Realtime publishers

The Essentials Series: Configuring High Availability for Windows Server 2008 Environments

Windows Server 2008 Native Technologies

by Richard Siddaway

Copyright Statement

Windows Server 2008 Native Technologies

In Article 1, we explored the reasons for needing high availability in our Windows Server 2008 environments. This article will examine the high-availability options that are native to Windows. We will discuss the advances in clustering that Windows Server 2008 brings (including Windows Server 2008 R2) and look at some of the possible obstructions to adopting the high-availability technologies that are "out-of-the-box" in Windows Server 2008.

High-Availability Options

The standard approach for high availability is clustering using the Microsoft Cluster Service (MSCS). However, we can also achieve high availability through the use of multi-instance applications and Network Load Balancing (NLB).

Multi-Instance

The main examples of high availability through a multi-instance approach in a Windows Server 2008 environment are Active Directory (AD) and DNS. In both cases, there are a number of servers that can supply the functionality. Data is automatically replicated between the domain controllers and DNS servers. Client machines are configured to use more than one of the possible targets. This approach works in a limited number of cases. It also supplies a disaster recovery capability in that the replication can be across sites. We still need to back up the data!

Network Load Balancing

NLB is used to provide load balancing and high availability across a number of servers. The applications used in this scenario are usually Web based, that is, based on HTTP or HTTPS. However, it is possible to load balance other TCP protocols.

The nodes of the NLB cluster will balance the traffic to the cluster between themselves. If a node fails, the traffic will be re-distributed amongst the remaining nodes of the cluster. NLB can be used also to provide high availability for applications such as ISA server or SQL Server (in read-only mode). There are some applications that won't work with NLB and need hardware load balancing instead.

Clustering

The majority of our high availability instances will involve clustering of Windows servers. These clusters could support databases, email systems, or other business-critical applications. We could cluster our Web servers, though we wouldn't be able to put as many nodes into the cluster as we could with NLB and we wouldn't gain load balancing.

In many organizations, the standard approach is to build a two-node cluster for each application. One node is active and the other is passive waiting to host the application in the event of failure on the first node. This setup ensures that each application has a dedicated failover path and that the correct level of resources (memory and CPU) is available for the application.

The maximum number of nodes supportable in a Windows Server 2008 failover cluster depends on the version of the operating system (OS) that is being used:

- 32-bit versions of Windows Server 2008 Enterprise and Datacenter Editions can support up to 8 nodes in a failover cluster (same as Windows Server 2003)
- 64-bit versions of Windows Server 2008 Enterprise and Datacenter Editions can support up to 16 nodes in a failover cluster

Windows Server 2008 R2 is 64-bit only so it will support up to 16 nodes in a failover cluster. It is not possible to support both 32- and 64-bit servers in the same cluster.

These figures suggest that it may be possible to reduce the cost of clustering by "sharing" the passive nodes. By this, I mean build a multi-node cluster with *m* active nodes and *n* passive nodes, where $m + n$ is less than 16 (or 8 if 32 bit). Most new clusters will be 64 bit, as email and database systems can take advantage of the memory that becomes available through using a 64-bit OS.

The "m+n" approach maximizes the use of the cluster resources, but it makes management a little more difficult. In a two-node cluster, the second node is the only failover target. In an "m+n" configuration, there are a number of failover targets available. The possible failover nodes for each application need to be managed to ensure that resources are available and all the applications don't end up on a single node.

Organizations have attempted to reduce the cost of failover clustering by running the cluster in an active-active configuration. In this case, both nodes are running an application, often database instances, and they are configured to failover to the other node. This can lead to both applications on the same node with an adverse impact on performance. Active-active cluster configuration is not recommended, and modern applications are appearing that no longer support it.

We have seen that Windows Server 2008 can support more nodes in a failover cluster than previous versions of Windows. What else is new?

Windows Server 2008 Advances in Clustering

Failover clustering has been available in the Windows OS since two-node clusters were introduced to Windows NT 4.0. One issue with failover clustering in earlier versions of Windows has been the restrictions on hardware that was supported when clustering.

Hardware Restrictions

The cluster configuration of servers and storage had to be on the approved, and tested, list before the cluster configuration was fully supported. The Hardware Compatibility List (HCL) and Windows Server Catalog provided this information. One major issue was the support of hardware drivers. If the drivers hadn't been tested and approved, the cluster couldn't be upgraded to the new drivers.

This has changed, as we will see. There are still hardware restrictions, but they are of a more "common-sense" variety than hard-and-fast rules. When creating a cluster, it makes sense to use matching servers. There is an argument that says that the passive node could be less powerful than the active node because the passive node won't really be used much. This is a false economy. In the event of failover, the passive node may become the only node available in a two-node cluster. We don't want business-critical systems suffering performance problems.

