The University of Auckland
Information Technology, Bachelor of Technology

BTech450: Industrial Project

Final Project Report

Huan-Chun Peng Hsu
hpen009@ec.auckland.ac.nz

Abstract

This is the final report for the BTech450 project. This project is to develop a model information system for a medium sized company, to be used as a deployment template for system integration.

This report details the various aspects of the project, including the project objective, approach, developments and problems.

Although the project does not reach its final phase due to the lack of information, this report concludes with some recommendations on how the project could proceed in the future.

Attached to the end are contents of the reports from the research phase, along with the work done in previous phases, they could serve as valuable knowledge base and foundation for the future of this project

Table of Contents

1. Introduction

The goal of this project is to design a model information system (hardware and software) for a medium sized company with two satellite branches and several mobile staff members.

The design has to consider in depth the following functional requirements: high availability (24x7) and high performance for users including remote access and mobile workers, high security against hacker and virus, secure and easy to retrieve storage and fast disaster recovery.

The approach must be based on academic considerations as well as commercial practicality.

The student will need to learn or know Microsoft Windows 2003, Mandriva Linux and Citrix Metaframe as the commercial references. We will take a fitness-for-purpose and Total-cost-of-ownership approach, and design the procedure for developing and implementing information systems.

This project is sponsored by Computers New Zealand (CNZ). TN Chan, CEO of CNZ acts as the industrial mentor and oversees the project progress.

1.1 Initiative

CNZ believes that currently in New Zealand, system integrators are not doing a very good job on providing a robust and comprehensive information system.

Most companies just put together something that “works”, with no consideration major issues such as: system management, usability, accessibility, expendability, and security. Important long term factors such as: operating cost, maintenance cost and disaster prevention/recovery are also overlooked.

1.2 Objectives

The requirements on the final system are:

- High Availability (24/7)
- High Performance
- Remote Access / Mobile Workers
- High Security
- Fast Disaster Recovery

- Commercial Practicality

1.3 Perspectives

We will be looking at things from the following perspectives:

- Academic Perspective
  System design and evaluation techniques
  Unbiased decisions to commercial products

- Commercial Perspective
  Commercial practicality

- Fitness for Purpose, FFP
  Does the design satisfy our needs?

- Total Cost of Ownership, TCO
  Startup and operating costs

1.4 “The Company”

The “medium sized” (or “mid-sized”) company is defined as a company which:

- Is not directly related to computer industry
- Has a head office with 30 employees
- Has two branches, each with 5 employees
- Has 7 mobile works (5 from HQ + 2 from branches)

1.5 Approach

The proposed steps of the project are:

1. Set requirements
   For the different components, such as software, hardware and services

2. Research
   Verify the requirements and find suitable solutions

3. Design
   Design the template; put everything together into a whole system

4. Refine
   Refine and redesign the template until satisfied

2. Set Requirements

In this phase we set the requirements for the various components of the system, including: software, hardware and services. We first visualize the final deliverable, and then try to identify the requirements as well as problems which may need to be investigated in the research phase.

2.1 Visualization

The visualization of the final deliverable will help us to identify the major components and requirements of the system. The visualization is based on the student's personal understanding of information systems, along with some basic research.

Depending on the result of the research phase, the final design may or may not be based on this.

The main components of the visualization are listed below.

2.1.1 Workstations

Two types of configuration:

- Process Worker
- Knowledge Worker

The Process Worker workstation will have basic level of hardware that is suitable for everyday office tasks such as word processing and spreadsheet.

The Knowledge Worker workstation will have more powerful hardware that can be used to run resource intensive applications efficiently, such as Photoshop and AutoCAD.

The personal files are not stored locally on the workstations, but on a central file server. This will give us the benefits of:

- Easy remote access
- Easy to backup files

- Easy to update the workstation configuration

2.1.2 Software

Three major software categories:

- Operating system: Microsoft Windows XP Professional
  Windows offers users with familiar working environment for most people; supports most software and hardware available on the market.

