Technical report:

Microsoft Cluster Services and IBM System Storage N series

Best-Practices Integration

Document NS3375-0

September 24, 2007

© Copyright IBM Corporation, 2007. All Rights Reserved.
All trademarks or registered trademarks mentioned herein are the property of their respective holders

Table of contents

Abstract

Best-practice storage implementations deliver efficiency and value in Microsoft environments. This paper outlines optimal architecture and the mechanisms used to create integrated, highly available server clusters using IBM System Storage N series targets and Microsoft Cluster Services. The intent of this document is to provide an architectural overview of the functioning of Microsoft clustering technology such that IBM N series can be deployed to provide shared resources to a Microsoft cluster. This document is not intended to replace the administration guides that provide the detailed step-by-step process that should be followed to deploy an IBM N series device within a Microsoft cluster.

Introduction

Clusters are a set of loosely coupled systems that behave as a single system. By coupling multiple independent systems to present the appearance of a single system, clients and applications can interact with the single system presented by the cluster. The cluster can then provide for system and component failure with minimal impact to the applications and clients. In many cases failures in the cluster are transparent to clients.

After the introduction of Microsoft® Windows NT® Server, Microsoft added the ability to cluster Windows® servers. These clustering services are referred to as Microsoft Cluster Services (MSCS). Microsoft continued to enhance this clustering ability with its follow-on server operating systems.

This paper explains the basics of Microsoft clustering and how to make use of IBM® Systems Storage™ N series in Microsoft Windows cluster deployments. Again, this paper is not intended to replace the information provided in the administration guides. In these guides the reader can find the detailed steps to be followed when configuring IBM N series storage for use in Microsoft Windows clusters.

Cluster Models

There are two main software models used for clustering:

- *Shared resource,* in which software running on any system in the cluster may access any resource connected to any system in the cluster. If more than one system needs to access the same resource, then a way of serializing access to the resource is required (typically through a lock manager).
- *Shared nothing,* where each system in the cluster owns a subset of the cluster's resources. Only one system may own and access a particular resource at a time, and requests from clients are automatically routed to the system that owns the resource.

Microsoft clustering is based on the *shared nothing* model and has several elements: Windows servers, Ethernet networks (usually two), a storage network, and one or more applications. In a Microsoft cluster each server that is a member of the cluster is termed a *node*. On each node of the cluster there is a collection of components that provides cluster-specific functionality—these components are collectively referred to as the *cluster services*. Each of the cluster services components is referred to as a *resource.* While each node in a cluster has its own local resources, the cluster as a whole will have a set of common resources—these common resources are accessible by all nodes in the cluster. These shared resources include such things as disk drives, Ethernet networks, IP addresses, applications, and databases.

Resources can be grouped together for management purposes into a single logical entity called a *resource group*—usually a resource group contains resources that are related to the functioning of a specific application. There is a particular resource group, the *cluster resource group*, that is created as part of the initialization of the cluster. This resource group contains the resources associated with the management of the cluster.

There is a special shared resource known as a *quorum resource*; typically a quorum resource is a shared disk. The quorum resource provides the means by which a cluster manages itself. The quorum resource contains all of the data necessary for the operation (and recovery) of the cluster. This data is maintained in a *cluster database,* which contains information about all physical and logical elements in a cluster, including cluster objects, their properties, and configuration data. When a node comes back online after a failure, the other cluster nodes update the failed node's copy of the cluster database.

The quorum resource can be owned by only one node at a time. A node can form a cluster only if it can gain control of the quorum resource. Similarly, a node can join a cluster (or remain in an existing cluster) only if it can communicate with the node that controls the quorum resource.

Another important component in the functioning of a Microsoft cluster is the *cluster network*, sometimes called the *heartbeat* or *private network*. Each node that is part of a cluster will have a connection to this cluster network. The purpose of the cluster network is to enable each node in the cluster to send heartbeat messages to the other nodes in the cluster and from this to detect a failure in the cluster.

