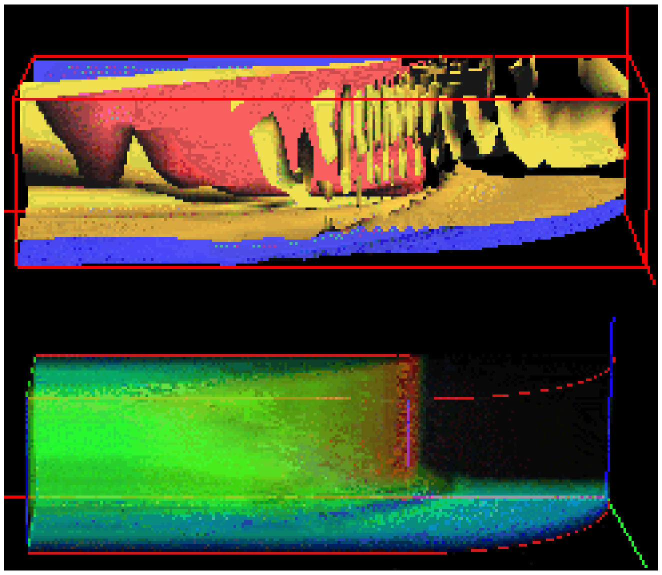


# March-April 1996, Vol. 2, No. 16

# CRAY J90 Cluster To Help C90 Users "Parallelize" Codes

*By Elisabeth Wechsler*

Helping NAS C90 users "parallelize" their vector production codes -- cost effectively and in a familiar computing environment -- is the goal of the latest parallel cluster workstation project in the NAS Systems Division. Delivery of four J90 systems, manufactured by Cray Research Inc., was expected at NAS last month, and represents the first phase of the Newton project.

"Fortran applications representative of the vectorized NAS CRAY C90 workload show that the J90 systems are more cost effective than any workstation out there, including those from IBM and SGI [Silicon Graphics Inc.]," said Bob Ciotti, project lead, adding that the four J90 systems together offer equivalent performance of five C90 CPUs.

The initial J90 configuration is intended to serve as proof-of-concept and to identify problem areas in internodal communication and scheduling issues specific to parallel computing. Each baseline system will consist of 4 CPUs, 1 gigabyte (GB) of main memory, two network interfaces, and 36 GB of disk storage. One J90 workstation will have additional processors and disk capacity to perform tasks such as compilation.

## Time Saved Recoding Applications

Ciotti believes that C90 vectorized Fortran codes will run unmodified very efficiently on the J90. "The challenge will be to maintain the multitasked vector efficiency while adding in the communication mechanisms required to parallelize the code over multiple distributed memory systems. We really don't know how well this will work, compared with the effort required to recode applications -- traditionally, the most significant obstacle to parallel computing."

"This system will provide users with a transitional system to develop message-passing techniques in a UNICOS environment," said George Myers, NAS scientific consultant. "For C90 users, the advantages of the J90 system over other clusters are likely to be a more stable operating system environment and familiar code development tools." In addition to the tools that Cray provides, the Newton project will incorporate such NAS-developed tools as PBS (the Portable Batch System) and p2d2, a parallel debugger.

Over the next several weeks, proposals will be solicited from current NAS C90 users and evaluated based on their CFD applications' potential for being parallelized. A limited number of those proposals, representing different NASA centers, will be accepted this spring. Access to Newton is being intentionally restricted so that a few applications developers can make progress quickly with a larger portion of the resource (all four Newton machines will be used for a single problem). "There will be less emphasis on eliminating idle time and more emphasis on getting good response and turnaround," Ciotti said.

## Higher Bandwidth, Lower Latency

According to Ciotti, the "really interesting configuration" will be available with an upgrade scheduled for delivery in the third quarter of this year. "The new hardware will support I/O bandwidth of over 1 gigabyte per second and latency of less than 100 nanoseconds between machines," he said, adding that these features will give the upgraded systems "great potential for running large parallel applications effectively."

If the systems perform well, there is a possibility that each could be upgraded to 16 or even 32 processors. Four 32-processor J90s have an approximate capacity of 8 gigaflops, or about twice that of vonneumann (the NAS CRAY C90), he explained.

To put these statistics in context, Ciotti observed that "the current crop of workstations were designed for the $75 billion workstation market, not the $1 billion supercomputing market." As a result, he believes that workstation clusters have "serious architectural weaknesses in the areas of memory bandwidth and I/O capability when running supercomputing applications."

## Scalar Performance a "Weakness"

However, Ciotti acknowledged that scalar performance is "a weakness in the J90 architecture," compared with IBM 590 or SGI R8000 workstations. "For applications that don't vectorize well or are inherently scalar, better performance is likely on babbage [IBM SP2] or davinci [SGI Power Challenge cluster]." While the upgraded system will have increased scalar performance, he explained that other factors such as memory bank busy time, bank conflict, and latency also have a significant effect on performance. "All we know is that the upgraded architecture will be better for scalar codes and about the same for the vector codes from vonneumann."

For users coding applications or running existing applications, vectorization is relatively straightforward, Ciotti continued. Cache-based systems require developers to worry about locality of reference as it relates to the first and second level, size, and line width of the cache. More involved algorithms are required to work around this limitation in memory bandwidth, he explained. "With the Newton cluster, we can avoid this step and concentrate solely on distributing the application across multiple systems. Hopefully, this will result in increased productivity in the development cycle of parallel applications."

## File Sharing Is Faster

Another unique feature of the Newton system is its ability to share files directly between systems much faster than NFS (Network File System) or DFS (Distributed File System). This feature, Shared File System, may help developers of parallel applications do I/O more efficiently, currently a general problem in parallel computing, Ciotti explained.

"By the end of this project we hope to have a road map for how to move production code from a high-speed processor environment to a parallel environment. We think the experience we've gained so far in other parallel environments at NAS will help us avoid some bumps in the road," Myers said.

Next Article | Contents | Main Menu | NAS Home

# Reconfiguration Increases davinci's CPU Utilization

*By Elisabeth Wechsler*

What began as a temporary reconfiguration of CPUs for davinci, the NAS workstation cluster testbed, has become a longer-term change that emphasizes multiprocessor node utilization. The original reason for the change was to run calculations for the NAS Parallel Benchmarks (NPB) on the Silicon Graphics Inc. (SGI) Power Challenge array portion of the cluster.

This first change came in October when the NAS high-speed processing group needed a 9-CPU node to run a perfect squares problem for the NPB 2.0 test suite, according to James Jones, who until recently was the lead system administrator for davinci. Because the SGI multiprocessor CPUs come only in multiples of two, it was necessary to create a 10-CPU node instead. To accomplish the reconfiguration, a 2-CPU board was removed from davinci-05, one of the testbed's eight nodes, and reinstalled in davinci-08.

