ORACLE
SOLARIS

An Oracle White Paper
February 2011

Zone Clusters—How to Deploy Virtual Clusters and Why

Introduction

Many organizations are seeking ways to better utilize computer systems. Virtualization technologies provide a safe way to consolidate multiple applications on a single system. This paper introduces the *zone cluster* (also called an Oracle Solaris Containers cluster), a virtual cluster in which an Oracle Solaris Zone is configured as a virtual node. The zone cluster supports the consolidation of multiple cluster applications on a single cluster.

This paper addresses the following topics:

- “Cluster Application Consolidation” on page 2 presents the forces driving consolidation.
- “Server Virtualization Technologies” on page 3 provides an overview of Oracle's virtualization technologies, with an emphasis on Oracle Solaris Zones.
- “Zone Cluster Overview” on page 7 introduces the zone cluster and further identifies numerous use cases that demonstrate its utility.
- “Zone Cluster Design” on page 16 describes the overall design of the zone cluster.
- “Administration Overview” on page 23 provides an overview of zone cluster administration.
- “Example Zone Cluster Configuration” on page 25 contains step-by-step instructions for an example zone cluster configuration.
- “Zone Cluster Administration” on page 30 describes common zone cluster administrative tasks.

This paper assumes familiarity with Oracle Solaris Cluster and Oracle Solaris Zones concepts.

Cluster Application Consolidation

Up until quite recently, it was common to dedicate a single cluster to one application or a closely related set of applications. The use of a dedicated cluster simplified resource management and provided application fault isolation. The relatively low cost of computer hardware made this approach affordable. Figure 1 shows this typical approach to supporting multiple applications, with multiple clusters supporting different databases.

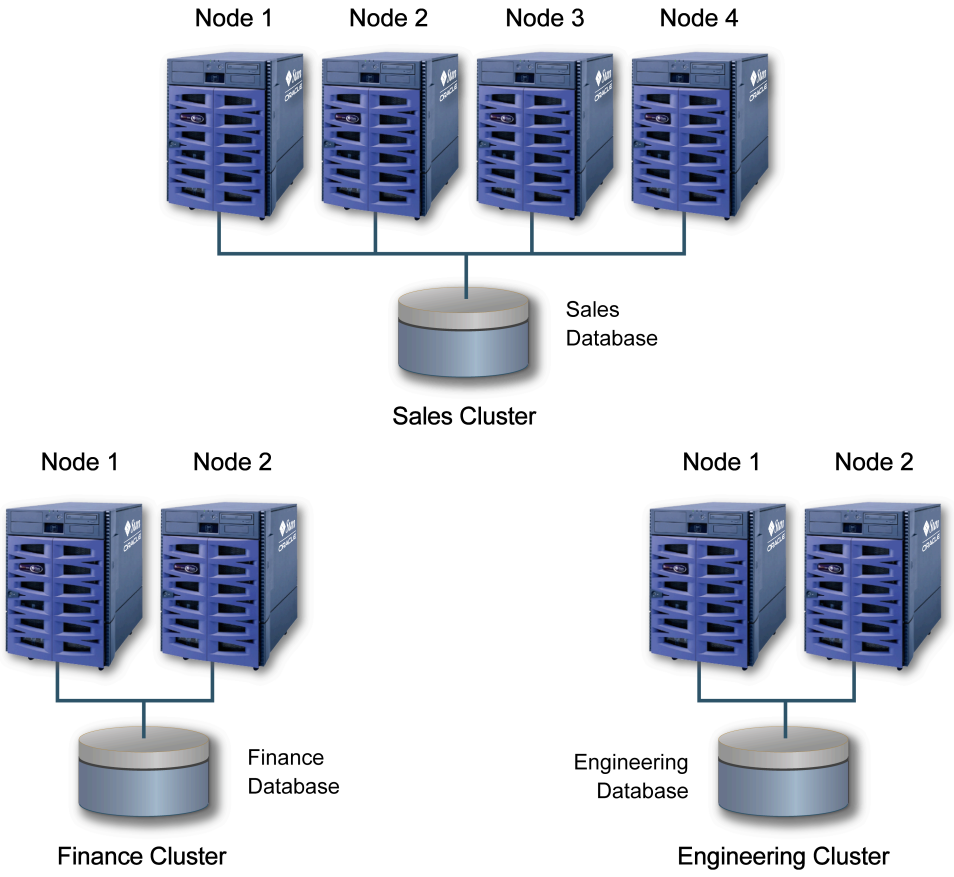

Figure 1. Example configuration: multiple clusters supporting different databases.

Moore's Law continues to apply to computers, and the industry continues to produce ever more powerful computers. The trend towards more powerful processors has been accompanied by gains in other resources, such as increased storage capacity and network bandwidth. Along with greater power has come improved price/performance ratios. While application processing demands have grown, in many cases these demands have grown at a much slower rate than that of the processing capacity of the system. As a result, many clusters now sit mostly idle with significant surplus processing capacity in all areas, including processor, storage, and networking.

Such large amounts of idle processing capacity present an almost irresistible opportunity for better system utilization. Organizations seek ways to reclaim this unused capacity, and thus are moving to host multiple applications on a single cluster. However, concerns about interactions between applications, especially in the areas of security and resource management, make people wary. Virtualization technologies address these security concerns and provide safe ways to host multiple applications in different clusters on a single hardware configuration.

Server Virtualization Technologies

Oracle offers a wide range of virtualization technologies that address network, storage, desktop, server, and operating system virtualization. This section focuses on server and operating system virtualization choices from Oracle. These virtualization choices facilitate hosting multiple applications on a single machine system, and include:

- Hardware partitions
- Virtual machines (VM)
- Operating system virtualization
- Resource management

Figure 2 provides a summary comparison of the virtualization technologies offered by Oracle. These technologies provide differing levels of isolation, resource granularity, and flexibility. The following sections provide an overview of each of these technologies.

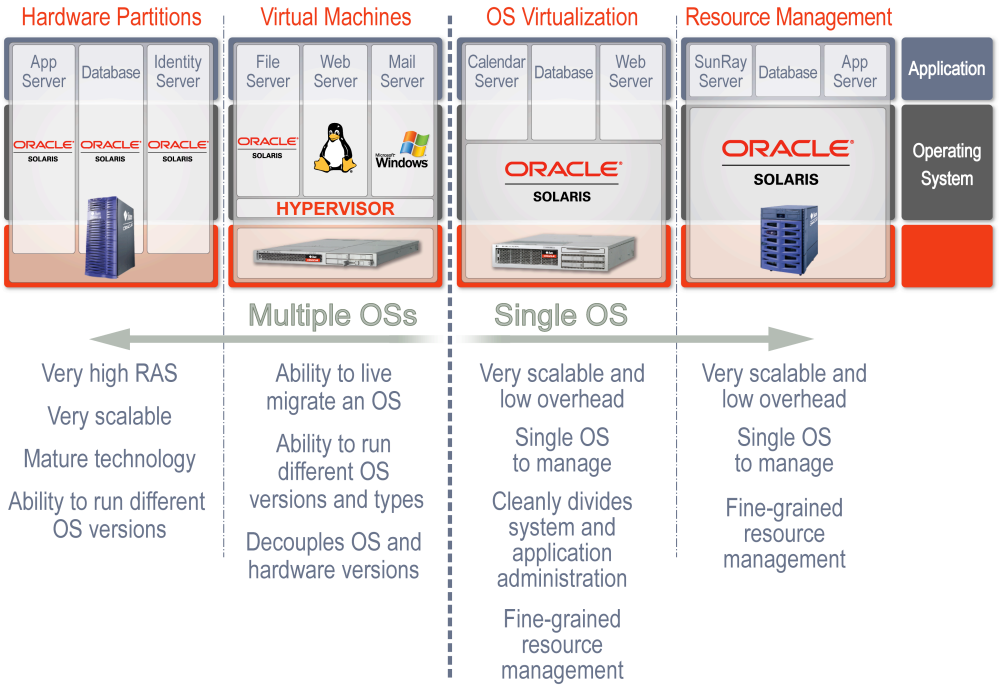

Figure 2. Virtualization technology comparison.

Hardware Partitions

Some high-end systems, such as select SPARC® Servers from Oracle, use physical hardware separation and can be configured into multiple hardware partitions. Each hardware partition is electrically isolated from the other partitions and runs a separate copy of the operating system. As a result, neither hardware nor software failures in one hardware partition can affect the other partitions.

The software sees each hardware partition as an independent computer capable of running an operating system, and each hardware partition can run a different operating system type and version. Each hardware partition is independent, and there is no interference between the hardware partitions, providing good scalability for applications.

With dynamic domains, hardware can be added to or removed from a hardware partition without shutting down any applications. The system administrator can move resources between hardware partitions, but resources cannot be shared as easily as when managed under a single operating system. The major disadvantage of this virtualization approach is that it is costly, and thus is limited to high-end systems.

Virtual Machines

Virtual machine (VM) technology supports multiple operating systems from different vendors and can support many OS environments per physical machine. The VM approach places a layer of firmware, called the hypervisor, between the traditional operating system and the physical machine. The hypervisor presents a virtual machine interface to the operating system and arbitrates requests from the operating system in the virtual machines. Thus, the hypervisor can support the illusion of multiple machines, each of which can run a different operating system image.

A variety of virtualization technologies exist, including Oracle VM Server for SPARC (previously called Sun Logical Domains) that uses the on-board hypervisor of the UltraSPARC® T1/T2-T2 Plus-based servers with chip multithreading (CMT) technology and Oracle VM Server for x86. These solutions use a technique called paravirtualization, in which the operating systems in the virtual machines have been modified to deal more efficiently with the hypervisor. Other companies offer virtualization techniques, such as Hyper-V and VMware, that also run on Oracle's Sun x86 servers.

Each virtual machine can run a different operating system type and version, which allows different operating systems to run on a single physical server. Virtual machines also provide the ability to migrate an entire live operating system from one machine to another. No hardware fault isolation is provided by virtual machine approaches. However, the virtual machine interface provides a generic machine interface to the operating system, which decouples the operating system from details of the underlying hardware. This reduces the need to change the operating system for different platforms.

One disadvantage of the virtual machine approach is the overhead of the hypervisor layer, due to the work needed for arbitration. Some of this overhead can be eliminated by dedicating a resource to a virtual machine, but then that resource cannot be shared.

Operating System Virtualization

The operating system virtualization approach creates an isolated environment for an application or set of applications under a single operating system image. The Oracle offering in this area, Oracle Solaris Zones[1], is an integral part of the Oracle Solaris 10 OS. Oracle Solaris Zones isolate software applications and services using flexible, software-defined boundaries and allow many private execution environments to be created within a single instance of the Oracle Solaris 10 OS.

The underlying Oracle Solaris OS has a single *global zone*, which is both the default zone for the system and the zone used for system-wide administrative control (see Figure 3). The system administrator of the global zone can create one or more non-global zones, and identifies all resources that will be made available to these non-global zones. An application or user within a non-global zone cannot see or affect things outside of the enclosing zone, thus providing strong security.

