

# Windows Server Multi-site Clustering With Double-Take® Availability

**Technical Whitepaper**  
**Double-Take Software, Inc.**  
**Published: August 2009**

## Introduction

Applications, like all man-made creations, fail; it's just a fact of life in the information technology world. The goal of the IT department is to mitigate the risk of downtime and prepare for recoverability of critical applications in an efficient manner. There are many different methods for restoring application service depending on the criticality of the application including rebuilding servers and restoring from backups, which may take days to complete. For business critical applications, faster recovery is imperative to keeping the business processes which depend on them going. Most organizations consider applications like e-mail, financial accounting software, ERP and CRM applications to be the backbone of their business processes. So what's an IT administrator to do?

Failover Clustering was created to address this fast recovery requirement by providing redundant servers that can resume processing as quickly as possible. The Microsoft Windows Server operating system has included clustering capabilities for many years. One of the challenges of most cluster technologies, however, is the requirement for having all servers attached to the same data storage devices. While the servers themselves are redundant, the shared storage requirement results in a single point of failure. Worse still, because of their close proximity, traditional single-site clusters don't protect the applications or data from regional disruptions like earthquakes, floods and hurricanes.

As part of its Workload Optimization suite of products, Double-Take Software provides market-leading solutions for data protection and recoverability. The GeoCluster feature, part of Double-Take® Availability, was created specifically to protect, extend and build upon the capabilities of Failover Clustering by providing redundant storage that eliminates the single point of failure inherent in traditional clusters as well as providing off-site cluster failover capabilities. Now data can be replicated to different storage devices locally or remotely and allow clustered applications to remain running regardless of the point of failure.

Microsoft Windows Server 2008 includes new features that provide significant improvements for managing clusters in multi-site architectures. For example, Windows Server 2008 Failover Clustering removes the dependency of having all cluster nodes on the same subnet. This eliminates the complexity of creating stretch VLANs across the wide area network which was required by previous editions of Failover Clustering. In addition, Windows Server 2008 includes a new quorum capability which stores a copy of the quorum outside of the cluster on a file share witness that acts as the tie-breaking vote during quorum arbitration. This makes it possible to place cluster nodes and the witness node at separate sites for additional redundancy in multi-site cluster architectures.

However, improvements to Failover Clustering do not yet provide a way to replicate the shared data used by a cluster which is truly the "last mile" in terms of building multi-site cluster architecture. GeoCluster provides a way to efficiently replicate these data changes and allow cluster nodes to quickly restore application services after a failure. Together, Windows Server 2008 Failover Clustering and GeoCluster provide the ultimate geographically distributed application recoverability solution for clustered applications.

## How Clustering Works

Failover clusters are composed of two or more servers (referred to as nodes) working together to run applications and resume operation if a cluster node fails. Traditional cluster implementations provide the ability to recover an application quickly by starting the failed application services on a remaining node, attaching to its data and resuming client connectivity so users can continue working. There are two primary types of failover clusters: those that have shared resources between them (typically their data storage is shared) and those that don't share resources.

### Shared Storage Clusters

Clusters that share their storage between nodes typically use multi-channel SCSI arrays or Storage Area Networks to create a shared connection to the application data. One node will own the data for read/write access at any given time, this node is referred to as the Active Node. This exclusive access prevents the same application on other nodes from trying to write to that same disk without the active node's knowledge - which would corrupt the data. In shared storage clusters disk ownership is typically performed using a 'first acquired' methodology. This ownership method is also used in quorum arbitration as a tie-breaker, so whichever node claims the quorum first will have one more vote than the other nodes and take ownership of the cluster.

This type of failover cluster architecture is known as an Active/Passive cluster; however, you can install other applications on the cluster with its own data storage that won't conflict with the other application. This is an Active/Active cluster since both nodes are running different applications at the same time and with access to the same shared storage array. If one node fails then that application can be restarted on the remaining node and attach to its data storage and continue running. The application can access the disks on the other node because the original node no longer has it locked for exclusive access since that node failed. Many applications like SQL Server allow you to install multiple instances of the application, storing data on different disks. For example, you could have one SQL Server that manages an accounting database while the other SQL Server instance manages an ERP database. While both nodes are running the same application, they are both managed independently and their data is logically separated.



