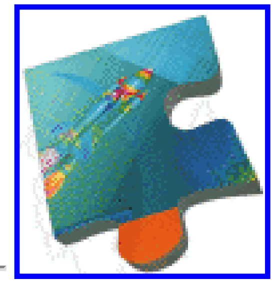
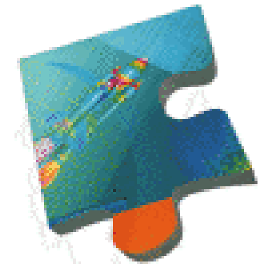
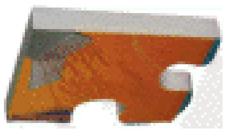
November-December 1996
Vol 2. No. 21



Updated: Tuesday, 03-Aug-1999 16:05:21 PDT
WebWork for: *Wade Roush*
NASA Responsible Official: *Dr. William J. Feiereisen*

NAS at Supercomputing '96

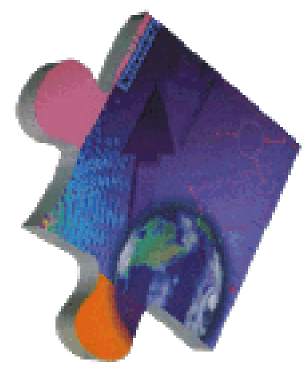




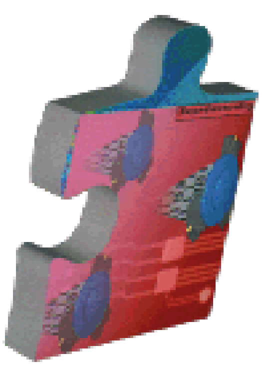
Here is an overview of NAS's involvement in Supercomputing '96 -- reflecting our latest accomplishments in high-performance computing. The NAS commitment to provide a unique supercomputing environment is underscored in the "puzzle" theme of its SC '96 booth, which highlights four capabilities that make up the "big picture" of the NAS facility: powerful vector and parallel supercomputers, visualization tools, mass storage capabilities, and high-speed networks. Along with specially developed systems software, these state-of-the-art elements create a high-performance system in which each piece is critical.

Four Puzzle Pieces That Create NAS's `Big Picture'

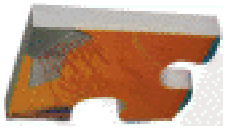
- Stereo Visualization Theater Features New 3D Flicks
- NASA Metacenter
- Virtual Windtunnel Adds Important Capabilities
- NAS Debugger Offers Enhancements
- Behind the Scenes at NAS a Multimedia Experience



NAS at SC '96



- Better Performance for Parallel Mesh Adaption of Unstructured Grids
- Panel Compares Archival Storage Systems
- High-performance Chips and Architectures Compared Using NAS Parallel Benchmarks
- Poster Session on Latest NAS Parallel Benchmarks
- Useful and `Opinionated' Discussion on MPI
- MPI-2 BOF Session
- Parallel I/O Tutorial
- NAS Demo: PMPIO



SC '96 Highlights Four Puzzle Pieces That Create NAS's `Big Picture'

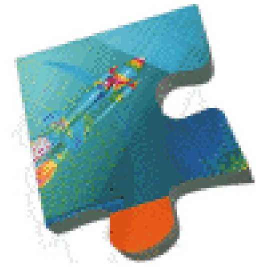
Stereo Visualization Theater Features New 3D Flicks

The NAS Facility's Stereo Visualization Theater offers a new presentation of datasets -- the wing rock, V22 tiltrotor, space shuttle, and F18 -- displaying aerospace designs in a 3D environment. Large problems are being solved in these simulations, which require the efficient integration of capabilities in storage, networking, supercomputing, and visualization. Stereo viewing offers insights into data analysis that traditional 2D methods can't provide.

The theater's graphics, digital audio, and stereo projection deliver a degree of resolution not possible with conventional videotape or multimedia. The theater, which can accommodate an audience of up to 20, runs on a Silicon Graphics Inc. Onyx RE2 system.

[TOP]

NASA Metacenter



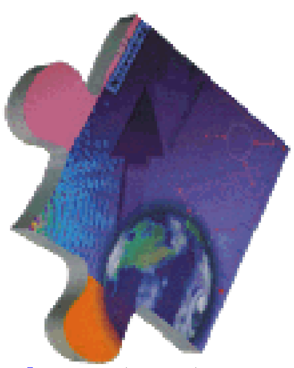
The NASA metacenter was formed between the NAS Facility at Ames Research Center and NASA Langley Research Center (LaRC) as an exploratory project. The goal is to make more effective use of supercomputing resources by expanding the accessibility and flexibility of IBM SP2 systems at both locations.

The metacenter consists of PBS (the Portable Batch System), a job management system, an external job scheduler, and system support. It allows researchers at NAS or LaRC to submit jobs to either system, with PBS running the job on the first available system that can provide the requested number of nodes.

Some of the features demonstrated include:

- routing jobs automatically from one system to another
- executing jobs on a user-specified system
- moving necessary files before and after job execution
- mapping user names and returning new output with electronic mail to the correct user
- running a single application (such as CFD and structural elements) across both SP2s simultaneously

[TOP]



Virtual Windtunnel Adds Important Capabilities

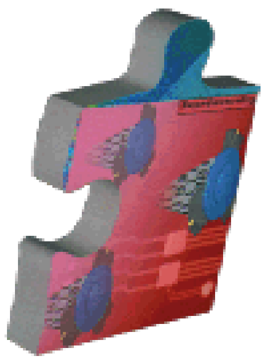
If you haven't seen the NAS Virtual Windtunnel (VWT), stop by the NAS booth to experience first-hand this virtual reality system that combines software and interface technology to visualize CFD simulations. A new feature, the Field Encapsulated Library (FEL), provides a high-performance, grid-independent approach to visualizing CFD algorithms. Applications programmers will appreciate these tools to access, organize, and manipulate data on numerical grids for interactive visualization.

Also new is a time-critical feature designed by Bryson to maintain interactivity in the VWT. It automatically assigns a time budget to each visualization, which then selects an algorithm to do the best job in the time allotted.

The VWT will be shown in combination with a new version of the Responsive Workbench (manufactured by Fake Space Labs Inc., Menlo Park, CA), which displays 3D images. The new workbench was built to withstand frequent moves to demonstrations around the country.

For more information, send email to bryson@nas.nasa.

[TOP]



NAS Debugger Offers Enhancements

Having one debugger available on a wide variety of systems is convenient for users. In addition, the ability to have an abstract view of a large collection of processes will make debugging distributed programs significantly easier, according to Robert Hood, developer of p2d2, a general-purpose portable debugger for distributed systems.

Hood will demonstrate p2d2's effectiveness in running a multiprocess MPI (Message Passing Interface) computation on a heterogeneous collection of hardware configurations at the NAS SC '96 booth. Other p2d2 features include a new online Help facility designed with integrated hypertext for Web Help. In addition, Hood will demonstrate automated display features that give users an overview of debugging processes and the values calculated.

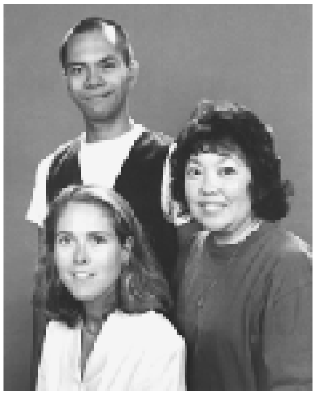
[TOP]

Behind the Scenes at NAS -- a Multimedia Experience

What really goes on at a supercomputing center? Find out more about the large-scale computing environment at the NAS Facility through an interactive multimedia demonstration designed for individual navigation.

"Aeronautical Supercomputing at NASA" illustrates how teraflop computing is helping to apply CFD technology to the design of tomorrow's aircraft. This online tour presents an overview of the major NASA programs comprising the NAS Systems Division: NAS Program, Computational Aerosciences, and the Aeronautics Consolidated Supercomputer Facility.

Learn how supercomputing components such as high-speed vector and parallel processing, networks, mass storage, visualization, and systems software are brought together to create a balanced environment. The hands-on program includes datasets and other computer-based exercises and will be housed in a kiosk at the Ames Visitor's Center after Supercomputing '96. The tour is designed to interest both technical and general audiences.



