



high availability
systems
engineering

september 2002



technical
white paper

data protection solutions for ProLiant Clusters utilizing NSI Double-Take

table of contents

abstract	2
assumptions and intended audience	2
introduction	2
NSI Double-Take overview	3
key concepts	3
source and target	3
mirroring and replication	3
choosing a replication target	4
using remote targets	5
remote target considerations	5
remote target backups	5
remote target implementation	5
using local targets	6
local target considerations	6
local target backups	6
local target implementation	7
summary	7
appendix a	8
appendix b	9
for more information	10

abstract

Microsoft Cluster Service (MSCS) leverages the shared storage infrastructure to enable failover of applications and their data. With shared storage, a single copy of the data fails over between the cluster nodes, which ensure that cluster nodes have the latest copy of the data when they host the application. This implementation presents a single point of failure if the data on the shared storage becomes inaccessible or damaged. The ability to have an exact copy of the data mitigates this failure point. NSI Double-Take working within MSCS enables the creation and maintenance of these copies of the clustered data. This paper will discuss implementation scenarios for NSI Double-Take within ProLiant Clusters.

assumptions and intended audience

The reader of this paper should have some familiarity with ProLiant servers, MSCS and general networking concepts.

introduction

ProLiant servers have many built-in high availability features. These include hot-plug redundant cooling fans and power supplies as well as PCI Hot Plug buses and hot-plug drives. Error Checking and Correction (ECC) memory, a standard feature of ProLiant servers, prevents single-bit, "soft" memory errors from propagating into double-bit, "hard" memory failures that would cause a complete server shutdown. While these hardware based features protect against common hardware failures, they cannot protect against Operating System or Application failure or against catastrophic hardware failures.

Microsoft Cluster Service (MSCS) running on ProLiant servers connected HP storage units provides a highly available platform that allows continuation of application services in the event of catastrophic server, operating system or application failure. MSCS implements this by connecting two or more servers to one or more shared storage units. Each cluster server and the shared storage units contain all the application information necessary to allow either or any server in the cluster to run the application.

HP storage devices contain many high availability features, such as hot-plug drives, hardware RAID and redundant array controllers. However, there are events that can adversely impact the usability of the data on the shared storage. The failure of multiple physical disk drives in a RAID set (more than two failed disks in a RAID 1+0 or RAID 5, more than three failed disks in an ADG RAID set) will prevent applications from accessing the data on the remaining drives. Furthermore, corrupted or missing files can prevent applications from starting or functioning properly, even when protected by MSCS. In the conditions outlined above, the shared storage unit is a single point of failure in the MSCS configuration.

Recovering from a shared storage failure, whether from drive failure (physical data failure) or missing or corrupted files (logical data failure) can be a complex process. HP has a series of white papers that describe how to backup and restore clustered data, but it is highly likely that the data restored from a tape backup will be older than what was on the disks at the time of failure. The data created or modified since the tape backup will be lost. In these circumstances, an online backup copy of the shared storage data can be invaluable.

For example, if a data file becomes corrupt, the online backup copy can be immediately copied over the corrupted file. Or, if a data file is accidentally deleted, it can in some cases be restored from the online backup.

NSI Double-Take overview

NSI Double-Take is one tool that can be used to create and maintain a continuously updated and immediately accessible copy of the cluster data. Besides providing an additional level of high availability in the event of physical or logical data failure, this copy of the data can be used as the source for tape backups, eliminating the time of day limitation on when backup can take place (“backup window”). For some customers, the online copy of the data may even serve as a replacement for daily tape backups.

key concepts

The rest of this paper will use some key terms to refer to the implementation and functionality of the NSI Double-Take solution. The following section explains how these terms are defined for the purpose of this paper.

source and target

When we describe creating and maintaining a copy of the data, we are discussing the source or original data and the target or copy of the original data. When replication is implemented via NSI Double-Take, the source data can be individual files and/or subdirectories on a disk or an entire disk. For most NSI Double-Take implementations there is considerable flexibility in the choice of the target location. For example, a source disk may be replicated to an equivalent disk on the target or to a subdirectory on a target disk.

