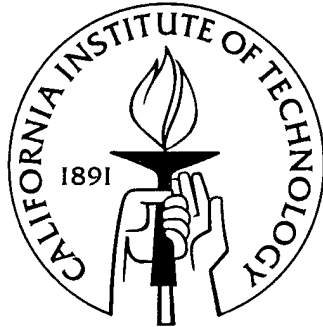


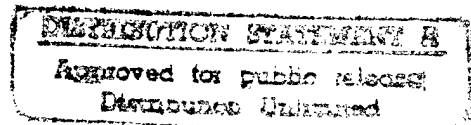
Final Technical Report
on
SCALABLE CONCURRENT PROGRAMMING PROJECT
Scalable Concurrent Programming Laboratory



Department of Computer Science
California Institute of Technology

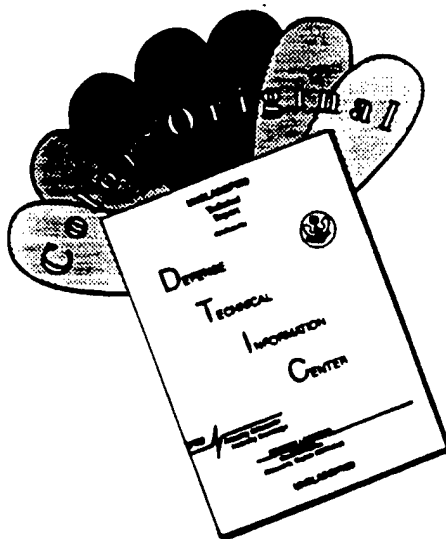
Office of Naval Research
Grant Number N00014-91-J-1986
Period July 1, 1991 through June 30, 1996

Principal Investigator: Dr. Stephen Taylor
Address: Scalable Concurrent Programming Laboratory
MS 256-80, California Institute of Technology
Pasadena, CA 91125
Email: steve@scp.caltech.edu
Phone/FAX: (818)395-3903 / (818)792-4257



19961009 114

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 9/23/96	3. REPORT TYPE AND DATES COVERED FINAL Technical 7/1/91- 6/30/96	
4. TITLE AND SUBTITLE Scalable Concurrent Programming Project			5. FUNDING NUMBERS G: N00014-91-J-1986	
6. AUTHOR(S) Dr. Stephen Taylor			8. PERFORMING ORGANIZATION REPORT NUMBER 61065-610650	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) California Institute of Technology Scalable Concurrent Programming Laboratory M/C 256-80 Pasadena, CA 91125			10. SPONSORING / MONITORING AGENCY REPORT NUMBER Project Code:4332741---01	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Department of the Navy Office of the Chief of Naval Research 800 North Quincy Street, Code 1511:TAP Arlington, VA 22217-5000			11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION / AVAILABILITY STATEMENT DISSEMINATION STATEMENT IS Approved for public release Distribution Unlimited			12b. DISTRIBUTION CODE N00179	
13. ABSTRACT (Maximum 200 words) See attached				
14. SUBJECT TERMS Scalable Concurrent Programming Irregular CMOS DeltaII Application Independent Three Dimensional				15. NUMBER OF PAGES
17. SECURITY CLASSIFICATION OF REPORT				16. PRICE CODE
18. SECURITY CLASSIFICATION OF THIS PAGE		19. SECURITY CLASSIFICATION OF ABSTRACT		20. LIMITATION OF ABSTRACT

Final Report

Proposed at a time when the community was primarily concerned with statically structured problems in scientific computing, the Scalable Concurrent Programming Project was motivated by the following general perceptions:

- For much of this century, scientific computing has been dominated by large regular problems and their associated computational techniques. This has been a direct result of relatively poor uniprocessor performance, simplistic software tools, and the associated education from previous generations.
- Device technology is rapidly approaching the physical limits of CMOS technology. During the next decade vast improvements in uniprocessor performance will be increasingly difficult and expensive to obtain. Concurrency is the door to the future; *Scalability* is the key.
- As performance improves, we will not solve the same problems faster, but rather we will solve completely new and more realistic computational problems. These *irregular problems* will combine computations in structures, materials, fluids, and electromagnetics. They will be three-dimensional in nature, involve complex moving boundaries, and resolve transient effects. They will become the cornerstone of design, test, and manufacturing in a broad range of industries currently dominated by empiricism.

With these perceptions in mind, the laboratory has been involved in two pioneering studies that concern *irregular concurrent programming problems* and *scalable software systems*.

The study of irregular problems has served to push the envelope of concurrent programming, drive the development of software tools, and provide realistic tests for the evaluation of experimental architectures such as the MIT J Machine and the Avalon A12. Irregular problems, when mapped to parallel machines, involve complex communication patterns with dynamic partitioning and load-balancing characteristics. At the beginning of the project, isolated attempts had been made to solve specific irregular problems on parallel machines. No cohesive *application independent* framework, founded on mathematics, existed for dealing with this general class of calculations. The central issues of scalability, portability, and maintainability were generally handled through ad-hoc methods, rather than an organized and systematic approach.

The study of scalable software systems has served to evaluate the lowest common denominator of scalable concurrent programming models: message-driven programming. This simple concept admits to a broad range of implementation techniques, including pointer copying on shared-memory machines and message-passing on distributed systems. The appearance of experimental hardware implementations, such as that provided by the MIT J-machine, motivated an integrated evaluation of both architectures and software systems based on the message-driven concept. During the project, a complete programming environment for the J-machine was designed and implemented around message-driven programming concepts. This system included a compiler, concurrent file system, and numerous programming tools. These tools were used to evaluate the architecture and experiment with irregular applications.