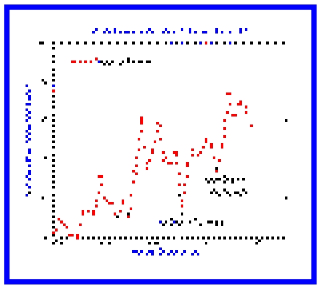
After the first reconfiguration, several other users found it useful to scale code across ten CPUs instead of eight in order to solve bigger problems, Jones explained, so it was decided to leave the alteration in place for the time being.

While helping NAS researchers prepare demonstrations for Supercomputing '95, a second temporary configuration change was made in November. Two 75-MHz, 2-CPU boards were added to davinci-05, as the testbed staff wanted to explore the effects on applications of mixing 75-MHz and 90-MHz CPUs within the same system. Since these extra four CPUs were intended to be part of a computer system on order, it was decided to "loan" the CPUs to davinci.

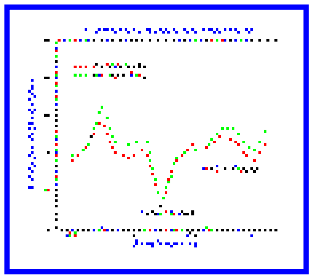
Before the reconfiguration, davinci had four multiprocessor nodes with eight 90-MHz CPUs each. Now, davinci-05 has ten CPUs (four 75-MHz, and six 90- MHz CPUs) while davinci-08 has ten 90-MHz CPUs.

According to Jones and current davinci lead system administrator Archemedes deGuzman, three important accomplishments have been achieved for the testbed: The reconfiguration has demonstrated the system's flexibility to move hardware for specific needs and to experiment with alternative configurations. It has also provided the NAS parallel systems staff with an opportunity to analyze usage patterns and application performance on a machine with mixed CPU speeds. Finally, the reconfiguration has offered a basis for comparison in determining the optimum long-term cluster configuration for davinci's 190 users, of which some 40 are considered active.

"We found, as predicted, that applications running on the mixed 75-MHz / 90-MHz CPU node experience a performance degradation -- effectively running at 75 MHz," Jones acknowledged. "Yet, even after warning users of this result, we see that this node is still a highly contended resource. We want to know if users are attempting to compensate for the performance hit by running their code across more CPUs, or if the 2 gigabytes of node memory is the more important factor."



To answer these types of questions, Jones and deGuzman began testing in January (see Figures 1 and 2) to determine how many jobs are fully utilizing the computing power of the multiprocessor nodes.



"Our gut feeling is that more users employ the large nodes for the memory resources rather than for the multiple CPUs," Jones said. "After analyzing the data currently being collected, we can determine whether to reconfigure the cluster to provide for a more evenly distributed number of CPUs -- or perhaps memory."

Jones emphasized that the primary goal is to support internodal (message-passing) parallel applications. "The cluster will become more attractive to parallel users once we receive and install SGI's low-latency HiPPI software, which will allow much faster communications between nodes."

Next Article | Contents | Main Menu | NAS Home

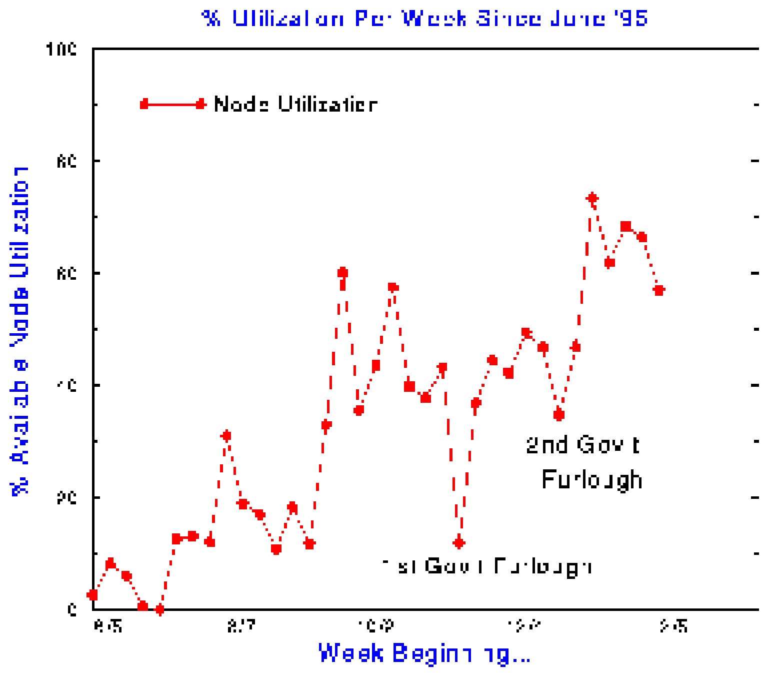


**Figure 1** shows that davinci's utilization (the percentage of node hours used, divided by the number of node hours available) has been slowly increasing since October, when the first CPU reconfiguration was implemented. In addition, davinci's utilization has remained, on average, over 20 percent higher during the same period. As the NAS parallel systems group continues to refine davinci's configuration to meet users' needs while also increasing overall system utilization, various data are being collected and analyzed. Such data has supported the staff observation that the four single-CPU nodes were used less than the multiprocessing nodes.

---

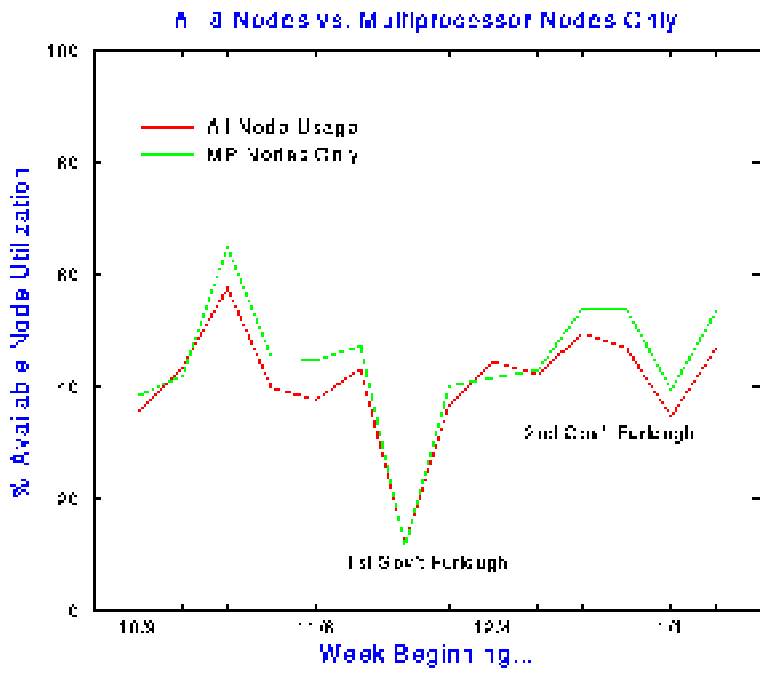


**Figure 2** shows that the overall cluster utilization would, in fact, increase if the four single-CPU nodes were omitted.