- Office application: Microsoft Office 2003
  Office contains applications for most common tasks such as word processing (Word), spreadsheet (Excel), presentation (PowerPoint) and e-mail (Outlook). Its formats are most commonly used, so the files will be easily exchangeable.

- Custom applications
  Depending on the nature of the company, most will run some kind of custom applications. For example: an engineering firm might want to use AutoCAD, and an advertising firm might want to use Photoshop.

2.1.3 Servers

Four main servers:

- Public Website and E-mail Server
  This provides an online presence of the company.

- Private Website
  This is an internal company website for posting announcements, events, discussions and sharing files.

- File Server
  This is the central location to store all employees' files. The files can be easily remote accessed and backed up.

- Backup Server
  This server will automatically backup files every night, and will be located in one of the branch offices.

2.1.4 Services

Two essential services:

- System Administrator
  One dedicated personnel to manage and support the company's information system. Most problems will be able to be resolved immediately without external resources.

- Backup
  In addition to the backup server, files should be backed up on to a physical medium such as a DVD or external hard disk. This will then be given to a trusted employee and to be kept at his house.

2.1.5 Remote Access

Possible remote access options:

- Virtual Private Network
  A VPN will connect all the offices together through the internet. This also allows employees to access their files from externally.

- WAP version of the internal website
  This will allow mobile devices, such as cell phones and PDAs, to access important company resources.

- Terminal Services ("Remote Desktop")
  This will allow off-site access to all resources, including some custom applications that are not normally available.

2.1.6 Preliminary Topology

This diagram is based on the visualization and shows the possible components and their locations.

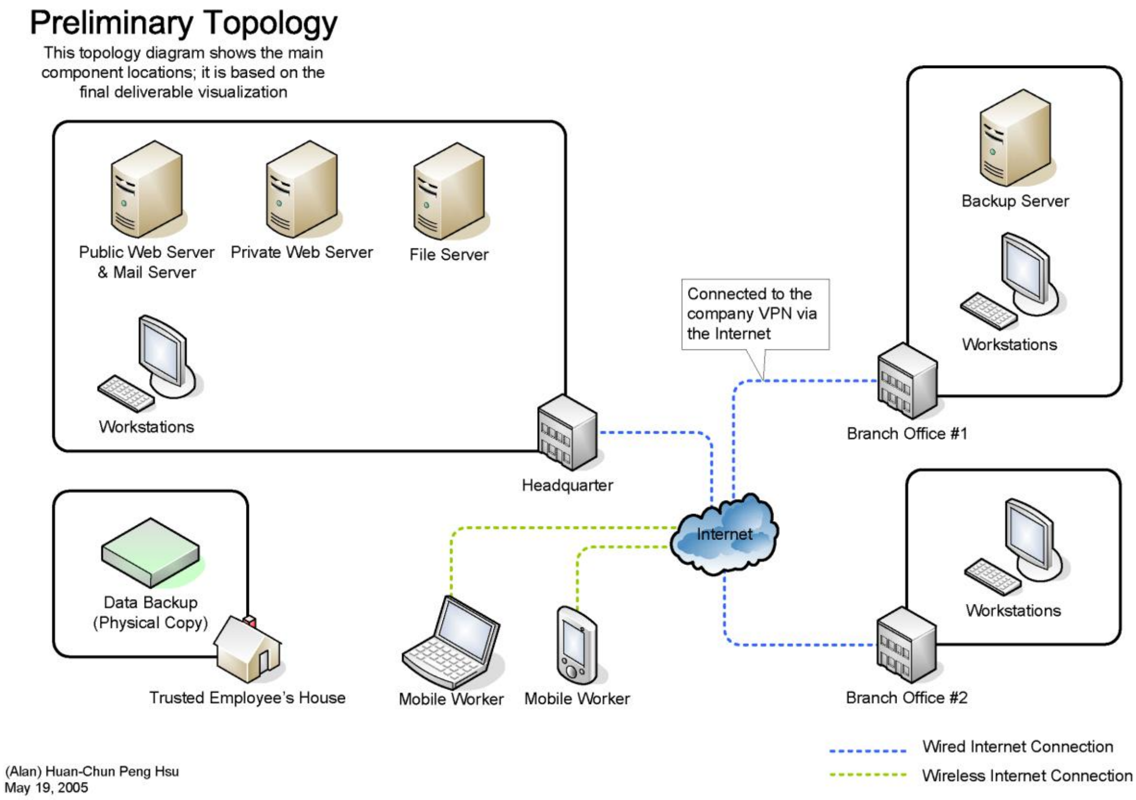


Figure 1: Preliminary Topology

2.2 Identified Requirements

The following are the main requirements which have been identified from the visualization:

- Software

  Operating System
  - Support many hardware and software
  - Easy to use and maintain

  Office Application
  - Exchangeable format

- Hardware

  Workstation
  - Process Worker
  - Knowledge Worker

Server
Public Website / E-mail
Private Website
File Server
Backup Server

- Services

System administrator
Off-site backup

2.3 Identified Problems

The following are some of the problems identified from the visualization:

- Network
How are the computers and servers connected?
What kind of internet connection is necessary?

- Servers
Are the different servers really required? Can they be combined?
What about redundant servers?
What about backup power?

- Remote Access
What kind of remote access is necessary?
Is terminal services really required?

- Mobile Worker
In order to ensure service quality, does the company need to supply notebook/PDA and the internet connection?

- Others
What about security, anti-virus and firewall?
What about disaster recovery?

3. Research

In this phase we need to verify the requirements and find suitable solutions, as well as solving the problems identified from the previous phase.

In order to obtain practical information, I have refined my research style and scope a numerous times. Below are the different research areas that I have focused on during the research phase:

1. Studies on system integration
2. Case studies on existing information systems
3. Gradually reducing the search scope
4. Focus only on high-availability methods
5. Focus only on network redundancy
6. Focus only on server clustering

3.1 Resources Consulted

The three major resources used in this phase are:

- The Internet
  Websites and discussion forums

- IEEE & ACM Database
  Technical publications

- Information System student
  Friend, final year INFOSYS student

3.2 Different Types of Information

During the research phase, I categorize the information I encounter into the following two types:

- Educational
  Educational information is the basic understanding of how things work.
  For example: what is RAID and how it works.

- Practical
  Practical information is the knowledge accumulated from experience.
  For example: How much bandwidth would a typical medium sized company require?

Because I do not understand many of the technologies involved in this project, such as SAN (storage area network), educational information is useful for me to learn them.

However in order to build a good information system, it is really the practical information we need in the research phase.

3.3 Obtaining Practical Information

From many years of web surfing experience, one would think practical information can be easily obtained. There are plenty of websites that teaches you how to build your own computer and how to maximize its performance. There are also websites teaching you how to setup your own web server. You can also find many online forums discussing topics like this. These are all good and practical information.

But wait. The practical information we need in this project is for commercial level applications. Are these easily obtained?

As I found out from the many hours spent on researching, the kind of practical information we need for this project is in an entirely different realm.

On the Internet, is it not easy to find answers to questions such as “What are some cost effective ways to build a high-availability cluster server?” You also cannot find forums discussing what kind of server redundancy solution is the best, or how to build a reliable LAN for a company.

I attribute the lack of information into the follow factors:

- Cost
  The costs involved in commercial level applications are very high. A single commercial level router could cost thousands of dollars.

- Scale
  The scale of the equipments involved, such as computers in a network, is very different from what is used in homes.

- Feasibility
  The equipments involved are not what normal consumers would come across or use.

Because of the above factors, ordinary people just do not have the interest and/or the ability to play around with such equipments and technologies. And with the lack of interest and motivation, practical information for commercial applications is very rare on the Internet.

3.4 Proprietary vs. Open

Another interesting trend I have noticed with the information regarding commercial applications is the differences between proprietary solutions and open source solutions.

Information on proprietary solutions tends to be more organized than open source solutions. Websites offering information on proprietary solutions are much more structured and arranged. For example, information on Microsoft's server clustering is much easier to obtain than Linux server clustering.

I believe this is simply due to the fact that there's more money to be made with proprietary solutions.

