Reliant® HA Release 1.1.4

For UnixWare® 7

A SCO Technical White Paper

October 2004

INTRODUCTION

Mission Critical Needs

If information is the lifeblood of business, then the ability to access and use this information on demand is like oxygen in the lungs! The rush towards downsizing from expensive proprietary systems to commodity hardware has led to the widespread acceptance of client-server computing on open systems. The growth in client-server computing has meant that more and more businesses now find themselves entrusting *mission critical* information to these systems. This has led to the demand for systems solutions offering high reliability with highly available access to information at all times.

THE COST OF DOWNTIME

Reasons for downtime

All business systems, even the most critical ones, can experience downtime for a variety of reasons. The reasons can include:

- CPU failures
- Memory board failures
- System panics or hangs
- Failed disk devices
- Administrator errors
- I/O component failures

Impact

The cost of downtime can be incredibly high. Reports have shown that even a small two server system with less than 200 clients experience average server down time costing in excess of $200,000 a year.[1] Each client failure alone can contribute another $1,000 to the total cost of the lost time. Many businesses seriously underestimate the cost of downtime when considering their system implementations. In addition to the financial costs, other consequences of downtime include:

- Loss of productivity
- Loss of critical data
- Loss of time
- Direct commercial impact, especially in areas like electronic commerce, retail, telemarketing or financial applications

Systems today must offer unprecedented availability for businesses to avoid the consequences of unexpected downtime.

[1] *What is the True Cost of Network Downtime*?, Gartner Group, Research Note

HIGH AVAILABILITY AND FAULT RESILIENCE

Levels of Availability

System availability is often described in terms of levels of availability. To obtain a minimum level, businesses can invest in single systems with UPS support or on-line disk replacement. Further protection for disk subsystems can be added by software storage management tools such as the Disk Mirroring or On-line Data Manager add-ons for UnixWare®, or by the use of RAID subsystems.

Expensive HW Dilemma

Unfortunately, to achieve the highest levels of availability, businesses have previously had to invest in expensive proprietary solutions from high-end system vendors. These systems offer fault-tolerance, primarily through the use of redundant hardware that spends most of its time sitting idle (in hot-standby mode) waiting to take over only during the instance of a system failure. While such systems are ideal for implementations such as banking or airline reservations, they also carry a significant and prohibitive hardware price premium which makes them unsuitable for many other types of businesses.

Fault Tolerance vs. Fault Resilience

In fact, the majority of businesses that rely on information access do not require the maximum level of availability offered by fault tolerant systems. Often, these systems can be reliably protected by a less expensive *fault resilient* solution. Such a solution can be implemented by combining a business's existing servers into a single highly available *cluster*.

Multisystem Parallelism

The cluster can provide a reasonably priced high availability solution, offering protection for existing mission critical applications. However, it can also offer even greater value through the support of distributed applications. These applications usethe combined processing power of systems in the cluster through shared access to data storage This can be of particular benefit to applications such as DBMSs.

Today many businesses are building data warehouses. These warehouses often incorporate greater amounts of new data that is often integrated from multiple sources, including external data. As the size of these databases increase, larger archival storage and more processing against the data is required.

CLUSTERED SOLUTIONS

Clusters

A LAN-based cluster of systems, each with its own processor, memory, and peripherals can appear as a single server to users and applications. It provides a cost effective way to gain features and benefits that to date have only been found on more expensive legacy systems. Figure One below shows a simple 2 node cluster.

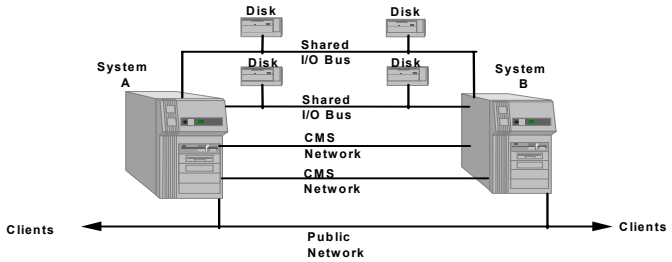

Figure One - A simple two-node cluster.

