

Uptime in Real Time

With NetWare Cluster Services for NetWare 5

Linda Kennard

Like most people, you want the peace of mind that comes from doing your job well. As a network administrator, this peace of mind comes from knowing that users have continuous, uninterrupted access to critical server-based resources, including licenses, applications, volumes, applications, services, applications, and did I mention applications?

To ensure that applications and other server-based resources are continually available, you try to maximize each server system's mean time between failure (MTBF) and minimize its mean time to repair (MTTR). You can thereby increase each server system's availability, which is the portion of time a system is usable and which is usually expressed as a percentage. For example, a server system with an MTBF of 10,000 hours and an MTTR of five hours delivers 99.95 percent availability, according to Robert Wipfel, Novell architect for the high-availability solutions code-named ORION.

Of course, unless your company's server systems (including both hardware and software) guarantee 100 percent availability, server-based resources will be unavailable periodically. Consequently, you can count on several minutes a year experiencing the antithesis of peace of mind. In fact, a system touting 99.95 percent availability may be unavailable for anywhere from 2,600 to 10,000 minutes per year, according to one estimate. ("Can't Tolerate Server Downtime? Cluster 'Em!" *Datamation*, Aug. 15, 1995. You can download this article from <http://www.datamation.com/PlugIn/issues/1995/august15/08bmm100.html>.)

In other words, to your company, a server system that provides 99.95 percent availability equates to as many as 10,000 minutes of lost revenue—and that loss can be considerable. (See "Did You Know?" on p. 10.) To you, a server system that provides 99.95 percent availability equates to as many as 166 sweat-filled hours spent restoring that system to calm your budget-conscious boss (not to mention anxious users). What is a lone network administrator to do?

WHY NOT MUSTER A CLUSTER?

One thing you can do to increase the availability of server-based resources is to muster a cluster (pardon the expression). A cluster is a combination of hardware and software that enables you to interconnect several servers, called nodes, that thereafter act as a single system. (For definitions of cluster-related terms, visit the *NetWare Connection* web site at <http://www.nwconnection.com>.)



Generally speaking, clusters provide between 99.9 and 99.999 percent availability. (See *Datamation*, Aug. 1995.) If you implement a cluster, you will probably experience no more than 500 minutes per year of downtime and possibly as few as five minutes. (See Figure 1 on p. 8.)

Traditionally, clusters have been proprietary solutions, but Novell has adopted and adapted the cluster concept to deliver NetWare Cluster Services for NetWare 5. Available in beta since March, NetWare Cluster Services for NetWare 5 will be released soon. (In fact, look for an announcement at NetWorld+Interop being held this month in Atlanta.)

NetWare Cluster Services is software that you install on industry-standard hardware to create a multinode cluster that ensures high availability of critical server-based resources, including the following:

- Connection licenses
- Data volumes
- Services, including Novell Distributed Print Services (NDPS) 2.0 and Novell's Dynamic Host Configuration Protocol (DHCP) and Domain Naming System (DNS) services
- Applications, including GroupWise 5.5, Netscape Enterprise Server, Oracle8 for NetWare, and various Java application servers such as IBM WebSphere

NOVELL'S CLUSTER CONCEPT

Code-named ORION II, NetWare Cluster Services is the second of Novell's clustering solutions. Novell introduced its first clustering solution, Novell High Availability Server (NHAS, pronounced *N-hass*) in September 1998. NHAS (formerly called ORION I) is software that you install on two interconnected NetWare 4.11 servers, both of which are actively

	Availability						
	90%	99%	99.9%	99.99%	99.999%	99.9999%	100%
Downtime Minutes Per Year	50,000	5,000	500	50	5	.5	0

Figure 1. Availability refers to the percentage of time a system is available. This table shows an estimated number of minutes a system will be down per year given the availability.

running applications. You then connect this two-node cluster to a shared fibre channel subsystem, where all data is stored.

What are the implied cluster limits of NHAS? You can build clusters with only two nodes using NHAS. Unlike NHAS, which supports only two-node clusters, NetWare Cluster Services is designed to support clusters with as many as 32 nodes. To date, Novell has officially tested and certified eight-node configurations of NetWare Cluster Services but will test cluster configurations with as many as 32 nodes on an individual customer basis as the need arises. At BrainShare '99, held during March in Salt Lake City, Novell engineers demonstrated two 12-node configurations of NetWare Cluster Services.

