Hitachi Storage Cluster for Microsoft Windows Server 2003

Geographically Distributed Clusters Using Hitachi TrueCopy™ Remote Replication Software

Technical Brief

By Tom Goulding, Carl Isenburg, and Steve Burr

March 2005
**Executive Summary**

Ensuring recovery of critical assets in the event of a disaster is a challenging proposition. A variety of techniques are available, but many do not go far enough to protect key information assets. Today’s organizations demand robust techniques to protect their critical computing infrastructures.

Remote site recovery and clustering are two techniques that are commonly used to mitigate disaster risks, but each has drawbacks that limit their effectiveness and appeal. However, distance clustering can combine the advantages of these two techniques while minimizing their drawbacks. Microsoft embraced distance clustering with its release of Microsoft Windows Server 2003 by introducing an important technique for management of geographically dispersed clusters: the majority node set.

Hitachi Data Systems provides support for Windows Server 2003 distance clusters and the majority node set technique using Hitachi TrueCopy™ Remote Replication software. Combining Windows Server 2003 majority node set clusters with TrueCopy Remote Replication software can provide for a robust, geographically distributed cluster. For many organizations, this configuration provides a valuable technique for mitigating disaster risk.
Contents

The Need for Real-time Recovery from a Site-wide Disaster ............................................................... 1
  Why Cluster Over Distance? .................................................................................................................. 1
  Why Are Additional Techniques Necessary for Distance Clustering? ............................................. 1

What is Hitachi Storage Cluster for Microsoft Windows? ................................................................. 2
  Features .................................................................................................................................................. 2
  Supported Platforms ............................................................................................................................ 3
  How Does It Work? ............................................................................................................................... 3
  What About “Split Brain”? .................................................................................................................. 3
  Microsoft Cluster Server Approaches to Avoiding Split Brain ........................................................ 4

Using Majority Node Sets Cluster Ownership ............................................................................... 5
  Comparing Majority Node Sets with the Shared Quorum Disk Model ................................................. 5
  How Does Hitachi Storage Cluster Work with Microsoft Quorum Models? ...................................... 6

Cluster Topologies .......................................................................................................................... 6
  Topology Nomenclature ...................................................................................................................... 7
  Topology 1: 1+1 (Windows 2000, Two Site) .......................................................................................... 7
  Topology 2: N+1 (Windows Server 2003, Two Site) ............................................................................. 8
  Topology 3: 1+1+1 (Windows Server 2003, Three Site) ...................................................................... 8
  Topology 4: N+N+1 (Windows Server 2003, Three Site) .................................................................... 9

Implementation Considerations ........................................................................................................ 10

Conclusion ........................................................................................................................................... 11

Appendix A: About the Authors ....................................................................................................... 12

Appendix B: Design Comparison ...................................................................................................... 13
  Topology #1: Cluster with Six Hosts and Two Sites ........................................................................... 13
  Topology #2: Cluster with Seven Hosts and Three Sites ................................................................. 14
Hitachi Storage Cluster for Microsoft Windows Server 2003
Geographically Distributed Clusters Using Hitachi TrueCopy™ Remote Replication Software

Technical Brief

By Tom Goulding, Carl Isenburg, and Steve Burr

The Need for Real-time Recovery from a Site-wide Disaster

Ensuring recovery of critical assets in the event of a disaster is a challenging proposition. A variety of techniques are available, but many do not go far enough to protect key information assets. Today’s organizations demand robust techniques to protect their critical computing infrastructures.

Historically, recovery techniques have relied on tape-based recovery, but these can be lengthy, error-prone processes. More recently, clustering for failover has provided a technique to automate this process and provide faster, more reliable recovery in the event of disaster. But for many, local-site clustering does not provide sufficient protection from wide-area events such as hurricanes or earthquakes.

Why Cluster Over Distance?

Organizations are increasingly aware of the risk associated with single-site recovery. Natural disasters, sabotage, power failures, and other disasters can easily affect all systems within a single location. Extending a cluster over distance provides an opportunity to remove recovery resources from the affected geography to better shield them for the effects of the disaster.

Why Are Additional Techniques Necessary for Distance Clustering?

Traditional failover clusters are not able to easily provide wide-area failover for two reasons:

1. It is relatively difficult to provide access to storage resources between sites separated over distance. A storage area network (SAN) is often used to facilitate sharing of storage between cluster nodes, but extending a SAN can be expensive, complex, and difficult.

