Realtime publishers

The Essentials Series

Configuring High Availability for Windows Server 2008 Environments

sponsored by

MARATHON
Run to Infinity

by Richard Siddaway

Copyright Statement

Article 1: The Art of High Availability

All organizations are becoming increasingly reliant upon their computer systems. The availability of those systems can be the difference between the organization succeeding and failing. A commercial organization that fails is out of business with the consequences rippling out to suppliers, customers, and the community.

This series will examine how we can configure our Windows Server 2008 environments to provide the level of availability our organizations need. The topics we cover will comprise:

- The Art of High Availability—What do we mean by high availability? Why do we need it, and how do we achieve it?
- Windows Server 2008 Native Technologies—What does Windows Server 2008 bring to the high-availability game, and how can we best use it?
- Non-Native Options for High Availability—Are there other ways of achieving high availability, and how can we integrate these solutions into our environments?

The first question we need to consider is why we need highly available systems.

Why Do We Need It?

This question can be turned on its head by asking "Do all of our systems need to be highly available?" The answer for many, if not most, organizations is no. The art of high availability comes in deciding which systems need to be made highly available and how this is going to be achieved. When thinking about these systems, we need to consider the effects of the systems not being available.

Downtime Hurts

Downtime is when the system is unavailable to the user or customer and the business process cannot be completed. If the server is up and the database is online but a network problem prevents access, the system is suffering downtime. Availability is an end-to-end activity. Downtime hurts in two ways: If a system is unavailable, the business process it supports cannot be completed and there is an immediate loss of revenue. This could be due to:

- Customer orders not being placed or being lost
- Staff not working
- Orders not being processed

The second way that downtime hurts is loss of reputation. This loss can be even more damaging in the long term if customers decide that your organization cannot be trusted to deliver and they turn to a competitor. The ability to gain business increases with ease of communication and access. The converse is that the ability to lose business increases just as fast if not faster.

Critical Systems on Windows

Critical business systems are hosted on the Windows platform. These can be customer facing or internal, but without them, the business grinds to a halt. Email may not seem to be a critical system, but it is essential to the modern business. More than 60% of person-to-person communication is via email in most businesses. This includes internal and external communications. If a company is non-responsive to communications, it is judged, perhaps harshly, as being out of business. This can become reality if it progresses too long.

24 × 7 Business Culture

The "Global Village" concept has been accelerated by the adoption of the Internet for business purposes. Globalization in this case means that business can come from anywhere in the world—not necessarily your own time zone. If your business competes at this level, high availability isn't an option, it's a necessity.

Legislation

Industries such as the financial services and health sector have a requirement to protect the data they store. This requirement can involve the availability of the data. In other cases, the systems must be highly available to meet safety requirements.

Once you know why you need it, you need to define what is meant by high availability.

What Is High Availability?

High availability is usually expressed in terms of a number of "9"s. Four nines is 99.99% availability. The ultimate goal is often expressed as 5 "9"s availability (99.999%), which equates to five and a quarter **minutes** of downtime per **year**. The more nines we need, the greater the cost to achieve that level of protection.

One common argument is scheduled downtime. If downtime is scheduled, for example, for application of a service pack, does that mean the system is unavailable? If the system is counted as unavailable, any Service Level Agreements (SLAs) on downtime will probably be broken. In hosting or outsourcing scenarios, this could lead to financial penalties. However, if scheduled downtime doesn't mean the system is counted as unavailable, impressive availability figures can be achieved—but are they a true reflection of availability to the users? There is no simple answer to these questions, but all systems require preventative maintenance or they will fail. The disruption to service can be minimized (for example, the patching nodes of a cluster in sequence) but cannot be completely eliminated. Probably the best that can be achieved is to ensure that maintenance windows are negotiated into the SLA.

These measurements are normally taken against the servers hosting the system. As we have seen, the server being available doesn't necessarily mean the system is available. We have to extend our definition of highly available from protecting the server to also include protecting the data.

The Clustering Service built-in to Windows is often our first thought for protecting the server. In the event of failure, the service automatically fails over to a standby server, and the business system remains available. However, this doesn't protect the data in that a failure in the disk system, or even network failures, can make the system unavailable.

> **Cross-Reference**
> We will return to these ideas in Articles 2 and 3.

Do We Still Need to Back Up?

One common question is "Do I still need to take a backup?" The only possible answer is YES! High availability is not, and never can be, a substitute for a well-planned backup regimen. Backup is your ultimate "get out of jail card." When all else fails, you can always restore from backup. However, this pre-supposes a few points:

- Test restores have been performed against the backup media. The last place you want to be is explaining why a business-critical system cannot be restored because the tapes cannot be read.
- A plan exists to perform the restore that has been tested and practiced. Again, you don't want to be performing recoveries where the systems and steps necessary for recovery are not understood.