Use identical servers when building the cluster. The servers should also be as resilient as possible with as much redundancy built in as possible, for instance fans, power supplies, and network cards.

Configuration Validation

A Windows Server 2008 cluster is self-tested and validated. Hardware should still be certified for Windows (all major manufacturers do this) and building clusters with servers from different manufacturers is not a recommended practice.

Clustering is now a feature for Windows Server 2008 rather than being treated as a service. Once the cluster servers, and storage, are assembled, the Failover Clustering feature can be enabled on each node. A validation wizard is then run from the Cluster Management MMC. The wizard will ask for the names of the servers in the cluster that the validation process will examine:

- Cluster nodes
- Network configuration
- Storage
- System configuration

There are a few essentials to remember before starting the validation wizard:

- The nodes have to be members of the same AD domain
- Do not mix domain controllers and member servers in the same cluster; it is still recommended not to use domain controllers
- Each node needs at least two network adapters
- If multipath I/O is supported, it will be tested
- Storage should use the SCSI-3 standards rather than SCSI-2
- Ensure all servers are the same service pack and patching level
- Ensure all drivers are signed

After testing, the results are saved to all nodes in the cluster. The cluster configuration will be supported by Microsoft if it passes the validation wizard's testing.

Configuration Choices

In previous versions of Windows, all IP addresses for a cluster had to be static. It is possible now to use DHCP-supplied addresses for a cluster. If this practice is adopted, ensure that the addresses are reserved in DHCP.

Windows Server 2003 and earlier versions had a single point of failure in terms of the quorum disk. This disk had to be available for the cluster to continue as it determined which node controlled the cluster. Failure was not a common occurrence, as clusters usually use SAN storage with its higher reliability. The use of the quorum disk was the most common scenario due to the prevalence of two-node clusters. Windows Server 2003 introduced the majority node set model where each node has the quorum resource replicated to local storage. This model provides better resiliency at the cost of reduced flexibility in terms of an increase in the number of nodes that must be online for the cluster to function.

Windows Server 2008 combines these models into a majority quorum model. Each node in the cluster plus the quorum (now known as the witness) resource is assigned a vote. A majority of votes controls the cluster. If only the witness resource is assigned a vote, the configuration duplicates the Windows Server 2003 quorum disk behavior, alternatively only assigning votes to the nodes duplicating the majority node set configuration.

The witness can be a separate disk or even a file share on a server outside of the cluster. This share can't be part of a DFS environment. It is possible to change the quorum model after the cluster has been created, but this is not recommended.

Geographically Distributed Clusters

We have already seen some of the networking changes that failover clustering in Windows Server 2008 introduces. The biggest change is that the cluster nodes no longer need to be on the same logical subnet. This restriction has been lifted, enabling us to create geographically dispersed clusters without the need for VLANs spanning our sites.

The heartbeat timeout between the cluster nodes is configurable, which means that the network latency (within reason) doesn't become an issue for a dispersed cluster. At first sight, this may seem to solve our high availability and disaster recovery issues in one go. However, there are still a few points to consider:

- Networking—Can failover occur across the network? If the failure involves the network router, for instance, failover can't happen and the cluster nodes in the primary data center are unavailable.
- Data—How will the data be replicated between the data centers? Will it be up to date when a failover occurs?
- Control and Management—Will there be issues because the nodes are in different data centers with, possibly, different administrators? Will patching and other maintenance occur at the right time?

New features in Windows Server 2008 R2

Windows Server 2008 R2 brings some further enhancements to failover clustering:

- Cluster Shared Volumes allow multiple nodes in the same cluster to access the same LUN. This is also used for the Hyper-V Live Migration feature.
- A Best Practice Analyzer is built-in to the cluster validation tool to ensure our new clusters are built according to best practice. The tool can be run periodically to confirm we are still adhering to best practice.
- Command-line setup. PowerShell becomes the command-line method of creating and administering clusters. It is possible to create a new cluster with a single line of PowerShell script.

Obstructions to Native Clustering

The enhancements to failover clustering are welcome, but there are still some obstructions to using native clustering:

- Cost—The cost of a clustered solution can be high when the cost of the SAN and passive nodes are taken into account.
- Skills—The mix of skills to create and manage the cluster hardware, storage, and application can be difficult to obtain. Bringing in contractors to create the cluster leaves us with a system that is not understood and is effectively unsupportable.
- Application Suitability—Can the application be configured easily to give high availability in a cluster? Does the cost rise for a clustered version? Will the clients automatically reconnect after a failover?
- Failover time—Is it acceptable? Some large clusters with large arrays of storage can take several minutes to fail over.

Summary

The primary high-availability option within Windows Server 2008 is failover clustering. This is enhanced and easier to work with compared with previous versions. There are still some obstructions to the use of native high-availability options. We will see possible solutions to these obstructions in the next article.