What is Clustering?

A cluster is a group of independent computer systems, referred to as nodes, working together as a **unified computing resource**. A cluster provides a single name for clients to use and a single administrative interface, and it guarantees that data is consistent across nodes.

Microsoft servers provide three technologies to support clustering: Network Load Balancing (NLB), Component Load Balancing (CLB), and Microsoft Cluster Service (MSCS) Failover Cluster.

Network Load Balancing

Network Load Balancing acts as a front-end cluster, distributing incoming IP traffic across a cluster of servers, and is ideal for enabling incremental scalability and outstanding availability for e-commerce Web sites. Up to 32 computers running a member of the Windows Server 2003 family can be connected to share a single virtual IP address. NLB enhances scalability by distributing its client requests across multiple servers within the cluster. As traffic increases, additional servers can be added to the cluster; up to 32 servers are possible in any one cluster. NLB also provides high availability by automatically detecting the failure of a server and repartitioning client traffic among the remaining servers within 10 seconds, while it provides users with continuous service.

Component Load Balancing

Component Load Balancing distributes workload across multiple servers running a site's business logic. It provides for dynamic balancing of COM+ components across a set of up to eight identical servers. In CLB, the COM+ components live on servers in a separate, COM+ cluster. Calls to activate COM+ components are load balanced to different servers within the COM+ cluster. CLB complements both NLB and Cluster Service by acting on the middle tier of a multi-tiered clustered network. CLB is provided as a feature of Application Center 2000. Both CLB and Microsoft Cluster Service can run on the same group of machines.

Failover Clustering

Cluster Service acts as a back-end cluster; it provides high availability for applications such as databases, messaging and file and print services. MSCS attempts to minimize the effect of failure on the system as any node (a server in the cluster) fails or is taken offline.

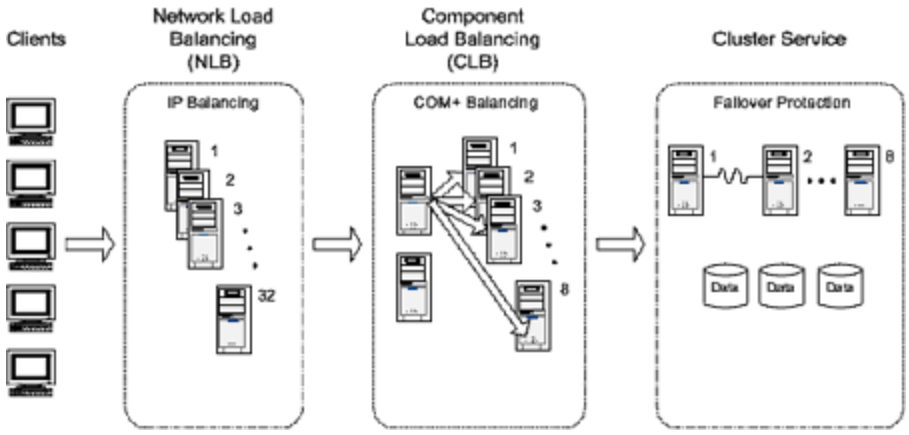


Figure 1. Three Microsoft server technologies support clustering

MSCS failover capability is achieved through redundancy across the multiple connected machines in the cluster, each with independent failure states. The Windows Server 2003, Datacenter Edition supports up to 8 nodes in a cluster.

Each node has its own memory, system disk, operating system and subset of the cluster's resources. If a node fails, the other node takes ownership of the failed node's resources (this process is known as "failover"). Microsoft Cluster Service then registers the network address for the resource on the new node so that client traffic is routed to the system that is available and now owns the resource. When the failed resource is later brought back online, MSCS can be configured to redistribute resources and client requests appropriately (this process is known as "failback").

Microsoft Cluster Service is based on the shared-nothing clustering model. The shared-nothing model dictates that while several nodes in the cluster may have access to a device or resource, the resource is owned and managed by only one system at a time.

Microsoft Cluster Service is comprised of three key components: the Cluster Service, Resource Monitor and Resource DLLs.

