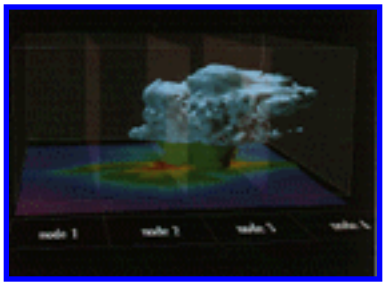
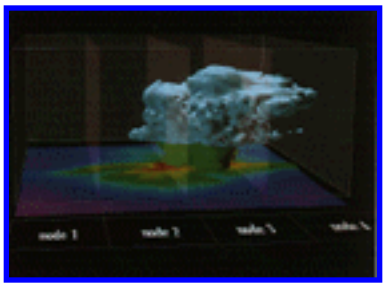
January-February 1994
Vol. 2, No. 1



Challenges and Solutions Shared at DCAA Workshop



by Elisabeth Wechsler

Distributed, or cluster, computing has come a long way in terms of performance benchmarks, and obviously is a technology that's here to stay. However, based on panel discussions and presentations at the NAS-sponsored DCAA Workshop in October, it clearly faces major challenges in terms of setting standards, improving system management, and achieving broader usability. It also is unclear at present how cost effective this style of computing is, overall.

"There's a lot of enthusiasm about distributed computing, but it's still an emerging technology," said Bill Kramer, Chief, NAS Computational Services Branch, and workshop chair. He commented that vendors have shown interest and support only recently, and that the technology "hasn't been used with great success by a wide variety of users," as factors in its immaturity.

The Distributed Computing Aerosciences Applications Workshop (DCAA) was held at NASA Ames, October 18-20. More than 160 participants attended.

Major Issues

Most researchers indicated in their workshop presentations that they were actively grappling with the issues of standardization, cost effectiveness, system management, and usability, while still trying to push the envelope of new technologies in high-performance distributed computing.

Gordon Bell, an independent consultant, advises using general purpose workstations whenever possible to get better performance value per dollar invested, adding that the workstation market's higher volume keeps prices low and encourages vendors to supply software, leading to increased standardization and usability. He touted impressive performance results with both tightly-coupled computers and distributed workstations by various vendors.

Louis Turcotte, of Mississippi State University, seemed to agree: "The risk factor is less with clustered workstations because if the system doesn't work, you at least end up with a bunch of nice workstations."

He reported on experiments with clustered, off-the-shelf, heterogeneous workstations conducted by the National High Performance Distributed Computing Consortium. Gigabit test beds using these commercial products will be available next spring, Turcotte said, adding that the software is based on open standards, such as Message Passing Interface (MPI), and includes C++, parallel I/O, and checkpoint

restarting.

Current Development Work

H.T. Kung, of Harvard University, is designing an Asynchronous Transfer Mode (ATM) switch for high-speed networking that uses portable virtual circuits. In his presentation, Kung described "very ambitious" performance goals of high bandwidth and low latency, along with reliable multitasking with synchronization of information. He claims that ATM runs 100 times faster than Ethernet and permits shared media on a single wire. Kung admits that control is a challenge, especially when running the network at speeds of more than 600 megabits.

Trade-offs include a current high cost ($1,000 for an ATM adapter compared with $60 for an Ethernet card) and the fact that ATM is not yet part of a standard. "This is very important," said Kung, who said he has spent about one-third of his technical time on standards issues over the last four to five months.

Dennis Duke, of Florida State University, reported on cluster computing research conducted on high-performance networks by the Supercomputer Computations Research Institute (SCRI) at Florida State University. Prompted by strong competition for resources and an interest in distributed computing, SCRI began networking high-performance workstations by creating a uniform password map and file system, and by making a commitment to heterogeneous architectures.

SCRI developed the Distributed Queuing System (DQS), which is simple, portable, secure, and offers high fault tolerance, generic queue support, and concurrent multiple node support. DQS 3.0, scheduled for release by the end of 1993, features enhanced scheduling, interactive shell load balancing, increased toolkit support, POSIX compliance, and a user-level migration library.

Vaidy Sunderam, of Emory University, urged the development of more basic solutions, where possible, to cluster computing problems. In terms of schedulers, he likes DQS because "it's not fancy, it does very simple things, and that works well for a number of people." He recommended changing the paradigm of cluster computing so that interactive work can be done more efficiently.

