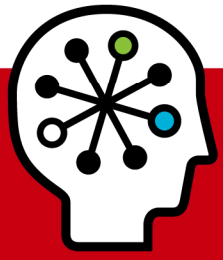
Sample Configurations with SGeRAC and Oracle RAC 10gR2

Version 1.6, March 2009



Introduction 3
Audience 3
Terms and definitions 3
Additional information 4

Planning considerations 4
Capacity planning 4
Processor capacity 4
Memory 5
Network: clients 5
Network: cluster interconnect 5
Storage 5
Failover time requirements 5
Planning for high availability 6
Public network 6
Network for cluster communication 7
Storage high availability 12
Multiple nodes 12
Power 12
Storage 13
Oracle Clusterware requirements: OCR and voting disk 13
RAC instance data files 13
Oracle binaries (Clusterware and RAC) 13
Archive log files 13

Prerequisites 14
Software 14
Server hardware 14
Network 14
Storage 14
Cluster for SLVM (eenie and meenie) 15
Cluster for CFS (minie and mo) 16

Sample configurations 16
Configuring Oracle RAC 10g on SLVM 16
Assumptions for this sample configuration 17
Creating a SGeRAC cluster with SLVM for Oracle RAC 10g 17
Prerequisites for Oracle 10g (sample installation) 19
Installing and configuring Oracle 10g Clusterware on local file system 22
Installing Oracle RAC 10g on local file system 22
Creating a RAC demo database on SLVM 23
Configuring Serviceguard packages using Serviceguard Extension for RAC 24
Configuring Serviceguard packages without Serviceguard Extension for RAC 26

Configuring Oracle RAC 10g on CFS 29
Assumptions for this sample configuration 29
CFS and ODM requirement 29
Creating a SGeRAC cluster with CFS for Oracle 10g 29
Prerequisites for Oracle 10g (sample installation) 32
Installing and configuring Oracle 10g Clusterware on local file system 34
Installing Oracle RAC 10g on CFS 34
Configuring ODM 35
Creating RAC demo database on CFS 36
Configuring Serviceguard packages with Serviceguard Extension for RAC 37
Cluster start and stop 40
Start and stop Oracle Clusterware 10g 40
Start and stop Oracle RAC 10g Instance 40

Appendix 41
Sample configuration for SLVM with Serviceguard Extension for RAC 41
Cluster configuration File for Cluster (eenie and meenie) 41
Package configuration for SLVM (Oracle Clusterware MNP) 41
Package control script for CFS (Oracle Clusterware MNP) 41
Serviceguard Extension for RAC configuration for Oracle Clustware 42
Sample configuration for SLVM without Serviceguard Extension for RAC 42
Cluster configuration file for cluster (“clm.asc”) 42
Package configuration file for node eenie for SLVM (“crs_eenie_pkg.conf”) 43
Package control file for node eenie for SLVM 43
Package configuration file for node “meenie” for CRS for SLVM 43
Package control file for node “meenie” for CRS for SLVM 43
Sample “cmviewcl –v” output 44
Sample configuration for CFS 46
Cluster configuration Ffile for cluster (minie and mo) 46
Package configuration for CFS (Oracle Clusterware MNP) 46
Package control script for CFS (Oracle Clusterware MNP) 47
Serviceguard Extension for RAC configuration for Oracle Clusterware 47
Sample scripts used by package control scripts 47
Sample script to start or stop Oracle Cluster Software (“cssd.sh”) 47
Document revision history 52

For more information 52
HP documentation 52
Oracle documentation 53

Introduction

This document discusses the various aspects of architecting, planning, and implementing an Oracle Real Application Cluster (RAC) 10g Release 2 solution on HP-UX 11i with Serviceguard and Serviceguard Extension for RAC (SGeRAC). The document also includes a brief description about planning considerations and presents two sample configurations for Oracle Real Application Cluster (RAC) 10g Release 2 with Serviceguard, Serviceguard Extension for RAC (SGeRAC), Shared Logical Volume Manager (SLVM), and Cluster File System (CFS).

This document also provides step by step installation instructions for creating an SGeRAC cluster, creating shared storage using Shared Logical Volume Manager (SLVM) and CFS, installing Oracle Clusterware (OC) and RAC, creating a demo database, and creating Serviceguard packages to synchronize start and stop of the complete solution stack.

Audience

The target audiences are those who are interested in architecting, planning, and implementing SGeRAC and Oracle RAC 10g clusters.

The reader should be familiar with Serviceguard (SG), Serviceguard Extension for RAC (SGeRAC), Oracle RAC 10g software, Shared Logical Volume Manager (SLVM), Symantec Veritas Cluster Volume Manager (CVM), Symantec Veritas Cluster File System (CFS), HP-UX 11i v2, and HP-UX 11i v3.

Terms and definitions

- APA – Auto Port Aggregation provides bonding of multiple networking interface cards where traffic is distributed to all interface cards.
- APA/Hot Standby – Auto Port Aggregation Hot Standby mode provides high availability through bonding of a primary and a standby interface card. Traffic is not distributed.
- CFS – Cluster File System allows multi-system shared access to common file system.
- CSS – Cluster Synchronization Service is a component of Oracle Clusterware that maintains Oracle cluster membership and heartbeat.
- CSS-HB – Cluster Synchronization Service heartbeat traffic
- CVM – Cluster Volume Manager allows multi-system shared access to volumes.
- GAB – Group Membership Service/Atomic Broadcast manages cluster membership and cluster communication for Symantec Veritas CFS 4.1/5.0 and CVM 4.1/5.0.
- GMS – Group Membership Service refers to HP's implementation of the NMAPI2 API on HP-UX with SGeRAC that provides group membership notification and process monitoring facility to monitor group status.
- HA – High Availability refers to configurations that are resilient to single failure.
- JBOD – Just a Bunch of Disks refers to single disk or a set of disk in disk enclosures that do not provide RAID capability for HA.
- LLT – Low Latency Transport provides kernel-to-kernel communications at link level and monitors network connections for Symantec Veritas CFS 4.1/5.0 and CVM 4.1/5.0. Distributes Symantec Veritas traffic amount network connections and maintains Symantec Veritas heartbeat.
- MNP – Multi-node package, a Serviceguard package that runs on multiple nodes at the same time and can be independently started and halted on individual nodes.
- NIC – Network Interface Card, host bus adapter for network communications, for example Ethernet card.
- OC – Oracle Clusterware can run in conjunction with Serviceguard Extension for RAC and provides Oracle cluster membership and resource management services.

- OCR – Oracle Cluster Registry is shared storage used to keep Oracle cluster and configuration information.
- ODM – Oracle Disk Manager is a standard API specified by Oracle for database I/O.
- RAC – Real Application Cluster enables a multi-instances concurrent shared access database.
- RAC-DB-IC – Real Application Cluster Interconnect traffic for both Global Cache Service and Global Enqueue Service.
- RAID – Redundant Array of Independent Disks refers to disk storage that provides HA through redundancy within an array of disks by internal mirroring or use of parity disks.
- RIP – Serviceguard Relocatable IP Address user for client application access and failovers with package
- SG-HB – Serviceguard Heartbeat traffic.
- SGeRAC – Serviceguard Extension for RAC extends Serviceguard to support Oracle RAC.
- SLVM – Shared Logical Volume Manger allows multi-system shared access to LVM volumes for RAC.
- VIP – Virtual IP address is used by OC to configure access to Oracle clients and for remote failover to reject client connections.
- Voting Disk – Shared storage used by Oracle Clusterware as vote tie breaker and for disk based heartbeat.

Additional information

The audience is encouraged to use these additional documentations in conjunction with this document. Sources are listed in the For More Information section.

- HP Serviceguard Storage Management Suite Version Release Notes
- Serviceguard Version A.11.19 Release Notes
- Serviceguard Extension for RAC Version A.11.19 Release Notes
- Managing Serviceguard Sixteenth Edition
- Using Serviceguard Extension for RAC Manual Sixth Edition
- Oracle Clusterware and Oracle Real Application Clusters Installation Guide version 10g Release 2 (10.2)
- Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide version 10g Release 2 (10.2)
- Symantec Veritas Storage Foundation 5.0 for Oracle RAC Configuration Guide Extracts for HP Serviceguard Storage Management Suite Extracts
- Symantec Veritas Storage Foundation 4.1 for Oracle RAC HP Serviceguard Storage Management Suite Extracts

Planning considerations

Proper planning is a requirement for high availability configurations. There are important considerations when deciding the cost-benefit tradeoffs.

Capacity planning

Proper capacity planning ensures sufficient resources are available to meet the expected service levels.

Processor capacity

In a cluster environment, cluster members maintain heartbeat traffic between nodes. Insufficient CPU processing capacity affects heartbeat processing and thus affects the heartbeat timeout. A larger timeout means a longer cluster reconfiguration time and a longer time before a node failure or network failure can be detected.

Memory

Sufficient physical memory should be available for all processes. Insufficient memory may result in swapping activities that affect the CPU processor availability to components that have timed heartbeat communications. Generally, one should consider sufficient memory to reduce paging activities, fit Oracle System Global Area (SGA) into main memory, and allow for user processes.

Network: clients

Insufficient bandwidth on the client network affects availability to the client.

Network: cluster interconnect

Insufficient bandwidth from the cluster interconnect affects communication between cluster components.

Storage

Sufficient storage bandwidth and storage space are required to maintain optimal database service. Additionally, sufficient space should be allocated for database archives for recovery purposes.

Failover time requirements

In a RAC configuration with SGeRAC, each node may have concurrent access to the database. The database service is accessible from all nodes and each node provides a client connection endpoint (IP address, port, and listener). When one node fails, clients can connect to another node for services. The client connection endpoint does not need to failover for the clients to continue service. However, even though an alternate connection endpoint is available, upon certain failures (for example node or network) and until failure detection and recovery, new client connections may not connect or the database service may be unavailable while the cluster goes through reconfiguration and/or recovery.

The failover time is the time between when a failure occurs to when the service is once again available to the client. A failover time requirement is important for the following reasons:

How fast the clients reconnect.

- On local LAN failover, this depends on detection time and the local LAN failover scheme.
- On remote failover, this depends on whether clients are enabled with Oracle Fast Application Notification (FAN), how fast cluster reconfiguration happens, how soon the VIP address fails over, and how soon the client connection times out.

How fast the cluster and RAC go through reconfiguration before database service resumes.

- On node failure, the reconfiguration time depends on the Serviceguard heartbeat timeout, the number of nodes, and the type of quorum device used.
- On cluster network interconnect failures, the database service availability depends on how soon the interconnect failure is discovered, speed of recovery actions by affected components (for example SG, GMS, SLVM, CVM, CFS, CSS, and/or RAC), and database recovery.
    - On a complete SG cluster interconnect failure, SG sees the failure within the SG heartbeat timeout.
    - With SG/CFS, GAB/LLT and SG shared the same networks and SG sees the interconnect failure within SG heartbeat timeout.
    - With CSS traffic, on configurations where CSS and SG share the same interconnect network, SG sees the failure within the SG heartbeat timeout. If CSS traffic is on a SG monitored network, SG can be configured to take actions via SG packages and optionally use Cluster Interconnect Subnet Monitoring[1]. If CSS traffic is on a non-SG monitored network, CSS sees the interconnect failure within the CSS timeout.

[1] Cluster Interconnect Subnet Monitoring is available starting SGeRAC A.11.18.

- With RAC traffic, if the interconnect is not configured to be monitored and acted upon by the other components, RAC discovers the interconnect failure within the Instance Membership Recovery (IMR) timeout.

The failover time requirement determines important timeouts, such as Serviceguard heartbeat timeout, network polling intervals, and cluster interconnect monitoring.

Planning for high availability

A properly configured high availability configuration should survive single failures and continue to operate.

Public network

The following describes the two levels of client public network high availability (HA): redundant components and client failover.

Redundant network interfaces and switches with local LAN failover by Serviceguard or bonding by Auto-Port Aggregation (APA) protects against single point network failures.

Client failover is needed with failures that impact existing and new client sessions. These failures include node failures (such as those caused by a power failure) and network failures (for example all redundant network interface/link failed). Protection is available at three levels: Oracle Fast Application Notification (FAN), remote VIP failover, and client connect timeout. Clients that are FAN integrated or using the FAN API may interrupt existing sessions and failover. Remote VIP failover is useful for non-FAN clients attempting to connect to the local node to avoid the TCP connect timeout. The client connect timeout is useful when client connect takes a long time for whatever reason.

VIP high availability

This section describes high availability for VIP.

Note:
Previously, Oracle virtual IP address (VIP) and Serviceguard relocatable IP address (RIP) should not exist on the same subnet on the same node due to potential collisions on IP address configuration. This issue has been addressed in Oracle 10.2.0.2 for Integrity platform and in Oracle 10.2.0.3 for HP 9000 platforms.

Serviceguard local LAN failover mechanism – preferred choice

For client public network HA in a SGeRAC configuration, the preferred method for network HA is to use Serviceguard primary and standby. Serviceguard monitors the redundant network and additional APA software is not required.

When the client network is configured with Serviceguard local LAN failover, Serviceguard performs the local LAN failover and Oracle Clusterware (OC) configures the VIP after Serviceguard local LAN failover. Since OC performs monitoring and manages the VIP address, client connectivity maybe unavailable until OC detects the outage and configures the VIP address on the local node[2].

[2] See Doc ID 296874.1 Configuring the HP-UX Operating System for the Oracle 10g VIP at https://metalink.oracle.com/ (Oracle MetaLink account required)

Local LAN failover using APA

When APA is used where the network interface cards are bonded, APA provides traffic distribution and load balancing capability among multiple physical network interface cards (NIC) or links. Load balance may be a benefit which is desirable to configurations where a single interface is insufficient to handle the network traffic. When a physical NIC or link fails, APA provides HA by distributing traffic among remaining NIC or links. One virtual link is presented to OC and APA network load balancing is transparent to OC. APA requires the same type of NIC. Since APA network connections go to the same switch, a switch failure means outage of the client network.

When APA/Hot Standby is used, APA/Hot Standby provides the primary-to-hot-standby failover by rerouting traffic from failed primary link to hot standby link. APA/Hot Standby does not load balance. Serviceguard does not monitor this network. One virtual network link is presented to OC and the physical failover is transparent to OC because the same virtual network link remains available. Both NICs must be the same type as in APA.

Remote failover

OC fails over the VIP address to a surviving node on a catastrophic failure such as node failure, instance failure, storage failure, or network failure.

Network for cluster communication

Serviceguard, OC, and each RAC instance maintain communication with peers on other nodes. When communication is broken, either through network partition or node failure, each of these components needs to reform its membership and eject non-members as needed.

