DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DOE/ER/25001--T6

DE92 000032

CEDAR PROJECT — ORIGINAL GOALS AND PROGRESS TO DATE

Progress Report

For Period January 25, 1991–January 24, 1992

G. Cybenko, D. Kuck, D. Padua, and E. Gallopoulos

Center for Supercomputing Research and Development
University of Illinois at Urbana–Champaign
305 Talbot
104 S. Wright Street
Urbana, IL 61801

September 5, 1991

Prepared for

THE U.S. DEPARTMENT OF ENERGY
AGREEMENT NO. DE-FG02-82ER25001

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Table of Contents

Introduction	4
Cedar System Hardware	5
Performance Monitoring Hardware	5
Cedar Memory Upgrade	5
Cedar Performance Characterization	6
Cedar Performance Analysis of the Perfect Codes	7
Cedar MTBF Analysis	7
Cedar Diagnostics	9
Cedar Hardware Tracing Support and Kernel Instrumentation	10
Architecture	11
<u>Compiler and Software Issues on Memory Management</u>	11
Cache Performance in Multiprocessor Systems	11
Software Coherence Scheme with Assistance of Directories	11
Memory System Performance Analysis	12
Detecting Redundant Accesses to Array Data	12
An Integrated Hardware/Software Solution for Effective Management of Local Storage in High-Performance Systems	12
Compiler-Directed Data Prefetching in Multiprocessors with Memory Hierarchies	13
Software	13
<u>Operating Systems</u>	14
Single User Batch	14
XPS – System and Process Status Reporting	14
Parallel Debugging	14
Cedar Console and Diagnostic Station	15
<u>Compilers</u>	15
Cedar Fortran Compiler	15
Multilingual Compilation with MIPRAC	17
Interactive Compilation	18
Sparse Matrices	18
Studies in Scheduling	19
Compiler-Generated and Optimized Synchronization for Data Dependences	20
Multiprocessor Performance on Algorithms and Applications	21
Domain Decomposition Method for Structural Analysis	22
Circuit Simulation	22
Ocean Simulation	22

Computational Fluid Dynamics	23
Sparse Matrix Computations	23
Direct Sparse System Solvers	23
Iterative Sparse System Solvers	24
Wavelets	24
<u>Perfect Benchmarks</u>	24
Workload Analysis and Perfect Benchmarks Verification	25
Perfect Version 2	25
Instrumentation of the Perfect Codes	25
Perfect Performance Database	26
<u>Performance Environments and Tools</u>	26
Cedar Performance Environment	26
Parallel Performance Advisor (PPA)	26

Introduction

Since its founding in 1984, CSRD has become a highly productive force in the supercomputing research community. Under the leadership of Director David J. Kuck and Associate Directors David Padua and George Cybenko, approximately 135 faculty, staff, and students engage in cooperative research in high performance computing. In a relatively few short years, CSRD has produced a number of innovative research results. In addition to Cedar, our pioneering efforts include the BLAS 3; the parallelizing LISP compiler, Parcel; the Perfect benchmarking effort; the restructuring compiler prototyper, Delta; the Maxpar tool for estimating parallelism limits; and a fundamental theorem of neural nets, among others.

Graduate education is a high priority at CSRD; to date, 30 PhD and 60 MS degrees have been awarded to students who have done graduate research at CSRD through the Departments of Computer Science, Mathematics, and Electrical and Computer Engineering. Many of these graduates have joined industrial firms in high performance computing, while a number of others hold faculty positions at major universities.

In addition to extensive research and development activities, CSRD contributes to the academic and industrial supercomputing community at large through a series of meetings, seminars, and workshops. In 1990, for example, CSRD sponsored 51 seminars by outside speakers, a summer workshop on parallelizing compilers, and a two-day workshop for undergraduate engineering students at area colleges and universities who are interested in graduate work in computational science and engineering computing.

The corporate world participates in the research at CSRD through the CSRD Affiliates program, which currently includes 10 members, among them Cray Research and IBM. Affiliate members, who contribute substantial amounts of both funds and equipment to support the research activities at CSRD, learn about the results of CSRD activities through an extensive series of publications that includes technical reports, newsletters, and an annual research review. Affiliates also attend the two-day annual meeting held each spring, at which faculty, staff, and students present their work and demonstrate the most recent achievements of the group. In August 1991, CSRD hosted the SPEC Benchathon VI, a week-long benchmarking effort by 21 members of SPEC, the System Performance Evaluation Cooperative.

CSRD provides its researchers with substantial equipment resources, including an 8-processor Alliant FX/80, an Alliant FX/1, two single processor Alliant FX/8s, a MIPS M/120, a Sun 4/490, a Pixar Image computer, several Sun file servers, 80 Sun workstations, and 20 Macintoshes and PC's. In addition to the 32-processor Cedar System which is now in use, we have received a 28-processor Alliant FX/2800, which is used for performance studies as well as service computing. In the future we plan to install other parallel machines for comparative performance studies.

The sections that follow describe in detail the work accomplished during 1991 and outline the goals and objectives for 1992.

Cedar System Hardware

The 32-processor Cedar integration was completed in December 1990 [Sta91]. This machine, the final phase of Cedar 1 hardware development, supports 32 CEs, 8 CEs in each of four clusters. The machine also has 32 global memory boards (GMB) of 2 Mbytes each. The interface between processor and global hardware is provided by the CSRD-designed global interface board (GIB). The GM and CEs (via GIBs) are connected via two unidirectional, 32-port Omega packet switched networks. Each global network board (GNB) in the network is an 8 by 8 crossbar switch. Global memory is also 32-way interleaved and supports the set of Cedar 1 synchronization primitives.

The machine has been quite stable since its release in January 1991. Thanks to the completed diagnostic hardware, any faults that do occur are quickly isolated and repaired.

Performance Monitoring Hardware

Performance monitoring hardware development is ongoing and, to date, significant instrumentation of Cedar has been accomplished. The GIBs, GNBs, and GMBs were all designed with "hooks" to attach performance gathering hardware. CSRD has designed Tracer and Histogrammer boards to fulfill performance monitoring needs.

The Tracer and Histogrammer boards are designed around the industry standard VME bus system and reside in cabinets strategically placed around the Cedar hardware. The Tracer board is capable of tracing 20 bits of data with a depth of 1 million samples and can acquire data at a rate of one sample per 170 nanoseconds. The Histogrammer board is capable of histogramming 64K different events at a rate of one sample per 170 nanoseconds. Both of these board types can be interleaved to support higher acquisition rates. A Sun processor board resides within each VME chassis with the performance evaluation (PE) hardware to extract the data from the PE boards and to provide analysis capabilities as well as an Ethernet connection to the other computers at CSRD.

Currently, 12 Tracer boards are extant in Cedar, providing 36 channels of trace capability. Histogramming hardware is in manufacture and will be ready to attach to Cedar in late September 1991, providing 32 channels of histogramming capability.

Cedar Memory Upgrade

CSRD is currently involved in a project to increase extensively the size of Cedar memory. Dubbed Cedar 1.5, this upgrade of Cedar can support a GM size of 2 GBytes and can go as high as 4 GBytes with a modification to each CE. This upgrade will allow CSRD to run extremely large codes on Cedar.

To take advantage of improvements that CSRD has already learned about from the Cedar machine and to attack the packaging problem associated with large machines, we will also enhance the GIB/network system. CSRD is redesigning the Omega network and designing a small daughter board for each GIB. The GIB daughter board will facilitate interface to the new network/memory system as well as provide blocking/deblocking of vector requests to enhance the system throughput in vector

mode.

To attack the packaging issues, CSRD is modifying the inter-board interface. Currently, boards communicate with each other via a 64-bit data path with bitwise parity protection and 4 control signals. This information is transmitted differentially every 85 nanoseconds, which amounts to 3 cables of 60 wires (180 wires total) for each port. CSRD will change the interface to an 8-bit interface running at 10 nanoseconds. This maintains the effective interface bandwidth while greatly reducing the number of cables and wires per cable.

CSRD will also be adopting a number of strategies that will improve the throughput of the shared memory and network system. We will increase the queue depth at each point of the system. The memory along with the GIB daughter boards will support block reads of 2 or 4 64-bit words from memory during vector prefetch. Each network board will have its queueing along each switch path rather than on input or output ports only, which serves to reduce significantly the blocking resulting from heavy traffic, because any full queue blocks only one input-to-output path and allows an input wishing to go to another output to pass. (A 4 by 4 switch, for example, is said to have 16 input-to-output paths.) The interface described above will help to improve latency because the finer grained transmission of control information will allow action to start before all control and data information is present at a board (e.g., a GMB can start a write cycle before it knows the address or data to write). Our simulations show that the enhancements described above can provide roughly twice the throughput, compared to the current Cedar machine.

The enhanced memory/network system will use existing Cedar physicals to keep the design cycle as short as possible. This, of course, has the added benefit of reducing cost and manpower requirements. CSRD will use board form factors approximately half the size of current boards (new form factor approximately 11 by 18 inches) to enhance manufacturability. Thus it will be possible to put 2 new boards where one now exists.

To date, the architecture of the enhanced memory has been completed. Board design is quite far along, with schematic capture close to finished. Simulation of the designs began in early August 1991. Board manufacturers and loading houses have been identified, and the project is proceeding on schedule.

Cedar Performance Characterization

Work is underway to characterize the performance of the Cedar memory system and its interaction with the control structures in the Cedar Fortran runtime library. This effort is a generalization of previous work at CSRD to characterize and predict the performance of a single Cedar cluster (the Alliant FX/8). An extensive empirical database that quantitatively characterizes the memory hierarchy has been accumulated on the 32-processor Cedar. It also demonstrates clearly the effectiveness of the prefetch unit in reducing the impact of global memory latency and providing data to all four clusters at rates that approach, and under certain circumstances exceed, those of cluster memory. When a subset of the clusters access global memory, the rates using the

prefetch unit can approach the observed cluster cache performance. This characterization will be used to predict and tune the performance of some codes on Cedar; it will also serve as calibration data for simulators used in a scalability study by the Architecture group.

Cedar Performance Analysis of the Perfect Codes

The Compiler group and the Applications group have begun a cooperative effort to port, benchmark, analyze, and improve the performance of some of the Perfect Benchmarks application codes. Twelve of the codes presently run on Cedar with varying degrees of performance. The major goal of this project is to identify the potential compiler transformations needed to improve the performance of the codes and to assess their automatability. The architectural characterization and algorithm design work of the Applications group will be applied to a subset of the code in order to explore the details of the architecture/application mapping and the strengths and weaknesses of the Cedar architecture.

Cedar MTBF Analysis

The Cedar system full configuration (32 processors and 32 memory boards) became operational in December 1990. Hardware and system software testing was performed throughout January, and the system was released to the Cedar user community on January 29, 1991. Since then, data have been gathered relating to up/down time, number of crashes, and cause of each crash.

Table 1 shows the number of hours the system was up and down from January 30 through July 20. Up time (the first column) counts only those hours when all four clusters were up and the system was fully operational. If any one cluster crashed or hung, Cedar was considered down, even if the remaining clusters continued to be used. The down times are categorized according to the causes for which the system went down. In almost every case, the cause of the crash was clear. When the cause wasn't clear, it was most probably either hardware or software. The "other" category includes crashes and down times caused by power outages, air conditioning trouble, and Sun network problems that affected Cedar. The "maint" category includes system preventive maintenance, diagnostics, and reconfiguration. This category was added in April, when it became clear that maintenance should be accounted for separately. (Before April, system testing was included in the other down times.)

	Uptime		Downtime							
	hours	(%)	Hardware hours	(%)	Software hours	(%)	Other hours	(%)	Maint hours	(%)
February*	529	(73)	121	(17)	48	(7)	22	(3)	0	(0)
March	541	(73)	101	(13)	53	(7)	49	(7)	0	(0)
April	452	(63)	167	(23)	47	(7)	1	(0)	53	(7)
May	476	(64)	167	(22)	24	(3)	20	(3)	57	(8)
June	485	(67)	84	(12)	9	(1)	87	(12)	55	(8)
July	543	(73)	110	(15)	14	(2)	56	(8)	21	(3)
Total	3027	(69)	750	(17)	194	(4)	235	(5)	186	(4)
* includes Jan. 30, 31										

Table 1. Cedar Up and Down Times (in hours and percentage of total time).

In general, the amount of down time caused by a crash is unrelated to the cause of the crash. In fact, the down time for a hardware crash tends to be long while that for a software crash tends to be short, possibly because the hardware often crashes at night when there is nobody available to reboot the system, so the system remains down for long periods when it could be brought up right away. Software crashes tend to occur during the day when somebody is running a program and watching it. The user will often notify an administrator when a crash occurs, and prompt attention results.

The data for mean time between failure (MTBF) are shown in Table 2. The number of crashes is categorized by cause, while the MTBF column is based on the total number of crashes. The overall MTBF for the 179-day period was 13.7 hours. The longest uptime was 114 hours, from May 17 to May 22.

	Uptime (hours)	Number of Crashes				MTBF (hours)
		Hardware	Software	Other	Total	
February	529	15	17	3	35	15.1
March	541	22	22	5	49	11.0
April	452	44	12	1	57	7.9
May	476	13	5	2	20	23.8
June	485	17	6	4	27	18.0
July	543	23	7	1	31	17.5
TOTAL	3027	134	69	16	219	13.8

Table 2. Cedar Mean Time Between Failure

The number of hardware crashes increased in April because of intermittent problems in the network. When the MTBF dropped to unreasonable levels, individual hardware failures were not treated as a crash; rather, a series of quick crashes was treated as a single problem and counted as one crash. Connecting probes or wiggling connectors would generally solve these problems. After enough information was collected from the diagnostic processors, it became clear that practically all of the hardware crashes were caused by a small number of bad solder joints or connectors. From May through July, these intermittent problems continued to occur and were found to be related to a connector pin, which was re-soldered on July 23rd.

Software bugs that accounted for many of the crashes through April were fixed, and there was a marked improvement in this category starting in May. Two more major bugs were fixed in July, and the system had no crashes caused by software from July 10 to July 27. With a number of recurring hardware and software problems now solved, we hope to see improved uptime percentages and much better MTBF figures.

Cedar Diagnostics

Two Cedar hardware diagnostics have been developed in order to test general reliability and to help pinpoint quickly problems in the networks or global memory. The first of these was designed to find intermittent errors in memory modules or the return network. The second was built to put a heavy traffic load on the network and check for routing or data errors in either the forward or return network.

The first diagnostic works by filling a page with a specific value; then each processor repeatedly reads it back (in parallel), checking the values. This program runs under Xylem, so the actual physical page used is assigned by the operating system. As such, it does not check the memory itself, but because of the way memory is interleaved, the page offset where bad data are found points at a specific memory module or return network path.

A more sophisticated diagnostic was needed to load down the network with output and input traffic, and also to pinpoint routing errors and other problems. This diagnostic does not run under Xylem and can test specific areas of the global memory. It works by having each processor (CE) independently write a vector, then optionally read it back and check it. The vector length is fixed at 1024, but the starting physical address, stride, and data pattern can be entered interactively. If no data pattern is given, the diagnostic writes the physical address of each element into itself, along with a processor ID and sequence counter. This variable pattern provides a general routing test.

Processors can also be selected, so that some may sit idle. The processor mask, starting address, and stride can be used to control which paths and memory modules actually get exercised. Data can be read back and checked with either scalar or vector instructions; if vector instructions are used, Cedar prefetching can be enabled or disabled.

This second diagnostic has proved to be a more useful general test; in addition, it can select subsets of the network or memories, down to a specific path and memory module. However, this diagnostic is somewhat cumbersome for general testing because it requires shutting down the operating system and it interactively requests a handful of parameters. For this reason, we have designed a special-case version which runs under Xylem, and defaults all parameters except the processor mask. This version cycles through a sequence of data patterns instead of using just one, in order to check for stuck-at and coupling faults.

Cedar Hardware Tracing Support and Kernel Instrumentation

The goal of the Xylem tracing project is to design, develop, implement, and document a method for gleaning information concerning the Cedar operating system using Cedar tracing hardware. After the design and implementation of a complex parallel processing operating system such as Xylem, there are many unknowns. A good implementation of operating system event tracing attempts to address and resolve these unknowns. In addition, there are unforeseen benefits, such as parallel debugging, that are facilitated by tracing.

The project entails installing operating system tracing calls at key locations throughout Xylem. The desired trace event data are written out to a memory location, which triggers a hardware event that forwards them over a network to a trace analysis system designed by the Performance Evaluation group.

The Operating System group has been working continuously with the Performance Evaluation group and the Hardware group to instrument the kernel and garner the trace

events. Xylem is currently implemented with user-selectable tracing event calls that write data to the tracing hardware which are then interpreted by the Performance Evaluation group. The instrumentation of the kernel is not yet complete.

Architecture

Compiler and Software Issues on Memory Management

Cache Performance in Multiprocessor Systems

Directory and software schemes have long been considered the two major approaches to managing the coherence among private caches in multiprocessor systems [ChVe91]. However, the cache performance and the overall system performance of the two approaches have not been adequately evaluated and compared. The goal of this project has been to define a framework for realistic performance evaluation and to compare the above-mentioned cache coherence schemes.

Our approach to performance evaluation has been to define a multiprocessor system using MIPS R3000 processors with various cache organizations. Parallel Perfect Benchmarks codes were used as benchmarks. The codes were executed (serially) on a MIPS M-120 system, and a 32-processor memory reference trace was produced and used as input to a multiprocessor cache simulator.

