VERITAS

VERITAS WHITE PAPER

VERITAS Cluster Server™ v2.0

Technical Overview

VERITAS

Table of Contents

Executive Overview

This document is intended for system administrators, system architects, information technology (IT) managers and other IT professionals interested in increasing application availability through the use of VERITAS Cluster Server™. This white paper will describe the terminology and technology associated with VERITAS Cluster Server.

Why VERITAS Cluster Server?

VERITAS Cluster Server is a state-of-the-art, multiplatform High Availability package currently shipping on Sun Solaris, HP/UX and Windows NT 4 and planned on Windows 2000, AIX and Linux. Key features of VERITAS Cluster Server include:

- Extremely scalable (up to 32 nodes in a cluster). This allows building larger clusters to support increasingly complex applications, as well as reducing total number of "spare" systems needed.
- Supports multiple environments. Windows NT, Solaris and HP/UX are supported. Support for additional operating systems is planned. (Individual clusters must be comprised of the same operating system family. Clusters of multiple operating system types can all be managed from the VERITAS Cluster Manager™ console.)
- Reduces administration cost. Because VERITAS Cluster Server is a multiplatform solution, administrators only need to learn one clustering technology to support multiple environments. A single group of administrators can become experts about High Availability administration.
- Provides a new approach to managing large server clusters. Through a Web- or Java-based graphical management interface, administrators can manage large clusters automatically or manually, and migrate applications and services among them.
- Supports all major third-party storage providers and works in small computer system interface (SCSI), network attached storage (NAS) and storage area network (SAN) environments.
- Provides flexible failover possibilities: one-to-one, any-to-one, any-to-any and one-to-any failovers.
- Optimizes availability based on dynamic choice of failover node. Target node can be chosen based on system load, number of service groups online (round robin) or defined priority.
- Supports parallel and failover service groups.
- Integrates seamlessly with other VERITAS products to increase availability, reliability and performance.
- Provides a simple method to develop support for new applications via agents.

VERITAS Cluster Server Technology Concepts

VERITAS Cluster Server is a very powerful application availability package with many advanced features. By breaking down the technology into understandable blocks, it can be explained in a simple fashion. The following section will describe each major building block in a VERITAS Cluster Server configuration. Understanding each of these items, as well as interaction with others, is key to understanding VERITAS Cluster Server. The primary items to discuss include the following:

- Clusters
- Resources and resource types
- Agents
- Agent classifications
- Service groups
- Resource dependencies
- Heartbeat

Clusters

A single VERITAS Cluster Server cluster consists of multiple systems connected in various combinations to shared storage devices. VERITAS Cluster Server monitors and controls applications running in the cluster and can restart applications in response to a variety of hardware or software faults.

A cluster is defined as all systems with the same cluster identification and connected via a set of redundant heartbeat networks.

Clusters can have from one to 32 member systems, or "nodes." All nodes in the cluster are constantly aware of the status of all resources on all other nodes. Applications can be configured to run on specific nodes in the cluster. Storage is configured to provide access to shared application data for those systems hosting the application. In that respect, the actual storage connectivity will determine where applications can be run. Nodes sharing access to storage will be "eligible" to run an application. Nodes without common storage cannot fail-over an application that stores data to disk.

Within a single VERITAS Cluster Server cluster, all member nodes must run the same operating system family. For example, a Solaris cluster would consist of entirely Solaris nodes; likewise with HP/UX and NT clusters. Multiple clusters can be managed from one console with the Cluster Server Cluster Manager.

The cluster manager allows an administrator to log in and manage a virtually unlimited number of NT and UNIX VERITAS Cluster Server clusters, using one graphical user interface (GUI) and command line interface (CLI). The common graphical user interface and command line interface is one of the most powerful features of VERITAS Cluster Server.

Resources and Resource Types

Resources are hardware or software entities, such as disks, network interface cards (NICs), IP addresses, applications and databases which VERITAS Cluster Server controls. Controlling a resource means bringing it online (starting), taking it offline (stopping) as well as monitoring the health or status of the resource.

Resources are classified according to *types,* and multiple resources can be of a single type. For example, two disk resources are both classified as type "disk." How VERITAS Cluster Server starts and stops a resource is specific to the resource type. For example, mounting starts a file system resource, and an IP resource is started by configuring the IP address on a network

interface card. Monitoring a resource means testing it to determine if it is online or offline. How VERITAS Cluster Server monitors a resource also is specific to the resource type. For example, a file system resource tests as online if mounted, and an IP address tests as online if configured. Each resource is identified by a name that is unique among all resources in the cluster.

