OpenVMS Technical Journal V5

A Survey of Cluster Technologies

A Survey of Cluster Technologies

Ken Moreau, Solutions Architect, OpenVMS Ambassador

Overview

This paper surveys the cluster technologies for the operating systems available from many vendors, including IBM AIX, HP HP-UX, Linux, HP NonStop Kernel, HP OpenVMS, PolyServe Matrix Server, Sun Microsystems Solaris, HP Tru64 UNIX and Microsoft Windows 2000/2003. In addition, it discusses some technologies that operate on multiple platforms, including MySQL Cluster, Oracle 9i and 10g Real Application Clusters, and Veritas clustering products. It describes the common functions that all of the cluster technologies perform, shows where they are the same and where they are different on each platform, and introduces a method of fairly evaluating the technologies to match them to business requirements. As much as possible, it does not discuss performance, base functionality of the operating systems, or system hardware.

The focus for the audience of this document is a person who is technically familiar with one or more of the clustering products discussed here and who wishes to learn about one or more of the other clustering products, as well as anyone who is evaluating various cluster products to find which ones fit a stated business need.

Introduction

Clustering technologies are highly interrelated, with almost everything affecting everything else. This subject is broken down into five areas:

- Single/multisystem views, which defines how you manage and work with a system, whether as individual systems or as a single combined entity.
- Cluster file systems, which defines how you work with storage across the cluster. Cluster file systems are just coming into their own in the UNIX world, and this article will describe how they work, in detail.
- Configurations, which defines how you assemble a cluster, both physically and logically.
- Application support, which discusses how applications running on your single standalone system today, can take advantage of a clustered environment. Do they need to change, and if so how? What benefits are there to a clustered environment?
- Resilience, which describes what happens when bad things happen to good computer rooms. This covers host-based RAID, wide area "stretch" clusters, extended clusters, and disaster tolerant scenarios.

This article covers the capabilities of IBM High Availability Cluster Multiprocessing (HACMP) 5.1 for AIX 5L, Linux LifeKeeper V4.3, Microsoft SQL Server 2000 Enterprise Edition, MySQL Cluster 4.1, HP NonStop Kernel G06.22, HP OpenVMS Cluster Software V7.3-2, Oracle 9i/10g Real Application Clusters, PolyServe Matrix Server and Matrix HA for Linux and Windows, HP Serviceguard 11i for HP-UX, HP Serviceguard A.11.16 for Linux, Sun Microsystems SunCluster 3.1 in a SunPlex cluster of Solaris 9 servers, HP TruCluster V5.1b, Veritas Cluster Server and SANPoint Foundation Suite V3.5 and Windows 2000/2003 with the Cluster Service. It also discusses Windows SQL Server 2005 Enterprise Edition, which offers additional capabilities beyond SQL Server 2000. This has not been officially released by Microsoft at this time, but is close enough that it is safe to describe its functionality.

For Linux, the focus is on the high availability side, not the HPTC (i.e., Beowulf) technologies.

Single-System and Multisystem-View Clusters

In order to evaluate the cluster technologies fairly, you need to understand four terms: scalability, reliability, availability and manageability.

- Availability defines whether the application stays up, even when components of the cluster go down. If you have two systems in a cluster and one goes down but the other picks up the workload, that application is available even though half of the cluster is down. Part of availability is failover time, because if it takes 30 seconds for the application to fail over to the other system, the users on the first system think that the application is down for those 30 seconds. Any actions that the users are forced to take as part of this failover, such as logging in to the new

system, must be considered as part of the failover time, because the users are not doing productive work during that time. Further, if the application is forced to pause on the surviving system during the failover, to the users on the second system the application is down for those 30 seconds.

- Reliability defines how well the system performs during a failure of some of the components. If you get subsecond query response and if a batch job finishes in 8 hours with all of the systems in the cluster working properly, do you still get that level of performance if one or more of the systems in the cluster is down? If you have two systems in a cluster and each system has 500 active users with acceptable performance, will the performance still be acceptable if one of the systems fails and there are now 1,000 users on a single system? Keep in mind that the users neither know nor care how many systems there are in the cluster; they only care whether they can rely on the environment to get their work done.

  Notice that reliability and availability are orthogonal concepts, and it is possible to have one but not the other. How many times have you logged into a system (it was available), but it was so slow as to be useless (it was not reliable)?

- Scalability defines the percentage of useful performance you get from a group of systems. For example, if you add a second system to a cluster, do you double performance, or do you get a few percentage points less than that? If you add a third, do you triple the performance of a single system, or not?

- Manageability defines how much additional work it is to manage those additional systems in the cluster. If you add a second system to the cluster, have you doubled your workload because now you have to do everything twice? Or have you added only a very small amount of work, because you can manage the cluster as a single entity?

Multisystem-view clusters are generally comprised of two systems, where each system is dedicated to a specific set of tasks. Storage is physically cabled to both systems, but each file system can only be mounted on one of the systems. Applications cannot simultaneously access data from both systems at the same time, and the operating system files cannot be shared between the two systems. Therefore, a fully-independent boot device (called a "system root" or "system disk") with a full set of operating system and cluster software files for each system is required.

Multisystem-View Clusters In Active-Passive Mode

Multisystem-view clusters in active-passive mode are the easiest for vendors to implement. They simply have a spare system on standby in case the original system fails in some way. The spare system is idle most of the time, except in the event of a failure of the active system. It is called active-passive because during normal operations, one of the systems is active and the other is not . This is classically called N+1 clustering, where N=1 for a two-system cluster. For clusters with larger numbers of systems, one or more spare servers can take over for any of the active systems.

Failover can be manual or automatic. Because the systems are all cabled to the same storage array, a spare system monitors the primary system and starts the services on the spare system if it detects a failure of the primary system. The "heartbeat" function can come over the network or from a private interface.

Comparing a single system to a multisystem-view cluster in an active-passive mode, the availability, reliability, scalability, and manageability characteristics are as follows:

- Availability is increased because you now have multiple systems available to do the work. The odds of all of the systems being broken at the same time are fairly low but still present.

- Reliability can be nearly perfect in this environment, because if all of the systems are identical in terms of hardware, the application will have the same performance no matter which system it is running on.

- Scalability is poor (non-existent) in an active-passive cluster. Because the applications cannot access a single set of data from both systems, you have two systems' worth of hardware doing one system's worth of work.

- Manageability is poor, because it takes approximately twice as much work to manage two systems as it does to manage a single system. There are multiple system roots, so any patches or other updates need to be installed multiple times, backups need to be done multiple times, and so

forth. Furthermore, you have to test the failover and failback, which adds to the system management workload

Notice that the spare system is idle most of the time, and you are getting no business benefit from it. The other alternative is to have all of the systems working.

Multisystem-View Clusters in Active-Active Mode

The physical environment of a multisystem-view cluster in active-active mode is identical to that of active-passive mode. Two or more systems are physically cabled to a common set of storage, but only able to mount each file system on one of the systems. The difference is that multiple systems are performing useful work as well as monitoring each other's health. However, they are not running the same application on the same data, because they are not sharing any files between the systems.

For example, a database environment could segment their customers into two groups, such as by the first letter of the last name (for example, A-M and N-Z). Then each group would be set up on separate disk volumes on the shared storage, and each system would handle one of the groups. This is known as a "federated" database. Or one of the systems could be running the entire database and the other system could be running the applications that access that database.

In the event of a failure, one system would handle both groups.

This is called an N+M cluster, because any of the systems can take over for any of the other systems. One way to define N and M is to think about how many tires you have on your automobile. Most people automatically say four, but including the spare, they really have five tires. They are operating in an N+1 environment, because four tires are required for minimum operation of the vehicle. A variation is to use the equivalent of the "donut" tire – a server that offers limited performance but enough to get by for a short period of time. This can be thought of as having 4½ tires on the vehicle. The key is to define what level of performance and functionality you require, and then define N and M properly for that environment.

Failover can be manual or automatic. The "heartbeat" function can come over the network or from a private interface. Comparing a single system to a multisystem-view cluster in active-active mode, the availability, reliability, scalability, and manageability characteristics are as follows:

- Availability is increased because you now have multiple systems available to do the work. As in the active-passive environment, the odds of all systems being broken at the same time are fairly low but still present.
- Reliability may be increased, but is commonly decreased. If two systems are each running at 60% of capacity, the failure of one will force the surviving system to work at 120% of capacity, which is not optimum because you should never exceed about 80% of capacity.
- Scalability for any given workload is poor in this situation because each workload must still fit into one system. There is no way to spread a single application across multiple systems.
- Manageability is slightly worse than the active-passive scenario, because you still have two independent systems, as well as the overhead of the failover scripts and heartbeat.

Failover of a Multisystem-View Cluster (Active-Active or Active-Passive)

One of the factors affecting availability is the amount of time it takes to accomplish the failover of a multisystem-view cluster, whether active-active or active-passive. The surviving system must:

- Notice that the other system is no longer available, which is detected when the "heartbeat" function on the surviving system does not get an answer back from the failed system.
- Mount the disks that were on the failing system. Remember that the file systems are only mounted on one system at a time: this is part of the definition of multisystem-view clusters. The surviving system must mount the disks that were mounted on the other system, and then possibly perform consistency checking on each volume. If you have a large number of disks, or large RAID sets, this could take a long time.
- Start the applications that were active on the failing system.
- Initiate the recovery sequence for that software. For databases, this might include processing the journalling logs in order to process any in-flight transactions that the failing system was performing at the time of the failure.

In large environments, it is not unusual for this operation take 30-60 minutes. During this recovery time, the applications that were running on the failed system are unavailable, and the applications that were running on the surviving system are not running at full speed, because the single system is now doing much more work.

Single-System-View Clusters

In contrast, single-system-view clusters offer a unified view of the entire cluster. All systems are physically cabled to all shared storage and can directly mount all shared storage on all systems. This means that all systems can run all applications, see the same data on the same partitions, and cooperate at a very low level. Further, it means that the operating system files can be shared in a single "shared root" or "shared system disk," reducing the amount of storage and the amount of management time needed for system maintenance. There may be spare capacity, but there are no spare systems. All systems can run all applications at all times.

In a single-system-view cluster, there can be many systems. Comparing a series of independent systems to the same number of systems in a single-system-view cluster, the availability, reliability, scalability, and manageability characteristics are as follows:

- Availability is increased because you now have multiple systems to do the work. The odds of all systems being broken at the same time is now much lower, because potentially you can have many systems in the cluster.
- Reliability is much better, because with many systems in the cluster, the workload of a single failed system can be spread across many systems, increasing their load only slightly. For example, if each system is running at 60% capacity and one server out of four fails, 1/3 of the load is placed on each of the other systems, increasing their performance to 80% of capacity, which will not affect reliability significantly.
- Scalability is excellent because you can spread the workload across multiple systems. If you have an application that is simply too big for a single computer system (even one with 64 or 128 CPUs and hundreds of gigabytes of memory and dozens of I/O cards), you can have it running simultaneously across many computer systems, each with a large amount of resources, all directly accessing the same data.
- Manageability is much easier than the equivalent job of managing this number of separate systems, because the entire cluster is managed as a single entity. There is no increase in management workload even when you have many systems.

The advantages in failover times over multisystem-view clusters comes from not having to do quite so much work during a failover:

- The surviving systems must detect the failure of the system. This is common between the two types of clusters.
- The surviving systems do not have to mount the disks from the failed system; they are already mounted.
- The surviving systems do not have to start the applications; they are already started.
- The execution of the recovery script is common between the two schemes, but it can begin almost instantly in the single-system-view cluster case. The application recovery time will be similar on both types of clusters, but if you have a large number of small systems, you can achieve parallelism even in recovery, so that your recovery can be faster in this case as well.

One criticism of shared root environments with a single root for the entire cluster is that this represents a single point of failure. If a hardware failure causes the shared root device to be inaccessible, or an operator error causes corruption on the shared root (such as applying a patch incorrectly or deleting the wrong files), the entire cluster will be affected. These concerns must be balanced against the amount of work involved in maintaining multiple system roots. Furthermore, an incorrect patch on one system root can cause incompatibility with the other cluster members. Such problems can be difficult to diagnose.

The system administrator must set up the operational procedures (including the number of shared roots) for their environment in such as way that the possibility of failure is minimized, and services are still delivered in a cost-effective manner. Frequent backups, hardware and software RAID, and good

quality assurance and testing procedures can help reduce the possibility of failure in either environment.

Now that the terms are defined, you can see how different cluster products work.

	Multisystem view	**Single-system view**	**Shared root**
HACMP AIX, Linux	Yes	No	No
LifeKeeper Linux, Windows	Yes	No	No
MySQL Cluster AIX, HP-UX, Linux, Solaris, Windows	Yes (MySQL Server)	Yes	No
NonStop Kernel	Yes	Yes	Each node (16 CPUs)
OpenVMS Cluster Software OpenVMS	Yes	Yes	Yes
Oracle 9i/10g RAC Many O/S's	Yes (Oracle DB)	Yes	Effectively yes ($ORACLE_HOME)
PolyServe Matrix Linux, Windows	Yes	Yes	No
Serviceguard HP-UX, Linux	Yes	No	No
SQL Server 2000/2005 Windows	Yes	No	No
SunCluster Solaris	Yes	No	No
TruCluster Tru64 UNIX	No	Yes	Yes
Veritas Cluster Server AIX, HP-UX, Linux, Solaris, Windows	Yes	No	No
Windows 2000/2003 Cluster Service Windows	Yes	No	No

Figure 1 Types of Clusters

HACMP

High Availability Cluster Multiprocessing (HACMP) 5.1 for AIX 5L runs on the IBM pSeries (actually an RS/6000 using the Power4 chip), and for Linux runs on a variety of platforms. It is a multisystem image cluster, where each system in the cluster requires its own system disk. Management is done either through the included Cluster Single Point Of Control (C-SPOC) or by the layered product Cluster Systems Management (CSM), which can manage mixed AIX and Linux systems in the same cluster. In both cases, you issue commands one time and they are propagated to the different systems in the cluster. Clusters can be configured either as active-passive (which IBM calls "standby") or active-active (which IBM calls "takeover") configurations.

Previous versions of HACMP came in two varieties: HACMP/ES (Enhanced Scalability) and HACMP (Classic). V5.1 includes all of the features of HACMP/ES.

Linux Clustering

Linux clustering is focused either on massive system compute farms (Beowulf and others) or a high availability clustering scheme. This article specifically does not address the High Performance Technical Computing market here, which breaks down a massive problem into many (hundreds or thousands) tiny problems and hands them off to many (hundreds or thousands) small compute engines. This is not really a high availability environment because if any of those compute engines fails, that piece of the job has to be restarted from scratch.

Most of the Linux high availability efforts are focused on multisystem-view clusters consisting of a small number of systems from which applications can fail over from one system to the other. Cluster file system projects such as Lustre and GFS are discussed later, but these do not offer shared root, so systems in Linux clusters require individual system disks.

There are some other projects that are focused on single-system-view clusters. One of these is the work being done by HP as part of the Single System Image Linux project. Another is from Qlusters Corporation, specifically the ClusterFrame XHA and ClusterFrame SSI products based on OpenMosix. At this time these are focused on the HPTC market, but when they prove themselves in the commercial high availability market space, they will have significant capabilities that match or exceed every other clustering product. Visit http://openssi.org for more information on the HP project, and http://www.qlusters.com for more information on ClusterFrame XHA and SSI.

MySQL Cluster

MySQL Cluster is a layer on top of MySQL, the open source database that runs on AIX, HP-UX, Linux (Red Hat and SUSE), Mac OS X, Windows 2000 and XP, and is being planned for OpenVMS. The software and intellectual property were acquired from Ericsson, and was integrated as open source into the Storage Engine of MySQL Server.

There are three components to a MySQL Cluster: application nodes, database server or storage nodes, and management nodes. Application nodes run MySQL Server and connect to the database server nodes running MySQL Cluster, and are managed by the management nodes. The different nodes can either be processes on a single server or distributed on multiple servers. MySQL Cluster is designed to work on "shared nothing" operating systems, where each node has private storage.