The first phase of this work has been to measure the hit ratios and shared memory traffic of the two coherence schemes using otherwise identical system organization. The results show that the hit ratios achieved by the simplest software scheme are comparable to those of a distributed directory scheme. The overall memory traffic of the software scheme is also shown to be smaller. Given comparable performance and the fact that the software scheme requires very simple hardware, it is therefore preferable to the directory schemes. The second phase of this project, which is just starting, will measure the execution time of various cache systems through simulation.

Software Coherence Scheme with Assistance of Directories

A pure software coherence scheme relies on compile-time analysis of a program and sometimes has to make conservative decisions about validity of cache data which can lead to a loss of cache performance. Several hardware enhancements have been proposed to supply additional runtime information and thus improve the performance [ChVe91].

We defined a new coherence scheme in which a special directory is used to supply the runtime information and assist in the invalidation process. This scheme improves on the directory scheme by allowing write-sharing of cache lines, and it improves on the pure software scheme by allowing inter-epoch locality to be preserved. The performance of the new scheme was evaluated, and it was consistently shown to achieve equal or better performance than either the pure software or pure directory schemes.

Memory System Performance Analysis

The memory system of a large-scale multiprocessor consisting of forward and reverse Omega networks and an interleaved memory was studied under burst traffic as exhibited by vector units or caches. The scalability of such systems in the range from 32 to 512 processors was demonstrated through detailed simulation, in addition to presenting their performance characteristics under various traffic and architectural parameters [GrTV91].

We have shown that such a memory system can deliver to processors a full memory bandwidth when the number of processors equals the number of memory modules and the memory is "slow," i.e., 4 to 5 clocks versus single clock network transfers. Doubling the number of memory modules or reducing their cycle time was shown not to result in either proportionate or scalable performance improvements. The largest scalable improvement was shown to be achieved by storing data in a block-interleaved fashion and making memory requests in terms of such blocks.

Detecting Redundant Accesses to Array Data

In typical numerical applications, a significant portion of global memory data traffic arises from accesses to blocks of array elements or regions. Memory system delays for this traffic can be reduced by using compile-time information to detect when a locally available copy of a region can be reused, thereby eliminating redundant global memory accesses. A compile-time algorithm has been developed that applies combined flow and dependence analysis to programs with vector and parallel constructs to detect such redundancies across loops nests and in the presence of conditionals [GrVe91b].

An Integrated Hardware/Software Solution for Effective Management of Local Storage in High-Performance Systems

Local storage in the form of private caches or local memory in vector or multiprocessor systems is extremely important in reducing the memory access latency. Unfortunately, neither one of these types of storage is easy to use to its full potential. Effective management of such local storage, especially for array accesses, is essential to achieving high performance.

We propose a new local storage facility called a priority data cache that allows combined hardware and software control [GrVe91b]. Using compile-time knowledge of array usage, lines containing data that are going to be reused are not allowed to be replaced by lines with data less likely to be used; that is, an access is not cached at all if data in the cache that it would have replaced have higher priority. The decision is made by hardware at runtime based on the priority information computed by the compiler. Priority is given to the data that are going to be reused the soonest. We describe compile-time techniques to compute priorities easily and effectively and to utilize this type of local storage.

Compiler-Directed Data Prefetching in Multiprocessors with Memory Hierarchies

Block fetching and block prefetching are the most effective ways to move data between the levels of memory hierarchy. We propose compile-time algorithms for extracting block fetch or prefetch requests from Fortran loops and scheduling them as early as possible subject to data and control dependence constraints. Static analysis of a number of subroutines is performed to determine the fraction of references that can be accessed in these modes and how early can this be done [GoGV90].

We found that 58% of all array accesses can be prefetched at least with respect to the innermost loop in which they were nested, 22% can be prefetched with respect to all surrounding loops, and 3% can be prefetched at the beginning of a subroutine. Where prefetching was not possible, data dependences were involved in 74% and control dependences in 72% of the instances. (We only allowed unconditional prefetches.) In a number of instances, both types of dependences were involved.

We also performed detailed simulations, including network and memory behavior, of several subroutines to determine the effectiveness of block fetches and prefetches. When prefetching was overlapped with computation, the speedups ranged from 8 to 68% compared with no-prefetching/overlap cases. The largest improvement was obtained when we also allowed prefetch requests to be initiated before the earlier prefetches were completed. However, this leads to significant network and memory contention and can actually result in performance degradation. Further study is necessary to determine the number of outstanding prefetch requests that should be allowed.

Software

In this section we discuss some of the accomplishments of the Operating Systems and Compiler groups, including some facilities developed by the Operating Systems group to make Cedar easier to use, such as:

1. A facility to queue jobs and then dispatch them one at a time to allow the use of Cedar in single-user mode without human intervention.
2. A facility to display information on processes and tasks running on Cedar at a given time.
3. A software system to display and accept commands related to the console.

We also discuss our recent work on the Cedar Fortran compiler. During the past year our efforts have been concentrated on evaluating the effectiveness of the parallelization phase of the compiler and on trying to develop new parallelization techniques to overcome current limitations.

Several projects studying different aspects of parallelization are discussed.

1. The Parafrase-2 project, whose main objective is to study issues of program decomposition for scheduling.
2. The MIPRAC project, in which our goal is to study the interprocedural analysis of programs, including multilingual programs. To accommodate languages such as C

and Lisp, this project also deals with the very difficult problem of pointer analysis for parallelization.

3. Our work on automatic synchronization instruction generation and on compile-time cache management.

Operating Systems

Single User Batch

Many Cedar users are benchmarking or running large problems that require the use of the system in a single-user mode. However, most of these benchmarks and problems are run from scripts or otherwise can be done in background. It quickly became desirable to have a facility for submitting jobs to be run automatically overnight in a single-user environment. The implementation of single-user batch was simplified by starting with the MDQS software queueing facility, which is a public domain device queueing and batch processing system. It normally runs as a background daemon which accepts job submissions and control commands and schedules these jobs. For our single-user purposes, we modified it to run as a foreground program, process the job queue strictly sequentially, display the queue between jobs, and exit after the last job finishes. An undocumented hook in init allows us to have a system shutdown, automatically run an arbitrary script, and then restart multi-user operation when the script is finished.

XPS – System and Process Status Reporting

This project is to design, develop, implement, and document a method for gleaning information concerning Xylem processes (and tasks). This project was completed with the production of a new Xylem command called "xps". The command displays information about Xylem processes and tasks. Xps was modeled after the standard Unix ps command, which displays only process and task information for the particular cluster on which the command was executed. Xylem processes and tasks can spread across multiple Cedar clusters, and the xps command displays Xylem process and task information across clusters, and also displays crucial information about the Xylem operating system. Xps runs as a Concentrix process; thus it can provide information in some cases even when one or more other processes are hung. This work led to the development of several new methods for gathering information from the Cedar operating system.

Xps has benefited the CSRD community with design of the helper-task spreading algorithm by the Compiler groups, design of the operating system cluster balance task algorithm, global and cluster memory usage, hard and soft page faults, Cedar system load, detailed Xylem process and task information, and snapshot functionality without having to instrument application code with tracing software.

Parallel Debugging

A software package, *mdb*, has been developed for debugging parallel programs on Cedar. This package is almost complete at this stage and is available to the general

Cedar user community. Mdb comes in two versions, a standard version, which is used by linking a program with the `-ldebug` library, and a stripped-down version, which is part of the C library. The stripped-down version is automatically linked with every program, so users always get at least this level of debugging support and can simply consider it part of the operating system.

To the user, mdb is similar in nature to a conventional breakpoint debugger, such as adb. However, mdb knows about the multi-tasking model available in Xylem as well as the architecture of the Alliant multi-processor. If any task enters the debugger, either because of an error or by request, this task interrupts all the other tasks in the program as quickly as possible, using facilities available in Xylem. At this point the user may examine the entire state of the program, which includes shared memory, private memories, any of the registers in any of the processors, and certain system state variables.

Mdb differs from many interactive debuggers in that it is a package of functions that are linked with a target program rather than a separate process controls the target program. Thus, although mdb is not totally isolated from the target, it is relatively simple to implement and can examine large amounts of program state very efficiently because program state is directly accessible without requiring any operating system services.

Responses from the user community have generally been favorable, and it is known that a significant number of real bugs have been easily and quickly found once the faulty program was linked with `-ldebug`.

Cedar Console and Diagnostic Station

The project, which is to design, develop, implement, and document a console and diagnostic station for Cedar, was completed by developing a Cedar console running X-Windows on a workstation. The Cedar console program starts up the Cedar Master console, sets up the Cedar "master" environment, and initiates Cedar cluster monitor sessions. An X-Window is opened and a communication session initiated for each of the Cedar cluster monitor sessions. The Cedar master console window, which is used to run and operate individual diagnostic programs, multi-cluster console writing sessions, or information gathering sessions, is opened for users input.

Among the many advantages to using a windowing system in the implementation of the Cedar console are central diagnostic and cluster console control and logging, multi-cluster console writes, and standard X-Window features, such as multiple sessions.

Compilers

Cedar Fortran Compiler

Two main objectives of the original proposal pertained to the Cedar Fortran compiler project: (1) providing a high-quality compiler to the programmers of the Cedar system, and (2) conducting research on compilers and restructurors.

We proposed to do this by:

1. Providing a parallel dialect of Fortran – Cedar Fortran – and a corresponding compiler that generates efficient code for the Cedar system. One specific goal was to provide a way to spread work across Cedar quickly.
2. Providing Cedar with Fortran 77 to a Cedar Fortran Restructurer based on Kap, the commercial restructurer by KAI. To make the restructurer more efficient and usable, we planned to study several topics including:
  - Making the restructurer accept Cedar Fortran as input;
  - Adding loop-spreading and interprocedural analysis capabilities; and
  - Finding better heuristics to allocate variables in the Cedar memory hierarchy and tuning the compilation strategies as the Cedar system grows.

In our last report we described progress in a number of these areas.

1. We added a dispatching facility in the Cedar Fortran runtime library that was able to reduce the overhead to 40 microseconds when spreading work across Cedar. Also, we conducted experiments with the Cedar synchronization hardware for self-scheduling.
2. We reported on experiments with the Cedar Restructurer on algorithms and some real application programs from the Perfect Benchmarks suite. These studies helped us to identify some of the weaknesses of the current restructuring technology and indicated some new transformations that can overcome them. These experiments were run on the 16-processor Cedar machine. Early findings from these studies indicated that although current restructuring technology can demonstrate remarkable performance when applied to program kernels, there is little speedup with real application programs. However, there seemed to be significant potential for improvement by developing new and refined restructuring techniques.

Since January 1991 we have been expanding on these previously reported restructurer experiments and their verification on the full 32-processor Cedar machine. We have engaged in a major effort to optimize the Perfect Benchmarks codes on Cedar, first by automatic translation with Kap/Cedar and then by manual modifications, to gain additional program performance. In these manual experiments we have done a mechanical analysis of the codes in a way that is most likely to be automatable.

The results during the last six months have also been very encouraging. In half of the Perfect Benchmarks codes we obtained a significant speedup over the program variants restructured by the 1990 compiler. Some of these programs are still under investigation, and better numbers are to be expected. In general, we are confident that our findings remain valid beyond Cedar and Cedar-like systems.

Based on these results we are planning the next compiler implementation, which is necessary not only to make available and to evaluate the developed compiler technology at CSRD, but also to provide conclusive evidence of the automatability of the new transformations. An important complementary issue will be the appropriate architecture of a restructuring compiler to accommodate the newly developed capabilities.

Multilingual Compilation with MIPRAC

The MIPRAC system is now in an advanced stage of development. The system has complete front-ends for C, ANSI C, and Fortran 77; that is, these translators accept complete, standard dialects. The Common Lisp translator is not yet complete. It accepts all of the special forms of Common Lisp, and most of the built-in functions of the language. We are continuing to develop the Common Lisp front end.

Support for multilingual compilation in MIPRAC means that an input program is translated into the intermediate language MIL, a language- and machine-independent assembly language. The primitive data types in MIL are numbers, memory addresses, and procedures. The input program is, practically speaking, compiled by the translation into MIL, which means that the subscript computations, pointer displacements, casts, and alignments, are made explicit in the translation to MIL. There is a direct correspondence between the instruction set of a conventional CPU and the constructs of MIL. Parallelization in MIPRAC therefore operates on a much lower-level representation of the program than that of a conventional parallelizer. Then, MIL versions of the library routines used by the program are included in the compilation.

This unconventional approach has several advantages. First, the compiler is able to understand machine-specific operations that would ordinarily confuse and defeat a parallelizer. For example, in SPICE (the circuit simulator in the Perfect Benchmarks suite) there are calls to non-standard library routines (*loc* and *floc*) which return the addresses of variables. These pointers are stored as integers in the Fortran code, but are used to read and write the memory during the computation. This is a violation of the Fortran standard, and therefore cannot be treated by a conventional parallelizer. MIPRAC, however, operates upon a translation of the program into MIL, wherein such pointer manipulations are made explicit. MIPRAC is therefore able to analyze this code correctly. The same is true of Fortran programs with aliases that violate the language standard. There are many more examples of such machine-specific operations to be found in C programs. It is especially common to find that library routines and linkages between components in different programming languages violate a language standard or perform machine-dependent actions that are sensible only from a low-level, operational perspective like that of MIL.

The second advantage of this approach to multilingual parallelization is that the intermediate language, being a complete implementation language (as would be the assembly language of any target machine), need not grow as additional languages are treated by the system. The task of adding a new language to MIPRAC is twofold: a translator (compiler) into MIL is written, and the standard library of the language is provided, either in MIL or in one or more languages accepted by MIPRAC. Beyond this, parallelization and code generation are oblivious to the source language of the input program. In contrast, intermediate representations of the syntax of an input program (e.g., abstract syntax trees) must grow when an additional source language is to be treated by the compiler, whenever the new language contains structures not present in the languages already accepted by the compiler.

Finally, MIL has been designed with formal semantics in mind. We have written several denotational semantics for MIL which are the basis for the interprocedural analysis of MIL programs. This means that when these semantics are coupled with the front ends, we in effect have denotational descriptions of C and Fortran programs (even when those programs are not standard-conforming). Moreover, MIL programs are normalized by MIPRAC into a simplified form, with an especially simple denotational semantics in which they may be easily analyzed and transformed.

Interactive Compilation

MIPRAC performs an interprocedural analysis of the semantics of its input program and summarizes the results in tables that include: a calling graph, the side effects of procedures, the side effects of expressions, access sites of variables, object lifetimes, closed forms (for induction variables), and a dependence graph.

The side effects, access sites, and object lifetimes include references to any dynamically allocated data in the program. These tables are too large to be useful in their entirety, but when they are examined interactively the user can select the objects and expressions of interest.

When used in this mode, MIPRAC is an advisor by which one may learn the structure of a large system of software, possibly involving modules in several languages. The dependences between selected portions of the program, and the side effects of various modules, may be deduced automatically and examined selectively.

Sparse Matrices

MIPRAC provides several kinds of support for sparse matrix computations. First, MIPRAC is able to apply conventional dependence testing to pointer-based structures. Second, MIPRAC provides for the allocation of dynamically allocated structures (e.g., sparse rows) in a hierarchical memory, and for the automatic deallocation of these structures when their lifetimes have expired.

For example, consider the following C fragment:

```
void *
displaced_malloc(int size_obj, int n_obj, int offset)
{
    char *p;
    p = (char *) malloc(n_obj * size_obj);
    return p - (offset * size_obj)
}
```

This is a typical allocation routine for a sparse computation. Suppose we have a 100 by 100 matrix a with three non-zero elements per row, at positions $i-1$, i , $i+1$. We may build a sparse representation for a in C by writing

```
float *a[NROWS];
for (i=0; i<N; i++)
    a[i] = displaced_malloc (size of float, 3, i-1);
```

Then by writing $a[i][j]$ (where j is a non-zero element of row i) we access this sparse structure using ordinary subscript computations. MIPRAC's analysis allows a conventional subscript test to be applied to the displaced rows of a , because it represents pointers as the addresses of object, with integer displacements added to them. It also provides for the rows of a to be allocated and deallocated in the hierarchical memory according to their lifetime. We are working to test these techniques and to extend their power, using the Perfect and SPEC benchmarks, and sparse matrix codes written at CSRD.

Studies in Scheduling

Extensive research has been conducted on auto-scheduling and on automatic approaches to functional parallelism extraction from ordinary serial (Fortran and C) programs. We now have a comprehensive methodology for extracting unstructured parallelism from serial programs. This approach, which attacks the general instance of the problem, is based on data and control dependence analysis. In addition, the problem was proved to be NPcomplete. Efficient optimization heuristics were developed to eliminate redundant control and data dependences, and thus to expose more parallelism.

Finally, based on the work described above, we have developed a new universal intermediate program representation called the Hierarchical Task Graph, which is suitable for both serial and parallel programs. Auto-scheduling is carried out on the HTG by having the compiler interject special drive-code between the nodes of the HTG. The resulting graph represents a program that can package its parallelism and schedule its tasks transparently from the operating system or the runtime library. This facilitates efficient exploitation of fine-grain parallelism without incurring the "traditional" overhead of thread creation and scheduling.

