, Naval Postgraduate School

96/A37
Study of Program Termination Using Z-Transform Theory
P. A. Venkatachalam, Asian Institute of Technology, Thailand and S. Arumugam, Computer Centre, GCT, India

97/A37
Parallel Distributed Processing Models for Natural Language Inference
Peter G. Tripodes, Los Angeles City College

100/A37
A Methodology for Dynamic Scheduling of Computation Tasks in Parallel Processing Systems
Ed P. Andert Jr., California State University, Fullerton

101/A37
Analysis of Fortran and C Performance on the FPS T Series Parallel Supercomputer
Thomas E. Bauer and Ian A. Taylor, Floating Point Systems, Inc.

102/A37
Application of Book Thickness of Graphs to Parallel Computing
Paul C. Kainen, The Analytic Sciences Corporation

106/A38
Parallel Optimization of Large-Scale Linear and Nonlinear Networks
Robert R. Meyer, University of Wisconsin-Madison

110/A38
Pipeline SOR: A Parallel Iterative Method for Shared Memory Machines
John P. Bonomo and Wayne R. Dyksen, Purdue University

111/A38
Optimization Strategies for Pattern Selection Problems and Their Implementation on the BBN Butterfly Parallel Processor
William Jeffrey, Robert Rosner, Robert Simon, Harvard-Smithsonian Center for Astrophysics, William Celmaster and Eric Tenenbaum, Bolt Beranek and Newman ACI

113/A38
Efficient Implementation of Parallel 3-D Seismic Algorithms on the CRAY X-MP
Mickey Edwards, Christopher C. Hsiung, Cray Research Inc., Dan D. Kosloff, Tel Aviv University, Israel and Moshe Reshef, Cray Research, Inc.

114/A39
Sparse Matrix Solution for Circuit Simulation on Multiprocessors
P. Sadayappan, Ohio State University and V. Visvanathan, AT&T Bell Laboratories

115/A39
Reliable Parallel Eliptic PDE Solution
Bruce M. McMillin and Lionel M. Ni, Michigan State University

116/A39
Efficient Matrix Multiplication on Hypercube Multiprocessors
Chung-Ta King and Lionel M. Ni, Michigan State University

117/A39
Parallel Simulated Annealing Optimization of Distributed Database Networks
G. S. Stiles, Allison Lee, Utah State University and Philippe Julien-Laferriere, Xerox Corporation, France

118/A40
Parallel Branch-and-Bound Algorithms on Hypercube Multiprocessors
Tarek S. Abdel-Rahman and Trevor N. Mudge, University of Michigan

119/A40
Parallel Programming Without Tears
Richard Foster, Chris Thomson and Dan Wilson, Myrias Research Corporation

L-4/A40
Multiplying Arbitrarily Shaped Matrices on a Data Parallel Computer
S. Lennart Johnsson, Alan Ruttenberg, Alex Vasilevsky, and Jill Mesirov, Thinking Machines Corporation

L-5/A40
Computing Fast Fourier Transforms on a Data Parallel Computer
S. Lennart Johnsson, Alan Ruttenberg, Alex Vasilevsky, and Guy Blelloch, Thinking Machines Corporation

Note: The numbers that follow the individual times for each presentation represent the numerical order and page number of the abstract.

L-7/A41
A Product Integration Method for Vortex Dynamics
Michael Shelley and Moshe Israeli, Princeton University

L-10/A41
A Parallel Algorithm for Matrix Inversion
Jerry F. Magnan, Florida State University

7:00 PM/Pasadena Exhibit Hall
Meeting of the SIAM Activity Group on Supercomputing/Cash Bar

Thursday, December 3/AM

8:30 AM/San Diego Room
Invited Presentations 9 and 10
Chair: Robert G. Voigt
ICASE

8:30
A Fast Multipole Algorithm for Large-Scale Particle Simulations
Leslie Greengard
Yale University

9:15
Lattice Gas Methods for Solving Partial Differential Equations
Gary Doolen
Los Alamos National Laboratory

10:00 AM/Pasadena Exhibit Hall
Coffee

10:40 AM
CONCURRENT SESSIONS

Thursday, December 3, 10:40 AM–12:00 Noon
Contributed Presentations 11/Santa Barbara A–C

Chair: Milo Dorr, Lawrence Livermore National Laboratory

10:40/15/A12
Large Scale FE Parallel Nonlinear Computations on High Performance Architectures
Charbel Farhat and Luis Crivelli
University of Colorado, Boulder

11:00/34/A12
A Family of Concurrent Algorithms for the Solution of Transient Finite Element Equations
B. Nour-Omid, Lockheed Palo Alto Research Laboratory
M. Ortiz, Brown University

11:20/103/A12
A Parallel Algorithm for Rapid Computation of Transient Fields
Robert J. Krueger
Iowa State University

11:40/169/A13
Analysis of a Parallelized Elliptic Solver for Reacting Flows
David E. Keyes and Mitchell D. Smooke
Yale University

Thursday, December 3, 10:40 AM–12:00 Noon
Contributed Presentations 12/San Diego Room

Chair: Eugene Brooks III, Lawrence Livermore National Laboratory

10:40/196/A13
Performance Analyses of the Nested-Dissection Algorithms for MIMD Multiprocessor Systems
Vijay K. Naik and Merrell L. Patrick
Institute for Computer Applications in Science and Engineering, NASA Langley Research Center and Duke University

11:00/190/A13
A Set of Level 3 Basic Linear Algebra Subprograms
Jack Dongarra, Argonne National Laboratory
Jeremy Du Croz, Numerical Algorithms Group Ltd., Iain Duff, Harwell Laboratory, Sven Hammarling, Numerical Algorithms Group Ltd.

11:20/56/A14
Gaussian Techniques on Shared Memory Multiprocessor Computers
Gregory A. Darmohray and Eugene D. Brooks, III
Lawrence Livermore National Laboratory

11:40/41/A14
LU Factorization on Distributed-Memory Multiprocessors
George A. Geist and Charles H. Romine
Oak Ridge National Laboratory

Thursday, December 3, 10:40 AM–12:00 Noon
Contributed Presentations 13/San Gabriel A–C

Chair: Raymond C. Y. Chin, Lawrence Livermore National Laboratory, and National Science Foundation

10:40/36/A14
Semi-Analytical Shape Functions in Parallel Computing Environment
J. A. Puckett and R. J. Schmidt
University of Wyoming

11:00/145/A14
The P-S Shape Functions in a Parallel Computing Environment
J. A. Puckett and R. J. Schmidt
University of Wyoming

11:20/90/A15
Invariant Imbedding and the Method of Lines for Multiprocessor Computers
Richard C. Allen, Jr., Lorraine S. Baca and David E. Womble
Sandia National Laboratories

11:40/128/A15
A Portable Parallel Algorithm for Multivariate Numerical Integration and Its Performance Analysis
E. deDoncker and John Kapenga
Western Michigan University

Thursday, December 3, 10:40 AM–12:00 Noon
Contributed Presentations 14/Sacramento Room

Chair: Richard Hickman, Lawrence Livermore National Laboratory

10:40/84/A15
Preconditioned Conjugate Gradient Methods for General Sparse Matrices on Shared Memory Machines
Edward Anderson and Youcef Saad
University of Illinois, Urbana-Champaign

11:00/52/A15
Parallel Multi-Level Finite Element Method with Hierarchical Basis Functions
G. Brussino, R. Herbin and V. Sonnad
IBM Corporation

11:20/48/A16
A Double Augmented Tree Architecture for Parallel Execution of Multigrid Algorithms
H. C. Wang and Kai Hwang
University of Southern California

11:40/38/A16
Implementation of a Parallel Multigrid Method on a Loosely Coupled Array of Processors
Raphaele Herbin, IBM Corporation
Stephane Gerbi, E.N.T.P.E., France
Vijay Sonnad, IBM Corporation

12:00 Noon
Lunch

Thursday, December 3/PM

1:30 PM/San Diego Room
Invited Presentations 11 and 12
Chair: Michael R. Raugh
NASA Ames Research Center

1:30
Strategies for the Mapping and Scheduling of Scientific Computations
Joel Saltz
Yale University

2:15
Graph Theory Plus Group Theory Equals Answers to Communication Problems
Vance Faber
Los Alamos National Laboratory

3:00 PM/Pasadena Exhibit Hall
Coffee

Thursday, December 3, 3:40–5:00 PM
Contributed Presentations 15/San Diego Room

Chair: Wlodek Proskurowski, University of Southern California

3:40/127/A16
SCHEDULE: An Environment for Developing Transportable Explicitly Parallel Codes in Fortran
J. J. Dongarra and D. C. Sorensen
Argonne National Laboratory

4:00/140/A17
Domain Decomposition Methods for Three-Dimensional Elliptic Problems
O. Axelsson
University of Nijmegen, The Netherlands

4:20/146/A17
Fast Poisson Solver on Irregular Regions by Boundary Integral-based Domain Decomposition
Daeshik Lee
University of Illinois, Urbana-Champaign

4:40/178/A17
A High-Performance FFT Algorithm for Vector Supercomputers
David H. Bailey
NASA Ames Research Center

Note: The numbers that follow the individual times for each presentation represent the numerical order and page number of the abstract.

FINAL PROGRAM

Thursday, December 3/PM
Continued

Thursday, December 3, 5:00–7:00 PM
Poster Session 3/Pasadena Exhibit Hall/Cash Bar

121/A41
Obtaining Nonuniform Systolic Designs Using Modified Mapping Techniques
Keith R. Allen and Roy P. Pargas, Clemson University

130/A42
A Parallel Processing Pre-Processor for the Fortran Language
Bruce W. Suter, Robert M. Hyatt and Joydeep Roy, University of Alabama at Birmingham

132/A42
Inhomogenous Parallel Supercomputers
Anthony E. Terrano, Rutgers University

133/A42
Fault-Tolerant Mapping of Large Scale Resource Allocation Algorithms onto Concurrent Processors
Rong T. Lee, Krishna R. Pattipati and Peter B. Luh, University of Connecticut

135/A42
Research Issues in 1000-Processor Linear Algebra Algorithms in Finite Element Analysis
Robert E. Benner and Gary R. Montry, Sandia National Laboratories

137/A43
Implementation of Iterative and Direct Solvers on The UNIX CRAY X-MP24
Michael A. Driscoll and Elissa C. Pariser, AT&T Bell Laboratories

138/A43
A Multiprocessor Model for a System Evaluation Tool
Kathleen M. Nichols, AT&T Bell Laboratories

139/A43
A Case Study in Scientific Parallel Processing: Prefix-Based Computation
Binay Sugla, AT&T Bell Laboratories

141/A43
The Effects of Parallel Processing on Matrix Computations
Laurie Melany Darmody, Laurel, MD

144/A44
The Behavior of Conjugate Gradient Methods on a Multi-Vector Processor with a Hierarchical Memory
William Jalby, Ulrike Meier and Ahmed Sameh, University of Illinois-Urbana-Champaign

147/A44
The Impact of Hierarchical Memory Systems on Linear Algebra Algorithm Design
Kyle Gallivan, William Jalby, Ulrike Meier and Ahmed Sameh, University of Illinois, Urbana-Champaign

148/A44
An Analysis of a Multivector Processor Implementation of Cyclic Reduction
Kyle Gallivan, Youcef Saad, University of Illinois at Urbana-Champaign and William Jolly, INRIA, France

149/A44
An Out-Of-Core Algorithm for Solving Dense Sets of Linear Equations on the FPS-164
Mark S. Woodyard, Martin Marietta Orlando Aerospace and Hassan N. Srinidhi, University of Central Florida

152/A45
Linear Permutations on a Multistage Interconnection Network
Anujan Varma, IBM Thomas J. Watson Research Center

156/A45
Solution of Elliptic Systems on Highly Parallel Computers
Paul O. Frederickson, Los Alamos National Laboratory

158/A45
Minimum Spanning Tree on The Hmesh Architecture
R. V. Boppana and C. S. Raghavendra, University of Southern California

159/A45
Comparison of Methods of Solving Symmetric Positive Definite Linear Systems on a Hypercube Concurrent Computer
Lois Mansfield, University of Virginia

161/A45
A Simulation Tool for Optimizing Production Programs
Steve Howard, Scientific Computer Systems Corporation

163/A48
Task Migration in Partitionable Parallel Processing Systems
Thomas Schwederski, Thomas L. Casavant, Purdue University and Howard Jay Siegel, Supercomputing Research Center

164/A46
Early Experience With the PASM System Prototype
Samuel A. Fineberg, Thomas L. Casavant and Thomas Schwederski, Purdue University

165/A46
Iterative Algorithms in a Data Driven Environment
P. Evripidou and J. L. Gaudiot, University of Southern California

166/A46
Computational and Reliability Measures of Homogeneous Multiprocessor Systems
Walid Najjar and Jean-Luc Gaudiot, University of Southern California

168/A46
Adaptive Grids of Multivariate, NTH-Order Calculi
Peter Nwoye O. Mbaeyi, West Germany

171/A47
Compiler-Drive Cache Management
Chi-Hung Chi and Henry Dietz, Purdue University

172/A47
CRegs: A Hardware Solution to the Alias Problem
Henry Dietz and Chi-Hung Chi, Purdue University

173/A47
Barrier MIMD: Beyond VLIW
Henry Dietz and Thomas Schwederski, Purdue University

174/A47
Efficient Iterative Solution of Elliptic PDE's
Fabian Waleffe, Massachusetts Institute of Technology and Thinking Machines Corporation

176/A47
A MIMD Eigenvalue Algorithm for a Parallel Transputer-Microcomputer System
Ina Vincent and Peter W. Aitchison, University of Manitoba, Canada

179/A48
Parallel Conjugate Gradients on a 1024-Node Hypercube
John L. Gustafson and Robert E. Benner, Sandia National Laboratories

183/A48
A Parallel Algorithm for Computing the Generalized Singular Value Decomposition (GSVD)
Zhaojun Bai, University of Maryland

185/A49
Finding Public Domain, Mathematical Software
Greg Astfalk, AT&T, Jack Dongarra, Argonne National Laboratory and Eric Grosse, AT&T Bell Laboratories

186/A49
The Lawrence Livermore Loops with a Vectorizing C Compiler
Greg Astfalk, AT&T and Dale Lancaster, Convex Computer Corp.

194/A49
A Graphical Programming Language for the Navier-Stokes Computer
Sherryl Tomboulian, Thomas W. Crockett and David Middleton, NASA Langley Research Center

195/A50
A Comparison of Different Parallel Programming Languages Using Choleski's Method of Multi-Processor Computer
Mark Jones, Duke University and Robert Voigt, NASA Langley Research Center

191/A51
Generalized SOR—New Spectra from Old
John de Pillis, Bergen Science Center, Norway

L-8/A50
Performance of Some CFD Codes on the Alliant FX/8
Walid Abu-Sufah and Amjad Daoud, Virginia Polytechnic Institute and State University

L-9/A50
Experiences with the Use of the Cray X-MP/48 Performance Monitoring Tools
Walid Abu-Sufah and Amjad Daoud, Virginia Polytechnic Institute and State University

L-11/A50
Combined Vector and Parallel Processing on the IBM 3090
John Tesch and Ken Jacobs, IBM, Dallas

L-12/A51
Applications of Level 2 BLAS in the NAG Library
Richard Brankin, Jeremy Du Croz, and Peter Mayes, The Numerical Algorithms Group Limited, Oxford, England

Note: The numbers that follow the individual times for each presentation represent the numerical order and page number of the abstract.

FINAL PROGRAM

Friday, December 4/AM

8:30 AM/San Diego Room
Invited Presentations 13 and 14
Chair: John Van Rosendale
Argonne National Laboratory

8:30
Parallel Processing Research at IBM
Tilak Agerwala
IBM T. J. Watson Research Center

9:15
Programming Environments for Multiprocessor Architectures
Piyush Mehrotra
Purdue University

10:00 AM/Pasadena Exhibit Hall
Coffee

10:40 AM
CONCURRENT SESSIONS

Friday, December 4, 10:40 AM–12:00 Noon
Contributed Presentations 16/San Gabriel A–C

Chair: Merrell Patrick, Duke University

10:40/153/A17
Parallel Vision Algorithms: An Approach
Sharat Chandran and Larry S. Davis
University of Maryland

11:00/154/A18
Performance of an Ocean Circulation Model on LCAP
Hsiao-Ming Hsu, Woods Hole Oceanographic Institution
Jih-Kwon Peir, IBM T. J. Watson Research Center, Dale B. Haidvogel, The Johns Hopkins University

11:20/28/A18
Ising Spin on a Shared Memory Machine: Computational Experience
James L. Blue and Francis Sullivan
National Bureau of Standards

11:40/189/A18
Parallel Iterative Algorithms To Solve The Discrete LQR Optimal Control Problem With Hard Control Bounds
Gerard G. L Meyer, The John Hopkins University
Louis J. Podrazik, Allied Signal, Inc.

Friday, December 4, 10:40 AM–12:00 Noon
Contributed Presentations 17/Sacramento Room

Chair: Mark K. Seager, Lawrence Livermore National Laboratory

10:40/175/A19
A Comparison of the Performance of LINPACK and Vectorized ICCG in Solving a Poisson Equation for a Problem of Two-Phase Flow in a Centrifuge
Eugene L. Poole, NASA Langley Research Center, Jeffery W. Frederick, University of Virginia

11:00/187/A19
On the Solution of Block Tridiagonal Systems of Equations on Multi-Vector Machines—A Comparative Study
Ken Dowers, S. Lakshmivarahan, Sudarshan K. Dhall and Julio Diaz, University of Oklahoma

11:20/112/A19
Linear Algebra at 30 MFLOPS on a 9 MFLOP Machine
Stewart F. Reddaway, ICL, England

11:40/134/A19
Performance of Blocked Gaussian Elimination on Multiprocessors
Elizabeth Jean O'Neil, BBN Advanced Computers, Inc., Henno Allik, BBN Laboratories Inc.

Friday, December 4, 10:40 AM–12:00 Noon
Contributed Presentations 18/Santa Barbara A–C

Chair: Scott Baden, University of California Berkeley

10:40/98/A20
Loop Blocking for Parallel Memory Optimization
Michael Wolfe, Kuck and Associates, Inc.

11:00/150/A20
Parallel FORTRAN: Why You Can't. How You Can.
Clifford Arnold, ETA Systems, Inc.

11:20/180/A20
Performance Based Distributed Programming
Phillip Q. Hwang, Naval Surface Weapons Center

11:40/20/A20
A Programming Aide for Message-passing Systems
Min-You Wu and Daniel D. Gajski
University of California, Irvine

Friday, December 4, 10:40 AM–12:00 Noon
Contributed Presentations 19/San Diego Room

Chair: William D. Gropp, Yale University

10:40/10/A21
A Gray-Code Scheme for Local Uniform Mesh Refinement on Hypercubes
William D. Gropp and Ilse C. F. Ipsen
Yale University

11:00/43/A21
Iteration with Mesh Refinement for Systems of ODE's
Olavi Nevanlinna
Helsinki University of Technology, Finland

11:20/108/A21
Parallelization of Adaptive Grid Domain Mappings
Calvin J. Ribbens, Virginia Polytechnic Institute and State University

11:40/25/A21
Automated Mesh Decomposition and Concurrent Finite Element Analysis for Hypercube Multiprocessor Computers
James G. Malone
General Motors Research Laboratories

1:00 PM
Conference Adjourns

Note: The numbers that follow the individual times for each presentation represent the numerical order and page number of the abstract.

ABSTRACTS: CONTRIBUTED PRESENTATIONS*

TUESDAY, DECEMBER 1
10:40 AM - 12:00 Noon
San Diego Room

Contributed Presentations 1

#46/10:40 AM

The Design and Implementation of Parallel Algorithms in Ada

The features of Ada that support parallel computing in mathematical software are covered and their advantages emphasized. The design and implementation of reusable parallel mathematical software components based on Ada's package, generic, data abstraction, and information hiding features are discussed. We also describe how to create an Ada shell that permits the use of existing Fortran software while maintaining the advantages of Ada. Mathematical software examples, primarily linear algebra, are presented that illustrate these Ada features.

Richard F. Sincovec
Computer Science Department
University of Colorado
Colorado Springs, CO 80933-7150

#123/11:00 AM

DINO -- A New Language for Numerical Computation on Local Memory Multiprocessors

We describe a new language for parallel processing that is specifically oriented towards writing numerical programs for local memory multiprocessors. DINO consists of extensions to standard C. The key new feature of DINO is the declaration by the user of a problem topology, which can be thought of as a virtual processor topology related to the natural physical or graphical topology of the problem. The topology is used in two additional DINO language extensions, composite procedure calls and distributed variables, that allow the execution of parallel procedures and communication between them without explicit communication statements. Experience using a prototype of DINO written in C++ will be described.

Matthew Rosing
Robert B. Schnabel
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

#160/11:20 AM

Scientific Parallel Processing with LGDF2

Large-Grain Data Flow was developed for writing portable, parallel scientific programs. The method uses a graphical language for specifying parallelism, combining dataflow-like programming constructs with blocks of ordinary sequential code. Macro expansion automatically generates source code containing efficient parallel scheduling mechanisms for a particular (parallel) processor. The original LGDF model was used successfully to implement several medium-sized programs on the Denelcor HEP. An improved version of the programming model (LGDF2) allows more efficient scheduling. Performance results for the Sequent Balance 21000 are given. Also, software tools are described that allow reliable construction, tuning, and debugging of parallel programs.

Robert G. Babb II
David C. DiNucci
Dept. of Computer Science and Engineering
Oregon Graduate Center
19600 NW Von Neumann Dr
Beaverton, OR 97006

---

* All contributed abstracts are numbered in the order that they were received at the SIAM office.

#157/11:40 AM

"The BF (Boundary-Fitted) Coordinate Transformation Technique of DEQSOL (Differential EQuation SOlver Language)"

DEQSOL is a high-level programming language specially designed to describe PDE problems in a way quite natural for numerical analyses. The DEQSOL translator automatically generates highly vectorizable simulation codes from high-level descriptions.
The BF facility has been developed in addition to the existing FDM and FEM facility. This facility automatically transforms any PDEs and regions defined in physical space into equivalent PDEs and rectangular regions in transformed space. This facility has made DEQSOL applicable to the development of simulators on curved and moving boundary regions, such as the impurity diffusion simulator in semiconductor devices.
Despite this extension, programming productivity is still an order of magnitude higher than FORTRAN programming, and generated codes achieve extremely high vectorization ratios(over 90%) on the Hitachi S-810 vector processor.

Chisato Konno, Michiru Yamabe, Miyuki Saji, Yukio Umetani
8th Dept., Central Research Laboratory
Hitachi Ltd.
Kokubunji, Tokyo 185, Japan

TUESDAY, DECEMBER 1
10:40 AM - 12:00 Noon
Sacramento Room

Contributed Presentations 2

#105/10:40 AM

Communication Complexity of Matrix Multiplication

Suppose a parallel processing machine multiplies two $n \times n$ matrices so only one copy of each input exists at the start of computations. We show that no matter what algorithm is used, it is possible to halve the machine so the amount of data exchanged between the halves is at least $(4n^4 - L^2)/16n^2$ where L is the maximum number of inputs entering one port. This supercedes a similar bound of Savage (Journal of Computer and System Science, **22**, 230-242 (1981)). This bound is used to find lower bounds on the time needed to do matrix multiplication on various parallel architectures.

David C. Fisher
Department of Mathematics
Harvey Mudd College
Claremont, CA 91711

#19/11:00 AM

ORDERINGS FOR PARALLEL SPARSE SYMMETRIC FACTORIZATION

We have developed a new heuristic for obtaining a nested dissection ordering, based on graph bisection algorithms. We use the Kernighan and Lin algorithm, as modified by Fiduccia and Matheyses, to partition the relevant section graphs. We present a new graph model so that the grid partitionings provide nodal separators, rather than the usual edge separators. We have refined the standard algorithm to further reduce the size of the separator set.

