TOOLS

Distributed Performance Management Tools

By James N. Chen and Niels Christiansen

This article presents an overview of performance management concepts that are useful in understanding how the IBM AIX system, network, and performance management tools work together in a heterogeneous UNIX environment. New performance management tools in the Performance Toolbox for AIX (PTX) are highlighted.

In a client/server environment, smooth and optimal performance is achieved when all available resources exist in harmonious balance. Reaching this balance is difficult because it requires juggling many machines across systems and networks. Clearly, any attempt to attain this system and network management harmony depends on effective performance management.

Effective performance management can appear to be an elusive goal. To find this "sweet spot" of optimal performance, system administrators must continually adjust and readjust a multitude of parameters. The resources of an enterprise are in balance when the utilization of all system resources is high but not overstressed, all user service goals are met, response times are minimal, and throughputs are maximized. In a perfect world, optimum performance management means *no waiting*; resources are always available.

James N. Chen

Management in a Client/Server Environment

In a distributed client/server environment, system, network, and performance management disciplines can be viewed as an interlocking triad. (See Figure 1.) Each discipline has some functions and data in common with the other two, but each discipline also has individual focus, scope, data, and process model requirements. Although system overheads, protocols, and audience characteristics can vary widely, there is a strong need for commonality of Application Programming Interfaces (APIs) and User Interfaces (UIs) so that various tools can be used together without major compatibility problems among applications. The following are examples of IBM tools in each of these areas:

- System management tools, such as the System Management Interface Tool (SMIT), Distributed SMIT, and Visual System Management (VSM), focus on the details of managing resources for individual systems.
- Network management tools, such as NetView, System Monitor, and Trouble Ticket/6000, focus on global management of system network resources based on Simple Network Management Protocol (SNMP).
- Performance management tools, such as Performance Toolbox for AIX (PTX), focus on detailed resource utilization at individual system nodes across the network.

Performance management tools have very special requirements: low overhead, variable data granularity, high-density data presentation, and high-bandwidth data transport and storage. At the same time, these tools have a common need to access configuration and network data managed by system and network management tools. For these tools to be used both separately and in concert, some degree of overlapping functionality is useful.

Performance Management Phases

Primary performance management tasks can be grouped into four major phases. By using the proper performance management tools in each of these phases, users can obtain optimal performance from the hardware and software they are developing or have purchased.

Component Design and Development

In the component design and development phase, software engineers must create high-quality programs within a specific time that provide functionality and performance at a given price level. To meet the time and cost constraints of customer demands, there are trade-offs between function and performance. Performance-conscious programmers generally focus on the micro-optimization of their individual components. To achieve this optimization of performance, they need detailed modeling, calibration, and measurement tools to visualize the resource utilization of their components in conjunction with the system.

System Integration

The system integration phase brings together component pieces of hardware, operating system, and applications to create a functioning system that can operate by itself or with other systems. At this time, software engineers should focus on the utilization of shared client/server system resources, such as CPU, disk, memory, and communication channels. These resources may require additional trade-offs and usage prioritizations between component pieces to achieve optimal overall system performance.

Software engineers need tools that enable them to see the big picture as well as to focus on micro sections of the system. Such tools enable system designers to create system performance profiles that can be used by component designers for improving component performance as well as by Information Services (IS) personnel in their pre-operations planning phase.

Pre-Operations

In the pre-operations phase, IS personnel design and layout systems and networks to meet the needs of their particular installation. Before deployment in their production environment, systems must be integrated so that the equipment and software will meet the performance capabilities of their planned systems and networks. IS personnel then set performance goals and service

