



Delivering High Availability Solutions with Red Hat Cluster Suite

Abstract

This white paper provides a technical overview of the Red Hat Cluster Suite layered product. The paper describes several of the software technologies used to provide high availability and provides outline hardware configurations. The paper is suitable for people who have a general understanding of clustering technologies, such as those found in Microsoft Windows 2000 Advanced Server and Sun Cluster products.

Revision 3d - June 2004

Table of Contents

Introduction.....	3
Red Hat Cluster Suite Overview.....	3
Application Support.....	3
Cluster Manager Technology.....	4
Basic Cluster Manager Operation.....	5
Detailed Operation.....	7
Application Services.....	7
Application Failover times.....	7
Active/Active and Active/Passive.....	8
Cluster communication.....	9
Quorum Partitions.....	9
I/O Barriers.....	10
Watchdog timers.....	11
I/O Subsystem Requirements.....	12
Parallel SCSI Configuration Notes.....	12
RAID considerations.....	13
Heterogeneous File Serving.....	13
Service Monitoring.....	14
Management.....	14
Summary.....	16
Oracle RAC and Cluster Manager.....	16
Cluster Manager and IP Load Balancing.....	16
References.....	17

Introduction

In early 2002 Red Hat introduced the first member of its Red Hat Enterprise Linux family of products - Red Hat Enterprise Linux AS (originally called Red Hat Linux Advanced Server). Since then the family of products has grown steadily and now includes Red Hat Enterprise Linux ES (for entry/mid range servers), Red Hat Enterprise Linux WS (for workstations) and Red Hat Desktop. These products are designed specifically for use in enterprise environments to deliver superior application support, performance, availability and scalability.

Red Hat Cluster Suite is one of two primary clustering products provided by Red Hat. The other is Red Hat Global File System (GFS). Red Hat GFS provides a multi-system, concurrent data sharing environment for applications, and is based on technology obtained by Red Hat with its acquisition of Sistina, in late 2003. Red Hat Cluster Suite and Red Hat GFS utilize the same configuration topologies (and are configured using the same hardware). They also share many of the underlying technologies described in this paper, such as support for I/O Fencing. In many cases small configurations or those using standard applications will be deployed using Red Hat Cluster Suite; medium/large systems or those with parallel applications will be suited to Red Hat GFS. Red Hat Cluster Suite is provided as a component of Red Hat GFS.

Both products are supported on Red Hat Enterprise Linux AS and ES, and are available for Intel x86, Itanium2, EM64T and AMD AMD64 systems.

Red Hat Cluster Suite Overview

Red Hat Cluster Suite includes two distinct clustering features. The major feature, and the focus of this white paper, is the high availability clustering capability, called Cluster Manager. This provides continued application operation in the event of server shutdown or failure. The second feature, called IP Load Balancing (originally called Piranha), provides network load balancing. It allows a front-end server to redirect IP network packets to a group of back-end servers in a balanced manner, thereby improving the total network performance. The IP Load Balancing feature is briefly described at the end of this paper.

The remainder of this white paper will primarily concentrate on the Cluster Manager feature, and also show how Cluster Manager and IP Load Balancing can be used together to create sophisticated multi-tier highly available configurations.

Application Support

When designing a high availability configuration the first task is to identify whether the customer's applications will be supported by the planned system. This section describes the applications that can benefit from Cluster Manager

capabilities.

Cluster Manager provides a failover infrastructure for applications that fall into several categories:

- Generic, unmodified applications. Most custom in-house applications can be used in Cluster Manager environments. This applies to any application that can tolerate a few seconds of downtime.
- Databases. Cluster Manager is the ideal way to deliver highly available databases, including Oracle 8i/9i, DB2, MySQL and Red Hat Database.
- Heterogeneous File Serving. Cluster Manager brings high availability to file serving environments such as NFS and SMB/CIFS (using Samba).
- Mainstream Commercial Applications. Cluster Manager can be used with applications such as SAP, Oracle Application Server and Tuxedo.
- Internet, and Open Source applications. Cluster Manager fully supports the most popular Internet and Open Source applications (e.g. Apache).
- Messaging. Using applications such as Sendmail and Domino.