RELIANTHA 1.1.4

UnixWare 7 High Availability Clustering

UnixWare 7is an open, standards-based operating system that supports the simultaneous execution and interaction of both traditional UNIX® applications along with Linux® applications. UnixWare already provides performance, scalability and other features important for business critical application environments. Reliant®HA 1.1.4 is an add-on Optional Service for UnixWare 7.1.4, UnixWare 7.1.3, UnixWare 7.1.2 (Open UNIX 8) or UnixWare 7.1.1 systems that offers high availability and continued access to system services and applications through fail-over. Designed as a high-availability alternative to fault tolerant machines, ReliantHA is ideal for environments that need to ensure rapid recovery from failure using failover. There is also no cost associated with an idle standby system because during normal operations, all systems in the cluster can be fully utilized.

Configuration Monitor

High availability in ReliantHA is provided through the Configuration Monitor, a process thatruns on all machines in the cluster. System resources such as file systems and applications are configured to be available on a particular system, and in the event of a component failure the Configuration Monitor takes corrective action. This may involve making the resource available from another system in the cluster. Component failures are identified by small processes called *detectors* which report component status to the Configuration Monitor. Failure of complete systems is identified by monitoring messages (*heartbeats)* which are passed between the systems in a cluster using a redundant private network.

Cluster Management

ReliantHA includes graphical tools for managing the cluster and systems within the cluster, and provides for user-configurable detection of and recovery from hardware and software failures in the cluster. The system's graphical cluster Configuration Manager provides a single point of cluster-wide administration. It can be run from any node's console in the cluster and displayed from any appropriately connected X display device.

Only minor manual editing of files or use of command line tools is necessary for cluster configuration management. One time kernel configuration is handled via installation screens. All subsequent management can be done via the graphical tools.

STANDARD FEATURES

Features

Features of ReliantHA include:

- Ability to configure, monitor and manage clusters from 2 to 4 nodes
- Automatic failover support for resources ranging from applications to whole systems
- Automatic failover for a mix of UNIX and Linux applications is supported on UnixWare under the Linux Kernel Personality (LKP)
- Easy to use, graphical Configuration Manager
- Tools for creating custom configurations, with example detectors and failover scripts provided for SCO*office* Mail Server, NFS, webservers, database, and generic applications
- A dedicated channel to monitor the state of the cluster, allowing cluster recovery in the event of a node failure
- Support for commodity Intel Pentium class and compatible single processor and SMP nodes

DESIGN HIGHLIGHTS

Design Benefits

ReliantHA has the capabilities of more expensive proprietary high availability systems delivered in a cost-effective, standards-based shrink-wrap package:

Multi-threaded Optional Service add-on product for UnixWare systems

- TCP/IP Ethernet client network support
- Industry Standard bus support, including SCSI, SCSI-2, and SCSI-3 and Fiber Channel peripherals for shared disk I/O configurations
- Private Ethernet network connections for cluster monitoring

RELIANTHA COMPONENTS

Intelligent Monitoring

The Configuration Monitor is a state engine that responds to changes in the configuration of the cluster, ensuring a high level of availability. For each resource that is being managed, ReliantHA runs processes to check the resource status, using recovery scripts to restart the resource, switch between alternative data paths, or migrate the resources to another system in the event of a service failure.

ReliantHA also communicates with the Configuration Monitors of other systems in the cluster using monitor messages. A missing message is used to detect catastrophic failure, initiating recovery of the lost services by the surviving systems.

Configuration Files

The Configuration Monitor and the Configuration File used in ReliantHA reside on every system in the cluster. Configuration Files contain the following:

- Definitions of all resources in the cluster
- Interrelations between resources in the cluster
- Detector specifications (see below) and associated resources
- Association of resources and respective transition scripts

Identical copies of the Configuration Monitor on each system communicate with each other and exchange the monitoring messages via the private Ethernet network.

