Testing Cluster Solutions[1]

By Steve Nasypany, Mike Panico, Sonia Weaver, and Juan Zalles

Our RS/6000 development team installed, configured, and tested a variety of existing products to see how they functioned in a clustered environment. We focused on how these individual products increased the availability, manageability, and scalability of the cluster. Some of the products tested include HACMP, Distributed SMIT, Network Installation Management, Tivoli Management Environment, ADSTAR Distributed Storage Manager, Performance Toolbox, and Lotus Notes.

Today's business-computing environment consists mainly of networked machines. Many businesses are beginning to "cluster" their machines to improve performance, availability, manageability, and scalability. A *cluster* is a group of servers that appear to client machines, administrators, and programmers as a single computing resource.

The move to clustering is already taking place within businesses, and now hardware and software developers must play catch-up to engineer products that take advantage of clustering principles. But you don't have to wait for a cluster solution—you can create one now.

What is a "cluster solution" and exactly what problems are these clusters solving? The answer to this question depends largely upon your specific environment, and what you wish to achieve. For the purposes of our testing, we focused on a RISC System/6000® (RS/6000™) environment using the AIX® operating system. In this environment, three goals were of top priority:

- High availability to isolate or reduce the impact of machine, resource, or device failures
- Manageability to balance loads and reduce system management costs
- Scalability to expand the capacity of servers, clients, users, or other resources

Our cluster development team at the IBM Austin site has taken existing software products running concurrently on a collection of RS/6000 hardware to produce a clustered environment. We performed dozens of test cases to discover how these products achieved the above goals. This article summarizes the results of these tests.

After reading the article, you should have a general understanding of the considerations involved in using a variety of products in a common cluster configuration. You should also come away with information (including suggestions) about specific products that might be useful for your environment, even if

Steve Nasypany

Mike Panico

[1] This article also appears on the Web. Open URL: http://www.rs6000.ibm.com/resource/technology. Select the "High Availability & Clustering" link or scroll down to that section. Select the link entitled "Testing Cluster Solutions in an RS/6000 Environment."

you do not use the exact RS/6000 environment we have tested. This article also includes pointers to product information that you can order or view from the World Wide Web.

Summary of Test Cases and Results

For testing, we selected a wide variety of products that are representative of a typical AIX environment, including system management, maintenance, load balancing, printing, and middleware products (as they became available).

We tested the software products shown in Figure 1 in a clustered environment. For each product, we discuss the test case results as they apply to high availability, manageability, and scalability. While some products serve all three goals, other products are applicable to only one or two of these goals.

A 4x2 cluster scenario (four two-node clusters) was used to most accurately emulate a realistic customer cluster environment. Our test plans called for one of the two-node clusters to represent the "Corporate" home office and the other three two-node clusters to represent remote sites, or "Branches." We planned to manage two of the remote sites with the corporate cluster, leaving one to be self-managed (the "Super Branch") with backup from the Corporate cluster. The test plans also called for us to create and extend the remote clusters.

Figure 2 shows the hardware used in our test cases.

Additionally, we used an IBM 3116 Page Printer with a Token-Ring network interface.

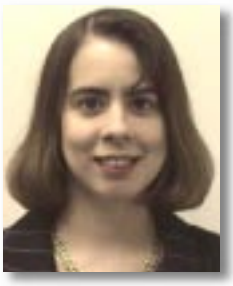
The following sections describe each product we tested, the purpose of each test case, and the results as they apply to high availability, manageability, and scalability.

AIX 4.1.4

We chose the AIX 4.1.4 operating system because this release of AIX was available and supported each of the products we tested. While we are reporting the results of tests for specific hardware and software configurations, these products run on other RS/6000 platforms (such as the RS/6000 SP™) and software (AIX 4.1.5). These test case results should still be useful for planning cluster solutions in your environment.

We installed the base Server package for AIX 4.1.4 on all servers. Roughly three dozen additional operating system filesets were installed, primarily to support the test suite products and device drivers specific to Differential SCSI and Serial Storage Architecture (SSA) drives. We also installed 15 Program Temporary Fixes (PTFs), most of which were specific to new releases of Lotus Notes® (4.11 and 4.5). As of this writing, the fileset updates include:

bos.adt.include	4.1.4.4
bos.adt.prof	4.1.4.17
bos.net.tcp.client	4.1.4.16
bos.rte.libc	4.1.4.18
bos.rte.libs	4.1.4.12
bos.rte.streams	4.1.4.2
bos.rte.tty	4.1.4.13
bos.sysmgt.serv_aid	4.1.4.5
bos.rte.up	4.1.4.17
bos.rte.mp	4.14.17
bos.rte.security	4.1.4.2
bos.rte.libnetsvc	4.1.4.2
bos.rte.printers	4.1.4.7
devices.mca.8ef4	4.1.4.2
devices.mca.ffe1.rte	4.1.4.1



Sonia Weaver

Juan Zalles

Software Products Tested

- AIX 4.1.4
- HACMP/HANFS 4.1.1
- Network Installation Management
- Distributed SMIT 2.2.1
- Tivoli Management Environment 3.0
- Tivoli/Sentry 3.0
- Tivoli/Admin 3.0
- Tivoli/Courier 3.0
- LoadLeveler 1.2.1
- Interactive Session Support 1.2.1
- ADSTAR Distributed Storage Manager 2.1.5.6
- NetTAPE 1.1
- Performance Toolbox 2.1.4
- Print Services Facility 2.1
- Lotus Notes 4.11 and 4.5.

Figure 1. Software products tested in a clustered environment

Hardware Used in Testing

Corporate Cluster

- 2 Model 7013 590 RS/6000s (HACMP servers) with 128 MB of RAM
- 1 Model 7013 530 RS/6000 (HACMP client) with 128 MB of RAM
- 2 SCSI2-DE external hard drives (a 7204 325 with 4.5 GB and a 7204 317 with 2.2 GB)
- 2 Token-Ring adapters for each server node (to allow for HACMP IP address takeover)

Super Branch Cluster 1

- 1 Model 7013 580 RS/6000 (HACMP server) with 128 MB of RAM
- 1 Model 7013 530 RS/6000 (HACMP server) with 128 MB of RAM
- 1 Model 7248 43P PowerPC (HACMP client) with 64 MB of RAM
- 1 7204 325 SCSI2-DE external hard drive with 4.5 GB
- 1 Token-Ring adapter per node

Branch Cluster 2

- 2 Model 7013 530 RS/6000s (HACMP servers) with 48 MB of RAM
- 1 7204 315 SCSI2-DE external hard drive with 2.0 GB
- 1 Token-Ring adapter per node

Branch Cluster 3

- 2 Model 7013 J40 RS/6000s (HACMP servers) with 128 MB of RAM
- 4 Model 010 SSA hard drives with 4.5 GB
- 1 Token-Ring adapter per node
- 1 Ethernet™ adapter per node
- 1 Asynchronous Transfer Mode (ATM) adapter per node
- 1 Fiber-optic Data Distribution Interface (FDDI) adapter per node

Figure 2. Hardware used in testing

HACMP 4.1.1 for AIX (with HANFS)

High Availability Cluster Multiprocessing (HACMP) for AIX provides high availability in a clustered environment. By using shared resources between cluster nodes, it eliminates single points of failure and provides limited downtime in the event of a failure.

For the purposes of this article, the following terms and definitions are used to describe the types of HACMP configurations:

Cascade mode: If a primary node fails, the resources (such as applications, disks, and network address) are acquired by a backup node. When the primary node returns to service, it resumes ownership of its resources.