Our heuristic produces consistently and sometimes significantly better parallel orderings than the usual modifications of scalar orderings. In particular the depth of the elimination tree is reduced. THis is the appropriate measure of time for our target machine, the Connection Machine, and for other parallel machines. The orderings are also comparable in space requirements to the best scalar orderings.

Charles E. Leiserson*
Laboratory for Computer Science
NE43-321
Massachusetts Institute of Technology
Cambridge MA 02139

&

John G. Lewis+
Boeing Computer Services Company
Mail Stop 7L-21
PO Box 24346
Seattle WA 98214

* and Thinking Machines Corporation
+ and MIT Center for Advanced Engineering Study

#162/11:20 AM

An Efficient Fixed Size Array for Solving Large Scale Toeplitz Systems

In this paper systolic implementation and partitioning of Bareiss algorithm for solving an $(n+1)$ by $(n+1)$ Toeplitz system of equations is studied. While systolic solutions to Toeplitz systems has been well studied, the main problem with the known results is that they cannot solve any arbitrary Toeplitz system of equations on a fixed size array of processors. Our systolic implementation solves the Bareiss algorithm on a fixed size linear array. Our algorithm achieves $O(\frac{n^2 \log_2(p)}{p})$ computing time and $O(n^2)$ storage on p (p may be fixed) processors. Even without the partitioning, our implementation has superior time performance compared to known results within a constant factor. Furthermore, our implementation achieves a processor utilization of almost 100%. Our solution allows the real time solution of large Toeplitz matrices which occur in most Toeplitz applications.

Afshin Daghi
Systolic Solutions Inc.
V.K. Prasanna Kumar
Department of Electrical Engineering-Systems
University of Southern California
LA, California 90089-0781
Ali Safavi
USC Information Sciences Institute
and Computer Science Department

#188/11:40 AM

Architecture and Operation of a Systolic Sparse Matrix Engine

This paper presents the architecture and operation of the FEM, a systolic array tuned to sparse matrix operations. It enables high-performance computation of the preconditioned conjugate gradient method for solving the large, sparse, linear systems of equations common to applications such as finite element analysis. Matrix operations are typically very simple and regular and thus ideally suited for systolic computation. The desire was to design a robust system that could handle sparse matrices with a general sparsity structure. Dense matrices are treated as special cases of sparse matrices instead of the other way around which is typical of vector processors. The objective of this work is to provide an efficient systolic architecture for computing the operations of iterative techniques.

Robert J. Dunki-Jacobs, Steven W. Hammond,
Robert M. Hardy, and Terry M. Topka
GE Corporate Research and Development Center
P.O. Box 8
Schenectady, NY 12301
USA

TUESDAY, DECEMBER 1
10:40 AM - 12:00 Noon
San Gabriel Rooms A-C

Contributed Presentations 3

#170/10:40 AM

Accelerating with Rank-One Updates

Consider iterations for solving $Ax = b$ (A n by n nonsingular) which are based on splitting $A = M - N$. We discuss rank-one updates to improve inv(M) as an approximation to inv(A) during the iteration. The update kills and reduces singular values of N inv(M) and thus speeds up the convergence. The basic algorithm terminates after at most n sweeps, and if full n sweeps are needed, then inv(A) has been computed. Suitability for parallel processing and the effect of restricted memory length for the updates are also discussed.

Timo Eirola and Olavi Nevanlinna
Institute of Mathematics
Helsinki University of Technology
SF-02150 Espoo, Finland

#91/11:00 AM

A Parallel, Hybrid Algorithm for the Generalized Eigenproblem

We present a parallel algorithm for computing all eigenvalues, and their corresponding eigenvectors, in a specified interval for the generalized eigenproblem, $Ax = \lambda Bx$, where A and B are real, symmetric and B is positive definite. Eigenvalues are isolated, in parallel, using the Sturm sequence property of leading principal minors of $A - \mu B$. Concurrently, eigenvalues and eigenvectors are computed accurately using a combination of inverse and Rayleigh quotient iterations.

Results obtained from implementation of this algorithm on a shared memory MIMD architecture are presented. Factors which affect the efficiency of the algorithm are discussed.

Shing Ma
Merrell Patrick
Daniel Szyld

Department of Computer Science
Duke University
Durham, NC 27706

#143/11:20 AM

Multiprocessor Algorithm for the Symmetric Generalized Eigenvalue Problem

A trace minimization algorithm for computing a few of the smallest (or largest) eigenvalues and associated eigenvectors of the generalized eigenvalue problem $Ax = \lambda Bx$ is presented. Here we assume that the matrices A and B are symmetric of order n, with B being positive definite, and that both A and B are so large and sparse that a factorization of either matrix is impractical. In each iteration of this algorithm, first investigated by A. Sameh and J. Wisniewski, we are simultaneously approximating the p desired eigenpairs $p<<n$ by minimizing the trace of a $p \times p$ matrix subject to quadratic constraints. This talk presents an improved computational scheme which incorporates a variety of acceleration strategies, as well as demonstrates the suitability of the algorithm on a multiprocessor with two levels of parallelism, such as the Alliant FX/8

Bill Harrod, Ahmed Sameh and Mark Schaefer
University of Illinois at Urbana–Champaign
Center for Supercomputing R & D
305 Talbot Laboratory
104 South Wright Street
Urbana, Illinois 61801–2932
USA

#182/11:40 AM

Parallel QR Factorization on a Hypercube Multiprocessor

We describe a new algorithm for computing an orthogonal decomposition of a rectangular $m \times n$ matrix A on a hypercube multiprocessor. The algorithm uses Givens rotations, and requires the embedding of a two-dimensional grid on the hypercube network. We design a global communication scheme which uses redundant computation to maintain the data proximity, and we employ a mapping strategy for data allocation so that the processor idle time remains constant for a fixed number of processors regardless of the size of a square matrix. We describe how the algorithm is easily generalized to include the case when $m > n$. The proposed global communication scheme and the data mapping strategy result in reduced computation and communication cost compared to other known results for the same problem. Complexity results and numerical experiments on an Intel iPSC hypercube will be presented.

Eleanor Chu
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Alan George
Department of Computer Science
University of Tennessee
Knoxville, Tennessee 37996
and
Oak Ridge National Laboratory
Mathematical Sciences Section
Engineering Physics and Mathematics Division
P. O. Box Y, Bldg. 9207-A, MS-3
Oak Ridge, Tennessee 37831

TUESDAY, DECEMBER 1
10:40 AM - 12:00 Noon
Santa Barbara Rooms A-C

Contributed Presentations 4

#14/10:40 AM

Shock Calculations on Multi Processors

We present results from computing solutions to the compressible two dimensional Euler equations on a hypercube multi processor. We use modern high resolution finite difference schemes to get sharp shock profiles, mainly the Roe scheme and its generalizations to higher order accuracy. We discuss how to decompose the computational domain into different processors and we analyze if we can increase the computational speed by not doing communication on every time level. Some results from solving the same problems on the Connection Machine will also be presented.

Bjorn Sjogreen
Department of ScientificComputing
Uppsala University
Sturegatan 4B
S-752 23 Uppsala
SWEDEN

#126/11:00 AM

Parallel Processing of a Domain Decomposition Method

We present a parallel algorithm for the efficient solution of a time dependent singularly perturbed convection diffusion equation. The method is based upon domain decomposition that is dictated by singular perturbation analysis. A transformation is made to a coordinate system induced by the characteristics of the reduced hyperbolic equation. Asymptotic analysis is used to determine regions where certain reduced equations may be solved in place of the full equation. This reduces the number and size of the domains where the full equation must be solved. Treatment of the internal layer leads naturally to a shock detection technique appropriate for the iterative solution of nonlinear problems. Parallelism is evident at 3 levels. Domain decomposition provides parallelism at the highest level, and within each subdomain we exploit parallel-vector processing. Important features of the method include independent solution of the characteristic curves at the lowest level and low communication requirements between processes devoted to solving in the various domains. Tightly coupled processes are only required in domains where the full equation must be solved. We shall discuss the implementation and some aspects of the performance of this algorithm on existing parallel computers.

R.C.Y. Chin and G.W. Hedstrom
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94550

J.S. Scroggs
Center for Supercomputer Research and Development
University of Illinois at Urbana-Champaign
Urbana, IL 61801

D.C. Sorensen
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4844

#18/11:20 AM

Moving Point and Particle Methods for Convection-Diffusion Problems

We present several recently proposed numerical methods for the solution of multi-dimensional convection-diffusion problems. The common feature of all methods is the presence of a set of points which move according to the local fluid velocity field. However, the type of information that is associated with each of these points may be different. In addition, there may be a fixed mesh present, providing even greater variety of numerical method. In particular, we consider a moving point method employing a fixed grid and a grid-free particle method. The suitability of these algorithms to vector processors will be discussed.

Michael Rees
Numerical Analysis Group
Computing Laboratory
8-11 Keble Road
Oxford, United Kingdom

#177/11:40 AM

Random-Walk Simulation of Diffusion-Reaction-Convection

A simple stochastic model for populations of diffusing, interacting, and drifting particles is used to derive simulation algorithms which are closely related to random-walk methods and particle methods for the solution of partial differential equations of reaction-diffusion and fluid flow. These algorithms are then directly realized as simple vector or parallel programs. Examples are given which compare the resulting simulations with some standard numerical solutions of diffusion-reaction and diffusion-convection systems of partial differential equations.

D. J. Hebert
Department of Mathematics and Statistics
University of Pittsburgh
Pittsburgh, Pa. 15260

TUESDAY, DECEMBER 1

3:40 - 5:00 PM
San Diego Room

Contributed Presentations 5

#3/3:40 PM

Parallel Elliptic Solvers

We consider parallel versions of the standard fast elliptic solvers and discuss their merits and disadvantages. In particular, we present an adaptation of the Block Cyclic Reduction (BCR) algorithm to vector and parallel machines. The main bottleneck of BCR lies in the solution of linear systems whose coefficient matrix is the product of tridiagonal matrices. This bottleneck is handled by expressing the rational function corresponding to the inverse of this product as a sum of elementary fractions. This leads to parallel solutions of tridiagonal systems. FFT based algorithms are also discussed. We discuss implementations of these algorithms on shared memory and distributed memory machines.

E. Gallopoulos
Y. Saad
University of Illinois at Urbana-Champaign
Center for Supercomputing R & D
305 Talbot Laboratory
104 South Wright Street
Urbana, Illinois 61801-2932
USA

#17/4:00 PM

The Parallel Waveform Relaxation Multigrid Method

A great deal of attention has recently been given to the development of the highly parallel Waveform Relaxation method (WR) for solving very large systems of ordinary differential equations. Attempts to use WR to solve the equations arising from the numerical method of lines have not been very successful due to the slow convergence of the normally used Jacobi, Gauss-Seidel and SOR methods.

In this paper we present a new algorithm that combines the parallel nature of WR with the fast convergence of multigrid to solve time dependent partial differential equations.

We also discuss its implementation on the Caltech Hypercube.

Vandewalle Stefan
Katholieke Universiteit Leuven
dept computerscience
Celestijnenlaan 200A
B - 3030 Heverlee, Belgium

#31/4:20 PM

A Pipelined Block QR Algorithm for a Ring of Vector Processors

We present a parallel algorithm for computing the QR factorization of a dense matrix on a ring of vector processors. Regarding the matrix A as a matrix of block columns, we use the WY representation (C. Bischof and C. Van Loan, "The WY Representation for Products of Householder Matrices" SIAM J. Scientific and Statistical Computing, Vol. 8, No. 1, 1987) to bundle Householder updates. This strategy results in an algorithm that is rich in matrix-matrix multiplication, the operation of choice for pipelined vector units. Here we show how this technique can be used in the coarse-grained parallel setting. We distribute block columns across the ring of processors and stagger the block column updates so that the computation can proceed in a pipelined fashion. We have performed experiments on the LCAP system at IBM Kingston which show that this method is indeed well suited for coarse-grained distributed systems.

Christian H. Bischof
Dept. of Computer Science
5133 Upson Hall
Cornell University
Ithaca, NY 14853

#32/4:40 PM

CONJUGATE GRADIENT PRECONDITIONERS FOR PARALLEL COMPUTERS

In this paper, we show how to derive efficient preconditionners for the conjugate gradient method on parallel computers with a large number of processors. These preconditioners use domain decomposition techniques, the domain being divided either in strips or in boxes. Numerical experiments will show that for a given problem dimension, the increase in the number of iterations as a function of the number of processors, is very small, making these preconditioners very attractive for future parallel computers.

Gérard MEURANT
Département de Mathématiques Appliquées
CEL-V - B.P. N° 27
94190 - VILLENEUVE St GEORGES
FRANCE

WEDNESDAY, DECEMBER 2

10:40 AM - 12:00 Noon
Santa Barbara Rooms A-C

Contributed Presentations 6

#129/10:40 AM

Concurrent Management of Priority Queues for Adaptive Algorithms

A study of the demands of several types of parallel numerical algorithms, which are synchronized by a task pool, is presented. This is used to evaluate different methods of concurrent priority queue management on various shared memory architectures. Both an analytic model and empirical results from the Denelecor HEP, Sequent Balance, Alliant FX-8 and Encore Multimax are considered.

This work is part of our effort to develop a set of macros, layered over the Argonne macro package, which provide a means of producing portable programs for adaptive partitioning problems. Portable efficient concurrent priority queue management is critical to this endeavor.

John Kapenga
Elise deDoncker-Kapenga
Department of Computer Science
Western Michigan University
Kalamazoo, MI 49008-5021

#83/11:00 AM

Programming Abstractions for Run-Time Partitioning of Scientific Continuum Calculations Running on Multiprocessors

I will discuss a set of software abstractions for dynamically partitioning various math-physics calculations on a team of processors. I tried out the abstractions on Anderson's Method of Local Corrections, a type of vortex method for computational fluid dynamics. I ran experiments on 32 processors of the Intel iPSC -- a message-passing hypercube architecture -- and on 4 processors of a Cray X-MP -- a shared-memory vector architecture -- and achieved good parallel speedups of 24 and 3.6, respectively. The abstractions should apply to diverse applications, including finite difference methods, and to diverse architectures without requiring that the application be reprogrammed extensively for each new architecture.

Scott B. Baden
Mathematics Department
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

#64/11:20 AM

Implementation of the Acceptance-Rejection Method on Parallel Processors: A Case Study in Scheduling

The Acceptance-Rejection method (AR) is often used in connection with the generation of nonuniform random distributions. We examine the problem of generating, via AR, vectors of nonuniform numbers on various kinds of vector and parallel processors. Vector and SIMD implementations are generally quite inefficient compared to MIMD implementations. Various scheduling models are presented and the theoretical predictions are verified by tests done on a BBN Butterfly Parallel Processor.

William Celmaster

Bolt Beranek and Newman
Advanced Computers Inc.
10 Fawcett Street, Cambridge, MA 02238

#74/11:40 AM

On the Placement of Parallel Processes

An important problem of parallel computation is to determine the best placement in a multiprocessor system of each of a set of related processes. An earlier approach to this problem attempted to maximize the number of communicating processes which lie on adjacent processors. We demonstrate that this approach is not suitable for large problems through a comparison of mappings generated by this approach with random mappings and with mappings generated by a heuristic which attempts to minimize the distance from each process to all processes with which it communicates. The various approaches are experimentally compared with respect to a specific problem.

Michael R. Leuze and Stephen R. Schach
Computer Science Department
Vanderbilt University
Box 77 Station B
Nashville, TN 37235

WEDNESDAY, DECEMBER 2

10:40 AM - 12:00 Noon
San Gabriel Rooms A-C

Contributed Presentations 7

#104/10:40 AM

Mapping Large Scale Computational Problems on a Highly Parallel SIMD Computer

The mapping strategies employed to solve various size computational problems on an SIMD computer with p = 1000 or more processors are described. For 'small' problems, whose size n << p, many problems can be solved simultaneously, and when n is approximately equal to p, straightforward 'direct' mappings can be used. The most interesting, and difficult, situation arises for large problems, n >> p where various techniques have been applied to a wide variety of computational problems. These include sheet mapping, crinkled mapping, linear array, multi-serial, parallel data transforms, permutations or a combination of mappings. The problem of changing mappings is also considered. The application of these strategies to a number of problems and their implementation on the various DAP systems will be described

Heather M. Liddell
Dennis Parkinson
Department of Computer Science & Statistics
Queen Mary College
University of London
Mile End Road
London
E1 4NS

#24/11:00 AM

Multiprogramming and the Performance of Parallel Programs

Tight synchronization in parallel programs executed on multiprogrammed multiprocessors may result in catastrophic performance losses as a result of the absence of swapped out processes. Our work introduces a programming methodology that utilizes computational synchronization and avoids tight synchronization in control flow parallel programs. In this methodology, each phase of the computation is assigned a status that can be ready, blocked or completed, and tasks in each computational phase are selfscheduled to ensure computational progress by the available executing processes. Results obtained indicate that this methodology avoids the catastrophic performance losses resulting from the swapping of processes in multiprogrammed multiprocessors.

Muhammad S. Benten
Harry F. Jordan
Electrical and Computer Engineering Department
University of Colorado
Boulder, CO 80309-0425

#122/11:20 AM

Using Mathematical Modeling to Aid in Parallel Algorithm Development

The intent of this presentation is to demonstrate the advantages of using mathematical modeling in conjunction with parallel computer implementation in designing and understanding parallel algorithms. We describe the formation of a model that accurately matched execution times of a rather complex parallel global optimization method on an Intel hypercube. We then discuss the application of this model to detect weakness in the algorithm and to analyze several possible improvements to it. We believe that this example helps to indicate that the use of mathematical modeling together with computer implementation is useful in making parallel algorithm development more efficient in human and computer time.

Elizabeth Eskow
Robert B. Schnabel
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

#4/11:40 AM

The Cerberus Multiprocessor Simulator

We describe a simulation facility for scalable shared memory multiprocessors. The processors are a RISC architecture with a minimum of additional instructions added to support synchronization. The functional units of the processors, including access to the shared memory, are fully pipelined. The multiprocessor is simulated at a very detailed level, with pipeline delays and conflicts in the packet switched shared memory server being accurately accounted for. Applications are written in C. The simulator can be used to examine algorithm performance as the number of processors in the multiprocessor is scaled, and accurately measures the effects of pipeline delays and conflicts in the shared memory subsystem.

Eugene D. Brooks, III, Timothy S. Axelrod,
Gregory A. Darmohray

Computational Physics Division
University of California,
Lawrence Livermore National Laboratory
P. O. Box 808
Livermore, California 94550

WEDNESDAY, DECEMBER 2

10:40 AM - 12:00 Noon
San Diego Room

Contributed Presentations 8

#109/10:40 AM

A New Parallel Algorithm for Linear Triangular Systems

A parallel algorithm for solving linear triangular system is presented and analyzed. This algorithm has a small start up phase which is different from the rest of the parallel phases so that the first processor will be able to accumulate enough results before shipping them to the all other processors. In the rest of the parallel phases, the first processor makes a substitution of the results of the rest of the processors from the previous phase and then makes an elemination of more variables. In the same time the rest of the processors substitute the results of the first processor from the previous phase. The most important features of this algorithm are: (1) It works well for any number of processors. (2) It achieves the optimal speed up, optimal efficiency and very low communication complexity. (3) It can be used in both distributed parallel computing environment and tightly coupled parallel computing system. (4) This algorithm can be mapped easily onto any parallel architecture without any programming difficulties.

Avi Lin
He Zhang
Department of Mathematics
Temple University
Philadelphia, PA 19122

#142/11:00 AM
A Parallel Algorithm for the Singular Value Decomposition of Rectangular Matrices

The classical sequential algorithm for computing the singular value decomposition (SVD) of a matrix has time complexity $O(n^3)$ and therefore can only be used on small matrices. Parallel algorithms for computing the SVD have recently been proposed. The best of these use Jacobi rotations to diagonalize the matrix. However, Jacobi-SVD algorithms require that the matrix be square, and, when this is not the case, the matrix must be preprocessed. A parallel algorithm is presented that computes the SVD of a rectangular matrix without preprocessing. The running time and numerical stability of this algorithm is compared against a selection of other parallel SVD algorithms.

William D. Shoaff and David C. Chan
Mathematical and Computer Sciences
Florida Institute of Technology
150 W. University Boulevard
Melbourne, FL 32901-6988

#125/11:20 AM
A Parallel Algorithms for the Singular Value Decomposition

A parallel algorithm for computing the singular value decomposition will be presented. The basic step of the algorithm is a divide and conquer step based upon a rank one modification of a bidiagonal matrix. Numerical difficulties associated with forming the product of a matrix with its transpose are avoided and numerically stable formulae for obtaining left singular vectors after computing updated right singular vectors are derived. A deflation technique is described which together with a robust root finding method will assure computation of the singular values to full accuracy in the residuals and will also assure the orthogonality of the singular vectors.

E.R. Jessup
Department of Computer Science
Yale University
New Haven, CT 06520

D.C. Sorensen
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439-4844

#89/11:40 AM
A Multiprocessor Scheme for Singular Value Decomposition

We present a multiprocessor scheme for determining the singular value decomposition of rectangular matrices in which the number of rows is substantially larger or smaller than the number of columns. In this scheme, we perform an initial **QR** factorization on the tall matrix (either A or A^T) using a multiprocessor block Householder algorithm. We then apply a one-sided Jacobi multiprocessor method on the resulting upper triangular **R** to yield $RV = U\Sigma$, from which the the desired singular value decomposition is obtained. Preliminary experiments on an Alliant FX/8 computer system with *8* processors indicate speedups near *5* for our scheme over an optimized implementation of the Linpack/Eispack routines which perform the classical bi-diagonalization technique. Our scheme performs exceptionally well for rank deficient matrices as well as for those rectangular matrices having clustered or multiple singular values, and may be well suited for applications such as real-time signal processing. We present performance results on the Alliant FX/8 and Cray X-MP/48 computer systems with particular emphasis on speed-ups obtained for our scheme over classical SVD algorithms.

Michael Berry
and
Ahmed Sameh

Center for Supercomputing Research and Development
University of Illinois at Urbana-Champaign
305 Talbot Lab
104 S. Wright St.
Urbana, IL 61801-2932

WEDNESDAY, DECEMBER 2

10:40 AM - 12:00 Noon
Sacramento Room

Contributed Presentations 9

#35/10:40 AM
A Parallel Homotopy Method for Solving a System of Polynomials

The homotopy method is theoretically able to find all the roots of a system of polynomials. In its numerical implementation, each root is obtained by following a distinct homotopy curve. Tracking the curve very closely may guarantee the finding of all roots, but computationally expensive. A loose following of the curve can expedite the computation, but some roots will be missing due to the merge of curves and the undesirable divergence may occur. Our proposed method can expedite the computation process and can automatically detect and correct curve merging. Potential parallelism of the proposed method is fully exploited to allow the implementation in parallel processors.

Lionel M. Ni
Department of Computer Science

Shui-Nee Chow and Yün-Qiu Shen
Department of Mathematics
Michigan State University
East Lansing, Michigan 48824

#94/11:00 AM

The Granularity of Homotopy Algorithms for Polynomial Systems of Equations

Polynomial systems consist of n polynomial functions in n variables, with real or complex coefficients. Finding zeros of such systems is challenging because there may be a large number of solutions, and Newton-type methods can rarely be guaranteed to find the complete solution list. There are homotopy algorithms for polynomial systems of equations that are globally convergent from an arbitrary starting point with probability one, are guaranteed to find all the solutions, and are robust, accurate, and reasonably efficient. There is inherent parallelism at several levels in these algorithms. Several parallel homotopy algorithms with different granularities are studied on several different parallel machines, using actual industrial problems from chemical engineering and solid modelling.