You can install NetWare Cluster Services on any Intel-based server running NetWare 5 with Support Pack 3. (As we go to press, Novell expects the Consolidated Support Pack 2, which includes NetWare 5 Support Pack 3, to be available on the Novell Support Connection web site soon.) Officially, Novell has tested NetWare Cluster Services using server models from several vendors, including the following:

- Compaq ProLiant (various models, including 6500 and 1850R)

- Dell PowerEdge (2300 and 4300 models)
- Hewlett-Packard (HP) NetServer LPr
- IBM Netfinity (7000 series)

Cluster Configurations

The 12-node clusters that Novell engineers featured at BrainShare '99 demonstrate a typical NetWare Cluster Services configuration. (See Figure 2.) Novell engineers installed NetWare Cluster Services on NetWare 5 (with Support Pack 3), which was running on HP NetServer LPr servers for one demonstration and on Compaq ProLiant servers for another demonstration.

The Novell engineers then connected the clusters' nodes through a Fast Ethernet (100 Mbps) switch and standard Fast Ethernet network interface boards. To meet the requirements of NetWare Cluster Services, the Novell engineers then configured all of the nodes to run the IP protocol and placed the nodes on the same IP subnet. Collectively, the nodes then ran NetWare 5 services and Oracle8 for NetWare, Netscape Enterprise Server for NetWare, and GroupWise 5.5.

The Novell engineers also connected four clients to each of the 12-node clusters. In addition to running the latest versions of the NetWare client software (namely Novell Client 4.11b for Win-

dows NT and Novell Client 3.1 for Windows 95/98), the nodes ran Netscape Navigator and the GroupWise 5 client software. To enable access to the Oracle8 database, the Novell engineers used the Java Runtime Environment and the Java Database Connectivity (JDBC) application programming interface (API) to write a simple, Java-based client. (Both the Java Runtime Environment and the JDBC API are included with NetWare 5.)

Lighthearted

Although the Novell engineers used a Fast Ethernet switch to connect the nodes within their demonstration clusters, you can also use a hub. The traffic that flows between nodes within a cluster is light. This node-to-node traffic consists mainly of packets generated by the cluster's heartbeat protocol. To detect potential failures, the nodes transmit and listen for heartbeat packets. (For more information about this heartbeat protocol and how it is used to detect potential failures, see the "GIPC Heartbeat" section on p. 12.)

Because the traffic generated by the heartbeat protocol is so light, performance is not an issue for node-to-node traffic within the cluster. As a general rule, a switch is needed only if the cluster's node-to-node traffic shares the same network forwarding device (such as the switch or hub) as the cluster's node-to-client traffic.

SAN or Sans SAN?

In the BrainShare '99 demonstrations, Novell engineers connected the 12-node clusters to a shared disk storage system and used Redundant Array of Independent Drives (RAID) level 5 for fault tolerance on both storage subsystems. For one demonstration, the Novell engineers used Fibre Channel boards, fiber optic cable, and a multiport hub to connect the cluster to the Compaq ProLiant RAID Array 8000 system. (Fibre Channel is a fiber-optic channel standard that supports several grades of fiber-optic and copper cable with bandwidth ranging from 100 Mbps to 1 Gbps.)

When you connect a cluster to a shared disk storage system, you form a storage area network (SAN). (SAN is an industry term for a network that connects servers and storage peripherals.) Novell strongly recommends using NetWare Cluster Services in SAN configurations.

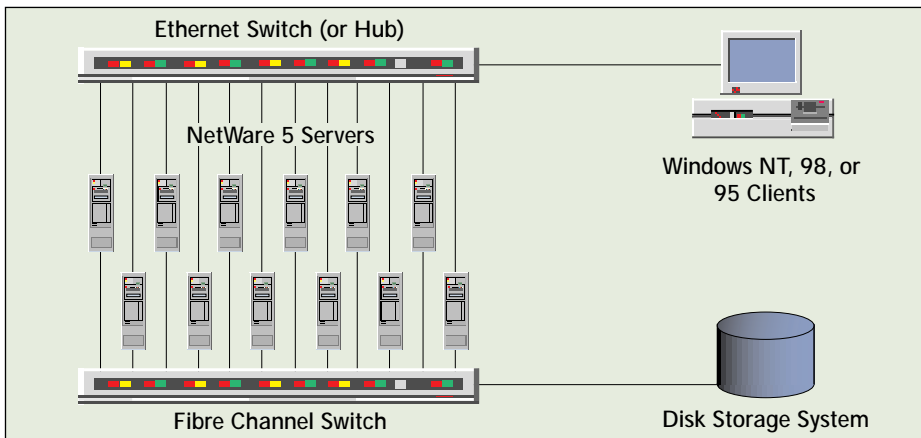


Figure 2. At BrainShare '99, Novell engineers demonstrated two 12-node configurations of clusters running NetWare Cluster Services in a SAN configuration.