Mutual takeover: If any node fails, the resources of that node are distributed to other available nodes in the cluster. When the failed node returns to service, it resumes ownership of its resources.

We set up the Corporate cluster for mutual takeover with Internet Protocol Address Takeover (IPAT). IPAT allows another node in the cluster that has a standby network adapter card to assume the IP address of a node that fails. This cluster also had two external SCSI2-DE hard drives, with one node controlling one hard drive and one node controlling the other. If a node failed, the other node acquires control of the external hard drive of the failed node, along with the IP address of the failed node.

We tested the IPAT functions exclusively at the Corporate cluster level. The first two Branch clusters were set up to cascade. In this scenario, a primary node owns a resource (our external SCSI2-DE hard drive). When the primary node fails, the secondary (standby) node acquires the external hard drive. This allows access to any data on that hard drive by logging into the secondary node.

The third Branch cluster consisted of J40s using the PowerPC™ 604 (two-way), along with four SSA external hard drives in a Model 010 drawer. We also configured this cluster for cascading failover. If the primary node fails, the SSA hard drives would be acquired by the secondary node. We used the High Availability Network File System (HANFS) 4.1.1 implementation provided with HACMP instead of using HANFS Version 4.2. The primary reason for using this configuration was because HANFS Version 4.2 cannot operate on nodes running HACMP. Because high availability was an integral part of our clustering test strategy, we used HANFS 4.1.1 combined with HACMP. If your environment requires file locking services, you should implement HANFS 4.2 on a separate cluster without HACMP.

HACMP lets you customize the functionality of the product's operation with scripts that define actions for specific HACMP events. We tried to configure our cluster as generically as possible to reduce the amount of scripting required. Any specific scripting requirements for our test cases are described within the applicable product section.

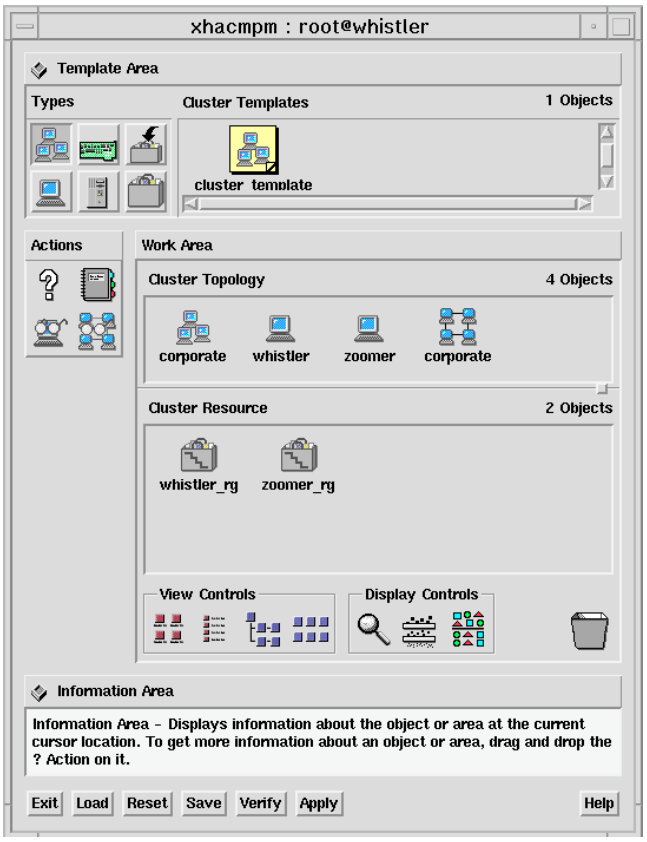


Figure 3. HACMP's Visual System Management interface

High Availability

A suite of regression tests was performed to test each of the major functions of HACMP and HANFS, and to verify our configuration. Other than the problems discussed in the following two sections, the testing went as planned.

Manageability

Each cluster configuration was updated easily using the System Management Interface Tool (SMIT) menus provided with HACMP. We were able to quickly add and modify resources, and the other nodes in the cluster were easily synchronized from the primary node. However, we did encounter one cluster-specific hardware problem during installation and configuration.

This problem involved the SCSI2 Differential Y-cables and device-to-device cables losing contact with one or more of the adapters or drives. This was caused by the weight of the cables, which were not well supported. To avoid this problem, we recommend the following:

- Properly support cables attached to adapters and drives.
- Be careful when moving SCSI2-DE equipped platforms, because movement might loosen the cable or adapter connections.

We encountered another problem in getting HANFS to work properly. The HANFS file systems containing the home directories of all users in the clusters were not available to other components that relied on these directories being available during startup. Until HACMP/HANFS starts, these file systems are not available for mounting because the server node that controls the HANFS file systems can mount these file systems only through HACMP/HANFS.

To avoid this problem, we recommend the following:

- Write post-event HACMP scripts that mount any needed file system upon completion of the HACMP node-up event.
- Carefully plan the startup order of resources on the cluster nodes (particularly in `/etc/inittab`), and adjust for any dependencies required by middleware applications.

Additionally, we encountered a problem with a cluster server node booting on its service address instead of its boot address when the cluster was configured for IPAT. This problem occurred when we manually stopped and started the cluster services repeatedly. To troubleshoot this problem, we checked the network addresses with a command (such as `netstat -i`) and manually restarted the cluster.

Lastly, we encountered a problem importing a volume group from one node to another on the SSA hard drives. To avoid this problem, install the latest updates for the SSA device drivers:

- devices.mca.8f97 and devices.ssa.disk for level 4.1.5.0.

Scalability

When we tested for scalability, we updated one of the three Branch clusters with products to duplicate the Corporate cluster and serve as a "Super Branch." We ran scalability tests in which additional hard drive resources were allocated to the Super Branch cluster. Our testing went as planned.

Network Installation Management

Network Installation Management (NIM) for AIX 4.1.4 installs the Base Operating System (BOS) and optional software on one or more machines in an AIX network environment. A NIM configuration consists of one or more servers providing resources for installations, a set of installable clients, and definitions of the available networks in the environment. An administration server for configuring, controlling, and updating the NIM environment is known as the *NIM Master*. NIM can install to clients and allow clients to download from NIM resource servers.

> ***HACMP lets you customize the functionality of the product's operation with scripts that define actions for specific HACMP events.***

For BOS installations, two primary resources are required. First, the Shared-Product Object Tree (SPOT) provides file system support when a client boots from the network. Second, a source for required software packages used in the installation process must be defined.

High Availability

The NIM tests focused on demonstrating the feasibility of creating and maintaining a highly available resource server and master—one capable of performing BOS and custom fileset installs if a primary NIM master fails. We had to customize the HACMP configuration to create highly available volumes upon which the SPOT, install packages, and `/export` and `/tftpboot` directories could reside.

We began by installing and configuring a NIM master on the primary node of the first cluster. This typically entails defining the network environments and identifying the clients. The HACMP software was started, allowing access to the highly available file systems. The NIM SPOT and install packages were then built on the highly available file systems, along with optional install packages for the AIX server and several test products. This process took from one to two hours, based on the number of optional packages we selected. We then defined optional products to allow them to be installed as custom packages from the NIM master.

Once we had configured the first NIM master, we initiated an HACMP failover using the HACMP SMIT menus. This

allowed the backup node to access the SPOT and licensed product resources defined on the highly available file system. The NIM network, client, SPOT, and install packages were then defined on the backup. This procedure, while repetitive, is not time consuming.