Transition Scripts

Transition Scripts are programs that are associated with each type of resource in a cluster. The scripts are executed in response to a change in a resource's state

Daemons

ReliantHA employs a communication daemon to exchange monitoring messages and status information between the systems in the cluster. A second GUI daemon handles communication to the graphical cluster Configuration Manager.

Detectors & Terminators

Detectors are scripts or C and C++ programs that report on the current state of specified cluster resources. This information is reported to a detector report queue at user-defined intervals. ReliantHA 1.1.4 comes with pre-defined detectors for common system resources, however, additional detectors can be written by the user to meet the needs of their configuration. Terminators shut down a node in the cluster should a severe fault occur.

Customizable

The Configuration Monitor comes with pre-defined recovery schemes to deal with hardware and software failures for a number of common services. Customers also have the flexibility to define their own switchover scheme and alter hardware or software failure detection and action policies by modifying the files and scripts that the Configuration Monitor uses. Thus, the Configuration Monitor can be customized to accommodate almost any computing and applications environment.

Node

A single computer system in the cluster. Each system has its own CPU(s) and memory resources. A node may be a uniprocessor or a shared memory multiprocessor (SMP) system. Each system runs a copy of the UnixWare operating system, the ReliantHA software and one or more applications.

Shared Disks

Disks shared among all nodes in the cluster.

Private Networks

Networks used exclusively for cluster communication. Only nodes that are members of the cluster are connected to the private network.

Public Networks

Network used by database clients to connect with the database.

OPERATIONAL OVERVIEW

Sharing resources

A ReliantHA cluster can support shared disk configurations. In the case where the application is not cluster-aware, all nodes can have access to the same storage subsystem but not to the same partitions. For shared data access, a distributed lock manager or a cluster-aware application is required. When sharing a resource such as a disk block, processes need a mechanism to synchronize modification to the resource, so that the resource is not placed in an inconsistent state.

Monitoring "Heartbeat" Messages

The private network is also used to determine the health of a node. This is accomplished by sending monitoring messages between the nodes in the cluster. A monitor message must be received from a node every few seconds. To guarantee that such a message is not missed due to network traffic, the network should be a private network. If a monitor message is not received, the remaining nodes co-ordinate a kill of the failed node using the serial lines connected between every node in the cluster.

Preventing Network Partitioning

Killing a node prevents a condition known as network partitioning, which can occur when nodes form groups of two or more sub-clusters. Each sub-cluster assumes that all other nodes outside of the sub-cluster have failed. Sub-clusters form when nodes are still up and running, but unable to communicate. If the sub-clusters continue to operate on the shared disks, data corruption occurs.

CLUSTER COMMUNICATION

Redundant Communications

Each system is connected to others in the cluster via a dedicated redundant private network using industry standard hardware. This allows each system to communicate its status. To eliminate any single points of failure, all inter-system connections are physically redundant. So, for every connection there is more than one physical communication path that goes from one system to another within the cluster. Should any one connection fail, then communications will continue via other routes and continue to provide service to the cluster. All bandwidths are used on both the primary and redundant connections.

The use of redundant connections requires the use of at least two communication paths per system. Thus, each system in the cluster will typically employ two Ethernet ports. There is no technical limitation on the number of physical connections between systems other than the physical limitations of the system's cabinetry. The addition of more than

four connections per system will provide additional redundancies and bandwidth to inter-system communication.

Low-Latency Transport

The communication protocols used with ReliantHA are TCP/IP for the cluster monitoring traffic. “Public” traffic (unrelated to ReliantHA monitoring) can be used on this network but will introduce the likelihood of performance degradation. Whenever possible, it is recommended that the inter-system network not be used for public traffic.

Virtual Network

The monitor networks use what is termed a Virtual Network to transmit data over more than one connection transparently. Virtual Networks are used to make multiple physical network connections appear as one so that data can be transmitted via more than one physical line. It automatically load balances message packages down redundant paths fully utilizing all bandwidth. It also facilitates switchover should one line fail.

