BTech 450 Industrial Project

Project Report

8 October 2006

(Alan) Huan-Chun Peng Hsu
hpen009@ec.auckland.ac.nz
UPI: hpen009, ID: 3161985

High-Availability (HA) Clusters

This report on HA clusters consists of three main parts: First, an introduction on server clustering covering the basics, giving an introduction on how server clusters function. The latter two sections are focused on Microsoft and Linux server clustering solutions.

1. Introduction

High-availability clusters are implemented primarily for the purpose of improving the availability of services which the cluster provides. They operate by having redundant nodes, which are then used to provide service when system components fail. The most common size for an HA cluster is two nodes, which is the minimum requirement to provide redundancy. HA cluster implementations attempt to manage the redundancy inherent in a cluster to eliminate single points of failure. (1)

Normally, if a server with a particular application crashes, the application will be unavailable until someone fixes the crashed server. HA clustering remedies this situation by detecting hardware/software faults, and immediately restarting the application on another system without requiring administrative intervention. As part of this process, clustering software may configure the node before starting the application on it. For example, appropriate file systems may need to be imported and mounted, network hardware may have to be configured, and some supporting applications may need to be running as well. (2)

HA clusters are often used for key databases, file sharing on a network, business applications, and customer services such as electronic commerce websites. (2)

HA cluster implementations attempt to build redundancy into a cluster to eliminate single points of failure, including multiple network connections and data storage which is multiply connected via shared storage such as storage area networks (SAN) or network attached storage (NAS) systems. (2)

HA clusters usually use a heartbeat private network connection which is used to monitor the health and status of each node in the cluster. One subtle, but serious condition where every clustering software must be able to handle is split-brain. Split-brain occurs when all of the private links go down simultaneously, but the cluster nodes are still running. If that happens, each node in the cluster may mistakenly decide that every other node has gone down and

attempt to start services that other nodes are still running. Having duplicate instances of services may cause data corruption on the shared storage. (2)

1.1 Common Configurations (2)

The most common size for an HA cluster is two nodes, since that's the minimum required to provide redundancy, but many clusters consist of many more, sometimes dozens, of nodes. Such configurations can sometimes be categorized into one of the following models:

- Active/Active — traffic intended for the failed node is either passed onto an existing node or load balance across the remaining nodes. This is usually only possible when the nodes utilize a homogeneous software configuration.

- Active/Passive — provides a fully redundant instance of each node, which is only brought online when its associated primary node fails. This configuration typically requires the most amount of extra hardware.

- N+1 — provides a single extra node that is brought online to take over the role of the node that has failed. In the case of heterogeneous software configuration on each primary node, the extra node must be universally capable of assuming any of roles of the primary nodes it is responsible for. This normally refers to clusters which have multiple services running simultaneously; in the single service case, this degenerates to Active/Passive.

- N+M - In cases where a single cluster is managing many services, having only one dedicated failover node may not offer sufficient redundancy. In such cases, more than one (M) standby servers are included and available. The number of standby servers is a tradeoff between cost and reliability requirements.

The term *Logical host* or *Cluster logical host* is used to describe the network address which is used to access services provided by the cluster. This logical host identity is not tied to a single cluster node. It is actually a network address/hostname that is linked with the service(s) provided by the cluster. If a cluster node with a running database goes down, the database will be restarted on another cluster node, and the network address that the users use to access the database will be brought up on the new node as well so that users can access the database again.

1.2 Application Design Requirements (2)

Not every application can run in a high-availability cluster environment, and the necessary design decisions need to be made early in the software design phase. In order to run in a high-availability cluster environment, an application must satisfy at least the following technical requirements:

- There must be a relatively easy way to start, stop, force-stop, and check the status of the application. In practical terms, this means the application must have a command line interface or scripts to control the application, including support for multiple instances of the application.

- The application must be able to use shared storage (NAS/SAN).