L. T. Watson
J. Harimoto
J. P. Bixler

Department of Computer Science
Virginia Polytechnic Institute & State University
Blacksburg, VA 24061

#167/11:20 AM

A Parallel Nonlinear Integer Programming Algorithm Based on Branch and Bound and Simulated Annealing

In this talk, we shall present a model problem which we wish to solve, a sketch of standard branch and bound techniques applicable to the solution of the problem, a short outline of the idea behind simulated annealing, and lastly an upper level description of an algorithm which uses a parallel branch and bound technique simultaneously with a modified simulated annealing process. The two techniques, operating in parallel, and "weakly" connected, complement each other in efficiency and determinacy. The parallel algorithm is being implemented on a multiprocessor transputer network system operating under an OCCAM shell. We will present comparisons made with standard sequential and parallel branch and bound algorithms, and with sequential and parallel simulated annealing algorithms.

Ken W. Bosworth
Mathematics Department
Utah State University
Logan, Utah 84322-3900

G. S. Stiles
Electrical Engineering Department
Utah State University
Logan, Utah 84322-4120

#131/11:40 AM

Recursive Binary Partitions

A Recursive Binary Partition (RBP) is a partition of space or data that is obtained from a single recursive partitioning scheme. The generic scheme makes recursive calls to a splitting function. By varying the definition of the splitting function and the depth of the recursion, one can obtain many of the currently popular partitioning strategies that are being used for vortex methods, domain decomposition and multiobject tracking. The recursive nature of the scheme allows all RBP's to be easily computed and maintained on parallel architectures with binary recursive structures such as hypercubes, the Connection Machine, RP3 and the Butterfly.

George Cybenko
Department of Computer Science
Tufts University
Medford, MA 02155

Tom Allen
Alphatech, Inc.
111 Middlesex Turnpike
Burlington, MA 01803

WEDNESDAY, DECEMBER 2

3:40 - 5:00 PM
San Diego Room

Contributed Presentations 10

#49/3:40 PM
Parallel Architectures for Optical Computing

Recent advances in optical devices, in conjunction with the inherent parallelism in optics have created the hope of designing a computer that can achieve tremendous speed improvement over existing electronic computers. In this paper, we present a massively parallel architecture for optical computing. The system is very well suited for applications that are 2-D in nature such as signal and image processing. Next we show how parallel algorithms can be mapped very efficiently onto the system. Then we present the optical implementation of different units comprising the system. Then we discuss its programmability, and predict its performance, as compared with conventional computers. The key message of this article is that optical computers can be potentially 10^3 to 10^5 faster than today's computers.

Kai Hwang and Ahmed Louri
Computer Research Institute
University of Southern California
University Park, SAL 324B
Los Angeles, CA 90089-0781

#76/4:00 PM
Synchronization of Nonhomogeneous Parallel Computations

We investigate the effect of non deterministic execution times upon the "speed-up" factor in the parallel execution of algorithms. This investigation has been triggered by the observations in the modeling of domain splitting algorithm for the numerical solution of partial differential equations. While the modeling of an unsynchronized execution leads to a very high processor utilization, requiring synchronization conditions results in a considerably lower level of performance. In this paper we show that this effect is quite general for nonhomogeneous parallel computations. It is an algorithmic effect independent of hardware and we give analytic results for a variety of nonhomogeneous computations.

Dan C. Marinescu
John R. Rice
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

#120/4:20 PM
Intrinsically Parallel Algorithms

Most algorithms implemented on parallel computers have been optimal serial algorithms, slightly modified or parallelized. An exciting possibility is the search for intrinsically parallel algorithms. These are algorithms which do not have a sensible serial equivalent - any serial equivalent is so inefficient as to of little use.

We describe a multiscale algorithm for the solution of PDE systems that is designed specifically for massively parallel super-computers. Unlike conventional multigrid algorithms, the new algorithm utilizes the same number of processors at all times. Convergence rates are much faster than for standard multigrid methods - the solution error decreases by up to three digits per iteration.

We present results of the implementation of the algorithm on the 65,536 processor Conection Machine, and we compare the results with more standard algorithms implemented on that computer.

Oliver A. McBryan
Dept. of Computer Science,
University of Colorado,
Boulder, CO 80309-0430.

#124/4:40 PM
A Software Tool For Building Supercomputer Applications

In this paper we describe a programming tool designed to help users of parallel supercomputers retarget and optimize application codes. The system is an interactive program editing and transformation system that helps the user with this task. Each program that enters the system is completely parsed and all data dependences are recorded. The user then works with the system to restructure his code to a form suitable for a given target architecture. At any time the user can ask the system to make estimates of program performance based on potential memory hierarchy conflicts or vector instruction utilization. The system will generate code for the BBN butterfly or Alliant FX/8.

Dennis Gannon
Daya Atapattu
Mann Ho Lee
Bruce Shei

Department of Computer Science
Indiana University
Bloomington, Indiana 47405

THURSDAY, DECEMBER 3

10:40 AM - 12:00 Noon
Santa Barbara Rooms A-C

Contributed Presentations 11

#15/10:40 AM

Large Scale FE Parallel Nonlinear Computations on High Performance Architectures

Here we revisit Finite Element algorithms based on homotopy equations, for implementation on shared memory and local memory multiprocessors, which represent both extremes of today's high performance architectures. To achieve this goal, two non-numerical algorithms for automatic domain decomposition are developed first. Then, a computational strategy that ties these with a numerical nonlinear algorithm based on homotopy equations is presented and discussed. Its implementation on parallel architectures features two levels of parallelism, namely, concurrency and vectorization. It requires little storage and minimizes synchronization and/or interprocessor communication. Numerical experiments conducted on a variety of multiprocessors (CRAY XMP-4, Alliant FX/8, Encore Multimax, Sequent Balance-8000, iPSC-d7) validate the computational strategy and assess its performance.

Charbel Farhat
Department of Aerospace Engineering Sciences and Center for Space Structures and Controls
University of Colorado at Boulder - Campus Box 429
Boulder, CO 80309-0429

Luis Crivelli
Department of Mechanical Engineering
University of Colorado at Boulder - Campus Box 427
Boulder, CO 80309-0427

#34/11:00 AM

A Family of Concurrent Algorithms for the Solution of Transient Finite Element Equations

In this paper, we describe a solution method for problems in structural dynamics. This method makes effective use of the architecture of concurrent computers. The algorithm is based on partitioning the structure into substructures. Each substructure is then processed over a time step independently of the others. Thus, high levels of concurrency can be achieved at this phase of the analysis. The solution of the complete system is constructed by gluing the solution for the substructures. This involves averaging the solution at the interfaces between the subsystems. This averaging scheme is derived by enforcing the consistency condition on the underlying algorithm. The resulting two parameter algorithm is shown to be unconditionally stable.

B. Nour-Omid
Mechanics and Materials Engineering Laboratory,
Lockheed Palo Alto Research Laboratory,
3251 Hanover Street,
Palo Alto, CA 94304.
and
M. Ortiz
Division of Engineering,
Brown University,
Providence, RI 02912.

#103/11:20 AM

A Parallel Algorithm for Rapid Computation of Transient Fields

The problem considered here is that of determining the internal electromagnetic field produced by an arbitrary incident pulse impinging on a one-dimensional medium with spatially varying permittivity and conductivity profiles. This is done by introducing a special set of Green's functions for the problem. These functions have the property that they can be computed for all time (at a fixed spatial location) via an algorithm which uses as input data the values of the functions for a fixed initial period of time. Although this technique is more computationally intensive than standard methods for computing Green's functions, it is ideally suited for a parallel environment because there is no sequential code involved. Examples comparing this method versus standard techniques are presented.

Robert J. Krueger
Applied Mathematical Sciences
Ames Laboratory
Iowa State University
Ames, Iowa 50011

#169/11:40 AM

Analysis of a Parallelized Elliptic Solver for Reacting Flows

A parallelized finite difference code for systems of nonlinear elliptic boundary value problems in two dimensions, based on Newton's method, is analyzed in terms of computational complexity and parallel efficiency. An approximate cost function depending on 15 dimensionless parameters (including discrete problem dimensions, convergence parameters, and machine characteristics) is derived for algorithms based on stripwise and boxwise decompositions of the domain and a one-to-one assignment of the strip or box subdomains to processors. The sensitivity of the cost function to the parameters is explored throughout a region of parameter space corresponding to a coupled system of nineteen equations with very expensive function evaluations (a detailed-kinetics diffusion flame). The algorithm has been implemented on parallel hardware and some experimental results for a flamesheet calculation with stripwise decompositions are presented and compared with the theory.

David E. Keyes & Mitchell D. Smooke
Department of Mechanical Engineering
Yale University
P.O. Box 2159 Yale Station
New Haven, CT 06520

THURSDAY, DECEMBER 3

10:40 AM - 12:00 Noon
San Diego Room

Contributed Presentations 12

#196/10:40 AM

Performance Analyses of the Nested-Dissection Algorithms for MIMD Multiprocessor Systems

Sequential algorithms based on the idea of nested dissection orderings are known to be effective in solving sparse positive definite linear systems which usually arise from grid problems. Recently, several attempts have been made in developing efficient parallel algorithms for solving such sparse linear systems. Most of these parallel algorithms assume mesh-connected, systolic type architectures or massively parallel systems. In this paper we consider various performance issues related to implementing the nested-dissection algorithms on some of the currently available MIMD architectures. Analytic models are developed and are verified by implementing the methods on a 64-node BBN Butterfly multiprocessor system.

Vijay K. Naik
Merrell L. Patrick

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA 23665

and

Computer Science Department
Duke University
Durham, NC 27706

#190/11:00 AM

A Proposal for a Set of Level 3 Basic Linear Algebra Subprograms

This Presentation describes a proposal for a set of Level 3 Basic Linear Algebra Subprograms (Level 3 BLAS), which are targeted at matrix-matrix operations, with the aim of providing more efficient, portable implementations of algorithms on high-performance computers, especially those with hierarchical memory and parallel processing capability.

Jack Dongarra
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, Illinois 60439-4844
U.S.A.

Jeremy Du Croz; Sven Hammarling
Numerical Algorithms Group Ltd
NAG Central Office, Mayfield House
256 Banbury Road
Oxford OX2 7DE
U.K.

Iain Duff
Computer Science and Systems Division
Harwell Laboratory
Oxfordshire OX11 ORA
U.K.

#56/11:20 AM
Gaussian Techniques on Shared Memory Multiprocessor Computers

Implementations of Gaussian elimination and Gauss–Jordan algorithms for a shared memory multiprocessor simulator are presented. Gaussian elimination algorithms require processor synchronization due to data dependencies. We show the necessity for a hardware implementation of processor synchronization. The efficiency of Gauss–Jordan versus Gaussian elimination with software and hardware processor barriers is also given. Even though Gauss–Jordan requires more operations than Gaussian elimination for a given size problem, Gauss–Jordan is faster for certain matrix sizes and numbers of processors. The cross-over points are shown for both hardware and software barriers. The algorithms are written in C and are run on the Cerberus Multiprocessor Simulator, a simulation facility for scalable shared memory multiprocessors with pipelined functional units.

Gregory A. Darmohray, Eugene D. Brooks III

Computational Physics Division
University of California
Lawrence Livermore National Laboratory
P.O. Box 808, L-298
Livermore, CA 94550

#41/11:40 AM
LU Factorization on Distributed–Memory Multiprocessors

We discuss and compare two methods of improving the efficiency of LU factorization with pivoting on distributed-memory multiprocessors. The first method uses a dynamic load balancing scheme to avoid work imbalances caused by pivoting. While this method increases the amount of communication required, it significantly reduces the execution time. The second method uses pipelining to mask the cost of pivoting. While pipelining complicates the algorithm, the execution time is slightly less than in the first method. Both algorithms have achieved efficiences of 85% to 95% running on a 64 node iPSC hypercube.

George A. Geist
Charles H. Romine

Mathematical Sciences Section
Engineering Physics and Mathematics Division
Oak Ridge National Laboratory
P. O. Box Y, Bldg. 9207-A, MS-3
Oak Ridge, Tennessee 37831

THURSDAY, DECEMBER 3

10:40 AM - 12:00 Noon
San Gabriel Rooms A-C

Contributed Presentations 13

#36/10:40 AM
Semi-Analytical Shape Functions in a Parallel Computing Environment

Semi-analytical shape functions have been used successfully to model many physical systems in continuum mechanics. The orthogonality of appropriately selected shape functions allows a continuous system to be effectively transformed into a series of smaller discrete subsystems that can be solved independently. The uncoupling of the subsystems produces a highly parallel algorithm that is well suited to either a shared or a distributed memory environment. The semi-analytical shape functions are discussed with regard to the algorithmic implications. Examples are presented to illustrate the methodology.

J. A. Puckett
R. J. Schmidt
Institute for Scientific Computation
University of Wyoming
Laramie, Wyoming 82071

#145/11:00 AM
The P-S Shape Functions in a Parallel Computing Environment

A shape function which combines polynomial and sine series functions (P-S Shape Functions) is used to develop an efficient finite element algorithm for the solution of the elliptic equations that model many phenomena in continuum mechanics. This nontraditional approach results in an algorithm that is massively parallel, adaptive, and easily implemented using standard numerical techniques. The key to these favorable features is the orthogonality of the sine series which uncouples element matrices associated with the series terms. Because of the special structure of the system equation, condensation methods are used to form a series of matrix equations that are much smaller than the original system equation. These equations can be triangularized independently and thus the algorithm is highly parallel.

J. A. Puckett
R. J. Schmidt
Department of Civil Engineering
University of Wyoming
Laramie, Wyoming 82071

#90/11:20 AM
Invariant Imbedding and the Method of Lines for Multiprocessor Computers

Recursive relations have been used to allow the solution of invariant imbedding equations with singularities. We demonstrate that these same relations can be used in an efficient implementation of invariant imbedding for multiprocessor computers and investigate the maximum attainable speedup.

The parallel implementation of imbedding can be used in conjunction with the method of lines to solve partial differential equations. We consider the problem of assigning lines to processors to minimize communication delays. We also consider the question of convergence of the resulting asynchronous algorithm. Timing and speedup data are presented for algorithms implemented on the NCUBE hypercube.

Richard C. Allen, Jr.
Lorraine S. Baca
David E. Womble

Sandia National Laboratories
Albuquerque, NM 87185

#128/11:40 AM
A Portable Parallel Algorithm for Multivariate Numerical Integration and Its Performance Analysis

An adaptive algorithm for numerical integration over an N-dimensional cube will be discussed, together with its implementation in a portable manner on a range of MIMD shared memory architectures.

Results from different machines will be given, showing good speedups especially in higher dimensions, where a large granularity is in effect.

We shall investigate the influence of various elements of the algorithm on the performance. A technique for a-priori performance analysis, on the basis of the serial code, will be applied and justified at the hand of performance results on the Sequent Balance, Alliant FX-8 and Encore Multimax.

E. deDoncker
John Kapenga
Department of Computer Science
Western Michigan University
Kalamazoo, MI 49008-5021

THURSDAY, DECEMBER 3

10:40 AM - 12:00 Noon
Sacramento Room

Contributed Presentations 14

#84/10:40 AM
Preconditioned Conjugate Gradient Methods for General Sparse Matrices on Shared Memory Machines

For very large general sparse linear systems such as those arising from three-dimensional models, preconditioned Krylov subspce methods represent a good alternative to using direct methods. When implementing these methods on parallel machines one faces the problem that the standard preconditioners are very sequential and become a serious bottleneck. The first remedy is to use a polynomial preconditioning which consists of approximating A^{-1} by a polynomial in A. This preconditioning technique offers a high degree of parallelism and its performance can be enhanced by a simple diagonal or block-diagonal scaling. An alternative is to parallelize the standard Incomplete LU (ILU) preconditioners by performing the incomplete factorization in a rank-one modification manner instead of computing one row of L and U at a time as is usually done. This implies manipulating the row and column structures of A but the resulting overhead is outweighed by the benefits of parallelism. Numerical experiments will be presented comparing these approaches.

Edward Anderson
Youcef Saad
University of Illinois at Urbana-Champaign
Center for Supercomputing R & D
305 Talbot Laboratory
104 South Wright Street
Urbana, Illinois 61801-2932
USA

#52/11:00 AM
PARALLEL MULTI-LEVEL FINITE ELEMENT METHOD WITH HIERARCHICAL BASIS FUNCTIONS

The p-version of the finite element method using a hierarchy of basis functions is employed in a multi-level computational scheme closely resembling the multi-grid approach. The advantage is that it is possible to preserve the speed of multi-grid techniques while avoiding the very difficult problem of generating a sequence of nested grids on complex geometries. The parallel implementation of this technique on a Loosely Coupled Array of Processors at IBM Kingston is described, and results are presented for the Poisson equation on a square with Dirichlet boundary conditions, using a standard smoothing technique.

G. Brussino
IBM Corporation
Department 48B
Kingston, NY 12401, USA.

Present address:
Alliant Computer Systems Corp.
Littleton
Massachusetts 01460 USA.

R. Herbin
IBM Corporation
Department 48B
Kingston, NY 12401, USA.

Present address:
Ecole Polytechnique de Lausanne
GASOV / ASTRID
1015 Lausanne, Switzerland.

V. Sonnad
IBM Corporation
Department 48B
Kingston, NY 12401, USA.

#48/11:20 AM

A Double Augmented Tree Architecture for Parallel Execution of Multigrid Algorithms

Abstract: We propose a multiprocessor architecture for the efficient execution of multigrid PDE algorithms. Using a binary tree as the skeleton, the processors of the same tree level are connected into a ring to form an *augmented tree.* Two augmented trees are stacked together with their roots coalesced to form an X-shaped tree structure. Leaf nodes of the two augmented trees are also used for I/O purpose. Feedback lines are provided from output nodes to input nodes. The architecture is shown to be well suited to parallel processing of multigrid algorithms via a close match of hardware links with the communication patterns of the algorithms. Both conventional and concurrent multigrid algorithms are considered. Performance of this architecture is compared with the hypercube architecture.

H.C. Wang and Kai Hwang

Department of Electrical Engineering-Systems

University of Southern California

University Park, MC-0781

Los Angeles, CA 90089

#38/11:40 AM

Implementation of a Parallel Multigrid Method on a Loosely Coupled Array of Processors

A parallel method with multicolour ordering has been implemented on the loosely Coupled Array of Processors at IBM Kingston. We consider various smoothers, restrictions and prolongations. The parallelization is obtained by an a-priori decomposition of the finest grid; each step of the multigrid method is then vectorized and parallelized, up to the direct solve on the coarse grid, which is done sequentially; data communication via the shared memory occurs at the boundaries of the subregions thus defined. We show that because of the relative parallel unefficiency of the smoother on coarser grids, it is more interesting to do the direct solve on a grid of level superior to 1. We present the numerical results that we obtained for the different algorithms, along with the speed-ups according to the number of processors that we use.

Raphaele Herbin
IBM Corporation,
Department 48B
Kingston, NY, USA.

Present address:
Ecole Polytechnique de Lausanne,
GASOV / ASTRID
1015 Lausanne, Switzerland.

Stephane Gerbi
E.N.T.P.E.
Lyon, France.

Present address:
Yale University,
Computer Science Department,
New Haven, CT, USA.

Vijay Sonnad
IBM Corporation,
Department 48B
Kingston, New York, USA.

THURSDAY, DECEMBER 3

3:40 - 5:00 PM
San Diego Room

Contributed Presentations 15

#127/3:40 PM

SCHEDULE: An Environment for Developing Transportable Explicitly Parallel Codes in Fortran

SCHEDULE is a software package designed to aid in programming explicitly parallel algorithms for numerical calculations. The design goal of SCHEDULE is to aid a programmer with implementation of explicitly parallel algorithms in a style of Fortran programming that will lend itself to transporting the resulting programs across a wide variety of parallel machines. The approach relies upon the user adopting a particular style of expressing a parallel program. Once this has been done the subroutines and data structure provided by SCHEDULE will allow implementation of the parallel program without dependence on specific machine intrinsics. The basic philosophy taken here is that Fortran programs are naturally broken into subroutines which identify units of computation that are self-contained and which operate on shared data structures.

A graphics post processor has been developed to analyze the execution pattern and structure of a parallel program. A graphics pre-processor is under development that will aid in constructing a parallel program directly from the large grain data dependency graph.

J.J. Dongarra and D.C. Sorensen
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4844

#140/4:00 PM

Domain decomposition methods for three-dimensional elliptic problems.

Decomposing a domain in $\mathbb{R}^3$ by parallel vertical planes leads to a block structure of the stiffness matrix where it is natural to form the reduced Schur complement matrix corresponding to the dividing planes. Formation of the matrix is in general impossible so the Schur complement system must be solved by iteration. This requires only the computation of the action of the matrix but this action requires the solution of the same problem on the subdomains. We describe how an effective (optimal order) preconditioner can be constructed and how a recursive subdomain method can be used for the solution of the subdomain problems and of the preconditioner. Pre- and postsmoothing of the residuals on the fine mesh implies that the correction can be computed on the coarse mesh which can decrease the cost by a factor close to 1/8. We describe also how one can implement the method on a cluster of loosely coupled parallel processors where each cluster consists of more tightly coupled processors.

O. Axelsson
Department of Mathematics
University of Nijmegen
Toernooiveld
6525 ED Nijmegen
The Netherlands.

#146/4:20 PM

Fast Poisson Solver On Irregular Regions by Boundary Integral-based Domain Decomposition

A method, called the Boundary Integral-based Domain Decomposition (BIDD) method, is proposed for a fast solution of the Poisson equation on irregular regions. The idea of the method is to adopt the domain decomposition approach by embedding a number of regular subdomains into the irregular problem domain, but to use the standard integral equation formulation of the potential theory to compute the interface values. The method provides a particularly efficient algorithm for parallel processing. Some results of numerical experiments on the Alliant multiprocessor will be reported.

Daeshik Lee, E. Gallopoulos
Center for Supercomputing Research and Development
University of Illinois
305 Talbot Lab
104 S. Wright St.
Urbana, IL 61801-2932

#178/4:40 PM

A High-Performance FFT Algorithm for Vector Supercomputers

Many traditional algorithms for computing the fast Fourier transform (FFT) on conventional computers are unacceptable for advanced vector and parallel systems because they employ nonunit, power-of-two memory strides. This paper presents a technique for computing the fast Fourier transform that avoids such strides and appears to be near-optimal for a variety of current vector and parallel computers. Performance results of a program based on this technique are presented. Notable among these results is that a Fortran implementation of this algorithm on the Cray-2 runs up to 75% faster than Cray's assembly-coded library routine.

David H. Bailey
NASA Ames Research Center
Mail Stop 258-5
Moffett Field, CA 94035

FRIDAY, DECEMBER 4

10:40 AM - 12:00 Noon
San Gabriel Rooms A-C

Contributed Presentations 16

#153/10:40 AM

Parallel Vision Algorithms: An Approach

In this work, we address the issue of implementing some common image processing applications on parallel processors. Our work involved two non Von-Neumann machines, viz. the Butterfly and the NCUBE. Our approach is fundamentally algorithmic in nature.

One basic technique used in computer vision is the Hough technique of feature detection. An application of this is the detection of collinear or nearly collinear feature points. These are evidences of edges, and thus shapes of objects in images.

We define an approach to proper processor management depending on the two quantities as parameters: the image size and the number of processors. Our approach led to a linear speedup.

Sharat Chandran
Larry S. Davis
Center for Automation Research
University of Maryland
College Park, MD 20742-3411

#154/11:00 AM

Performance of a Ocean Circulation Model on LCAP

A numerical ocean model based on the primitive equations has been parallelized and run on the 1CAP system - an experimental parallel machine with an IBM host and ten FPS-x64 attached processors. Using coarse-grain parallelism approach, the computational domain is partitioned into a number of sub-domains according to one of the horizontal directions. Each attached processor computes one sub-domain. Inter-processor communication is accomplished through a set of shared bulk memories. Five synchronization (barrier) points are inserted into the program to guarantee a correct sequence of execution.

A set of ocean model benchmarks have been carried out. The best result shows an impressive speed-up of 6.9 on a 10-processors scale can be achieved.

Hsiao-Ming Hsu
Department of Physical Oceanography
Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

Jih-Kwon Peir
Computer Science Department
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

Dale B. Haidvogel
The Chesapeake Bay Institute
The John Hopkins University
Baltimore, Maryland 21211

#28/11:20 AM

Ising spin on a shared memory machine: computational experience.