System Management Disciplines

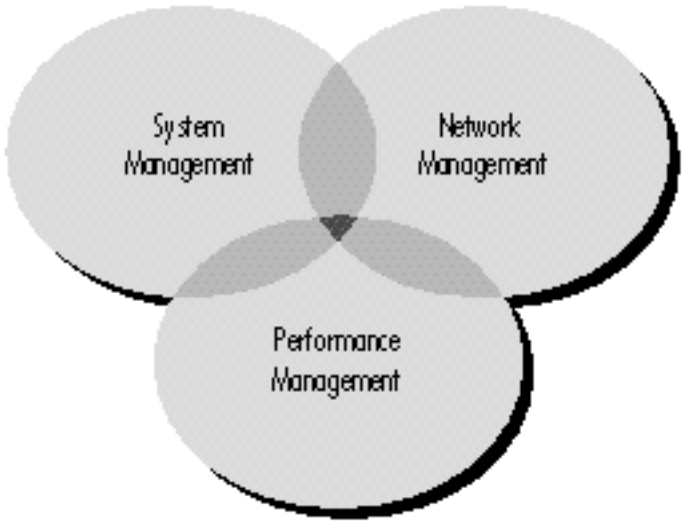

Figure 1. The interlocking relationship of management disciplines

levels that determine what will be considered normal or abnormal performance in the operations phase. Performance calibration tools such as PTX are key to understanding and setting parameters for service-level monitoring.

Operations

In the operations phase, the systems and networks have been deployed into the customer's production environment. System administrators must monitor key resource parameters to detect out-of-range performance, find persistent abnormalities, and fine-tune system resource utilization. System administrators must also be able to determine if performance and service-level goals are being met. They require tools to help diagnose and debug performance problems online. They must also collect long-term data for trend analysis and future capacity planning.

All phases of performance management involve performance calibration and benchmarking. To accomplish these tasks, system administrators need accurate instrumentation of systems and applications, and tools that can read, record, and present this data coherently. System administrators use this data to establish realistic performance goals, track actual performance to desired performance, and implement trend analysis and capacity planning. Recorded performance data is also key for the diagnosis, debugging, and post-mortem analysis of disastrous system failures.

PAIDE and the Agent Component of PTX

- Support for selected non-AIX platforms
- Server side recording of statistics to local file

Manager Component of PTX

- Enhanced recording facilities
- A recording analysis program
- A set of utility programs to post-process recording files
- A dedicated exception monitoring program

Figure 2. Enhancements over PTX 1.1

Performance Management Users

Performance management tools are used by performance analysts, programmers, system and network administrators, support personnel, and general users. Each user group may have special requirements for performance tools suited to their jobs. Most tools are designed with a special group in mind.

- **Performance analysts** characteristically have detailed knowledge of the inner workings of system hardware and software. After many years of experience and detailed study of performance data, they can usually spot problems by examining a few key indicators. Performance analysts need tools that provide detailed information about the inner workings of the system.
- **Programmers** typically focus first on functionality and consider performance aspects only after the coding is completed. This can be especially problematic if they rely heavily on system services without realizing the resource requirements of these services. Programmers need tools that show them if their programs are using system resources efficiently.
- **System and network administrators and support personnel** are primarily concerned with keeping the systems and networks up and running for their customers. They need forecasting tools to alert them before problems happen, diagnosis and debug tools to help them when problems occur, and recording and analysis tools to help them understand how to prevent these problems from occurring in the future.
- **End users** typically need some sensory feedback to inform them of the availability of system resources. They need tools such as alarm lights and resource usage meters to let them know that work is being done.

All groups measure performance management tools by the key criteria of response time, resource availability, and throughput. The Performance Toolbox has facilities to help each of these user groups address their key performance problems in the four phases of performance management.

Performance Tuning Model

A simple performance tuning model consists of six main steps:

1. Monitor key performance statistics.
2. Log and record statistics.
3. Analyze resource utilization statistics and identify bottlenecks.
4. Prescribe a remedy to a performance problem.
5. Set and adjust alert/alarm conditions and notification policies.
6. Make prescribed tuning adjustments and return to step 1.

With the proper performance tools, these steps can be repeated to obtain a fine-tuned environment that meets the desired performance goals. Each step in this model can be done by a variety of tools—manual, semi-automatic, and automatic. In an ideal "lights out" operation, this model would be totally automatic; tools would be preprogrammed to perform each step without human intervention. However, today's technology has not yet reached the point at which total automation occurs with high reliability and consistency. Therefore, it is very important to have semi-automatic tools that perform data collection, sorting, and preliminary analysis, while deferring to trained human operators on critical decisions.