2. Even if the SAN is extended, these shared storage resources are wholly located at the primary site and are susceptible to the impact of the site-wide disaster. For example, if the primary facility is flooded, failover resources at the secondary site will be unable to connect to the water-logged devices located in the primary site.
What is Hitachi Storage Cluster for Microsoft Windows?

Hitachi Storage Cluster for Microsoft Windows is a disaster recovery solution from Hitachi Data Systems (see Figure 1). It “cluster-enables” disk resources that have been replicated to a distant site using Hitachi TrueCopy™ Remote Replication software. Since a recovery-suitable disk image is available at a remote location, the cluster can fail over to the recovery facility and pick up processing using that I/O-consistent image of production disks.

Figure 1: A Hitachi Storage Cluster Example

In a Hitachi Storage Cluster, production disk resources are replicated to the recovery site and recovery nodes are ensured appropriate access to the cluster’s disk resources. Operations can be resumed with minimal outage.

In other words, Hitachi Storage Cluster “stretches” a cluster across distance. Production disk resources are replicated to the recovery site, and Hitachi Storage Cluster ensures that recovery nodes have appropriate access to the cluster’s disk resources to resume operation with minimal outage.

The net effect is that production processing is removed from the production data center following a disaster. This enables real-time recovery of critical applications across widely separated geographic locations.

Features

- Extends Microsoft Cluster Service with a TrueCopy resource to monitor and manage the replicated data (groups of disks)
- Coordinates replication activity (replication takeover) with cluster operations (online, move, etc.)
- No additional cluster commands to learn or manage
:: Straightforward configuration
:: Uses TrueCopy software for robust off-host disk replication to the remote facility

Supported Platforms
:: Any Hitachi storage system that supports TrueCopy Remote Replication software
:: Microsoft Windows Server 2003 (three or more nodes, using Majority Node Set quorum model)
:: Microsoft Windows 2000 (2 nodes only) pending availability of Hitachi TrueCopy Agent for Windows 2000

How Does It Work?
1. TrueCopy software is used to maintain exact duplicates of production disks at the recovery location.
2. Hitachi Storage Cluster resources are deployed on each node.
3. Dependencies are set on cluster disk resources so that disks/nodes will not come online before Hitachi Storage Cluster software can prepare the replication environment.
4. Prior to a node is coming online, Hitachi Storage Cluster software ensures that the local storage system will support I/O from the cluster and that replication traffic is moving in the proper direction.

What About “Split Brain”?

“Split Brain” is a potential problem for all clustered recovery solutions. Consider what happens if cluster nodes cannot communicate with one another. Could these independent nodes each begin servicing transactions on their own while other nodes do the same? Could each think that they are the cluster and begin services requests independently of one another? This condition is known as the “split-brain” (see Figure 2).
The condition known as split brain occurs when independent nodes begin servicing transactions on their own—indipendently of one another.

With split-brain, transaction loss and data corruption are possible—maybe even probable. But with multiple sites, split-brain is even more relevant since single site assumptions fail and data integrity will likely depend on those assumptions. As a rule, if cluster ownership is in doubt, halting production resources is the safest course of action.

Integrity is better than availability.

Microsoft Cluster Server Approaches to Avoiding Split Brain

With the release of Windows Server 2003, Microsoft has introduced an important technique for management of geographically dispersed clusters: the majority node set, as shown in Figure 3.
Microsoft introduced the majority node set cluster to make sure cluster configuration data is kept consistent across different disks.

**Using Majority Node Sets Cluster Ownership**

Microsoft documentation describes a majority node set cluster like this: “A cluster configuration that has two or more nodes and that is configured so that the nodes may or may not be attached to one or more cluster storage devices. The cluster configuration data is stored on multiple disks across the cluster, and the cluster service makes sure that this data is kept consistent across the different disks. Each node maintains its own copy of the cluster configuration data. The quorum resource ensures that the cluster configuration data is kept consistent across the nodes.”

This means that an election process is used to arbitrate ownership of the cluster, and that under normal operation ownership will only change once a strict majority of cluster members agree. In addition, since each node maintains its own copy of the cluster configuration data (via communication with other cluster members over the IP network), a separate physical resource is not involved, which means cluster resource and management requirements may be substantially reduced.