Manageability

The configuration described in the "High Availability" section allows the backup NIM master to access the SPOT, install packages, and `/export` and `/tftpboot` directories if the primary NIM master fails. If a failover occurs during an install, you must enter the identity of the resources and clients being installed to restart the install from the backup.

Because there is no way to synchronize NIM masters, you should carefully plan how you will define resources and clients. For example, the definition of multiple install packages adds to the configuration of the backup master, since each resource definition must be duplicated. If possible, minimize manageability problems by placing optional packages in a single, defined NIM resource. In this case, you would centrally locate installation images in a single directory.

For resource definitions, the highly available file system must be failed over to the backup node. For client definitions, no failover is necessary since the client information is stored locally.

Scalability

The NIM failover capability described in the "High Availability" section does not increase the scalability of NIM by allowing a larger set of machines to be installed. That is because in this setup, only one NIM master is active at a time. For scalability tests, we configured a second cluster identically to the first and performed installs. This capability also provided redundancy if the first cluster failed completely.

A more functional alternative for an environment concerned with maintaining access to non-BOS packages would be to duplicate the packages on several servers. These packages typically require a fraction of the installation time of the BOS packages, and clients can download them from another server if the primary master is busy or is otherwise unavailable.

NIM's overall performance is a function of the resource server, its load, the networks involved, and the number of resources and clients being installed.

Distributed SMIT 2.2.1

Distributed SMIT (DSMIT) lets you build and distribute SMIT commands to other clients on a network. A DSMIT configuration consists of the DSMIT servers, a configuration file server, and its clients. The clients can be grouped into domains, which specify a permanent list of clients to be managed together. During runtime, any set of clients or domains can be managed as a whole. You can define multiple servers, allowing management functions to be distributed across several hosts.

> ***DSMIT lets you build and distribute SMIT commands to other clients on a network.***

For this test case, we focused on making the DSMIT configuration file server a highly available resource, allowing system administration tasks to be performed if the primary node failed.

High Availability

In its current design, the configuration file server holds the DSMIT security configuration files. It is a single point of failure that can hamper high availability. Our test case involved installing a DSMIT server on a clustered pair. To initialize the primary server, we defined it as the configuration file server. We input a list of managing machines that would then have the DSMIT server software installed. Lastly, we defined the administrators and the list of clients on the network to be administered. DSMIT then creates a set of security keys for defining the file server, managing machines, clients, and administrators.

DSMIT was then run to build the lists of clients and servers to be grouped into domains, and to specify which operating system they were using. (DSMIT can support AIX, Sun®, and HP™ clients.) DSMIT can then be started by specifying these domain names. Next, we initialized the backup node and clients using the appropriate keys. The DSMIT file server configuration files were then copied to the backup node.

Upon failover, the backup server assumes the primary master's IP address and can provide the file server functions as needed for a local or remote DSMIT server interface to operate. We successfully performed a series of system administration tasks.

Manageability

DSMIT, like SMIT, provides an easy-to-use interface with comprehensive system management functions. You can use DSMIT to manage one or many clients. For this test case, the primary cluster operated in cascade mode, thus only a single DSMIT server was active at any given time. However, specifying additional hosts as managing platforms allows for multiple DSMIT sessions.

Typically, multiple managing machines are defined to support more than a single administrator. As with all operations in which multiple administrators are performing system management functions, be sure to coordinate the operation of maintenance tasks.

Scalability

While our configuration involved copying identical domain and client information to each DSMIT server, this is not a requirement. Each server can customize its domains. For each group of managing machines tied to a DSMIT file server, you must use a password defined on the file server to operate the DSMIT interface. However, the DSMIT administrator's password does not have to be the same as the root password.

This configuration of DSMIT does not increase the scalability of DSMIT to allow for larger numbers of machines to be managed. Because of the dependency on IP address takeover, only one server in a clustered pair can be actively serving as the file and DSMIT server of its domains.

Alternatively, additional DSMIT servers in a larger cluster can be added and will operate as long as the primary or backup file server is available. Any number of managing machines can be defined, depending upon how many clients are to be administered. However, each managing machine requires a complete DSMIT server license.

Tivoli Management Environment 3.0

Tivoli Management Environment® (TME®) lets you manage systems in a distributed environment. TME uses the hostname/IP address to uniquely identify a machine. Given this implementation, TME 3.0 cannot take full advantage of HACMP's mutual takeover mode to isolate machine failures. TME 3.0 only works with a dedicated hot-standby machine (that is, a machine that takes over the identity of the failed machine).

TME 3.0 addresses the system management problems created by the rapid growth of networks and distributed processing. Its goal is to provide one-touch system management. That is, if you want something to happen on all the machines you manage, you have to push only one button.

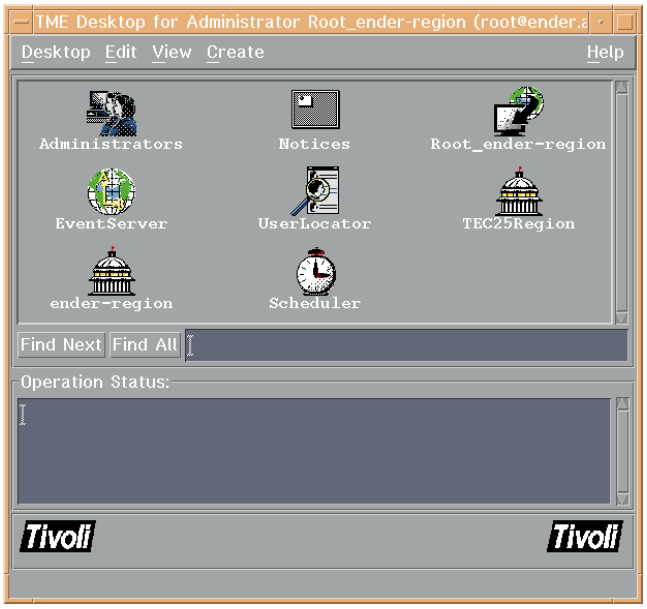


Figure 4. Tivoli Desktop Manager

TME 3.0 also reduces the differences in managing disparate hardware and software. The product includes a fairly robust security model that circumvents the traditional UNIX® problem of sharing root-level authority. TME 3.0 provides more granularity for system administration, allowing you to provide various types of permissions on a need-to-have basis. TME also contains a bulletin board on which users can post notices. TME 3.0 provides these features through both a command-line interface and a graphical desktop.

TME 3.0 lets you put sets of machines into a group called a Tivoli Management Region (TMR). These TMRs can be interconnected so that you can make the region layout transparent to the system administrators. This function also isolates node failures to within the TMR; that is, if a server node fails within a TMR, you cannot manage the entire TMR but you can still manage other TMRs.

One of the most important functions of TME is the *profile manager*, a mechanism that lets you establish a relationship between a profile and a set of subscribers to that profile. The subscribers can also be profile managers, thus enabling hierarchical definitions that allow you to easily manage large numbers of machines. Information about how profiles and profile managers are used is in the "Tivoli/Sentry," "Tivoli/Admin," and "Tivoli/Courier" sections.

The purpose of this test case was to verify that TME could be used to manage a cluster of machines. We believe that the cost associated with mutual takeover mode is optimal for a small cluster, so that is the mode we tested. TME 3.0 also should be able to take advantage of HACMP's mutual takeover mode to isolate network and SCSI adapter failures as well as DASD failures.