These complementary studies have resulted in *basic programming technology* that supports the development of portable, large-scale, irregular applications. The technology, collectively termed the *concurrent graph*, consists of a collection of concurrent algorithms, libraries and compiler techniques. It is based on the abstract view of an application as composed of a collection of partitions where

each partition is a distinct entity that is able to execute concurrently, interact with other partitions, move between computers to achieve load-balance, dynamically adjust its granularity, render itself, and allow interactive understanding and modification of its data structures.

The long-term vision is one of a large-scale calculation that can be interactively *monitored* and *steered* through appropriate concurrent algorithms.

Each component of this approach to computational research has been investigated through a distinct component of the research effort and has been described in an attached color plate provided with this report. In summary these plates are:

- Scalable Software Systems:
 - Architecture and Software System Experiments.
 - Message-Driven File System Experiments.
 - Concurrent Graph Library.
 - Avalon A12: Technology Transfer.
- Irregular Programming Problems:
 - Titan IV Launch Vehicle Simulations.
 - Delta II Launch Vehicle Simulations.
 - Ion Thruster Simulations.
 - Gas Flow Simulations.
 - Concurrent Scientific Visualization.

The concurrent graph requires that an application problem be described as a graph composed on nodes and edges. The nodes correspond to *partitions* of the problem, and the edges correspond to data dependencies. Multiple nodes may be mapped to a single computer, or collection of computers sharing memory, in order to overlap communication and computation. Nodes are implemented as processes, light-weight threads, or simply compiled code. This approach separates the logical structure of the application from the underlying machine, which allows the graph to alter the relationship *dynamically* during program execution. This characteristic allows a broad range of dynamic load-balancing algorithms to be utilized, and allows for adjustment of granularity through purely local modifications to the structure of the graph. The first generation of a programming library based on this concept was designed and implemented under a complementary NSF-PYI award, and subsequently applied to an NSF Grand-Challenge problem in Material Science.

During the Scalable Concurrent Programming Project, a novel approach to load-balancing has been invented, implemented, analyzed, and integrated into the concurrent graph framework. This approach can be applied to a variety of parallel algorithms that are based on the parabolic heat equation: Workload at a given computer is treated as heat to be diffused to other computers. These methods have a variety of attractive qualities. They are simple and scalable to implement, involving only nearest neighbor communication. They are guaranteed to converge for arbitrary asynchronously introduced load imbalances, and their rate of convergence has been determined analytically. These results follow directly by leveraging decades of standard mathematical analysis. The algorithms allow a trade-off to be made between the quality of load-balance and the time taken to achieve the balance. Finally, when used for grid-based calculations, the methods maintain locality present in the original grid. This latter quality facilitates their use as both a solution to the dynamic load-balancing problem and the static mapping problem. A variety of implementation issues have been addressed in order to deliver practical methods based on this concept to applications. These issues include determining the profitability of balancing, deciding when to balance, determining what to move and where, and successfully initiating and terminating the balancing process. This work received the 1995 Outstanding Paper Award at the International Conference on Parallel Programming.

The graph technology has enabled the development of a variety of large-scale, three-dimensional simulation capabilities that have had substantial industrial impact. The capabilities have been applied to a broad variety of problems that are among the most complex and aggressive simulations attempted using parallel machines. The capabilities include:

- **Hawk.** This capability was developed in collaboration with the Technology Computer Aided Design (TCAD) group at Intel Corporation, and the Gas Dynamics group at Phillips Laboratory, Edwards AFB. It is based on the Direct Simulation Monte Carlo method applied to complex three-dimensional grid structures. This method has enabled the first full-scale simulations of neutral flow in realistic plasma reactors. In the past, reactors were developed and evaluated solely on the basis of expensive experimental characterization. This capability provides the opportunity for a substantial reduction in the time and cost of developing new plasma-processing technologies; it has already been used by Intel process engineers in the evaluation of proprietary reactor designs.
- **Concurrent-ALSINS.** In a collaborative effort with The Aerospace Corporation, a concurrent, implicit Navier-Stokes capability has been developed. This capability is the primary fluid dynamics tool used by The Aerospace Corporation for a broad range of practical simulations of multi-body launch vehicle configurations. It has been successfully applied to studies of the Titan IV and Delta II launch vehicles. These studies have provided insights into the flight characteristics of the vehicles and the interaction between the vehicle aerodynamics and the plume, at different altitudes and angle-of-attack.
- **PlumePIC.** This capability, developed in collaboration with the MIT Space Power and Propulsion Laboratory, is based on the Particle-In-Cell technique. It was de-

signed to enable the analysis of spacecraft contamination resulting from the use of ion thrusters. These electric-propulsion devices are to be deployed by Hughes, as part of the Galaxy program, and JPL, as part of the New Millennium program, in the 1996-7 time-frame. The capability has been applied to the first full-scale simulation of a realistic satellite employing the Hughes ion thruster.