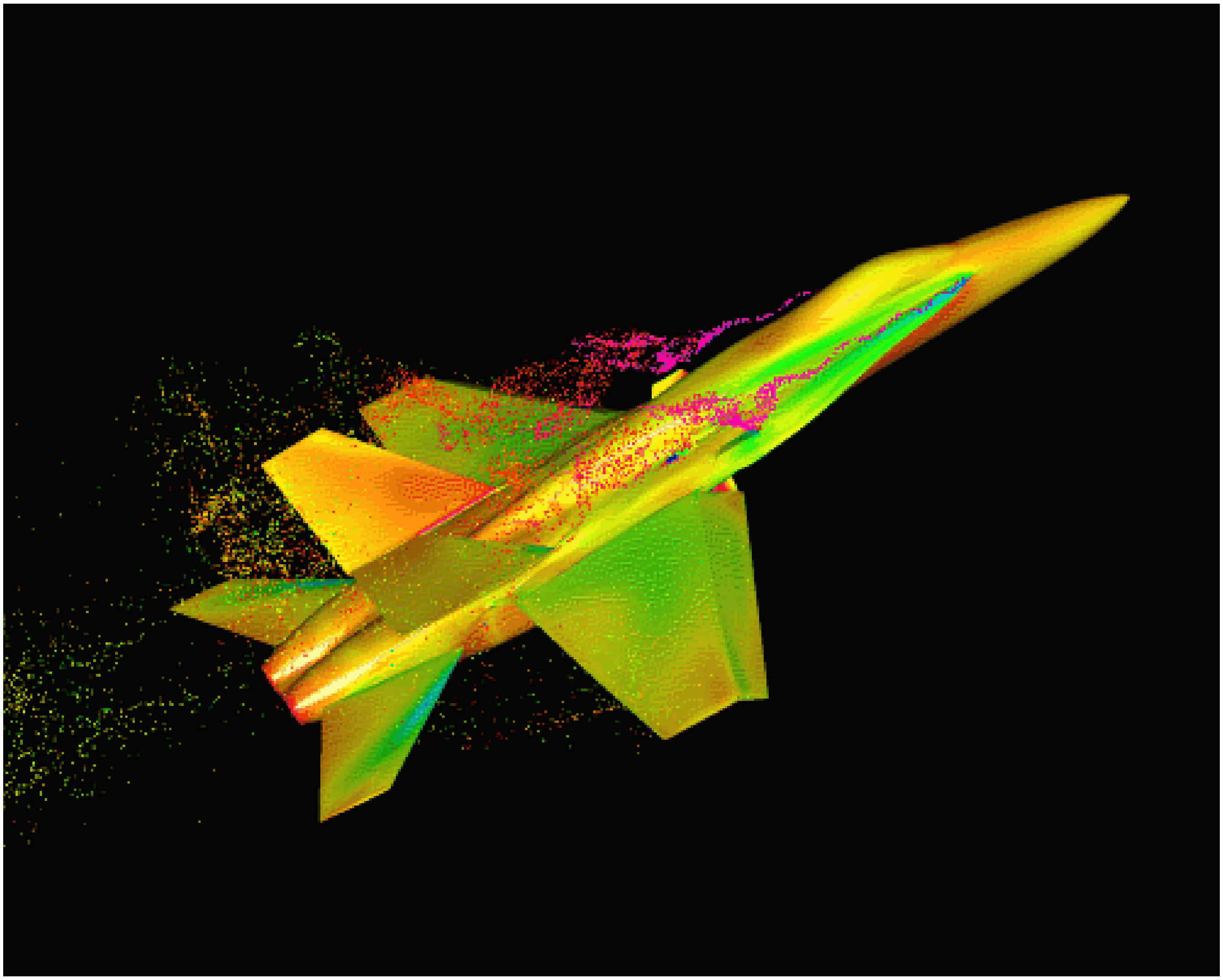
For more information, contact Kevin McCabe at kmccabe@nas.nasa.gov.

The NAS mulitmedia team: Jeol Antipiesto (top), Chris Gong (left), Fay Pattee (right).

[TOP]

Next Article | Contents

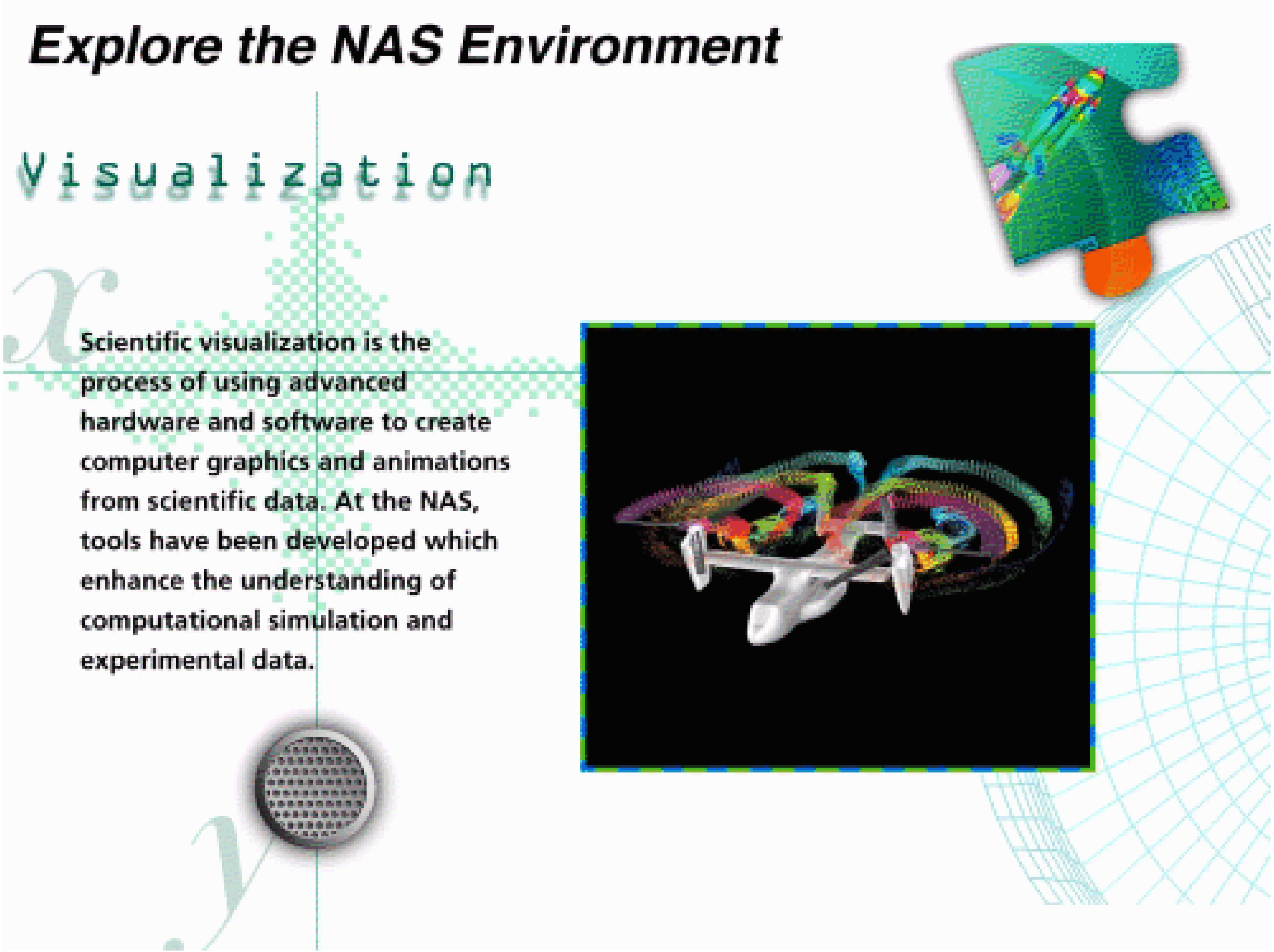
Main Menu | NAS Home



A frame from one animation in the Stereo Visualization Theater. Shown is the F-18 at high angle, with survace colored by pressure and streaklines colored by time.

Data by Ken Gee. Image by Chris Gong and David Kao, using FAST and UFAT.

Back to the article.

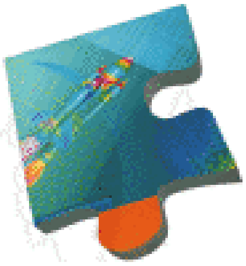


A screen from the interactive multimedia tour "Aeronautical Supercomputing at NASA," which gives visitors an overview of the NAS Systems Division. This particular screen features scientific visualization tools through the use of text and Quicktime video clips, such as the Bell Boeing V22 Tiltrotor, shown here. Clicking on the "speaker" icon at lower left gives a sound bite with more information. The tour is being presented at Supercomputing '96 in Pittsburgh, November 17 -- 21.

Image by Fay Pattee.

Back to the article.

Next Article | Contents | Main Menu | NAS Home



NAS at SC '96

Better Performance for Parallel Mesh Adaption of Unstructured Grids

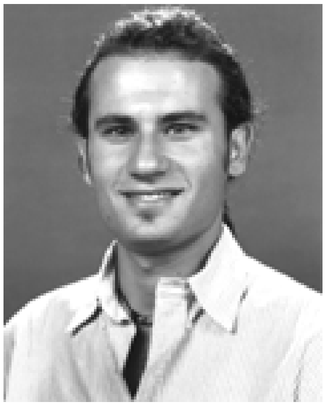
(6A) Algorithms II: Wednesday, 3:30 p.m.

The results of a new heuristic remapping algorithm that minimizes redistribution costs for dynamic mesh adaption on unstructured grids will be presented by NAS scientist Rupak Biswas.

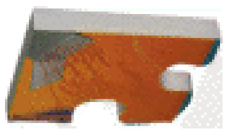
The paper, "Global Load Balancing with Parallel Mesh Adaption on Distributed-memory Systems," by Biswas, Leonid Oliker, and Andrew Sohn, discusses the results of research collaboration at the NAS Facility.

An approximate 55 percent speedup on 64 processors of the IBM SP2 is achieved when one-third of the mesh is randomly adapted. For large-scale scientific computations, the load-balancing strategy gives almost a six-fold reduction in solver execution times over non-balanced loads. The heuristic remapper yields processor assignments that are less than 3 percent off the optimal solutions but require only 1 percent of the computation time.