**Comparing Majority Node Sets with the Shared Quorum Disk Model**

Prior to Windows Server 2003, quorum disks were used to arbitrate ownership of the cluster. Under this model, a physical storage device (the quorum disk) was used to storage cluster configuration data, and ownership of this device indicated ownership of the cluster.

This model is reasonably straightforward to implement, but it begins to suffer as cluster nodes are moved apart from one another. Over relatively short distances, the quorum disk can be accessible by all cluster nodes via the SAN. Beyond those distances, however, additional techniques must be applied to ensure that the quorum device is appropriately distributed to all locations.
At the disk level, ownership of the quorum disk is arbitrated every three seconds using low-level SCSI protocol commands. If the current owner fails to renew ownership, a challenge/defense protocol is initiated to determine the new owner of the cluster.

Over distance, it is this reliance on low-level SCSI commands that cause issues. Not only must all nodes have access to the quorum disk (or a viable duplicate of the quorum disk) but the technique that enables wide area disk access must also support transmission or emulation of those SCSI commands.

How Does Hitachi Storage Cluster Work with Microsoft Quorum Models?

For Windows Server 2003, Hitachi Storage Cluster uses Majority Node Set, built into Microsoft Cluster Server. No additional third-party software components are necessary.

For Windows 2000, Hitachi Data Systems will soon provide a software agent to facilitate the Hitachi Storage Cluster. This agent uses the shared quorum model and includes a quorum filter driver to emulate SCSI reserve in the arbitration process, as shown in Figure 4.

Figure 4: Shared Quorum

The shared quorum model is reasonably easy to implement and works well over short distances, but it begins to suffer as cluster nodes are moved apart from one another.

Cluster Topologies

A cluster can be configured in a variety of topologies, with a varying number of nodes at either the production or recovery facilities. Each of these topology options carries distinct costs and benefits. It is important to choose a topology based on recovery and cost requirements for each environment. The “best” topology is the one that best fits your needs within your budget.
Topology Nomenclature

Cluster topologies are often designated using a shorthand expression in the form X+Y or X+Y+Z, such as 1+1, or 3+2, or N+N+1. The first number (or variable) represents the number of cluster nodes at the production facility. The second value represents the number of cluster nodes running at the recovery site. Subsequent terms represent nodes running at additional sites.

Using this convention, “3+2” would depict a cluster topology with three nodes at the production site and two nodes for recovery.

Topology 1: 1+1 (Windows 2000, Two Site)

This topology looks a lot like a traditional “local site” cluster that has been extended to the recovery site. For this configuration, Hitachi Storage Cluster would require a TrueCopy Agent for Microsoft Cluster Service (pending availability) to manage and arbitrate ownership of the cluster disk resources, as shown in Figure 5. This agent is composed of a quorum filter driver and resource DLL.

Figure 5: A 1+1 Cluster

A 1+1 cluster provides for automatic failover if Node A fails, but requires manual intervention in other failure scenarios.

This configuration provides for automatic failover if Node A fails, but other failure scenarios will require manual intervention, including primary disk failure, primary site failure, and WAN failure.
Topology 2: N+1 (Windows Server 2003, Two Site)

This topology consists of multiple nodes located at the primary site with a single recovery node at the remote facility. It provides local failover at the production location for “minor” disasters rather than triggered a site-wide failover. But, in the event of a primary site failure, manual intervention is required to bring up the cluster at the recovery site.

**Figure 6: An N+1 Cluster**

Multiple nodes at the primary site and a single node at the recovery site ensure local failover at the production location for “minor” disasters.

Topology 3: 1+1+1 (Windows Server 2003, Three Site)

With this topology, a “witness node” (tie-breaker) is placed at third site to arbitrate quorum. This configuration can provide automatic (“lights out”) failover from the production to recovery sites. In addition, the three-site architecture provides additional resiliency, since a two-site failure (out of three sites) is necessary to bring down the cluster. This increases odds in your favor, since a two-site disaster is much less likely than a single-site event.
Figure 7: An 1+1+1 Cluster

This three-site configuration can provide automatic failover from the production to recovery sites, with additional resiliency.

Note that the witness node is only in place to vote for cluster ownership and therefore has very low hardware and software requirements. Beyond the operating system (and cluster software) this node only needs network connectivity to participate in quorum voting. It does not require any connectivity to shared storage resource. Ideally this third node is placed near the “customer” of the service.