Figure 1: Shared Storage Cluster

### Shared Nothing Clusters

The biggest problem with shared storage clustering is that it has a single point of failure within the storage sub-system. If the disk storage fails then none of the nodes can access the shared data and all of the applications on that cluster will fail. SANs (Storage Area Networks) were created in part to combat this problem by having the storage protected in different arrays, but implementations and success vary between vendors. Another problem is storage vendor lock-in since competing vendors' solutions are incompatible with each other. This is why Shared Nothing Clusters were created to guarantee that a failure in the storage sub-system wouldn't cause the entire cluster to fail.

Shared Nothing Clusters, just as their name states, don't share any resources between their nodes. They each have their own storage and can continue processing if the storage attached to other nodes fails. Data is typically replicated between the different nodes' storage to facilitate failover of applications between nodes. So if a SQL Server commits transaction 'A' before the node fails, when it restarts on another node then transaction 'A' will be there waiting for it to continue.

However, quorum arbitration is different in Shared Nothing Clusters because each node has an independent quorum disk that it can claim. In this scenario, quorum arbitration can't occur solely in the storage sub-system layer because there is no shared bus and arbitration must be done by some other external mechanism.

GeoCluster removes the shared disk restrictions of Microsoft Windows Server clusters and turns them into Shared Nothing Clusters. The diagram here shows a GeoCluster-enabled failover cluster with completely independent storage.

GeoCluster replicates the real-time data changes between the nodes and also performs quorum arbitration outside of the Shared Nothing Cluster using one or more witness file shares on non-cluster servers that act as tie-breaking votes. Once the challenge of shared storage has been overcome, the next logical step to provide the ultimate in recoverability is to physically distance the nodes to provide multi-site data center redundancy.

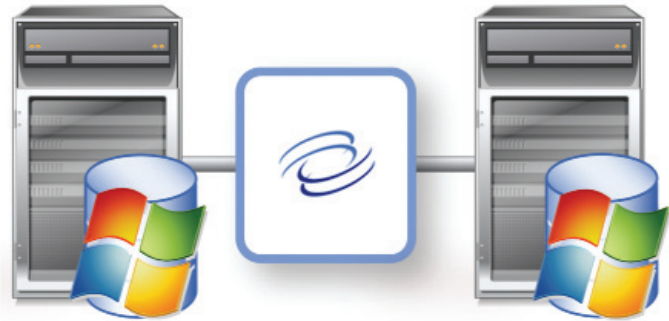


Figure 2: Shared Nothing Cluster

## Implications for Multi-Site Clustering

The goal of multi-site clustering is to extend a cluster across a wide area network and provide site resiliency. Applications can then continue to run if a node or an entire site fails, which provides the ultimate recoverability strategy for your data and applications. This creates a few new challenges that must be overcome to ensure that everything will work as planned. Regardless of the cluster type (Shared or Shared Nothing), there is basic functionality that must be present in any cluster implementation for it to work correctly. First and foremost, one of the nodes has to be in charge of the cluster to maintain its state and govern activities. These can be activities such as allowing addition and removal of nodes to the cluster or moving an application workload to one particular node instead of another node. To manage these activities, clusters maintain a central repository of management information known as a quorum which is used to act as a tie-breaker and determine which node "owns" the cluster responsibility as a whole. This is a critical feature, otherwise the nodes would all think that they were in-charge and wouldn't coordinate properly with each other - a scenario known as "split brain". The process of the cluster deciding which node "owns" the quorum is known as cluster arbitration. In a multi-site cluster, quorum arbitration cannot be done through traditional methods since there is no shared storage sub-system and not even a shared site to perform arbitration.

There must also be a mechanism for ensuring that changes to shared data are written to all of the other sites in a timely enough fashion and that the database can remain in a consistent state. There must be enough network bandwidth between the sites to handle the data transmission rates and to allow your users to reconnect and continue working with acceptable performance.

### Quorum Arbitration

Determining cluster ownership between nodes becomes more challenging in multi-site configurations because wide area networks are inherently far less reliable than local area networks. Therefore multi-site cluster arbitration must take extra precautions to prevent split brain and ensure the integrity of the cluster.

### File Share Arbitration

This method of quorum arbitration is a great solution to provide the tie breaking votes necessary to prevent split brain. Typically the file share is placed at a third site that has connectivity to the production and failover site that can act as a tie-breaking arbiter. If the production node, site or network failed, then one of the remaining cluster nodes in another site could claim the quorum file share and take ownership of the cluster. It would then start the applications and use its copy of the application data to resume services.

