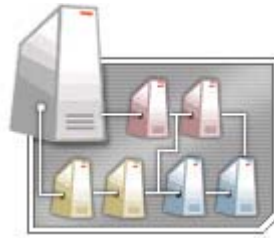


Windows Server 2003 Clustering Features



Windows Server 2003 clustering consists of two different technologies: Server Cluster and Network Load Balancing (NLB). Each of these technologies can be used to provide high availability for different types of services. Server Cluster is primarily used to provide availability for mission critical applications through fail-over. Database, ERP or CRM, OLTP, file and print, e-mail, and custom application services are typically clustered using Server Cluster. NLB is used to provide high availability for applications that scale out horizontally, such as Web servers, proxy servers, and other services that need client requests distributed across nodes in a cluster.

Table 1 compares the two sets of clustering features.

Table 1: Server Cluster and NLB compared

Server Cluster	NLB
Used for databases, e-mail services, line of business (LOB) applications, and custom applications	Used for Web servers, firewalls, and Web services
Included with Windows Server 2003, Enterprise Edition, and Windows Server 2003, Datacenter Edition	Included with all four versions of Windows Server 2003
Provides high availability and server consolidation	Provides high availability and scalability
Can be deployed on a single network or geographically distributed	Generally deployed on a single network but can span multiple networks if properly configured
Supports clusters up to eight nodes	Supports clusters up to 32 nodes
Requires the use of shared or replicated storage	Doesn't require any special hardware or software; works "out of the box"

Server Cluster

Server Cluster is a dramatically improved version of the Microsoft Cluster Service (MSCS) component included with Windows 2000 Advanced Server and Windows 2000 Datacenter Server. When you deploy Server Cluster, you first configure it between two and eight servers that will act as *nodes* in the cluster. Then you configure the cluster resources that are required by the application you're clustering. These resources may include network names, IP addresses, applications, services, and disk drives. Finally, you bring the cluster online so that it can begin processing client requests.

Most clustered applications, and their associated resources, are assigned to one cluster node at a time. If Server Cluster detects the failure of the primary node for a clustered application, or if that node is taken offline for maintenance, the clustered application is started on a backup cluster node. Client requests are immediately redirected to the backup cluster node to minimize the impact of the failure.

Note: Though most clustered services run on only one node at a time, a cluster can run many services simultaneously to optimize hardware utilization. Some clustered applications may run on multiple Server Cluster nodes simultaneously, including Microsoft SQL Server.

Nodes in a cluster use a *quorum* to track which node owns a clustered application. The quorum is the storage device that must be controlled by the primary node for a clustered application. Only one node at a time may own the quorum. When an application fails over to a backup node, the backup node takes ownership of the quorum. When the cluster nodes are all attached to a single storage device, the quorum may be created on the storage device. This type of cluster is called a *single quorum device server cluster* when built with Windows Server 2003. Figure 1 shows a four-node single quorum device server cluster.

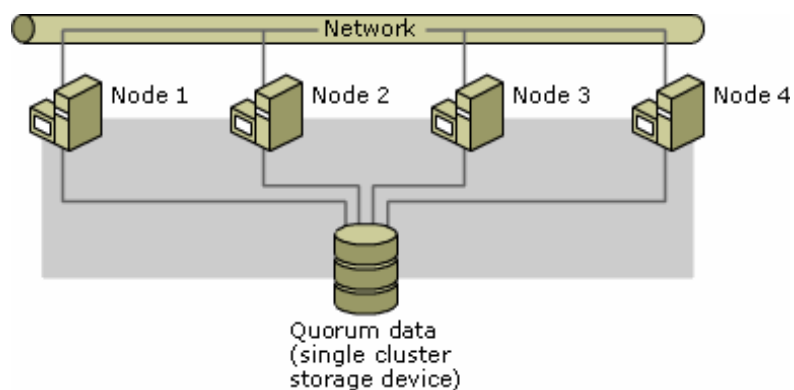


Figure 1: An example of a single quorum device server cluster

Connecting all nodes to a single storage device simplifies the challenge of transferring control of the data to a backup node. However, this architecture has weaknesses. If the storage device fails, the entire cluster fails. If the storage area network SAN fails, the entire cluster fails. Both the storage device and the SAN can

be designed with complete redundancy, but there's one component in this architecture that will never be redundant—the facility. Floods, fires, earthquakes, extended power failures, and other serious problems will cause the entire cluster to fail. If your business continuity requirements mandate that work continue even if a facility is taken offline, a single quorum device server cluster solution alone won't meet your needs.

Majority node set (MNS) server clusters store the quorum on a locally attached storage device connected directly to each of the cluster nodes. Of course, for a backup node to assume control of the quorum, the backup node must have a copy of the data stored within the quorum. Server cluster handles this requirement by replicating quorum data across the network. As Figure 2 shows, majority node set clusters require only that the cluster nodes be connected by a network. That network doesn't need to be a local area network (LAN), either. It can be a wide area network (WAN) or a virtual private network (VPN) connecting cluster nodes in different buildings or cities—allowing a cluster to overcome geographic restrictions imposed by the storage connections.

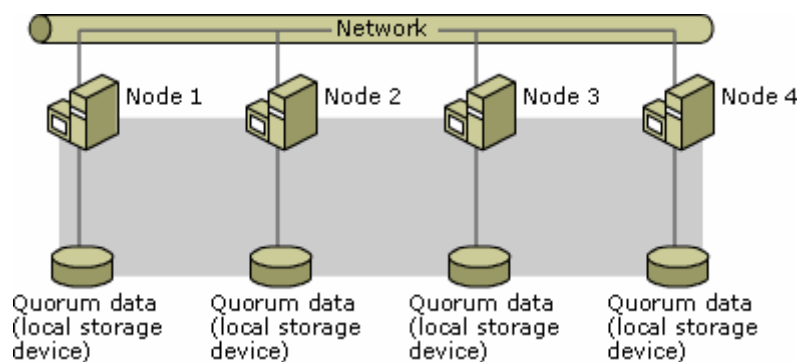


Figure 2: An example of a majority node set server cluster

Majority node set clustering does have requirements that single quorum device server clusters lack. To effectively fail over between nodes, majority node set clusters must have at least three nodes. More than half of the cluster nodes must be active at all times. So, if you design a cluster with three nodes, two of them must be active for the cluster to be functional. Eight node clusters must have five nodes active to remain online. Single quorum device server clusters require that only a single node continues to function.

Windows Server 2003, Enterprise Edition, and Windows Server 2003, Datacenter Edition, support Server Cluster on both 32-bit and 64-bit server platforms.

Network Load Balancing

The second clustering technology included with Windows Server 2003 is NLB. NLB clusters don't use a quorum, and so don't impose storage or network requirements on the cluster nodes. If a node in the cluster fails, NLB automatically redirects incoming requests to the remaining nodes. If you take a node in the cluster offline for

maintenance, you can use NLB to allow existing client sessions to be completed before taking the node offline. This eliminates any end-user impact during planned downtime. NLB is also capable of weighting requests, which allows you to mix high-powered servers with legacy servers and ensure all hardware is efficiently utilized.

Most often, NLB is used to build redundancy and scalability for firewalls, proxy servers, or Web servers, as illustrated in Figure 3. Other applications commonly clustered with NLB include virtual VPN endpoints, streaming media servers, and terminal services. NLB is included with all versions of Windows Server 2003, including Windows Server 2003, Web Edition. NLB clusters can scale to 32 nodes.

