the Availability Digest
www.availabilitydigest.com

Windows Server Failover Clustering

April 2010

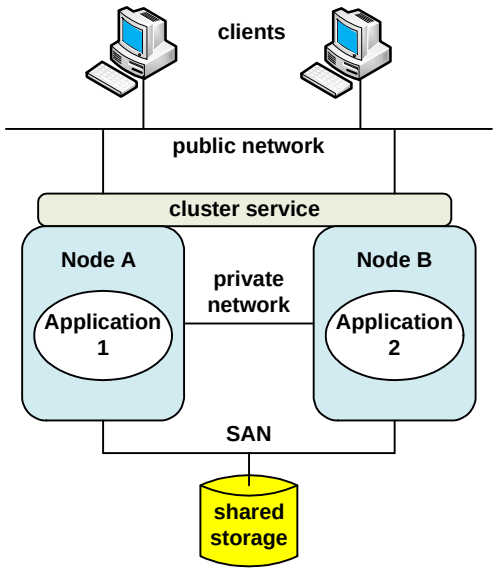
Windows Server Failover Clustering (WSFC) is the successor to Microsoft Cluster Service (MSCS). WSFC and its predecessor, MSCS, offer high availability for critical applications such as email, databases, and line-of-business applications by implementing a redundant cluster of Windows servers that provide a single-system image to the users.

MSCS has been Microsoft's solution to building high-availability clusters of Windows servers since it was first introduced with Windows NT Server 4.0. MSCS has been significantly enhanced and simplified and renamed WSFC with the release of Windows Server 2008. WSFC for Windows Server 2008 R2 has seen even further enhancements to Windows clustering.

What is a Windows Cluster?

Microsoft defines a cluster as follows:[1]

> "A failover cluster is a group of independent computers, or nodes, which are physically connected by a local-area network (LAN) or a wide-area network (WAN) and that are programmatically connected by cluster software. The group of nodes is managed as a single system and shares a common namespace. The group usually includes multiple network connections and data storage connected to the nodes via storage area networks (SANs). The failover cluster operates by moving resources between nodes to provide service if system components fail."



The nodes in a Windows cluster are Windows servers that are physically interconnected by a redundant private network for node monitoring and failover. The nodes have access to a common set of redundant disk resources through a storage area network (SAN). The cluster service is the software that programmatically connects the nodes in the cluster and provides a single-system view to the clients that are using the cluster.

The clients are unaware that they are dealing with a cluster. The cluster appears to them to be a single Windows server. In effect, the application is running in a *virtual server*.

[1] Failover Clustering in Windows Server 2008 R2, *Microsoft White Paper*; April 2009.

An application runs in only one node at a time. However, the redundancy built into the cluster provides protection against any single component failure. Should a server, communication link, storage link, or application fail, the failure is automatically detected by the cluster service, which will move the failed application to a surviving node. Users may experience temporary degraded performance but will not completely lose access to their applications.

All hardware used in a WSFC cluster must be certified by Microsoft in order to obtain Microsoft support for the cluster. Certified hardware is listed in Microsoft's Hardware Compatibility List (HCL). Furthermore, Microsoft highly recommends that all nodes in a cluster be identically configured.

Resource Groups

Fundamental to the operation of a cluster is the notion of *resources* and *resource groups*.[2] A resource is a hardware or software component that is managed by the cluster service. Resources include application executables, disks, logical storage units, IP addresses, network names, and network interface cards (NICs). Every resource has a *resource monitor* that allows it to report its status to the cluster service and that allows the cluster service to query the resource status and to send directives to the resource to bring it online or take it offline.. The most important function of the resource monitor is to monitor the health of its resource and to report health changes to the cluster service.

A resource group is the group of resources that comprise an application. It includes all resources needed for an application. A resource group typically includes a set of application executables, one or more logical storage units (identified via LUNs, or logical unit numbers), an IP address, and a network name. Clients know the application only by its IP address or network name.

The cluster service treats a resource group as an atomic unit. A resource group can only be running in one node at a time. That node is said to *own* the resources of a resource group currently assigned to it. A node can be running (can own) several resource groups at any one time. That is, a node can be supporting several applications simultaneously.

Failover

Should an application be impacted by a hardware or a software fault, the cluster service can take one of several actions:

- It can attempt to restart the application on its owning node.
- It can move the resource group to another node in the cluster.
- If the problem is a node failure, it can move all resource groups currently owned by that node to other nodes in the cluster.