We installed TME on all machines. One machine in each cluster was a TME server and the other was a TME client. We also defined a profile manager for each cluster that contained all the machines in the cluster. Once a server is installed, all the clients can be installed from the TME desktop on the server. You need the root password and network access to the client machines to perform the client installations.

We set up each cluster (of two machines) as its own TMR (that is, one server and one client). The clusters were then interconnected. The Corporate cluster was connected to all Branch clusters. The interconnection to clusters that had their own system administrators was one-way (that is, Corporate could manage each Branch, but Branch administrators could not manage the Corporate machines). The interconnection to clusters that did not have their own system administrators was two-way.

The goal of Tivoli Management Environment 3.0 is to provide one-touch system management.

High Availability

TME 3.0 was installed on an HACMP file system to isolate SCSI adapter and drive failures. Node failure of the server in a TMR made the region unmanageable, but we could still manage the other TMRs in the unaffected clusters (that is, the Branch clusters).

We encountered a problem implementing the two-way interconnection using the unsecured interconnect method, which forced us to reinstall an entire TMR. To avoid this problem, we recommend the following:

- Always back up your TME database before making significant changes.
- Always back up your TME database after successfully making significant changes.
- Use the secure method to interconnect TMRs.

Manageability

We were able to quickly install and configure TME on every machine in each cluster. The testing went well and enabled us to manage the cluster as planned.

Scalability

This product lets you add a new machine to a cluster fairly easily. Once the machine was added as a client and all the appropriate software installed, the machine was then added to the cluster profile manager. All the profile managers were redistributed, which caused all the appropriate actions on the new machines. However, it was not easy to add completely new clusters (which necessitates adding a new TMR). TMR configurations cannot be copied, which increases the amount of time needed for installations. To add a new cluster, we had to repeat all of the installation and configuration tasks.

We were able to successfully add machines and entire clusters as planned. One procedure that worked well was creating a profile manager that contained all the machines in the cluster. That way, when you add a new machine, you need only add the machine to the cluster profile and redistribute the data targeted for the cluster.

Tivoli/Sentry 3.0

Tivoli/Sentry® (Sentry) sets up event adapters on machines being managed with TME. It is built on top of the TME framework. The adapter can be configured to detect special events (such as a file system becoming full) and take a specific action. Examples of the types of actions that you can configure include running a predefined program, sending mail to a specific person, or posting an event to either Tivoli's Enterprise Console® or TME's Sentry Monitor.

The Enterprise Console collects and manages events on machines in a network. The Sentry Monitor lets system administrators manage events from one location. It is built into the TME desktop and lets you quickly detect events that need attention. The collection mechanism has fairly sophisticated filtering abilities so that specific events can be routed to specific system administrators. It also keeps the events in a shared database so that multiple system administrators can coordinate among themselves to address the events.

Sentry defines profiles that contain definitions of the events to watch, of how frequently to watch, and of what action should take place when an event occurs. You can define these profiles centrally and distribute them to a large number of machines. When these profiles are distributed, the Sentry adapters on each machine read the profiles and react accordingly.

The purpose of this test case was to monitor the events in a cluster. We programmed a profile in Sentry to watch for file systems nearing capacity. Each cluster had a profile manager for monitoring DASD. The profile manager contained a profile with entries for each file system we were monitoring (for example, `/`, `/usr`, `/tmp`). The subscriber to the profile manager monitoring the DASD was the cluster profile manager, which means that the profile was distributed to all machines in the cluster. For hosts with unique file systems, we set up profile managers with subscribers that were specific machines.

We configured Sentry to provide an escalating set of actions depending on how full the file systems were getting. The lowest level of action was to post a notice on the TME bulletin board. The next level of action was to send mail and post urgent events to both the Sentry and Enterprise Console monitors. The highest level of action was to display a warning dialog on a system administrator's desktop.

> ***Tivoli/Sentry sets up event adapters on machines being managed with TME. It is built on top of the TME framework.***

High Availability

Sentry is an add-on application to TME 3.0 and uses TME's high-availability functions.

Manageability

Although we had to enter a quantity of duplicate information when defining profiles, we were able to quickly define the profile and distribute it to every machine in each cluster. Most of our actions worked

well, with one exception: we were unable to forward Sentry events to the Enterprise Console.

Scalability

Sentry is an add-on application to TME 3.0 and uses TME's scalability functions.

Tivoli/Admin 3.0

Tivoli/Admin® (Admin) is a Tivoli product that uses profiles to manage user and group information. Admin is built on top of the TME framework. When the profile is distributed to the target machines, TME data is exported to the appropriate file systems. Backups of the original system files are also maintained so that changes can be backed out.

The purpose of this test case was to manage user and group information. We used Admin to provide user logins on the cluster machines. The profile managers were set up hierarchically so that users could be defined for the entire enterprise or restricted to specific clusters. We used NFS-mounted home directories. We also took advantage of the Admin support to establish E-mail aliases for users in the cluster.

High Availability

Admin is an add-on application to TME 3.0 and uses TME's high-availability functions.

Manageability

We were able to manage user and group information as planned. One problem arose when we accidently distributed the profiles by enabling the option that replaced the profile information (that is, normal user information) on the target machines with information from the database rather than merging it with the profile information on the machines. Because the profiles in the database only included a subset of the normal user information, all the system logins were destroyed. Although not part of our original test case, we were able to verify that you can indeed recover backup versions of the system files.

Scalability

Admin is an add-on application to TME 3.0 and uses TME's scalability functions.

Tivoli/Courier 3.0

Tivoli/Courier® (Courier) uses profiles to identify files to be distributed to machines. Courier is built on top of the TME 3.0 framework. The profiles specify where to get the files and where to put them on the target machines. The profiles can also contain pointers to programs that can be run at various times before or after the distribution of files.

The purpose of this test case was to distribute files to clients in the cluster. We used Courier to distribute the `/.profile` and `/.kshrc` files for root user to all machines in the cluster. In doing so, the root login environment becomes the same on all machines in the cluster.

High Availability

Courier is an add-on application to TME 3.0 and uses TME's high-availability functions.

Manageability

The test case went well and we were able to distribute the files as planned.

Scalability

Courier is an add-on application to TME 3.0 and uses TME's scalability functions.

LoadLeveler 1.2.1

LoadLeveler® lets users run jobs more efficiently by matching their processing needs to available resources. LoadLeveler defines a pool of managing and submit-only nodes. When a job is submitted to LoadLeveler, it is distributed among the nodes (excluding submit-only nodes) based on a user-defined metric.

The purpose of this test case was fourfold. We wanted to determine the ease of adding and deleting nodes, and determine the ease of changing the configuration of the LoadLeveler pool in a clustered environment with other applications configured in the cluster. We also wanted to determine how LoadLeveler's built-in high availability would interact with HACMP. Lastly, we were interested in how LoadLeveler would behave during an HACMP IP Address Takeover (IPAT).

There were no special installation considerations for LoadLeveler.

High Availability

LoadLeveler has built-in high-availability capabilities. It defines alternate central managers in the LoadLeveler pool. LoadLeveler uses a cascading methodology to implement high availability. LoadLeveler also works well with HACMP, except when using IPAT.