As a result of these activities, the Scalable Concurrent Programming Laboratory's work was a finalist for the 1996 Smithsonian Award for Science. Collectively, these applications are representative of a broad range of computational techniques with vastly different physical and chemical models. All of these capabilities, however, employ the same concurrent implementation. This allows them to share many data structures, primitive operations, load-balancing algorithms, and granularity-control mechanisms. Moreover, they operate on a wide range of networked workstations, distributed-memory multicomputers such as the Intel Paragon and Cray T3D, and shared-memory multiprocessors such as the SGI Power-Challenge. Networked workstations and shared-memory machines are typically used for validation and simple two-dimensional axisymmetric simulations, while large parallel machines are used for full-scale, three-dimensional simulations.

The concurrent graph concepts employed in these applications were used to drive the evaluation of experimental architectures such as the MIT J-machine and Avalon A12. These machines provide hardware implementations of message-driven programming concepts. All of the the primary software systems for the J-machine hardware experiment, including the compiler, linker, loader, file system, floating point libraries, and microkernel, were developed in this project. The J-machine communication technology was eventually incorporated in the design on the Cray T3D architecture through the efforts of Bill Dally at MIT. The message passing system and associated libraries for the Avalon A12 were designed and implemented by the Scalable Concurrent Programming Laboratory in collaboration with Avalon, Inc. This work, although funded under the ARPA Multi-level Compiler Project, builds directly on experience gained in the Scalable Concurrent Programming Project.

The J-machine experiment, while interesting from an academic viewpoint, has been surrounded by miss-conceptions. Our interest in exploring the use of small memories was motivated by the desire to force alternative implementation techniques, including distributed code-management, microkernel design, and message-driven distributed file systems. Many useful lessons were learned in the experiment, including what constitutes minimal microkernel support, the complexity required to implement distributed code management, how messages can be used to implement file-system technologies, and what design problems in the J-machine memory and message systems should be improved in the M-machine design. This experience proved valuable in working with the Avalon A12, which employs a small microkernel, simple message passing system, DMA hardware support for message transmission without copying, and leverages a large, low-cost, distributed memory.

Approximately 50% of the industrial applications efforts undertaken in the project were successful in that they both pushed the envelope of understanding and impacted an industrial partner. There were many reasons for the failure of the other 50%. These included the normal turnover of personnel, unrealistic short term expectations of parallel computing, and unsuccessful technical approaches. However, we consider a 50% failure

rate on complex multi-disciplinary activities was an acceptable cost in time and effort, given the progress made.

Overall, the project has led to substantive explorations of concurrent programming and systems technology. It has also laid the groundwork for a variety of new research efforts. The Hawk and Concurrent-ALSINS capabilities are evolving to incorporate continued research and continue to be supported by industrial partners. The Hawk capability is in use by Intel Corporation, Tegal Corporation, and IDA. Concurrent-ALSINS has subsequently been used in a flight failure investigation related to the Delta II vehicle by the Aerospace Corporation. The ARPA sponsored Multi-Level Compiler Project at the laboratory is developing software for a new experimental architecture, the MIT M-machine, and continues to develop software for the Avalon A12 machine. A second-generation of the concurrent graph technology for heterogeneous machines is under development and aspects of the technology concerning the relationship between granularity control and load balancing are yet to be investigated. These projects could not have been undertaken without the support provided as part of the Scalable Concurrent Programming Project or the experience gained during the associated experiments.

For further information, see <http://www.scp.caltech.edu>.

Publications

Software Systems Papers

Foster and Taylor, *A Compiler Approach to Scalable Concurrent-Program Design*, ACM Transactions on Programming Languages and Systems, Vol. 16, No. 3, May 1994, pp 577-604.

Heirich and Taylor *Load Balancing by Diffusion*, Proceedings of 24th International Conference on Parallel Programming, Vol. 3, CRC Press pp 192-202, 1995. 1995 Outstanding Paper Award.

Maskit and Taylor, *A Message-Driven Programming System for Fine-Grain Multicomputers*, Software Practice and Experience, Vol 24(10), pp 953-980, Oct 1994.

Taylor, Watts, Rieffel, and Palmer, *The Concurrent Graph: Basic Technology for Irregular Problems*, IEEE Parallel and Distributed Technology (1995), Vol. 4, No. 2, pp 15-25, Summer 1996.

Irregular Programming Papers

Fyfe, Taylor, and Williams, *A Concurrent, Tetrahedral Fluid Solver, based on Wave Propagation*, Parallel CFD 94, Elsevier Science Publishers B.V., Kyoto Japan, 1994.

Harrar, Keller, Lin, and Taylor, *Parallel Computation of Taylor-Vortex Flows*, Parallel CFD 92, Stuttgart, Germany, Elsevier Science Publishers B.V., pp 193-206, 1991.

Palmer and Taylor, *Rotation Invariant Partitioning for Computational Scientific Visualization*, Parallel CFD 94, Elsevier Science Publishers B.V., Kyoto Japan, 1994.

Palmer, Taylor and Totty, *Interactive Volume Rendering on Clusters of Shared-Memory Multiprocessors*, Proceedings of Parallel CFD 95, Elsevier Science Publishers B.V., Kyoto Japan, 1995.

Samanta Roy, Hastings, and Taylor, *Three-Dimensional Plasma Particle-in-Cell Calculations of Ion Thruster Backflow Contamination*, Journal of Computational Physics (1995),

Samanta-Roy, Hastings, and Taylor, *Three-Dimensional Plasma Particle-in-Cell Calculations of Ion Thruster Backflow Contamination*, Proceedings of 34th AIAA Aerospace Sciences Meeting, 1996, Reno, NV.