Each application can have a *preference list* indicating in which node it prefers to run and to which nodes it should fail over in preference order. It also species dependencies, indicating for each resource what other resources must first be available. When cluster service detects a failure, it determines to which node to move a failed resource group based on several factors, such as nodal load and preference. The resources of the resource group being moved are then started on the new node in the order specified by the resource-group dependencies.

When a node is restored to service and rejoins the cluster, all resource groups that have specified the restored node as their preferred node are moved back to that node.

[2] Server Clusters: Architecture Overview for Windows Server 2003, *Microsoft White Paper*; March 2003.

Since clients know their application only by its IP address or network name, and since these are the resources that are transferred to the new node upon failure, cluster component failures are transparent to the clients. They simply keep on using the application even though it is now running on a different node. One caveat is that session state and memory-resident application state will be lost following a failover. Therefore, a cluster provides high-availability but not fault tolerance.

Note that a resource group can only be owned by one node at a time. That means that LUNs can only be accessed by one node at a time. However, all nodes must have a connection to all LUNs that they may have to own following a failure. This requirement is satisfied by having all LUNs be resident in the shared storage provided by the SAN.

Though applications generally do not need to be modified to run in a cluster, "cluster-aware" applications can often take advantage of additional facilities built into the resource monitors for extended high-availability and scalability features.

Quorum

In addition to the resources that can be owned by nodes, a cluster has a very important common resource – the *quorum*. The quorum is a cluster configuration database that is hosted on shared storage and that is therefore accessible to all nodes. The configuration database includes such information as which servers are currently members of the cluster, which resources are installed in the cluster, and the current state of each resource. A node can participate in a cluster only if it can communicate with the quorum.

The quorum has two main functions:

Consistency

The quorum is a definitive repository of all configuration information related to the cluster. It provides each physical server with a consistent view of how the cluster is currently configured. It also provides the configuration information required by a node being returned to the cluster or by a new node being added to the cluster.

Arbitration

As in any multinode application network, a cluster is subject to the split-brain syndrome. If a network fault breaks the cluster so that there are two or more isolated groups of nodes, and if no action were taken, each of the isolated groups might conclude that it is the surviving remnant of the cluster and will take ownership of the resource groups owned by the nodes that it considers to have failed. Resource groups are now owned by multiple nodes in the cluster, leading to database corruption as independent and uncoordinated updates are made to the databases of the affected applications.

Split-brain operation must be avoided. This is a function provided by the quorum. It will detect that the cluster has been broken and will select the surviving cluster according to *majority*. Majority means that the surviving group of nodes selected to carry on the cluster functions must contain more than half of the nodes configured for the cluster. If there are *n* nodes, the surviving group must contain at least *n*/2+1 nodes. All of the other nodes will be removed from cluster membership, and this new configuration will be noted in the quorum-configuration database. The surviving group is said to have "quorum." If no group has quorum, the cluster is down; and it must wait for nodes to rejoin the cluster.

This leaves the problem of a cluster with an even number of nodes. If the cluster is evenly split, neither group has quorum; and the cluster is down. To avoid this, the quorum database can be

given a vote so that there are effectively an odd number of nodes, allowing a quorum to be established.

The Cluster Service

The cluster service is a collection of software components that run on each node and that perform cluster-specific activity. Cluster-service components interact with each other over the private network interconnecting the cluster nodes. The components include the following:

Node Manager

The Node Manager runs on each node and maintains a list of all nodes that belong to the cluster. It monitors the health of the nodes by sending heartbeat messages to each node. If it does not receive a response to a heartbeat message after a number of tries, it multicasts to the entire cluster a message requesting that each member verify its view of the current cluster membership. Database updates are paused until the cluster membership has stabilized.

If a node does not respond, it is taken out of service; and its active resource groups are moved to other operating nodes according to the preferences of each resource group.

Database Manager

The Database Manager runs on each node and maintains the cluster configuration database. This database contains information on all physical and logical entities in the cluster, such as the cluster itself, node membership, resource types and descriptions, and resource groups. This information is used to track the current state of the cluster and to determine its desired state.

The Database Managers cooperate to ensure that a consistent view of the cluster is maintained at each node. The Database Manager on the node making a configuration change initiates the replication of its update to the other nodes. Replication is atomic and serial and uses a one-phase commit. If a node cannot make an update, it is taken out of service.

Changes are also written to the quorum resource as a log for node-recovery purposes.

