Realtime
publishers

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

Non-Native Options for High Availability

by Richard Siddaway

Copyright Statement

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.