It is worth noting that during the last few years many of our research results have been adopted by industrial labs and incorporated in commercial products within a few months after their publication (notably DEC, Cray Research, Convex, Gypsi, and TERA Computers). Publications based on part of this work include [GiPo91a, GiPo91b, GiPo91c].

In the same period we completed a powerful simulator for studying the effects of various schedulers on multiprocessing and multiprogramming environments. Using the same simulator we can also study the effects of auto-scheduling in a multiprogramming environment. The ability of auto-scheduling to adjust task granularity dynamically by reflecting dynamic changes in the workload gives us additional opportunities to boost system utilization and to make better use of processor and memory resources. This work is continuing with more experiments.

During the next academic year we plan to continue our work on the implementation of the HTG in the Parafrase-2 compiler, conduct experiments to determine the amount of functional parallelism that can be extracted automatically from real scientific applications, and focus on the architectural support required to exploit fully the advantages offered by the auto-scheduling model and the HTG.

Compiler-Generated and Optimized Synchronization for Data Dependences

Compiler generated synchronization is useful because it allows a parallelizing compiler to generate concurrent loops with data dependences, and to automatically synchronize those data dependences. Because synchronization operations consume resources, it is also desirable that the compiler reduce the number of synchronization instructions executed at compile time and to reduce the amount of machine resources consumed by synchronization.

The generation and optimization of synchronization for data dependences is driven by the data dependence graph. The basic principle in generating synchronization is to wait at the data access at the head of a dependence arc (the *sink*) for the data access at the tail of the arc (the *source*) to complete. For example, in Figure 1, the dependence from *S3* to *S2* is synchronized by inserting an instruction before statement *S2* to wait until the write to *b* at *S3* in the previous iteration completes. After *S3*, an instruction is placed to signal that the write to *b* in that iteration has completed. If the two data accesses are always separated by a constant number of iterations – that is, the dependence has a constant distance – this is quite simple to do. If the number of iterations separating the two accesses varies, or cannot be computed at compile time, it is necessary either to wait until all previous accesses to the source of the dependence have completed, or to determine at runtime in what accesses at the dependence source must complete before performing the sink access. Details on all these methods can be found in [TaYZ88, MiPa87, MiPa86, Midk86, ZhYe84].

```

doacross i = 1, n
S1      a(i) = c(i) + d(i)
S2      e(i) = b(i-1) + d(i)
S3      b(i) = c(i) + d(i)
S4      e(i) = a(i-1) + d(i)
end doacross

```

Figure 1: A loop with dependences

The synchronization for a dependence, along with the serial execution of some parts of the program, may cause another dependence to be synchronized. Thus, in Figure 1, synchronizing the dependence from *S3* to *S2* will cause the dependence from *S1* to *S4* to be synchronized also. Therefore the dependence from *S1* to *S4* does not need to be explicitly synchronized. This situation can be recognized using two general methods – pattern matching on the lexical relation and distances of pairs of dependences [MiPa91,

MiPa87, LiAb87, LiAb85, LiZh85], and pathfinding in sections of the iteration space of the loop [MiPa91, Jaya88, MiPa87, MiPa86, Midk86]. Pattern matching methods have the advantage of running faster and being able to handle dependences with an unknown distance, whereas the pathfinding methods can detect when the synchronization from two or more dependences synchronizes another dependence. Preliminary data [MiPa91] indicate that with Fortran programs from the Perfect Benchmarks, the pattern matching methods are nearly as good as the pathfinding methods. These data also indicate that taking into account the actual synchronization used, instead of just using the dependence structure of the program, is extremely beneficial.

Use of the synchronization optimization methods just discussed is important for two reasons. Most obviously, the runtime overhead is reduced as the number of synchronization operations executed is reduced. Less obviously, reducing the number of synchronization points in a loop can lead to reductions in the amount of synchronization hardware needed, allowing either the synchronization hardware to be faster, or allowing the resources to be used in other parts of the machine. Because compile times are also a concern, the preliminary data suggest that compilers should either only use pattern matching methods, or use pattern matching methods as a "prefilter" to minimize the amount of work done by the path-finding methods.

Multiprocessor Performance on Algorithms and Applications

All of the work projected for dense matrix computations has been completed. This work resulted in the first basic primitives based on matrix-matrix operations (BLAS3) that enhanced the performance of block-based algorithms for solving linear systems and the algebraic eigenvalue problem on parallel computers with hierarchical memory systems (e.g., Alliant FX/8 and FX/80, as well as Cedar). Some of the algorithmic innovations actually led to higher performance than previously known on vector machines, such as 1 CPU of the Cray X-MP. This collection of algorithms constitutes Linpack- and Eispack-like libraries for one cluster of Cedar, with some implemented on four clusters [GaPS90]. With minor modifications, these algorithms can be adapted for multiprocessing on the Cray-2 or Cray Y-MP. We have also introduced two rapid elliptic solvers: one based on FFT's and tridiagonal solvers and the other based on a modified block-cyclic reduction scheme [GaSa89a, GaSa89b, GaSa89c, GaSa89d, GaSa89e, Fran90, GaFM89]. Both schemes achieve high performance on one Cedar cluster and 1 CPU of a Cray X-MP, and both have been implemented on the multicluster Cedar. Based on some of the ideas developed in our parallel rapid elliptic solvers, we have developed an implicit parallel algorithm for solving general linear parabolic partial differential equations [GaSa90]. This parabolic solver achieves impressive speed improvement compared to the best sequential algorithm on both the Alliant FX/8 and Cray Y-MP multiprocessors, and it obtains more accurate solutions. It has also been implemented on the multicluster Cedar with good speedup results.

Domain decomposition methods are ideally suited for large grain parallelism and serve as the first step in problem decomposition among the clusters of Cedar or the CPU's of the Cray X-MP or Cray-2. In the case of Cedar, an algorithm designer has

two more levels of "parallelism": (i) taking advantage of the concurrency available in each cluster (medium grain parallelism), and (ii) taking advantage of the vector capability of each processor within a cluster (fine-grain parallelism). For the Cray architecture, on the other hand, an algorithm designer has to tailor his algorithm to capitalize on vectorization only, once the problem is decomposed among the various CPU's.

For this DOE-supported research we have concentrated our efforts on domain decomposition techniques in several large-scale applications [ChSa87, ChSa89a, ChSa89b, Heal91, ACCG91].

Domain Decomposition Method for Structural Analysis

We are continuing our work on symmetric and antisymmetric domain decomposition methods for three-dimensional elasticity problems with nonsymmetric domains and boundary conditions. So far, most of the experiments used to illustrate the effectiveness of this approach on both vector and parallel machines have been on simple model problems. We plan to obtain actual problems from industry or the national laboratories to test and improve our approach. Our experiments with this scheme show good speedup on the multicluster Cedar vs. one cluster in solving the linear systems resulting from this application. We have also parallelized an explicit finite element code for structural dynamics on the Alliant FX/80, Cray Y-MP, and Cedar, and plan to investigate the feasibility of using domain decomposition techniques in such schemes.

Circuit Simulation

Along with suitable partitioning schemes, we are in the process of developing hierarchical relaxation algorithms that employ waveform relaxation across the Cedar clusters and use efficient methods, such as iterative time analysis or waveform relaxation – Newton, for solving the differential subsystems generated by waveform relaxation on each cluster [XiaE88, SGCH90, HWGS90]. We also plan to take advantage of certain partitioning strategies (dissection-like schemes) in order to use a direct approach (like SPICE [Yang90a, Yang90b, Yang90c]) on all Cedar clusters.

Ocean Simulation

We plan to investigate the use of domain decomposition techniques in an ongoing study for the parallelization of an ocean simulation code on Cedar [DeGG89, DeGG91]. The mathematical model involves the solution of steady-state and time-dependent boundary value problems defined on complex domains (e.g., with holes, in order to model islands). In conjunction with atmospheric general circulation these models can be used to predict weather. These models are written in finite-difference form in order to treat the complex boundary conditions involved. A major difference between ocean and atmospheric circulation models is that the resolution necessary to model the dynamics in the former is considerably higher. As a result, a much finer grid is required, implying the manipulation of significantly larger amounts of data. Hence, apart from the intrinsic interest in the features of the numerical algorithms involved and their parallelization, we

are interested in the performance of the code on Cedar-like architectures, particularly the impact of the substantial I/O requirements, as well as the impact of the virtual memory system on performance.

In addition to developing algorithms for Cedar and the Cray multiprocessors, we plan also to investigate the possibility of experimenting with and porting our software to other architectures such as Intel's Delta hypercube.

Computational Fluid Dynamics

Last year we initiated an effort to design a finite-element, operator-splitting scheme for solving the time-dependent Navier-Stokes equation for three-dimensional configurations which is suitable for parallel computers [SeAs91]. The resulting scheme has been used to solve well-known model problems such as 2-D and 3-D driven cavities. Experiments and analysis show the superiority of this scheme over others, such as Crank-Nicolson. Furthermore, its performance on the Alliant FX/80 indicates efficient utilization of both concurrency and the vector capability of each processor.

Another effort in this area is aimed at the parallelization of a production code, N3S, for handling the Navier-Stokes equations for both internal and external flows. Software optimization and algorithm improvements led to speed improvements on 1 CPU of the Cray Y-MP. Currently, the code is being ported to the Alliant FX/80 and Cedar, as well as a 4-CPU Cray Y-MP.

Sparse Matrix Computations

Work on this topic has been in progress for several years. Much of the work concerns the design of robust sparse linear system solvers, both direct and preconditioned iterative schemes. Actual implementation on parallel machines such as the Alliant FX/80, Cedar, and multiple-CPU Cray Y-MP lags somewhat behind implementation on their counterparts for dense matrix computations.

Direct Sparse System Solvers

Effective parallel methods in this area depend on the development of novel reordering [Wijs89b] and factorization schemes as well as the development of a set of high-performance primitives akin to the BLAS3 for dense matrix computations. We have concentrated mainly on parallel direct solvers that are suitable for unstructured sparse general nonsymmetric linear systems. Toward this goal, we have developed three direct sparse solvers for the multicluster Cedar. Two are not yet utilizing the hierarchical memory organization of Cedar; that is, dealing with Cedar as a "flat" multiprocessor [Yang90a, Davi86, Davi89]. The third incorporates a reordering strategy to produce a matrix structure that is more suitable for the multicluster architecture [Mars91]. Numerous experiments on Cedar show reasonable speedup. However, much work remains to be done toward fine-tuning all three solvers.

Iterative Sparse System Solvers

Such solvers are needed for three-dimensional problems because direct sparse solvers tend to be too time consuming and demand higher storage capacity. Conjugate gradient (CG) schemes for symmetric positive definite systems [MeSa88, MeEi91, Voeg91] and CG-like methods [GaSZ90a, GaSZ90b, MeSG91] for nonsymmetric systems have been investigated, by us and others, on a variety of parallel architectures. The efficiency of CG schemes depends heavily on the choice of the preconditioner; an effective preconditioner for one architecture is not necessarily the preconditioner of choice for another architecture. For CG-like schemes, success in solving the nonsymmetric linear system depends critically on the choice of the preconditioner. We have investigated the choice of preconditioners for these classes of methods for the Cedar and Cray multiprocessors. Also, because most CG-like schemes are successful only for matrices with positive definite symmetric parts, we plan to develop more robust solvers that are either based on (block) row-action methods, for example block-Kaczmarz [BrSa88b, BrSa90a, BrSa90b] or CG-based schemes for the normal equations. These methods can be designed to be ideally suited for a variety of parallel computers. Our aim in developing robust parallel iterative solvers is that they can be used for applications as varied as computational fluid dynamics, circuit simulation (e.g., SPICE [Yang90a, Yang90b]), and semiconductor device simulation [GaKR91].

More attention will be paid to developing algebraic multigrid schemes for such systems and incorporating them in the above methods on the Cedar and Cray multiprocessors.

Wavelets

Wavelets are a new family of functions that show promise for solving various types of numerical partial differential equations and signal processing problems. The interesting feature of a wavelet basis is that it provides an orthogonal decomposition of a function into basis functions that are translations and scalings of one another. The same property holds for classical trigonometric decompositions as well, but wavelets can have finite support, which is important for localizing in both time and frequency domains. We have explored the implementation of wavelet transform algorithms on a number of parallel and vector machines. The wavelet transform is similar to the FFT but is more efficient requiring only $O(n)$ operations instead of $O(n \log n)$ steps. This work is described in [Sull90].

Perfect Benchmarks

Work on supercomputing benchmarking and performance evaluation has continued in a number of complementary directions. The overall effort is still guided by CSRD personnel, with active participation by American vendors such as Cray Research and Convex. Our progress during this reporting period is described below.

Workload Analysis and Perfect Benchmarks Verification

A fundamental question that must be asked about any computer benchmark is the extent to which that benchmark is representative of a supercomputer workload. Because the Perfect Benchmarks were drawn from a variety of sites and user communities, there is no simple correlation that can be made a priori about the representativeness of the benchmark suite. Our methodology for evaluating the generality of the suite consisted of sampling the major users of the University of Illinois NCSA Cray, obtaining their codes and typical datasets, and then performing low-level code characterizations. Our metrics included instruction frequencies, average vector lengths, instruction buffer fetch frequencies, and total running time. We compared the metrics for the NCSA workload with similar metrics from the Perfect codes and other commonly used Fortran benchmarks. Our findings were reported in [Brad91] and [Berr91], and quantitative evidence was presented that indicates that the Perfect codes resemble the NCSA workload more closely than do kernel- and algorithm-based benchmarks. This work will be continued in the direction of measuring average theoretical parallelism using the Parallel Performance Advisor (PPA) tool.

Perfect Version 2

Since the inception of the Perfect Benchmarks effort in 1987, a number of changes have occurred in supercomputing. The two most relevant to the Perfect activity are: (1) workstation speeds and memories have increased to the point where the small memory and two-dimensional models of many of the Perfect codes do not clearly separate supercomputers from advanced workstations, and (2) relatively mature explicitly parallel codes that solve meaningful scientific and engineering problems have been developed. These developments have enabled an updated suite of codes and datasets that have a more comprehensive coverage of typical supercomputer applications as well as larger, 3-dimensional models and explicitly parallel codes in some cases to accompany the basic Fortran 77 codes. We are cooperating with IBM, Cray Research, Cray Computer, Los Alamos National Laboratory, and Arco among other institutions in this endeavor. A number of other benchmarking activities are expressing interest in collaborating with us. We are discussing these directions with Euroben and SPEC representatives.

Instrumentation of the Perfect Codes

Current instrumentation of the Perfect suite is limited to elapsed and CPU times as determined by internally defined timing points. A beta-version of a more refined timing instrumentation has been released for verification by Perfect collaborators and is scheduled to be released generally during the fall of 1991. The lessons learned from this instrumentation will be applied to Version 2, as described in [Raus90].

Perfect Performance Database

A database system designed explicitly for benchmarking performance data was completed and implemented during this reporting period. The system is written in Ingress and allows for remote usage. The existing Perfect performance data have been entered in the database and will be put on-line this fall for general use.

Performance Environments and Tools

Cedar Performance Environment

The implementation of the Cedar Performance Hardware is complete and is now undergoing integration into the system. Checks comparing software and hardware tracing have been performed, and the results verify the hardware tracing facility. The histogrammer boards are complete and are being integrated into the system in the summer of 1991. The performance environment has already been used to explore performance of some Perfect codes on Cedar, and a report documenting that application of the environment is underway.

Parallel Performance Advisor (PPA)

CSRD has developed a large number of performance tools and simulation packages during the past 5 years, as have other computer researchers. The variety of tools can overwhelm users, especially if there is significant startup time associated with learning how to use and interpret each tool. In an attempt to overcome this difficulty but to retain the utility of the various tools that have been developed, we are designing and implementing a comprehensive performance and simulation facility that works in a default way, running a number of tools on a user program, combining the results, and interpreting them for the user. Our goal is to enable the user to focus on small program segments that present the greatest performance problems together with suggestions about how those problems might be resolved.

Bibliography
DOE DE-FG02-85ER25001

- [ACCG91] S. Aslam, H. C. Chen, G. Cybenko, E. Gallopoulos, H. Gao, M. Ham, A. Malony, A. Sameh, S. Sharma, T. Canfield, D. Leibfritz, M. Minkoff, C. Mueller and E. Plaskacz. "The Advanced Software Development and Commercialization Project Progress Report PR-2", CSRD Report No. 1129, Univ. of Illinois at Urbana-Champaign, Cntr. for Supercomputing Res. & Dev., April 1991.
- [AnPo91] John B. Andrews and Constantine D. Polychronopoulos. *An Analytical Approach to Performance/Cost Modeling of Parallel Computers*. **To appear in the Journal of Parallel and Distributed Computing and to appear in the special issue on Performance Modeling of Parallel Computers in August.**, April 1991.
- [BCGL91] D. Bradley, G. Cybenko, H. Gao, J. Larson, F. Ahmad, J. Golab and M. Straka. *Supercomputer Workload Decomposition and Analysis*. **Proc. of ICS'91 Conference, Cologne, Germany, June 17-21, 1991.**
- [BCVY91] John Bruner, Hoichi Cheong, Alexander Veidenbaum and Pen-Chung Yew. *Chief: A Parallel Simulation Environment for Parallel Systems*. **Presented at the 5th Int'l Parallel Processing Symp. in Anaheim, CA, April 30 - May 2, 1991.**
- [BePo91] Carl J. Beckmann and Constantine D. Polychronopoulos. *Broadcast Networks for Fast Synchronization*. **Proc. of ICPP'91, Vol. I, pp. 220-224, August 1991.**
- [BePo91] Carl J. Beckmann and Constantine D. Polychronopoulos. *The Effect of Scheduling and Synchronization Overhead on Parallel Loop Performance*. **Submitted for publication, April 1991.**
- [Brun91] M. C. Brunet. "An Algorithm for Transforming Iterative Numerical Algorithms to Adaptive Equivalents (Application to the Stokes Problem)", CSRD Report No. 1063, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1991.
- [Chen91] Hsin-Chu Chen. *Circulative Matrices of Degree O*. **A revised version of the manuscript "Two Special Classes of Matrices", submitted to SIAM Journal on Matrix Analysis and Applications., February 1991.**
- [ChLH91] Pohua P. Chang, Daniel M. Lavery and Wen-mei W. Hwu. *The Importance*

of Prepass Code Scheduling for Superscalar and Superpipelined Processors.
Submitted for publication, May 1991.