Send email to rbiswas@nas.nasa.gov for more information.

 Rupak Biswas  Leonid Oliker  Andrew Sohn

[TOP]



Panel Compares Archival Storage Systems

Wednesday, 3:30 p.m.

Four archival storage systems -- DMF, HPSS, NAStore, and UniTree -- will be examined by the following panelists at their respective sites: Terry Jones (Naval Oceanographic Office/Northrop Grumman), Douglas Carlson (Cornell Theory Center), John Lekashman (NAS), and Ellen Salmon (NASA Goddard Space Flight Center).

Configurations, customers, site usage, and drawbacks for each archival system will be presented. Discussion will cover panelists' views of the tradeoffs of each storage solution, plus a look at future storage needs. The effectiveness of common tape formats, archive commands, benchmarks, and desired features, will also be addressed.

[TOP]

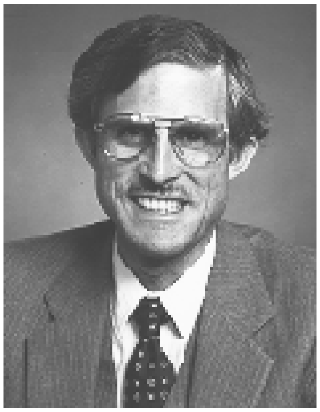
High-performance Chips and Architectures Compared Using NAS Parallel Benchmarks

(M6) Monday, 8:30 a.m.

The architecture of several high-performance microprocessors and a description of the supercomputers based on these processors will be presented in an all-day tutorial by NAS scientists Subhash Saini and David Bailey. Performance of various hardware/programming model combinations will then be compared, based on the latest NAS Parallel Benchmark results. Discussion of general trends in high-performance computing and future directions in hardware and software technology will be included.

The tutorial, "Hot Chips for High Performance Computing," will cover these CMOS-based processors: DEC Alpha 21164 (CRAY T3D); MIPS R10000 (SGI Power Challenge Array); Intel Pentium Processor (DoE ASCI system); PowerPC 604 (IBM SMP); Hewlett-Packard PA 8000 (Convex Exemplar SPP2000); NEC (NEC SX-4); Fujitsu VPP700; Hitachi SR2201.

Send email to saini@nas.nasa.gov for more information about this research.

 Subhash Saini  David Bailey

[TOP]

Poster Session on Latest NAS Parallel Benchmarks

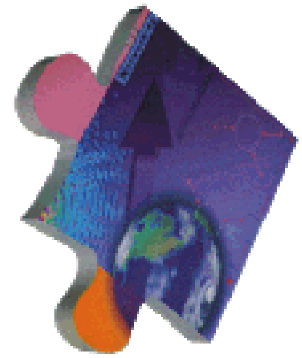
All day Tuesday & Wednesday -- Reception: Wednesday, 6:00 p.m.

Make time at the poster reception to meet with Bill Saphir, Alex Woo, and Maurice Yarrow to discuss the new set of results for version 2.1 of the NAS Parallel Benchmarks (NPB). Included are first-time results for the CRAY T3E. Other architectures tested are the CRAY T3D, IBM SP2, Intel Paragon, and the SGI Power Challenge Array.

One surprise that researchers turned up: the system with the best relative peak performance on the benchmarks doesn't give the best relative performance on the benchmarks -- a change from the NPB 1 results. NPB 2, announced at Supercomputing '95, is based on Fortran 77 and the MPI message-passing standard. This suite of eight benchmarks is run with little or no tuning, which gives users an approximate idea of real-world performance on these parallel computers.

[TOP]

Useful -- and `Opinionated' Discussion on MPI



(S3) Sunday, 8:30 a.m.

Beginners and experienced users alike will benefit from "An Intensive and Practical Introduction to the Message Passing Interface (MPI)," a tutorial by Bill Saphir. As the title implies, this full-day session is jam-packed with information, from core functionality to more advanced MPI features. Saphir promises "opinionated discussion" -- for example, he states that "there are lots of send modes, but all except one of them are usually worthless."

That kind of practical, cut-to-the-chase approach, with an emphasis on high performance, is what this session is all about: looking at what works in reality versus theory, as well as how to avoid common mistakes. You'll also learn the differences between MPI and PVM (Parallel Virtual Machine) and the latest developments in MPI-2, the next version of the MPI standard, expected to be released in 1997.

For more information, send email to wcs@nas.nasa.gov. Look for a Web site listing for the complete tutorial in the next issue of NAS News.

[TOP]

MPI-2 BOF Session

Wednesday, 3:30 p.m.

Parallel users are eagerly awaiting the unveiling of MPI-2, the effort to include extensions to the message-passing interface standard. The draft released at Supercomputing '96 reflects work in progress, with a final document expected next June.

Join other interested users, vendors, and software developers -- including NAS staff members who have been actively participating in MPI-2's formation -- to find out what's going on and exchange ideas at this birds-of-a-feather gathering.

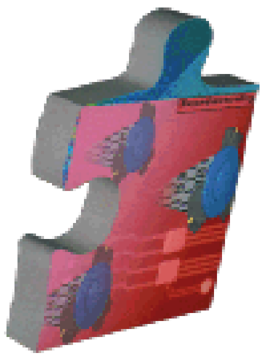
[TOP]

Parallel I/O Tutorial

(M5) Monday, 8:30 a.m.

The full-day tutorial "Parallel I/O on Highly Parallel Systems" by Samuel Fineberg and Bill Nitzberg, takes a comprehensive look at the latest in parallel I/O, from the basics to recent advances. Fineberg and Nitzberg (both in the NAS parallel systems group) will present multivariate findings for commercial products, including those from Convex, Cray Research, IBM, Intel, and Meiko, as well as academic and research projects.

[TOP]



NAS Demo: PMPIO

Stop by the NAS booth for a first-hand look at PMPIO, a successful, portable version of the MPI-IO interface. Fineberg, Nitzberg, and co-developer Parkson Wong will demonstrate features that contribute to high performance, such as collective buffering. Don't expect pretty pictures -- just the best library out there -- so far.

The NAS News online archives contain several articles on PMPIO, the most recent of which is written from a user's perspective.

Next Article | Contents | Main Menu | NAS Home

`Surprise' in New NAS Parallel Benchmark Results

by Bill Saphir, Alex Woo, and Maurice Yarrow

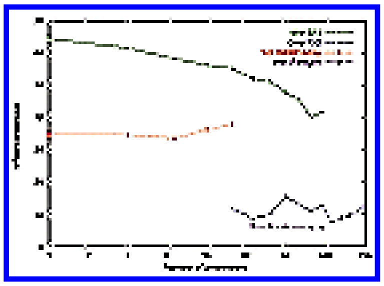
The NAS Parallel Benchmarks (NPB), developed in 1991 to compare the performance of highly parallel computers with that of traditional supercomputers, have become a widely recognized measure of supercomputer performance. The benchmarks are unique in that they are specified algorithmically and are implemented by computer vendors using techniques and optimizations appropriate to specific computers. This allows the comparison of widely different computer architectures. These vendor-optimized NPB implementations will be referred to as NPB 1.

In late 1995, NAS announced NPB 2, a set of NPB implementations based on Fortran 77 and MPI (the Message Passing Interface). NPB 2 implementations are intended to run with little or no tuning -- in contrast to those in NPB 1, which are highly optimized by vendors for specific architectures. They are designed for computers with cache-based hierarchical memories, though some may be appropriate for vector machines, as well.

Unlike that of proprietary NPB 1 implementations, the source code of NPB 2 implementations is freely available. One advantage of this is that anyone can run them on any machine, allowing data collection on a wide variety of systems and configurations. Another is that the techniques used in NPB 2 can be studied for possible use in other codes.

Because they have not been optimized for any particular machine, NPB 2 implementations approximate the performance a typical user can expect for a portable parallel program on a distributed-memory parallel computer. Collectively, NPB 2 results present a well-calibrated comparison of the real-world performance of several parallel computers. These results complement, rather than replace, NPB 1 results. The first NPB 2 results were released in August. Complete results, as well as specifications and codes, are available.

Real Performance Not Proportional to Peak



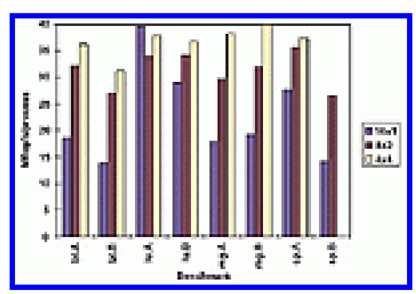
While it is well known that the typical realized performance of most RISC processors is far below peak performance, the latter still appears prominently in the specifications of most supercomputers. Because NPB 2 codes run unmodified, their floating point operation count is essentially the same on different machines, providing an opportunity to report actual performance in Mflop/s (millions of floating point operations per second). Noteworthy is that the relative actual performance does not match relative peak performance. Peak performance (in Mflop/s) per processor is as follows: 266 for the IBM SP2; 360 for the SGI Power Challenge Array; 150 for the CRAY T3D; and 75 for the Intel Paragon. We attribute the relatively high performance of the SP2 nodes to its extremely high-memory bandwidth -- over 2 gigabytes per second (GB/s) between the processor and main memory. On the other hand, the SP2 shows the worst scalability, indicating that its network is underpowered relative to its processors.