If you only have two sites, then you can place the file share in the production site so that the services will not failover automatically if the network connection between the sites fails. If the production site has failed and you really do need to failover, then you can manually override the other node and bring the application back on-line.

### Node Majority Arbitration

If having a third off-site file share isn't an option, then you can still have reliable arbitration using Node Majority arbitration. This quorum arbitration model only counts votes of cluster nodes to determine ownership. Therefore, it requires an odd-number of nodes to facilitate arbitration with the site containing the majority of the nodes acting as the cluster owner.

### Automatic versus Manual Failover

On the surface automatic failover seems like the preferred method since it can happen without an administrator's intervention. However, clusters are operating under strict failover criteria that won't take external factors into consideration so you should plan your multi-site failover scenarios carefully. Manual failover isn't nearly the cumbersome process that some administrators and managers believe it to be and it can save you a lot of pain with larger data sets. For example, your cluster doesn't know that once it fails over a one terabyte database over a slow network connection that it might take days or weeks to resynchronize once the production problem is fixed.

### Storage Replication

Disk storage is a critical component of a failover cluster. Not only is your data stored on the disk sub-system, but quorum arbitration also relies upon it. Having a solid storage strategy is a key decision that you need to consider wisely. Like most technology topics, storage is implemented and managed using different layers that perform specialized tasks that its adjoining layers rely upon, ultimately allowing the application to read and write data to persistent storage like disks, tape or optical storage media.

Applications read and write their data as either records or files. Database applications manage their data as records in tables and as such require that the files that ultimately store their data are in a good consistent format that they can recognize. Most database applications also use data integrity algorithms that ensure the consistency of the data and if they find problems with the data many algorithms can gracefully recover from those problems. This is known as crash consistency and means the difference between a running and corrupted database. Replication technology was created to get changes made to the active production node's storage to other nodes in the cluster. Preserving database transactional write-order integrity and thus crash consistency during the replication process is the key to a successful replication methodology. Storage replication architecture design can make or break a multi-site cluster initiative.

Quorum Arbitration Method	Production Failure Type	Production Site Failover Method
<b>Node and File Share Majority</b> <b>2 Site Stretch Cluster</b>	Node	<b>Automatic:</b> A surviving node will claim ownership of the cluster and transition the failed application to itself or another node.
	WAN	<b>None:</b> The production node will continue owning the cluster since it can still see the file share and have majority ownership.
	Site	<b>Manual:</b> The cluster will go down since the file share isn't available until an administrator overrides the non-majority and brings the applications on-line.
<b>Node and File Share Majority</b> <b>Multi-site Cluster</b>	Node	<b>Automatic:</b> Same as 2 Site Stretch node failure.
	WAN	<b>Automatic:</b> The remaining nodes will still be able to see the file share at another site and gain cluster ownership and failover the applications.
	Site	<b>Automatic:</b> Same as a WAN failure above.
<b>Node Majority</b> <b>2 Site</b>	Node	<b>Automatic:</b> The production site's cluster will continue to function as long as it still contains more than 50% of the remaining cluster nodes. I.e. Cluster Nodes / 2 + 1.
	WAN	<b>Automatic:</b> The production cluster will continue to function since it still contains a majority of the cluster nodes.
	Site	<b>Manual:</b> The cluster will fail until an administrator overrides the non-majority and brings the applications on-line.

Figure 3: Cluster Quorum and Failover Methods



## Replication Integration Considerations

Replication technology can integrate at four main points in a system and there are four primary considerations when evaluating technologies that integrate at each of these layers. The main considerations are transactional awareness, manageability and network impact. The fourth consideration is scalability which transcends all of the integration layers and is the most critical factor.

While scalability is something that is nice to have in most areas of information technology, it is the single most important consideration when evaluating replication technology. Scalability should be measured carefully under production loads to ensure that the replication technology can properly scale to meet your organization's needs. You should also evaluate scalability using real-world scenarios that include sudden breaks in the network connection between the systems. This is because you never get to choose how or when a failure will occur. Replication technologies that don't scale to your processing load will either bring your production application to a halt or create data integrity problems on the replica trying to keep up with the production load.

Transactional awareness is the ability of the replication technology to understand the logical application operations and its' ability to preserve those order of operations and maintain crash consistency. This is another critical capability because most applications have the native ability to recover from a system crash, but only if their data files are in a state that they can recognize as recoverable. It is also important to evaluate replication technologies that will be used for multi-site cluster initiatives over the WAN circuits that they will be using once in production. Replication technology that can't ensure the crash consistent state of databases, especially under load and real-world network latency conditions, can't provide reliable data integrity and are thus unsuitable for recoverability solutions especially with the strict demands of multi-site clusters.

