Chapter 15

Serviceguard

HP-UX Handbook
Revision 13.00

TERMS OF USE AND LEGAL RESTRICTIONS FOR THE HP-UX RECOVERY HANDBOOK

ATTENTION: PLEASE READ THESE TERMS CAREFULLY BEFORE USING THE HP-UX HANDBOOK. USING THESE MATERIALS INDICATES THAT YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THESE TERMS, DO NOT USE THE HP-UX HANDBOOK.

THE HP-UX HANDBOOK HAS BEEN COMPILED FROM THE NOTES OF HP ENGINEERS AND CONTAINS HP CONFIDENTIAL INFORMATION.

THE HP-UX HANDBOOK IS NOT A PRODUCT QUALITY DOCUMENT AND IS NOT NECESSARILY MAINTAINED OR UP TO DATE. THE HP-UX HANDBOOK IS HERE MADE AVAILABLE TO HP CONTRACT CUSTOMERS FOR THEIR INTERNAL USE ONLY AND ON THE CONDITION THAT NEITHER THE HP-UX HANDBOOK NOR ANY OF THE MATERIALS IT CONTAINS IS PASSED ON TO ANY THIRD PARTY.

Use of the HP-UX Handbook: Hewlett-Packard Company ("HP") authorizes you to view and download the HP-UX Handbook only for internal use by you, a valued HP Contract Customer, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials. You may not modify the HP-UX Handbook in any way or publicly display, perform, or distribute or otherwise use them for any public or purpose outside your own business. The materials comprising the HP-UX Handbook are copyrighted and any unauthorized use of these materials may violate copyright, trademark, and other laws. If you breach any of these Terms, your authorization to use the HP-UX Handbook automatically terminates and you must immediately destroy any downloaded or printed materials.

Links To Other Web Sites: Links to third party Web sites provided by the HP-UX Handbook are provided solely as a convenience to you. If you use these links, you will leave this Site. HP has not reviewed all of these third party sites and does not control and is not responsible for any of these sites or their content. Thus, HP does not endorse or make any representations about them, or any information, software or other products or materials found there, or any results that may be obtained from using them. If you decide to access any of the third party sites linked to this Site, you do this entirely at your own risk.

Disclaimer: THE HP-UX HANDBOOK IS PROVIDED "AS IS" WITHOUT ANY WARRANTIES OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. HP further does not warrant the accuracy and completeness of the materials in the HP-UX Handbook. HP may make changes to the HP-UX Handbook at any time without notice. The HP-UX Handbook may be out of date, and HP makes no commitment to update the HP-UX Handbook. Information in the HP-UX Handbook may refer to products, programs or services that are not available in your country. Consult your local HP business contact for information regarding the products, programs and services that may be available to you.

Limitation of Liability: IN NO EVENT WILL HP, ITS SUPPLIERS, OR OTHER ANY THIRD PARTIES MENTIONED IN THE HP-UX HANDBOOK BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOST PROFITS, LOST DATA OR BUSINESS INTERRUPTION) ARISING OUT OF THE USE, INABILITY TO USE, OR THE RESULTS OF USE OF THE HP-UX HANDBOOK, WHETHER BASED ON WARRANTY, CONTRACT, TORT OR ANY OTHER LEGAL THEORY AND WHETHER OR NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS DOES NOT APPLY IN CASE OF INTENT OR IF LIABILITY IS LEGALLY STIPULATED. IF YOUR USE OF THE HP-UX HANDBOOK RESULTS IN THE NEED FOR SERVICING, REPAIR OR CORRECTION OF EQUIPMENT OR DATA, YOU ASSUME ALL COSTS THEREOF.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Applicable Laws: These Terms will be governed by and construed in accordance with the laws of the State of California, without giving effect to any principles of conflicts of laws.

General: HP may revise these Terms at any time by updating this posting. ***Revised Oct 2013***

FEEDBACK or QUESTIONS: please email essam.ackleh@hp.com
(please use subject syntax: *HP-UX Handbook v13.00 Chapter <YY> - <Feedback Title>*

TABLE OF CONTENTS

Introduction

When Serviceguard was first marketed, it was titled MC ServiceGuard. Today, marketing has simplified this to just **Serviceguard**. Any document that uses the older title was probably generated last century in the days of HPUX 10.20 and Serviceguard A.10.06.

This chapter introduces High Availability solutions using Serviceguard and how to setup, maintain, and troubleshoot them. It is *not* intended to replace the official documentation, especially the *Managing Serviceguard* manual, downloadable from http://www.hp.com/go/hpux-SG-docs. In fact, some parts are extracted and concentrated from that and from other internal and external documents.

This chapter will cover Serviceguard on HPUX only. Product development for Serviceguard on Linux has ceased and information on it is not part of this chapter.

All commands and troubleshooting procedures are based on Serviceguard Version A.11.19.00 for HPUX 11iv2 (11.23) and/or A.11.20 for HPUX 11iv3 (11.31) and before.

Special configurations like MetroCluster and ContinentalClusters will not be discussed. Also only a very short introduction will be provided for Serviceguard Extension for SAP (SGeSAP) and Serviceguard Extension for RAC (SGeRAC).

Since Serviceguard configurations for these special extensions may be complex to configure and maintain, **it is strongly recommended that you recommend Hewlett-Packard's high availability consulting services to ensure a smooth installation and rollout.**

Sections of this Chapter:

- What is Serviceguard?
- Supported versions of Serviceguard and associated features
- Useful Procedures and Commands
- Troubleshooting Serviceguard
- Common Issues
- Prevent Split Brain
- Type of Volume Managers
- SGeSAP
- SGeRAC
- Serviceguard Manager
- Patches
- Command Overview
- Serviceguard related Files
- Additional Informations

What is Serviceguard?

Serviceguard allows you to create high availability clusters of HP Integrity or HP 9000 servers. A high availability cluster promotes critical business application availability in spite of a hardware or software failure. Highly available systems protect users from software failures as well as from failure of a system processing unit (SPU), disk, or local area network (LAN) component. Hardware redundancy is key to high availability. In the event that one component fails, the redundant component takes over. Serviceguard and other high availability subsystems coordinate the transfer between components. Serviceguard can operate on both physical systems and virtual machines.

What is a Serviceguard Cluster

A Serviceguard **cluster** begins with is a group of 1-16 HP 9000 or HP Integrity servers or virtual machines called **nodes** that are configured with hardware and software redundancy to avoid a **single point of failure** (SPOF). Redundant components should use a separate power source to prevent a single point of failure. Serviceguard is automation software that starts and stops the cluster, starts/monitors/stops **packages**, and reacts to network failures and node and application service failures by either bringing standby NICs online or moving packages to **adoptive** nodes.

Figure 1 illustrates a robust cluster configuration that eliminates SPOFs. Note the user’s path to data for each application.