To speed up the failover times from the central manager to the alternate central manager during our tests, we needed to add the following lines to the `LoadL_config` file:

```
CENTRAL_MANAGER_HEARTBEAT_INTERVAL = 30
CENTRAL_MANAGER_TIMEOUT = 4
```

Also, for the client to detect the alternate central manager after a failover, we needed to change the following line in the `LoadL_admin` file on the alternate central manager from:

```
SCHEDD_RUNS_HERE = False
```

to:

```
SCHEDD_RUNS_HERE = True
```

We did encounter a significant problem using LoadLeveler with HACMP. We discovered that after a failover in a cluster using HACMP's IPAT, the alternate central manager would not start the `LoadL_negotiator` and `LoadL_collector` daemons. After IPAT, the central manager node was still communicating with the alternate central manager using its HACMP boot address. In this case, the LoadLeveler central manager constantly sends out keep-alive packets to the alternate central manager, instead of the alternate central manager querying the central manager about the status of LoadLeveler. This becomes a problem because the central manager's IP address now resides on the alternate central manager's standby Token-Ring adapter. The clients were now going to the alternate central manager's standby adapter and not finding the daemons running.

To avoid this problem, we recommend that you write an HACMP post-event script for `node_down_local_complete` that would stop LoadLeveler on the central manager node after node failure. Then the alternate central manager can detect it and become the new central manager. However, it will appear to clients that the old central manager is still running because of IP address takeover.

LoadLeveler lets users run jobs more efficiently by matching their processing needs to available resources.

Manageability

We encountered two problems during testing. The initial version of AFS® installed on the cluster nodes caused all nodes in the LoadLeveler pool to crash. Another problem that affected LoadLeveler was inconsistent IP naming conventions—some nodes had fully qualified hostnames while others did not. To avoid these problems, we recommend the following:

- If you use AFS, ensure that AFS 3.1.1 or later is installed on a system that is included in a LoadLeveler pool.
- Fully qualify the hostnames in the `.rhosts` file on each node.

Other than the problems mentioned above, we were able to successfully use LoadLeveler's job-management functions to submit, manage, and control LoadLeveler jobs.

Scalability

Scalability in LoadLeveler is easily accomplished by editing several stanza files on each node in the pool. We successfully and easily added and deleted nodes, and changed, added, and deleted load metrics.

Interactive Session Support 1.2.1

Interactive Session Support (ISS) lets you manage the distribution of remote sessions.

Clients are transparently connected to servers that have the lowest loads. ISS, using an associated Domain Name Service (DNS) server, dynamically resolves pseudo-host-names that refer to pools of machines.

ISS provides an IP address resolution to clients that points to a minimally loaded machine from the pool. To measure machine loads, ISS can use `rsh` to periodically run user-defined metrics on pool machines, or the LoadLeveler product's load metrics can be used, if available. DNS resources are modified as load conditions in the pool change. You can also define multiple and overlapping pools with different load metrics.

Most installation and configuration work is done on the cluster providing ISS services. Machines that are members of server pools only need to allow `rsh` operations. Client machines only need modifications to the name resolver file `/etc/resolv.conf`.

We installed ISS and DNS applications and data on the cluster's shared disks. A new ISS-specific DNS name service and the ISS process itself were configured as highly available applications. This allowed a second cluster node to take over ISS services upon failure of the first node. An independent metric (not the LoadLeveler metric) was used. For some of the tests we used a simple metric set by the tester to force ISS resolutions to point to the desired machine.

In general, we configured each two-node cluster to be a server pool. For our configuration, we created short scripts to start and stop DNS and ISS. We configured HACMP with ISS and DNS as highly available applications. We then edited the configuration files to define the server pools, load metrics, and IP addresses. Lastly, we altered the clients' DNS configuration fileset.

High Availability

The high-availability aspects of ISS in a cluster were tested with simulated failures of the ISS/DNS node itself. Upon node failure, ISS clients continued to obtain load-leveling connections to the pool. Clients' ISS/DNS requests were then serviced by the takeover node (using IPAT), and client applications continued to connect to unloaded nodes from the pools.

Manageability

The manageability aspects of ISS in a cluster were tested by altering the makeup of the server pools. These tests included changing a pool machine's IP address and changing the metric used to measure machine load. Editing configuration files was all that was required. Clients needed to be modified only if the IP address of the ISS name server changed or if the server pool naming scheme changed.

Scalability

We tested the scalability of ISS in a cluster environment by adding new machine pools, adding machines to existing pools, and creating an overlapping pool that included several clusters and pools. These tests were similar to the manageability tests. The new machine pools consisted of two-node clusters, and an eight-node pool consisting of four two-node clusters. The test case went as planned, and we were able to successfully obtain and use load-managed connections to these pools.

ADSTAR Distributed Storage Manager 2.1.5.6

The ADSTAR® Distributed Storage Manager (ADSM) is an IBM client/server product that lets you consolidate and automate backup and restore functions. ADSM's backup clients support a wide variety of UNIX and Microsoft® platforms.

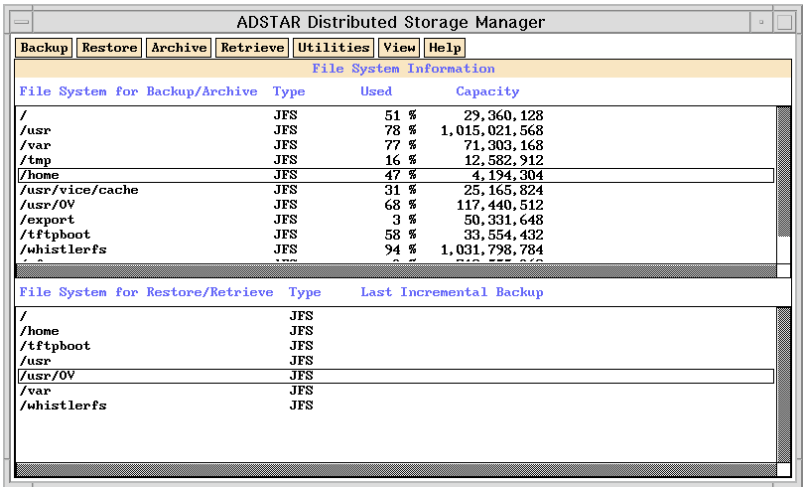


Figure 5. ADSM backup and archive client interface

We followed the standard ADSM installation. ADSM is a complex product with many features for scaling to very large, distributed environments. You should not install ADSM without prior preparation and planning of server and tape storage resources. After becoming familiar with the installation procedures, we were able to configure the ADSM server within a few hours.

High Availability

ADSM has been successfully tested with HACMP by the International Technical Support Center (see the *HACMP/6000 Customization Examples* Redbook, SG24-4498) for one-sided and mutual takeovers. This configuration requires you to use two servers physically connected to a twin-tailed tape storage system. We did not have access to a twin-tailed tape resource, so the purpose of this test focused on providing highly available backup capabilities by installing several ADSM servers in the test environment. We then simultaneously updated the configuration of the ADSM backup clients (using Tivoli's Courier) to allow them to register with a second ADSM server.

We grouped clients together to permanently use 60-70% of the licenses available on their primary ADSM server. The additional licenses could be used by another set of clients if their ADSM servers failed by allowing open client registration. Alternatively, open registration could be allowed for all licenses available on a server. In this case, users would have to keep track of which server they used for backup and restore functions.

Manageability

The ADSM backup client allowed us to easily perform backup and restore functions from any client or server in the network. ADSM can also perform automated backup functions for registered clients. This capability requires you to use a customized Exclude/Include script that specifies the directories and files on the client to be backed up.

We encountered three problems during this test case. The first problem involved basic errors with client environment variables and configuration. The second problem occurred when we registered several clients from the administrator's user interface without fully qualified hostnames, making them unrecognizable to the server when the ADSM backup clients were started on those hosts. The third problem arose when several clients registered remotely with the server and the users manually selected the wrong storage pool.