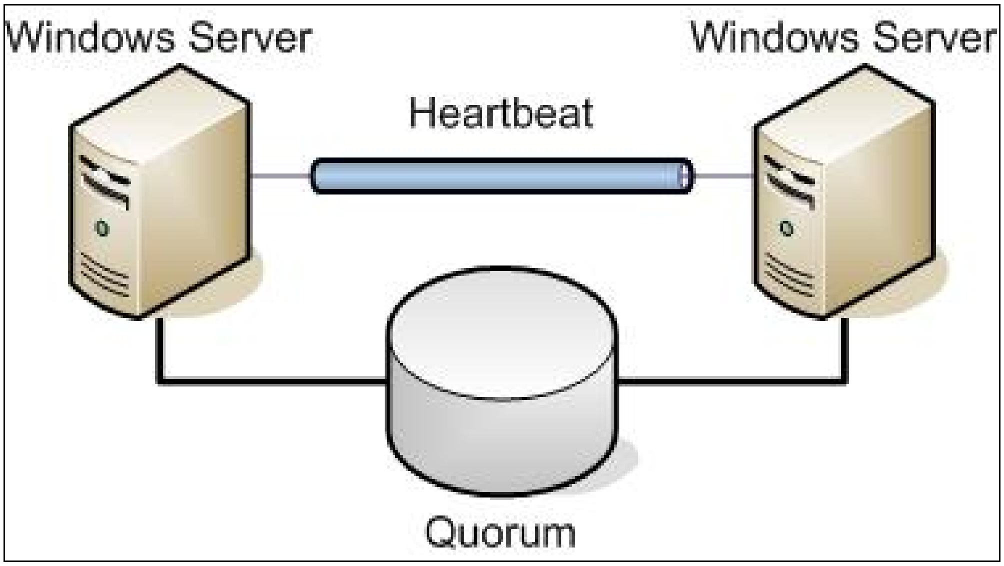


Figure 1) Two-node Microsoft cluster.

Figure 1 shows a two-node Microsoft cluster with the major physical components of the cluster—the shared resource quorum (in this case a disk drive on a shared fabric), the cluster network, and the minimum number of nodes required.

Since the purpose of a cluster is to provide highly available services to clients, Microsoft clustering provides the concept of a *virtual server*. A virtual server is an abstraction of a real server that is hosted on one of the cluster nodes and is providing application services to its client community. Should the cluster node hosting the virtual server cease to function, this will be detected by other nodes in the cluster, and the virtual server will be reinstantiated on another cluster node.

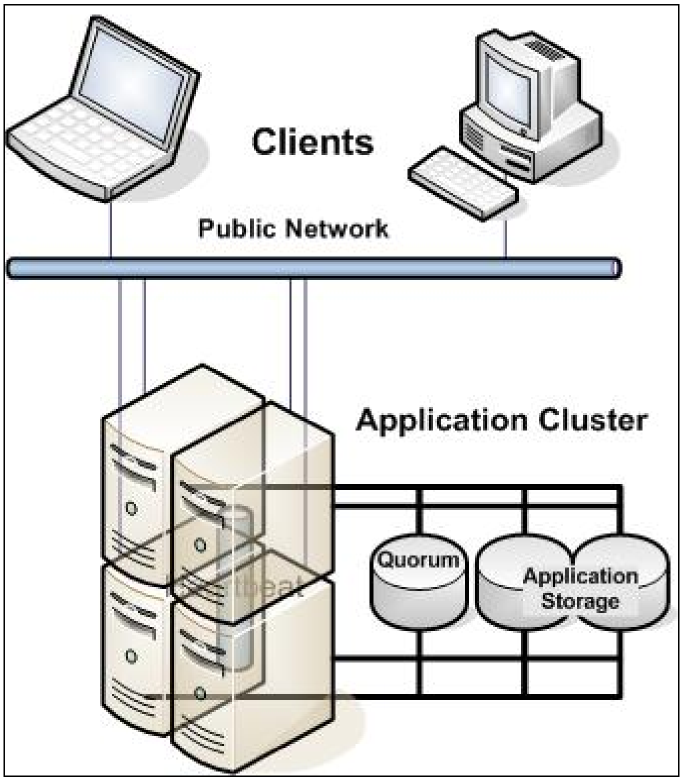


Figure 2) Microsoft application cluster deployment.

Figure 2 is a diagram of a typical cluster deployment. Client machines access their applications via the *public network* by connecting to an IP address assigned to the virtual server. A Microsoft cluster can host many virtual servers; the limitation for this hosting is a practical one based on the ability of the virtual server to handle its client community in a timely fashion.

