Activities and Operations of the Advanced Computing Research Facility

October 1986 - October 1987

Gail W. Pieper

Mathematics and Computer Science Division

This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research of the U.S. Department of Energy, under contract W-31-109-Eng-38.
From left to right, the ACRF staff members are as follows:

**David Levine.** Dave is the primary consultant for the Alliant and hypercube computers. He helps solve problems about Fortran and parallel compilers and vectorization and is involved in collaborative projects in large-scale scientific computing. He also carries out benchmarking and performance analysis on the parallel computers.
Burton Garbow. Burt helps set up new accounts for ACRF computer users, in particular, those enrolled in the classes offered at Argonne. In addition, Burt coordinates the response to proposals for ACRF project authorization and thus acts as a de facto contact for documentation and questions about the facility.

Jack Dongarra, ACRF Scientific Director. Jack's primary research interests are devising linear algebra algorithms and developing portable algorithms for high-performance computers. Jack is also working with researchers at the Center for Supercomputing Research and Development at the University of Illinois. He has been involved in the LINPACK and EISPACK projects, is codesigner of the Level 2 and Level 3 BLAS which exploit the special features of advanced architectures, and is working on a linear algebra package (LAPACK) designed to supersede LINPACK and EISPACK.

Eugene Rackow. Gene is responsible for maintaining the hardware, installing systems software, configuring the file systems and mailers, and coordinating systems backups. He is also in charge of site preparation and physical installation of new machines, and he maintains the MCS Division workstations and VAX computers.

Teri Huml. As ACRF secretary, Teri coordinates the numerous seminars, workshops, and classes that staff members offer to encourage use of the ACRF computers. She also handles requests for information about the ACRF and mails out appropriate documentation to interested users.

Ewing Lusk, ACRF Deputy Scientific Director. Focusing on parallel programming methodologies, Rusty was one of the developers of a new approach that implements synchronization through monitors written as macros. At present, he is investigating the use of logic programming as a language to control parallelism in numerical programs.

Gail Pieper. As technical editor, Gail prepares proposals, press releases, and progress reports and helps write articles about the ACRF. She is also coordinating the new ACRF affiliates programs for universities and industry.

Rick Stevens, Manager. As manager of the computing resources for the MCS Division, Rick coordinates the integration of new hardware, the acquisition of software, and the production of user documentation. He also does systems programming, consults with various user groups working on parallel algorithms, and carries out performance analysis studies,
## Contents

1. Advanced Computing Research Facility: Its Role and Use...............................................................1

   1.1 Upgrading of Computers ........................................................................................................2
   1.2 Installations and Configuration Changes ...........................................................................2
   1.3 Networking Changes ........................................................................................................3

2. Advanced Scientific Computing Research in the ACRF.................................................................4

   2.1 Algorithms and Software ......................................................................................................4
   2.2 Software Tools .....................................................................................................................5
   2.3 Parallel Programming Methodologies ...............................................................................7
   2.4 Programming Languages and Dialects ..............................................................................7
   2.5 Automated Deduction ........................................................................................................9
   2.6 Publications and Presentations ............................................................................................9

3. User Facility Activities..................................................................................................................12

   3.1 Projects ................................................................................................................................12
   3.2 Proposals .............................................................................................................................16
   3.3 University Classroom Users ..............................................................................................16

4. User Education............................................................................................................................16

   4.1 Visitors ................................................................................................................................17
   4.2 Seminars ...............................................................................................................................20

Appendix - Meetings, Workshops, and Conferences ....................................................................24

References.........................................................................................................................................25
Activities and Operations of the Advanced Computing Research Facility

October 1986 - October 1987

prepared by

Gail W. Pieper

1. Advanced Computing Research Facility: Its Role and Use

This report discusses research activities and operations of the Advanced Computing Research Facility (ACRF) from October 1986 through October 1987.