In CFS and CVM (4.1/5.0) configurations, Symantec Veritas' Group Membership Service/Atomic Broadcast and Low Latency Transport (GAB/LLT) also uses the cluster interconnect for peer to peer communications.

The categories of traffic between nodes are distinguished as follows:

- SG-HB – Serviceguard heartbeat and communications traffic. Supported over single or multiple subnet networks.
- CSS-HB – Cluster Synchronization Service (CSS) heartbeat traffic and communications traffic for Oracle Clusterware. CSS-HB uses a single logical connection over a single subnet network.
- RAC-DB-IC – RAC instance peer to peer traffic and communications for Global Cache Service (GCS) and Global Enqueue Service (GES), formerly Cache Fusion (CF) and Distributed Lock Manager (DLM). Network HA is provided by the HP-UX 11i platform (Serviceguard or APA bonding).
- ASM-IC – Applicable only when using Automatic Storage Management (ASM). ASM instance peer to peer traffic. When it exists, ASM-IC should be on the same network as CSS-HB. Network HA is required either through Serviceguard failover or APA bonding.
- GAB/LLT – Applicable only when using CFS/CVM. Symantec cluster heartbeat and communications traffic. GAB/LLT communicates over link level protocol (DLPI) and supported over Serviceguard heartbeat subnet networks, including primary and standby links. GAB/LLT is not supported over APA or virtual LANs (VLAN).

Note that each category maintains its own timeout for which nodes may be evicted from its respective membership.

The interconnect network requires HA configurations. When a single network failure occurs, for example LAN card or switch failures, all the cluster nodes continue to operate. Without HA, a single network failure results in a network partition between the nodes and evicts nodes are halted.

Using Serviceguard primary and standby links is the preferred HA model to provide HA for the cluster communications interconnect network HA. With redundancy through Serviceguard primary and standby, Serviceguard monitors the network and performs local failover if the primary network becomes unavailable.

General principles

It is preferred to have all interconnect traffic for cluster communications on a single heartbeat network with a standby interface so that Serviceguard will monitor the network and resolve interconnect failures by cluster reconfiguration. This preferred configuration is the recommended common configuration.

The following examples are instances when it is not possible to place all interconnect traffic on the same network:

- RAC GCS (cache fusion) traffic may be very high, so a separate network for RAC-DB-IC may be needed[3]. One RAC-DB-IC may interfere with another RAC-DB-IC on the same cluster. RAC-DB-IC may also interfere with heartbeat traffic.
- Some networks are not supported by CFS/CVM, so the RAC-DB-IC traffic may be on a separate network.
- Certain configurations for fast re-configuration require at least SG heartbeat network, while CSS-HB and and RAC-DB-IC does not support multiple network for HA purposes. Note: Starting A.11.19, the faster failover capability is in the base Serviceguard product.

In these cases, you may see a longer time to recover some network failures beyond those protected by primary and standby, unless special logic is developed.

Cluster Interconnect Configurations for SLVM

Configurations with SLVM have configurations that are combinations of SG-HB, CSS-HB and RAC-DB-IC on Ethernet. The following figures show several examples on how cluster interconnect traffic can be distributed. This is not an exhaustive list.

Figure 1. Preferred: SG-HB, CSS-HB, and RAC-DB-IC on same subnet

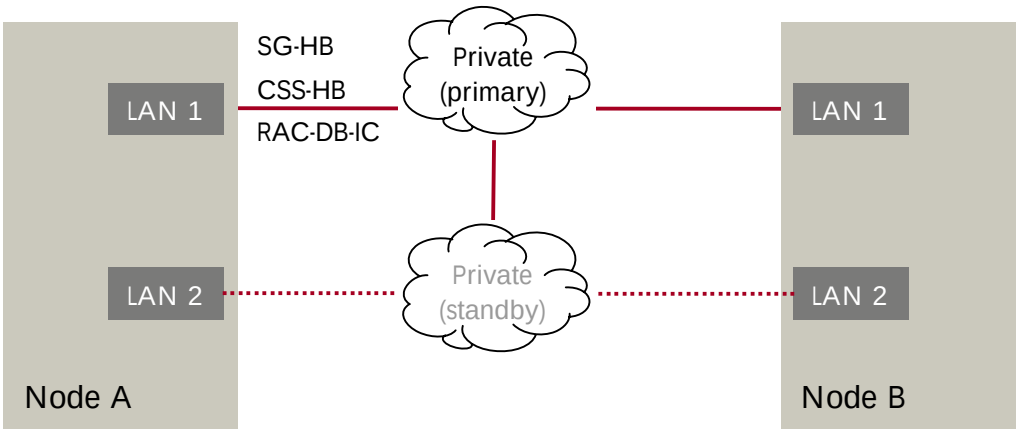


Figure 1 shows a common configuration where all inter-cluster traffic flows through the primary network. This configuration is suitable for most common installations where the RAC traffic does not saturate the network and negatively affect other components. There is one network and the network

[3] See CLUSTER_INTERCONNECTS, page 5-11, Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide version 10g Release 2 (10.2) (http://download-west.oracle.com/docs/cd/B19306_01/rac.102/b14197.pdf)

has sufficient bandwidth. If the primary network fails, Serviceguard performs a local LAN failover to use the standby network. Node failure is detected when Serviceguard misses heartbeats.

Configurations with heavy RAC-DB-IC traffic may place a limit on how aggressive the Serviceguard heartbeat timeout can be used since SG-HB may not be processed in time. Therefore, a longer Serviceguard heartbeat timeout may be needed to avoid false cluster reconfigurations.

Figure 2. Dual SG-HB with CSS-HB and RAC-DB-IC on same subnet

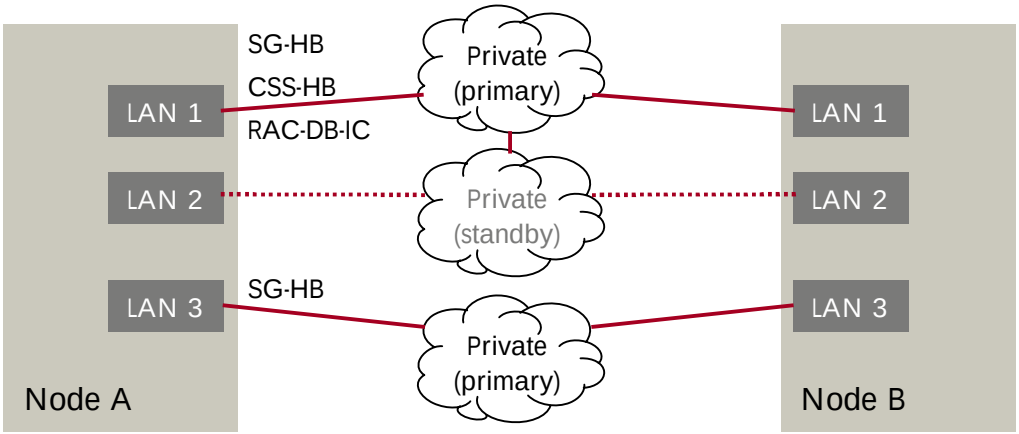


Figure 2 shows one method to overcome the heavy RAC-DB-IC traffic affecting SG-HB. This configuration uses two SG-HB networks. Two SG HB networks are required in faster failover configurations. The Serviceguard heartbeat timeout can be as aggressive as the software configurations allow. The subnet for CSS-HB and RAC-DB-IC has redundancy with primary and standby adapters and switches configured. If the primary (lan1) fails, Serviceguard performs a local LAN failover to the standby (lan2). If a node fails, Serviceguard detects the failure from the loss of SG-HB.

If both primary (LAN 1) and standby (LAN 2) fail, Serviceguard logs the failure but will not take action unless Serviceguard packages with monitored subnets are configured to take action, for example node failfast. The advantage for Serviceguard to take action on a monitored subnet is that the failure detection can be faster than the CSS-HB timeout, and thus recovery action can be quicker. Use of Serviceguard subnet monitoring has a limitation where if all interconnect fails (for example primary and standby switch failed at the same time), all the nodes are halted. If there is a concern with simultaneous failure of both switches, starting with SGeRAC A.11.18, Serviceguard supports cluster interconnect subnet monitoring.

If there are no Serviceguard packages configured to take action, CSS-HB timeout occurs and CSS goes through reconfiguration and reboot the node that is not part of the CSS cluster membership. When the node reboots, Serviceguard will reform with the new membership.

Figure 3. Single SG-HB with CSS-HB and RAC-DB-IC on separate subnet

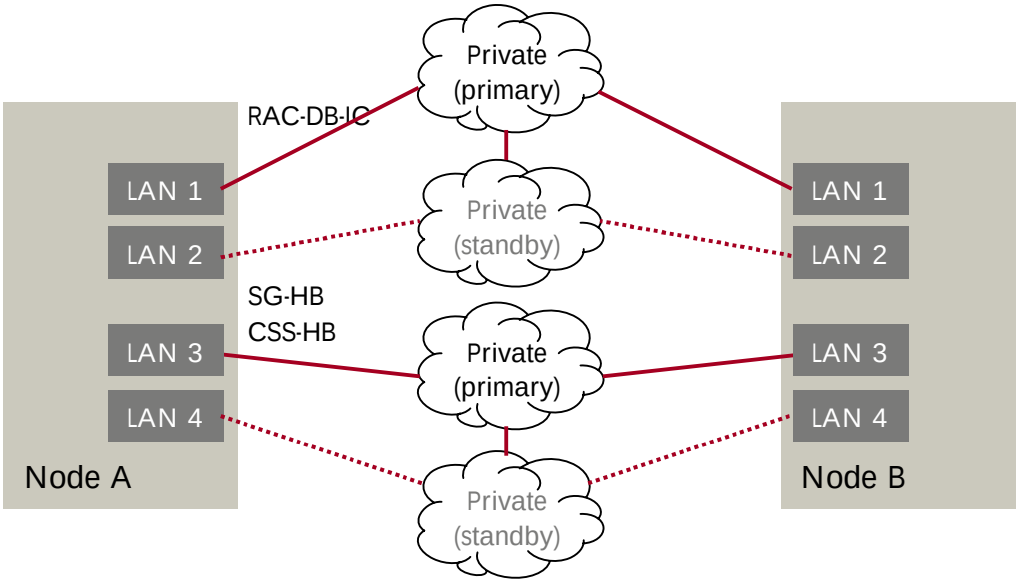


Figure 3 is a variation of figure 2 and shows where the CSS-HB residing on the same subnet as SG-HB. The RAC-DB-IC is on a separate network and thus do not affect the HB traffic. If the primary (lan1) fails, Serviceguard performs local LAN failover. If both primary (lan1) and standby (lan2) fails, RAC Instance Membership Recovery (IMR) reforms and evicts suspect nodes. Eviction reboots the node. The IMR timeout is by default longer than the Serviceguard reconfiguration time and CSS reconfiguration time.

The advantage of this configuration is that RAC instance traffic is separate from heartbeat traffic. The RAC traffic does not interfere with heartbeat traffic. A Serviceguard package can be configured to monitor the RAC-DB-IC subnet. If the RAC-DB-IC subnet fails (both primary and standby), the Serviceguard package can be configured to shutdown the RAC instance in order to avoid RDBMS IMR timeout, and only the specific RAC instance is affected. This configuration allows halting the RAC instance rather than evicting and halting the node. Use of Serviceguard subnet monitoring has a limitation where if all interconnect fails (for example primary and standby switch failed at the same time), all the instances are halted. If there is a concern with simultaneous failure of both switches, starting with SGeRAC A.11.18, Serviceguard supports cluster interconnect subnet monitoring.

Cluster Interconnect Configurations for CFS and CVM

Configurations with CFS and CVM add GAB/LLT as an additional set of traffic.

Figure 4. Preferred: single subnet with Ethernet primary and standby including GAB/LLT

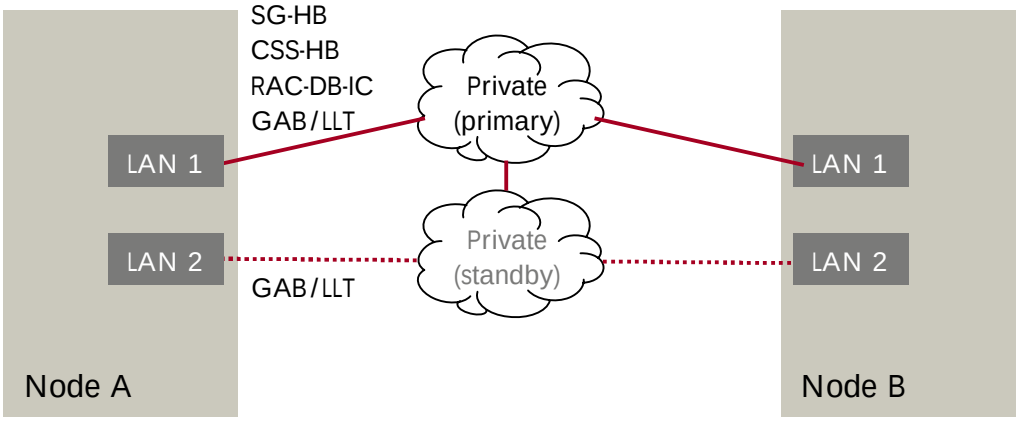


Figure 4 shows a common configuration where SG-HB, CSS-HB, RAC-DB-IC, and GAB/LLT share the same network for cluster communications. This configuration is common for most installations where the RAC traffic does not saturate the network and negatively affect other components (SG heartbeat, CSS heartbeat, and GAB/LLT traffic.) There is one network and the network has sufficient bandwidth. A failure of primary is protected by Serviceguard local LAN failover for SB-HB, CSS-HB and RAC-DB-IC. GAB/LLT uses both primary (lan1) and standby (lan2) for communications, so GAB/LLT can tolerate a failure of either the primary or standby.