In Windows 2003 Microsoft further refines the term *cluster model* by the manner in which the quorum resource is used in the server cluster. Three types of cluster models are available for deployment with server clusters:

- *Local quorum*, which is typically used for deploying dynamic file shares on a single cluster node to ease home directory deployment and administration or for testing and development.
- *Single quorum device*. This is the most widely used of the cluster models. The cluster configuration data is contained on a single storage device connected to all nodes of the cluster.
- *Majority node set*. This is new in Windows 2003. Each node in the cluster maintains its own copy of the cluster configuration, and the quorum resource ensures the consistency of this data across all the nodes in the cluster. Majority node set quorums are typically used in geographically dispersed clusters.

Cluster Network

For a Microsoft cluster to function successfully, it needs an Ethernet network connecting all the cluster nodes. This network handles the heartbeat messages used to detect the failure of cluster nodes. In addition to this network for handling the cluster management traffic, Microsoft recommends providing an additional Ethernet network, the public network, which is used by clients of the cluster to access their application services. It is important to separate the cluster management traffic from that associated with client services—cluster health management needs to occur in a timely fashion, and a dedicated network helps ensure this.

In a Microsoft cluster there is a process that is running on all nodes—the *Node Manager*. The job of the Node Manager is to manage node membership of the cluster; this includes joining nodes into the cluster, evicting nodes from the cluster, and detecting node failure. The Node Manager on each node communicates using a heartbeat message (every 1.2 seconds) with its counterparts on the other nodes of the cluster. The loss of two consecutive heartbeat message responses will result in a message being logged to the event log (event 1123), and the loss of six consecutive responses will result in the Node Manager forcing the cluster into a *regroup event*. During a regroup event all the nodes of a cluster are forced to verify their view of the cluster membership. While this regroup event is taking place, all write operations to any common storage in the cluster are suspended. Nodes that do not respond are removed from the cluster, and the active resources from the failed node are moved to another node in the cluster.

Quorum Management

The quorum device in a cluster is used to ensure that there is only a single management process for the cluster—this is intended to prevent *split-brain syndrome*. Split-brain syndrome is where more than one node claims ownership of some critical resource.

The quorum resource belongs to only a single node of a cluster at a time. The first node to create the cluster takes ownership of the quorum resource. Since the clusters described in this document make use of a shared disk as the quorum resource, the way in which the node takes ownership and maintains ownership of the quorum resource is through small computer system interface (SCSI) commands. When using a disk as a quorum resource, the drive must be a physical disk resource and not a partition, since changing ownership of the quorum involves moving the entire resource to another cluster node.

In a Microsoft cluster the first node in the cluster becomes the initial quorum owner. The quorum owner issues a reserve request for the quorum disk, and so long as it continues to be the quorum owner, it will continue to issue a reserve request every three seconds. Should the cluster enter a regroup event, the quorum owner will be forced to defend its ownership of the quorum through a challenge/defense mechanism.

When a regroup event is initiated, all nodes will issue a device or bus reset. This reset releases the reservation held by the quorum owner. Once a nonowner has issued a reset request, it waits 10 seconds before checking to see if the quorum resource is available; if the quorum owner is functioning correctly, it will regain its reservation (through its regular three-second reservation request) and thus defend its ownership of the quorum resource.

Integration of IBM N series Storage Targets

Integrating IBM N series storage targets with MSCS can be broken into a few basic steps:

- Cabling up the storage target to the cluster nodes
- Configuring the storage to present logical unit numbers (LUNs) to the cluster nodes
- Initializing and formatting the LUNs
- Creating the cluster.

Knowing that the way in which clusters are created and function depends on the proper access to a shared quorum device allows the correct configuration of IBM N series storage appliances. The quorum device (and indeed all the shared storage used by the cluster) must be visible to all nodes that are part of the cluster. In order for this to occur, the IBM N series storage appliance must be physically connected to all the servers that will become the nodes of the cluster.