Network impact should also be measured carefully when evaluating replication technology. Many organizations find out the hard way after making expensive purchase decisions that their networks will need massive increases in bandwidth. This has resulted in canceling many off-site recoverability projects because of the high total cost of ownership of some replication technologies. Over time, the network will be one of the most expensive components of your recoverability architecture, so make sure that you spend time examining those requirements and factoring them into your long-term budget expectations.

Finally, manageability is the amount of skill and effort required to maintain your enterprise replication technology and the off-site recoverability architecture as a whole. Now let's explore each integration layer works with regard to these four primary criteria.

### Network Connectivity

The network connectivity between sites will provide the data transmission functionality as well as the communication and heart beat between the cluster nodes. Bandwidth and latency requirements vary based on the replication tools chosen and should be rigorously tested with real production loads.

### Bandwidth Requirements

You will need at least enough bandwidth between sites to handle the average rate of application data change. The amount needed can vary greatly depending on the replication integration point. You should also take into account the amount of bandwidth required by users during operation of the application. This is important after failover when your production users will need to connect over the WAN to access their data and resynchronization now has to compete with that traffic. Some replication tools go as far as requiring dedicated WAN circuits to function, thus eliminating the user contention, but significantly increasing the cost of their solutions.

Some replication tools provide bandwidth management features that will allow you to throttle the amount of replication and resynchronization traffic to manageable levels. You may also want to consider burstable network bandwidth to use during resynchronization prior to failback. You should measure your bandwidth requirements wisely because they will be a significant portion of your total cost of ownership.

### Latency Requirements

Site-to-site network latency are also constraining points of multi-site clusters and some replication tools don't fare well at all with latency over long distances. It is also important to note that IP Ping tests to measure production latency are not very reliable; they were designed to test whether another node is online and use ICMP management protocol, which has the highest priority of all IP packets on a network. Thus, their statistics are best possible performance indicators which won't fare well under real production loads.

### Application Reconnectivity

It's important to understand that when performing failover across subnets some applications may not immediately reconnect immediately and continue processing. Operating systems and some client applications make networking calls to a DNS server and acquire the IP address to perform their initial connection to the server application. If client applications adhere to DNS Time to Live (TTL) parameters associated with DNS records then they will retry connecting to the failed-over application server. Otherwise you may need to make the OS perform a DNS refresh or restart the client application to force the client to get the new IP address and reconnect. It may also be desirable to failover of all layers of client server applications along with their database back-ends. For example, Microsoft Outlook clients will automatically retry to connect regardless of the subnet where the Exchange server is running. So it's important to understand the complete application recoverability requirements and plan accordingly.

### WAN Accelerators

These products are somewhat new to the networking world and can provide help with both bandwidth and latency concerns for WANs with long distances between sites. It's important to note that while many WAN accelerators can compress data, simulate IP ACK packets and perform other network optimization, they are still no substitute for properly measuring your bandwidth and latency needs before implementation.

### Additional Considerations

Replication tools and methodologies are the primary concerns when designing a multi-site cluster solution. However, there are many others that must factor into the decision making process because they too can be critical to the long-term success of the project.

### Resynchronization

All replication tools must perform a baseline of the data set when they are initially connected. If there is a prolonged network outage between sites then the data will have to be resynchronized to continue replication. There are different methods for performing this process and all require time and bandwidth to complete. Some will allow your application to continue running while others may require you to stop the application until the resynchronization process completes. Resynchronization of replicas can take hours or even days for larger data sets depending on network bandwidth, so you should be prepared to handle this scenario.

### Vendor/Hardware Requirements

Many replication tools, especially those that integrate in the storage layer, are proprietary and will only work with the same vendor hardware in all sites. This means vendor lock-in so you won't be able to mix storage hardware or change storage vendors without significant effort.

Some storage replication tools also need additional hardware devices to assist with the network transmission of data. These devices must be factored into the total costs and management of the multi-site cluster implementation. When dealing with multiple vendors you must also take special care when updating firmware and associated hardware components in all sites to minimize application downtime and ensure that the new combinations are fully certified and supported.