The Cluster Service

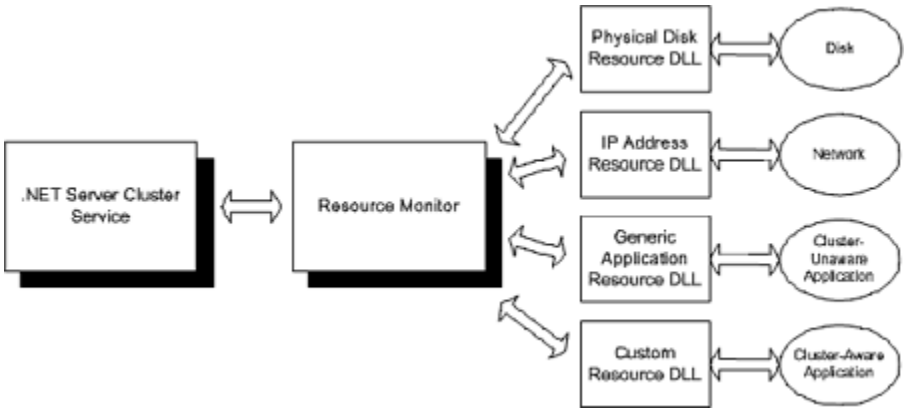
The Cluster Service is the core component and runs as a high-priority system service. The Cluster Service controls cluster activities and performs such tasks as coordinating event notification, facilitating communication between cluster components, handling failover operations and managing the configuration. Each cluster node runs its own Cluster Service.

The Resource Monitor

The Resource Monitor is an interface between the Cluster Service and the cluster resources, and runs as an independent process. The Cluster Service uses the Resource Monitor to communicate with the resource DLLs. The DLL handles all communication with the resource, so hosting the DLL in a Resource Monitor shields the Cluster Service from resources that misbehave or stop functioning. Multiple copies of the Resource Monitor can be running on a single node, thereby providing a means by which unpredictable resources can be isolated from other resources.

The Resource DLL

The third key Microsoft Cluster Service component is the resource DLL. The Resource Monitor and resource DLL communicate using the Resource API, which is a collection of entry points, callback functions and related structures and macros used to manage resources.



What is a Quorum?

What is a quorum? To put it simply, a quorum is the cluster's configuration database. The database resides in a file named \MSCS\quolog.log. The quorum is sometimes also referred to as the quorum log.

Although the quorum is just a configuration database, it has two very important jobs. First of all, it tells the cluster which node should be active. Think about it for a minute. In order for a cluster to work, all of the nodes have to function in a way that allows the virtual server to function in the desired manner. In order for this to happen, each node must have a crystal clear understanding of its role within the cluster. This is where the quorum comes into play. The quorum tells the cluster which node is currently active and which node or nodes are in standby.

It is extremely important for nodes to conform to the status defined by the quorum. It is so important in fact, that Microsoft has designed the clustering service so that if a node cannot read the quorum, that node will not be brought online as a part of the cluster.

The other thing that the quorum does is to intervene when communications fail between nodes. Normally, each node within a cluster can communicate with every other node in the cluster over a dedicated network connection. If this network connection were to fail though, the cluster would be split into two pieces, each containing one or more functional nodes that cannot communicate with the nodes that exist on the other side of the communications failure.

When this type of communications failure occurs, the cluster is said to have been partitioned. The problem is that both partitions have the same goal; to keep the application running. The application can't be run on multiple servers simultaneously though, so there must be a way of determining which partition gets to run the application. This is where the quorum comes in. The partition that "owns" the quorum is allowed to continue running the application. The other partition is removed from the cluster.Local

Types of Quorums

So far in this article, I have been describing a quorum type known as a standard quorum. The main idea behind a standard quorum is that it is a configuration database for the cluster and is stored on a shared hard disk, accessible to all of the cluster's nodes.

In Windows Server 2003, Microsoft introduced a new type of quorum called the Majority Node Set Quorum (MNS). The thing that really sets a MNS quorum apart from a standard quorum is the fact that each node has its own, locally stored copy of the quorum database.

Types:

1) Quorum Disk
2) Local Only Quorum
3) MNS (Majority Node Set)

How Clustering Works

In a two-cluster node Active / Active setup, if any one of the nodes fail, then the another active node will take over the active resources of the failed instance. It is always preferred while creating two-node cluster that each node be connected to a shared disk array using either fiber channel or SCSI cables.