SANs offer a number of benefits, including increased I/O scalability and storage consolidation. (Storage consolidation is the term used to refer to the ability of multiple servers to share a common storage area.)

Storage consolidation enables you to move disks between servers and also increases the availability of those resources. Of course, how easily you can move disks and the extent to which you increase availability depends in large part on the SAN management software you use.

If you use NetWare Cluster Services to manage a SAN-configured cluster, you can easily move disks between servers without disrupting access to server-based resources. NetWare Cluster Services also ensures that when one node fails, any of the surviving nodes can take over the failed node's workload: Any surviving node can access the exact same data—stored on the shared disk storage system—to which the failed node had access.

NetWare Cluster Services also supports non-SAN configurations, although Novell does not expect many companies to be interested in these configurations. In a few cases, however, a SAN may not be necessary. For example, if your company has two web servers and the data they present to clients changes very little, you may store copies of that static data on the servers' local drives. Wipfel explains, "If one server goes down, it wouldn't really matter because the data on the other server is essentially identical." Under these circumstances, you may opt to use NetWare Cluster Services in a non-SAN configuration to spare the expense of a disk storage system.

You can also use replication software, such as Novell Replication Services (NRS), to replicate files between nodes in a non-SAN configuration. However, replicated data is not synchronized exactly with original data. If the primary server fails, the secondary server will start accessing the replicated data, which will not be the same as the original. "For some applications, that's a bad thing," Wipfel says, noting that if you restart Oracle with an inconsistent copy of its files, Oracle will fail. "Applications like Oracle and GroupWise," Wipfel says, "actually require shared [storage] disks."

SANsense

With NetWare Cluster Services, you can use either Fibre Channel or serial

storage architecture (SSA) to connect a cluster to a disk storage system. (SSA is a high-speed interface specification that enables serial data transfers of 40 Mbps in full-duplex mode.) You can also use SCSI to connect small clusters to a storage subsystem.

DID YOU KNOW?

In a white paper titled "The Changing Costs of System Availability," Stratus Computer Inc. reports that an organization will lose a minimum of U.S. \$1,000 worth in user productivity if a 20-user system is unavailable for only 60 minutes. If the unavailable resource is a revenue-generating application, such as an airline reservation or credit card application, 60 minutes can cost your organization at least thousands and possibly millions of dollars. (You can download this white paper from http://www.stratus.com/docs/service/whitep/srap_1.htm#2.)

Novell engineers have tested NetWare Cluster Services using storage devices from several vendors, including the following:

- Compaq ProLiant Raid Array 8000
- Compaq FibreRaid Array
- IBM NetFinity EXP15
- XIOTech MAGNITUDE
- Data General Corp. CLARiiON
- Dell PowerVault 650F

Fail-Over Easy

When a node in a NetWare Cluster Services SAN configuration fails or is otherwise unavailable, any other node can take over all or some of the resources once provided by the failed node.

During the configuration process, you specify if NetWare Cluster Services should move the resources from a failed node to a surviving node or if NetWare Cluster Services should distribute the resources among several surviving nodes.

When surviving nodes take over resources for a failed node, the process is called a failover. Failovers typically take less than one minute, says Wipfel, and occur automatically when nodes unexpectedly fail. You can also manually respond to or invoke failovers when you need to upgrade or service a node.

If a node is temporarily unavailable (through either a manual or an automatic failover) and then rejoins the cluster, the cluster can restore itself to its original state. That is, all the resources can be returned to wherever they were located prior to the failover. This process is called a failback. Like a failover, a failback can occur automatically or manually.

END USERS: WHAT THEY DON'T KNOW . . .

When a failover occurs, users probably won't know about it. For example, suppose you configure a three-node cluster. (See Figure 3 on p. 12.) Further suppose that each node runs Netscape Enterprise Server for NetWare and hosts two web sites:

- Web server 1 hosts web sites A and B.
- Web server 2 hosts web sites C and D.
- Web server 3 hosts web sites E and F.

All of the data and graphics associated with the six web sites are stored on a storage device that is connected to the cluster via Fibre Channel. (See Figure 3.) You have configured the cluster so that if web server 1 fails, NetWare Cluster Services will move web site A to web server 2 and web site B to web server 3.

Now suppose that web server 1 experiences an abnormal end (otherwise known as the dreaded "abend"). What happens?