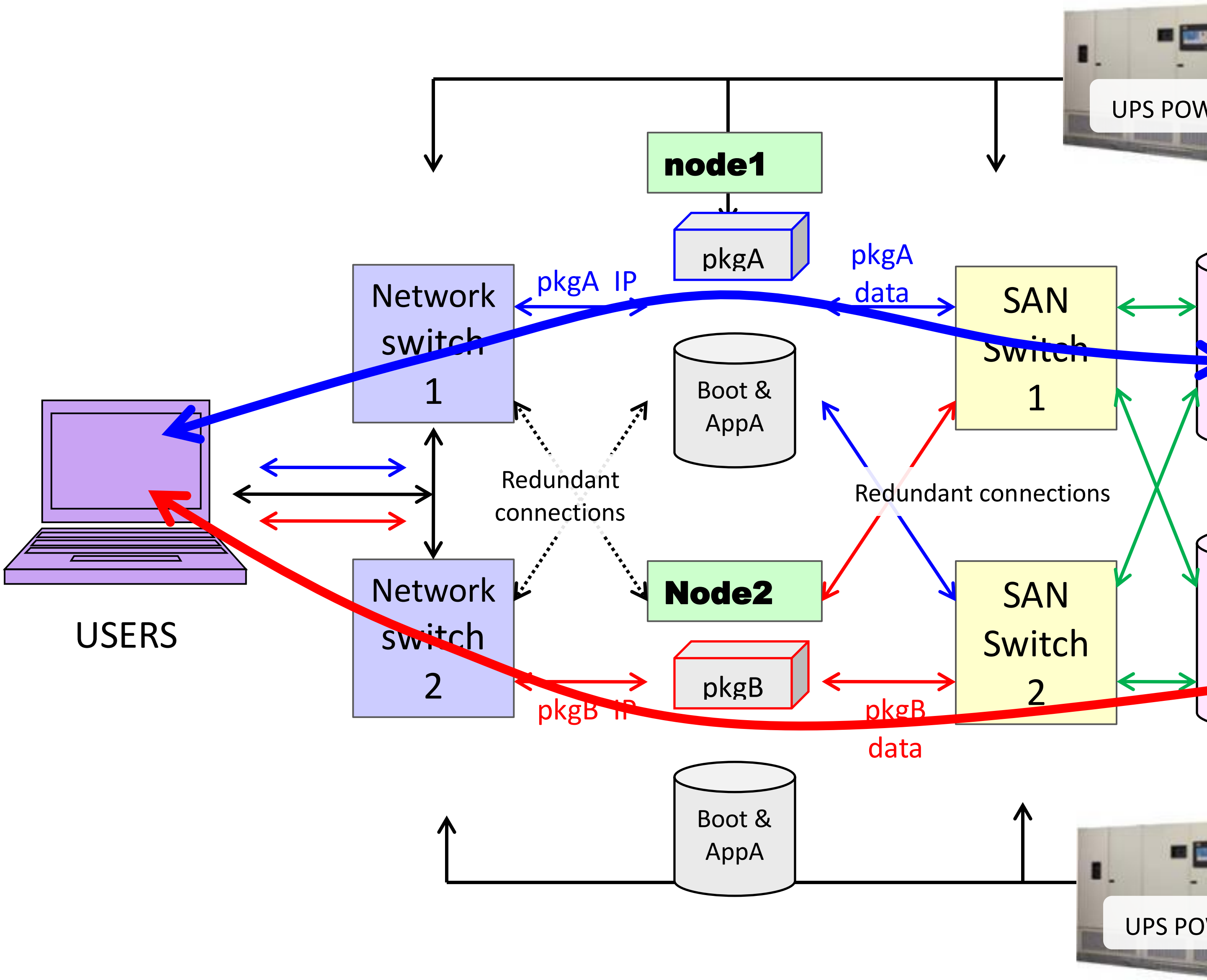


Figure 1: Robust Cluster Configuration

Figure 1 also illustrates the concept of active/active cluster. Each system is a **primary node** for one or more Serviceguard packages and **adoptive node** for the other node's packages in the event of a node or site failure. Active/Active environments distribute the solution load. If possible, half of the cluster should reside at a different site to avoid outages due to localized failures. It is possible to create a cluster without packages.

Clusters are created, managed and deleted using cmquerystg, cmpreparestg, cmpreparecl, cmdeploycl, cmquerycl, cmcheckconf, cmapplyconf, cmdeleteconf, cmgetconf, cmhaltcl, cmhaltnode, cmmodnet, cmruncl, cmrunnode, cmversion, cmviewcl and cmviewconf.

Packages

A cluster can operate up to 300 packages. A package identifies unique system resources, business applications that need those resources and application monitor services, and the manner in which they are activated or started when the package is started. The package also defines the manner in which to halt those applications and deactivate system resources. Since a package may operate on a different node, packages requiring SAN storage require that adoptive nodes are zoned to the shared storage.

Package styles

There are two styles of packages – **legacy** and **modular**. A package is defined by a modular* configuration file or legacy configuration file and a control script. Modular and legacy packages can co-exist on a cluster. *modular package capability was introduced in A.11.18.
The legacy-style package configuration file contains some parameters that reference the package name, package control script path and other features. The legacy control script contains additional parameters defining volume groups or disk groups, logical volumes and their mount directories and options, and relocatable IPs. The control script usually also identifies external application start/stop scripts and application monitor scripts. One drawback of a 2-file concept was that the admin often forgot to copy updates to package scripts to adoptive nodes.

The modular-style package configuration file includes all customizable parameters found in both legacy configuration and control files. This enables the control script functionality to be generic across nodes in the cluster. Many enablement toolkits include modular package integration.

Both legacy and modular package configuration file parameters must be loaded in the cluster binary file using cmapplyconf. cmapplyconf cannot be used with legacy package control scripts.

Failover packages are created/managed/deleted using the commands cmapplyconf, cmgetpkgenv, cmhaltpkg, cmmakepkg, cmmigratepkg, cmmodpkg and cmrunpkg and cmdeleteconf.

Package classes

There are also 3 classes of packages - **failover**, **system multi-node (SMN)**, and **multi-node (MN)**. Only failover packages can be modular packages until A.11.20 Patch B is installed. Then the admin can also create modular MN packages. A failover package can run on only one system at a time. This means that package-related system resources can only be activated on one node at any given time. Failover packages may TOC a node when configuration criteria are met.

SMN and MN packages are non-failover packages and were introduced with a patch for A.11.17. The only supported SMN package is named SG-CFS-pkg. If the CFS bundle has been installed and the admin configures SG-CFS-pkg, it starts VERITAS Cluster Volume Manager (cvmd) and it must run on all active nodes in order to enable CFS MN packages. Failure of the SG-CFS-pkg on a node causes that node to TOC. MN packages can run on multiple active nodes simultaneously and are typically dedicated to supporting Cluster File System which is available in specific **Storage Management Suite** (SMS) products. SMS-based packages use only VxVM disk groups and volumes. As of A.11.18, customers could create non-CFS-based multi-node packages. As of this writing, SMS and MN packages based on CFS are built and destroyed using cfscluster, cfsdgadm cfsmntadm, cfsmount and cfsumount

Serviceguard monitors node heartbeat transmission, NIC failure and package status by default. Additional monitors can be added to a package. If any of these **failover triggers** occurs, Serviceguard may switch traffic to a standby NIC or cause a node to TOC and failover-class packages to start on adoptive nodes to automatically restore business services with minimal interruption.

Supported versions of Serviceguard and associated features

Current supported versions of Serviceguard are identified in a customer-viewable document titled 'HP Serviceguard Support Model", which is located at this link http://www.hp.com/softwarereleases/releases-media2/. Additionally, the customer-viewable document titled "HP Serviceguard/SGeRAC/Storage Management Suite/Serviceguard Manager Plug-in Compatibility and Feature Matrix" at http://www.hp.com/go/hpux-SG-docs provides quite a bit of useful compatibility and other information about each supported version.

Failure Management

The following figure demonstrates that though hardware may fail, users can still get to the data.

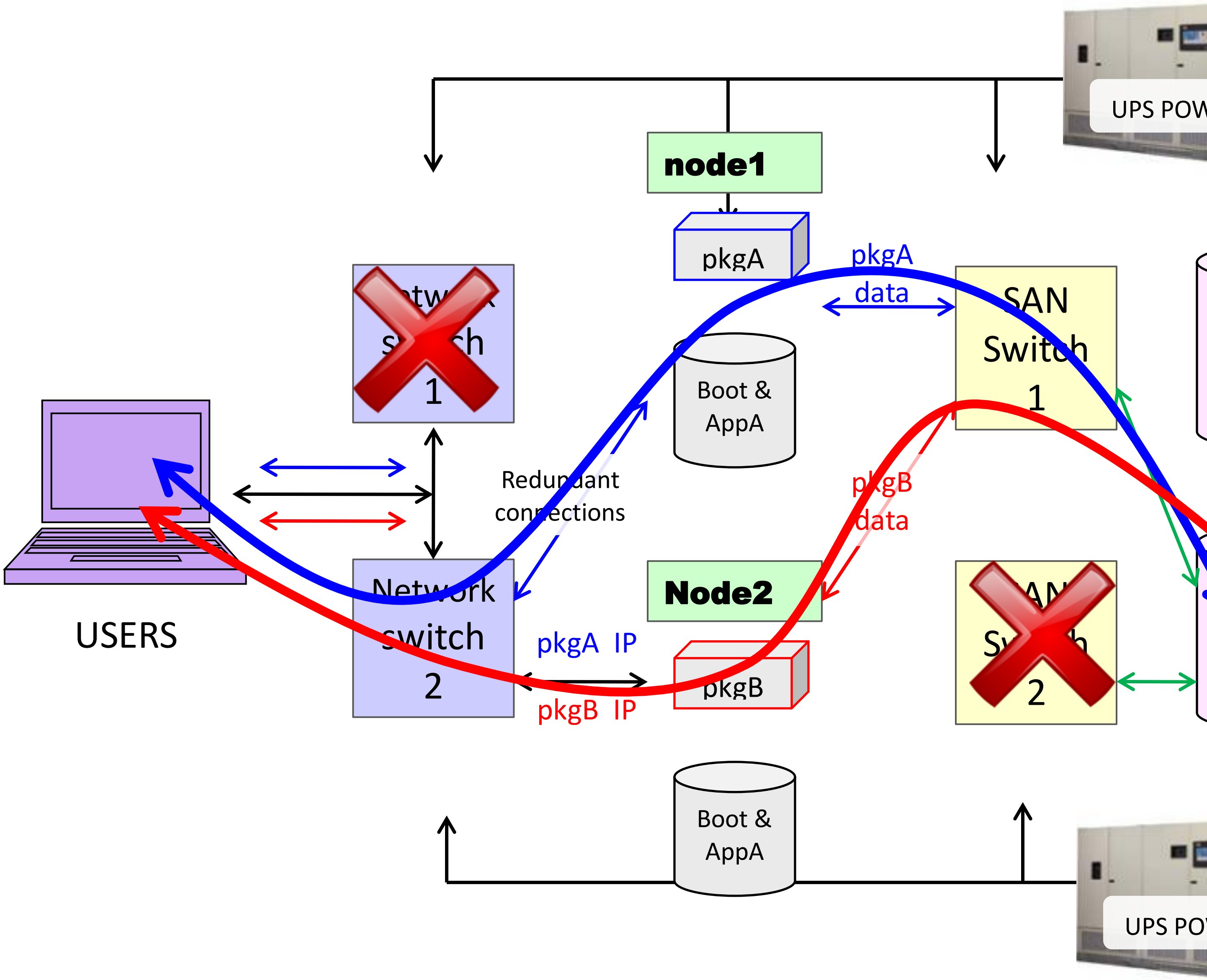


Figure 2: Continuous operations in spite of multiple network or SAN failures

Network and SAN path rerouting is automated by Serviceguard standby LAN failover or HPUX multi-pathing so that critical application operations continue unaffected. Since the packages run on their original nodes, users can still get to data.