Sunderam, who was part of the development team for Parallel Virtual Machine (PVM) in association with the NAS Applied Research Branch, reported that PVM is proving to be a good technology because it is flexible and built on a networked cluster of heterogeneous machines. PVM is designed to enhance the functionality, cost effectiveness, and performance of a multifaceted environment, he said. User interface changes continue to be cost effective and the software reflects new research within three to six months of release, he added.

"PVM has been used to utilize the existing model and implementation," Sunderam said. "Now we must try out additional platforms to enhance productivity."

An Industry Example

Jeff Johnson, of McDonnell Douglas Aerospace, who reported on distributed parallel processing in Computational Fluid Dynamics (CFD), noted the problem of reduced resources while demands for CFD are up. McDonnell Douglas is experimenting with event driven architecture to improve load balancing in parallel processing.

The key, Johnson said, is using a simple design, which makes the architecture more flexible. Two queues were created for the McDonnell Douglas model: the processor idle queue, which is ordered by relative processor speed, with the most powerful processor at the top of the queue, and the work queue, which is ordered by relative compute time with the longest running entry at the top of the queue. This model defines the cycle as complete only when all tasks are complete and applies the algorithm only to the first iteration. Computing one cycle in one zone is the unit of work, he explained.

Fault tolerance is a "very big issue" at McDonnell Douglas, Johnson said, because other people's machines are being used and testing is heavy. To reduce potential stress, the company has emphasized education and communication between the supercomputing and workstation groups.

From the work done on fault tolerance at McDonnell Douglas, Johnson observed that there are more possible points of failure when hardware and software is involved, and the chance of failure increases with jobs running for more than 12 hours. Since failures cause significant productivity loss, most jobs are run during off shifts.

Standards Needed

Although distributed computing has proven to be viable in some specific work applications, many problems remain to be solved.

"We need a set of experience in order to look at standards issues in a meaningful way," Kramer said, urging patience.

Bell, a strong advocate for RISC-based cluster computing, stated the dilemna of standards more bluntly: "If you're after teraflops you won't be buying a general purpose computer, and with a specialized system you'll be lucky to find an application that runs on it. Many of the specialized machines don't have any secondary memory, so after you run the problem on those machines, there's no way to store the data."

Another problem, Bell said, is that you have to start programming from scratch every three to four years to develop software for the new generation of scalable machines because the older machines' code isn't portable. "This ensures that you'll never have any software for new systems," he said. He urged users and vendors to work harder toward standardizing on libraries, languages, and environments.

Turnaround time is too slow to keep rewriting software," said Miron Livny, of the University of Wisconsin, in a panel discussion on parallelism and distributed schedulers. "We should deal with the

reality of what's available, no matter what the problems, before designing something different."

Jim Patterson, of Boeing Computer Services and also a member of the same panel, put it another way: "We can't continue with everyone rolling their own."

Judging by comments from workshop participants, it's clear that there isn't a lot of agreement yet about standards for distributed computing or how to make the emerging technology work in industry.

As the DCAA Workshop demonstrated, there is a high level of interest in distributed computing and significant work being accomplished, due to the increasing price performance of workstations. While the immaturity of the technology precludes it from being a general solution at this time, this situation may change soon. Some groups, such as the one described above at McDonnell Douglas, are already exploiting the potential of this technology in a very real way, today.

More High Performance Fortran

by Subhash Saini

This article explains HPF data distribution directives, parallel statements, mapping inquiry subroutines, and computational library subroutines.

The High Performance Fortran Language Specification (HPF), includes three data distribution directives that allow programmers to control the distribution of data to processors. The directives are: ALIGN, DISTRIBUTE, and TEMPLATE. They tell the compiler how data objects should be assigned to processors.

The ALIGN Directive

The ALIGN Directive specifies that certain data objects (typically arrays) be mapped in the same way as certain other data objects. Operations between aligned data objects are more efficient because they are mapped to the same abstract processor.

A variety of alignments are available: simple stride, offset, axis collapse, axis transposition, axis reversal and replication.

The following table contains examples of some of these alignments.