[to the article](#)

# MPI-IO: NAS Helps Define An Emerging Standard for File Access on Parallel Systems

*By Bill Nitzberg*

NAS, along with researchers and engineers from IBM, Lawrence Livermore National Laboratory (LLNL), and over 100 other organizations in 18 countries, is defining MPI-IO, a new standard interface for parallel file operations. MPI-IO is an extension to the popular MPI (Message Passing Interface) standard. Together, they will enable truly portable programs to be written for parallel machines. NAS has implemented a generic portable version of MPI-IO, called PMPIO. This prototype achieved a 20-fold speedup in writing files on the IBM SP2.

## Performance and Portability "Inseparable"

Existing parallel I/O systems evolved directly from I/O systems for serial machines, which are heavily tuned for sequential, large accesses, and limited file sharing between processes. In a parallel environment, none of these assumptions hold. Further, in the scientific world, a program is not considered truly portable unless it not only compiles but also runs efficiently -- thus, portability and performance are inseparable.

This leaves programmers with two equally bad choices: use a standard file I/O interface (UNIX) and cross their fingers, hoping for good performance, or hand code intricate I/O optimizations for every parallel system.

The UNIX interface, and sequential file I/O interfaces in general, lead parallel programmers to specify file operations as an uncoordinated series of seek and read/write operations. This "naive" method for performing parallel I/O can lead to serious load imbalance, poorly ordered accesses, and significant contention for the network, caches, disks, and other system resources.

In addition, common data distribution patterns result in small accesses. For example, a three-dimensional distribution of a 1-million element array (100 x 100 x 100) onto 64 processesr (4x4x4) yields file accesses of only 25 elements long. Small accesses are particularly bad for performance from the perspective of the network and disks because they translate into small messages on the network, where performance will be dominated by network latency and will never approach the full bandwidth. Disks are notoriously inefficient when accessing less than a full disk block (due to costly read-modify-write cycles and inefficient caching).

## Collective I/O is Key

MPI-IO has the potential to solve both performance and portability problems. It will provide a standard interface for file operations in both the C language and Fortran. The key to performance is exploiting the highly synchronized nature of scientific programs though collective I/O. The uncoordinated nature and small size of the I/O operations in parallel applications is not inherent to the application, nor is it inherent to parallel I/O -- instead, it is an artifact of a poor file I/O interface.

A collective interface allows the programmer to specify a file access as a high-level global operation (for example, "save array X") rather than as a series of seeks and writes. This provides a more natural interface, as well as the ideal framework for optimizing the data transfer. In a collective operation, all processes are synchronized and ready to perform file accesses, and the entire operation can be orchestrated using the globally shared knowledge of the data distribution, file layout, and physical system characteristics.

## Collective Buffering Ups Performance

The collective interface in MPI-IO permits significant optimization to be performed with tools such as the NAS-developed "collective buffering" algorithm. Collective buffering is an extension of the file buffering provided by sequential operating systems, where the data is permuted before writing (or after reading) in order to collect and combine small file accesses into large blocks. This technique was demonstrated at the NAS booth during Supercomputing '95, and delivered a 20-times improvement in parallel I/O performance.

MPI-IO implementations are currently being developed and tested on the IBM SP2, Intel Paragon, CRAY T3D, Meiko CS-2, and Silicon Graphics Inc. (SGI) systems by groups at NAS, IBM, and LLNL.

NAS's PMPIO runs on all systems that can run the MPICH MPI library (developed at Argonne National Laboratory and Mississippi State University), including the IBM SP2 and Intel Paragon, as well as workstation clusters from IBM, SGI, and Sun Microsystems). The initial version of PMPIO implements a subset of the MPI-IO interface, including basic file operations, collective operations, and support for arbitrary data distributions. PMPIO runs with both C and Fortran on standard UNIX-based file systems or UNIX-based parallel file systems. PMPIO will be available for beta-testing in early April.

To find out about becoming a beta tester, send email to **nitzberg@nas.nasa.gov**.

Here's more information on MPI-IO.

*In the May-June '96 issue of* NAS News, *Sam Fineberg, of NAS's parallel systems group, will cover the basics of the MPI-IO interface.*

Next Article Contents Main Menu NAS Home

# Getting Familiar With PBS on the CRAY C90s

*By Terry Nelson*

The Portable Batch System (PBS) will be implemented soon on both CRAY C90 supercomputers housed in the NAS Facility -- current plans call for availability to users in April. PBS is designed to provide a seamless computing environment for job submittal and control across a heterogeneous collection of computing resources. A parallel version has been in production on the IBM SP2 and workstation cluster testbeds at NAS since early 1995. *For more background information on PBS, see* NAS News *May-June 1995.*

Cray users will need to become familiar with PBS commands and script conversion methods in order to run their jobs on these systems. This effort should be minimal, since PBS is POSIX 1003.2d compliant, and is close to the current Network Queuing System (NQS) in structure. This article highlights some differences between PBS and NQS.

## Scripts Showing Command Comparisons

The following is a sample script in PBS for the Cray C90s. The analogous NQS commands follow each line in parentheses. PBS reads the .cshrc (csh) or .profile (sh) files and starts out in $HOME, just as NQS does.

```
% cat script1
#PBS -S /bin/csh
#  (QSUB -s /bin/csh)
#PBS -lcput=1000
#  (QSUB -lT 1000)
#PBS -lmem=10Mw
#  (QSUB -lM 10Mw)
#PBS -l srfs_big=150mw
#  (QSUB -lr $BIGDIR,150Mw)
#PBS -l srfs_fast=5mw
#  (QSUB -lr $FASTDIR,5Mw)
cd $HOME/exampldir
cp inp.file $BIGDIR
```

```
cd $BIGDIR
...
$HOME/exampldir/bigbin < inp.file
...
# End of PBS script1
```

Jobs are submitted with the command *qsub* (as with NQS) in the above case, *qsub* script1.

A second example contains comments to show some additional PBS features:

```
% cat script2
# Time in PBS can be expressed in
# hours:minutes:seconds
#PBS -lcput=1:5:10
#PBS     -lmem=10Mw
# One can combine separate requests on the
# same line
# Either Mw or mw is acceptable
#PBS -l srfs_big=100Mw,srfs_fast=5mw
# There are a number of handy PBS
# environment variables.
# Here PBS_O_WORKDIR refers to the location
# from which the job
#  was submitted.
cd $PBS_O_WORKDIR
...
# End of PBS script2
```

Attributes can be included on the *qsub* line, in which case they take precedence over similar attributes in the actual script. For example:

```
qsub -lcput=30:00,mem=8mw script2
```

submits script2, but with limits of 30 minutes and 8 megawords (MW) of memory.

## Critical Differences in Process Limits

The distinction between job and process limits exists in PBS but with a difference. In NQS, this distinction is expressed by -lT and -lM (job) or -lt and -lm (process). In PBS, process limits are expressed by the pcput and pmem attributes and job limits are expressed by cput (job CPU time) and mem (job memory). There is, however a crucial difference between NQS and PBS usage.