Figure 5. Dual SG-HB with CSS-HB and RAC-DB-IC on single subnet including GAB/LLT

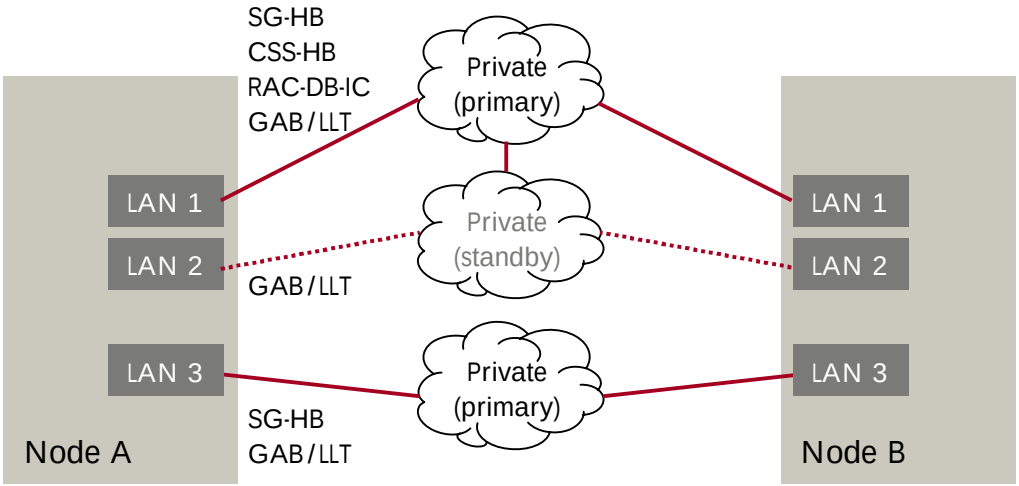


Figure 5 shows a configuration to address the possibility that heavy RAC-DB-IC traffic affects SG-HB. The pros and cons are same as figure 3.

Figure 6. Dual primary and standby Ethernet including GAB/LLT

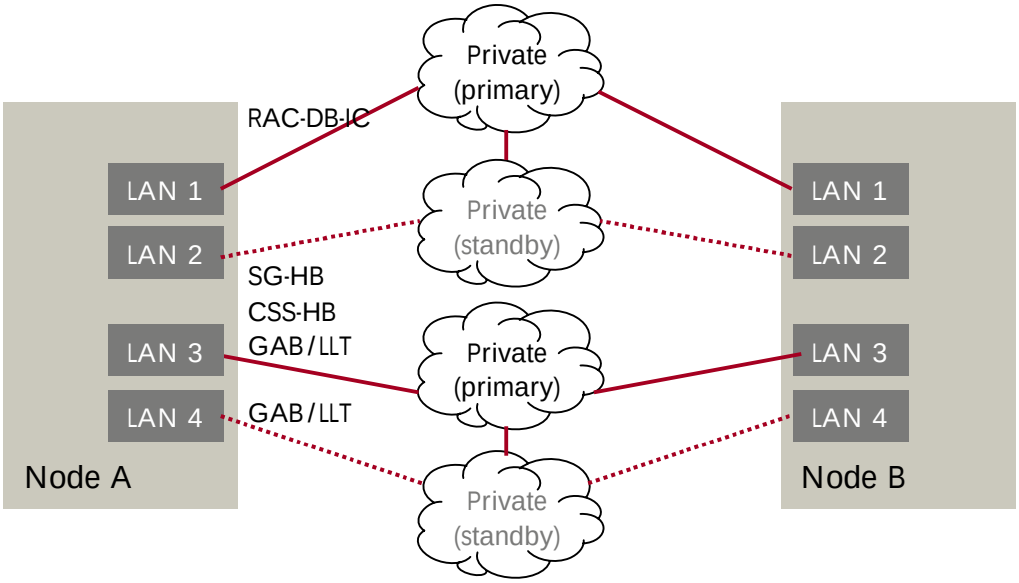


Figure 6 shows the same variation as figure 3 except this configuration is for CFS and CVM. This configuration is for heavily loaded configurations where RAC-DB-IC traffic interferes with heartbeats and other cluster communications. Placing RAC-DB-IC traffic on a separate network allows more aggressive SG-HB timeout values. The drawback is that the RAC-DB-IC should be monitored so network failures can be detected sooner than the Instance Membership Recovery (IMR) timeout and recovery can be started earlier.

Storage high availability

Storage HA is available at several levels, as follows:

- Redundant links to the same disk device.
- Storage arrays provide redundancy at the disk level.
- Volume Manager mirroring to multiple devices, for example by SLVM or CVM.
- Multiple copies on multiple disks.

OC relies on the platform to provide transparent redundant links to the same device. OC provides redundancy for the Oracle Cluster Registry (OCR) by providing mirroring capability, and thus disks from JBODs ("just a bunch of disk") can be used. When JBODs are used and the disk themselves are not highly available, two physical disks are required to protect against link failure and disk failure. In the case of redundancy for the Voting disk, three of more copies, that is disks, are required.

When redundant links and storage arrays are used, the configuration of OC is simplified by configuring a single OCR and Voting disk with HA provided by SLVM, CVM, and/or storage array.

Multiple nodes

Multiple nodes protect against failures at the system node level. For cluster HA, a minimum of two nodes is required.

Power

Redundant components should be separately powered so that a single power failure does not impact all nodes, all switches, and/or all storage.

Storage

Oracle Clusterware (OC) assumes the required storage is available when OC starts. Therefore, OC does not perform any storage activation and leaves it up to the platform or users to activate the storage prior to starting OC. For SGeRAC configurations, Serviceguard packages are used as the mechanism to activate storage prior to starting OC. For SLVM and CVM configurations, the shared storage activation is performed by the Serviceguard package that starts OC. For CFS configurations, the shared storage activation is performed by multi-node packages; therefore, the Serviceguard package that starts OC must have a dependency on the relevant multi-node packages.

Oracle Clusterware requirements: OCR and voting disk

Oracle Cluster Registry (OCR)

The OCR requires 100 MB of disk space. The OCR must be shared and accessible by all cluster nodes. OC uses the OCR to keep Oracle cluster information and configuration information regarding cluster databases. It is also used to keep track of processes that the Oracle Clusterware controls.

Voting disk

The voting disk requires 20 MB of disk space. As with the OCR, the voting disks is shared and accessible by all cluster nodes. Oracle uses the voting disk to manage Oracle cluster software membership. The voting disk is used as a health check device; case of network failure; it is used to arbitrate cluster ownership among the instances.

Shared storage

Each OCR and voting disk can exist as follows:

- SLVM – The file can reside on SLVM as a raw logical volume where the whole logical volume is used as the vote disk. In this case, the vote disk is the raw logical volume.
- CFS – The file can reside on a cluster file system as a regular file. In this case, it is not really a disk in the traditional sense of a physical device.
- CVM – The file can reside on CVM as a raw volume similar to SLVM raw logical volume.

Note:
Oracle uses the term voting disk. Sometimes, the vote disk is also referred as the voting disk. It is supported to place the OCR and voting disk on SLVM, CFS, or CVM. The advantages include multi-path support, mirroring, and controlled access.

RAC instance data files

With SGeRAC, RAC instance data files may reside on SLVM or CVM raw volumes, CFS, and ASM over SLVM, ASM over raw disks, and raw disks.

Oracle binaries (Clusterware and RAC)

Oracle binaries must reside on either local file system or cluster file system (CFS).

Archive log files

Oracle archive log files may be needed for database recovery and should be available to the node that performs database recovery. The archive logs may reside on cluster file system (CFS), or a file

system that can fail over within a Serviceguard package.

Prerequisites

In the sample configurations, the following prerequisites apply:

Software

- HP-UX 11i v2 0505 Enterprise Operating Environment
- Serviceguard A.11.16 or A.11.17/A.11.18/A.11.19 (A.11.17 or later required for CFS support)
- Serviceguard Extension for RAC A.11.16 or A.11.17/A.11.18/A.11.19 (A.11.17 or later required for CFS support)
- HP Serviceguard Management Suites Bundles A.01.00 or later.
- Oracle 10g R2 Clusterware and RAC

Server hardware

- One two-node cluster for SLVM (node names: “eenie” and “meanie”)
- One two-node cluster for CVM/CFS (node names: “mo” and “minie”)

Network

- Public – Ethernet with two redundant NICs for primary and standby
  - OC requires one VIP address per node
  - HA is provided by Serviceguard local LAN failover.
- Private –Ethernet with redundant NICS for primary and standby
  - All private cluster communications flow through the private network.
  - HA is provided by Serviceguard local LAN failover.

Storage

- Each node as internal storage for OC and RAC binaries.
- Shared Storage:
  - Primary Link
    - `/dev/dsk/c4t0d0  /dev/rdsk/c4t0d0`
    - `/dev/dsk/c4t1d0  /dev/rdsk/c4t1d0`
    - `/dev/dsk/c4t2d0  /dev/rdsk/c4t2d0`
    - `/dev/dsk/c4t3d0  /dev/rdsk/c4t3d0`
  - Redundant Link
    - `/dev/dsk/c5t0d0  /dev/rdsk/c5t0d0`
    - `/dev/dsk/c5t1d0  /dev/rdsk/c5t1d0`
    - `/dev/dsk/c5t2d0  /dev/rdsk/c5t2d0`
    - `/dev/dsk/c5t3d0  /dev/rdsk/c5t3d0`

Cluster for SLVM (eenie and meenie)

Figure 7. Cluster for SLVM

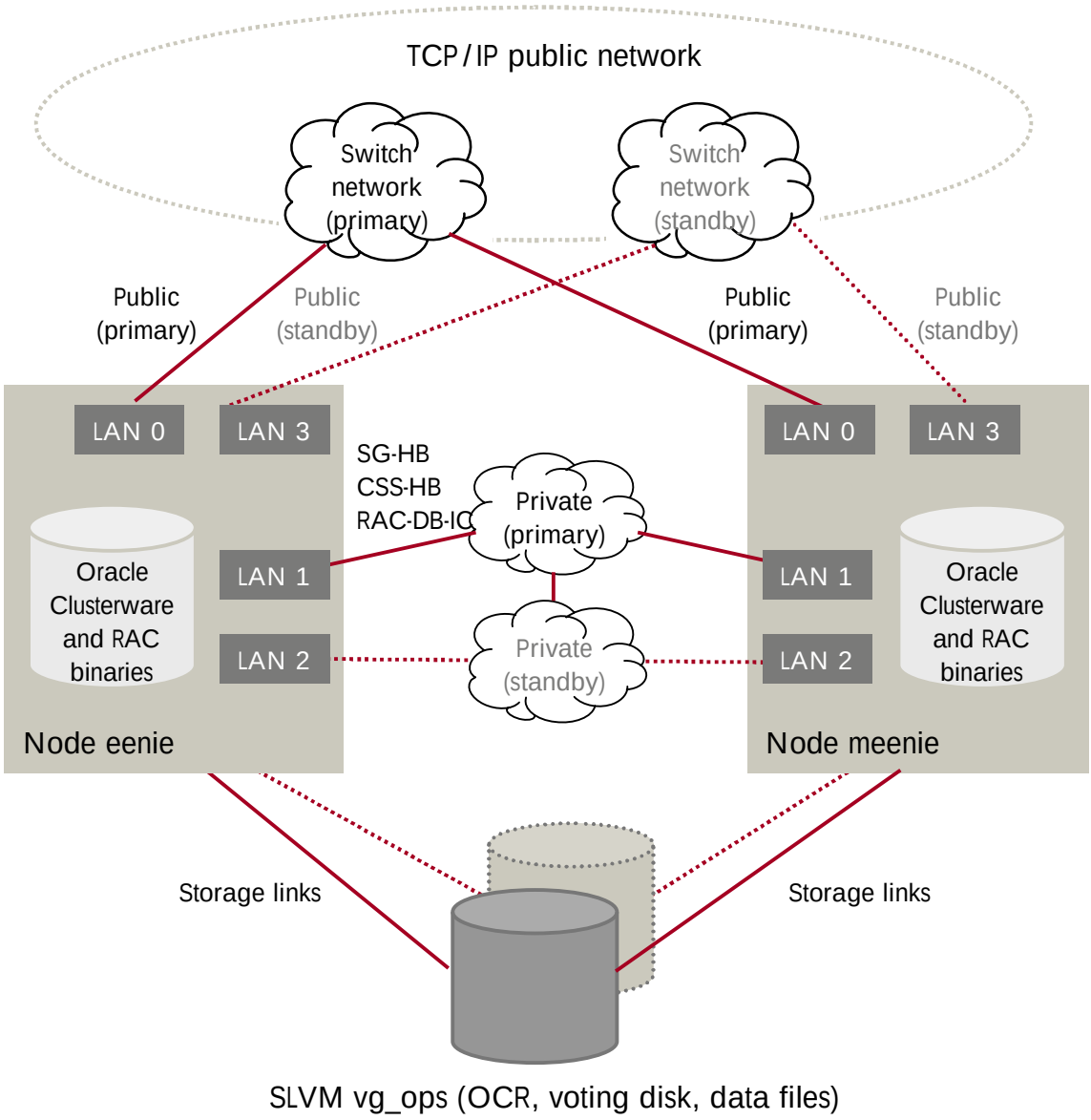
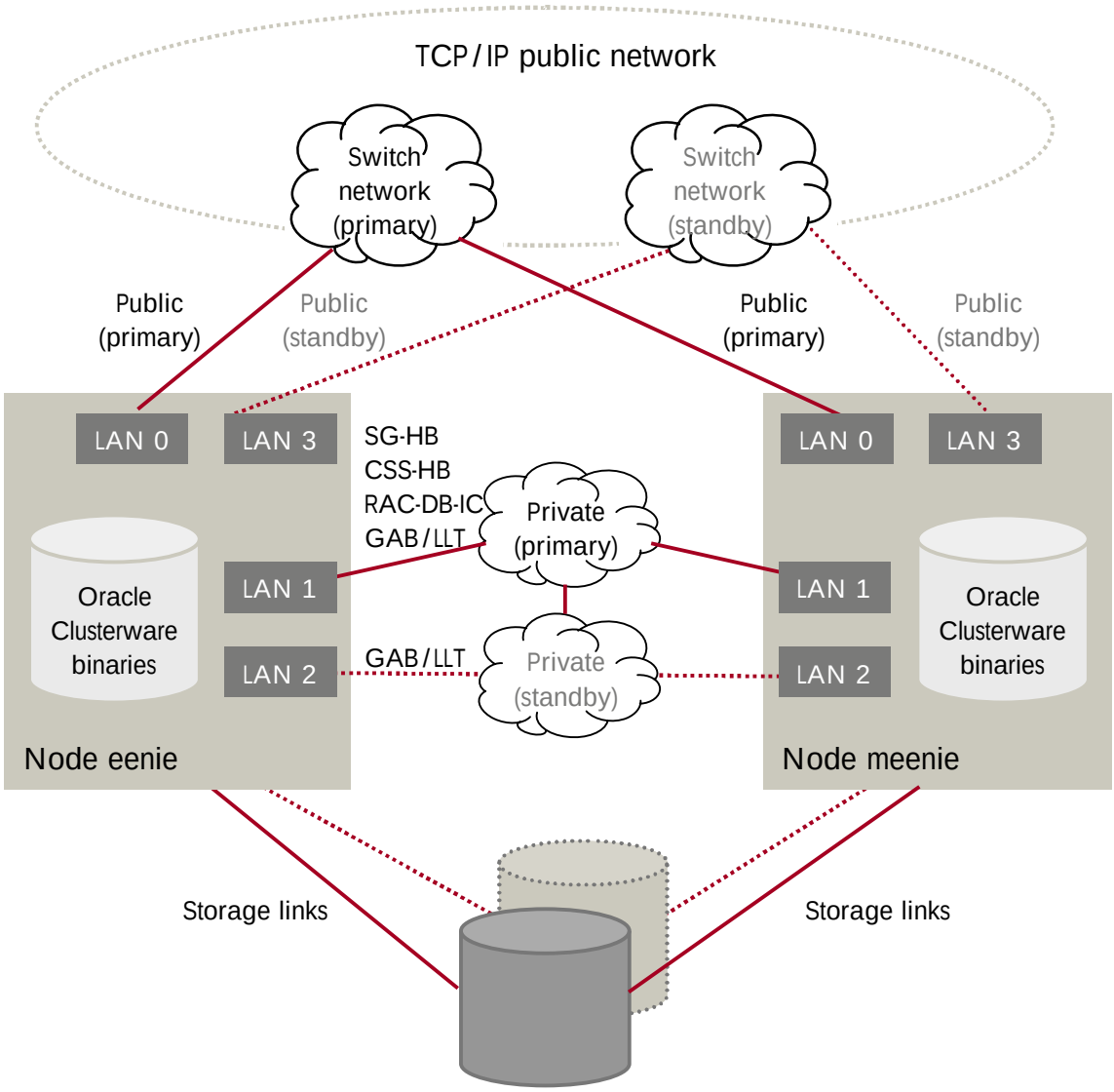


Figure 8. Cluster for CFS



Sample configurations

The following sections describe sample configuration steps for Oracle RAC 10g and SLVM.