The ACRF was established by Argonne National Laboratory in 1984 to allow researchers access to a wide variety of advanced computer architectures. The underlying belief was that new architectures may well revolutionize computing and that competent researchers must have access to a number of systems to allow comparative analysis to begin. The facility is operated by the Mathematics and Computer Science Division and supported by funding from the Department of Energy.

The first machine in the ACRF was a Denelcor HEP, a commercially available, general-purpose multiprocessor. The facility has expanded considerably since then and now features six machines of varying parallel architectures: an 8-vector-processor Alliant FX/8, a 20-processor Encore Multimax, a 24-processor Sequent b21, a 32-processor Intel iPSC, a 16-
vector-processor Intel iPSC, and a 1024-processor Active Memory Technology DAP. These machines offer substantial capabilities for both numerical and symbolic applications. For example, the Alliant enables researchers to study the tradeoffs in vector and parallel computation for numerical applications, while the hypercubes offer the ability to study numerical computation in a distributed environment (this is particularly true of the 16-processor hypercube, which includes vector units on each node). Periodically, other multiprocessors are available in the ACRF on loan from vendors for testing, and we expect to add other promising systems as they become available.

To encourage researchers to use the ACRF machines, we teach three-day courses on parallel computing, sponsor workshops and conferences (see the Appendix), and grant time on the ACRF machines via ARPANET, MFENET, and TYMNET. Currently, researchers from universities, government laboratories, and industrial research centers are using the machines in applications ranging from linear algebra and differential equations to logic programming and theorem proving.

Our ultimate goal is to make the ACRF a national center for advanced-computing research.

1.1. Upgrading of Computers

Since establishing the ACRF in 1984, we have continued to select and operate computers with innovative designs that are likely to be effective for a wide range of scientific computing tasks (see Refs. 1-2 for earlier progress reports on the ACRF). These computers are used to investigate parallel programming methodologies, to study automated reasoning, to develop performance measurement techniques, and to devise tools for program transformation.

1.2. Installations and Configuration Changes

As part of our continuing efforts to have state-of-the-art machines available for experimentation, we have added to the ACRF an AMT DAP. This computer system features a bit parallel SIMD architecture with 1024 processors arranged in a 32 x 32 array. A distinct advantage of this new machine is that it is an attached processor to the Sun, thus enabling users to do software development in Unix and run a simulator on any Sun workstation to prepare for running on the DAP.
In addition to the DAP, the ACRF features five other multiprocessors:

- Alliant FX/8, with 8 vector processors sharing 32 megabytes of memory. The vector capability of this machine makes it ideal for research in high-performance numerical computations.

- Encore Multimax, with 20 processors sharing 64 megabytes of memory. During the past year, the Encore was upgraded from NS32032 processors rated at .75 mips (millions of instructions per second) to NS32332 processors rated at 2.0 mips.

- Sequent b21, with 24 processors and 24 megabytes of shared memory.

- Intel iPSC/d5, a five-dimensional hypercube architecture with 32 nodes, each having a 0.5 megabyte of memory. The hypercube design enables researchers to develop algorithms for processors that communicate through message passing rather than shared memory.

- Intel iPSC-VX/d4, a four-dimensional hypercube architecture with 16 nodes, each having 1.5 megabytes of memory. A special feature of this system is its vector capability, making the machine suitable for research in high-performance numerical computations. The Argonne iPSC-VX/d4 was the first vector system installed at a customer site.

These five multiprocessors all run a version of Unix, with Fortran and C compilers. In addition, the Sequent b21 is equipped with Pascal and Ada compilers, and the Encore Multimax has a Pascal compiler.

1.3. Networking Capabilities

The ACRF machines are linked to each other and to the Division's VAX 11/780 dual-processor system. The VAX itself is a host on the ARPANET/MILNET and is connected to MFENET and TYMNET. Thus, the ACRF machines are readily available throughout the country.
Figure 1. The ACRF multiprocessors are linked to each other and various networks.

2. Advanced Scientific Computing Research in the ACRF

The advanced computing research program has as its objective the creation of portable algorithms, software, and programming techniques for advanced computer architectures. The following sections highlight recent projects conducted by scientific staff in the Mathematics and Computer Science Division. Principal investigators are listed in parentheses after each heading.