Some tools can operate in a semi-automatic mode by monitoring key statistics and notifying a user when certain actions and decisions need to be made. For example, PTX can alert an operator to pathological performance behaviors to enable the operator to respond with the appropriate corrective actions. PTX also offers users the option of customizing their set of tools with a flexible and expandable toolbox format. Its powerful graphic presentation facility can also be dynami-

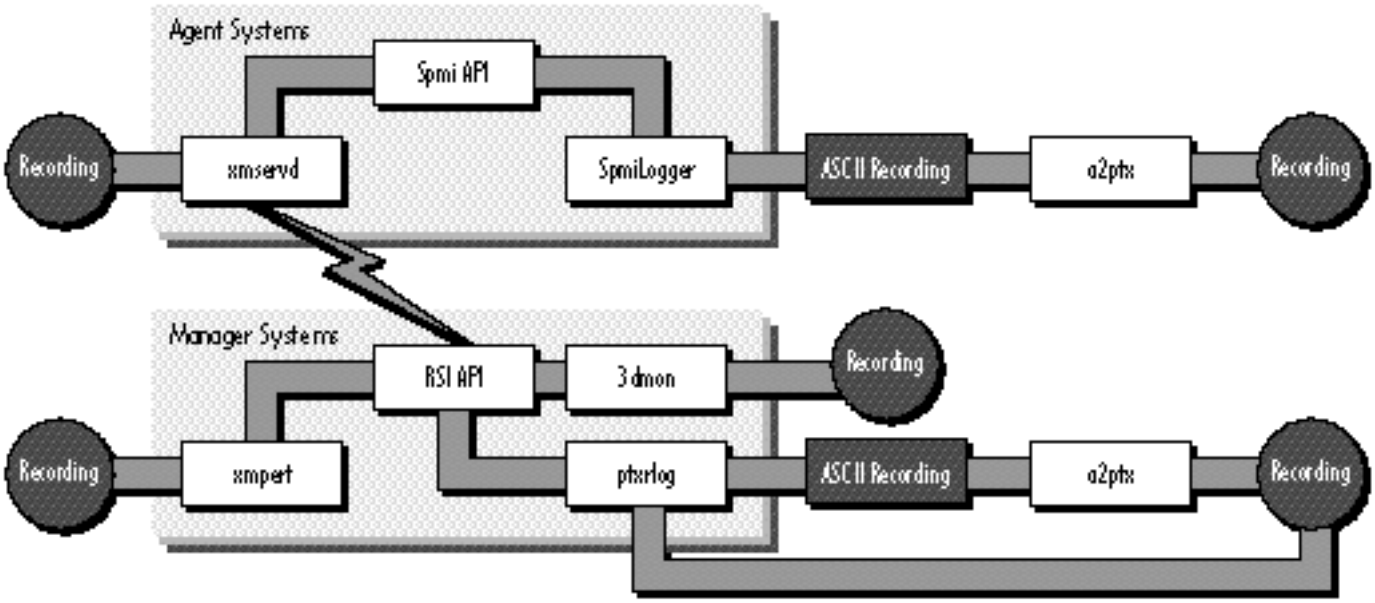

Figure 3. PTX recording facilities

cally customized for individual users. The Performance Toolbox Version 1.1, as described in the article "AIX Performance Toolbox/6000" (*AIXpert*, November 1993, pp. 24-28), provides a good basis and framework to do this. This article also provides a description of the AIX Performance Aide (PAIDE) product, which is the agent component of PTX.

Key enhancements described in the next section build upon and strengthen this tuning model in a distributed heterogeneous client/server environment.

Enhancements in PTX Version 1.2

Two versions of PTX are announced for shipment in the third quarter of 1994. Version 1.2 runs on AIX Version 3.2.5 and Version 2.1 runs on AIX Version 4.1. The two versions are almost identical; they differ only in the supported statistics necessitated by differences in the operating systems. Figure 2 shows the most important enhancements in PTX over Version 1.1.

Support for Selected Non-IBM Platforms

Beginning with Hewlett-Packard Series 9000/700 systems running HP/UX Version 9.03, PAIDE and the agent component of PTX contain everything necessary to install and run the agent on non-AIX systems. Support for HP 9000/700 will be included in the initial shipment of PTX and PAIDE Versions 1.2 and 2.1. Agents for Sun SPARCstations™ with SunOS 4.1.3 will follow shortly. The non-AIX agents will provide all the functionality of the AIX agents, except the ability to export their statistics to Simple Network Management Protocol (SNMP). A basic set of statistics is available across all agents. However, because of differences in hardware and software and varying levels of instrumentation, each platform requires a customized set of statistics to monitor its most important probing points.