From the perspective of NetWare Cluster Services, a lot happens. NetWare Cluster Services running on web servers 2 and 3 notices that NetWare Cluster Services running on web server 1 is no longer transmitting heartbeat packets. NetWare Cluster Services on web servers 2 and 3 decides to read web server A's silence as a potential failure and begins the failover process.

The result of the failover process is that NetWare Cluster Services moves web site

A—and all applicable IP addresses—to web server 2. NetWare Cluster Services also moves web site B—and all applicable IP addresses—to web server 3.

From the users' perspective, nothing happens. The failover process happens so quickly users regain access to web sites A and B within seconds—and, in most cases, without having to log in again. Users notice only the typical hourglass on their monitor, indicating passing seconds.

KEEP THE SERVER SIDE UP!

Although you now understand what NetWare Cluster Services does, you may be wondering how it works. To understand how NetWare Cluster Services increases availability, you need to understand its architecture. (See Figure 4 on p. 14.)

You install NetWare Cluster Services on every node in the cluster. As Figure 4 shows, NetWare Cluster Services includes NetWare Loadable Modules (NLMs):

- Cluster Configuration Library (CLSTRLIB)
- Group Interprocess Communication (GIPC)
- Split Brain Detector (SBD)
- Virtual interface (VI) architecture Link Layer (VLL)
- Cluster Resource Manager (CRM)
- Trustee Migration (TRUSTMIG)
- Cluster Management Agent (CMA)

Your Local Library

The CLSTRLIB module loads a copy of the Novell Directory Services (NDS) cluster objects into local memory. When you install NetWare Cluster Services, the software creates a Cluster container object in your company's NDS tree. All of the cluster resources are represented as objects that are stored within this Cluster container. (For more information about NDS cluster objects and managing those objects, see "Cluster Control With Console-One" on p. 16.)

Each cluster object holds configuration information, which includes policies that you create when you install and configure NetWare Cluster Services. Among other things, these policies indicate where resources should run when the cluster is operating normally and where the resources should be relocated in the event of a node failure.

When you bring up the first node in a cluster, the CLSTRLIB module on that node accesses NDS and stores the cluster objects in memory. The first node then becomes the master node, and its CLSTRLIB module sends the cluster resource data to all other nodes that join the cluster. As a result, these nodes need not access NDS. (In fact, the CLSTRLIB module on the master node refers again to NDS only if you create a new resource or if you change the properties of an existing resource.)

The master node, in case you were wondering, does not represent a potential single point of failure. If the master node fails, the cluster detects that failure and simply elects a new master. The cluster elects as master the node with the highest numeric IP address.

GIPC Heartbeat

The GIPC (pronounced like the word *gypsy*) module runs the cluster group membership protocols, including the heartbeat protocol. As mentioned earlier, each node within a cluster transmits heartbeat packets over the Ethernet connection at regular intervals. For example, the default setting is once every second. Each node also continually checks for heartbeat packets from other nodes. When one node stops transmitting heartbeat packets for a defined time period, the other nodes detect this silence. (By default, this time period is eight seconds, but you can change this setting.)

After detecting a silent node, the other nodes figuratively talk amongst themselves to decide, as a group, whether or not they will interpret that silence as a failure. All of the surviving nodes must agree simultaneously that the silent node has failed. (The nodes use a two-phase commit algorithm to reach consensus.)

When the nodes collectively agree upon the possibility of a failure, the GIPC module on each node notifies the VLL module of this potential failure.

Avoiding Split Brain Blues

Before taking action on the message of potential failure from the GIPC layer, the VLL module first consults the SBD module. The SBD module detects and protects against split brains. A split brain refers to a bad cluster situation in which nodes form mutually exclusive views of the cluster's state and act accordingly.

For example, a split brain can occur if someone accidentally pulls the Ethernet cord from one of the nodes within a cluster, such as node A. After losing its Ethernet link, node A is unable to send or hear heartbeat packets. The GIPC modules on nodes B and C detect node A's silence, and conclude that node A is dead. As a result of this mistaken view of the cluster's state, nodes B and C attempt to restart node A's resources.