2.1. Algorithms and Software
(J. J. Dongarra and D. C. Sorensen)

Our efforts to create algorithms and software for advanced-computer architectures involved several collaborative projects:

- With J. Scroggs (CSRD, University of Illinois at Urbana) and R. Chin and G. Hedstrom (Lawrence Livermore National Laboratory), we have developed a parallel algorithm for solving singularly perturbed parabolic PDEs. The algorithm is based upon asymptotic analysis of the equation to determine the domain decomposition. It was written using SCHEDULE and has been running successfully on the Alliant.
• With J. Demmel and A. Greenbaum of the Courant Institute and J. DuCroz and S. Hammarling of the Numerical Algorithms Group, we received NSF funding to partially support the design and implementation of a transportable linear algebra library in Fortran 77 for efficient use on high-performance computers. The project will facilitate the development of scientific codes on advanced computers, increase the portability of scientific codes between different computing environments, and provide tools to help evaluate computer performance. Central to this proposal is the use of the advanced computers in the ACRF as well as other high-performance computers.

• With I. Duff of AERE Harwell and J. DuCroz and S. Hammarling of NAG, we prepared a proposal to design a Level 3 BLAS (Basic Linear Algebra Subprograms). The new BLAS would be better able to exploit the features of high-performance computers.

In addition to these collaborative efforts, we developed several parallel algorithms for determining the singular value decomposition of a matrix. These algorithms are based on a divide-and-conquer strategy. One of the algorithms appears to be competitive in accuracy with the corresponding LINPACK algorithm and shows considerable speedup in both parallel and serial mode.

We also ported to the Sun workstations a large computer program (COMMIX-1B) for thermal hydraulic analysis and have developed an interactive preprocessor and postprocessor. We have begun studying the use of the COMMIX module on the workstations in conjunction with the parallel processors in the ACRF.

2.2. Software Tools
(B. S. Garbow and W. R. Cowell)

Toolpack/l has been installed on the VAX, Sun workstations, the Sequent b21, the Encore, and the Alliant. The Fortran-language tools access a library of low-level C-language subroutines. Since the Fortran-C interface is different on each of the machines, the major task in the installations involves tailoring the C routines to accommodate the corresponding interface. We are also developing a specialized interface for Sun workstations using windows; we expect that this interface will facilitate the interactive use of Toolpack tools.
As an outgrowth of our work on Toolpack, we have developed a collection of software tools intended to improve the performance of Fortran programs on advanced-architecture machines. This work has involved devising a way to identify data dependencies between blocks of code in Fortran programs and making this information available to the programmer and to the transformation tools in the software collection. The data dependency analysis is conducted interactively between the tools and the user; when a tool discovers a possible dependency it cannot resolve, it asks the user for clarification. One of the tools in the collection uses the data dependency information to determine whether different iterations of a DO loop are independent and thus can be parallelized. Some of the automatic transformations carried out by the tools are designed to increase the number of arithmetic operations in DO loops performed per access to main memory (a procedure that often improves performance on vector machines). Other transformations reduce the number of dependencies by making certain substitutions. Preliminary timings have shown that significant speedups (factors of 6 to 8) can be obtained on the Sequent b21.

Figure 2. Performance of the elastic wave code EWAVE. The solid line shows measured speedups; the dashed line is the 100% efficiency curve.
2.3. Parallel Programming Methodologies  
*J. J. Dongarra, E. L. Lusk, R. A. Overbeek, and D. C. Sorensen*

In 1985 we developed a package called SCHEDULE for writing explicit parallel programs in Fortran. During the past year, we extended the package and implemented it on the Alliant, Sequent, and Encore machines of the ACRF, the VAX, Sun workstations, the CRAY-2, Flex, NCUBE, Ultra Computer, and IBM VMS systems. A significant addition to the original package is a postprocessing graphical analyzer, which allows a user to visualize the dynamic parallel behavior of a program.