In the cluster, Virtual Network is used strictly for inter-system communications, transmitting state and monitoring information for configuration management. It must be running on both ends of the communications line to function so it cannot be used for connections to systems outside of the cluster. A MAC (Media Access Control) Switch pseudo network device driver is what provides the virtual network view of two or more network connections. This driver is how the Virtual Network is provided, selecting the physical connections to transmit message packets and the alternate use of connections to optimize communications.

Supported NICs

Currently, certified interface cards for use with ReliantHA are available from Intel® and other vendors. For the latest list of certified interface cards see SCO's Certified and Compatible Hardware web page which may be accessed at the following URL: http://www.sco.com/chwp.

DISK STORAGE AND I/O SUBSYSTEM

Disk Mirroring

Disk mirroring should be employed to eliminate single points of failure in the cluster. This can be implemented for non-shared disks via the volume management capabilities of the SCO's Online Data Manager add-on product. Currently, hardware RAID solutions are the only solutions available for shared disks. All disks in the cluster should be mirrored and configured redundantly with separate controllers for both the primary and mirrored disks. Note that root disks do not failover in the event of a system failure and

cannot be on shared disks in the cluster. They can, however, be mirrored using Online Data Manager.

SCSI

ReliantHA 1.1.4 uses a SCSI-based I/O subsystem that supports either eight addresses per SCSI chain with standard SCSI or 16 addresses per chain with wide SCSI2. It is possible to mix standard SCSI and wide SCSI controllers and host adapters in the cluster, provided they do not reside on the same SCSI bus. Each host and each device use one address so the number of devices supported per SCSI chain depends on the cluster configuration and switchover schemes. The greater the number of switchover options that are required, the greater the number of addresses that will be needed. Due to SCSI cable length restrictions, each system in the cluster must be no more than 20 feet apart from each other. Currently, certified HBAs for use with ReliantHA are available from Adaptec, Qlogic and other vendors. For the latest list of certified adapters see SCO's Certified and Compatible Hardware web page which may be accessed at the following URL: http://www.SCO.com/chwp.

RAID

Hardware RAID solutions are also available for use with clusters. For the latest list of certified solutions see SCO's Certified and Compatible Hardware web page which may be accessed at the following URL: http://www.SCO.com/chwp.

A TECHNICAL DISCUSSION

Detection and Failover

The primary functionality offered in the high availability component of the system is the ability to fail over arbitrary services (up to and including an entire system) within a luster of 2-4 nodes.[2] This fail over can happen automatically or can be initiated manually e.g. for the purposes of maintenance on a particular resource.

In Figure Two below, a highly redundant hardware architecture for a 2 node cluster. The two systems are connected to each other using redundant private networks for the Configuration Monitoring System (CMS) to monitor the system and also share redundant I/O buses. In the high availability solution, the systems are not both accessing the same disks. System A “owns” Disks J and L and System B “owns” disks K and M. For maximum reliability and availability, each system would be using the Hardware RAID to mirror the data on the two disks it owns across the two IO controllers. For example, Disk J and Disk L would be configured as mirrors of each other. Note

[2] Though most of the discussed cases in this document involve failover between systems, there is nothing precluding detection and failover of resources within a single system.

that ReliantHA itself does not require mirroring per se, but it is advisable in order to eliminate single points of failure. Both System A and System B are connected to the same public network to allow client access.

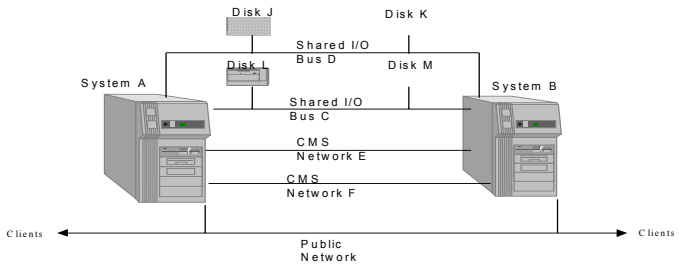

Figure Two – A two-node cluster with redundant hardware.