MySQL offers a multisystem view of the database, and MySQL Cluster adds single-system view. It does not support sharing of disks, but transparently fragments the database over the systems in the cluster with real-time replication, so that the database information can be accessed from any system in the cluster.

NonStop Kernel

NonStop Kernel (NSK, formerly the Tandem Guardian operating system) runs on the HP NonStop servers (formerly NonStop Himalaya or Tandem Himalaya servers), and is configured as a single-system-view cluster. It offers true linear scalability as you add processors to the environment, because of the shared-nothing architecture and superb cluster interconnect. 2 to 16 processors can be configured to have a shared root and be considered one system. A cluster of systems, both local and geographically distributed, is centrally managed with the Open Systems Manager (OSM) console.

OpenVMS Cluster Software

OpenVMS Cluster Software has always been the gold standard of clustering, with almost linear scalability as you add systems to the cluster. It can be configured as either multisystem view or single-system view, although the most common is single-system view. It supports single or multiple system disks.

Oracle 9i/10g Real Application Clusters

Oracle 9i/10g Real Application Clusters (RAC) is the next generation of Oracle Parallel Server, and runs on the Oracle 9i and 10g database on every major computing platform. It offers a single-system-view of the database files, such that external applications can connect to the database instance on any of the systems in the cluster. It does not offer a multisystem-view of the database, but this is easily achieved by simply running the database without RAC.

Oracle achieves the functionality of a shared root (called $ORACLE_HOME), but accomplishes it differently on the different host operating systems. On single-system-view operating systems that offer

clustered file systems, $ORACLE_HOME is placed in a shared volume and made available to all of the systems in the cluster. On multisystem-view operating systems that do not offer clustered file systems, Oracle replicates all of the operations to individual volumes, one per system in the cluster, without forcing the user to take any action. The installation, patches, and monitoring are the same whether there is one $ORACLE_HOME or multiple, replicated $ORACLE_HOMEs.

Oracle is steadily adding functionality to RAC, which requires less support from the base operating systems. For example, 9i RAC required the addition of Serviceguard Extensions for RAC (SGeRAC) on HP-UX, while 10g RAC does not require SGeRAC. Further, 10g RAC is capable of running without the underlying operating system itself being clustered. As a result, HACMP, Serviceguard, SunClusters, and Windows 2000/2003 Cluster Server are now optional for Oracle 10g RAC.

PolyServe Matrix HA and Matrix Server

PolyServe Matrix Server is a clustered file system for Linux and Windows which includes a high availability and a cluster management component. The HA component provides automated failover and failback of applications. Each node in a PolyServe cluster requires its own system disk, which can be local or SAN boot. Matrix Server allows the underlying disk volumes to be accessed for read-write simultaneously from all nodes. It also allows a unified view of device management, such that device names are common across all systems in the cluster regardless of the order that the devices were discovered during a boot. The management application is CLI and GUI based, and allows the cluster to be managed as a single entity from any node in the cluster. Matrix Server is primarily an installable file system, and so does not itself offer a multisystem view because the underlying operating systems offer that as the default. Similarly, Matrix Server does not offer a shared root, because it is a layer on top of the operating system and is activated late in the boot process.

Serviceguard

Serviceguard (also known as MC/Serviceguard) is a multisystem-view failover cluster. Each system in a Serviceguard cluster requires its own system disk. There are excellent system management capabilities from the Service Control Manager and the Event Management Service, including the ability to register software in the System Configuration Repository, get system snapshots, compare different systems in the cluster, and install new instances of the operating system and applications by copying existing instances using Ignite/UX. It is also well integrated with HP/OpenView.

Serviceguard Manager can configure, administer, and manage HP-UX and Linux Serviceguard clusters through a single interface. Each cluster must be homogeneous; that is, each cluster can only be running one operating system. Business continuity solutions to achieve disaster tolerance are available. HP-UX offers Campuscluster, Metrocluster, and Continentalcluster. Metrocluster functionality is offered on Linux through Serviceguard for Linux integration with Cluster Extension XP. Additional complementary products on Linux include Serviceguard Extension for SAP for Linux and an application toolkit for Oracle. Contributed toolkits are available for leading Linux applications.

SQL Server 2000/2005 Enterprise Edition

Microsoft SQL Server 2000 Enterprise Edition is a multisystem-view failover clustered database, running on Microsoft Windows 2000/2003. SQL Server 2005 is the next release of this product, and is available on Windows 2003. They are available in both 32-bit and 64-bit versions for the various hardware platforms. They provide both manual and automatic failover of database connections between servers. A database can be active on only a single instance of SQL Server, and each server requires its own installation. Unless specifically noted, all references to functionality in this article apply to both versions equally.

SunCluster

SunCluster 3.1 is a multisystem-view failover cluster. A group of Solaris servers running SunCluster software is called a SunPlex system. Each system in a SunPlex requires its own system disk, and Sun recommends keeping the "root" passwords the same on all systems. This has to be done manually, which gives you some idea about the level of management required by a SunPlex. The Cluster File System (CFS) offers a single-system-view of those file systems that are mounted as a CFS. The Sun Management Center and SunPlex Manager are a set of tools that manage each system as a separate entity but from a centralized location.

TruCluster V5.1b

TruCluster V5.1b represents a major advance in UNIX clustering technology. It can only be configured as a single-system-view. The clustering focus is on managing a single system or a large cluster in exactly the same way, with the same tools, and roughly the same amount of effort. It offers a fully-shared root and a single copy of almost all system files.

Veritas

Veritas offers several products in this area, but then offers many combinations of these products under separate names. The base products are:

- Veritas Cluster Server (VCS) manages systems in a cluster, with a GUI interface. It is unique in that it can manage multiple different clusters at a time. It can simultaneously manage systems running AIX, HP-UX, Linux, Solaris, and Windows running the Veritas Cluster Server software. Each cluster must be homogeneous; that is, each cluster can only be running one operating system.
- Veritas File System (VxFS) is a journaled file system that works in either a standalone system or a cluster. A "light" version of this product is included with HP-UX 11i Foundation Operating Environment, and the full version is included in the Enterprise Operating Environment as Online JFS.
- Veritas Global Cluster Manager (GCM) manages geographically-distributed Veritas Cluster Server clusters from a central console. Applications can be monitored across multiple clusters at multiple sites, and can be migrated from one site to another. The application service groups in each cluster must be setup by VCS, but can then be monitored and migrated through GCM.
- Veritas Volume Manager (VxVM) manages volumes, whether they are file systems or raw devices. A "light" version of this product is included with HP-UX 11i, and offers similar functionality as the HP-UX 11i Logical Volume Manager and the TruCluster Logical Storage Manager.
- Veritas Cluster Volume Manager (CVM) offers the same functionality as VxVM but does it across multiple systems in a cluster. An important distinction is that CVM requires that every system mount every shared volume.
- Veritas Volume Replicator (VVR) allows disk volumes to be dynamically replicated by the host, both locally and remotely. This is similar to the "snap/clone" technology in the StorageWorks storage controllers.

Veritas combines these into many different packages. The two important ones for this discussion are:

- SANPoint Foundation Suite – HA (SPFS – HA), which includes VxFS with cluster file system extensions, VxVM with cluster extensions, and the Veritas Cluster Server
- Veritas DataBase Extension Advanced Cluster (DBE/AC) for Oracle 9i/10g Real Application Clusters (RAC), which includes VxFS, VxVM, and CVM, along with an implementation of the Oracle Disk Manager (ODM) API for Oracle to use to manage the volumes

The Veritas Network Backup (NBU) is not a cluster technology; therefore, it is not addressed in this paper.

Most of these products run under AIX, HP-UX, Linux, Solaris, and Windows, but SANPoint Foundation Suite – HA runs only under HP-UX and Solaris. Check with Veritas for specific versions and capabilities of the software for specific versions of the operating systems, and look for more discussion of these in later sections of this paper. In some cases the products replace the operating system's clusterware (Cluster Server, Cluster Volume Manager), and in other cases they are enhancements to the operating system's products (Cluster File System, Volume Replicator). All of the products are offered by both HP and Veritas, and supported by either company through the Cooperative Service Agreement (ISSA).

Windows 2000/2003 DataCenter is a multisystem-view failover cluster. Applications are written to fail over from one system to another. Each system in a Windows 2000/2003 cluster requires its own system disk, but the Cluster Administrator tool can centralize the management of the cluster.

Cluster File Systems

Cluster file systems are how systems communicate with the storage subsystem in the cluster. There are really two technologies here: one addresses how a group of systems communicates with volumes that are physically connected to all of the systems, and the other addresses how a group of systems communicates with volumes that are only physically connected to one of the systems.

Network I/O allows all of the systems in a cluster to access data, but in a very inefficient way that does not scale well in most implementations. Let's say that volume A is a disk or tape drive which is physically cabled to a private IDE or SCSI adapter on system A. It cannot be physically accessed by any other system in the cluster. If any other system in the cluster wants to access files on the volume, it must do network I/O, usually by some variation of NFS.

Specifically, if system B wants to talk to the device that is mounted on system A, the network client on system B communicates to the network server on system A in the following way:

1. An I/O connection is initiated across the cluster interconnect from system B to system A.
2. System A receives the request, and initiates the I/O request to the volume.
3. System A gets the data back from the volume, and then sends an I/O request back to system B.

Notice that there are three I/Os for each disk access. For NFS, there is also significant locking overhead with many NFS clients. This leads to poor I/O performance in an active-active system.

Every system offers network I/O in order to deal with single-user devices that cannot be shared, such as tapes, CD-ROM, DVD, or diskettes, and to allow access to devices that are on private communications paths, such as disks on private IDE or SCSI busses. This type of access is known as "proxy file system."

In contrast, direct access I/O (also known as "concurrent I/O") allows each system to independently access any and all devices, without going through any other node in the cluster. Notice that this is different from UNIX direct I/O, which simply bypasses the file system's cache. Most database systems do direct I/O both in a clustered and non-clustered environment, because they are caching the data anyway, and don't need to use the file system's cache.

Implementing direct access I/O allows a cluster file system to eliminate two of the three I/Os involved in the disk access in network I/O, because each system talks directly over the storage interconnect to the volumes. It also provides full file system transparency and cache coherency across the cluster.

You may object that we could overwhelm a single disk with too many requests. This is absolutely true, but this is no different from the same problem with other file systems, whether they are clustered or not. Single disks, and single database rows, are inevitably going to become bottlenecks. You design and tune around them on clusters in exactly the same way you design and tune around them on any other single-member operating system, using the knowledge and tools you use now.

These technologies are focused on the commercial database environments. But in the High Performance Technical Computing (HPTC) environment, the requirements are slightly different. The IBM General Parallel File System (GPFS) offers direct access I/O to a shared file system, but focuses on the HPTC model of shared files, which differs from the commercial database model of shared files in the following ways:

- The commercial model optimizes for a small number of multiple simultaneous writers to the same area (byte range, record or database row) of a shared file, but assumes that this occurs extremely frequently, because commercial databases and applications require this functionality.
- The HPTC model optimizes for throughput because, while the number of multiple simultaneous writers to any given file may be large (hundreds or even thousands of systems), the applications are designed so that only one process is writing to any given byte range. In the unlikely event of multiple writers to a single byte range of a shared file, the HPTC model switches to network I/O semantics, and ships all of the data to a single master system for that

byte range. This has been found to be more efficient overall because the condition occurs so infrequently in the HPTC world.

This paper focuses on the commercial database environment.

The I/O attributes of cluster products are summarized in the following table.

	Network I/O	**Direct Access I/O**	**Distributed Lock Manager**
HACMP AIX, Linux	Yes	Raw devices and GPFS	Yes (API only)
LifeKeeper Linux, Windows	NFS	Supplied by 3rd parties	Supplied by 3rd parties
MySQL Cluster AIX, HP-UX, Linux, Solaris, Windows	No (supplied by native O/S)	Yes (effectively)	Yes (for database only)
NonStop Kernel	Data Access Manager	Effectively Yes	Not applicable
OpenVMS Cluster Software OpenVMS	Mass Storage Control Protocol	Files-11 on ODS-2 or -5	Yes
Oracle 9i/10g RAC Many O/S's	No (supplied by native O/S)	Yes, both raw devices & Oracle file systems	Yes
PolyServe Matrix Linux, Windows	No (supplied by native O/S)	Yes	Yes (for file system only)
Serviceguard HP-UX, Linux	Yes	Supplied by 3rd parties	Supplied by 3rd parties
SQL Server 2000/2005 Windows	No (supplied by the native O/S)	No	No
SunCluster Solaris	Yes	Supplied by 3rd parties	Supplied by 3rd parties
TruCluster Tru64 UNIX	Device Request Dispatcher	Cluster File System (requires O_DIRECTIO)	Yes
Veritas SPFS HP-UX, Solaris	No (supplied by the native O/S)	Yes (SPFS or DBE/AC)	Yes (SPFS or DBE/AC)
Windows 2000/2003 Cluster Service Windows	NTFS	Supplied by 3rd parties	Supplied by 3rd parties

Figure 2 Cluster I/O Attributes

Every system in the world can do network I/O in order to share devices that are on private storage busses.

HACMP, LifeKeeper, and Serviceguard do network I/O using NFS; NonStop Kernel does it with the Data Access Manager (DAM, also called the "disk process" or DP2); OpenVMS Cluster Software does it with the Mass Storage Control Protocol (MSCP); SunCluster does it with NFS or the Cluster File System; TruCluster does it both with the Device Request Dispatcher (DRD) and the Cluster File System; and Windows 2000/2003 does it with NTFS and Storage Groups. MySQL, Oracle, SQL Services, and Veritas use the native I/O system of the operating system on which they are running.

The more interesting case is direct access I/O.

HACMP offers direct access I/O to raw devices for two to eight systems in a cluster. However, HACMP does not itself handle the locks for raw devices. Instead, it requires that applications use the Cluster Lock Manager APIs to manage concurrent access to the raw devices. The Concurrent Logical Volume Manager provides "enhanced concurrent mode," which allows management of the raw devices through the cluster interconnect, which should not be confused with a cluster file system as it applies only to raw devices.

Linux has projects being done by HP and Cluster File Systems Inc for the US Department of Energy to enhance the Lustre File System originally developed at Carnegie Mellon University. This enhancement is focused on high-performance technical computing environments and is called the Scalable File Server. This uses Linux and Lustre to offer high throughput and high availability for storage, but does not expose this clustering to the clients.

MySQL Cluster does not offer direct access I/O, but it achieves the same effect by fragmenting the database across the systems in the cluster and allowing access to all data from any application node in the cluster. This provides a unified view of the database to any application that connects to the MySQL Cluster. Each database node is responsible for some section of the database, and when any data in that section is updated, the database nodes synchronously replicate the changed information to all other database nodes in the cluster. It is in fact a "fragmented" (using MySQL terminology) and a "federated" (using Microsoft and Oracle terminology) database, and yet it behaves as a single-system image database.

NonStop Kernel is interesting because, strictly speaking, all of the I/O is network I/O. But because of the efficiencies and reliability of the NSK software and cluster interconnect, and the ability of NSK to transparently pass ownership of the volume between CPUs within a system, it has all of the best features of direct access I/O without the poor performance and high overhead of all other network I/O schemes. Effectively, NSK offers direct access I/O, even though it is done using network I/O. The NonStop Kernel (including NonStop SQL) utilizes a "shared-nothing" data access methodology. Each processor owns a subset of disk drives whose access is controlled by the Data Access Manager (DAM) processes. The DAM controls and coordinates all access to the disk so a DLM is not needed.

OpenVMS Cluster Software extends the semantics of the Files-11 file system transparently into the cluster world, offering direct I/O to any volume in the cluster from any system in the cluster that is physically connected to the volume. For volumes that are not physically connected to a specific system, OpenVMS Cluster Software transparently switches to network I/O. Opening a file for shared access by two processes on a single system, and opening the same file for shared access by two processes on two different systems in a cluster, works identically. In effect, all file operations are automatically cluster-aware.