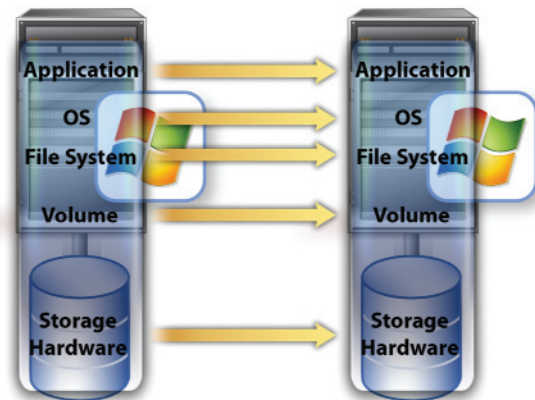


Figure 4: Replication Integration Models

## Multi-Site Clustering using GeoCluster

Double-Take Availability the GeoCluster feature integrate at the OS/file system layer to provide replication services between nodes of the Microsoft Windows cluster. As such, it also integrates within Microsoft Windows Server Failover Clustering as a clustered disk resource. Once it's implemented, the management of the cluster is the same as a standard shared disk scenario so system administrators' pre-existing cluster management skills transfer seamlessly. Since Double-Take Availability replicates the underlying storage of Microsoft Clusters, you can stretch cluster nodes between physical locations and still use the native failure detection and failover functionality that cluster-aware applications use to communicate with Failover Clustering.

The integration point of replication above the file system allows it to have complete transactional awareness as well as the ability to limit the scope of replication to just the files and directories that the application requires for failover. Thus, it can eliminate changes to unnecessary files like TempDB and the windows pagefile as well as changes due to disk defragmentation that other replication methods are forced to replicate over the network.

As a component of Double-Take Availability, GeoCluster provides the industry's recognized performance leading data replication technology. It includes patented data integrity algorithms that ensure write order preservation of logical application writes and thus completely maintains crash consistency at any point of failure. This time-tested replication scalability has been shown to scale linearly with hardware and network configurations especially in multi-terabyte configurations.

GeoCluster uses an asynchronous method of replication to ensure that the production nodes will never suffer performance degradation because of intermittent network failures or node failure. It uses performance-tuned methods to get replication data on the network as quickly as possible after the production storage commits the write, without sacrificing crash consistency. GeoCluster also integrates with Microsoft Volume Shadow Services to provide scheduled point-in-time snapshots of the replica data to preserve recoverability in the event of logical errors caused by administrator/user error and other unwanted changes so that you can easily roll data back to previous points-in-time.

Network impact is minimized by replicating just the application's logical changes to its files while preserving order of operations. This is especially important because this is the smallest level of change possible that the application expects to see on the replica to provide crash consistency. For example, if SQL Server changes a single byte in its database files the Double-Take replication engine will replicate just that byte to the other cluster nodes. When a database transaction is logged or a page changes, then just that transaction is replicated in its entirety regardless of the size and not the entire block that other replication tools are forced to replicate. This gives Double-Take Availability and GeoCluster a huge advantage in network performance up front because they doesn't have to replicate extraneous non-recoverable I/O.

Double-Take Availability can also perform real-time data compression on these changes. The tuned compression algorithm uses minimal CPU overhead while delivering real-world measured performance of 65-75% compression for database applications like SQL Server and Exchange. Compression of changes before their sent over the wire for replication further decreases the amount of bandwidth required to support your replication needs.

Resynchronization of replicas in the event of long-term network outages like network cuts or after failover is performed using the Double-Take differential mirroring algorithm. Large database files are analyzed for block-level differences with only the changes copied across the network. This process saves a tremendous amount of time when performing resynchronization of large databases. It also allows you to seed or pre-stage the target replica before the initial baseline of the database using a recent backup restored on the off-site node thus dramatically reducing the time to baseline the replica.

There are no additional hardware requirements for the product above and beyond standard Windows servers and TCP/IP connectivity between nodes. The GeoCluster feature of Double-Take Availability is completely storage independent and doesn't require additional specialty hardware like other solutions. Therefore, you can mix and match storage of different types and from different vendors. You can, for example, run your production node on high-speed Fiber storage from vendor A while storing your replica data on SAS or iSCSI storage from vendor B. This gives you complete flexibility by virtualizing the storage subsystem and allowing you to change storage by simply swapping it out. Thus you can avoid vendor lock-in and architect your storage based on application need rather than vendor limitations.