We updated our "monitors and macros" package for programming shared-memory multiprocessors portably, and added to the package a message-passing capability which allows portable programming of the local-memory model of computation. In collaboration with several colleagues, we prepared a book *Portable Programs for Parallel Processors*, describing a complete set of parallel-programming tools for multiprocessors of various computational models. The book discusses the methodology in general, together with examples using the specific tools, and includes diskettes containing the software itself.

We have also been developing a visual debugging tool for message-passing parallel programs. This tool is intended for programmers using Sun workstations.

In collaboration with B. Glickfeld of Northern Illinois University, we considered the issue of how to use a set of shared-memory multiprocessors to work cooperatively on a single task. As an initial experiment, we wrote a program that performed a number of straightforward state-space searches spread over multiple machines. In the most impressive test, we succeeded in using 50 processors spread over four machines (2 Encore Multimaxes, a Sequent b8, and a Sequent b21) to achieve a speedup of about 36; the effect was to reduce the actual time to perform the computation from almost 3 days to just over 2 hours.

In another project, in collaboration with S. Wallace from the University of North Florida, we wrote a program that gets excellent speedups merging two sorted lists. Using that algorithm, we were able to write a program that implements a merge-sort efficiently. The program is of interest as an example of where the use of monitors might offer substantial performance advantages over other approaches. We intend to examine these issues over the next several months.

2.4. Programming Languages and Dialects  
*K. W. Dritz, E. L. Lusk, and R. A. Overbeek*

The implementation of dialects of logic programming on multiprocessors is a topic that has gained worldwide attention. We implemented a version of the Warren Abstract Machine (WAM) for use on shared-memory multiprocessors. This implementation was used to construct an OR-parallel implementation of Prolog, which in turn was used to investigate a number
of critical issues. As a continuation of this effort, we participated in the Gigalips Project (a joint effort including Argonne, the University of Manchester, Imperial College London, and the Swedish Institute of Computer Science). During 1986-87 we initiated the creation of a higher performance OR-parallel Prolog, called Gigalips Prolog.

In connection with the OR-parallel Prolog work, we implemented a graphics program to trace the execution of individual parallel programs. This package has proven invaluable in enabling us to understand, debug, and tune the OR-parallel logic programming system itself.

In a related piece of work, J. Crammond from Imperial College visited Argonne to run his shared-memory implementation of Parlog on our Sequent and our Encore. The results were excellent and exceeded our expectations. We hope to have Crammond visit again, along with others from Imperial College and the Weizmann Institute, to generalize this early work.

We also completed the restructuring of one of our Ada tasking programs to serve as a suitable benchmark for assessing the efficiency of a vendor’s implementation of the Ada rendezvous mechanism. The program revealed one implementation of Ada tasking on a multiprocessor of the ACRF to be relatively naive in its current state of development. More advanced implementations, those embodying what is known as “Habermann-Nassi optimization,” will find this program to be an excellent test of the success of that optimization. Accordingly, we have offered the program to vendors of multiprocessing implementations of Ada for testing purposes. Three vendors are currently using the program.

Figure 3. A parallel logic program in action
2.5. Automated Deduction  
(*E. L. Lusk and R. A. Overbeek*)

We began implementing an equality-based deduction system. We expect that the system will be completed, in its original form, during early 1988. It will utilize a set of loosely coupled machines, each running the Gigalips Prolog. Our hope is that this project will produce not only a better equality-based theorem prover, but a new set of software tools for implementing of deduction systems on multiprocessors.

2.6. Publications and Presentations

Listed below are publications and presentations relating to advanced computing research carried out by Mathematics and Computer Science Division staff. This list updates the lists given in previous progress reports [1,2].