Oracle 9i/10g RAC does not offer network I/O, but requires that any volume containing database files be shared among all systems in the cluster that connect to the shared database. 9i/10g RAC offers direct access I/O to raw devices on every major operating system, with the exception of OpenVMS, where it has used the native clustered file system for many years (starting with the original version of Oracle Parallel Server). Oracle has implemented its own Oracle Clustered File System (OCFS) for the database files on Linux and Windows as part of Oracle 9i RAC 9.2, and is extending OCFS to other operating systems in Oracle 10g as part of the Automated Storage Manager (ASM).

In general, the OCFS cannot be used for the Oracle software itself ($ORACLE_HOME), but can be used for the database files. The following table shows which cluster file systems can be used for Oracle software and database files:

	Oracle software	**Oracle database files**
HACMP/ES on AIX	Local file system only	Raw, GPFS
LifeKeeper on Linux	Local file system only	RAW, OCFS
OpenVMS Cluster SW	OpenVMS cluster file system	OpenVMS cluster file system
Serviceguard on HP-UX	Local file system only	RAW, Veritas DBE/AC
Serviceguard on Linux	Local file system only	Raw, OCFS
SunClusters on Solaris	Solaris GFS	Raw, Veritas DBE/AC

		(Solaris GFS is not supported for database files)
TruCluster on Tru64 UNIX	TruCluster CFS	Raw, TruCluster CFS
Windows 2000/2003 Cluster	OCFS	Raw, OCFS

Figure 3 Cluster File Systems for Oracle

"Local file system only" means that the Oracle software ($ORACLE_HOME) cannot be placed on a shared volume; each server requires its own copy, as described above. Interestingly, the Solaris Global File Service does support a shared $ORACLE_HOME, but does not support shared Oracle database files.

PolyServe Matrix Server does not offer network I/O as such, because it is available in the underlying operating systems. Matrix Server performs direct access I/O to any volume of the cluster file system under its control and uses its distributed lock manager to perform file locking and cache coherency. It supports on-line addition and removal of storage, and the meta-data is fully journaled. PolyServe Matrix Server and OpenVMS Cluster Software are the only cluster products with fully distributed I/O architectures, with no single master server for I/O. PolyServe manages the lock structure for each file system independently of the lock structures for any other file systems, so there is no bottleneck or single point of failure.

Serviceguard and Windows 2000/2003 Cluster Service do not offer a direct access I/O methodology of their own, but rely on 3rd party tools such as Oracle raw devices or the Veritas SANpoint Foundation Suite – High Availability. Serviceguard uses an extension to the standard Logical Volume Manager for clusters, called the Shared Logical Volume Manager (SLVM) to create volumes that are shared among all of the systems in the cluster. Notice that this only creates the volume groups: the access to the data on those volumes is the responsibility of the application or the 3rd party cluster file system.

SQL Services 2000/2005 does not offer direct access I/O.

SunCluster does not offer direct access I/O in its cluster file system (Global File Service, or GFS), which simply allows access to any device connected to any system in the cluster, independent of the actual path from one or more systems to the device. In this way devices on private busses such as tapes or CD-ROMs can be accessed transparently from any system in the SunPlex. The GFS is a proxy file system for the underlying file systems, such as UFS or JFS, and the semantics of the underlying file system are preserved (that is, applications see a UFS file system even though it was mounted as GFS). Converting a file system to a GFS destroys any information about the underlying file system. GFSs can only be mounted cluster-wide, and cannot be mounted on a subset of the systems in the cluster. There must be entries in the /etc/vfstab file on each system in the cluster, and they must be identical. (SunClusters does not provide any checks on this or tools to help manage this.)

Multiported disks can also be part of the GFS, but Sun recommends that only two systems be connected to a multiported disk at a time (see below). Secondary systems are checkpointed by the primary system during normal operation, which causes significant cluster performance degradation and memory overhead. The master system performs all I/O to the cluster file system upon request by the other systems in the cluster, but cache is maintained on all systems that are accessing it.

SunCluster manages the master and secondary systems for multiported disks in a list of systems in the "preferenced" property, with the first system being the master, the next system being considered the secondary, and the rest of the systems being considered spares. If the master system fails, the next system on the "preferenced" list becomes the master system for that file system and the first spare becomes the secondary. This means that the "preferenced" list must be updated whenever systems are added to or removed from the cluster, even during normal operation.

TruCluster offers a cluster file system that allows transparent access to any file system from any system in the cluster. However, all write operations, as well as all read operations on files smaller than 64K bytes, are done by the CFS server system upon request by the CFS client systems. Thus, TruCluster generally acts as a proxy file system using network I/O. The only exceptions are applications that have been modified to open the file with O_DIRECTIO. Oracle is the only application vendor that has taken advantage of this.

Veritas offers a cluster file system in two different products. The SANPoint Foundation Suite – High Availability (SPFS – HA) enhances VxFS with cluster file system extensions on HP-UX and Solaris, providing direct I/O to any volume from any system in the cluster that is physically connected to the volume. SPFS requires that any volume managed this way be physically connected to every system in the cluster. This offers direct I/O functionality for the general case of file systems with flat files. For Oracle 9i/10g RAC, the Veritas Database Edition/Advanced Cluster (DBE/AC) for 9i/10g RAC supports direct access I/O to the underlying VxFS. Note that if you do not use either SPFS – HA or DBE/AC, the Veritas Volume Manager defines a volume group as a "cluster disk group" (a special case of a "dynamic disk group"); this is the only type of volume that can be moved from one system in a cluster to another during failover. This is not a cluster file system, since Veritas emphasizes that only one system in a cluster can make use of the cluster disk group at a time.

All of the above systems that implement direct access I/O use a "master" system to perform meta-data operations. Therefore, operations like file creations, deletions, renames, and extensions are performed by one of the systems in the cluster, but all I/O inside the file or raw device can be performed by any of the systems in the cluster using direct access I/O. OpenVMS Cluster Software and PolyServe have multiple "master" systems to optimize throughput and reduce contention.

An advantage of direct access I/O, whether implemented with a file system or with raw devices, is that it allows applications to be executed on any system in the cluster without having to worry about whether the resources they need are available on a specific system. For example, batch jobs can be dynamically load balanced across all of the systems in the cluster, and are more quickly restarted on a surviving system if they were running on a system that becomes unavailable. Notice that the availability of the resources does not address any of the recovery requirements of the application, which must be handled in the application design.

Every operating system has a lock manager for files in a non-clustered environment. A distributed lock manager simply takes this concept and applies it between and among systems. There is always a difference in performance and latency between local locks and remote locks (often up to an order of magnitude difference (10x)), which may affect overall performance. You must take this into account during application development and system management.

HACMP offers the Cluster Lock Manager, which provides a separate set of APIs for locking, in addition to the standard set of UNIX System V APIs. All locking is strictly the responsibility of the application. The Cluster Lock Manager is not supported on AIX with the 64-bit kernel. HACMP also offers the General Parallel File System, which was originally written for the High Performance Technical Computing (HPTC) environment but is now available in the commercial space.

MySQL Cluster tracks the locks for the database itself, but does not offer a generalized distributed locking mechanism.

NSK does not even have the concept of a distributed lock manager, as none is required. Ownership of all resources (files, disk volumes, and so forth) is local to a specific CPU, and all communication to any of those resources uses the standard messaging between CPUs and systems. The DAM responsible for a given volume keeps its locks and checkpoints this information to a backup DAM located on a different CPU. Because of the efficiencies of the messaging implementation, this scales superbly.

OpenVMS Cluster Software uses the same locking APIs for all locks, and makes no distinction between local locks and remote locks. In effect, all applications are automatically cluster-aware.

Oracle implements a distributed lock manager on HACMP, Linux LifeKeeper, SunClusters, and Windows 2000/2003, but takes advantage of the native distributed lock manager on OpenVMS Cluster Software, Serviceguard Extension for OPS/RAC, and TruCluster.

SQL Server 2000/2005 tracks the locks for the database itself, but, because only a single instance of the database can be running at one time, there is no distributed lock manager.

SunCluster and TruCluster extend the standard set of UNIX APIs for file locking in order to work with the cluster file system, resulting in a proxy for, and a layer on top of, the standard file systems. Keep in mind that even though the file system is available to all systems in the cluster, almost all I/O is performed by the master system, even on shared disks.

Veritas uses the Veritas Global Lock Manager (GLM) to coordinate access to the data on the cluster file system.

Windows 2000/2003 does not have a distributed lock manager.

Quorum

When discussing cluster configurations, it is important to understand the concept of quorum. Quorum devices (which can be disks or systems) are a way to break the tie when two systems are equally capable of forming a cluster and mounting all of the disks, but cannot communicate with each other. This is intended to prevent cluster partitioning, which is known as "split brain."

When a cluster is first configured, you assign each system a certain number of votes (generally 1). Each cluster environment defines a value for the number of "expected votes" for optimal performance. This is almost always the number of systems in the cluster. From there, we can calculate the "required quorum" value, which is the number of votes that are required in order to form a cluster. If the actual quorum value is below the required quorum value, the software will refuse to form a cluster, and will generally refuse to run at all.

For example, assume there are two members of the cluster, system A and system B, each with one vote, so the required quorum of this cluster is 2.

In a running cluster, the number of expected votes is the sum of all of the members with which the connection manager can communicate. As long as the cluster interconnect is working, there are 2 systems available and no quorum disk, so the value is 2. Thus, actual quorum is greater than or equal to required quorum, resulting in a valid cluster.

When the cluster interconnect fails, the cluster is broken, and a cluster transition occurs.

The connection manager of system A cannot communicate with system B, so the actual number of votes becomes 1 for each of the systems. Applying the equation, actual quorum becomes 1, which is less than the number of required quorum required to form a cluster, so both systems stop and refuse to continue processing. This does not support the goal of high availability; however, it does protect the data, as follows.

Notice what would happen if both of the systems attempted to continue processing on their own. Because there is no communication between the systems, they both try to form a single system cluster, as follows:

1. System A decides to form a cluster, and mounts all of the cluster-wide disks.
2. System B also decides to form a cluster, and also mounts all of the cluster-wide disks. The cluster is now partitioned.
3. As a result, the common disks are mounted on two systems that cannot communicate with each other. This leads to instant disk corruption, as both systems try to create, delete, extend, and write to files without locking or cache coherency.

To avoid this, we use a quorum scheme, which usually involves a quorum device.

Picture the same configuration as before, but now we have added a quorum disk, which is physically cabled to both systems. Each of the systems has one vote, and the quorum disk has one vote. The connection manager of system A can communicate with system B and with the quorum disk, so expected votes is 3. This means that the quorum is 2. In this case, when the cluster interconnect fails, the following occurs:

1. Both systems attempt to form a cluster, but system A wins the race and accesses the quorum disk first. Because it cannot connect to system B, and the quorum disk watcher on system A observes that at this moment there is no remote I/O activity on the quorum disk, system A becomes the founding member of the cluster, and writes information, such as the system id of the founding member of the cluster and the time that the cluster was newly formed, to the quorum disk . System A then computes the votes of all of the cluster members (itself and the quorum disk, for a total of 2) and observes that it has sufficient votes to form a cluster. It does so, and then mounts all of the disks on the shared bus.
2. System B comes in second and accesses the quorum disk. Because it cannot connect to system A, it thinks it is the founding member of the cluster, so it checks this fact with the quorum disk, and discovers that system A is in fact the founding member of the cluster. But system B cannot communicate with system A, and as such, it cannot count either system A or the quorum disk's votes in its inventory. So system B then computes the votes of all of the cluster members (itself only

for a total of 1) and observes it does not have sufficient votes to form a cluster. Depending on other settings, it may or may not continue booting, but it does not attempt to form or join the cluster. There is no partitioning of the cluster.

In this way only one of the systems will mount the cluster-wide disks. If there are other systems in the cluster, the value of required quorum and expected quorum would be higher, but the same algorithms allow those systems that can communicate with the founding member of the cluster to join the cluster, and those systems that cannot communicate with the founding member of the cluster are excluded from the cluster.

This example uses a "quorum disk," but in reality any resource can be used to break the tie and arbitrate which systems get access to a given set of resources. Disks are the most common, frequently using SCSI reservations to arbitrate access to the disks. Server systems can also be used as tie-breakers, a scheme that is useful in geographically distributed clusters.

Cluster Configurations

The following table summarizes important configuration characteristics of cluster products.

	Max Systems In Cluster	**Cluster Interconnect**	**Quorum Device**
HACMP AIX, Linux	32	Network, Serial, Disk bus (SCSI, SSA) (p)	No
LifeKeeper Linux, Windows	16	Network, Serial (p)	Yes (Optional)
MySQL Cluster AIX, HP-UX, Linux, Solaris, Windows	64	Network	Yes
NonStop Kernel	255	ServerNet (a)	Regroup algorithm
OpenVMS Cluster Software	96	CI, Network, MC, Shared Memory (a)	Yes (Optional)
Oracle 9i/10g RAC Many O/S's	Dependent on the O/S	Dependent on the O/S	n/a
PolyServe Matrix Linux, Windows	16	Network	Yes (membership partitions)
Serviceguard HP-UX, Linux	16	Network, HyperFabric (HP-UX only)	Yes = 2, optional >2
SQL Server 2000/2005 Windows	Dependent on the O/S	Dependent on the O/S	n/a
SunCluster Solaris	8	Scalable Coherent Interface (SCI), 10/100/1000Enet (a)	Yes (Optional), recommended for each multiported disk set
TruCluster Tru64 UNIX	8 generally, 512 w/Alpha SC	100/1000Enet, QSW, Memory Channel (p)	Yes (Optional)
Veritas Cluster Server AIX, HP-UX, Linux, Solaris, Windows	32	Dependent on the O/S	Yes (using Volume Manager)
Windows 2000/2003 DataCenter	4/8	Network (p)	Yes

Figure 4 Cluster Configuration Characteristics

HACMP can have up to 32 systems or dynamic logical partitions (DLPARs, or soft partitions) in a system. Except for special environments like SP2, there is no high speed cluster interconnect, but serial cables and all Ethernet and SNA networks are supported as cluster interconnects. The cluster interconnect is strictly active/passive, and multiple channels cannot be combined for higher throughput. The disk busses (SCSI and SSA) are also supported as emergency interconnects if the network interconnect fails. Quorum is supported only for disk subsystems, not for computer systems.

LifeKeeper supports up to 16 systems in a cluster, connected by either the network or by serial cable. These are configured for failover only, and are therefore active/passive. Any of the systems can take over for any of the other systems. Quorum disks are supported but not required.

MySQL Cluster can have up to 64 systems in the cluster, connected with standard TCP/IP networking. These can be split among any combination of MySQL nodes, storage engine nodes, and management nodes. MySQL uses the management node as an arbitrator to implement the quorum scheme.

NonStop Kernel can have up to 255 systems in the cluster, but, given the way the systems interact, it is more accurate to say that NonStop Kernel can have 255 systems * 16 processors in each system = 4,080 processors in a cluster. Each system in the cluster is independent and maintains its own set of resources, but all systems in the cluster share a namespace for those resources, providing transparent access to those resources across the entire cluster, ignoring the physical location of the resources. This is one of the methods that NSK uses to achieve linear scalability. ServerNet is used as a communications path within each system as well as between relatively nearby S-series systems. ServerNet supports systems up to 15 kilometers and remote disks up to 40 kilometers, with standard networking supporting longer distances. The ServerNet-Fox gateway provides the cluster interconnect to the legacy K-series. The cluster interconnect is active/active. NSK uses a message-passing quorum scheme called Regroup to control access to resources within a system, and does not rely on a quorum disk.