### Implementing Double-Take Availability for Windows Server 2008

The first step in creating a multi-site cluster using Double-Take Availability and GeoCluster is to install and configure the Failover Clustering role in Windows Server 2008. Next, Double-Take Availability can be installed on the first node in the multi-site cluster. This will install the Double-Take file system filter drivers and GeoCluster cluster resources. Using the Microsoft Cluster Administrator (or another method), a base cluster group can be created with the appropriate cluster resources within it. Next, install the clustering role on the other remaining member nodes of the cluster (up to 16 nodes in Windows 2008) and configure quorum arbitration. You can use either multi-site cluster quorum arbitration method provided by Windows Server 2008, Node Majority or Node Majority with File Share Witness arbitration. Optionally, you can also configure Double-Take Availability to use a specific network connection to handle all replication traffic and configure bandwidth throttling and compression if needed.

From this point forward you can install the cluster-aware applications just as if you were using a shared storage cluster. Adding the GeoCluster replicated disk resources in lieu of standard shared disk resources allows you to replicate the database and transaction log disks on a clustered SQL Server, for example. Double-Take will replicate any logical data changes to each of the nodes in the cluster that you specify in real-time. When the database application makes changes to its data files, they will be replicated while maintaining order of operations to provide complete crash consistency of the application.

Once the initial synchronization of the shared data is complete, failover between nodes can then occur. Failover Clustering will perform its native application monitoring and initiate failover if any application resources fail or the cluster group is manually moved between nodes. Upon failover, Failover Clustering will start the application and, because Double-Take Availability ensures write order preservation of replicated data, the application will start normally and perform any necessary roll forward processing. Once the application services have been started, end-user applications can then reconnect.

Once the original production node has been fixed, Double-Take Availability can resynchronize the data disks used by the cluster using its differential remirroring feature which will find and resynchronize only the differences between the data files. Failback can be configured to happen as soon as resynchronization completes or manually which will gracefully shutdown the resources on the failover node and restore them on the production node. Double-Take Availability will restore replication to the other nodes of the cluster after the primary node resumes processing.

### GeoCluster Multi-Site Cluster Use Cases

The GeoCluster feature of Double-Take Availability natively works with all of the supported Windows Server 2008 Failover Clustering architectures. Any application that can run on Microsoft Windows in a failover cluster is fully supported including cluster configurations like multi-site and global clustering. Management of Double-Take Availability, the GeoCluster feature and Failover Clustering is performed using the Microsoft Cluster Administrator. Thus the management requirement is minimal because it doesn't differ much from managing a failover cluster without Double-Take Availability installed.

### Shared Nothing Same-Site Clustering

Double-Take Availability and GeoCluster can allow you to replicate cluster-aware storage between nodes in the same site. This allows you to eliminate the shared storage single point of failure, or put one node on SAN storage from vendor A and have the other node on SAS or iSCSI storage from vendor B. If one vendor's



storage fails, perhaps during firmware and driver updates, then the other nodes will be completely insulated and allow the clustered application to continue running. This architecture works with Windows Server 2008 Failover Clustering and has been a proven solution by Double-Take Availability for many years on both the Windows Server 2000 and Windows Server 2003 clustering platforms.

Quorum arbitration is performed using Node Majority (available in Windows Server 2003 and Windows Server 2008) or using Node Majority with File Share Witness (available in Windows Server 2008 only). GeoCluster provides its own native File Share Witness arbitration for Windows 2003 clusters. Networking between nodes uses standard TCP/IP communications protocols over a standard network connection. Since the cluster nodes are all on the same subnet, client applications will reconnect immediately since the IP address is also failed over as part of failover clustering.

### "Stretch" Clustering Architecture

This clustering architecture uses the GeoCluster feature of Double-Take Availability and Windows Server 2008 to stretch the cluster nodes across campuses or across a WAN to different sites. Windows Server 2008 Failover Clustering allows you to have cluster nodes on different subnet which makes failover a seamless process. The applications can reconnect as soon as the failover completes which usually takes just a few minutes while the clusters resources are brought online on the secondary node.

### "Global" Cluster Recoverability Architecture

This architecture is unique to Double-Take Availability. It works well for environments that want the option of replicating their cluster data over very long distances - perhaps even spanning oceans. Cluster data is replicated to another independent cluster for global cluster recoverability. For example, you could have two nodes in San Francisco and Los Angeles with a third node stretched all the way to Hamburg, Germany. The Hamburg cluster can be a single node cluster or a member of a cluster currently used for production in Germany. In this configuration, you don't have to force arbitration traffic to travel over such a great distance and suffer from the extreme latency.