Backup also forms an essential part of your disaster recovery planning.

Disaster Recovery vs. High Availability

These two topics, high availability and disaster recovery, are often thought of as being the same thing. They are related but separate topics. High availability can be best summed up as "keeping the lights on." It is involved with keeping our business processes working and dealing with day-to-day issues. Disaster recovery is the process and procedures required to recover the critical infrastructure after a natural or man-made disaster. The important point of disaster recovery planning is restoring the systems that are critical to the business in the shortest possible time.

Traditionally, these are two separate subjects, but the technologies are converging. One common disaster recovery technique is replicating the data to a standby data center. In the event of a disaster, this center is brought online and business continues. There are some applications, such as relational database systems and email systems, that can manage the data replication to another location. At one end of the scale, we have a simple data replication technique with a manual procedure required to bring the standby data online in place of the primary data source. This can range up to full database mirroring where transactions are committed to both the primary and mirror databases and failover to the mirror can be automatically triggered in the event of applications losing access to the primary. In a geographically dispersed organization where systems are accessed over the WAN, these techniques can supply both high availability and disaster recovery.

We have seen why we need high availability and what it is. We will now consider how we are going to achieve the required level of high availability.

Achieving High Availability

When high availability is discussed, the usual assumption is that we are talking about clustering Windows systems. In fact, technology is one of three areas that need to be in place before high availability works properly:

- People
- Processes
- Technology

People and Processes

These are the two points that are commonly overlooked. I have often heard people say that clustering is hard or that they had a cluster for the application but still had a failure. More often than not, these issues come down to a failure of the people and processes rather than the technology.

The first question that should be asked is "Who owns the system?" The simple answer is that IT owns the system. This is incorrect. There should be an established business owner for all critical systems. They are the people who make decisions regarding the system from a business perspective—especially decisions concerning potential downtime. A technical owner may also be established. If there is no technical owner, multiple people try to make decisions that are often conflicting. This can have a serious impact on availability. Ownership implies responsibility and accountability. With these in place, it becomes someone's job to ensure the system remains available.

A second major issue is the skills and knowledge of the people administering highly available systems. Do they really understand the technologies they are administering? Unfortunately, the answer is often that they don't. We wouldn't make an untrained or unskilled administrator responsible for a mainframe or a large UNIX system. We should ensure the same standards are applied to our highly available Windows systems. I once worked on a large Exchange 5.5 to Exchange 2003 migration. This involved a number of multi-node clusters, each running several instances of Exchange. One of the Exchange administrators asked me "Why do I need to know anything about Active Directory?" Given the tight integration between Exchange and Active Directory (AD), I found this an incredible question. This was definitely a case of untrained and unskilled.

Last, but very definitely not least, we need to consider the processes around our high-availability systems. In particular, two questions need to be answered:

- Do we have a change control system?
- Do we follow it?

If the answer to either of these is no, our system won't be highly available for very long. In addition, all procedures we perform on our systems should be documented and tested. They should always be performed as documented.

Technology

Technology will be the major focus of the next two articles, but for now, we need to consider the wider implications of high availability. We normally concentrate on the servers and ensure that the hardware has the maximum levels of resiliency. On top of this, we need to consider other factors:

- Network—Do we have redundant paths from client to server? Does this include LAN, WAN, and Internet access?
- Does the storage introduce a single point of failure?
- Has the operating system (OS) been hardened to the correct levels? Is there a procedure to ensure it remains hardened?
- Does our infrastructure in terms of AD, DNS, and DHCP support high availability?
- Does the application function in a high-availability environment?

Costs

Highly-available systems explicitly mean higher costs due to the technology and people we need to utilize. The more availability we want, the higher the costs will rise. A business decision must be made regarding the cost of implementing the highly-available system when compared against the risk to the business of the system not being available.

This calculation should include the cost of downtime internally together with potential loss of business and reputation. When a system is unavailable and people can't work, the final costs can be huge leading to the question "We lost how much?"

Summary

We need high availability to ensure our business processes keep functioning. This ensures our revenue streams and business reputation are protected. We achieve high availability through the correct mixture of people, processes, and technology.

Article 2: 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.

Article 3: Non-Native Options for High Availability

The previous two articles in this series covered the need for high availability and how we can satisfy that need with the native Windows technologies. There are situations where those technologies do not meet our needs and we have to look to third-party or non-native solutions. Those options will be covered in this article.

Suitability and Cost

There are a number of possibilities for supplying high availability to our systems. We must remember that not all options are suitable for a given scenario and that "just because we can doesn't mean we should." By this, I mean that we match the solution to our needs and don't apply a technology just because it seems a good thing to experiment with. When shopping for a solution, we must remember the criticality of the systems we want to protect and whether the cost of downtime justifies the cost of the solution.