Server-Side Recording

In response to customer requests, the `xmservd` daemon of PAIDE and the agent component of PTX has been enhanced to enable recording directly to a local file instead of sending packets over the network. A configuration file provides the following information to the `xmservd` daemon:

- When to start and stop recording
- Which statistics to record
- The sampling interval for each statistic
- How to create and how long to preserve recording files

The overhead of recording from `xmservd` is normally so insignificant that it makes continuous

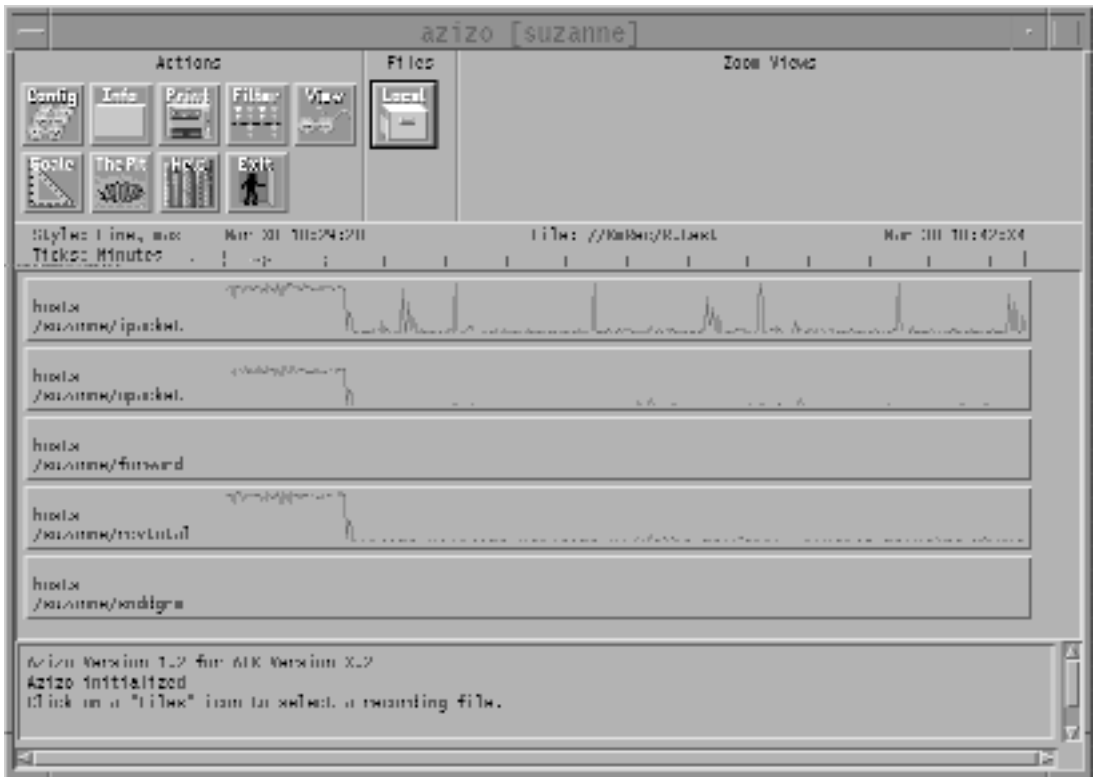

Figure 4. The azizo main window

recording feasible. With continuous recording, you always have the performance data for analyzing a performance problem that you did not detect while it occurred. The recordings produced by `xmservd` can be played back with the `xmperf` monitor and post-processed and analyzed with other programs of the PTX manager component.

Enhanced Manager Recording Facilities

The manager component of PTX has also been enhanced to provide more versatile recording environments. For example, you can now use PTX to record monitoring sessions from the `3dmon` monitor. Another improvement, `ptxrlog`, enables you to record from a remote system without having an active graphical monitor.

Figure 3 shows the various recording facilities of PTX Versions 1.2 and 2.1.

Recording Analysis Program

PTX Version 1.1 had only two ways to analyze a recording: play it back with `xmperf` or print out all details with the `xmtab` program. The new versions of PTX provide a new program—`azizo`—to analyze recordings. This program enables you to work with graphical or tabular views of the data and zoom-in on any part of a recording in any detail needed. The `azizo` program is designed to work in concert with the recording file utility programs described below.

Figure 4 shows the `azizo` main window. Figure 5 shows an example of `azizo`'s zoomed-in graphical window, which gives a partial view of the data in the recording file.