This configuration also provides you with a manual fail-over option in-case you don't want to automatically failover the clustered applications because of very unreliable networks or limited bandwidth. You could even create an encrypted VPN and use the public Internet to act as the backbone so that you don't have to invest in leased lines over such a great distance. The Hamburg replica could be brought on-line as a last resort during a critical regional outage.

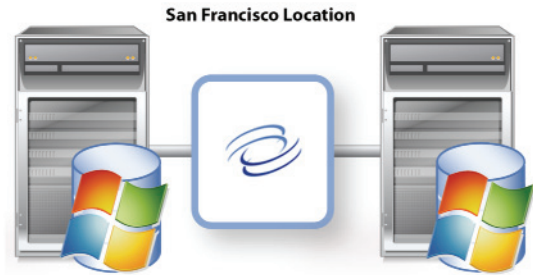


Figure 5: Shared Nothing Same-Site Clustering

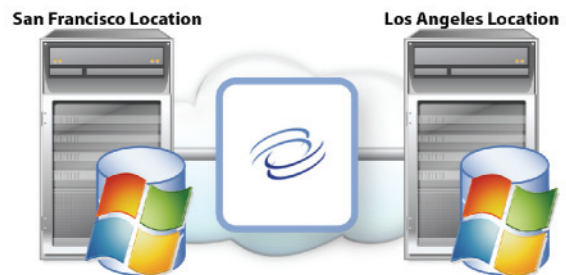


Figure 6: "Stretch" Clustering Architecture

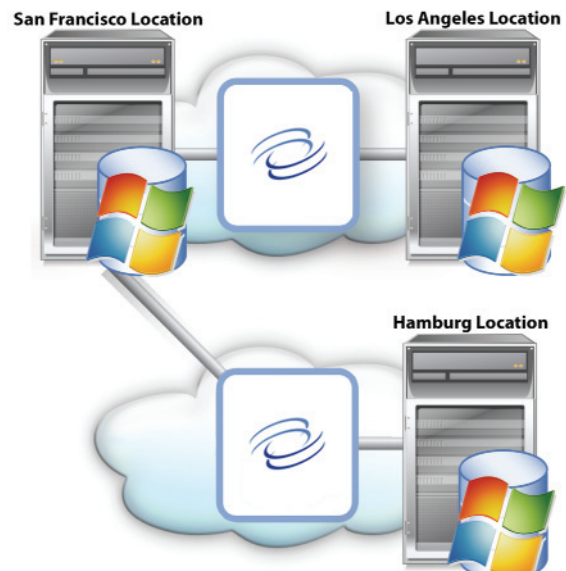


Figure 7: "Global" Cluster Recoverability Architecture

## Summary

Microsoft Windows Failover Clustering has come a long way over the years and Double-Take Availability with the GeoCluster feature is still a cornerstone for protecting and enhancing cluster capabilities. Double-Take Software provides market-leading solutions for data protection and recoverability. Double-Take Availability from Double-Take Software was created specifically to protect extend and build upon the capabilities of Failover Clustering by providing redundant storage that eliminates the single point of failure inherent in traditional clusters as well as providing off-site cluster failover capabilities. Now data can be replicated to different storage devices locally or remotely and allow clustered applications to remain running regardless of the point of failure.

Manage your subscription to eNews. Visit: [www.doubletake.com/subscribe](http://www.doubletake.com/subscribe)

**Double-Take Availability** 

 Printed on recycled paper.

Get the standard today: [www.doubletake.com](http://www.doubletake.com) or 888-674-9495

© Double-Take Software, Inc. All rights reserved. Double-Take, Balance Double-Take Cargo, Double-Take Flex, Double-Take for Hyper-V, Double-Take for Linux, Double-Take Move, Double-Take ShadowCaster, Double-Take for Virtual Systems, GeoCluster, Livewire, netBoot/i, NSI, sanFly, TimeData, TimeSpring, winBoot/i and associated logos are registered trademarks or trademarks of Double-Take Software, Inc. and/or its affiliates and subsidiaries in the United States and/or other countries. Microsoft, Hyper-V, Windows, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. Red Hat is a registered trademark of Red Hat, Inc. Novell, the Novell logo, the N logo, SUSE are registered trademarks of Novell, Inc. in the United States and other countries. All other trademarks are the property of their respective companies.