VERITAS Cluster Server includes a set of predefined resources types. For each resource type, VERITAS Cluster Server has a corresponding *agent.* The agent provides the resource type specific logic to control resources.

Agents

The actions required to bring a resource online or take it offline differ significantly for different types of resources. Bringing a disk group online, for example, requires importing the disk group. Bringing an Oracle database online would require starting the database manager process and issuing the appropriate startup command(s) to it. From the cluster engine's point of view, the same result is achieved — making the resource available. The actions performed are quite different, however. VERITAS Cluster Server handles this functional disparity between different types of resources in a particularly elegant way, which also makes it simple for application and hardware developers to integrate additional types of resources into the cluster framework.

Each type of resource supported in a cluster is associated with an agent. An agent is an installed program designed to control a particular resource type. For example, for VERITAS Cluster Server to bring an Oracle resource online, it does not need to understand Oracle; it simply passes the online command to the Oracle agent. The Oracle agent knows to call the server manager and issue the appropriate startup command. Because the structure of cluster resource agents is straightforward, it is relatively easy to develop agents as additional cluster resource types are identified.

VERITAS Cluster Server agents are "multithreaded." This means a single VERITAS Cluster Server agent monitors multiple resources of the same resource type on one host. For example, the disk agent manages all disk resources. VERITAS Cluster Server monitors resources when they are online as well as when they are offline (to ensure resources are not started on systems where they are not supposed to be currently running). For this reason, VERITAS Cluster Server starts the agent for any resource configured to run on a system when the cluster is started.

Classifications of VERITAS Cluster Server Agents

Bundled Agents

Agents packaged with VERITAS Cluster Server are referred to as bundled agents. They include agents for disk, mount, IP and several other resource types. For a complete description of bundled agents shipped with VERITAS Cluster Server, see the bundled agents guide.

Enterprise Agents

Enterprise agents are separately packaged agents that that can be purchased from VERITAS to control popular third-party applications. They include agents for Informix, Oracle, DB2, VERITAS NetBackup™ and Sybase. Each enterprise agent ships with documentation about the proper installation and configuration of the agent.

Storage Agents

Storage agents provide control and access to specific kinds of enterprise storage, such as the Network Appliance Filer series and the VERITAS ServPoint™ (NAS) Appliance.

Custom Agents

If a customer has a specific need to control an application that is not covered by the agent types listed above, a custom agent must be developed. VERITAS Enterprise Consulting Services provides agent development. Or customers can write their own. For more information, refer to the *"VERITAS Cluster Server Agent Developers' Guide,"* which is part of the standard documentation..

Service Groups

A service group is a set of resources working together to provide application services to clients.

For example, a Web application service group might consist of:

- Disk groups on which the Web pages to be served are stored
- A volume built in the disk group
- A file system using the volume
- A database whose table spaces are files and whose rows contain page pointers
- The network interface card or cards used to export the Web service
- One or more IP addresses associated with the network card(s)
- The application program and associated code libraries

VERITAS Cluster Server performs administrative operations on resources, including starting, stopping, restarting and monitoring at the service group level. Service group operations initiate administrative operations for all resources within the group. For example, when a service group is brought online, all the resources within the group are brought online. When a failover occurs in VERITAS Cluster Server, resources never fail-over individually – the entire service group that the resource is a member of is the unit of failover. If there is more than one group defined on a server, one group may fail-over without affecting the other group(s) on the server.

From a cluster standpoint, there are two significant aspects to this view of an application service group as a collection of resources:

- If a service group is to run on a particular server, all of the resources it requires must be available to the server.
- The resources comprising a service group have interdependencies; that is, some resources (e.g., volumes) must be operational before other resources (e.g., the file system) can be made operational.

Resource Dependencies

One of the most important parts of a service group definition is the concept of resource dependencies. As mentioned above, resource dependencies determine the order specific resources within a service group are brought online or offline when the service group is brought offline or online. For example, a VERITAS Volume Manager™ Disk Group must be imported before volumes in the disk group can be started and volumes must start before file systems can be mounted. In the same manner, file systems must be unmounted before volumes are stopped and volumes stopped before disk groups deported.

Types of Service Groups

VERITAS Cluster Server service groups fall in two categories, depending on whether they can be run on multiple servers simultaneously.