In NQS, scripts that provide only process attributes have those values used for the job limits, as well. This is not true in PBS. Instead, such jobs are limited to 300 seconds and 4 MW of memory -- regardless of the values of pcput or pmem in the script. Users must provide the cput and mem attributes, whether process limits are provided or not. Type *man pbs_resources* for a detailed explanation of PBS attributes and their formats.

## Linking Jobs

One useful new feature in PBS is the ability to link jobs in a variety of dependence relationships. For example, to run job scr3 only if jobs scr1 and scr2 complete successfully, use the -W attribute, as follows:

```
> qsub scr1
2463.vn.nas.nasa.gov
> qsub scr2
2466.vn.nas.nasa.gov
>
```

When you type *qsub*, PBS returns the full job identifier -- not just the number (for example, 2463). The whole identifier is required to do the next submit, such as:

```
> qsub -Wdepend=afterok:2463.vn.nas.nasa.gov:2466.vn.nas.nasa.gov scr3
2471.vn.nas.nasa.gov
>
```

After both jobs 2463 and 2466 complete successfully, PBS will automatically queue job 2471. If jobs 2463 and 2466 have already completed, using -Wdepend=afterok will result in an error. In this case, it would not be needed anyway since the two preceding jobs had already run. Type *man qsub* for the full range of possible dependencies between jobs.

## Translate Scripts In Advance

Transforming scripts from NQS to PBS is primarily a mechanical task. A script (called nqs2pbs) is provided to accomplish this task, so that users have some lead time to prepare scripts. Simply provide the old and new filenames, as in:

```
nqs2pbs scr1nqs scr1pbs
```

This script translates the NQS command, then places it on the line after the new PBS command. This has two purposes: First, it shows very clearly the differences between the two command styles. Second, the new script can currently be used for NQS submission (since #PBS is a comment to NQS). After PBS is in production, it can still be used (since #QSUB is a comment to PBS).

## Job Monitoring Command

The main PBS command to monitor jobs is called qstat. A small sample display in the POSIX standard form is shown below.

```
Job id          Name    User            Time Use        S       Queue
---------------------------------------------------------------------

2463.vn         pcr4    usrnam1         0               H       pending
2464.vn         pcr5    usrnam2         00:00:01        R       batch
```

Type *man qstat* for explanations of the table headings and other available options.

## Additional Commands for Job Control

PBS offers several other commands to help users control their jobs. These include:

- *qdel* - delete a batch job from the queue OR from execution (There is no extra parameter for an executing job, unlike in NQS.)

- *qselect* - list job identifiers by a list of selection criteria

Both commands have man pages to explain their use and syntax. The SEE ALSO section at the end of each man page mentions others commands that may be of interest.

## For More Information

Several sources of documentation on PBS, as implemented on the CRAY C90s at the NAS Facility, are available.

- a PBS page on the World Wide Web.
- Several man pages on PBS, the most useful of which include:
  *man pbs_resources*, *man qstat*, *man qsub*, and *man qdel*

Next Article | Contents | Main Menu | NAS Home

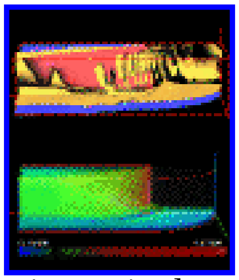
# Methods for Direct Volume Visualization of Curvilinear Grids Developed Under NAS Grant

*by Jane Wilhelms & Allen Van Gelder*

Much of the visualization of data from computational fluid dynamics concentrates on imaging vector data as, for example, particle traces and flow lines. Visualization of scalar data, such as pressure, temperature, energy, and helicity, is also of interest, and presents unique challenges.

## Visualizing Scalar Data

There are two dominant methods of visualizing three-dimensional scalar data: isosurfaces and direct volume rendering. Isosurfaces, the older method, involve finding two-dimensional surfaces of a given value within the three- dimensional volume. These surfaces are often extracted as a mesh of polygons, and can be rapidly drawn on graphics workstations. Multiple surfaces can be shown in one image, each with its own threshold, and differentiated by color or other properties. Isosurfaces clearly present information on data at the threshold values, but cannot show data between these values.



The top of Figure 1 shows the NASA Ames blunt fin dataset drawn with three isosurfaces. The image was rendered using a hierarchical isosurface program (called *oct*) developed by the UCSC team. The blunt fin is a small curvilinear dataset of resolution 40x32x32. The isosurfaces shown take a fraction of a second elapsed time to extract on a Silicon Graphics Inc. (SGI) Reality Engine II high-end graphics workstation. Once extracted as polygons, they can be animated at a rate of about five frames per second, using a single processor.

An alternate approach is direct volume rendering. This method treats the volume as a semi-transparent, colored medium in which all regions may potentially contribute to the final image. Scalar data values are mapped to values of color and opacity using a transfer function that is normally under the user's control. Direct volume rendering provides a very general tool for visualization, but usually takes longer to create images -- and sometimes the sheer amount of information presented in a single image can be confusing.

The bottom of Figure 1 shows the blunt fin dataset as a direct volume rendering. The mapping of data to color is shown as a band below the image; note that the free stream is mapped as clear and does not appear. (The mapping to opacity is not shown on the bar.) Creating this image took about 16 seconds elapsed time on the Reality Engine using another method developed at UCSC (called *qp*), which takes advantage of graphics hardware. Although the image must be re-created each time the volume is rotated, the image can be redrawn in about one second if only the transfer function, translation, or scaling is changed.

## Challenges of Non-rectilinear Grids

Isosurface extraction and direct volume rendering were first developed for regular rectilinear grids. The UCSC team is actively working to extend these algorithms and develop new approaches for visualizing curvilinear grids, multizone curvilinear grids, and other irregular grids.

These grids present interesting challenges for several reasons, for example: the complex shapes of cell regions defined by grid points; the wide variation in the sizes of cells in different regions of the grid; and the intersecting nature of multi-grids. While it was fairly easy to extend standard isosurface methods for curvilinear grids, direct volume rendering -- particularly of multigrids -- has proven to be more of a challenge.

## General Volume Rendering Method

UCSC's *qp* program,which created the bottom image in Figure 1, works well on single grids -- though there is some sacrifice of image quality in exchange for speed. Also, *qp* cannot handle multiple intersecting grids, such as the space shuttle, which consists of nine intersecting grids with nearly a million data points.

To deal with multigrids, other types of irregular grids, and inclusion of polygonal surface data in the image together with volume data, the team developed a general volume rendering method in which the faces that connect data points are treated as independent polygons. These polygons are sorted by screen region and distance from the user. During the drawing phase, the polygons at each screen pixel are drawn front-to-back, taking into account their color and opacity (mapped from data values using the transfer function) and the distance between polygons. A unique feature of this method is that polygons are inserted into a spatial hierarchy -- so that polygons occupying the same region of space are grouped and can be ignored as a group if not visible.