Configuring Oracle RAC 10g on SLVM

The following sections describe the process for configuring Oracle RAC 10g on SLVM.

Assumptions for this sample configuration

1. Cluster hardware configured.
2. HP-UX 11i v2 0505 Enterprise Operating Environment.
3. Serviceguard and Serviceguard Extension for RAC installed.
4. Same private interconnect used for all inter-node traffic (Serviceguard, RAC, CSS)
5. One shared disk for shared volume group.
6. Two private disks (one disk per node for local file system for local Oracle storage)

Creating a SGeRAC cluster with SLVM for Oracle RAC 10g

The following sections describe the process for configuring SLVM.

SLVM Logical Volume Planning

For the sample configuration with SLVM, the following shared logical volumes are used. Note that one SLVM volume group is used for both Oracle Clusterware (OC) required files and RAC database files. When more than one RAC instance database or when the cluster will have more than one RAC instance database, it is recommended that separate volume groups are used for OC and each RAC instance database.

Using multiple volume groups permits better maintainability and manageability. For example, with SLVM single node online volume reconfiguration (SNOR), it is possible to reconfiguration SLVM volumes online while the RAC instance is running. Since SNOR requires that the target volume group be deactivated on all but one node, if only one volume group is used, RAC instances and OC on other nodes are impacted.

Figure 9. SLVM logical volume worksheet

```
                              RAW LOGICAL VOLUME NAME    SIZE (MB)
Oracle Cluster Registry: _____/dev/vg_ops/rora_ocr_____108___ (one per
cluster)
Oracle Cluster Vote Disk: ____/dev/vg_ops/rora_vote_____28___ (one per
cluster)

Oracle Control File: _____/dev/vg_ops/ropsctl1.ctl______118______
Oracle Control File 2: ___/dev/vg_ops/ropsctl2.ctl______118______
Oracle Control File 3: ___/dev/vg_ops/ropsctl3.ctl______118______
Instance 1 Redo Log 1: ___/dev/vg_ops/rops1log1.log_____128______
Instance 1 Redo Log 2: ___/dev/vg_ops/rops1log2.log_____128_______
Instance 1 Redo Log 3: ___/dev/vg_ops/rops1log3.log_____128_______
Instance 1 Redo Log: ____________________________________________________
Instance 1 Redo Log: ____________________________________________________
Instance 2 Redo Log 1: ___/dev/vg_ops/rops2log1.log____128________
Instance 2 Redo Log 2: ___/dev/vg_ops/rops2log2.log____128________
Instance 2 Redo Log 3: ___/dev/vg_ops/rops2log3.log____128_________
Instance 2 Redo Log: ____________________________________________________
Instance 2 Redo Log: ____________________________________________________
Data: System ___/dev/vg_ops/ropssystem.dbf___508__________
Data: Sysaux ___/dev/vg_ops/ropssysaux.dbf___808__________
Data: Temp ___/dev/vg_ops/ropstemp.dbf______258_______
Data: Users ___/dev/vg_ops/ropsusers.dbf_____128_________
Data: User data ___/dev/vg_ops/ropsdata1.dbf_208__________
Data: User data ___/dev/vg_ops/ropsdata2.dbf__208__________
Data: User data ___/dev/vg_ops/ropsdata3.dbf__208__________
Parameter: spfile1 ___/dev/vg_ops/ropsspfile1.ora __5_____
Password: ______/dev/vg_ops/rpwdfile.ora__5_______
Instance 1 undotbs1: /dev/vg_ops/ropsundotbs1.dbf___508___
Instance 2 undotbs2: /dev/vg_ops/ropsundotbs2.dbf___508___
Data: example1__/dev/vg_ops/ropsexample1.dbf__________168____
```

Creating volume group and logical volumes

1. Initialize LVM disk on node (“eenie”)

   ```
   # pvcreate /dev/rdsk/c4t3d0
   ```

2. Create the volume group on node (“eenie”).

   ```
   # mkdir /dev/vg_ops
   # mknod /dev/vg_ops/group c 64 0x070000
   ```

   Note: <0x070000> is the minor number on this sample configuration.

   ```
   # vgcreate /dev/vg_ops /dev/dsk/c4t3d0
   # vgextend /dev/vg_ops /dev/dsk/c5t3d0
   ```

   Note: <c5t3d0> is a redundant link to <c4t3d0>

3. Create logical volumes on node (“eenie”). For each of the logical volumes in the worksheet, create the logical volumes.

   ```
   # lvcreate –L <size> -n <lv name> <vg name>
   ```

   Example:

   ```
   # lvcreate  -L 128 –n ora_ocr /dev/vg_ops
   ```

   Repeat this step for each logical volume specified in the worksheet.

4. Export Volume Group on node (“eenie”)

   ```
   #  vgexport –s –p –m /tmp/vg_ops.map /dev/vg_ops
   ```

5. Import Volume Group on node (“meenie”). Assuming remote shell is configured, for example permission on /.rhosts or /etc/host.equiv, and logon on node (“meenie”).

   ```
   # rcp eenie:/tmp/vg_ops.map /tmp
   # mkdir /dev/vg_ops
   # mknod /dev/vg_ops/group c 64 0x070000
   # vgimport –s –m /tmp/vg_ops.map /dev/vg_ops
   ```

Create cluster ASCII file

```
# cd /etc/cmcluster
# cmquerycl –C clm.asc –n eenie –n meenie
```

Edit the cluster ASCII file. A two node configuration requires either a Serviceguard quorum server or cluster lock device. In this sample, the shared disk from the SLVM volume group is used as a cluster lock device.

```
CLUSTER_NAME            cluster_eenie
```

```
FIRST_CLUSTER_LOCK_VG   /dev/vg_ops

NODE_NAME         eenie
  NETWORK_INTERFACE     lan0
    STATIONARY_IP 15.13.170.64
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP  192.1.1.1
  NETWORK_INTERFACE     lan2
  FIRST_CLUSTER_LOCK_PV /dev/dsk/c4t3d0

NODE_NAME         meenie
  NETWORK_INTERFACE     lan0
    STATIONARY_IP 15.13.170.80
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP  192.1.1.2
  NETWORK_INTERFACE     lan2
  FIRST_CLUSTER_LOCK_PV /dev/dsk/c4t3d0

AUTO_START_TIMEOUT      600000000
NETWORK_POLLING_INTERVAL    2000000
NETWORK_FAILURE_DETECTION   INOUT
MAX_CONFIGURED_PACKAGES     150

OPS_VOLUME_GROUP  /dev/vg_ops
```

For A.11.18 and prior, the heartbeat timeout is

```
HEARTBEAT_INTERVAL      1000000
NODE_TIMEOUT            2000000
```

For A.11.19 and later, the heartbeat timeout is

```
MEMBER_TIMEOUT          14000000
```

Create cluster (sample)

Create and start the cluster on node.

```
# cmapplyconf –C clm.asc
# cmruncl
# cmviewcl

CLUSTER         STATUS
cluster_eenie   up

  NODE           STATUS       STATE
  eenie          up           running
  meenie         up           running
```

Prerequisites for Oracle 10g (sample installation)

The following are sample steps to prepare a SGeRAC cluster for Oracle 10g:

Note:
Consult Oracle documentation for Oracle installation details.

Create groups on each node

Create the Oracle Inventory group if one does not exist, create the OSDBA group, and create the Operator Group (optional).

```
# /usr/sbin/groupadd oinstall
# /usr/sbin/groupadd dba
# /usr/sbin/groupadd oper
```

Create Oracle user on each node

```
# /usr/bin/useradd –u 203 –g oinstall –G dba,oper oracle
```

Change password on each node

```
# passwd oracle
```

Enable remote access (ssh or remsh) for Oracle user on all nodes

For remsh, add oracle user to the .rhosts file or host.equiv file.

Create symbolic links

Required if Motif 2.1 Development Environment Package is not installed.

```
# ln -s /usr/lib/libX11.3 /usr/lib/libX11.sl
# ln -s /usr/lib/libXIE.2 /usr/lib/libXIE.sl
# ln -s /usr/lib/libXext.3 /usr/lib/libXext.sl
# ln -s /usr/lib/libXhp11.3 /usr/lib/Xhp11.sl
# ln -s /usr/lib/libXi.3 /usr/lib/libXi.sl
# ln -s /usr/lib/libXm.4 /usr/lib/libXm.sl
# ln -s /usr/lib/libXp.2 /usr/lib/libXp.sl
# ln -s /usr/lib/libXt.3 /usr/lib/libXt.sl
# ln -s /usr/lib/libXtst.2 /usr/lib/libXtst.sl
```

Create file system for Oracle directories

In the following samples, `/mnt/app` is a mounted file system for Oracle software. Assume there is a private disk `c2t0d0` at 18 GB size on all nodes. Create the local file system on each node.

```
# umask 022
# pvcreate /dev/rdsk/c2t0d0
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgcreate /dev/vg01 /dev/dsk/c2t0d0
# lvcreate –L 16000 /dev/vg01
# newfs –F vxfs /dev/vg01/rlvol1
# mkdir –p /mnt/app
# mount /dev/vg01/lvol1 /mnt/app
# chmod 775 /mnt/app
```

Create Oracle Cluster Software home directory

For installing Oracle Cluster Software on local file system, create the directories on each node.

```
# mkdir –p /mnt/app/crs/oracle/product/10.2.0/crs
# chown –R oracle:oinstall /mnt/app/crs/oracle
# chmod -R 775 /mnt/app/crs/oracle
```

The Oracle Cluster Software home directory is as follows:

```
/mnt/app/crs/oracle/product/10.2.0/crs
```

When installing Oracle Cluster Software, you should set the ORACLE_HOME environment to specify this directory. Please note at installation and before running the root.sh script, the parent directories of the Oracle Cluster Software home directory must be changed to permit only the root user to write to those directories.

Create Oracle base directory (for RAC binaries on local file system)

If installing RAC binaries on local file system, create the oracle base directory on each node.

```
# mkdir –p /mnt/app/oracle
# chown –R oracle:oinstall /mnt/app/oracle
# chmod –R 775 /mnt/app/oracle
```

Modify oracle user to new home directory on each node.

```
# usermod –d /mnt/app/oracle oracle
```

Prepare shared storage on SLVM

This section assumes the OCR, Vote device, and database files are created on SLVM volume group vg_ops.

Change permission of shared logical volume group

```
# chmod 755 /dev/vg_ops
```

Change permission and ownership of Oracle Cluster Software vote device and database files

```
# chown oracle:oinstall /dev/vg_ops/r*
# chmod 660 /dev/vg_ops/r*
```

Change permission of OCR device

```
# chown root:oinstall /dev/vg_ops/rora_ocr
# chmod 640 /dev/vg_ops/rora_ocr
```

Change permission of voting device

```
# chown root:dba /dev/vg_ops/rora_vote
# chmod 644 /dev/vg_ops/rora_vote
```

Create raw device mapping file for Oracle Database Configuration Assistant

```
# chown root:dba /dev/vg_ops/rora_vote
# chmod 644 /dev/vg_ops/rora_vote
```

In this example, the database name is “ver10”.

```
# ORACLE_BASE=/mnt/app/oracle ; export ORACLE_BASE
# mkdir –p $ORACLE_BASE/oradata/ver10
# chown –R oracle:oinstall $ORACLE_BASE/oradata
# chmod –R 755 $ORACLE_BASE/oradata
```

The following is a sample of the mapping file for DBCA.

```
system=/dev/vg_ops/ropssystem.dbf
sysaux=/dev/vg_ops/ropssysaux.dbf
```

```
undotbs1=/dev/vg_ops/ropsundotbs01.dbf
undotbs2=/dev/vg_ops/ropsundotbs02.dbf
example=/dev/vg_ops/ropsexample1.dbf
users=/dev/vg_ops/ropsusers.dbf
redo1_1=/dev/vg_ops/rops1log1.log
redo1_2=/dev/vg_ops/rops1log2.log
redo2_1=/dev/vg_ops/rops2log1.log
redo2_2=/dev/vg_ops/rops2log2.log
control1=/dev/vg_ops/ropsctl1.ctl
control2=/dev/vg_ops/ropsctl2.ctl
control3=/dev/vg_ops/ropsctl3.ctl
temp=/dev/vg_ops/ropstmp.dbf
spfile=/dev/vg_ops/ropsspfile1.ora
```

In this sample, create the DBCA mapping file and place at: `/mnt/app/oracle/oradata/ver10/ver10_raw.conf`.