Likewise, the GIPC module on node A detects no heartbeat from nodes B and C and concludes that node A is the sole

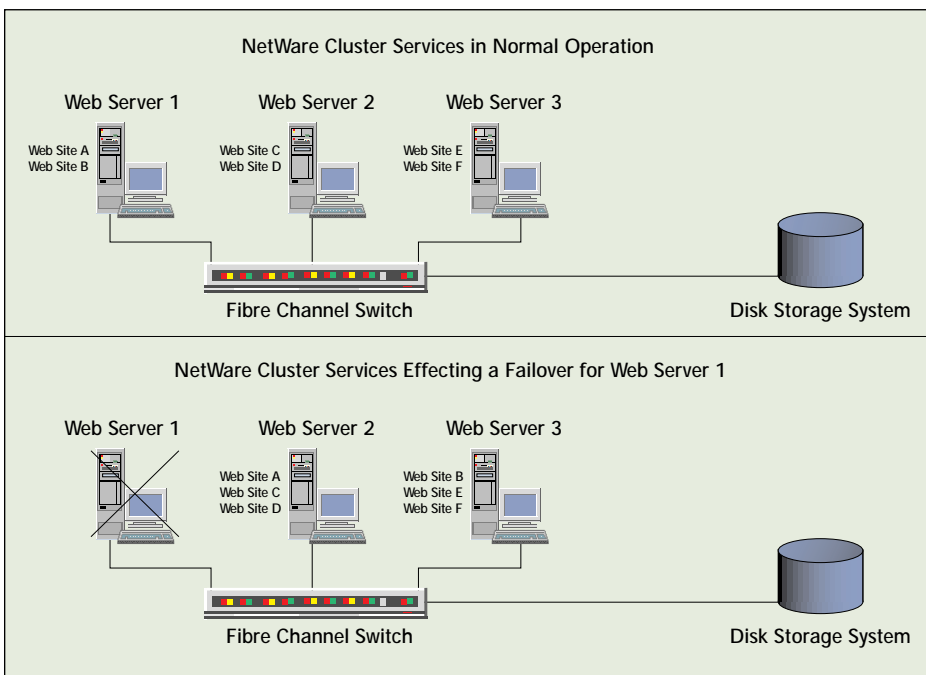


Figure 3. When one cluster node fails, the surviving cluster nodes take over the resources previously running on the failed node, following the failover policies you specified in NDS.

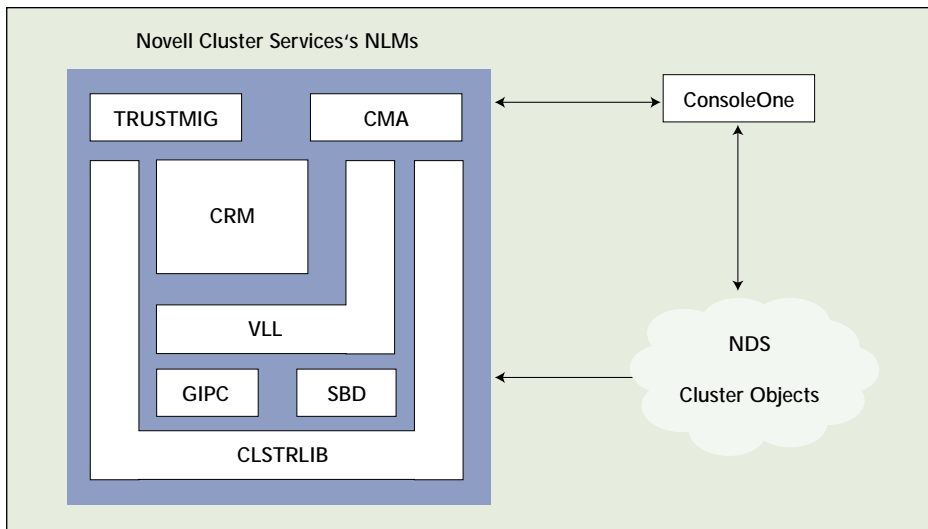


Figure 4. NetWare Cluster Services includes seven NLMs that interact with each other and with NDS via ConsoleOne to increase the availability of cluster resources.

survivor in what must have been a cluster catastrophe. As a result of this lonely view of the cluster's state, node A attempts to restart all of the cluster's resources itself.

Fortunately, the SBD module protects the cluster from split brains. However, the SBD module can only detect and protect against split brains if you are using NetWare Cluster Services in a SAN configuration.

When you are installing a SAN-configured cluster, the installation program detects the SAN and creates a special partition on the disk storage system. This partition essentially becomes an electronic white board. Each node reads from and writes to this white board, so the nodes can compare notes about their understanding of the state of the cluster. For example, when the VLL module notifies the SBD module that a failure might have occurred, the SBD module checks the white board (that is, the special partition on the shared disk storage system) for messages from the suspected failed node.

The Poison Pill

If the SBD module finds messages on the white board from the suspected failed node, the SBD module concludes that the node is still active. If there are no messages from the suspected failed node, the SBD module concludes that the node has in fact failed.