The next figure shows the result of Serviceguard moving a package to an adoptive node in the event of a node failure.

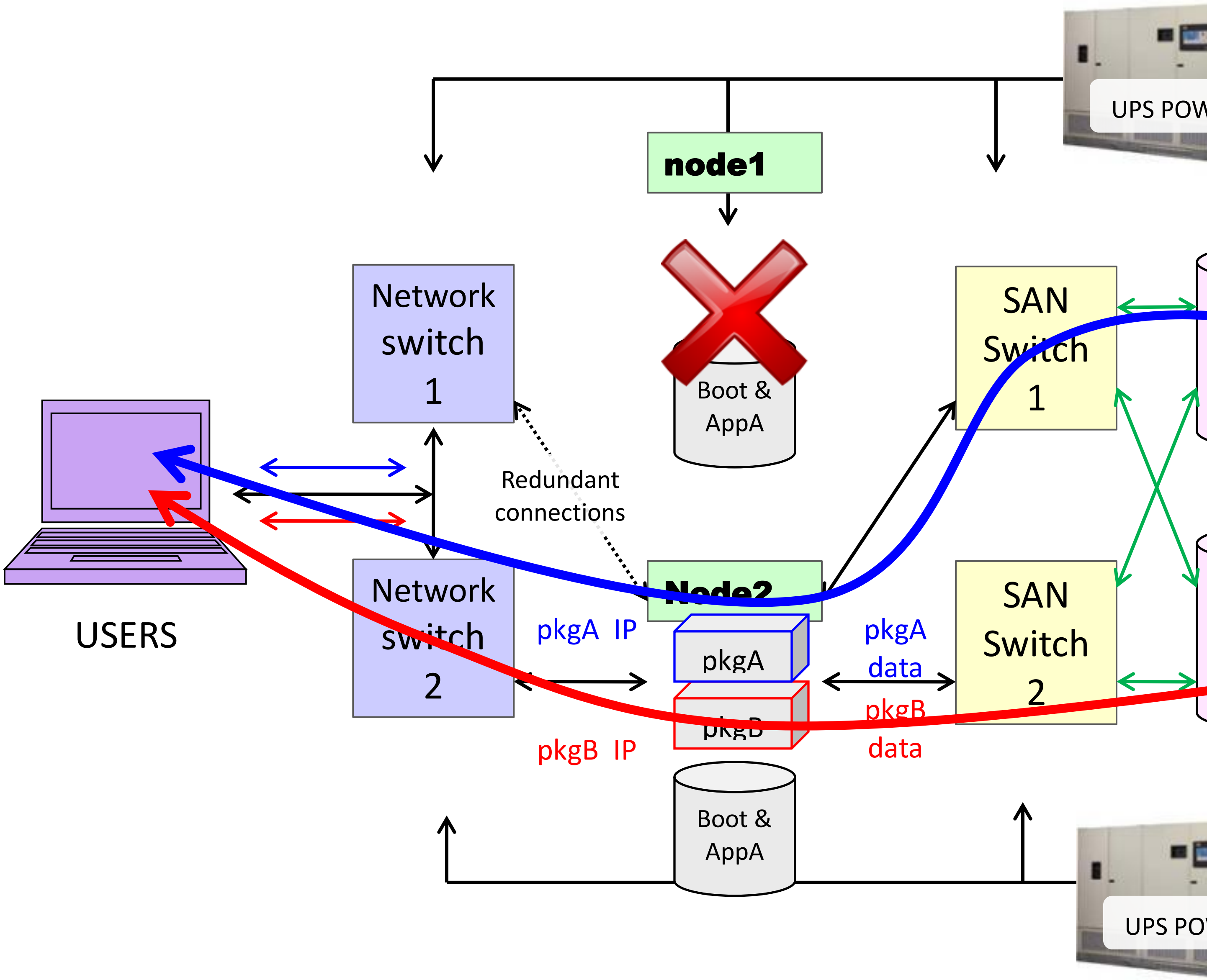


Figure 3: Reduced business outage due to redundant hardware and Serviceguard intervention

Though a node failure occurred, all other components continue to operate as expected. Package failover incurs an application outage for a period of time represented by the time it takes for Serviceguard to reform the cluster + identify an adoptive node that suits package startup + package startup time (including file system check/mount) + application startup time and data recovery if necessary.

A cluster configuration can exist in a single computer room, which is vulnerable to a site outage, but an **extended distance cluster** or **campus cluster** can deal with the effects of a site failure.

The following figure portrays the result of a complete site failure:

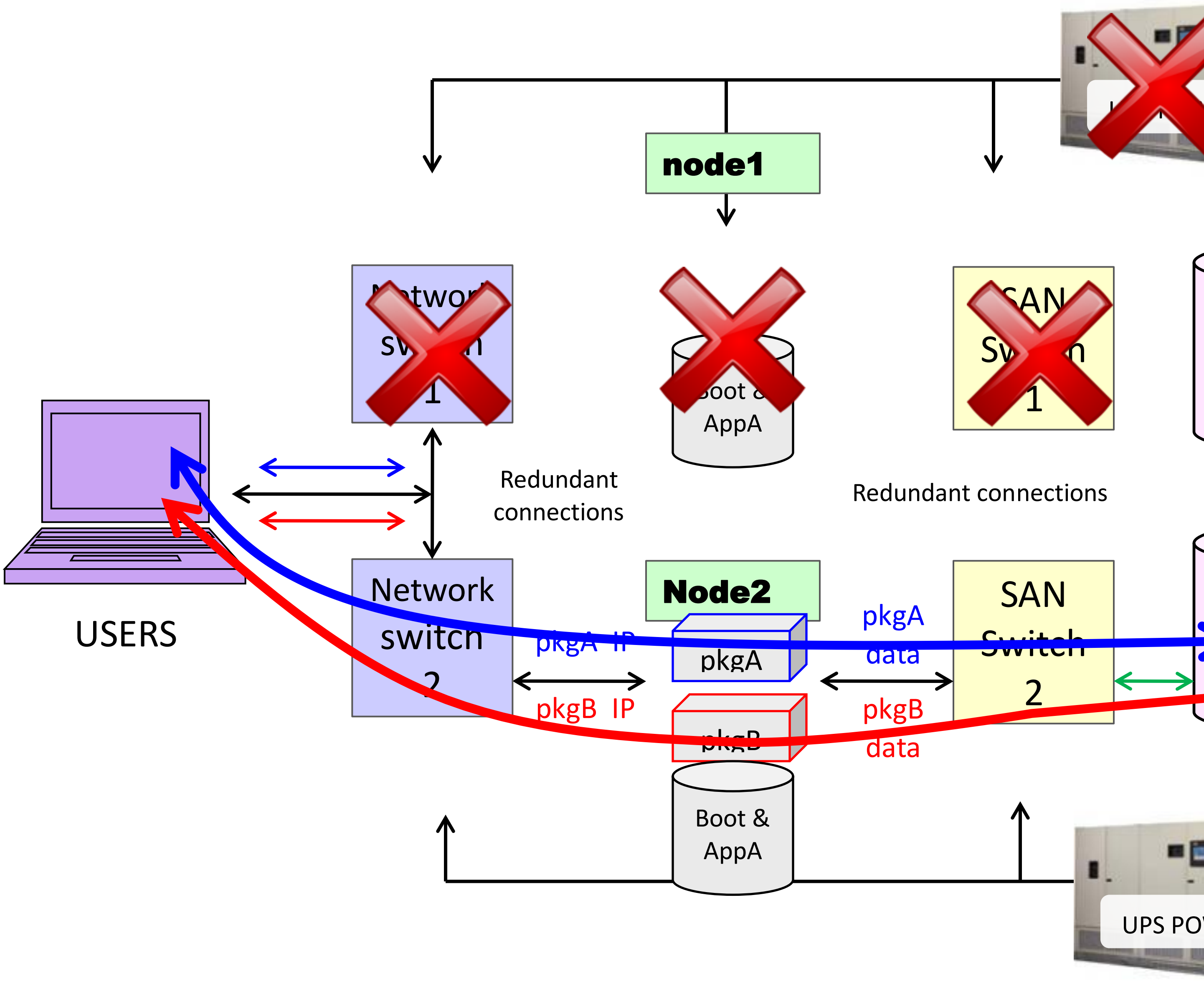


Figure 4: Site failure

Node failures are detected by Serviceguard (by loss of heartbeat), and cause a cluster reformation and package re-assignment to an adoptive node. Package failover stops the business solution for the period of time that it takes to detect node failure + reform the cluster + move the package to the adoptive node + fsck the data file systems + start the application on the failover node. Though not fault tolerant in all aspects, Serviceguard clusters are a cheaper solution than full fault tolerance systems such as the HP NonStop product line.