A critical feature of Cluster Manager is that applications do not have to be modified before they can be deployed in a cluster system. In most cases applications are not even aware that they are running in a cluster - they become high availability applications automatically.

Red Hat Enterprise Linux products have many features designed for enterprise environments, so the few applications that are not suitable for deploying in Cluster Manager configurations can still benefit from the other Red Hat Enterprise Linux capabilities. Examples would be Real-Time applications that have low latency requirements (less than a few seconds) and limited buffering capability in their data collection devices, or applications that provide their own clustering infrastructure, such as Oracle Real Application Clusters (RAC) or Veritas Cluster Server configurations.

Cluster Manager Technology

Cluster Manager provides high availability by using a technology widely used by other operating systems - application failover. Application failover is used in most high availability clustering products, such as Microsoft Windows 2000 Advanced Server, Sun Cluster, and Compaq TruClusters. With Cluster Manager, customers benefit from a clean implementation of a well-understood and mature technology.

As a modern clustering implementation, Cluster Manager has been specifically developed for use with today's commodity hardware products; it does not require expensive, special-purpose hardware components. All the configurations described in this paper can be built using standard commodity products. In some cases optional items can be added to further increase system availability, such as an Uninterruptible Power Supply (UPS).

Basic Cluster Manager Operation

The simplest Cluster Manager configuration comprises a pair of servers and an external SCSI or Fibre Channel storage subsystem. As many as eight servers can be configured in a Cluster Manager configuration provided they are all connected to the same external storage subsystem. This allows them to access all shared disks directly. The Cluster Manager software controls access to individual storage partitions, so that only one server can access a particular partition at a time. This is required because standard applications do not support concurrent access to their data files from multiple systems.

Each server will then operate in the same manner as if it were a single, standalone system, running applications and accessing data on its allocated storage partitions. Using multiple servers in this fashion is often referred to as scale-out computing, that is, adding compute power to a configuration with additional systems; scale-up computing, on the other hand, refers to supporting larger numbers of processors in an SMP system.

In addition to their connections to the shared storage array, the servers are also connected using a network or serial interface so that they can communicate with each other using a network polling mechanism. In the event that one of the servers shuts down or fails the other servers will detect the event (due to failed network poller) and automatically start to run the applications that were previously running on the failed server. Selection of which remaining server an application will be restarted on can be pre-selected by the system administrator or randomly selected by the Cluster Manager. This migration of applications from a failed server to the remaining servers is called *failover*. Because all servers are connected to the external shared storage the operational servers can access the failed server's disk partitions and its applications can continue to operate normally. If necessary a remaining server will also take over the IP address of the failed server, so that network operations can continue without interruption. The general layout of a small, 3 node Cluster Manager configuration is shown in Figure 1.