If the suspected failed node is actually still active and, therefore, the cluster has a split brain, the cluster acts swiftly (and brutally). Nodes on one side of the split

brain are allowed to continue living, but the other nodes must die. NetWare Cluster Services grants life to whichever side has the most nodes or, in the case of a tie, to the side where the master node resides.

The side of the split brain that will live knows it is winning, just as the side that will die knows it is losing. When the nodes on the losing side realize their position, they basically commit mass suicide by forcing abends. Novell engineers refer to this suicide process as "eating the poison pill."

If the SBD module informs the VLL module that the suspected failed node is in fact dead, NetWare Cluster Services still issues the poison pill—just in case the suspected failed node later becomes active. If a presumably dead node somehow regains life, this node consults the white board to see what's happening on the cluster. On the white board, it finds a message from the cluster, which may as well read "kill yourself now, please." As ordered, the dutiful node eats the poison pill by forcing an abend.

The cluster's actions sound brutal, Wipfel says, but if the cluster weren't brutal "what can happen, is much worse." What can happen, of course, is a split brain.

Finding and Restarting Resources

When the SBD module notifies the VLL module that a suspected failed node is active or that it is dead, the VLL module notifies the CRM module. If the message to the CRM module is "ding, dong the node is dead," the CRM module takes the next step toward the failover process.

The CRM module tracks where cluster resources are actually running and restarts the resources in the event of a failure, thereby ensuring that they are continually available. When you bring up the master node, its CRM module begins creating a data structure in local memory for storing information about the current runtime state of the cluster. The CRM module on the master node shares this runtime data structure with other nodes as they join the cluster. Thereafter, the CRM module on each node maintains this data structure.

When the VLL module notifies the CRM module that a failure has occurred, the CRM module first checks its runtime data to see whether or not the suspected failed node was running any resources. If the CRM module learns that the failed node was in fact running resources, the CRM module will execute the failover policies specified in the NDS configuration data, which the CLSTRLIB module read into local memory when you first installed the cluster. Among other things, these failover policies include a preferred nodes list. When a node fails, the node that is next on the preferred nodes list restarts the failed node's resources.

For example, extend the earlier example by supposing node A had been running the Oracle8 database. Further suppose that the preferred nodes list identifies node A, node B, and node C, in that order. This means that if node A fails, node B should restart node A's resources because node B is next on the list.

When the CRM module on node C learns that node A failed, it checks the cluster resource data in node C's memory, finds the preferred nodes list, and does nothing. (Node C knows node B is still alive and that node B is next on the preferred nodes list.) At the same time, the CRM module on node B checks the cluster resource data in node B's memory, realizes that it follows node A on the preferred nodes list, and accordingly, restarts the Oracle8 database previously running on node A.

Trustee Migration

If a data volume is also mounted on a failed node, the CRM module will remount that volume according to the failover policies. For example, suppose the failover policies indicate that node C should remount the data volume on node A if node A fails. When the CRM

Cluster Control With ConsoleOne

ConsoleOne is a Java-based management tool with a GUI that enables you to manage many (and eventually all) Novell products from a single workstation running the appropriate product-specific snap-in modules. For example, for the demonstrations at Brain-Share '99 in Salt Lake City, Novell engineers ran a beta version of the ConsoleOne snap-in modules for NetWare Cluster Services for NetWare 5 on a Windows NT workstation.

NEW NDS OBJECTS FOR NETWARE CLUSTER SERVICES

By running the ConsoleOne snap-in modules for NetWare Cluster Services on a Windows NT or 95 workstation, you can configure and manage all of the nodes and resources in a cluster. These nodes and resources are represented as objects in the Novell Directory Services (NDS) tree. Specifically, NetWare Cluster Services defines the following objects to represent the cluster:

- Cluster container object
- Cluster Node objects
- Cluster Resource objects
- Cluster-enabled Volume objects
- Cluster Resource Template objects

The Cluster Resource Template objects contain information you use to create new Cluster Resource objects. If you use these template objects, you do not have to reenter the same information every time you create a new Cluster Resource object.

Using the ConsoleOne snap-in modules for NetWare Cluster Services, you can manipulate the data for any of the cluster objects. For example, you can configure, control, and monitor Cluster Resource objects in any of the following ways:

- Create or modify load and unload scripts and timeouts for applications
- View the current location of resources
- Add, move, or remove resources
- Take resources online or offline
- Specify failover and failback policies
- Respond to or invoke manual failovers or failbacks

VIEWS IN CONSOLEONE

When you view a cluster in ConsoleOne, you will see a split screen. (See Figure 5 on p. 18.) The left side of the ConsoleOne screen is called the Tree View and shows the NDS tree. The right

side of the ConsoleOne screen is called the Console View. Assuming you have selected Console View from the pull-down menu that appears when you select View from the toolbar at the top of the screen, the Console View shows the objects within the container object you select in the Tree View.