Serviceguard uses TCP/IP network services for reliable inter-node communication, including the transmission of heartbeat messages; periodic signals from each functioning node which are central to the operation of the cluster. TCP/IP services also are used for other types of inter-node communication. The network hardware should include redundant LAN interfaces on each node to permit redundant cluster heartbeat paths to increase cluster availability.

Summarization of the key features of a hardware cluster configuration:

- Redundant networking connectivity (standby LAN NIC or APA for each business IP).
- Redundant networking infrastructure (e.g. one redundant network using two switches and one dedicated network for inter-node communication).
- Redundant mass storage access (more than one SCSI or FC interface).
- Redundant mass storage infrastructure for data protection. (e.g. using hardware features of disk arrays (RAID) or software solutions like MirrorDisk/UX or Veritas Volume Manager).
- Redundant power supply (using UPS, more than one power circuit).
- Additionally use Event Monitoring Service (EMS), which lets you monitor and detect failures that are not directly handled by Serviceguard

Summarization of the key Features of a package configuration:

- Each package has exclusive access to system features such as LVM volume groups, VxVM disk groups, and one or more ‘ relocatable’ IP addresses.
- Each package starts a business application and monitor services. Any monitored service failure triggers package halt and failover.

For more examples of HA Hardware and Software Configurations see *Managing Serviceguard*: http://www.hp.com/go/hpux-serviceguard-docs.

Quorum Rules and Cluster Arbitration Device – Split-brain prevention

To insure packages have owners, Serviceguard transmits UDP heartbeat messages amongst all nodes periodically. If a node fails to transmit a heartbeat in the **NODE_TIMEOUT** (A.11.18 and earlier) or **MEMBER_TIMEOUT** (A.11.19 and later) window identified in the cluster configuration, the cluster will use the quorum rules to reform the cluster and find owners for packages that were operated by the missing-in-action node. To prevent Serviceguard from starting a failover package on a *second* node when all heartbeat traffic fails, a cluster arbitration device is required in a 2-node cluster and recommended in 3-16 node clusters. The arbitration function can take the form of a cluster lock VG, lock LUN or quorum server (on a system outside of the cluster).

Quorum Rules:

- Nodes that continue to exchange heartbeat messages in a **greater than 50%** subset of the previous cluster will reform a new cluster. The arbitration device will not be used. Example: If 2 nodes cannot exchange HB with a 3rd node, the 2 nodes will reform a cluster and adopt node 3 packages. Node 3 will automatically TOC (see next rule).

- Nodes that can only exchange heartbeat messages in a **less than 50%** subset of the previous cluster are forced to TOC to preserve data integrity, under the assumption that the packages will be managed by other nodes forming a majority. Example: If 1 node cannot exchange HB with 2 other nodes, the node will TOC.

- When **exactly 50%** of nodes cannot interact with the other half (a ***split-brain*** condition), they must negotiate with the arbitration device (cluster lock disk, lock LUN or Quorum Server) to decide which half forms the new cluster and which half TOCs. Example: If 1 node cannot exchange HB with its partner node, they both seek the arbitration device. The first node to attain abitration reforms the cluster and the late one performs a TOC.

- If no node can reach the cluster arbitration device, all nodes TOC.

Serviceguard commands

In order to understand explanations in the rest of the manual, it is helpful to have a Serviceguard command reference. The following list of Serviceguard commands are arranged by category and typical order of operation. Most Serviceguard commands can be run from any node in the cluster. NOTE: Some of the commands (or options) are new with Serviceguard A.11.20.

```
==============================================================================
=
========================                         CONFIGURATION        COMMANDS
==========================
==============================================================================
=

cmpreparecl - modifies system files as required for cluster creation. - (new
with A.11.20)
Usage: cmpreparecl [-n node_name ]... [-t]
Option meaning:
-n                  Hostname of server intended to be a member of the cluster
-t                  Preview (test) system file changes

______________________________________________________________________________
_
cmsetdsfgroup - Create and modify a cDSF (cluster device special file) group.
(new with A.11.20)
Usage: cmsetdsfgroup [-v] [-n node_name ] [-a [-n node_name ] ... ]
                     [-r [-f] [-n node_name ] ... ] [-v] [-f]
                     [[-n node_name ]...] [-c|-q]
Option meaning:
-n node_name        Specifies the name of a node to include in the cDSF group.
-a                  Adds nodes specified by -n node_name arguments to an
                    existing cDSF group.
-r                  Removes the nodes specified by -n node_name arguments from
                    an existing cDSF group.
-c                  Creates a cDSF group from the members of the existing local
                    Serviceguard cluster.
-f                  When used with -r and -n, forces the removal of the
                    specified nodes, even if those nodes are unreachable or
                    have an inconsistent configuration.
-q                  Returns info on the current cDSF group.

______________________________________________________________________________
_
cmpreparestg - Create or modify LVM or VxVM/CVM storage shared between
Serviceguard nodes.  (new with A.11.20)
Usage: cmpreparestg {-l vgname | -g dgname} [{-p pv_path...  |-P pvs_file}]
                    {-L [lvname] |-c lv_counts} [-m mountpoint]
                    [-o " option string " ... ] [-n node_name]...  [-t]
Option meaning:
-l vgname           Creates or modifies a LVM or a shared LVM (SLVM) VG group.
-g dgname           Creates or modifies a VxVM/CVM disk group.
-p pv_path          The device path names of a physical volume on the node
                    where the command is run.
```

```
-P filename       Alternatively, you can supply a list of physical volumes to
                  cmpreparestg in a file specified by  filename.
-L lvname         The name of the new logical volume created on a new or
                  existing VG or disk group.
-c lv_counts      The -c option is used to create multiple logical volumes on
                  a LVM VG or a VxVM/CVM disk group.
-m mountpoint     Specify a mount point for a new logical volume, or for
                  multiple new logical volumes specified via the -c option.
-o fs_opts=options_string
                  Use  -o fs_opts to specify options for new file systems
                  created via the  -m option.
-o lv_opts=options_string
                  Use   -o  lv_opts  to  specify  options  for  newly  created
logical
                  volumes.
-o vg_opts=options_string
                  Use this option to specify options for created VG.
-o dg_opts=options_string
                  Use this option to specify options for newly created
                  VxVM/CVM disk groups.
-n node_name      Perform the operation on a specified set of nodes.
-t                Test (preview) only.
```

```
_____________________________________________________________________________
_
cmdeploycl - automatically creates a cluster with up to 16 nodes.  (new with
A.11.20)
Usage: cmdeploycl [-t] [-s site ]... [-n node ]... [-N net_template ]
                  [-c clustername] [-q qs_host [qs_ip] | -L locklun] [-cfs]
       cmdeploycl [-t] [-s site ]... [-n node ]... [-N net_template ]
                  [-c clustername] [-b] -L vg:pv [-cfs]
       cmdeploycl [-t] [-s site ]... [-n node ]... [-N net_template ]
                  [-c clustername] [-L vg:pv ] [-cfs]
       cmdeploycl [-h]
Option meaning:
-c clustername    The name of the cluster that will be created.
-s site           Defines a logical site name where nodes are situated.
-n node           Specifies the node name to be included in the set of nodes
                  to query.
-N net_template   Specifies a file output by  cmquerycl -N, and edited if
                  necessary to configure additional subnets.
-q qs_host [qs_ip]   Identify hostname or IP of quorum server
-L Lock Device    Identify lock device: (locklun, vg, pv, vg:pv)
             -L   and -q are mutually exclusive.
-b                Builds storage shared by all the Serviceguard nodes.
-cfs              Must use to deploy a CFS cluster.
-t                Test (preview) only.
```

```
_____________________________________________________________________________
_
cmquerycl - query cluster or node configuration info
Usage:
   cmquerycl [-k] [-v] [-f format] [-l limit] [-w probe_type]
             [-h ipv4|ipv6] [-a ipv4|ipv6|any] [-c cluster_name]
             [-C cluster_ascii_file]
             [-q quorum_server [qs_ip2] | -L lock_lun_device|lock_vg:lock_pv]
             [-n node_name [-L lock_lun_device]]...

   cmquerycl [-v] -N network_template_file -n node_name ... [-f format]
             [-a ipv4|ipv6|any]
Option meaning:
-C cluster_ascii_file
                  Cluster configuration info will be saved in
                  cluster_ascii_file
-N network_template_file
                  Causes cmquerycl to discover connected LAN interfaces on
                  each node specified via the -n option, and to write that
                  info to network_template_file.
-c cluster_name   Cluster info will be queried from cluster cluster_name.
-f format         Select the output format to display.  (table/line)
-k                Eliminates some disk probing.  Does not return info about
                  potential cluster lock VGs and lock physical volumes.

The -k and -L lock_vg:lock_pv options are mutually exclusive.
-l limit          Limit the info included to type limit. (lvm/net)
-L lock_lun_device | lock_vg:lock_pv
                  Specifies the device to be used as the cluster lock
-n node_name      Specifies systems that should be included in the query.
-q quorum_server [qs_ip2]
                  Specify the hostname or IP address of the quorum server.
-v                Verbose output
-h address_family
                  Specifies the heartbeat IP address family. (ipv4/ipv6)
-a address_family
                  This flag specifies the Internet Protocol Version to which
                  Serviceguard will attempt to resolve cluster node names and
                  quorum server host names. (ipv4/ipv6/any/none/local/full)


______________________________________________________________________________
_
cmmakepkg - create a high availability package configuration file
Usage:
      cmmakepkg [-v (0 | 1 | 2)] [-f format] -l [[-m module_file_name]...]
      cmmakepkg [-v (0 | 1 | 2)] [-n package_name] [-i pkg_ascii_file]
                [-t file_name] [[-m module_file_name [-t file_name]]...]
                [output_file_name]
      cmmakepkg  [-v  (0  |  1  |  2)]  -u  pkg_ascii_file  [[-m  module_file_name]
...]
      cmmakepkg [-v] {-s | -p} [output_file_name]
Option meaning:
-v {0 | 1 | 2}    Specifies the verbosity level of the output.
                  0  (machine_parsable  mode)  1  (headline  mode)  2  (verbose
mode)
-n package_name   Configures the package name in the generated configuration
                  file.
-i pkg_ascii_file
                  Add    additional    modules    to    an    existing    package
configuration.
-m module_file_name...
```

```
                  Name of the package module files to include in the package
                  configuration file.
-u pkg_ascii_file
                  Upgrade a package to use the most recent version of the
                  modules.
-l                If used by itself, lists the available package module files
                  and their versions and brief descriptions of each module.
-f format         Select the output format to display. (line/table)
-p                Will be obsolete in a future release of Serviceguard.
-s                Will be obsolete in a future release of Serviceguard.
-t file_name      Use a toolkit configuration file in the POSIX shell format.
                  Requires -m option.
```