The target can be located on the same server as the source or it can be located on a separate server or Network Attached Storage (NAS) device. When the target is a separate server, NSI Double-Take must be installed on both the source and target machines. HP NAS devices can use HP DataCopy, which is fully interoperable with NSI Double-Take installed on the source server. If the target is on the same machine as the source, only one copy of NSI Double-Take is required.

Throughout the rest of this paper, a target located on a separate device from the source will be referred to as a remote target. A target located on the same device as the source will be referred to as a local target.

mirroring and replication

NSI Double-Take uses two main mechanisms to create and maintain copies of data, mirroring and replication. Mirroring is the bit by bit copying of the data from the source location to the target location. Replication is the transmission of data modifications on the source data to the target location. When a source and target have been defined to Double-Take, the first operation that occurs is a mirroring of the data from the source to the target. After the initial mirroring operation is complete, Double-Take automatically uses replication to maintain the target data. In case the rate and amount of data changes on the source exceeds the throughput available for replication, the unreplicated changes are stored in a queue file. When throughput increases or the rate of change at the source decreases, the queue file is read and changes are replicated to the target.

NSI Double-Take also possesses a third mechanism, partial mirroring, that examines the data blocks on the source and target and copies changed data only from source to target. This mechanism is invoked when the rate of change at the source has exceeded the bandwidth available to such an extent that the queue file has been completely filled. When the queue file is full, no further changes can be stored in it. In order to prevent any data inconsistencies between the source and target copies of the data, NSI Double-Take will automatically undertake a partial mirror of the source data.

choosing a replication target

The choice of a replication target has an impact on the functionality of the resulting solution. The differences and implications of the choice will be discussed in detail in this paper.

In all cases, the replication target must have an equivalent amount of available space to the replication source. In some cases, the target may require more available space than exists on the source. A chief reason for this concerns “orphan” behavior, which is a replication rule that governs the handling of the deletion of files on the replication source. Controlling orphan behavior is more significant in file share environments, where accidental file deletions occur more frequently, than in messaging or database environments. When Double-Take is configured to allow orphans, deletion of a file on the source will not delete that file from the target. This allows restoration of deleted files from the online backup at the target rather than having to restore the file from tape.

Allowing orphans on the target means that the amount of space used on the target will exceed the amount of space used on the source, as the space occupied by these files on the target is not in use on the source after the files have been deleted.

A list of HP products by storage capacity is in appendix a. This list can be used to compare products for use as remote replication targets.

IMPORTANT: Deleting files locally on a file server will normally send the files to the Recycle Bin. Double-Take does not see this event as a deletion, but as a move of the files out of the replication set. This means the file will be deleted on the target regardless of the orphan setting. Files deleted by network clients do not go into the Recycle Bin, so they will be kept on the target.

A benefit of having a replicated copy of production data is that files on the target are closed (i.e., not opened by an application) which means the target files can be used for tape backup without having to shut down the application or using backup application specific add-ins to allow the backup of open files. This means that the traditional “backup window”, the period of time that applications are unavailable while the data is copied to tape, is no longer a limiting factor.

For some customers, the amount of data to be backed up exceeds the abilities of their tape backup solutions, in speed of backup solution, storage capacity of backup solution or both. For these customers, backing up from disk to disk becomes the only viable solution. In this scenario, continuous replication (as offered by NSI Double-Take) is more attractive than scheduling backup jobs, because the network impact of continuous replication is distributed across the production day instead of being concentrated in a specific backup time frame.

using remote targets

Replicating to a remote target offers the benefit that the copy of the data is physically separate from the source cluster and nodes, which allows for data recovery in case of catastrophic storage or cluster system failure. When replicating data from a cluster, remote replication allows the replication from the cluster to be itself clustered, providing continued replication in the event that disks being replicated are moved from one cluster node to another.