Ising spin simulations occur in areas such as studies of phase transitions, and alloy solidification. For a large system, hundreds of thousands of updates spin sites must be performed in order to approach a single equilibrium configuration, and averages over many configurations are required.

The kernel process appears to be "iteration on regular grids." This suggests parallel algorithms related to those used for solving elliptic pde's by finite difference approximations. We have devised algorithms which have similarities with the techniques of red-black ordering and chaotic relaxation. Preliminary tests indicate that, for a five processor machine, one can expect a speedup factor of 3.5 to 4.5.

James L. Blue and Francis Sullivan
Center for Applied Mathematics
National Bureau of Standards
Building 101, Room A438
Gaithersburg, MD 20899

#189/11:40 AM

Parallel Iterative Algorithms To Solve The Discrete LQR Optimal Control Problem With Hard Control Bounds

In this paper we present two gradient based parallel iterative algorithms to solve the linear quadratic regulator (LQR) optimal control problem with hard control bounds. The first algorithm is a parallel implementation of a parametrized gradient projection method and the second algorithm is a parallel implementation of a combination of the first algorithm with a linearly constrained version of the Fletcher-Reeves conjugate gradient method. We show that at each iteration parallel evaluation of the step length and projected gradient of the quadratic cost function can be efficiently performed as a function of the number of processors. We then embed our parallel step length and gradient projection procedures to produce two parallel algorithms which are suitable for real-time online implementation on a SIMD machine.

Gerard G. L. Meyer
Electrical and Computer Engineering Department
The Johns Hopkins University
Baltimore, Maryland 21218

Louis J. Podrazik
Bendix Environmental Systems Division
Allied-Signal, Inc.
Baltimore, Maryland 21284-9840

FRIDAY, DECEMBER 4

10:40 AM - 12:00 Noon
Sacramento Room

Contributed Presentations 17

#175/10:40 AM

A Comparison of the Performance of LINPACK and Vectorized ICCG in Solving a Poisson Equation for a Problem of Two-Phase Flow in a Centrifuge

A problem in fluid dynamics, two phase fluid flow in a centrifuge, is used to compare the direct solution of linear systems using LINPACK routines with the iterative solution of linear systems using Incomplete Cholesky Conjugate Gradient (ICCG). We compare two different approaches to vectorize ICCG,using the diagonal ordering,which yields shorter vector lengths, $O(n)$, for an $n \times n$ grid, and using multicolor orderings which yield long vector lengths, $O(n^2)$.The key consideration is the greater degree of vectorization for the multi-color approach versus the somewhat better convergence results for the diagonal ordering. Results are given for runs on a CYBER 855, Convex C-1, Cray-2, Cray-XMP, and CYBER 205.

Dr. Eugene L. Poole
Mail Stop 244
Nasa Langely Research Center
Hampton, Va. 23665

Jeffery W. Frederick
Dept. of Mechanical Engineering
University of Virginia
Charlottesville, Va. 22901

#187/11:00 AM

On the Solution of Block Tridiagonal Systems of Equations on Multi-Vector Machines - A Comparitive Study

The problem of solving scalar tridiagonal systems has been extensively studied by various researchers. This paper examines the solution of tridiagonal systems in which the non-zero elements of the system are mxm matrices, forming a block tridiagonal system of equations.

Two solution methods are examined: Ritchmeyer's algorithm, and a cyclic reduction based algorithm. It is shown that on multi-vector machines such as the Alliant FX/8, the reduction algorithm takes less time to execute compared to the Ritchmeyer's algorithm.

Ken Dowers, S. Lakshmivarahan, Sudarshan K. Dhall, and Julio Diaz
Parallel Processing Institute
School of Electrical Engineering and Computer Science
University of Oklahoma
Norman, OK 73019

#112/11:20 AM

High Performance Linear Algebra on the AMT DAP500

New techniques will be presented that greatly increase the linear algebra performance of the AMT DAP500, a highly parallel SIMD array machine rated at 9 MFLOPS. Many algorithms, such as solving equations or matrix multiplication, can be expressed as an accumulation of outer products of pairs of vectors. By mapping one vector across the array and multiplying it successively by numbers from the other vector, use can be made of new standard functions defined at a level above the individual array operation. Results will be presented: for example, solving 1000 x 1000 equations improves from 7 to over 30 MFLOPS at 32-bits.

Stewart F. Reddaway
ICL
Cavendish Road
Stevenage, Herts
SG1 2DY England

#134/11:40 AM

Performance of Blocked Gaussian Elimination on Multiprocessors

A performance analysis of blocked Gaussian elimination of a symmetric positive-definite system on a simple MIMD model is presented. The adopted MIMD model assumes a high bandwidth switch or network with bounded bandwidth ports to global and local memories. The blocking is shown to reduce global references to a non-interfering level, under certain mild restriction, so that synchronization and block-overhead are the important performance considerations. The analysis explains our experimental results on the BBN Butterfly multiprocessor: both the strongly limited speedup of the row-based algorithm and the far more extensive speedup of the block-based algorithm.

Elizabeth Jean O'Neil
BBN Advanced Computers, Inc.
10 Fawcett Street
Cambridge, MA 02238

Henno Allik
BBN Laboratories Inc.
Union Station
New London, CT 06320

FRIDAY, DECEMBER 4

10:40 AM - 12:00 Noon
Santa Barbara Rooms A-C

Contributed Presentations 18

#98/10:40 AM

Loop Blocking for Parallel Memory Optimization

Researchers have found that blocking of loops, judiciously applied, can improve the performance of computers by improving the utilization of virtual memory, cache memory, local memory or registers. We present loop blocking as a compiler optimization that is closely related to strip mining, a method used by compilers to generate code for vector registers. Loop blocking combined with index set splitting, loop interchanging and concurrentization is used to automatically optimize linear algebra algorithms for parallel machines with private memories.

Michael Wolfe
Kuck and Associates, Inc.
1808 Woodfield Drive
Savoy, IL 61874

#150/11:00 AM

Parallel FORTRAN: Why You Can't. How You Can.

With the advent of a variety of parallel processing systems, computational investigators are looking for convenient programming models and tools for executing their favorite codes. The first choice of investigators in the physical sciences and engineering appears to be access to parallel processing within FORTRAN. Unfortunately, the basic rules and requirements of an operational FORTRAN product strongly conflict with the most obvious primitives for parallel processing. I will discuss reasons why FORTRAN can not be formally extended for parallel processing without transforming the language beyond recognition (and beyond its operational requirements). I will then discuss an alternative for altering FORTRAN for parallel processing without any formal extensions. This is done with a set of a dozen compiler directives. The goal is to invent a set of directives that will be portable and allow efficient code to be generated for a large variety of parallel processing architectures.

Clifford Arnold
ETA Systems, Inc.
1450 Energy Park Dr.
St. Paul, MN 55108, USA

#180/11:20 AM

Performance Based Distributed Programming

Writing parallel and distributed programs will be very difficult. A number of approaches have been investigated to look at the viability of writing normal, sequential programs without consideration for the hardware architectures or the environmental characteristics. A separate decomposition configuration or specification is then written that will partition the software at runtime. By analyzing the software and hardware as separate steps with an automated specification generator, the program can optimally applied to a larger number of architectures. The automation techniques will be very important especially for the development of large systems for hard real-time systems.

Phillip Q. Hwang
Director, ACSAD Laboratory
Naval Surface Weapons Center, U33
White Oak Detachment
Silver Spring, Maryland 20903-5000
(202)394-1351

#20/11:40 AM

A Programming Aide for Message-passing Systems

A programming aide, which will automatically partition and insert synchronization primitives where needed may increase the programming productivity and reduce the number of errors by an order of magnitude. This paper describes an interactive tool for message-passing systems. The tool performs the dataflow and dependency analyses, partitions programs into serial and parallel segments and simple loops into parallel processes, and inserts synchronization primitives automatically or interactively. It also provides analysis and performance estimates for complex loops. The time for developing programs on message-passing systems is expected to be drastically reduced.

Min-You Wu and Daniel D. Gajski
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801

FRIDAY, DECEMBER 4

10:40 AM - 12:00 Noon
San Diego Room

Contributed Presentations 19

#10/10:40 AM

A Gray-Code Scheme for Local Uniform Mesh Refinement on Hypercubes

Adaptive methods for PDEs can be viewed as a graph problem. Parallel methods must distribute this graph efficiently among the processors. In doing this, the cost of communication between processors and the structure of the graph must be considered. We divide this problem into two phases: labeling of graph nodes and subsequent mapping of these labels onto processors. We describe a new form of Gray-code which we call an *interleaved* Gray-code that allows easy labeling of graph nodes even when the maximal level of refinement is unknown, allows easy determination of nearby nodes in the graph, is completely deterministic, and often (in a well-defined sense) distributes the graph efficiently across a hypercube.

Computational experiments will be presented.

William D. Gropp
Ilse C.F. Ipsen
Dept. of Computer Science
Yale University
P.O. Box 2158 Yale Station
New Haven, CT 06520

#43/11:00 AM

Iteration with Mesh Refinement for Systems of ODE's.

We consider block Picard - Lindelöf iteration (or "waveform relaxation") for systems of initial value problems. A rigorous environment for the following fact is given: one can refine the mesh during the iteration in a reliable way so that the total amount of work is a small multiple of that for the decoupled system.

Olavi Nevanlinna
Helsinki University of Technology
Institute of Mathematics
SF-02150 Espoo, Finland

#108/11:20 AM

Parallelization of Adaptive Grid Domain Mappings

Adaptive grid domain mappings are used as a tool for grid adaption in the numerical solution of partial differential equations. They allow the power of adaptive moving-grid methods to be applied to difficult problems, without introducing irregularities into the grid which may make good vector or parallel numerical methods harder to achieve. The three main computational steps associated with these mappings are: 1) constructing the mapping by tensor product least squares with constraints, 2) evaluating the mapping by tensor product spline evaluation, 3) inverting the mapping by a two-dimensional secant method. The speaker will describe parallel implementations of these three steps on the Sequent Balance 21000 multiprocessor. Issues affecting performance and speedup will be discussed.

Calvin J. Ribbens
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

#25/11:40 AM

Automated Mesh Decomposition and Concurrent Finite Element Analysis for Hypercube Multiprocessor Computers

A concurrent finite element formulation for linear and nonlinear transient analysis using an explicit time integration scheme has been developed for execution on hypercube multiprocessor computers. The formulation includes a new decomposition algorithm which automatically divides an arbitrary finite element mesh into regions and assigns each region to a processor on the hypercube. The decomposition algorithm is deterministic in nature and relies on a scheme which reduces the bandwidth of the matrix representation of the connectivities in the mesh. Numerical results obtained from a 32 processor Intel hypercube (the iPSC/d5 machine) are presented. Speedup factors of greater than 31 have been obtained thereby demonstrating 95% efficient use of the machine.

James G. Malone
Engineering Mechanics Department
General Motors Research Laboratories
Warren, MI 48090-9057

TUESDAY, DECEMBER 1

5:00 - 7:00 PM
Pasadena Exhibit Hall

POSTER SESSION 1

#1
Scheduling of Precedence-Constrained Tasks on Multiprocessors

A major factor in the intelligent utilization of multiprocessor systems is the determination of how to assign computational tasks among processors and to schedule their execution. To this end, we consider a set of precedence-constrained tasks, with arbitrary communication among them. We investigate several approaches to the problem of assigning and scheduling the tasks in order to achieve maximum parallelism and minimum communication overhead in both fully-connected and hypercube multiprocessor ensembles. Three simple heuristics and an adaptation of simulated annealing are described. Results of computational experiments provide a comparative analysis of the performance of these heuristics, and give insight into their appropriate application for scheduling distributed task systems.

C. C. Price
Department of Computer Science
Stephen F. Austin State University
Nacogdoches, Texas 75962

M. A. Salama
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

#2
ITERATIVE SUBOPTIMAL STRATEGIES FOR IMPROVING THE TRANSIENTS IN ADAPTIVE CONTROL SYSTEMS

Iterative solutions which involve the use of iterative optimization techniques are given to improve the adaptation transients in adaptive systems. Two suboptimal controllers are developed and applied to an equivalent discrete-time system valid to describe the behaviour of a recent adaptive control scheme when at least one of the time-varying parameters entering the adaptation algorithm varies within a close domain being admissible for convergence of the estimates. The resulting suboptimal control strategies are translated into corrections of this parameter so as to recompute the adaptation algorithm over a finite horizon. Numerical simulations illustrate the efficiency of the proposed scheme.

M. DE LA SEN AND M.J. GONZALEZ-GOMEZ

DEPARTAMENTO DE FISICA.FACULTAD DE CIENCIAS

UNIVERSIDAD DEL PAIS VASCO

LEIOA-BIZKAIA-SPAIN

#5
A Comparison of Hydrodynamics Multitasking Algorithms

Three one-dimensional, time-dependent hydrodynamics algorithms were multitasked on a Cray X-MP/416. An Eulerian method, a Lagrangian method with artificial viscosity, and a Lagrangian method using a Riemann solver were compared. Speedup results were obtained on a dedicated machine; that is with no one else running. The algorithm changes needed to multitask each of the methods are discussed. Differences in the speedups are discussed as well as the debugging problems encountered. The results were obtained for a sample problem involving a shock wave moving down a channel. The multitasking results are identical to the serial results and agree with the analytical solution.

David A. Mandell
Computational Physics Group
Los Alamos National Laboratory
Los Alamos, NM 87545

* All contributed abstracts are numbered in the order that they were received at the SIAM office.

#6

EFFICIENT PARALLEL ALGORITHMS FOR CONTROLABILITY AND EIGENVALUE ASSIGNMENT PROBLEMS*

The design and analysis of time-invariant linear control systems give rise to a variety of interesting linear algebra problems. Numerically viable sequential algorithms now exist for most of these problems; however, efficient parallel algorithms are virtually non-existent. In this paper, we propose efficient parallel algorithms for multi-input controllability problem and the single-input pole assignment problem. A desirable feature of these algorithms is that they are composed only of basic linear algebraic operations such as vector-matrix multiplication, solution of a linear system and computationa of eigenvalues or singular values of a symmetric matrix, for which efficient parallel algorithms have already been developed. Thus, the proposed algorithms ahve potentials for implementations on some existing and future parallel processors.

B. N. Datta and Karabi Datta
Department of Mathematical Sciences
Northern Illinois University
DeKalb, IL 60115 USA

#7

Use of Matrix Determinant of Kronecker Product In Parallel Matrix Computations

Using the idea of matrix determinant of Kronecker product of two matrices A and B, we show here how several computational problems in linear algebra like (a) finding if two matrices A and B have eigenvalues in common (b) finding a symmetrizer of a matrix and (c) computing the inertia of a nonhermitian matrix can be solved in parallel in $o(\log^2 n)$ steps using a bounded number of processes. At this point, the results are of theoretical interests only. Some new properties of the matrix determinant are also developed in this paper.

Karabi Datta
Department of Mathematical Sciences
Northern Illinois University
DeKalb, IL 60115 USA

#8

A New Efficient Parallel Algorithm for Solving Tridiagonal Linear Systems

CANCELLED

How to solve tridiagonal linear
efficiently is of great signi
new parallel algorithm is ... irst
we transform the tridia ... ions
to a boundary value ... second
order linear recur ... then we
take [log N] ma ... o compute the
necessary in ... coefficients
simultane ... ly we take one step
to get ... on. If N processors are
available ... only take 5[log N]+O(1)
times. Oth ... wise 22.5[N/P]+5[log P]+O(1)
times are needed if we have P processors (P<N). Comparing with the other methods, such as recursive doubling method, cyclic reduction method, etc., the new algorithm presented in this paper is more efficient.

Xiaoshu Pan
Department of Mathematics, Physics & Mechanics
Nanjing Aeronautical Institute
Nanjing, 210002
People's Republic of China

#11

Task Assignment in a Multiprocessor System

This paper presents a simple and efficient heuristic algorithm for assigning a set of partially ordered computational tasks with computation and communication constraints onto a three-dimensional bus-based multiprocessor system so that the total execution time is minimized. The algorithm uses two cost functions: (i) "Minimum Distance" and (ii) "Parallel Transfer" to minimize the completion time. Extensive simulation results show that the heuristic algorithm is highly effective. The results are compared to a "semi-random" assignment algorithm. The proposed algorithm has also been used to solve some practical numerical problems where optimal assignments have been obtained.

C. Siva Ram Murthy, V. Rajaraman
Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560012, INDIA

#12

Algorithm and Performance Notes Concerning Block LU Factorization

A common matrix structure occurring in static finite element analysis is a symmetric banded matrix having a significant number of zeros below the band, known as a skyline system. Since most finite element applications lead to symmetric, positive definite matrices, pivoting is not necessary, which preserves the original structure. This paper first reviews the standard implementation of Crout factorization for solving such a system which is limited to vector speeds. The remainder of the paper discusses blocking strategies for the matrix skyline for two dense algorithms. The first is the matrix-vector version of Crout factorization and the second is an approach based on products of multiple Gauss transformation matrices.

Jim Armstrong
Staff Research Mathematician
Optimization & Research Group
CONVEX Computer Corporation
701 N. Plano Road
Richardson, Texas 75081

#13

Use of Mathematical Switches to Solve Differential Equation Problems.

Mathematical switches have been used to solve differential equation problems in a multischeme approach. The use of the switches is equivalent to select automatically a suitable scheme among several schemes at every computing step. The use of three switches has been examined in a multiprocessor computer. Numerical results of one switch, the switch of exact error, confirm that a combined solution of several schemes is indeed more accurate than one of a single scheme. The use of switch of estimated accumulated error demonstrates that a suitable scheme selection automatically at every computing step is feasible. A multischeme approach with properly designed mathematical switches can certainly resolve many numerical difficulties which are not solvable today.

Yi-ling F. Chiang
Computer and Information Science
New Jersey Institute of Technology
Newark, New Jersey, 07102

#16

Block Truncated-Newton Methods for Unconstrained Minimization

Truncated-Newton methods are a class of optimization methods suitable for large-scale problems. At each major iteration, a search direction is obtained by approximately solving the Newton equations using an iterative method. In this way, matrix costs and second-derivative calculations are avoided, hence removing the major drawbacks of Newton's method. In this form, the algorithms are well-suited for vectorization. Further improvements in performance are sought by using block iterative methods for computing the search direction. In particular, conjugate-gradient-type methods are considered. These still permit vectorization of inner loops, while introducing parallelism at a coarser level.

Stephen G. Nash
Dept. of Operations Research
and Appl. Statistics
George Mason University
Fairfax, VA 22030

#21

Solving non-structural problems via a structural analysis package

MSC/Nastran is widely known and accepted as the
world's leading finite element package for struc-
tural analysis problems.To accomplish th the
package has very advanced mathematical t,
especially in the numerical linear a.
It is not well known however,tha n has
a high level languague,which a ser to
utilize these mathematical r non-struc
tural problems.
The name of this lang P,which stands
for Direct Matrix A CANCELLED Programming.
Large linear lea roblems arise when sol-
ving nonsquar r linear equations,whose
equations a s differ in number.
The pape ent the method of solving large
linea are problems with DMAP .
Prac eriences will be demonstrated on
scala cor and parallel computers.

Louis Komzsik
Senior Numerical Analyst
Technical Services Department
The MacNeal-Schwendler Corporation
815 Colorado Blvd.
Los Angeles,CA 90041

#22

Calculating Good Initial Guesses for Iterative Solutions of Parabolic Partial Differential Equations in Parallel

An iterative procedure for solving parabolic partial differential equations with multiple space dimensions is presented. Such problems arise in fluid dynamics. Areas of rapid change are located by taking an explicit time step that exceeds the Courant condition on a coarse mesh and applying gradient based domain decomposition. The resulting subareas are solved in parallel. Schwarz outer iterations are used for over-lapping subareas. The advanced subareas are superimposed on the explicit solution. The result is used as the initial guess for the coarse mesh iteration. Application of this method improved convergence rates on the coarse mesh tenfold.

A. Louise Perkins
Graduate Group in Computing Science
University of California at Davis in Livermore
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94550

#23
Systematic Method of Condition/Event Nets Design

Many different equivalence notions for comparing concurrent systems have been introduced in different contexts. Aim of this paper is to present minimization procedure and systematic method of condition/event nets design. The notions considered are based on the assumption that the behaviour of a system can be observed, or experienced by an observer only communicating with it. The observer can be considered as a possible environment in which the system is embedded and with whom the system interacts. If two different systems cannot be distinguished by any such observer, then they can be substituted in a consistent way with respect to the interactions with the environment, whatever the environment is. The equivalence notions should be therefore congruences at least with respect to the parallel composition, i.e. in terms of nets with respect to the superposition of events.

Boleslaw Mikolajczak
Computer and Information Science Department
Southeastern Massachusetts University
North Dartmouth, MA 02747

#26
Neural Nets and Sparse Matrices

CANCELLED

Recently there has been increased interest in neural nets as novel approach to the handling of noisy and inexact data. Neural networks are modeled after powerful parallel biological computing systems and are of superior performance for a variety of cognitive tasks. Clearly the structure of a neural net can be modeled by a large, rectangular, sparse matrix. In th's talk we show that some of the recent work b' 'rson and his collaborators can be formul' 'near algebra terms as sparse matrix ur' ..ns. We will express some of Anders' .chms in sparse matrix terminology, .e convergence of a learning r' the combination of recent advances in 'ix computations with neural nets or ssibility of large scale simulatior nets on todays supercomputers.

Horst D. Simon

Engineering Technology Applications Division
Boeing Computer Services, M/S 7L-21
P.O. Box 24346
Seattle, WA 98124

#27
Large Memory Applications on the SCS-40

The SCS-40 is a minisupercomputer with a peak performance of about 44 MFLOPS, 4 Mwords of memory, and a Cray X-MP compatible architecture. In this paper we report about implementations of linear algebra algorithms on the SCS-40, which are designed to solve large dense problems using out-of-core techniques. Based on block algorithms we have implemented linear equation solvers and solvers for the generalized eigenvalue problem, which reduce I/O traffic and perform well on the vector architecture of the SCS-40. We will report numerical results and performance figures for problems with several thousand degrees of freedom.

R. E. Anderson, R. G. Grimes, A. C. Mong, and H. D. Simon

Engineering Technology Applications Division
Boeing Computer Services, M/S 7L-21
P.O. Box 24346
Seattle, Washington 98124-0346

#29
A Parallel Matrix Inversion Algorithm for Triangular Toeplitz Systems

In this paper, a parallel matrix inversion algorithm for n-order triangular Toeplitz systems is given. The number of processors needed by this algorithm is P=n, the number of parallel steps is $T = O(\log^2 n)$. Thus, the speed up is $S = O(n/\log n)$. On the other hand, the fast division of polynomial has also corresponding parallel processing, and an obvious expression of the triangular Toeplitz matrix inversion is presented. Theorem 1. Suppose A is a n-order triangular Toeplitz matrix, then A can be calculated in $T=O(\log^2 n)$ time steps and needs at most P=n/2 processors. Corollary 1. The triangular system of equations Ax=b, where A is a triangular Toeplitz matrix of order n, can be solved in $T=O(\log^2 n)$ time steps and needs P=n processors. Corollary 2. The division of two polynomials $f(x)/g(x)$, where $\deg(f(x))=2n-1$, $\deg(g(x))=n$, can be calculated in $T=O(\log^2 n)$ time steps and at most P=n processors.

You Zhaoyong
Li Lei
Department of Mathematics, Xian Jiaotong University
Shan Xi, China

#30

Two Supernodal Implementations of General Sparse Factorization for Vector Computers

The introduction of hardware gather/scatter on recent vector supercomputers, such as the Cray X-MP, has made it possible to vectorize the computationally intensive kernels at the heart of general sparse Cholesky factorization (an indexed SAXPY, or so-called SAXPYI). By exploiting the supernode structure of the factor matrix, we are able to organize two new implementations around double-loop and even triple-loop computational kernels, whose speed may be greatly increased with standard loop unrolling techniques. The new implementations are generally twice as fast as an optimized sparse factorization module from SPARSPAK, with overall Computational rates ranging from 60 to 85 MFLOPS on several structural analysis matrices.