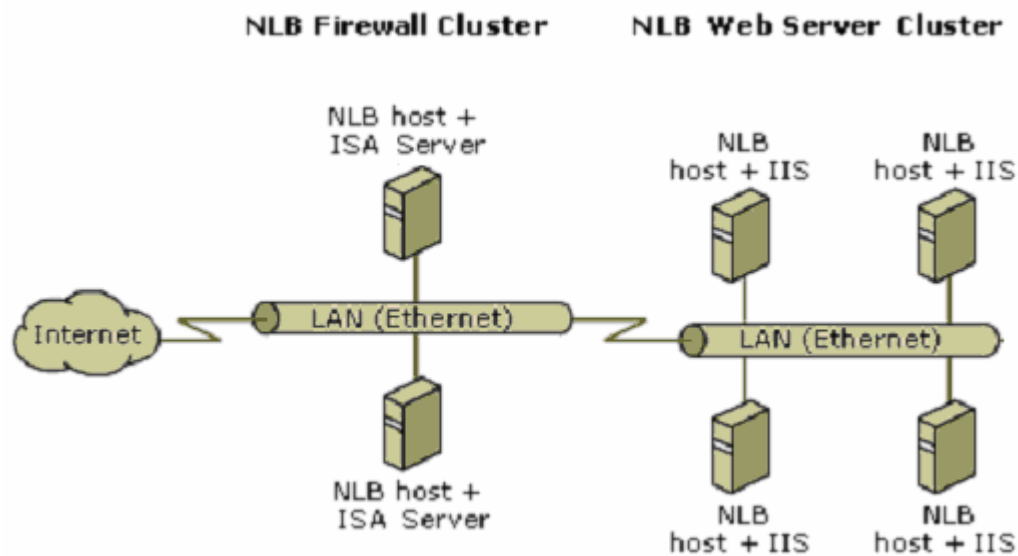


Figure 3: An example of a network load balancing cluster

Meeting Your Availability Goals

The primary reasons organizations make use of clustering are to provide application availability and data integrity and to reduce costs associated with downtime. These costs may be incurred because of the reduced end-user productivity or lost business opportunities. If you offer Service Level Agreements (SLAs) to your customers and are unable to meet your SLAs because of extended downtime, the costs may be even more tangible.

Many different types of events can cause downtime. You should take the likelihood and seriousness of each type of event into account when designing architectures. Common events include the following:

Planned downtime due to upgrade, service pack, or security patching

Unexpected hardware and software failures

Infrastructure failures and natural disasters

The sections that follow discuss ways you can use Windows Server 2003 clustering features to reduce downtime in these scenarios and ways that you can ensure even custom applications stay online.

Patching and Upgrades

Patching and upgrading are key parts of managing systems and, as you're aware, it's more than a little likely that your systems will need patches and upgrades applied during their life span. Unfortunately, updating software on critical systems requires planned downtime, because most updates require the service or the server to be restarted. Any type of update carries a risk of extended downtime, because the update may be incompatible with critical applications or may introduce a problem that doesn't allow the system to start.

Server Cluster and NLB can be used to completely eliminate downtime associated with deploying patches by using rolling upgrades. At a high level, you perform a rolling upgrade by taking redundant cluster nodes gracefully offline, upgrading them, and then bringing them back online. This process is repeated until all cluster nodes have been successfully upgraded.

Hardware and Software Failures

When most people think about downtime, the first thing that comes to mind is hardware failures. Failed disks, memory, processors, power, and network equipment are all common sources of unplanned downtime. Both Server Cluster and NLB can be used to provide availability in the event of a failure of a processor, memory chip, power supply, or other hardware component. However, Windows Server 2003 clusters can be designed to provide availability at many other layers. By connecting cluster nodes to separate network equipment and UPSs (Uninterruptible Power

Supplies), Windows Server 2003 clusters can ensure that critical services continue even when infrastructure that the server depends on fails.

Critical applications are rarely hosted by a single server. More often applications consist of multiple layers. For example, many applications have layers for the firewalls, application servers, and a database on the back end. To provide complete redundancy, all layers of your application must be clustered. As Figure 4 shows, Windows Server 2003 clustering is capable of providing availability for all of these layers. In the three-layer architecture described above, NLB provides availability and scalability for the firewalls, front-end servers, and application servers, while Server Cluster provides high availability for the database.

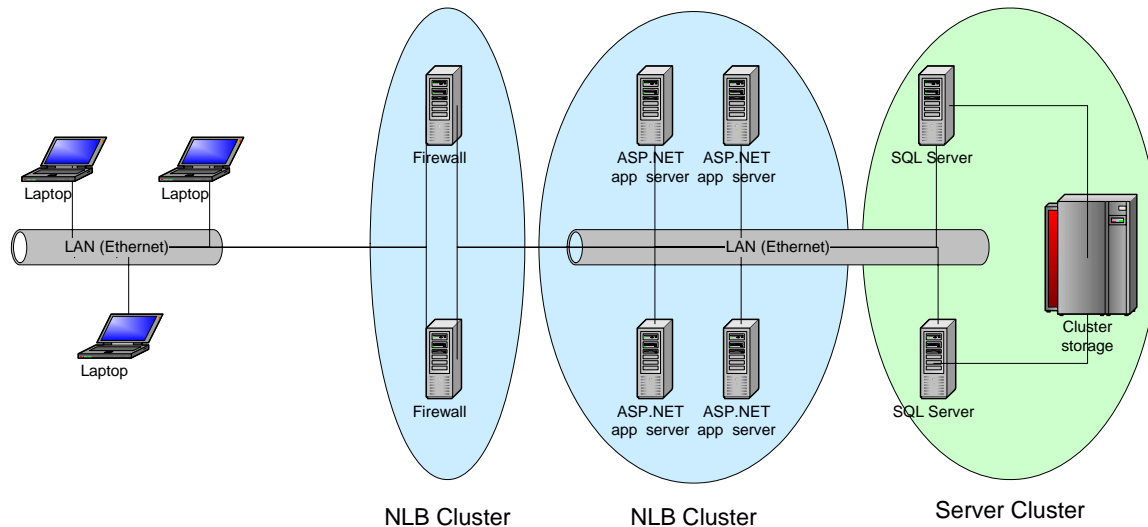


Figure 4: Windows Server 2003 provides clustering for all layers of your mission-critical applications

Availability is directly related to capacity planning. When you are designing clusters, plan to meet peak capacity requirements with one failed node. If you're designing a two-node cluster, you need to ensure that each node is less than 50 percent utilized during peak time to that ensure a single surviving node can handle all requests. The fewer nodes in the cluster, the more otherwise unused computing resources must be dedicated to providing capacity in the event of a fail-over. To ensure redundancy while providing high efficiency, Windows Server 2003 supports eight-node server cluster, requiring as little as 12 percent excess capacity to ensure availability. NLB supports up to 32 nodes in a cluster, which can provide availability with less than 4 percent excess capacity.

Natural Disasters, Power Outage, or Terrorist Attacks

"Clusters are great—but redundancy within a data center doesn't meet my BC requirements."

When you build a cluster in a single data center, you can provide redundancy for just about everything except the data center. If there's a natural disaster, a WAN infrastructure failure, or the entire data center loses power for an extended period of time, all cluster nodes will be offline. Fortunately, Windows Server 2003 provides Majority Node Set clustering. Majority Node Set clustering allows cluster nodes to be connected by using a LAN or WAN instead of a SAN or other type of storage connection with distance limitations. Any network that can provide guaranteed round-trip latency of less than half a second will work—even a VPN. Figure 5 illustrates a three-node majority node set cluster with a VPN interconnection.

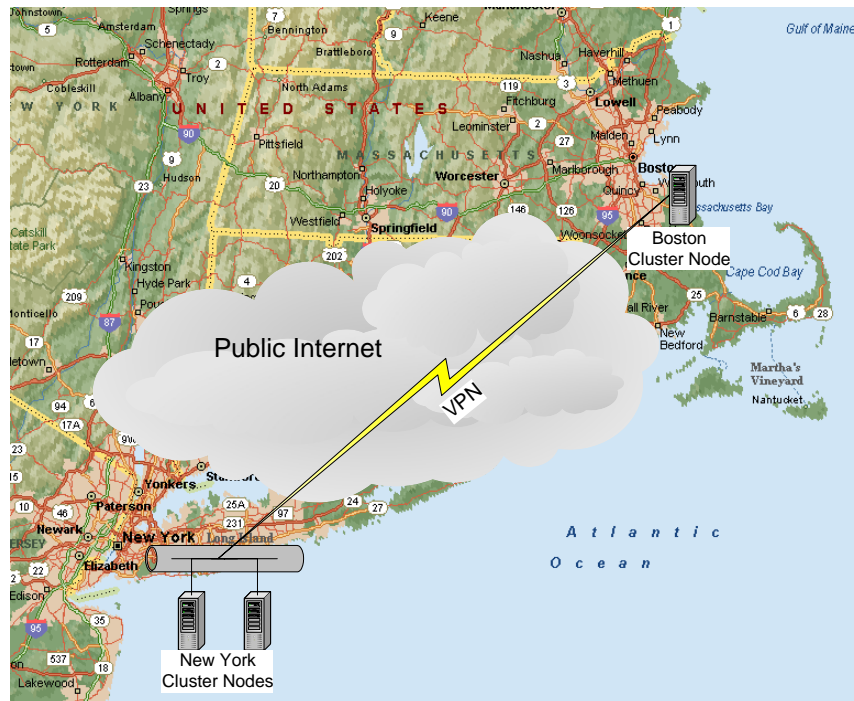


Figure 5: Majority node set clusters can be geographically distributed

One of the challenges of geographically distributed clusters is avoiding the split-brain phenomenon. Backup nodes in a cluster take control of clustered resources when the primary node appears to have failed. However, network connectivity failures can make each node in a cluster think the others have failed. If such a failure occurred on a cluster not built on Windows Server 2003, a backup node might start a clustered application at the same time as the primary node—causing a split brain.