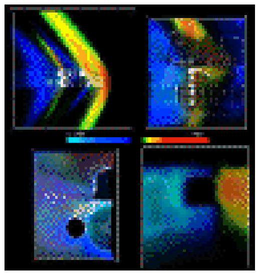


Figure 2 shows several images of the nine grids of the space shuttle using this direct volume rendering program, called *scremun*. The color bar indicates the mapping of data values to color (the free stream is again clear). These images (500 x 500 pixels in size) take between 15 and 90 seconds elapsed time using four processors on the Reality Engine. (In comparison, the blunt fin dataset takes about 22 seconds for images such as those in Figure 1, using *scremun* with one processor.) The rendering is done in software and does not require a graphics workstation, and it can be runin parallel on multiple processors -- in this case, on a multiprocessor SGI workstation.

## Hierarchies for Visualizing Large Datasets

Datasets are often so large that visualizing them takes an "uncomfortable" amount of time. UCSC's solution to this problem has been to create a hierarchy over the data that provides rapid access to regions of interest, as well as intelligent summary information for general understanding of the data's content. These hierarchies can also point back to the original data, so accuracy need not be lost.

Hierarchies -- a form of data compression -- can handle data of multiple dimensions, irregular grids, and varying data representations. Visualization algorithms can calculate images from different levels of the hierarchy, providing rapid rough estimates. Whole subtrees that are not visible can be ignored during rendering, making it much faster. Error metrics can indicate the closeness of estimated data values to actual ones.

UCSC has implemented error-controlled hierarchies over regular grids that offer these features. Currently, these hierarchies over irregular datasets are only used to limit traversal to visible regions. The team is presently extending this method to include models of the data at varying levels of resolution.

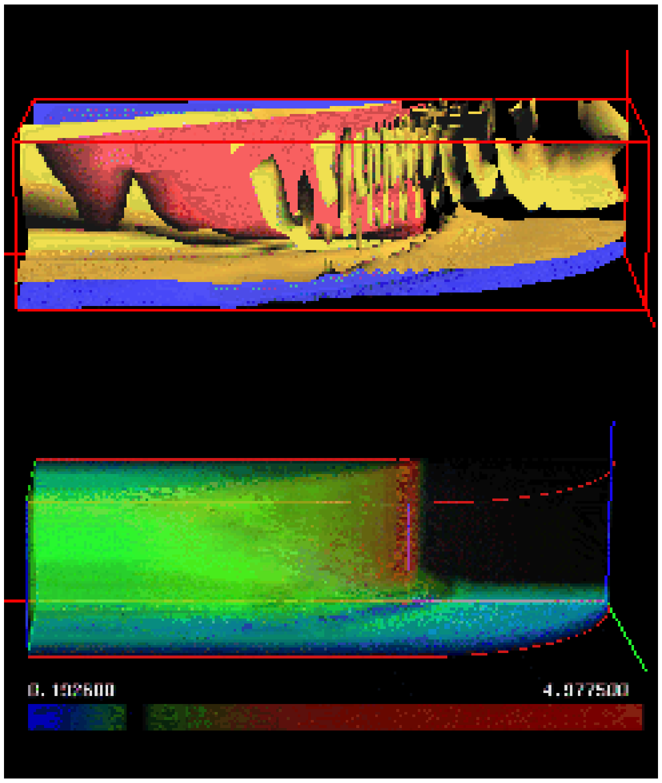
For more information on UCSC's visualization work, send email to **wilhelms@cse.ucsc.edu**. Here's more about NAS's scientific visualization work and more on UCSC's visualization work and other research work.

---

*Former team members -- Naim Alper, Judy Challinger, Tom Goodman, Andy John, Shankar Ramamoorthy, and Orion Wilson -- have graduated from UCSC.*

Next Article | Contents | Main Menu | NAS Home

**Figure 1**. The NASA Ames blunt fin dataset visualized using isosurfaces on the top and direct volume rendering below. The color bar indicates the mapping of data values to color in the bottom image. The top image shows threshold values 1.5 (reddish), 1.0 (gold), and 0.7 (purple). These images were produced using methods developed at the University of California, Santa Cruz, under a NASA grant funded by NAS.



to the article

# Using PIOFS Efficiently on the IBM SP2

by Mark R. Smith

IBM's Parallel I/O Filesystem (PIOFS) has undergone testing on the NAS Facility's SP2 test nodes for several months. Currently, PIOFS is being enabled by special request for interested users. Sample programs located on the SP2 in /usr/contrib/piofs/ provide examples that run faster using PIOFS than alternative SP2 filesystems.

Here are some tips to ease the transition from the NFS-mounted shared filesystem (/scratch1) to the PIOFS filesystem, /piofs1.

- You can access /piofs1 from the SP2 nodes only -- not from the babbage1 and babbage2 front-end systems.
- Use large [1 megabyte (MB) or greater] blocks in I/O to and from PIOFS (serial I/O from one reader/writer).
- Use file I/O from within your source code to stage files to and from PIOFS and minimize use of AIX file commands.
- Take advantage of the data checkpointing feature to keep track of large data filesets, in the event that an error occurs during program execution.

The examples cited below have been tested, and source code is provided in /usr/contrib/piofs/. One feature of this filesystem, called "parallel views," provides support for multiple writers into one logical PIOFS file. For detailed information on parallel views and other features, see the *IBM AIX Parallel I/O Filesystem* guide (number SH34-6065), available through the NAS Documentation Center.

The NAS parallel systems staff are still evaluating the effectiveness of PIOFS, and will report information on measured performance in a future issue of *NAS News*.

PIOFS appears as a generic AIX filesystem to codes running on the SP2 nodes. More than 20 gigabytes (GB) of high-performance IBM 9333 disks are allocated to PIOFS and are available to the NAS SP2 user community. These disks are in a pool that exhibits peak read and write performance more than an order-of-magnitude greater than the NFS-mounted filesystem. NAS and IBM staff members are working to increase both the size and performance of the PIOFS partition.

Although PIOFS can be viewed and manipulated as an AIX filesystem, typical AIX file manipulation commands such as *cp, mv, dd,* and *tar* use small block sizes that result in poor performance. NAS and IBM staff are working to optimize the standard AIX file management commands for PIOFS. Until then, use file I/O with large blocks inside a Fortran or C program to incur less overhead. The following pseudo-code strategy for staging files is recommended:

```
c pseudo-code for PIOFS file I/0
  program user_code
  character*8 piofsfile, infile
  integer     io_process
     ...
  if(io_process)
     stage_to_piofs(infile, piofsfile)
     ...
c Real program work goes here
     ...
  if(io_process)
     stage_from_piofs(piofsfile, outfile)
     ...
  stop
  end

c stage_from_piofs follows similar logic to
  stage_to_piofs
  subroutine stage_to_piofs(infile, piofsfile)
  parameter(inunit=12, piofsunit=13,
  CHUNK=262144)
  character   infile(64),
  piofsfile(64)

  real*8      data(TOTAL)
  integer     low_index, high_index
     ...
  open(inunit, file=infile)
  open(piofsunit, file=piofsfile)
     ...
  blocks=(Compute_Number_CHUNKS())
     ...
c Read chunks of the input file and stage to
  the PIOFS file
  do index=1, blocks
      low_index  = (index-1)*CHUNK + 1
      high_index = index*CHUNK
      read(inunit) data(low_index,
  high_index)
      write(piofsunit) data(low_index,
  high_index)
  enddo
  close(inunit)
  close(piofsunit)
  return

c end PIOFS example pseudo-code
```

While this example glosses over important detail in real program I/O, it does illustrate a stage-in function. Complete Fortran and C sample codes with Makefiles are located in /usr/contrib/piofs/stage2piofs/, along with detailed stage-in and stage-out functions. Since each of your data files has data structures that are specific to your applications, these are generic working source code implementations that you can adapt to your application on the SP2.

