Delivering High Availability Solutions with Red Hat Linux Advanced Server Version 2.1

Abstract

This white paper provides a technical overview of Red Hat Linux Advanced Server V2.1 high availability features. The paper describes several of the software technologies used to provide high availability and provides several example 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.*

*This white paper is a longer version of a paper originally written by Red Hat Inc. and Dell Computer Corporation covering use of Red Hat Linux Advanced Server with Dell server and storage systems. The original paper provided less technical detail but included example Dell hardware configurations. Please refer to Dell Computer Corp. for copies of the original paper.

Table of Contents

Introduction....3
- Application Support....3
- Performance....3
- Availability....3
- Stability....4
- Support....4

Red Hat Linux Advanced Server High Availability Overview....4

Application Support....5

Cluster Manager Technology....6

Basic Cluster Manager Operation....6

Detailed Operation....7
- Application Services....8
- Application Failover times....8
- Active/Active and Active/Passive....8
- Cluster communication....10
- Quorum Partitions....10
- I/O Barriers....11
- Watchdog timers....12
- I/O Subsystem Requirements....12
- Parallel SCSI Configuration Notes....13
- RAID considerations....14
- Heterogeneous File Serving....15
- Service Monitoring....15
- Management....16

Summary....17
- Oracle RAC and Cluster Manager....17
- Cluster Manager and IP Load Balancing....17
- References....18

Introduction

In May 2002 Red Hat Inc. introduced its latest Linux release designed for use in Enterprise environments - Red Hat Linux Advanced Server V2.1. The Red Hat Linux Advanced Server product significantly extends the capabilities provided by the earlier Red Hat Linux High Availability Server products.

Red Hat Linux Advanced Server V2.1 is based on the Red Hat Linux V7.2 and Red Hat Linux Professional V7.2 products. It has been designed explicitly for the enterprise computing market to deliver superior application support, performance, availability, and scalability. The following paragraphs outline a few of the features that deliver these benefits:

Application Support

Advanced Server is the Red Hat Linux operating system on which Independent Software Vendors (ISVs) will qualify and certify their products. Major commercial applications, such as Oracle and SAP, will qualify and support their products exclusively on Advanced Server.

Performance

Advanced Server is based on Linux Kernel 2.4.9 and includes numerous performance and scalability enhancements. These include support for features such as:

- Asynchronous I/O - applications need no longer pause after issuing I/Os until they are complete.
- Increased SMP granularity - particularly in the SCSI I/O subsystem, permitting increased I/O throughput.
- SMP Scheduler enhancements - support process-CPU affinity, which improves performance by increasing the CPU cache hit rate.

Availability

One of the major new features of Advanced Server, and the focus of this white paper, is the inclusion of a fully featured high availability clustering capability, called Cluster Manager. This provides continued application operation in the event of server shutdown or failure. Additionally, the IP Load Balancing (Piranha) network load balancing feature provided in the earlier Red Hat Linux High Availability Server product is retained and enhanced.

Stability

Advanced Server will provide significantly enhanced system stability for enterprise customers due to the way that Red Hat plans to deliver releases. While consumer market editions of Red Hat Linux are released approximately every six months, Advanced Server releases will be less frequent, approximately every 12-18 months, and will be supported for longer periods of time. Advanced Server releases will be subjected to longer and more rigorous test cycles, greatly reducing the testing requirements of the ISV community. Additionally Red Hat will preserve and stabilize API semantics so that ISVs are not required to make major application modifications with each release. In combination these features will make Advanced Server a very stable platform for long-term enterprise deployments and high quality application support.

Support

To ensure the availability of highly qualified support and maintenance, Advanced Server is only available for deployment as a support and maintenance subscription. In contrast, consumer-focused Red Hat Linux products can be purchased without on-going support and maintenance. Red Hat provides Advanced Server on an annual subscription basis with remedial and errata (patch) services included. There are three variants of Advanced Server, each providing a different level of service, with coverage ranging from basic installation/configuration support, through to enterprise-class service with 24x7 coverage and 1-hour response. All variants include access to Red Hat Network for system maintenance and management services, Red Hat's Internet-based systems management facility. With Red Hat Network, systems are maintained with the latest tested, certified patches, ensuring maximum reliability and security.