Majority node set clustering prevents split brain by requiring that a majority of the cluster nodes agree on which node controls a clustered resource. In Figure 5, if the network connection to the primary node in the Boston office fails, both of the backup nodes will detect this failure, agree that the primary node is offline, and assign a node to take control of the quorum and start the clustered application. Meanwhile, the primary node will detect that it cannot communicate with the two backup nodes. Because the primary node cannot communicate with a majority of the nodes, it will take itself offline. The Windows Server 2003 cluster will successfully fail over in this

complicated scenario, without possibility of a split brain. Business can continue as usual from the New York offices. When the Internet connection in Boston has stabilized, you can manually fail back the cluster services to the Boston node.

If the default behavior doesn't meet your needs, you can choose to manually control the failover, too.

Majority node set clustering replicates the quorum, but is not responsible for replicating the application data. However, Microsoft SQL Server and most other modern databases provide replication mechanisms suited for synchronizing data across a WAN. Majority node set clustering also works with storage-based replication mechanisms.

Geographically distributed clusters require more than just software, of course. The Microsoft Geographic Cluster qualification program requires vendors to list supported solutions on the Hardware Compatibility List (HCL). Microsoft performs rigorous checks for various failure cases to ensure data integrity and to ensure that the cluster guarantees are always met.

Clustering Custom Applications

“Sure, I can cluster SQL—but it doesn't do me any good unless I can cluster my home-grown front-end app, too.”

Until now, most clusters were built around infrastructure services such as Microsoft SQL Server and Microsoft Exchange Server. Unfortunately, clustering those back-end services only provides part of the solution. In order to provide true business continuity, you must provide redundancy for the applications you've built on top of the infrastructure. For example, if you have a homegrown application that front-ends the clustered back-end SQL Server database, your end users will still be stranded if your homegrown application fails.

To help you provide availability, server cluster includes three resource types specifically designed for custom applications:

Generic Application

Generic Script

Generic Service

As Figure 6 shows, the New Resource Wizard walks administrators through the process of configuring an existing application as a clustered resource. Most applications require that other resources, such as an IP address, are failed over at the same time as the clustered application. The New Resource Wizard makes it simple to add these.

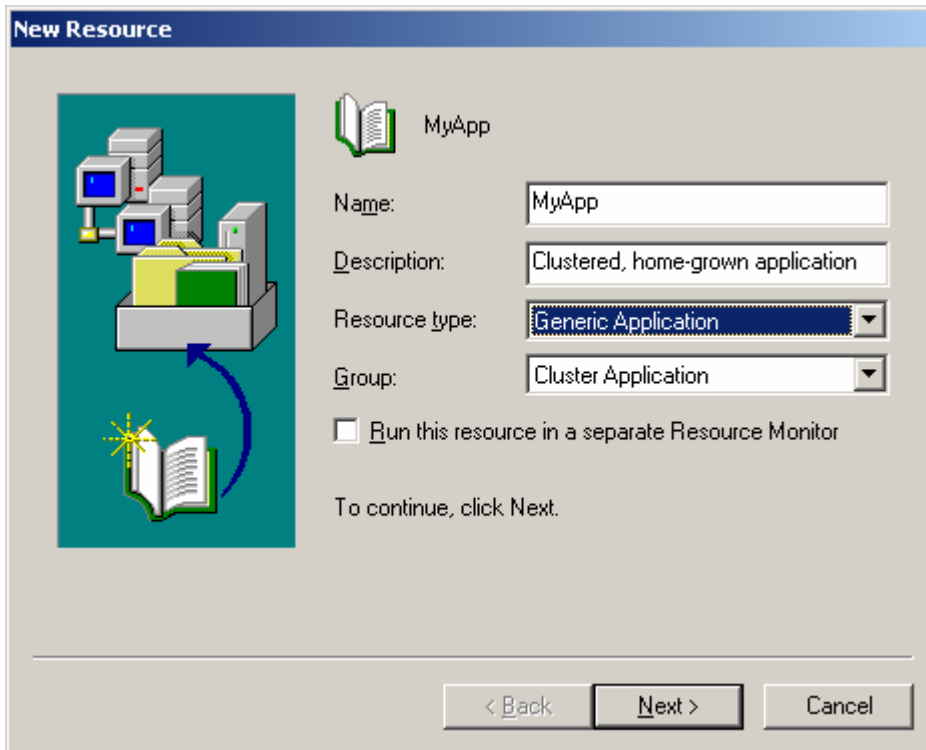


Figure 6: The New Resource Wizard makes it simple to add custom applications

For example, clients probably communicate with the custom application at a specific IP address. Creating this clustered IP address is as simple as using the New Resource Wizard and selecting **IP Address**, and then specifying the IP address and subnet mask that will be shared between systems. You may add several other resources for your clustered application, possibly including a network name to identify your custom application on the network, and a majority node set quorum to store and replicate the custom application's data. You can also replicate configuration information contained within the registry.

To take advantage of all the benefits of Server Cluster, your developers can extend the Windows Server 2003 clustering platform to meet your needs. Developers can call clustering functions from an application, service, or script. This includes writing detailed troubleshooting information to the cluster log, performing an automated cluster-wide setup, and even creating custom cluster resource types.

If your custom front end is a Web application or Web service, use NLB to provide redundancy and scalability.

Meeting Your Scalability Needs

Everyone needs to plan for future growth. When planning information systems to last several years, you can conserve capital by designing small, inexpensive systems that can quickly scale to meet demand. Planning for future scaling reduces the cost and risk of deploying systems and lets you keep your infrastructure costs tied closely to your growth.

NLB supports the philosophy of starting small and scaling capacity only as needed. In that way you can make conservative estimates of utilization for new services and scale them only after establishing that the demand is real. For example, if you're deploying a new Web service, it's difficult to anticipate your capacity needs accurately because there are many significant questions that are difficult to answer precisely, such as:

How many users will the Web service have?

How many requests will each user submit?

What will requests be at peak time?

What computing resources will the Web service require to serve each request?

Perhaps you've been considering using Web services technology to streamline supply chain management with your vendors. While you know there are significant efficiencies to be realized, you're not sure your vendors are tech-savvy enough to take advantage of the new Web service, so you'd benefit by keeping your initial capital investment limited. You could start with a low-end server: a single processor Windows Server 2003, Standard Edition. If the Web service becomes popular, or requires more computing resources than initially estimated, you can add an additional server and configure the NLB feature to scale the service to additional servers. Adding additional servers scales capacity almost linearly, and provides availability, too. If the Web service isn't as popular as expected, then you've avoided wasting money on unnecessary hardware and software.

While NLB is perfect for scaling Web applications, not all applications can be scaled horizontally. Windows Server 2003 provides vertical scalability for clusters based on Server Cluster, too. Each cluster node can support up to 64 processors and 512 gigabytes (GB) of RAM with the 64-bit version and up to 32 processors and 64 GB of RAM with the 32-bit version on Windows Server 2003, Datacenter Edition. Storage won't be a problem either, because Windows Server 2003 provides native support for Fibre Channel SANs.

Meeting Your Manageability Requirements

“Thanks to the recent reductions in force (RIFs), my IT staff doesn’t have the time to support additional clusters.”