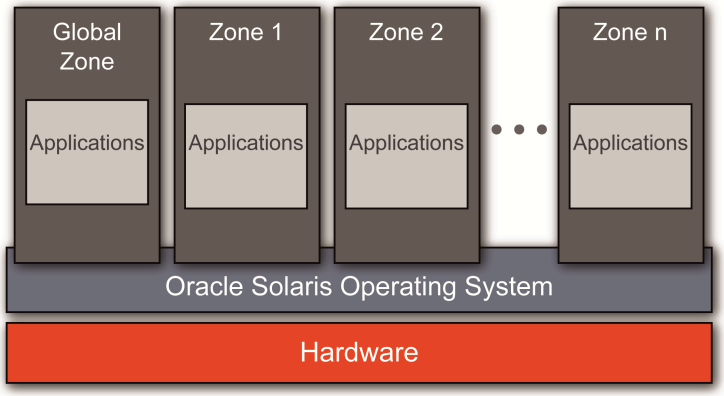

Figure 3. Oracle Solaris Zones.

The zone is a unit of granularity for resource management. Applications within a zone must share the resources assigned to the zone. A system administrator can dedicate resources to a zone or grant some specific share of resources for use by a zone. For example, a system administrator can grant a Fair Share Scheduler share of 50 to zone A, 30 to zone B, and 20 to zone C; and the result would be that zone A gets 50% of CPU resources, while zone B get 30% and zone C gets 20%.

The zone provides a unit of fault isolation. For example, an attempt within a zone to execute a reboot results in a zone reboot instead of a machine reboot. This isolation helps prevent application actions or faults from affecting applications in other zones.

Compared to the technologies using multiple operating systems, the Oracle Solaris Zones model has less administrative overhead as there is only one operating system to manage. In addition, Oracle Solaris Zones do not have the overhead of the hypervisor, do not require any special hardware, and are available on all platforms starting with Oracle Solaris 10. While Oracle Solaris Zones isolate applications, they do not isolate operating system failures nor hardware failures.

---

[1] An Oracle Solaris Container combines the boundary separation provided by Oracle Solaris Zones with system resource controls. However, zones can be used independently of resource management. Many people use the terms *zones* and *containers* interchangeably.

Resource Management

Resource management is a feature of the Oracle Solaris OS that manages the utilization of resources by an application or a collection of applications called a project. Within a single copy of the Oracle Solaris OS, resources can be dedicated to zones, applications, or projects. This allows resources such as CPU, swap, and memory to be bound to a specific program. However, all applications see each other and can affect each other. As such, resource management provides neither security nor application fault isolation.

Selecting a Server Virtualization Approach

The critical factors that drive the selection of a virtualization approach in terms of high availability are the following:

- Electrical isolation
- Operating system fault isolation
- Support for different operating systems versions
- Application fault isolation
- Security isolation
- Resource management

One of the most common scenarios motivating people to adopt virtualization is the desire to consolidate applications supporting multiple organizations, and this is also a major focus of this report. Consider how a virtualization approach would be selected for this scenario:

- Modern hardware achieves a good level of reliability, and the high availability features of a cluster enable the system to overcome many of the more common hardware faults. Thus electrical isolation is not essential.
- The Oracle Solaris operating system is reliable and can support a wide variety of applications, which means that a single operating system image meets most customer needs in this scenario.
- There are important applications that recover from errors by rebooting machines. Thus application fault isolation is essential to ensure that the applications of one organization do not affect others.
- Different organizations want to keep their data private, which makes security isolation essential.
- Each organization is usually charged for computer services. So each organization wants to be assured that it gets what it paid for, and resource management provides that assurance.

Zones satisfy the requirements of this scenario with low overhead in a well integrated feature set. This paper now restricts itself to the operating system virtualization technology approach.

Zone Cluster Overview

The virtualization technology products discussed so far are all single machine products. The zone cluster extends the Oracle Solaris Zone principles to work across a cluster, providing support for applications.

Cluster Terminology

Changes to the Oracle Solaris 10 OS cause a rethinking of basic cluster concepts, such as *Oracle Solaris host*, *cluster*, and *cluster node*.

- *Oracle Solaris host*—the Oracle Solaris host is a configuration that supports exactly one Oracle Solaris image and one Oracle Solaris Cluster image. The following entities can be an Oracle Solaris host:
  - A "bare metal" physical machine that is not configured with a virtual machine or as a hardware domain
  - An Oracle VM Server for SPARC guest domain
  - An Oracle VM Server for SPARC I/O domain
  - A hardware domain
- *Cluster node*—A cluster node has two properties: A cluster node hosts cluster applications, and a cluster node can be a member of the cluster that contributes votes towards cluster membership.
- *Cluster*—A cluster is a collection of one or more cluster nodes that belong exclusively to that collection.

Oracle Solaris 10 introduces the concept of a zone. Oracle Solaris 10 runs all applications in a zone, which can either be a global zone or a non-global zone. Since cluster applications always run in a zone, the cluster node is always a zone.

Since a physical machine can now host multiple Oracle Solaris hosts, there can be multiple Oracle Solaris 10 OS images on a single machine. A single Oracle Solaris 10 OS image will have exactly one global zone, and may have any number of zones belonging to different zone clusters. Thus, a single Oracle Solaris host can support multiple cluster nodes. However, each cluster node on a single Oracle Solaris host will belong to separate clusters.

Global Clusters and Zone Clusters

Two types of clusters can be configured with Oracle Solaris 10: *global clusters* and *zone clusters*.

- *Global cluster*—The global cluster contains all global zones in a collection of Oracle Solaris hosts. The global cluster can optionally contain non-global zones with no membership votes.
- *Zone cluster*—A zone cluster is composed of one or more non-global zones, which are all of zone brand type cluster. Each cluster node of a zone cluster resides on a different Oracle Solaris host.

A zone cluster node requires that the global zone on that same Oracle Solaris host must be booted in cluster mode in order for the zone cluster node to be operational. All zone cluster nodes must be on Oracle Solaris hosts belonging to the same global cluster. The zone cluster nodes can be a subset of Oracle Solaris hosts for that same global cluster. While a zone cluster depends upon a global cluster, a global cluster does not depend upon any zone cluster.

Figure 4 shows a four-machine hardware configuration supported by Oracle Solaris Cluster.

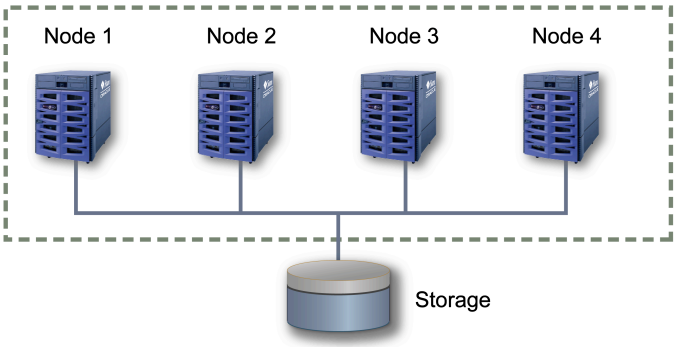

Figure 4. A four-machine hardware configuration for clustering.

Figure 5 shows multiple clusters running on that same four-machine hardware configuration. An important point is that a cluster node is a cluster software construct that does not necessarily have a one-to-one relationship to hardware. [When Sun Cluster ran on Solaris 9, the host and the cluster node were the same thing. And, except for hardware domains, the cluster node was also the physical machine. This traditional cluster was perhaps simpler to understand, but was also less flexible.]

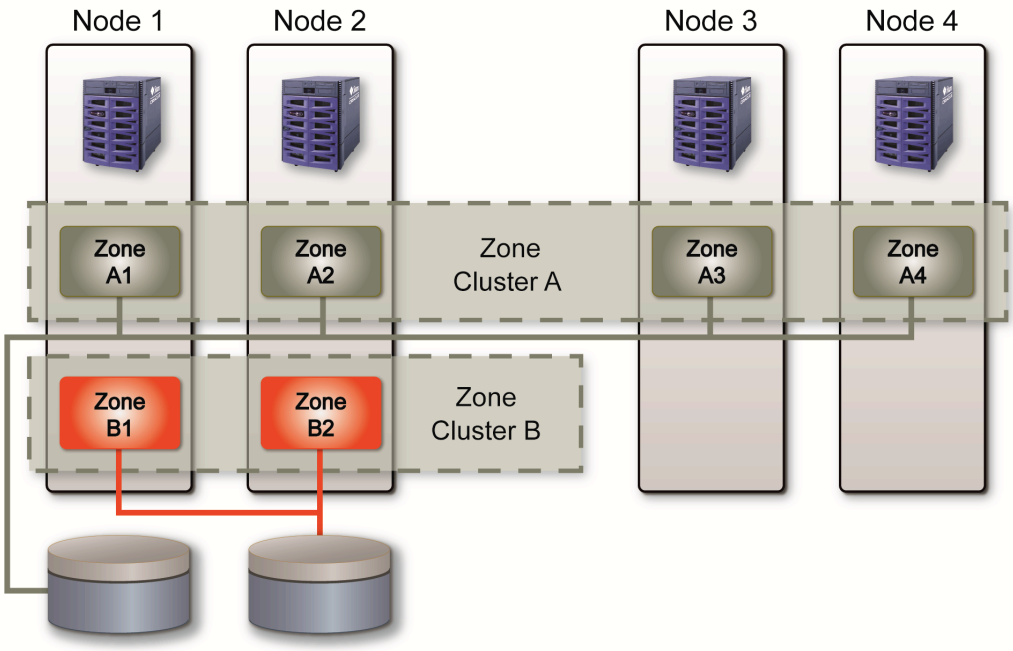

Figure 5. Multiple clusters deployed on a four-machine configuration.

The zone cluster can be viewed as a *virtual cluster*: the zone cluster presents the illusion to a cluster application that the zone cluster is the traditional cluster.

The following sections provide more details on the features of the zone cluster, and present various potential use cases for this technology.

Security Isolation

Cluster applications come in two basic types: *failover* and *scalable*. A *failover application* is a single instance application that can run on just one node at a time. The system will restart the failover application on another node as a result of a node failure or administrative request. A *scalable application* consists of multiple application instances running on different nodes.

An Oracle Solaris Zone is created and exists on only one machine. Thus, the current Oracle Solaris Zone cannot be used as the security container for either failover or scalable applications. This paper introduces the concept of a *cluster-wide zone*, intended specifically for use as a security container for both failover and scalable applications. The zone cluster provides the implementation for a cluster-wide zone.

The cluster-wide zone is the security container for both failover and scalable applications. No instance of a failover or scalable application can run outside of the encapsulating cluster-wide zone. In other words, all instances of a scalable application always run within a single cluster-wide zone, and a failover application cannot switchover or failover outside of the cluster-wide zone.

The zone cluster follows the zone model of security. The only resources that can be seen or affected from within a zone cluster are those resources that the system administrator from the global zone has configured into the zone cluster. It is impossible to add resources to a zone cluster from within the zone cluster. Similarly, it is impossible to change zone cluster configuration parameters from within the zone cluster. For example, it is impossible to change the share of CPUs allocated to a zone cluster from within the zone cluster.

Application Fault Isolation

Operating systems provide some level of application fault isolation. For example, the panic of one application instance does not cause failures in all other applications. However, there are actions that an application can take that will cause failures of other applications. For example, an application can order the node to reboot, which obviously affects all applications on that node.