*Publications*


Reports


W. R. Cowell and B. S. Garbow, "Users' Guide to Toolpack/1 (Release 2) in a Unix Environment," ANL-87-12 (March 1987)

K. W. Dritz and J. M. Boyle, "Beyond 'Speedup': Performance Analysis of Parallel Programs," ANL-87-7 (February 1987)


Technical Memoranda


W. J. Cody, "ELEFUNT Test Results under NS320000 FORTRAN V2.5.3 on the Sequent Balance," MCS-TM-80 (September 1986)

W. R. Cowell, "UNIX Shell Scripts to Invoke a Set of Toolpack/1 Tools," MCS-TM-77 (July 1986)


Oral Presentations


3. User Facility Activities

The ACRF is intended to be a national user facility focused on carrying out advanced computing research. Toward this end, the Mathematics and Computer Science Division encourages use of the ACRF's specialized resources. During the past year, users from all over the world have taken advantage of free access to the machines to conduct a wide variety of research projects.

3.1. Projects

The following list is representative of the wide range of research projects pursued by recent ACRF participants.
Parallel algorithms for
  bin packing problem
  combinatorial problems
  computational fluid dynamics
  finite element structural analysis
  linear algebra
  number theoretic quadrature rules
  oil reservoir simulation
  stochastic linear programming
  subsumption
  unitary eigenvalue problem

Graphics
  Parallel graphics
  Partitioning program graphs for execution on multiprocessors

Tools for analyzing and debugging parallel programs

Methods
  Banded systems of linear equations in parallel
  Design of the Level 2 BLAS
  Design of the Level 3 BLAS
  Linear Algebra Package
  Parallel enhancement mechanisms
  Parallel implementation of multifrontal schemes
  Parallel methods to solve the least squares problem
  Pipelined Givens method for the QR factorization of a sparse matrix
  Sparse matrix operations on parallel architectures

Timing studies
  Incremental detection of synchronization errors
  The NAG library for multiprocessors
  Resource allocation within a prototypical linear algebra library

Physics problems
  Analysis and reconstruction of auditory signals
  Monte Carlo simulation of electron showers
  Numerical modeling of turbulent transport problems
  Properties of electronic states in solids

Chemistry problems
  Explicit finite element structural codes on the hypercube
  Interaction of a GUT magnetic monopole with a hydrogen atom
  Modifying large quantum chemistry codes with new algorithms
Results from many of these projects have already been presented at conferences and reported in technical reports. Among those from outside users we note the following:

**Conference Presentations**

Yi-Ling Chiang, "Mathematical Switches to Solve Differential Equation Problems," at the Third International Conference on Parallel Processing, December 1987


**Books and Papers Citing Argonne's ACRF**


Yi-Ling F. Chiang, "Use of Mathematical Switches to Solve Differential Equation Problems," Research Report 13, New Jersey Institute of Technology


"Exploring Supercomputing," Horizons 16, no. 2 (Summer 1987) 14


F. B. Hanson, "Computational Dynamic Programming for Stochastic Optimal Control on a Vector Multiprocessor," preprint

T. C. Oppe and D. R. Kincaid, "Numerical Experiments with a Parallel Conjugate Gradient Method," Report CNA-208, Center for Numerical Analysis, University of Texas at Austin, April 1987


R. J. Plemmons, Student reports prepared for class offered at North Carolina State University entitled "Parallel Algorithms and Supercomputing," Spring 1987. Twenty-two students and three faculty members participated; the ACRF machines were used.

L. Reichel, "A Matrix Problem with Application to Rapid Solution of Integral Equations," Report No. ICM-86-009, University of South Florida, 1986 (the numerical experiments were carried out here)


3.2. Proposals

As classes and workshops familiarize researchers with the new machines in the ACRF, potential users propose new applications and techniques to implement on our advanced computers. Over the past year, proposals from more than 110 research proposals have been submitted and accepted.

The first step in obtaining access to the high-performance computers of the ACRF is to submit an informal proposal to the reviewers in the MCS Division. Interested researchers should contact the ACRF scientific director for more information on the submission of proposals.

3.3. University Classroom Users

Several universities are using the ACRF machines in their classroom instruction on parallel computing techniques. These universities include

- North Carolina State University, in a course entitled "Parallel Algorithms and Supercomputing," taught by R. J. Plemmons.