The sample codes use a simple REAL*8 array as the data structure staged to and from another filesystem accessible on SP2 nodes in the programs "piofsdemo.f" and "piofsdemo.c".

## Large Blocks for Better Performance

As noted above, using large blocks in Fortran, C, (or C++) I/O will lead to far greater performance. Block sizes of 2 MB or larger work well and achieve as much as 8 MB/sec when writing to the filesystem -- much greater than the peak of just over 1 MB/sec achieved when writing to the /home filesystems or the approximately 2.5 MB/sec when writing to nonshared local disk (/tmp) on each node.

Another feature available with PIOFS is data checkpointing. PIOFS can track the state of a PIOFS file at any point after a READ or WRITE operation has completed. Sample source codes in /usr/contrib/piofs/checkpt/ illustrate this feature. Checkpointing is implemented in the following pseudo-code:

```
c checkpt.f - checkpointing pseudo-code
  program checkpt
  real*8    data(MAX)
  integer*4 piofsunit, rc, ios
  character piofsfile(64)
c include file for PIOFS file chkpt
  include "piofs/piofs.inc"
c obtain piofs filename
  open(unit=piofsunit, iostat=ios,\
     file=piofsfile,...)
     ...
c perform writes to PIOFS file
     ...
c checkpoint the file
  rc = piofschkpt(piofsunit)
  if(rc.ne.0) call error(rc) ! handle error
     ...
c do more work on the file, to recover:
  call flush_(piofsunit)
  rc = piofsrchkpt(piofsunit)
  if(rc.ne.0) call error(rc) !
c successful restore replaces file with
  checkpointed data
     ...
  stop
  end
c end PIOFS example pseudo-code
```

To take full advantage of PIOFS, use the data partitioning scheme detailed in IBM's PIOFS documentation. PIOFS relies on setting up "views" into the data consisting of a number of subfiles. NAS is evaluating the efficacy of this paradigm versus simply using large blocks on single I/O streams within code. In performance tests on the current configuration, with simple demonstration codes, using parallel views increased peak write performance nearly proportional to the number of PIOFS servers (currently, three). When these servers are in heavy use, performance gains are marginal.

As the staff gathers more information on the performance and stability of PIOFS features, results and tips will be posted. Messages of the Day will also be posted with new information.

If you're interested in using PIOFS, contact the NAS parallel systems group by sending email to **nashelp@nas.nasa.gov**.

# Quick Tips for Getting Accurate Wall-clock Timings on the IBM SP2

*by Mark R. Smith*

When code timing was discussed in the November-December '95 issue of *NAS News*, the routine *rtc* was cited. Though accurate, this routine uses an unsupported library and is not portable to all other AIX systems. Here's a method for getting accurate, sub-microsecond resolution wall-clock timings on any AIX system.

IBM documentation refers to a more portable AIX timing framework, which will also return the precision of the measurement using the RS/6000 Real Time Clock registers. Sample C and Fortran codes using these routines are provided on the SP2 in /usr/contrib/timing/ and require simple relinking of your code to pick up this updated version of *rtc()* in /usr/contrib/timing/libutil.a. The only change is the use of the routines *read_real_time()* and *time_base_to_time()* which are in the system library /usr/lib/libc.a.

To get a breakdown of run-time performance by routine, *prof* and *gprof* have been modified to work with the AIX Parallel Environment (MPI and MPL) on the IBM SP family. The *-p* and *-q* compile flags cause files "mon.out.N" (prof) or "gmon.out.N" (gprof) to be created at run-time, where N is the task id (one for each node) that is summarized and reported.

To use profiling:

- Add the *-p* (for *prof*) or *-pg* (for *gprof*) flags to your compile step; for example:

  ```
  mpxlf -O -pg -g test.f -o test
  ```

  will generate an executable "test" from source file test.f that will generate mon.out.N files when run.

- Run the code, whether from an interactive or batch PBS session. Make certain that the working directory is a shared filesystem. (Your home filesystem is fine for profile traces; for large trace files use /scratch1.)

- Reduce the files by using the *prof* or *gprof* command, followed by:

  ```
  prof test >prof.outfile
  ```

or

```
gprof test > gprof.outfile
```

from babbage1 or babbage2. Note that the *prof* and *gprof* commands run only from babbage1 or babbage2.

For more information on timing and performance measurements on the SP2, contact Steve Heistand at **heistand@nas.nasa.gov**.

Next Article | Contents | Main Menu | NAS Home

# NAS to Enhance Remote Communications

*By Marcia E. Redmond*

How would you like to attend a NAS class or meeting without ever leaving your office, or put on a pair of earphones and listen to parts of these events while working on other tasks? The NAS local area network (LAN) group is currently installing and testing router software that makes multicasting -- the ability of one or more computers to view and listen to the same information from a remote host -- possible.

Investigation of the multicasting capability began a year ago but was shelved until November, when software was released that would allow existing routers to support the concept efficiently. Since then, LAN group members Bryan Juliano and Kevin Lahey have been working to make multicasting more widely available.

Currently, several users are downloading new versions of multicasting applications, including the MBone (Multitasking Backbone), to NAS-supported fileservers for testing. Juliano pointed out that good production-quality releases are scarce since the software is new and changing rapidly.

For now, the group is "concentrating efforts on getting multicasting to a reliable working state so that we can begin using it as soon as possible," Juliano said, adding that testing and troubleshooting were to be completed in February.

For more information about multicasting at NAS, send email to **juliano@nas.nasa.gov**, and watch for updates in future issues of *NAS News*.

Next Article | Contents | Main Menu | NAS Home

# International Virtual Reality Conference to Focus on Applications, Real-world Problems

VRAIS '96

*By John Hardman*

Researchers from around the world will gather in Silicon Valley at the end of this month for the IEEE Virtual Reality Annual International Symposium ( VRAIS '96). The conference, to be held March 30-April 3 at the Santa Clara Marriott, will feature participants from the U.S., U.K., Austria, Canada, France, Japan, Switzerland, and other countries. This year's conference will reflect the maturation of VR technology by highlighting virtual reality applications. Jim Thomas of Battelle Pacific Northwest Laboratory will kick off the conference with a keynote address that examines the impact of virtual reality on the industrial enterprise.