The Oracle Solaris Zone feature set has been designed to reduce the possibility that the misbehavior of one application will negatively impact other applications. Continuing with the same example, the system treats a reboot command issued within a zone as a "zone reboot" command, which ensures that the reboot command does not affect applications outside of that zone.

Oracle Solaris Zones disallow many operations that can negatively impact other applications outside of that zone. The zone cluster retains support of this principle.

Resource Management

The Oracle Solaris operating system has long included a resource management subsystem. The administrator can use Oracle Solaris resource management software to dedicate resources for some purpose or assign a particular share of a resource type to a project or application.

The Oracle Solaris Resource Manager software has added a level of granularity for managing resources at the zone level. The system administrator from the global zone can manage the resource usage of the zone, and the operating system ensures that these controls cannot be changed from within the zone. This is particularly important when consolidating Oracle Solaris Cluster applications on a single system. The zone cluster retains this zone feature.

Dedicated Cluster Model

The zone provides the illusion to a single machine application that the zone is a machine dedicated for the use of the applications within that zone. The zone cluster provides the illusion to cluster applications that the zone cluster is a cluster dedicated for the use of cluster applications within that zone cluster. Similarly, when a user logs in to the zone cluster, the user sees the zone cluster as a traditional cluster.

The zone cluster is a simplified cluster. Employing a minimalist approach, only the components needed to directly support cluster applications are present, including such things as:

- File systems
- Storage devices
- Networks
- Cluster membership

Those cluster components that are not needed by cluster applications, such as quorum devices and heartbeats, are not present. Zone clusters do not mimic all aspects of the physical system. For example, zone clusters do not support zones nested within the zones of the zone cluster.

Zone Cluster Use Cases

This section demonstrates the utility of zone clusters by examining a variety of use cases, including the following:

- Multiple organization consolidation
- Functional consolidation
- Multiple-tier consolidation
- Cost containment
- Administrative workload reduction

Multiple Organization Consolidation

Any solution that consolidates cluster applications from multiple organizations must satisfy the following critical requirements:

- *Security Isolation*—The solution must ensure that applications and users from different organizations cannot see or affect others. Different organizations insist upon ensuring that their own information remain private.
- *Application Fault Isolation*—The solution must not allow the failure of one application to affect applications in other areas. Different organizations do not want their schedules impacted by problems of other organizations.
- *Resource Management*—The solution must provide controls on the utilization of resources by the applications of each organization. Computer resources are not free. Costs must be allocated to the different organizations, and the different organizations want guarantees that they receive the resources for which they have paid.

Zone clusters satisfy these core requirements. Figure 6 shows an example of the consolidation of databases from two different organizations upon one physical four-node cluster. In this example, the Sales Zone Cluster spans all four machines, while the Finance Zone Cluster spans two machines.

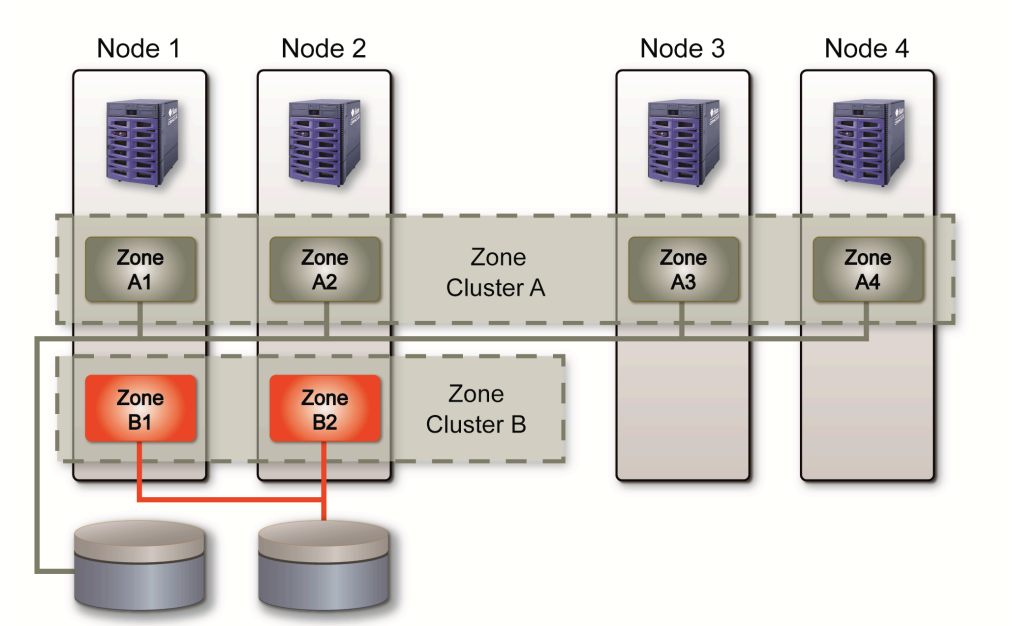

Figure 6. Example consolidation of applications from two organizations.

Functional Consolidation

Many organizations dedicate different clusters for different purposes. One of the most common divisions of responsibility is as follows:

- Production
- Test
- Development

Some more cautious organizations are unwilling to risk any extraneous activity upon their production systems, but many organizations may be willing to consolidate test and development activity upon a single physical cluster. Figure 7 shows an example of test and development activity consolidated on a single physical cluster using zone clusters.

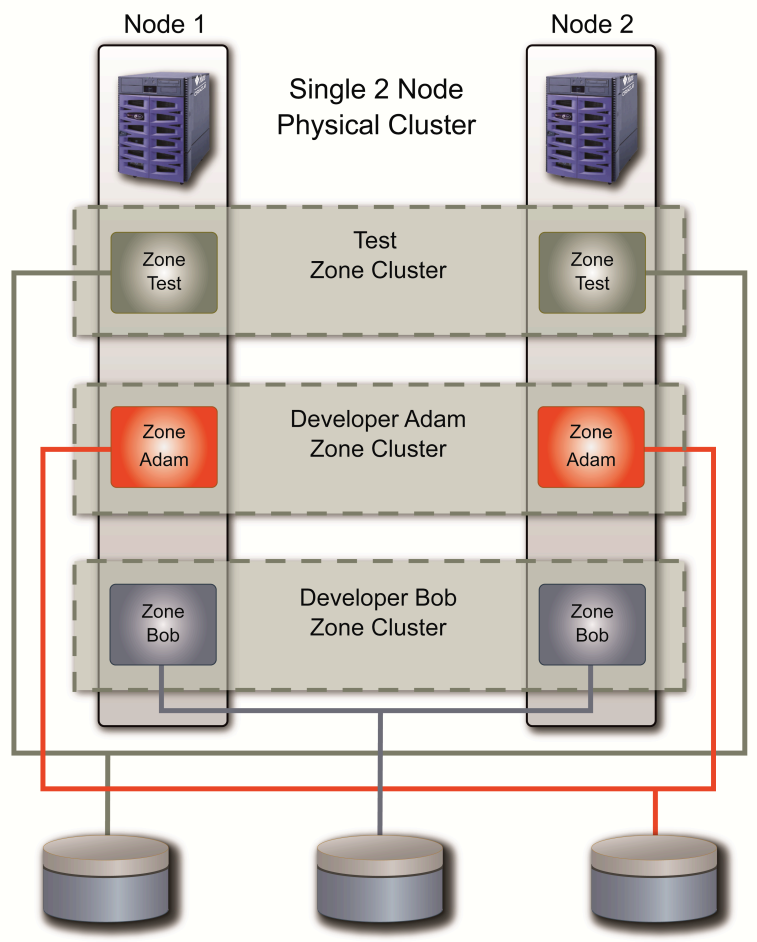

Figure 7. Example of functional consolidation.

The critical factors in functional consolidation are application fault isolation and resource management.

It has been quite common for developers to schedule dedicated time on physical clusters, because the activity of other developers could negatively impact work. Zone clusters enable applications in

different zone clusters to operate independently of the behavior of applications in other zone clusters. Even application failures in other zone clusters do not impact the applications beyond the containing zone cluster. Thus engineers can reboot zones or switchover applications in one zone cluster without impacting any other zone cluster.

The system administrator can create or destroy zone clusters dynamically without impacting other zone clusters. The limit on the number of zone clusters is so large at 8,192 zone clusters that the number of zone clusters is effectively unlimited. The combination of these factors means that an organization can configure a zone cluster for each developer and each developer can work concurrently. Configuring one or more zone clusters for each developer can eliminate the need for developers to schedule dedicated cluster time (often at inconvenient off-hours) and can help speed development.

Some cluster applications assume that the cluster application runs on a dedicated cluster. Zone clusters support the ability to run multiple cluster applications of this class. This feature is especially important when testing changes or release levels in this class of applications.

Multiple-Tier Consolidation

The well-known three-tier datacenter model identifies the following tiers:

- Front-end
- Application
- Database

Zone clusters support the consolidation of applications from all three tiers. Figure 8 shows a consolidation example using Scalable Apache Web server for the front-end tier; a Java™ 2 Platform, Enterprise Edition (J2EE) application server for the application tier; and a RAC database for the database tier. All tiers use separate zone clusters.

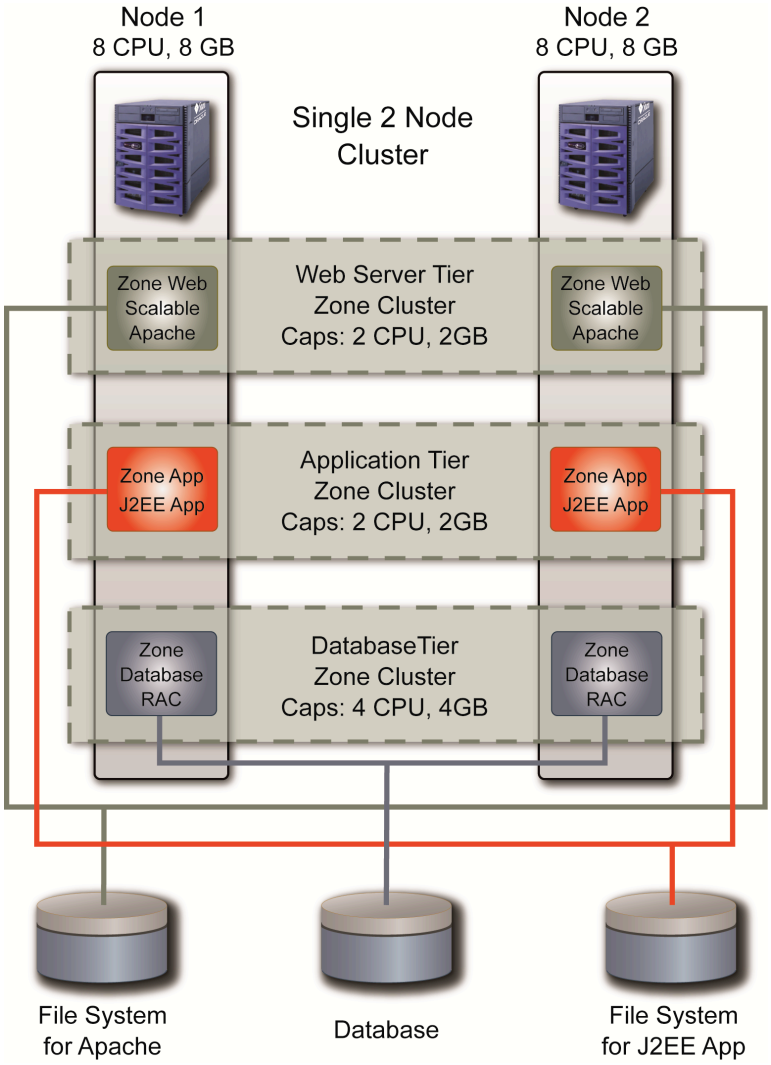

Figure 8. Example of multiple-tier consolidation.