Recording File Utility Programs

With many more sources of recording files, a need arises for programs to split, merge, tabulate, and process recording files. For example, if the `xmservd` daemon has recorded activity on a client system and a server system, it may be interesting to play those recordings back simultaneously or analyze them together. The program `ptxmerge` can merge the two recordings into one file. With the `ptxsplit` program, you can choose from many options to split one recording file into several. You can even create recording files from ASCII input files so that data from other sources can be analyzed together with true recordings. Also, if you want to use other analysis tools, new PTX programs enable you to convert recordings to a variety of output formats for input to spreadsheet or presentation programs.

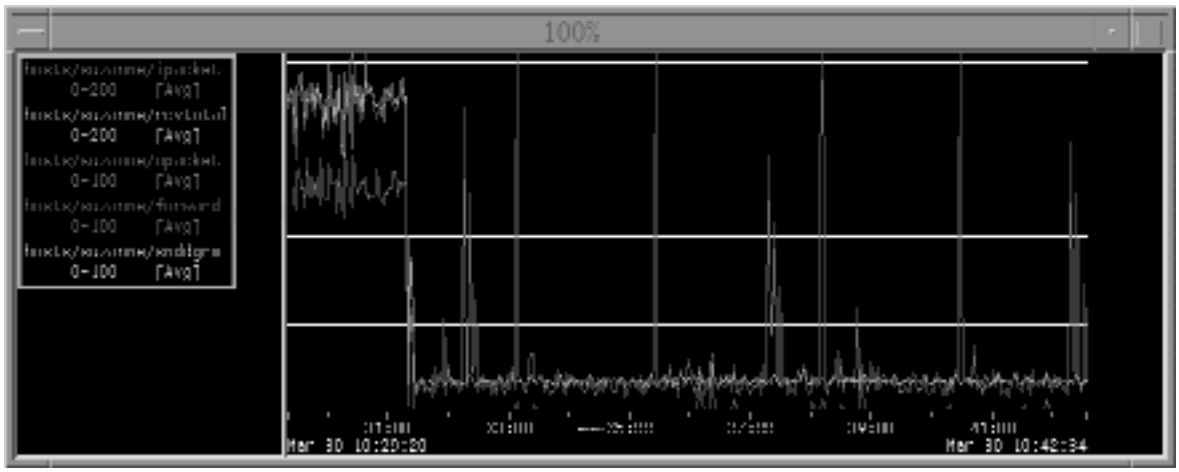

Figure 5. An azizo zoomed-in graph

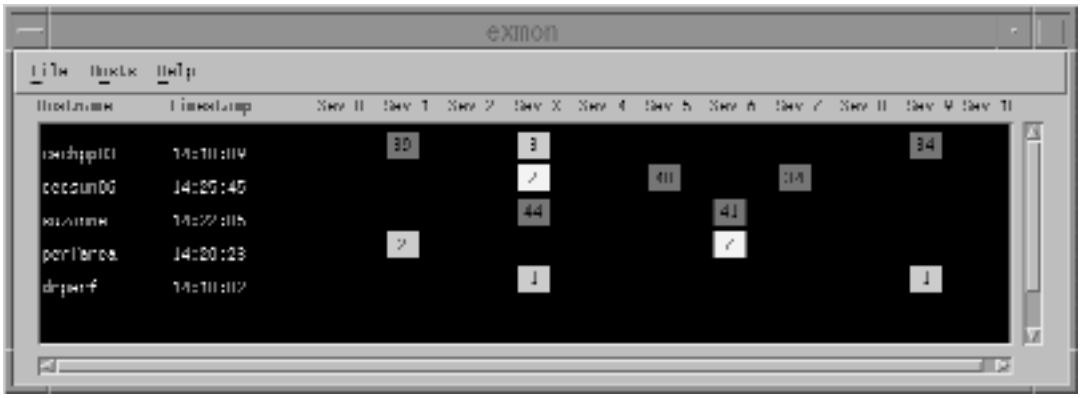

Figure 6. The exmon main window

Exception Monitoring Program

A new program called `exmon` monitors alarms generated by the filter daemons on remote systems. A matrix display of the alarms received brings the latest alarms to your attention and tracks alarm frequency. The `exmon` program also enables you to execute other programs to further analyze the performance on any system that generates an alarm. For example, you can invoke `3dmon` by clicking on a host name and selecting `3dmon` from a user-customizable menu. Figure 6 shows the matrix display of `exmon`.