Table 1: HPF Available Alignments

Simple alignment	`CHPF$ ALIGN x(i) WITH y(i)` `CHPF$ ALIGN a(i,j) WITH b(i,j)`
Collapse to a smaller dimension	`CHPF$ ALIGN a(i,j) WITH y(i)` `CHPF$ ALIGN c(i,*,k) WITH b(i,k)`
Embedding into a larger dimension	`CHPF$ ALIGN x(i) WITH y(i,1)` `CHPF$ ALIGN y(i) WITH a(2,i)`

The DISTRIBUTE Directive

On a four-processor computer, the DISTRIBUTE directive would resemble the following example:

```
     REAL*8 x(512,512), y(512,512), z(100)

CHPF! DISTRIBUTE x(BLOCK, *), y(*, CYCLIC), z(BLOCK(5))
```

Using the above example, the first processor would store the following array sections:

```
     x(385:512, 1:512)

     y(1:512, 4:512:4)

     z(1:5), z(21:25), z(41:45), z(61:65), z(81:85)
```

The TEMPLATE Directive

The TEMPLATE directive is like an array whose elements have no content, and therefore, occupy no storage. It is an abstract space of index positions that can be distributed, and with which arrays can be aligned.

Templates are useful when several arrays must be aligned relative to one another, but there is no need to declare a single array that spans the entire index space of interest.

For example, the following code is written to align four `n` by `n` arrays to the four corners of a template measuring `(n+1)` by `(n+1)`:

```
     PROGRAM abc

     INTEGER, PARAMETER :: n = 512

CHPF$ TEMPLATE, DISTRIBUTE(BLOCK, BLOCK),

CHPF$&DIMENSION(n+1,n+1)::x

     DOUBLE PRECISION, DIMENSION

     &(n,n) :: a, b, c, d
```

```
CHPF$ ALIGN a(m, k) WITH x(m , k)

CHPF$ ALIGN b(m, k) WITH x(m , k+1)

CHPF$ ALIGN c(m, k) WITH x(m+1, k)

CHPF$ ALIGN d(m, k) WITH x(m+1, k+1)
```

An object can be realigned or redistributed. Redistributing an object causes all objects aligned with it to be redistributed also, in order to maintain the alignment relationships.

An array that does not have the DYNAMIC attribute cannot be realigned, and an array or template that does not have the DYNAMIC attribute cannot be redistributed. Templates cannot be in COMMON, and templates are not passed through a subprogram argument interface.

Parallel Statements

HPF provides a FORALL statement, FORALL construct, and INDEPENDENT directive to allow explicit parallelism in the computations.

Any procedure used in a FORALL statement or construct must be pure. A pure procedure obeys certain syntactic constraints, which ensure that it produces no side effects. A pure function can be invoked concurrently in a FORALL statement.

An example of the use of a FORALL statement follows:

```
FORALL (i=1:n) d(i) = a(i,i)
```

An example of a FORALL construct follows:

```
FORALL (i =1:n-1)

     FORALL (j = i+1, n)

          x(i,j) = x(j,i)

     END FORALL

END FORALL
```

The INDEPENDENT Directive

The INDEPENDENT directive specifies to the compiler that there is no data dependency; for example, no array element is defined by one iteration of a DO loop which is read or written by another. This directive implies that the DO loop can be executed in parallel, or that the FORALL statement can be executed without copying an accessed array.

An example of an INDEPENDENT directive follows:

```
CHPF$ INDEPENDENT

      DO i = 1,512

           x(index(i)) = y(i)

      ENDDO
```

Here, the code asserts that the index does not have any repeated entries, and `x` and `y` are not storage associated.

System Inquiry Functions

HPF provides two system inquiry functions. They are NUMBER_OF_PROCESSORS and PROCESSORS_SHAPE.

NUMBER_OF_PROCESSORS displays the total number of processors available to the program along a specified dimension of the array. PROCESSORS_SHAPE displays the shape of the array.

These functions allow HPF programs to run on computers whose configurations are not known at compile time. For example, on the NAS Intel Paragon XP/S-15 with 208 compute nodes, the syntax is as follows:

```
      NUMBER_OF_PROCESSORS() is 208

      NUMBER_OF_PROCESSORS(DIM=1) is 16

      NUMBER_OF_PROCESSORS(DIM=2) is 13

      PROCESSORS_SHAPE() is (/16,13/)
```

Mapping Inquiry Subroutines

HPF provides three mapping inquiry subroutines that describe how an array is actually mapped onto the physical processors of the computer. They are: HPF_ALIGNMENT, HPF_TEMPLATE, and HPF_DISTRIBUTION. These subroutines tell the user which HPF distribution directives the compiler has ignored. Since these directives are guidelines to the compiler, and, as such, are advisory in nature, the compiler may ignore these directives and distribute data in a manner more efficient than the user specified.