Failover Groups

A failover group runs on one system in the cluster at a time. Failover groups are used for most application services, such as most databases, network file system (NFS) servers and any other application not designed to maintain data consistency when multiple copies are started.

The VERITAS Cluster Server engine assures that a service group is only online, partially online or in any states other than offline (such as attempting to go online or attempting to go offline).

Parallel Groups

A parallel group can run concurrently on more than one system in the cluster at a time.

A parallel service group is more complex than a failover group. It requires an application that can be started safely on more than one system at a time, with no threat of data corruption.

Cluster Communications (Heartbeat)

VERITAS Cluster Server uses private network communications between cluster nodes for cluster maintenance. This communication takes the form of nodes informing other nodes they are alive, known as *heartbeat.* With *cluster status,* nodes inform all other nodes of actions taking place and the status of all resources on a particular node. This cluster communication takes place over a private, dedicated network between cluster nodes. VERITAS requires two independent, private networks between all cluster nodes to provide necessary communication path redundancy and allow VERITAS Cluster Server to discriminate between a network failure and a system failure.

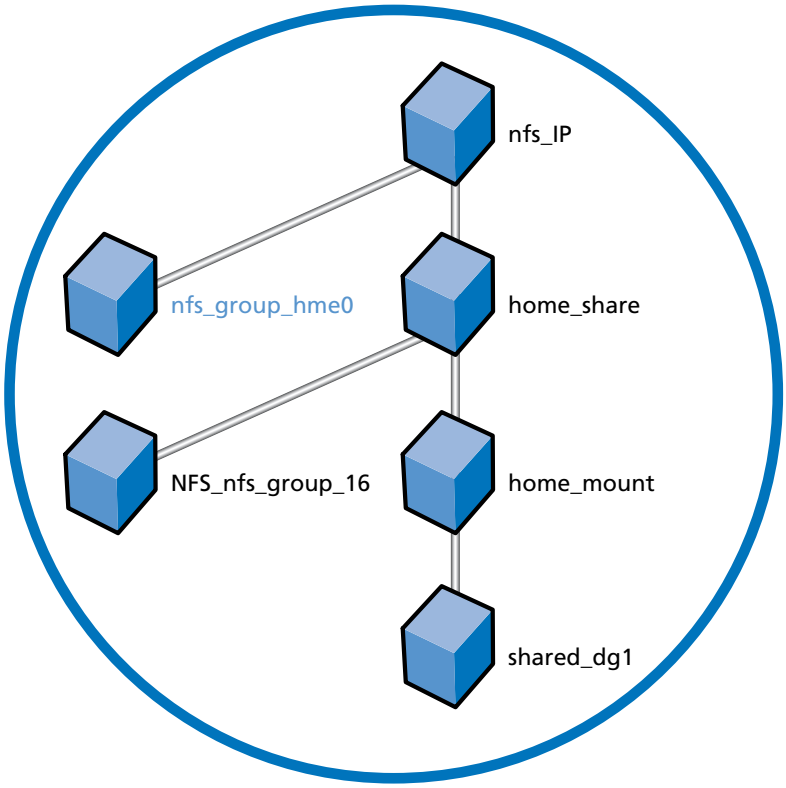
VERITAS Cluster Server uses a communication package comprised of the low latency transport (LLT) and group membership/atomic broadcast (GAB). These packages function together as a replacement for the IP stack and provide a robust, high-speed communication link between systems without the latency induced by the normal network stack.

Putting the Pieces Together

How do all these pieces tie together to form a cluster? Understanding how the pieces fit makes the rest of VERITAS Cluster Server fairly simple. Let's take a very common example, a 2-node cluster serving a single network file system to clients. The cluster itself consists of two nodes; connected to shared storage to allow both servers to access the data needed for the file system export.

In this example, we are going to configure a single service group called NFS_Group that will be failed over between server "A" and server "B" as necessary. The service group, configured as a *failover group,* consists of *resources,* each one with a different *resource type.* The resources must be started in a specific order for everything to work. This is described with *resource dependencies.* Finally, to control each specific resource type, VERITAS Cluster Server will require an *agent.* The VERITAS Cluster Server engine will read the configuration file and determine what agents are necessary to control the resources in this group, as well as resources in any other service group configured to run on this system. It also will start the corresponding VERITAS Cluster Server Agents. VERITAS Cluster Server will determine the order to bring up the resources

based on resource dependency statements in the configuration. When it is time to put the service group online, VERITAS Cluster Server will issue online commands to the proper agents in the proper order. The following drawing is a representation of a VERITAS Cluster Server service group, with the appropriate resources and dependencies for the NFS_Group. The method used to display the resource dependencies is identical to the VERITAS Cluster Server graphical user interface.