C. Cleve Ashcraft
John G. Lewis
Barry W. Peyton

Engineering Technology Applications
Boeing Computer Services
P.O. Box 24346, M/S 7L-21
Seattle, Washington 98124-0346

#33

Recursive Doubling Algorithm for Solution of Tridiagonal Systems on Hypercube Multiprocessors

The recursive doubling algorithm as developed by Stone solves a tridiagonal linear system of size n on a parallel computer with n processors using O(log n) parallel arithmetic steps. We give a limited processor version of the recursive doubling algorithm which solves the tridiagonal linear system using O(n/p + log p) parallel arithmetic steps on a parallel computer with p processor where $p < n$. The main techniques rely on fast parallel prefix algorithms, for which we describe an efficient mapping on the hypercube using binary reflected Gray code. We show that for $p \ll n$ the recursive doubling algorithm solves the tridiagonal linear system with linear speed-up and constant efficiency over the sequential LU decomposition algorithm. The experimental results obtained on the Intel iPSC/d5 hypercube multiprocessor are presented. These results indicate greater gain in speed-up and efficiency over the existing sequential algorithms than our theoretical limits predict.

Omer Egecioglu
Department of Computer Science
University of California Santa Barbara
Santa Barbara, CA 93106
Alan J. Laub
Cetin K. Koc
Department of Electrical and Computer Engineering
University of California Santa Barbara
Santa Barbara, CA 93106

#37

Matrix Processing Computers--The Logical Step into the Future

Matrix processing methods of computation generalize the fundamental concepts of scalar and vector computers. Naturally occurring multi-dimensional arrays of data form the basic operands for the matrix processing kernel operations. Control information, data transfer, processor organization, and memory management deal with matrices rather than with scalars. The multi-dimensional parallelism inherent in matrix processing can be effectively exploited in developing novel computer architectures for high-speed scientific computations.

David E. T. Foulser
Saxpy Computer Corporation
255 San Geronimo Way
Sunnyvale, CA 94086

#39

Parallel Implementation of a Neumann-Dirichlet Preconditioner in Domain Decomposition.

Alternating Neumann-Dirichlet subregions with intersecting separators are considered. Comparison of a domain decomposition method in parallel and sequential implementation is discussed. Investigation of computationally optimal number of processors on hypercube architecture is presented.

Wlodek Proskurowski
Majid Haghoo
Department of Mathematics, DRB 306
University of Southern California
Los Angeles, CA 90089-1113

#40
The Network Emulation Problem

Loosely speaking, the network emulation problem asks how best to represent the processor adjacencies of one network with those of another. In general, we are willing to incur additional processor redundancy as long as we can maintain constant communication costs. Herein we address several questions about emulation, including: Which networks can be emulated in the plane? How much redundancy is necessary? How difficult is it to determine whether one network can emulate another?

James M. Abello
Department of Computer Science
University of California
Santa Barbara, CA 93106

Donna J. Brown
Coordinated Science Lab
University of Illinois
Urbana, IL 61801

Michael R. Fellows
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131

Michael A. Langston
Department of Computer Science
Washington State University
Pullman, WA 99164-1210

#44
Performance Evaluation for Stochastic Dynamic Programming on Vector Multiprocessor

Performance has been measured for predictor corrector methods applied to the PDE of stochastic dynamic programming on the Argonne National Laboratory ACRF Alliant FX/8 vector multiprocessor. The dependence of the performance on the number of processors and the number of mesh points are given, as well as the relation to the vector length and memory size. The code has a large Amdahl fraction for both vectorization and parallelization.

Floyd B. Hanson
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439-4844
and
Department of Mathematics, Statistics
and Computer Science
University of Illinois at Chicago
Chicago, IL 60680

#45
PARALLEL RUNGE-KUTTA METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS.

We investigate Runge-Kutta methods for ordinary differential equations, suitable for parallel processing across the method.

Classical explicit methods are not suitable for parallel processing. For non-stiff equations the best candidates seem to be fully implicit methods of high order constructed by collocation . We use fixpoint iteration and as starting values for the internal vectors we use a high order interpolant. A result is found giving the order of an iterate as a function of the order of the predictor and the number of iterations.

Numerical experiments on Cray X-MP/2 and Alliant show that we achieve the predicted speed up within reasonable limits. We also show that an ad-hoc constructed parallel variable step method performs better than the well-known code RKF45 on examples where the function evaluations are compute - intensive.

For stiff equations we construct efficient diagonally implicit RK methods and compare them with a well-known singly diagonally implicit method.

Professor Syvert P. Nörsett
Institute for numerical mathematics
University of Trondheim
7034 Trondheim
Norway

Dr. Ivar Lie
Institute for numerical mathematics
University of Trondheim
7034 Trondheim
Norway

#47
Scientific Computation in FORTRAN 8X

Earlier this year the ANSI FORTRAN Committee completed its work on the Fortran 8X Standard and is in the process of submitting that standard to public review before its adoption by ANSI and ISO as the new standard.

Examples of the use of Fortran 8X in scientific computation will be given which illustrate the four main areas in which Fortran 8X extends beyond (its subset) Fortran 77: simplified array processing, generalised numerical precision, and convenient definition by the programmer of new types and new operations on those types.

Special emphasis will be given in the examples to the use of Fortran 8X in parallel algorithms.

ALAN WILSON
ACTIVE MEMORY TECHNOLOGY
Suite 103, 16802 Aston Street
Irvine, California 92714

#50

Simulation of Larger Dimensions on the Intel iPSC Hypercube

A library of subroutines has been written and implemented on the Intel iPSC Hypercube which allows a programmer to simulate Hypercubes of higher dimensions. This subroutine library runs on a 5-dimensional MX Hypercube and it is capable of simulating up to 9-dimensions. A special Gray code was developed to minimize communication overhead for 2-D and 3-D finite difference calculations. Tests which were run on a 2-D finite difference code are presented. The tests show that under certain conditions programs run faster when higher dimensions are simulated.

Johnny Petersen and Jan Lindheim
Department of Computer Science
Chr. Michelsen Institute
Fantoftvegen 38
N-5036 Fantoft
NORWAY

#51

Parallel Implementation of a New Linear Matrix Solver

An Intel iPSC Hypercube implementation of a new stationary iterative method developed by Dr. John de Pillis is presented. This algorithm finds the solution vector x for the invertable n x n linear system Ax = (I-B)x = f where A is a normalized matrix with real spectrum. The solution method coverges quickly because the Jacobi iteration matrix B is replaced by an equivalent iteration matrix with a smaller spectral radius. The parallel algorithm partitions A row-wise among all the processors in order to keep memory load to a minimum and to avoid duplicate computations. This method reduces only the outer loop in the iteration, which allows for exploitation of vector hardware. Example problems and timings will be presented.

Lisette de Pillis and Johnny Petersen
Department of Computer Science
Chr. Michelsen Institute
Fantoftvegen 38
N-5036 Fantoft
NORWAY

John de Pillis
Bergen Scientific Centre
Allégaten 36
N-5000 Bergen
NORWAY

#53

Parallel Implementation of Fourier Pseudospectral Methods on a Loosely Coupled Array of Processors.

A domain decomposition algorithm was applied for the parallel performance of two dimensional Fast Fourier Transform (2-D FFT) in order to be used in the parallel solution of time dependent partial differential equations using pseudospectral methods. The algorithm was based on a hardware configuration consisting of shared bulk memories connected to attached processors. The pseudospectral method has been implemented in parallel using data sorting for the 2-D FFT, and results have been obtained by solving a two-dimensional time dependent parabolic differential equation. It was found that the parallel efficiency of the method is satisfactory for reasonable problem sizes.

Zaphiris Christidis, and Vijay Sonnad
IBM Corporation,
Data Systems Division
Dept. 48B / MS428
Neighborhood Rd.
Kingston, N.Y. 12401

#54

Optimal Algorithms for Parallel Matrix Multiplication on Vector Computer

In this paper, two parallel algorithms for n-order matrix multiplication on vector computer are given. (1) Optimal inner product algorithm: the number of processors $P = n^3/(\log n)$; the number of parallel steps $T = 3\log n + (\log n)^{2/3} - \log\log n + O(1)$; efficiency $E = O(1)$; (2) Optimal parallel strassen algorithm: the number of processors $p = n^{2.81}/(\log n)$; the number of parallel steps $T = O(\log n)$; efficiency $E = O(1)$. This two algorithms all achieve lower bound of complexity of parallel matrix multiplication in order of magnitude, and bound of the parallel processor attain optimal with the efficiency is O(1). The inner product algorithm whose efficiency is O(1/logn) and parallel strassen algorithm are improved.

You Zhaoyong
Li Lei
Xian Jiaotong University
China

#55

Exploiting Multilevel Parallelism in PDE Software on the CRAY X-MP

This presentation provides an overview of how different types of parallelism in PDE software can be exploited on the CRAY X-MP with vectorization and the two available modes of multitasking: microtasking and macrotasking. A description is given of characteristics of these multitasking mechanisms and of experience in using them to enhance performance of two large codes applied to compressible full potential flow problems. The codes are (1) the general purpose package PLTMG of R. Bank et. al. which incorporates adaptive refinement of unstructured triangle meshes and multigrid solution techniques, and (2) a code utilizing logically uniform structured grids and preconditioned GMRES iterations. Various factors affecting multiprocessing performance are discussed.

Michael Bieterman
Applied Mathematics
Engineering Technology Applications
Boeing Computer Services
P.O. Box 24346, MS 7L-21
Seattle, WA 98124-0346

#57

An Extension of Ryckaer's Algorithm

A numerical algorithm integrating the Cartesian equations of motion for a system of interacting point particles subject to holonomic constraints is formulated. The algorithm is based on the original Ryckaert's method. However, it can be conveniently applied to the molecules in which some centers of force are coincident with very small masses, or are not coincident with any mass.

Sheng-Bai Zhu
Picosecond and Quantum Radiation Laboratory
P.O. Box 4260
Texas Tech University
Lubbock, TX 79409

#58

Solving Very Large Dense Systems of Linear Equations

Several experiments involving a block-based variant of the LU decomposition will be described. We will attempt to find out how much of a CRAY-XMP/48 with a 256 million work SSD (a fast external storage device) can be used to solve a very large system of linear equations (orders exceeding 10,000.) We hope to equal or exceed the computation rate of 718 Mflops reported by Dongarra and Hewitt (January 1986 issue of SIAM J. Sci. Stat. Comput..) As time and machine availability permit, we will also report on similar experiments attempted on a CRAY-2.

Michael Pernice
National Center for Atmospheric Research
Scientific Computing Division
P.O. Box 3000
Boulder, Colorado 80307-3000

#59

Low-Accuracy, High Speed Methods in the Numerical Solution of Ordinary Differential Equations

This paper discusses some new approaches to the numerical solution of ordinary differential equations utilizing parallel architecture. Historically, many problems have required small stepsizes and unacceptably slow integration times. The central idea of this paper is to predetermine h, the stepsize, and based on the speed of the computer, optimize the method for that given step size. Real time integration can be achieved, but the cost is a decrease in accuracy. The approach is to maximize the speed while sacrificing the least possible accuracy. While accuracy may be reduced, other properties may be preserved, for example, stability.

Chris S. Coray, Department of Mathematics, Utah State University, Logan, Utah, 84321

Jean Michel Favre, Tektronix Inc., PO Box 500, 14150 Karl Braun Drive, Beaverton, OR, 97077

#60

Application of Supercomputers in MD study of Liquid Phase Isomerizations

With the aid of supercomputers and vectorization technique, we are able to perform full MD simulations for isomerizations in liquid. Some new features have been discovered. The entropy effect becomes weaker by tightening the freedom of motion for bond angle bending and bond stretching, and becomes negligible for the rigid isomers. The memory kernal and the rotational kinetic energy of the isomer may depend on the reaction coordinate and the barrier height. This dependence is related to the solvent viscosity and to the mass ratio of the isomer. Consideration of these effects may improve the Kramers-Grote-Hynes theoretical rates.

Sheng-Bai Zhu
Picosecond and Quantum Radiation Laboratory
P.O. Box 4260
Texas Tech University
Lubbock, TX 79409

#61

Fortran 9y: Extensions to FORTRAN for supporting network connected computers

The FORTRAN 8x Standard seems well-suited for a single vector processor. However, the Standard does not address issues relating to architectures like hypercubes.
There are two major problem areas to address. One is the problem of communications between and control of (C^2) independent processes. The second is the programmers' "virtual" program - the program seen as a whole.
We describe our fundamental model of network connected computer systems. This model is used to define the C^2 issues. We propose specific primitives based on our FORTRAN implementation on the FPS T-series. We indicate some ideas for defining virtual programs.

D. E. Stevenson
Department of Computer Science
Clemson University
Clemson, SC 29634 -1906

#62

Parallel Algorithms for Market Equilibrium Problems

In this paper we introduce both nonlinear and linearized parallel algorithms for the solution of a variety of market equilibrium problems formulated as variational inequality problems. These algorithms can be used to solve equilibrium problems such as the general spatial price equilibrium problem and the related constrained matrix problem. We provide computational results for the parallel algorithms and compare their performance to nonlinear and linearized serial algorithms. The computer utilized for this purpose is the IBM 3090 at the Cornell National Supercomputing Facility.

Anna Nagurney
Department of General Business and Finance
School of Management
University of Massachusetts
Amherst, MA 01003

#63

Concurrent I/O System for the Hypercube Multiprocessor

One of the deficiencies of existing parallel machines is the lack of adequate, high-bandwidth, parallel I/O facilities. Because of this, many interesting and important problems (image processing, data base) cannot be efficiently solved on parallel architectures. In this paper we will present the Concurrent I/O system (CIO) for the Hypercube multiprocessor. Parallel I/O systems differ from conventional I/O systems. Among challenging problems are efficient coordination of processors operating on an I/O resource (e.g. file), design of a distributed naming convention, design of the distributed cache, and a general facility for implementing I/O operations. We will discuss fundamental issues of I/O systems for parallel machines, describe the architecture of CIO, describe the CIO remote procedure call facility, and present some performance issues related to the cache design and routing strategy.

Andrew Witkowski, M/S 138/208
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

#193

A Pseudo-Spectral Method Implemented on a SIMD Machine

In this paper we compare how effectively a vector computer and a SIMD machine solve a PDE using spectral methods. We solve Burger's equation using an implicit Chebyshev pseudo-spectral technique and optimize the method on both machines. We utilize parallel algorithms for a tri-diagonal solver, FFT, and the Chebyshev recursion and compare run times on a Cray-1 and a Connection Machine I. It is found that the method runs in NlnN time on the Cray-1 and lnN time on the Connection Machine I, but the Cray-1 still obtains the solution in a shorter time.

J.V.W. Reynders
Laboratory for Computational Physics
Code 4400, Naval Research Laboratory
Washington, DC 20375-5000

permanent address:

Program in Applied and Computational Mathematics
Princeton University
Princeton, NJ 08540

#L-1

Parallel Multigrid Algorithms in Computational Fluid Dynamics

We will present progress on an ongoing project at RIACS/NASA Ames on parallel multigrid algorithms with applications to computational fluid dynamics (CFD). The long term goal of the project is to determine whether parallel computers and algorithms can make a significant contribution to the field of CFD in the near future. Our approach has been both theoretical and experimental - varying from studying model problems to actual parallel implementations of CFD codes. In particular, we wil try to speak on our experience in implementing a transonic flow code (FLO52) on the Intel iPSC hypercube.

Eric Barszcz
Mailstop 233-14,
NASA Ames, Moffet Field, Ca 94035

Tony F. Chan
Mathematics Department
UCLA, LOS Angeles, Ca 90024
and
Research Institute for Advanced Computer Science (RIACS), NASA Ames,
Moffett Field, Ca 94035

Dennis Jespersen
Mailstop 202a-1
NASA Ames, Moffet Field, Ca 94035

Ray S. Tuminaro
Computer Science Department
Stanford U., Stanford, Ca 94305
and
RIACS, NASA Ames,
Moffett Field, Ca 94035

#L-2

Matrix Decomposition on Message passing Architectures

We study various algorithms for solving linear systems of equations on different models of parallel architectures: Gauss, Jordan and Givens decompositions on message passing multiprocessors. We first recall some results concerning the shared data model which can be considered as a complete network of processors. In case of message passing model, the processors communicate through processor to processor channels. Y. Saad introduced a parallel Gaussian elimination algorithm on a linear array of processors based on a pipelined broadcasting of the pivot row. We derive a new algorithm by transmitting various pivot rows between neighbouring processors. We show the asymptotic optimality of this algorithm. The preceding technics are applied to derive new parallel algorithms for the Gauss - Jordan method and the orthogonal decomposition using Givens rotations. Lastly we discuss various ways to incorporate pivoting strategies for numerical stability reasons and present results of execution on a 16 nodes Hypercube FPS T-20.

M. COSNARD and B. TOURANCHEAU
Equipe d'Algorithmique Parallèle
CNRS, Laboratoire TIM3
Institut National Polytechnique de Grenoble
46 Avenue Félix Viallet
31031 GRENOBLE CEDEX - FRANCE

#L-3

A Linear Processor Network for the Knapsack Problem

A linear processor network for solving the Knapsack Problem is presented. The parallel algorithm is based on a sequential algorithm from Horowitz & Sahni (E. Horowitz and S. Sahni, "Computing partitions with applications to the knapsack problem", *J. of the ACM*, vol 21, n° 2, april 74, pp 277-292) which is a depth-first branch and search one. Given n positive integers $W = (w_1, w_2, \ldots, w_n)$ and a positive integer M, the Knapsack Problem consists in finding an ensemble $I \subseteq [1,n]$, such that $\sum w_i = M$, for $i \in I$. Each processor (PE) receives as input a combination c of $\{0,1\}^n$, its sum S(c) and a position i. Each PE stores n, the vector W and the searched value M. The basic arithmetic operation is a binary addition performed on the input for generating a new combination, the evaluation of the new sum which requires only one integer addition and the test on the equality between M and the generated sum. Strategies for pruning the search tree and for stopping the network are discussed, as well as how the parallelism is introduced into the starting algorithm. Some results of simulation on the FPS T-20 4-Cube Parallel Computer are presented.

M. COSNARD and A.G. FERREIRA
Equipe d'Algorithmique Parallèle
CNRS, Laboratoire TIM3
Institut National Polytechnique de Grenoble
46, Avenue Félix Viallet
38031 Grenoble Cedex - FRANCE

WEDNESDAY, DECEMBER 2

5:00 - 7:00 PM
Pasadena Exhibit Hall

POSTER SESSION 2

#65
Fast Fourier Transforms on the BBN Butterfly Parallel Processor

We analyze the computation of one-dimensional FFT's on the Butterfly Parallel Processor. Our analysis de-emphasizes the mathematical transformations which could be used to improve performance, and focusses instead on programming techniques. In particular, we describe the use of the Uniform System library to perform storage and task allocation for the scheduling-on-pairs FFT algorithm.

William Celmaster

Bolt Beranek and Newman
Advanced Computers Inc.
10 Fawcett Street, Cambridge, MA 02238

#66
SCIENTIFIC COMPUTER PERFORMANCE EVALUATION: A GUIDE FOR THE PERPLEXED

It is easy to compare the speed of scalar computers, but no recognized standards or methods exist for comparing the speed of advanced scientific computers. Typically, a user would like to know:

1. On average, how fast is a given computer on the class of problems it is expected to run?

2. How reproducible is this average? The speed of vector processors is notoriously sensitive to how much of the workload has vector structure in both program and data.

We will present some answers to these questions, based on real application benchmarks, spreadsheet databases and nonparametric statistics. The methods used (jackknife and bootstrap) give means and variances for computer speed parameters, and we will explore speculations about a "modern portfolio theory" of computer performance based on these means and variances.

M. Edward Borasky
Sr. Staff Analyst
Floating Point Systems, Inc.
P.O. Box 23489
Portland, Oregon 97223

#67
Fast Banded Matrix Multiplication on Parallel Computers

A fast algorithm for multiplying banded matrices is presented. The algorithm's implementation can be several orders of magnitude faster than some commercially available subroutines for banded matrix multiplication on vector and parallel computers. It can also be one order of magnitude faster on scalar machines. The performance of the method on various vector and parallel computers (including the ETA[10], IBM 3090, Alliant FX/8, and Cyber 205) is given and compared. (This algorithm was developed for use in matrix inversion. See the accompanying paper by Magnan for a discussion of this application.)

Richard Bertram and Jerry F. Magnan
Dept. of Mathematics and Supercomputer Computations Research Center
Florida State University
Tallahassee, FL 32306

#68
Parallel and Vector Processing of the Incremental Dynamic Programming Algorithm

This paper discusses modifications to the incremental dynamic programming (DP) algorithm which allow the efficient solution of ten-dimensional problems on a supercomputer. These modifications include (1) parallel processing of the outer-loop over the state variables, (2) vector calculation of the inner-loop minimization over the control variables, and (3) the use of control vectors to accomplish the feasibility tests.

Timing comparisons are made for the algorithm running on a CDC CYBER 205, ETA-10, CRAY X-MP, ALLIANT FX/8, and a VAX 780. The CPU times show the ETA-10 to be over 4 times as fast than the CRAY X-MP for 64-bit precision. Timings on the ALLIANT are in progress with multitasking to split the work to be done in the stage loop among the eight processors.

Daan Sandee, Senior Consultant, Control Data Corp.
Professional Services Division, 100 Sliger Bldg.
2035 E. Paul Dirac Drive, Tallahassee, FL 32304

Dennis Morrow, Research Associate and Mohan Ramamurthy, Research Associate
Supercomputer Computations Research Institute,
523 Keen Bldg., Tallahassee, FL 32306-4052

#69

Simulation of Transonic Flow on a Concurrent Computer

With the increasing use of computational fluid dynamics for aerodynamic design, the demand for cost-effective computers has also increased. The transonic flow regime is one area of study that is important for all supersonic and high subsonic aircraft. This work involves the implementation of the widely-used FL057 code, an Euler equation-based transonic flow solver, on the AMETEK local-memory concurrent computer. As a result of various factors in the design of the original program, the conversion process is straightforward, and the concurrent version is shown to run very efficiently. The paper discusses the implementation philosophy, as well as the memory requirements, verification of results and performance, including speedups.

Mark E. Bassett
Christopher J. Catherasoo
AMETEK Computer Research Division
610 North Santa Anita Avenue
Arcadia, California 91006
(818)445-6811

#71

Distributed Orthogonal Factorization: Givens and Householder Algorithms

We design and implement several algorithms for orthogonal factorization on distributed memory multiprocessors (e.g. hypercubes). Each algorithm is analyzed for its arithmetic and communication complexity. Two of the algorithms, the standard Givens sequence, and the greedy Givens sequence, are based on Givens rotations. Two algorithms employ Householder transformations but with different communication schemes (broadcast and pipelined ring). A fifth algorithm is a hybrid: it uses Householder transformations in an internal phase in which there is no communication, and Givens rotations in a recursive elimination phase. This algorithm possesses the low arithmetic cost of the Householder algorithms and the low communication cost of the Givens algorithms. Computation times from implementations on an intel IPSC hypercube agree quite well with our theoretical analyses.

Alex Pothen
Padma Raghavan
Computer Science Department
The Pennsylvania State University
University Park, PA 16802

#72

A Constraint Algorithm for Maintaining Rigid Bonds in Molecular Dynamics Simulations of Large Molecules

An algorithm is described for efficiently maintaining rigid bonds in molecular dynamics simulations of polyatomic molecules. In this algorithm, a leapfrog integration scheme with constraint conditions is applied to all the bound particles. The procedure for computing the constraint forces requires a small number of iterations. The bond lengths do not have to be kept exactly fixed, but can deviate from their assigned values by small amounts. The algorithm is similar to the SHAKE algorithm and the algorithm of Memon et al. in that it iteratively corrects the constrained equations of motion. However, instead of applying successive corrections to particle positions or inverting a matrix, our method iteratively corrects the constraint forces required for maintaining the rigid bonds.