OpenVMS Cluster Software supports up to 96 systems in a cluster, spread over multiple datacenters up to 500 miles apart. Each of these can also be any combination of VAX and Alpha systems, or (starting in 2005) any combination of Itanium and Alpha systems, in mixed architecture clusters. There are many cluster interconnects, ranging from standard networking, to Computer Interconnect (the first cluster interconnect ever available, which was introduced in 1984), to Memory Channel, and they are all available active/active. The quorum device can be a system, a disk, or a file on a disk, with the restriction that this volume cannot use host-based shadowing.

Oracle 9i/10g RAC uses the underlying operating system functionality for cluster configuration, interconnect, and quorum. The most common type is 100BaseT or 1000BaseT in a private LAN, often with port aggregation to achieve higher speeds. For low latency cluster interconnects, HP offers HyperFabric and Memory Channel, and Sun offers Scalable Cluster Interconnect. Oracle 9i/10g RAC does not use a quorum scheme; instead, it relies on the underlying operating system for this functionality.

PolyServe Matrix Server uses the underlying operating system functionality for cluster configuration and interconnect. PolyServe on both Linux and Windows primarily uses gigabit Ethernet or Infiniband in a private LAN. Matrix Server uses a quorum scheme of membership partitions, which contain the metadata and journaling for all of the file systems in the cluster. There are three membership partitions: all data is replicated to all of them, providing redundancy. One or even two of these partitions could fail, and PolyServe could still rebuild the information from the surviving membership partition. These membership partitions provide an alternate communications path, allowing servers to correctly arbitrate ownership and coordinate the file systems, even if the cluster interconnect fails. It is good practice to place the three membership partitions on three separate devices that are not the devices of the file systems themselves.

Serviceguard can have up to 16 systems, using standard networking or HyperFabric (HP-UX only) as a cluster interconnect, and uses Auto Port Aggregation for a high speed active/active cluster interconnect. A special requirement is that all cluster members must be present to initially form the cluster (100% quorum requirement), and that >50% of the cluster must be present in order to continue operation. Serviceguard can use either one or two quorum disks (two in an Extended Campus Cluster), a quorum server that is not a member of the cluster, or an arbitrator system that is a member

of the cluster. The quorum disk, quorum system, or arbitrator system is used as a tie breaker when there is an even number of production systems and a 50/50 split is possible. For two nodes, a quorum device is required: either a system (on HP-UX and Linux), a disk volume (on HP-UX), or a LUN (on Linux). Quorum devices are optional for any other size cluster. Cluster quorum disks are supported for clusters of 2-4 nodes, and cluster quorum systems are supported for clusters of 2-16 systems. Single quorum servers can service up to 50 separate Serviceguard clusters for either HP-UX or Linux. Note that quorum servers are not members of the clusters that they are protecting.

SQL Server 2000/2005 uses the underlying operating system functionality for cluster configuration, interconnect, and quorum. The maximum number of servers in an SQL Server environment grew from four with Windows 2000 to eight with Windows 2003.

SunCluster can have up to eight systems in a cluster, using standard networking or the Scalable Coherent Interconnect (SCI) as a cluster interconnect. SCI interfaces run at up to 1GByte/second, and up to four can be striped together to achieve higher throughput, to support active/active cluster interconnects. However, Sun only offers up to a four-port SCI switch, so only four systems can be in a single SunPlex domain. Quorum disks are recommended by Sun, and each multiported disk set requires its own quorum disk. So, for example, if there are four systems (A, B, C and D) with two multiported disk arrays (X and Y) where disk array X is connected to systems A and B, and disk array Y is connected to systems C and D, two quorum disks are required.

TruCluster can have up to eight systems of any size in a cluster. For the HPTC market, the Alpha System SuperComputer system farm can have up to 512 systems. This configuration uses the Quadrix Switch (QSW) as an extremely high-speed interconnect. For the commercial market, TruCluster uses either standard networking or Memory Channel as an active/passive cluster interconnect.

Both OpenVMS Cluster Software and TruCluster recommend the use of quorum disks for 2-node clusters, but make it optional for clusters with a larger number of nodes.

Veritas Cluster Server can have up to 32 systems in a cluster of AIX, HP-UX, Linux, Solaris, and Windows 2000/2003 systems. Standard networking from the underlying operating system is used as the cluster interconnect. Veritas has implemented a special Low Latency Transport (LLT) to efficiently use these interconnects without the high overhead of TCP/IP. Veritas implements a standard type of quorum in Volume Manager, using the term "coordinator disk" for quorum devices.

Windows 2000 clusters can have up to four systems in a cluster, while Windows 2003 extends this to eight. Keep in mind that Windows 2000/2003 DataCenter is a services sale, and only Microsoft qualified partners like HP can configure and deliver these clusters. The only other cluster interconnect available is standard LAN networking, which works as active/passive.

Application Support

The following table summarizes application support provided by the cluster products.

	Single-instance (failover mode)	**Multi-instance (cluster-wide)**	**Recovery Methods**
HACMP AIX, Linux	Yes	Yes (using special APIs)	Scripts
LifeKeeper Linux, Windows	Yes	No	Scripts
MySQL Cluster AIX, HP-UX, Linux, Solaris, Windows	No (MySQL Server)	Yes	Failover
NonStop Kernel	Yes (takeover)	Effectively Yes	Paired Processing
OpenVMS Cluster Software OpenVMS	Yes	Yes	Batch /RESTART
Oracle 9i/10g RAC Many O/S's	No (Oracle DB)	Yes	Transaction recovery

PolyServe Matrix Linux, Windows	Yes	No	Scripts
Serviceguard HP-UX, Linux	Yes	No	Packages and Scripts
SQL Server 2000/2005 Windows	Yes	No	Scripts
SunCluster Solaris	Yes	No	Resource Group Manager
TruCluster Tru64 UNIX	Yes	Yes	Cluster Application Availability
Veritas Cluster Server AIX, HP-UX, Linux, Solaris, Windows	Yes	No	Application Groups
Windows 2000/2003 Cluster Service Windows	Yes	No	Registration, cluster API

Figure 5 Cluster Support for Applications

Single-Instance and Multi-Instance Applications

With respect to clusters, there are two main types of applications: single-instance applications and multi-instance applications. Notice that these are the opposite of multisystem-view and single-system-view. Multisystem-view clusters allow single-instance applications, providing failover of applications for high availability, but don't allow the same application to work on the same data on the different systems in the cluster. Single-system-view clusters allow multi-instance applications, which provide failover for high availability, and also offer cooperative processing, where applications can interact with the same data and each other on different systems in the cluster.

A good way to determine if an application is single-instance or multi-instance is to run the application in several processes on a single system. If the applications do not interact in any way, and therefore run properly whether there is only one process running the application or multiple processes on a single system are running the application, then the application is single-instance.

An example of a single-instance application is telnet. Multiple systems in a cluster can offer telnet services, but the different telnet sessions themselves do not interact with the same data or each other in any way. If a system fails, the users on that system simply log in to the next system in the cluster and restart their sessions. This is simple failover. Many systems, including HACMP, Linux, Serviceguard, SunCluster, and Windows 2000/2003 clusters set up telnet services as single-instance applications in failover mode.

If, on the other hand, the applications running in the multiple processes interact properly with each other, such as by sharing cache or by locking data structures to allow proper coordination between the application instances, then the application is multi-instance.

An example of a multi-instance application is a cluster file system that allows the same set of disks to be offered as services to multiple systems. This requires a cluster file system with a single-system-view cluster, which can be offered either in the operating system software itself (as on OpenVMS Cluster Software) or by other clusterware (as on Oracle 9i/10g RAC). Although HACMP, LifeKeeper, Serviceguard, SunCluster, and Windows 2000/2003 do not support multi-instance applications as part of the base operating system cluster-ware, 3rd party tools can add multi-instance application capability. For example, NonStop Kernel uses messaging and DAMs to provide this functionality.

Applications, whether single-instance or multi-instance, can be dynamically assigned network addresses and names, so that the applications are not bound to any specific system address or name. At any given time, the application's network address is bound to one or more network interface cards (NICs) on the cluster. If the application is a single-instance application running on a single system,

the network packets are simply passed to the application. If the application is a single-instance application running on multiple systems in the cluster (like the telnet example above), or is a multi-instance application, the network packets are routed to the appropriate system in the cluster for that instance of the application, thus achieving load balancing. Future communications between that client and that instance of the application may either continue to be routed in this manner, or may be sent directly to the most appropriate NIC on that server.

Recovery Methods

The recovery method is the way the cluster recovers the applications that were running on a system that has been removed, either deliberately or by a system failure, from the cluster.

HACMP allows applications and the resources that they require (for example, disk volumes) to be placed into resource groups, which are created and deleted by running scripts specified by the application developer. These scripts are the only control that HACMP has over the applications, and IBM stresses that the scripts must take care of all aspects of correctly starting and stopping the application, otherwise recovery of the application may not occur. The resource groups can be concurrent (that is, the application runs on multiple systems of the cluster at once) or non-concurrent (that is, the application runs on a single system in the cluster, but can fail over to another system in the cluster). For each resource group, the system administrator must specify a "node-list" that defines the systems that are able to take over the application in the event of the failure of the system where it is currently running. The node-list specifies the "home node" (the preferred system for this application) and the list of other systems in the cluster for takeover of the application, in order of priority.

For multisystem-view clusters like Linux, recovery is done by scripts that are invoked when the heartbeat messages between the cluster members detects the failure of one of the systems. Two example scripts for Linux are mon and monit.

MySQL Cluster does not support applications as such, since it is focused only on the database. Any applications which connect to the database are external to MySQL Cluster, even if they are running on the same system as the database instance. Failover of connections from application nodes to database nodes is transparent to the external applications, and failover of external application connections to the application nodes is seamless between instances of the database.

For fault-tolerant systems like HP NonStop servers, recovery within a system is done by paired processes, where a backup process is in close synchronization with a primary process, ready to take over in the event of any failure. Data replication is the foundation for recovery between systems.

Oracle does an excellent job of recovering failed transactions. Oracle 9i/10g RAC offers this same type of recovery of failed transactions between systems in a cluster. The simple way to think about it is to understand the actions that a standalone Oracle database would perform if the single system went down unexpectedly and then came back up and began recovery. These actions are the same ones that Oracle 9i/10g RAC performs on the surviving systems in a cluster if one of the systems went down unexpectedly. This was easy for Oracle to implement, as the code was already thoroughly tested, and it is easy for database administrators to understand, as they are already familiar with the way Oracle performs these operations in standalone databases.

PolyServe Matrix HA uses application-specific scripts to perform failover and failback of applications. Failover can be from one named system to another (1:1), from the failed system to a group of systems (1:n), or from a group of systems to another group of systems (n:m).

Serviceguard has extensive tools for grouping applications and the resources needed to run them into up to 150 "packages" that are then managed as single units. A set of attributes are specified for each application package, including zero or more network identities, disk volumes, application services, and other resources required by the application. The system administrator can define procedures for recovering and restarting the applications on one of a prioritized list of systems in a cluster, in the event of server failure, and can also define whether to fail back when the primary node is available, or to operate in "rotating standby" mode.

SQL Server 2000/2005 does not support application failover as such, since it is focused only on the database. Any applications that connect to the database are external to SQL Server, even if they are running on the same system as the database instance, and application failover is handled by the underlying operating system. The connections to the database automatically fail over to the surviving server.

SunClusters has tools for grouping applications into either "failover resource groups" or "scalable resource groups." Failover resource groups perform recovery of single-instance applications running on a single server, while scalable resource groups perform recovery of single-instance applications that are running on multiple servers. Be aware that "scalable" applications are single-instance applications; the tools provide IP redirection, routing, and load balancing only.

Both OpenVMS Cluster and TruCluster multi-instance applications have built-in methods that enable recovery from failing systems. They can monitor some applications and recover them automatically. OpenVMS Cluster Software specifies recovery using the /RESTART qualifier on the batch SUBMIT command; TruCluster does it with the Cluster Application Availability facility.

Veritas has extensive tools for grouping applications and their required resources into "resource groups," and defines "agents" to manage those groups. There are "failover groups," which can run on only one system at a time, and "parallel groups," which can run on multiple systems at a time. Veritas has created agents for many standard applications, including DB2, Oracle, and SAP for the Veritas Cluster Server. Failover of resource groups is extremely flexible, and can be done by priority (the next server on the list is chosen), round robin (the server running the least number of resource groups is chosen), and load-based (where the actual load at the moment of failover imposes predefined limits on the number of resource groups on a system).

With Windows 2000/2003 Cluster Service, there are three main recovery methods:

- Generic application/generic service. This doesn't require development of any kind. There is a one-time registration of the application for protection by Windows 2000/2003. A wizard guides administrators through this process.
- Custom resource type. The application itself is unchanged, but the application vendor (or other party) develops a custom resource DLL that interfaces an application with Windows 2000/2003 Cluster Service to do application-specific monitoring and failover. Again, this doesn't require any development at integration time, but merely requires registering the application using the custom resource DLL.
- Cluster Service API. The application is modified to explicitly comprehend that it is running in a clustered environment and can perform cluster-related operations (failover, query nodes, and so forth).

Cluster Resilience

One of the major selling points of clusters is high availability. The cluster must be resilient, even when things go wrong, such as system or storage subsystem failures or peak workloads beyond expectations, and even when things go very wrong, such as physical disasters. The following table summarizes the cluster resilience characteristics of cluster products.

	Dynamic Partitions	**Disk High Availability**	**Disk Path High Availability**	**Cluster Network Alias**
HACMP AIX, Linux	DLPARs (AIX 5.2 and later)	RAID-1 (Logical Volume Manager)	Multipath I/O (active/passive)	Not shared
LifeKeeper Linux, Windows	No	Distributed Replicated Block Device (DRBD)	Multipath I/O (active/passive)	Not shared
MySQL Cluster AIX, HP-UX, Linux, Solaris, Windows	No	Dependent on the O/S	Dependent on the O/S	Not shared
NonStop Kernel	No	RAID-1, Process Pairs	Multipath I/O (passive)	Shared
OpenVMS Cluster Software OpenVMS	Galaxy	RAID-1 (Host Based Volume Shadowing)	Multipath I/O (passive)	Shared

Oracle 9i/10g RAC Many O/S's	No	Dependent on the O/S	Dependent on the O/S	Dependent on the O/S
PolyServe Matrix Linux, Windows	No	Dependent on the O/S	Dependent on the O/S	Dependent on the O/S
Serviceguard HP-UX, Linux	vPars	RAID-1 (MirrorDisk/UX)	Multipath I/O (active)	Not shared
SQL Server 2000/2005 Windows	No	Dependent on the O/S	Dependent on the O/S	Dependent on the O/S
SunCluster Solaris	Hot add/swap	RAID-1 (Solaris Volume Mgr)	Multipath I/O (passive)	Not shared
TruCluster Tru64 UNIX	No	RAID-1 (Logical Storage Manager)	Multipath I/O (active)	Shared
Veritas SPFS HP-UX, Solaris	No	RAID-1 (Veritas Volume Manager)	Multipath I/O (passive)	No (simulated by Traffic Director)
Windows 2000/2003 DataCenter Windows	No	No (RAID-1 NTFS is not supported in a cluster)	Multipath I/O (active/passive)	Not shared

Figure 6 Cluster Resilience Characteristics

Dynamic Partitions

Almost all of the high-end server systems offer hard partitions, which electrically isolate sections of a large SMP system from each other. But some operating environments also have the ability to dynamically move hardware components between these hard partitions without rebooting the running instance of the operating system. These are called soft, or dynamic partitions.