To avoid these problems, we recommend the following:

- Standardize the environment for the backup clients (we used Tivoli's Courier product).
- Ensure that you configure the hostnames consistently on the ADSM server and its backups.
- If you want to use open registration with the ADSM server, users must know the servers to which to register.

Once we became familiar with the product, ADSM presented no difficulties in operation and significantly eased the backup and restore tasks associated with the other test products. We used ADSM to provide backup functions for nearly all of the products we tested.

Scalability

No scalability problems were encountered during our tests. Several ADSM servers were configured, and we were able to back up the original ADSM server. At the IBM Austin site, ADSM provides on-demand and nighttime backup for several hundred clients.

NetTAPE 1.1

NetTAPE improves and simplifies the management of tape operations and the accessibility of tape devices in RS/6000 network environments. NetTAPE offers consolidated tape operations for all network tape devices from a single user interface, and lets users gain access to remote tape resources transparently, using either a command line or application programming interface.

The purpose of this test focused on allowing global access to tape resources for the servers and clients in our environment. NetTAPE supports a wide variety of devices, from single tape devices to automated tape library servers. For our tests, library servers and twin-tailed tape devices were unavailable for testing.

Initially, several configuration files must be edited to configure the shared tape devices, their hosts, device pools, access control, and any tape libraries. You should carefully read the NetTAPE manual's configuration chapter before undertaking this task. Many of the customization options should be easy to understand for a moderately skilled system administrator.

High Availability

Our tests involved allowing any of the 4x2 cluster nodes and their clients to access six 8mm tape devices spread throughout our environment. The majority of these devices were attached to the HACMP pairs. Because these devices were not twin-tailed, it was not possible to failover the tape drives and recover data from them. However, it was possible to define NetTAPE domains for each cluster node and allow the clients to access an operating domain if the client's primary tape device failed. In real-world operation, a requesting restoration from a failed tape resource would require physically moving the proper tape cartridge to a domain with an operational tape device.

Manageability

NetTAPE allowed easy access to a number of tape devices dispersed in the test environment. We were able to perform tape-management operations, including additions and deletions to the pool of tape devices, with a graphical interface. We could also monitor the status of tape devices.

NetTAPE does require you to define a `.netrc` file for each user. Administrators can use this file to control user access to specific physical tape device hosts. Users must have login access to the tape device host.

Scalability

Our tests showed no problems in allowing users or applications on the server and client platforms to access tape resources over the network. In a local environment, the failure of a single tape device or its host was quickly overcome by simply accessing another available tape drive. In environments where critical backup and restore functions are geographically separated, or where physically relocating a tape cartridge is impractical, a twin-tailed or highly available tape library system would be required.

No scalability limits were discovered using NetTAPE, although our testing did not simulate heavy usage or loading by clients. If your environment has large-scale client tape operations and you want to distribute these functions, you should consider NetTAPE as an option to providing tape devices on every host, or installing and maintaining a tape library server.

Performance Toolbox for AIX 2.1.4

The Performance Toolbox for AIX (PTX) consists of a set of visual monitoring applications (collectively called *the manager*) and data supplier agents that provide performance- and resource-use information from the clients to the manager. The purpose of these tests was

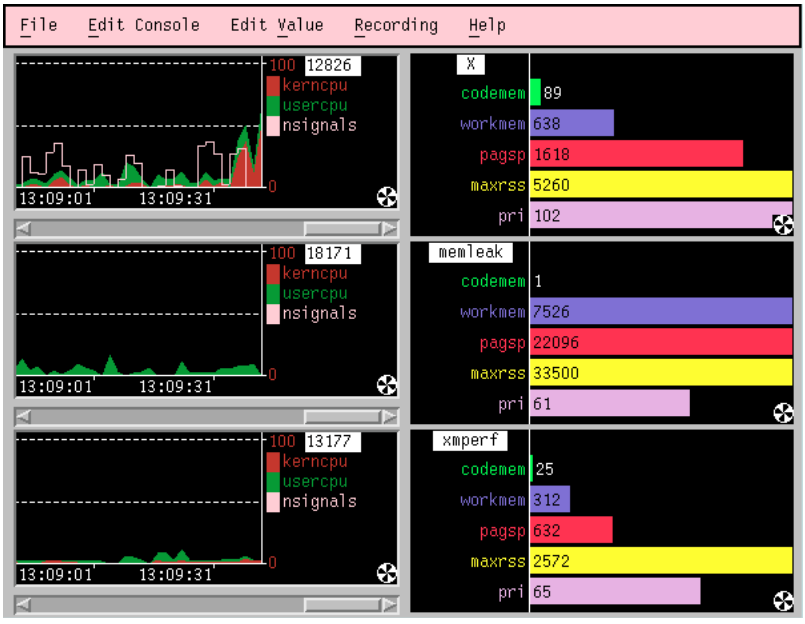


Figure 6. PTX xmperf remote IP monitor

to verify the general operation of PTX in a cluster.

No HACMP-specific procedures were required to install the manager and agents in the test environment.

High Availability

Because the failure of an HACMP node running a manager or agent is likely to terminate the associated processes, the only high-availability capability is the operation of similarly configured managers monitoring the same set of clients. In these tests, the manager could be operating or started on the backup node and acquire information on the set of clients being monitored.

Manageability

The PTX manager allows users to customize the information displayed, which also updates a single configuration file. Once the primary node's configuration was customized, we were able to copy the configuration files onto the backup node.

To aid in managing or monitoring the performance of the clustered nodes, the monitoring application was run on a client in the cluster and various remote resources were monitored on the cluster servers. Various resources could be monitored, including processor, memory, and the network performance of each node. However, be careful when interpreting the clustered nodes' information during a failover. Due to our IPAT configuration (where the backup node assumes the primary's IP address), information from the backup node was reported under the primary node's listing.

Scalability

The PTX Manager can monitor many clients simultaneously. However, the visual representation of the data from a large number of clients can quickly overwhelm a user and become difficult to interpret. System and network administrators using PTX should focus on customizing the PTX displays to report the minimal amount of information needed to monitor a server or client, and troubleshoot specific problem areas or indicators.

AIX 4.1.4 Printing and Print Services Facility 2.1

Print Services Facility™ (PSF™) for AIX is an intelligent printer driver that provides Advanced Function Presentation™ (AFP™) capabilities. PSF supports shared network print server functions and a wide variety of data streams, formats, printers, and platforms.

The purpose of this test case was to verify that basic AIX and PSF printing functions would operate in our cluster environment. When printing in a clustered environment, you must consider several failure issues. When a printer is attached to a clustered RS/6000 platform, both the printer and its host are single points of failure. The same is true for printers attached to terminal servers on a Local Area Network (LAN). For printers with their own network interfaces, the printer and network are single points of failure. In all cases, a clustered print environment would typically require more than one printer resource to be available.

> ***Print Services Facility supports shared network print server functions and a wide variety of data streams, formats, printers, and platforms.***

High Availability

The AIX spooler has been successfully tested with HACMP by the International Technical Support Center (*HACMP/6000 Customization Examples* Redbook, SG24-4498) to create a highly available print system in a cascade configuration. This setup requires you to create remote print queues on each of the cluster nodes and locate printers on non-clustered platforms, terminal servers, or directly on the network. Additionally, the AIX spooling directory is configured on both clustered nodes to reside on a highly available file system.