The following hypothetical example illustrates this process. If System A were to go down, System B would assume ownership of the mirrored disk pair J and L. System B would also assume the network address of System A. Finally, the applications previously running on System A would be started instead on System B. Assuming that the starting time for the application is negligible, the whole failover process should take less than 3 minutes. In this way, networked clients would see the failure of System A as a temporary network disconnect. They would then reconnect to what they thought was System A and continue work. However, there may be, of course, performance differences once all the applications that were running on System A were resumed on System B, depending on what other applications were running on System B and its relative power. If this is a concern, non-critical applications on System B could be shutdown automatically as part of the failover process to help address the performance issue. Meanwhile, work could be done to restore System A and, when it was back on line, its applications and services could be moved back from System B. Providing the applications on System B are being monitored by ReliantHA, this process is symmetrical, that is, if system B fails, they will be maintained on system A.

Software Architecture

The software architecture to support the failover processes is shown in Figure Three on the following page. The heart of the cluster software on each system is the Configuration Monitor. This process relies on a set of detector processes, each of which monitors a separate resource on the system.

Typically, there would be detectors for the application(s) of interest on the system as well as for the various software and hardware components of the system that may be failed over. When a detector identifies a failure, it notifies the Configuration Monitor through a messaging queue. The Configuration Monitor, using information from user supplied configuration files, initiates the appropriate failover or recovery actions.

Graphical administration is performed through the Reliant Cluster Visual Manager (RCVM) which communicates with the Configuration Monitor through a GUI daemon (GUID).

The Configuration Monitor communicates with other nodes on the system through the Communication Daemon (COMMD) processes which interface to the networking subsystem. Just above the actual hardware drivers in the networking stack is the Virtual Networking Driver. Its function is to load balance ReliantHA's private networking communications across the redundant physical networks available to it.

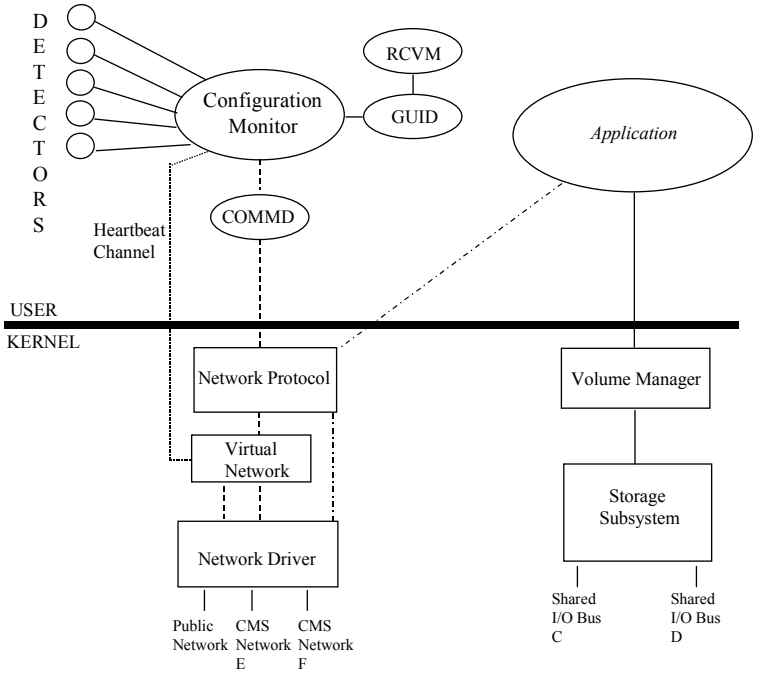

Figure Three. Software architecture for failover.

The Configuration Monitor also has a private channel of communication with the Virtual Networking Driver which allows it to generate the monitoring messages for the system. This "heartbeat" is cluster monitored by the other systems in the cluster and its absence is taken as an indication that the system has failed and that the other systems in the cluster should take over for it according to the user-defined configuration files. Meanwhile, the application is communicating across the network to any clients and is

accessing data through the storage subsystems and shared I/O buses on the disks owned by this system.