Cost Containment

Each zone cluster provides a unit of granularity for resource management. Organizations can use this basic capability of zone clusters to control costs. Many software vendors use pricing models based upon the processing power of the system as represented by the number of CPUs. Administrators can configure a zone to use either specific CPUs or a specific number of CPUs. When an application runs in a zone, the application can use only the CPUs available to the zone containing the application. Either of these limits on CPUs can be used to determine the application license fee, which can result in significant savings.

Consider an example four-node cluster in which each machine is configured in the same way as the two-node cluster shown in Figure 8, and assume each machine has eight CPUs. The physical cluster hosts three software applications. In order to avoid legal issues about specifying charges for specific

products, this example assumes that the Front-end Tier hosts application AppFrontEnd, the Application Tier hosts AppApp, and the Database Tier hosts AppDB. Now assume the following fictional charges for software licenses:

- AppFrontEnd Per CPU Fee (FEfee): $10,000 per CPU
- AppAPP Per CPU Fee (APfee): $15,000 per CPU
- AppDB Per CPU Fee (DBfee): $20,000 per CPU

The formula for the pricing would be:

```
(FEfee + APfee + DBfee)× Number of CPUs per node × Number of Nodes
```

The total license cost for this configuration would be:

```
($10,000 + $15,000 + $20,000)× 8 × 4 = $1,440,000
```

Now consider the same physical cluster with the same applications. However, this time a separate zone cluster is used for each application. This configuration assigns two CPUs for the zone cluster hosting AppFrontEnd, assigns two CPUs for the zone cluster hosting AppApp, and assigns four CPUs for the zone cluster hosting AppDB.

The formula for the pricing in this configuration would be:

```
((FEfee × CPUs) + (APfee × CPUs) + (DBfee × CPUs)) × Number of Nodes
```

The total license cost for this configuration would be:

```
(($10,000 × 2) + ($15,000 × 2) + ($20, 000 × 4)) × 4 = $520,000
```

This example reduces costs by nearly two thirds. The simple formula for costs as shown above can be used to calculate cost savings when using zone clusters. When multiple applications run on the same cluster, each of the applications will be running on some CPUs. In this situation a single application will not be utilizing all CPUs. So there is no need to pay license fees for CPUs that will not be used for that application. Thus, even if an organization has already consolidated cluster applications on a single physical cluster, there can be significant cost savings by using zone clusters.

This cost containment feature has been officially recognized since the initial release of Oracle Solaris 10 by using Oracle Solaris resource management software. Under the earliest approach, the administrator defined a processor set for a pool and mapped the zone onto that pool. The result is that all applications in that zone are limited to those specific CPUs. Today, there are easier ways to accomplish this. The administrator can now specify either the number of CPUs or select the specific CPUs when configuring a zone.

Note—See “References” on page 41 for more information on this cost containment policy.

Administrative Workload Reduction

Zone clusters are considerably simpler than global clusters. For example, there are no quorum devices in a zone cluster, as a quorum device is not needed. A global cluster and all of the zone clusters on that

global cluster share the same operating system. Operating system updates and patches need only be applied once for both the global cluster and all of its zone clusters. This translates into a reduction of administrative work when zone clusters can be substituted for global clusters.

As an example of this principle, consider one large financial institution with a large number of databases. This financial institution plans to consolidate 6 to 8 databases per cluster, with a cluster size up to 16 nodes, for a total number of nodes approaching 200. The financial institution plans to use a zone cluster per database. One motivation for this approach is that there is a lot less administration work for a set of zone clusters versus a set of global clusters.

Zone Cluster Design

Earlier sections presented an overview of the concept of a zone cluster, and described how zone clusters can be used. This section describes the zone cluster design at a high level, and identifies how and what is supported.

Virtual Node

The zone cluster consists of a set of zones, where each zone represents a virtual node. Each zone of a zone cluster is configured on a separate machine. As such, the upper bound on the number of virtual nodes in a zone cluster is limited to the number of machines in the global cluster.

Oracle Solaris supports the modification and enhancement of a zone through use of the *BrandZ framework*. The zone cluster design introduces a new brand of zone, called the *cluster brand*. The cluster brand is based on the original native brand type, and adds enhancements for clustering. The BrandZ framework provides numerous hooks where other software can take action appropriate for the brand type of zone. For example, there is a hook for software to be called during the zone boot, and zone clusters take advantage of this hook to inform the cluster software about the boot of the virtual node.

Note—Because zone clusters use the BrandZ framework, at a minimum Oracle Solaris 10 5/08 is required.

The user does not need to be concerned about the brand type of zone. From the view point of customers and applications, the cluster brand zone looks and acts just like a native zone with cluster support enabled.

Cluster Membership

Each zone cluster has its own notion of membership. The format of membership for a zone cluster is identical to that of a cluster running on Solaris 9 OS without zones. Applications running in a zone cluster receive the same kind of information as when running in the global zone. This means that applications run identically in the zone cluster and global zone with respect to membership.

Naturally, a zone of a zone cluster can only become operational after the global zone on the hosting machine becomes operational. A zone of a zone cluster will not boot when the global zone is not booted in cluster mode. A zone of a zone cluster can be configured to automatically boot after the

machine boots, or the administrator can manually control when the zone boots. A zone of a zone cluster can fail or an administrator can manually halt or reboot a zone. All of these events result in the zone cluster automatically updating its membership.

Membership Monitoring

The system maintains membership information for zone clusters. Each machine hosts a component, called the Zone Cluster Membership Monitor (ZCMM), that monitors the status of all cluster brand zones on that machine. The ZCMM knows which zones belong to which zone clusters.

First consider the case where global cluster node membership changes, because of either a node join or node departure. The node reconfiguration process in the global zone determines the new cluster membership. Upon completing the machine reconfiguration process, the system selects a new ZCMM leader. If the previous ZCMM leader is still around, there will be no change; otherwise, the system arbitrarily picks one ZCMM as the leader. The ZCMM leader collects zone cluster virtual node information from the ZCMMs on each node, compiles the information, and then distributes the new membership to all ZCMMs.

In the case where a zone cluster virtual node status changes, the ZCMM on that machine forwards that information to the ZCMM leader. This triggers a zone cluster reconfiguration. The ZCMM Leader distributes new membership information for that zone cluster to the ZCMM on each machine.

This design quickly updates and delivers the zone cluster membership information. Those with cluster experience have probably noted that it takes time for global cluster recovery to complete after a global cluster reconfiguration. After a global cluster node reconfiguration, the zone cluster membership process begins and completes long before the global cluster recovery process completes. This prevents significant delays in updating zone cluster membership.

Security

The zone cluster security design follows the security design for the Oracle Solaris Zone feature. The zone is a security container. The operating system checks all requests to access resources, such as file systems, devices, and networks, to determine whether such access has been granted to that zone. When permission has not been granted, the operating system denies access. Applications can send requests to software via a limited number of communication channels, such as system calls and doors. The operating system tags each communication with information identifying the zone from which the request originated. Software in the kernel or global zone is considered to be trusted, and has the responsibility to check the access permissions based upon the originating zone and deny any unauthorized access. Applications in a non-global zone cannot tamper with software in the kernel or in the global zone. The overall result is that the system restricts application access to just authorized items.

Figure 9 shows the overall security architecture for zone clusters.

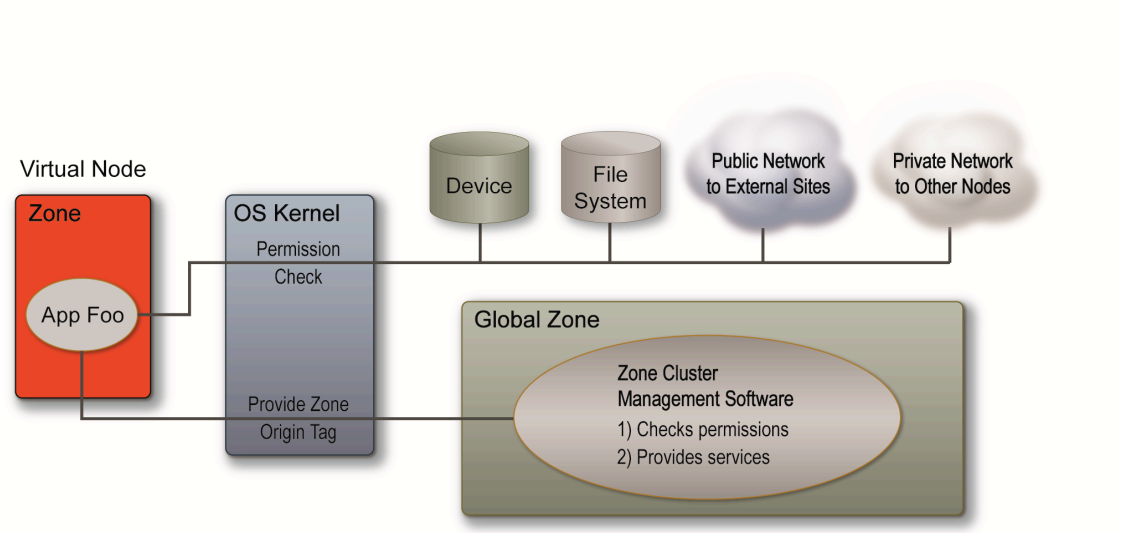

Figure 9. Zone cluster security architecture.

The zone components of a zone cluster are equivalent from a security perspective. The design ensures that the privilege related parameters are the same for all zone components of a zone cluster. For example, the zone name and zone root path are the same for each zone component.

Support for Solaris Trusted Extensions

Government organizations, such as military and intelligence agencies, and large financial institutions have very stringent security policies. They classify data using various security labels and users are given fine grained access rights to such data. Traditionally, this requirement has been termed "Multi-Level Security MLS". Oracle Solaris Trusted Extensions is a Multi-Level Secure (MLS) operating system.

Trusted Extensions is based upon the concept of a single machine security container and uses Solaris Containers as the security container. However, a single machine is a single point of failure. A group of zones running on multiple systems can act together as a single security container as well as providing a high availability platform. Processes, file systems, shared devices and networks belonging to such a single cluster-wide security container can move from one system to another as needed in the case of failures or administrator request, but always within the confines of that "cluster-wide security container," thus providing both high availability and multi-level security. Together the capabilities of Oracle Solaris Cluster and Trusted Extensions can provide these cluster-wide security containers.

File Systems

Zone clusters support access to a variety of different types of file systems, including local, highly available, cluster, and NFS file systems. The subsequent subsections describe how the various types of file systems are supported in zone clusters.