In the past, many organizations chose not to deploy clusters because the systems administration staff didn’t have the spare cycles to go to training to deploy the cluster, not to mention the additional time performing day-to-day maintenance on the cluster. Server Cluster and NLB require minimal training because the clustering components are part of the operating system and have an easy-to-use, wizard-based interface, which Figure 7 shows.



Figure 7: Almost all administration tasks are wizard-based

Ongoing administration requirements are minimal. In fact, you don’t even need to develop a new process to manage the patching or monitoring of Windows Server 2003’s clustering features, as you would if you used clustering services available from a third party as an add-on to the operating system. Patches for the clustering components will be delivered to customers through the Microsoft standard update distribution process. You can monitor the cluster components just as you would monitor any other Windows service, so your existing monitoring systems will work with little or no modification.

If you plan to deploy many clusters, you’ll be pleased to know that there are command-line tools for managing almost every aspect of Windows Server 2003,

including the cluster components. The Windows Management Instrumentation Command-line (WMIC.exe) tool provides an interface to almost every imaginable function useful for deployment and managing servers. The Cluster.exe tool allows for automating of just about any cluster task, including adding and removing cluster nodes, analyzing cluster configuration, and configuring clustered applications. Your staff can use Cluster.exe and WMIC.exe to deploy dozens of clustered applications to remote sites with minimal effort.

Backing up and restoring clustered application data has always been one of the challenges of supporting clusters. After all, how does the backup system know which cluster node controls a clustered resource? This isn't a concern with Windows Server 2003—you can back up cluster data using any standard backup tool, including the Backup utility included freely with the operating system. You can even take advantage of volume snapshot services, which provide a point-in-time backup of a volume or cluster quorum. Recovering from serious problems is simplified, too, because Automated System Recovery (ASR) allows administrators to quickly recover a corrupted server—including cluster servers.

“I worked with clusters before, but won't try them again until the technology matures. They took so long to troubleshoot that they had more downtime than a standalone server.”

If problems do occur, troubleshooting is simplified by the detailed logging generated by the clustering services. By analyzing these logs, a systems administrator can isolate complex problems in a matter of minutes. Without this detailed logging, administrators would spend hours performing trial-and-error troubleshooting to identify the source of the problem. The quicker a problem can be resolved, the shorter the downtime—and that's what clusters are all about.

Meeting Your Cost Goals

“The numbers don’t add up. It costs me more to build and support clusters than the additional uptime will save the business.”

Availability, scalability, and manageability are all factors of cost. Clusters should save you money, not cost you money. If the benefits of increased availability and capacity provided by clustering aren’t greater than the costs, it’s not worthwhile for you to deploy clusters.

To extend the benefits of clustering to as many organizations as possible, Microsoft has designed the Windows Server 2003 clustering technologies to be the lowest-cost solutions of their type. First, the initial acquisitions costs are low because clustering components are included with the operating system. If you plan to deploy an NLB cluster, it’s included free with Windows Server 2003, Web Edition, Standard Edition, Enterprise Edition, and Datacenter Edition. If you plan to deploy a Server Cluster, you’ll need to buy Enterprise Edition or Datacenter Edition; no other software is necessary.

There’s no extra software to buy for management or monitoring of the cluster technologies, because it’s all included with Windows Server 2003. If you already have an enterprise network management solution, you can easily tie the clustering technologies into your existing system. The tight integration with the operating system and standard Windows patch management process keeps maintenance costs low—a surprisingly significant part of the cost of competitive solutions. The straightforward user interface and improved maturity of the software means you won’t need to pay consulting services just to get it up and running.

Hardware costs are lower than other clustering solutions, too, because Windows runs on industry-standard low-cost hardware, unlike much more expensive reduced instruction set computer (RISC)-based systems. You won’t have to buy significant amounts of redundant hardware, either, because Windows Server 2003 supports Server Cluster with up to eight nodes and NLB with up to 32 nodes. This enables you to provide redundancy with as little as 1/8 or 1/32 of the total cluster capacity available.

If you plan to cluster a Web application or firewall, there are ways to do it without using NLB—but they’re all much more expensive. Many organizations have used dedicated hardware devices to load-balance traffic between multiple servers. However, this is costly. Not only would you need to buy the load-balancing device, but to ensure availability you would also need to buy at least two load-balancing devices per cluster. Then you would need to train your staff on how to deploy and manage the cluster. You would also need to customize your monitoring and backup systems to support the new device. With NLB, you don’t need to purchase any additional hardware or software, and training costs are next to nothing. You can even deploy multiple Web sites and Web services using a single cluster to achieve the highest level of performance and efficiency.

Summary

“Clusters are all marketing promises. When the rubber meets the road, they just don’t work like they’re supposed to.”

The clustering services included with Windows Server 2003 are the result of years of real-world experience building Microsoft clusters. As a result, Windows Server 2003 clusters work in today’s real world of reduced budgets and limited manpower. Windows Server 2003 clusters save you more than they cost you, and they do more than merely provide availability—they fulfill your business continuity requirements. The latest generation of Microsoft clusters is inexpensive to deploy, yet capable of growing quickly to keep up with demand. Windows Server 2003 clustering works in *your* world.

Technical Overview of Clustering in Windows Server 2003

Server Clusters

NOTE: Server clusters is a general term used to describe clusters based on the Microsoft® Cluster Service (MSCS), as opposed to clusters based on Network Load Balancing.

General

Larger Cluster Sizes

Microsoft Windows® Server 2003 Enterprise Edition now supports 8-node clusters (was two), and Windows Server 2003 Datacenter Edition now supports 8-node clusters (was four).

Benefits

Greater Flexibility – this provides much more flexibility in how applications can be deployed on a Server cluster. Applications that support multiple instances can run more instances across more nodes; multiple applications can be deployed on a single Server cluster with much more flexibility and control over the semantics if/when a node fails or is taken down for maintenance.

64-Bit Support

The 64-bit versions of Windows Server 2003 Enterprise Edition and Datacenter Edition support Cluster Service.

Benefits

Large Memory Needs – Microsoft SQL Server™ 2000 Enterprise Edition (64-bit) is one example of an application that can make use of the increased memory space of 64-bit Windows Server 2003 (up to 4TB – Windows 2000 Datacenter only supports up to 64GB), while at the same time taking advantage of clustering. This provides an incredibly powerful platform for the most computer intensive applications, while ensuring high availability of those applications.

NOTE: GUID Partition Table (GPT) disks, a new disk architecture in Windows Server 2003 that supports up to 18 exabyte disks, is not supported with Server clusters.

Terminal Server Application Mode

Terminal Server can run in application mode on nodes in a Server cluster. NOTE: There is no failover of Terminal Server sessions.

Benefits

High Availability - Terminal Server directory service can be made highly available through failover.

Majority Node Set (MNS) Clusters

Windows Server 2003 has an optional quorum resource that does not require a disk on a shared bus for the quorum device. This feature is designed to be built in to larger end-to-end solutions by OEMs, IHVs and other software vendors rather than be deployed by end-users specifically, although this is possible for experienced users. The scenarios targeted by this new feature include:

Geographically dispersed clusters. This mechanism provides a single, Microsoft-supplied quorum resource that is independent of any storage solution for a geographically dispersed or multi-site cluster. NOTE: There is a separate cluster Hardware Compatibility List (HCL) for geographic clusters.

Low-cost or appliance-like highly available solutions that have no shared disks but use other techniques such as log shipping or software disk or file system replication and mirroring to make data available on multiple nodes in the cluster.

NOTE: Windows Server 2003 provides no mechanism to mirror or replicate user data across the nodes of an MNS cluster, so while it is possible to build clusters with no shared disks at all, it is an application specific issue to make the application data highly available and redundant across machines.

Benefits

Storage Abstraction – frees up the storage subsystem to manage data replication between multiple sites in the most appropriate way, without having to worry about a shared quorum disk, and at the same time still supporting the idea of a single virtual cluster.

No Shared Disks – there are some scenarios that require tightly consistent cluster features, yet do not require shared disks. For example, a) clusters where the application keeps data consistent between nodes (e.g. database log shipping and file replication for relatively static data), and b) clusters that host applications that have no persistent data, but need to cooperate in a tightly coupled way to provide consistent volatile state.

Enhanced Redundancy – if the shared quorum disk is corrupted in any way, the entire cluster goes offline. With Majority Node Sets, the corruption of quorum on one node does not bring the entire cluster offline.