Computational Library Functions

Fortran 90 contains few basic operations useful for designing parallel codes. As a result, HPF includes several classes of such parallel basic operations.

HPF defines a library of 45 computational functions to be provided as a Fortran 90 module: four reduction functions; 12 combining-scatter functions; 12 parallel-prefix functions; two sorting functions; and three bit-manipulation functions.

Some operations are difficult or impossible to express directly in a high-level machine-independent language such as HPF. Extrinsic procedures define an explicit interface to procedures written in other paradigms, such as SPMD or explicit message-passing subroutine libraries. This interface can also be used to interface HPF to other languages, such as C++.

Sequence and Storage Association

Storage association is the association of two or more data objects that occurs when two or more storage sequences share, or are aligned with, one or more storage units.

Sequence association refers to the order of array elements that Fortran requires when an array expression or element is associated with a dummy array arrangement. For example, the rank and shape of the actual argument need not agree with the rank and shape of the dummy argument.

Both storage association and sequence association are natural concepts only in computers with linearly addressed memory. HPF allows the mapping of variables across multiple processors in order to improve parallel performance through the locality of data by aligning the different data items.

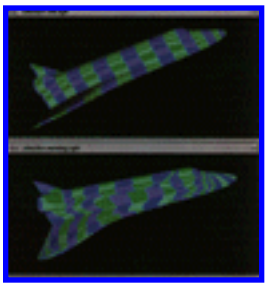
Fortran 77 and Fortran 90 COMMON and EQUIVALENCE statements constrain the alignment of different data items to localize the data for computation. HPF allows the codes that rely on storage and sequence association, but their full support is not compatible with the HPF goal of high-performance through data distribution.

In both cases, HPF provides directives to assert that full sequence and storage association for affected variables must be maintained. Without these explicit directives, reliance on the properties of association is not allowed. A compiler may then choose to distribute any variables across processor memories to

improve performance. HPF also defines the relationship between HPF data mapping and Fortran sequence and storage association.

For Saini's research on HPF, see http://www.nas.nasa.gov/Pubs/TechReports/NASreports/NAS-95-010/NAS-95-010.html.

Visualization '93



by Pamela P. Walatka

The IEEE Visualization '93 (Vis93) conference held in San Jose, CA in October was a small conference representing sky-high-end scientific visualization. Vis93 brought together the top experts in the field of making scientific data visible.

NASA Ames Research Center was well represented. Steve Casner, Ames Flight Human Resources Branch, and Kristina Miceli, NAS Applied Research Branch, organized the workshop "Intelligent Visualization Systems."

Other members of the Applied Research Branch presented tutorials. Steve Bryson was a co-instructor for "Virtual Reality for Visualization." Horst Simon taught "A guided Tour of High Performance Computing: Architecture, Software, Applications." And "Vector Field Topology" was presented by Daniel Asimov, Al Globus, and Creon Levit.

David Lane, who developed the Unsteady Flow Analysis Toolkit (UFAT) and is a member of the NAS System Development Branch, presented a paper, *Visualization of Time-Dependent Flow Fields*.