Shankar, Rieffel, Taylor, Weaver, and Wulf, *Low-Pressure Neutral Transport Modelling for Plasma Reactors*, Proceedings of Workshop on Industrial Applications of Plasma Chemistry, Aug. 21-25, 1995, Eds. A. Wendt and J.V. Heberlein, Volume A, pp 31-40.

Taylor and Wang, *Launch Vehicle Simulations using a Concurrent, Implicit Navier-Stokes Solver*, Journal of Spacecraft and Rockets (1995). (To appear.)

Taylor and Wang, *Launch Vehicle Simulations using a Concurrent, Implicit Navier-Stokes Solver*, 33rd Aerospace Sciences Meeting, AIAA, Reno, NV, 1994.

Wang and Taylor, *Studies of the Multi-body Launch Vehicle Flowfield Characteristics using CFD: Upstream plume effects*. Parallel-CFD 96, Capri, Italy May 20-23.

Wang and Taylor, *A Concurrent, Nodal Mismatched, Implicit Navier-Stokes Solver*, Parallel CFD 94, Elsevier Science Publishers B.V., Kyoto Japan, 1994.

Wang and Taylor, *A Concurrent Navier-Stokes Solver for Implicit Multibody Calculations*, Parallel CFD 93, Elsevier Science Publishers B.V., Paris, France, 1993.

Thesis Research

Chow, *Compiler Techniques for Loosely-Coupled Multi-Cluster Architectures*, Masters Thesis, Department of Computer Science, California Institute of Technology, 1996.

Rieffel, *Concurrent Simulations of Plasma Reactors for VLSI Manufacturing*, Masters Thesis, Department of Computer Science, California Institute of Technology, CS-TR-95-012, 1995.

Watts, *A Practical Approach to Dynamic Load Balancing*, Masters Thesis, Department of Computer Science, California Institute of Technology, CS-TR-95-013, 1995.

Heirich, *Load Balancing with Differential Equations*, Masters Thesis, Department of Computer Science, California Institute of Technology, CS-TR-94-13, 1994.

Maskit, *A Message-Driven Programming System for Fine-Grain Multicomputers*, Masters Thesis, Department of Computer Science, California Institute of Technology, CS-TR-94-03, 1994.

Palmer, *Immersing the Scientist in Data*, Masters Thesis, Department of Computer Science, California Institute of Technology, CS-TR-94-06, 1994.

2 Ph.D. Students, Daniel Maskit and Michael Palmer are currently writing their theses. Daniel Maskit was partially funded by NSF and Michael Palmer was partially funded by a DOD AASERT award. Two other Ph.D. Students, Marc Rieffel and Jerrell Watts are half way through their Ph.D. studies. Alan Heirich has changed to an alternative project. Brian Chow has decided to persue an industrial career.

Technical Reports

Maskit and Taylor, *Experiences Programming the J-Machine*, Department of Computer Science, California Institute of Technology, Technical Report CS-TR-93-11, 1993.

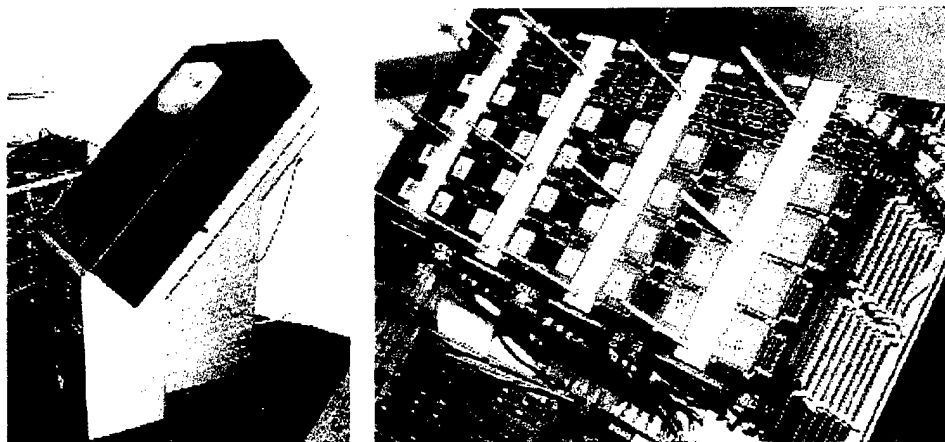
Maskit, Zadik, and Taylor, *System Tools for the J-Machine*, Department of Computer Science, California Institute of Technology, Technical Report, CS-TR-93-12, 1993.

Zadik and Taylor, *A File System for the J-Machine*, Department of Computer Science, California Institute of Technology, Technical Report CS-TR-93-27, July 1993.

Chandy and Taylor, *Primer for Program Composition Notation*, Department of Computer Science, California Institute of Technology, Technical Report: Caltech-CS-TR-90-10.

**MESSAGE DRIVEN SOFTWARE SYSTEM
EXPERIMENTS**

Architecture and Software System Experiments.