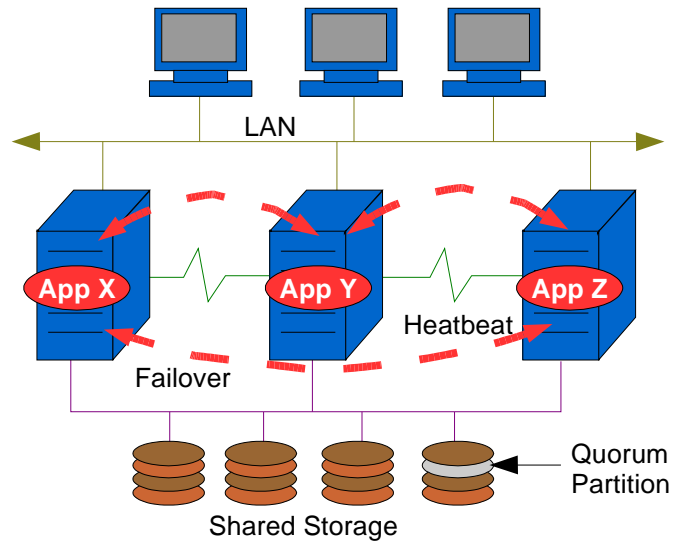


Figure 1 - Typical Cluster Manager Configuration

The crucial technical feature of a Cluster Manager cluster is that the storage is shared, allowing any server to host any application and directly access its data. Cluster Manager provides high availability for applications by managing redundant server resources. However, to make the entire configuration highly available it is necessary to consider the other essential components of the configuration. Enterprise-strength clusters are configured to have no-single-point-of-failure by including redundancy at all levels. This will generally include redundant power supply systems and redundant network interfaces. Also, it is important to realize that a failure of the shared external storage could bring the entire cluster down, so it is vital to use a high availability storage system. This will typically include dual controllers for redundancy and all the storage will be configured in RAID-1 (mirroring) or RAID-5 (parity) sets. A useful analogy is to consider that clustering is to servers what RAID-1 is to disk storage. The two technologies work together to create a complete high availability solution.

Detailed Operation

While the general mechanics of cluster systems are relatively simple, a closer understanding of some of the techniques used to ensure data integrity and high availability can be helpful in ensuring that deployments function as expected and that customer expectations are appropriately set.

This section provides a closer examination of several important Cluster Manager features.

Application Services

Cluster Manager uses the concept of services to implement application failover; the unit of failover is a service rather than an application. A service comprises several items, including:

- A service name
- Any IP address that the application requires
- Mount points
- Device names
- Name of the application stop/start/status control script
- Preferred server node(s) and recovery policy
- Service monitoring interval

During a failover the Cluster Manager will mount partitions at the correct mount points, configure the IP address for the service, and then call the service start script, which will start the application itself. The application will find the same environment that it had on its original server - the failover is essentially invisible to it. Multiple preferred server nodes can be defined, so that when a failover is required Cluster Manager can restart a service on a server that the system administrator has previously selected.

Application Failover times

Application failover times are dependent on two factors:

- The time taken after a failure to trigger the failover process
- The application specific recovery time

The default timer for Network polling is 2 seconds. If the poller fails to receive a response, 6 retries are attempted, making the total time before triggering a failover 14 seconds. Polling interval and retry counters are adjustable during Cluster Manager installation.

Application specific recovery times vary greatly, and can include activities such as rebuilding file systems (fsck) and playing Database recovery journals. Cluster Manager supports the use of Linux journaled file systems, such as Ext3, which greatly reduce file system rebuild times.

In the case where an application is relocated across servers by the system administrator using the Cluster Manager utilities, the service will be shutdown and restarted cleanly using the stop/start script. This eliminates all poller delays and application recovery procedures, so is typically rapid.

Active/Active and Active/Passive

Understanding how applications can be distributed across multiple servers in a cluster is important. In the simplest case, a customer wishes to run several unrelated applications. Each application is set up to access files located on different disk partitions. In a Cluster Manager environment the customer can simply spread the application services across the clustered servers in any way that seems appropriate. All nodes are actively running a share of the total load. This is called Active/Active clustering, since all servers are indeed active. If one server shuts down, the other servers will pick up the load of running all its services.

In the event that the customer wishes to run a single large application on the cluster, it must be remembered that servers cannot access the same disk partition at the same time - because few applications available today provide support for concurrent data update from multiple systems (Oracle RAC is one of the few applications that does support multi-system concurrent update). So, it is necessary to restrict these applications to a single server, leaving the other servers as ready-to-go backups in case of failure. This is called Active/Passive operation. This style of operation typically leaves the Passive systems idle, which is a waste of valuable computing power. To make the Passive systems Active, it is necessary to either find additional applications to run, or to somehow split the data files of the main application so that they can be placed on different disk partitions. An example might be to run a separate MySQL service on each server, each accessing a different database. Example Active/Active and Active/Passive application deployments are shown in Figure 2.