The shared data in the cluster must be stored on shared disks, otherwise, when a failover occurs; the node which is taking over in the cluster pack cannot access it. As we are already aware, clustering does not help protect data or the shared disk array that it is stored on. So it is very important that you select a shared disk array that is very reliable and includes fault tolerance.

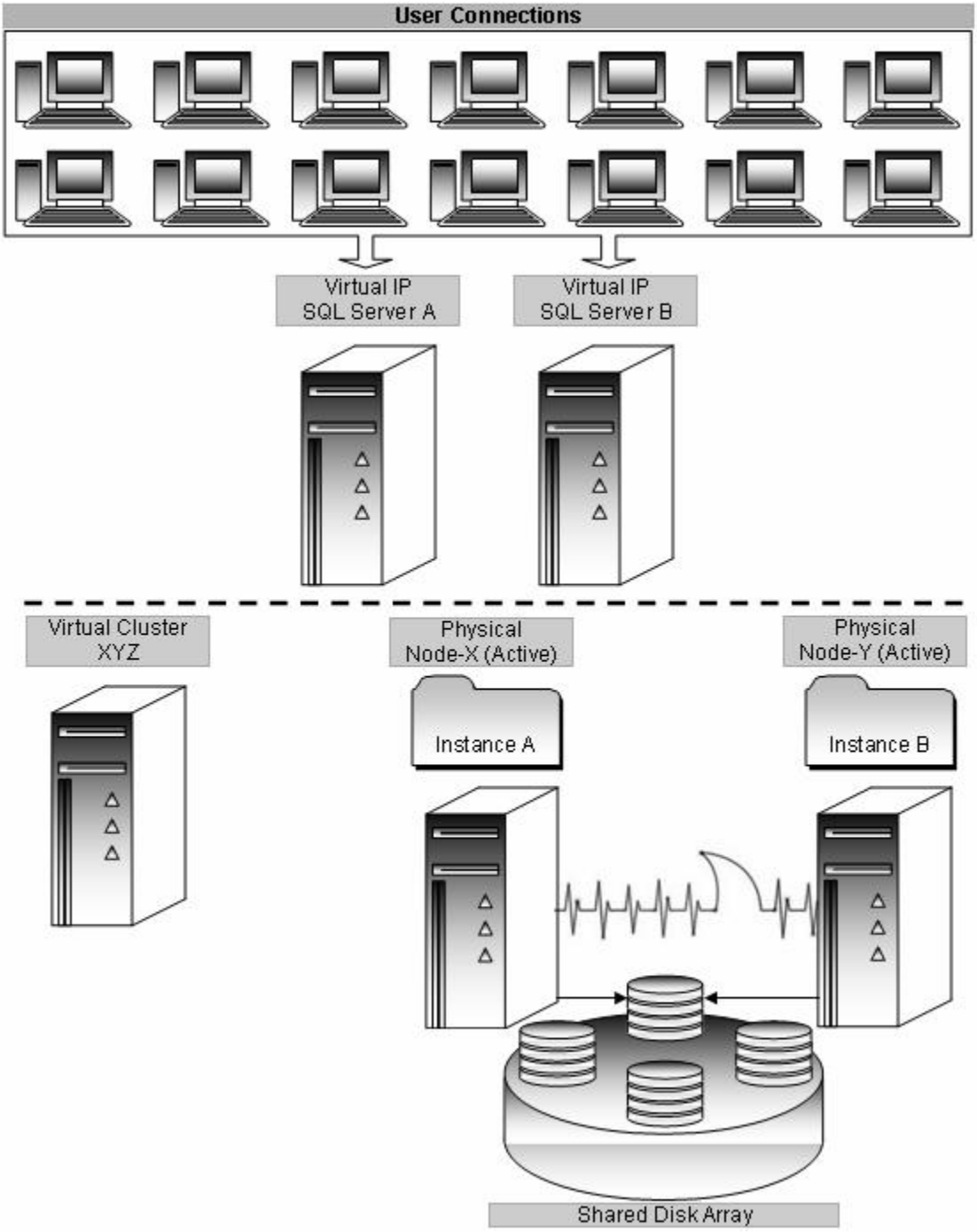
Both nodes of the cluster are also connected to each other via a private network. This private network is used for each node to keep track of the status of the other node. For example, if one of the node experiences a hardware failure, the other node will detect this and will automatically initiate a failover.