Installation

Installed by Default

Clustering is installed by default. You only need to configure a Cluster by launching Cluster Administrator or script the configuration with Cluster.exe. In addition, third-party quorum resources can be pre-installed and then selected during Server cluster configuration, rather than having additional resource specific procedures. All Server cluster configurations can be deployed the same way.

Benefits

Easier Administration – you no longer need to provide a media CD to install Server clusters.

No reboot – you no longer need to reboot after you install or uninstall Cluster Service.

Pre-configuration Analysis

Analyzes and verifies hardware and software configuration and identifies potential problems. Provides a comprehensive and easy-to-read report on any potential configuration issues before the Server cluster is created.

Benefits

Compatibility – Ensures that any known incompatibilities are detected prior to configuration. For example, Service for Macintosh (SFM), Network Load Balancing (NLB), dynamic disks, and DHCP issued addresses are not supported with Cluster Service.

Default Values

Creates a Server cluster that conforms to best practices using default values and heuristics. Many times for newly created Server clusters, the default values are the most appropriate configuration.

Benefits

Easier Administration – Server cluster creation asks fewer setup questions, data is collected and the code makes decisions about the configuration. The goal is to get a “default” Server cluster up and running that can then be customized using the Server cluster administration tools if required.

Multi Node Addition

Allows multiple nodes to be added to a Server cluster in a single operation.

Benefits

Easier Administration – makes it quicker and easier to create multi-node Server clusters.

Extensible Architecture

Extensible architecture allows applications and system components to take part in Server cluster configuration. For example, applications can be installed prior to a server being server clustered and the application can participate in (or even block) this node joining the Server cluster.

Benefits

Third-Party Support – allows applications to setup Server cluster resources and/or change their configuration as part of Server cluster installations rather than as a separate post-Server cluster installation task.

Remote Administration

Allows full remote creation and configuration of the Server cluster. New Server clusters can be created and nodes can be added to an existing Server cluster from a remote management station. In addition, drive letter changes and physical disk resource fail-over are updated to Terminal Server client's sessions.

Benefits

Easier Administration – allows for better remote administration via Terminal Services.

Command Line Tools

Server cluster creation and configuration can be scripted through the cluster.exe command line tool.

Benefits

Easier Administration – much easier to automate the process of creating a cluster.

Simpler Uninstallation

Uninstalling Cluster Service from a node is now a one step process of evicting the node. Previous versions required eviction followed by uninstallation.

Benefits

Easier Administration – Uninstalling the Cluster Service is much more efficient as you only need to evict the node through Cluster Administrator or Cluster.exe and the node is unconfigured for Cluster support. There is also a new switch for Cluster.exe which will force the uninstall if there is a problem with getting into Cluster Administrator:

```
cluster node %NODENAME% /force
```

Quorum Log Size

The default size of the quorum log has been increased to 4096 KB (was 64 KB).

Benefits

Large number of shares – a quorum log of 4,096 KB allows for large numbers of file or printer shares (e.g. 200 printer shares). In previous versions, the quorum log would run out of space with this many shares, causing inconsistent failover of resources.

Local Quorum

If a node is not attached to a shared disk, it will automatically configure a "Local Quorum" resource. It is also possible to create a local quorum resource once Cluster Service is running.

Benefits

Test Cluster – This makes it very easy for users to create a test cluster on their local PC for testing out cluster applications, or for getting familiar with the Cluster Service. Users do not need special cluster hardware that has been certified on the Microsoft Cluster HCL to run a test cluster.

Note: Local quorum is only supported for one node clusters (i.e. lonewolf). In addition, the use of hardware that has not been certified on the HCL is not supported for production environments.

Recovery – in the event you lose all of your shared disks, one option for getting a temporary cluster working (e.g. while you wait for new hardware) is to use the cluster.exe /fixquorum switch to start the cluster, then create a local quorum resource and set this as your quorum. In the case of a print cluster, you can point the spool folder to the local disk. In the case of a file share, you can point the file share resource to the local disk, where backup data has been restored. Obviously, this does not provide any failover, and would only be seen as a temporary measure.

Quorum Selection

You no longer need to select which disk is going to be used as the Quorum Resource. It is automatically configured on the smallest disk that is larger than 50 MB and formatted NTFS.

Benefits

Easier Administration – the end user no longer has to worry about which disk to use for the quorum. NOTE: The option to move the Quorum Resource to another disk is available during setup or after the Cluster has been configured.

Integration

Active Directory

Cluster Service now has much tighter integration with Active Directory™ (AD), including a "virtual" computer object, Kerberos authentication, and a default location for services to publish service control points (e.g. MSMQ).

Benefits

Virtual Server – by publishing a cluster virtual server as a computer object in the Active Directory, users can access the virtual server just like any other Windows

2000 server. In particular, it removes the need for NetBIOS to browse and administrator the cluster nodes, allowing clients to locate cluster objects via DNS, the default name resolution service for Windows Server 2003. NOTE: Although the network name Server cluster resource publishes a computer object in Active Directory, that computer object should NOT be used for administrative tasks such as applying group policy. The ONLY roles for the virtual server computer object in Windows Server 2003 are:

To allow Kerberos authentication to services hosted in a virtual server, and
For cluster-aware and Active Directory-aware services (such as MSMQ) to publish service provider information specific to the virtual server they are hosted in.

Kerberos Authentication – this form of authentication allows users to be authenticated against a server without ever having to send their password. Instead, they present a ticket that grants them access to the server. This contrasts to NTLM authentication, used by Windows 2000 Cluster Service, which sends the user's password as a hash over the network. In addition, Kerberos supports mutual authentication of client and server, and allows delegation of authentication across multiple machines. NOTE: In order to have Kerberos authentication for the virtual server in a mixed mode cluster (i.e. Windows 2000 & Windows Server 2003), you must be running Windows 2000 Advanced Server SP3 or higher. Otherwise NTLM will be used for all authentications.

Publish Services – now that Cluster Service is Active Directory-aware, it can integrate with other services that publish information about their service in AD. For example, Microsoft Message Queuing (MSMQ) 2.0 can publish information about public queues in AD, so that users can easily find their nearest queue,. Windows Server 2003 now extends this to allow clustered public queue information to be published in AD.

NOTE: Cluster integration does not make any changes to the AD schema.

Extend Cluster Shared Disk Partitions

If the underlying storage hardware supports dynamic expansions of a disk unit, or LUN, then the disk volume can be extended online using the DISKPART.EXE utility.

Benefits

Easier Administration – Existing volumes can be expanded online without taking down applications or services.

Resources

Printer Configuration

Cluster Service now provides a much simpler configuration process for setting up clustered printers.

Benefits

Easier Administration – To set up a clustered print server, you need to configure only the Spooler resource in Cluster Administrator and then connect to the virtual server to configure the ports and print queues. This is an improvement over previous versions of Cluster Service in which you had to repeat the configuration steps on each node in the cluster, including installing printer drivers.

MSDTC Configuration

The Microsoft Distributed Transaction Coordinator (MSDTC) can now be configured once, and then be replicated to all nodes.

Benefits

Easier Administration – in previous versions, the COMCLUST.EXE utility had to be run on each node in order to cluster the MSDTC. It is now possible to configure MSDTC as a resource type, assign it to a resource group, and have it automatically configured on all cluster nodes.

Scripting

Existing applications can be made Server cluster-aware using scripting (VBScript and Jscript) rather than writing resource dlls in C or C++.

Benefits

Easier Development – makes it much simpler to write specific resource plug-ins for applications so they can be monitored and controlled in a Server cluster. Supports resource specific properties, allowing a resource script to store Server cluster-wide configurations that can be used and managed in the same way as any other resource.

MSMQ Triggers

Cluster Service has enhanced the MSMQ resource type to allow multiple instances on the same cluster.

Benefits

Enhanced Functionality – allows you to have multiple clustered message queues running at the same time, providing increased performance (in the case of Active/Active MSMQ clusters) and flexibility.

NOTE: You can only have one MSMQ resource per Cluster Group

Network Enhancements

Enhanced Network Failover

Cluster Service now supports enhanced logic for failover when there has been a complete loss of internal (heartbeat) communication. The network state for public communication of all nodes is now taken into account.