In this configuration, the VERITAS Cluster Server engine would start agents for disk group (shared_dg1), mount (home_mount), share (home_share), network file system (NFS_nfs_group_16), network interface card (nfs_group_hme0) and IP on all systems configured to run this group. The resource dependencies are configured as follows:

- The /home file system, shown as home_mount, requires the disk group shared_dg1 to be online before mounting.
- The network file system export of the home file system requires the home file system to be mounted as well as the network file system daemons to be running.
- The High Availability IP address, nfs_IP, requires the file system to be shared as well as the network interface to be up, represented as nfs_group_hme0.
- The network file system daemons and the disk group have no lower (child) dependencies, so they can start in parallel.
- The network interface card resource is a persistent resource and does not require starting.

The NFS_Group can be configured to start automatically on either node in the example. It then can move or fail-over to the second node based on operator command, or automatically if the first node fails. VERITAS Cluster Server will put resources offline starting at the top of the graph and start them on the second node, starting at the bottom of the graph.

VERITAS Cluster Server Advanced Capabilities

This section will outline capabilities found in VERITAS Cluster Server that set it apart from other application clustering packages.

Service Group Dependencies

Service group dependencies provide the capability to link entire service groups to provide startup and failover control. For example, an application group accessing a database group must wait to start until the database is started. Similarly, if the database group faults, the application group may need restarting.

VERITAS Cluster Server has a comprehensive set of service group dependency possibilities. VERITAS Cluster Server provides three possible online groups and one offline group: online local, online global, online remote and offline local.

- In an *online group dependency,* the parent (upper) group must wait for the child (lower) group to be brought online before it can start. For example, to configure an application and a database service as two separate groups, you would specify the application as the parent and the database service as the child. If the child faults, the parent is stopped and restarted after the child restarts. The online group dependency has three forms:
  - In an *online local* dependency, an instance of the parent group depends on an instance of child group being online on the same system. This typically is used in a database and application service configuration where the application directly connects to the database.
  - In an *online global* dependency, an instance of parent group depends on an instance of the child group being online on any system. This typically is used in a database environment with a front-end Web server connecting via IP.
  - In an *online remote* dependency, an instance of parent group depends on an instance of the child group being online on any system other than the system on which the parent is online. This configuration is useful where the load of the combined resource groups is too great for a single system.
- In an *offline local group dependency,* the parent group can be started only if the child group is offline on the system and vice versa. This prevents conflicting applications from running on the same system. For example, you can configure a production application on one system and a test application on another. If the production application fails, the test application will be put offline before the production application starts.

Triggers

VERITAS Cluster Server provides a method for the administrator to carry out specific actions when events occur in the cluster, such as a resource or group fault, or to carry out specific tasks outside the service group before it comes online or goes offline. This capability is know as triggers. Triggers provide the capability to extend the functionality of the cluster engine with simple shell scripts, perl or compiled programs. For example, the pre-online trigger can be used to carry out specific actions such as reconfiguring storage area network zoning before bringing up a group. The post-online and post-offline triggers can be used to control applications outside the cluster, such as signalling for a Web server restart following a database switchover.

Service Group Workload Management

Deciding which server should bring up a service group following a failure plays a major role in how well the cluster will function in a server consolidation environment. Lack of a dedicated, redundant server or cascading failovers complicates the decisions greatly. For example, if a node in a multinode cluster is running 10 service groups at time of failure, the administrator may wish all groups to move to an empty, redundant server. If this is not available, then the administrator may wish the groups to be spread out across remaining servers. How the cluster software determines the "best" takeover node varies among vendors.

Most cluster implementations provide a simple method to determine what node will act as a takeover node after a failure. The order of nodes listed in the configuration sets what node will take over a group. The first node in the system list in a running state will be chosen. This is a very restrictive policy, as it does not scale well in multinode environments. VERITAS Cluster Server provides three possible failover policies to make managing multiple, critical applications simpler. The three primary policies are priority, round robin and load.