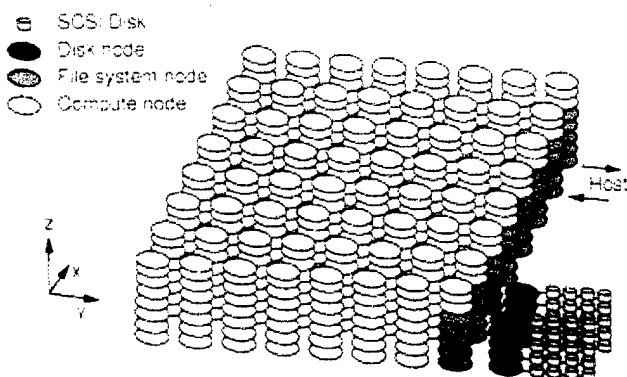
The continued development of state-of-the-art parallel architectures presents new challenges to the compiler writer and programming system designer. This project sought to drive these experiments through genuine applications development. The project explored the problem of designing programming tools that efficiently utilize advanced hardware mechanisms for communication and synchronization. A complete programming system, Message-Driven C (MDC), was developed targeted at the MIT-J machine -- an ARPA-sponsored machine architectural experiment. The programming system provided a compiler, linker, loader, assembler, distributed-code micro-kernel, concurrent file system, networking, and performance evaluation tool. In addition, applications were developed using a portable programming tool, the *concurrent graph*. This tool allowed applications to operate on a broad range of parallel architectures and networked workstations -- including the J-machine. Thus applications are able to execute both on today's machines, and be utilized in state-of-the-art architectural experiments.

Concepts from the J-machine communication structure have been directly integrated into the Cray T3D and T3E architectures. Direct feedback has been provided to the MIT group through regular visits and several technical reports and publications. This feedback has directly influenced the design of the next generation of hardware -- the M-Machine. In addition, the results are being used as the foundation for the next generation of programming systems at Caltech. Experience from this project led directly to programming systems for the Avalon A12 architecture.

Maskit and Taylor, "A Message-Driven Programming System for Fine-Grain Multicomputers," *Software - Practice and Experience*, 24, 953-980 (1994).

Foster and Taylor, "A Compiler Approach to Scalable Concurrent-Program Design", *ACM Transactions on Programming Languages and Systems*, Vol. 16, No. 3, May 1994, pages 577-604.

Message-Driven File System Experiments



As parallel machines continue to scale, the data sets that are generated continue to grow in proportion. Additionally, large production runs make take several week or months, so it is important to save intermediate results periodically to guard against unexpected shutdowns and mechanical failures. Thus, frequent storage of large quantities of information on backing store is required. Conventional file storage systems, while adequate for small numbers of computers, do not take advantage of the bandwidth opportunities provided by three-dimensional, mesh-connected architectures such as the CRAY T3D and MIT J and M-machines.

A scalable, concurrent file system has been developed in which disks are attached via a two-dimensional plane of edge connections at the periphery of the machine. There are two levels of concurrency in the file system. Compute nodes may send messages to arbitrary locations within the file system plane, and data blocks are striped across physical disks. The implementation is "stateless" and thus consistent with the industry standard Network File System (NFS). Thus the file system appears over the internet as a conventional Unix file system.

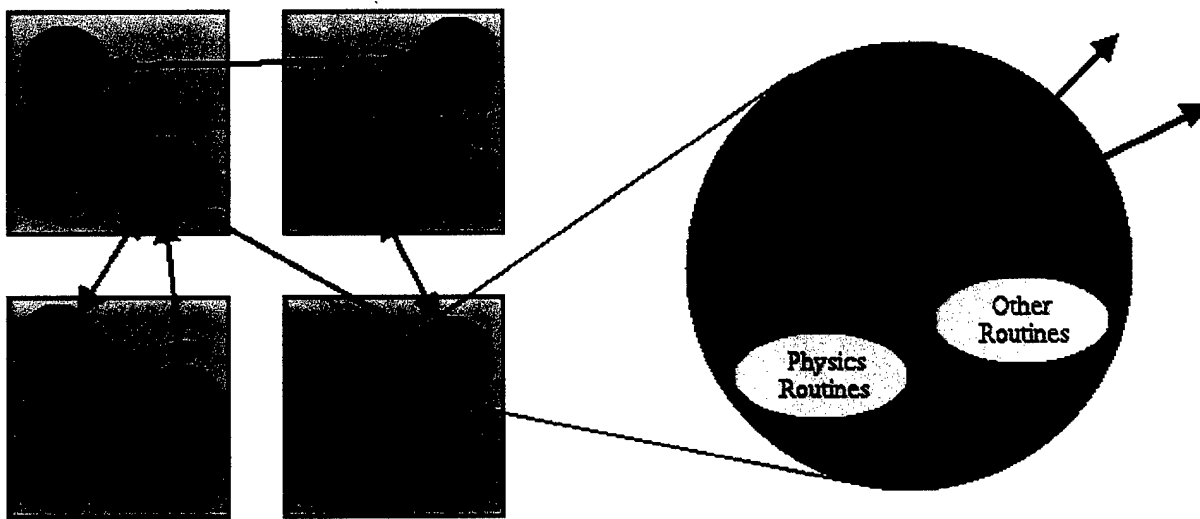
The file system is an enabling technology that is necessary for applications experiments. The system has provided feedback to hardware engineers on how to organize disk hardware so as to improve throughput. This feedback has been incorporated into the M-machine design.

Zadik, "The Message Driven File System: A Network Accessible File System for Fine-Grain Message Passing Multicomputers", Masters Thesis, Department of Computer Science, California Institute of Technology. 1995

Concurrent Graph Library

The Concurrent Graph Library provides basic programming technology to support irregular applications on scalable concurrent hardware. Developed under NSF-PYI support, the library has been successfully applied to a wide variety of large-scale industrial application problems as part of the Scalable Concurrent Programming Project.