- [ChYe91] Ding-Kai Chen and Pen-Chung Yew. *An Empirical Study on DOACROSS Loops.* To appear in *Proceedings of Supercomputing'91*, Albuquerque, NM., November 18-22, 1991.
- [DGGN91] L. DeRose, K. Gallivan, E. Gallopoulos and A. Navarra. *Parallel Ocean Circulation Modeling on Cedar.* *Proc. 5th SIAM Conf. on Parallel Processing for Scientific Computing*, Houston, TX, March 1991.
- [EiBl91] Rudolf Eigenmann and William Blume. *An Effectiveness Study of Parallelizing Compiler Techniques.* *Proc. of ICPP'91*, St. Charles, IL, August 12-16, 1991.
- [Eijk91] Victor Eijkhout. *Beware of Modified Incomplete Point Factorizations.* *IMACS Conference on Iterative Methods*, Brussels, Belgium, April 2-4, 1991.
- [Eijk91] Victor Eijkhout. *Beware of Unperturbed Modified Incomplete Factorizations.* Presented at the *IMACS Int'l. Symposium on Iterative Methods in Linear Algebra*, Brussels, Belgium, April 2-4, 1991.
- [EiVa91] Victor Eijkhout and Panayot Vassilevski. "The Role of the Strengthened C.B.S. Inequality In Multilevel Methods", May 1991.
- [GaJW91] K. Gallivan, W. Jalby and H. Wijshoff. "Some Basic Performance Measurements of the 16 x 16 Cedar Configuration", CSRD Report No. 1146, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1991.
- [GaKR91] Albert T. Galick, Thomas Kerkhoven and Umberto Ravaioli. *Iterative Solution of the Eigenvalue Problem for a Dielectric Waveguide.* Submitted to *IEEE Transactions on Microwave Theory and Techniques*, July 1991.
- [Gall91] E. Gallopoulos. *Algorithms and Applications Research at CSRD.* *Proceedings of the Rennes Sister Cities Computer Science Research Centers Workshop*, Rennes, FRANCE, pp. 140-151, April 17-19, 1991.
- [GiPo91] Milind Girkar and Constantine D. Polychronopoulos. *The HTG: An Intermediate Representation for Programs Based on Control and Data Dependencies.* To appear in *IEEE Transactions on Parallel and*

Distributed Systems, May 1991.

- [GJTV91] K. Gallivan, W. Jalby, S. Turner, A. Veidenbaum and H. Wijshoff. *Preliminary Basic Performance Analysis of the Cedar Multiprocessor Memory Systems*. **Proc. of ICPP'91, St. Charles, IL, Vol. I, pp. 71-75, August 12-16, 1991.**
- [GrRu91] Elana D. Granston and Vincent F. Russo. *Signature-Based Polymorphism for C++*. **Proc. of the Usenix C++ Technical Conference, Washington, DC, April 21-25, 1991.**
- [GrVe90] Elana D. Granston and Alexander V. Veidenbaum. *Detecting Redundant Accesses To Array Data*. **To appear in Proceedings of Supercomputing'91, Albuquerque, NM, November 18-22, 1991.**
- [GrVe91] Elana D. Granston and Alexander V. Veidenbaum. *An Integrated Hardware/Software Solution for Effective Management of Local Storage in High-Performance Systems*. **Proceedings of ICPP'91, St. Charles, IL, Vol. II, pp. 83-90, August 12-16, 1991.**
- [Heal91] Brian Edward Healy. "Parallel and Vector Algorithms in Nonlinear Structural Dynamics Using the Finite Element Method", CSRD Report No. 1139, UILU-ENG-91-8040, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1991.
- [HsYe91] William Tsun-yuk Hsu and Pen-Chung Yew. *An Effective Synchronization Network for Large Multiprocessor Systems*. **Submitted for Publication, May 1991.**
- [HuGS91] G. G. Hung, K. Gallivan and R. Saleh. *Parallel Circuit Simulation Based on Nonlinear Relaxation Methods*. **To appear in the Proc. of ISCAS'91, Singapore, June 11-14, 1991.**
- [Hung91] Gih-Guang Hung. "Parallel Circuit Simulation Techniques Using Nonlinear Relaxation", CSRD Report No. 1126 UILU-ENG-91-8037, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1991.
- [JaTu91] David Jablonowski and Allan Tuchman. "Vista Users Manual", CSRD Report No. 1068, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1991.
- [KoCG91] C. K. Koc, P. R. Cappelo and E. Gallopoulos. *Decomposing Polynomial Interpolation for Systolic Arrays*. **Int'l Journal of Computer Mathematics,**

Vol. 38, pp. 219-239, April 1991.

- [Kona91] Pavlos Konas. "Parallel Discrete Event Simulation on Shared Memory Multiprocessors", CSRD Report No. 1074, UILU-ENG-91-8035, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1991.
- [KoYe91] Pavlos Konas and Pen-Chung Yew. *Parallel Discrete Event Simulation on Shared-Memory Multiprocessor*. Proc. of the 24th Annual Simulation Symposium, New Orleans, LA, April 1-5, 1991.
- [LimH91] Hock-Beng Lim. "Characterization of Parallel Program Behavior on a Shared-Memory Multiprocessor", CSRD Report No. 1138, UILU-ENG-91-8039, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., July 1991.
- [LiYe91] David Lilja and Pen-Chung Yew. *The Performance Potential of Fine-Grain and Coarse-Grain Parallel Architecture*. Proc. of Hawaii Int'l. Conference on System Sciences, Hawaii, Vol. 1, No. Architecture, pp. 324-333, January 1991.
- [LiYe91] David J. Lilja and Pen-Chung Yew. *Combining Hardware and Software Cache Coherence Strategies*. Presented at the Int'l. Conference on Supercomputing, Cologne, Germany, June 17-21, 1991.
- [LiYe91] Hock-Beng Lim and Pen-Chung Yew. *Parallel Program Behavioral Study on a Shared Multiprocessor*. Condensed version in Proc. of ICS'91 Conference, Cologne, Germany, June 17-21, 1991.
- [LiYe91] David J. Lilja and Pen-Chung Yew. *Architectures and Compiler Techniques for Exploiting Parallelism in Loops*. Submitted for publication, March 1991.
- [LiZh91] Zhiyuan Li. *Compiler Algorithms for Event Variable Synchronization*. Proceedings of ICS'91, Cologne, Germany, June 17-21, 1991.
- [MaHJ91] Allen Malony, David Hammerslag and David Jablonowski. *TraceView: A Trace Visualization Tool*. Submitted for Publication, February 1991.
- [Mars91] Bret Andrew Marsolf. "Large Grain Parallel Sparse System Solver", CSRD Report No. 1125, UILU-ENG-91-8036, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1991.
- [MeEi91] Ulrike Meier and Rudolf Eigenmann. *Parallelization and Performance of*

Conjugate Gradient Algorithms on the Cedar Hierarchical-Memory Multiprocessor. Proc. of the 3rd ACM Sigplan Symp. on Principles and Practice of Parallel Programming, Williamsburg, VA, April 21-24, 1991.

- [MiPa91] Sam P. Midkiff and David A. Padua. *A Comparison of Four Synchronization Optimization Techniques. Proceedings of ICPP'91, St. Charles, IL, August 12-16, 1991.*
- [MeSG91] U. Meier, G. Skinner and J. Gunnels. "A Collection of Codes for Sparse Matrix Computations", CSRD Report No. 1134, INTERNAL, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., July 1991.
- [SABB91] Ahmed Sameh, Sohail Aslam, Michael Berry, Randall Bramley, Hsin-Chu Chen, Victor Eijkhout, Kyle Gallivan, Stratis Gallopoulos, Michael Heath, Fen-Lien Juang, Ulrike Meier and Gung-Chung Yang. "Algorithms and Applications Group: Center for Supercomputing Research and Development Semiannual Report July 1990-December 1990.", CSRD Report No. 1127, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1991.
- [SaBC91] S. Saarinen, R. Bramley and G. Cybenko. *The Numerical Solution of Neural Network Training Problems. Submitted for publication, January 1991.*
- [Staf91] CSRD Staff. *The Cedar Project. Proc. of ICPP'91, St. Charles, IL, August 12-16, 1991.*
- [SuYe91] Hong-Men Su and Pen-Chung Yew. *Efficient Doacross Execution on Distributed Shared-Memory Multiprocessors. Proc. of ICPP, St. Charles, IL, Vol. I, pp. 45-48, August 12-16, 1991.*
- [SuYe91] Hong-Men Su and Pen-Chung Yew. *A Generalized Information Dissemination Scheme and Its Applications. Submitted for Publication, February 1991.*
- [TCJB91] Allan Tuchman, George Cybenko, David Jablonowski, Brian Bliss and Sanjay Sharma. *VISTA: A Systems for Remote Data Visualization. Presented at the Fifth SIAM Conference on Parallel Processing for Scientific Computing, Houston, TX, March 25-27, 1991.*
- [TuJC91] Allan Tuchman, David Jablonowski and George Cybenko. *Run-Time Visualization of Program Data. To appear in the Proceedings of Visualization '91, San Diego, CA, October 22-25, 1991.*

- [Voeg91] Naomi Voegtli. "Chebyshev Polynomial Preconditioning for the Conjugate Gradient Method", CSR D Report No. 1137, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., July 1991.
- [WeGS91] Yen-Cheng Wen, Kyle Gallivan and Resve Saleh. *Parallel Event-Driven Waveform Relaxation*. To appear in the Proceedings of the Int'l. Conference on Computer Design, Boston, MA, October 14-16, 1991.
- [Will91] Peter L. Williams. *Applications of Computational Geometry to Volume Visualization*. Proc. of the Third Canadian Conference on Computational Geometry, Vancouver, BC, CANADA, August 1991.
- [AbPa90] Seth Abraham and Krishnan Padmanabhan. *Constraint Based Evaluation of Multicomputer Networks*. Proc. 1990 Int'l Conference on Parallel Processing, Vol. 1, pp. 521-525, August 13-17 1990.
- [Abra90] Seth Abraham. "Issues in the Architecture of Direct Interconnection Schemes for Multiprocessors", CSR D Report No. 977 UILU-ENG-90-8019, Univ. of Illinois at Urbana-Champaign Center for Supercomputing Res. & Dev., March 1990.
- [AmHa90] Zahira Ammarguella and Luddy Harrison. *Automatic Recognition of Induction & Recurrence Relations by Abstract Interpretation..* Proc. of Sigplan 1990, Yorktown Heights, Vol. 25, No. 6, pp. 283-295, June 1990.
- [Amma90] Zahira Ammarguella. "A Control-Flow Normalization Algorithm and its Complexity", CSR D Report No. 1042, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1990.
- [Beck90] Carl Beckmann. "CARL: An Architecture Simulation Language", CSR D Report No. 1066, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., December 1990.
- [BePo90] Carl Beckmann and Constantine Polychronopoulos. *Fast Barrier Synchronization Hardware*. Proc. Supercomputing '90, New York, NY, pp. 180-189, November 12-16, 1990.
- [Berr90] Michael Waitzel Berry. "Multiprocessor Sparse SVD Algorithms and Applications", CSR D Report No. 1049 UILU-ENG-90-8031, Univ. of Illinois at Urbana-Champaign, Cntr. for Supercomputing Res. & Dev., November 1990.
- [BlBG90] Brian Bliss, Marie-Christine Brunet and E. Gallopoulos. *Automatic Parallel Program Instrumentation: Applications in Performance and Error Analysis*.

Proc. of 2nd ACM Int'l. Conf. on Expert Systems for Numerical Computing, W. Lafayette, IN, May 24-26, 1990.

- [BIBG90] B. Bliss, M. C. Brunet and E. Gallopoulos. "Automatic Parallel Program Instrumentation with Applications in Performance and Error Analysis", CSRD Report No. 1025, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1990.
- [BrSa90] Randall Bramley and Ahmed Sameh. *Row Projection Methods for Large Non-symmetric Linear Systems*. **To appear SIAM Journal Scientific Statistical Computing**, January 1992.
- [BrSa90] Randall Bramley and Ahmed Sameh. *Domain Decomposition for Parallel Row Projection Algorithms*. **To appear in Applied Numerical Mathematics**, Vol. 8, January 1990.
- [Brun90] John Bruner. "PARSIM User Interface Tutorial", CSRD Report No. 1040, Univ. of Illinois at Urbana-Champaign, Cntr. for Supercomputing Res. & Dev., September 1990.
- [Chen90] Hsin-Chu Chen. "Two Special Classes of Matrices", CSRD Report No. 979, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., March 1990.
- [Chen90] Mei-Qin Chen. *A Parallel Quasi-Newton Method For Partially Separable Large-Scale Minimization*. **Presented at ORSA/TIMS Conference**, October 1990.
- [ChHa90] Jyh-Herng Chow and Luddy Harrison. *Switch-Stacks: A Scheme for Microtasking Nested Parallel Loops*. **Proc. of Supercomputing '90**, New York, NY, pp. 190-199, November 12-16, 1990.
- [ChHa90] Jyh-Herng Chow and Williams Ludwell Harrison III. *Microtasking Recursive, Parallel Programs*. **Proc. of the Int'l. Conference on Parallel Processing**, St. Charles, IL., Vol. II, No. Software, pp. 282-283, August 13-17, 1990.
- [Chow90] Jyh-Herng Chow. "Parallel Execution of Lisp Programs in the Parcel Run Time Environment", CSRD Report No. 991, UILU-ENG-90-8021, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1990.
- [ChSa90] Mei-Qin Chen and Ahmed Sameh. *Conjugate Subspaces Decomposition and Its Application in Solving Linear Systems with Many Right-Hand Sides*.

The Fourth Auburn Linear Algebra Conference, March 20–23, 1990, March 1990.

- [ChVe90] Hoichi Cheong and Alexander Veidenbaum. *Compiler-directed Cache Management in Multiprocessors*. **IEEE Computer Journal**, Vol. 23, No. 6, pp. 39–47, June 1990.
- [ChVe90] Yung-chin Chen and Alexander V. Veidenbaum. *Comparison and Analysis of Software and Directory Coherence Schemes*. To appear in **Supercomputing'91**, Albuquerque, NM, November 18–22, 1991.
- [CKPK90] George Cybenko, Lyle Kipp, Lynn Pointer and David Kuck. *Supercomputer Performance Evaluation and the Perfect Benchmarks^{TM}*. **Proceedings of ICS, Amsterdam, Netherlands**, March 1990.
- [DaYe90] Timothy Davis and Pen-Chung Yew. *A Stable Non-deterministic Parallel Algorithm for General Unsymmetric Sparse LU Factorization*. **SIAM J. on Matrix Analysis and Applications**, Vol. 2, No. 3, pp. 383–403, July 1990.
- [EHJP90] R. Eigenmann, J. Hoeflinger, G. Jaxon and D. Padua. *Cedar Fortran and Its Compiler*. **Lecture Notes in Computer Science: Proc. of the Joint Conf. on Vector and Parallel Processing**, Zurich, Switzerland, Vol. 457, pp. 288–300, January 1990.
- [EHJP90] R. Eigenmann, J. Hoeflinger, G. Jaxon and D. Padua. *Cedar Fortran and its Restructuring Compiler*. In: **Languages and Compilers for Parallel Computing II**, T. Gross D. Gelernter A. Nicolau and D. Padua, ed. MIT Press, 1990.
- [Eijk90] Victor Eijkhout. *Analysis of Parallel, Incomplete Point Factorizations*. To appear in **Linear Algebra and Its Applications**, October 1990.
- [Fran90] George N. Frank. "Experiments on the Cedar Multicluster with Parallel Block Cyclic Reduction and an Application to Domain Decomposition Methods", CSRD Report No. 1056 UILU-ENG-90-8032, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., November 1990.
- [FuCh90] Chuigang Fu. "Evaluating the Effectiveness of Fortran Vectorizers by Measuring Total Parallelism", CSRD Report No. 1033, UILU-ENG-90-8029, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1990.