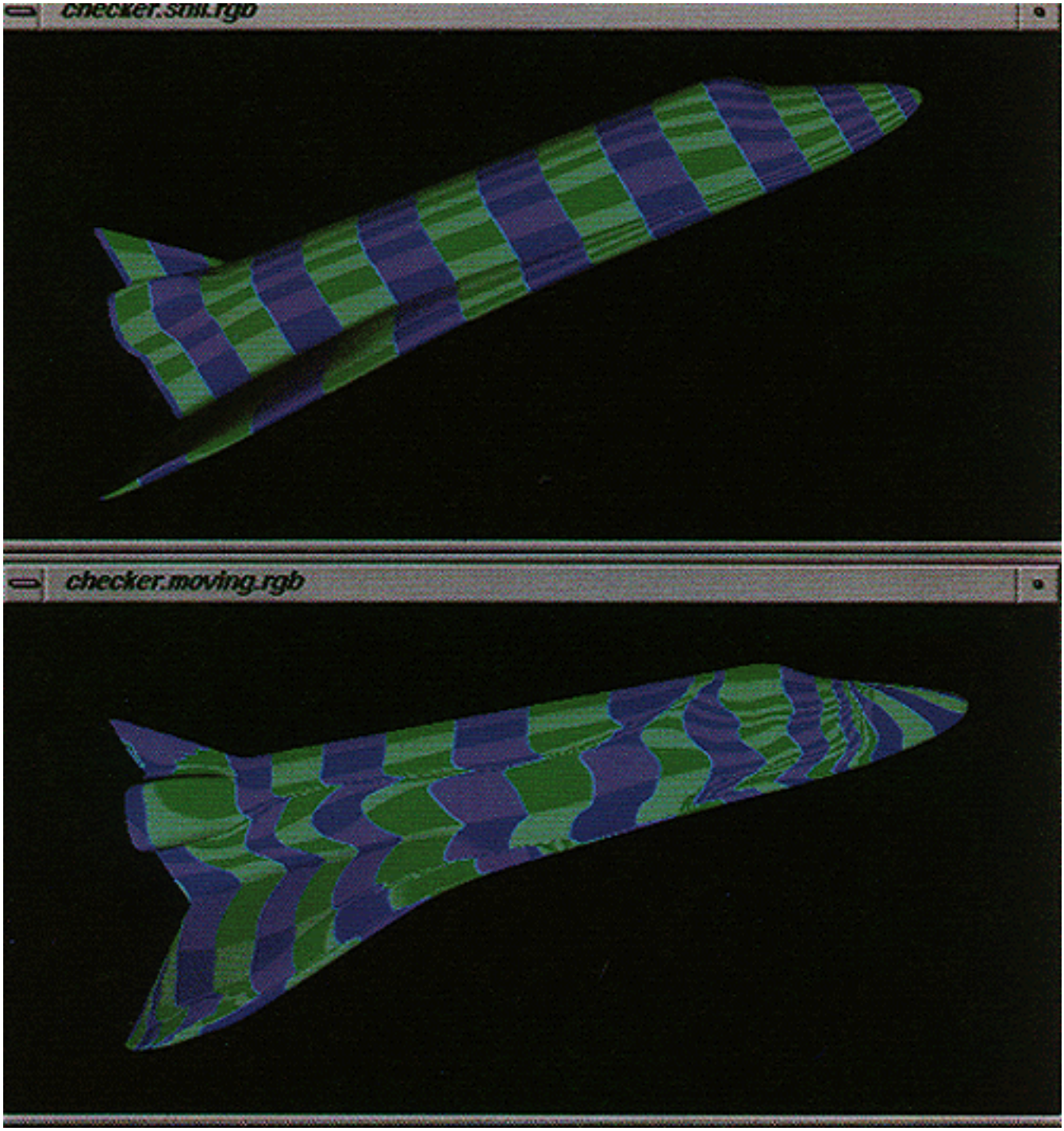
The NASA Ames demonstration booth was staffed by members of the FAST (Flow Analysis Software Toolkit) team: Jean Clucas, Rajiv Baronia, Kevin McCabe, Fergus Merritt, Tim Sandstrom, David Schmitz, and John West (all of the NAS Systems Development Branch). Merritt, who heads the FAST team, said, "One of the best things about the Visualization conference is meeting face-to-face with people that we have worked with over the phone and via email." Clucas offered copies of the first-ever FAST programmer's Manual.

A new video, *NAS Video Production Using FAST*, was released in December. The nine-minute video was produced by Chris Gong and narrated by Kevin McCabe (both of the NAS Computational Services Branch), and gives an overview of FAST and the video production services available through the NAS Graphics Lab. Copies of the video are available from the NAS Documentation Center.

Lisa Kerstin Forssell, a Stanford graduate student interning at NAS, demonstrated a flow visualization technique she developed. Forssell maps a texture onto a surface, and then stretches the texture as it would be moved by flow.

Overall, the conference was a forum for new ideas and trends. For example, the visualization of *software* is gaining momentum. Stephen Eick, of AT&T Bell Laboratories, Naperville, IL, presented a tutorial on visualization of large software systems. Images are used to convey information such as code history and

performance. An example here at NAS is the NAS Trace Visualizer (NTV), developed by Louis Lopez of the Systems Development Branch. NTV visualizes what each processor has done during execution of a parallel program.