- The University of Illinois at Chicago, in a Workshop Program on Scientific Computing, under the sponsorship of Floyd Hanson.

- The University of Illinois at Urbana, under Ahmed Sameh.

- The University of Tulsa, under the direction of Julio Diaz.

Additionally, two universities have submitted proposals to use the ACRF for classroom instruction:

- The University of California at Riverside, under the direction of John De Pillis

- The University of Illinois at Urbana, under Joseph Hardin

4. User Education

Argonne conducted nine classes on parallel computing during the past year. The attendees, averaging twenty per class, represented universities, industry, and various research laboratories throughout the country. The intent of the classes was to familiarize the attendees with the ACRF environment, to offer ample hands-on experience on the parallel computer systems, and to apply parallel programming to each attendee's area of research. During the
classes, the attendees were taught how to write and run several programs, with Fortran and C being the primary programming languages. Session topics that were addressed included monitors and their implementation with macros, the SCHEDULE package, and compiler directives on the Sequent and Alliant systems and the Intel hypercubes.

We plan to continue these classes in 1988. Tentative dates are March 2-4, April 27-29, June 15-17, July 27-29, September 21-23, and November 2-4. Researchers interested in attending one of these classes should contact the ACRF scientific director.

We also sponsored two workshops—one on performance evaluation of parallel computers and programs, the other on the Level 3 BLAS—and held a two-week institute in parallel computing for graduate students and post-doctoral researchers. The response to these workshops and the institute was extremely positive, and we hope to sponsor similar efforts during the next few years.

4.1. Visitors

The MCS Division invites scientists from industry, universities, and other research laboratories to participate in the various ACRF research projects. Graduate and undergraduate students, postdoctoral candidates, and faculty stay for periods of time ranging from two weeks to several months. During the past year, we have had numerous visitors here working on projects involving computer graphics, high-performance algorithms, and software tools for advanced-architecture computers:

K. A. Ariyawansa - Washington State University
Parallel schemes to approximate values and subgradients of the recourse function in certain stochastic programs

Ralph Butler - University of North Florida
Study of the "indexing problem," the problem of rapidly extracting logical formulas with specified properties from a large database

Ralph Byers - North Carolina State University
Implementation and analysis of variants of the QR algorithm for solving the algebraic Riccati equation through a Hamiltonian-Schur decomposition method

Kathryn Connolly - Northwestern University
Implementation of SCHEDULE and the new graphics trace facility on the Encore Multimax and the Sequent b21; design of two preprocessors that take an Alliant SCHEDULE code as input and produce as output a code ready for the Encore or Balance; implementation of SCHEDULE on netlib
Julio Diaz - University of Oklahoma
  Development of parallel methods for solving linear systems of equations arising from convective-diffusion PDEs; implementation in parallel with the use of SCHEDULE

Terry Disz - Illinois Benedictine College
  Development of graphics algorithms

Floyd Hanson - University of Illinois at Chicago
  Evaluation of various implementations of parallel algorithms

Rolf Hempel - G.M.D., West Germany
  Implementation of a communications library for parallel multigrid algorithms for the hypercube; design of parallel multigrid codes and software tools to support their semi-automatic parallelization

Kirk Jordan - Exxon Engineering & Research
  Development of performance evaluation methods for parallel and vector architectures

Nicholas Karonis - Northern Illinois University
  Applications of automated reasoning technology to expert systems of practical use, in particular, for netlib and for managing phase-diagram calculations

Jim Kohl - Purdue University
  Debugging of the program used for transferring files to the MFENET from the ACRF; rewriting of the file system used to send mail to the MFENET; installation of INGRES; debugging of problems with sockets and systems administration

Ramamohanarao Kotagiri - University of Melbourne, Australia
  Study of the usefulness of distributed array processors such as the DAP for performing indexing operations for clause selection

Robert Olson - University of Illinois at Urbana
  Study of synchronization issues in the new implementation of WAM