- Most importantly, the application must store as much of its state on non-volatile shared storage as possible. Equally important is the ability to restart on another node at the last state before failure using the saved state from the shared storage.

- Application must not corrupt data if it crashes or restarts from the saved state.

The last two criteria are critical to reliable functionality in a cluster, and are the most difficult to satisfy fully. Finally, licensing compliance must be observed.

2. Microsoft Cluster Server

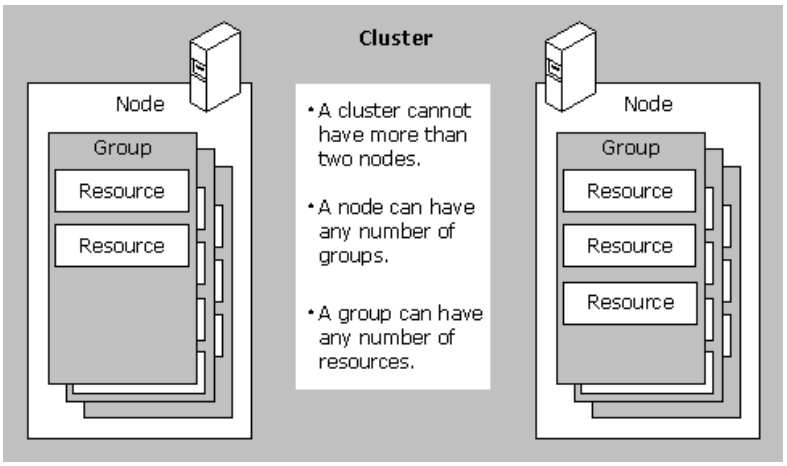
Server clustering is offered by Microsoft for the Windows Server platform. This section will be primarily focused on Windows Server 2003 clustering as it is the latest version available. The clustering software, Microsoft Cluster Server (MSCS), is shipped with the Enterprise Edition of Windows Server 2003.

2.1 MSCS Basics (4)

MSCS supports clusters nodes which are specially linked servers running the cluster service. The primary function of MSCS occurs when one server in a cluster fails or is taken offline. With MSCS, the other server in the cluster takes over the failed server's operations. Clients using server resources experience little or no interruption of their work because the resource functions move from one server to the other. The primary purpose of clustering is to provide failover and re-instantiation of services and resources, thereby providing increased availability for the services (e.g., messaging, database, file and print, etc.).

MSCS is comprised of two main components: clustering software and the Cluster Administrator (cluadmin.exe, a GUI and cluster.exe, a command-line management tool). The clustering software enables the two servers of a cluster to exchange specific types of messages that trigger the transfer of resources at the appropriate times. The clustering software has two primary components: the Cluster Service and the Resource Monitor. The Cluster Service runs on each cluster server. It controls cluster activity, communication between cluster servers, and failure operations. The Resource Monitor handles communication between the Cluster Service and the application resources. The Cluster Administrator is a graphical application that is used to manage a cluster. It runs on any version of NT (server, workstation) that has Service Pack 3 or later installed, Windows 2000, Windows XP and Windows 2003.

In MSCS, a cluster is a configuration of two nodes, each of which is an independent computer system. Together, these independent servers create a "server cluster." The cluster appears to users as a single server. For MSCS, both nodes must be running NT Server - Enterprise Edition, Windows 2000 Advanced/Datacenter Server or Windows Server 2003 Enterprise/Datacenter Server. The network applications, data files, and other tools you install on the nodes are the cluster resources, which provide services to network clients. A resource is hosted on only one node at any time. The figure below shows the relationship between nodes, groups, and resources.