Red Hat Linux Advanced Server High Availability Overview

As mentioned in the Introduction, a major feature of Advanced Server in the inclusion of two complementary high availability features:

- Red Hat Cluster Manager, which provides high application availability through failover clustering. Cluster Manager is based on Kimberlite Open Source technology with significant Red Hat enhancements.
- The IP Load Balancing (Piranha) feature provided in earlier Red Hat Linux High Availability Server releases. IP Load Balancing is based on Linux Virtual Server (LVS) Open Source technology with significant Red Hat enhancements.

Cluster Manager and IP Load Balancing are automatically installed as part of the standard Advanced Server installation process. Once the Advanced Server installation has completed they can be quickly configured and deployed. A comprehensive installation and configuration manual is included with Advanced Server.

The remainder of this white paper will primarily concentrate on the new Cluster Manager feature in Advanced Server, 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 running in a cluster - they become high availability applications automatically.

Advanced Server has many new features for enterprise environments, so the few applications that are not suitable for deploying in Cluster Manager configurations can still benefit from the other Advanced Server 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.

redhat

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, Red Hat Linux Advanced Server customers benefit from a clean, new implementation of a well-understood and mature technology.

As a new 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. It 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 array. Both servers are connected to the external storage array and access its disks directly. The Cluster Manager software is used to control access to 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 two servers are also connected to each other using a network or serial interface so that they can communicate with each other. In the event that one of the servers shuts down or fails the other server will detect the event and will automatically start to run the applications that were previously running on the failed server. This migration of the application from the failed server to the remaining server is called failover. Because both servers are connected to the external shared storage the operational server can access the failed server's disk partitions and its applications can continue to operate normally. If necessary the 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 Cluster Manager configuration is shown in Figure 1.

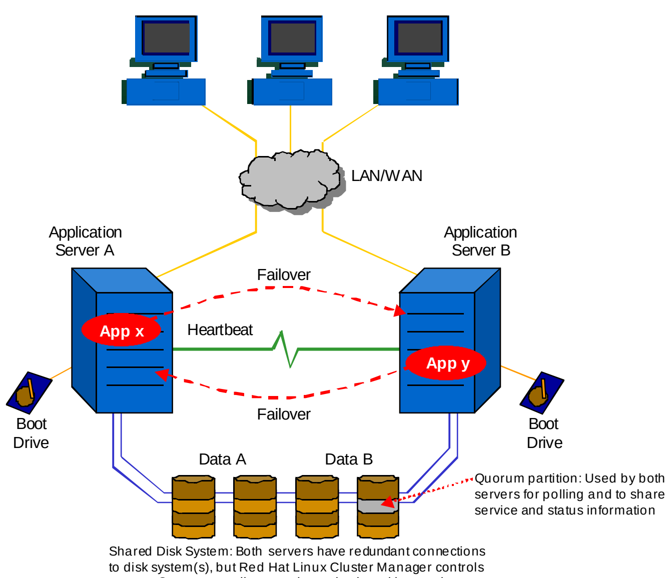


Figure 1 - Typical Cluster Manager Configuration

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 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 the other server - the failover is essentially invisible to it.

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 and Quorum partition polling is 2 seconds. If both pollers fail to receive responses 6 retries are attempted, making the total time before triggering a failover 14 seconds. If the Quorum poller fails to receive a response while the Network poller is functioning normally an additional 11 retries are attempted, making the total time before triggering a failover 24 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 manager using the Cluster Manager administration tools, 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 the two 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 two servers in any way that seems appropriate. Both nodes are actively running a share of the total load. This is called Active/Active clustering, since both servers are indeed active. If one server shuts down, the other server will pick up the load of running all services.