S.G. Lambrakos, E.S. Oran and J.P. Boris
Laboratory for Computational Physics and Fluid Dynamics
Naval Research Laboratory

M. Nagumo
Biomolecular Engineering
Naval Research Laboratory
Washington, DC 20375-5000

#75

Pattern Recognition by Neural Networks on Hypercube

The objective of this work is to compare the performance characteristics of the parallel implementations of the Hopfield's model and the "back-propagation" model (Rumelhart et al.) in pattern recognition, and to investigate the transition from one model to the other. Specifically, the test case of recognition of Chinese characters is studied on hypercube topology computers. Quantitative analyses of the relationship between the amount of overlap in the training samples and the reliability of both models is presented. Also, the smooth transition linking the two models is discussed. The interpolating network has higher capacity than in the Hopfield's model and learns faster than in the "back-propagation" model.

Alex W. Ho, Wojtek Furmanski
Concurrent Computation Project
Mail Stop 206-49
California Intstitute of Technology
Pasadena, CA 91125

#77

A Parallel Tridiagonal Equations Solver for the Butterfly Multiprocessor

A parallel block partitioning method to solve a tridiagonal system of linear equations is adapted to the Butterfly multiprocessor. A performance analysis of the programming experiments on the 32-node Butterfly is presented. The computational results verify the theoretical speedup and efficiency results of the parallel algorithm over its serial counterpart. Also, included is a comparative study of the performance runs of the same code on the Butterfly processor with hardware floating point versus the one with software floating point facility. The achieved results are shown to be within 82-90% of the predicted performance.

Swarn P. Kumar

Boeing Computer Services
Engineering Technology Applications
P.O. Box 24346, M/S 7L-21
Seattle, Washington 98124-0346

#78

Electromagnetic Scattering Analysis on the Hypercube

To investigate the applicability of a parallel computing architecture to the solution of large scale electromagnetic scattering problems, two existing sequential electromagnetic codes utilizing different algorithms have been converted to run on the Jet Propulsion Laboratory/California Institute of Technology Mark III Hypercube. The first was a frequency domain method of moments approach developed at Lawrence Livermore National Laboratory, "Numerical Electromagnetics Code" (NEC-2). It requires the solution of a large system of linear equations. The second was a time domain finite difference solution of Maxwell's equations for scattered fields. This presentation will describe the algorithms used in the parallel decompositions and discuss the performance results.

Jean E. Patterson
Ruel H. Calalo
Farzin Manshadi
Paulett C. Liewer
William A. Imbriale
James R. Lyons
Jet Propulsion Laboratory
Mail Stop 138/208
4800 Oak Grove Drive
Pasadena, California 91109

#79

A Performance Comparison of Three Mini Supercomputers: the Alliant FX/8, the SCS-40, and the Convex C-1

A benchmark of three leading mini supercomputers has been carried out. Using the Los Alamos Standard Benchmark Set, a series of tests were run on each of the three machines. The results are compared with each other and with the same set of tests run on the Cray Research, Inc. X-MP series machine. In addition, the asynchronous parallel processing capability of one of the machines is explored in some detail. The results suggest a strong showing for this class of computers.

**Harvey J. Wasserman
Margaret L. Simmons
Olaf M. Lubeck
P.O. Box 1663, MS B265
Computing and Communications Division
Los Alamos National Laboratory
Los Alamos, NM 87545**

#80

Multitasking a sparse matrix code on the Alliant FX/8

We have recently multitasked a code for the direct solution of sparse linear equations on the Alliant FX/8. We discuss several issues which are involved, all of which are of relevance to any shared memory multiprocessor. Among these issues are the dynamic allocation of data, the management of task queues, task spawning, and the effect of controlling the granularity. We also show runs of our code under the Schedule package from Argonne which presents a portable interface to users of parallel machines, allows the user to define his computational graph, and has very useful graphic output to a SUN workstation. Our tailored code attains a speed-up of about 6 on the eight processors of the Alliant. We suggest ways of improving it further.

**Iain S. Duff
Computer Science and Systems Division
Harwell Laboratory,
Oxfordshire, England OX11 ORA**

#81

Parallel Triangular Matrix Inversion

The speaker will show that the information flow in parallel triangular matrix inversion is virtually identical to that in parallel Cholesky factorization. So, parallel triangular matrix inversion algorithms can be found by modifying parallel Cholesky factorization algorithms. Drawing from the extensive literature on parallel Cholesky factorization, the speaker will give parallel algorithms for triangular matrix inversion on systolic arrays, array processors, shared-memory multiprocessors and data flow machines.

David C. Fisher, Harvey Mudd College, Claremont, CA 91711 (Mathematics Dept.)

Linda Sattler, The Claremont Graduate School, Claremont, CA 91711 (graduate student in the Mathematics Dept.)

#82

Parallel Processing Bibliography (in Machine-Readable Form)

Rather than print yet another bibliography, the author is doing something different. Hardcopy bibliographies are just that: HARD! This means 1) they don't change, they are not 2) correctable, 3) updateable, or 4) reformattable for different citation styles, and 5) searching becomes difficult when bibliographies get too large. Soft bibliographies (in machine-readable form) are updateable, correctable, are reformattable, and search very quickly. It offers control librarians cannot offer.

The author started with several existing parallel processing bibliographies (including annotations) and has maintained this list for three years. The bibliography is available free of charge to ARPAnet sites (with updates) or for a nominal fee from COSMIC, NASA's software distribution facility. It contains over 7,000 references regarding parallel processing hardware, software, algorithms, and other useful citations. SIAM members are encouraged to contribute. Key word fields note surveys, required, and recommended readings for beginners.

E. N. Miya

Computational Research Branch
NASA Ames Research Center
Moffett Field, CA 94035
eugene@ames-nas.arpa
UUCP: ames!eugene

#85

Analysis of vectorisable preconditioners for 3D problems

Incomplete factorisations of matrices deriving from 3D problems can be based on point, line or plane pivots. Numerical tests will be presented showing that, while recursive methods seem to display largely the same behaviour in all cases, strategies to obtain vectorisability (mainly by explicitly using approximations to inverses of pivots) will result in widely varying convergence speeds (including non-convergence!). Line methods appear the most interesting in terms of numbers of iterations/operations (and with vectorlength h-2/2 on average). A limit matrix analysis on sever strongly anisotropic model problems will analyse the difference between convergence of recursive and vectorisable methods.

Victor Eijkhout
Department of Mathematics,
University of Nijmegen
Toernooiveld 5,
6525 ED Nijmegen
The Netherlands

#86

Decomposition and Solution of Transient Finite Element Problems on Hypercubes

Current research demonstrates that concurrent computers with the distributed-memory, hypercube MIMD architecture can be highly effective machines for finite element applications. Our research applies domain decomposition methods and concurrent equation solvers to examine performance issues for realistic structural and transient solid mechanics problems. The relative merits of iterative (conjugate gradient and Lanczos) and direct or hybrid solvers for a range of problems will be discussed, as well as the effective solution of a sequence of right-hand sides. High concurrent efficiency and transparency at the user level are realized in application on the JPL/Caltech Hypercubes.

Gregory A. Lyzenga and Arthur Raefsky
Jet Propulsion Laboratory,
California Institute of Technology
Mail Stop 238/640
4800 Oak Grove Drive
Pasadena, CA 91109

Bahram Nour-Omid
Mechanics and Materials Engineering Lab.
Lockheed Palo Alto Research Laboratory
Palo Alto, CA 94304

#87

Serial-to-Parallel Conversion of Real Time Continuous System Simulations

Many existing computer simulation programs of real time systems are amenable to parallel execution on local-memory, message-passing architecture parallel processors. We have developed a set of tools which partition an existing simulation program along module boundaries and map the resulting system to a hypercube parallel processor. Our conversion approach seeks to balance the computational load, minimize communication costs and maximize the number of processors employed.

Alan M. Baum
Donald J. McMillan
Computer Science Department
General Motors Research Laboratories
Warren, Michigan 48090-9055

#88

Design of a Real Time, Interactive Simulation Computer with a Parallel Processing Organization

A special purpose computer organization is proposed for fast, real time, interactive simulation. The organization comprises a central processor, tens to thausants of identical peripheral processors (PP's), specially organized memory modules and a set of registers, achieving MIMD (Multiple Instruction stream, Multiple Data stream) parallel processing. Data are transfered by and among the memory modules, rather than by and among PP's, at the speed of memory cycle. A simple example of such organization will be discussed for a simulation of a typical dynamic phenomenon.

Yukoh Kobayashi, Ph.D.
California Research and Development
1118 S. Lake Street
Los Angeles, CA 90006

#92

Parallel Subspace Iteration for Solving the Generalized Eigenvalue Problem

Subspace iteration is used to compute m eigenpairs of a generalized eigenvalue problem, $KX = \lambda MX$, of size n, n >> m. A parallel execution time model of subspace iteration broken into basic phases of the computation, e.g., inverse iteration and Rayleigh-Ritz approximations, is defined. Model parameters account for computation and communication costs, synchronization and parallel processing overheads, both within and between phases.

Results of a parallel implementation of subspace iteration on a shared memory MIMD machine are presented. These results are used to test the validity of the execution time model.

Umesh Mahajan
Merrell Patrick
Department of Computer Science
Duke University
Durham, NC 27706

#93

Performance of the NCUBE Hypercube

The objective of this work is to study the performance characteristics of the NCUBE hypercube, and to investigate the use of the NCUBE parallel graphics interface. This talk also addresses the performance of the high-speed CrOS III crystalline operating system on the NCUBE. As a test problem we chose to study two-dimensional, convectively-dominated, fluid flows using a finite-difference technique which incorporates the flux-corrected transport algorithm. For this algorithm a speed of ~90 Kflops per node is attained. In designing algorithms for the NCUBE it is important to take into account the high message latency of the native VERTEX operating system.

Geoffrey C. Fox, David W. Walker
Concurrent Computation Project
Mail Stop 206-49
California Intstitute of Technology
Pasadena, CA 91125

Gary R. Montry
Sandia National Laboratories
Albuquerque, NM 87185

#95

CONVERGENCE OF CERTAIN PARALLEL SOURCE ITERATION SCHEMES IN PARTICLE TRANSPORT

Combined inner/outer source iteration is a well-developed and widely implemented scheme for the serial solution of (linear) particle transport problems. Brickner, Hiromoto and Wienke (of LANL) have presented several parallel variants of inner/outer source iteration, and results from implementations of these upon a hypercube (as well as a shared-memory machine). These parallel source iteration schemes all correspond to decomposition by energy group.

We present a general model of parallel source iteration that encompasses all schemes in the work cited above. Matters related to convergence of this general class of parallel source iterations also are discussed.

Paul Nelson
Air Force Weapons Laboratory (AWYS)
Kirtland AFB NM 87117-6008
and
Department of Nuclear Engineering
Texas A&M University
College Station TX 77843-3133

C.P. Katti
School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi 110067 India

Beny Neta
Department of Mathematics
Naval Postgraduate School
Monterey CA 93943

ABSTRACTS: POSTER PRESENTATIONS

#96

Study of Program Termination Using Z-Transform Theory

This paper deals with the termination of programs. A method of generating a set of recurrence equations for a given program is explained. Z-transform techniques are applied on these equations and the resulting equations in z are analysed to study the termination of the programs. This is a new approach.

The left shifting property is used for Conversion from discrete form to Z-transform. Then using the well known theorems and inverse transform the response of the equations is estimated. From the response termination is decided.

1. P.A. Venkatachalam
Computer Science Division
Asian Institute of Technology
GPO Box 2754
Bangkok 10501
THAILAND

2. S. Arumugam
Computer Centre, GCT
Coimbatore-641013
INDIA

#97

Parallel Distributed Processing Models for Natural Language Inference

Parallel procedures are defined on distributed network representations of natural language sentences that identify their logical and probabilistic consequences. These procedures are efficient on large natural language data bases and resolve computationally to the parallel identification of requisite patterns of interconnections among distributed occurrences of certain basic elements of which all network representations are wholly composed.

The proposed network representations can be systematically modified to yield a spectrum of parallel distributed processing models for natural language inference, ranging from logically complete models that yield only correct inferences to logically incomplete models that also yield certain incorrect inferences.

Peter G. Tripodes
Department of Mathematics
Los Angeles City College
Los Angeles, CA 90029

#100

A Methodology for Dynamic Scheduling of Computation Tasks in Parallel Processing Systems

The use of parallel processor computer systems for high-performance scientific computing today requires systems designers to partition an application into functions for each processor allocating spare processors for fault tolerance. A methodology is presented that avoids this static partitioning by dynamic assignment and reassignment of application tasks to processors. The computational intractability problem of optimal multiprocessor task assignment is circumvented through the use of heuristic and sub-optimal scheduling algorithms. The development and simulation results of two major categories of algorithms are presented including a centralized, computational-level scheduling approach, and a distributed, general multiprocessor ready-que oriented approach.

Ed P. Andert Jr.
Department of Computer Science
California State University, Fullerton
800 N. State College, E100G
Fullerton, CA 92634

#101

Analysis of Fortran and C Performance on the FPS T Series Parallel Supercomputer.

Benchmark Fortran and C programs have been written for the FPS T Series to evaluate the performance of these languages on this highly parallel scientific computer. Benchmarks were chosen which span a wide cross-section of computational parameters and were run on configurations of up to 128 nodes. Algorithm specification and detailed performance information are presented for each of the benchmarks, and the implications for general performance expectations are discussed.

Thomas E. Bauer and Ian A. Taylor
Floating Point Systems, Inc.
3601 S.W. Murray Blvd.
Beaverton, Oregon 97005

#102

Application of Book Thickness of Graphs to Parallel Computing

This paper applies the notion of book thickness to VLSI design and to the parallelization of sequential code. The book thickness of a graph is the smallest number of pages in a "book" in which the graph can be drawn with vertices along the binding. If the I/O ports of a chip, or smaller processing element, are regarded as vertices with computational paths as edges, then book thickness corresponds directly to time-division-multiplexing of the calculation. The pages of a program flow graph can be implemented in Mealy-type processing elements; an additional set of Moore-type PE's supervise the calculation.

Paul C. Kainen
The Analytic Sciences Corporation
8301 Greensboro Dr.
McLean, VA 22102

#106

Parallel Optimization of Large-Scale Linear and Nonlinear Networks

As a result of the availability of multiprocessors and multicomputers, there has been a renewed interest in decomposition techniques for large-scale network optimization problems. In network optimization the granularity of the decompositions that are possible ranges from very coarse (corresponding to collections of network optimization subproblems) to very fine (in the sense of vector arithmetic). These topics will be addressed in the context of research on large-scale generalized networks and nonlinear networks. We will describe computational experience with two parallel computing systems at the Computer Sciences Department of the University of Wisconsin and possibly with other advanced architectures as well.

Robert R. Meyer
University of Wisconsin-Madison
Computer Sciences Department
1210 West Dayton Street
Madison, Wisconsin 53706

#110

Pipeline SOR: A Parallel Iterative Method for Shared Memory Machines

We present a new parallel version of SOR called Pipeline SOR (PiSOR). This algorithm is designed to make efficient use of processors on a shared memory machine while maintaining the traditional serial SOR convergence properties. Iterations in the PiSOR algorithm are performed by different processors and are pipelined one after the other. Synchronization flags are used to prevent any iteration from overtaking the ones ahead of it. Results will be given from an implementation of the PiSOR algorithm on the Balance 21000.

John P. Bonomo
Wayne R. Dyksen
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

#111

Optimization Strategies for Pattern Selection Problems and Their Implementation on the BBN Butterfly™ Parallel Processor

We have used optimization techniques for finding steady solutions to dynamical systems for which Lyapunov functionals can be constructed. These algorithms have been implemented on a parallel processing computer, and we have compared the efficacy of such calculations with what can be accomplished by direct integration on a serial processor machine. The particular problem addressed is pattern selection in Rayleigh-Bénard convection, for which the corresponding dynamical model equation is the Swift-Hohenberg equation. We conclude that since the dynamics induced by the various minimization algorithms leads to identical steady-state solutions, and as the solution time is a very strong function of the dynamical process used to move along the Lyapunov "surface", the construction of efficient dynamical processes on the Lyapunov "surface" is an effective method for finding steady solutions to the underlying dynamical system.

William Jeffrey
Robert Rosner
Robert Simon
Harvard-Smithsonian Center for Astrophysics
Cambridge, MA 02138

William Celmaster
Eric Tenenbaum
Bolt, Beranek and Newman ACI
Cambridge, MA 02138

#113

Efficient Implementation of Parallel 3-D Seismic Algorithms on the CRAY X-MP

The recent need in 3-D seismic processing both in forward modeling and in migration has fostered the growth in algorithm development on the current generation of vector and/or parallel computers. Computation of meaningful sized problems requires extensive calculations and large 3-D data which must be retrieved and restored several thousand times during the computation.

With the proper choice of algorithms, problem sizes previously thought intractable can now be solved within reasonable time. This concept is demonstrated on the CRAY X-MP/48 with a 128MW SSD with examples of 3-D acoustic and elastic forward modeling using the Fourier method and a 3-D post stack depth migration.

Mickey Edwards
Cray Research, Inc.
5847 San Felipe
Suite 3000
Houston, TX
77057

Christopher C. Hsiung
Cray Research, Inc.
900 Lowater Road
Chippewa Falls, WI
54729

Dan D. Kosloff
Dept. of Geophysics
Tel Aviv University
Tel Aviv, Israel

Moshe Reshef
Cray Research, Inc.
2000 Bering Drive
Suite 830
Houston, TX
77057

#114
Sparse Matrix Solution for Circuit Simulation on Multiprocessors

Motivated by the need to speed up the computationally intensive process of circuit simulation, this paper presents two approaches to parallel sparse LU decomposition: one which exploits parallelism at the level of row operations and the other at the level of elementary arithmetic operations. The various factors that determine the actual achieved performance of a parallel implementation of an algorithm are identified and a general framework is developed for evaluating their relative effects. This framework is then used along with measured results on various circuit matrices obtained from implementations on the ALLIANT FX/8 to contrast the performance of the two algorithms.

P. Sadayappan
Department of Computer and Information Science
Ohio State University
Columbus, OH 43210

V. Visvanathan
AT&T Bell Laboratories, Room 3A-529
Murray Hill, NJ 07974

#115
Reliable Parallel Eliptic PDE Solution

Message-based multiprocessor systems are ideal vehicles for the implementation of relaxation algorithms. However, due to hardware and/or software errors, the end result of the computation may not be reliable. This work embeds reliability, through the use of a constraint predicate, into the class of eliptic PDE relaxation solutions to both flag faulty and ensure correct behavior of the problem solution. The presentation will show both graphically and analytically the effects of faulty behavior and the action of the constraint predicate on the faults.

Bruce M. McMillin and Lionel M. Ni
Department of Computer Science
Michigan State University
East Lansing, Michigan 48824-1027

#116
Efficient Matrix Multiplication on Hypercube Multiprocessors

Matrix multiplication on hypercube multiprocessors is studied to determine the effectiveness of partitioning, pipelining, and architectural change on the performance of algorithms. The study shows that a linear speedup can be achieved by combining the above three factors together, even the input matrices must be downloaded from a host and the resultant matrix must be transferred back. This linear speedup follows from a minimization in the communication overhead through the overlapping between communication and computation. The approach presented here can be applied to other problems to develop more efficient parallel algorithms on hypercube multiprocessors.

Chung-Ta King and Lionel M. Ni
Department of Computer Science
Michigan State University
East Lansing, Michigan 48824-1027

#117
Parallel Simulated Annealing Optimization of Distributed Database Networks

Design of a minimum cost network using standard linear programming codes may require hours of IBM 4341 time for a network as small as five nodes. Simulated annealing can find near optimal solutions in minutes on a PC. We have implemented a parallel version of simulated annealing on Transputers by subdividing the network, optimizing the sub-networks, then optimizing the entire network. The parallel version yields more tightly grouped results and a lower average cost than the sequential version for the same computing time. The parallel version performs better if the subnetworks are not compact but overlap each other entirely.

G. S. Stiles
Allisen Lee
Center for Computer Information Networks Research
Utah State University
Logan, UT 84322-4120

Philippe Julien-Laferriere
Xerox Corporation
Paris, France

#118

Parallel Branch-and-Bound Algorithms on Hypercube Multiprocessors

Branch-and-bound (BB) algorithms are solution techniques for many optimization problems in Operations Research. They require considerable computational effort, making them candidates for parallel processing. This paper presents preliminary work on implementing parallel BB algorithms on hypercube multiprocessors. Integer programming is taken as a case study. Initially considering the special case of 0-1 integer programming, two generic parallel implementations of a BB algorithm for solving that problem on hypercube multiprocessors are presented. The two implementations serve as a feasibility study to determine factors that affect the performance of the general case. Experimental results from the NCUBE/ten with 64 processors demonstrate the performance of the two implementations.

Tarek S. Abdel-Rahman and Trevor N. Mudge
Dept. of Electrical Engineering and Computer Science
University of Michigan
1101 Beal Avenue
Ann Arbor, Michigan 48109-2110

#119

Parallel Programming Without Tears

The design of the Myrias parallel computing system has emphasized ease of use. Selected do statements in Fortran 77 programs are replaced with pardo statements, which create sets of tasks that run in independent address spaces that are merged at task completion. pardos can be nested and combined with recursion. The Myrias implementation assigns tasks to processing elements, balances load between PEs, and moves data between tasks, all automatically. Programs are repeatable, and can be debugged using a symbolic debugger. This paper explains pardo, and develops a particle-in-cell code as an example application.

Richard Foster
Chris Thomson
Dan Wilson

Myrias Research Corporation
900 Park Plaza
10611 - 98 Avenue
Edmonton, Alberta
Canada
T5K 0C7

#L-4

Multiplying arbitrarily shaped matrices on a Data Parallel Computer

The Connection Machine is a Data Parallel Computer for which we have implemented routines for the multiplication of arbitrarily shaped matrices. Hence, inner-product, rank-1, AXPY, and matrix-vector computations are covered. All these operations are asymptotically logarithmic in time if there are sufficently many processors. However, practically they may be considered constant time operations on the Connection Machine by the availability of fast global operations such as copy and summation. We present data allocation schemes, matrix partitioning for maximally concurrent computation, complete algorithms and kernel functions.

S. Lennart Johnsson, Alan Ruttenberg, Alex Vasilevsky, and Jill Mesirov

Thinking Machines Corp.
245 First Street
Cambridge, MA 02142

#L-5

Computing Fast Fourier Transforms on a Data Parallel Computer

The computation of large Fourier Transforms is of great interest for several signal processing applications, the solution of partial differential equations in three dimensional space, and in crystallography. In a data parallel computer an FFT on N elements can be performed in a time proportional to $log_2 N$ with a suitable communication system, and a number of processors proportional to N. The Connection Machine is a highly concurrent computer with a large primary storage. We discribe the implementation of a radix-2, complex FFT on the Connection Machine. Two techniques for the efficient use of the communication system that are described and implemeted are pipelining and cube rotation. We also address the issue of allocation and reuse of twiddle factors.

S. Lennart Johnsson, Alan Ruttenberg, Alex Vasilevsky, and Guy Blelloch

Thinking Machines Corp.
245 First Street
Cambridge, MA 02142

L-7
A Product Integration Method for Vortex Dynamics

A new high-order method for evaluating the multi-dimensional integrals appearing in the vorticity formulation of inviscid, incompressible fluid dynamics is presented. Unlike discrete vortex methods, the singularity in the integrand is analytically reduced, and does not require the introduction of a smoothing (finite-core) function, and its associated non-physical length scale.

The support of a general compact vorticity distribution is covered by a coordinate system periodic in (at least) one direction. By evaluating the integrals in iterated fashion, the integrals over the periodic coordinate are evaluated with spectral accuracy. The resulting scheme yields good accuracy at moderate resolution, and is naturally parallelizable on medium and fine grained levels.