- Priority is the most basic method. The first available running system in the system list is chosen. This is ideal for a simple two-node cluster, or a small cluster with a very small number of service groups.
- Round robin chooses the system running the least number of service groups as a failover target. This is ideal for larger clusters running a large number of service groups of essentially the same server load characteristics (for example, similar databases or applications).
- Load is the most flexile and powerful policy. It provides the framework for true server consolidation at the data center level in large clusters. In large clusters, with dozens of nodes and potentially hundreds of applications, the cluster software must be able to decide the best possible system for failover. Load policy is made of two components, system limits and group prerequisites.
  - System limits set a fixed capacity to servers and a fixed demand for service groups. For example, a Sun 6500 is set to a capacity of 400 and a Sun 4500 to 200. A database may have a determined load of 150. On failover, the server in the cluster with the highest remaining capacity will be chosen. When this group starts coming online, the 150 load is subtracted from the server's remaining capacity. The next service group will re-evaluate the remaining capacity of all servers and choose the best candidate.
  - The actual load on a server also can be calculated in real-time by an outside application and provide to VERITAS Cluster Server through a simple command line interface.
  - Service group prerequisites and limits add additional capability to the load policy. The user can set a list of finite resources available on a server (limits), such as shared memory segments, semaphores and others. Each service group then is assigned a set of prerequisites. For example, a database may need three shared memory segments and 10 semaphores. VERITAS Cluster Server load policy first will determine a subset of all systems that meet these criteria and then choose the lowest loaded system from this set. In this way, an unloaded system that does not meet all the prerequisites of a group will not be chosen.
  - System zones provide a subset of systems to use in an initial failover decision. A service group will try to stay within its zone before choosing a host in another zone. For example, imagine a typical three-tier application infrastructure with Web servers, application servers and database servers. The application and database servers are configured in a single cluster. Using `SystemZones` means a service group in the application zone will try to fail to another application zone server if it is available. If not, it then would fail to the database zone based on load and limits. In this configuration, excess capacity and limits available on the database backend essentially would be kept in reserve for the larger load of a database failover, while application servers would handle the load of any groups in the application zone. During a cascading failure, excess capacity in the cluster still is available to any service group. The system zones feature allows fine-tuning application failover decisions, yet still retains the flexibility to fail anywhere in the cluster if necessary.
  - Overload warning provides the final piece of the load policy. When a server sustains a predetermined load level (static or dynamically determined), the load warning trigger is initiated. (See the "Triggers" section for a full description of event management with triggers). The overload trigger is a user-defined script or application designed to carry out

the proper actions. Sample scripts detail simple operator warning on overload as well as a method to move or shut down groups based on user-defined priority values. For example, if load on a server running a business-critical database reaches and stays above a user-defined threshold, operators will be notified immediately. The load warning trigger then could scan the system for any service groups with a lower priority than the database (such as an internal human resources application) and move the application to a lesser-loaded system — or even shut down the application. The key here is that the framework is completely flexible. The installer or user is free to implement any overload management scheme desired.

Notification

VERITAS Cluster Server 2.0 provides full simple network management protocol (SNMP), management information base (MIB) support and a Highly Available SNMP/simple mail transfer protocol (SMTP) notifier. Features include:

- Configurable recipient list
- Configurable severity levels
- Configurable number of SNMP/SMTP servers
- SNMP notification
  - Support for HP OpenView, Novell ManageWise, IBM Tivoli
- SMTP notification
  - Mail to specified recipients based on type/location of failure

User Privileges

VERITAS Cluster Server 2.0 extends administrative capabilities by allowing multiple classes of cluster operators. Administrative privileges can be assigned at the cluster and group levels. Privileges can be granted for guest, operator and administrator levels. For example, a user can have complete control to modify and operate a specific service group (group administrator), yet only be allowed to view the remainder of the cluster configuration (cluster guest). Granting operator privileges only allows starting and stopping service groups. For example, a database administrator can start and stop a database service group, but not change its configuration.

Summary

The use of VERITAS Cluster Server, along with VERITAS Foundation Suite™ and VERITAS NetBackup, can increase availability of critical applications significantly in your environment. To help design and deploy complex application availability architecture, VERITAS offers a complete portfolio of professional services for assessment, architecture and design and deployment.

VERITAS™

WHITE PAPER

VERITAS Software Corporation
Corporate Headquarters
350 Ellis Street
Mountain View, CA 94043
650-527-8000 or 800-327-2232

For additional information about VERITAS Software, its products, or the location of an office near you, please call our corporate headquarters or visit our Web site at
www.veritas.com
e-mail us at sales@veritas.com

90-01657-399