In the event that the customer wishes to run a single large application on the cluster, it must be remembered that both servers cannot access the same disk partition at the same time - because no popular applications available today (apart from Oracle RAC) provide support for concurrent data update from multiple systems. So, it is necessary to restrict these applications to a single server, leaving the other server as a ready-to-go backup in case of failure. This is called Active/Passive operation. This style of operation typically leaves the Passive system idle, which is a waste of valuable computing power. To make the Passive system Active, it is necessary to either find another application 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.

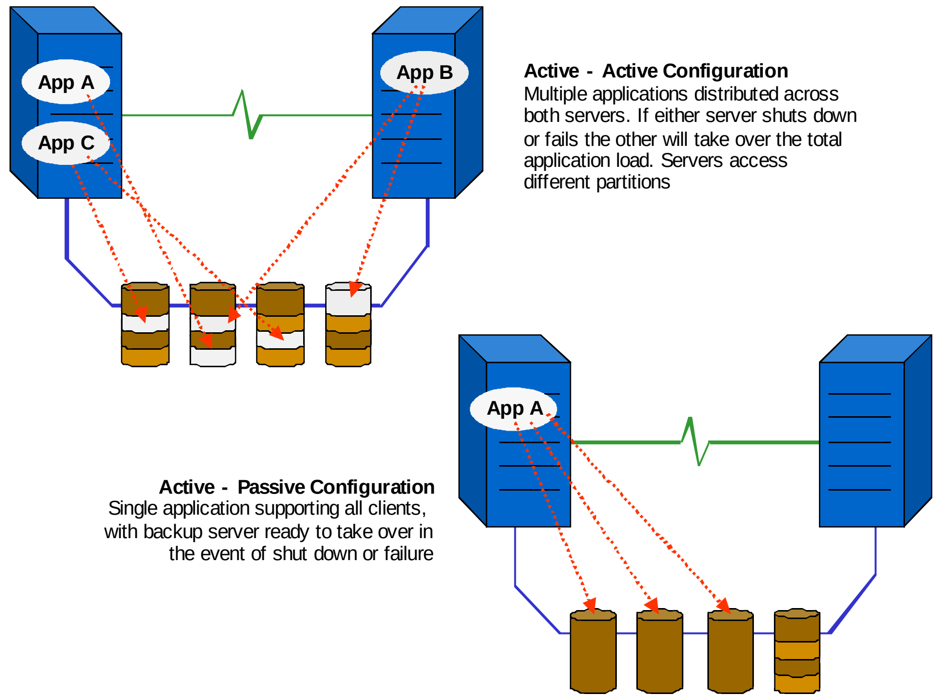


Figure 2 - Active/Active and Active/Passive Configurations

Cluster communication

As described earlier, Cluster Manager configurations comprise two server systems connected to a shared external storage array. Additionally, the two servers are connected by LAN or RS-232 serial cables. Cluster Manager software uses these connections for heartbeating between the servers. Each server heartbeats the other with regular short messages to check that it is operating correctly. If heartbeats do not receive an acknowledgement then a server will use this as an indication that the remote server has failed.

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

Quorum Partitions

While the use of standard LAN and serial communication methods for heart-beating is common among all failover clustering products, Cluster Manager provides an additional, more sophisticated, polling mechanism. This mechanism uses a Quorum partition that is configured on the shared storage subsystem. Both systems read and write the Quorum partition on a regular basis, indicating their status, service definitions, and which services they are running. (Note that in practice, to increase system resiliency, two Quorum partitions are configured, a primary and a backup. Quorum partitions are small, at 10MB each.)

Cluster Manager uses the Quorum heartbeat as the final arbiter of server operation, that is, if a server does not update its Quorum data regularly then it is considered to have failed, and application failover will be initiated. Even if all LAN and Serial connections fail the cluster will continue to operate normally if Quorum heartbeats are still operational. As such, the LAN and Serial connections are used as accelerators for detecting a failed server (since Quorum partition updates may be delayed due to heavy I/O loads or bus re-initializations). The Cluster Manager software will use the status of network pollers to determine whether a failed Quorum update indicates a failed server or simply a very busy server.