Dynamic partitions protect against peaks and valleys in your workload. Traditionally, you build a system with the CPUs and memory for the worst-case workload, accepting the fact that this extra hardware will be unused most of the time. In addition, with hard partitioning becoming more popular because of system consolidation, each hard partition requires enough CPUs and memory for the worst-case workload. But dynamic partitioning lets you share this extra hardware between partitions of a larger system. For example, you can allocate the majority of your CPUs to the on-line processing partition during the day, and move them to the batch partition at night. AIX 5.2 with DLPARs, HP-UX 11i with vPars, and OpenVMS V7.3-2 with Galaxy offer this functionality, as follows:

	Move CPUs	**Move I/O slots**	**Move memory**	**Share memory between partitions**
DLPARs AIX 5.2	Yes	Yes	Yes	No
Galaxy OpenVMS V7.3-2	Yes	No	No	Yes
vPars HP-UX 11i	Yes	No	No	No

Figure 7 Dynamic Partitioning

Do not confuse dynamic partitions with dynamic reconfiguration. Dynamic reconfiguration refers to the hot-add and hot-swap capabilities of the servers, where CPU boards, memory boards, and PCI boards can be added or removed and replaced without powering down or even rebooting the server. This requires cooperation with the operating systems, but it is not associated with clustering. The GS-

series of AlphaServers, the IBM pSeries, HP Integrity servers, HP NonStop servers, the SunFire 10K, 12K and 15K systems, and the Superdome systems all offer these capabilities, but they have nothing to do with dynamic partitioning. Sun calls this capability "Dynamic System Domains;" HP calls this "instant Capacity" (iCAP); and IBM call this "On Demand."

Both HP-UX and OpenVMS offer tools to dynamically balance CPUs across dynamic partitions. HP-UX Work Load Manager (WLM) works with the Process Resource Manager (PRM) to offer goal- based performance management of applications. CPUs can be moved between vPars in order to achieve balanced performance across the larger system. OpenVMS offers the Galaxy Balancer (GCU$BALANCER) utility, which can load balance CPU demand across multiple Galaxy instances.

Notice that all of the above software runs in the base operating system, not just in the clustering product. DLPARs, Galaxy, and vPars partitions can be clustered just as any other instances of the respective operating systems can be clustered. WLM and the Galaxy Balancer do not require the dynamic partitions to be clustered.

Disk and Disk Path High Availability

The storage subsystem level includes redundant host adapters, redundant paths from the host adapters to the storage controllers (through redundant switches if you are using FibreChannel), redundant storage controllers configured for automatic failover, and the appropriate RAID levels on the disks themselves. But some of this redundancy requires cooperation from the host operating system, specifically in the area of multipath I/O.

Multipath I/O allows the system to have multiple physical paths from the host to a specific volume, as when multiple host adapters are connected to redundant storage controllers. This is common with FibreChannel, but it is also achievable with SCSI and HP's Computer Interconnect (CI).

Support for multipath I/O can be either active or passive. With active multipath I/O, both paths are active at the same time, and the operating system can load balance the I/O requests between the multiple physical paths by choosing the host adapter that is least loaded for any given I/O operation. In the event of a path failure (caused by the failure of the host adapter, a switch, or a storage controller), the operating system simply reissues the I/O request to another path. This action is transparent to the application.

With passive multipath I/O, only one path is active at one time, but the other path is ready to take over if the first path fails. This is accomplished in the same way as the active multipath I/O, by having the system re-issue the I/O request.

Figure 6 shows whether the operating system supports active or passive multipath I/O. Many operating systems enable multipath I/O using EMC PowerPath, HP SecurePath, or Veritas Foundation Suite. PowerPath, SecurePath, and Veritas File System (using Dynamic Multipathing) allow both passive multipath I/O and static active multipath I/O, where the storage administrator can set the preferred and alternate paths from the host to the storage subsystem for a specific disk volume. For Linux, HP provides multipath I/O through open source FC HBA drivers, and the Linux MD driver for SCSI.

But if you have multiple simultaneous disk failures, such as by physical destruction of a storage cabinet, these technologies are not adequate.

The first level of defense against these types of failures is host-based RAID, which performs mirroring or shadowing across multiple storage cabinets.

AIX Logical Volume Manager offers RAID-1 (mirrored disks) with up to 3 copies of any disk, but does not offer multipath I/O. Network Interface Takeover allows the system administrator to configure multiple network adapters, where one or more is designated as a backup for the others (passive multipath), but this is not provided for storage interface devices. HP offers SecurePath on AIX.

HP-UX uses MirrorDisk/UX to maintain up to three copies of the data. The software for enabling active multipath I/O varies depending on the storage system being used by HP-UX. EMC PowerPath is included in the HP-UX kernel to give active multipath I/O to EMC storage arrays; the HP Logical Volume Manager (LVM) gives active multipath I/O to the EVA and XP storage subsystems.

Linux uses Distributed Replicated Block Device (DRBD), where one of the systems writes to a local disk and then sends an update over the network so that the other system can write a copy of that data to its local disk. HP offers SecurePath on Linux.

MySQL Cluster uses the underlying operating system functionality for disk access and does not offer any enhancements in this area. However, MySQL Cluster does allow simultaneous access to the database from multiple systems in the cluster, a scheme that is both highly scalable and highly available.

NonStop Kernel provides data high availability using a combination of RAID-1 (mirrored disks), passive multipath I/O with multiple ServerNet Fabrics, multiple controller paths and so forth, as well as process pair technology for the fault-tolerant Data Access Managers.

OpenVMS supports RAID-1 with Host-Based Volume Shadowing, which can maintain up to three copies of any disk, including the system disk. OpenVMS supports passive multipath I/O, with operator-controlled load balancing.

Oracle 9i/10g RAC is entirely dependent on the underlying operating system for all of these features, and does not implement any in the database or clustering software.

PolyServe Matrix Server does not offer RAID functionality, relying instead on the underlying operating system, and contends that dual redundant disks and multiple FibreChannel paths are not required to build high availability clusters. Matrix Server supports various multipath drivers, including PowerPath and SecurePath on Linux and Windows, and QLogic failover driver on Linux.

Solaris Volume Manager offers RAID 1+0, which can maintain up to three copies of any disk, including the system disk. Solaris supports passive multipath I/O with operator-controlled load balancing.

SQL Server 2000/2005 does not offer any capabilities in this area, relying instead on the underlying operating system.

Tru64 UNIX supports RAID-1 and RAID-5 with the Logical Storage Manager, which can protect any disk, including the system root. Up to 32 copies of a disk are supported. However, LSM does not support RAID-5 in a cluster, nor can an LSM volume contain the boot partitions of cluster members. LSM supports active multipath I/O.

Veritas supports RAID-1 and RAID-5 with the Veritas SANPoint Foundation Suite on HP-UX and Solaris. Dynamic Multipathing provides passive multipath I/O.

Windows 2000/2003 supports RAID-1 with NTFS mirroring on standalone systems but not in a cluster. HP offers SecurePath on Windows.

Accessibility of the mirrored volumes by the other systems in the cluster is the same as for any shared volume. AIX Logical Volume Manager, HP-UX MirrorDisk/UX, Linux DRBD, NSK RAID-1, Solaris Volume Manager, Veritas Volume Manager, and Windows 2000/2003 NTFS mirrors do not allow access to the remote copy of the data. OpenVMS Host Based Volume Shadowing and Tru64 UNIX Logical Storage Manager allow access to all systems in the cluster.

Cluster Network Alias

There are three types of cluster network alias:

- The Dynamic Name System (DNS) server has a single network name, which is the alias for the cluster. A list of "A" records specifies the network addresses of the actual systems in the cluster. The DNS server can "round robin" the requests among the list of network addresses, offering simple load balancing. However, in some cases the DNS server has no information about the utilization of a given system in the cluster, even whether the system is running. Requests may be sent to a system that is down at that moment. In other cases, one or more of the systems in the cluster dynamically update the list of "A" records in the DNS server to reflect system availability and load. This offers load balancing and higher availability, as requests will only be sent to systems that are running and able to do the work.
- A network address is bound to one of the network interface cards (NIC) on one of the systems in the cluster. All connections go to that NIC until that system exits the cluster, at which time the network address is bound to another NIC on another system in the cluster. This offers failover only, with no load balancing, but it does allow a client to connect to the cluster without needing to know which NIC on which system is available at that moment.
- A network address is bound to one of the NICs on one of the systems in the cluster. Again, all connections go to that NIC until that system exits the cluster, at which time the network

address is bound to another NIC on another system in the cluster. However, in this case, when a request is received, the redirection software on that system chooses the best system in the cluster to handle the request by taking into account the services offered by each system and the load on the systems at that moment, and then sends the request to the chosen system. This offers high availability and dynamic load balancing, and is more efficient than constantly updating the DNS server.

Transparent failover is important for highly available applications. However a connection-oriented application requires information about the state of the connection between the cluster and the client at the time the NIC or the system failed, which the system must be able to preserve.

Single-instance applications (which can only run on one system in the cluster at a time) have their network address bound to a single NIC, but allow failover of that network address to another NIC on a surviving system. This is indicated by "not shared" in Figure 6. Multi-instance applications can have their network address shared between multiple NICs, which is indicated by "shared" in the table.

MySQL Cluster, Oracle 9i/10g RAC, PolyServe Matrix Server and SQL Server rely on the underlying operating system functionality for network alias, and do not offer any enhancements in this area.

HACMP allows multiple IP addresses to be mapped to the same NIC, which provides cluster alias functionality by using IP Address Takeover (IPAT) to move the network address from a failed system to a surviving system. However, each of these IP addresses must be on their own subnet. (Note that this is the opposite of the Serviceguard requirement.) HACMP does not preserve the state of existing connections during a failover; the client must reconnect to the cluster alias, and the new incoming requests are distributed across the surviving systems.

LifeKeeper allows an additional IP address to be mapped to a NIC, and the failover of that address to a surviving system. LifeKeeper does not preserve the state of existing connections during a failover.

NonStop servers use the NonStop TCP/IPv6 to offer Ethernet failover. Multiple IP addresses can exist on the same Ethernet ServerNet Adapter, and they can be designated as either "non-shared IP" or "shared IP." Shared IP addresses gain scalability because NSK uses all available Ethernet ServerNet Adapters in the system for outbound traffic, which is, effectively, active multipath I/O. NSK does preserve the state of existing connections during a failover.

OpenVMS Cluster Software offers a cluster alias for DECnet (also known as DECnet Phase IV), DECnet-Plus (also known as DECnet Phase V), and TCP/IP. DECnet requires that one or more of the systems in the cluster be a routing node, but allows any or all of the systems in the cluster to accept incoming packets addressed to up to 64 cluster aliases. DECnet-Plus requires that there be an adjacent DECnet Phase V router somewhere on the network, but allows any or all of the systems in the cluster to accept incoming packets addressed to up to 64 cluster aliases. TCP/IP Services for OpenVMS allows the use of DNS load balancing (called DNS Clusters) with either static or dynamic load balancing. TCP/IP Services also offers failSAFE IP, which provides failover for NIC failures. None of these preserves the state of existing connections during a system failover.

Serviceguard allows multiple IP addresses to be mapped to the same NIC as a relocatable IP address. Multiple IP addresses can exist on the same NIC only if they are on the same subnet. (Note that this is the opposite of the HACMP requirement.) Up to 200 relocatable IP addresses can exist in a Serviceguard cluster. In the event of a NIC failure, HP-UX preserves the state of existing connections, but during a system failover, Serviceguard does not. On Linux, Serviceguard implements network redundancy by grouping two or more NICs together in a Linux process known as channel bonding, which can be configured in high availability mode or load balancing mode. In the high availability mode, one interface transmits and receives data while the others are available as backups. If one interface fails, another interface in the bonded group takes over. Load balancing mode allows all interfaces to transmit data in parallel in an active/active arrangement. In this case, high availability is also provided, because the bond still continues to function (with less throughput) if one of the component LANs fails. To achieve highly available network services, HP highly recommends channel bonding in each critical Internet Protocol (IP) subnet in order. Failover from one NIC to another prevents the package from failing over to another node and is transparent to the application.

SunClusters allows multiple IP addresses to be mapped to the same NIC for either "failover resource groups" or "scalable resource groups," discussed in Application Support. IP addresses that belong to failover resource groups are accessed through and run on a single system in the cluster and are

reassigned if that system fails. Scalable resource groups are accessed through and run on multiple systems in the cluster. Solaris does preserve the state of the existing connection in the event of a NIC failure, but SunCluster does not preserve the state of existing connections during a system failover.

TruCluster offers a cluster alias for TCP/IP using either host routing or network routing (only for virtual subnets). Any system in the cluster can register to receive incoming packets, which are then redirected to the correct system in the cluster, which is determined using weighted round-robin scheduling. A service port that is accessed through a cluster alias can be either "in_single" or "in_multi." "In_single" services are accessed through and run a single system in the cluster, which is reassigned if that system fails. "In_multi" services are accessed through and run on multiple systems in the cluster. The maximum number of cluster aliases is controlled by max_aliasid, with a default of 8 and a maximum value of 102,400, although this value may never be reached due to memory constraints. Tru64 UNIX preserves the state of existing connections in the event of a NIC failure by using NetRAIN, but TruCluster does not preserve the state of existing connections during a system failover.

Veritas Cluster Server does not offer a cluster alias as such, but uses a set of front-end systems running the Traffic Director package. This is equivalent to the F5 BigIP or Cisco ReDirector functionality, in that incoming requests are distributed to a large number of servers, using round robin, weighted round robin, least connections, and weighted least connections algorithms. The Traffic Director is designed to work with Cluster Server on any platform, but the Traffic Director itself runs only on Solaris systems.

Windows 2000/2003 Cluster Service allows multiple IP addresses to be mapped to the same NIC as a failover IP address. Windows does not allow the same network address to be assigned to NICs on different servers, and Windows does not preserve the state of existing connections during a system failover.

Disaster Tolerance

Disaster tolerance protects computer operations and data from site-wide disasters. For example, in Florida we worry about hurricanes, especially after the devastation of four major hurricanes in six weeks in 2004. In other areas of the world, we worry about tornadoes or blizzards. Everybody worries about power failures and fires. The only way to protect your valuable data from these types of disasters is to make sure that it is stored somewhere far away, and to keep it up to date as close to real-time as possible. There are many kinds of data replication, but the two major types are physical replication and logical replication.

Physical Replication

Physical replication can be performed either by the operating system or by the storage subsystem.

Some operating systems use the same software that they use for data high availability in the same computer room, except that the second (or, in some cases, the third) copy of the data is in another physical location. Serviceguard uses MirrorDisk/UX, NSK uses host-based replication to create a Nomadic Disk, SunCluster uses Solaris Volume Manager, and OpenVMS uses Host Based Volume Shadowing for OpenVMS. In other cases, the systems use different software than they use for local copies; for example, HACMP uses GeoRM.

Distance limitations are based on storage interconnect physical limits, which for FibreChannel is usually about 100km.

Configuration is the same as with single-room clusters because the connections to the disk systems use standard FibreChannel (with either long-wave GBICs, or FibreChannel over IP (FCIP)). The exceptions to this are HAGEO, which is limited to eight systems in two locations, and SunClusters which is limited to two systems in two locations, with the option of a quorum disk in a third location, separated by no more than 200 kilometers.

By doing the replication over FibreChannel instead of using networking between the servers, you can replicate the information from the local site to the remote site even if some of the systems in the remote site are not working.

Storage subsystems also perform replication, using either HP Continuous Access or the EMC Symmetrix Remote Datacenter Facility (SRDF). Connecting the FibreChannel switches together, exactly as in the operating system replication described above, allows the storage controllers to perform the replication. The advantage of this method is that the host does not have to be aware that

the replication is occurring, which means that any environment can use this, no matter which operating system, or even mix of operating systems, is using the storage system. Another advantage is that there is no load on the computer systems for this replication because it is all being done in the storage controllers. Host-based mirroring requires two or more host I/O operations for every write: one to each volume in the mirror-set. Controller-based mirroring requires only one host I/O operation for every write. The controller takes care of replicating the write operation to all of the local or remote volumes in the mirror-set. The disadvantage is that the target volume of the replication is inaccessible to any host while replication is occurring. To get around this restriction, have the remote site periodically take "snapshots" of the target volume, and then mount the snapshot volume. This gives the remote systems full access to data that is very close to up-to-date, which is sufficient for many purposes.

Failover requires you to explicitly stop the replication process in the surviving data center, and then explicitly mount the storage subsystems on the systems in the surviving data center to get back into production. Failback requires the same type of operation: you have to synchronize the data between the two storage subsystems, and then place one of them back into source mode and the other into target mode in order to restore the standard configuration.