When clients or servers submit print jobs to a queue residing on the primary cluster node, the job is placed in the highly available spool directory and sent to the remote print server. If a primary cluster node fails, the backup node's queue daemon (`qdaemon`) and remote queue are started. The backup node continues to accept print requests and processes those already in the spool. Print jobs in the process of being submitted when a cluster platform fails are lost and must be resubmitted once the backup has initialized.

To achieve this capability, you must develop two short HACMP event scripts: one that defines the processes the backup must start on takeover, and one that terminates the processes on the backup when the primary node becomes available again.

PSF is not enabled for cascade capability. While some of PSF's capabilities use AIX print-queue concepts, PSF is a much more sophisticated product with functionality well beyond basic printing. If your environment requires highly available print functions, the optimal solution is to have more than one PSF server active (just as most environments have multiple print queues). Clients simply submit print jobs to any available queue, with each print server having multiple queues and printers.

Manageability

Basic AIX print management and PSF resources are easily managed using standard SMIT configuration panels. We were able to successfully reconfigure and update resources in the test environment. Because of the size of some font files used by PSF, you might want to locate these resources on an HANFS platform if you are using multiple print servers.

Scalability

With PSF, you can define multiple print resources and queues. Scaling of resources could entail additional servers, printers, disks, and network resources. For our tests, we reconfigured the Super Branch to be a PSF server, and defined its own printing resources. Printing requests were then serviced by either the Corporate or Super Branch. No problems were encountered in these tests.

Lotus Notes 4.11 and 4.5

Lotus Notes® (Notes) is a Lotus® product for messaging and groupware collaboration. Notes can provide electronic mail, communication, database, document management, and workflow capabilities.

We chose Notes 4.11 and 4.5 because they were the latest releases available when we began our test cases. Notes Release 4.11 does not support the built-in clustering and database partitioning features available in Release 4.5.

In our Notes 4.11 installation, we focused on the operation of multiple servers within our cluster without using HACMP. For the Domino Advanced Services (Notes 4.5), we focused on testing the server mail and replication features in our cluster without using HACMP. For our client operations, we tested the use of Notes executables and resources with HANFS.

High Availability

Existing Notes Server Release 4 solutions have been developed to work with HACMP. For more information, see *Executive Summary: Scalable Lotus Notes on the RS/6000 SP*, available on the Web at the following URL: http://www.rs6000.ibm.com/resource/technology/notesSP/. This document also discusses using the Domino Advanced Services with HACMP.

We installed Notes executables on each server, and placed our data directories and databases on the highly available file systems. The highly available file systems are required when using HACMP, but they are not when using the built-in clustering functions in Domino.

In our installation, the Domino Server automatically enabled the billing functions. You can turn off these functions by removing the billing references in the `notes.ini` files `ServerTasks` field.

Domino comes with its own built-in cluster functions that are dependent upon its database replication. When a server with a replicated database fails, Domino Advanced Services responds to user and application queries by routing those requests to a server with a copy of the needed database. Domino Advanced Services also increases availability

by providing load-balancing features for client requests in a cluster.

HACMP can provide failover capability for mail routing. In environments where mail routing is critical, an HACMP solution is preferable to replicating mail database and routing functions over several servers. Domino Advanced Services does not provide HTTP/IP address failover, which can be provided in an HACMP environment. Alternatively, using LoadLeveler's ISS functions across a Domino cluster with replicated data can create highly available HTTP functions.

If a clustered server replicator is shut down or fails before a replication can be performed, the automatic modification events are lost. For this reason, we recommend that you implement a scheduled replication procedure with all members of the cluster and not depend completely on automatic replication. With HACMP, you can script a forced replication procedure when restarting an HACMP node with the Notes server.

Partitioning allows multiple Domino servers to run on a single platform. While partitioned servers operating on a single host can participate in a cluster, we recommend that other Domino servers in a cluster be located on separate systems. With partitioning, a single server instance can fail while the others continue to operate. Each partitioned server uses its own data directories and resources, requiring additional memory and disk space for each instance of the server.

Installing the Notes client executables and user account on our HANFS cluster allowed clients to operate after a failover. During a failover, the loss of services typically appears to the clients as a standard NFS timeout until the backup platform has completed a takeover. This capability should also be practical in using other file systems, such as AFS or DFS.

If you choose to operate multiple clients on a single server (as we did), you need to set the following variable in the `notes.ini` file for each client and server:

```
Notes_ISOLATION=1
```

Although you can set this variable using the client user's environment files, we recommend using the `notes.ini` file because it is easier to maintain than a user profile or login script.

Overall, we encountered no significant problems operating Notes 4.11 or 4.5 in our cluster. Because Notes uses shared memory functions extensively, we recommend that you implement a script to clean shared memory and release semaphores when a server is restarted. This might prevent or reduce the impact of shared memory problems in your environment.

Domino Advanced Services increases availability by providing load-balancing features for client requests in a cluster.

Manageability

All Notes server-management functions are performed with administration user interfaces. We successfully set up servers, address books, mail routing, and database replication policies without any significant problems. We strongly recommend that you use the Lotus Notes Release 4 on *AIX Systems Installation, Customization and Administration* Redbook (SG24-4694) for planning any installation. Client registrations also presented no significant problems.

For better performance and manageability, we changed the following operating system parameters in our environment:

- Increased the number of processes allowed to 1024
- Expanded paging space to a minimum of 256 MB
- Increased the number of licensed users to 16

We also recommend that each grouping of a Notes' user and mail file system be located in its own volume group and logical

volume. To allow additional resources to be added easily as the number of users grow, groups of Notes users should be broken up into different volumes. Logical volumes can also be mirrored for environments that require it. Ideally, you should place each volume group's log file on a separate physical disk. While this might be impractical for small installations, it reduces the impact of logging activity on other heavily used file systems and eases maintenance tasks.

Finally, centrally locating the Notes client executables and the user databases using HANFS eases maintenance, but does affect performance and can cause client sessions to fail in the event of a prolonged takeover. Alternatively, Notes clients executables can be located on each client (requiring approximately 100 MB of disk space), while user accounts and databases can be maintained on a global file system (NFS, HANFS, AFS, or DFS).

Scalability

A single Notes server's scalability is primarily a function of network and platform performance and the number of clients being supported. Our testing involved creating another Notes server in the test environment and verifying its operation. No problems were encountered in registering or configuring mail routing.

With Notes 4.5, we successfully installed and configured additional Domino servers. To improve both scalability and performance, we recommend that you replicate the most heavily used databases across multiple servers. Partitioning allows additional Domino servers to be located on the same host, and you can add additional resources (CPU, disk, memory) to increase performance.

We did not perform client scalability tests. In any large organization, this would be a function of network, server, and the chosen file system limitations. With middleware applications such as Notes, performance issues become an important concern. We recommend using striped file systems to improve database performance, and that Fast/Wide SCSI or SSA devices be used if possible. Optimizing database locations and replication procedures can also increase performance.

Overall, we recommend that you approach each performance issue separately, without implementing an aggregate of changes simultaneously. Focusing on performance changes one at a time allows you to analyze the effect of each change, and helps isolate and identify the particular bottlenecks in your environment.

Conclusions

Having reached the end of our testing, we have learned that up-front planning plays an integral part in the ease of creating and managing a cluster. We have also learned that some products lend themselves to clustering easily, while others do not. We began with some misconceptions about the number of products that can be used within clusters, and about the costs associated with backup resources. We also arrived at a list of questions that you can ask yourself before you begin clustering (see Figure 7). Our final thoughts cover these topics.