Benefits

Better Failover – in Windows 2000, if Node A owned the quorum disk and lost all network interfaces (i.e. public and heartbeat), it would retain control of the cluster, despite the fact that no one could communicate with it, and that another node may have had a working public interface. Windows Server 2003 cluster nodes now take the state of their public interfaces into account prior to arbitrating for control of the cluster.

Media Sense Detection

When using Cluster Service, if network connectivity is lost, the TCP/IP stack does not get unloaded by default, as it did in Windows 2000. There is no longer the need to set the DisableDHCPMediaSense registry key.

Benefits

Better Failover – in Windows 2000, if network connectivity is lost, the TCP/IP stack was unloaded, which meant that all resources that depended on IP addresses were taken offline. Also, when the networks came back online, their network role reverted to the default setting (i.e. client and private). By disabling Media Sense by default, it means the network role is preserved, as well as keeping all IP address dependant resources online.

Multicast Heartbeat

Allows multi-cast heartbeats between nodes in a Server cluster. Multi-cast heartbeat is automatically selected if the cluster is large enough and the network infrastructure can support multi-cast between the cluster nodes. Although the multi-cast parameters can be controlled manually, a typical configuration requires no administration tasks or tuning to enable this feature. If multicast communication fails for any reason, the internal communications will revert to unicast. All internal communications are signed and secure.

Benefits

Reduced Network Traffic – by using multicast, it reduces the amount of traffic in a cluster subnet, which can be particularly beneficial in clusters of more than two nodes, or geographically dispersed clusters.

Storage

Volume Mount Points

Volume mount points are now supported on shared disks (excluding the quorum), and will work properly on failover if configured correctly.

Benefits

Flexible Filesystem Namespace - volume mount points (Windows 2000 or later) are directories that point to specified disk volumes in a persistent manner (e.g. you can configure C:\Data to point to a disk volume). They bypass the need to associate each disk volume with a drive letter, thereby surpassing the 26 drive letter limitation (e.g. without volume mount points, you would have to create a G: drive to map the "Data" volume to). Now that Cluster Service supports volume mount points, you have much greater flexibility in how you map your shared disk namespace.

NOTE: The directory that hosts the volume mount point must be NTFS since the underlying mechanism uses NTFS reparse points. However the file system that is being mounted can be FAT, FAT32, NTFS, CDFS, or UDFS.

Client Side Caching (CSC)

Client Side Caching (CSC) is now supported for clustered file shares.

Benefits

Offline File Access –Client Side Caching for clustered file shares allows a client to cache data stored on a clustered share. The client works on a local copy of the data that is uploaded back to the Server cluster when the file is closed. This allows the failure of a server in the Server cluster and the subsequent failover of the file share service to be hidden from the client.

Distributed File System

Distributed File System (DFS) has had a number of improvements, including multiple stand-alone roots, independent root failover, and support for Active/Active configurations.

Benefits

Distributed File System (DFS) allows multiple file shares on different machines to be aggregated into a common namespace (e.g. [\\dfsroot\share1](#) and [\\dfsroot\share2](#) are actually aggregated from [\\server1\share1](#) and [\\server2\share2](#)). New clustering benefits include:

Multiple Stand-Alone Roots – previous versions only supported one clustered stand-alone root. You can now have multiple ones, giving you much greater flexibility in

planning your distributed file system namespace (e.g. multiple DFS roots on the same virtual server, or multiple DFS roots on different virtual servers).

Independent Failover – granular failover control is available for each DFS root, allowing you to configure failover settings on an individual basis and resulting in faster failover times.

Active/Active Configurations – you can now have multiple stand-alone roots running actively on multiple nodes.

Encrypted File System

With Windows Server 2003, the encrypting file system (EFS) is supported on clustered file shares. This allows data to be stored in encrypted format on clustered disks.

Storage Area Networks (SAN)

Clustering has been optimized for SANs, including targeted device resets and the shared storage buses.

Benefits

Targeted Bus Resets - the Server cluster software now issues a special control code when releasing disk drives during arbitration. This can be used in conjunction with HBA drivers that support the extended Windows Server 2003 feature set to selectively reset devices on the SAN rather than full bus reset. This ensures that the Server cluster has much lower impact on the SAN fabric.

Shared Storage Bus – shared disks can be located on the same storage bus as the Boot, Pagefile and dump file disks. This allows a clustered server to have a single storage bus (or a single redundant storage bus). NOTE: This feature is disabled by default due to the configuration restrictions. This feature can/should only be enabled by OEMs and IHVs for specific and qualified solutions. This is NOT a general purpose feature exposed to end users.

Operations

Backup and Restore

You can actively restore the local cluster nodes cluster configuration or you can restore the cluster information to all nodes in the Cluster. A node restoration is also built into Automatic System Recovery (ASR).

Benefits

Backup and Restore – Backup (NTBackup.exe) in Windows Server 2003 has been enhanced to enable seamless backups and restores of the local Cluster database, and to be able to restore the configuration locally and to all nodes in a Cluster.

Automated System Recovery – ASR can completely restore a cluster in a variety of scenarios, including: a) damaged or missing system files, b) complete OS reinstallation due to hardware failure, c) a damaged Cluster database, and d) changed disk signatures (including shared).

Enhanced Node Failover

Cluster Service now includes enhanced logic for node failover when you have a cluster with three or more nodes. This includes doing a manual “Move Group” operation in Cluster Administrator.

Benefits

Better Failover – during failover in a cluster with three or more nodes, the Cluster Service will take into account the “Preferred Owner List” for each resource, as well as the installation order for each node, in order to work out which node the group should be moved to.

Group Affinity Support

Allows an application to describe itself as an N+1 application. In other words, the application is running actively on N nodes of the Server cluster and there are 1 “spare” nodes available if an active node fails. In the event of failure, the failover manager will try to ensure that the application is failed over to a spare node rather than a node that is currently running the application.

Benefits

Better performance – applications are failed over to spare nodes before active nodes.

Node Eviction

Evicting a node from a Server cluster no longer requires a reboot to clean up the Server cluster state. A node can be moved from one Server cluster to another without having to reboot. In the event of a catastrophic failure, the Server cluster configuration can be force cleaned regardless of the Server cluster state.

Benefits

Increased Availability – not having to reboot increases the uptime of the system.

Disaster Recovery – in the event of a node failure, the cluster can be cleaned up easily.

Rolling Upgrades

Rolling upgrades are supported from Windows 2000 to Windows Server 2003.

Benefits

Minimum downtime – rolling upgrades allow one node in a cluster to be taken offline for upgrading, while other nodes in the cluster continue to function on an older version. NOTE: There is no support for rolling upgrades from a Microsoft Windows NT 4.0 cluster to a Windows Server 2003 cluster. An upgrade from Windows NT 4.0 is supported but the cluster will have to be taken offline during the upgrade.

Queued Changes

The cluster service will now queue up changes that need to be completed if a node is offline.

Benefits

Easier Administration – ensures that you do not need to apply a change twice if a node is offline. For example, if a node is offline and is evicted from the Cluster by a remaining node, the cluster service will be uninstalled the next time the first node attempts to join the Cluster. This also holds true for applications.

Disk Changes

The Cluster Service more efficiently adjusts to shared disk changes in regards to size changes and drive letter assignments.

Benefits

Dynamic Disk Size – If you increase the size of a shared disk, the Cluster Service will now dynamically adjust to it. This is particularly helpful for SANs, where volume sizes can change easily. It does this by working directly with Volume Mount Manager, and no longer directly uses the DISKINFO or DISK keys. NOTE: These keys are maintained for backwards compatibility with previous versions of the Cluster Service.

Password Change

Cluster Service account password changes no longer require any downtime of the cluster nodes. In addition, passwords can be reset on multiple clusters at the same time.

Benefits

Reduced Downtime – In Windows Server 2003, you can change the Cluster Service account password on the domain as well as on each local node, without having to take the cluster offline. If multiple clusters use the same Cluster service account, you can change them simultaneously. In Microsoft Windows NT 4.0 and Microsoft Windows 2000, to change the Cluster service account password, you have to stop the Cluster service on all nodes before you can make the password change.

Resource Deletion

Resources can be deleted in Cluster Administrator or with Cluster.exe without taking them offline first.

Benefits

Easier Administration – in previous versions, you first had to take a resource offline before you could delete it. Now, Cluster Service will take them offline automatically, and then delete them.