## Comprehensive Survey of VR

Other invited talks, papers, panels, and tutorials will focus on the application of virtual reality to medicine and other fields. In total, 30 high-quality, peer-reviewed papers will be presented over the course of VRAIS '96, providing a comprehensive survey of ongoing work. Steve Bryson, of NAS's data analysis group, is the VRAIS '96 General Chair. He sees the last two years as being very significant in the development of practical VR technology. "We're not the 'hot hype' field any more -- that distinction now belongs to the Web -- but serious applications of virtual reality technology are really starting to evolve."

## Special Events

Special events include an exclusive new technology preview from SGI and the CyberEdge Journal annual awards ceremony. On April 2, Silicon Graphics, Inc. (SGI) will demonstrate its recently unveiled Onyx InfiniteReality system, which will drive an immersive, interactive, 150-degree field-of-view, SEOS projection system. This event will take place at the SGI Advanced Systems Division Innovation Center at the SGI campus in Mountain View. On Monday, April 1, VRAIS '96 will host the CyberEdge Journal

Product of the Year Award. Since 1992, this award has been presented annually to outstanding people and products in virtual reality in recognition of their accomplishments.

## "Mysteries" of VR Will Be Examined

This year's conference will feature not only the successes in applying VR to real-world problems, but also serious consideration of many of the technical issues still to be resolved in the field. "Virtual reality is still an unsolved problem," Bryson said. "While great progress has been made, there is still a lot of research to be done. We continue to be dogged by hardware limitations, new and difficult software requirements, and the mysteries of how humans operate in VR."

Other NAS staff members involved in presenting VRAIS '96 are Local Arrangements Co-Chairs Sandy Johan and Leslie Schlecht.

*John Hardman, who designed the VRAIS '96 logo and other promotional material, is a member of the NAS customer communications group.*

Next Article | Contents | Main Menu | NAS Home

Next Article | Contents | Main Menu | NAS Home

# FAST Wins 1995 NASA Software of the Year Award

*From left: Tim Sandstrom, Todd Plessel, Paul Kelaita, John West, Fergus Merritt, Gordon Bancroft. Not shown: Al Globus, Kevin McCabe and Val Watson. Other FAST team members included: Rajiv Baronia, Jean Clucas, Pat Elson, Chris Gong, Vee Hirsch, and Pam Walatka (all at NAS), and Clyde Gumbert and Robert Neely (both at Langley Research Center).*

---

No, this somber-looking crew is not ready for the firing squad -- it's the original FAST (Flow Analysis Software Toolkit) development team, reunited at a ceremony held at Kennedy Space Center last October to receive the 1995 NASA Software of the Year award.

The Software of the Year Competition, which was initiated in 1994 to recognize the intellectual property of NASA software developers (who cannot own a copyright or receive royalties for their work), was cosponsored by NASA's Office of Safety and Mission Assurance and the Inventions and Contributions Board. FAST was selected by a team of software experts from NASA centers and Jet Propulsion Laboratory because "it has become indispensable to the analysis of computation fluid dynamics," according to a NASA press release.

The software is used by researchers at all NASA centers, over 40 U.S. aerospace companies, and universities around the world. Here's more information and visualizations rendered with FAST.

Next Article Contents Main Menu NAS Home

## Semiconductor Workshop -- Register Now

There's still time to register for the two-day Semiconductor Device Modeling Workshop to be held March 28-29. For further information, send email to Marcia Redmond, NAS training coordinator, at **redmond@nas.nasa.gov**, or call (415) 604-4373.

## Register Now for June CFD Conference

Participants are encouraged to register as soon as possible for the 15th International Conference on Numerical Methods in Fluid Dynamics, to be held June 24-28 in Monterey, CA (about 85 miles south of Ames Research Center). On April 1, fees increase $50 (to $425).

**Last date to register is May 31.**

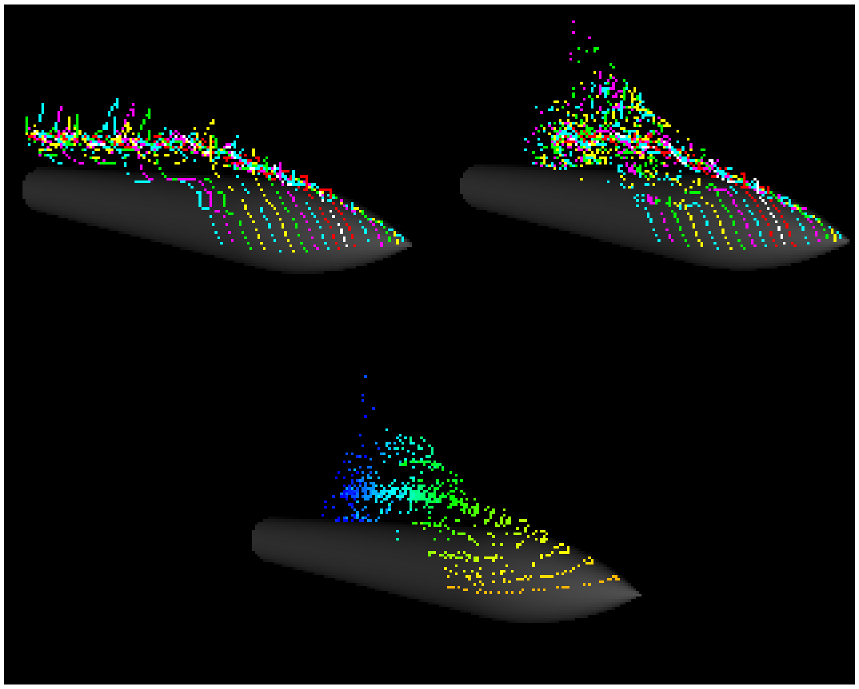
The biennial event provides a forum for technical presentations covering a broad range of computational fluid dynamics topics and features a three-day exhibit with a dozen corporate sponsors. Some 300-350 scientists and researchers from around the world are expected to attend.

For more information, phone Lisa Harris-Calhoun, administrative chairperson, at (415) 604-2084, send email to **lisa_harris-calhoun@qmgate.arc.nasa.gov**.

## Interactive Workbench Arrives at NAS

The Interactive Workbench, a virtual reality-type display invented by Wolfgang Krueger in Germany and built at Stanford University under a NASA research contract, was installed in the NAS Visualization Lab in February. Using head-tracked stereo glasses combined with video images back-projected on the surface of a table, the workbench gives the effect of three-dimensional objects floating above the table's surface. The workbench -- currently used as a display for the NAS Virtual Windtunnel, which uses a dataglove for interactive manipulation of the visualization environment in 3D -- was demonstrated at the NAS Supercomputing '95 booth. The workbench and Virtual Windtunnel will be part of the Large-Scale Interactive Visualization Environment (LIVE) system being placed in the NAS Vis Lab. The LIVE system will be described in the next issue of *NAS News*.