Jim Patterson - Eastern Kentucky University
  Systematization and documentation of a monitor/macro package; writing local enhancements to the ACRF systems as needed
Nathan Pfluger - University of South Florida
  Design of a Prolog database containing information about the structure of protein molecules

Margaret Purtill - University of California at Berkeley
  Implementing on the Intel hypercubes Argonne’s monitors-macros package

Dan Ross - University of Texas
  Generalizing the debugger in ANL-WAM; design of software to be used in a high-performance equality-based theorem prover now under development

R. C. Schmitt - MCT Division, Argonne
  Development of COMMIX-1C tailored to run in a combined workstation and parallel processor environment

Kish Shen - Cambridge University
  Benchmarking of the Argonne parallel Prolog; generation of Prolog search trees to test certain codes to be used on the new implementation of parallel Prolog for the Gigalips project

Chris Thompson - AERE Harwell
  Research on numerical methods for solving PDEs; study of Fortran transformations that enhance program performance on vector architectures; design of numerical algorithms for advanced computers

John Van Rosendale - University of Utah
  Study of the BLAZE language for expressing parallel algorithms

Stephen Vavasis - Stanford University
  Design of parallel and vector algorithms for numerical linear algebra computations; study of the solution of PDEs on high-performance computers

T. C. Wang - University of Texas at Austin
  Implementation of an equality-based inference system for multiprocessors

Robert White - North Carolina State University
  Study of parallel iterative methods for the numerical solution of PDEs
4.2. Seminars

For the past several years MCS and Computing Services have jointly sponsored a series of seminars on high-performance computing. Below are listed the names of our most recent speakers, their affiliations, and the title of their talk.

J. B. Rosen and A. T. Phillips  
*Multitasking and Algorithms for Constrained Optimization*  
University of Minnesota  
October 2, 1986

Lee Higbie  
*The CHoPP Parallel Processor*  
Sullivan Computer Corporation  
October 13, 1986

Alvin Bayliss  
*Numerical Analysis of a Solid Combustion Model Using an Adaptive Pseudo-Spectral Method*  
Northwestern University  
October 16, 1986

Robert White  
*Parallel Iterative Methods and Multisplittings*  
North Carolina State University  
November 6, 1986

Howard C. Elman  
*Approximate Schur Complement Preconditioners for Serial and Parallel Computers*  
University of Maryland  
November 14, 1986

Burton Smith  
*Processor Architecture for a Shared Memory MIMD Computer*  
Institute for Defense Analysis  
November 19, 1986
Sanzheng Qiao
Recursive Least Squares Algorithm for Linear Prediction Problems
Cornell University
November 25, 1986

Robert Plemmons
Algorithms and Experiments for Structural Optimization on High Performance Architectures
University of Illinois at Urbana
December 2, 1986

Elizabeth Jessup
Solving the Symmetric Tridiagonal Eigenvalue Problem on the Hypercube
Yale University
December 9, 1986

Adam Bojanczyk
Parallel Algorithms in Linear Algebra
Washington University
January 6, 1987

William R. Martin
Monte Carlo Photon Transport on Advanced Computers
University of Michigan
January 8, 1987

Thu V. Vu
Link Scheduling Protocols and Routing Algorithms for a Packet-Switched Network
Harris Corporation, Govt. Aerospace Systems Div.
January 9, 1987

Wayne J. Davis
Online Production Scheduling Using Parallel Computation
University of Illinois at Urbana
January 22, 1987
Karen Thompson
*Asynchronous Parallel Successive Overrelaxation Algorithm*
University of Wisconsin
March 3, 1987

G. W. Stewart
*Communications in Parallel Matrix Computations*
University of Maryland
March 23, 1987

James Solinsky
*The SAIC Σ–1 © Neurocomputer*
Scientific Applications International Corporation
May 11, 1987

Youcef Saad
*Numerical Methods for Large Matrix Eigenvalue Problems*
Center for Supercomputing Research and Development,
University of Illinois
May 26, 1987