These are textures mapped to show flow on surfaces.
Forssell also creates images of the vector field using Line Integral Convolutinn, which blurs a noise texture according to the vector field. The convolution filter can then be phase-shifted in successive images to create the impression of flow.
Images courtesy of Lisa Kerstin Forssell

to the article.

NAS Contributes to Supercomputing '93 Success

by Elisabeth Wechsler

Supercomputing '93 held few surprises for most NAS attendees, and the conference seemed to reflect a maturing industry, with its turnout of more than 5,200 participants, dozens of vendor exhibitions (including a booth by PC-based Microsoft Corp.), the presence of more lawyers than ever before on the exhibit floor, and the fact that most announcements of new technology had been orchestrated to the press several weeks before the November 14-19 conference began in Portland, OR.

However, the very fact that this conference is held annually and gets bigger every year, attracts most vendors even remotely associated with supercomputing, and enjoys wide participation in the user community, makes it an important benchmark for year-round efforts at NAS and at other institutions, said Doreen Cheng, who moderated two SC'93 sessions.

SC'93 acts as a catalyst by setting a big milestone, agrees Horst Simon, adding that a lot gets done in the name of the conference; for example, *NAS Parallel Benchmarks Results*, which was distributed informally at the conference.

"A lot of important work is accomplished in small informal sessions among people for whom there is the one big annual meeting where we all get together," he continued. For example, High Performance Fortran, Message Passing Interface, and Parkbench were developed this way.

Sense of Realism

"There's more of a sense of realism and scientific objectivity in the field of supercomputing now than there was before," commented David H. Bailey. He observed a move away from the "initial giddiness" that characterized the industry a few years ago, to a feeling of pessimism more recently, and now back to a middle ground. "People recognize the challenges of parallel supercomputing and are trying to meet them by doing good, responsible work," he said.

Among the NAS contributions at SC'93 were: the entire conference network configuration, which achieved some technological "firsts"; the participation of several NAS scientists and researchers who presented papers and tutorials or chaired workshop panels; and a well-attended booth with popular demos of FAST, NTV, and AIMS software.

In addition, there were highlights of personal achievement such as the announcement of David H. Bailey as the winner of the 1993 Sidney Fernbach Award for his algorithms in support of high performance computing applications; and the fact that three vendors exhibiting at SC'93--Thinking Machines Corp.,

nCUBE and SGI/Fluent Software, all showed applications using Horst Simon's spectral dissection algorithm.

SCinet'93, the NAS-designed network that was constructed on site, had two goals, according to Jim McCabe; one was to extend network performance capabilities beyond what had previously been attempted, and the other was to conduct various network-based experiments in the resource-hungry supercomputing conference environment.

Although networking was regarded in the past as an afterthought in supercomputing, McCabe said, SCinet'93 demonstrated that networking is now a "vital part of supercomputing systems".

Among the accomplishments of SCinet'93 were:

- First use of Asynchronous Transfer Mode (ATM) as a backbone network.
- Highest capacity metropolitan area network, using Synchronous Optical NETwork (SONET) OC-12 (622 megabits per second).
- Longest SONET OC-3 (155 megabits per second) link, from Portland to Albuquerque.
- Largest ATM wide area network ever built.
- Largest High Performance Parallel Interface (HiPPI) network ever built.
- First combination of ATM, Ethernet, FDDI, Fibre Channel, HiPPI, and SONET network technologies.

"In a normal environment, it would take six months to build something of this complexity, and we did it in a week," McCabe said, acknowledging the help of colleagues Nick Butzen, Jude George, Alfred Nothaft, Kelly Russell, and Leslie Schlecht.

Those presenting papers were:

- David H. Bailey (NAS Applied Research Branch), *RISC Microprocessors and Scientific Computing* and *Parallel Computing: Promise and Challenge*, an invited talk as recipient of the Fernbach Award. The *NAS Parallel Benchmarks Results* were presented in the latter talk, as well as in a November 19 panel presentation.
- Steve Barnard (Cray Research at NAS) and Horst Simon (NAS Applied Research Branch), *A Spectral Algorithm for Envelope Reduction of Sparse Matrices*. In addition, Simon and Subhash Saini (NAS Computational Services Branch) gave a poster presentation of performance

benchmarks for the week of the conference.