The strength of the SP2 also shows up strikingly in a comparison of "efficiency" -- that is, the percentage of peak performance achieved on NPB 2 codes. For small numbers of processors, the SP2 nodes attain close to 25 percent efficiency, while the others hover around 10 percent.

Regression Testing and Configuration

NPB 1 results are typically available for only one machine configuration, with one set of system software. Because NPB 2 source code is available, NPB 2 codes can be run on many configurations of the "same" computer. We examined different configurations of the HPCC-funded Power Challenge cluster at the NAS Facility. The cluster, which offers much flexibility in how jobs are run, comprises several multiprocessor SMP nodes that are connected by a HiPPI channel, using a low-latency MPI implementation. The NPB team at NAS ran 16-process jobs in the following configurations:

- 16 processes on a single node (with 18 processors)
- 8 processes on each of two nodes (with 8 processors each)
- 4 processes on each of four nodes (with 8 processors each)



There are several competing effects. Within a node, all processors share a single bus, so there's more memory contention with more active processors; however, communication latency is quite low -- around 18 microseconds. Between nodes, communication transfers over HiPPI, which has a higher latency (about 110 microseconds) and lower aggregate bandwidth. To our surprise, the results showed quite clearly that it can be more effective to distribute 16 processes over four nodes, reducing memory contention. Results are shown in Figure 2.

Future NPB Releases

By the end of 1996, the NPB team will release a complete set of MPI -- F77 codes for NPB 2. At present, the suite includes the SP, BT, LU, and MG benchmarks; to this will be added FT, IS, CG, and EP.

The NPB group is also working on HPF (High Performance Fortran) versions of several of the benchmarks. At least two HPF implementations will be available by the end of the year.

For a quick explanation of NPB 1, see NAS News, January -- February 1996.

Next Article | Contents | Main Menu | NAS Home

Ames' Semiconductor Device Modeling in the 21st Century

by Subhash Saini

Editor's Note: *First of a two-part series. In the next issue of NAS News, a discussion of the importance of parallel adaptive device modeling using unstructured meshes will be presented.*

NASA Ames Research Center (ARC) has instituted a new program in semiconductor process and device modeling to ensure that NASA will be able to meet future requirements in high-performance computing, advanced scientific instruments, and advanced materials. This program involves using the NAS Facility's highly parallel computer systems to simulate semiconductor devices. The creation of partnerships with Stanford University and UC Berkeley, as well as Silicon Valley companies, will also be considered.

NASA requirements for high-performance computers are different from those of mainstream industry. For many applications, such as the HPCC Remote Exploration and Experimentation (REE) project, NASA requires low-powered, ultra-compact, high-performance computers that are resistant to radiation damage. Inevitably, NASA must pursue the modeling of semiconductor devices that further the possibility of manufacturing electronic and nanoelectronic devices based on quantum mechanical effects to achieve the desired functionality.

In spite of the seemingly obvious need for highly parallel computing as a simulation and design tool in this arena, there has been relatively little utilization of parallel systems anywhere, up to now. The principal impediments include access to parallel testbeds, as well as the availability of usable software and parallel computing expertise. The ARC program is designed to help remove these impediments by developing and porting device modeling codes on highly parallel machines, as discussed in the following sections.

Computational Methods

Boltzman transport equation (BTE), a basic physics equation used to study the transport of particles, is used in such diverse fields as plasma physics, rarefied gas dynamics, neutron transport in reactors, and electron transport in semiconductors. Semiclassical BTE is an integro-differential equation in the seven-dimensional phase space (three spatial coordinates, three momentum coordinates, and one time dimension) for which a closed form solution does not exist. The only possible solution for semiclassical BTE requires making several assumptions. The local rate of change of the distribution function for BTE is equal to the sum of the three terms: (a) diffusion term due to spatial gradient, (b) drift term due to an external electric field, and (c) various scattering events computed with quantum mechanical methods.

Currently, no single method can be employed to perform device modeling and simulations. Much like computational fluid dynamics methods, which offer tradeoffs between accuracy and CPU time consumed, semiconductor device modeling relies on a hierarchy of computational methods, as shown in the table.

MICROSCOPIC METHODS	MACROSCOPIC METHODS
Stochastic Methods	Series Expansion Methods
Full-band Monte Carlo	Spherical Harmonics
Reduced-band Monte Carlo	Laguerre Polynomials
Deterministic Methods	Moment Methods
Scattering Matrix	Classical Hydrodynamics
Cellular Automata	Quantum Hydrodynamics
Non-equilibrium Methods	Energy Transport Methods
Green's Function	Drift-diffusion
	Energy Balance

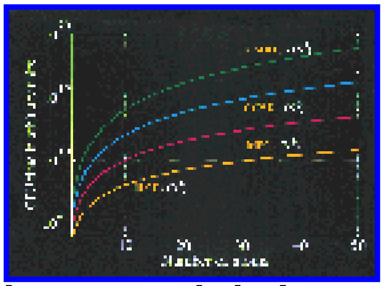
Most of the existing computational methods use either one-dimensional (1D) or two-dimensional (2D) approaches implemented on a single workstation processor. Through the drastic reduction of feature size, integrated circuit (IC) technology has reached a point where the complexity of devices requires a fine-grained three-dimensional (3D) analysis to obtain sufficient accuracy in computer simulations. The computational resources required by 3D semiconductor device modeling are very high and can be met only with high-performance parallel computers. The ARC program will implement selective 2D and 3D versions of these computational methods on high-performance computers such as the HPCC-funded IBM SP2 at the NAS Facility.

Atomic Level Simulations

The accurate simulation of silicon device structures begins with accurate geometry and material properties of that structure. To make advances at a deep sub-micron regime, the physics of processes at the atomic level must be understood. These processes include the formation, activation, and interface properties of point defects formed at the silicon dioxide-silicon interface of a semiconductor device. Understanding defect structures and the interface effect is crucial for accurate device modeling at this level. Accurate -- but practical -- ab initio methods are needed to compute energy and charge density for a silicon dioxide defect in silicon. The other need for atomic-level simulation involves ion implantation, where relevant physics quantities of interest must be extracted using ab initio methods of quantum chemistry and made integral to device modeling.

Full configuration interaction (FCI) is the most accurate method to compute these physical quantities. In this method, careful mixing of multiple wave functions, including the excited states, amounts to incorporating electron correlations. However, the set of possible configurations becomes so large that the complexity of FCI is on the order of (10 N)!, where N is the number of atoms, and where we assume 10 basis functions in an atom. As a result, FCI computations converge slowly and are very expensive, rendering this method useful only for one- or two-atom systems.

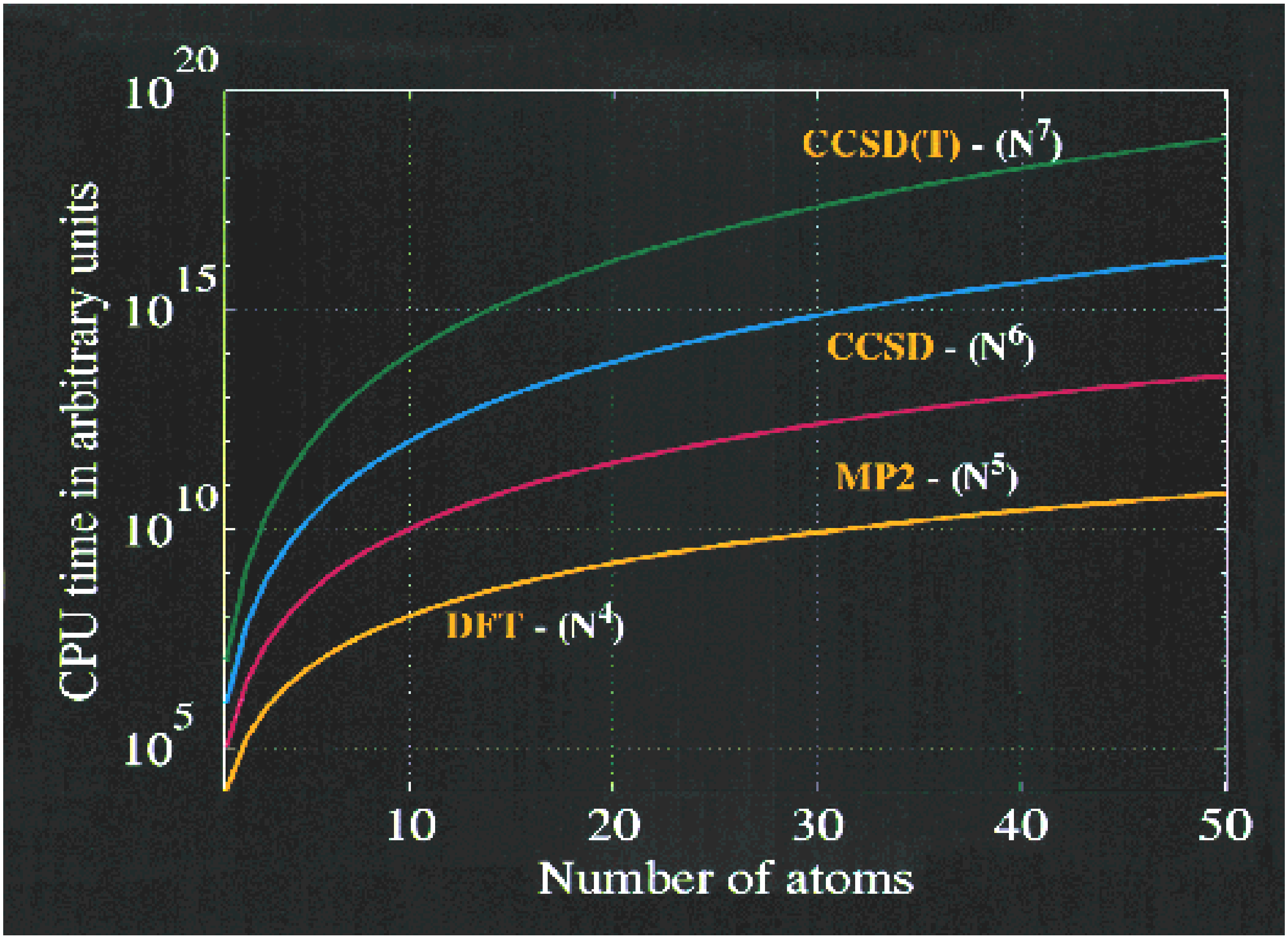
Since the FCI method is not feasible for most problems, the coupled-cluster singles and doubles (CCSD) method, with a perturbational estimate of the connected triple excited CCSD(T) method, is probably the most accurate and practical approach in use today. The computational complexity of CCSD(T) is proportional to (10 N)7. This can be reduced by ignoring the triple excited states. The computational complexity of CCSD is proportional to (10 N)6. Another approximation is the Moller-Plesset second-order perturbation theory (MP2), with a computational complexity on the order of (10 N)5. An approximation is the density functional theory method with a complexity of (10 N)4.