This mechanism differs slightly from Microsoft Windows 2000 Advanced Server, which also uses LAN and Serial heartbeats with a single Quorum partition in a SCSI Reservation model. During cluster state changes the two servers race to reserve the Quorum partition. In the event that the two servers cannot poll successfully across the LAN and Serial connections, the winner of the race takes over the applications of the loser. The net effect is that both servers will not continue to operate if all the LAN and Serial heartbeat connections fail.

An important feature of Cluster Manager is that, due to the fully shared nature of the Quorum partition, the two 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, 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 disconnect.

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 both servers do not issue I/Os to the same 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 server would automatically take over its applications, thereby providing the necessary application availability. 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 the other server. This condition would ultimately be detected by the Quorum polling mechanism, 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 programmable power controllers each server can directly control the system power applied to the other server. 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, the recovery server will power cycle it prior to taking over any of its applications, 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 server.

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 both 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 when using these types of HBA. However, some vendors produce purpose-designed HBAs that not only provide multi-initiator RAID capabilities but also mask all SCSI bus error and initialization issues from the operating system.

Cluster Manager supports these "intelligent" RAID HBAs in multi-initiator configurations. To identify specific supported HBAs refer to the Hardware Compatibility List at http://hardware.redhat.com (this is especially important when selecting a RAID HBA).

The diagrams in Figure 3 show multi-initiator configurations with off-the-shelf HBAs (not supported) and with intelligent RAID HBAs (supported). Note that, while intelligent RAID HBAs are highly cost effective, they generally provide lower performance than external RAID systems, due to complexities in maintaining cache coherence.

Given sufficient budget, 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, and most products 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 of using external RAID controllers is that optimum performance is realized because there are no cache coherence issues and the separate SCSI buses can handle I/O operations simultaneously. While more expensive, these configurations are simpler to install and maintain, and they provide the best parallel SCSI performance. An example configuration is shown in Figure 3.

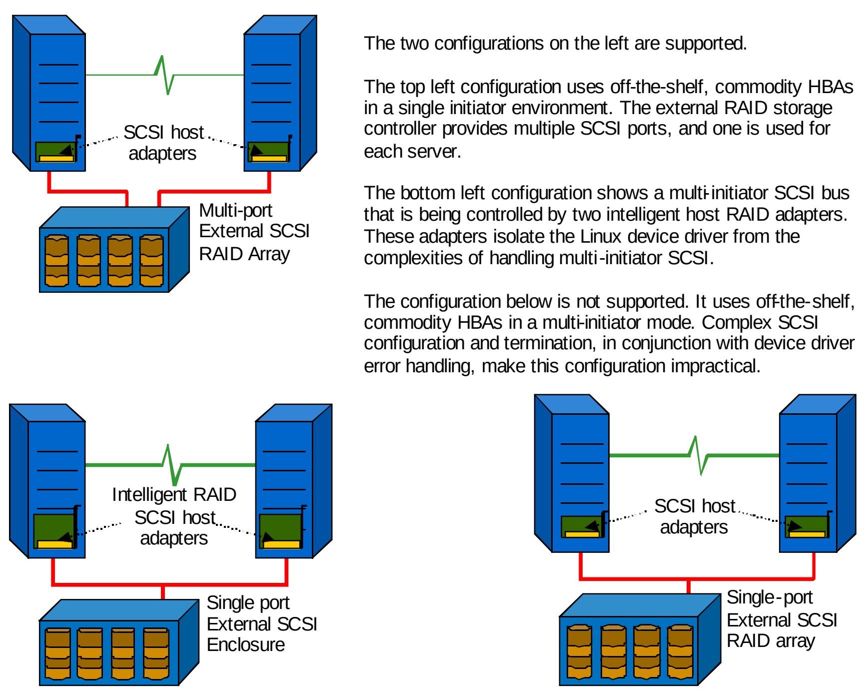


Figure 3 - Supported and Unsupported Parallel SCSI Configurations

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 three 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.
- The server I/O adapter. This is the most cost effective RAID option for a Cluster Manager configuration, but requires the two RAID adapters (one in each server) to coordinate RAID set status between themselves. Adapters that support this capability are now becoming available on the market (see Parallel SCSI Configuration Notes above).
- Shared external controller RAID. This is a more expensive RAID solution and is suitable for larger 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 both servers are required to have symmetrical I/O systems, which means that the NFS server on either server 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 to 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 Java-based Browser interface that can be used to monitor one or several clusters from any remote PC. For security reasons, the Java GUI cannot be used to setup or control Cluster Manager.