Installing and configuring Oracle 10g Clusterware on local file system

Login as "oracle" user.

```
$ export DISPLAY=<display>:0.0
$ cd <10g Cluster Software disk directory>
$ ./runInstaller
```

Note:

1. Specify CRS HOME as `/mnt/app/crs/oracle/product/10.2.0/crs`. This is a local file system.
2. Specify OCR Location as `/dev/vg_ops/rora_ocr`.
3. Specify Vote Disk Location as `/dev/vg_ops/rora_vote`.
4. Supply the VIP addresses
    a. Oracle clusterware requires one VIP address for each node.
5. Specify the public network and private network
    a. In this sample, the private network is 192.1.1.0
6. When prompted, run `orainstRoot.sh` on each node
7. When prompted, run `root.sh` on each node

When Oracle Clusterware is installed, the Oracle cluster is also created. For configuring Oracle VIP in a configuration with Serviceguard Local LAN failover, see Oracle MetaLink Note:296874.1.[4]

Installing Oracle RAC 10g on local file system

Login as "oracle" user.

```
$ export ORACLE_BASE=/mnt/app/oracle
$ export DISPLAY=<display>:0.0
$ cd <Oracle RAC 10g installation disk>
$ ./runInstaller
```

Note:

1. In this example, the path to ORACLE_HOME is on a local file system `/mnt/app/oracle/product/10.2.0/db_1`
2. Select installation for database software only.
3. When prompted, run `root.sh` on each node

[4] Doc ID: Note:296874.1 Configuring the HP-UX Operating System for the Oracle 10g VIP at https://metalink.oracle.com/ (Oracle MetaLink account required).

Creating a RAC demo database on SLVM

Export environment variables for "oracle" user.

```
export ORACLE_BASE=/mnt/app/oracle
export DBCA_RAW_CONFIG=/mnt/app/oracle/oradata/ver10/ver10_raw.conf

export ORACLE_HOME=$ORACLE_BASE/product/10.2.0/db_1
export ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs

LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:$ORACLE_HOME/rdbms/lib
SHLIB_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME/rdbms/lib32
export LD_LIBRARY_PATH SHLIB_PATH
export PATH=$PATH:$ORACLE_HOME/bin:$ORA_CRS_HOME/bin:/usr/local/bin:
CLASSPATH=$ORACLE_HOME/jre:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib:$ORA
CLE_HOME/network/jlib
export CLASSPATH

export DISPLAY={display}:0.0
```

Setting up listeners with Oracle Network Configuration Assistant

$ netca

Notes:

1. Select Cluster Configurations
2. Select all nodes
3. Select Listener configuration
4. Select Add
5. Provide Listener name
6. Select Protocols
7. Select TCP/IP port number for listener

Creating demo database with Database Configuration Assistant

$ dbca

Unless specified, the default options are used. Notes:

1. Unless specified, the default options are used.
2. Select Oracle Real Application Clusters database
3. Select Create a Database
4. Select all nodes
5. Select General Purpose template
6. Provide Global Database Name
   a. In this sample, the global database name and SID prefix are "ver10".
7. Select Management Options
   a. In this sample, no management options chosen.
8. Provide passwords for user accounts
9. Select Listeners to register database
   a. In this sample, the listeners used are "LISTENER_EENIE" and "LISTENER_MEENIE".
10. Select Storage Options
    a. In this sample, Select the storage option for Raw Devices
11. Provide Raw Device Mapping File Location
    a. In this sample, the file is located at `/mnt/app/oracle/oradata/ver10/ver10_raw.conf`.

12. Choose Recovery Configuration
    a. In this sample, use default parameters (no flash recovery and archiving.).
    b. Flash Recovery Area and archiving can be configured. When configuring archiving, choose Enable Archive Mode Parameter and specify where to place archive logs. If Flash Recovery Area is configured, archive logs default to the Flash Recovery area.
    c. Without CFS, for simplicity, the archives logs should be on a file system that can be accessed by any node that would be performing a database recovery.
13. Select Database Content
14. Configure Database Services
15. Configure Initialization Parameters
16. Configure Database Storage
17. Create Database

Configuring Serviceguard packages using Serviceguard Extension for RAC

It is recommended to start and stop Oracle Cluster Software in a Serviceguard package, as that will ensure that Oracle Cluster Software will start after SGeRAC is started and will stop before SGeRAC is halted. Serviceguard packages should also be used to synchronize storage activation and deactivation with Oracle Cluster Software and RAC instances. Additionally, the Serviceguard package also checks CSS in case CSS is halted outside the package. If CSS is halted outside the package, the package halts.

For SGeRAC A.11.17, there is a whitepaper Use of Serviceguard Extension For RAC Toolkit with Oracle RAC 10g Release 2 or later (http://docs.hp.com/en/8987/sgeractoolkit-wp.pdf) to configure packages with multi-node packages and simple dependency features. The SGeRAC Toolkit can be downloaded from the software depot (http://software.hp.com/ à High availability à Serviceguard Extension for RAC Toolkit).

SGeRAC A.11.18 and later includes the SGeRAC Toolkit along with a README document describing the use of SGeRAC Toolkit.

Prepare Oracle Cluster Software for Serviceguard packages

Login as "root" user.

```
# export ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs
# export PATH=$PATH:$ORA_CRS_HOME/bin
```

Stop Oracle Clusterware on each node

For 10g 10.2.0.1 or later:

```
# crsctl stop crs
```

Wait until Oracle Cluster Software completely stops. (Check CRS logs or check for Oracle processes, for example `ps –ef | grep ocssd.bin`)

Change Oracle Cluster Software from starting at boot time on each node

For 10g 10.2.0.1 or later:

```
# crsctl disable crs
```

Creating Serviceguard packages

In this configuration, the cluster is configured with one Serviceguard package multi-node package that will start and stop Oracle Clusterware.

Creating Serviceguard package for Oracle Clusterware

1. Create package directory and copy toolkit files.

```
# mkdir /etc/cmcluster/crsp-slvm
# cd /etc/cmcluster/crsp-slvm
# cp /opt/cmcluster/SGeRAC/toolkit/crsp/* ./
```

2. Create package files

```
# cmmakepkg –p crsp-slvm.conf
# cmmakepkg –s crsp-slvm.ctl
```

3. Edit the package configuration file `crsp-slvm.conf`.

```
PACKAGE_NAME                    crsp-slvm
PACKAGE_TYPE                    MULTI_NODE
#FAILOVER_POLICY                CONFIGURED_NODE
#FAILBACK_POLICY                MANUAL
NODE_NAME                       eenie
NODE_NAME                       meenie
RUN_SCRIPT                      /etc/cmcluster/crsp-slvm/crsp-slvm.ctl
HALT_SCRIPT                     /etc/cmcluster/crsp-slvm/crsp-slvm.ctl
SERVICE_NAME                    crsp-slvm-srv
SERVICE_FAIL_FAST_ENABLED       NO
SERVICE_HALT_TIMEOUT            300
```

4. Edit the package control script `crsp-slvm.ctl`.

```
VGCHANGE="vgchange -a s"
VG[0]="vg_ops"
SERVICE_NAME[0]="crsp-slvm-srv"
SERVICE_CMD[0]="/etc/cmcluster/crsp-slvm/toolkit_oc.sh check"
SERVICE_RESTART[0]=""

function customer_defined_run_cmds
{
# ADD customer defined run commands.
       /etc/cmcluster/crsp-slvm/toolkit_oc.sh start
        test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
       /etc/cmcluster/crsp-slvm/toolkit_oc.sh stop
        test_return 52
}
```

5. Edit the toolkit configuration file `oc.conf`.

```
ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs
```

6. Add the package to the cluster. Distribute Oracle Clusterware multi-node package (MNP) directory to all nodes.

```
# cd /etc/cmcluster
# rcp –r crsp-slvm root@meenie:/etc/cmcluster
```

Add package to cluster.

```
# cd /etc/cmcluster/crsp-slvm
# cmapplyconf –P crsp-slvm.conf

Modify the cluster configuration ([y]/n)? y
Completed the cluster creation
```

Starting and stopping Serviceguard packages and Oracle RAC

On each node, halt Oracle Clusterware if running.

```
# $ORA_CRS_HOME/bin/crsctl stop crs
```

Start the complete stack by running the Serviceguard Package.

```
# cmrunpkg crsp-slvm
Running package crsp-slvm on node eenie
Successfully started package crsp-slvm on node eenie
Running package crsp-slvm on node meenie
Successfully started package crsp-slvm on node meenie
cmrunpkg: All specified packages are running
```

```
# cmviewcl
CLUSTER         STATUS
cluster_eenie   up

  NODE           STATUS       STATE
  eenie          up           running
  meenie         up           running

MULTI_NODE_PACKAGES

  PACKAGE        STATUS       STATE        AUTO_RUN     SYSTEM
  crsp-slvm      up           running      enabled      no
```

Configuring Serviceguard packages without Serviceguard Extension for RAC

It is recommended to start and stop Oracle Cluster Software in a Serviceguard package, as that will ensure that Oracle Cluster Software will start after SGeRAC is started and will stop before SGeRAC is halted. Serviceguard packages should also be used to synchronize storage activation and deactivation with Oracle Cluster Software and RAC instances. Additionally, the Serviceguard package also checks CSS in case CSS is halted outside the package. If CSS is halted outside the package, the package halts.

It is recommended to use SGeRAC Toolkit. This section is a sample if the SGeRAC Toolkit is not used.

Prepare Oracle Cluster Software for Serviceguard packages

Login as “root” user.

```
# export ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs
# export PATH=$PATH:$ORA_CRS_HOME/bin
```

Stop Oracle Clusterware on each node
For 10g 10.2.0.1:

```
# crsctl stop crs
```

Wait until Oracle Cluster Software completely stops. Check CRS logs or check for Oracle processes, for example `ps –ef | grep ocssd.bin`

Change Oracle Cluster Software from starting at boot time on each node
For 10g 10.2.0.1:

```
# crsctl disable crs
```

Creating Serviceguard Packages
In this configuration, each node is configured with one Serviceguard package that will start and stop Oracle Clusterware.

Creating Serviceguard package for node “eenie”

1. Create Package Directory.

```
# cd /etc/cmcluster
# mkdir pkg
# mkdir pkg/crs_eenie_pkg
# cd pkg/crs_eenie_pkg
```

2. Create package files

```
# cmmakepkg –p crs_eenie_pkg.conf
# cmmakepkg –s crs_eenie_pkg.sh
```

3. Edit the package configuration file crs_eenie_pkg.conf.

```
SERVICE_NAME                    css_check_eenie
SERVICE_FAIL_FAST_ENABLED       NO
SERVICE_HALT_TIMEOUT            300

PACKAGE_NAME                    crs_eenie_pkg
NODE_NAME                       eenie
RUN_SCRIPT
/etc/cmcluster/pkg/crs_eenie_pkg/crs_eenie_pkg.sh
HALT_SCRIPT
/etc/cmcluster/pkg/crs_eenie_pkg/crs_eenie_pkg.sh
```

4. Edit the package control script crs_eenie_pkg.sh.

```
SERVICE_NAME[0]="css_check_eenie"
SERVICE_CMD[0]="/etc/cmcluster/pkg/crs_eenie_pkg/cssd.sh monitor"
SERVICE_RESTART[0]=""

function customer_defined_run_cmds
{
# ADD customer defined run commands.
```

```
        /etc/cmcluster/pkg/crs_eenie_pkg/cssd.sh start
        test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.

        /etc/cmcluster/pkg/crs_eenie_pkg/cssd.sh stop
        test_return 52
}
```

Note:
The `cssd.sh` script is a sample script that is in the Appendix for starting, monitoring, and stopping OC.

5. Add the package to the cluster.

```
# cmapplyconf –P crs_eenie_pkg.conf
Modify the cluster configuration ([y]/n)? y
Completed the cluster creation
```

Creating Serviceguard package for node “meenie”
Create the Serviceguard package for node “meenie”. The steps are the same as for node “eenie”. Samples of the package configuration file and control scripts are in the Appendix.

Starting and stopping Serviceguard packages and Oracle RAC
Start the complete stack by running the Serviceguard Package.

```
# cmrunpkg –n eenie crs_eenie_pkg
Running package crs_eenie_pkg on node eenie
Successfully started package crs_eenie_pkg on node eenie

# cmrunpkg –n meenie crs_meenie_pkg
Successfully started package crs_meenie_pkg on node meenie
cmrunpkg: All specified packages are running

# cmviewcl
CLUSTER        STATUS
cluster_eenie   up

  NODE           STATUS       STATE
  eenie          up           running

    PACKAGE          STATUS       STATE        AUTO_RUN     NODE
    crs_eenie_pkg    up           running      enabled      eenie

  NODE           STATUS       STATE
  meenie         up           running

    PACKAGE          STATUS       STATE        AUTO_RUN     NODE
    crs_meenie_pkg   up           running      enabled      meenie
```

Configuring Oracle RAC 10g on CFS

The following sections describe the process for configuring Oracle RAC 10g on CFS.

Assumptions for this sample configuration

1. Cluster hardware configured
2. HP-UX 11i v2 0505 Enterprise Operating Environment
3. HP Serviceguard Storage Management Suite (A.01.00 or later) Installed
4. Same private interconnect used for all inter-node traffic (Serviceguard, RAC, CSS, GAB/LLT)
5. One shared disk for CFS
6. Two private disks (one disk per node for local file system for local Oracle storage)

CFS and ODM requirement

ODM is required when using Oracle RAC with CFS and SGeRAC.