Failover Manager

The Failover Managers, which run on each node, work together to arbitrate ownership of resource groups following a component failure. They are responsible for initiating failover of resource groups and for starting and stopping resources according to the dependencies specified by each resource group.

If a resource fails, the Failover Manager in that node might try to stop and restart the resource. If this doesn't correct the problem, the Failover Manager stops the resource, which will trigger a failover of the failed resource group to another node. In the event of a resource-group move, the Failover Manager will update the configuration database via the Database Manager.

Failover may be triggered in response to an unplanned hardware or application fault, or it may be triggered manually by the cluster administrator so that a node can be upgraded. In the latter case, the shutdown is orderly. If failover is triggered by a component failure, shutdown can be sudden and disruptive. Should this happen, extra steps are required to evaluate the state of the cluster and the integrity of the application database before the failed resource groups can be returned to service on a surviving node.

When a node is returned to service and rejoins the cluster, the Failover Manager manages the failback of resource groups. It decides which resource groups to move to the recovered node

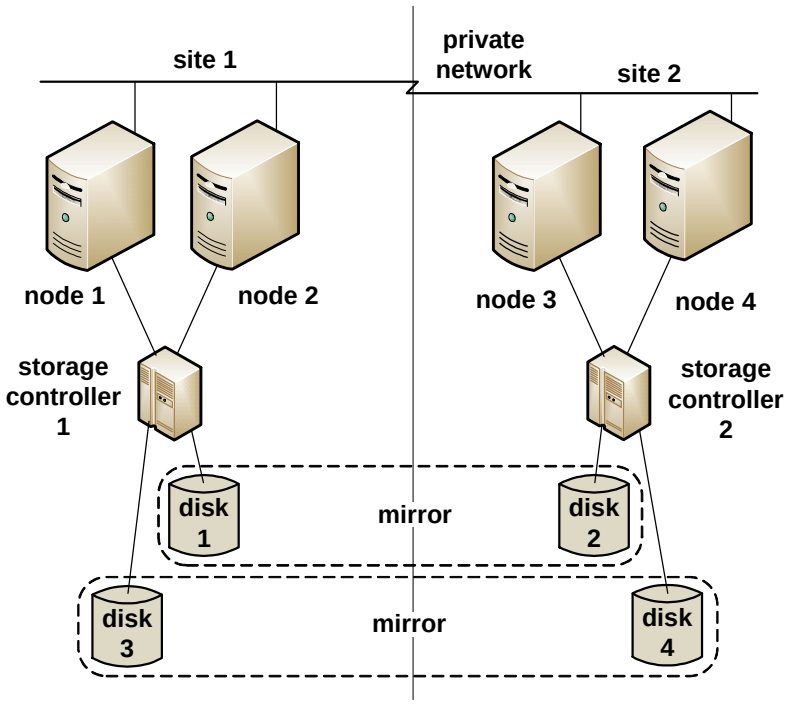
based on preferences. Moving resource groups to a recovered node can be restricted to certain hours to prevent mass movements during peak activity times.

The Failover Manager is also responsible for the backup and restoration of the quorum logs and other critical files.

Multisite Clusters

Though high-availability clusters reduce the impact of single-component failures, the cluster is still vulnerable to site-location disasters such as fires and floods. The only protection against site disasters is to have another cluster located far enough away that it is unlikely that any one disaster will affect both cluster sites. This can be achieved with geographically dispersed multisite clusters,[3] in which interconnected clusters are located at two or more geographically-separate sites.

In a multisite configuration, data replication must be used to ensure that both sites have an up-to-date view of all files and databases. Data disks may be optionally mirrored either asynchronously or synchronously, though the applications must be able to deal with some data loss if asynchronous replication is used. However, the quorum disk must be replicated synchronously to ensure a consistent view of the distributed cluster at any point in time. If asynchronous replication of application databases is used, distance is not a problem since the response time of a quorum update does not directly affect the performance of the applications. Clusters can be separated by hundreds of miles. If the application databases are synchronously replicated, the distance separating the sites is limited.



Replication may either be software-based at the host level or hardware-based at the SAN controller level. However, if SAN block replication is used, it must be guaranteed that the order of writes is preserved to maintain target database consistency.

[3] Geographically Dispersed Clusters, *Microsoft TechNet*; 2010.