Local File Systems

A virtual node of a zone cluster can have access to a file system that is only available to that particular virtual node, and this is called a *local file system*. The rules for zone cluster support of local file systems

are identical to that of a native zone. The zone cluster relies upon the basic zone file system support for this feature. Zone cluster local file system support includes Oracle Solaris Zettabyte File System (ZFS), UFS, Veritas VxFS, and Sun QFS.

Highly Available File Systems

A *highly available file system* is accessible to multiple virtual nodes but is mounted in exactly one virtual node at a time. The cluster will mount the highly available file system in another virtual node in response to a virtual node failure or the manual switchover command from an administrator. Once mounted, a highly available file system works exactly like a local file system.

Zone clusters directly mount highly available file systems in the virtual node when mounted with read and write privileges. Only one zone cluster can access a particular highly available file system at any given time. Zone clusters can mount highly available file systems for the virtual node using a loopback mount with only read privileges. Multiple zone clusters can share a highly available file system with read-only privileges.

Zone clusters support highly available file systems using the HAStoragePlus subsystem. Zone clusters record information about highly available file systems in the Cluster Configuration Repository. Zone clusters validate access permissions both when the administrator specifies an application dependency upon a highly available file system and when actually mounting the file system.

Zone cluster highly available file system support includes Oracle Solaris ZFS, UFS, Veritas VxFS, and Sun QFS.

Cluster File Systems

Zone clusters use two methods to make *cluster file system*, also sometimes called *global file systems*, available to applications running in the virtual nodes.

The first method mounts the cluster file system on all virtual nodes directly, as opposed to using a loopback mount. This approach is employed when using shared QFS file systems. The shared QFS file system is under the control of a scalable mount point (`SUNW.ScalMountPoint`)_resource running in the global zone. The resource mounts the file system in a location that is relative to the virtual node's zone path. When a shared QFS file system is mounted in this way, it can only be mounted in one zone cluster at any one point in time.

The second method mounts the cluster file system on all virtual nodes using a loopback mount. This approach is employed when using the *Proxy File System*. The Proxy File System technology allows UFS and Veritas VxFS file systems to be used as global file systems. The UFS or VxFS cluster file system is under the control of an HAStoragePlus (SUNW.HAStoragePlus) resource running in the zone cluster. The methods used by this resource run in the global zone enabling the file system to be loopback mounted on to the zone cluster's virtual nodes. In contrast to cluster file systems using shared QFS, those using UFS or VxFS can be mounted on virtual nodes of multiple zone clusters concurrently. Furthermore, the file system can be mounted either read-write or read-only, on a per-virtual node basis.

However, the two methods described above do share a number of common features. The zone cluster mounts the cluster file system at the same mount point for all virtual nodes of the zone cluster. Any mounted cluster file system is mounted in all virtual nodes of the zone cluster. The failure of one virtual node does not stop access to the cluster file system from other virtual nodes. The zone cluster ensures that any joining virtual node will automatically have the same cluster file systems mounted in exactly the same mount points as is currently the situation with other virtual nodes of the zone cluster.

Zone clusters record information about cluster file systems in the Cluster Configuration Repository. Zone clusters validate access permissions both when the administrator specifies an application dependency upon a cluster file system and when actually mounting the file system.

NFS File Systems

An NFS file system can be mounted within a zone cluster in one of two ways. The first method is the basic approach of mounting the NFS file system on an individual node, in this case a virtual node, as required. The second method mounts the NFS file system on all virtual nodes within the zone cluster under the control of a scalable mount point (`SUNW.ScalMountPoint`) resource. This second method also shares the features described in the cluster file system section relating to automatic mounting of the NFS mount on virtual nodes that join the zone cluster. This method is also available within the global zone.

An NFS file system cannot be imported in one zone, such as the global zone, and then exported to another zone using loopback mounts.

Storage Devices

Zone clusters support direct access to storage devices including disks and RAID units, and network-attached storage (NAS) devices.

Disks and RAID Devices

Zone clusters support direct access to both ordinary disks and RAID units. Zone clusters also support access to volume manager devices as if they were a disk. Zone clusters allow only one zone cluster direct access to any particular disk or RAID unit.

When data on a disk or RAID unit is accessed via a file system, it is recommended that the administrator not grant direct access to the disk or RAID unit. When an application has direct access to a disk or RAID unit, the application can issue IOCTL calls. A malicious application can issue a bad IOCTL and cause a driver malfunction, which can cause operating system failure. This opens up a reliability concern, but does not present a security risk. The use of file systems eliminates this reliability concern. However, some users require direct access and understand the risk, and therefore zone clusters support this feature.

Zone clusters support fencing of disks and RAID devices. The fencing feature ensures that a node that has left the cluster cannot continue to modify data on shared storage. This is a very important data

integrity feature that has been supported by Oracle Solaris Cluster for a long time. The zone cluster feature supports fencing for the nodes of the zone cluster, and thus provides data integrity.

When a virtual node fails on a machine where the global zone remains operational, the operating system does not mark the zone as down until all I/O has terminated. Zone clusters do not mark a virtual node as down until the operating system has marked the zone as down. Thus once the virtual node is down, Oracle Solaris Cluster can guarantee that no more I/O will come from the departed virtual node. In all cases, Oracle Solaris Cluster ensures that no I/O will come from a departed node.

Oracle Solaris Cluster supports fencing of disks and RAID units in all of the following cases:

- When access is direct to the device
- When access is via a volume manager
- When access is via a file system

Zone clusters also support access to volume manager devices. Zone clusters rely upon the basic zone support for volume manager devices. Both Oracle Solaris Volume Manager (SVM) and Veritas Volume Manager administration must be done from the global zone. Zone clusters support automatic volume manager reconfiguration after zone cluster membership changes from the global zone.

Note—At the time of this writing, Oracle Solaris Volume Manager for Sun Cluster devices work in zone clusters, while other kinds of Oracle Solaris Volume Manager devices and Veritas volume manager devices do not. However, a file system mounted on top of a volume in the global zone can be configured to work in a zone cluster even when the volume cannot.

NAS Devices

Some NAS devices can export storage via the iSCSI protocol, which makes the NAS device appear to be a storage device similar to a disk. In this case the cluster only sees a storage device and manages the iSCSI LUN just like a disk, including fencing support.

It is possible to access NAS units via NFS. In this case the administrator performs an NFS client mount inside the zone cluster node, and data access follows the NFS protocol.

Networks

Zone clusters support network communications over both the public and private networks. Public networks refer to communications outside of the cluster; private networks refer to communications between cluster nodes.

Private Interconnect

From the application perspective, the private interconnect support in the zone cluster is identical to the private interconnect support in the global zone. The system stripes traffic across all available paths using the `clprivnet` driver, and guarantees the data delivery without duplicates as long as at least one

path remains operational. There can be up to six private interconnects. The cluster transparently recovers from the failure of any number of private interconnects, until there are none left.

Zone clusters automate the private interconnect setup. The zone cluster automatically selects a net mask and IP addresses for the zone cluster virtual nodes from the pool of private network net masks and IP addresses that had been established at the time the global cluster was configured. The zone cluster automatically discovers the correct NICs based upon the private interconnect configuration of the global cluster.

The zone cluster uses the same naming conventions as used by the global cluster. Each zone cluster has a separate name space for cluster private network names. When a lookup occurs, the name space for the zone cluster of the requestor is used when the requestor comes from a zone cluster; otherwise, the name space for the global zone is used. The zone cluster uses the same physical private interconnect as that of the global cluster, but uses a unique net mask and unique IP addresses in order to separate the traffic of the zone cluster, and thus provide security isolation.

Zone clusters can also take advantage of the Reliable Datagram Sockets version 1 (RDS v1) when the cluster uses Infiniband (IB) networks for the private interconnects.

Public Network

The zone cluster, like an individual zone, communicates to public networks using an IP address across a NIC. The system administrator in the global zone grants privileges for that combination of IP address and NIC to that zone cluster through the `clzonecluster` command. The system administrator specifies only one NIC of an IP network multipathing (IPMP) group, and the system grants privileges for using that IP address with any NIC in that IPMP group. These networking privileges cannot be changed from within the zone cluster, for security reasons.

The LogicalHost resource is an IP address that is active from only one virtual node at a time. The administrator can switchover the LogicalHost resource from one virtual node to another. The system automatically configures the LogicalHost resource on the node where the application uses that LogicalHost resource. Should a virtual node fail, the system will automatically move the LogicalHost resource along with the dependent application to a surviving node. The zone cluster validates the permissions of the LogicalHost when creating a LogicalHost resource and when activating a LogicalHost resource on a node. Any LogicalHost resource always stays within one zone cluster.

The SharedAddress resource is an IP address that appears to be active for the entire cluster, while in reality the IP address is hosted on one machine and the cluster distributes incoming traffic across the cluster based upon the selected load balancing policy. The system automatically changes the node hosting the IP address for the SharedAddress in case of node failure. The zone cluster validates the permissions of the SharedAddress when creating a SharedAddress resource and when activating a SharedAddress resource on a node. A SharedAddress resource operates strictly within one zone cluster.

Some cluster applications issue `ifconfig` commands to administer the IP addresses used by that cluster application. The zone cluster supports the following `ifconfig` commands:

- `ifconfig -a`

Lists logical interfaces belonging to the zone cluster, physical NICs that were explicitly configured to support an IP address authorized for the zone cluster, NICs that belong to an IPMP group with at least one NIC explicitly configured to support an authorized IP address, and `clprivnet` interfaces.

- `ifconfig addif`

  Adds a logical interface where the IP address is authorized for the zone cluster and the NIC has been explicitly configured to support this IP address or the NIC belongs to an IPMP group where a NIC has been explicitly configured to support this IP address.

- `ifconfig [up | down | plumb | removeif]`

  Performs the specified action on the logical interface. This logical interface must already belong to the zone cluster.

The `ifconfig` command has a very large number of subcommands and options. Zone clusters do not allow any other subcommands and do not allow most options. The goal was to enable commands needed by cluster applications, while at the same time disallowing commands that would permit someone in a zone environment to affect other zones.

Each zone cluster typically requires access to two NICs to form one IPMP group for public network communications, and to two NICs for the private interconnect. If each zone cluster required dedicated physical NICs for these connections, the number of physical NICs required would grow as fast as new zone clusters were deployed. But since zone clusters use network configuration `ip-type=shared`, physical NICs can be shared among zone clusters. This is done in a safe manner such that different zone clusters cannot see each other's traffic.

The other option for network access from a local zone is `ip-type=exclusive`. This approach requires a dedicated NIC for each connection. Zone clusters do not support `ip-type=exclusive`.

Where IP addresses are in short supply, zone cluster nodes can be configured without public IP addresses. Doing so places some restrictions on the zone cluster nodes use. First, without a public IP address, the only way to access the zone is through the `zlogin` command from the global zone in which the zone is configured. Second, a zone cluster configured in this a way cannot host scalable services because such services require a public network to transmit the outgoing packets.

Administration Overview

There are two distinct administrative roles related to zone clusters: *system administrator* and *application administrator*. The system administrator manages the zone cluster platform. The application administrator manages the applications and their resources within a zone cluster. The database administrator is one common application administrator role. The following subsections delineate these administrative activities.