Creating a SGeRAC cluster with CFS for Oracle 10g

In this sample, both the Oracle RAC software and datafiles reside on CFS. There is a single Oracle home. Three CFS files systems are created for Oracle home, Oracle datafiles, and for the Oracle Cluster Registry (OCR) and vote device. The Oracle Cluster Software home is on local file system.

```
/cfs/mnt1 – for Oracle Base and Home
/cfs/mnt2 – for Oracle datafiles
/cfs/mnt3 – for OCR and Vote device.
```

Initializing the VERITAS Volume Manager

If have not already done so, install VxVM license key on all nodes.

```
# vxinstall
```

Create Cluster ASCII file

```
# cd /etc/cmcluster
# cmquerycl –C clm.asc –n mo –n minie
```

Edit the cluster ASCII file. A two node configuration requires either a Serviceguard quorum server or cluster lock device. In this sample, since there is no SLVM volume group to be used for a cluster lock device, a quorum server is used.

```
CLUSTER_NAME            cluster_mo
QS_HOST                 white
QS_POLLING_INTERVAL     120000000
QS_TIMEOUT_EXTENSION    2000000

NODE_NAME               minie
  NETWORK_INTERFACE     lan0
    STATIONARY_IP       15.13.170.82
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP        192.1.1.3
  NETWORK_INTERFACE     lan2

NODE_NAME               mo
  NETWORK_INTERFACE     lan0
    STATIONARY_IP       15.13.171.137
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP        192.1.1.4
```

```
  NETWORK_INTERFACE     lan2

AUTO_START_TIMEOUT      600000000
NETWORK_POLLING_INTERVAL 2000000
NETWORK_FAILURE_DETECTION       INOUT
MAX_CONFIGURED_PACKAGES         150
```

For A.11.18 and prior, the heartbeat timeout is

```
HEARTBEAT_INTERVAL      1000000
NODE_TIMEOUT            5000000
```

For A.11.19 and later, the heartbeat timeout is

```
MEMBER_TIMEOUT          14000000
```

Create cluster (sample)

```
# cmapplyconf –C clm.asc
```

Start the Cluster

```
# cmruncl
# cmviewcl

CLUSTER        STATUS
cluster_mo     up

  NODE           STATUS       STATE
  minie          up           running
  mo             up           running
```

Configuring Cluster Volume Manager (CVM)

Configure the system multi-node package, **SG-CFS-pkg**, to configure and start the CVM / CFS stack.

```
# cfscluster config -s
CVM is now configured
Starting CVM...
It might take a few minutes to complete
```

When CVM starts up, it selects a master node, and this is the node from which you must issue the disk group configuration commands. To determine the master node, issue the following command from each node in the cluster.

```
# vxdctl –c mode
mode: enabled: cluster active - SLAVE
master: minie
```

Initializing disks for CVM / CFS

You need to initialize the physical disks that will be employed in CVM disk groups. If a physical disk has been previously used with LVM, you should use the `pvremove` command to delete the LVM header data from all the disks in the volume group (this is not necessary if you have not previously used the disk with LVM)

To initialize a disk for CVM, log on to the master node, then use the `vxdiskadm` program to initialize multiple disks, or use the `vxdisksetup` command to initialize one disk at a time, as in the following example:

```
# /etc/vx/bin/vxdisksetup -i c4t1d0
```

Create disk groups for RAC

Use the vxdg command to create disk groups. Use the –s option to specify shared mode, as in the following example:

```
# vxdg –s init cfsdg1 c4t1d0
```

Create disk group multi-node package

Add the disk group to the cluster.

```
# cfsdgadm add cfsdg1 all=sw
Package name "SG-CFS-DG-1" was generated to control the resource
Shared disk group "cfsdg1" was associated to the cluster
```

Activate disk group

```
# cfsdgadm activate cfsdg1
```

Creating volumes and adding a Cluster File System

```
# vxassist -g cfsdg1 make vol1 7000m
# vxassist -g cfsdg1 make vol2 7000m
# vxassist -g cfsdg1 make vol3 300m
# newfs -F vxfs /dev/vx/rdsk/cfsdg1/vol1
    version 7 layout
    7168000 sectors, 7168000 blocks of size 1024, log size 16384 blocks
    largefiles supported
# newfs -F vxfs /dev/vx/rdsk/cfsdg1/vol2
    version 7 layout
    7168000 sectors, 7168000 blocks of size 1024, log size 16384 blocks
    largefiles supported
# newfs -F vxfs /dev/vx/rdsk/cfsdg1/vol3
    version 7 layout
    307200 sectors, 307200 blocks of size 1024, log size 1024 blocks
    largefiles supported
```

Configure mount point

```
# cfsmntadm add cfsdg1 vol1 /cfs/mnt1 all=rw
Package name "SG-CFS-MP-1" was generated to control the resource
Mount point "/cfs/mnt1" was associated to the cluster

# cfsmntadm add cfsdg1 vol2 /cfs/mnt2 all=rw
Package name "SG-CFS-MP-2" was generated to control the resource
Mount point "/cfs/mnt2" was associated to the cluster

# cfsmntadm add cfsdg1 vol3 /cfs/mnt3 all=rw
Package name "SG-CFS-MP-3" was generated to control the resource
Mount point "/cfs/mnt3" was associated to the cluster
```

Mounting Cluster File System

```
# cfsmount /cfs/mnt1
# cfsmount /cfs/mnt2
# cfsmount /cfs/mnt3
```

Check CFS mount points

```
# bdf | grep cfs
/dev/vx/dsk/cfsdg1/vol1
```

```
                    7168000   35644 6686584    0% /cfs/mnt1
/dev/vx/dsk/cfsdg1/vol2
                    7168000   25644 6686584    0% /cfs/mnt2
/dev/vx/dsk/cfsdg1/vol3
                    307200    3264  284657     1% /cfs/mnt3
```

Viewing configuration

```
# cmviewcl

CLUSTER        STATUS
cluster_mo     up

  NODE           STATUS       STATE
  minie          up           running
  mo             up           running

MULTI_NODE_PACKAGES

  PACKAGE        STATUS       STATE        AUTO_RUN     SYSTEM
  SG-CFS-pkg     up           running      enabled      yes
  SG-CFS-DG-1    up           running      enabled      no
  SG-CFS-MP-1    up           running      enabled      no
  SG-CFS-MP-2    up           running      enabled      no
  SG-CFS-MP-3    up           running      enabled      no
```

Prerequisites for Oracle 10g (sample installation)

These are sample steps to prepare a SGeRAC cluster for Oracle 10g. Consult Oracle documentation for Oracle installation details.

Create groups on each node

Create the Oracle Inventory group if one does not exist, create the OSDBA group, and create the Operator Group (optional).

```
# /usr/sbin/groupadd oinstall
# /usr/sbin/groupadd dba
# /usr/sbin/groupadd oper
```

Create Oracle User on each node

```
# /usr/bin/useradd –u 203 –g oinstall –G dba,oper oracle
```

Change password on each node

```
# passwd oracle
```

Enable remote access (ssh or remsh) for Oracle user on all nodes

For remsh, add oracle user to the `.rhosts` file or `host.equiv` file.

Create symbolic links

Required if Motif 2.1 Development Environment Package is not installed.

```
# ln -s /usr/lib/libX11.3 /usr/lib/libX11.sl
# ln -s /usr/lib/libXIE.2 /usr/lib/libXIE.sl
# ln -s /usr/lib/libXext.3 /usr/lib/libXext.sl
# ln -s /usr/lib/libXhp11.3 /usr/lib/Xhp11.sl
# ln -s /usr/lib/libXi.3 /usr/lib/libXi.sl
# ln -s /usr/lib/libXm.4 /usr/lib/libXm.sl
# ln -s /usr/lib/libXp.2 /usr/lib/libXp.sl
```

```
# ln -s /usr/lib/libXt.3 /usr/lib/libXt.sl
# ln -s /usr/lib/libXtst.2 /usr/lib/libXtst.sl
```

Create file system for Oracle directories

In the following samples, `/mnt/app` is a mounted file system for Oracle software. Assume there is a private disk `c2t0d0` at 18 GB size on all nodes. Create the local file system on each node.

```
# umask 022
# pvcreate /dev/rdsk/c2t0d0
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgcreate /dev/vg01 /dev/dsk/c2t0d0
# lvcreate –L 16000 /dev/vg01
# newfs –F vxfs /dev/vg01/rlvol1
# mkdir –p /mnt/app
# mount /dev/vg01/lvol1 /mnt/app
# chmod 775 /mnt/app
```

Create Oracle Cluster Software home directory

For installing Oracle Cluster Software on local file system, create the directories on each node.

```
# mkdir –p /mnt/app/crs/oracle/product/10.2.0/crs
# chown –R oracle:oinstall /mnt/app/crs/oracle
# chmod -R 775 /mnt/app/crs/oracle
```

Create Oracle base directory (for RAC binaries on Cluster File System)

If installing RAC binaries on Cluster File System, create the oracle base directory once since this is CFS directory visible by all nodes. The CFS file system used is `/cfs/mnt1`.

```
# mkdir –p /cfs/mnt1/oracle
# chown –R oracle:oinstall /cfs/mnt1/oracle
# chmod –R 775 /cfs/mnt1/oracle
# chmod 775 /cfs/mnt1
```

Change directory permission on each node.

```
# chmod 775 /cfs
```

Modify oracle user to new home directory on each node.

```
# usermod –d /cfs/mnt1/oracle oracle
```

Prepare shared storage on Cluster File System

This section assumes the OCR, Vote device, and database files are created on CFS directories. The OCR and vote device reside on `/cfs/mnt3` and the demo database files reside on `/cfs/mnt2.`

Create OCR and vote device on Cluster File System

Create directories for OCR and vote device on Cluster File System. Run commands only on one node.

```
# chmod 755 /cfs/mnt3
# cd /cfs/mnt3
# mkdir OCR
# chmod 755 OCR
# mkdir VOTE
# chmod 755 VOTE
# chown –R oracle:oinstall /cfs/mnt3
```

Change directory permission on each node (if needed).

```
# chmod 775 /cfs
```

Create directory for Oracle demo database on Cluster File System
Create the CFS directory to store Oracle database files. Run commands only on one node.

```
# chmod 775 /cfs/mnt2
# cd /cfs/mnt2
# mkdir oradata
# chown oracle:oinstall oradata
# chmod 775 oradata
```

Change directory permission on each node (if needed).

```
# chmod 775 /cfs
```

Installing and configuring Oracle 10g Clusterware on local file system
Login as "oracle" user.

```
$ export DISPLAY=<display>:0.0
$ cd <10g Cluster Software disk directory>
$ ./runInstaller
```

Note:

1. Specify CRS HOME as `/mnt/app/crs/oracle/product/10.2.0/crs`. This is a local file system.
2. Specify OCR Location as `/cfs/mnt3/OCR/ocr_file` if using CFS for OCR.
3. Specify Vote Disk Location as `/cfs/mnt3/VOTE/vote_file` if using CFS for vote device.
4. Supply the VIP addresses
    a. Oracle clusterware requires one VIP address for each node.
5. Specify the public network and private network.
    a. In this sample, the private network is 192.1.1.0.
6. When prompted, run `orainstRoot.sh` on each node.
7. When prompted, run `root.sh` on each node.

When Oracle Clusterware is installed, the Oracle cluster is also created. For configuring Oracle VIP in a configuration with Serviceguard Local LAN failover, see Oracle MetaLink Note:296874.1.[5]

Installing Oracle RAC 10g on CFS
Login as "oracle" user. This step installs the Oracle RAC binaries on CFS. It is recommended that the RAC binary installation and demo database create to be separate steps.

```
$ export ORACLE_BASE=/cfs/mnt1/oracle
$ export DISPLAY=<display>:0.0
$ cd <Oracle RAC 10g installation disk>
$ ./runInstaller
```

Note:

1. In this example, the path to ORACLE_HOME is located on a CFS directory. `/cfs/mnt1/oracle/product/10.2.0/db_1.`
2. Select installation for database software only.
3. When prompted, run `root.sh` on each node.

[5] Doc ID: Note:296871.1 Configuring the HP-UX Operating System for the Oracle 10g VIP at https://metalink.oracle.com/ (Oracle MetaLink account required).

Configuring ODM

ODM is required when using Oracle RAC with SGeRAC and CFS. For this sample configuration, the ODM libraries are included with the HP Serviceguard Storage Management Suite bundle for RAC.

Previously, there was a confirmed problem with creating an Oracle database with dbca after enabling ODM (linking the ODM library). The Oracle bug # is 5103839. The workaround was to create the database (see §4.2.8) first and then link ODM (§4.2.7). Starting with Oracle 10.2.0.3, this problem has been resolved and the workaround is not needed.

Check ODM availability

1. Check the VRTSdbed license.

```
# /opt/VRTS/bin/vxlictest -n "VERITAS Storage Foundation for Oracle" -
f "ODM"
ODM feature is licensed
```

2. Check that the VRTSodm package is installed.

```
# swlist VRTSodm

VRTSodm                     5.0.01.01 Veritas Oracle Disk Manager
VRTSodm.ODM-KRN                   5.0.01.01 Veritas ODM kernel files
VRTSodm.ODM-MAN                   5.0.01.01 Veritas ODM manual pages
VRTSodm.ODM-RUN                   5.0.01.01 Veritas ODM commands
```

3. Check that libodm.sl is present.

```
# ls -lL /opt/VRTSodm/lib/libodm.sl
-r-xr-xr-x 1 root sys 81464 May 15 2006 /opt/VRTSodm/lib/libodm.sl
```

Configure Oracle to use ODM library

1. Login as Oracle user
2. Shutdown database (if running)
3. Link the Oracle Disk Manager library into Oracle home for Oracle 10g

For HP 9000 systems:

```
$ rm ${ORACLE_HOME}/lib/libodm10.sl
$ ln -s /opt/VRTSodm/lib/libodm.sl ${ORACLE_HOME}/lib/libodm10.sl
```

For HP Integrity systems:

```
$ rm ${ORACLE_HOME}/lib/libodm10.so
$ ln -s /opt/VRTSodm/lib/libodm.sl ${ORACLE_HOME}/lib/libodm10.so
```

Configure Oracle to stop using ODM library

1. Login as Oracle user
2. Shutdown database (if running)
3. Link the original Oracle Disk Manager library into Oracle home for Oracle 10g

Note:
Some versions of the SGeRAC manual incorrectly removed the installed library ("libodmd10.sl"for HP 9000 systems and "libdodmd10.so" for HP Integrity systems). The correct behavior is to remove the link.

For HP 9000 systems:

```
$ rm ${ORACLE_HOME}/lib/libodm10.sl
$ ln -s ${ORACLE_HOME}/lib/libodmd10.sl ${ORACLE_HOME}/lib/libodm10.sl
```

For HP Integrity systems:

```
$ rm ${ORACLE_HOME}/lib/libodm10.so
$ ln -s ${ORACLE_HOME}/lib/libodmd10.so ${ORACLE_HOME}/lib/libodm10.so
```

Creating RAC demo database on CFS

Export environment variables for "oracle" user.

```
export ORACLE_BASE=/cfs/mnt1/oracle

export ORACLE_HOME=$ORACLE_BASE/product/10.2.0/db_1
export ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs

LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:$ORACLE_HOME/rdbms/lib
SHLIB_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME/rdbms/lib32
export LD_LIBRARY_PATH SHLIB_PATH
export PATH=$PATH:$ORACLE_HOME/bin:$ORA_CRS_HOME/bin:/usr/local/bin:
CLASSPATH=$ORACLE_HOME/jre:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib:$ORA
CLE_HOME/network/jlib
export CLASSPATH

export DISPLAY=<display>:0.0
```

Setting up listeners with Oracle Network Configuration Assistant

```
$ netca
```

Notes:

1. Select Cluster Configurations
2. Select all nodes
3. Select Listener configuration
4. Select Add
5. Provide Listener name
6. Select Protocols
7. Select TCP/IP port number for listener

Creating demo database with Database Configuration Assistant

```
$ dbca
```

Unless specified, the default options are used.

Notes:

1. Unless specified, the default options are used.
2. Select Oracle Real Application Clusters database.
3. Select Create a Database.
4. Select all nodes.
5. Select General Purpose template.
6. Provide Global Database Name.
   a. In this sample, the global database name and SID prefix are "ver10".
7. Select Management Options.
   a. In this sample, no management options chosen.