---

cmcheckconf - validate HA cluster configuration and/or package configuration files against current system configuration

```
Usage: cmcheckconf [-v] [-C cluster_ascii_file [-k | -K] ]
                   [[-p pkg_reference_file] | [-P pkg_ascii_file]...]
       cmcheckconf [-v] [-k | -K]
Option meaning:
-v                Verbose output will be displayed.
-k                Using the -k option means that cmcheckconf only checks disk
                  connectivity to the LVM VGs that are identified in the
                  cluster configuration file, and does not check VxVM disk
                  groups.
-K                Only check disk connectivity for cluster lock VGs.
-C cluster_ascii_file
                  Name of the cluster cluster configuration file to read.
-P pkg_ascii_file...
                  Name of the package configuration file(s) to read.
-p  pkg_reference_file
                  Name of the file containing a list of package configuration
                  file(s) to read.
```

---

cmapplyconf - perform cmcheckconf (behind the scenes) and then builds or updates and distributes the cluster binary file.

```
Usage: cmapplyconf [-f] [-v] [-k | -K] [-C cluster_ascii_file]
                   [[-p pkg_reference_file] | [-P pkg_ascii_file]...]
       cmapplyconf [-v] -N network_template_file
Option meaning: (uses the same options as cmcheckconf)
```

---

cmdeleteconf - Delete either the cluster or the package configuration

```
Usage: cmdeleteconf [-v] [-f] [-c cluster_name] [[-p package_name]...]
Option meaning:
-f                Force the deletion.
-v                Display verbose output.
-c cluster_name   Name of the (halted) cluster to delete.
-p package_name   Name of (halted) package to delete from the cluster.
```

```
_
cmmigratepkg - Migrate Serviceguard legacy Package to a Module Package.
Usage: cmmigratepkg -p <pkgname> [-x externalscript] [-e] [-s] -o
<outputfile>
Option meaning:
-p package_name
                  Name of an existing configured legacy package to convert.
-x external_script
                  Name of the external script file to create.
-e                Generate PEV's from non Serviceguard parameters.
-s                Comments out service attributes in the output_file.
-o output_file
                  Name of the output file for converted package
configuration.


=============================================================================
=
========================= CONTROL AND STATUS COMMANDS
======================
=============================================================================
=
cmruncl - run a high availability cluster
Usage: cmruncl [-f] [-v] [-n node_name...] [-t | -w none]
Option meaning:
-f                Force cluster startup without warning message and
                  continuation prompt that are printed with the -n option.
-v                Verbose output will be displayed.
-n node_name...   Start the cluster daemon on the specified subset of
node(s).
-t                Test only.
-w none           Disables network probing before starting cluster.(CAUTION)

_____________________________________________________________________________
_
cmrunnode - run a node in a high availability cluster
Usage: cmrunnode [-v] [node_name...] [-t | -w none]
Option meaning:
-v                Verbose output will be displayed.
node_name...      Start the cluster daemon on the specified node(s).
-t                Test only.
-w none           Disables network probing before starting cluster.(CAUTION)

_____________________________________________________________________________
_
cmrunpkg - run a high availability package
Usage: cmrunpkg [-v] [-t] [[-a] | [-n node_name]]... package_name...
       cmrunpkg [-v] [-m module_name] [-e exclude_module_name]
                [-n node_name] package_name
Option meaning:
-n node_name      Act on the specified set of nodes.
-a                Let Serviceguard select an eligible node from the package's
                  configured node list on which to start the failover
package.
-v                Verbose output will be displayed.
```

```
-t                  Test only.
-m  module_name     Modular packages: Partial package startup ends after the
                    identified module is completes.
-e  exclude_module_name
                    Name of the module that should be excluded from the package
                    run sequence.
```

```
______________________________________________________________________________
_
cmrunserv - run a service from the high availability package run script
Usage: cmrunserv [-v] service_name service_command_string
       cmrunserv [-v] [-r restarts] service_name service_command_string
       cmrunserv [-v] [-R] service_name service_command_string
Option meaning:
-v                  Verbose output will be displayed.
-r restarts         'restarts' indicates how many times the service may fail
                    before the package should be halted.
-R                  Restart the package service an unlimited number of times if
                    it fails.
service_name        Name of the service as it exists in the package
                    configuration info.
service_command_string
                    Process string to be started.
```

```
______________________________________________________________________________
_
cmhaltserv - halt a service from the high availability package halt script
Usage: cmhaltserv [-v] service_name
Option meaning:
-v   Verbose output will be displayed.
```

```
______________________________________________________________________________
_
cmhaltpkg - halt a high availability package
Usage: cmhaltpkg [-v] [-t] [-n node_name]...  package_name...
       cmhaltpkg [-v] [-s] [-n node_name]...  package_name...
Option meaning:
-n node_name        Acts only on a specific node.
-v                  Verbose output
-t                  Test only.
-s                  Symmetric halt is only for packages in maintenance mode.
```

```
______________________________________________________________________________
_
cmhaltnode - halt a node in a high availability cluster
Usage: cmhaltnode [-f | -d [ -t]] [-v] [-t] [node_name]...
Option meaning:
-f                    Force the node to halt even if packages or group members
are
                    running on it.
-d                       Halt the node and detach the running packages on the
halting
                    node.
-v                  Verbose output will be displayed.
-t                  Test only.
```

```
-d -t                Test only. Show a list of packages that will be detached if
                     the node is halted with -d option.
node_name...         The name of the node(s)to halt.

_____________________________________________________________________________
_
cmhaltcl - halt a high availability cluster
Usage: cmhaltcl [-f | -d [ -t]] [-v]
Option meaning:
-f                   Force the cluster to shutdown even if packages or group
                     members are currently running.
-v                   Verbose output will be displayed.
-d -t                Test only.

_____________________________________________________________________________
_
cmdisklock - manage Serviceguard cluster lock devices.
Usage: cmdisklock check path
       cmdisklock [-f] reset path
Option meaning:
check                Check and report current status of the cluster lock
reset                Reset (initialize) the state of the cluster lock device.

_____________________________________________________________________________
_
cmdo - Execute process on multiple remote Serviceguard nodes.
Usage: cmdo [[-n node_name]...] [-t timeout] command

_____________________________________________________________________________
_
cmexec - Execute process on a remote Serviceguard node.  (new with A.11.20)
Usage:  cmexec  node_name  [-o  outfile]  [-t  timeout]  {[command]  |  [-k
cmd_label]}
Option meaning:
-o outfile           Write output to the specified file instead of stdout.
-t timeout           Kill the executed command if it does not exit within the
                     specified timeout (in seconds).
node_name            Serviceguard node to execute command on.
cmd                  Command to execute on remote node. May be quoted.
-k cmd_label         Only supported for applications provided by HP

_____________________________________________________________________________
_
cmmodnet - add, remove or check whether an address address can be added to a
           subnet, or enable or disable a LAN interface in a high
           availability cluster
Usage: cmmodnet [-v] {-a | -r | -t}
                -i { IP_address subnet_name | IPv6_address IPv6_subnet }
       cmmodnet [-v] {-d | -e} LAN_name
Option meaning:
-v                   Verbose output will be displayed.
-a                   Add IPv4_address to its IPv4_subnet or IPv6_address to its
                     IPv6_subnet
                     from its IPv6_subnet
-t                   Check (test) only.
```

```
-i                  Required. IPv4_Address or IPv6_address
-d                  Disable LAN interface configured in the cluster.
-e                  Enable LAN interface configured in the cluster.
```