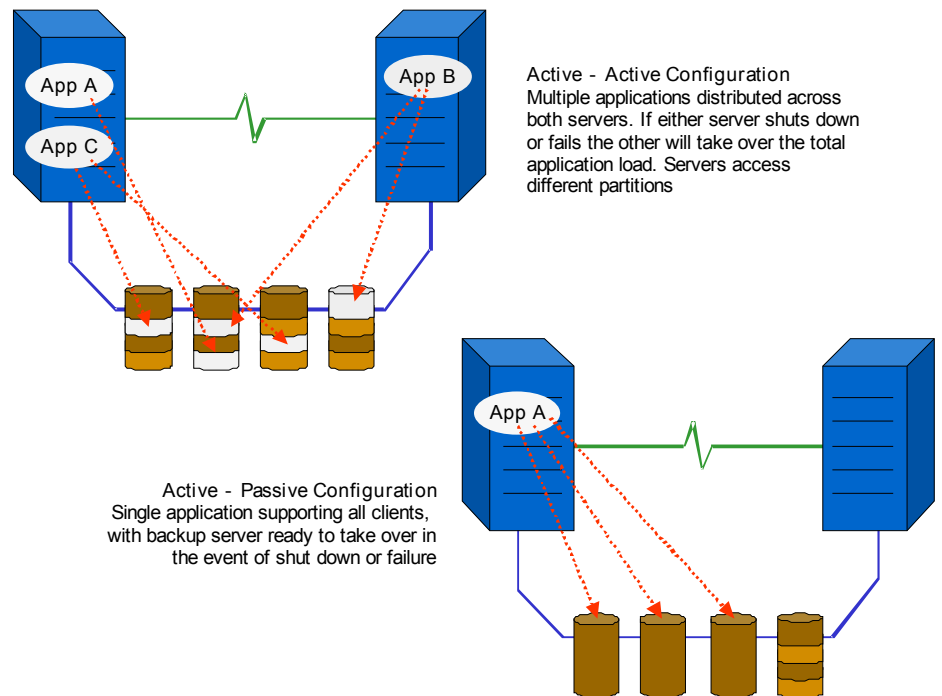


Figure 2 - Active/Active and Active/Passive Configurations

Cluster communication

As described earlier, Cluster Manager configurations include 2-8 server systems connected to a shared external storage array. Additionally, the servers are configured on a shared LAN. Cluster Manager software uses these connections for heartbeating between the servers. Each server heartbeats the others with regular short messages to check that they are operating correctly. If a series of retried heartbeats do not receive an acknowledgment then a server will assume that the remote server has failed and trigger a cluster transition, which removes the failed server from the configuration and initiates application service failover.

In most cases it makes sense to configure more than one LAN connection between the servers for additional redundancy. Cluster Manager supports as many LAN connections as are configured.

Quorum Partitions

Cluster Manager software requires two small (10MB) partitions to be created for use as *Quorum* partitions. Only one partition is strictly needed, the other merely acts as a safeguard against partition corruption. The Quorum partition is used to store static Application Service definitions and dynamic service state (such as where services are currently running). It is the only partition that is shared concurrently by all nodes in the cluster. By reading the Quorum partition a server will know which services are defined, their preferred nodes, and on which node they are currently running. A Quorum daemon checks the Quorum partition every few seconds so that changes are rapidly propagated across the cluster.

An important feature of Cluster Manager is that, due to the fully shared nature of the Quorum partition, all servers are always aware of configuration changes. For example, if server A changes a service while server B is shutdown, server B will automatically learn of the changes when it next boots and joins the cluster. In fact, in a 2 node cluster, if server A changes a service while server B is shutdown and then shuts down itself, server B will still be aware of the update even if it is the first server to reboot. A shared Quorum data area allows Cluster Manager to correctly handle all these types of timing-related synchronization issues.