Performance Management Trends and Directions

Performance tools cover the spectrum from manual to automatic operations, from character-based to live color graphic presentations, from single-node analysis to large distributed networks of heterogeneous machines, from novice to expert tools. These tools can be broadly viewed as first-, second-, or third-generation tools.

First-generation tools tend to be character-based, manually operated tools that only experts would use and understand. They usually present volumes of raw data about a single subsystem; this data typically must be decoded and interpreted by someone very knowledgeable in specific subsystems. Most first-generation tools do not work well, or at all, in a heterogeneous client/server environment. Their data and interfaces are not integrated or correlated across a family of applications. Typical UNIX tools such as `vmstat`, `netstat`, `iostat`, `sar`, and `ps` fit into this category.

Second-generation tools, such as the Performance Toolbox for AIX, tend to use live and interactive color graphic presentations that are easy to customize and use. A semi-automatic mode of operation relies on the pattern recogni-

tion skills of the user to learn to choose pathological performance patterns, characterize them, and then feed back that information to the system so that the system can later detect these patterns and automatically notify the user, or take some prescribed corrective action.

Second-generation tools are designed to operate in a heterogeneous client/server environment. These tools enable users to select the granularity and grouping of statistics they want to collect depending on the resources available. The data collected through program interfaces fits into more globally accepted syntax and semantic definitions so that it can be processed by a variety of applications. These tools usually provide data filtering and alarm facilities for interoperation between different applications.

Performance management tools should address the needs of a wide audience, from novice to expert.

Third-generation tools will be a more fully automated and integrated suite of tools that can be used by users with less detailed system and network knowledge and experience. Because the tools will incorporate more sophisticated analysis and control algorithms, the results will be more accurate. These tools will typically be more knowledge-based so they can automatically learn to recognize new pathological performance patterns. Once recognized, they can respond to correct those new performance problems by balancing available resources or by making detailed recommendations for new resources. Human intervention will be minimized.

Conclusions

Performance management tools are specialized tools that work with a suite of system and network management tools. They should address the needs of a wide audience, from novice to expert. Together, they should allow users to complete a simple performance-tuning cycle—monitoring, recording, analyzing, prescribing a remedy, setting alarm and alert conditions, and adjusting performance resource parameters either manually or automatically. These tools should be easy to install, configure, and use. They should be flexible and expandable. Finally, they should be based on a common set of instrumentation and interoperate in a heterogeneous networked environment.

Key enhancements to the Performance Toolbox for AIX position it as an advanced second-generation performance management tool that can easily evolve into a key third-generation tool for AIX and other platforms.

James N. Chen, *IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Internet: jchen@perfmap.austin.ibm.com. Mr. Chen is a senior programmer in the Systems Management Architecture and Development area. He has been involved in AIX software development, technical marketing, support, software architecture, and performance tools since 1982. He has a BS in Electrical Engineering from Purdue University and an MS in Electrical Engineering from Rutgers University.*

Niels Christiansen*, Encee, Inc., 1015 Quail Park Drive, Austin, TX 78758. Internet: nchris@perfmap.austin.ibm.com. Mr. Christiansen worked in system engineering for IBM Denmark from 1965 to 1990. He is now president of Encee, Inc., a vendor to IBM's Systems Management Architecture and Development area. Mr. Christiansen currently works with design and implementation of performance tools. He received a BS-equivalent degree in Mechanical Engineering from the Copenhagen Teknikum College.*

Industry-Leading Benchmark Results for IBM SP2 POWERparallel System

The most powerful IBM POWERparallel™ system yet shipped was recently installed at the Maui High-Performance Computing Center. The 80-node SP2™, a RISC-based UNIX parallel processing computer, is the initial installation of a machine that the center plans to scale to 400 processors later this year. The 400-node system will be capable of delivering up to 100 billion calculations per second, making it one of the most powerful scalable, parallel computers in the world.

On the NAS benchmark suite, the IBM POWERparallel System SP2 dramatically outperformed the established competition, with up to twice the price/performance of the Cray® T3D. The NAS benchmark suite, comprising both pseudo-application and kernel benchmarks, was run on 16 nodes and 64 nodes of the 80-node system being installed at the Maui Center. n