However in either case, most information is educational.

3.5 Review of Resources

In this section we will review the different resources that have been consulted in the research phase.

3.5.1 The Internet

The Internet, as discussed above, has a lot of practical information at consumer level, available from many enthusiast sites and forums. Practical information at commercial level is very rare.

3.5.2 IEEE and ACM Database

The IEEE and ACM databases contain millions of technical publications. They have the latest technologies and studies. However all the papers I have looked at are academic studies – most are too theoretical and does not have commercial implementations, so they cannot be easily put into practical use.

3.5.3 Information System Student

Information Systems (INFOSYS) majors learn more about commercial equipments, such as networking and routing applications. I have discussed many different aspects of this project with one of my friend, who is a final year INFOSYS student. However, to my surprise, when it comes into practical knowledge, she appears to be as clueless as I am.

3.6 Example of Practical Information

One of the practical information I have encountered is a paper published by GarrettCom (a networking equipment vendor), titled “Redundancy with Standards in Industrial Ethernet LANs”.

This paper suggests the use of ring topology and IEEE 802.1w Rapid Spanning Tree Protocol to provide redundancy for a local area network:

- Ring topology
  Is easy to setup and also minimizes cabling and cost
  Can be easily converted into a partial mesh for higher redundancy

- Rapid Spanning Tree Protocol
  Has very fast route convergence time, hence fast recovery from link failures
  Is available in most commercial routers

3.7 Practical Information from the University

There was not much information I could use on this project from the papers I have taken in the University. There is one thing from Computer Science 742 (“Data Communications and Networks”) that came into mind when I was researching about network redundancy:

IPv6 multihoming using “shim6” (Level 3 shim for IPv6) protocol

Using this could reduce the single-point-of-failure problem with IPv4 multihoming because it relied on a single router. However currently *shim6* is only an IETF draft (http://tools.ietf.org/wg/shim6/) and it is yet to be finalized.

4. Conclusion

Although with the continuous refinement of the search style and scope, there was not information to progess to the design phase in the end.

For the future of this project, I recommend that the scope to be reduced. For example, focus only on server clustering. With a much smaller scope like that, I believe it will be possible to conduct some more in-depth research, such as interviewing system administrators at different organizations.

Such investigation was not feasible with the current project scope because of the sheer breadth involved, and the student’s time constraints.

Also because this project is research based, and does not utilize much of the skills learned in the BTech IT degree, it would be a good idea to let future students know in advance that research project like this is much different from “ordinary” programming projects. The main differences would be:

Programming projects

- Can get a taste of commercial development environment, which could be a great valuable experience for the future
- Can put the skills learned in the University into use

Research projects

- No programming
- Get to learn new things
- Refine your research skills

Attached to the end of this document is the various information from the reports I have produced in the research phase. Together with the planning and the ground work established in the earlier phases, they could serve as valuable knowledge base or the foundation for the future continuation of this project.

_
10/24/2006

5. Appendix

5.1 High-Availability Subject Areas

Things can be researched on:

- Network (LAN)
  How to design a fault tolerant network? What would be the best topology to use?

- Network (External)
  How to ensure high level of internet availability?

- Storage
  What is a good storage solution?

- Workstation
  How do we ensure a high level of workstation availability?
  If one break down, can we immediately switch to another one and continue working?

- Servers
  How to maintain high server uptime?

- Backup and recovery
  What are some fast backup and recovery methods?
  What about disaster recovery plan?

5.2 High-Availability Resources

Some resources that might be useful:

- Reliability, Availability and Optimization
  http://www.weibull.com/systemrelwebcontents.htm

- Reliability and Availability
  http://www.eventhelix.com/RealtimeMantra/FaultHandling/reliability_availability_basics.htm
  http://www.eventhelix.com/RealtimeMantra/FaultHandling/system_reliability_availability.htm

- Hardware Fault Tolerance
  http://www.eventhelix.com/RealtimeMantra/HardwareFaultTolerance.htm

- High-availability cluster
  http://en.wikipedia.org/wiki/High-availability_cluster