Topology 4: N+N+1 (Windows Server 2003, Three Site)

This topology provides for the most comprehensive protection and the least potential for manual intervention. It uses multiple nodes at both the production and recovery site so that “local disaster” can be handled before failing to the secondary facility. In addition, it uses the three-site architecture mentioned earlier to further distribute recovery decision making. At first glance, it appears that this architecture would be the most expensive. After all, it requires the most nodes of any of the architecture options presented here. But as environments scale, this option may actually be the most cost-effective way to provide robust recovery characteristics. For example, if multiple applications are deployed within a single cluster, each can share a single witness node, and you effectively distribute the witness node investment among a greater pool of services.
Using multiple nodes at both the production and recovery site, this topology provides for the most comprehensive protection and the least potential for manual intervention. A local disaster can be handled before failing to the secondary facility.

**Implementation Considerations**

When architecting a solution, it is important to consider your risks and exposures, and apply those techniques that are most appropriate. In most cases, you will find that a layered approach to protection is best since any single recovery technique will probably not serve to protect against every conceivable outage.

1. Recoverability of data is highly dependent on the algorithms employed to transfer data to the recovery site in real time. If disk writes from the production hosts are not applied in the same order at the recovery site as at the primary (i.e., the replication must maintain write-order fidelity) the recovery disks will likely contain subtle (or not so subtle) errors that can either prevent recovery or render the recovery data worthless or worse. TrueCopy Remote Replication software, with its asynchronous storage-based replication capabilities, makes use of the large cache provided by Hitachi storage products to provide in-order delivery of every production I/O to the volumes at the recovery location. This ensures I/O consistency and allows for a nearly contemporaneous recovery point in a failover scenario.
2. Although asynchronous capabilities of TrueCopy software can provide replication of clustered storage resources over unlimited distance, Microsoft clusters are effectively distance limited due to heartbeat traffic. To identify dead nodes, Microsoft Cluster Services uses heartbeats, which are periodic messages to indicate a node is still operational. If a cluster does not respond to heartbeat traffic within a specified period of time, that node is considered dead. In terms of stretched clusters, heartbeat traffic takes some time to traverse the network and at some point network latency will delay heartbeat messages to the point that the cluster is no longer functional. Microsoft Cluster Service has a 500-millisecond limit on heartbeat responses, which will translate into a physical limit dependent on distance and network configuration.

3. Clusters do not protect against data corruption or human error. If bad data is written to disk (e.g., human error, virus, etc.) it will be replicated to the remote location. For these circumstances, it is important to have point-in-time images of data that can be used to restore data to a time prior to the corruption. Tape backup is one example of this type of recovery. To decrease recovery time, you can consider disk-based recovery options. Hitachi SplitSecond™ Solutions, a family of recovery solutions offered by Hitachi Data Systems, provide rapid application recovery without imposing application outages to create point-in-time disk replicas.

**Conclusion**

Combining Windows Server 2003 majority node set clusters with TrueCopy Remote Replication software can provide for a robust, geographically distributed cluster. For many organizations, this configuration provides a valuable technique for mitigating disaster risk.
Appendix A: About the Authors

Tom Goulding and Carl Isenburg are both Solution Architects for Continuity Services with Hitachi Data Systems Global Solution Services. They focus on storage solutions involving data replication.

Steve Burr is Principal Consultant for Hitachi Data Systems UK, with a special focus on business continuity.

Other contributors to this paper included Advanced Technical Consultants Robert Burch and Joe Pena, both supporting the Americas region, and Wendy Roberts, supporting the Asia Pacific region.
# Appendix B: Design Comparison

In general, there is a correlation between the number of nodes in a SAN and the level of availability that the cluster provides. However, this correlation is not exact and can vary depending on specific recovery objectives.

## Topology #1: Cluster with Six Hosts and Two Sites

(Production Facility Nodes: A1, A2, P1, P2; Recovery Site Nodes: P3, P4)