Because of the complexity of sharing volumes by multiple systems in a multisystem image cluster, GeoRM, MirrorDisk/UX, Nomadic Disk, and Solaris Volume Manager do not allow access to the remote copy of the data that is mirrored by the operating system; similarly, HP Continuous Access and EMC SRDF do not allow access to the target volumes mirrored by the storage controllers.

Whether it is done by the host operating system or by the storage controller, physical replication offers bidirectional data replication.

Assume you have a production system in Boston MA, which is working fine. You want to safeguard your data, so you decide to build a disaster recovery site in Nashua NH, which is within 100km (60 miles) of Boston.

First, you put in a group of FibreChannel switches, and connect them using the FibreChannel to ATM adapters to the FibreChannel switches in your Boston data center. Then you put a duplicate set of storage in Nashua, and begin replicating the production system's data from Boston to Nashua. This is known as active/passive replication, because Nashua is simply a data sink: no processing is going on there, because there are no systems at that site.

However, you need processing to continue in Nashua even if your Boston data center is unavailable. So you put some systems in Nashua, and physically cable them to the storage. The target volume of the storage-based replication is not available for mounting by the systems, no matter what operating system they are running, so the Nashua site is idle, simply waiting for a signal to take over the operations from Boston. This signal can be automated or manual, but in either case, the storage subsystem would break the FibreChannel link, the systems would mount the volumes, and would then initiate any recovery mechanisms that have been defined for the application.

But your CFO strenuously objects to having a group of systems in Nashua just sitting idle, so you split your workload and give half of it to Boston and half of it to Nashua. Notice that this is a multisystem-view implementation, so the target volume of a physical copy is not available for mounting by the systems. So, just as you duplicated the Boston storage in Nashua, now you duplicate the Nashua storage in Boston, which you then connect to the systems in Boston as well, and you set up replication from Nashua to Boston.

Now you have your Boston production data being replicated to Nashua, and your Nashua production data being replicated to Boston. You could survive the loss of either data center, and have all of your data in the other one.

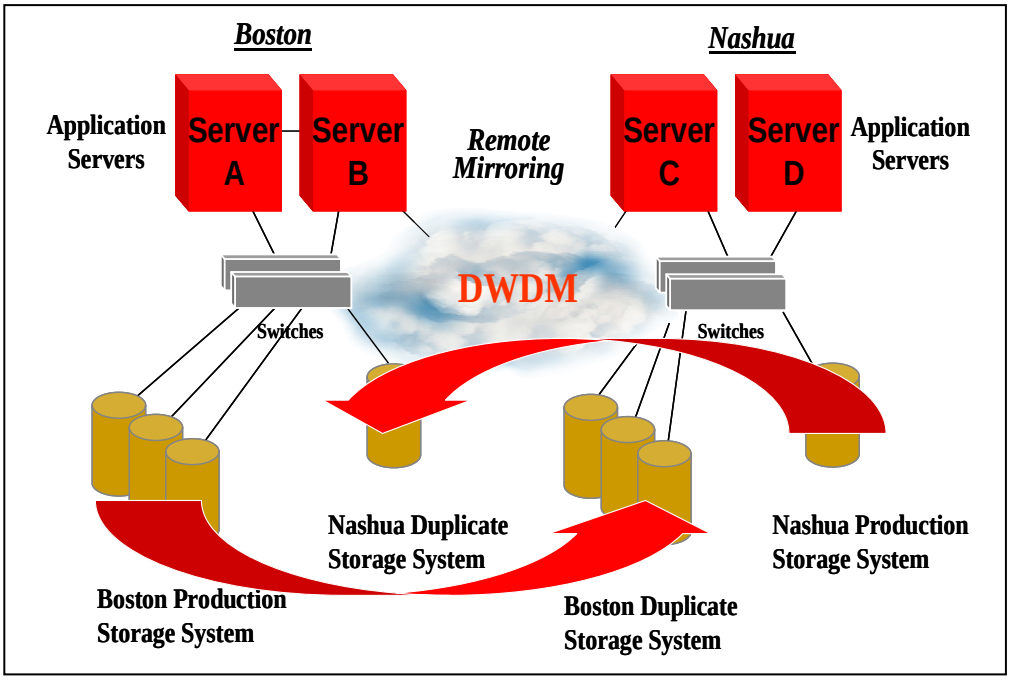

Figure 8 Remote Mirroring

This is known as active/active bidirectional data replication. Even though each set of storage is only being replicated in one direction, your business has data being replicated across the enterprise

Notice that with systems-based data high availability software, or StorageWorks Continuous Access, or EMC SRDF, the data being replicated is actually being written multiple times, by the system or by the storage controller. All of the data on the source disk is being written to the target disk, mission-critical database files as well as temporary files in a scratch area. Careful planning is required in order to ensure that you are replicating everything you need (such as the startup scripts for the database application, which exist outside the database files and are probably not on the same disk volumes as the database files), but are not replicating too much (such as /tmp or paging files).

This is how disaster tolerance works when it is being done by the storage subsystem, or by a multisystem image clustering environment. But some environments have clustered file systems. HAGEO allows raw devices to be accessed on both the local and remote systems, and OpenVMS Host Based Volume Shadowing (HBVS) allows full read/write access to all systems in either site. In these environments, you don't need to have separate storage subsystems that are effectively idle at the remote site; all of your volumes can be in active use. In addition, all of your servers can be used to process all of the information: reads tend to be done by the local disks so they are efficient, and writes are propagated to the other site by the host-based replication technology used in a single computer room.

Another advantage of this solution is that, because OpenVMS HBVS allows three copies of the replicated data, there could be a third site, which would allow continued disaster tolerance for the remaining two sites even if one of the sites was inaccessible.

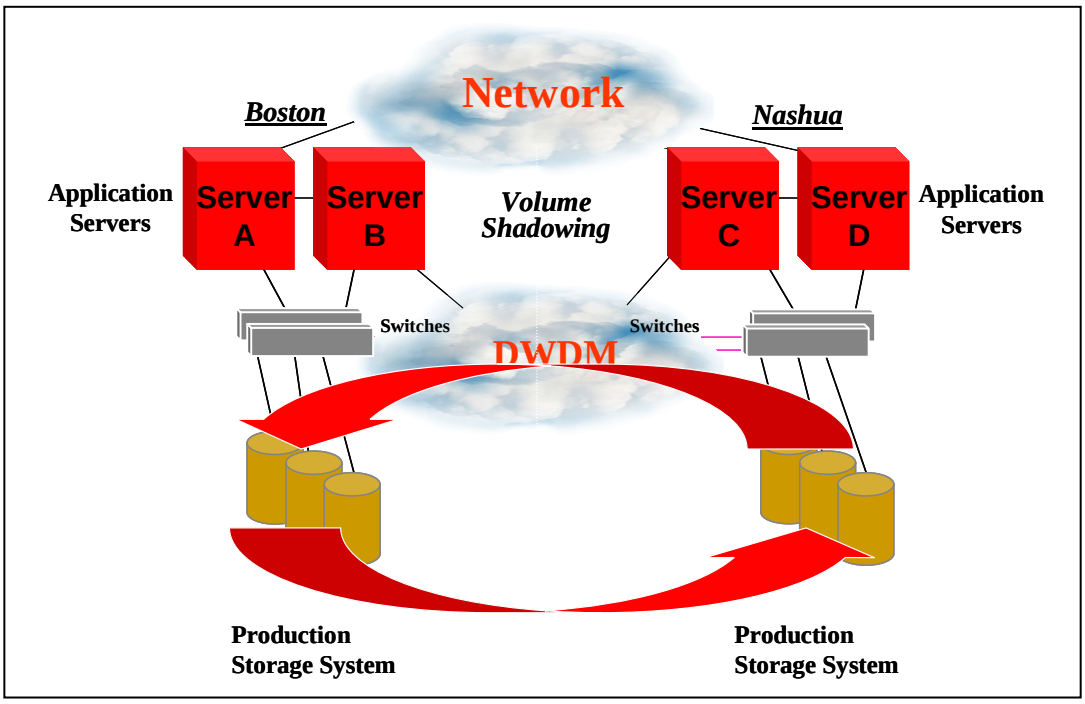

Figure 9 Volume Shadowing

Logical Replication

Logical replication differs from physical replication in both what is being replicated and how the replication is being done.

Logical replication ignores the disk volumes and replicates the transactions themselves. In the same way that physical replication takes a single write operation and applies it to multiple disk volumes, logical replication takes a single update transaction and applies it to multiple databases. The communications can be done over standard networking, and the systems that operate the multiple databases may or may not be clustered, depending on the database software chosen.

Recall the example data center in Boston. As before, you need a disaster recovery site. By using logical replication, you can put that disaster recovery site anywhere you want, because you are using standard networking technology. So you choose Los Angeles for your disaster recovery site.

You put a duplicate of your data center in Los Angeles, and then connect them with a fast networking link. Notice that this is a fully functional duplication of the data center, as it requires both systems and storage. However, it is not required to be a physical duplication: if you only require that some data is replicated, or if you can accept some lesser performance when a failover occurs, you can have fewer computing resources in Los Angeles than in Boston. Then you use logical replication to replicate the database records.

You can either leave it like this, in active/standby mode, or you can have the New York site run some type of production. Notice that it has to be different from the production run in Boston, because the two systems or clusters cannot access the same data. However, you can have bidirectional logical replication, just as you can have bidirectional physical replication. Both data protection and failover capabilities are provided from New York to Los Angeles. This is an active/active logical replication scheme. As in the previous example, failover and failback are semi-automated processes, with some human intervention required.

It is important to know what is being replicated. In the example of physical replication, the storage subsystem takes the bits that were written to the disk and copies them across to the other side. The advantage of this is that everything is being replicated: database files, scripts, flat files, everything. But this approach has some disadvantages as well.

- Everything is being replicated, which might require extremely high bandwidth in order to replicate data that may not needed at the remote site, such as log files.

- A simple operator error (like "rm –r " or "DELETE [...]*.*;*") will be replicated to the target disk in real time.
- Because the storage system is not aware of the database transactions, the information is not guaranteed to be transactionally consistent.

It is impossible to ensure atomicity of a database transaction with physical replication. Physical replication only duplicates the disk blocks that have changed, without regard to the files to which those blocks belong or whether the disk block that was changed was a database row or a database index. For example, a database row may be replicated, but the communications path may fail before the database index information can be replicated, or vice versa. Also, if a transaction spans files or volumes, some parts of the transaction may not be replicated in the proper sequence. If the systems that originated the transaction fail, or if the communications link fails at that moment, the replicated database will probably be corrupted and must be recovered by the database manager before the backup system can begin processing.

Logical replication, however, replicates database transactions, not volumes. It does not replicate security information, patches to the operating system or applications, scripts, or any of the myriad other files needed for maintaining the site; these have to be replicated and maintained by hand or by whole file replication. But the advantage is that an operator error that works on files cannot destroy your entire environment. Furthermore, the logical replication technology is aware of the database transaction as an atomic entity, and so can ensure that the entire transaction is either fully committed or fully rolled back.

Logical replication is done using various technologies. HP Reliable Transaction Router (RTR), HP NonStop Remote Database Facility (RDF) for the NonStop Kernel, IBM MQSeries, and Oracle DataGuard for both Oracle 9i/10g and Oracle Rdb are all examples of logical replication software. Also, many application server suites such as BEA WebLogics, IBM WebSphere, and Microsoft Windows iAS can replicate transactions across multiple sites.

Logical replication has three advantages over physical replication:

- The ability in some cases to read the database at the remote site
- Flexibility of the configurations
- Higher distance capabilities

If the remote database accepts the updates in their original form, as transactions, there is no requirement that the two databases be identical, and because the remote database simply accepts transactions, it can be used at the remote site. In most cases the remote database would be read-only, which is a "best practice" for MQSeries and RTR instead of a requirement, whereas it is a requirement with Oracle DataGuard using logical replication.

If, on the other hand, the remote database accepts updates as transaction logs (also known as "log-shipping" or "log-mining"), the two databases must be identical and the remote database cannot be updated (or even read, depending on the database software) by the remote site. This is the way that RDF and DataGuard work. RDF replicates individual tables, files, or the entire database, and it supports read access to the remote database. Oracle DataGuard using physical replication replicates the entire database, but it does not allow any access to the remote database.

Logical replication can replicate the transaction to multiple sites by duplicating the network packets that carry the update information and sending them to multiple geographically diverse sites. By replicating the transactions, a single database can act as the replication target for multiple other databases. 1:1, 1:n, and n:1 configurations are easily achievable using logical replication technology.

Logical replication usually operates in asynchronous mode, which does not require the remote site to complete the update before continuing the operation at the primary site. Therefore, there is no distance limitation between logical replication sites. However, you should consider the possibility of "in-flight" transactions being lost in the event of a failure of either the primary system or the communications path.

There a distance limitation for physical replication but not for logical replication because physical replication requires a great deal of two-way communication between the sites, and because of the limitation of the speed of light.

In a vacuum, light travels at 186,282 miles per second. Round that off to 200,000 miles per second (to make the math easier), or 200 miles per millisecond. We require confirmation of any message, so we must use round-trip distances. Therefore, light can travel up to 100 miles away and back in 1 millisecond. But light travels somewhat slower in fibre than in a vacuum, and there are the inevitable switch delays, so the conservative rule of thumb is that it takes 1 millisecond for every 50-mile round trip. Therefore, 500 miles adds 10 milliseconds to the latency of a disk access. Given normal disk access latency of 10-15 milliseconds, this merely doubles the latency -- the shadowing software can cope with that. But if the latency is more than that, the software might think the disk at the other end has gone off-line and will incorrectly break the shadow set. If we increase the timeout feature to get around this, the software will think the disk is just being slow when it really is inaccessible, and we will have unacceptable delays in breaking the shadow set when it needs to be broken.

In both the physical and logical asynchronous replication scenarios, each volume on one side is merely sending data to the other side, and eventually getting an acknowledgement back. The acknowledgement is not time-critical, and you can keep sending new data while waiting for the acknowledgement of the data you have already sent. The other side cannot write data to the target disk, so conflict resolution is not necessary. The additional latency of the speed of light is not as important, so the distance limitations are almost nonexistent as long as you don't exceed the distance limitations of your interconnect technology.

If the replication is done synchronously, the write at the local site and the remote sites must be performed before the operation is considered complete. This means that the latency of the transmission is added to the normal disk latency, which slows down each transaction and limits the number of transactions that can be completed in a given period of time. For example, 100km of distance adds slightly over one millisecond to the time needed to complete a write operation, as we discussed before (one millisecond per 50 miles, and 100km is about 62 miles). The advantage of this is that the information is guaranteed to be replicated to the remote site if the operation completed successfully. If the remote site is unable to perform the operation (that is, if a communications failure occurs), the local site can roll back the operation, and the two sites are still synchronized. The disadvantage of this is that each operation will necessarily take longer to perform.

If the replication is done asynchronously, the write at the remote site is queued but is not necessarily performed when the operation is considered complete. The advantage of this is that the local site can continue processing without having to wait the potentially long time until the remote site performs the operation, and eventually the remote site will catch up and be synchronized with the local site. The disadvantage of this is that if the local site becomes inaccessible or physically destroyed before the remote site has received those write operations, they may be lost with no way to recover them. The users at the local site received confirmation that their operation succeeded (and it did, at the local site), but the remote site never got some number of those transactions. This type of problem can be very difficult to resolve.

NonStop Kernel is pioneering a new approach which guarantees no lost transactions while not suffering the latency of synchronous transactions. The transaction logs, also known as audit trails, of the transactions which are being performed at the local site are synchronously replicated at the remote site by using disk mirroring. The primary site can perform all of the transactions at full speed without having to wait for the transactions to be performed by the remote system. If the primary site fails, the remote site can reexecute all of the transactions that took place at the primary site, because it has an up-to-the-last-transaction copy of the transaction log. This technique can be used by any system that has disk mirroring (which may be done by either the host or the storage subsystem) and a transaction log.