The scaling of several available computational ab initio methods is shown in the figure, which clearly illustrates how quantum mechanical methods can exhaust the capabilities of even the most advanced supercomputers. Ab initio methods could be used to compute the potential energy surface of larger systems. The ARC program will use the density functional theory method, in conjunction with both the molecular mechanics method and the molecular dynamics with the long-range forces method, to simulate various physical processes in semiconducting materials.

Next Article | Contents | Main Menu | NAS Home

MICROSCOPIC METHODS	MACROSCOPIC METHODS
Stochastic Methods	Series Expansion Methods
Full-band Monte Carlo	Spherical Harmonics
Reduced-band Monte Carlo	Laguerre Polynomials
Deterministic Methods	Moment Methods
Scattering Matrix	Classical Hydrodynamics
Cellular Automata	Quantum Hydrodynamics
Non-equilibrium Methods	Energy Transport Methods
Green's Function	Drift-diffusion
	Energy Balance

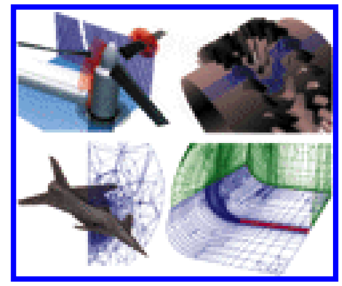


Scaling of ab initio methods for semiconductor process and device modeling. Even for small systems, only density function theory calculations are feasible on currently available high-performance computers.

Graphic by Subhash Saini.

Back to the article.

Field Encapsulation Library Solves Grid Dependence Problems



by Steve Bryson and David Kenwright

A new tool that allows developers to write scientific visualization algorithms and systems in a grid-independent manner is now available to NAS users. The Field Encapsulation Library (FEL) makes it simpler and faster to develop these algorithms and systems and to support many computational grid types.

Frees Programmers From `Tedious Work'

FEL's design uses a C++ class hierarchy, which permits new visualization techniques to be developed without knowledge of the underlying grid structure. This is a significant benefit for applications programmers -- freeing them from the tedious work of loading and managing data, and allowing them to concentrate on developing and testing new visualization techniques.

Furthermore, they need only do a single implementation of a given algorithm, and it will automatically work with every supported grid type. A new grid type can be rapidly implemented by following a standard object template.

The real benefit of this system is that all visualization techniques that access data through FEL will automatically work with the new grid -- without doing a single modification.

Addresses Problems In CFD Simulation

CFD (computational fluid dynamics) simulations are performed on a variety of grid types, including:

- time-varying, multizone, curvilinear grids
- periodic, time-varying, multizone curvilinear grids
- adaptive, time-varying, Cartesian grids
- adaptive hybrid grids (combined structured and unstructured)
- adaptive unstructured grids with multiple element types

This wide variety of grid types has complicated the development of general-purpose visualization systems. Algorithms used for interpreting the results of CFD simulations can depend strongly on the type of grid -- an algorithm written for one grid type may not work on another. FEL was designed with NAS industrial partners such as Boeing and Pratt & Whitney in mind, as well as other NASA centers, all of whom use many grid types.

Designed for `Very High Performance'

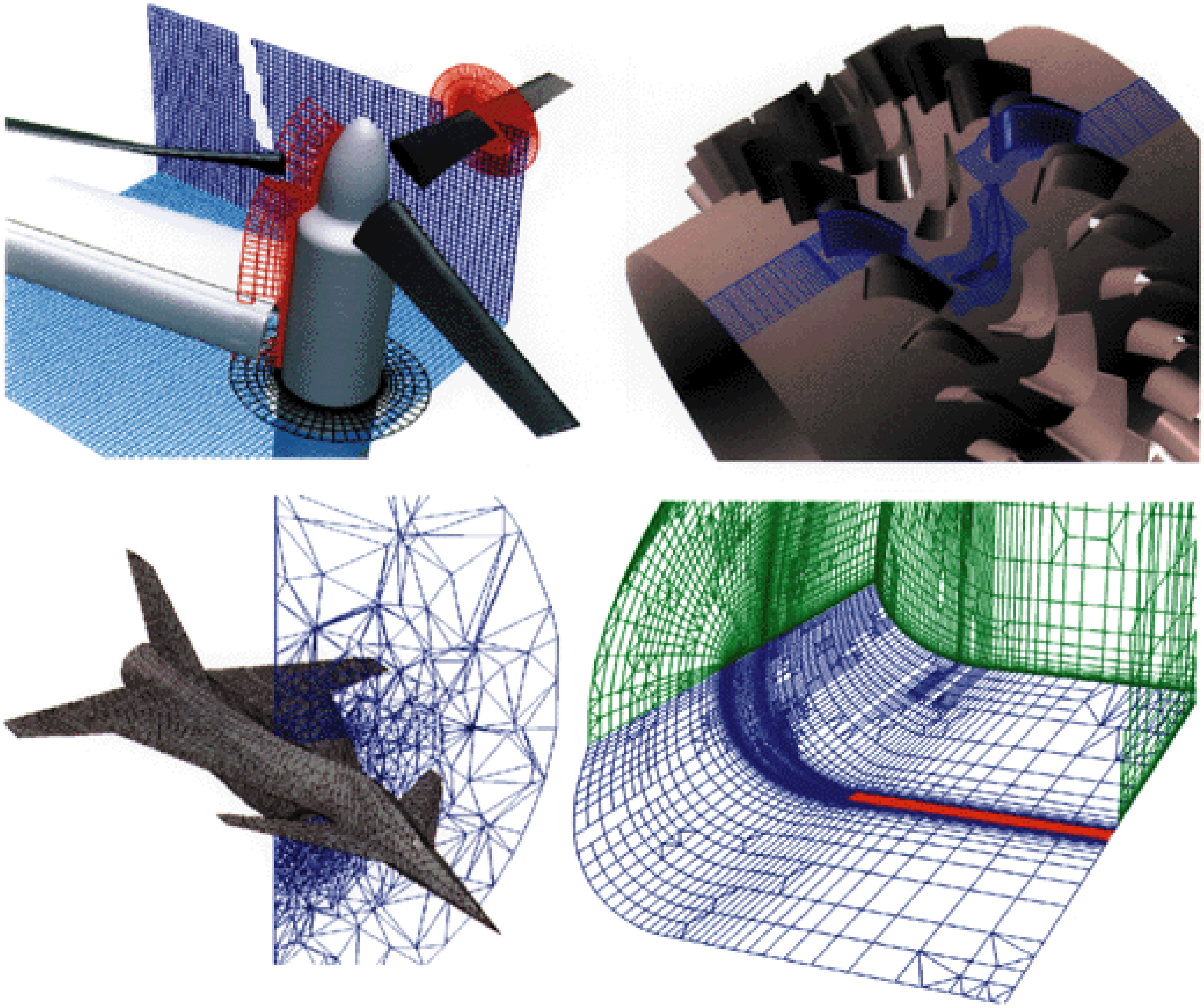
Achieving very high performance was an overriding factor in FEL's design. Near-real-time computations are essential for interactive visualization systems such as the Virtual Windtunnel (VWT), developed at NASA Ames. In order to maintain the performance of such systems, optimized search routines were developed to provide fast data access.

FEL has been integrated in two visualization systems: the VWT and pV3, developed at Massachusetts Institute of Technology. The alpha version, released in March, supported time-varying, multizone, curvilinear grids. Initial tests in the VWT showed that computation times for time-independent visualizations were not degraded. Support for other grid types, including unstructured grids, periodic curvilinear grids, and Cartesian grids, is underway.