- [GaBe90] Hui Gao and Michael Berry. "Performance Studies of LAPACK on Alliant FX/80 and 1 Cedar Cluster", CSRD Report No. 1001, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1990.
- [GaPS90] K. A. Gallivan, R. J. Plemmons and A. H. Sameh. *Parallel Algorithms for Dense Linear Algebra Computations*. SIAM Review, Vol. 32, No. 1, pp. 54-135, March 1990.
- [GaSa90] E. Gallopoulos and Y. Saad. "Efficient Solution of Parabolic Equations by Polynomial Approximation Methods", CSRD Report No. 969, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., February 1990.
- [GaSZ90] Kyle Gallivan, Ahmed Sameh and Zahari Zlatev. *Solving General Sparse Linear Systems Using Conjugate Gradient-type Methods*. Proc. 1990 Int'l. Conf. on Supercomputing, Amsterdam, Netherlands, ACM Press, pp. 132-139, June 1990.
- [GoGV90] Edward H. Gornish, Elana D. Granston and Alexander V. Veidenbaum. *Compiler-directed Data Prefetching in Multiprocessors with Memory Hierarchies*. Proc. of ICS'90, Amsterdam, The Netherlands, Vol. 1, pp. 342-353, May 1990.
- [GPHL90] Mark D. Guzzi, David A. Padua, Jay P. Hoeflinger and Duncan H. Lawrie. *Cedar Fortran and Other Vector and Parallel Fortran Dialects*. Jour. of Supercomputing, Vol. 4, No. 1, pp. 37-62, March 1990.
- [GrTV91] Elana Granston, Stephen Turner and Alexander Veidenbaum. *Design and Analysis of a Scalable, Shared-Memory System with Support for Burst Traffic*. Proceedings of the 1990 Workshop on Shared-Memory Multiprocessors, Seattle, WA, May 26-17, 1990.
- [HaAm90] Luddy Harrison and Zahira Ammarguellat. *A Comparison of Automatic versus Manual Parallelization of the Boyer-Moore Theorem Prover*. In: *Languages and Compilers for Parallel Computing*, A. Nicolau D. Gelernter and D. Padua, ed. MIT Press, pp. 307-330, 1990.
- [HWGS90] G. G. Hung, Y. C. Wen, K. Gallivan and R. Saleh. *A Parallel Circuit Simulation using Hierarchical Relaxation*. Proc. 27th Design Automation Conf., pp. 394-399, June 1990.
- [KrCV90] David W. Krumme, George Cybenko and K. N. Venkataraman. *Gossiping in Minimal Time*. To appear in SIAM Journal on Computing, August 1990.

- [Leun90] Bruce Paul Leung. "Issues on the Design of Parallelizing Compilers", CSRD Report No. 1012, UILU-ENG-90-8027, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1990.
- [LiYe90] David J. Lilja and Pen-Chung Yew. *The Interaction of Cache Block Size and Parallel Loop Scheduling Strategy*. Submitted for Publication, December 1990.
- [LiYe90] David J. Lilja and Pen-Chung Yew. "A Compiler-Assisted Directory-Based Cache Coherence Scheme", Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., November 1990.
- [LouG90] Gang Lou. "Parallel Methods for Solving Linear Systems Via Overlapping Decomposition", CSRD Report No. 1065, UILU-ENG-90-8033, Univ. of Illinois at Champaign-Urbana, Center for Supercomputing Res. & Dev., December 1990.
- [MiPa90] Sam Midkiff and David Padua. *Issues in the Compile-Time Optimization of Parallel Programs*. Proc. of Int'l Conf. on Parallel Processing 1990, Vol. II, pp. 105-113, August 1990.
- [Padu90] David Padua. *Problem Solving Environments for Parallel Computing*. Proc. of an International Conf. organized by the IPSJ to Commemorate the 30 Anniversary, pp. 323-330, November 1990.
- [PoJa90] David Pointer and Greg Jaxon. "Cedar Synchronization Processor Instruction Set Reference", CSRD Report No. 1017, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res & Dev., July 1990.
- [Poly90] Constantine D. Polychronopoulos. "Parallel Programming Issues", CSRD Report No. 1004, Univ. of Illinois at Urbana-Champaign, Ctr. for Supercomputing Res. & Dev., June 1990.
- [Poly90] Constantine D. Polychronopoulos. "Auto Scheduling: Control Flow and Data Flow Come Together", CSRD Report No. 1058, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., December 1990.
- [Saad90] Youcef Saad. "SPARSKIT: A Basic Tool Kit for Sparse Matrix Computation", CSRD Report No. 1029, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1990.
- [Scho90] Dale Allan Schouten. "An Overview of Interprocedural Analysis Techniques for High Performance Parallelizing Compilers", CSRD Report No. 1005,

UILU-ENG-90-8025, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1990.

- [ShLY90] Zhiyu Shen, Zhiyuan Li and Pen-Chung Yew. *An Empirical Study of Fortran Programs for Parallelizing Compilers*. **IEEE Trans. on Parallel and Distributed Systems**, Vol. 1, No. 3, pp. 350-364, July 1990.
- [SMBS90] Sanjay Sharma, Allen Malony, Michael Berry and Priyanvada Sinval-Sharma. *Run-Time Monitoring and Performance Visualization of Concurrent Programs*. **Proc. of Supercomputing 90**, November 1990, New York, NY, pp. 784-793, November 12-16, 1990.
- [TaYZ90] Peiyi Tang, Pen-Chung Yew and Chuan-Qi Zhu. *Compiler Techniques for Data Synchronization in Nested Parallel Loops*. **Proc. of ICS'90**, Amsterdam, Holland, Vol. 1, pp. 177-186, May 1990.
- [TuBe90] Allan Tuchman and Michael Berry. *Matrix Visualization in the Design of Numerical Algorithms*. **ORSA Journal of Computing**, Vol. 2, No. 1, 1990.
- [Will90] Peter L. Williams. *Issues in Interactive Direct Projection Volume Rendering of Nonrectilinear Meshed Data Sets*. Invited Work in Progress Report for San Diego Workshop on Volume Visualization, San Diego, CA, December 1990.
- [Yang90] Gung-Chung Yang. *DSPACK: A Parallel Direct Sparse Matrix Package for Shared-Memory Multiprocessors*. **Proc. of 1990 Int'l. Conference on Parallel Processing**, St. Charles, IL, Vol. 3, pp. 197-200, August 1990.
- [Yang90] Gung-Chung Yang. *PARASPICE: A Portable Parallel Circuit Simulator*. **Advances in Electrical Engineering Software (Electrosoft '90): Proc. of the 1st Int'l. Conference on Electrical Engineering Analysis and Design**, Editor: P.P. Silvester, Computational Mechanics Publications, Springer-Verlag, Lowell, MA, pp. 3-14, August 1990.
- [Yang90] Gung-Chung Yang. *An Integrated CAD System for Device Model Design, Parameter Extraction, and Circuit Simulation*. **Advances in Electrical Engineering Software (Electrosoft '90): Proc. of the 1st Int'l. Conference on Electrical Engineering Analysis and Design**, Editor: P.P. Silvester, Computational Mechanics Publications, Springer-Verlag, Lowell, MA, pp. 31-42, August 1990.
- [Yang90] Gung-Chung Yang. *PARASPICE: A Parallel Circuit Simulator for Shared-*

Memory Multiprocessors. Proceedings of the 27th ACM/IEEE Design Automation Conference, pp. 400–405, June 1990.

- [YeBr90] Pen-Chung Yew and John Bruner. *SEE: A System Evaluation Environment for Studying Parallel Systems*. Presented at the First Workshop on Parallel Processing, National Tsing Hua, Univ., Taiwan, Dec. 20–21, 1990, December 1990.
- [YiHa90] Kwangkeun Yi and Luddy Harrison. *On-the-fly Circuit to Measure the Average Working Set Size*. Proc. of '90 IEEE Int'l. Conference on Computer Design: VLSI in Computers and Processors, pp. 471–475, September 1990.
- [YiKw90] Kwang Keun Yi. "On-the-Fly Methods to Measure the Locality of Programs", CSRD Report No. 988, UILU-ENG-90-8020, Univ. of Illinois at Urbana-Champaign Center for Supercomputing Res. & Dev., May 1990.
- [AbPa89] Seth Abraham and Krishnan Padmanabhan. *An Analysis of the Twisted Cube Topology*. Proc. of 1989 Int'l. Conf. on Parallel Processing, St. Charles, IL, Vol. I, pp. 116–120, August 1989.
- [AbPa89] Seth Abraham and Krishnan Padmanabhan. *An Analysis of the Star Graph Topology for Multicomputer Systems*. Subm. for publ. to 10th Int'l. Conf. on Dist. Computing Systems, May 1990, Paris, France, October 1989.
- [Amm89] Zahira Ammarguella. "Normalization of Program Control Flow", CSRD Report No. 885, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1989.
- [AnSa89] Edward Anderson and Youcef Saad. *Solving Sparse Triangular Linear Systems on Parallel Computers*. Int'l. Journal of High Speed Computing, Vol. 1, No. 1, pp. 73–95, 1989.
- [BCGM89] Michael Berry, Hsin-Chu Chen, E. Gallopoulos, Ulrike Meier, Allan Tuchman, Harry Wijshoff and Gung-Chung Yang. *Algorithmic Design on the Cedar Multiprocessor*. Proc. of 4th Int'l. Conf. on Supercomputing & 3rd World Computer Exhibition, Santa Clara, CA, Vol. 27, pp. 78–88, April 1989.
- [BCKK89] M. Berry, D. Chen, P. Koss, D. Kuck, L. Pointer, S. Lo, Y. Pang, R. Roloff, A. Sameh, E. Clementi, S. Chin, D. Schneider, G. Fox, P. Messina, D. Walker, C. Hsiung, J. Schwarzmeier, K. Lue, S. Orszag and F. Seidl. *The Perfect Club Benchmarks: Effective Performance Evaluation of*

Supercomputers. Int'l. Jour. of Supercomputer Applications, Fall 1989, Vol. 3, No. 3, pp. 5-40, Fall 1989.

- [BCMS89] Randall Bramley, Hsin-Chu Chen, Ulrike Meier and Ahmed Sameh. *On Some Parallel Preconditioned CG Schemes. Proc. of Conf. on Preconditioned Conjugate Gradient Methods, 1989.*
- [BePo89] Carl J. Beckmann and Constantine Polychronopoulos. *The Effect of Barrier Synchronization and Scheduling Overhead on Parallel Loops. Proc. of 1989 Int'l. Conf. on Parallel Processing, St. Charles, IL, Vol. II, pp. 200-204, August 1989.*
- [BeSa89] Michael Berry and Ahmed Sameh. *An Overview of Parallel Algorithms for the Singular Value and Dense Symmetric Eigenvalue Problems. Jour. of Computational and Applied Mathematics, Vol. 27, pp. 191-213, 1989.*
- [BiEm89] Eric J. Bina and Perry A. Emrath. *A Faster fsck for BSD Unix. Proc. of USENIX Technical Conference, Winter 1989, San Diego, CA, pp. 173-185, Winter 1989.*
- [Blis89] Brian E. Bliss. "Instrumentation of Fortran Programs for Automatic Roundoff Error Analysis and Performance Evaluation", CSRD Report No. 936, UILU-ENG-89-8015, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., October 1989.
- [Bram89] Randall B. Bramley. "Row Projection Methods for Linear Systems", CSRD Report No. 881, UILU-ENG-89-8008, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1989.
- [Bram89] Randall Bramley. *Block Stiefel Acceleration. Submitted for Publication, October 1989.*
- [Chen89] Ding-Kai Chen. "MaxPar: An Execution Driven Simulator for Studying Parallel Systems", CSRD Report No. 917, UILU-ENG-89-8013, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., October 1989.
- [Cheo89] Hoichi Cheong. "Compiler-Directed Cache Coherence Strategies for Large-Scale Shared-Memory Multiprocessor Systems", CSRD Report No. 953, UILU-ENG-89-8018, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., December 1989.
- [ChSa89] Hsin-Chu Chen and Ahmed Sameh. *A Matrix Decomposition Method for Orthotropic Elasticity Problems. SIAM Jour. on Matrix Analysis and*

Applications, Vol. 10, No. 1, pp. 39-64, 1989.

- [ChSa89] Hsin-Chu Chen and Ahmed Sameh. *A Domain Decomposition Method for 3D Elasticity Problems. Applications of Supercomputers in Engineering: Fluid Flow and Stress Analysis Applications*, pp. 171-188, September 1989.
- [ChSY89] Ding-Kai Chen, Hong-Men Su and Pen-Chung Yew. *The Impact of Synchronization and Granularity on Parallel Systems. Proc. of the 17th Int'l. Symp. on Computer Architecture, Seattle, WA, December 1989.*
- [ChVe89] Hoichi Cheong and Alexander Veidenbaum. *A Version Control Approach to Cache Coherence. Proc. of 1989 Int'l. Conf. on Supercomputing*, pp. 322-330, June 1989.
- [Cybe89] George Cybenko. *Approximations by Superpositions of a Sigmoidal Function. Mathematics of Control, Signals & Systems (1989), Vol. 2*, pp. 303-314, August 1989.
- [CyBe89] George Cybenko and Michael Berry. *Hyperbolic Householder Algorithms for Factoring Structured Matrices. SIAM Jour. on Matrix Analysis and Applications*, May 1989.
- [Cybe89] George Cybenko. *Mathematical Problems in Neural Computing. Proc. of Int'l. Symp. on Mathematical Theory of Networks and Systems*, June 1989.
- [Cybe89] George Cybenko. *Designing Neural Networks. Proc. First Annual IEEE Int'l. Conf. on Artificial Neural Networks*, pp. 1-3, October 1989.
- [Davi89] Timothy Alden Davis. "A Parallel Algorithm for Sparse Unsymmetric Factorization", CSRD Report No. 907, UILU-ENG-89-8012, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., September 1989.
- [DeGG89] Luis DeRose, Kyle Gallivan and E. Gallopoulos. "Trace Analysis of the GFDL Ocean Circulation Model: A Preliminary Study", CSRD Report No. 863, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., April 1989.
- [EmGP89] Perry Emrath, Sanjoy Ghosh and David Padua. *Event Synchronization Analysis for Debugging Parallel Programs. Proc. of Supercomputing '89, Reno, Nevada, November 1989.*

- [EmPY89] Perry Emrath, David Padua and Pen-Chung Yew. *Cedar Architecture and Its Software*. **Proc. of Hawaii Int'l. Conf. on System Sciences** (January 1989), pp. 306–315, 1989.
- [Emra89] Perry Emrath. *Program Laborious*. **Unix Review**, pp. 51–60, April 1989.
- [GaFM89] E. Gallopoulos, G. Frank and U. Meier. *Experiments with Elliptic Problem Solvers on the Cedar Multicluster*. **Proc. of Fourth SIAM Conf. Par. Proc. Sci. Comput., Chicago, IL**, pp. 245–250, December 1989.
- [GaGJ89] Yogesh Gaur, Vincent A. Guarna Jr. and David Jablonowski. *An Environment for Performance Experimentation on Multiprocessors*. **Proc. of Supercomputing '89, Reno, NV**, pp. 589–596, November 1989.
- [GaGL89] Dennis Gannon, Vincent A. Guarna Jr. and Jenq Kuen Lee. *Static Analysis and Runtime Support for Parallel Execution*. **Monographs in Parallel and Distributed Computing**, Chris Jesshope and David Klappholz, eds., Pitman Publishing, 1989., 1989.
- [GaLe89] E. Gallopoulos and Daeshik Lee. *Fast Laplace Solver by Boundary Integral-Based Domain Decomposition Methods*. In: **Parallel Processing for Scientific Computing**, G. Rodrigue, ed. SIAM, pp. 141–145, 1989.
- [GaSa89] E. Gallopoulos and Y. Saad. *Parallel Rapid Elliptic Solvers*. **Parallel Processing for Scientific Computing**, pp. 51–55, 1989.
- [GaSa89] E. Gallopoulos and Youcef Saad. *Parallel Block Cyclic Reduction Algorithm for the Fast Solution of Elliptic Equations*. **Parallel Computing**, Vol. 10, No. 2, pp. 143–160, 1989.
- [GaSa89] E. Gallopoulos and Ahmed Sameh. *Solving Elliptic Equations on the Cedar Multiprocessor*. In: **Aspects of Computation on Asynchronous Parallel Processors**, M.H. Wright, ed. Elsevier Science Pub. B.V. (North-Holland), pp. 1–12, 1989.
- [GaSa89] E. Gallopoulos and Youcef Saad. *On the Parallel Solution of Parabolic Equations*. **Proc. of 1989 Int'l. Conf. on Supercomputing, Herakleion, Greece**, pp. 17–28, June 1989.
- [GaSa89] E. Gallopoulos and Ahmed Sameh. *Solving Elliptic Equations on the Cedar Multiprocessor*. In: **Aspects of Computation on Asynchronous Parallel Processors**, M.H. Wright, ed. Elsevier Science Pub. B.V. (North-Holland), pp. 1–12, 1989.