Figure 4 shows an example screenshot taken from the Cluster Manager GUI.

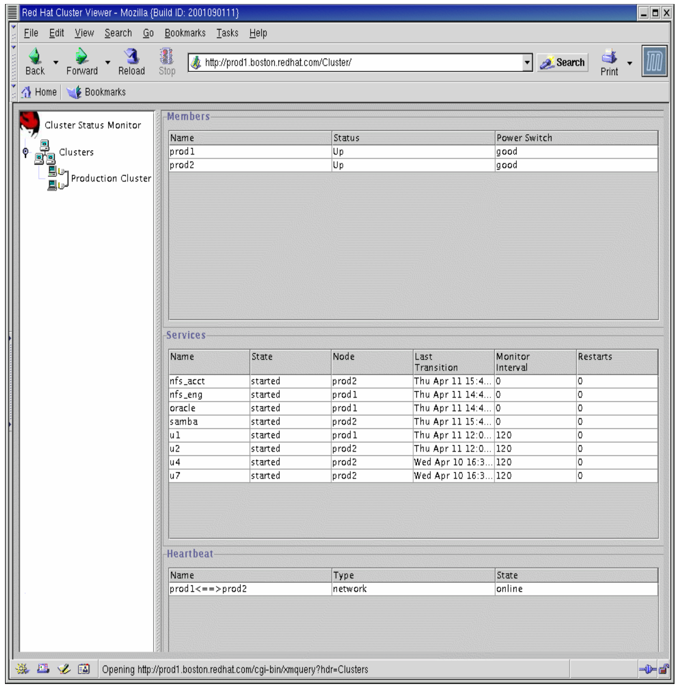


Figure 4 - Example Cluster Manager GUI window

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. The Red Hat Linux 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 increased scalability (such as support for more than two servers) and greatly 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 (more than two servers are supported) and increase the transaction rate against a single database.

Cluster Manager and IP Load Balancing

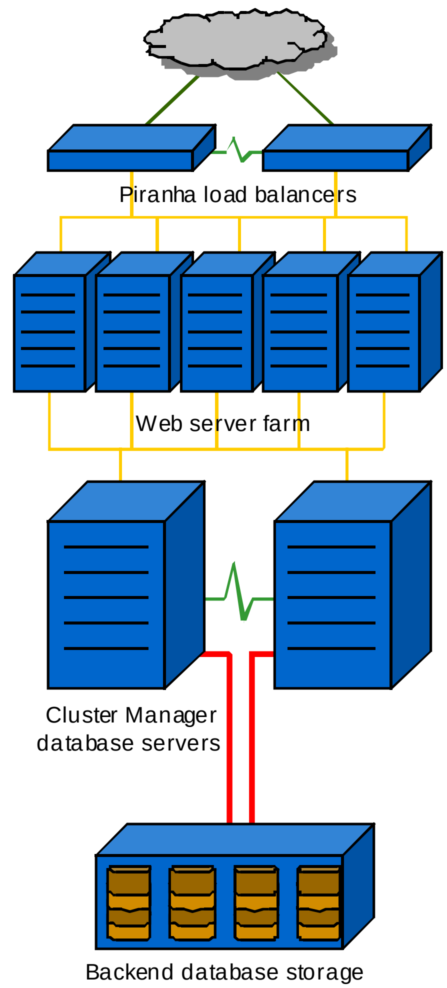
Cluster Manager and IP Load Balancing (Piranha) are complementary high availability technologies that can be used separately or in combination, depending on application requirements. Both of these technologies are integrated in Advanced Server.

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. Due to the fact that the servers do not utilize any shared storage it is most effective for static or read-only applications. 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 Linux Advanced Server 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 Linux Advanced Server Hardware Compatibility List at http://hardware.redhat.com