The library's function will largely be transparent to end users (such as CFD scientists), although they will appreciate the ability to load and visualize previously unsupported grid types.

FEL was developed by a NAS team consisting of Steve Bryson, Chris Henze, David Kenwright, and Ravi Samtaney. Past contributors include Eric Barszcz, Sandy Johan, Hung Nguyen, and John West.

Next Article | Contents | Main Menu | NAS Home



The Field Encapsulation Library will permit current and future visualization systems to support a wide variety of computational grid types including (clockwise, from top left) multizone curvilinear, multizone periodic, hybrid, and unstructured, grids. Its object-oriented design also permits other grid types to be rapidly integrated.

Images courtesy of Robert Meakin, Karen Gundy-Burlet, NASA Langley Research Center, and Rupak Biswas, respectively.

Back to the article.

Converting PVM Codes to MPI

by Steve Heistand

The process of converting existing PVM (Parallel Virtual Machine) code to MPI (Message Passing Interface) code is not as involved as you might first think. The most difficult part lies in the fact that MPI does not spawn processes; thus, both the "master" and "slave" processes must be in the same executable program. Unlike PVM, MPI does not use the master/slave programming paradigm. Instead, MPI starts up the same binary on all processes in the group. The function of this executable may differ on each processor but the same executable is used.

Ideally, these differences are taken into account when writing the code. After the fact, the easiest way to incorporate the PVM method into the MPI method is to use the *if..then* .. *else* .. *endif* statement, placing the master source in one part of the *if* check and the slave source in the other.

What this ends up looking like is:

```
C    Initialize MPI
     call MPI_Init(integer ierror)
     call MPI_Comm_size(MPI_COMM_WORLD,
         integer size, integer ierror)
     call MPI_Comm_rank(MPI_COMM_WORLD,
         integer rank, integer ierror)
     if ( rank .eq. 0 ) then
C master part
     else
C slave part
     endif
```

Core Subroutines

The conversion from PVM to MPI is relatively simple because the core subroutines in both are almost identical. The subroutines used in initializing are slightly different, but the send and receive calls are very similar. Using the above code segment, the following example explains the calls.

Initialization:

The PVM calls to subroutines PVMFMYTID() and PVMFPARENT() -- if used -- are replaced with the following calls:

```
MPI_Init(integer ierror)
MPI_Comm_rank(MPI_COMM_WORLD, integer rank,
    integer ierror)
MPI_Comm_size(MPI_COMM_WORLD, integer size,
    integer ierror)
```

The first subroutine "enrolls" the process in MPI. The value of *ierror* on return from the subroutine is normally zero, with negative indicating an error.

The next two calls return information about the overall process setup. The size variable gives the total number of processes, and the rank is a number that corresponds to the relative process number. This value ranges from zero to size-1.

MPI_COMM_WORLD is a predefined parameter which refers to the communicator that includes all the processes. In MPI, a communicator is similar to a group in PVM. There is also a similar group concept in MPI. Although the communicator and group are similar, they each have a distinct and separate purpose under MPI. Since the communicator will suffice for the most part, it will be discussed here.

Sends and Receives Explained

The calls for passing messages are in general slightly less complicated under MPI than in PVM, although once you start exploring the full range of possibilities in MPI it does get complex.

Gone are the buffer initialization calls and the calls to pack data found in PVM. Just one call is needed to specify what data to send, what type it is, and who it's going to. A similar single call is done for the receiving end.

The complex parts of even a simple send/receive pair are the concepts of buffered versus non-buffered, synchronous versus asynchronous, and blocking versus non-blocking. Only the latter will be discussed here.

- More information on the other concepts..

PVM uses blocking receive calls; the send calls are mostly non-blocking, particularly as they return after the data has been sent to the remote end. (A receive need not have been called at that point.) The call can also block until the message will fit into memory on the receiving end. If one node gets overloaded with messages, the buffer space may fill up such that the send will not return right away.

MPI's normal mode of blocking calls goes like this: the send blocks until a receive has been called and the message is safely on its way; the receive blocks until all data has been loaded into memory. The send can also return if there is sufficient buffer space in which to copy the message before sending it; however, this is highly implementation dependent and not guaranteed. In either case, once the call has returned it is safe to use the data that was sent or received.

Non-blocking Feature is Efficient

The non-blocking aspect of MPI is useful for those who want to (or can) overlap computation and communication efficiently. This can be done by posting the send or receive call and then going to another task that doesn't involve the data just sent or about to be received. The data can't be used until the call has been successfully completed. A "request" variable in the non-blocking calls can be checked periodically to determine the status of the call.

Putting this all together, the following examples show the original PVM code and the corresponding MPI code.

Initialization:

```
    call pvmfmytid( mytid )
```

to

```
    call MPI_Init( info )
    call MPI_Comm_rank(MPI_COMM_WORLD,
        rank, info)
    call MPI_Comm_size(MPI_COMM_WORLD,
        size, info)
```

Messages:

```
    call pvmfinitsend(PVMDEFAULT, info)
    call pvmfpack(REAL8,    data, n, 1, info)
    call pvmfsend(to_tid, msgtype, info)
```

to

```
    call MPI_Send(data, n, MPI_REAL, to_tid,
        msgtype, MPI_COMM_WORLD, info)
    call pvmfrecv(from, msgtype, bufid)
    call pvmfunpack(REAL8, data, n, 1, info)
```

to

```
    call MPI_Recv(data, n, MPI_REAL, from,
        msgtype, MPI_COMM_WORLD, status, info)
```

Barriers:

```
    call pvmfjoingroup(group_name,
        my_group_rank)
    call pvmfbarrier(group_name, count, info)
```

to

```
    call MPI_Barrier(MPI_COMM_WORLD, info)
```

These are just a few examples of converting PVM code to MPI. A more extensive list Heistand is creating includes other examples. If you have any questions about PVM to MPI conversion, send email to heistand@nas.nasa.gov.

Next Article | Contents | Main Menu | NAS Home

New Program for Generating Surface Oil Flow Visualizations is Now Available to Researchers

by David Kao

Editor's Note: *AFLIC was written by Arthur Okada, formerly at the NAS Facility. Okada worked with David Kao to investigate numerical surface oil flows. The program is based on early work by Brian Cabral and Leith Leedom at Silicon Graphics Inc., who introduced a method for blurring texture images using Line Integral Convolution (LIC). Later, Lisa Forssell, formerly at NAS, extended this method to include curvilinear grids.*

A new visualization program, AFLIC, is now available to NAS researchers who need to generate surface flow patterns from their numerical flow simulations. These patterns can reveal certain flow characteristics, such as separation and reattachment. In wind-tunnel experiments, a common technique is to coat the surface of the model -- for example, a turbine engine -- with an oil and paint mixture. During the experiment, an air stream flows over the solid body and disturbs the oil and paint, forming surface flow patterns. AFLIC generates patterns that resemble the results of these experiments.

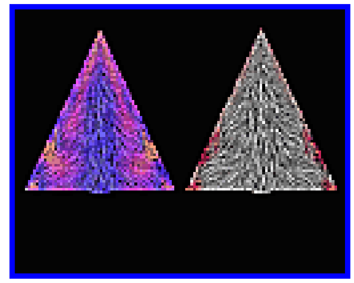
Provides Continuous Flow Patterns

CFD scientists have traditionally used two techniques to visualize flow patterns on numerical flow fields. The first is to generate vector arrow plots; for each grid point, a vector arrow is created. The length and direction of the vector are based on flow magnitude and direction at each grid point, and the vectors reveal an overall flow pattern. The second is to track a particle from each grid point near the surface for some time interval. The path of the particles gives an overall flow pattern.

While both techniques are effective, the resulting images depend on the placement of the vectors and particles. This placement is not required in AFLIC. Furthermore, the flow patterns generated are continuous rather than discrete.

AFLIC uses the Line Integral Convolution (LIC) method to create texture-mapped images that resemble flow patterns. The researcher starts with an image and a vector field as the input. The image is then blurred to reveal the flow pattern of the vector field. For example, if a black-and-white noise image (similar to a TV tuned to an empty channel) is used as the image and a circular velocity field is used as input, the resulting output is a black-and-white image with circular flow patterns.

Helps Detect Flow Features



AFLIC provides several techniques to help researchers detect flow features. In addition to several filtering schemes to sharpen the LIC images, it includes techniques for color flow textures that are computed based on velocity direction and velocity angle. Coloring the flow texture by direction makes changes in flow direction easy to see. However, when the grid surface is completely shaded by some color maps (left, in figure) the color sometimes appears more dominant than the underlying flow pattern.

AFLIC solves this by coloring the flow texture only at locations where the flow changes abruptly (right, in figure). This allows scientists to easily detect flow separation and reattachment features. These and other techniques are detailed in "Enhanced Line Integral Convolution with Flow Feature Detection," NAS Technical Report NAS-96-007.

AFLIC handles PLOT3D multizone curvilinear grids. The inputs to the program are the flow data files; the output is the computed flow textures. Another program, AFVIEW, written by Arthur Okada, reads AFLIC's output and renders the flow textures. The rendering is disconnected from AFLIC so that users can view the textures repeatedly without recalculation -- running AFLIC once and viewing the results later with AFVIEW.