- Leo Dagum (NAS Applied Research Branch), *Automatic Partitioning of Unstructured Grids into Connected Components*.

- Sid Chatterjee, Rob Schrieber (both NAS Applied Research Branch), and John Gilbert (Xerox/PARC), *Mobile and Replicated Alignment for Data Parallel Languages*.

- Rob Van Der Wijngaart (NAS Applied Research Branch), *Efficient Implementation of a 3-d ADI Method on the iPSC/860*.

Leading workshops and tutorials, and chairing panels were:

- Bill Kramer (NAS Computational Services Branch), "Computational Services in the UNIX Supercomputing Center."

- Rob Schreiber was involved as presenter in two tutorials, "High Performance Fortran Standards" and "Parallel Linear Algebra Algorithms."

- Horst Simon, "A Guided Tour of High Performance Computing: Architecture, Software and Algorithm."

- Doreen Cheng, one of the speakers in the "Debaters for High Performance Computers" workshop, session chair for "Runtime Support for Parallelism," and moderator for "Heterogeneous Distributed Computing." In addition, Cheng and Robert Hood distributed two NAS documents, *A Protocol for Client Server Interaction in a Debugger* and *Facilitating a Portable Debugger for Parallel and Distributed Programs*.

Cheng believes the workshops with 50-60 conference participants were "very productive" because there were opportunities to make headway on issues of importance to both users and vendors. For example, in the workshop on debugging, vendors asked about waiting for language models to become stabilized before standardizing debuggers, and they were quite receptive to hearing about the approach used at NAS.

"We can't wait for that standardization of languages," Cheng said. "We define the protocol, the vendor provides the information, and we present the content in the way users want." She added that the advantages include greater flexibility and practicality. Users often complain that they get too much information and that it's not presented in a useful way, while vendors complain about constantly having to restructure their debuggers to increase portability.

The NAS booth, designed by John Hardman and located on the exhibit floor, was a busy place, according to Pat Elson, NAS User Interface Manager, who coordinated the NAS SC'93 effort. Aside from providing

technical handouts, the NAS booth ran three demos concurrently--FAST, NTV, and AIMS, which drew many conference participants browsing the exhibit floor:

- FAST (Flow Analysis Software Toolkit) is a multiprocessed environment for the visualization of computational fluid dynamics simulations; it was developed by a NAS team headed by Fergus Merritt. The demo of distributed FAST by Kevin McCabe showed one copy of the software and data at the NAS booth in Portland and another copy at the other end of the hookup at NAS headquarters at NASA Ames Research Center at Moffett Field; only commands were transmitted. What fascinated many observers was the use of "plain, vanilla" Internet--with no extra, special high bandwidth capability.

  An adjunct to this demo showed stereo glasses, CrystalEyes, a commercial product running in FAST to produce right and left eye views, which, when worn by users, create a 3d effect. A hardware controller alternates the active view.

- NTV (NAS Trace Visualizer) is a tool to visualize execution traces from MPP computers. Louis Lopez, who heads its development at NAS, demonstrated NTV, along with David DiNucci.

- AIMS (Automated Instrumentation and Monitoring System) instruments and monitors Fortran and C parallel programs: CMMD (on the CM-5), NX (on iPSC/860, iPSC/Delta, and Paragon), and PVM (on HP workstation clusters). Jerry Yan, of NASA Ames, leads the development team.

Other NASA staffers who assisted SC'93 behind the scenes include Lyz Baumiller, Chris Gong, Tony Lisotta and Serge Polevitsky.

Bailey Recieves IEEE Award

David H. Bailey of NAS was awarded the 1993 Sidney Fernbach Award at Supercomputing '93, in Portland OR, on November 18 for his algorithms in support of high performance computing applications.

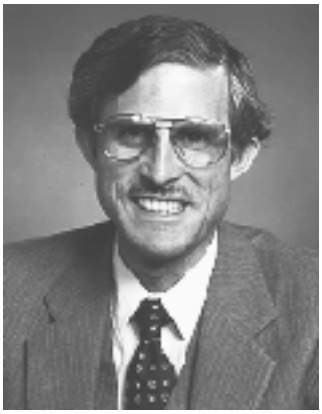
The annual Sidney Fernbach Award was established in 1992 by the Board of Governors of the IEEE Computer Society in honor of the late pioneer's work in the development and application of high-performance computers for the solution of large computational problems. Bailey is the first recipient.