You may also select Cluster View from the View pull-down menu. (See Figure 5 on p. 18.) Among other things, the Cluster View shows you the number and names of cluster nodes, the percentage of nodes available, and the percentage of resources available. In addition, the master cluster node is identified with a yellow dot. Figure 5, for example, shows the yellow dot on node 2. If node 2 fails, another node takes over its function as master node, and the yellow dot on the Cluster View reflects this change.

In the event of a failure, the Cluster View also shows you the state and location of resources that were affected during a failover. For example, Figure 5 shows that GroupWise is running on node 3 and has had two lives, which means that it has been restarted or experienced failover twice. The MONITOR.NLM is running on node 1 and is still on its first life. The VOL0_SERVER is running on node 2 and has been restarted four times.

The Cluster View also reports the state of cluster nodes and resources. For example, the cluster nodes in Figure 5 are "Running." (Other examples of states include offline, loading, unloading, and unassigned.)

From the Cluster View screen, you may also select the Event Log and HTML Report tabs. Event Log shows a table of cluster events, such as a resource changing from loading to running state, in the order in which these events occurred. The HTML Report shows a full report of the cluster state at the time the tab is clicked. The HTML Report also shows the properties of all cluster resources and node objects. You can save the HTML Report to a text file and view it later using a web browser.

NDS INTEGRATION

NetWare Cluster Services's full integration with NDS makes it unique from other cluster solutions. That is, with NetWare Cluster Services, you have only one copy of management data for your entire cluster, regardless of the number of nodes. Some clustering solutions force you to create and change configuration files once per cluster node. If you needed to edit a file and have 32 nodes, you would have to make the same change on each node.

In contrast, NetWare Cluster Services has only one copy of its management data. If you need to edit a NetWare Cluster Services file and have a 32-node cluster, you make the change only once in ConsoleOne. ●

module on node C learns that node A failed, this CRM checks the cluster resource data in node C's memory. Because the failover policies indicate that node C is responsible for remounting node A's data volume, the CRM on node C finds the volume in the shared disk storage system and remounts it.

When the CRM moves a data volume from one node to another, the TRUSTMIG module ensures that the NDS trustee rights on that volume are

translated so that they will be valid on the new node.

The Cluster Commissioner

The CMA module interfaces with the management console, ConsoleOne. ConsoleOne enables you to manage the entire cluster, regardless of the number of nodes, from one location. You can manage a cluster from any workstation running the ConsoleOne snap-in modules for NetWare Cluster Services. (For more information

about managing NetWare Cluster Services with ConsoleOne, see "Cluster Control With ConsoleOne.")

CLUSTER NLMS IN ACTION

To solidify your understanding of NetWare Cluster Services, consider a final example of how its seven modules work. Suppose you have a three-node cluster with nodes A, B, and C in a SAN configuration. Further suppose that GroupWise 5.5 is running on node A. The policies

you created in NDS dictate that when node A fails, node B should restart GroupWise 5.5 and remount node A's data volume, which is located on the shared disk storage system.

In the middle of a typical workday, the SYS: volume on node A inexplicably fails, causing node A to fail. NetWare Cluster Services immediately affects a failover.

When node A fails, the GIPC modules on the surviving nodes, nodes B and C, notice that node A has not transmitted heartbeat packets during the defined time period (by default, eight seconds). The GIPC modules on nodes B and C exchange messages with their respective VLL modules regarding this suspected failure.

The VLL modules send a message to the SBD modules, which check the SAN and find that node A is no longer writing messages to the white board. The SBD modules send a message to the VLL modules, confirming that node A has failed. After nodes B and C reach consensus regarding that failure, the VLL modules on nodes B and C simultaneously send the notice of node A's failure to the CRM modules.

The CRM modules on nodes B and C check the cluster resource data that the CLSTRLIB read from NDS when the cluster first started. These CRM modules find, at the same time, that node B is responsible for restarting node A's resources. Accordingly, node C does nothing, and node B begins the failover process by running the load script for GroupWise 5.5. This load script would contain commands to start the GroupWise post office agent and the message transfer agent.