WMI Support

Server clusters provides WMI support for:

Cluster control and management functions including starting and stopping resources, creating new resource and dependencies etc.

Application and cluster state information. WMI can be used to query whether applications are online, whether cluster nodes are up and running as well as a host of other status information.

Cluster state change events are propagated via WMI to allow applications to subscribe to WMI events that show when an application has failed, when an application is restarted, when a node fails etc.

Benefits

Better Management – allows Server clusters to be managed as part of an overall WMI environment.

Supporting and Troubleshooting

Offline/Failure Reason Codes

These provide additional information to the resource as to why the application was taken offline, or failed.

Benefits

Better Troubleshooting – allows the application to have different semantics if the application has failed or some dependency of the application has failed, versus the administrator specifically moved the group to another node in the Server cluster.

Software Tracing

Cluster Service now has a feature called software tracing that will produce more information to help with troubleshooting Cluster issues.

Benefits

Better Troubleshooting – this is a new method for debugging that will allow Microsoft to debug the Cluster Service without loading checked build versions of the dll's (symbols).

Cluster Logs

A number of improvements have been made to the Cluster Service log files, including a setup log, error levels (info, warn, err), local server time entry, and GUID to resource name mapping.

Benefits

Setup Log – during configuration of Cluster Service, a separate setup log (%SystemRoot%\system32\Logfiles\Cluster\CICfgSrv.log) is created to assist in troubleshooting.

Error Levels – this makes it easy to be able to highlight just the entries that require action (e.g. err).

Local Server Time Stamp – this will assist in comparing event log entries to Cluster logs.

GUID to Resource Name Mapping – this assists in understanding the cluster log references to GUIDs. A Cluster object file (%windir%\Cluster\Cluster.obj) is automatically created and maintained that contains a mapping of GUID's to Resource Name mappings.

Event Log

Additional events are written to the event log indicating not only error cases, but showing when resources are successfully failed over from one node to another.

Benefits

Better Monitoring – this allows event log parsing and management tools to be used to track successful failovers rather than just catastrophic failures.

Clusdiag

A new tool called clusdiag is available in the Windows Server 2003 Resource Kit.

Benefits

Better Troubleshooting – makes reading and correlating cluster logs across multiple cluster nodes and debugging of cluster issues more straight forward.

Validation and Testing – Clusdiag allows users to run stress tests on the server, storage and clustering infrastructure. As such, it can be used as a validation and test tool before a cluster can be put into production

Chkdsk Log

The cluster service creates a chkdsk log whenever chkdsk is run on a shared disk.

Benefits

Better Monitoring – this allows a system administrator to find out and react to any issues that were discovered during the chkdsk process.

Disk Corruption

When Disk Corruption is suspect, the Cluster Service reports the results of CHKDSK in event logs and creates a log in %systemroot%\cluster.

Benefits

Better Troubleshooting – results are logged in the Application and Cluster.log. In addition, the Cluster.log references a log file (e.g. %windir%\CLUSTER\CHKDSK_DISK2_SIGE9443789.LOG) in which detailed CHKDSK output is recorded.

Network Load Balancing

Network Load Balancing Manager

In Windows 2000, to create an NLB Cluster, users had to separately configure each machine in the cluster. Not only was this unnecessary additional work, but it also opened up the possibility for unintended user error because identical Cluster Parameters and Port Rules had to be configured on each machine. A new utility in Windows Server 2003 called the *NLB Manager* helps solve some of these problems by providing single point of configuration and management of NLB clusters. Some key features of the NLB Manager:

Creating new NLB clusters and automatically propagating Cluster Parameters and Port Rules to all hosts in the cluster and propagating Host Parameters to specific hosts in the cluster.

Adding and removing hosts to and from NLB clusters.

Automatically adding Cluster IP Addresses to TCP/IP.

Managing existing clusters simply by connecting to them or by loading their host information from a file and then saving this information to a file for later use.

Configuring NLB to load balance multiple web sites or applications on the same NLB Cluster, including adding all Cluster IP Addresses to TCP/IP and controlling traffic sent to specific applications on specific hosts in the cluster. [See the *Virtual Clusters* feature below]

Diagnosing mis-configured clusters.

Virtual Clusters

In Windows 2000, users could load balance multiple web sites or applications on the same NLB Cluster simply by adding the IP Addresses corresponding to these web sites or applications to TCP/IP on each host in the cluster. This is because NLB, on each host, load balanced all IP Addresses in TCP/IP, except the Dedicated IP Address. The shortcomings of this feature in Windows 2000 were:

Port Rules specified for the cluster were automatically applied to all web sites or applications load balanced by the cluster.

All the hosts in the cluster had to handle traffic for all the web sites/ applications hosted on them.

To block out traffic for a specific application on a specific host, traffic for all applications on that host had to be blocked.

A new feature in Windows Server 2003 called *Virtual Clusters* overcomes the above deficiencies by providing per-IP Port Rules capability. This allows the user to:

Configure different Port Rules for different Cluster IP Addresses, where each Cluster IP Address corresponds to a web site or application being hosted on the NLB Cluster. This is in contrast to NLB in Windows 2000, where Port Rules were applicable to an entire host and not to specific IP Addresses on that host.

Filter out traffic sent to a specific website/application on a specific host in the cluster. This allows individual applications on hosts to be taken offline for upgrades, restarts, etc. without affecting other applications being load balanced on the rest of the NLB cluster.

Pick and choose which host in the cluster should service traffic sent to which website or application being hosted on the cluster. This way, not all hosts in the cluster need to handle traffic for all applications being hosted on that cluster.

Multi-NIC support

Windows 2000 allowed the user to bind NLB to only one network card in the system. Windows Server 2003 allows the user to bind NLB to multiple network cards, thus removing the limitation.

This now enables users to:

Host multiple NLB clusters on the same hosts while leaving them on entirely independent networks. This can be achieved by binding NLB to different network cards in the same system.

Use NLB for Firewall and Proxy load balancing in scenarios where load balancing is required on multiple fronts of a proxy or firewall.

Bi-directional Affinity

The addition of the *Multi-NIC support* feature enabled several other scenarios where there was a need for load balancing on multiple fronts of an NLB Cluster. The most common usage of this feature will be to cluster ISA servers for Proxy and Firewall load balancing. The two most common scenarios where NLB will be used together with ISA are:

Web Publishing

Server Publishing

In the Web Publishing scenario, the ISA cluster typically resides between the outside internet and the front-end web servers. In this scenario, the ISA servers will have NLB bound only to the external interface, therefore, there will be no need to use the *Bi-directional Affinity* feature.

However, in the Server Publishing scenario, the ISA cluster will reside between the Web Servers in the front, and the Published Servers in the back. Here, NLB will have to be bound to both the external interface [facing the Web servers] and the internal

interface [facing the Published Servers] of each ISA server in the cluster. This increases the level of complexity because now when connections from the Web Servers are being load balanced on the external interface of the ISA Cluster and then forwarded by one of the ISA servers to a Published Server, NLB has to ensure that the response from the Published Server is always routed to the same ISA server that handled the corresponding request from the Web Server because this is the only ISA server in the cluster that has the security context for that particular session. So, NLB has to make sure that the response from Published Server does not get load balanced on the internal interface of the ISA Cluster since this interface is also clustered using NLB.

This task is accomplished by the new feature in Windows Server 2003 called *Bi-directional Affinity*. Bi-directional affinity makes multiple instances of NLB on the same host work in tandem to ensure that responses from Published Servers are routed through the appropriate ISA servers in the cluster.

Limiting switch flooding using IGMP support

The NLB algorithm requires every host in the NLB Cluster to see every incoming packet destined for the cluster. NLB accomplishes this by never allowing the switch to associate the cluster's MAC address with a specific port on the switch. However, the unintended side effect of this requirement is that the switch ends up flooding all of its ports with all incoming packets meant for the NLB cluster. This can certainly be a nuisance and a waste of network resources. In order to arrest this problem, a new feature called *IGMP support* has been introduced in Windows Server 2003. This feature helps limit the flooding to only those ports on the switch that have NLB machines connected to them. This way, non-NLB machines do not see traffic that was intended only for the NLB Cluster, while at the same time, all of the NLB machines see traffic that was meant for the cluster, thus satisfying the requirements of the algorithm. It should, however, be noted that IGMP support can only be enabled when NLB is configured in *multicast* mode. Multicast mode has its own drawbacks which are discussed extensively in KB articles available on www.microsoft.com. The user should be aware of these shortcomings of multicast mode before deploying IGMP support. Switch flooding can also be limited when using *unicast* mode by creating VLANs in the switch and putting the NLB cluster on its own VLAN. Unicast mode does not have the same drawbacks as Multicast mode, and so limiting switch flooding using this approach may be preferable.