The technology is based on the concept of a *concurrent graph* that provides an adaptive collection of processes that may relocate between computers dynamically. The graph is portable to a wide range of high-performance multicomputers (e.g. Cray T3D/E and Intel Paragon), shared-memory multiprocessors (e.g. SGI PowerChallenge), and networked workstations (e.g. IBM, SGI, or Sun workstations). For each architecture it is optimized to take advantage of the best available underlying communication and synchronization mechanisms. The graph provides a framework for automatic load balancing and granularity control, and interactive, on-the-fly visualization. Load balancing is based on the notion of heat diffusion and has rigorously provable convergence and correctness properties.



Taylor, Watts, Rieffel and Palmer, "The Concurrent Graph: Basic Technology for Irregular Problems", *IEEE Parallel and Distributed Technology*, 4(2):15-25, 1996.

Heirich and Taylor, "Load Balancing by Diffusion", Proceedings of 24th International Conference on Parallel Programming, Vol. 3, CRC Press pp 192-202, 1995. 1996 Outstanding Paper Award.

Avalon A12: Technology Transfer



The Avalon A12 architecture (shown above) is a scalable multicomputer for high-performance computing and embedded systems applications. The first system was recently delivered to Caltech and is part of a joint development effort between Avalon and the Scalable Concurrent Programming Laboratory. The A12 has the following system design:

- Freestanding or rugged rack mount configuration.
- 400MHz DEC-Alpha Processor technology using *standard PC memory*.
- Hot-swappable Processors.
- PCI interface at each processor.
- Reconfigurable worm-hole routing network - 400Mb/sec.
- Distributed power supply and power control.
- Packs 9.6 Gigaflops into a single rack.
- Real-time micro-kernel operating system.
- MPI and NX Parallel Programming.
- Wide variety of cross-compilers for embedded applications available.
- State-of-the-art debugging and software tools available.

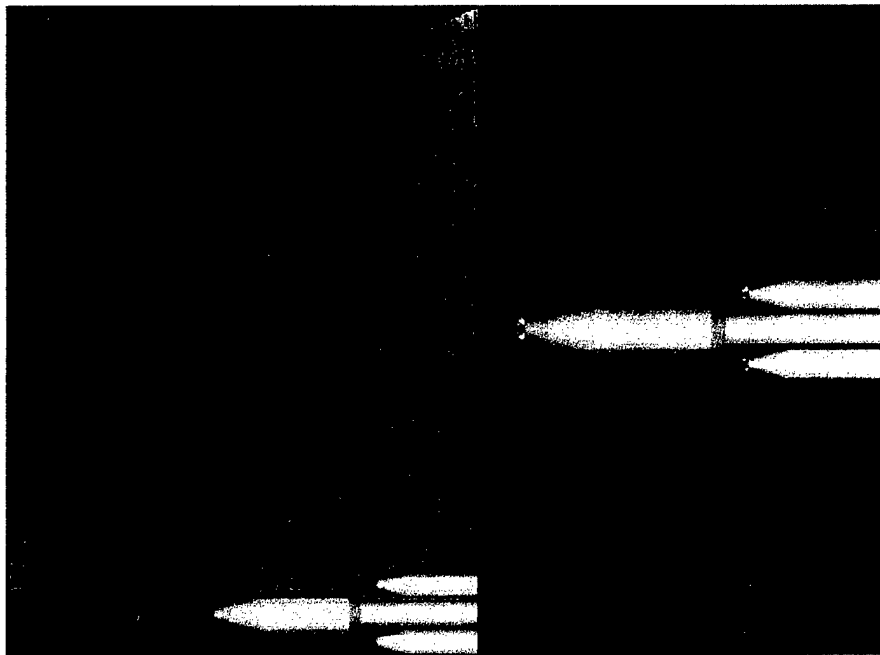
The A12 message passing system and associated libraries were designed and implemented by members of the Scalable Concurrent Programming Laboratory in collaboration with Avalon Inc.

**IRREGULAR PROGRAMMING
EXPERIMENTS**

Titan IV Launch Vehicle Simulations

The Scalable Concurrent Programming Laboratory, in collaboration with The Aerospace Corporation, has developed a capability for modeling complex launch vehicle configurations. This capability computes both steady state and unsteady solutions to the three-dimensional, compressible Navier-Stokes equations. It employs a variety of features that increase practical utility such as multi-body support, turbulence modeling, mismatched grid structures, and implicit flow solution. The capability is portable to a broad range of parallel machines including networks of workstations, multicomputers, and shared-memory multiprocessors.

Flow-field simulations of the Titan IV vehicle have been produced at a variety of free-stream velocities and angles-of-attack. The pictures below show a cross-section through the computational grid, consisting of approximately 750,000 cells (left), and an example pressure field at Mach 1.6 (right). This simulation executed on the 256-node Intel Delta machine and required 37,000 node hours.