- [GGJM88] Vincent Guarna Jr., Dennis Gannon, David Jablonowski, Allen Malony and Yogesh Gaur. *Faust: An Integrated Environment for the Development of Parallel Programs*. IEEE Software, pp. 20-27, July 1989.
- [GJMW89] Kyle Gallivan, William Jalby, Allen Malony and Harry Wijshoff. *Performance Prediction of Loop Constructs on Multiprocessor Hierarchical*. Proc. Third Int'l. Conf. on Supercomputing, Crete, Greece, pp. 433-442, June 1989.
- [Gorn89] Edward H. Gornish. "Compile Time Analysis for Data Prefetching", CSRD Report No. 949, UILU-ENG-89-8016, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., December 1989.
- [Harr89] Luddy Harrison. "The Interprocedural Analysis and Automatic Parallelization of Scheme Programs", CSRD Report No. 860, UILU-ENG-89-8003, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1989.
- [JaGu89] David Jablonowski and Vince Guarna Jr. *GMB: A Tool for Manipulating and Animating Graph Data Structures*. Software-Practice and Experience, Vol. 19, No. 3, pp. 283-301, March 1989.
- [Karl89] Steven Karlovsky. "Automatic Management of Programmable Caches Algorithms and Experience", CSRD Report No. 892, University of Illinois at Urbana-Champaign Center for Supercomp. Res & Dev., July 1989.
- [KaSa89] Chandrika Kamath and Ahmed Sameh. *A Projection Method for Solving Nonsymmetric Linear Systems on Multiprocessors*. Parallel Computing, Vol. 9, pp. 291-312, 1989.
- [KaSa89] E. Kamgnia and A. Sameh. *A Fast Elliptic Solver for Simply Connected Domains*. Computer Physics Communications, North Holland, Amsterdam, Vol. 55, pp. 43-69, 1989.
- [KGAR89] T. Kerkhoven, A. Galick, J. H. Arends, U. Ravaioli and Y. Saad. *Efficient Numerical Simulation of Electron States in Quantum Wires*. Journal of the American Physical Society, Vol. 68, No. 7, pp. 3461-3469, October 1989.
- [Koss89] Peter Koss. "Application Performance on Supercomputers", CSRD Report No. 847, UILU-ENG-89-8001, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1989.
- [Krau89] James John Krause. "A Graphical User Interface for the XDBX Debugger",

CSRD Report No. 867, UILU-ENG-89-8004, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1989.

- [Kuck89] David J. Kuck. *Keynote Address 15th Annual Int'l. Symp. on Computer Architecture, May 30-June 3, 1988, Honolulu, HI. Computer Architecture News*, Vol. 17, No. 1, pp. 5-26, March 1989.
- [LaMe89] J. Laminie and U. Meier. *Solving Navier-Stokes Equations on the Cedar Multi-Cluster System. Proc. 4th SIAM Conference Parallel Processing for Scientific Computing, Chicago, IL.*, pp. 174-179, December 1989.
- [Lave89] Daniel M. Lavery. "The Design of a Hardware Performance Monitor for the Cedar Supercomputer", CSRD Report No. 866, UILU-ENG-89-8006, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1989.
- [Lilj89] David Lilja. *Efficient Generation of Poisson Distributed Random Numbers. Trans. of Soc. for Comp. Simulation*, Vol. 6, No. 1, pp. 31-41, 1989.
- [LiMY89] David Lilja, David Marcovitz and Pen-Chung Yew. "Memory Referencing Behavior and a Cache Performance Metric in a Shared Memory Multiprocessor", CSRD Report No. 836, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., February 1989.
- [LiRe89] Zhiyuan Li and Edward M. Reingold. *Solution of a Divide-and-Conquer Maximin Recurrence. SIAM Journal of Computing*, 1989.
- [LiYZ89] Zhiyuan Li, Pen-Chung Yew and Chuan-Qi Zhu. *Data Dependence Analysis on Multi-Dimensional Array References. Proc. of 1989 Int'l. Conf. on Supercomputing*, pp. 215-224, June 1989.
- [LiZh89] Zhiyuan Li. "Intraprocedural and Interprocedural Data Dependence Analysis for Parallel Computing", CSRD Report No. 910, UILU-ENG-89-8011, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1989.
- [Malo89] Allen D. Malony. *JED: Just an Event Display. Proc. of Workshop on Performance Instrumentation for Parallel Systems, Santa Fe, NM*, June 1989.
- [MiPC89] Samuel P. Midkiff, David Padua and Ronald G. Cytron. *Compiling Programs with User Parallelism. Research Monographs in Parallel & Distributed Computing*, C. Jesshope, D. Klappholz (eds.), Pitman

Publishing, 1989., 1989.

- [NeTu89] Henry Neeman and Allan Tuchman. "Simulation Time Animation System", CSRD Report No. 859, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., February 1989.
- [Perf89] Lynn Pointer Editor. "Perfect Report: 1", CSRD Report No. 896, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., July 1989.
- [Pete89] Paul Marx Petersen. "PDE Tutor: A System for the Automatic Solution of Partial Differential Equations", CSRD Report No. 891, UILU-ENG-89-8009, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1989.
- [PGHL89] Constantine Polychronopoulos, Milind Girkar, Mohammad Reza Haghighat, Chia-Ling Lee, Bruce Leung and Dale Schouten. *Parafrase-2: A New Generation Parallelizing Compiler*. *Proc. of 1989 Int'l. Conf. on Parallel Processing, St. Charles, IL*, Vol. II, pp. 39-48, August 1989.
- [PoKP89] Constantine Polychronopoulos, David Kuck and David Padua. *Utilizing Multidimensional Loop Parallelism on Large-Scale Parallel Processor Systems*. *IEEE Trans. on Comp.*, Vol. 38, No. 9, pp. 1285-1296, September 1989.
- [Poly89] Constantine Polychronopoulos. *Multiprocessing versus Multiprogramming*. *Proc. of 1989 Int'l. Conf. on Parallel Processing, St. Charles, IL*, Vol. II, pp. 223-230, August 1989.
- [SGCH89] R. Saleh, K. Gallivan, M. Chang, I. Hajj, D. Smart and T. Trick. *Parallel Circuit Simulation on Supercomputers*. *Proc. of IEEE*, Vol. 77, No. 12, pp. 1915-1931, December 1989.
- [ShLY89] Zhiyu Shen, Zhiyuan Li and Pen-Chung Yew. *An Empirical Study on Array Subscripts and Data Dependences*. *Proc. of 1989 Int'l. Conf. on Parallel Processing, St. Charles, IL*, Vol. II, pp. 145-152, August 1989.
- [ShMa89] Sanjay Sharma and Allen Malony. "XCOUNT: A Tool to Interpret Counting Related Events", CSRD Report No. 897, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1989.
- [ShNe89] Peter Shirley and Henry Neeman. *Volume Visualization at the Center for Supercomputing Research & Development*. *Proc. of Chapel Hill Workshop on Volume Visualization*, pp. 15-19, January 1989.

- [Skee89] Robert D. Skeel. "Macromolecular Dynamics on a Shared-Memory Multiprocessor", CSRD Report No. 929, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., October 1989.
- [SuYe89] Hong-Men Su and Pen-Chung Yew. *On Data Synchronizations for Multiprocessor Systems. Proc. of 16th Annual Int'l. Symp. on Computer Architecture, Jerusalem, Israel*, pp. 416-423, May 1989.
- [Tang89] Ju-Ho Tang. "Performance Evaluation of Vector Machine Architectures", CSRD Report No. 850, UILU-ENG-89-8002, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1989.
- [TaYZ90] Peiyi Tang, Pen-Chung Yew and Chuan-Qi Zhu. *A Parallel Linked List for Shared-Memory Multiprocessors. Proc. of the 13th Annual Int'l Computer Software & Application Conf.*, pp. 130-135, September 1989.
- [Turn89] Stephen Wilson Turner. "Shared Memory and Interconnection Network Performance for Vector Multiprocessors", CSRD Report No. 876, UILU-ENG-89-8007, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1989.
- [TuVe89] Stephen Turner and Alexander Veidenbaum. "Burst Traffic in MIN-Based Shared Memory Systems", CSRD Report No. 855, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., February 1989.
- [Wart89] Nancy Jeanne Warter. "An Environment Monitor for the Cedar Supercomputer", CSRD Report No. 871, UILU-ENG-89-8005, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1989.
- [Wijs89] Harry A. G. Wijshoff. "Implementing Sparse BLAS Primitives on Concurrent/Vector Processors: A Case Study", CSRD Report No. 843, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1989.
- [Wijs89] Harry A. G. Wijshoff. "Symmetric Orderings for Unsymmetric Sparse Matrices", CSRD Report No. 901, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1989.
- [ZhFL89] Chuan-Qi Zhu, Zhixi Fang and Xiaobo Li. *A New Parallel Sorting Approach with Sorting Memory Module. Jour. of Parallel and Distributed Computing*, Vol. 7, pp. 482-502, 1989.
- [AbDa88] Santosh Abraham and Timothy Davis. *Blocking for Parallel Sparse Linear*

System Solvers. Proc. of 1988 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 166–173, August 1988.

- [Abra88] Santosh Abraham. "Reducing Interprocessor Communication in Parallel Architectures: System Configuration and Task Assignment", CSRD Report No. 726, UILU-ENG-88-8001, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1988.
- [Ande88] Edward Charles Anderson. "Parallel Implementation of Preconditioned Conjugate Gradient Methods for Solving Sparse Systems of Linear Equations", CSRD Report No. 805, UILU-ENG-88-8007, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1988.
- [Bina88] Eric J. Bina. "Modifications to the UNIX File System Check Program FSCK for Quicker Crash Recovery", CSRD Report No. 811, UILU-ENG-88-8010, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1988.
- [BrSa88] Randall Bramley and Ahmed Sameh. *A Robust Parallel Solver for Block Tridiagonal Systems. Proc. of 1988 Int'l. Conf. on Supercomputing, St. Malo, France, ACM Press*, pp. 39–54, July 1988.
- [BrSa88] Randall Bramley and Ahmed Sameh. "A Robust Parallel Solver for Block Tridiagonal Systems", CSRD Report No. 806, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1988.
- [Chen88] Hsin-Chu Chen. "The SAS Domain Decomposition Method for Structural Analysis", CSRD Report No. 754, UILU-ENG-88-8003, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., March 1988.
- [Chen88] Hsin-Chu Chen. *The SAS Domain Decomposition Method. Presented at SIAM Conf. on Applied Linear Algebra, Madison, WI*, May 1988.
- [ChVe88] Hoichi Cheong and Alex Veidenbaum. *Cache Coherence Enforcement Scheme with Fast Selective Invalidation. 15th Int'l. Symp. on Comp. Architecture*, p. 299, June 1988.
- [ChVe88] Hoichi Cheong and Alex Veidenbaum. *Detection of Stale Copy Accesses and Fast Selective Cache Coherence Enforcement Using a Flow Analysis Approach. Proc. of 1988 Int'l. Conf. on Parallel Processing, Vol. I: Architecture*, pp. 138–145, August 1988.
- [Cont88] Thomas Martin Conte. "The Simulation and Tuning of the Global Memory Subsystem of a Multiprocessor", CSRD Report No. 821, UILU-ENG-88-

8013, M.S. Thesis, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., September 1988.

- [CyKM88] R. Cytron, S. Karlovsky and K. McAuliffe. *Automatic Management of Programmable Caches*. **Proc. of 1988 Int'l. Conf. on Parallel Processing**, St. Charles, IL, pp. 229-238, August 1988.
- [DaDa88] Timothy Davis and Edward Davidson. *Pairwise Reduction for the Direct, Parallel Solution of Sparse, Unsymmetric sets of Linear Equations*. **IEEE Trans. on Computers**, Vol. 37, No. 12, pp. 1648-1654, December 1988.
- [EmPa88] Perry Emrath and David Padua. *Automatic Detection of Nondeterminacy in Parallel Programs*. **Proc. of ACM SIGPLAN/SIGOPS Workshop on Parallel & Distributed Debugging**, Madison, WI, pp. 89-99, May 1988.
- [GaJG88] Kyle Gallivan, William Jalby and Dennis Gannon. *On the Problem of Optimizing Data Transfers for Complex Memory Systems*. **Proc. of 1988 Int'l. Conf. on Supercomputing**, St. Malo, France, pp. 238-253, July 1988.
- [GaLe88] E. Gallopoulos and Daeshik Lee. *Boundary Integral Domain Decomposition on Hierarchical Memory Multiprocessors*. **Proc. of 1988 Int'l. Conf. on Supercomputing**, St. Malo, France, pp. 488-499, July 1988.
- [GGGJ88] Vincent Guarna Jr., Dennis Gannon, Yogesh Gaur and David Jablonowski. *Faust: An Environment for Programming Scientific Applications*. **Proc. of Supercomputing '88**, pp. 3-10, November 1988.
- [GiPo88] Milind Girkar and Constantine Polychronopoulos. *Partitioning Programs for Parallel Execution*. **Proc. of 1988 ACM Int'l. Conf. on Supercomputing**, St. Malo, France, pp. 216-229, July 1988.
- [GiPo88] Milind Girkar and Constantine Polychronopoulos. *Compiler Issues for Supercomputers*. **Proc. of Supercomputing '88**, pp. 164-173, November 1988.
- [GJMS88] Kyle Gallivan, William Jalby, Ulrike Meier and Ahmed Sameh. *The Impact of Hierarchical Memory Systems on Linear Algebra Algorithm Design*. **Int'l. Jour. of Supercomputer Applications**, Vol. 2, No. 1, pp. 12-48, Spring 1988.
- [GKLS88] Kyle Gallivan, Peter Koss, S. Lo and R. A. Saleh. *A Comparison of Parallel Relaxation-Based Circuit Simulation Techniques*. **Professional Program Session Record 43: Simulation of Analog and Mixed-Mode Circuits**, Electro '88, pp. 1-6, May 1988.

- [GKPR88] M. M. Gooley, L. V. Kale, D. A. Padua, B. Ramkumar, U. S. Reddy, D. C. Sehr, W. W. Shu and B. W. Wah. *Prolog at the University of Illinois. Proc. of COMPCON '88*, pp. 68–73, 1988.
- [Guar88] Vincent Guarna Jr. *A Technique for Analyzing Pointer and Structure References in Parallel Restructuring Compilers. Proc. of 1988 Int'l. Conf. on Parallel Processing, St. Charles, IL*, pp. 212–220, August 1988.
- [GuGa88] Vince Guarna Jr. and Yogesh Gaur. *A Portable User Interface for a Scientific Programming Environment. Proc. of Siggraph Symp. on User Interface Software, Banff, Alta., Canada*, pp. 211–220, October 1988.
- [HaPa88] W. Ludwell Harrison III and David Padua. *PARCEL: Project for the Automatic Restructuring and Concurrent Evaluation of Lisp. Proc. of 1988 Int'l. Conf. on Supercomputing, St. Malo, France*, pp. 527–538, July 1988.
- [HuKP88] Harlan Husmann, David Kuck and David Padua. *Automatic Compound Function Definition for Multiprocessors. Proc. of 1988 Int'l. Conf. on Parallel Processing, St. Charles, IL*, pp. 34–41, August 1988.
- [Jaya88] D. N. Jayasimha. *Distributed Synchronizers. Proc. of 1988 Int'l. Conf. on Parallel Processing, St. Charles, IL*, pp. 23–27, August 1988.
- [Jaya88] Doddaballapur Narasimha-Murthy Jayasimha. "Communication and Synchronization in Parallel Computation", CSRD Report No. 819, UILU-ENG-88-8012, Ph.D. Thesis, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., September 1988.
- [KaPS88] Laxmikant Kale, David Padua and David Sehr. *OR Parallel Execution of Prolog Programs with Side Effects. Journal of Supercomputing*, Vol. 2, pp. 209–223, 1988.
- [LeeD88] Daeshik Lee. "Performance Enhancement of the Extra-Stage Multistage Interconnection Networks", CSRD Report No. 645, Univ. of Illinois at Urbana-Champaign, Cntr. for Supercomputing Res. & Dev., January 1988.
- [LeLe88] Kyungsook Y. Lee and Daeshik Lee. *On the Augmented Data Manipulator Network in SIMD Environments. IEEE Trans. on Computers*, Vol. 37, pp. 574–584, May 1988.
- [Lilj88] David Lilja. *Methods of Reducing the Branch Penalty in Pipelined Processors. Computer*, Vol. 21, No. 7, pp. 47–55, July 1988.

- [LiYe88] Zhiyuan Li and Pen-Chung Yew. *Interprocedural Analysis for Parallel Programs*. Proc. of 1988 Int'l. Conf. on Parallel Processing: Vol. II Software, St. Charles, IL, pp. 221-228, August 1988.
- [LiYe88] Zhiyuan Li and Pen-Chung Yew. *Program Parallelization with Interprocedural Analysis*. Jour. of Supercomputing, Vol. 2, pp. 225-244, 1988.
- [LiYe88] Zhiyuan Li and Pen-Chung Yew. *Efficient Interprocedural Analysis for Program Parallelization and Restructuring*. Proc. of ACM/SIGPLAN PPEALS, New Haven, CT, pp. 85-99, July 1988.
- [Malo88] Allen Malony. *Regular Processor Arrays*. Presented at 2nd Symposium on the Frontiers of Massively Parallel Computation, Fairfax, VA, October 1988.
- [Malo88] Allen D. Malony. *Multiprocessor Instrumentation: Approaches for Cedar*. Proc. of Workshop on Instrumentation for Future Parallel Systems, May 1988.
- [MaPi88] Allen D. Malony and Joseph R. Pickert. "An Environment Architecture and Its Use in Performance Data Analysis", CSRD Report No. 829, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., October 1988.
- [Marc88] David Michael Marcovitz. "A Multiprocessor Cache Performance Metric", CSRD Report No. 813, UILU-ENG-88-8011, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1988.
- [MaRe88] Allen D. Malony and Daniel A. Reed. *Visualizing Parallel Computer System Performance*. Proc. of Workshop on Instrumentation for Future Parallel Systems, Santa Fe, NM, May 1988.
- [MeSa88] Ulrike Meier and Ahmed Sameh. *The Behavior of Conjugate Gradient Algorithms on a Multivector Processor with a Hierarchical Memory*. Journal of Computational and Applied Mathematics, Vol. 24, pp. 13-32, 1988.
- [MiIy88] Samir G. Mitra and Ravishankar Iyer. "Measurement-Based Analysis of Multiple Errors and Near-Coincident Fault Discovery in a Shared Memory Multiprocessor", CSRD Report No. 744, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., February 1988.
- [Mitr88] Samir G. Mitra. "Measurement-Based Analysis of Multiple Errors and Near-Coincident Fault Discovery on a Shared Memory Multiprocessor", CSRD Report No. 743, UILU-ENG-88-8002, Univ. of Illinois at Urbana-

Champaign, Center for Supercomputing Res. & Dev., February 1988.