First Users Found AFLIC Useful

Ames CFD scientists Neal Chaderjian and Robert Meakin have had an opportunity to try AFLIC. Chaderjian used the program to generate surface oil flows from his simulation of a delta wing with rolling motion. The resulting flow patterns gave Chaderjian a good impression of the flow reattachment and separation near the leading edge of the delta wing.

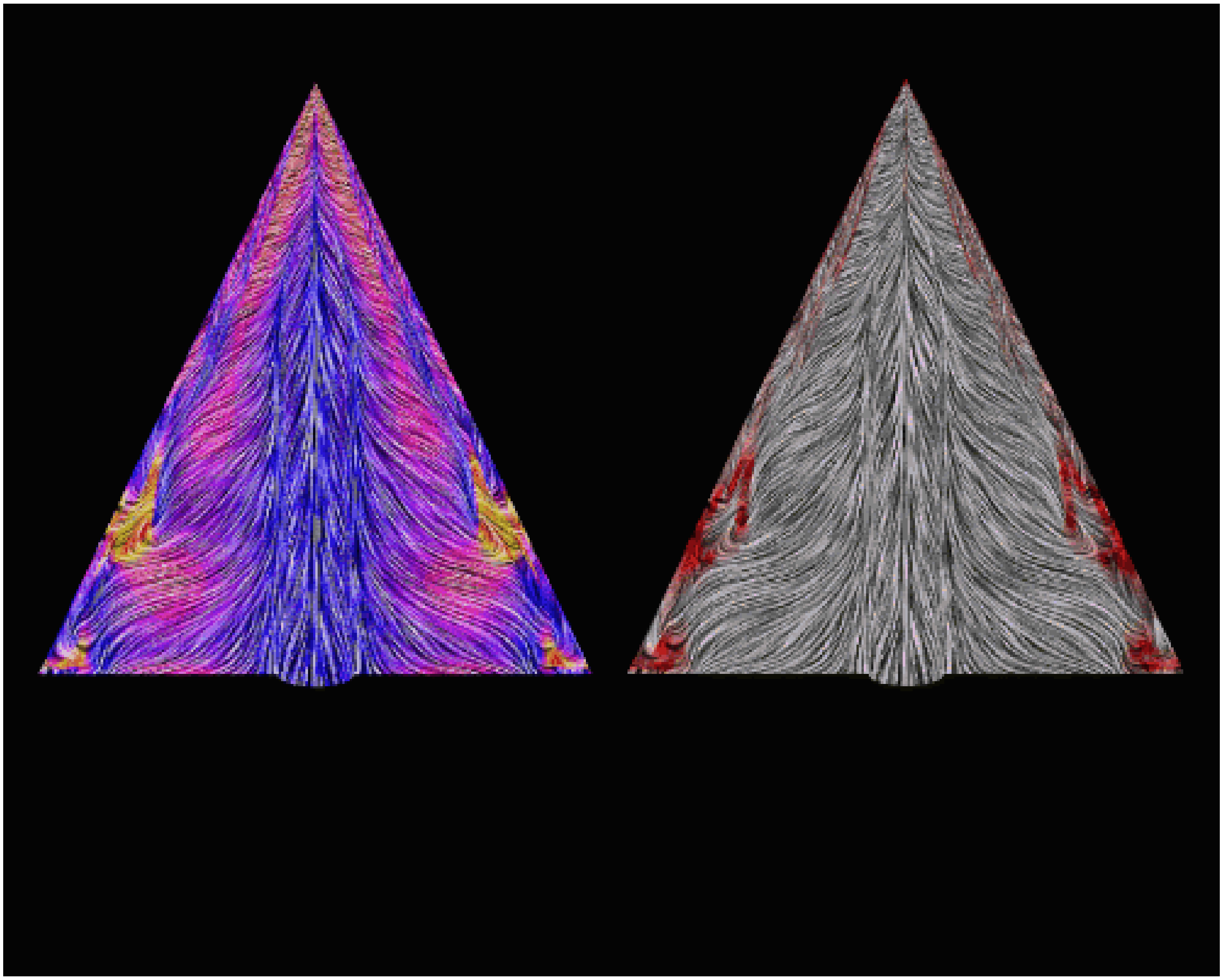
Meakin and Scott Ackroyd (an intern from Utah State University) used AFLIC to compare full-span and semi-span simulations of a V22 rotor and wing configuration. The surface oil flows helped validate the asymmetric flow in the former.

Next Challenge -- Unsteady Flows

NAS researchers are working to add unsteady surface oil techniques to AFLIC. Visualizing flow fields using instantaneous flow visualization techniques can be misleading because the time variable is not considered in the calculation. Based on comparisons of several visualization techniques, this has shown to be true for particle tracing methods. (See "Unsteady Flow Technique Reveals New Phenomena," in the NAS News special section on Unsteady Visualization, July -- August 1995.)

If you're interested in trying AFLIC and AFVIEW, send email to davidkao@nas.nasa.gov.

David Kao works in the NAS visualization technologies group. He is currently developing numerical flow visualization techniques. Kao will be presenting this work at the IS&T/SPIE Symposium on Electronic Imaging Science and Technology next February in San Jose.

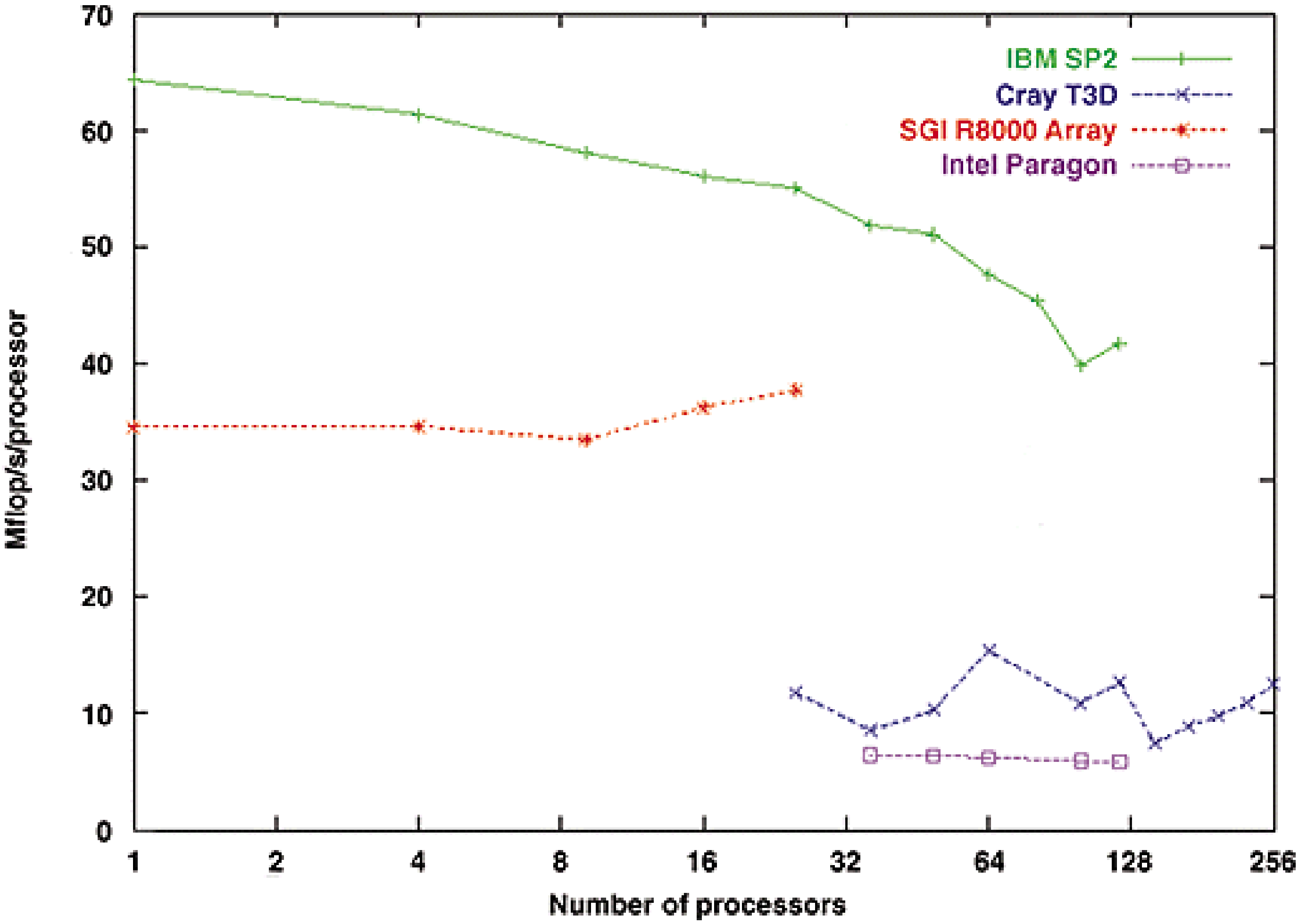


At left is the survace oil flow pattern on a rolling delta wing at 30 degree angle of attack. The wing body is colored based on flow direction.

The identical image (right), colored by AFLIC in certain areas only, allows CFD researchers to quickly spot those regions where abrupt flow changes occur. A detection feature in AFLIC highlights flow searations and reattach;ments in magenta.

Graphic by Neal Chaderjian.