In two node clusters, where there are failure conditions in which it is possible for both nodes to be able to access the storage but be unable to poll each other, the Quorum partition is also used as a inter-node communication mechanism. The Cluster Manager will use the Quorum partition as a tie breaker to identify which node should continue operation in these 'split cluster' situations.

I/O Barriers

A critical design goal of Cluster Manager is to ensure the highest level of data integrity. This means taking great care to ensure that multiple servers do not issue I/Os to the same disk partition at the same time. During normal operation this is achieved by ensuring that only one server mounts a partition at a time; Cluster Manager application control scripts coordinate all mount and unmount operations.

However, failure conditions can occur that, without an I/O barrier, circumvent the Cluster Manager's control of mounting and unmounting partitions. For example, if a server that is running an application were to hang for long enough to expire the Cluster Manager's connection pollers the remaining servers would automatically take over its applications (thereby meeting application availability requirements). If the hung server subsequently recovered from its error condition it would continue to run its applications, unaware that they had been failed-over by another server. This condition would ultimately be detected by the Quorum service poller, through which the recovered server would detect that it should no longer be running any applications. But the detection will take one or two seconds, during which time it is quite possible for a few application I/Os to be incorrectly issued. These could easily be sufficient to corrupt an application database.

The solution to this type of scenario is the I/O barrier. By using an I/O barrier mechanism an unhealthy server can be prevented from spuriously restarting applications that have been failed-over. Cluster Manager uses two methods to create an I/O barrier:

- Watchdog timers. A watchdog timer (either hardware or software) is installed in each server and is used to monitor server operation. If the server fails to activate the watchdog timer correctly the watchdog will automatically trigger a

shutdown/reboot. The watchdog timer is set to a lower value than the Cluster Manager's failover timers, ensuring that a hung server is shutdown before any applications are failed-over. Note that with watchdog timers each server triggers the shutdown/reboot of itself.

- Programmable power controllers. Using cross-coupled or network-based programmable power controllers each server can directly control the system power applied to the servers. Power controllers are connected to each server by an RS-232 serial connection or across a LAN. If a server hangs, failing to respond to any pollers, a remaining server will power cycle it prior to triggering application failover, thereby ensuring that it cannot spring back to life and issue spurious I/Os. Note that, unlike watchdog timers, in a configuration using programmable power controllers each server can trigger the shutdown/reboot of the other servers.

Other cluster products implement I/O barriers using various different techniques. The most common method is to use SCSI Reservations. A SCSI Reservation permits a server to allocate a disk entirely to itself; the disk will not respond to I/O requests from another server. This prevents more than one server issuing I/O to a disk at a time. After a failover the recovery server can break the old reservation and reserve the disk itself. This technique is effective but has a few drawbacks. The main disadvantages are that many storage controllers do not implement SCSI Reservations reliably and that entire disks, rather than individual partitions, are reserved at a time. Reserving entire disks to a single server can significantly reduce the flexibility of application usage in the cluster, especially with today's large RAID arrays. As a result of these (and other) limitations SCSI Reservations are not widely used in modern clustering products, and are not used by Cluster Manager.

Watchdog timers

Cluster Manager supports three types of watchdog timer. The simplest is an entirely software-based watchdog that is driven off the Linux kernel interrupt handling subsystem and controlled by the Cluster Manager's Quorum daemon. This watchdog will detect all hangs except those in the very lowest levels of the kernel, which should be extremely rare.

The Linux kernel also supports a hardware-based NMI (non-maskable interrupt) watchdog that relies on specific server hardware (usually an Intel i810 TCO chipset on the system motherboard). The NMI watchdog hardware will trigger a reboot of the system if it does not detect a steady level of system interrupts occurring.

Lastly, it is possible to configure a traditional hardware watchdog timer. There are a variety available on the market, often as PCI modules with associated device drivers. These devices will force a system shutdown/reboot if their device driver does not regularly reset them.

All of these watchdog mechanisms provide a very robust I/O barrier for the Cluster Manager.