8. Provide passwords for user accounts.
9. Select Listeners to register database.
    a. In this sample, the listeners used are "LISTENER_MO" and "LISTENER_MINIE".
10. Select Storage Options.
    a. In this sample, Select the storage option for Cluster File System.
11. Provide Database File Locations.
    a. In this sample, choose "Use Common Location for all Database Files" and enter `/cfs/mnt2/oradata` as the common directory.
12. Choose Recovery Configuration.
    a. In this sample, use default parameters (Flash Recovery Area configured) and select Enable Archiving.
    b. If Flash Recovery Area is configured, archive logs default to the Flash Recovery area.
    c. With CFS, the archives can be on a file system that is accessed by any node that would be performing a database recovery.
13. Select Database Content
14. Configure Database Services.
15. Configure Initialization Parameters.
16. Configure Database Storage.
17. Create Database.

Configuring Serviceguard packages with Serviceguard Extension for RAC

It is recommended to start and stop Oracle Cluster Software in a Serviceguard package, as that will ensure that Oracle Cluster Software will start after SGeRAC is started and will stop before SGeRAC is halted. Serviceguard packages should also be used to synchronize storage activation and deactivation with Oracle Cluster Software and RAC instances. Additionally, the Serviceguard package also checks CSS in case CSS is halted outside the package. If CSS is halted outside the package, the package halts.

For SGeRAC A.11.17, there is the Use of Serviceguard Extension For RAC Toolkit with Oracle RAC 10g Release 2 or Later White Paper (http://docs.hp.com/en/ha.html) to configure packages with multi-node packages and simple dependency features. The SGeRAC Toolkit can be downloaded from the HP Software Depot (http://software.hp.com/ à High availability à Serviceguard Extension for RAC Toolkit).

SGeRAC A.11.18 and later includes the SGeRAC Toolkit along with a README document describing the use of SGeRAC Toolkit.

Prepare Oracle Cluster Software for Serviceguard packages

Login as "root" user.

```
# export ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs
# export PATH=$PATH:$ORA_CRS_HOME/bin
```

Stop Oracle Clusterware on each node
For 10g 10.2.0.1 or later:

```
# crsctl stop crs
```

Wait until Oracle Cluster Software completely stops. (Check CRS logs or check for Oracle processes, for example `ps –ef | grep ocssd.bin`)

Change Oracle Cluster Software from starting at boot time on each node
For 10g 10.2.0.1 or later:

```
# crsctl disable crs
```

Creating Serviceguard packages

In this configuration, the cluster is configured with one Serviceguard multi-node package that will start and stop Oracle Clusterware.

Creating Serviceguard package for Oracle Clusterware

1. Create package directory and copy toolkit files

```
# mkdir /etc/cmcluster/crsp
# cd /etc/cmcluster/crsp
# cp /opt/cmcluster/SGeRAC/toolkit/crsp/* ./
```

2. Create Package Files

```
# cmmakepkg –p crsp.conf
# cmmakepkg –s crsp.ctl
```

3. Edit the package configuration file `crsp.conf`. Since the OCR, Voting Disk, RAC binaries, and demo database files reside on CFS, the OC package is configured to depend on the CFS packages.

```
PACKAGE_NAME                    crsp
PACKAGE_TYPE                    MULTI_NODE
#FAILOVER_POLICY                CONFIGURED_NODE
#FAILBACK_POLICY                MANUAL
NODE_NAME                       mo
NODE_NAME                       minie
RUN_SCRIPT                      /etc/cmcluster/crsp/crsp.ctl
HALT_SCRIPT                     /etc/cmcluster/crsp/crsp.ctl

DEPENDENCY_NAME                 SG-CFS-MP-1
DEPENDENCY_CONDITION            SG-CFS-MP-1=UP
DEPENDENCY_LOCATION             SAME_NODE

DEPENDENCY_NAME                 SG-CFS-MP-2
DEPENDENCY_CONDITION            SG-CFS-MP-2=UP
DEPENDENCY_LOCATION             SAME_NODE

DEPENDENCY_NAME                 SG-CFS-MP-3
DEPENDENCY_CONDITION            SG-CFS-MP-3=UP
DEPENDENCY_LOCATION             SAME_NODE

SERVICE_NAME                    crsp-srv
SERVICE_FAIL_FAST_ENABLED       NO
SERVICE_HALT_TIMEOUT            300
```

4. Edit the package control script `crsp.ctl`.

```
SERVICE_NAME[0]="crsp-srv"
SERVICE_CMD[0]="/etc/cmcluster/crsp/toolkit_oc.sh check"
SERVICE_RESTART[0]=""

function customer_defined_run_cmds
{
```

```
# ADD customer defined run commands.

        /etc/cmcluster/crsp/toolkit_oc.sh start
        test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
        /etc/cmcluster/crsp/toolkit_oc.sh stop
        test_return 52
}
```

5. Edit the toolkit configuration file `oc.conf`.

```
ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs
```

6. Add the package to the cluster. Distribute Oracle Clusterware multi-node package (MNP) directory to all nodes.

```
# cd /etc/cmcluster
# rcp –r crsp root@minie:/etc/cmcluster
```

Add package to cluster.

```
# cmapplyconf –P crsp.conf
Modify the cluster configuration ([y]/n)? y
Completed the cluster creation
```

Starting and stopping Serviceguard packages and Oracle RAC
On each node, halt Oracle Clusterware if running.

```
# $ORA_CRS_HOME/bin/crsctl stop crs
```

Start the complete stack by running the Serviceguard Package.

```
# cmrunpkg crsp

# cmviewcl

CLUSTER        STATUS
cluster_mo     up

  NODE           STATUS       STATE
  minie          up           running
  mo             up           running

MULTI_NODE_PACKAGES

  PACKAGE        STATUS       STATE        AUTO_RUN     SYSTEM
  SG-CFS-pkg     up           running      enabled      yes
  SG-CFS-DG-1    up           running      enabled      no
  SG-CFS-DG-2    up           running      enabled      no
  SG-CFS-DG-3    up           running      enabled      no
  SG-CFS-DG-4    up           running      enabled      no
  SG-CFS-MP-1    up           running      enabled      no
  SG-CFS-MP-2    up           running      enabled      no
  SG-CFS-MP-3    up           running      enabled      no
  crsp           up           running      enabled      no
```

Verify Oracle Clusterware status.

```
# $ORA_CRS_HOME/bin/crsctl check crs
CSS appears healthy
CRS appears healthy
EVM appears healthy
```

Cluster start and stop

The following sections describe the process for starting and stopping Oracle 10g Clusterware.

Start and stop Oracle Clusterware 10g

Placing the start and stop of Oracle Clusterware in Serviceguard packages ensures that the shared storage required by Oracle Clusterware is available.

When halting the Serviceguard cluster ("`cmhaltcl –f`"), the package dependencies ensure that the Oracle Clusterware packages are halted first before the cluster file systems are unmounted.

When using "`crsctl stop crs`" to stop Oracle Clusterware while Oracle RAC instances are running, stopping Oracle Clustware causes the instances to shutdown abort. This is the default behavior since other shutdown modes may not complete in time for Oracle Clusterware to stop in a timely manner.

If a different Oracle RAC instance shutdown mode (for example normal, immediate) is desired, the instances should be halted prior to initiating Oracle Clusterware shutdown.

Start and stop Oracle RAC 10g Instance

In this sample configuration, the Oracle RAC instances startup and shutdown are controlled by Oracle Clusterware.

Appendix

Sample configuration for SLVM with Serviceguard Extension for RAC

Cluster configuration File for Cluster (eenie and meenie)

```
CLUSTER_NAME            cluster_eenie
FIRST_CLUSTER_LOCK_VG   /dev/vg_ops

NODE_NAME          eenie
  NETWORK_INTERFACE     lan0
    STATIONARY_IP 15.13.170.64
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP  192.1.1.1
  NETWORK_INTERFACE     lan2
  FIRST_CLUSTER_LOCK_PV /dev/dsk/c4t3d0

NODE_NAME          meenie
  NETWORK_INTERFACE     lan0
    STATIONARY_IP 15.13.170.80
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP  192.1.1.2
  NETWORK_INTERFACE     lan2
  FIRST_CLUSTER_LOCK_PV /dev/dsk/c4t3d0

AUTO_START_TIMEOUT      600000000
NETWORK_POLLING_INTERVAL    2000000
NETWORK_FAILURE_DETECTION   INOUT
MAX_CONFIGURED_PACKAGES     150

OPS_VOLUME_GROUP  /dev/vg_ops
```

For A.11.18 and prior, the heartbeat timeout is

```
HEARTBEAT_INTERVAL      1000000
NODE_TIMEOUT            2000000
```

For A.11.19 and later, the heartbeat timeout is

```
MEMBER_TIMEOUT          14000000
```

Package configuration for SLVM (Oracle Clusterware MNP)

```
PACKAGE_NAME                    crsp-slvm
PACKAGE_TYPE                    MULTI_NODE
#FAILOVER_POLICY                CONFIGURED_NODE
#FAILBACK_POLICY                MANUAL
NODE_NAME                       eenie
NODE_NAME                       meenie
RUN_SCRIPT                      /etc/cmcluster/crsp-slvm/crsp-slvm.ctl
HALT_SCRIPT                     /etc/cmcluster/crsp-slvm/crsp-slvm.ctl
SERVICE_NAME                    crsp-slvm-srv
SERVICE_FAIL_FAST_ENABLED       NO
SERVICE_HALT_TIMEOUT            300
```

Package control script for CFS (Oracle Clusterware MNP)

```
VGCHANGE="vgchange -a s"
VG[0]="vg_ops"
SERVICE_NAME[0]="crsp-slvm-srv"
```

```
SERVICE_CMD[0]="/etc/cmcluster/crsp-slvm/toolkit_oc.sh check"
SERVICE_RESTART[0]=""

function customer_defined_run_cmds
{
# ADD customer defined run commands.
       /etc/cmcluster/crsp-slvm/toolkit_oc.sh start
        test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
       /etc/cmcluster/crsp-slvm/toolkit_oc.sh stop
        test_return 52
}
```

Serviceguard Extension for RAC configuration for Oracle Clustware

```
ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs
```

Sample configuration for SLVM without Serviceguard Extension for RAC

The following sections describe sample configurations with SLVM if the SGeRAC Toolkit is not used.

Cluster configuration file for cluster (“clm.asc”)

```
CLUSTER_NAME            cluster_eenie
FIRST_CLUSTER_LOCK_VG   /dev/vg_ops

NODE_NAME         eenie
  NETWORK_INTERFACE     lan0
    STATIONARY_IP 15.13.170.64
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP  192.1.1.1
  NETWORK_INTERFACE     lan2
  FIRST_CLUSTER_LOCK_PV /dev/dsk/c4t3d0

NODE_NAME         meenie
  NETWORK_INTERFACE     lan0
    STATIONARY_IP 15.13.170.80
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP  192.1.1.2
  NETWORK_INTERFACE     lan2
  FIRST_CLUSTER_LOCK_PV /dev/dsk/c4t3d0

AUTO_START_TIMEOUT      600000000
NETWORK_POLLING_INTERVAL    2000000
NETWORK_FAILURE_DETECTION   INOUT
MAX_CONFIGURED_PACKAGES     150

OPS_VOLUME_GROUP  /dev/vg_ops
```

For A.11.18 and prior, the heartbeat timeout is

```
HEARTBEAT_INTERVAL      1000000
NODE_TIMEOUT            2000000
```

For A.11.19 and later, the heartbeat timeout is

```
MEMBER_TIMEOUT          14000000
```

Package configuration file for node eenie for SLVM (“crs_eenie_pkg.conf”)

```
PACKAGE_NAME      crs_eenie_pkg
NODE_NAME         eenie
RUN_SCRIPT        /etc/cmcluster/pkg/crs_eenie_pkg/crs_eenie_pkg.sh
HALT_SCRIPT       /etc/cmcluster/pkg/crs_eenie_pkg/crs_eenie_pkg.sh

SERVICE_NAME                css_check_eenie
SERVICE_FAIL_FAST_ENABLED   NO
SERVICE_HALT_TIMEOUT        300
```

Package control file for node eenie for SLVM

```
VGCHANGE="vgchange -a s"
VG[0]="vg_ops"

SERVICE_NAME[0]="css_check_eenie"
SERVICE_CMD[0]="/etc/cmcluster/pkg/crs_eenie_pkg/cssd.sh monitor"
SERVICE_RESTART[0]=""

function customer_defined_run_cmds
{
# ADD customer defined run commands.

        /etc/cmcluster/pkg/crs_eenie_pkg/cssd.sh start
        test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.

        /etc/cmcluster/pkg/crs_eenie_pkg/cssd.sh stop
        test_return 52
}
```

Note:
The `cssd.sh` script is a sample script that is in the Appendix for starting, monitoring, and stopping OC.

Package configuration file for node “meenie” for CRS for SLVM

```
PACKAGE_NAME      crs_meenie_pkg
NODE_NAME         meenie
RUN_SCRIPT        /etc/cmcluster/pkg/crs_meenie_pkg/crs_meenie_pkg.sh
HALT_SCRIPT       /etc/cmcluster/pkg/crs_meenie_pkg/crs_meenie_pkg.sh

SERVICE_NAME                css_check_meenie
SERVICE_FAIL_FAST_ENABLED   NO
SERVICE_HALT_TIMEOUT        300
```

Package control file for node “meenie” for CRS for SLVM

```
VGCHANGE="vgchange -a s"              # Default
```

```
VG[0]="vg_ops"

SERVICE_NAME[0]="css_check_meenie"
SERVICE_CMD[0]="/etc/cmcluster/pkg/crs_meenie_pkg/cssd.sh monitor"
SERVICE_RESTART[0]=""


function customer_defined_run_cmds
{
# ADD customer defined run commands.

        /etc/cmcluster/pkg/crs_meenie_pkg/cssd.sh start
        test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.

        /etc/cmcluster/pkg/crs_meenie_pkg/cssd.sh stop
        test_return 52
}
```

Note:
The `cssd.sh` script is a sample script that is in the Appendix for starting, monitoring, and stopping OC.