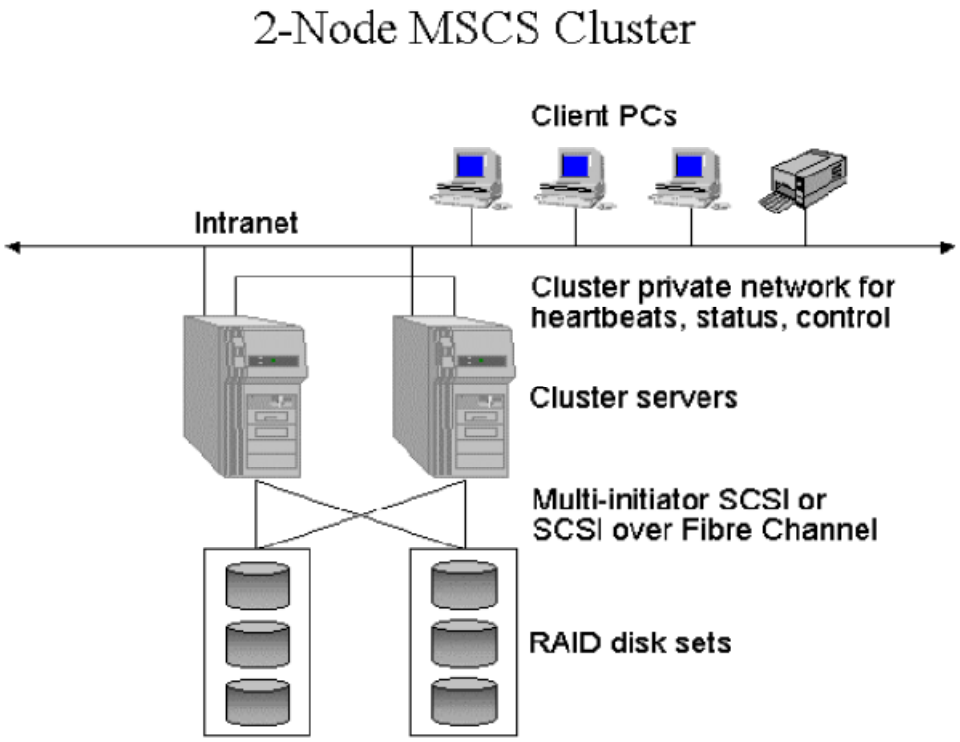
- Node: The term used to refer to a server that is a member of a cluster.

- Resource: A hardware or software component that exists in a cluster, such as a disk, an IP address, a network name, or an instance of an Exchange 2000 component.

- Group: A combination of resources that are managed as a unit of failover. Groups are also known as resource groups or failover groups.

Microsoft Windows Server 2003 Enterprise Edition now supports 4-node clusters (was two in previous versions), and Windows Server 2003 Datacenter Edition now supports 8-node clusters (was four in previous versions). (5)

The following figure illustrates components of a two-node server cluster with shared storage device connections using SCSI or SCSI over Fiber Channel. (6)



2.2 Clustered Applications (3)

Most clustered applications, and their associated resources, are assigned to one cluster node at a time. If Server Cluster detects the failure of the primary node for a clustered application, or if that node is taken offline for maintenance, the clustered application is started on a backup cluster node. Client requests are immediately redirected to the backup cluster node to minimize the impact of the failure.

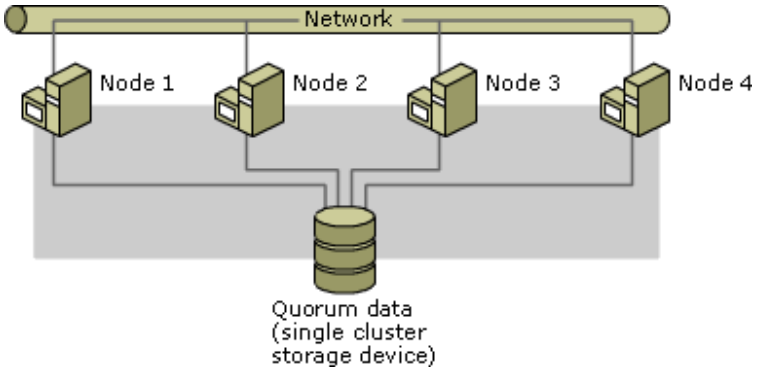
Though most clustered services run on only one node at a time, a cluster can run many services simultaneously to optimize hardware utilization. Some clustered applications may run on multiple Server Cluster nodes simultaneously (for example, Microsoft SQL Server) if it is cluster aware.