> ***The most important conclusion of our testing is the requirement for detailed planning for any cluster.***

Planning

The most important conclusion of our testing is the requirement for detailed planning for any cluster. While many of the tasks associated with installing and configuring over a dozen products in an HACMP environment seemed daunting at first, hindsight showed that good planning can make these tasks very manageable. Additionally, HACMP itself can seem quite complicated at first, but once we had become proficient with its operation, it was easily maintained, could be quickly configured, and was reliable.

We recommend that administration staff be familiar with all of the products mentioned in this article before attempting to install, configure, and operate them in a

clustered environment. Clustering can complicate the analysis of simple problems that are easily diagnosed in a nonclustered environment. The vast majority of problems we encountered were related to hardware and basic product setup, rather than the result of operating in a cluster.

Products

Several products, not designed for a clustered environment, were successfully configured to work with HACMP. Applications such as DSMIT and NIM can be adapted to operate under HACMP, but if you do not adequately plan for their use in a highly available environment, your customization of the product and use of resources might render them inefficient (or even unusable).

One surprise from our tests was how many of the products could actually be installed and operated from the same cluster. We expected that each cluster would eventually become unmanageable after being loaded with several products, but our experience showed us that clustering does not require a "one-product-per-cluster" model.

For instance, a business using Notes primarily during the day and performing network installs or backup functions mainly at night could reasonably load a single cluster with both capabilities. Other organizations might require only one application to be highly available and not require this of other installed products.

We also had thought that some clustering solutions would be doubly expensive because backup nodes were unused resources until a failover occurs. In our tests, we used many of our backup nodes for operating other products and performing other tasks. For instance, an organization might have a custom, highly available application requiring 24-hour operation, but might have a noncritical need for a second product. The cluster could be configured to run the noncritical application on a backup node, and HACMP could even terminate the other applications to guarantee performance if a failover occurred.

Additionally, disparate hardware and resources can be grouped to create clusters.

Questions to Ask Yourself

We recommend that you ask yourself the following questions before beginning work on your cluster solution:

- How do I prioritize high availability, manageability, and scalability?
- What hardware, software, or tasks require high availability?
- What hardware, software, or tasks do not require high availability?
- What existing hardware or software is compatible with HACMP?
- Do you require your environment to be highly available 24 hours a day?
- To what extent are your system administrators familiar with the products involved?
- What existing equipment is most prone to failure?
- What are the worst-case hardware, software, and resource failures?
- What are the potential single points of failure in your environment?
- What level of performance loss is tolerable in a failover?
- Can nonprimary nodes and resources be used for other tasks?
- What are the growth needs of the environment?
- Does a single operation environment meet all customer's needs?
- How should hardware, software, and resources be distributed to optimize availability, manageability, and scalability?

Figure 7. Cluster-planning questions

Some environments might have rare failover events, and would be willing to tolerate some level of performance loss during these times. These environments can pair less powerful backups with better equipped primary nodes. For instance, instead of creating two clusters out of pairing two J40

Web Pages

The following Web pages can be reached by anyone with an Internet connection:

- HACMP
  http://www.rs6000.ibm.com/software/Appfinder/clustering.html
  http://www.austin.ibm.com/software/Apps/hacmp.html
  http://www.clam.com

- LoadLeveler/ISS
  http://www.ics.raleigh.ibm.com/ics/issfact.htm

- Lotus Notes
  http://www.lotus.com/products/

- NetTAPE
  http://www.rs6000.ibm.com/software/Appfinder/datamanagement.html

- NIM
  http://www.developer.ibm.com/library/ref/about4.1/df4insta.html

- Tivoli
  http://www.tivoli.com/
  http://www.tivoli.com/tivevery/contacts.html

The following Web pages are internal to IBM and can be reached only by IBM employees:

- ADSM
  http://w3.austin.ibm.com/~adsm
  http://w3.austin.ibm.com/afs/austin/depts/itso/itso_web/htmlbooks/sg244601.00/4601fm.html

- DSMIT
  http://w3.austin.ibm.com/afs/austin/depts/itso/itso_web/htmlbooks/gg244380.00/4380fm.html

- HACMP
  http://hacmp.aix.dfw.ibm.com

- LoadLeveler/ISS
  http://w3.austin.ibm.com/afs/austin/depts/k76b/public_html/lxperf/loadleveler/main_load_page.html

- NIM
  http://w3.austin.ibm.com/afs/austin/depts/d57s/nim_html

- PSF
  http://w3.austin.ibm.com/afs/austin/depts/itso/itso_web/htmlbooks/gg243570.01/3570fm.html
  http://w3.austin.ibm.com/afs/austin/depts/itso/itso_web/htmlbooks/gg243570.01/psfdescr.html

- PTX
  http://w3.austin.ibm.com/afs/austin/depts/itso/itso_web/htmlbooks/gg242541.00/2541fm.html
  http://w3.austin.ibm.com/afs/austin/depts/itso/itso_web/htmlbooks/gg242541.00/2541ch6h.html

- Tivoli
  http://w3.austin.ibm.com/afs/austin/depts/tivoli/public_html/index.html

and two 59X models, it might make more sense to have the less powerful nodes serve as the backup resources.

Finding Product Information

For more information about the products described in this article, you can order IBM Redbooks or access product Web pages.

References

The following books can be ordered from your IBM sales representative or, in the U.S., from IBM Customer Publications Support at 1-800-879-2755:

- *ADSM V2 Guide* (SG24-4532)
- *ADSM for AIX: Advanced Topics* (SG24-4601)
- *AIX System Management* (GG24-4380)
- *An HACMP Cookbook* (SG24-4553)
- *Examples Using TME10 on AIX* (SG24-4867)
- *Getting Started with ADSM AIX Clients* (GG24-4242)
- *HACMP/6000 Customization Examples* (SG24-4498)
- *High Availability on the RISC System/6000 Family* (SG24-4551)
- *IBM LoadLeveler Administration Guide—Release 2.1* (SH26-7220)
- *IBM Print Services Facility for AIX: Print Administration* (S544-3817)
- *IBM SystemView for AIX: An Overview* (GG24-2541)
- *Implementing High Availability on a RISC System/6000 SP* (SG24-4742)
- *Lotus Notes Release 4 on AIX Systems Installation, Customization, and Administration* (SG24-4694)
- *Network Installation Management Guide and Reference* (SC23-2627)
- *Printing Under AIX Version 4* (GG24-3570)

Steve Nasypany, *IBM Corporation, 11400 Burnet Road, Internal ZIP 9564, Austin, TX, 78758. E-mail: nasypany@austin.ibm.com. Mr. Nasypany is a staff software engineer in AIX System Management/User Interface. He has a BS in Computer Science from the University of Houston—Downtown.*

Mike Panico, *IBM Corporation, 11400 Burnet Road, Internal ZIP 9564, Austin, TX, 78758. E-mail: panico@austin.ibm.com. Mr. Panico is a software engineer in AIX System Management/User Interface. He has a BS in Computer Science from Florida Atlantic University.*

Sonia V. Weaver, *IBM Corporation, 11400 Burnet Road, Internal ZIP 9561, Austin, TX, 78758. E-mail: soniaw@austin.ibm.com. Mrs. Weaver is a software engineer in AIX Information Design and Development. She has an AAS in and is completing her BLS in Technical Communication at St. Edward's University.*

Juan Zalles, *IBM Corporation, 11400 Burnet Road, Internal ZIP 9564, Austin, TX, 78758. E-mail: juanz@austin.ibm.com. Mr. Zalles is an advisory software engineer in AIX System Management/User Interface. He has a BA in Mathematical Sciences from Rice University.*