remote target considerations

When replicating data to a remote target, besides the question of adequate space on the target, there is also the issue of network throughput. If the throughput on the network is insufficient for the replication traffic, the solution may not be acceptable. Throughput is not the same as bandwidth. A network connection may have a bandwidth of 100 MB/sec, but the actual throughput of the connection from source to target will be less than that. For a more complete discussion of throughput vs. bandwidth, please refer to appendix b later in this document. Double-Take has mechanisms to recover from temporary throughput shortages, such as queuing and partial re-mirroring. However, ongoing disparities between throughput and the amount of data changes to be replicated will lead to outdated data on the target.

remote target backups

Replicating data to a remote target not only provides a recovery mechanism for handling shared storage failures; it can also simplify backing up clustered data. Backing up clustered data without data replication can be complicated. For example, if backing the data up over a network, the backup server must connect to the cluster virtual servers to ensure successful operations in the event of a group failover. If backing up to a directly connected backup unit, successful operation can only be assured when the backup software is cluster-able and the backup device is connected to all cluster nodes. Normally, the cluster applications need to be quiesced before beginning a backup, so cluster specific commands for taking the application offline need to be executed in these cases.

Backing the data up from the replicated copy on a remote target is a much simpler procedure. Since the replica files are closed, there is no need to quiesce the application for long periods of time to back them up. Since the backup software is not dealing with a cluster, there is no need for connecting to virtual servers or cluster-aware backup software.

remote target implementation

To establish replication from a cluster there are two configuration steps that must be performed. The first is to define the replication set, i.e., to describe what data should be replicated. A replication set definition is very flexible. It can be either an entire disk or any combination of files and subdirectories on that disk. The replication set is not "cluster-aware" and must be defined and named identically on each cluster node.

The second step is to establish a connection to a target server for the replication. This second configuration step is made cluster-aware by the MSCS resource DLL added by NSI Double-Take to the cluster. This DLL is installed automatically when NSI Double-Take is installed on a cluster node. This resource type ensures that a connection is reestablished when the replication set fails over.

There are several limitations of this resource type:

- The resource type does not allow the specification of a drive and path on the target server. It instead expects the exact same drive letter and path to exist on the target as on the source. This means if "R:\test" is being replicated from the cluster, then "R:\test" must exist on the target. This only applies to clustered connections; non-clustered connections may have their target on any disk in any subdirectory. This

may cause customers to have to reconfigure their target machines for clustered replication. It also effectively prevents intra-cluster mirroring because a cluster will only have a single "R:\test" directory.

- Several connection configuration options are not available in the MSCS connection resource. One in particular has to do with handling file deletions on the source disk – replicate or not to the target. These settings have to be established through the NSI Double-Take Administrator, after the clustered connection has been established. These settings are not reliably persistent from session to session, so they need to be manually reestablished (outside of Cluster Administrator) after every failover.
- Another connection configuration option that is unavailable in the resource type is the ability to specify the network that should be used for a particular replication set. This hampers the ability to use a dedicated replication network to reduce public network traffic in a clustered implementation. It is possible to set a default network for all replication for a server, but this does not provide enough flexibility for complex replication topologies.

using local targets

Replicating to a local target offers the benefit that the copy of the data is physically separate from the cluster-shared storage, which allows for data recovery in case of data corruption. Replicating data from the cluster shared storage to a node's local storage does not allow the replication from the cluster to be itself clustered.

Local target replication is best suited to branch office deployments, where network bandwidth constraints impact the ability to replicate (and backup) large amounts of data to the central office.

local target considerations

Replicating data from clustered shared storage to server node local drives has some restrictions as compared to remote replication. Most importantly, the connection that ensures replication will not be clustered. This means that a failover of the clustered disk being replicated will cause replication of the data to stop. Data modifications that occur while the disk is accessed on the other node will not be replicated to the local disks on the original node. This restriction occurs because of the constraint imposed by the NSI Double-Take connection resource DLL that defines the connection to the cluster, which requires that the drive letters for source and target are the same. Since it is not possible for a computer to host multiple disks with the same drive letter, the clustered implementation is not possible.