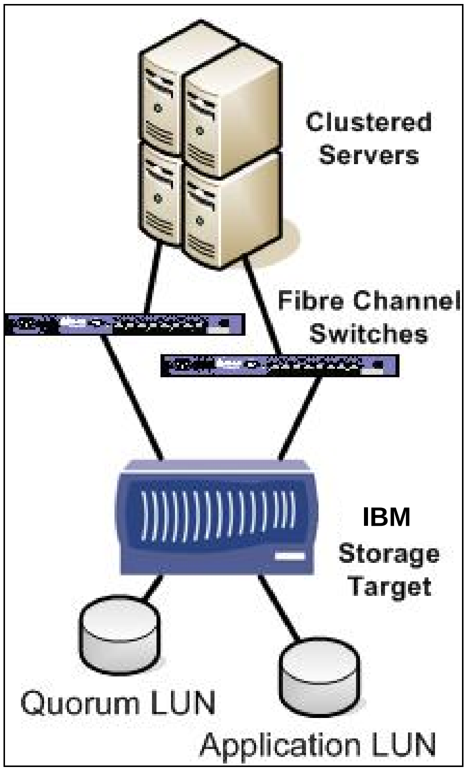


Figure 3) Fibre Channel–based cluster.

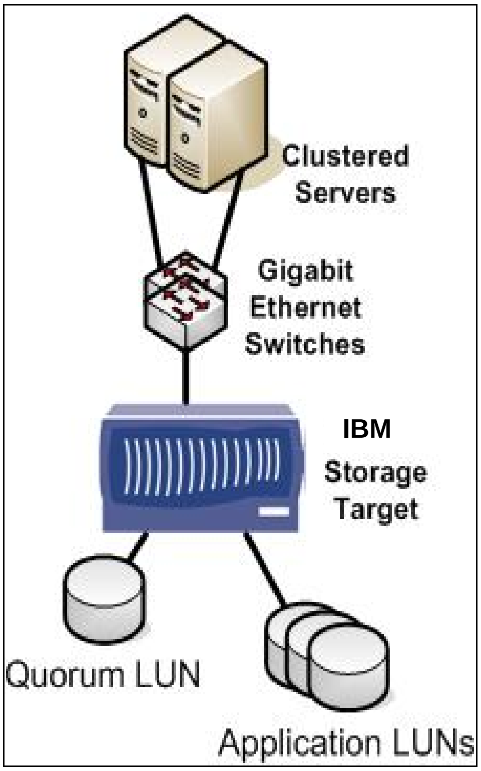


Figure 4) iSCSI-based cluster.

For a Fibre Channel (FC) deployment, this means that the FC storage targets in the IBM N series storage appliance need to be connected to a FC switching fabric that also has all the FC initiator host bus adaptors (HBAs) in the cluster nodes connected (refer to *Figure 3*). Once this cabling exercise is completed, the LUNs on the IBM N series storage appliance need to be unmasked (or mapped) to all the potential cluster nodes. It is not possible to form the cluster until this step occurs, since the quorum device needs to be available to all cluster nodes as part of the cluster creation process.

For iSCSI deployments the Gigabit Ethernet interfaces in the IBM N series storage appliance need to be connected to the appropriate Gigabit Ethernet interfaces in each of the cluster nodes (refer to *Figure 4*). Again, as in the case of the FC deployment, the LUNs on the IBM N series storage appliance need to be unmasked to all the nodes that will compose the cluster. Before the cluster is created, it will be necessary to log in using iSCSI from each prospective cluster node to the IBM N series storage target in order for the LUNs to become exposed to the nodes. As before, the cluster cannot be created until at least the quorum device is visible to all nodes that will become the cluster.

Once the LUNs have been unmasked to the potential cluster nodes, the LUNs themselves need to be “initialized”—this involves writing a signature on the drive. As all nodes that have had the LUNs unmasked will usually detect the presence of the new disks, it is important to remember that until the cluster has been formed, the drives should not be accessed from other than a single server.