---

cmmodpkg - enable or disable switching attributes for a high availability package

```
Usage: cmmodpkg {-e[-t]|-d} [-n node_name]...  [-v] package_name...
       cmmodpkg [-v] [-n node_name] -R -s service_name package_name
       cmmodpkg [-v] [-m on [-n node_name] | off] package_name ...
Option meaning:
-e          Enables package switching.
-d          Disables package switching.
-n  node_name       Act on a specific node.
-R                  Resets the service restart counter to zero for the service
                    name specified by the -s option.
-s  service_name    Name of the package service whose restart counter is to be
                    reset.
-v                  Verbose output will be displayed.
-t                  Test only.
-m  on [-n node_name] | off
                    Enables/disables maintenance mode.
```

---

cmstartres - starts resource monitoring on the local node for an EMS resource that is configured in a Serviceguard package.

```
Usage: cmstartres [-v] [-u] -p package_name resource_name
Option meaning:
-v                  Verbose output will be displayed.
-u                  Wait for the EMS resource to be available before starting
                    resource monitoring.
-p   package_name    The name of the package where the EMS resource is
configured
                    in.  Required parameter.
resource_name       Name of the EMS resource to start resource monitoring for.
```

---

cmstopres - stops resource monitoring on the local node for an EMS resource that is configured in a Serviceguard package.

```
Usage: cmstopres [-v] -p package_name resource_name
Option meaning:
-v                  Verbose output will be displayed.
-p package_name     The name of the package in which the EMS resource is
                    configured.  Required parameter.
resource_name       Name of the EMS resource to stop resource monitoring.
```

---

cmvolmond - monitor LVM logical volumes and VxVM volumes for a high availability package. (new with A.11.20)

```
Usage: /usr/sbin/cmvolmond [-h|-v] [-O log_file] [-D log_level]
                           [-t poll_interval] volume_path...
```

```
Option meaning:
-h                  Displays the usage, as listed above, and exits.
-v                  Displays the monitor version and exits.
-O log_file         Specifies a file for logging (log messages are printed to
                    the console by default).
-D log_level        Specifies the log level.
-t poll_interval    Specifies the interval between volume probes.
volume_path         Full block device path to at least one VxVM volume or LVM
                    logical volume device file for monitoring.  Required.
```

```
_____________________________________________________________________________
_
cmvxserviced - monitor VxVM and CVM volumes for a high availability package.
Usage: cmvxserviced  [-h, --help] [-v, --version] [-O, --log-file <log_file>]
                     [-D, --log-level <1-7>] [-t, --poll-interval <seconds>]
                     <volume_path> [<volume_path>...]
Option meaning:
-h                  Displays the usage, as listed above, and exits.
-v                  Displays the monitor version and exits.
-O log_file         Log file (messages are sent to console by default)
-D log_level        Specifies the log level.
-t poll_interval    Specifies the interval between volume probes.
```

Informational Commands

```
_____________________________________________________________________________
_
cmeval - evaluate the impact of Serviceguard configuration and state changes
on the packages (new with A.11.20)
Usage: cmeval [-v] [-o output_file] input_file  (where input file is a
modified
                                                   cmviewcl-v-f_line output)
Option meaning:
-o output_file      The output file contains the list of affected packages and
                    is designed for simple machine parsing
input_file          The input file contains the output of 'cmviewcl -v -f line'
                    that has been modified for the configuration/state
                    changes that are to be evaluated.
-v                  Verbose output will be displayed.
```

```
_____________________________________________________________________________
_
cmversion - display SG version info
Usage: cmversion
```

```
_____________________________________________________________________________
_
cmviewcl - view info about a high availability cluster
Usage: cmviewcl [-v] [-f {table|line}] [-s config]
                [-l {package|cluster|node|group}]
                [-r {A.11.12|A.11.16|11.12|11.16}]
                [-c cluster_name]
                {[-n   node_name]...   |   [-p   package_name]...   |   [-S
site_name]...}
Option meaning:
-c cluster_name     Name of the cluster to view.
```

```
-f format          Select the output format to display.  (line/table)
-l type            Limit the type of data displayed. (group/node/package)
-n node_name       View info only about the specific node_name,
                   including info about the packages that are running
                   on these nodes.
-p package_name    View info only about the specific package_name(s).
-S site_name       View info only about the specific site_name(s), the
                   nodes assigned to the site(s), and relevant packages.
-r release         Make the tabular formatted output mimic that of the
                   specified release.
-s config          Displays only the static configuration info, not the
                   dynamic status info.
-v                 Verbose output will be displayed.


_____________________________________________________________________________
_
```

cmcompare - Compare files on multiple nodes

Usage: **cmcompare [-v] [-m|--match content|owner|perm|time] [[-n node_name]...]**
[file ...]

```
Option meaning:
-m content|owner|perm|time
                   Specifies a particular comparison to perform (content,
                   ownership, permissions, or modification timestamp).
-v                 Verbose output will be displayed.
-n node_name       Serviceguard node to compare file(s) on.
file               Path to file to be compared.


_____________________________________________________________________________
_
```

cmcp - Copy files between Serviceguard nodes

Usage: **cmcp [-rv] [src_node:]src_file [...] [dest_node:]dest_file**

```
Option meaning:
-r                 Recursively copy entire directories.
-v                 Verbose output will be displayed.
src_node           Serviceguard node to retrieve the file from. Defaults to
                   the local node.
src_file           Path to source file on src_node.
dest_node          Serviceguard node to copy file to. Defaults to the local
                   node.
dest_file          Path to copy src_file to on the dest_node.


_____________________________________________________________________________
_
```

cmsync - copy files and/or directories to multiple Serviceguard nodes.

Usage: **cmsync [-rv] [[-n node_name]...] [src_node:]src_file [...]**

```
Option meaning:
-r                 Recursively copy entire directories.
-v                 Verbose output will be displayed.
-n node_name       Serviceguard node to copy file to. If no -n options are
                   specified, default is to copy to all nodes within the
                   configured cluster.
src_node           Serviceguard node to retrieve the file from. Defaults to
                   the local node.
src_file           Path to source file on src_node.
```

```
_____________________________________________________________________________
_
cmquerystg - Displays info about the cluster DSFs.  (new with A.11.20)
Usage: cmquerystg -f format  [[-p path]...] [{[-l]| -d} -n node...]
Option meaning:
-f format         The output format ('line' is the only output format)
-p dsf_path | vg_path
                  Absolute path name of a device special file or a volume
                  group.
-l                Limits the output to display only info about the
                  VGs on the specified nodes.
-n                Limits the display to the devices visible on the specified
                  nodes.
-d                Limits the output to display only info about
                  persistent device special files on the specified nodes.

_____________________________________________________________________________
_
cmgetconf - Get cluster or package configuration info
Usage:
      cmgetconf [-v {0 | 1 | 2}] [[-K] -c cluster_name] [-p package_name]
      [output_filename]
      cmgetconf [-v {0 | 1 | 2}] [[-K] -c cluster_name] [-P dirname]
      cmgetconf [-v {0 | 1 | 2}] [-K]
Option meaning:
-v 0|1|2  Verbose output will be displayed.
-c cluster_name   Name of the cluster for which to query cluster info.
-K                 Causes the probing of VGs to be skipped, reducing the time
for
                  the command to complete.
-p package_name   Name of an existing package for which to query package
                  info.
-P                Write all package's configuration info to a specified
                  output_filename. Mutually exclusive to the -p option.
output_filename   If -c and/or -p options are specified, the name of the
                  file into which cmgetconf will copy cluster or package
                  configuration info.

_____________________________________________________________________________
_
cmgetpkgenv - get the configured environment for a high availability package
Usage: cmgetpkgenv package_name

=============================================================================
=
========================                          DIAGNOSTIC        COMMANDS
===============================
=============================================================================
=
cmscancl - gather system config info from nodes with Serviceguard installed.
Usage: cmscancl [-n node ...] [-s| -o output_file ]
Note: cmquerycl can be used as a diagnostic command
Option meaning:
-n node_name...   Specify the node(s) to be scanned.
```

```
-o  output_file   Write configuration info to a specified output file.
-s                Display the configuration info to the  screen only.
```