<table>
<thead>
<tr>
<th>Failure Scenario</th>
<th>Hosts at Production Facility</th>
<th>Hosts at Recovery Site</th>
<th>SAN at Production Facility</th>
<th>SAN at Recovery Site</th>
<th>Quorum</th>
<th>Replication Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server A1 goes down; SAN okay</td>
<td>A1 not working A2 active P1 takes over A1 P2 passive</td>
<td>P3 passive P4 passive</td>
<td>No impact; continue to replicate between sites</td>
<td>No impact; continue to replicate between sites</td>
<td>Cluster still has majority</td>
<td>Production to recovery</td>
</tr>
<tr>
<td>Both A1 and A2 go down; SAN okay</td>
<td>A1 not working A2 not working P1 takes over A1 P2 takes over A2</td>
<td>P3 passive P4 passive</td>
<td>No impact; continue to replicate between sites</td>
<td>No impact; continue to replicate between sites</td>
<td>Cluster still has majority</td>
<td>Production to recovery</td>
</tr>
<tr>
<td>A1 down, A2 down, and P1 down; left with single P2</td>
<td>Propose processes to declare site down Force quorum required to recover</td>
<td>P3 takes over P4 takes over After force quorum and bring up applications on P3 and P4</td>
<td>No impact; continue to replicate between sites</td>
<td>No impact; continue to replicate between sites</td>
<td>Cluster does not have majority, but is running due force quorum</td>
<td>Recovery to production</td>
</tr>
<tr>
<td>Both A1 A2 okay, but SAN goes down at main site</td>
<td>A1 no disk access A2 no disk access P1 no disk access P2 no disk access</td>
<td>P3 takes over P4 takes over</td>
<td>Not available; servers available, therefore cluster still has majority</td>
<td>SAN at main site down; cannot replicate</td>
<td>Cluster still has majority</td>
<td>No replication</td>
</tr>
<tr>
<td>Entire main site goes down</td>
<td>A1 not working A2 not working P1 not working P2 not working</td>
<td>P3 takes over after force quorum P4 takes over after force quorum</td>
<td>Not available</td>
<td>Main site down; cannot replicate</td>
<td>Cluster does not have majority, but is running due force quorum</td>
<td>No replication</td>
</tr>
<tr>
<td>Link failure between sites</td>
<td>A1 active A2 active P1 passive P2 passive</td>
<td>P3 passive P4 passive</td>
<td>SAN connectivity between sites down; cannot replicate</td>
<td>SAN connectivity between sites down; cannot replicate</td>
<td>Cluster still has majority</td>
<td>No replication</td>
</tr>
</tbody>
</table>
### Topology #2: Cluster with Seven Hosts and Three Sites

(Production Facility Nodes: A1, A2, A3; Recovery Site Nodes: P1, P2, P3; Tie-breaker Site with Witness Node)

<table>
<thead>
<tr>
<th>Failure Scenario</th>
<th>Status</th>
<th>Replication Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server A1 goes down; SAN okay</td>
<td>A1 not working P1 takes over A1</td>
<td>Recovery to production (A1)</td>
</tr>
<tr>
<td></td>
<td>A2 active P2 passive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A3 active P3 passive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No impact; continue to replicate between sites</td>
<td>Cluster still has majority</td>
</tr>
<tr>
<td></td>
<td>No impact; continue to replicate between sites</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cluster still has majority</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recovery to production (A2/A3)</td>
<td></td>
</tr>
<tr>
<td>Production facility hosts down; SAN okay</td>
<td>A1 not working P1 takes over A1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A2 not working P2 takes over A2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A3 not working P3 takes over A3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No impact; continue to replicate between sites</td>
<td>Cluster still has majority</td>
</tr>
<tr>
<td></td>
<td>No impact; continue to replicate between sites</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cluster still has majority</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recovery to production</td>
<td></td>
</tr>
<tr>
<td>All servers okay; SAN goes down at production facility</td>
<td>A1 no disk access P1 takes over A1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A2 no disk access P2 takes over A2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A3 no disk access P3 takes over A3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAN down; no replication</td>
<td>Cluster still has majority</td>
</tr>
<tr>
<td></td>
<td>SAN at main site down; cannot replicate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No replication</td>
<td></td>
</tr>
<tr>
<td>Entire production facility goes down</td>
<td>A1 not working P1 takes over A1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A2 not working P2 takes over A2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A3 not working P3 takes over A3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not available</td>
<td>Cluster still has majority</td>
</tr>
<tr>
<td></td>
<td>SAN at main site down; cannot replicate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cluster still has majority</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No replication</td>
<td></td>
</tr>
<tr>
<td>Link failure between sites</td>
<td>A1 active P1 passive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A2 active P2 passive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A3 active P3 passive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAN available; no replication</td>
<td>Cluster still has majority</td>
</tr>
<tr>
<td></td>
<td>SAN available; no replication (failover not available)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No replication</td>
<td></td>
</tr>
</tbody>
</table>

Note: After replication fails, further failover cannot and will not occur.