Because disk mirroring must be done synchronously in order to guarantee that the transaction log at the remote site has all of the transactions performed by the primary site, the distance is usually limited to about 100 km, but the precise distance varies depending on the replication technology being used.

Disaster Tolerance Features

The following table summarizes the disaster tolerant characteristics of cluster products. Unlike most of the other tables in this paper, this table has multiple options for each operating system, reflecting the multiple technologies that are available as options for each operating system.

Logical replication is done by software outside of the operating system; therefore, all systems are capable of it. Logical replication capabilities are not included in the following chart.

	Data Replication Method/Mode	Link Type and Distance	Cluster Topology and Size
HACMP/GeoRM AIX	Host based physical, sync/async, target is inaccessible	Dark fibre, 103km. Any networking, unlimited distance	Single cluster in 2 sites, 8 systems -or- not clustered
HAGEO AIX	Host based physical, sync/async, target can be read/write	Dark fibre, 103km. Any networking, unlimited distance	Single cluster in 2 sites, 8 systems
LifeKeeper Linux, Windows	Host based physical, sync/async, target is inaccessible	Any networking, unlimited distance	Single cluster in 2 sites, 8 systems
MySQL Cluster AIX, HP-UX, Linux, Solaris, Windows	Host based logical, sync only, target is inaccessible	Any networking, LAN distances	Not clustered
NonStop Kernel, RDF	Host based logical, async only, target is read only	Any networking, unlimited distance	Single cluster in 255 sites, 255 systems
NonStop Kernel, Nomadic Disk	Host based physical, sync only, target is inaccessible	ServerNet, 40km	Single cluster in 2 sites, 255 systems
OpenVMS Cluster Software OpenVMS	Host based physical, sync/async, target is read/write	Dark fibre, 100km. Any networking, 800km	Single cluster in 3 sites, 96 systems
Oracle Data Guard Many O/S's	Host based logical, sync/async, target is read-only	Any networking, unlimited distance	Dependent on the O/S
PolyServe Matrix Linux, Windows	n/a	n/a	Single cluster in 2 sites, 16 systems
Serviceguard Extended Campus Cluster HP-UX	Host based physical, sync only, target is inaccessible	Dark fibre, 100km	Single cluster in 2 sites, 4 systems -or- Single cluster in 3 sites, 16 systems
Serviceguard Metro Cluster HP-UX	StorageWorks Continuous Access, sync -or- EMC SRDF, sync/async	Dark fibre, 100km	Single cluster in 3 sites, 16 systems (3rd site is for arbitrator system only)
Serviceguard Continental Cluster HP-UX	StorageWorks Continuous Access, sync -or- EMC SRDF, sync/async -or- 3rd party host based replication	Dependent on replication technology, effectively unlimited	2 clusters in 2 sites, 16 systems in each site
Serviceguard for Linux with Cluster Extension XP	StorageWorks Continuous Access, sync	Dark fibre, 100km	Single cluster in 3 sites, 16 systems (3rd site is for arbitrator system only)
SQL Server 2000/2005 Windows	Host based logical, sync/async, target is read-only	Any networking, unlimited distance	2 clusters in 2 sites, number of systems is dependent on the O/S
SunCluster Enterprise Continuity Solaris	Host based physical, sync/async, target is read-only	Dark fibre, 200km	Single cluster in 3 sites, 2 systems (3rd site is for quorum

			device only)
TruCluster Tru64 UNIX	Host based physical, sync only, target is read/write	Dark fibre, 6km	Single cluster in 2 sites
Veritas Cluster Server AIX, HP-UX, Linux, Solaris, Windows	Host based physical, sync/async, target is read-only (snapshot)	Dark fibre, 100km. Any networking, unlimited distance	64 clusters in 8 sites, each with 32 systems
Windows 2000/2003 DataCenter Windows	StorageWorks Continuous Access, sync only	Dark fibre, 100km	2 clusters in 2 sites

Figure 10 Disaster Tolerance Characteristics

HP-UX Metro Cluster and Windows 2000/2003 DataCenter support only storage-based physical replication. All of the other products offer host-based storage replication and automated failover/failback, but they vary in how this is implemented.

"Dark fibre" in Figure 10 refers to the FibreChannel links that must be dedicated to this operation, and not shared with any other cluster or other task.

AIX uses host-based Geographical Remote Mirroring (GeoRM) to replicate raw devices, logical volumes, or file systems across 103km using FibreChannel, or across any distance using standard networking. The target of the replication (called the GeoMirror remote site) is not accessible by systems at the remote site. GeoMirror devices can operate in synchronous mode (write the local volume and then write the remote volume, but wait until both writes are complete), "Mirror Write Complete" mode (write the local volume and the remote volume at the same time, but wait until both writes are complete), or asynchronous mode. GeoRM can operate between up to 8 AIX servers, which may but do not have to be in an HAGEO cluster.

HAGEO extends HACMP to a geographically distributed configuration. It uses GeoRM to replicate raw devices, logical volumes, or file systems, but can be set up in a cascading failover or concurrent access configuration using GPFS. This is a single logical cluster in multiple sites, so the same configuration rules for an HACMP cluster apply.

LifeKeeper for Linux and Windows uses SteelEye Disaster Recovery Solution to replicate the file systems across any distance using standard networking. The target of the replication is not accessible to systems at the remote site. This is a single logical cluster that is geographically distributed, so the same configuration rules for LifeKeeper clusters apply.

MySQL Cluster replicates the master database changes to all of the slave databases. These changes are tracked in the "binary log," which is enabled in the master database. The slave databases connect to the master database and are updated whenever a change is made to the master. The master database has no formal connection to the slave databases, and is not aware of how many slave databases exist, so this is not really a cluster. Note that the slave databases must start as perfect copies of the master database. This operates over any distance, although latency might be a factor for the slave databases.

NonStop Kernel uses the NonStop Remote Database Facility (NonStop RDF) software to stream transactions across any distance using standard networking. The target of the logical replication is another database, whose DAMs apply the updates. RDF can replicate the data to multiple sites, and supports 1:1, 1:n, and n:1 configurations. Each system is independent of the others, but they are all part of the loosely-coupled cluster, linked by Expand networking software. This allows the remote system to have read access to the logical standby databases.

In addition to the NonStop RDF for real-time transaction replication, NonStop AutoSYNC software will replicate and synchronize application environments automatically. It monitors user defined files on the source system and uses whole-file replication to update the file on the remote systems. While it can replicate any file type, it is most commonly used for batch files, source files, configuration files, and any flat file that needs to be coordinated between sites.

NonStop Kernel also uses standard host-based disk replication technology across up to 40km of ServerNet to create a Nomadic Disk. The target of the replication is not accessible to systems at the remote site, and it does not have to be in the same cluster. A "zero lost transactions" environment can be created by combining NonStop RDF with Nomadic Disks that contain the transaction logs, to create a physical replication backup to the logical replication.

OpenVMS Cluster Software extends the standard clustering technology to the remote sites, and uses Host Based Volume Shadowing (HBVS) to replicate the disk volumes at the remote sites. The target of the replication is fully accessible by all systems at all sites. This is a single logical cluster that is geographically distributed, so the same configuration rules for OpenVMS Cluster Software apply.

Oracle Data Guard can run with or without 9i/10g RAC, and replicates transactions to other databases at up to nine other sites, as follows:

- Transactions are replicated and copied simultaneously to all of the databases where they are executed directly. The databases are fully accessible at all sites, but conflict resolution may complicate the environment.
- Transactions are replicated to a set of databases that have on-disk structures that are identical to the primary database on a block-for-block basis by applying the journal logs of the primary database to the physical standby databases. These databases are not accessible to systems at the remote sites.
- Transactions are also replicated to a set of databases that contain the same logical information as the primary database but which may have a different set of on-disk structures and are not identical to the original on a block-for-block basis. This is done by "mining" the journal logs and transforming those transactions into SQL statements that are then executed on the logical standby databases. This provides read access to the logical standby databases to systems at the remote sites, so that they can be used for queries and reports while still providing disaster tolerance to the original database.

Note that in any case. only the database information is replicated, not all the other information required for full functionality at the remote site.

PolyServe Matrix Server offers stretch clusters for disaster tolerance by using storage based replication. PolyServe offers no remote replication capabilities.

Serviceguard for HP-UX offers three different approaches to disaster tolerance:

- Extended Campus Cluster uses either MirrorDisk/UX (100km) or Veritas Cluster Volume Manager (10km) to replicate the file systems across FibreChannel. The target of the replication is not accessible at the remote site. This is a single logical cluster, which can have up to four systems at two sites when using a lock (quorum) disk, or up to 16 systems at three sites with the quorum server acting as an arbitrator node. Note that the use of either Veritas Cluster Volume Manager or HP Shared Logical Volume Manager (SLVM) is limited to two systems, one in each site, in an Extended Campus Cluster.
- Metro Cluster uses storage-based replication instead of host-based replication. Whether HP StorageWorks Continuous Access or EMC Symmetrix Remote DataCenter Facility (SRDF) is used, the target of the replication Is not accessible at the remote site. This is a single logical cluster which can have up to 16 systems, but it requires a third site for the quorum servers acting as arbitrator nodes.
- Continental Clusters does not have a native replication technology, but can use any other replication technology such as HP StorageWorks Continuous Access, EMC SRDF, or Oracle DataGuard. The accessibility of the target of the replication, and the distance between sites, is dependent on the technology used for replication. Continental Clusters are two separate clusters, each with up to 16 systems. Unlike the other two methods, which offer automated or manual failover and failback, failover and failback in a Continental Cluster is strictly manual.

Serviceguard for Linux integrates with Cluster Extension XP by using storage based replication instead of host based replication with HP StorageWorks Continuous Access. Similar to Metrocluster, the target of the replication is not accessible at the remote site. This is a single logical cluster that can have up to 16 systems, but it requires a third site for the quorum server acting as an arbitrator node.

SQL Server 2000 has offered replication for many years, either replicating the transactions to a remote database or through log shipping. In both cases the remote database is read-only.

SQL Server 2005 adds database mirroring, which provides "instant standby" with zero data loss and failover times of less than three seconds. This also eliminates much of the scripting involved in transaction replication, requiring only a few SQL statements to set up the mirror, fail over the primary database to the mirror, and fail back the mirror to the primary database. The mirror database is read-only. SQL Server 2005 also adds the concept of the "witness," which is a third instance of SQL Server that acts as a quorum server to prevent split brain. Database mirroring is available in two safety levels: "full" (synchronous replication) and "off" (asynchronous replication). Clients are aware of both the principal and mirror servers, and if the session to the principal server is dropped, the client attempts to reconnect first to the principal and then to the mirror.

Both SQL Server 2000 and SQL Server 2005 are dependent on the underlying operating system for the number of systems in the environment, but each site is a separate cluster.

Sun Infrastructure Solution for Enterprise Continuity uses host-based Remote Mirror to replicate either file systems or raw devices across up to 200km using standard networking technology with the Nortel Networks OPTera 5200. The target of the replication is available at the remote site if it is a raw device, but not if it is a file system, which is the same rule as if the replication were being done inside a single computer room. This is a single logical cluster that is geographically distributed, but there is a further restriction: there can only be a single system at each site. This restriction effectively gives you the choice between high availability (using multiple systems at a single site) or disaster tolerance (using one system at each of two sites). A third site can be used for a quorum device.

TruCluster offers storage-based replication using the StorageWorks Continuous Access to replicate file systems across up to 6km using FibreChannel. The target of the replication is not available at the remote site during the replication, but the source volume can be part of an LSM volume. This is a single logical cluster that is geographically distributed, so all of the configuration rules for a TruCluster still apply.

Veritas offers multiple host-based replication products for multiple platforms:

- Veritas Storage Replicator for Windows offers physical replication across unlimited distances over standard networking. It offers 1:1, 1:n, or n:1 schemes, and operates asynchronously.
- Veritas Volume Manager offers physical replication across FibreChannel using its standard RAID-1 technology for the standard VxFS file system. This is available for AIX, HP-UX, Linux, Solaris, and Windows systems.
- Veritas Volume Replicator is an optional capability of Volume Manager, offering physical replication over standard networking. Volume Replicator can replicate up to 32 remote copies of any volume, either 1:1, 1:n, or n:1. This works either synchronously or asynchronously.

None of the Veritas products allow the target volume to be mounted and used directly by the remote systems, but Veritas offers FlashSnap, which allows the remote system to create a snapshot of the target volume that can then be mounted and used.

Windows 2000/2003 DataCenter does not offer host-based replication across any large distance, but depends on storage-based replication. As such, the target of the replication is not accessible at the remote site during replication. Each site is an independent cluster, and requires manual intervention for failover and failback.

Summary

Every operating system offers a high availability option, but their capabilities vary widely. Some systems offer 2 nodes with manual failover time measured in minutes, other systems offer 16 nodes with automated failover time measured in seconds, while still others offer hundreds or thousands of processing units with absolutely transparent recovery from failure. Each system knows how to protect itself in this increasingly insecure world: some systems do it by depending on FibreChannel replication, others do it by depending on a database to move the data around the country. Still others offer true active/active multi-site clusters over hundreds of miles.

Once the business requirements are determined through a continuity planning program, it is up to you to understand these technologies and select the right solution to satisfy these requirements. But you

must also make sure your senior management are aware of the capabilities and limitations of the technology you have chosen. You can correctly implement a 2-node, manual failover system with no disaster tolerance, but your management may assume it is an infinitely expandable fault tolerant system with unattended zero recovery time, even if the primary datacenter is completely and unexpectedly wiped out. This causes a problem that will only be discovered when the system behaves exactly as you implemented it, and management discovers it does not meet their expectations.

To prevent this problem, document exactly what your chosen solution will and won't do, and get full agreement from management. If they need it to do more, you will have to request more money. In any case, HP can provide the hardware, software, and services to help you design and implement a solution that meets both business and operational requirements.