--- end of command list ---

Preparation to build a cluster

Version of Serviceguard installed

Depending in the information sought; use one of the following methods:

```
# cmversion
A.11.19.00

# what /usr/lbin/cmcld | grep A.11
         A.11.19.00 Date: 03/17/11 Patch: PHSS_41902

# swlist | grep guard ; swlist -l product | grep guard
  B5140BA                  A.11.31.06     Serviceguard NFS Toolkit
  T2775DB                      A.03.00          HP Serviceguard Cluster File
System
  PHSS_41902               1.0            Serviceguard A.11.19.00
  -snipped-
```

Network configuration

Serviceguard commands use networking ports identified with the ‘hacl’ string in /etc/services and /etc/inetd.conf file. Commands are dependent on *identd*, called out in **/etc/inetd.conf**, to authenticate Serviceguard commands. Before attempting to run Serviceguard commands, insure the following:

- TO avoid Serviceguard command failure if DNS fails, configure **/etc/nsswitch.conf** with this line:

  ```
  hosts: files dns
  ```
- Each node’s **/etc/hosts** must list every Ethernet IP assigned to any node in the cluster and those IPs must be aliased to the simple hostname of the host node.
- Create **/etc/cmcluster/cmclnodelist**, and populate it like .rhosts, listing every cluster member. Example:

  ```
  node1 root
  ```

Check whether Serviceguard network ports:

```
# netstat -a | grep hacl
tcp        0      0  *.hacl-probe          *.*                          LISTEN
tcp        0      0  *.hacl-cfg            *.*                          LISTEN
udp        0      0  *.hacl-cfg            *.*
```

Check for clusters

The `cmquerycl` command can be used to either search for existing Serviceguard clusters in the network or create a cluster configuration ASCII file. This basic cmquerycl command may take some time to complete.

```
# cmquerycl

Cluster Name   Node Name
UNUSED
               rxh17u07

rxh17u09_cluster
               rxh17u09
```

LVM preparation

When a volume group is created, /etc/lvmtab (or for version 2.X volume groups, /etc/lvmtab_p) is loaded with references to the volume group and related physical devices. Every node that will run a package must have an lvmtab file that will allow activation of a package VG. After creating a volume group on one node, import it into the other adoptive nodes. *Note that unless cDSFs are specified, the device special files representing a shared LUN may differ across systems.* Use this example (Node2 was where the vg was created) to update /etc/lvmtab* on adoptive nodes:

1. On Node2, create a map files.

```
# for vg in `strings /etc/lvmtab | grep -v -E 'disk|dsk' | sed 's_/dev/__g'`
  do
  vgexport -pvs -m /etc/lvmconf/$vg.map /dev/$vg
  done
```

2. Copy the map file to nodeC:

```
root@Node2:/ # rcp / etc/lvmconf/vgspare.map nodeC:/etc/lvmconf/vgspare.map
```

3. On nodeC, create the volume group directory:

```
root@nodeC:/ # mkdir /dev/vgspare
```

4. Still on nodeC create a control file named *group* in the directory /dev/vgspare, as follows:

```
root@nodeC:/ # mknod /dev/vgspare/group c 64 0xhh0000
```

`hh`: Use the same minor number as on *Node2* as used on nodeC when possible. Use the following command to display a list of used minor numbers:

```
root@nodeC:/ # ls -l /dev/*/group
```

5. Import the volume group data using the map file from node Node2. On node nodeC, enter:

```
root@nodeC:/ # vgimport -s -m /etc/lvmconf/vgspare.map /dev/vgspare
```

Note that the disk device names on nodeC may be different from their names on Node2. Make sure the physical volume names are correct throughout the cluster. When the volume group can be activated on this node, perform a `vgcfgbackup`. (This backup will be available in the unlikely event that a `vgcfgrestore` must be performed on this node because of a disaster on the primary node and an LVM problem with the volume group.) Do this as shown in the example below:

```
# vgchange -a y /dev/vgspare
# vgcfgbackup /dev/vgspare
# vgchange -a n /dev/vgspare
```

NOTE: When LVM mirroring is used, consider creating a `/etc/lvmpvg` file on each node to ensure that each mirror is created on a different array, if the cluster has such a configuration. Note that the lvmpvg on each node will contain different device special files if lvmtab differs between nodes. See the SAW document titled “`HPUX Serviceguard - Using /etc/lvmpvg to put mirror data on different arrays`”

6. Make sure that you have deactivated the volume group on Node2. Then enable the volume group on nodeC:

```
root@nodeC:/ # vgchange -a y /dev/vgspare
```

7. Create a directory to mount the disk:

```
root@nodeC:/ # mkdir /spare1
```

8. Mount and verify the volume group on nodeC:

```
# mount /dev/vgsparee/lvspare1 /spare1
```

9. Unmount the volume group on *nodeC*:

```
# umount /spare1
```

10. Deactivate the volume group on *nodeC*:

```
# vgchange -a n /dev/vgspare
```

Other cluster requirements are itemized in the Managing Serviceguard manual that applies to the version installed. The manual is located at this website: http://www.hp.com/go/hpux-SG-docs

Fast deployment of a cluster – cmpreparecl, cmdeploycl

To insure the files are configured properly, use cmpreparecl (a feature of A.11.18 and later):

```
# cmpreparecl [-n node_name ]...

Example:
# cmpreparecl -n rxh17u09 -n rxh17u07
Running cmpreparecl on nodes rxh17u09 rxh17u07
Saving command output to /var/adm/cmcluster/sgeasy/easy_deployment.log
Running cmpreparecl on nodes rxh17u09 rxh17u07
Configuring cmclnodelist on rxh17u09.
Changing cmclnodelist permissions to 644 on node rxh17u09
Commenting out line
 'auth         stream tcp6 wait   bin  /usr/lbin/identd   identd '
 ininetd.conf on node rxh17u09.
Adding
 auth stream tcp6 wait bin /usr/lbin/identd identd -t120

 to inetd.conf on node rxh17u09.
Generating an updated inetd.conf for rxh17u09.
Updating /etc/hosts on rxh17u09 to contain all cluster node IP addresses.
Finalizing update of /etc/hosts on node rxh17u09.
Finalizing /etc/nsswitch.conf file on node rxh17u09
Configuring cmclnodelist on rxh17u07.
Changing cmclnodelist permissions to 644 on node rxh17u07
Commenting out line
 'auth         stream tcp6 wait   bin  /usr/lbin/identd   identd '
 ininetd.conf on node rxh17u07.
Adding
 auth stream tcp6 wait bin /usr/lbin/identd identd -t120

 to inetd.conf on node rxh17u07.
Generating an updated inetd.conf for rxh17u07.
Updating /etc/hosts on rxh17u07 to contain all cluster node IP addresses.
Finalizing update of /etc/hosts on node rxh17u07.
cmcp: Failed to get attributes for /etc/nsswitch.conf on rxh17u07
Adding
 'host: files dns'
```

```
entry to /etc/nsswitch.conf on node rxh17u07.
Finalizing /etc/nsswitch.conf file on node rxh17u07
```

cmdeploycl

After cmpreparecl succeeds, use cmdeploycl to quickly build a cluster.
Example (building a one-node cluster):

```
# cmdeploycl -n rxh17u07
Running cmdeploycl on nodes rxh17u07
Saving subcommand output to /var/adm/cmcluster/sgeasy/easy_deployment.log
Running cmdeploycl on nodes rxh17u07
Calling cmpreparecl (/usr/sbin/cmpreparecl -n rxh17u07  )
cmpreparecl may take a while to complete ...
Command cmpreparecl succeeded
Calling cmquerycl (/usr/sbin/cmquerycl -v -w full -n rxh17u07   -C
/etc/cmcluster/sgeasy/cluster.ascii.generated)
cmquerycl may take a while to complete ...
Command cmquerycl succeeded
Calling cmapplyconf (/usr/sbin/cmapplyconf -v -C
/etc/cmcluster/sgeasy/cluster.ascii )
cmapplyconf may take a while to complete ...
Command cmapplyconf succeeded
Calling cmruncl (/usr/sbin/cmruncl -v -w none )
cmruncl may take a while to complete ...
Command cmruncl succeeded

CLUSTER            STATUS
rxh17u07_cluster   up

  NODE           STATUS       STATE
  rxh17u07       up           running

    Network_Parameters:
    INTERFACE    STATUS                       PATH                NAME
    PRIMARY      up                           0/4/1/0/6/0         lan3
    STANDBY      up                           0/1/2/1             lan1
Cluster configuration saved to /etc/cmcluster/sgeasy/cluster.ascii
```