I/O Subsystem Requirements

Cluster Manager configurations support SCSI and Fibre Channel storage subsystems. Fibre Channel is the preferred storage interconnect for medium and large systems due to its robustness, performance and ease of configuration. Fibre Channel configurations can use direct connections or hubs/switches. For smaller systems traditional parallel SCSI provides high performance and is extremely cost effective, although some care must be taken to ensure correct configuration, as described below.

Parallel SCSI Configuration Notes

In many shared-storage clustering products that support parallel SCSI it is common to configure all servers and the external storage array on the same physical bus. This type of configuration is called multi-initiator or multi-host because there is more than one I/O command initiator/host on the bus.

Due to the complexities of SCSI bus cable length and termination rules, multi-initiator configurations are invariably hard to configure correctly. They can also be difficult to repair without shutting the entire cluster down. Additionally, correct handling of SCSI error conditions when there is more than one host on the bus is extremely complex for the Linux SCSI device drivers. These issues are especially true when using commodity, off-the-shelf SCSI host bus adapters (HBAs). Consequently, Cluster Manager does not support multi-initiator SCSI configurations. Instead, parallel SCSI configurations should be configured with external storage controllers that support *multiple, single-initiator* buses. These controllers support two (or more) electrically separate SCSI buses, each connected to a different server. Additionally, these controllers usually offer a range of RAID capabilities. Since each server is connected to a separate SCSI bus the servers can be configured with commodity, off-the-shelf SCSI HBAs and the Linux SCSI device drivers do not have to handle complex multi-initiator error conditions. An additional benefit is that the separate SCSI buses can handle I/O operations simultaneously, improving performance. Example configurations are shown in Figure 3.

To identify Red Hat certified HBAs and external storage controllers refer to the Hardware Compatibility List at <http://hardware.redhat.com> (note that some vendors self-certify their products, so it is necessary to contact them directly for certification information).

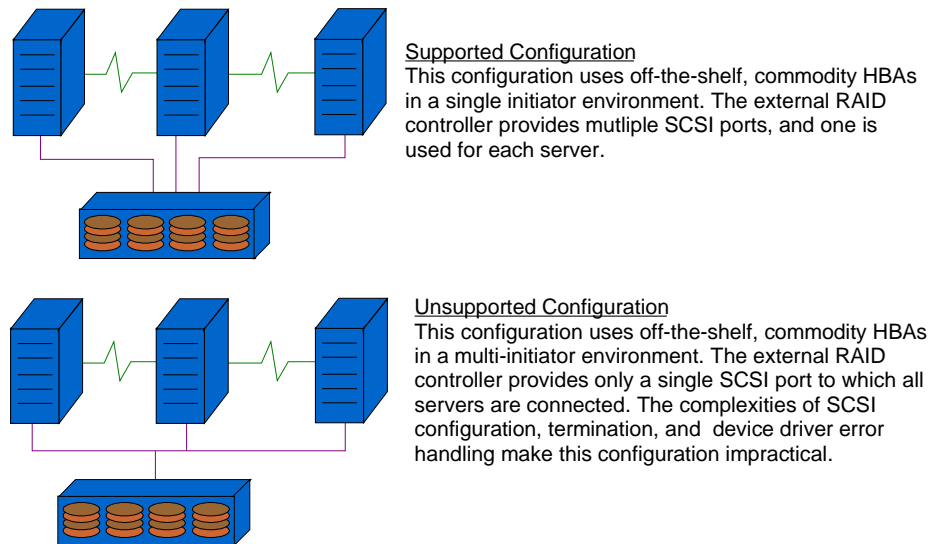


Figure 3 - Supported and Unsupported Parallel SCSI Configurations

RAID considerations

As described earlier, configuring the storage subsystem to use RAID sets is an important part of any high availability solution. Both RAID-1 and RAID-5 will provide excellent availability; RAID-1 generally being higher performance and higher priced than RAID-5.