- [Padu88] David Padua. *The Cedar Parallel Processor: Machine Organization and Software. Speedup*, Vol. 2, No. 1, January 1988.
- [PaHK88] D. Padua, W. L. Harrison III and D. J. Kuck. *Languages and Compilers for Parallel Symbolic Computing*. In: **Biological and Artificial Intelligence Systems**, E. Clementi and S. Chin, ed. ESCOM, Leiden, The Netherlands, pp. 453–461, 1988.
- [Poly88] Constantine Polychronopoulos. *The Impact of Run-Time Overhead on Usable Parallelism*. **Proc. of 1988 Int'l. Conf. on Parallel Processing**, St. Charles, IL, pp. 108–112, August 1988.
- [Poly88] Constantine Polychronopoulos. *Compiler Optimizations for Enhancing Parallelism and Their Impact on Architecture Design*. **IEEE Trans. on Computers**, Vol. C-37, No. 8, pp. 991–1004, August 1988.
- [Poly88] Constantine Polychronopoulos. *Toward Auto-Scheduling Compilers*. **Journal of Supercomputing**, May 1988.
- [ReMa88] Daniel A. Reed and Allen Malony. *Parallel Discrete Event Simulation: The Chandy-Misra Approach*. **Proc. of SCS Multiconference on Distributed Simulation**, San Diego, CA, pp. 8–13, February 1988.
- [Saad88] Youcef Saad. *Preconditioning and Indefinite Linear Systems*. **Special Issue of Jour. of Computational & Applied Mathematics (CAM)**, May 1988.
- [SaWi88] Youcef Saad and Harry Wijshoff. *Benchmark Package for Sparse Matrix Computations*. **Proc. of 1988 Int'l. Conf. on Supercomputing**, St. Malo, France, pp. 500–509, July 1988.
- [Scro88] Jeffrey S. Scroggs. "Solution of a Parabolic Partial Differential Equation via Domain Decomposition: The Synthesis of Asymptotic and Numerical Analysis", CSRD Report No. 791, UILU-ENG-88-8005, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1988.
- [Sehr88] David Christopher Sehr. "OR-Parallel Execution of Prolog Programs with Side Effects", CSRD Report No. 822, UILU-ENG-88-8014, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., October 1988.
- [SRSK88] Wennie Shu, Balkrishma Ramkumar, David Sehr and Laxmicant Kale. "The

- REDUCE-OR Process Model for Logic Programs on Parallel Machines", CSRD Report No. 737, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1988.
- [Staf88] CSRD Staff. "Papers Presented at the Third Siam Conference on Parallel Processing for Scientific Computing", CSRD Report No. 747, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., February 1988.
- [TaDa88] Ju-Ho Tang and Edward S. Davidson. *An Evaluation of Cray-1 and Cray X-MP Performance on Vectorizable Livermore Fortran Kernels. Proc. of 1988 Int'l. Conf. on Supercomputing, St. Malo, France, pp. 510-518, July 1988.*
- [Tang88] Peiyi Tang. "Self-Scheduling, Data Synchronization and Program Transformation for Multiprocessor Systems", CSRD Report No. 809, UILU-ENG-88-8009, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., September 1988.
- [TaYZ88] Peiyi Tang, Pen-Chung Yew and Chuan-Qi Zhu. *Impact of Self-Scheduling Orders on Performance. Proc. of 1988 Int'l. Conf. on Supercomputing, St. Malo, France, pp. 593-603, July 1988.*
- [TzYZ88] Nian-Feng Tzeng, Pen-Chung Yew and Chuan-Qi Zhu. *Realizing Fault-Tolerant Interconnection Networks via Chaining. Special Issue on Fault-Tolerant Computing, IEEE Trans. on Computers, Vol. 37, No. 4, pp. 458-462, April 1988.*
- [XiaE88] Eugene Zhu Xia. "Parallel Waveform-Relaxation-Newton for Circuit Simulation", CSRD Report No. 772, UILU-ENG-88-8004, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1988.
- [YewP88] Pen-Chung Yew. *Architecture of the Cedar Parallel Supercomputer. In: Parallel Systems and Computation, G.S. Almasi G. Paul, ed. North-Holland, Amsterdam, pp. 137-148, 1988.*
- [Zhon88] Jialin Zhong. "Computer Visualization of Electron Movement in Gallium Arsenide Simulation", CSRD Report No. 797, UILU-ENG-88-8006, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1988.
- [AbPa87] Santosh Abraham and Janak Patel. *Parallel Garbage Collection on a Virtual Memory System. Proc. of 1987 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 243-246, August 19-22, 1987.*

- [AlPa87] Todd R. Allen and David Padua. *Debugging Parallel Fortran on a Shared Memory Machine*. Proc. of 1987 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 721-727, August 19-22, 1987.
- [AnSa87] Edward Anderson and Youcef Saad. *Preconditioned Conjugate Gradient Methods for General Sparse Matrices on Shared Memory Machines*. Presented at 3rd SIAM Conf. on Parallel Processing for Scientific Computing, Los Angeles, CA, December 1-4, 1987.
- [BePl87] Michael Berry and Robert Plemmons. *Algorithms and Experiments for Structural Mechanics on High Performance Architecture*. Computer Methods in Applied Mechanics & Engrg, Vol. 64, pp. 487-507, 1987.
- [BeSa87] Michael Berry and Ahmed Sameh. *A Multiprocessor Scheme for the Singular Value Decomposition*. In: Parallel Processing for Scientific Computing, G. Rodrigue, ed. SIAM, pp. 67-71, 1987.
- [Bren87] Glen Alan Brent. "Using Program Structure to Achieve Prefetching for Cache Memories", CSR D Rpt. No. 647, Univ. of Illinois at Urbana-Champaign, Ctr. for Supercomputing Res. & Dev., January 1987.
- [BrSa87] P. N. Brown and Y. Saad. "Hybrid Krylov Methods for Nonlinear Systems of Equations", CSR D Report No. 699, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., November 1987.
- [ChHa87] M. Q. Chen and S. P. Han. "A Parallel Quasi-Newton Method for Partially Separable Large Scale Minimization", CSR D Report No. 689, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., April 1987.
- [ChSa87] Hsin-Chu Chen and Ahmed Sameh. *Numerical Linear Algebra Algorithms on the Cedar System*. In: Parallel Computations and Their Impact on Mechanics, A. Noor, ed. The American Society of Mechanical Engineering, pp. 101-125, 1987.
- [CHSS87] R. C. Y. Chin, G. W. Hedstrom, J. S. Scroggs and D. C. Sorensen. *Parallel Computation of a Domain Decomposition Method*. Proc. of Sixth IMACS Int'l. Symp. on Computer Methods for Partial Differential Equations, March 1987.
- [ChVe87] Hoichi Cheong and Alex Veidenbaum. *The Performance of a Software-Managed Multiprocessor Caches on Parallel Numerical Programs*. In: Lecture Notes in Computer Science No. 297: Proc. of First Int'l. Conf. on Supercomputing, Athens, Greece, T.S. Papatheodorou E.N.

Houstis C.D. Polychronopoulos, ed. Springer-Verlag, New York, NY, pp. 316-337, 1987.

- [Defe87] Daniel Defend. "Parallel Ray Tracing in a Vector-Multiprocessing Environment", CSR D Report No. 677, UILU-ENG-87-8005, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1987.
- [DoSo87] J. J. Dongarra and D. C. Sorensen. *A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem*. SIAM Journ. on Scientific & Statistical Computing, Special Issue on Parallel Processing, Norfolk Mtg., Vol. 8, No. 2, pp. s139-s154, March, 1987.
- [FTYZ87] Zhixi Fang, Peiyi Tang, Pen-Chung Yew and Chuan-Qi Zhu. *Dynamic Processor Self-Scheduling for General Parallel Nested Loops*. Proc. of 1987 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 1-10, August 19-22, 1987.
- [GaEK87] E. Gallopoulos, O. Egecioglu and C. Koc. "Fast and Practical Parallel Polynomial Interpolation", CSR D Report No. 646, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1987.
- [GaJa87] Dennis Gannon and William Jalby. *The Influence of Memory Hierarchy on Algorithm Organization: Programming FFTs on a Vector Multiprocessor*. In: *Characteristics of Parallel Algorithms*, D. Gannon, L. Jamison and B. Douglas, eds. MIT Press, Cambridge, MA, 1987.
- [GaJM87] K. Gallivan, W. Jalby and U. Meier. *The Use of BLAS3 in Linear Algebra on a Parallel Processor with a Hierarchical Memory System*. SIAM J. Sci. Stat. Comput., Vol. 8, No. 6, pp. 1079-1084, November 1987.
- [GaSa87] E. Gallopoulos and Youcef Saad. *A Parallel Block Cyclic Reduction Algorithm for the Fast Solution of Elliptic Equations*. In: *Lecture Notes in Computer Science No. 297: Proc. of First Int'l. Conf. on Supercomputing, Athens, Greece*, T.S. Papatheodorou E.N. Houstis C.D. Polychronopoulos, ed. Springer-Verlag, New York, NY, pp. 563-575, 1987.
- [GiSo87] Milind Girkar and Milind Sohoni. "On Finding the Vertex Connectivity of Graphs", CSR D Report No. 669, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1987.
- [Guar87] Vincent A. Guarna Jr. "Analysis of C Programs for Parallelization in the Presence of Pointers", CSR D Report No. 695, UILU-ENG-87-8007, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev.,

December 1987.

- [Guar87] Vincent A. Guarna Jr. "VPC - A Proposal for a Vector Parallel C Programming Language", CSRD Report No. 666, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1987.
- [GuMa87] Vincent Guarna and Allen Malony. *Instrumentation for the Development of Parallel Programs. Presented at IEEE/ACM Sigarch Workshop on Instrumentation for Distributed Computing Systems, Sanibel Island, FL, January 1987.*
- [Guzz87] Mark Guzzi. "Cedar Fortran Programmer's Handbook", CSRD Report No. 601, Univ. of Illinois at Urbana-Champaign, Ctr. for Supercomputing Res. & Dev., June 1987.
- [HsYZ87] William Hsu, Pen-Chung Yew and Chuan-Qi Zhu. *An Enhancement Scheme for Hypercube Interconnection Networks. Proc. of 1987 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 820-823, August 19-22, 1987.*
- [Jaya87] D. N. Jayasimha. *Parallel Access to Synchronization Variables. Proc. of 1987 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 97-100, August 19-22, 1987.*
- [Kamg87] Emmanuel R. Kamgnia. "The von-Neumann-Ulam Monte Carlo Method for Solving Systems of Linear Algebraic Equations", Subm. for publ. to Jour. Parallel and Dist. Comp., July 1987.
- [KDLS87] David Kuck, Edward Davidson, Duncan Lawrie and Ahmed Sameh. *Parallel Supercomputing Today and the Cedar Approach. In: Experimental Parallel Computing Architectures, J. J. Dongarra, ed. Elsevier Science Publishers B.V. (North-Holland), New York, NY, pp. 1-20, 1987.*
- [KeSa87] T. Kerkhoven and Youcef Saad. "Acceleration Techniques for Decoupling Algorithms in Semiconductor Simulation", CSRD Report No. 686, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1987.
- [Koni87] Kris Gibson Konigsfeld. "An Ethernet Local Area Network Controller with Supporting Operating System Software", CSRD Report No. 668, UILU-ENG-87-8004, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1987.
- [Kwok87] Alex Yuen-Wai Kwok. "A Performance Analysis of Architectural

- Scalability", CSRD Report No. 679, UILU-ENG-87-8006, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1987.
- [LeeD87] Daeshik Lee. *The Lambda Network: A New Permutation Network*. **Proc. of 1987 Int'l. Conf. on Parallel Processing**, St. Charles, IL, pp. 407-410, August 19-22, 1987.
- [LeeR87] Roland Lun Lee. "The Effectiveness of Caches and Data Prefetch Buffers in Large-Scale Shared Memory Multiprocessors", CSRD Report No. 670, UILU-ENG-87-8005, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1987.
- [LeYL87] Roland Lee, Pen-Chung Yew and Duncan H. Lawrie. *Data Prefetching in Shared Memory Multiprocessors*. **Proc. of 1987 Int'l. Conf. on Parallel Processing**, St. Charles, IL, pp. 28-31, August 19-22, 1987.
- [LeYL87] Roland Lee, Pen-Chung Yew and Duncan H. Lawrie. *Multiprocessor Cache Design Considerations*. **Proc. of 14th Int'l. Symp. on Computer Architecture**, Pittsburgh, PA, pp. 253-262, June 1987.
- [LiAb87] Zhiyuan Li and Walid Abu-Sufah. *On Reducing Data Synchronization in Multiprocessed Loops*. **IEEE Trans. on Computers**, Vol. C-36, No. 1, pp. 105-109, Jan., 1987.
- [LoPS87] Sy-Shin Lo, Bernard Philippe and Ahmed H. Sameh. *A Multiprocessor Algorithm for the Symmetric Tridiagonal Eigenvalue Problem*. **SIAM Journ. on Scientific and Statistical Computing**, Vol. 8, No. 2, pp. s155-s165, March, 1987.
- [MaRM87] Allen D. Malony, Daniel A. Reed and Patrick J. McGuire. *MPF: A Portable Message Passing Facility for Shared Memory Multiprocessors*. **Proc. of 1987 Int'l. Conf. on Parallel Processing**, pp. 739-741, August 1987.
- [McEm87] Robert McGrath and Perry Emrath. *Using Memory in the Cedar System*. In: **Lecture Notes in Computer Science No. 297: Proc. of First Int'l. Conf. on Supercomputing**, Athens, Greece, T.S. Papatheodorou E.N. Houstis C.D. Polychronopoulos, ed. Springer-Verlag, New York, NY, pp. 43-67, 1987.
- [McGu87] Patrick John McGuire. "A Measurement-Based Study of Concurrency in a Multiprocessor", CSRD Report No. 674, UILU-ENG-87-2210, CSG-61, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1987.

- [Mill87] Alan Ray Miller. "Non-Preemptive Run-Time Scheduling Issues on a Multi-tasked, Multiprogrammed Multiprocessor with Dependencies, Bidimensional Tasks, Folding, and Dynamic Graphs", CSRD Report No. 656, UILU-ENG-87-8002, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1987.
- [MiPa87] Samuel Midkiff and David Padua. *Compiler Algorithms for Synchronization*. **IEEE Trans. on Computers**, Vol. C-36, No. 12, pp. 1485-1495, December 1987.
- [PaGL87] David A. Padua, Vincent A. Guarna Jr. and Duncan H. Lawrie. *Supercomputer Programming Environments*. **Proc. of Symp. on Parallel Computations and Their Impact on Mechanics**, Boston, MA, pp. 55-79, December 1987.
- [PoBa87] Constantine Polychronopoulos and Utpal Banerjee. *Processor Allocation for Horizontal and Vertical Parallelism and Related Speedup Bounds*. **IEEE Trans. on Computers**, Vol. 36, No. 4, April 1987.
- [PoKu87] Constantine Polychronopoulos and David J. Kuck. *Guided Self-Scheduling: A Practical Scheduling Scheme for Parallel Supercomputers*. **IEEE Trans. on Computers**, Vol. C-36, No. 12, pp. 1425-1439, December 1987.
- [Poly87] Constantine Polychronopoulos. *Loop Coalescing: A Compiler Transformation for Parallel Machines*. **Proc. of 1987 Int'l. Conf. on Parallel Processing**, St. Charles, IL, pp. 235-242, August 19-22, 1987.
- [Poly87] Constantine Polychronopoulos. "Compile-Time Reduction of Interprocessor Communication for Parallel Computers", CSRD Report No. 642, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., January 1987.
- [Poly87] Constantine Polychronopoulos. "More on Advanced Loop Optimizations", CSRD Report No. 667, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., September 1987.
- [Poly87] Constantine Polychronopoulos. *Advanced Loop Optimizations for Parallel Computers*. In: **Lecture Notes in Computer Science No. 297: Proc. of First Int'l. Conf. on Supercomputing**, Athens, Greece, T.S. Papatheodorou E.N. Houstis C.D. Polychronopoulos, ed. Springer-Verlag, New York, NY, pp. 255-277, 1987.
- [Saad87] Youcef Saad. *On the Design of Parallel Numerical Methods in Message Passing and Shared Memory Environments*. **Proc. of Int'l. Seminar on**

Scientific Supercomputers, Paris, France, pp. 217-238, February 2-6, 1987.