The WSFC cluster service and clustered applications are unaware of geographical separation. All cluster functions are performed in the same way no matter where the cluster members are located. Microsoft does not provide a replication product for multisite clusters. Third-party products must be used, such as the NeverFail ClusterProtector,[4] which provides synchronous replication, fault detection, and remote-site failover services.

Geographically-dispersed cluster configurations supported by Microsoft appear in the Microsoft Hardware Compatibility List.

WSFC Enhancements over MSCS

WSFC has been significantly enhanced over MSCS in many areas.

Cluster Administration

A major challenge historically with clusters has been the complexity of building, configuring, and managing clusters. WSFC hides the clustering "nuts and bolts" behind a new GUI interface, the Failover Cluster Management snap-in for the Microsoft Management Console. Microsoft claims that a cluster expert is no longer needed to successfully deploy and maintain a cluster. These functions can now be performed by an IT generalist.

The Failover Cluster Management administration tool is task-oriented rather than resource-oriented and simplifies administration via several new wizards. For instance, with MSCS, in order to create a highly available file share, the administrator had to create a group, create a disk resource, create an IP address, create a network name, configure heartbeat messages, establish a preferred node list, and specify resource dependencies. With WSFC, all the administrator has to do is to specify a network name. The High Availability Wizard does the rest.

With Failover Cluster Management, an administrator can maintain multiple clusters in the organization. Clusters can be managed remotely via the Remote Server Administration Tools.

For those experienced cluster administrators who want to further tune the cluster configuration, the MSCS cluster.exe commands are still available and allow full access to all MSCS administrative capabilities. However, the cluster.exe commands will be replaced with new Windows PowerShell Cmdlets in later versions.

Scalability

Under MSCS, the maximum number of nodes that could be in a cluster was eight. WSFC has increased this limit to sixteen x64 nodes in a cluster.

Security

Kerberos is now used for user authentication. All communication between nodes is signed and may be encrypted.

[4] NeverFail ClusterProtector, http://extranet.neverfailgroup.com/download/DS-cluster-08-09-4page-lo.pdf.

Networks

Microsoft strongly encourages the use of redundant, separate, and distinctly routed networks for providing fault tolerance in the private network connecting the nodes. If this is not provided, WSFC will generate a warning message; and the cluster may not be accepted by Microsoft for support.

In the case of redundant networks, the fastest network will be given priority for internal traffic.

Under MSCS, the maximum allowed latency over the private network was 500 milliseconds. This was due to heartbeat limitations. Heartbeat intervals were not configurable. In WSFC, heartbeat parameters are configurable; and the latency restriction has been removed. In addition, rather than broadcasting heartbeats, WSFC now uses TCP/IP connections to improve heartbeat reliability. IPv6 as well as IPv4 is supported.

Under MSCS, the cluster members at both sites in a geographically-dispersed multisite cluster had to be on the same subnet. This meant that the private network interconnecting the two sites had to be a VLAN (virtual LAN) stretched over a WAN communications link. This restriction has been removed by WSFC. The cluster members at each site can now be on different subnets connected by a simple (redundant) WAN link. No VLAN needs to be created.

Hyper-V Integration

WSFC is integrated with Microsoft's Hyper-V virtualization services. Any node in the cluster may host virtual machines, and virtual machines may be failed over individually or en masse. Live migration of virtual machines commanded by the administrator occurs within milliseconds with no perceived downtime and with no lost connections.

Validation

The Validate a Configuration Wizard can be run to ensure that a cluster configuration will be supported by Microsoft. It validates all hardware components against Microsoft's Hardware Compatibility List and validates the cluster configuration. It is useful not only when a cluster is created, but it can also be used to periodically validate the cluster configuration.

Rolling Upgrades

Nodes may be upgraded by removing them one at a time from the cluster, upgrading them, and then returning them to the cluster. However, migration from MSCS clusters is not specifically supported. The Migration Wizard is available to help these migrations.

Summary

The WSFC cluster service monitors cluster health and automatically moves applications from a failed node to surviving nodes, bringing high availability to critical applications. WSFC also brings high availability to Microsoft's Hyper-V virtualization services.

WSFC brings many enhancements to the stalwart MSCS clustering services. With WSFC, up to sixteen Windows servers can be organized into a multisite, geographically-dispersed cluster with cluster sites separated by hundreds of miles. A convenient GUI administrator tool supported by several wizards removes the need for a cluster specialist to configure and manage the cluster.

WSFC makes cluster technology even more attractive to small businesses and large enterprises alike.