# NAS Visualization Techniques Presented at AIAA Conference in Reno



This graphic shows an instantaneous flow visualization technique (streamlines, top left) compared with two time-dependent flow visualization techniques for an ogive cylinder at high angles of attack. Using a three-dimensional dataset provided by Ames scientist and UFAT user Scott Murman, NAS scientist David Lane presented his findings in a paper, entitled "Visualizing Time-varying Phenomena in Numerical Simulations of Unsteady Flows," at the 34th AIAA meeting in Reno in January. Although instantaneous flow visualization techniques are effective for viewing the flow at one point in time, they do not accurately show time-varying physical phenomena such as boundary layer shedding (streaklines, top right) and vortex core (timelines, bottom), according to Lane.

Next Article | Contents | Main Menu | NAS Home

# Mar - Apr 1996, Vol. 2, No. 16

**Executive Editor:** Marisa Chancellor

**Editor:** Jill Dunbar

**Senior Writer:** Elisabeth Wechsler

**Contributing Writers:** John Hardman, Terry Nelson, Bill Nitzberg, Marcia Redmond, Mark Smith, Allen Van Gelder, Jane Wilhelms

**Image Enhancements:** Chris Gong

**Other Contributors:** David Bailey, Steve Bryson, Bob Ciotti, Archemedes deGuzman, James Donald, Sam Fineberg, Toby Harness, James Jones, Chris Kuszmaul, Brian Juliano, George Myers, Bill Saphir, Leslie Schlecht, Leigh Ann Tanner, Dani Thompson, Bob Thurman, Dave Tweten

**Editorial Board:** Marisa Chancellor, Jill Dunbar, Chris Gong, Mary Hultquist, David Lane, Chuck Niggley, Elisabeth Wechsler

*Our thanks to Editorial Board members who volunteered their ideas, time, and effort to NAS News over the past year: Eric Hibbard, Kristina Miceli, Serge Polevitsky, and Pam Walatka.*

# NAS NEWS

Volume 2, Number 16 March - April 1996

## Reconfiguration Increases davinci's CPU Utilization

by Elisabeth Wechsler

What began as a temporary reconfiguration of CPUs for davinci, the NAS workstation cluster testbed, has become a longer-term change that emphasizes multiprocessor node utilization. The original reason for the change was to run calculations for the NAS Parallel Benchmarks (NPB) on the Silicon Graphics Inc. (SGI) Power Challenge array portion of the cluster.

This first change came in October when the NAS high-speed processing group needed a 9-CPU node to run a perfect square problem for the NPB 2.0 test suite, according to James Jones, who until recently was the lead system administrator for davinci. Because the SGI multiprocessor CPUs come only in multiples of two, it was necessary to create a 10-CPU node instead. To accomplish the reconfiguration, a 2-CPU board was removed from davinci-05, one of the testbed's eight nodes, and reinstalled in davinci-08.

After the first reconfiguration, several other users found it useful to scale code across ten CPUs instead of eight in order to solve larger problems, Jones explained, so it was decided to leave the alteration in place for the time being.

While helping NAS researchers prepare demonstrations for Supercomputing '95, a second temporary configuration change was made in November. Two 75-MHz, 2-CPU boards were added to davinci-05, as the testbed staff wanted to explore the effects on applications of mixing 75-MHz and 90-MHz CPUs within the same system. Since these extra four CPUs were intended to be part of a computer system on order, it was decided to "loan" the CPUs to davinci.

Before the reconfiguration, davinci had four multiprocessor nodes with eight 90-MHz CPUs each. Now, davinci-05 has ten CPUs (four 75-MHz and six 90-MHz CPUs) while davinci-08 has ten 90-MHz CPUs.

According to Jones and current davinci lead system administrator Archemedis deGuzman, these

*Continued on page 2*

## CRAY J90 Cluster To Help C90 Users 'Parallelize' Codes

by Elisabeth Wechsler

Helping NAS C90 users "parallelize" their vector production codes—cost effectively and in a familiar computing environment—is the goal of the latest parallel cluster workstation project in the NAS Systems Division. Delivery of four J90 systems, manufactured by Cray Research Inc., was expected at NAS last month, and represents the first phase of the Newton project.

"Fortran applications representative of the vectorized NAS CRAY C90 workload show that the J90 systems are more cost effective than any workstation cluster, including those from IBM and SGI (Silicon Graphics Inc.)," said Bob Ciotti, project lead, adding that the four J90 systems together offer equivalent performance of five C90 CPUs.

The initial J90 configuration is intended to serve as proof-of-concept and to identify problems seen in internodal communication and scheduling issues specific to parallel computing. Each baseline system will consist of 4 CPUs, 1 gigabyte (GB) of main memory, two network interfaces, and 36 GB of disk storage. One J90 workstation will have additional processors and disk capacity to perform tasks such as compilation.

### Time Saved Recoding Applications

Ciotti believes that C90 vectorized Fortran codes will run unmodified very efficiently on the J90. "The challenge will be to maintain the multitasked vector efficiency while adding in the communication mechanisms required to parallelize the code over multiple distributed memory systems. We really don't know how well this will work, compared with the effort required to recode applications—traditionally, the most significant obstacle to parallel computing."

"This system will provide users with a transitional system to develop message-passing techniques in a UNICOS environment," said George Myers, NAS scientific consultant. "For C90 users, the advantages of the J90 system over other clusters are likely to be a more stable operating system environment and familiar code development tools." In addition to the

*Continued on page 3*

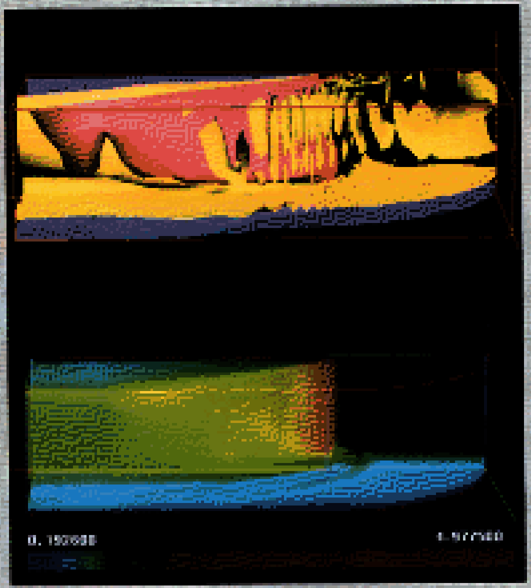


Figure 1. The NASA Ames blunt fin dataset visualized using isosurfaces on the top and direct volume rendering below. The color bar indicates the mapping of data values to colors in the bottom image. The top image shows threshold values 1.1 (reddish), 1.0 (gold), and 0.7 (purple). These images were produced using methods developed at the University of California, Santa Cruz, under a NASA grant funded by NAS. See page 5.

## *Fill Out Reader Survey—Inside*

Please take a few moments to fill out the NAS News Reader Survey, located in the center of this issue. Or, complete the online questionnaire (http://www.nas.nasa.gov/NAS/NASnews/current-issue/survey). Your opinions are very important to us and will help us tailor future issues of NAS News to your interests. **Responses must be received by April 15, 1996.**

We also need your help to update our subscription list so that you can continue receiving your free copy of *NAS News*. Please fill out and return the update form attached to the survey.