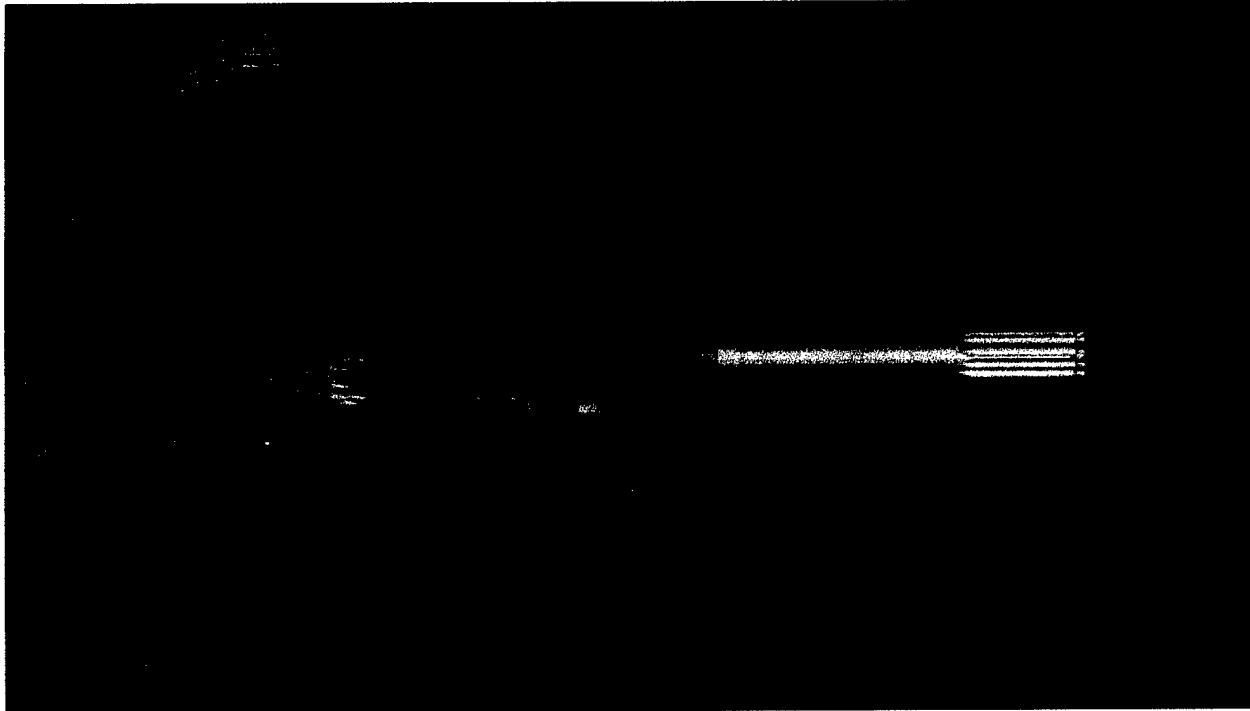


Taylor and Wang, "Launch Vehicle Simulations using a Concurrent, Implicit Navier-Stokes Solver", Journal of Spacecraft and Rockets, 1995. (In Press)

Wang and Taylor, "A Concurrent, Nodal Mismatched, Implicit Navier-Stokes Solver", Parallel CFD 94, Elsevier Science Publishers B.V., Kyoto Japan, 1994.

Wang and Taylor, "A Concurrent Navier-Stokes Solver for Implicit Multibody Calculations", Parallel CFD 93, Elsevier Science Publishers B.V., Paris, France, 1993.

Delta II Launch Vehicle Simulations



An early version of the Delta II vehicle was flight tested and found to be close to its control margins. Post flight reconstruction of the aerodynamic forces were inconsistent with measured data from wind tunnel tests. As a result, it was necessary to make assumptions in the pressure distribution on the vehicle so as to find the center of pressure. Unfortunately, the calculated center of pressure differs substantially from the wind tunnel results.

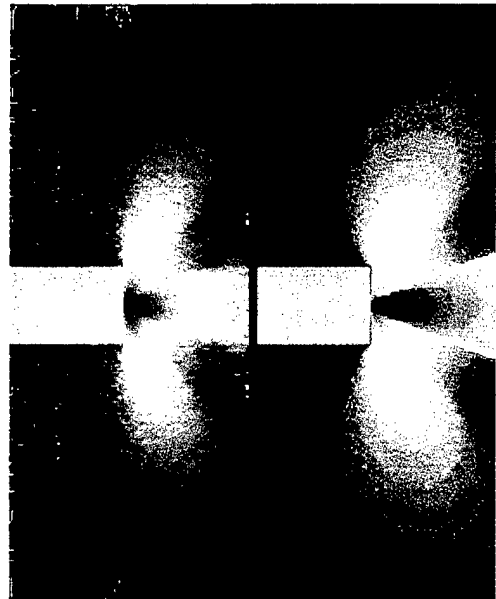
A scalable, concurrent flow solver for computing both steady state and unsteady solutions to the three-dimensional compressible Navier-Stokes equations was developed. It employs a variety of features that increase practical utility: multibody support, full viscous effects, turbulence modeling, and implicit flow capabilities. The capability is Portable to a broad range of parallel machines using *concurrent graph* technology.

Direct impact on Air Force Delta II mission. Poorly understood aerodynamic effects represent a concern for future flights of the Delta II and also for the design of the next generation of launch vehicles. Results with the plume turned off and also with the plume turned on, at both zero and 5 degree angle-of-attack, have been calculated.

Ion Thruster Simulations

The next generation of military and commercial satellites will utilize electric propulsion devices called ion thrusters. These thrusters emit effluents that may drift back around the vehicle when on orbit, contaminating spacecraft surfaces, such as solar panels. This contamination reduces the useful life of the vehicle.