Server cluster Architecture

Updated: January 01, 2003

This section discusses Server cluster and how to configure it for failover support for applications and services. Resource groups, cluster storage devices, network configuration and storage area networks are also discussed.

Server cluster

Server cluster is used to provide failover support for applications and services. A Server cluster can consist of up to eight nodes. Each node is attached to one or more cluster storage devices. Cluster storage devices allow different servers to share the same data, and by reading this data provide failover for resources.

Connecting Storage Devices

The preferred technique for connecting storage devices is fibre channel.

- When using three or more nodes, fibre channel is the only technique that should be used.
- When using 2-node clustering with Advanced Server, SCSI or fibre channel can be used to connect to the storage devices.

Configuring Server clusters

Server clusters can be setup using many different configurations. Servers can be either active or passive, and different servers can be configured to take over the failed resources of another server. Failover can take several minutes, depending on the configuration and the application being used, but is designed to be transparent to the end-user.

Server cluster and Failover

When a node is active, it makes its resources available. Clients access these resources through dedicated virtual servers.

Server cluster uses the concept of virtual servers to specify groups of resources that failover together. When a server fails, the group of resources configured on that server for clustering fails over to another server. The server that handles the failover should be configured for the extra capacity needed to handle the

additional workload. When the failed server comes back online, Server cluster can be configured to allow failback to the original server, or to allow the current server to continue to process requests.

Figure 6: Multi-node clusters with all nodes active

Figure 6 above shows a configuration where all nodes in a database cluster are active and each node has a separate resource group. With a partitioned view of the database, each resource group could handle different types of requests. The types of requests handled could be based on one or more factors, such as the name of an account or geographic location. In the event of a failure, each node is configured to fail over to the next node in turn.

Resource Groups

Resources that are related or dependent on each other are associated through resource groups. Only applications that need high availability should be part of a resource group. Other applications can run on a server cluster, but don't need to be a part of a resource group. Before adding an application to a resource group, IT staff must determine if the application can work within the cluster environment. Cluster-Aware Applications. Applications that can work within the cluster environment and support cluster events are called cluster-aware. Cluster-aware applications can register with the Server cluster to receive status and notification information.

Cluster-Unaware Applications. Applications that do not support cluster events are called cluster-unaware. Some cluster-unaware applications can be assigned to resource groups and can be failed over.

Applications that meet the following criteria can be assigned to resource groups.

- IP-based protocols are used for cluster communications. The application must use an IP-based protocol for their network communications. Applications cannot use NetBEUI, IPX, AppleTalk or other protocols to communicate.
- Nodes in the cluster access application data through shared storage devices. If the application is not able to store its data in a configurable location, the application data will not be available on failover.

- Client applications experience a temporary loss of network connectivity when failover occurs. If client applications cannot retry and recover from this, they will cease to function normally.

New Features for Resources and Resource Types

Windows Server 2003 adds new features for resources and resource types. A new resource type allows applications to be made cluster-aware using VBScript and JScript. Additionally, Windows Management Instrumentation (WMI) can be used for cluster management and event notification.

Architecting Resource Groups

When architecting resource groups, IT staff should list all server-based applications and services that will run in the cluster environment, regardless of whether they will need high availability. Afterward, divide the list into three sections:

- Those that need to be highly available
- Those that are not part of the cluster and on which clustered resources do not depend
- Those that are running on the cluster servers that do not support failover and on which the cluster may depend.

Applications and services that need to be highly available should be placed into resource groups. Other applications should be tracked, and their interactions with clustered applications and services should be clearly understood. Failure of an application or service that is not part of a resource group should not impact the core functions of the solution being offered. If it does, the application or service may need to be clustered.

Focus on selecting the right hardware to meet the needs of the service offering. A cluster model should be chosen to adequately support resource failover and the availability requirements. Based on the model chosen, excess capacity should be added to ensure that storage, processor and memory are available in the event a resource fails, and failover to a server substantially increases the workload.

With a clustered SQL Server configuration, IT staff should consider using high-end CPUs, fast hard drives and additional memory. SQL Server 2000 and

standard services together use over 100 megabytes (MB) of memory as a baseline. User connections consume about 24 kilobytes (KB) each. While the minimum memory for query execution is one MB of RAM, the average query may require two to four MB of RAM. Other SQL Server processes use memory as well.

Optimizing Cluster Storage Devices

Cluster storage devices should be optimized based on performance and availability needs. While the [Windows Datacenter Hardware Compatibility List](#) provides a detailed list of acceptable Redundant Array of Independent Disks (RAID) configurations for clusters, Table 2 below provides an overview of common RAID configurations. The table entries are organized from the highest RAID level to the lowest.

Table 2 RAID Configurations

RAID Level	RAID Type	RAID Description	Advantages & Disadvantages
5+1	Disk striping with parity + mirroring	Six or more volumes, each on a separate drive, are configured identically as a mirrored stripe set with parity error checking.	Provides very high level of fault tolerance but has a lot of overhead.
5	Disk striping with parity	Three or more volumes, each on a separate drive, are configured as a stripe set with parity error checking. In the case of failure, data can be recovered.	Fault tolerance with less overhead than mirroring. Better read performance than disk mirroring.
1	Disk mirroring	Two volumes on two drives are configured identically. Data is written to both drives. If one	Redundancy. Better write performance than disk striping with parity.

RAID Level	RAID Type	RAID Description	Advantages & Disadvantages
		drive fails, there is no data loss because the other drive contains the data. (Does not include disk striping.)	
0+1	Disk striping with mirroring	Two or more volumes, each on a separate drive, are striped and mirrored. Data is written sequentially to drives that are identically configured.	Redundancy with good read/write performance.
0	Disk striping	Two or more volumes, each on a separate drive, are configured as a stripe set. Data is broken into blocks, called stripes, and then written sequentially to all drives in the stripe set.	Speed/Performance without data protection.

Optimizing Network Configuration

The network configuration of the cluster can also be optimized. All nodes in a cluster must be a part of the same domain and can be configured as domain controllers or member servers. Ideally, multi-node clusters will have at least two nodes that act as domain controllers and provide failover for critical domain services. If this is not the case, the availability of cluster resources may be tied to the availability of the controllers in the domain.

Private and Public Network Addresses

Typically nodes in a cluster are configured with both private and public network addresses.

- Private network addresses are used for node-to-node communications.
- Public network addresses are used for client-to-cluster communications.

Some clusters may not need public network addresses and instead may be configured to use two private networks. In this case, the first private network is for node-to-node communications and the second private network is for communicating with other servers that are a part of the service offering.

Storage Area Networks

Increasingly, clustered servers and storage devices are connected over SANs. SANs use high-performance interconnections between secure servers and storage devices to deliver higher bandwidth and lower latency than comparable traditional networks. Windows 2000 Datacenter Server and Windows Server 2003 Datacenter Edition implement a feature called Winsock Direct that allows direct communication over a SAN using SAN providers.

SAN providers have user-mode access to hardware transports. When communicating directly at the hardware level, the individual transport endpoints can be mapped directly into the address space of application processes running in user mode. This allows applications to pass messaging requests directly to the SAN hardware interface, which eliminates unnecessary system calls and data copying.

SANs typically use two transfer modes. One mode is for small transfers, which primarily consist of transfer control information. For large transfers, SANs can use a bulk mode whereby data is transferred directly between the local system and the remote system by the SAN hardware interface without CPU involvement on the local or remote system. All bulk transfers are pre-arranged through an exchange of transfer control messages.

Other SAN Benefits

In addition to improved communication modes, SANs have other benefits.

- They allow IT staff to consolidate storage needs, using several highly reliable storage devices instead of many.
- They also allow IT staff to share storage with non-Windows operating systems, allowing for heterogeneous operating environments.