Rolf Hempel
*The SUPRENUM Communication Subroutine Library for PDE Application Software*
GMD, West Germany
June 4, 1987

Robert Schmitt
*COMMIX in an Engineering Workstation Environment*
Materials and Components Technology Division, Argonne
August 20, 1987

Roger Hockney
*Benchmark Measurements on MIMD Computers*
Reading University, UK
August 27, 1987
Christian Bischof
*The Merits of Block Algorithms with Examples for OR and SVD Decompositions*
Cornell University
September 11, 1987

John VanRosendale
*Synchronized Packet Interconnection Networks*
University of Utah and ANL
October 15, 1987

Bo Kagstrom
*Parallel Algorithms for Solving the Tridiagonal Sylvester Equation*
University of Umeå, Sweden
October 26, 1987
Appendix - Meetings, Workshops, and Conferences


Forty-six people attended a two-day workshop at Argonne designed to give researchers in the field of parallel computing an opportunity to exchange viewpoints, approaches, and results in the area of performance evaluation of parallel computers and programs. Among the topics discussed were performance measurement by instrumentation, analysis of real-time trace events, and effects of program structure and programming methodology on performance.


Two dozen mathematicians and computer scientists from the United States and England attended a two-day workshop at Argonne, to study topics relating to linear algebra software. In particular, the attendees discussed expanding the Basic Linear Algebra Subprograms to exploit the features of advanced computers, and algorithm implementations on today's high-performance computers.


Twenty-five graduate students and postdoctoral researchers were selected out of 80 applicants to participate in a two-week institute at Argonne sponsored in part by NSF and in part by DOE. Invited speakers included Gordon Bell, Bill Buzbee, Josh Fisher, David Kuck, Neil Lincoln, Chuck Seitz, Larry Smarr, Burton Smith, and Guy Steele. Each invited speaker was given the responsibility of organizing the three-hour morning session into any format he desired. The afternoon sessions were organized each day by a member of the Argonne research staff. These were "hands-on" sessions, in which the attendees used the ACRF machines. A variety of tutorial material was presented, with supervision of student exercises.
References


Distribution for ANL-87-50

Internal:

C. Adams
J. M. Beumer (3)
T. Fields
F. Y. Fradin
H. G. Kaper
A. B. Krisciunas
G. W. Pieper (1450)
J. Unik
D. Weber

ANL Patent Department
ANL Contract File
ANL Libraries
TIS Files (3)

External:

DOE-TIC, for distribution per UC-32 (112)
Manager, Chicago Operations Office, DOE
Mathematics and Computer Science Division Review Committee:
  J. L. Bona, Pennsylvania State U.
  T. L. Brown, U. of Illinois, Urbana
  P. Concus, LBL
  S. Gerhart, MCC, Austin, Texas
  H. B. Keller, California Institute of Technology
  J. A. Nohel, U. of Wisconsin, Madison
  M. J. O’Donnell, University of Chicago
  D. Austin, DOE-ER
  R. Barrow, DOE OADPM
  D. Baugatz, Alliant
  W. Buzbee, NCAR
  C. Gordon Bell, DANA
  M. Ciment, NSF
  J. Connolly, University of Kentucky
  D. Cole, Intel
  K. Cramer, DOE-CH
  F. Darema-Rogers, IBM
  J. Decker, DOE-ER
  K. Foote, Sequent
  A. Hayes, LANL
  D. Hitchcock, DOE-ER
  K. Hopper, Alliant
  R. Huddleston, LLNL
A. Karp, IBM
T. Kitchens, DOE-ER
M. McNeill, Encore
G. Michael, LLNL
D. Micciche, Alliant
C. Moler, DANA
C. Mundie, Alliant
D. Nelson, DOE-ER
C. Oliver, Kirkland AFB
R. Parsons, Sequent
L. Petzold, LLNL
J. Pool, NAG U.S.A.
G. Ringstad, Encore
F. Ris, IBM
M. Scott, DOE OADPM
C. Thomas, DOE OADPM
R. Ward, ORNL