Michael Shelley and
Moshe Israeli
Princeton University
The Program in Applied and
Computational Mathematics
217 Fine Hall
Princeton, New Jersey 08544

#L-10
A Parallel Algorithm for Matrix Inversion

A variational algorithm for inverting matrices is presented. Since the algorithm involves mostly matrix multiplication and addition it is well suited for use on vector and parallel computers. It can be several orders of magnitude faster than methods which are in common use on these computers. An advantage of the algorithm is that it can obtain an approximate, sparse inverse of a sparse matrix. In addition, it is suitable for general classes of matrices and is robust for ill-conditioned matrices. The algorithm's performance on vector and parallel computers is demonstrated with example problems. (See the accompanying paper by Bertram and Magnan for an efficient banded matrix multiplication method developed for the implementation of the inversion algorithm.)

Jerry F. Magnan
Department of Mathematics
Florida State University
Tallahassee, FL 32306

THURSDAY, DECEMBER 3

5:00 - 7:00 PM
Pasadena Exhibit Hall

POSTER SESSION 3

#121
OBTAINING NONUNIFORM SYSTOLIC DESIGNS USING MODIFIED MAPPING TECHNIQUES

Several systolic solutions for matrix triangularization using Givens' rotations have been derived manually. Many techniques for automatically mapping algorithms onto systolic arrays have been studied. Most mapping techniques assume uniformity of data dependence in the source algorithm or dataflow in the target design. The manually-derived designs for triangularization using Givens' rotations exhibit nonuniform dataflow, so it is difficult to apply proposed mapping techniques to derive known systolic designs for this problem. This study uses a systolic array design tool called SAGE to obtain systolic designs equivalent to several manually-derived designs semi-automatically. Several novel solutions are also obtained.

Keith R. Allen & Roy P. Pargas
Department of Computer Science
Clemson University
Clemson, SC 29634-1906

#130

A Parallel Processing Pre-Processor For the FORTRAN Language

Current implementations of parallel FORTRAN languages rely heavily on library routines accessed through the standard FORTRAN subroutine calling convention. This is neither convenient, reliable, or readable, and generally has a significant effect on programming production. This pre-processor introduces new FORTRAN "operators" or keywords that more clearly define the parallel operation being used and that also makes these parallel operations more readily apparent to the casual reader of the program (and even to the programmer!). The primary project goals were to 1) provide "correct" primitive operators such as barriers, critical regions, and others so that the programmer would have basic operators that are reliable; and 2) provide as much machine-independence as possible so that the programmer does not have to be concerned with what machine he will be using.

Bruce W. Suter
Robert M. Hyatt
and Joydeep Roy
Dept. of Computer and Information Sciences
Room 115A Campbell Hall
University of Alabama at Birmingham
University Station
Birmingham, AL. 35294

#132

Inhomogeneous Parallel Supercomputers

Parallel computers have not offered a sufficient advance in performance or cost-effectiveness over conventional supercomputers to inspire widespread interest among scientists and engineers with computation-intensive problems. By exploiting properties of the class of target applications, parallel computers can be designed which are at least 100 times more powerful and cost-effective than those built thus far. Such computers can be built using current off-the-shelf components. The speaker will present techniques for measuring the dependence of the ratio of communication to computation for computation-intensive problems on the decomposition grain size, and will show that this quantity is not scale invariant.

Anthony E. Terrano
Department of Electrical and Computer Engineering
Rutgers University
Piscataway, NJ 08854

#133

FAULT-TOLERANT MAPPING OF LARGE SCALE RESOURCE ALLOCATION ALGORITHMS ONTO CONCURRENT PROCESSORS

This paper is concerned with the mapping of large scale resource allocation algorithms onto parallel computing architectures. The mapping problem is viewed as one of assigning the nodes of a directed, acyclic problem graph (representing the logical and data dependencies among the tasks constituting the algorithm) onto the nodes of an undirected system graph (denoting the parallel computing architecture). The objective is to minimize the completion time of the algorithm such that the redundancy, processor memory, and security constraints are satisfied. The dealys introduced by task queueing, message collision, and task precedence relationships are explicitly modeled. We present three algorithms to solve the mapping problem. The first is a two-stage heuristic that determines the order of task execution based on the critical path method (CPM), and then employs a one-step optimization algorithm to determine the task allocation. The second uses the idea of pair-wise exchange on task execution order to improve the performance of the heuristic algorithm. Finally, we develop an optimal mapping strategy using the A* algorithm, wherein a lower bound on the cost-to-go is obtained from the CPM, and an upper bound from the two-stage heuristic. The performance of the algorithms is illustrated on a variety of examples.

Rong T. Lee
Krishna R. Pattipati
Peter B. Luh
Dept. of Electrical and Systems Engineering
Univ. of Connecticut
Storrs, CT 06268

#135

Research Issues in 1000-Processor Linear Algebra Algorithms in Finite Element Analysis*

Research in parallel finite element solution algorithms on a 10-processor ELXSI 6400 has identified several promising candidates for further investigation on 1000-processor systems. These algorithms include: (1) Jacobi preconditioned conjugate gradients (PCG), (2) incomplete LU factorization PCG, (3) unaccelerated and CG-accelerated multigrid based on Jacobi overrelaxation, and (4) hybrid direct-iterative solution methods. Limitations of these 10-processor algorithms are outlined, and progress towards achieving massive parallelism on a 1024-node NCUBE hypercube is discussed.

*This work performed at Sandia National Laboratories supported by the U.S. Department of Energy under contract number DE-AC04-76DP00789.

Robert E. Benner and Gary R. Montry
Sandia National Laboratories
Albuquerque, NM 87185-5800

ABSTRACTS: POSTER PRESENTATIONS

#137
Implementation of Iterative and Direct Solvers on the UNIX CRAY X-MP24

A comparison is made of a set of linear algebra routines on a vector - parallel supercomputer, CRAY X-MP24. The set contains routines for matrix multiplication and for the iterative and direct solutions of general systems of equations. These routines are designed to utilize a secondary solid stae storage device as well as the parallel processing features of the CRAY X-MP. Performance results are presented that indicate that the speed of these routines approach that of routines with all data in main memory and is close to the maximum speed of the processor.

Michael A. Driscoll and Elissa C. Pariser
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

#138
A Multiprocessor Model for a System Evaluation Tool

An interactive multiprocessor evaluation project is underway and a first version of the tool, PARET (Parallel Architecture Research and Evaluation Tool), has been implemented. A system model for evaluation must accurately represent significant behavior, but be simple and flexible enough for rapid construction and to maintain an interactive execution speed.

Directed flow graphs have been used extensively to model parallel algorithms with nodes representing software modules, and arcs showing the flow of data and control. This paradigm has been extended to include the entire multiprocessor, with nodes representing all primitive units of work which must be scheduled on a physical resource for execution. The author will discuss the model and its application to PARET.

Kathleen M. Nichols
Room No. 4G-622
Computer & Robotics Research Lab.
AT&T Bell Laboratories
Crawfords Corner Road
Holmdel, New Jersey 07733

#139
A Case Study in Scientific Parallel Processing: Prefix-Based Computation

In this presentation the speaker will compare and contrast several implementations of computational problems whose underlying parallelism can be modeled as a parallel prefix computation, and use these results to investigate the various factors that dictate the performance of parallel programs. The problem of prefix computation which is used here as a case study finds widespread applications in computing linear recurrences, matrix multiplication, generalized Horner's expressions, various signal processing and graph theoretic algorithms. Here the speaker will focus on actual implementations of prefix-based applications using a variety of parallel processors (S/NET, NCUBE), software environment (MEGLOS, LINDA) and different programming paradigms (asynchronous, synchronous).

Binay Sugla
Room No. 4G604
Computer and Robotics Research Laboratory
AT&T Bell Laboratories
Crawfords Corner Road
Holmdel, New Jersey 07733

#141
The Effects of Parallel Processing on Matrix Computations

Increased computational speed and the power to manipulate numbers simultaneously are obvious effects of parallel processing. Yet in order to use the technology to its fullest, one must gain insight into its different implementations. The purpose of this presentation is to describe the more versatile implementations of parallel processing on commonly used matrix computations: implementations on algorithms designed to compute the solution of linear systems, the solution of the linear and non-linear least squares problems, and the solution of eigenvalue problems. Also implementation of parallel processing on corrective techniques to improve the accuracy of ill-conditioned problems will be included.

Laurie Melany Darmody
9521 Baltimore Avenue
Laurel, Maryland 20707

#144

The Behavior of Conjugate Gradient Methods on a Multi–Vector Processor with a Hierarchical Memory

Conjugate gradient algorithms are efficient solvers for elliptic partial difference equations. In this paper, an analysis of some of the tradeoffs involved in the design of an efficient implementation of conjugate gradient based algorithms for a multi–vector processor with a two–level memory hierarchy is presented and supplemented by experimentals results obtained on an Alliant FX/8. The considered algorithms comprise the classical conjugate gradient algorithm, preconditioning techniques that are well suited for parallel computers such as polynomial preconditioners and a vectorized version of the incomplete Cholesky preconditioner as well as the reduced system approach.

William Jalby*, Ulrike Meier, Ahmed Sameh
University of Illinois at Urbana–Champaign
Center for Supercomputing R & D
305 Talbot Laboratory
104 South Wright Street
Urbana, Illinois 61801–2932
USA

* on leave from INRIA,
domaine de Voluceau – Rocquencourt,
B.P.105 78150 Le Chesnay, France

#147

The Impact of Hierarchical Memory Systems on Linear Algebra Algorithm Design

In 1986, work was done at CSRD which demonstrated that linear algebra algorithms based on the BLAS or extended BLAS were not able to achieve high performance on multivector processors with a hierarchical memory system due to a lack of data locality. For such machines, it is necessary to implement 'block' linear algebra algorithms in terms of matrix–matrix primitives. The design of efficient linear algebra algorithms for these architectures requires an analysis of the behavior of the matrix–matrix primitives and the resulting block algorithms as a function of certain system parameters (cache size, number of processors ...). This paper discusses the methodology and results of this analysis.

Kyle Gallivan, William Jalby*,
Ulrike Meier and Ahmed Sameh
University of Illinois at Urbana–Champaign
Center for Supercomputing R & D
305 Talbot Laboratory
104 South Wright Street
Urbana, Illinois 61801–2932
USA

* on leave from INRIA,
domaine de Voluceau – Rocquencourt,
B.P.105 78150 Le Chesnay, France

#148

An Analysis of a Multivector Processor Implementation of Cyclic Reduction

In this paper, the architectural issues involved in implementing algorithms to solve regular sparse linear systems, such as cyclic reduction, on a multivector processor with a hierarchical shared memory system are investigated. These issues include: data locality, cache bank conflict resolution strategies and the influence of MIMD execution on the address streams. Techniques are discussed which avoid some of the well–known difficulties associated with cyclic reduction on parallel and vector processors.

Kyle Gallivan, Youcef Saad

University of Illinois at Urbana–Champaign
Center for Supercomputing R & D
305 Talbot Laboratory
104 South Wright Street
Urbana, Illinois 61801–2932
USA

William Jalby

INRIA,
domaine de Voluceau – Rocquencourt,
B.P.105 78150 Le Chesnay, France

#149

An Out-Of-Core Algorithm for Solving Dense Sets of Linear Equations on the FPS-164

Solution of large and dense linear systems on commercial parallel processors has not been extensively studied in the literature. Such systems of equations arise in the moment method solution to many problems in electromagnetic theory. A LU decomposition algorithm is presented to solve large and dense sets of linear equations with complex coefficients on the FPS 164. The algorithm takes into account the necessary I/O and CPU overlapping for computing the solution. We analyze the interrelationship between the memory size, the CPU speed and the I/O transfer rate and give projections for improving the performance.

Mark S. Woodyard
Martin Marietta Orlando Aerospace
P O Box 5837, MP 243
Orlando, FL 32855
and
Hassan N. Srinidhi
Dept. of Computer Science
University of Central Florida
Orlando, FL 32816

#152

Linear Permutations on a Multistage Interconnection Network

Multistage interconnection networks with N inputs and outputs and log N stages pass only a limited set of permutations of data. In this paper we study the routing of the class of linear permutations on a multistage network. This class contains several permutations useful in parallel processing. A simple condition for passability of permutations in this class is formulated. For permutations which are not admissible in one pass, an algorithm for conflict-free routing using multiple passes is presented.

Anujan Varma
IBM Thomas J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598

#156

SOLUTION OF ELLIPTIC SYSTEMS ON HIGHLY PARALLEL COMPUTERS

The fastest known algorithms for the solution of an elliptic system all have one thing in common: they require $O(log(n))$ operations if the number of processors $P = O(n)$. They differ greatly in their performance, however, when one examines the constant in front of the $log(n)$ in this bound. They also differ in their accuracy when roundoff noise is present and in their ability to handle singular elliptic problems. Some, including the FFT based algorithms, will not handle variable coefficient problems. The fact that they differ in their communication requirements means that their performance is architecture dependent. One of the fastest of the highly parallel algorithms is the new algorithm PSMG, for Parallel Superconvergent Multigrid, which Oliver McBryan and the author are developing. PSMG reduces the error by a factor of 0.0014 on every iteration when solving a two-dimensional Poisson problem, for example, and converges almost as fast on nearby variable coefficient problems.

Paul O. Fredrickson
Computing and Communications Division
Los Alamos National Laboratory
Los Alamos, NM 87545

#158

Minimum Spanning Tree on the Hmesh Architecture

A fast algorithm to compute minimum spanning tree of a given undirected graph on Hierarchical MESH connected computer (HMESH) is presented. The time complexity of the algorithm is $O(\log^2 n)$, where n is the number of nodes in the graph. HMESH is a broadcast bus VLSI architecture which consists of $n \times n$ processing elements (PE's) in a mesh connected structure and a hierarchy of broadcast buses in each row and column of the mesh structure such that each broadcast bus is connected to exactly k PE's, where k is a small constant. It is also shown how to compute *connected components* and *transitive closure* of a given undirected graph in $O(\log^2 n)$ with a few modifications to the algorithm presented for computing minimum spanning tree.

R. V. Boppana C. S. Raghavendra
Dept. of Electrical Engineering–Systems
University of Southern California
Los Angeles, CA 90089

#159

Comparison of Methods of Solving Symmetric Positive Definite Linear Systems on a Hypercube Concurrent Computer

The solution of partitioned symmetric positive definite linear systems on a Hypercube concurrent computer using Cholesky decomposition, conjugate gradient iteration, and preconditioned conjugate gradient iteration are compared. The partitionings we use, in their most complete form, are similar to nested dissection carried out to a depth of k, the dimension of the hypercube. We give timings from the Intel iPSC Hypercube at Oak Ridge, as well as counts of significant computation units such as matrix-vector multiplications and forward and backward solves, along with counts of communication requirements such as fan in - fan outs and nearest neighbor exchanges.

Lois Mansfield
Department of Applied Mathematics
University of Virginia
Charlottesville, VA 22901

#161

A Simulation Tool for Optimizing Production Programs

Sim40 is a high-level tool which simulates pipelined vector processor instruction timing for the SCS-40 and Cray X-MP. Sim40 produces a graphic clock-by-clock display of instruction resource utilization which automatically highlights resource contentions preventing instruction issue. Critical code may thus be highly optimized without requiring the programmer to have a detailed knowledge of machine timing characteristics. The novel integration of Sim40 with a symbolic debugger provides a familiar user interface for simulation while adding a powerful "watch address" capability to the debugger. Sim40 is also used to evaluate the performance of new CPU designs.

Steve Howard
Scientific Computer Systems Corporation
10180 Barnes Canyon Road
San Diego, CA 92121

#164

Early Experience With the PASM System Prototype

The PASM (PArtitionable SIMD/MIMD) parallel processing system is dynamically reconfigurable in either MIMD or SIMD modes. This research involved performance evaluation of this multi-mode capability of the 16-processor PASM prototype applied to matrix multiplication. Variations of an $O(n^2)$ square matrix multiplication algorithm were examined for SIMD, MIMD and hybrid SIMD/MIMD modes of parallelism. Results indicated that the SIMD version of the algorithm achieved approximately 65% better efficiency than the MIMD version and was 25% better than the hybrid SIMD/MIMD version. This was due to the ability of the micro-controllers in PASM to execute flow control instructions in parallel with SIMD arithmetic and network transfer operations.

Samuel A. Fineberg
Thomas L. Casavant
Thomas Schwederski
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907
tomc@ee.ecn.purdue.edu

#165

Iterative Algorithms in a Data Driven Environment

A large class of numerical algorithms is iterative. These iterative algorithms have been traditionally implemented, in conventional Von Neuman environments, with REPEAT-UNTIL and WHILE loops. These REPEAT-UNTIL constructs severely limit the performance of parallel processors because they cannot be vectorized and/or multitasked. By inserting a FORALL loop (do i=1,n) inside these REPEAT-UNTIL constructs performance of these iterative algorithms on parallel processors can be improved. The performance of the proposed scheme on the MIT tagged token data-flow architecture has been simulated. Jacobi and Gauss Sidel iterations were used as benchmark algorithms. The proposed scheme nearly doubles the speedup compared to the conventional REPEAT-UNTIL loops.

P. Evripidou and J.L. Gaudiot
Computer Research Institute
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-0781

#166

Computational and Reliability Measures of Homogeneous Multiprocessor Systems

The advent of parallel processing has opened new possibilities in scientific and numerical computing. Computations that would be prohibitively long on uniprocessors can now be carried on parallel machines in a more reasonable time. However, the likelihood of one or more elements failing increases with the complexity and the number of elements in the system. Therefore, the overall reliability and fault-tolerance becomes a key issue in the design and implementation of such systems. The measures proposed here aim at evaluating the achievable trade-offs between faster execution on a larger number of processors and a more reliable computation. Computational oriented measures are proposed that evaluate the probability of a sage completion of a computation on a given syste, By expressing the reliability as function of the number of processor-hours instead of time one can determine the amount of reliable computational work a system can deliver.

Walid Najjar and Jean-Luc Gaudiot*
USC/Information Sciences Inst.
4676 Admiralty Way
Marina del Rey, CA 90292

*Dept. of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-0781

#168

Adaptive Grids of Multivariate, Nth-Order Calculi

This paper outlines some of the analytic-topological problems of decompositions of n-dimensional spaces with relevance for statistical problems; (it constitutes the preparatory stage for 'Nth-Order Canonical Calculi' - in preparation). The extension of wave function approaches to basic statistical problems (with a view to creation of compartible differential geometry of statistics) generates new problems of decompositions. embeddings and imbeddings of (hyper)spaces. These problems, taken together analytisch-topologically, constitute a form of 'intersection set' for unifying, extended 'Statistical Mechanics' for pattern recognition (e.g. neural pattern computing), (bio)chemical spatio-temporal trasformation of evolution structures, and monitored reliability of complex systems et al. This contribuion is on the precisions, open problems and hypotheses.

Dr. Peter Nwove O. MBAEYI
Postfach 2001
D-7400 Tuebingen 1
West Germany

#171
Compiler-Driven Cache Management

Cache performance is critical in cache-based supercomputers, where the cache-miss/cache-hit memory reference delay ratio is typically large. Using compile-time analysis, program behavior can be predicted, and cache control directives can be embedded in the generated code. Thus, cache performance can be improved in a way not possible using conventional techniques. Given hardware able to selectively bypass the cache, cache performance can be increased: pollution can be minimized. Cache line replacement can also be controlled by the compiler (rather than by LRU, etc.), further enhancing performance. This new methodology provides optimal, or near optimal, cache performance without complex hardware.

Chi-Hung Chi and Henry Dietz
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

#172
CRegs: A Hardware Solution to the Alias Problem

Often, pointer and subscripted array references touch memory locations for which there are several possible aliases, hence compilers are unable to optimize code implementing them. The CReg (Cache-Register) mechanism combines the hardware structures of cache and registers to create an entirely new associative memory structure, which can be used either as processor registers or as a replacement for conventional cache memory. By permitting aliased names to be grouped together, CRegs resolve ambiguous alias problems in hardware, thereby enabling compilers to generate more efficient code. Hardware implementing CRegs is of comparable complexity to that implementing a similar-sized conventional cache.

Henry Dietz and Chi-Hung Chi
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

#173
Barrier MIMD: Beyond VLIW

In terms of compiler technology, there is a wide gap between the static SIMD and dynamic MIMD control flow models. Static control brings efficient synchronization and greater opportunity for compile-time optimization, but the SIMD model incorporates many additional constraints. VLIW (Very Long Instruction Word) computers relax these, while preserving the benefits of static control. We propose the **Static-Barrier** machine model as an efficient extension of static control flow beyond the expressive power of VLIW. Synchronization barriers permit efficient VLIW-like instruction scheduling, but also insure reasonable performance where ambiguous inter-processor timing occurs either through differences in control-flow or through unpredictable instruction execution delays.

Henry Dietz and Thomas Schwederski
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

#174
Efficient Iterative Solution of Elliptic PDE's

Most iterative techniques can be very efficient provided one knows the value of one or several special parameters. For SOR, one needs to determine the optimal over-relaxation factor W. Recent theoretical results give new insight into this problem, and the local relaxation approach allows efficient practical implementation of SOR.

We will review the recent developments mentioned above and extend them to the general 9 point stencil case. In addition we observe that SOR does not seem to be well-suited for high-order stencils. Thus, another general iterative method is proposed and analyzed. This discussion will be illustrated with applications, including some Finite Element problems which were implemented on the Connection Machine Computer.

Fabian Waleffe
Department of Mathematics
M.I.T.
Cambridge, MA 02139

and

Thinking Machines Corporation
245 First Street
Cambridge, MA 02142

#176
A MIMD Eigenvalue Algorithm for a Parallel Transputer - Microcomputer System

A parallel MIMD algorithm is developed for determining the eigenvalues of a large matrix. It is implemented in the INMOS OCCAM language for a microcomputer-based Transputer system with the capability of utilizing a large number of Transputers operating in parallel. The algorithm is based on the QR algorithm and is tested on an IBM AT microcomputer combined with up to eight Transputers. Comparisons are made to a QR algorithm on a large computer pertaining to the speed of computation, utilization and efficiency. Improvement and extension of these results will be discussed.

Ms. Ina Vincent and Peter Aitchison
Applied Mathematics Department
University of Manitoba
Winnipeg, Manitoba, Canada
R3T 2N2

#179
Parallel Conjugate Gradients on a 1024-Node Hypercube*

The conjugate gradient method is a preferred approach for the solution of sparse, structured linear systems generated by finite difference and finite element analyses of many two and three dimensional problems. On an ensemble architecture such as the hypercube, much of the conjugate gradient method is easily made massively parallel. An exception is the inner-product calculation, which, for example, can be accomplished quickly using a logarithmic time-complexity exchange algorithm. We will present both theoretical and measured performance for these calculations on a 1024-processor hypercube ensemble.

*This work performed at Sandia National Laboratories supported by the U.S. Department of Energy under contract number DE-AC04-76DP00789.

John L. Gustafson and Robert E. Benner
Sandia National Laboratories
Albuquerque, NM 87185-5800

#181
Performance of Some CFD Loops on Several Parallel Processors

Three loops representative of most algorithms for computational fluid dynamics (CFD) have been run on three different parallel processors in ord r to understand the suitability of diff parallel architectures to CFD ar ns. The three loops have quite dif memory access requirements, .ie trouble with inherant depe , which would inhibit parallelisr .chines which have been used a .LLIANT FX/8, the Intel iPSC, and Butterfly. The loops and their pe .es will be presented.