State Change Messages

Before running the load script for GroupWise 5.5, node B sends messages to all of the other nodes in the cluster (in this case, node C), explaining that node B is loading GroupWise 5.5. The moment that GroupWise 5.5 is running, node B sends another message to all other nodes informing them of this fact.

These state-change messages are sent to all nodes in a cluster before changes are made to any cluster resource. All nodes must then commit to the change by using the same two-phase commit algorithm that the GIPC module uses to agree upon a suspected failure. That is, the state change has to be made by all of the nodes in the cluster at the same time and can never be made on just one node alone.

In this way, NetWare Cluster Services ensures that all nodes are always aware of the current state of the cluster and its resources. NetWare Cluster Services also ensures that one node cannot attempt to relocate a resource without committing that change across the entire cluster.

Because all nodes are always aware of the state of the cluster and its resources, all nodes know exactly what is happening if there is a failure during a failover process. For example, by apprising node C of what node B is doing, node B ensures that if something goes wrong while it is running the load script for GroupWise 5.5, node C will know exactly what has happened and when—and will be able to take over.

And the Workday Continues

GroupWise 5.5 is now running on node B, as is node A's GroupWise post office

volume, which node B remounted as dictated in the failover policies. The TRUSTMIG module has ensured that all NDS trustee rights previously associated with node A are now valid on node B.

If you do not have a cluster and your company's GroupWise server fails, you will probably spend several hours restoring GroupWise 5.5 on another server. You will also have to respond to countless complaints from panic-stricken users whose productivity depends in part upon the ability to send and receive e-mail.

If you have NetWare Cluster Services, however, users won't even notice if node A fails. NetWare Cluster Services moves the IP address previously associated with node A to the failover server, node B. The GroupWise 5.5 client software simply uses the same IP address to reconnect to what appears to be the same GroupWise server. After the failover is completed—which occurs in seconds—users are reading their e-mail messages without having to log in again and without having lost anything. Your users' workday continues as usual—without a hitch.

LIKE YOU'VE NEVER SEEN

The examples used in this article highlight the benefits of automatic failovers, but the ability to invoke manual failovers is equally important. After all, users don't care whether a server failed unexpectedly or was taken down intentionally. Users only want to know how long you will take to restore the server-based resources they use. Brian Howell, Novell product marketing manager, puts it succinctly—uninterrupted access to applications is "all end users care about."

Using the ConsoleOne snap-in modules for NetWare Cluster Services, you make a "simple click operation" to move an application from one node to another, Howell says. This kind of flexibility is extremely useful when you need to service or upgrade a server or when you need to divvy up a server's workload because the server is nearing the maximum level of its CPU utilization.

Whatever your reasons for moving resources, NetWare Cluster Services enables you to move these resources with ease. "This is flexibility like you've never seen," says Howell. And peace of mind like you've never known.

Linda Kennard works for Niche Associates, a technical writing and editing firm located in Sandy, Utah. ●

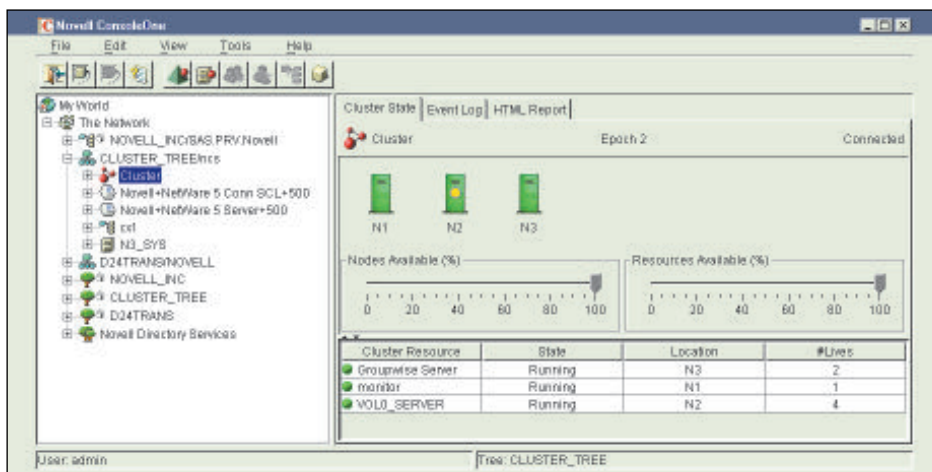


Figure 5. Using the ConsoleOne snap-in modules for NetWare Cluster Services, you can view the current state of the cluster, including which cluster node is master (indicated with the yellow dot) and the percentage of nodes and resources available on the cluster.