When clients initiate a connection, how will they know what to do when a failover occurs? This is the most intelligent part of Microsoft Cluster Services. When a user establishes a connection with SQL Server, it is through SQL Server's own virtual name and virtual TCP/IP address. This name and address are shared by both of the servers in the cluster. In other words, both nodes can be defined as preferred owners of this virtual name and TCP/IP address.

Usually, a client will connect to the SQL Server cluster using the virtual name used by the cluster. And as far as a client is concerned, there is only one physical SQL Server, not two. Assuming that the X node of the SQL Server cluster is the node running SQL Server 'A' in an Active/Active cluster design, then the X node will respond to the client's requests. But if the X node fails, and failover to the next node Y occurs, the cluster will still retain the same SQL Server virtual name and TCP/IP address 'A', although now a new physical server will be responding to client's requests.

During the failover period, which can last up to several minutes, clients will be unable to access SQL Server, so there is a small amount of downtime when failover occurs. The exact amount of time depends on the number and sizes of the databases on SQL Server, and how active they are.

http://msdn.microsoft.com/en-us/library/ms952401.aspx#wns-introclustermscs_topic1
http://www.good.com/documentation5/Exchange_Admin_Guide/Good%20Messaging%205.0%20Admin%20for%20Exchange-10-02.html


User Connections
Virtual IP
SQL Server A
Virtual IP
SQL Server B
Virtual Cluster
XYZ
Physical
Node-X (Active)
Physical
Node-Y (Active)
Instance A
Instance B
Shared Disk Array

Clustering Terms

Cluster Nodes

A cluster node is a server within a cluster group. A cluster node can be Active or it can be Passive as per SQL Server Instance installation.

Heartbeat

The heartbeat is a checkup mechanism arranged between two nodes using a private network set up to see whether a node is up and running. This occurs at regular intervals known as time slices. A failover is initiated, if heartbeat is not functioning, and another node in the cluster will take over the active resources.

Private Network

The Private Network is available among cluster nodes only. Every node will have a Private Network IP address, which can be ping from one node to another. This is to check the heartbeat between two nodes.

Public Network

The Public Network is available for external connections. Every node will have a Public Network IP address, which can be connected from any client within the network.

Shared Cluster Disk Array

A shared disk array is a collection of storage disks that is being accessed by the cluster. This could be SAN or SCSI RAIDs. Windows Clustering supports shared nothing disk arrays. Any one node can own a disk resource at any given time. All other nodes will not be allowed to access it until they own the resource (Ownership change occurs during failover). This protects the data from being overwritten when two computers have access to the same drives concurrently.

Quorum Drive

This is a logical drive assigned on the shared disk array specifically for Windows Clustering. Clustering services write constantly on this drive about the state of the cluster. Corruption or failure of this drive can fail the entire cluster setup.

Cluster Name

This name refers to Virtual Cluster Name, not the physical node names or the Virtual SQL Server names. It is assigned to the cluster as a whole.

Cluster IP Address

This IP address refers to the address which all external connections use to reach to the active cluster node.

Cluster Administrator Account

This account must be configured at the domain level, with administrator privileges on all nodes within the cluster group. This account is used to administer the failover cluster.

Cluster Resource Types

This includes any services, software, or hardware that can be configured within a cluster. Ex: DHCP, File Share, Generic Application, Generic Service, Internet Protocol, Network Name, Physical Disk, Print Spooler, and WINS.

Cluster Group

Conceptually, a cluster group is a collection of logically grouped cluster resources. It may contain cluster-aware application services, such as SQL Server 2000.

SQL Server Network Name (Virtual Name)

This is the SQL Server Instance name that all client applications will use to connect to the SQL Server.

SQL Server IP Address (Virtual IP Address)

This refers to the TCP/IP address that all client applications will use to connect to SQL Server; the Virtual Server IP address.

SQL Server 2000 Full-text

Each SQL Virtual Server has one full-text resource.