In Windows Server 2003 clustering, the application does not have to be cluster aware, custom applications can also be clustered through the support of generic application clustering. This can be any type of application, script or service.

2.3 “Quorum” (3)

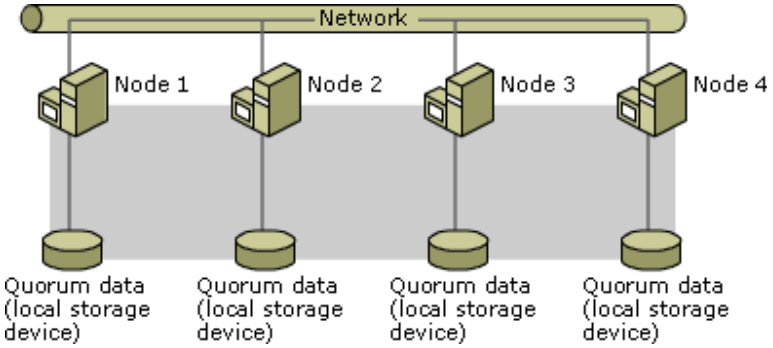
Nodes in a cluster use a *quorum* to track which node owns a clustered application. The quorum is the storage device that must be controlled by the primary node for a clustered application. Only one node at a time may own the quorum. When an application fails over to a backup node,

the backup node takes ownership of the quorum. When the cluster nodes are all attached to a single storage device, the quorum may be created on the storage device. This type of cluster is called a *single quorum device server cluster* when built with Windows Server 2003. Figure below shows a four-node single quorum device server cluster.



Connecting all nodes to a single storage device simplifies the challenge of transferring control of the data to a backup node. However, if the storage device fails, the entire cluster fails. If the storage area network (SAN) fails, the entire cluster fails. Both the storage device and the SAN can be designed with complete redundancy to prevent this.

Majority node set (MNS) server clusters store the quorum on a locally attached storage device connected directly to each of the cluster nodes. For a backup node to assume control of the quorum, the backup node must have a copy of the data stored within the quorum. Server cluster handles this requirement by replicating quorum data across the network. As the figure below shows, majority node set clusters require only that the cluster nodes be connected by a network. That network doesn't need to be a local area network (LAN), either. It can be a wide area network (WAN) or a virtual private network (VPN) connecting cluster nodes in different buildings or cities—allowing a cluster to overcome geographic restrictions imposed by the storage connections.



Majority node set clustering does have requirements that single quorum device server clusters lack. To effectively fail over between nodes, majority node set clusters must have at least three nodes. More than half of the cluster nodes must be active at all times. So, if you design a cluster with three nodes, two of them must be active for the cluster to be functional. Eight node clusters must have five nodes active to remain online. Single quorum device server clusters require that only a single node continues to function.

3. Linux Cluster Server

In this section we will examine some open source alternative of server clustering using Linux, and how it compares to Microsoft's solution. Since the implementation is quite similar, only the major differences will be outlined.