Zone Cluster Administration

The zone cluster platform can only be administered from the global zone. The zone cluster cannot be created from within the zone cluster, because it would not already exist. The other justification for performing zone cluster platform administration from only the global zone is that most zone cluster administration consists of authorizing the use of specific resources within that zone cluster. Security isolation is a primary concern, and therefore security related changes are not allowed from within the zone cluster.

The `clzonecluster` command can be executed from any cluster node to affect the entire cluster in a single command. In other words the `clzonecluster` command supports single point of administration, eliminating the need to issue commands repetitively for large clusters. The `clzonecluster` command combines the functionality of the Oracle Solaris `zonecfg` and `zoneadm` commands, while closely following the format of those commands. The `clzonecluster` command adds support for zone cluster specific features, such as a global context for file systems, devices, and networks. The `clzonecluster` command also introduces a node scope for resources local to a specific virtual node. The `clzonecluster` command automates the configuration of the private interconnect, and uses knowledge of the global cluster to automatically assign reasonable defaults for most Oracle Solaris `sysidcfg` settings, which reduces administrative work. The zone cluster software has interfaces with Oracle Solaris to ensure that misconfigurations cannot occur through `zonecfg` or `zoneadm` commands.

In addition to the command line interface, the Oracle Solaris Cluster Manager for Oracle Solaris software provides a graphical user interface for administration of a zone cluster. Oracle Solaris Cluster Manager is Web-based, so some may prefer to call it a browser-based user interface.

Application Administration

Applications running within a zone cluster can be administered from inside that specific zone cluster or from the global zone.

Oracle Solaris Cluster has a rich framework for managing applications under control of the Resource Group Manager (RGM) subsystem. Applications, file systems, storage devices, and network resources are all identified as RGM resources. Administrators identify functional relationships as *dependencies*. For example, the administrator can specify that application FOO is dependent upon file system `/BAR`, which means that the file system `/BAR` must be mounted successfully before launching application FOO. The administrator can specify location relationships as *affinities*. There are multiple forms of both dependency and affinity relationships. The administrator places the applications and associated resources of a data service into a resource group. Dependencies can be established between resources in the same or different resource groups, and affinities can be established between resource groups. The administrator can further select policies related to availability issues. All Oracle Solaris application management features that are available in the global zone are available for use within a zone cluster.

Zone clusters follow the zone model of security: users from within a zone can only see and affect things within that zone. Each zone cluster has its own name space for application management, thus providing an isolated environment for application management. This avoids name conflicts between

administrators in different zone clusters. Administrators often use working configurations as templates when creating new configurations. Different name spaces make it relatively easy to copy a working configuration.

Multiple data services can be placed within the same zone cluster. Therefore, any number of applications can be placed in a single zone cluster. However, there are often good reasons to place different data services in their own zone cluster. One such justification would be to use the zone level resource controls of a zone cluster to control license fees.

The Oracle Solaris Cluster set of commands for managing applications is the same for both the global zone and the zone cluster. When a command executes within a zone cluster, that command can only manage things within that cluster. When the same command executes within the global zone, the administrator can choose to have the command operate in any zone cluster or the global zone.

The Oracle Solaris Cluster Manager software provides a GUI interface for managing applications in the zone cluster. The Oracle Solaris Cluster Manager runs only from the global zone.

The Oracle Solaris Cluster product provides data service configuration wizards that significantly reduce the administrative work needed to install cluster application. The initial release includes data service configuration wizards that support various Oracle® Real Application Clusters (RAC) database configurations. The administrator accesses the wizard via the Oracle Solaris Cluster Manager. Applications can also be configured without the use of a configuration wizard.

Example Zone Cluster Configuration

This section contains a representative example that creates a typical zone cluster. “Zone Cluster Administration” on page 30 provides a more detailed explanation of the full set of zone cluster administrative tasks.

Preliminary Configuration

Before creating a zone cluster, the system administrator must satisfy a number of prerequisites:

- Install the Oracle Solaris operating system and Oracle Solaris Cluster software on all machines in the cluster.
- Configure the global cluster and boot the global cluster in cluster mode.
- Create any storage volumes that will be used by the zone cluster.
- Create any file systems that will be used by the zone cluster.
- Create the IPMP groups for the NICs that will be used by the zone cluster.
- Determine the encrypted value of the root password that will be used in the newly created zone cluster. (Refer to the `sysidcfg(4)` man page for more information.)

Refer to the Oracle Solaris and Oracle Solaris Cluster 3.3 documentation for instructions on performing these actions.

Zone Cluster Configuration

The command `clzonecluster` used in this example can be executed from any node and operates across the entire cluster. Oracle Solaris Cluster commands typically come in both a long form and an abbreviated short form. The short form of the `clzonecluster` command is `clzc`; both commands take the same parameters and carry out the same tasks.

This section walks through an annotated example of the configuration of a typical zone cluster consisting of two virtual nodes, as shown in Figure 10. This zone cluster is intended to support an Oracle RAC database using a Sun QFS shared file system on Oracle Solaris Volume Manager for Sun Cluster, plus an application running on a failover file system.

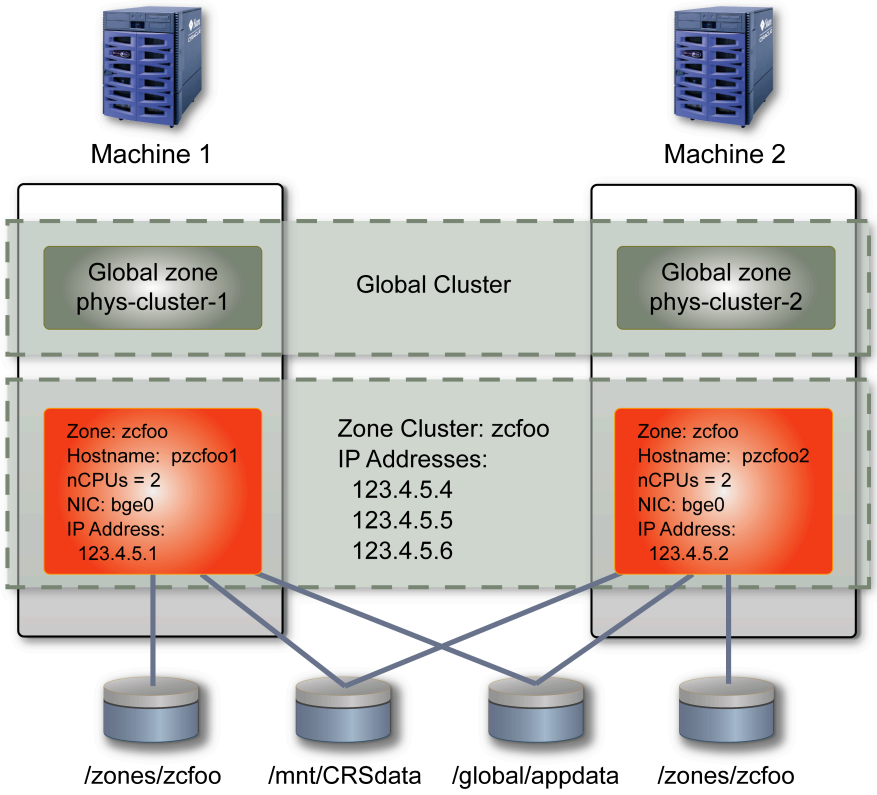

Figure 10. Example zone cluster configuration.

1. The first step is to create a zone cluster named `zcfoo` that will boot whenever the machine boots.

```
# clzonecluster configure zcfoo
clzc:zcfoo> create
clzc:zcfoo> set zonepath=/zones/zcfoo
clzc:zcfoo> set autoboot=true
```

The `create` command creates a zone cluster that consists of sparse-root zones. To configure a whole-root zone cluster, pass the `-b` option to the `create` command. One can also configure a

zone cluster using an existing zone cluster as a template. For example, to configure the zone cluster `zcbar` using `zcfoo` as the template, the following commands can be used:

```
# clzonecluster configure zcbar
clzc:zcbar> create -t zcfoo
```

This will result in the `zcbar` zone cluster having identical properties to `zcfoo`. The properties would then be customized as appropriate for `zcbar`.

2. Add dedicated CPUs. Using a fixed number of CPUs (less than the total amount available) can help reduce licensing costs. This example chooses to use a fixed number of two CPUs dedicated for the use of this zone cluster on each machine.

```
clzc:zcfoo> add dedicated-cpu
clzc:zcfoo:dedicated-cpu> set ncpus=2
clzc:zcfoo:dedicated-cpu> end
```

3. Next, add the nodes of the zone cluster. The physical host machine and the virtual node name must be specified, along with the public IP address and network interface for the node. The following commands first create a virtual node named `pzcfoo1` running on the machine `phys-cluster-1`. Then, a second virtual node `pzcfoo2` is created on machine `phys-cluster-2`.

```
clzc:zcfoo> add node
clzc:zcfoo:node> set physical-host=phys-cluster-1
clzc:zcfoo:node> set hostname=pzcfoo1
clzc:zcfoo:node> add net
clzc:zcfoo:node:net> set address=123.4.5.1
clzc:zcfoo:node:net> set physical=bge0
clzc:zcfoo:node:net> end
clzc:zcfoo:node> end

clzc:zcfoo> add node
clzc:zcfoo:node> set physical-host=phys-cluster-2
clzc:zcfoo:node> set hostname=pzcfoo2
clzc:zcfoo:node> add net
clzc:zcfoo:node:net> set address=123.4.5.2
clzc:zcfoo:node:net> set physical=bge0
clzc:zcfoo:node:net> end
clzc:zcfoo:node> end
```

4. Specify IP addresses. The zone cluster created in this example requires two IP addresses for use by Oracle RAC as Virtual IP (VIP) addresses and one IP address for use with the failover application. These IP addresses are specified in the global context, because these IP addresses can be used on any node in the zone cluster.

```
clzc:zcfoo> add net
clzc:zcfoo:net> set address=123.4.5.4
clzc:zcfoo:net> end

clzc:zcfoo> add net
clzc:zcfoo:net> set address=123.4.5.5
```

```
clzc:zcfoo:net> end

clzc:zcfoo> add net
clzc:zcfoo:net> set address=123.4.5.6
clzc:zcfoo:net> end
```

The system automatically determines the appropriate NIC for an IP address by using the specified subnet. The system looks for a NIC with an IP address on this subnet that has already been configured into this zone cluster. The system knows that this NIC is available for use in this zone cluster, and uses this NIC to host the specified IP address.

The system automatically performs the configuration needed to support use of the cluster private interconnect using the `clprivnet` interface. This includes allocating a subnet and set of IP addresses on the private network. There is no need to do anything additional to obtain that support.