CANCELLED

E.M. Murman, .es, and D. Modiano
Computatio .d Dynamica Laboratory
Dept. of .cics and Astronautics
and
E. Ducot . V. Klema
Concurren.. Computing Group
Statistics Laboratory

Massachusetts Institute of Technology
Cambridge, MA 02139

#183
A Parallel Algorithm for Computing the Generalized Singular Value Decomposition (GSVD)

One of the recent interesting topics in numerical linear algebra is the GSVD of any two matrices having the same number of columns. The GSVD provides a new tool for the study of matrix pencil. The advent of the wide applications of the GSVD has aroused much interesting in developing a parallel algorithm for computing the GSVD. We will present a parallel implementation of C.C.Paige's sequential Jacobi-like GSVD algorithm. The algorithm is designed to implement efficiently on distributed-memory parallel computer architectures, such as Hypercube and Finite Element Machine.

Zhaojun Bai
Department of Computer Science
University of Maryland
College Park, MD 20742

#163
Task Migration in Partitionable Parallel Processing Systems*

In parallel processing systems that can be partitioned into independent submachines a problem that needs to be addressed is the migration of tasks across partition boundaries. Motivations for task migration include freeing larger partitions, fault tolerance, and load balancing. In this research, the overhead and penalties associated with task migration in parallel systems with multistage cube networks are examined. The overhead of moving tasks is divided into direct and indirect overhead. The sum of direct and indirect overhead represents all delays experienced by the task that is migrated. Migration penalty is introduced as the delay incurred by tasks in other partitions as a consequence of the migration. This information will be used to make task migration decisions.

*This research was supported by the Air Force Office of Scientific Research under grant number F49620-86-K-0006, and by the Supercomputing Research Center under contract number MDA904-85-C-5027.

Thomas Schwederski**
Howard Jay Siegel***
Thomas L. Casavant**

**School of Electrical Engineering, Purdue University, W. Lafayette, IN 47907
***Supercomputing Research Center
4380 Forbes Boulevard, Lanham, MD 20706

#185

Finding Public Domain, Mathematical Software

A service exists by which users can request and receive, via electronic mail, free public domain mathematical software. The service is called *netlib*†. The service runs at Bell Laboratories in Murray Hill, NJ for UUCPnet and Argonne National Laboratories in Argonne, IL for ARPAnet.

Netlib operates without human intervention. Users must therefore be relatively specific in their request(s) to *netlib*. The database of software is built around a collection of independent libraries. An index for each libary is available to briefly describe its contents.

If a requestor is interested in a specific algorithm or topic, which is often the case, software for it may be present in several libraries. For instance, 95 routines to compute eigenvalues exist in 10 different libraries.

We detail here the *find* facility of *netlib* that allows users to locate routines, across all libraries, by using keyword(s). The requestor submits a *netlib* request to find a given keyword(s). Information is returned to briefly describe the capabilities of the routine(s), and the information necessary to submit another *netlib* request for the routine(s), relevant to the keyword.

Several examples are given to illustrate its usage.

† Jack Dongarra and Eric Grosse, "Distribution of Mathematical Software via Electronic Mail", *Comm. ACM*, **30**(403-407), 1987.

Greg Astfalk
AT&T
PO Box 900
Princton, NJ 08540

Jack Dongarra
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439

Eric Grosse
AT&T Bell Laboratories
600 Mountain Ave
Murray Hill, NJ 07974

#186

The Lawrence Livermore Loops with a Vectorizing C Compiler

The Lawrence Livermore Loops, hereafter LLL, was originally authored by F.H. McMahon† in Fortran. It is intended as a measure of the floating-point performance that is relevant to 'scientific computing.' It consists of 24 loops, or kernels, each representing a commonly used, or generally important, scientific algorithm. One of the more common utilizations of the LLL benchmark is for an approximate characterization of todays 'supercomputers.'

Clearly *the* language of scientific computation and supercomputing is, and will be for some time to come, Fortran. The C language offers many features that are utilitarian and aesthetic to some scientists involved in computation. While C is not dominant in scientific computation it is becoming more popular.

With a vectorizing C compiler the scientist can retain the features of C and also achieve the vector speed-ups that were previously confined to Fortran on the advanced architecture machines.

In this paper we present the results of the LLL coded in C, compiled with an automatic vectorizing C compiler and executed on a vector minisupercomputer, the Convex C1.

† F.H. McMahon, "The Livermore Fortran Kernels: A Computer Test of the Numerical Performance Range," Lawrence Livermore Laboratory, UCRL-53745, December 1986.

Greg Astfalk
AT&T
PO Box 900
Princeton, NJ 08540

Dale Lancaster
Convex Computer Corp.
7474 Greenway Center Drive, Suite 450
Greenbelt, MD 20770

#194

A GRAPHICAL PROGRAMMING LANGUAGE FOR THE NAVIER-STOKES COMPUTER

Traditional programming techniques are sometimes inadequate for dealing with the complexities of parallel architectures. Graphical programming tools are emerging as an alternative for dealing with certain aspects of parallelism, especially the definition of processor interactions. We describe a prototype graphical editor/assembler which has been implemented for a reconfigurable pipeline machine, the Princeton/NASA Navier-Stokes Computer. With this tool, users write programs by interactively "drawing" the pipeline configuration and annotating it with additional information about control flow and memory access. These schematic representations of chained vector operations are then automatically translated to object code. The concepts employed are extendable to machines with similar architectures.

Sherryl Tomboulian
Thomas W. Crockett
David Middleton

ICASE
MS/132C
NASA - Langley Research Center
Hampton, VA 23665

#195

A Comparison of Different Parallel Programming Languages Using Choleski's Method on a Multi-Processor Computer

Choleski's method for solving banded symmetric, positive definite systems is implemented on a multiprocessor computer using three FORTRAN based parallel programming languages, the Force, PISCES and Concurrent FORTRAN. The capabilities of the languages for expressing parallelism and their user friendliness are discussed, including readability of the code, debugging assistance offered, and expressiveness of the languages. The performance of the different implementations is compared. It is argued that PISCES, using the Force for fine-grained parallelism, is the appropriate choice for programming Choleski's method on the multiprocessor computer, Flex/32.

Mark Jones
Merrell Patrick
Department of Computer Science
Duke University
Durham, NC 27706

and

Robert Voigt
ICASE
NASA Langley Research Center
Hampton, VA 23665

#L-8

Performance of Some CFD Codes on the Alliant FX/8

Three CFD codes are ported to the Alliant FX/8. The first solves the 3-D unsteady Euler equations using an explicit finite-volume Runge-Kutta time stepping method. The second solves the same problem using the Beam and Worming implicit finite difference method. The third is ARC2D from NASA Ames which solves the unsteady 2-D Navier-Stockes equations using an implicit method. We present extensive observations and results on the performance of these codes on the FX/8. Careful interaction with parrallelizing compiler improves the performance some. Better results are obtained by simple recoding of different segments in the programs.

Walid Abu-Sufah and Amjad Daoud
Department of Computer Science
Virginia Polytechnic Institute and State Univ.
Blacksburg, VA 24061

#L-9

Experiences with the Use of the Cray X-MP/48 Performance Monitoring Tools

We report our experiences with the performance monitoring tools of the Cray X-MP/48. The goal is workload characterization of vectorized/multitasked/ and microtasked computational fluid dynamics codes. Our results revealed that the Cray tools are very informative when executing code on a single CPU. However, when codes are multitasked and/or microtasked, little multiprocessing relevant information could be obtained using the current tools. Some ideas on hardware/software instrumentation of the machine for workload characterization of multiprocessed codes are discussed.

Walid Abu-Sufah and Amjad Daoud
Department of Computer Science
Virginia Polytechnic Institute and State Univ.
Blacksburg, VA 24061

#L-11

Combined Vector and Parallel Processing on the IBM 3090

The IBM 3090 Computer System is capable of taking advantage of both vector and parallel features in a single FORTRAN program. Portions of code with large vector content can be run with reduced CPU time by taking advantage of the built-in 3090 Vector Facility. Portions of the code that are independent can be split off and run on separate processors of the multiprocessor 3090 system, thus reducing the elapsed time for running the entire program. For codes containing portions of independent code with vector content, the independent sections can be executed on separate processors each with their own vector facility. Vectorization of the code is done automatically by the VS FORTRAN Version 2 compiler. The use of the parallel processing capabilities can be exploited by the programmer by use of the VS FORTRAN multitasking facility, or automatically by the compiler by the recently announced Parallel Fortran Product available by special bid from IBM. The capabilities of the hardware and software for utilizing both the vector and parallel capabilities will be discussed. Examples of speedups in both CPU and elapsed runtime will be shown with applications currently taking advantage of these features.

John Tesch & Ken Jacobs
IBM
1503 LBJ Freeway
Dallas, Texas 75234

#L-12

Applications of Level 2 BLAS in the NAG Library

The Level 2 BLAS as specified by Dongarra et al (1986), "An extended set of Fortran basic linear algebra subprograms", Argonne National Laboratory Report, are becoming established as a standard interface to commonly occurring matrix-vector operations, for which efficient implementations are becoming available on various machines. The strategy for the NAG Library is to make calls to Level 2 BLAS wherever possible and maintain the portability of the higher level algorithms. We illustrate this by examples taken from the solution of non-linear equations and non-linear least squares problems. Performance measurements are quoted from various vector-processing machines in support of the strategy.

Richard Brankin
Jeremy Du Croz
Peter Mayes
The Numerical Algorithms Group Limited
Mayfield House
256 Banbury Road
Oxford
OX2 7DE
England

#191

Generalized SOR - New Spectra from Old

Linear system $(I - \tilde{B})\tilde{x} = \tilde{f}$ is equivalent to $Ax = (I_n - B)x = f$ if, for arbitrary initial vectors y_0 and $\tilde{y}_0$, the sequences $\{y_j\}, \{\tilde{y}_j\}$ by

$$y_j = By_{j-1} + f, \quad \tilde{y}_j = B\tilde{y}_{j-1} + \tilde{f} \qquad j = 0, 1, 2, 3, \ldots$$

where $\{\tilde{y}_j\} \rightarrow \tilde{x}$. . Solution vector $x = \pi(\tilde{x})$ must be easy to obtain from $\tilde{x}$. . If, moreover, $\tilde{y}_j \rightarrow \tilde{x}$ faster than $\{y_j\} \rightarrow x$, then $\{\tilde{y}_j\}$ is said to be an acceleration of $\{y_j\}$. . This is the case if and only if the spectral radii have the properties:

$$(1)\ \rho(\tilde{B}) < \rho(B) \text{ and } (2)\ \rho(\tilde{B}) < 1.$$

The SOR theory, for example, supplies a parameterized family of alternate matrices

$$\{\tilde{B} = B_\omega, \tilde{f} = f_\omega\} \text{ where } \tilde{x} = x.$$

We have $\tilde{y}_j \rightarrow \tilde{x}$ at least twice as fast as modified SOR when $\sigma(B)$ is pure imaginary, and $\tilde{y}_j \rightarrow \tilde{x}$ always converges when $\sigma(A)$ (hence, $\sigma(B)$) is real. In fact, our method converges for all systems where $\sigma(B)$ is either real or pure imaginary, even if $\rho(B) > 1$. No symmetry, consistent ordering or positivity is ever required.

John de Pillis
Bergen Science Centre
Allsgat 36
N-5007 Bergen
Norway

#L-13 (POSTER SESS. 1, TUESDAY, DEC. 1, 5:00-7:00 PM)

Using Almost Block Diagonal Codes when Solving ODE's on Supercomputers

We describe briefly the two algorithms which are in common use for solving the almost block diagonal systems which arise in the collocation solution of boundary value solvers in ordinary differential equations and we analyse the cost of using each of them on a vector machine. We present our analysis in terms of the size and order of the differential system and the distribution of the boundary conditions. By considering some extreme cases we are able to demonstrate conclusively the degree of superiority of the favoured algorithm for this application. Finally we make some observations about solving boundary value problems using collocation methods on shared memory parallel machines, highlighting those areas where more work needs to be done.

I. Gladwell
Department of Mathematics
Southern Methodist University
Dallas, TX 75275

R. I. Hay
Department of Statistics and Computational Maths.
University of Liverpool
Liverpool L69 3BX England

C/P = Contributed Presentation
I/P = Invited Presentation
P/S = Poster Session

AUTHOR INDEX

C/P = Contributed Presentation
I/P = Invited Presentation
P/S = Poster Session

C/P = Contributed Presentation
I/P = Invited Presentation
P/S = Poster Session

AUTHOR INDEX

C/P = Contributed Presentation
I/P = Invited Presentation
P/S = Poster Session

C/P = Contributed Presentation
I/P = Invited Presentation
P/S = Poster Session

AUTHOR INDEX

NAME	DAY	TIME	END-TIME	SESSION	ABST.PAGE	ROOM
Schnabel, Robert	Tue. AM	11:00	11:20	C/P 1	A1	San Diego
Schnabel, Robert	Wed. AM	11:20	11:40	C/P 7	A8	San Gabriel A-C
Schwederski, Thomas	Thurs. PM	5:00	7:00	P/S 3	A48	Pasadena Exh. Hall
Schwederski, Thomas	Thurs. PM	5:00	7:00	P/S 3	A46	Pasadena Exh. Hall
Schwederski, Thomas	Thurs. PM	5:00	7:00	P/S 3	A47	Pasadena Exh. Hall
Scroggs, Jeffrey	Tue. AM	11:00	11:20	C/P 4	A5	Santa Barbara A-C
Shei, Bruce	Wed. PM	4:40	5:00	C/P 10	A11	San Diego
Shelley, Michael	Wed. PM	5:00	7:00	P/S 2	A41	Pasadena Exh. Hall
Shen, Yun-Qiu	Wed. AM	10:40	11:00	C/P 9	A9	Sacramento
Shoaff, William	Wed. AM	11:00	11:20	C/P 8	A9	San Diego
Siegel, Howard Jay	Thurs. PM	5:00	7:00	P/S 3	A48	Pasadena Exh. Hall
Simmons, Margaret	Wed. PM	5:00	7:00	P/S 2	A34	Pasadena Exh. Hall
Simon, Horst	Tue. PM	5:00	7:00	P/S 1	A25	Pasadena Exh. Hall
Simon, Robert	Wed. PM	5:00	7:00	P/S 2	A38	Pasadena Exh. Hall
Sincovec, R. F.	Tue. AM	10:40	11:00	C/P 1	A1	San Diego
Siva Ram Murthy, C.	Tue. PM	5:00	7:00	P/S 1	A23	Pasadena Exh. Hall
Sjogreen, Bjorn	Tue. AM	10:40	11:00	C/P 4	A4	Santa Barbara A-C
Smooke, Mitchell	Thurs. AM	11:40	12:00	C/P 11	A13	Santa Barbara A-C
Sonnad, V.	Thurs. AM	11:00	11:20	C/P 14	A15	Sacramento
Sonnad, V.	Thurs. AM	11:40	12:00	C/P 14	A16	Sacramento
Sonnad, V.	Tue. PM	5:00	7:00	P/S 1	A28	Pasadena Exh. Hall
Sorensen, D. C.	Tue. AM	11:00	11:20	C/P 4	A5	Santa Barbara A-C
Sorensen, D. C.	Wed. AM	11:20	11:40	C/P 8	A9	San Diego
Sorensen, D. C.	Thurs. PM	3:40	4:00	C/P 15	A16	San Diego
Srinidhi, Hassan	Thurs. PM	5:00	7:00	P/S 3	A44	Pasadena Exh. Hall
Stevenson, D. E.	Tue. PM	5:00	7:00	P/S 1	A30	Pasadena Exh. Hall
Stiles, G. S.	Wed. PM	5:00	7:00	P/S 2	A39	Pasadena Exh. Hall
Stiles, G. S.	Wed. AM	11:20	11:40	C/P 9	A10	Sacramento
Sugla, Binay	Thurs. PM	5:00	7:00	P/S 3	A43	Pasadena Exh. Hall
Sullivan, Francis	Fri. AM	11:20	11:40	C/P 16	A18	San Gabriel A-C
Suter, Bruce	Thurs. PM	5:00	7:00	P/S 3	A42	Pasadena Exh. Hall
Szyld, Daniel	Tue. AM	11:00	11:20	C/P 3	A3	San Gabriel A-C
Tang, Wei Pai	Wed. AM	9:15	10:00	I/P 6	3	San Diego
Taylor, Ian	Wed. PM	5:00	7:00	P/S 2	A37	Pasadena Exh. Hall
Tenenbaum, Eric	Wed. PM	5:00	7:00	P/S 2	A38	Pasadena Exh. Hall
Terrano, Anthony	Thurs. PM	5:00	7:00	P/S 3	A42	Pasadena Exh. Hall
Tesch, John	Thurs. PM	5:00	7:00	P/S 3	A50	Pasadena Exh. Hall
Thomson, Chris	Wed. PM	5:00	7:00	P/S 2	A40	Pasadena Exh. Hall
Tomboulian, Sherryl	Thurs. PM	5:00	7:00	P/S 3	A49	Pasadena Exh. Hall
Topka, Terry	Tue. AM	11:40	12:00	C/P 2	A3	Sacramento
Tourancheau, B.	Tue. PM	5:00	7:00	P/S 1	A31	Pasadena Exh. Hall
Tripodes, Peter	Wed. PM	5:00	7:00	P/S 2	A37	Pasadena Exh. Hall
Tuminaro, Ray	Tue. PM	5:00	7:00	P/S 1	A31	Pasadena Exh. Hall
Umetani, Yukio	Tue. AM	11:40	12:00	C/P 1	A2	San Diego
Vandewalle, Stefan	Tue. PM	4:00	4:20	C/P 5	A6	San Diego
Varma, Anujan	Thurs. PM	5:00	7:00	P/S 3	A45	Pasadena Exh. Hall
Vasilevsky, Alex	Wed. PM	5:00	7:00	P/S 2	A 40	Pasadena Exh. Hall
Venkatachalam, P.A.	Wed. PM	5:00	7:00	P/S 2	A37	Pasadena Exh. Hall
Vincent, Ina	Thurs. PM	5:00	7:00	P/S 3	A47	Pasadena Exh. Hall
Visvanathan, V.	Wed. PM	5:00	7:00	P/S 2	A39	Pasadena Exh. Hall
Voigt, Robert	Thurs. PM	5:00	7:00	P/S 3	A50	Pasadena Exh. Hall
Waleffe, Fabian	Thurs. PM	5:00	7:00	P/S 3	A47	Pasadena Exh. Hall
Walker, David	Wed. PM	5:00	7:00	P/S 2	A36	Pasadena Exh. Hall
Wang, H. C.	Thurs. AM	11:20	11:40	C/P 14	A16	Sacramento
Wasserman, Harvey	Wed. PM	5:00	7:00	P/S 2	A34	Pasadena Exh. Hall
Watson, Layne	Wed. AM	11:00	11:20	C/P 9	A10	Sacramento
Wilson, Alan	Tue. PM	5:00	7:00	P/S 1	A27	Pasadena Exh. Hall
Wilson, Dan	Wed. PM	5:00	7:00	P/S 2	A40	Pasadena Exh. Hall
Witkowski, Andrew	Tue. PM	5:00	7:00	P/S 1	A30	Pasadena Exh. Hall
Wolfe, Michael	Fri. AM	10:40	11:00	C/P 18	A20	Santa Barbara A-C
Womble, David	Thurs. AM	11:20	11:40	C/P 13	A15	San Gabriel A-C
Woodyard, Mark	Thurs. PM	5:00	7:00	P/S 3	A44	Pasadena Exh. Hall
Wu, Min-You	Fri. AM	11:40	12:00	C/P 18	A20	Santa Barbara A-C
Yamabe, Michiru	Tue. AM	11:40	12:00	C/P 1	A2	San Diego
Zhang, He	Wed. AM	10:40	11:00	C/P 8	A8	San Diego
Zhaoyong, You	Tue. PM	5:00	7:00	P/S 1	A25	Pasadena Exh. Hall
Zhaoyong, You	Tue. PM	5:00	7:00	P/S 1	A28	Pasadena Exh. Hall
Zhu, Sheng-Bai	Tue. PM	5:00	7:00	P/S 1	A29	Pasadena Exh. Hall

C/P = Contributed Presentation
I/P = Invited Presentation
P/S = Poster Session

REGISTRATION INFORMATION

The registration desk will be located in the California Foyer on the second floor of the hotel and will be open as listed below.

Monday, November 30	4:00 PM–8:00 PM
Tuesday, December 1	7:00 AM–7:30 PM
Wednesday, December 2	7:30 AM–6:00 PM
Thursday, December 3	7:30 AM–6:00 PM
Friday, December 4	7:30 AM–1:00 PM

		SIAG/ SC*	SIAM Member	Non Member	Students
Conference Fees	Advance	$ 90	$ 95	$125	$20
	On-Site	$120	$125	$155	$20

* Discounted fee for members of the SIAM Activity Group on Supercomputing

Beef and Beer Party
Tuesday, December 1
7:30 PM
Pasadena Exhibit Hall
Westin Bonaventure
$15.00

Non SIAM Members
Non-member registrants are encouraged to join SIAM in order to obtain the member rate for conference registration and enjoy all the other benefits of SIAM membership.

Special Note
There will be no prorated fees. No refunds will be issued once the conference has started.

Telephone Messages
The telephone number at the Westin Bonaventure is 1-213-624-1000. The Westin Bonaventure will either connect you to the SIAM desk or forward a message. Messages received at the SIAM registration desk are posted on a bulletin board in the registration area.

SIAM CORPORATE MEMBERS
Non-member attendees who are employed by the following institutions are entitled to the SIAM member rate.

Aerospace Corporation
Amoco Production Company
AT&T Bell Laboratories
Bell Communications Research
Boeing Company
Cray Research, Inc.
Culler Scientific Systems Corporation
E.I. Du Pont de Nemours and Company
Eastman Kodak Company
Exxon Research and Engineering Company
General Electric Company
General Motors Corporation
Giers Schlumberger
GTE Laboratories, Inc.
Hollandse Signaalapparaten B.V.
IBM Corporation
Institute for Computer Applications in Science and Engineering (ICASE)
IMSL, Inc.
MacNeal-Schwendler Corporation
Marathon Oil Company
Martin Marietta Energy Systems
Mathematical Sciences Research Institute
Standard Oil Compay of Ohio (SOHIO)
Supercomputing Research Center, a division of Institute for Defense Analyses
Texaco, Inc.
United Technologies Corporation

Credit Cards
SIAM is now accepting credit cards for the payment of registration fees and special functions.

GENERAL INFORMATION

EXHIBITS

Software/Hardware Books/Journals

The exhibit area is in the Pasadena Exhibit Hall of the hotel. The exhibit times are 9:30 AM–10:00 PM, Tuesday, December 1; 9:30 AM–7:00 PM, Wednesday and Thursday, December 2 and 3; and 9:30 AM—1:00 PM Friday, December 4. The exhibits setup time will begin at 2:00 PM, Monday, November 30; breakdown will begin at 1:00 PM and will end at 4:00 PM, Friday, December 4.

Special Notice to:

All Conference Participants

SIAM requests conferees to refrain from smoking in the session rooms during lectures. Thank you.

UPCOMING CONFERENCES

May 23–26, 1988
Third SIAM Conference on Linear Algebra
The Concourse Hotel
Madison, WI

June 13–16, 1988
Fourth SIAM Conference on Discrete Mathematics
Cathedral Hill Hotel
San Francisco, CA

July 11–15, 1988
SIAM Annual Meeting
Hyatt Regency Hotel
Minneapolis, MN

EXHIBITORS

Software/Hardware

Active Memory Technology

Alliant Computer Systems

Amdahl Corporation

Applied Dynamics

Convex Computer Corp.

Floating Point Systems

IBM Corporation

Intel Scientific Computers

Kuck & Associates, Inc.

Multiflow Computer, Inc.

NCUBE

Numerical Algorithms Group

Numerix Corporation

Saxpy Computer Corporation

Sequent Computer Systems

Publishers

Academic Press, Inc.

American Institute of Physics

American Society of Mechanical Engineers

Baywood Publishing Company, Inc.

Butterworth Publishers

Cambridge University Press

North Holland—A Division of Elsevier Science Publishers B.V.

Penn State Press

Plenum Publishing

SIAM

Note: This list includes only those exhibitors who submitted their contracts for exhibit space prior to the printing of the final program.