- Storage area network
  http://en.wikipedia.org/wiki/Storage_area_network

- Disaster recovery
  http://en.wikipedia.org/wiki/Disaster_recovery

- IEEE Task Force on Cluster Computing: High Availability (HA)
  http://www.ieeetfcc.org/high-availability.html

5.3 Reviews of Some High-Availability Resources

Redundancy with Standards in Industrial Ethernet LANs
http://www.garrettcom.com/techsupport/papers/redundancy.pdf
This paper suggests that we use ring topology with IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) for redundancy and fast recovery. The older version of the protocol (Spanning Tree Protocol, STP, 802.1d) is available more widely in commercial equipments; its convergence time is 30 to 50 seconds. RSTP is designed to be faster and compatible with STP, and some vendor might offer their own proprietary version of it. The papers suggests the use of ring topology because using "daisy-chain" or sequential point-to-point topology is optimal for minimizing cabling, and routing the end of the cable back to the router/switch is fairly easy. This provides a ring network with redundancy.

Network Topology
http://en.wikipedia.org/wiki/Network_topology
http://www.delmar.edu/Courses/ITNW2313/network.htm
http://www.networkdictionary.com/networking/mesh.php
Many resources has suggested that mesh topology provides the best redundancy, but it is also the most expensive to install. In a fully connected mesh network, one node is connected to every other node; the number of connections can be reduced to form a partial mesh, which is less expensive to implement. Of course it would be impossible to build the mesh network by using individual computers as nodes, but it can be a good idea to use routers/switches as nodes. Another topology is the ring topology, it is easier to install and still provides redundancy, as mentioned above.

High-availability cluster
http://en.wikipedia.org/wiki/High-availability_cluster
HA clusters are computer clusters that are implemented primarily for the purpose of improving the availability of services they provide. It uses redundant computers (or "nodes") to provide service in case part of the system fails. This is often used for key databases, file sharing, business applications, and customer services. Products found extensively in commercial or research/academic use:
HA-OSCAR - High Availability Open Source Cluster Application Resources
Veritas Cluster Server - multi-platform
Sun Cluster - Solaris only
OpenVMS - The original clustering OS
HP ServiceGuard for HPUX and Linux
Linux-HA — a commonly used free software HA package for the Linux OS.
High Availability Cluster Multiprocessing aka IBM HACMP for AIX
Microsoft Cluster Server (MSCS) - Windows only

IEEE Task Force on Cluster Computing: High Availability

http://www.ieeetfcc.org/high-availability.html
Provides some links to common commercial HA products

Network attached storage
http://en.wikipedia.org/wiki/Network-attached_storage
This article is mostly educational but has noted a few free NAS software which can be used to build custom NAS servers: FreeNas, NASLite and Openfiler

Andrew File System (AFS)
http://en.wikipedia.org/wiki/Andrew_file_system
This is a distributed networked file system developed by Carnegie Mellon University, and is the file system used in the University of Auckland. The benefits include good security and scalability over traditional networked storage solutions. Free implementations include OpenAFS and Arla.

5.4 High-Availability 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.

5.4.1 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)

5.4.1.1 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.

5.4.1.2 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.

5.4.2 Microsoft Cluster Server

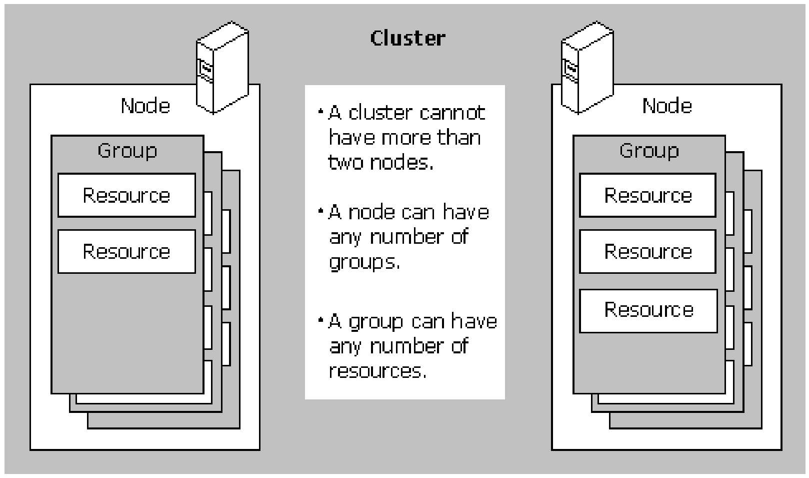
Server clustering is offered by Microsoft for the Windows Server platform. This section will be primarily focused on Windows Server 2003 clustering as it is the latest version available. The clustering software, Microsoft Cluster Server (MSCS), is shipped with the Enterprise Edition of Windows Server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