Microsoft Distributed Transaction Coordinator (MS DTC)

Certain SQL Server Components require MS DTC to be up and running. MS DTC is shared for all named / default instances in cluster group.

SQL Server Virtual Server Administrator Account

This is the SQL Server service account, and it must follow all the rules that apply to SQL Service user accounts in a non-clustered environment.

How to Cluster Windows Server 2003

Before Installing Windows 2003 Clustering

Before you install Windows 2003 clustering, we need to perform a series of important preparation steps. This is especially important if you didn't build the cluster nodes, as you want to ensure everything is working correctly before you begin the actual cluster installation. Once they are complete, then you can install Windows 2003 clustering. Here are the steps you must take:

- Double check to ensure that all the nodes are working properly and are configured identically (hardware, software, drivers, etc.).
- Check to see that each node can see the data and Quorum drives on the shared array or SAN. Remember, only one node can be on at a time until Windows 2003 clustering is installed.

- Verify that none of the nodes has been configured as a Domain Controller.
- Check to verify that all drives are NTFS and are not compressed.
- Ensure that the public and private networks are properly installed and configured.
- Ping each node in the public and private networks to ensure that you have good network connections. Also ping the Domain Controller and DNS server to verify that they are available.
- Verify that you have disabled NetBIOS for all private network cards.
- Verify that there are no network shares on any of the shared drives.
- If you intend to use SQL Server encryption, install the server certificate with the fully qualified DNS name of the virtual server on all nodes in the cluster.
- Check all of the error logs to ensure there are no nasty surprises. If there are, resolve them before proceeding with the cluster installation.
- Add the SQL Server and Clustering service accounts to the Local Administrators group of all the nodes in the cluster.
- Check to verify that no antivirus software has been installed on the nodes. Antivirus software can reduce the availability of clusters and must not be installed on them. If you want to check for possible viruses on a cluster, you can always install the software on a non-node and then run scans on the cluster nodes remotely.
- Check to verify that the Windows Cryptographic Service Provider is enabled on each of the nodes.
- Check to verify that the Windows Task Scheduler service is running on each of the nodes.
- If you intend to run SQL Server 2005 Reporting Services, you must then install IIS 6.0 and ASP .NET 2.0 on each node of the cluster.

These are a lot of things you must check, but each of these is important. If skipped, any one of these steps could prevent your cluster from installing or working properly.

How to Install Windows Server 2003 Clustering

Now that all of your physical nodes and shared array or SAN is ready, you are now ready to install Windows 2003 clustering. In this section, we take a look at the process, from beginning to end.

To begin, you must start the Microsoft Windows 2003 Clustering Wizard from one of the nodes. While it doesn't make any difference to the software which physical node is used to begin the installation, I generally select one of the physical nodes to be my primary (active) node, and start working there. This way, I won't potentially get confused when installing the software.

If you are using a SCSI shared array, and for many SAN shared arrays, you will want to make sure that the second physical node of your cluster is turned off when you install cluster services on the first physical node. This is because Windows 2003 doesn't know how to deal with a shared disk until cluster services is installed. Once you have installed cluster services on the first physical node, you can turn on the second physical node, boot it, and then proceed with installing cluster services on the second node.

These are some of the improvements Windows Server 2003 has made in clustering:

- Larger clusters:The Enterprise Edition now supports up to 8-node clusters. Previous editions only supported 2-node clusters. The Datacenter Edition supports 8-node clusters as well. In Windows 2000, it supported only 4-node clusters.
- 64-bit support: This feature allows clustering to take advantage of the 64-bit version of Windows Server 2003, which is especially important to being able to optimize SQL Server 2000 Enterprise Edition.
- High availability: With this update to the clustering service, the Terminal Server directory service can now be configured for failover.
- Cluster Installation Wizard: A completely redesigned wizard allows you to join and add nodes to the cluster. It also provides additional troubleshooting by allowing you to view logs and details if things go wrong. It can save you some trips to the Add/Remove Programs applet.
- MSDTC configuration: You can now configure MSDTC once and it is replicated to all nodes. You no longer have to run the comclust.exe utility on each node.

Madhusudhana Rao Gadipudi