For more information

On IBM HACMP

- http://www-1.ibm.com/servers/aix/products/ibmsw/high_avail_network/hacmp.html for general information on HACMP
- http://www-1.ibm.com/servers/eserver/pseries/library/hacmp_docs.html for the technical documentation on HACMP
    - Chapter 3, "Types of Cluster Resources" says that only raw devices offer direct access I/O
    - Administration and Troubleshooting Guide, Chapter 9, "Managing Shared LVM Components" says that applications must use the Cluster Lock Manager to arbitrate access to raw devices, and that HACMP does not perform this function
    - Planning and Installation Guide, Chapter 3, "Planning Cluster Networking" describes IP Address Takeover
- http://www-1.ibm.com/servers/eserver/pseries/hardware/whitepapers/dlpar.pdf for a white paper on dynamic LPARs
- http://www.almaden.ibm.com/StorageSystems/file_systems/GPFS/Fast02.pdf for a white paper on the General Parallel File System (GPFS), and http://publib.boulder.ibm.com/clresctr/windows/public/gpfsbooks.html for the documentation on GPFS on the various cluster types. Finally, http://www.redbooks.ibm.com/redbooks/SG246954.html for a specific discussion of implementing Oracle 9i RAC on GPFS
- http://www-1.ibm.com/servers/eserver/pseries/library/hacmp_hiavgeo.html for the technical documentation for HAGEO
- http://www-1.ibm.com/servers/eserver/pseries/library/hacmp_georm.html for the technical documentation for HACMP Geographic Remote Mirror (GeoRM)

On SteelEye LifeKeeper for Linux

- http://www.hp.com/linux for general information on HP servers and Linux
- http://h18000.www1.hp.com/solutions/enterprise/highavailability/linux/index.html for general information on HP servers and Linux and high availability solutions
- http://www.hp.com/hpinfo/newsroom/press/08aug02b.htm for information on Lustre
- http://h18022.www1.hp.com/solutions/enterprise/highavailability/linux/highperformance/index.html for specifications on LifeKeeper with ProLiant servers
- http://linux-ha.org , specifically "What Linux-HA Can Do Now", and "LAN Mirroring Technologies"
- http://technet.oracle.com/tech/linux/ for information on Oracle and Linux
- http://www.steeleye.com/products/linux/#2 and http://www.steeleye.com/pdf/literature/lkpr4linux.pdf for information on SteelEye LifeKeeper

- http://www.lifekeeper.com/pdf/literature/se_drs_architecture_overview_final.pdf for information on SteelEye LifeKeeper disaster tolerance capabilities
- http://www.kernel.org/software/mon/ for the Service Monitoring Daemon
- http://www.tildeslash.com/monit/ for information on the Monit Utility
- http://sourceforge.net/projects/ssic-linux for information on the Single System Image Linux project

On MySQL Cluster

- http://www.mysql.com/products/cluster/ for general information on MySQL cluster, along with pointers to white papers and FAQs with more detailed information
- http://dev.mysql.com/doc/mysql/en/Replication.html for information on replication

On HP NonStop Kernel

- http://h20223.www2.hp.com/nonstopcomputing/cache/76385-0-0-0-121.aspx for access to the NonStop Technical Library (NTL) product for full NSK information
- http://h20223.www2.hp.com/nonstopcomputing/cache/76385-0-0-0-121.aspx for details on NSK and high availability
- http://h71028.www7.hp.com/ERC/downloads/RDFSVDS[1].pdf for information on the Remote Database Facility (NonStop RDF)
- http://h20223.www2.hp.com/nonstopcomputing/cache/76385-0-0-0-121.aspx for information on the Parallel Library TCP/IP software for network failover

On HP OpenVMS Cluster Software

- http://h71000.www7.hp.com/ for general information on OpenVMS and OpenVMS Cluster Software
- http://h71000.www7.hp.com/openvms/products/clusters/index.html for information on OpenVMS Cluster Software V7.3-2
- http://h18000.www1.hp.com/info/SP2978/SP2978PF.PDF for OpenVMS Cluster Software V7.3 SPD
- http://h71000.www7.hp.com/doc/index.html for the documentation. Specifically:
  - http://h71000.www7.hp.com/doc/731FINAL/4477/4477PRO.HTML, OpenVMS Cluster Systems, Section 2.3.2 shows quorum algorithm, and Chapter 7, Setting Up and Managing Cluster Queues
  - http://h71000.www7.hp.com/doc/731FINAL/6318/6318PRO.HTML, Guidelines for OpenVMS Cluster Configurations, Chapter 8, Configuring OpenVMS Clusters for High Availability and Appendix D, Multi-Site OpenVMS Clusters
  - http://h71000.www7.hp.com/doc/731FINAL/5423/5423PRO.HTML, Volume Shadowing for OpenVMS, Section 1.5 discusses WAN based RAID-1 for disaster tolerance
  - http://h71000.www7.hp.com/doc/731FINAL/documentation/pdf/OVMS_731_galaxy_gd.pdf, OpenVMS Alpha Partitioning and Galaxy Guide, Appendix A discusses the Galaxy Balancer program
- For cluster alias, see
  - http://h71000.www7.hp.com/doc/73final/documentation/pdf/DECNET_OVMS_NET_MAN.PDF, section 1.2.5.2 for a description of DECnet Phase IV cluster alias
  - http://h71000.www7.hp.com/doc/73final/6499/6499pro_index.html, section 9.2 and section xxx for a description of DECnet-Plus cluster alias

    - http://h71000.www7.hp.com/doc/73final/6526/6526pro.HTML, Chapter 6 for a description of DNS load balancing and TCP/IP Load Broker
- http://h71000.www7.hp.com/openvms/whitepapers/Illuminata.pdf for a white paper written by Illuminata describing the disaster tolerant capabilities of the major clustering products, using OpenVMS Cluster Software as the benchmark

On Oracle 9i/10g Real Application Clusters

- http://otn.oracle.com/products/database/clustering/index.html for general 9i RAC information
- Oracle MetaLink note id # 183408.1 - Aug 2003 discusses the cluster file systems supported by 9i RAC (An Oracle MetaLink subscription is required to access the note)
- http://otn.oracle.com/products/database/clustering/RACWhitepapers.html for a series of technical white papers on 9i RAC
    - http://otn.oracle.com/products/database/clustering/pdf/Oracle9iRAC_BusinessWhitePaper.pdf is an excellent introduction
    - http://otn.oracle.com/products/database/clustering/pdf/rac_building_ha_rel2.pdf discusses the management of 9i RAC and some best practices
    - http://otn.oracle.com/deploy/performance/pdf/FederatedvsClustered.pdf covers a lot of the same topics as this paper, but from a database point of view
- http://otn.oracle.com/deploy/availability/htdocs/odg_overview.html for DataGuard, and defines the difference between physical copy (Redo Apply) and logical copy (SQL Apply)

On PolyServe Matrix Server and Matrix HA

- http://www.polyserve.com/products.html for general information on PolyServe products
- http://www.polyserve.com/products_literature.html to request white papers, case studies and other information on PolyServe products
- http://www.polyserve.com/pdf/mxs_datasheet.pdf, "File System Features" which states that direct I/O (which this white paper calls direct access I/O) as a mount option, and does not require application changes
- http://www.polyserve.com/pdf/matrixha_datasheet.pdf, "HA Features and Benefits" which discusses the replication engine
- http://www.polyserve.com/products_mslinux.html for Matrix Server on Linux
- http://www.polyserve.com/products_mswindows.html for Matrix Server on Windows
- http://www.polyserve.com/products_matrixha.html for Matrix HA

On HP Serviceguard for HP-UX and Linux

- http://www.hp.com/go/ha for general information on Serviceguard and high availability
- http://docs.hp.com/hpux/11i for the complete HP-UX 11i documentation set
- http://www.hp.com/products1/unix/operating/infolibrary/reports/2002Unix_report.pdf for the DH Brown 2002 UNIX Function Review report
- http://docs.hp.com/hpux/ha/ for Disaster Tolerant and Highly Available Cluster Technologies
- http://www.hp.com/products1/unix/highavailability/ar/mcserviceguard/infolibrary/index.html, Information Library
    - 5nines Architecture Overview
    - System Cluster Technologies and Disaster Tolerance
    - Data Protection

    - Process Resource Manager (PRM) and Work Load Manager (WLM)
- http://www.software.hp.com/cgi-bin/swdepot_parser.cgi/cgi/displayProductInfo.pl?productNumber=B2491BA for MirrorDisk/UX
- http://www.docs.hp.com/hpux/ha/index.html#Quorum%20Server for a discussion of the quorum server
- http://haweb.cup.hp.com/Support/Extended-SG-Clusters/Extended-SG-Clusters.pdf for a discussion of Serviceguard Extension for RAC (SGeRAC) with different volume managers – HP Internal Use Only
- http://docs.hp.com/hpux/onlinedocs/B3936-90065/B3936-90065.html, Managing MC/Serviceguard, "How The Network Manager Works" covers relocatable IP addresses
- http://docs.hp.com/hpux/onlinedocs/T1335-90018/T1335-90018.html, Installing and Managing HP-UX Virtual Partitions
- http://h18006.www1.hp.com/products/sanworks/secure-path/linux.html for SecurePath for Linux

On Sun Microsystems SunClusters 3.1

- http://wwws.sun.com/software/cluster/index.html for information on SunCluster
- http://wwws.sun.com/software/cluster/wp-clustereval/wp-clustereval.pdf for an IDC evaluation of SunCluster, which Sun paid for
- http://docs.sun.com/db/doc/816-3383/6m9lt7uuk?a=view#cacfchja, SunCluster 3.1 Administration and Application Development Guide,
    - "Quorum Configurations" for a description of quorum
    - "Multiported Disk Device Groups" for a description of disk sharing and failover
    - "Multihost Disks" which explicitly says that all I/O is performed by the master system, and that only OPS/RAC allows direct access I/O to cluster file systems on multiported disks
    - "Cluster File Systems" for management of the CFS
    - "Data Services" for shared IP addresses (aka, cluster alias)
- http://docs.sun.com/db/doc/816-3384/6m9lu6fid?a=view#cihcjcae, SunCluster 3.1 System Administration Guide, "How To Add A Cluster File System" for a description of the actions needed on each system in the cluster to create a CFS
- http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-5075-11.pdf, SunFire 15K/12K Dynamic Reconfiguration Utility, Chapter 2, "Introduction to DR on the SunFire 15K/12K", describes Dynamic System Domains and makes clear that this is for dynamic reconfiguration of failed components
- http://www.sun.com/solutions/infrastructure/continuity/index.html for information on SunCluster Enterprise Continuity

On HP TruCluster

- http://h30097.www3.hp.com/ for general information on Tru64 UNIX and TruCluster
- http://h18004.www1.hp.com/products/quickspecs/11444_div/11444_div.HTML for the QuickSpecs on TruCluster V5.1b
- http://h30097.www3.hp.com/docs/pub_page/cluster51B_list.html for the documentation. Specifically:
    - http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHGVETE/TITLE.HTM , Cluster Technical Overview
    - http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHGVETE/TITLE.HTM, Cluster Technical Overview, Section 2.2 and 3.0

- http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHGWETE/TITLE.HTM, Hardware Configuration, Section 1.3.1.4
- http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHGYETE/TITLE.HTM, Cluster Administration, Section 4.3, Calculating Cluster Quorum
- http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHH0ETE/TITLE.HTM, Highly Available Applications, Chapter 1, Cluster Applications

- http://h18004.www1.hp.com/products/quickspecs/10899_div/10899_div.HTML for the QuickSpecs for the Logical Storage Manager V5.1b
- http://www.hp.com/techservers/ for the AlphaServer SC home page
- http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARHGVETE/TITLE.HTM, Chapter 6 for a description of the cluster alias
- http://h30097.www3.hp.com/cluster/tru64_campus_clusters.html for a description of the disaster tolerant capabilities of TruCluster

On Veritas Cluster Server, Database Edition, Veritas Storage Foundation (which was known as the SANPoint Foundation Suite), Storage Replicator, Volume Manager and Global Cluster Manager

- http://www.veritas.com/van/articles/3245.html for a good overview of VCS
- http://www.veritas.com/products/listing/ProductDownloadList.jhtml;vrtsid=ES5UF1DWXTTUPQFIYCLCFEY?productId=clusterserver#whitepapers for white papers on Veritas Cluster Server, specifically the "Cluster Server Technical Overview" discusses the configurations and limits of VCS
- http://ftp.support.veritas.com/pub/support/products/ClusterServer_UNIX/252160.pdf, the Veritas Cluster Server 3.5 User's Guide
- http://eval.veritas.com/downloads/pro//gcm/gcm_wp_tech_overview.pdf for a technical overview of Veritas Global Cluster Manager
- http://eval.veritas.com/downloads/pro/DHBrown_Report.pdf discusses the SANPoint Foundation Suite, and compares it to SunClusters Cluster File System
- http://eval.veritas.com/downloads/pro/sp_fdn_suite_ha/spfs_datasheet_pdf.pdf and http://www.veritas.com/van/products/sanpointfoundationsuite.html for a full description of SPFS - HA
- http://eval.veritas.com/downloads/pro/db_edition/dbed_ac_ds.pdf for an overview of Veritas Database Edition/Advanced Cluster for Oracle 9i RAC. Also, http://eval.veritas.com/downloads/pro/vm-tech_review_guide_050803.pdf describes the differences and restrictions between cluster disk groups and the cluster file system for 9i RAC, and specifically says that Dynamic Multipathing is passive
- http://eval.veritas.com/downloads/pro/vsr/vsr_ds.pdf describes Storage Replicator for Windows
- http://eval.veritas.com/downloads/pro/vcs/vcs_td_datasheet_0802.pdf describes Veritas Traffic Director

On Microsoft SQL Server 2000/2005 Enterprise Edition

- http://www.microsoft.com/sql/techinfo/default.asp for general SQL Server 2000 information, and pointers to white papers and documentation
- http://www.microsoft.com/sql/techinfo/administration/2000/availability.asp for specific high availability features of SQL Server 2000
- http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/failclus.mspx for details on SQL Server 2000 failover clustering
- http://msdn.microsoft.com/library/en-us/replsql/replover_694n.asp for details on SQL Server 2000 replication
- http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/logship1.mspx and http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/logship2.mspx for a complete

description of log-shipping in SQL Server 2000

On Microsoft Windows 2000/2003 DataCenter

- http://www.microsoft.com/windows2000/en/datacenter for general Windows 2000 DataCenter information
- http://www.microsoft.com/windowsserver2003/technologies/clustering/default.mspx for information on Microsoft clustering options
- http://www.microsoft.com/windowsserver2003/default.mspx for general Windows 2003 information
- http://www.microsoft.com/windowsserver2003/evaluation/features/highlights.mspx#cluster for Windows 2003 Cluster Service information
- http://www.microsoft.com/windows2000/en/datacenter/help/ and http://www.microsoft.com/windowsserver2003/proddoc/default.mspx for the documentation. Specifically:
  - Choosing a Cluster Model, emphasizes that Windows 2000 DataCenter is a multisystem-view cluster, and the lack of single-system-view capabilities.
  - Checklist: Preparing a Server Cluster, states that each system in the cluster must have its own system disk.
  - Server Clusters says up to 4 systems with Windows 2000, and 8 systems with 2003
  - Cluster Hardware and Drivers says the network is the only cluster interconnect
  - Quorum Disk describes the use of the quorum disk, and Cluster Database discusses how the cluster database from each system is written to the recovery log on the quorum disk
  - Windows Server Clusters, How To…, Perform Advanced Administrative Tasks
  - Choosing a RAID Method
  - Disaster Protection discussing the lack of WAN RAID

On HP StorageWorks Continuous Access

- http://h18006.www1.hp.com/storage/software.html for general information on StorageWorks high availability solutions
- http://h18000.www1.hp.com/products/quickspecs/10281_na/10281_na.html, QuickSpecs for StorageWorks Continuous Access
- http://h18006.www1.hp.com/products/storage/software/conaccesseva/index.html, SANworks Continuous Access Overview and Features

Books

- "Clusters for High Availability", Peter Weygant, ISBN 0-13-089355-2
- "In Search of Clusters", Gregory F. Pfister, ISBN 0-13-899709-8

Acknowledgements

Acknowledgements to Wendy Bartlett, Kirk Bresniker, Dan Cox, Raymond Freppel, Jon Harms, Ron LaPedis, Scott Lerner, Kerry Main, Keith Parris, Bob Sauers, Wesley Sawyer and Chuck Walters for their invaluable input and review.

Revision History

V0.7, 1995 to 2001 – Developed and presented the "Unified Clusters" talk to many groups

V0.8, Sep 2001 – Was scheduled to present "A Survey of Cluster Technologies" at the Encompass 2001 in Anaheim CA, but was interrupted by other events (11-Sep-2001) ☹

V0.9, Oct 2002 – Presented "A Survey of Cluster Technologies" at the HP Enterprise Technology Symposium 2002 in St Louis MO

V1.0, Jul 2003 – Published "A Survey of Cluster Technologies" in the OpenVMS Technical Journal, http://h71000.www7.hp.com/openvms/journal/v2/articles/cluster.html

V1.0, Aug 2003 – Presented "A Survey of Cluster Technologies" at the HP Enterprise Technology Symposium 2003 in Atlanta GA, because the traditional HP customers had not attended HP ETS 2002, and there was a lot of interest in the material

V1.9, Sep 2003 – Added IBM HACMP and Sun Microsystems SunClusters material

V2.0, Nov 2003 – Added Oracle, PolyServe and Veritas material, re-organized the Disaster Tolerance section, presented at the OpenVMS Technical Update Days

V2.1, Feb 2004 – Updated extensively with new information on NSK and Veritas

V2.2, Aug 2004 – Added MySQL Cluster, updated Oracle information for 10g

V2.3 Oct 2004 – Added Microsoft SQL Server 2000/2005

V2.4 Jan 2005 -- Edited for publication in the OpenVMS Technical Journal