Consideration should be given to where the RAID capability is implemented within the Cluster Manager configuration. There are two options:

- The Linux host software. This is only suitable for non-clustered storage, so is not supported by Cluster Manager. The reason for this is that the current Linux host RAID software does not coordinate RAID set status across multiple servers. Note, however, that host-based RAID software can be used in Cluster Manager configurations for storage that is not shared, such as boot/root devices and other local storage.
- Shared external controller RAID. This is recommended for all external storage subsystems. Since all RAID functions are performed within the external controller there is no requirement for coordination between the servers or their I/O adapters.

Heterogeneous File Serving

As mentioned in the Applications section, Cluster Manager supports high availability file sharing environments using NFS V2 and V3, and SMB/CIFS (using Samba). Support for these environments is fully contained within Cluster Manager and can be configured quickly and easily.

In NFS environments, file handles (pointers to open files on the NFS server) are maintained in the client. In Cluster Manager configurations all servers are required to have symmetrical I/O systems, which means that all NFS servers can handle a client NFS request correctly. The result is that, apart from possibly noticing a few seconds delay during a failover, client applications will continue to work continuously across a failover (there will be no "stale NFS file handle" errors).

Similarly, in SMB/CIFS environments, state is maintained in the client, so failover of the Samba server is straightforward. How a client application responds to the temporary loss of access to its file share during a failover operation is entirely dependent on the application. Some applications will fail ungracefully, others will ask the user if he/she wishes to retry, and still others will retry quietly (and successfully after the failover is complete). Fortunately this situation is identical in Windows 2000 Advanced Server clusters and Microsoft has made significant progress in making its most popular applications (Microsoft Office, for example) cluster aware.

Service Monitoring

While cluster technology is primarily concerned with ensuring that applications continue to be available in the event of server shutdown or failure, it is just as important to be able to recover from an application software failure. To handle application failures, Cluster Manager supports Service Monitoring.

The Cluster Manager service monitoring feature can be enabled or disabled on a per-service basis. If enabled, the service control script will be called every few seconds. The script will perform application specific tests to check that it is operating correctly. For example the script could check that the correct processes are running and active, or that the appropriate log files are open. Sophisticated scripts could perform database queries or web page accesses. In the event of a failure, the script is called again to restart the service.

In the case of NFS and SMB/CIFS Cluster Manager automatically provides built-in service monitoring.

Management

Cluster Manager software provides two management interfaces for a system administrator:

- A comprehensive command line interface that provides the ability to setup, monitor and control Cluster Manager. Extensive on-line help is also provided.
- A GUI interface that can be used to monitor and control one or several clusters from any remote PC.

Figure 4 shows an example screen-shot taken from the Cluster Manager GUI.

File Cluster Help

Cluster Name: Rainbow Has Quorum

Status: Cluster is running On Member: red.lab.boston.redhat.com

Members

Name	Status
blue	Active
cyan	Active
green	Active
magenta	Active
red	Active
yellow	Active

Services

Enable
 Disable
 Restart
 Properties

Name	State	Member	Last Transition	Monitor Interval	Restarts
greenonly	Running	green	12:50:17 Jul 28	0	0
greenpref	Running	green	12:50:17 Jul 28	0	0
redonly	Running	red	16:49:05 Jul 28	0	0
redpref	Running	red	16:49:05 Jul 28	0	0
service_nfs_1	Running	magenta	16:56:33 Jul 28	0	0
service_nfs_2	Running	magenta	16:57:39 Jul 28	0	0

Figure 4 - Example Cluster Manager GUI window

Summary

The previous pages have outlined several important features of the Cluster Manager. Readers who have experience with other high availability failover clustering products will recognize many of the features and issues raised, and will appreciate how Red Hat engineers have worked to solve them. Red Hat's Cluster Manager engineering team has extensive clustering experience; Cluster Manager implements the latest software technologies, based on an Open Source foundation and designed for commodity hardware.

Red Hat will significantly enhance Cluster Manager in the future. Features that provide improved file system management (such as support for Distributed and Coherent file systems) are in active development today.