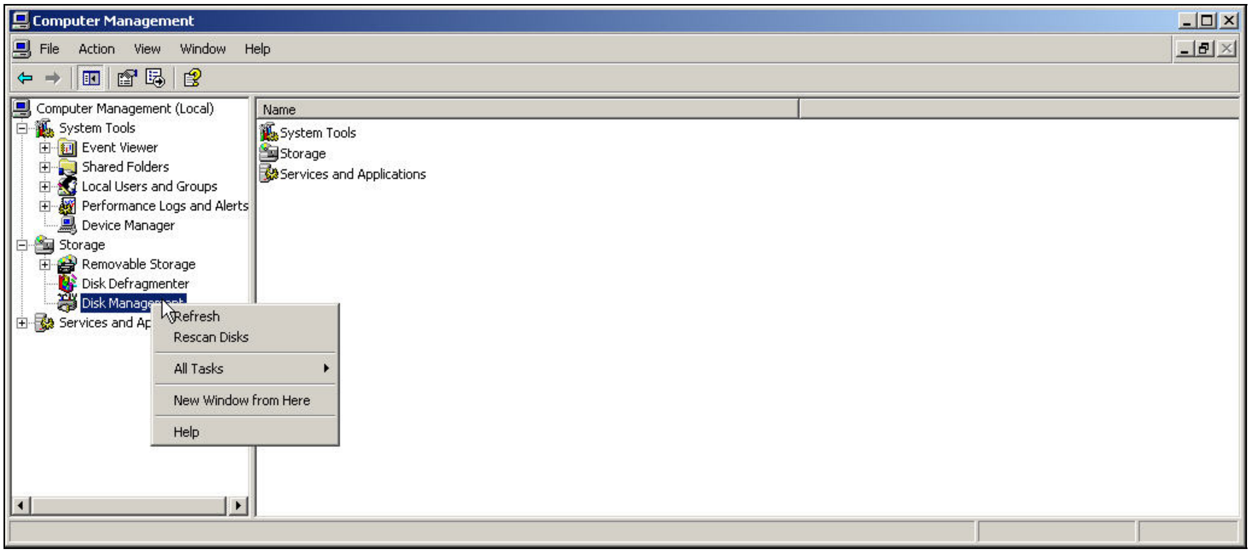
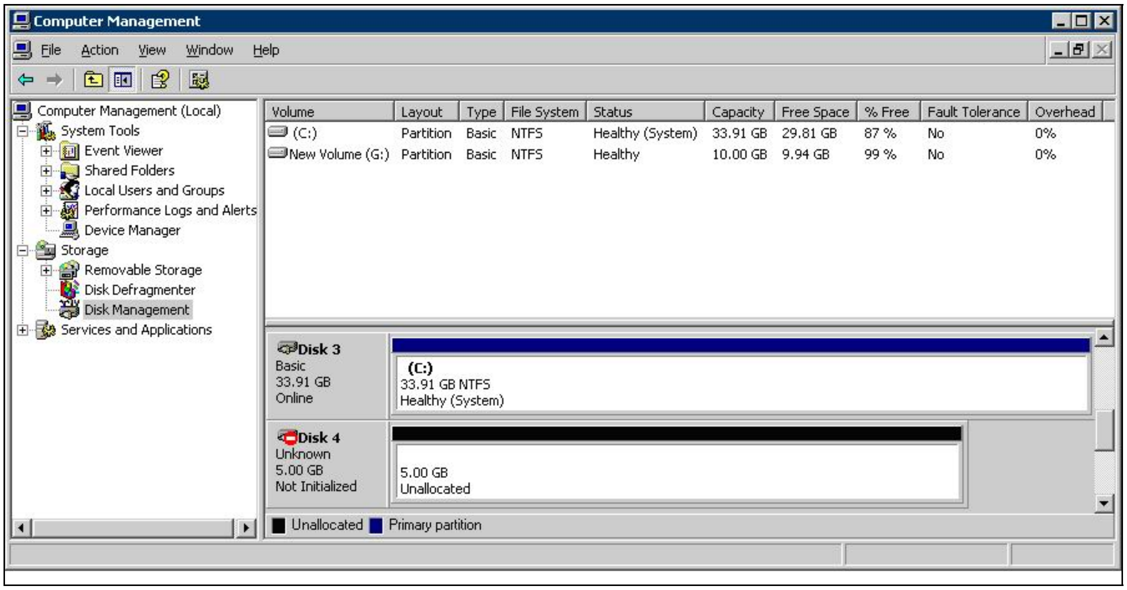


Figure 5) Rescan disks.

From one of the servers, open the Computer Management console—usually by this time a wizard will have started to assist in the initialization of the new storage that the server is seeing. If this has not occurred, then right-clicking Disk Management and “Rescan drives” will result in the drives being visible within the Computer Management console window (see *Figure 5*).



Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free	Fault Tolerance	Overhead
(C:)	Partition	Basic	NTFS	Healthy (System)	33.91 GB	29.81 GB	87 %	No	0%
New Volume (G:)	Partition	Basic	NTFS	Healthy	10.00 GB	9.94 GB	99 %	No	0%

Disk 3
Basic
33.91 GB
Online

(C:)
33.91 GB NTFS
Healthy (System)

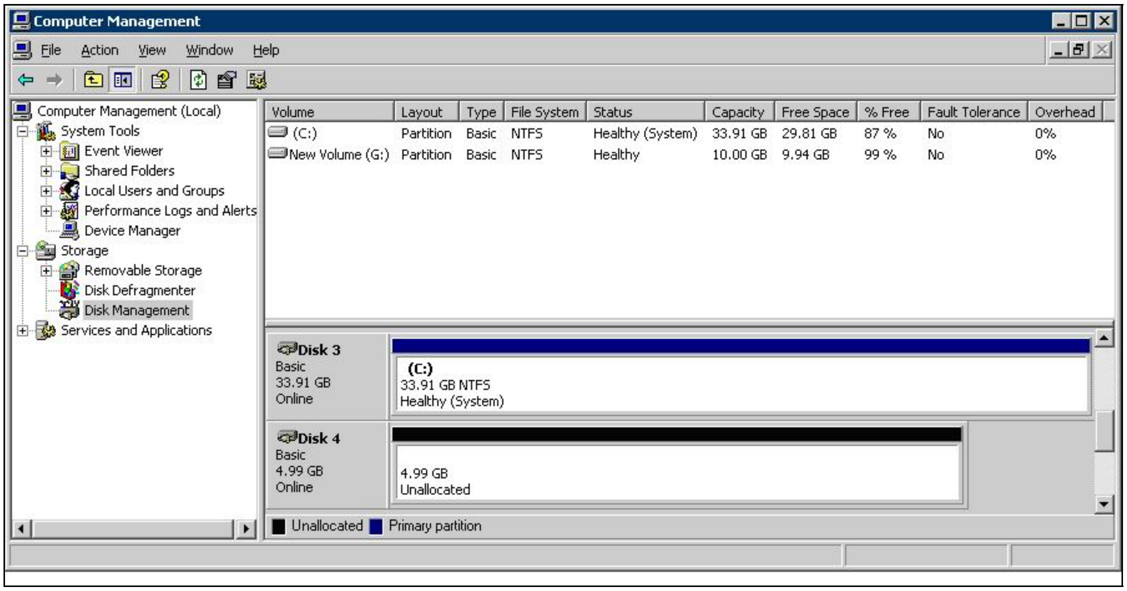
Disk 4
Unknown
5.00 GB
Not Initialized

5.00 GB
Unallocated

Figure 6) Disk management.

It is important when initializing the drives to keep them as *basic* disks, since at the time of this writing MSCS does not support *dynamic* disks as shared resources.

Once the LUNs are visible in Windows, they will appear in the Disk Management interface as *unknown* drives that have the *not initialized* attribute (see *Figure 6*). These drives should then be initialized, and they will appear as available and unallocated disks in the Disk Management interface (see *Figure 7*).



Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free	Fault Tolerance	Overhead
(C:)	Partition	Basic	NTFS	Healthy (System)	33.91 GB	29.81 GB	87 %	No	0%
New Volume (G:)	Partition	Basic	NTFS	Healthy	10.00 GB	9.94 GB	99 %	No	0%

Disk 3
Basic
33.91 GB
Online

(C:)
33.91 GB NTFS
Healthy (System)

Disk 4
Basic
4.99 GB
Online

4.99 GB
Unallocated

Figure 7) Disk management.

The next step is to create partitions on these drives and format the partitions. Once this operation is completed, the cluster can be created. The first node of the cluster will become the “owning” node of all the resources available to the cluster. As each subsequent node joins the cluster, MSCS will ensure that the shared resources are only visible to the node that is supposed to manage that particular resource.

Conclusion

Configuring IBM N series storage appliances as part of a Microsoft cluster environment is a relatively easy and painless process. It is important to remember that while the storage needs to be available to all cluster nodes, only one node is used to complete the initialization and formatting of the LUNs.

IBM N series has a Microsoft Management Console snap-in called IBM System Storage N series with SnapDrive® that makes the process outlined in the prior section even more transparent when connecting IBM N series storage appliances to Microsoft clusters—SnapDrive also works with nonclustered Windows servers.

Trademarks and Special Notices

© International Business Machines 1994-2007. IBM, the IBM logo, System Storage, and other referenced IBM products and services are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Network Appliance, the Network Appliance logo and SnapDrive are trademarks or registered trademarks of Network Appliance, Inc., in the U.S. and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.