3.1 Linux-HA (7)

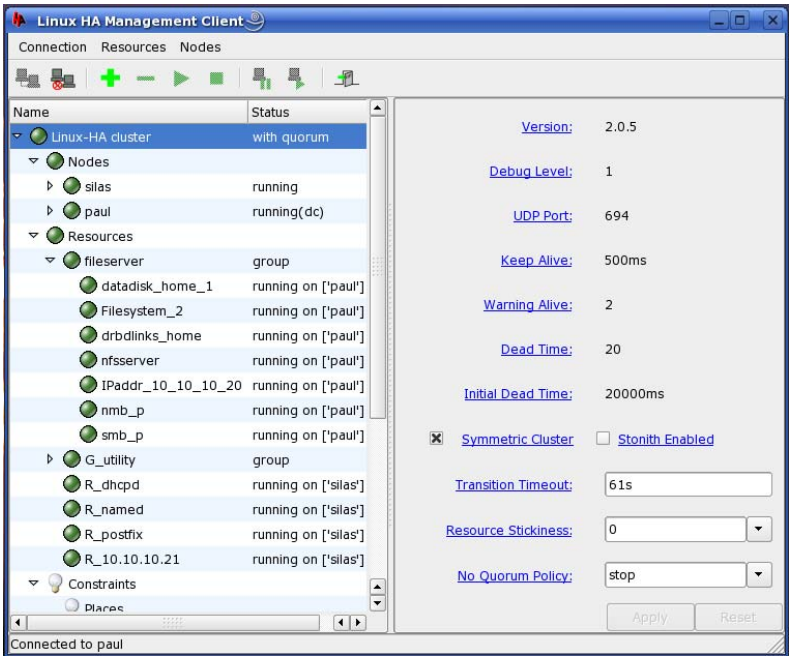
The Linux-HA (High-Availability Linux) project provides a high-availability (clustering) solution for Linux, FreeBSD and Solaris which promotes reliability, availability, and serviceability (RAS).

The project's main software product is *Heartbeat*, a GPL-licensed portable cluster management program for high-availability clustering. Its most important features are:

- No fixed maximum number of nodes - Heartbeat can be used to build large clusters as well as very simple ones
- Resource monitoring: resources can be automatically restarted or moved to another node on failure
- Fencing mechanism to remove failed nodes from the cluster
- Sophisticated resource management, resource inter-dependencies and constraints
- Time-based rules allow for different policies depending on time
- Several resource scripts (for Apache, DB2, Oracle, PostgreSQL etc.) included
- GUI for configuring, controlling and monitoring resources and nodes

In the first version of Heartbeat, monitoring the status of resources and applications must be done through monitor packages such as MON. With the second release of Heartbeat (v2), monitoring can be done with the included CRM (cluster resource monitor). All resource initialization and monitoring are done by scripts, some popular ones are included (see above) with Heartbeat. (8)

Following is a sample screenshot of the Linux HA management graphical user interface:



4. Windows or Linux?

When it comes to choosing between Windows and Linux, the major points to consider are:

- Cost. Windows solutions can cost a lot, while Linux is almost always free. Linux architectures are usually more flexible, while Windows are somewhat limited. For example, Windows Server 2003 supports clustering of up to 8 nodes, while there is no set limit for Linux-HA.

- Easy of use. Windows solutions tend to be more user friendly, but over the years, most Linux software have established graphical user interface as well (as seen above). However in case of server software, this really shouldn't be a great issue for an experienced system administrator.

- Support of applications. This depends on what kind of solution the company wants to use. If they want to create custom software solutions and services using Microsoft's platform (such as .NET) then using Windows servers is the most obvious way to go.

So ultimately the choice falls on what the company wants to use, and how much they are willing to pay.

5. References

(1) Wikipedia Computer Cluster
http://en.wikipedia.org/wiki/Computer_cluster

(2) Wikipedia HA Cluster
http://en.wikipedia.org/wiki/High-availability_cluster

(3) Microsoft Windows Server 2003 Clustering White Paper
http://www.microsoft.com/windowsserver2003/techinfo/overview/bdmtdm/default.mspx

(4) Scott Schnoll's Microsoft Cluster Server Center
http://www.nwnetworks.com/cluster.html

(5) Technical Overview of Windows Server 2003 Clustering Services
http://www.microsoft.com/windowsserver2003/techinfo/overview/clustering.mspx

(6) Windows Server 2003 Server Cluster Architecture
http://www.microsoft.com/windowsserver2003/techinfo/overview/servercluster.mspx

(7) Wikipedia Linux-HA
http://en.wikipedia.org/wiki/Linux-HA

(8) The High-Availability Linux Project
http://www.linux-ha.org/