We must also think of the skills available to our IT department. In many cases, these solutions are ideal for an organization with limited IT skills and presence. This could be small to midsize organizations or the "Branch Office" scenario in a larger, more distributed environment.

Data Replication

We have seen that for true high availability, we need to protect the server and data. One method of protecting the data is by using storage-controlled replication. This is sometimes referred to as *block-level replication*.

The concept is simple in that two sets of storage are linked across the network. Changes that occur on the primary are replicated to the secondary storage. The replication works at the disk block level to minimize the replication traffic. Data replication of this sort involves additional software for controlling the storage and replication. If both sets of storage are linked to the nodes of the cluster, it is possible for the storage to failover to the secondary in the event of a failure in the primary storage.

Although it might seem to be an ideal solution, there are some downsides to consider. The first potential issue is latency. Any delay in replication invites a situation where a failure means that the data on the secondary storage is incomplete, which could lead to errors in the business process. If the replication occurs continuously, there is the possibility that corrupt data will be replicated between the storage units. The network links between the storage units can contribute to the latency and need to be resilient to ensure a network failure doesn't prevent replication.

One other potential issue we need to consider is transactional consistency. If we are replicating database files in this manner, we have to ensure that the database transactions are replicated correctly so that the database will be in a consistent state in the event of failover. Storage-based replication can be used as part of a virtualization solution to enable cross-site high availability.

Virtualization

Virtualization is a major part of a modern infrastructure. Server virtualization is the first thought when this topic is mentioned, but we can also virtualize applications, application presentation, storage, and the desktop. Varying degrees of high availability can be achieved using these solutions.

Server Virtualization

Server virtualization in a production environment will consist of one, or more, hosts that are connected to storage. The hosts will run a hypervisor that enables them to have a number of virtual servers as guests. Each guest has at least one file containing the operating system (OS) and data. Virtualization enables us to make best use of our hardware, but if that hardware fails, we could lose more systems than we would in a physical environment. High availability is achieved by configuring the virtual hosts in a cluster.

Virtualization can supply high availability in a number of ways:

- In the event of a failure in the virtual server, it can be automatically restarted. This works well for some simple failures, but in the event of a failure in the OS, or application, the virtual server could immediately shut down again.
- If the virtual host hardware fails, it is possible for the virtual server to be moved to another host within the cluster. There may be downtime associated with this move, but it should be minimal, especially if the move is performed automatically.
- One advantage of a virtual environment is that if a host becomes overloaded, it is possible for virtual servers to be migrated to another host in the cluster that has a lighter workload. Moves of this sort can occur without downtime. This has the advantage of minimizing issues of servers being slow to respond because they are bottlenecked for resources.
- The movement of virtual servers to other hosts enables us to perform maintenance tasks on a host at minimal cost on planned downtime as far as the guest is concerned.

If we combine virtualization with the data replication we have already discussed, we could have a highly-available system that spans data centers, though failover wouldn't be automatic.

Application Virtualization

Application virtualization may seem out of place in this article, but high availability has to be considered as an end-to-end scenario. There is no point making the server resilient if the client keeps failing.

The application can be hosted on multiple servers and presented to the client machine as part of a remote desktop or as a single remote application. These "Terminal Servers" actually run the application with the client being used for display and to send input. The server farm can be configured to route the request to run the application to the most appropriate server. This provides high availability by preventing a single point of failure existing for the client application. There are also administrative gains to this approach, especially when considering installation and patching.

Unfortunately, all applications can't be delivered in this manner. We can adopt application virtualization and stream the applications from a server as required. An alternative approach is to virtualize the desktop and use the high-availability features we discussed earlier for those virtual machines. We can even let the application control our high availability, though these are server-side applications rather than client applications.

Application-Controlled High Availability

Recent versions of Microsoft Exchange Server and Microsoft SQL Server have introduced a number of high-availability options that don't directly use the Windows clustering features. Other database systems have high-availability features that fit this pattern.

Microsoft Exchange Server

In Microsoft Exchange Server 2003 and earlier, the only high-availability option for mailbox servers was clustering. We could use NLB for front-end servers but the primary target for resiliency was the mailbox servers.

Exchange 2007 changed the game by providing a number of replication-based high-availability options. Clustering is still available to protect the mailbox server, but there are other options that will protect the data and in some instances both server and data.

Mailboxes are stored in databases by Exchange. These databases are grouped into storage groups where the databases share log files. However, for the replication techniques to work, we need to limit ourselves to a single database per storage group. We may be configuring replication at the storage group level, but in reality, we are working with the databases. This is a lead into Exchange 2010 where storage groups don't exist and we only work with databases.