A concurrent simulation capability, PlumePIC, based on the plasma particle-in-cell (PIC) technique has been developed. This capability allows the structure of the deposition process to be predicted over realistic three-dimensional geometries. The capability models the production of slow charge-exchange (CEX) ions produced in the thruster and their transport around the exterior of the satellite. The self-consistent electrostatic potential is determined by solving the Poisson equation over the entire computational domain. The pictures at the right show an example simulation that uses a generic spacecraft geometry with an attached ion thruster. The pictures show, from upper left to lower right, the spacecraft geometry and vertical cross-sections through the predicted ion-density, charge-exchange ion density, and electrostatic potential. The simulations were produced on a 256-node Cray T3D at JPL using a grid of approximately 10 million cells and 30 million test particles. The solution converged after 120,000 node hours.

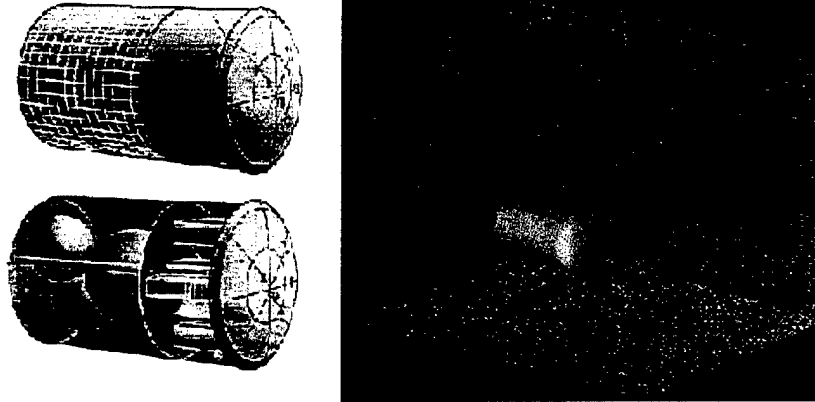


The capability is portable to a wide range of networked workstations, multicomputers, and shared-memory multiprocessors. It utilizes a state-of-the-art load balancing technique based on heat diffusion. Due to the unique two-phase nature of the PIC simulation (particle movement and field solution), current load balancing methods proved ineffective. A new strategy, based on the concept of load as *vector*, has been applied to overcome the problem.

Samanta Roy, Hastings, and Taylor, "Three-Dimensional Plasma Particle-in-Cell Calculations of Ion Thruster Backflow Contamination", *Journal of Computational Physics*, 1995. (In Print)

Samanta-Roy, Hastings, and Taylor, "Three-Dimensional Plasma Particle-in-Cell Calculations of Ion Thruster Backflow Contamination", Proceedings of 34th AIAA Aerospace Sciences Meeting, 1996, Reno, NV.

Gas Flow Simulations



The Skipper mission was a space experiment involving a small satellite (above left) traveling at about 8 km/sec making a controlled re-entry for the purpose of measuring ultraviolet radiation from the shock-heated gas at the nose region. To model the reentry, a concurrent, three-dimensional simulation capability based on the Direct Simulation Monte Carlo (DSMC) method was developed in collaboration with Phillips Laboratory, EAFB.

This capability employed regular rectangular grids and was carefully constructed using modern software engineering practices. As a result, the chemical and surface models were interchangeable and could be used for a variety of applications. It was subsequently applied to the modeling of neutral flow in plasma reactors by Intel Corporation. The capability included a variety of features to increase practicality: complex grids generated with standard CFD tools, multiple species, realistic chemistry models, inelastic collisions, velocity-dependent collisional cross sections. It was portable to a wide range of parallel machines using the *concurrent graph* technology.

Shankar, Rieffel, Taylor, Weaver, and Wulf, "Low-Pressure Neutral Transport Modelling for Plasma Reactors", Proceedings of Workshop on Industrial Applications of Plasma Chemistry, Aug 21-25, 1995, Eds A. Wendt and J.V. Heberlein, Volume A, pp 31-40.

Concurrent Scientific Visualization



As parallel machines continue to scale, the data sets produced by scientific simulations grow increasingly large. Scientists need to be able to obtain a quick visual check, as well as perform detailed analysis, of very large regular and irregular datasets.

The group has developed dynamically load balanced concurrent algorithms for visualization of both regular and irregular data sets on parallel machines. The intent is to connect to a running parallel application, step into its data structures, and walk around inside the changing data in order to understand its evolution. This project is partially supported by a DOD AASERT Award.

Our rendering algorithms, embodied in an implementation called *RendAsunder*, were demonstrated on the "Power Wall" in the Silicon Graphics booth at Supercomputing '95. Hardware consisted of eight Power Challenge machines and two Power Onyxes, with a resolution of 3200x2400 pixels on a screen measuring eight feet diagonally. On a 64 processor Power Challenge Array, the algorithms are capable of rendering a 350MB dataset at D1 video resolution (640x480 pixels) at 8-10 frames per second. The group has an ongoing research relationship with Silicon Graphics in this area.



In the large figure above and to the right, the rectangles overlaid on the screen each represent one of 32 processors; the 4 colors represent 4 Power Challenge nodes, which each contain 8 of the processors.

Palmer, M.E., Totty, B., and Taylor, S. "Ray Casting on Shared-Memory Architectures: Efficient Exploitation of the Memory Hierarchy," Submitted to *IEEE Parallel and Distributed Technology*.

Palmer, M.E., Taylor, S., and Totty, B. "Interactive Volume Rendering on Clusters of Shared-Memory Multicomputers," *Parallel Computational Fluid Dynamics '95*, Pasadena, CA. Elsevier Science Publishers B.V., 1995.