The award, which consists of a certificate and a $2,000 honorarium, is presented to an individual for "an outstanding contribution in the application of high-performance computers using innovative approaches".

Bailey joined NAS in 1984. In 1986, he wrote a program to compute the mathematical constant pi to high precision for a new CRAY-2 supercomputer, using an algorithm discovered in 1985 by J. Borwein and P. Borwein of Canada. Although his CRAY-2 computations of pi were soon outdistanced by others, the software that Bailey developed for performing multiple-precision arithmetic is now used by scientists world-wide for research in both pure and applied mathematics.

Bailey discovered that one key to fast multiple-precision computation is the same as that important to many fluid dynamics computations: an efficient scheme to perform the fast Fourier transform (FFT) on advanced computer architectures. Techniques that he subsequently developed with Paul Swarztrauber of NCAR are now the basis of FFT routines available on supercomputers from CRAY and other vendors. In 1988, Bailey demonstrated that Strassen's algorithm, long considered only an academic curiosity, could be used to realize significant savings for practical matrix multiplication. Most computer vendors now provide Strassen-based matrix routines in their scientific libraries.

Supercomputing benchmarking is another area where Bailey has played an active role. In 1984, he developed the "NAS Kernels", later incorporated into the SPEC suite. In 1991, he was a member of the NAS Parallel Benchmarks (NPB) team. Bailey contributed to the wide distribution and acceptance of NPB as a standard through numerous lectures and seminars. Today, NPB performance results are widely cited in the parallel computing community. Bailey's essay, *Twelve Ways to Fool the Masses*, is often mentioned in scientific discussions of questionable parallel performance reports.



Bailey received his B.S. in mathematics at Brigham Young University and his Ph.D. in mathematics from Stanford University. He has been active on the program committees of supercomputing conferences for several years and has served on the editorial boards of the *IEEE Transactions on Parallel and Distributed Computing*, the *Journal of Supercomputing*, and the *International Journal of High-Speed Computing*. He presently serves on the board of directors of ACM SIGNUM.

NAS Parallel Benchmarks Results Available

Updated results for the NAS Parallel Benchmarks, developed at NASA Ames Research Center to study the performance of parallel supercomputers, were published in October.

Details of the eight benchmark problems to be solved are given in the report, and discussion of the results is included.

Performance results for several systems, including previously unpublished results for the Intel Paragon and new results using Parallel Virtual Machine (PVM, as well as new or updated system results for Thinking Machines CM-5, IBM SP-1, CRAY Y-MP C90, and CRAY T3D represent the best results reported to NAS for the specific systems listed. In several instances, the results are improvements in both compilers and implementations.

The *NAS Parallel Benchmarks Results* report is updated when there are sufficient new results to publish, usually twice a year.

The results are reported in *NAS Parallel Benchmark Results 10-93*, Technical Report RNR-93-016. To request a copy send email to the NAS Documentation Center.

Sample Fortran programs implementing the NAS Parallel Benchmarks are also available. The sample codes are provided on Macintosh floppy disks and contain the Fortran source codes, README files, input data files, and reference output data files for correct implementation of the benchmark problems. These codes have been validated on several computer systems ranging from conventional workstations to supercomputers. To obtain the sample program disks, write to: NAS System Division, Mail Stop 258-6, NASA Ames Research Center, Moffett Field, CA 94035-1000, Attn: NAS Parallel Benchmark Codes.

January - February 1994
Vol. 2, No. 1

Executive Editor: Marisa Chancellor

Editor: Jill Dunbar

Senior Writer: Elisabeth Wechsler

Contributing Writers: Jean Clucas, Subhash Saini, Pamela Walatka

Other Contributors: David Bailey, Doreen Cheng, Leo Dagum, Chris Gong, Pat Elson, Steve Harding, Marcia Hartman, Bill Kramer, Mark Kremenetsky, Jim McCabe, Marcia Redmond, Kelly Russell, Subhash Saini, Horst Simon

Editorial Board: Marisa Chancellor, Jill Dunbar, Chris Gong, Creon Levit, Serge Polevitzky, Pamela Walatka, Elisabeth Wechsler, Rita Williams