- [Scha87] Mark Johannes Schaefer. "A Polynomial Based Iterative Method for Linear Parabolic Equations", CSRD Report No. 661, UILU-ENG-87-8003, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1987.
- [Skee87] Robert Skeel. "Waveform Iteration and the Shifted Picard Splitting", CSRD Report No. 700, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., November 1987.
- [Staf87] CSRD Staff. "Center for Supercomputing Research and Development Quarterly Report: Second Quarter 1987", CSRD Report No. 681, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1987.
- [Staf87] CSRD Staff. "Center for Supercomputing Research and Development Quarterly Report: First Quarter 1987", CSRD Report No. 684, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1987.
- [Szc87] Joseph Szczypek. "A Fault-Diagnosis Scheme for the Cedar Processor-Memory Interconnection Network", CSRD Report No. 716, UILU-ENG-87-8008, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., December 1987.
- [TYFZ87] Peiyi Tang, Pen-Chung Yew, Zhixi Fang and Chuan-Qi Zhu. *Deadlock Prevention in Processor Self-Scheduling for Parallel Nested Loops. Proc. of 1987 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 11-18, August 19-22, 1987.*
- [UhPK87] A. Uht, C. Polychronopoulos and J. F. Kolen. *On the Combination of Hardware and Software Concurrency Extraction Methods. Proc. of ACM Workshop on Microprogramming, Fort Collins, CO, December 1-4, 1987.*
- [YeTL87] Pen-Chung Yew, Nian-Feng Tzeng and Duncan H. Lawrie. *Distributing Hot-Spot Addressing in Large Scale Multiprocessors. IEEE Trans. on Computers, Vol. C-36, No. 4, pp. 388-395, April, 1987.*
- [AbDa86] Santosh G. Abraham and Edward S. Davidson. *A Communication Model for Optimizing Hierarchical Multiprocessor Systems. Proc. of the 1986 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 467-474, Aug. 19-*

22, 1986.

- [AbDa86] S. Abraham and E. S. Davidson. "Task Assignment Using Network Flow Methods for Minimizing Communication in n-Processor Systems", CSRD Report No. 598, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Sept. 3, 1986.
- [AbHK86] Walid Abu-Sufah, Harlan Husmann and David Kuck. *On Input/Output Speed-up in Tightly-Coupled Multiprocessors*. **IEEE Trans. on Computers**, Vol. C-35, No. 6, pp. 520-530, June 1986.
- [AbMa86] Walid Abu-Sufah and Allen D. Malony. "Experimental Results for Vector Processing on the Alliant FX/8", CSRD Rpt. No. 539, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Feb. 11, 1986.
- [AbMa86] Walid Abu-Sufah and Allen D. Malony. *Vector Processing on the Alliant FX/8 Multiprocessor*. **Proc. of the 1986 Int'l. Conf. on Parallel Processing**, St. Charles, IL, pp. 559-566, Aug. 19-22, 1986.
- [AbPa86] Santosh Abraham and Janek Patel. "Parallel Garbage Collection on a Virtual Memory System", CSRD Report No. 620, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Pittsburgh, PA, December 1986.
- [Bane86] Utpal Banerjee. "Direct Parallelization of Call Statements - A Review", CSRD Rpt. No. 576, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., April 1986.
- [BeSa86] Michael Berry and Ahmed Sameh. *Multiprocessor Jacobi Algorithms for Dense Symmetric Eigenvalue and Singular Value Decompositions*. **Proc. of the 1986 Int'l Conf. on Parallel Processing**, St. Charles, IL, pp. 433-440, Aug. 19-22, 1986.
- [BGHJ86] M. Berry, K. Gallivan, W. Harrod, W. Jalby, S. Lo, U. Meier, B. Philippe and A. H. Sameh. *Parallel Algorithms on the CEDAR System*. In: **CONPAR 86, Lecture Notes in Computer Science**, W. Handler, ed. et. al. Springer-Verlag, pp. 25-39, 1986.
- [CyrJ86] J. Cyr. "The Structured Memory Access Architecture: An Implementation and Performance Evaluation", CSRD Rpt. No. 597, UILU-ENG-86-8008, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1986.

- [Davi86] Timothy Davis. "PSolve: A Concurrent Algorithm for Solving Sparse Systems of Linear Equations", CSRD Report No. 612, Univ. of Illinois at Urbana-Champaign, Ctr. for Supercomputing Res. & Dev., December 1986.
- [DKLS86] E. Davidson, D. Kuck, D. Lawrie and A. Sameh. *Supercomputing Tradeoffs and the Cedar System*. In: **High-Speed Computing: Scientific Applications and Algorithm Design**, R. Wilhelmson, ed. University of Illinois Press, pp. 3-11, 1986.
- [Gann86] Dennis Gannon. "Restructuring Nested Loops on the Alliant Cedar Cluster: A Case Study of Gaussian Elimination of Banded Matrices", CSRD Rpt. No. 543, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., February 1986.
- [GoPS86] G. H. Golub, R. J. Plemmons and A. Sameh. *Parallel Block Schemes for Large Scale Least Squares Computations*. In: **High-Speed Computing: Scientific Applications and Algorithm Design**, R. Wilhelmson, ed. University of Illinois Press, pp. 171-179, 1986.
- [Guzz86] Mark Guzzi. "Multitasking Runtime Systems for the Cedar Multiprocessor", CSRD Rpt. No. 604, UILU-ENG-86-8009, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., July 1986.
- [HaPa86] W. Ludwell Harrison III and David A. Padua. *Representing S-Expressions for the Efficient Evaluation of Lisp on Parallel Processors*. **Proc. of the 1986 Int'l. Conf. on Parallel Processing**, St. Charles, IL, pp. 703-710, Aug. 19-22, 1986.
- [Harr86] W. Ludwell Harrison III. "Compiling Lisp for Evaluation on a Tightly Coupled Multiprocessor", CSRD Rpt. No. 565, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Mar. 20, 1986.
- [Husm86] Harlan Husmann. "Compiler Memory Management and Compound Function Definition for Multiprocessors", CSRD Rpt. No. 575, UILU-ENG-86-8007, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1986.
- [JaMe86] William Jalby and Ulrike Meier. *Optimizing Matrix Operations on a Parallel Multiprocessor with a Memory Hierarchy*. **Proc. of the 1986 Int'l. Conf. on Parallel Processing**, St. Charles, IL, pp. 429-432, Aug. 19-22, 1986.
- [JaMS86] William Jalby, Ulrike Meier and Ahmed Sameh. "The Behaviour of Conjugate Gradient Based Algorithms on a Multi-Vector Processor with a

- Memory Hierarchy", CSR D Report No. 607, Univ. of Illinois at Urbana-Champaign, Ctr. for Supercomputing Res. & Dev., November 23, 1986.
- [Kama86] C. Kamath. "Solution of Nonsymmetric Systems of Equations on a Multiprocessor", CSR D Rpt. No. 591, UILU-ENG-86-8004, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1986.
- [KDLS86] David J. Kuck, Edward S. Davidson, Duncan H. Lawrie and Ahmed H. Sameh. *Parallel Supercomputing Today and the Cedar Approach*. Science, Vol. 231, pp. 967-974, Feb. 28, 1986.
- [Kolc86] Alexander Kolchinsky. "Emulating Multitasking System Calls for a Multicluster Machine", CSR D Rpt. No. 560, UILU-ENG-86-8001, Univ. of Illinois at Urbana-Champaign, Dept. of Comput. Sci., May, 1986.
- [LeeG86] G. Lee. "Some Issues in General-Purpose Shared Memory Multiprocessing: Parallelism Exploitation and Memory Access Combining", CSR D Rpt. No. 589, UILU-ENG-86-6002, Univ. of Illinois at Urbana-Champaign, Ctr. for Supercomputing Res. & Dev., June 1986.
- [LeKK86] Gyungho Lee, Clyde P. Kruskal and David J. Kuck. *The Effectiveness of Combining in Shared Memory Parallel Computers in the Presence of 'Hot Spot'*. Proc. of the 1986 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 35-41, Aug. 19-22, 1986.
- [LoPh86] Sy-Shin Lo and Bernard Philippe. *The Symmetric Eigenvalue Problem on a Multiprocessor*. In: **Parallel Algorithms & Architectures**. North-Holland, Amsterdam, The Netherlands, pp. 31-43, 1986.
- [Malo86] Allen Malony. "Cedar Performance Measurements", CSR D Rpt. No. 579, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 25, 1986.
- [Malo86] A. Malony. "Cedar Performance Evaluation Tools: A Status Report", CSR D Rpt. No. 582, Univ. of Illinois at Urbana-Champaign, Ctr. for Supercomputing Res. & Dev., July 21, 1986.
- [Midk86] S. Midkiff. "Automatic Generation of Synchronization Instructions for Parallel Processors", CSR D Rpt. No. 588, UILU-ENG-86-6002, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., May 1986.
- [MiPa86] Samuel P. Midkiff and David A. Padua. *Compiler Generated Synchronization*

for DO Loops. Proc. of the 1986 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 544-551, Aug. 19-22, 1986.

- [Nand86] N. R. Nandakumar. "Polynomial Preconditioning of Symmetric Indefinite Systems", CSR D Rpt. No. 580, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1986.
- [PaWo86] D. Padua and M. Wolfe. *Advanced Compiler Optimization for Supercomputers*. CACM, Vol. 29, No. 12, pp. 1184-1201, December, 1986.
- [PoBa86] Constantine D. Polychronopoulos and Utpal Banerjee. *Speedup Bounds and Processor Allocation for Parallel Programs on Multiprocessors. Proc. of the 1986 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 961-968, Aug. 19-22, 1986.*
- [PoKP86] Constantine D. Polychronopoulos, David J. Kuck and David A. Padua. *Execution of Parallel Loops on Parallel Processor Systems. Proc. of the 1986 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 519-527, Aug. 19-22, 1986.*
- [Poly86] Constantine Polychronopoulos. "On Program Restructuring, Scheduling, and Communication for Parallel Processor Systems", CSR D Rpt. No. 595, UILU-ENG-86-8006, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., August 1986.
- [PSKD86] Andrew R. Pleszkun, Gurindar S. Sohi, Bassam Z. Kahhaleh and Edward S. Davidson. *Features of the Structured Memory Access (SMA) Architecture. Proc. of Spring COMPCON 86, pp. 259-263, Mar. 3-6, 1986.*
- [Swar86] Paul Swarztrauber. "Multiprocessor FFTs", CSR D Rpt. No. 608, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1986.
- [TaYe86] Peiyi Tang and Pen-Chung Yew. *Processor Self-Scheduling for Multiple-Nested Parallel Loops. Proc. of the 1986 Int'l. Conf. on Parallel Processing, St. Charles, IL, pp. 528-535, Aug. 19-22, 1986.*
- [Trio86] Remi Triolet. "Interprocedural Analysis for Program Restructuring with Parafrase", CSR D Rpt. No. 538, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Aug., 1986.
- [Tzen86] N. F. Tzeng. "Fault Tolerant Multiprocessor Interconnection Networks and Their Fault-Diagnoses", CSR D Rpt. No. 594, UILU-ENG-86-8005, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev.,

August 1986.

- [TzYZ86] Nian-Feng Tzeng, Pen-Chung Yew and Chuan-Qi Zhu. *Fault-Diagnosis in a Multiple-Path Interconnection Network*. **Proc. of The 16th Int'l. Symp. on Fault-Tolerant Computing**, pp. 98-103, July 1-3, 1986.
- [Veid86] Alex Veidenbaum. *A Compiler-Assisted Cache Coherence Solution for Multiprocessors*. **Proc. of the 1986 Int'l. Conf. on Parallel Processing, St. Charles, IL**, pp. 1029-1036, Aug. 19-22, 1986.
- [YeTL86] Pen-Chung Yew, Nian-Feng Tzeng and Duncan H. Lawrie. *Distributing Hot Spot Addressing in Large Scale Multiprocessors*. **Proc. of the 1986 Int'l. Conf. on Parallel Processing, St. Charles, IL**, pp. 51-58, Aug. 19-22, 1986.
- [AbKw85] Walid Abu-Sufah and Alex Y. Kwok. *Performance Prediction Tools for Cedar: A Multiprocessor Supercomputer*. **Proc. of the 12th Int'l. Symp. on Computer Architecture**, pp. 406-413, June 17-19, 1985.
- [AbuS85] Walid Abu-Sufah. "Enhancing the Performance of Small Grain Multiprocessors", Rpt. on NBS Workshop on Performance Evaluation of High-Speed Computers, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Gaithersburg, MD, June 5-6, 1985.
- [Emra85] Perry Emrath. *Xylem: An Operating System for the Cedar Multiprocessor*. **IEEE Software**, Vol. 2, No. 4, pp. 30-37, July, 1985.
- [Jack85] Daniel Thomas Jackson. "Data Movement in Doall Loops", CSRD Rpt. No. 524, UILU-ENG-85-8014, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Aug., 1985.
- [KaSa85] E. Kamgnia and A. Sameh. *A Numerical Conformal Mapping Method for Simply Connected Domains*. **Second SIAM Conf. on Parallel Processing for Scientific Computing**, Nov. 18-21, 1985.
- [Kuck85] David J. Kuck. *Supercomputers and Distributed Computing*. **Proc. of the 1985 ACM Thirteenth Annual Computer Science Conf. (CSC '85)**, pp. 34-46, Mar. 12-14, 1985.
- [LeKK85] Ghungho Lee, Clyde P. Kruskal and David J. Kuck. *The Effectiveness of Automatic Restructuring on Nonnumerical Programs*. **Proc. of the 1985 Int'l. Conf. on Parallel Processing**, pp. 607-613, August 20-23, 1985.
- [LeKK85] Gyungho Lee, Clyde P. Kruskal and David J. Kuck. *An Empirical Study of*

Automatic Restructuring of Nonnumerical Programs for Parallel Processors. Special Issue on Parallel Processing of IEEE Trans. on Computers, Vol. C-34, No. 10, pp. 927-933, Oct., 1985.

- [LiAb85] Zhiyuan Li and Walid Abu-Sufah. *A Technique for Reducing Synchronization Overhead in Large Scale Multiprocessors. Proc. of the 12th Int'l. Symp. on Computer Architecture*, pp. 284-291, June 17-19, 1985.
- [LiZh85] Zhiyuan Li. "A Technique for Reducing Data Synchronization in Multiprocessed Loops", CSRD Rpt. No. 521, UILU-ENG-85-8013, Univ. of Illinois at Urbana-Champaign, Dept. of Computer Sci., May, 1985.
- [LuLa85] Stephen F. Lundstrom and Duncan H. Lawrie. *Experiences with Distributed Systems. Special Issue of IEEE Software*, Vol. 2, No. 3, pp. 5-6, May, 1985.
- [McLa85] Timothy Alan McDaniel. "Non-Linear Recurrences and EISPACK", CSRD Rpt. No. 511, UILU-ENG-85-8011, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Oct., 1985.
- [PaLa85] Krishnan Padmanabhan and Duncan H. Lawrie. *Performance Analysis of Redundant-Path Networks for Multiprocessor Systems. ACM Trans. on Computer Sysys.*, Vol. 3, No. 2, pp. 117-144, May, 1985.
- [Pawl85] Laura Marie Pawlowski. "Conversion of Parafrase Output to Alliant's Concurrent Fortran", CSRD Rpt. No. 532, UILU-ENG-85-8009, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., Dec., 1985.
- [Phil85] Bernard Philippe. "Approximating the Square Root of the Inverse of a Matrix", CSRD Rpt. No. 508, UILU-ENG-85-8009, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., July 1985.
- [Same85] A. Sameh. *Parallel Linear System Solvers. Proc. of the Conf. on Vector and Parallel Processors for Scientific Computation*, May 27-29, 1985.
- [Same85] Ahmed H. Sameh. *Algorithms and Experiments for Parallel Linear Systems Solvers. Second SIAM Conf. on Parallel Processing for Scientific Computing*, Nov. 18-21, 1985.
- [TzYZ85] Nian-Feng Tzeng, Pen-Chung Yew and Chuan-Qi Zhu. *A Fault Tolerant Scheme for Multistage Interconnection Networks. Proc. of the 12th Int'l. Symp. on Computer Architecture*, pp. 368-375, June 17-19, 1985.

- [TzYZ85] Nian-Feng Tzeng, Pen-Chung Yew and Chuan-Qi Zhu. *The Performance of a Fault-Tolerant Multistage Interconnection Network*. **Proc. of the 1985 Int'l. Conf. on Parallel Processing**, pp. 458-465, Aug. 20-23, 1985.
- [Veid85] Alexander Veidenbaum. "Compiler Optimizations and Architecture Design Issues for Multiprocessors", CSRD Rpt. No. 85-520, UILU-ENG-85-8012, Univ. of Illinois at Urbana-Champaign, Dept. of Comput. Sci., May, 1985.
- [WoLD85] Michael Wolfe, Bruce Leasure and James Davies. "Understanding Paraphrase Output Listings", CSRD Rpt. No. 505, PR-85-6, UILU-ENG-85-8006, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., July 9, 1985.
- [AbuS84] Walid Abu-Sufah. *Cedar Performance Evaluation Tools*. **Proc. of Int'l. Workshop on Modeling and Performance Evaluation of Parallel Systems**, pp. 472-472, Dec. 13-18, 1984.

END

**DATE
FILMED**

10/2/91

II