Sample “cmviewcl –v” output

```
CLUSTER         STATUS
cluster_eenie   up

  NODE           STATUS       STATE
  eenie          up           running

    Cluster_Lock_LVM:
    VOLUME_GROUP          PHYSICAL_VOLUME       STATUS

    /dev/vg_ops           /dev/dsk/c4t3d0       up

    Network_Parameters:
    INTERFACE    STATUS       PATH                 NAME
    PRIMARY      up           0/0/0/0              lan0
    PRIMARY      up           0/8/0/0/4/0          lan1
    STANDBY      up           0/8/0/0/6/0          lan3
    STANDBY      up           0/8/0/0/5/0          lan2

    PACKAGE          STATUS       STATE        AUTO_RUN     NODE
    crs_eenie_pkg    up           running      disabled     eenie

      Policy_Parameters:
      POLICY_NAME     CONFIGURED_VALUE
      Failover        configured_node
      Failback        manual

      Script_Parameters:
      ITEM       STATUS   MAX_RESTARTS  RESTARTS   NAME
      Service    up       0             0          css_check_eenie
```

```
    Node_Switching_Parameters:
    NODE_TYPE    STATUS       SWITCHING    NAME
    Primary      up           enabled      eenie (current)

NODE           STATUS       STATE
meenie         up           running

  Cluster_Lock_LVM:
  VOLUME_GROUP          PHYSICAL_VOLUME       STATUS
  /dev/vg_ops           /dev/dsk/c4t3d0       up

  Network_Parameters:
  INTERFACE    STATUS       PATH                 NAME
  PRIMARY      up           0/0/0/0              lan0
  PRIMARY      up           0/8/0/0/4/0          lan1
  STANDBY      up           0/8/0/0/6/0          lan3
  STANDBY      up           0/8/0/0/5/0          lan2

  PACKAGE          STATUS       STATE        AUTO_RUN     NODE
  crs_meenie_pkg   up           running      disabled     meenie

    Policy_Parameters:
    POLICY_NAME     CONFIGURED_VALUE
    Failover        configured_node
    Failback        manual

    Script_Parameters:
    ITEM       STATUS   MAX_RESTARTS  RESTARTS   NAME
    Service    up       0             0          css_check_meenie

    Node_Switching_Parameters:
    NODE_TYPE    STATUS       SWITCHING    NAME
    Primary      up           enabled      meenie (current)
```

Sample configuration for CFS

The following sections describe sample configurations with CFS:

Cluster configuration Ffile for cluster (minie and mo)

```
CLUSTER_NAME            cluster_mo
QS_HOST                 white
QS_POLLING_INTERVAL     120000000
QS_TIMEOUT_EXTENSION    2000000

NODE_NAME               minie
  NETWORK_INTERFACE     lan0
    STATIONARY_IP       15.13.170.82
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP        192.1.1.3
  NETWORK_INTERFACE     lan2

NODE_NAME               mo
  NETWORK_INTERFACE     lan0
    STATIONARY_IP       15.13.171.137
  NETWORK_INTERFACE     lan3
  NETWORK_INTERFACE     lan1
    HEARTBEAT_IP        192.1.1.4
  NETWORK_INTERFACE     lan2

AUTO_START_TIMEOUT      600000000
NETWORK_POLLING_INTERVAL 2000000
NETWORK_FAILURE_DETECTION       INOUT
MAX_CONFIGURED_PACKAGES         150
```

For A.11.18 and prior, the heartbeat timeout is

```
HEARTBEAT_INTERVAL      1000000
NODE_TIMEOUT            5000000
```

For A.11.19 and later, the heartbeat timeout is

```
MEMBER_TIMEOUT          14000000
```

Package configuration for CFS (Oracle Clusterware MNP)

```
PACKAGE_NAME                crsp
PACKAGE_TYPE                MULTI_NODE
#FAILOVER_POLICY            CONFIGURED_NODE
#FAILBACK_POLICY            MANUAL
NODE_NAME                   mo
NODE_NAME                   minie
RUN_SCRIPT                  /etc/cmcluster/crsp/crsp.ctl
HALT_SCRIPT                 /etc/cmcluster/crsp/crsp.ctl

DEPENDENCY_NAME             SG-CFS-MP-1
DEPENDENCY_CONDITION        SG-CFS-MP-1=UP
DEPENDENCY_LOCATION         SAME_NODE

DEPENDENCY_NAME             SG-CFS-MP-2
DEPENDENCY_CONDITION        SG-CFS-MP-2=UP
DEPENDENCY_LOCATION         SAME_NODE

DEPENDENCY_NAME             SG-CFS-MP-3
DEPENDENCY_CONDITION        SG-CFS-MP-3=UP
```

```
DEPENDENCY_LOCATION           SAME_NODE

SERVICE_NAME                  crsp-srv
SERVICE_FAIL_FAST_ENABLED     NO
SERVICE_HALT_TIMEOUT          300
```

Package control script for CFS (Oracle Clusterware MNP)

```
SERVICE_NAME[0]="crsp-srv"
SERVICE_CMD[0]="/etc/cmcluster/crsp/toolkit_oc.sh check"
SERVICE_RESTART[0]=""

function customer_defined_run_cmds
{
# ADD customer defined run commands.

        /etc/cmcluster/crsp/toolkit_oc.sh start
        test_return 51
}

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.

        /etc/cmcluster/crsp/toolkit_oc.sh stop
        test_return 52
}
```

Serviceguard Extension for RAC configuration for Oracle Clusterware

```
ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs
```

Sample scripts used by package control scripts

The following sections provide sample scripts used by Package Control Scripts to Start and Stop Oracle Cluster Software. This is an example for use with SGeRAC A.11.16.

Sample script to start or stop Oracle Cluster Software ("cssd.sh")

The script is a sample script that is used to aid the Serviceguard package for starting, stopping, and monitoring Oracle Clusterware.

Please note that "crsctl stop crs" is used to stop Oracle Clusterware and any Oracle RAC instances running are shutdown abort by Oracle Clusterware. If a different RAC instance shutdown mode is desire, the instance needs to be halted before initiating Oracle Clusterware stop.

```
#!/usr/bin/sh
##############################################################################
###
# (C) Copyright 2005 Hewlett-Packard Development Company, L.P.
##############################################################################
###
# This script is provided as a sample.
##############################################################################
###
# Arguments
#         "Usage: ${0} <start|stop|monitor|enable|disable >"
#
# <start>   Starts the Oracle Cluster Software
# <stop>    Stops the Oracle Cluster Software
# <monitor> Monitors the Oracle Cluster Software (ocssd.bin)
```

```
# <enable>      Enable the Oracle Cluster Software to autostart
# <disable>       Disable the Oracle Cluster Software from autostart
################################################################################
###
# Function: log_message
#
# This function log any message with date, time and node name affixed
# to it. It accepts just one parameter.
# Parameter:
# 1. Message to be logged
################################################################################
####

function log_message
{
    if [ $# -eq 1 ]; then
        echo "$(date '+%b %e %T') - Node \"${HOST}\" $1 "
    else
        echo
    fi
}

################################################################################
####
# Function: cssd_run_cmds
#
# Start CRS cssd daemons
################################################################################
####
function cssd_run_cmds
{

  set -A TMP_MONITOR_PROCESSES ${CSSD_MONITOR_PROCESSES[@]}

  typeset -i c
  typeset -i tmp_num_procs=${#TMP_MONITOR_PROCESSES[@]}

  $ORA_CRS_HOME/bin/crsctl start crs

  # Wait for daemon to start

    while true
    do
        for i in ${TMP_MONITOR_PROCESSES[@]}
        do
            id=`ps -fu $ORACLE_USER | awk '/'${i}$'/ { print $2 }'`
            if [[ ${id} != "" ]]
            then
                print "\n *** ${i} process has started. ***\n"
                c=0
                while (( c < $tmp_num_procs ))
                do
                    if [[ ${TMP_MONITOR_PROCESSES[$c]} = $i ]]
                    then
                        unset TMP_MONITOR_PROCESSES[$c]
                        c=$tmp_num_procs
                    fi
                    (( c = c + 1 ))
                done
            fi
        done

        if [[ ${TMP_MONITOR_PROCESSES[@]} = "" ]]
```

```
        then
            break
        fi

        sleep $MONITOR_INTERVAL
    done
}

################################################################################
####
# Function: cssd_stop_cmds
#
# Stop cssd daemons
################################################################################
####
function cssd_stop_cmds
{
   typeset -i n=0

    # Grab the PID of the CSS daemon

    for i in ${CSSD_MONITOR_PROCESSES[@]}
    do
        CSSD_MONITOR_PROCESSES_PID[$n]=`ps -fu $ORACLE_USER | awk
'/'${i}$'/ { print $2 }'`
        print "Monitored process = ${i}, pid =
${CSSD_MONITOR_PROCESSES_PID[$n]}
"
        if [[ ${CSSD_MONITOR_PROCESSES_PID[$n]} = "" ]]
        then
            print "\n\n"
            ps -ef
            print "\n *** ${i} is not running  ***"
        fi
        (( n = n + 1 ))
    done

   $ORA_CRS_HOME/bin/crsctl stop crs

   # wait until CSS process goes away

    while true
    do
        for i in ${CSSD_MONITOR_PROCESSES_PID[@]}
        do
            kill -s 0 ${i} > /dev/null
            if [[ $? != 0 ]]
            then
                print "\n\n"
                print "\n *** ${i} has stopped.  ***"
                return 0
            fi
        done

        sleep ${MONITOR_INTERVAL}
    done

}

################################################################################
####
# Function: monitor_processes
#
```

```
# Monitor cssd daemons
################################################################################
####
function monitor_processes
{
   typeset -i n=0

    # Grab the PID of the CSS daemon

    for i in ${CSSD_MONITOR_PROCESSES[@]}
    do
        CSSD_MONITOR_PROCESSES_PID[$n]=`ps -fu $ORACLE_USER | awk
'/'${i}$'/ { print $2 }'`
        print "Monitored process = ${i}, pid =
${CSSD_MONITOR_PROCESSES_PID[$n]}
"
        if [[ ${CSSD_MONITOR_PROCESSES_PID[$n]} = "" ]]
        then
            print "\n\n"
            ps -ef
            print "\n *** ${i} is not running  ***"
          return 0
        fi
        (( n = n + 1 ))
    done

   # wait until CSS process goes away

    while true
    do
        for i in ${CSSD_MONITOR_PROCESSES_PID[@]}
        do
            kill -s 0 ${i} > /dev/null
            if [[ $? != 0 ]]
            then
                print "\n\n"
                ps -ef
                print "\n *** ${i} has stopped.  ***"
                return 0
            fi
        done

        sleep ${MONITOR_INTERVAL}
    done

}
################################################################################
####
# Function: css_enable_cmds
#
# Enable CSS to auto start
################################################################################
####
function css_enable_cmds
{
   $ORA_CRS_HOME/bin/crsctl enable crs
}
################################################################################
####
# Function: css_disable_cmds
#
# Disable CSS from auto start
```

```
#############################################################################
####
function css_disable_cmds
{
   $ORA_CRS_HOME/bin/crsctl disable crs
}

#############################################################################
####
# MAIN
# Check the command-line option and take the appropriate action.
#############################################################################
####
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/lbin
ORA_ver=10.2.0.1
HOST=`hostname`
DATE=`date`
PKG_DIR=${0%/*}
exit_code=0

set -A CSSD_MONITOR_PROCESSES ocssd.bin
ORA_CRS_HOME=/mnt/app/crs/oracle/product/10.2.0/crs

TIME_OUT=300
ORACLE_USER=oracle
MONITOR_INTERVAL=5

log_message "*** $0 called with $1 argument. ***"
case $1 in

  start)
      log_message ": Starting Oracle CSS at ${DATE}"
      cssd_run_cmds
  ;;

  stop)
      log_message ": Stopping Oracle CSS at ${DATE}"
      cssd_stop_cmds
  ;;

  monitor)
      monitor_processes
  ;;

  enable)
      css_enable_cmds
  ;;

  disable)
      css_disable_cmds
  ;;
    *)
        log_message "Usage: ${0} <start|stop|monitor|enable|disable>"
    ;;
esac
exit $exit_code
```

Document revision history

Revision	Date	Description	Comment
1.0	Dec 6,2005	First version	
1.1	Jan 23, 2006	Minor update	OC files on CFS
1.2	Feb 13, 2006	Minor update	From extended team feedback
1.2.1	Feb 21, 2006	Minor update	Directory ownership
1.2.2	Feb 28, 2006	Minor update	Directory ownership
1.2.3	Apri 7, 2006	Minor update	Add ODM issue and oracle bug #
1.3.0	May 10, 2006	Minor update	Update IB, add RIP/VIP co-existence
1.3.1	May 30, 2006	Minor update	Reduced material on IB, update ODM issue
1.4	July 2006	Ext. version	Update with external format and feedback
1.5	May 2007	Minor update	Update section on planning for HA. Update package configuration with SGeRAC toolkit
1.6	March 2009	Minor update	Update for A.11.19

For more information

www.hp.com/go/serviceguardsolutions
www.hp.com/go/sgerac

HP documentation

All of the following materials can be found on the HP Technical Documentation web site at http://docs.hp.com.

- HP Serviceguard Storage Management Suite Version Release Notes
  http://docs.hp.com/en/ha.html
- Serviceguard Version A.11.19 Release Notes
  http://docs.hp.com/en/ha.html
- Serviceguard Extension for RAC Version A.11.19 Release Notes
  http://docs.hp.com/en/ha.html
- Managing Serviceguard Sixteenth Edition
  http://docs.hp.com/en/ha.html
- Using Serviceguard Extension for RAC Manual Sixth Edition
  http://docs.hp.com/en/ha.html
- Symantec Veritas Storage Foundation 5.0 for Oracle RAC Configuration Guide Extracts for HP Serviceguard Storage Management Suite Extracts
  http://docs.hp.com/en/T2771-90046/T2771-90046_FINAL4-24.pdf
- Symantec Veritas Storage Foundation 4.1 for Oracle RAC HP Serviceguard Storage Management Suite Extracts
  http://docs.hp.com/en/7412/T2771-90010.pdf
- Use of Serviceguard Extension For RAC Toolkit with Oracle RAC 10g Release 2 or later RAC White Paper, April 2009
  http://docs.hp.com/en/ha.html
- Support of Oracle RAC ASM with SGeRAC White Paper, January 2008
  http://docs.hp.com/en/8988/ASM-SGeRAC-tk.pdf

Oracle documentation

All of the following materials can be found on the Oracle Technical Documentation web site at http://www.oracle.com/technology/documentation/database10gr2.html.

- Oracle Clusterware and Oracle Real Application Clusters Installation Guide version 10g Release 2 (10.2) for HP-UX
  http://download-west.oracle.com/docs/cd/B19306_01/install.102/b14202.pdf
- Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide version 10g Release 2 (10.2)
  http://download-west.oracle.com/docs/cd/B19306_01/rac.102/b14197.pdf
- Oracle Net Services Administrator's Guide
  http://download-west.oracle.com/docs/cd/B19306_01/network.102/b14212.pdf
- Client Failover Best Practices for Highly Available Oracle Database: Oracle Database 10g Release
  http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_ClientFailoverBestPractices.pdf
- Note:296874.1 Configuring the HP-UX Operating System for the Oracle 10g VIP at https://metalink.oracle.com/ (Oracle MetaLink account required)

April 2009