5. Specify the Sun QFS shared file system. The zone cluster created in this example requires a Sun QFS shared file system to support Oracle RAC. The administrator specifies cluster file systems in the global context, because these file systems operate on multiple nodes. The `special` parameter specifies the name of the Sun QFS shared file system as it appears in the master configuration file (MCF). Do not use the raw parameter with a Sun QFS shared file system.

```
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/mnt/CRSdata
clzc:zcfoo:fs> set special=CrsData
clzc:zcfoo:fs> set type=samfs
clzc:zcfoo:fs> end
```

6. Specify the UFS highly available local file system. The zone cluster created in this example requires a UFS highly available local file system for the failover application. The administrator specifies highly available local file systems in the global context, because these file systems operate on multiple nodes, even though they are active on only one node at a time.

```
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/global/appdata
clzc:zcfoo:fs> set special=/dev/md/app/dsk/d20
clzc:zcfoo:fs> set raw=/dev/md/app/rdsk/d20
clzc:zcfoo:fs> set type=ufs
clzc:zcfoo:fs> add options [logging,nodevices]
clzc:zcfoo:fs> end
```

7. Add `sysid` information. Oracle Solaris Zones need information for `sysidcfg`. The `root_password` is the only required field, and this is an encrypted value in the form stored in the `/etc/shadow` file. These parameters will be the same for each zone component of the zone cluster.

```
clzc:zcfoo> add sysid
clzc:zcfoo:sysid> set root_password=<encrypted passwd from /etc/shadow>
```

```
clzc:zcfoo:sysid> end
```

8. Now, verify and commit the zone cluster configuration.

```
clzc:zcfoo> verify
clzc:zcfoo> commit
clzc:zcfoo> exit
```

9. Install the zone. There is a separate step to install the zone.

```
# clzonecluster install zcfoo
```

Installation can take a while. When the installation completes, the zone cluster is installed but not yet booted.

10. Boot the zone cluster.

```
# clzonecluster boot zcfoo
```

The `boot` command will cause the component zones to be initialized, SMF manifests imported, etc. Once the system identification is done, each zone will be rebooted. At this point the zone cluster is ready for use.

Zone Cluster Administration

The `clzonecluster` command supports all zone cluster administrative activity, from creation through modification and control to final destruction. The `clzonecluster` command supports *single point of administration*, which means that the command can be executed from any node and operates across the entire cluster. The `clzonecluster` command builds upon the Oracle Solaris `zonecfg` and `zoneadm` commands and adds support for cluster features.

Zone clusters support a wide range of features. This section explains the process of configuring each of these features. This section is organized according to the primary features being configured. Zone clusters build upon the feature set provided by the native zone. This section covers those features that have been added for zone clusters or differ in some way from the native zone brand type support.

Note—Refer to the Oracle Solaris OS documentation for information about the native zone support.

Node and Cluster-Wide Scope

The native zone has two levels of scope that are used during configuration: *top-level scope* and *resource scope*. When top-level scope is used, the `zonecfg` command specifies properties of the zone. In contrast, the `zonecfg` command also has a resource scope for items such as device, file system, or network resources. The resource scope specifies information about a particular resource.

The `clzonecluster` command expands the scope possibilities. The top-level scope specifies information that applies cluster-wide. For example, the `zonepath` must be the same for all zones constituting the zone cluster. The `clzonecluster` command adds a node scope for specifying information that applies only to a specific node in the zone cluster. The node scope is comparable to the top-level scope of the `zonecfg` command.

The following example illustrates how to enter and leave a node scope of an existing node. The `add node` subcommand automatically enters the node scope for the node being created. The `end` subcommand exits the node scope and returns to the top-level scope. Later examples show tasks that can be performed within a node scope

```
clzc:zcfoo> add node
clzc:zcfoo:node> set physical-host=phys-cluster-3
clzc:zcfoo:node> end
clzc:zcfoo>
```

The `clzonecluster` command supports both the management of resources on a cluster-wide basis and resources local to a specific node. When the administrator enters a resource scope from the top-level or cluster scope, the resource applies cluster wide. When the administrator enters a resource scope from the node scope, the resource applies to that specific node. For example, the administrator specifies a cluster file system resource from a top-level or cluster scope; the administrator specifies a local file system from a node scope.

System identification

The Oracle Solaris Zone cannot become operational until the administrator specifies a set of system identification parameters via the Oracle Solaris `sysidcfg(4)` facility. This information should be specified when initially creating the zone cluster. These properties are the same on each zone of the zone cluster.

The `clzonecluster` command has a `sysid` resource scope for specifying this information for all zones of the zone cluster. In many cases, reasonable default values are supplied by the system, eliminating the need for the administrator to enter basic information that can be automatically determined. For example, the default time zone for the zone cluster is the same as that of the physical cluster.

At this time, the administrator is only required to enter an encrypted root password. Other information, such as the system locale or time zone, can also be entered, if needed.

The following shows an example of specifying numerous fields of `sysidcfg(4)` information for all zones of the zone cluster. This example assumes that the administrator has already started the process of configuring a zone cluster.

```
clzc:zcfoo> add sysid
clzc:zcfoo:sysid> set root_password=<encrypted passwd from /etc/shadow>
clzc:zcfoo:sysid> set name_service="NIS {domain_name=scdev.example.com
  name_server=timber(1.2.3.4)}"
clzc:zcfoo:sysid> set nfs4_domain=dynamic
clzc:zcfoo:sysid> set security_policy=NONE
clzc:zcfoo:sysid> set system_locale=C
clzc:zcfoo:sysid> set terminal=xterms
clzc:zcfoo:sysid> set timezone=US/Pacific
clzc:zcfoo:sysid> end
clzc:zcfoo>
```

Refer to the `sysidcfg(4)` man page for information about these parameters.

Node support

Zone clusters include support for adding and removing nodes from a zone cluster.

Adding a Node

Some set of nodes must be specified when initially creating the zone cluster. The administrator can also add nodes to a zone cluster after initial configuration. When the administrator adds a node to an existing zone cluster, the system automatically applies all global properties of the zone cluster to the added node. This includes information about global resources, such as cluster file systems.

The administrator must specify (1) the global cluster node host name that resides on the same machine as the zone cluster node, and (2) the host name for the zone cluster node. The host name has a specific IP address that a user can specify when attempting to reach the zone cluster node via the network, such as with `telnet`. In the vast majority of situations, the administrator must also specify the network

information that supports access to the zone from the network, which is required to enable logging in to the zone from the network. The zone host name is used when adding entries to an RGM resource group node list to specify allowed locations for an RGM resource group.

The following example adds a zone to the existing zone cluster `zcfoo`. The zone is added to the global cluster node `phys-cluster-3`; the virtual node is assigned the host name `zc-node-3`:

```
# clzonecluster configure zcfoo
clzc:zcfoo> add node
clzc:zcfoo:node> set physical-host=phys-cluster-3
clzc:zcfoo:node> set hostname=zc-node-3
clzc:zcfoo:node> add net
clzc:zcfoo:node:net> set physical=hme0
clzc:zcfoo:node:net> set address=123.4.5.5
clzc:zcfoo:node:net> end
clzc:zcfoo:node> end
clzc:zcfoo> exit
```

Removing a Node

The administrator can remove a node while in the process of configuring a zone cluster. The following command removes the node on the specified physical host:

```
clzc:zcfoo> remove node physical-host=phys-cluster-2
```

When the zone cluster is already configured and operational, the administrator must first use the `clzonecluster(1CL)` command to halt that zone cluster node, and then uninstall the node. Following this, the `remove` subcommand can be used to remove the virtual node.

The following commands illustrate removing a virtual node that is already configured and operational:

```
# clzonecluster halt -n phys-cluster-2 zcfoo
# clzonecluster uninstall -n phys-cluster-2 zcfoo
# clzonecluster configure zcfoo
clzc:zcfoo> remove node physical-host=phys-cluster-2
clzc:zcfoo> exit
#
```

File System Support

Zone clusters support three different kinds of file systems: local file systems, shared Sun QFS file systems, and highly available file systems. The following sections describe the support for each type.

Local File System

A local file system can be mounted on only one node. The local file system is the kind of file system that the native zone supports. The `clzonecluster` command does not currently support the ability to configure a local file system. Instead, the administrator can use `zonecfg` to configure a local file system.

Sun QFS Shared File System

A Sun QFS shared file system is accessible on all nodes of the zone cluster concurrently. The administrator specifies the Sun QFS shared file system in the top-level scope.

Here is an example of configuring a Sun QFS shared file system:

```
# clzonecluster configure zcfoo
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/qfs/ora_home
clzc:zcfoo:fs> set special=oracle_home
clzc:zcfoo:fs> set type=samfs
clzc:zcfoo:fs> end
clzc:zcfoo> exit
#
```

The `dir` entry is the mount point relative to the `zonepath`. The `special` entry is the name of the Sun QFS file system as it appears in the Sun QFS master configuration file (MCF). The raw entry is not used when configuring a Sun QFS file system. The `options` entry is not used with the `clzonecluster` command when configuring Sun QFS file systems; instead specify options in the MCF file and `vfstab` file.

UFS or Veritas VxFS Cluster File Systems

A UFS or VxFS cluster file system is accessible on all nodes of a zone cluster concurrently. The administrator specifies the cluster file system in the top-level scope.

Here is an example of configuring a cluster file system:

```
# clzonecluster configure zcfoo
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/oradata/flash_recovery_area
clzc:zcfoo:fs> set special=/global/zcfoo/orafs1
clzc:zcfoo:fs> set type=lofs
clzc:zcfoo:fs> end
clzc:zcfoo> exit
#
```

The dir entry is the mount point relative to the `zonepath`. The special entry is the mount point of the cluster file system in the global zone. The cluster file system is under the control of a `SUNW.HAStoragePlus` resource, in the zone cluster, whose method run in the global zone to perform the mount operation. The options entry is not used with the clzonecluster command when configuring a cluster file systems; instead specify options in the `vfstab` file.

Highly Available File System

A highly available file system, also called a failover file system, mounts on only one node at a time. The system can move the highly available file system between nodes in response to node failure or administrative command. The administrator specifies the highly available file system in the top-level scope.

Here is an example of configuring a UFS file system as a highly available file system:

```
# clzonecluster configure zcfoo
clzc:zcfoo> add fs
clzc:zcfoo:fs> set dir=/mnt/foo-app
clzc:zcfoo:fs> set special=/dev/md/foo-ds/dsk/d20
clzc:zcfoo:fs> set raw=/dev/md/foo-ds/rdsk/d20
clzc:zcfoo:fs> set type=ufs
clzc:zcfoo:fs> end
clzc:zcfoo> exit
#
```

Zone cluster also supports Oracle Solaris ZFS as a highly available file system. The zone cluster supports Oracle Solaris ZFS at the granularity of the Oracle Solaris ZFS storage pool. Oracle Solaris Cluster moves the entire Oracle Solaris ZFS storage pool between nodes, instead of an individual file system.

The following example configures the Oracle Solaris ZFS storage pool `zpool1` as highly available:

```
# clzonecluster configure zcfoo
clzc:zcfoo> add dataset
clzc:zcfoo:dataset> set name=zpool1
clzc:zcfoo:dataset> end
clzc:zcfoo> exit
#
```

Storage Device Support

Zone clusters support the direct use of storage devices, including local devices and cluster-wide devices.

Local Device

A local storage device is a device that can only connect to one machine. The `clzonecluster` command currently does not support local devices. Rather, the administration can configure a local device using the `zonecfg` command.

Cluster-Wide Devices

A cluster-wide storage device is a device that can be used by multiple nodes in one of two ways. Some devices, such as Oracle Solaris Volume Manager for Sun Cluster devices, can be used concurrently by multiple nodes. Other devices can be used by multiple nodes, but only one node can access the device at any given time. An example is a regular SVM device with connections to multiple machines.