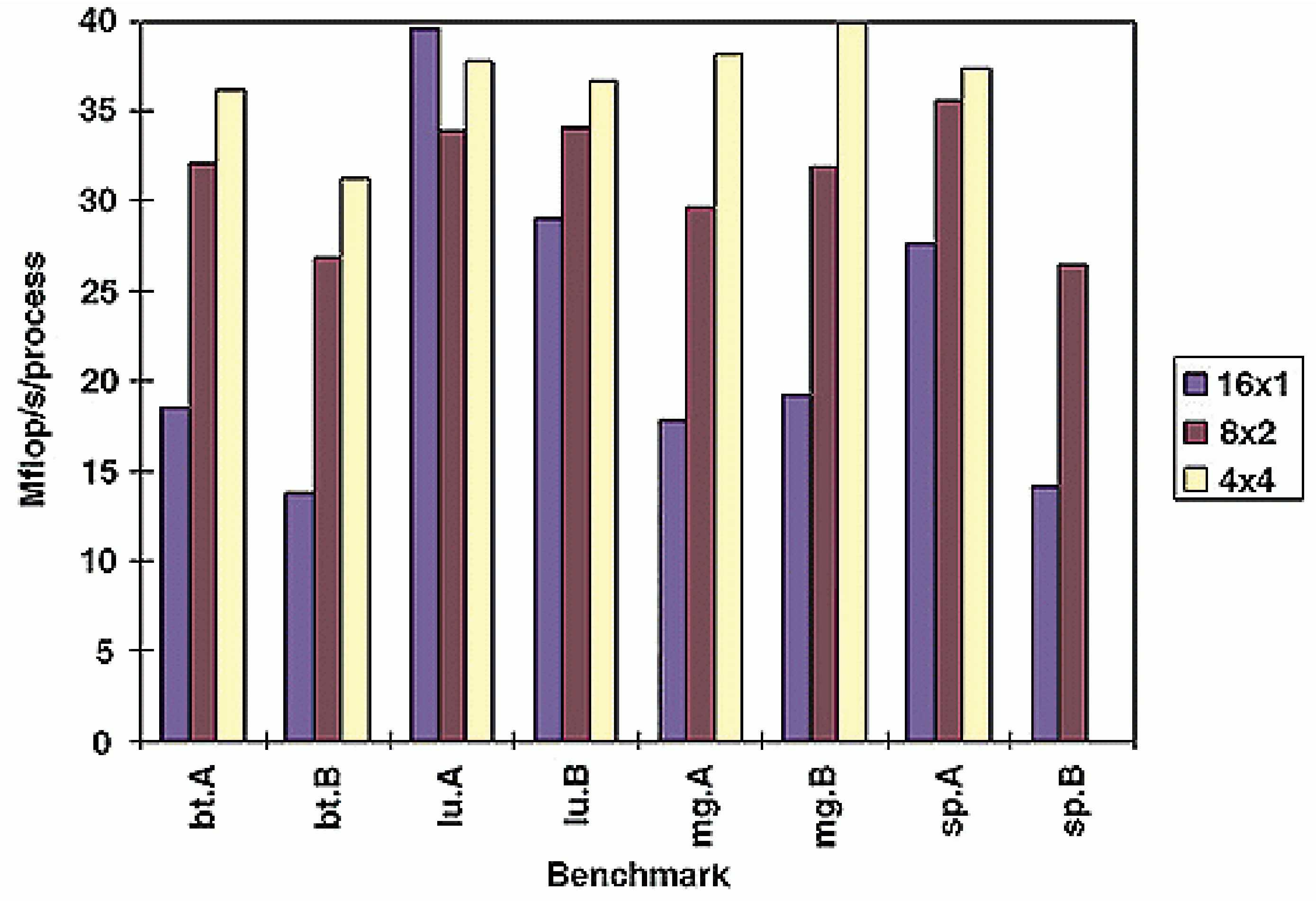
Back to the article.



Performance in Mflop/s per processor on the class A (small) version of the NPB 2 SP benchmark on the IBM SP2, CRAY T3D, SGI Power Challenge Array, and Intel Paragon. It reveals a large disparity in per-node performance between the four machines. The SP2 is the clear leader, obtaining as many as 65 Mflop/s per processor (but as few as 40 Mflop/s per processor on 100 nodes). The SGI is a distant second with about 35 Mflop/s per processor. The T3D trails substantially at about 10 Mflop/s per processor, while the Paragon registers barely more than 5 Mflop/s per processor. This ranking is consistent across all benchmarks.

Graphic by Bill Saphir.

Back to the article.



NAS Parallel Benchmarks performance on different configurations of an SGI Power Challenge Array. Note that 16 processes distributed over four nodes always performed better than the same number of processes on a single node. No data was available for the SP class B benchmark on four nodes.

Graphic by Bill Saphir.

Back to the article.

Lecture Sparks Ideas for Collaboration with Semiconductor Industry

by Elisabeth Wechsler

A presentation at the NAS Facility on September 19 by Ronald Goossens prompted discussion among researchers on the topic of possible collaboration between the semiconductor industry and Ames Research Center. Goossens is director of the Center for Semiconductor Modeling and Simulation and Technology CAD Sciences at Semiconductor Research Corp. (SRC).

SRC, established to solve technical challenges for the chip manufacturing industry, funds research at universities and manages cooperative research between the industry and national laboratories.

Ames Center Director Henry MacDonald introduced Goossens, commenting that the lecture topic, "Semiconductor Modeling and Simulation: A Personal Perspective of the Playing Field," represents an area "very dear to my heart."

Overview of SRC

Goossens gave an overview of his organization's goals and summarized results of SRC's research in the areas of topography simulation and interconnection reliability. Other long-term studies involve grids and computational science, bulk processes, device models, and interconnection performance prediction.

Following the lecture, researchers addressed the critical issues facing the semiconductor industry, areas in which Ames can perform useful services, and NASA's key requirements for semiconductor devices in the future. NAS scientist David Bailey moderated, with participants attending from Ames, Stanford University, and Pennsylvania State University.

Among Ames' principal advantages noted in the discussion were: its location in Silicon Valley; installation of leading-edge high-performance computer systems; expertise in 3D physical modeling, quantum chemistry, numerical methods, parallel computing, and scientific visualization; and the relative ease of software technology transfer.

Areas of Possible Future Study

A list of recommended tasks and research focus areas came out of the discussion:

- Develop a "virtual reactor" to facilitate study of chemical vapor deposition and plasma processes.
- Implement one or more large 3D device modeling codes on the NAS Facility's IBM SP2 or other highly parallel systems.
- Perform advanced quantum chemistry calculations to better understand certain key semiconductor surface phenomena.
- Incorporate some advanced algorithms into device and process codes, such as unstructured grid methods, adaptive grids, new grid generation schemes, and novel algorithms for device layout.
- Perform simulations of opto-electronic devices.
- Develop advanced visualization facilities for device and process codes.
- Study future issues, including quantum effects and the exploding complexity of the design process.

To request videotapes and handouts for this or other NAS training events, contact the NAS Documentation Center.

Next Article | Contents | Main Menu | NAS Home

Next Article | Contents | Main Menu | NAS Home

Seminar Videotapes Available

Most research talks given at the NAS Facility are accessible from the NAS Documentation Center videotape loan program. Procedures for obtaining training event materials are also available online. Information on past training events is also available.

Here is a summary of the some of the most widely attended New Technology Seminars in the last three months.

"An Implicit Algorithm for the Navier-Stokes Equations," presented by Robert W. MacCormack, Stanford University's Department of Aeronautics and Astronautics. He discusses a new procedure for solving the Navier-Stokes equations on structured grids that attempts to minimize approximate factorization error. Convergence rates of approximately 0.8 are achieved for high Reynolds numbers flow past a sphere or through a nozzle. Converged solutions, to at least engineering accuracy, are obtained in about 50 time steps. MacCormack, who worked at Ames for many years, is known as the "father of CFD." (10/1/96)

"Some Experiences with Schur Complement Parallel Preconditioning for CFD Calculations," presented by Tim Barth, NAS algorithms, architectures, and applications group. This work is a joint effort between Barth, Tony Chan (University of California, Los Angeles), and Wei-Pei Tang (University of Waterloo, Canada). The research looks at solving matrices that arise from the discretization of advection-diffusion field equations on arbitrary domains using stabilized numerical methods. Various CFD examples are shown to demonstrate the techniques' efficiency and robustness. (9/5/96)

"Weighted Essentially Non-Oscillatory (ENO) and Discontinuous Galerkin Methods for CFD Problems," presented by Chi-Wang Shu, Brown University. Recent work for conservation laws in developing weighted ENO finite differences schemes, as well as Galerkin finite element methods, is described. More efficient than ENO schemes, these methods enable convergence on smooth solutions and render good resolutions for CFD-related shocked problems. (8/23/96)

"The SUIF Parallelizing Compiler," presented by Monica Lam (Stanford University). This compiler, now in the public domain, automatically parallelizes and optimizes sequential programs for shared-memory multiprocessors. (8/22/96)

Next Article | Contents | Main Menu | NAS Home

November-December 1996 -- Vol. 2, No. 21

NAS NEWS

Field Encapsulation Library Solves Grid Dependence Problems

Ames' Semiconductor Device Modeling in the 21st Century

THIS ISSUE