5.4.2.1 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.

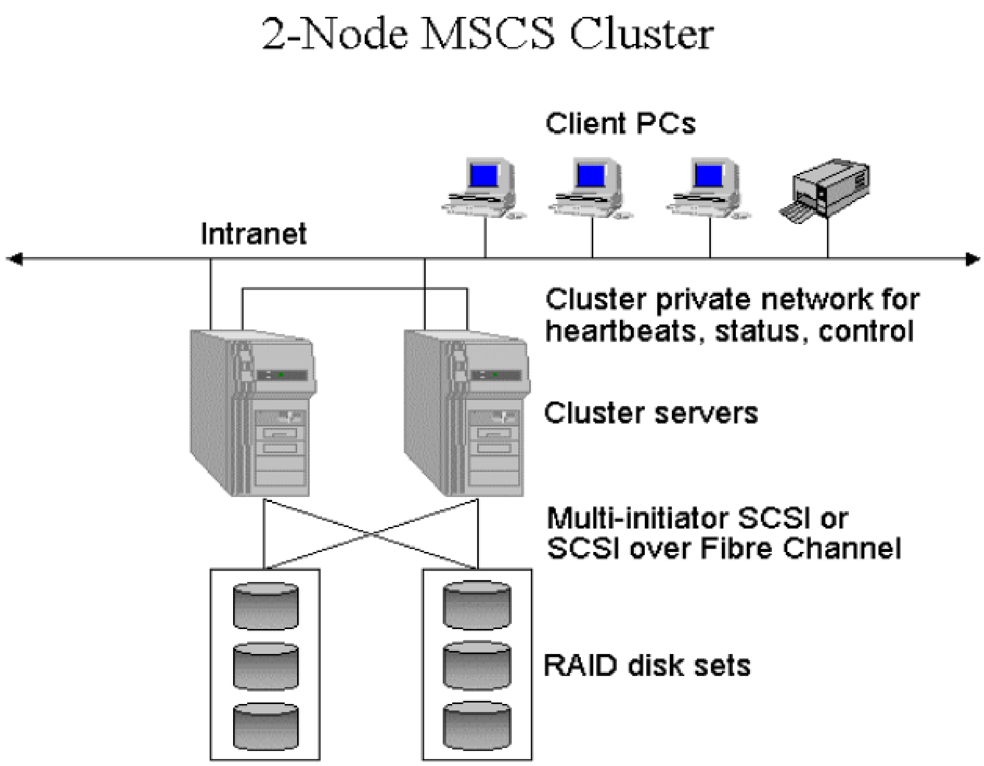
In MSCS, a cluster is a configuration of two nodes, each of which is an independent computer system. Together, these independent servers create a "server cluster." The cluster appears to users as a single server. For MSCS, both nodes must be running NT Server - Enterprise Edition, Windows 2000 Advanced/Datacenter 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)