The administrator configures cluster-wide devices in the top-level context. Wild cards can be used when identifying the device, as shown in the following example that configures a set of Oracle Solaris Volume Manager for Sun Cluster devices:

```
# clzonecluster configure zcfoo
```

```
clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/md/oraset/dsk/*
clzc:zcfoo:device> end

clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/md/oraset/rdsk/*
clzc:zcfoo:device> end

clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/md/1/dsk/*
clzc:zcfoo:device> end

clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/md/1/rdsk/*
clzc:zcfoo:device> end
clzc:zcfoo:> exit
```

Notice that both the *logical* and *physical* device paths must be specified when exporting Oracle Solaris Volume Manager for Sun Cluster metasets and/or metadevices to a zone cluster. In the above example, the *set number* of `oraset` is 1. The set number of a metaset can be found by running the `ls -l` command and specifying the set name, as shown in the following example.

The output from this command displays a symbolic link, which includes the set number.

```
# ls -l /dev/md/oraset
```

DID devices can also be configured. The following example configures the DID device `d10`:

```
# clzonecluster configure zcfoo
clzc:zcfoo> add device
clzc:zcfoo:device> set match=/dev/did/*dsk/d10s*
clzc:zcfoo:device> end
clzc:zcfoo:> exit
#
```

Networking Support

The zone cluster includes support for both public and private networking, as described in the following sections.

Private Interconnect

The private interconnect refers to the network connections between the nodes of the cluster. The system can automatically configure the zone cluster to support communications across the private interconnect. The system automatically selects a subnet from the pool of private network subnets specified when the physical cluster was installed. The system then assigns an IP address for each virtual node of the zone cluster. The system software isolates the private networks of different zone clusters

into separate name spaces. The result is that each zone cluster effectively has its own Oracle Solaris Cluster private network driver (`clprivnet`) support, while sharing the same physical networks.

If a zone cluster does not need private interconnect support, the administrator can disable, or turn off, this feature. A zone cluster that only supports one failover application is one example of a configuration that does not require a private interconnect.

The following example shows how to turn off this feature when creating the zone cluster. The property must be set in the top-level context:

```
clzc:zcfoo> set enable_priv_net=false
```

Note—This private interconnect feature cannot be changed on a running zone cluster.

Public Network

The public network refers to network communications outside of the cluster. Zone clusters include both local network and cluster-wide network support.

- *Local network support*

  A local network resource is used exclusively by one node. The `clzonecluster` command currently does not support a local network resource. Instead, the administrator can use the `zonecfg` command to configure a local network resource.

- *Cluster-wide network support*

  A network resource can be configured for use on multiple nodes of the cluster. An IP address can be hosted on only one node at a time. However, this kind of network resource can move between virtual nodes at any time.

  The following entities require this kind of network resource:

  - Logical Host
  - Shared Address
  - Oracle RAC Virtual IP (VIP) Address
  - An IP address directly managed by a cluster application (using commands such as `plumb`, `unplumb`, `up`, `down`, and `addif`).

The following example configures a network resource that can be used across the cluster:

```
# clzonecluster configure zcfoo
clzc:zcfoo> add net
clzc:zcfoo:net> set address=123.4.5.5
clzc:zcfoo:net> end
clzc:zcfoo> exit
#
```

Notice that the network *interface* cannot be specified by the user for a cluster-wide network resource.

The system determines the subnet of the specified network resource. The system will allow the specified IP address to be used on either (1) any NIC that has already been authorized for use in this zone cluster; or (2) any NIC in an IP network multipathing (IPMP) group that has already been authorized for use in this zone cluster. Normally, there is a network resource configured for use in each zone for such purposes as login. This follows the stringent zone security policy of checking both IP address and NIC.

Boot and Halt Operations

The administrator can manually boot or halt the entire zone cluster at any time just like a physical cluster. The example zone cluster will automatically boot after the node boots and halt when the node halts. The following commands boot and halt the entire zone cluster on all configured nodes.

```
# clzonecluster boot zcfoo
# clzonecluster halt zcfoo
```

The administrator can boot or halt individual nodes of the zone cluster. Typically the administrator halts and reboots individual nodes for administrative tasks, such as software upgrades. The following examples boot and halt the specified node:

```
# clzonecluster boot -n <base-cluster-node> zcfoo
# clzonecluster halt -n <base-cluster-node> zcfoo
```

Note—A zone component of a zone cluster can only be booted in cluster mode when the machine hosting the zone component is booted in cluster mode.

The `cluster shutdown` command can also be used to halt a zone cluster. Executing `cluster shutdown` in the global zone halts all zone clusters and the physical cluster. Executing `cluster shutdown` in a zone cluster halts that particular zone cluster, and is equivalent to the use of the `clzonecluster` command to halt the entire zone cluster.

Delete Operation

Before a zone cluster can be deleted, all the resource groups and their associated resources must be deleted. Then the zone cluster must be halted and uninstalled prior to deletion. The administrator executes the following commands to destroy the zone cluster:

```
# clzonecluster halt zcfoo
# clzonecluster uninstall zcfoo
# clzonecluster delete zcfoo
```

Displaying Zone Cluster Information

Two subcommands, `status` and `list`, are used to obtain information about a zone cluster. The `list` subcommand displays a list of zone clusters configured on the system.

The `status` subcommand displays information about zone clusters, including the host name and status for each node. The following example displays the information for a particular zone cluster:

```
# clzonecluster status -v zcfoo
=== Zone Clusters ===

--- Zone Cluster Status ---

Name   Node Name    Zone HostName    Status      Zone Status
----   ---------    -------------    ------      -----------
zcfoo   phys-cluster-1    giggles-1    Online    Running
        phys-cluster-2    giggles-2    Online    Running
        phys-cluster-3    giggles-3    Online    Running
```

Clone Operation

The `clone` subcommand clones a zone cluster, similar to the Oracle Solaris `zoneadm` clone command. Before executing the clone subcommand, the administrator must first configure a zone cluster. The `clone` subcommand uses the referenced zone cluster to determine how to install this particular zone cluster. The system can usually install a zone cluster more quickly using the clone subcommand.

Other Zone Subcommands

The Oracle Solaris `zonecfg` and `zoneadm` support additional subcommands. The `clzonecluster` command supports most, but not all, of these additional subcommands.

Specifically, the `clzonecluster` command does not support the following subcommands:

- `attach`
- `detach`

The `clzonecluster` command supports the other subcommands supported by the Oracle Solaris `zonecfg` and `zoneadm` commands. The difference is that the `clzonecluster` command applies the subcommand to all zones of the zone cluster. Refer to the `zonecfg` and `zoneadm` man pages for more information.

Note—The initial release of zone clusters is based upon the Oracle Solaris 10 5/08 OS. Check with the Oracle Solaris Cluster documentation and release schedule for information on support for additional subcommands.

Oracle Solaris OS Command Interaction

The Oracle Solaris OS contains commands for managing zones. Naturally, the Oracle Solaris `zonecfg` and `zoneadm` commands cannot manage items that do not exist in a single-machine zone, such as global file systems. The Oracle Solaris OS commands do not manage resources that must be the same on multiple nodes, such as the security related properties. Oracle Solaris OS commands are used to manage some local features of a zone component of a zone cluster.

For example, the `zonecfg` command can be used to configure the following:

- Local file system
- Local Oracle Solaris ZFS pool
- Local device
- Local network resource
- Resource control properties on one node

In contrast, the `zonecfg` command cannot configure the following:

- Zone name
- Zone path
- The `limitpriv` property
- Solaris Resource Manager pool
- The `inherit-pkg-dir` property
- Cluster wide resources, such as a cluster file system

The `zoneadm` command can boot and halt the zone on the local node or list the status of the zones on the local node.

Zone Cluster Administrative GUIs

Oracle Solaris Cluster provides both the text-based interactive `clsetup` command interface and the Oracle Solaris Cluster Manager for Oracle Solaris browser-based graphical user interface (GUI) for administrative actions. The Oracle Solaris Cluster Manager browser-based GUI has been enhanced so that the administrator in the global zone can both view and administer resource groups and resources in zone clusters. Oracle Solaris Cluster Manager does not run in a zone cluster.

Summary

Zone clusters provide secure environments for controlling and managing cluster applications. The cluster applications see this environment as a dedicated private cluster.

While this report is quite extensive, please refer to the Oracle Solaris Cluster documentation for complete information about this feature.

About the Author

Tim Read is a Software Developer for the Oracle Solaris Cluster Group. His main role is the development of the Oracle Solaris Geographic Edition product. He has written a number of whitepapers and books on high availability and disaster recovery including Oracle Solaris Cluster

Essentials, published in 2010. He has a B.Sc. in Physics with Astrophysics from the University of Birmingham in the UK.

This whitepaper has been updated from the original source material written by Dr. Ellard Roush.

Acknowledgements

The development of the zone cluster feature was done by the project team, and recognition must be extended to all team members who contributed in a variety of ways: Zoram Thanga, Pramod Rao, Tirthankar Das, Sambit Nayak, Himanshu Ashiya, Varun Balegar, Prasanna Kunisetty, Gia-Khanh Nguyen, Robert Bart, Suraj Verma, Harish Mallya, Ritu Agrawal, Madhan Balasubramanian, Ganesh Ram Nagarajan, Bharathi Subramanian, Thorsten Frueauf, Charles Debardeleben, Venkateswarlu Tella, Hai-Yi Cheng, Lina Muryanto, Jagrithi Buddharaja, Nils Pedersen, and Burt Clouse.

References

TABLE 20. REFERENCES FOR MORE INFORMATION

DESCRIPTION	URL
Oracle Solaris Cluster	http://www.oracle.com/us/products/servers-storage/solaris/cluster-067314.html
Sun Cluster Wiki	http://wikis.sun.com/display/SunCluster/Home
"Configuring a Zone Cluster," Sun Cluster Software Installation Guide for Solaris OS	http://download.oracle.com/docs/cd/E19680-01/821-1255/gqzen/index.html
Oracle Solaris Cluster 3.3 Documentation Center	http://download.oracle.com/docs/cd/E19680-01/821-1261/index.html
Oracle Solaris Cluster Concepts Guide for Solaris OS	http://download.oracle.com/docs/cd/E19680-01/821-1254/index.html
Oracle Solaris Cluster System Administration Guide for Solaris OS	http://download.oracle.com/docs/cd/E19680-01/821-1257/index.html
System Administration Guide: Solaris Containers—Resource Management and Solaris Zones	http://download.oracle.com/docs/cd/E19253-01/817-1592/index.html
clzonecluster(1CL) man page	http://download.oracle.com/docs/cd/E19680-01/821-1263/6nm8r5jgu/index.html
sysidcfg(4) man page	http://download.oracle.com/docs/cd/E19253-01/816-5174/6mbb98ujq/index.html
zoneadm(1M) man page	http://download.oracle.com/docs/cd/E19253-01/816-5166/6mbb1kqoa/index.html
zonecfg(1M) man page	http://download.oracle.com/docs/cd/E19253-01/816-5166/6mbb1kqoc/index.html
Oracle Partitioning and Pricing Policy	http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf

Zone Clusters—How to Deploy Virtual Clusters and Why
Feb 2011
Author: Tim Read
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

SOFTWARE. HARDWARE. COMPLETE.