With NSI Double-Take 4.1, when a replication source disk moves to another cluster node, the NSI Double-Take management console will not report an error, even though replication has ceased because the source drive is no longer controlled by the connection. Failback of the source disk to the original node will not result in an automatic resynchronization of the data to reflect modifications that occurred while the disk was hosted on the other node. This resynchronization will have to be manually initiated by performing a "verify" operation on the connection in the NSI Double-Take management console.

In general, there will be more storage on the clustered shared drives than will be available on the local nodes. In these scenarios, only the most frequently changed files should be replicated with the more static files protected through normal tape backups.

local target backups

A typical local replication backup implementation would have locally-attached backup devices on the same node with the replicated data. For example, a ProLiant Packaged Cluster could have both the replicated disks and internal AIT Tape drives in the available drive bays of a ProLiant DL380 G2 server node.

local target implementation

Establishing local replication in this configuration is similar to creating a remote replication implementation in that both a replication set and a connection need to be established. In this case, the replication set will not be created on the second node and the connection will not be clustered.

In a properly configured cluster, failover should be a rare occurrence, occurring only in case of node failure or when maintenance is performed on a cluster node or application. A local replication implementation should therefore be fairly reliable. Since Double-Take is not clustered, it should be noted that a reboot of the node that is the local replication target may result in the following results:

- When the node reboots, all groups fail over to another node
- When the rebooted node becomes active, the replication connection may start without the source drives being present on that node. This will not show up as an error condition, but replication cannot occur.
- When the source disk is returned to the rebooted node, resynchronization of the data will not automatically occur, but must be started by a verify operation.

summary

Data replication adds another level of high availability to ProLiant Clusters. By understanding the features and limitations of the available implementations, a satisfactory customer experience can be achieved.

appendix a

table 1. HP remote replication targets (Server configurations use 72 GB SCSI disks)

Product	Maximum Storage Capacity
ProLiant DL360	72 GB
ProLiant DL580	291 GB
ProLiant DL760	291 GB
ProLiant DL380	436 GB
ProLiant ML570	436 GB
ProLiant ML350	582 GB
ProLiant ML370	582 GB
StorageWorks NAS S1000	640 GB
ProLiant ML530	1,008 GB
ProLiant ML750	1,529 GB
StorageWorks NAS B2000	9 TB
SureStore NAS va	15 TB
StorageWorks NAS B3000	27 TB
StorageWorks Executor E7000	64 TB
SureStore NAS xp	73 TB

The storage capability of a ProLiant Server can be easily expanded by one to four terabytes by adding ProLiant storage options such as smart array controllers and storage enclosures.

appendix b

Calculating network throughput

The calculations here are rough estimates of how much data can be sent through a network connection (network throughput). The factors involved in this calculation are the bandwidth of the network, the latency between source and destination and the data transfer size.

The transfer size is not the frame size, but the size of the network buffer on the destination server. For Microsoft Windows 2000 servers, this buffer is 17,520 bytes by default for Ethernet. 17,520 is 12 packets of 1,460 bytes, the data segment size of an Ethernet frame.

The throughput calculation is:

Throughput = Transfer Size / Transfer Time; where

Transfer Time = Round Trip Time + Transfer size / Bandwidth.

What this means is that the round trip time (latency) is not incurred on each Ethernet packet, but only when the receive buffer on the destination is full—data will stream until the receive buffer is full, at which time a response from the destination to the target occurs and the round trip latency is in effect.

So the full equation is:

Throughput = Transfer size / (Round trip time + (Transfer Size/Bandwidth)),

which boils down to:

Throughput = 1/((Round Trip time/transfer size) + (1/Bandwidth))

**for more
information**

To learn more about HP High Availability and ProLiant Clusters visit the following Web site: <http://www.hp.com/servers/proliant/highavailability>.

Help us improve our technical communication. Let us know what you think about the technical information in this document. Your feedback is valuable and helps us structure future communications. Please send your comments to: <mailto:hawebserver@hp.com>

Microsoft is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

All other product names mentioned herein may be trademarks or registered trademarks of their respective companies.

The information in this document is subject to change without notice.

09/2002

P/N 17AH-0902A-WWEN