5.4.2.2 Clustered Applications (3)

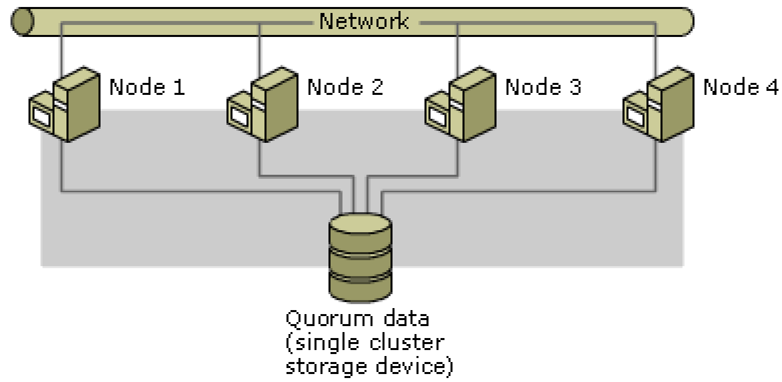
Most clustered applications, and their associated resources, are assigned to one cluster node at a time. If Server Cluster detects the failure of the primary node for a clustered application, or if that node is taken offline for maintenance, the clustered application is started on a backup cluster node.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

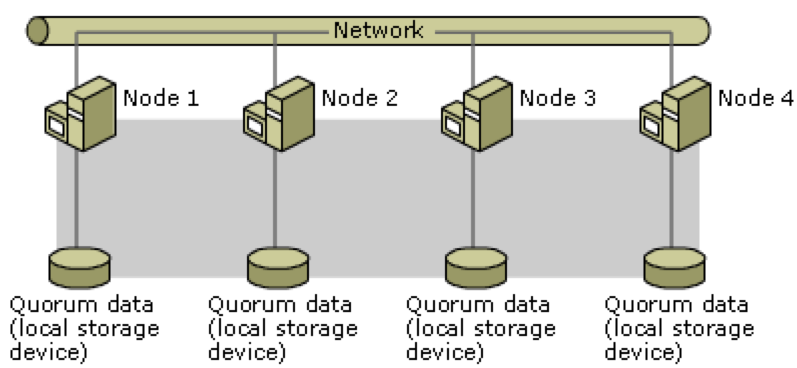
5.4.2.3 "Quorum" (3)

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.

5.4.3 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.

5.4.3.1 Linux-HA (7)

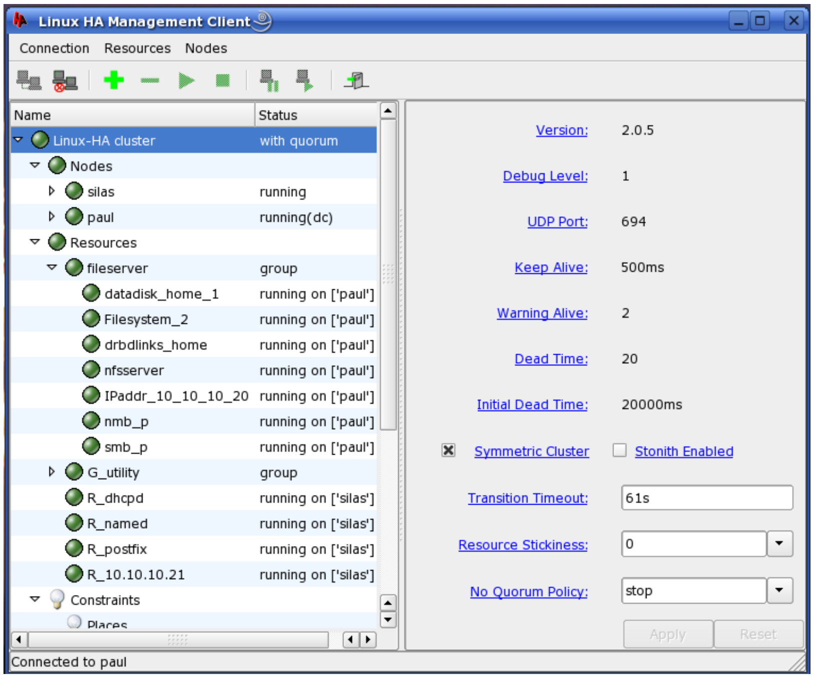
The Linux-HA (High-Availability Linux) project provides a high-availability (clustering) solution for Linux, FreeBSD and Solaris which promotes reliability, availability, and serviceability (RAS).

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:



5.4.4 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.

Please note that there is not enough information in this research on Linux solutions to make a better comparison.

5.4.5 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/