Oracle RAC and Cluster Manager

It is worth briefly contrasting Cluster Manager clusters with Oracle RAC clusters. As described earlier, Cluster Manager clusters are suitable for the very wide range of applications that have been designed to run on a single server system. Cluster Manager permits these applications to be deployed, unmodified, in a high availability environment.

Oracle RAC is one of the very few Unix/Linux applications on the market today that supports concurrent read-write access to a single database from multiple servers. This complex technology is suitable for single instance database applications that are too large to be handled by a single server. Using Oracle RAC it is possible to add servers and increase the transaction rate against a single database.

Cluster Manager and IP Load Balancing

Cluster Manager and IP Load Balancing (Piranha) are complementary high availability technologies that can be used separately or in combination, depending on application requirements. Both of these technologies are integrated in Red Hat Cluster Suite.

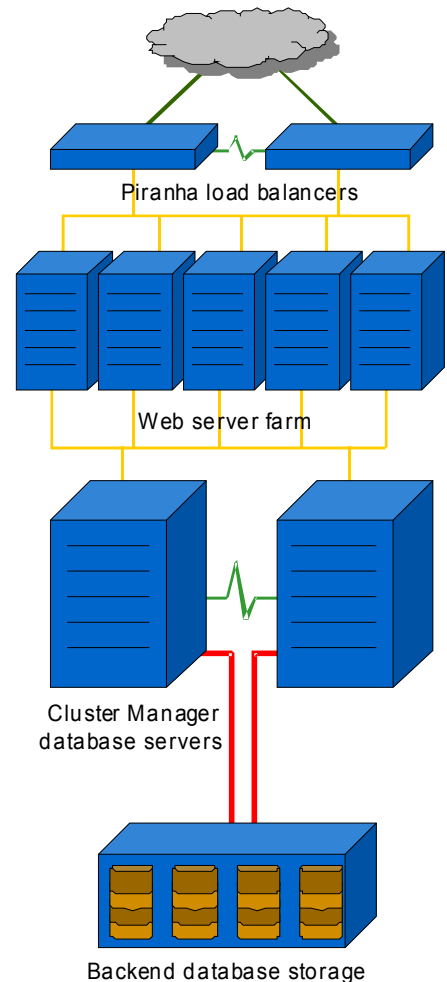
IP Load Balancing is based on the Open Source LVS project, and provides a superset of its capabilities. Notably, IP Load Balancing supports the configuration of a Backup Director, which will take over IP load balancing in the event that the Primary Director fails. Additionally an IP Load Balancing Director will proactively poll its IP clients to ensure that they are active, and will rapidly adjust to client status changes (when they join or leave the load balancing group).

IP Load Balancing technology is used for load balancing incoming IP requests across a group of servers, and is ideal for large-scale Web servers. Availability is enhanced because the configuration continues to operate if any server shuts

down or fails. Because the servers do not utilize any shared storage it is most effective for applications that use static or read-only data. However, when combined with a high availability backend technology, such as Cluster Manager, an extremely effective multi-tier solution with dynamically updated data can be created.

The configuration to the right shows a large, three tier Web server configuration, based entirely on Red Hat Enterprise Linux and commodity hardware. The top tier comprises two IP Load Balancing directors. The directors spread incoming Web requests to the second tier. Running the customer's web application and using primarily static data, these servers will handle the bulk of incoming web requests. For transactional web requests (placing orders, etc.) the second tier web application will issue database requests to the third tier of the system. This is a high availability Cluster Manager configuration running a database application that is accessing a database on the shared storage.

Using this approach the complete site will deliver excellent availability and performance. There is no single point of failure, and adequate compute power to deliver excellent web site performance.



References

For additional information please refer to the following web sites and articles:

- For information on deploying IP Load Balancing please refer to:
<http://www.redhat.com/support/wpapers/redhat/piranha/index.html>
- For additional information on Red Hat products refer to <http://www.redhat.com>
- Red Hat Enterprise Linux Hardware Compatibility List at
<http://hardware.redhat.com>