There are three replication techniques we can use with Exchange 2007:

- Local Continuous Replication (LCR) in which a second set of disks on the same server is used as a replication target. This protects the data, but if the server fails, both instances are unavailable.
- Cluster Continuous Replication (CCR) has the second set of storage on a different server. The clustering features of Windows are used for the heartbeat facility, as there is the capability of automatic failover between the two instances of the mailbox databases. Windows Server 2008 geographic clustering can be used so that the replication target is in a different data center, which provides disaster recovery as well as high availability.
- Standby Continuous Replication (SCR) uses a log shipping technique to replicate the mailbox database to another server. This can also be in a different data center. Automatic failover is not possible using this technique, but we gain the advantage of being able to run other Exchange roles on the server. This is a very good disaster recovery technique.

These techniques are taken a stage further with Exchange 2010. Clustering is not offered as a high-availability option! The CCR and SCR options of Exchange 2007 are combined into a replication scheme that can support instances of the same database on multiple servers. Any server hosting a copy of the database can make it available to clients in the event of the primary failing. This technique is also used by other email systems.

As we saw in Article 1, there is a convergence of disaster recovery and high-availability techniques. It has been suggested that if there are sufficient replicas of a database in an environment, backups could be ignored. Personally, I think it will be a long time before I stop backing up my mailbox databases.

Microsoft SQL Server

SQL Server 2008 still supports clustering as a high-availability option. It does, however, also supply a number of options that can be used for high availability and\or disaster recovery. These options are conceptually very similar to the techniques we discussed for Exchange:

- Database mirroring will mirror all changes to a replica of the database on another server. Several configurations are available, including one that provides automatic failover to the mirror. The client application must be correctly written to take advantage of this feature.
- There are several purely replication-based techniques that can produce copies of all, or some, of a database on another server. Transactional replication is probably best used as a high-availability technique. This will replicate all transactions performed on the database to one, or more, targets. Automatic failover isn't available, but a manual procedure would get the database back online very quickly.

- Log shipping involves taking a backup of the transaction log, copying the backup to another server, and restoring the backup to replay the transactions. The database is in a non-usable state during this process and a manual process is needed for failover.

These last two techniques may be regarded as better suited for disaster recovery, but if a small amount of downtime can be tolerated, they would make acceptable high-availability options. So far, we have looked at alternative ways to protect the data we use and to protect the server. There is one option left to look at that achieves both of these goals.

Synchronized Systems

This solution will effectively combine two Windows servers and present them to the world as a single server. The servers are monitored and tested by the synchronization software, and if one server fails, the other is available as an exact duplicate to continue providing the applications to support the business. The two servers are completely synchronized at the OS, application, and data levels by ensuring that changes happen to both servers simultaneously.

The advantage of this approach is that it is a single solution that can be implemented without needing a high degree of skill with the individual components that make up our systems. Unlike clustering, applications do not need to be aware that they are running in this environment. Thus, the same install and configuration is used whether the application is on a single server or a synchronized environment.

Data is included in the synchronization process so that it is automatically protected. The technique can cover physical or virtual servers and can be extended to provide a disaster recovery capability by spanning different physical sites. As the servers are continually synchronized, there is very rapid failover, no manual procedures, and no issues about data replication to remote sites within any distance limitations imposed by network latency.

Option	Advantages for Using for High Availability	Obstructions for Using for High Availability
Microsoft Native Clustering	• "In the box" • Automatic failover • Easier setup and management compared with earlier versions • Geographically distributed clusters • Integration with Microsoft applications	• Costs when factoring in SAN and passive nodes • Clustering management skills required • Cluster-aware versions of applications may be required • Failover can take several minutes in clusters with large storage arrays
Data Replication	• Cross-site capabilities become part of disaster recovery solution • Data is protected	• Cost • Complexity • Skill requirements • Manual failover
Virtualization	• Multiple options • Automatic failover • Workload balancing • Minimizes planned downtime	• Additional complexity • Need data replication for cross site
Application-Controlled High Availability	• Protects data and service • Under the control of the application team • Provides disaster recovery capability as well	• Additional servers and storage mean higher cost • May not be automatic failover
Synchronized Systems	• Single solution • Minimal setup and administration skills required • Automatic failover	• Cost of additional hardware and software

Table 1: Windows Server 2008 high-availability options at a glance.

Future of High Availability

High availability is going to be an increasing requirement as businesses require, and expect, that their processes will be available at all times. Market pressures will penalize organizations that can't supply access to their systems when the customer requires it.

Early thinking about high availability in a Windows environment focused on protecting the server and the data, but it is now recognized that the data is at least as important. High-availability techniques are evolving to ensure the data is protected. This will continue, and simple clustering may become a thing of the past.

The high-availability solutions of the future must be easy to install, configure, and maintain. "It's too hard" cannot be allowed as an excuse for downtime.