Manual method to create a cluster - cmquerycl

The manual method to create a cluster begins with cmquerycl.   This command should be run as the first step in preparing for cluster configuration. It may also be used as a trouble-shooting tool to identify the current configuration of a cluster, since it prints out an overview of its discovery results Example:

```
# cmquerycl -v -C /etc/cmcluster/cluster.ascii -n Node1 -n Node2

Looking for other clusters ... Done
Gathering storage information
Found 21 devices on node Node1
Found 25 devices on node Node2
Analysis of 46 devices should take approximately 6 seconds
0%----10%----20%----30%----40%----50%----60%----70%----80%----90%----100%
Found 3 volume groups on node Node1
Found 2 volume groups on node Node2
Analysis of 5 volume groups should take approximately 1 seconds
0%----10%----20%----30%----40%----50%----60%----70%----80%----90%----100%
Note: Disks were discovered which are not in use by either LVM or VxVM.
      Use pvcreate(1M) to initialize a disk for LVM or,
      use vxdiskadm(1M) to initialize a disk for VxVM.
Gathering network information
Beginning network probing
Completed network probing

Node Names:    Node1
               Node2

Bridged networks (local node information only - full probing was not
performed):

...

IP subnets:

IPv4:
...

IPv6:

Possible Heartbeat IPs:
...

Possible Cluster Lock Devices:
...
LVM volume groups:


LVM physical volumes:
...
LVM logical volumes:
...
Writing cluster data to /tmp/cluster.ascii.
```

The above cmquerycl command presupposes that the cluster arbitration device is a lock VG that is shared by both nodes and is registered in /etc/lvmtab on both nodes and that both nodes have NICs assigned IPs on the same subnet. The resulting output is `cluster.ascii`, which contains cluster-specific configuration information.

Required parameters:

```
CLUSTER_NAME                        rxh17u09_cluster
HOSTNAME_ADDRESS_FAMILY         IPV4
MEMBER_TIMEOUT          14000000                  <- default
AUTO_START_TIMEOUT      600000000                 <- default
NETWORK_POLLING_INTERVAL        2000000           <- default
NETWORK_FAILURE_DETECTION               INOUT     <- default
NETWORK_AUTO_FAILBACK           YES               <- default
MAX_CONFIGURED_PACKAGES         300               <- default
```

```
The following section is declared for each node:
NODE_NAME                <hostname>
  NETWORK_INTERFACE lan3
    HEARTBEAT_IP         <IP>
  NETWORK_INTERFACE lan1
```

Optional parameters:

```
FIRST_CLUSTER_LOCK_VG
  Or
CLUSTER_LOCK_LUN /dev/dsk/c1t2d3s1 (identified per node)
  Or
QS_HOST <qs_host>
QS_ADDR <qs_addr>
QS_POLLING_INTERVAL 120000000   <- default value if used
QS_TIMEOUT_EXTENSION 2000000    <- default value if used
```

The following 2 parameters are specified in each node section:

```
CAPACITY_NAME specifies a name for the capacity.
CAPACITY_VALUE specifies a value for the CAPACITY_NAME that precedes
```

Multiple declarations of the following 3 parameters is legal:

```
SUBNET <IP>
  IP_MONITOR ON
  POLLING_TARGET <IP>     <- default IP is gateway
```

WEIGHT parameters are associated with packages, and are partnered with node CAPACITY parameters. A maximum of 4 weights may be defined.

```
WEIGHT_NAME -
WEIGHT_DEFAULT - specifies a default weight for this WEIGHT_NAME.


USER_NAME  <ANY_USER | <actual_user>  (never root)
USER_HOST  ANY_SERVICEGUARD_NODE | CLUSTER_MEMBER_NODE | <hostname>
USER_ROLE  <FULL_ADMIN | MONITOR | PACKAGE_ADMIN >
```

Any number of the `VOLUME_GROUP` parameter may be declared (up to kernel limits)

At a minimum, customize the CLUSTER_NAME parameter in the file.

Validating a Cluster Configuration - cmcheckconf

NOTE: cmapplyconf performs cmcheckconf. If you plan on distributing the cluster binary file now, skip cmcheckconf and run cmapplyconf.

Use the `cmcheckconf` command to check a high availability cluster configuration and/or package configuration files and determine whether cmapplyconf will be successful. This validates the cluster configuration as specified by the cluster ASCII file and/or the package configuration files specified by each package ASCII file in the command and identifies whether the cmapplyconf can be performed (if the cluster is currently running).

If the cluster has already been configured previously, the `cmcheckconf` command will compare the configuration in the cluster ASCII file against the previously configuration information stored in the binary configuration file and validates the changes. The same rules apply to the package ASCII file.

```
root@Node1:/ # cmcheckconf -k -v -C /etc/cmcluster/cluster.ascii
```

This checks the following:

- Network addresses and connections.
- Cluster lock or lock LUN connectivity (if you are configuring a lock disk) or quorum server communication.
- Validity of configuration parameters for the cluster and packages.
- Uniqueness of names.
- Existence and permission of scripts specified in the command line.
- Verify specified nodes have the same heartbeat subnets.
- If you specify the wrong configuration filename.
- If all nodes can be accessed.
- No more than one CLUSTER_NAME, HEARTBEAT_INTERVAL, and AUTO_START_TIMEOUT are specified.
- The value for package run and halt script timeouts is less than 4294 seconds.
- Legitimate value of MEMBER_TIMEOUT.
- The value for AUTO_START_TIMEOUT variables is >=0.
- Heartbeat network minimum requirement. The cluster must have one heartbeat LAN configured with a standby, two heartbeat LANs, one heartbeat LAN and an RS232 connection, or one heartbeat network with no local LAN switch, but with a primary LAN that is configured as a link aggregate of at least two interfaces.
- At least one NODE_NAME is specified.
- Each node is connected to each heartbeat network.
- All heartbeat networks are of the same type of LAN.
- The network interface device files specified are valid LAN device files.
- If a serial (RS-232) heartbeat is configured, there are no more than two nodes in the cluster, and no more than one serial (RS232) port connection per node.
- VOLUME_GROUP entries are not currently marked as cluster-aware.
- There is only one heartbeat subnet configured if you are using CVM 3.5 disk storage.

If the cluster is online, the check also verifies that all the conditions for the specific change in configuration have been met.

Using the -k option means that cmcheckconf only checks disk connectivity to the LVM disks that are identified in the ASCII file. Omitting the -k option (the default behavior) causes cmcheckconf to test the connectivity of all LVM disks on all nodes. Using -k can result in significantly faster operation of the command.

```
root@Node1:/etc/cmcluster# cmcheckconf -k -v -C ./cluster.ascii -P \ ./sw/sw-
pkg.conf
Checking cluster file: ./cluster.ascii
Note : a NODE_TIMEOUT value of 2000000 was found in line 134. This value
is recommended if the top priority is to reform the cluster as fast
as possible in case of failure. If the top priority is to minimize
reformations, consider using a higher setting. For more information see
the cluster configuration ASCII file or the Managing Serviceguard manual.
Checking nodes ... Done
Checking existing configuration ... Done
Gathering storage information
Found 3 devices on node Node1
Found 3 devices on node Node2
Analysis of 6 devices should take approximately 1 seconds
0%----10%----20%----30%----40%----50%----60%----70%----80%----90%----100%
Found 1 volume groups on node Node1
Found 1 volume groups on node Node2
Analysis of 2 volume groups should take approximately 1 seconds
0%----10%----20%----30%----40%----50%----60%----70%----80%----90%----100%
Gathering network information
Beginning network probing (this may take a while)
Completed network probing
Parsing package file: ./sw/sw.conf.
Checking for inconsistencies
Modifying configuration on node Node1
Modifying configuration on node Node2
cmcheckconf: Verification completed with no errors found.
Use the cmapplyconf command to apply the configuration.
```

Generation and Distribution of a Cluster Configuration

Use the `cmapplyconf` command to validate and apply Serviceguard cluster configuration and package configuration files. cmapplyconf performs another cmcheckconf, creates or updates the binary configuration file called `/etc/cmcluster/cmclconfig`, and distributes it to all nodes. This binary configuration file contains the cluster configuration information as well as package configuration information for all packages specified. Cmapplyconf also installs the cluster lock structure on the lock VG (if defined) and it installs a cluster ID and the exclusive activation mode bit in LVM metadata on disks identified with VOLUME_GROUP parameters in the cluster ASCII file. It also removes the cluster ID and exclusive activation bit on volume groups that have been removed from the configuration file.

If changes to either the cluster configuration or to any of the package configuration files are needed, first update the appropriate ASCII file(s) (cluster or package), then validate the changes using the cmcheckconf command and then use cmapplyconf again to verify and redistribute the binary file to all nodes.

The cluster ASCII file only needs to be specified if configuring the cluster for the first time, or if adding or deleting nodes to the cluster. The package ASCII file only needs to be specified if the package is being added, or if the package configuration is being modified. It is recommended that the user runs the `cmgetconf` command to get either the cluster ASCII configuration file or package ASCII configuration file whenever changes to the existing configuration are required.

Before distributing the configuration, ensure that your security files permit copying among the cluster nodes. (For security file details see “Preparing Your Systems” in the *Managing Serviceguard*, Chapter 5 *Building an HA Cluster Configuration*.

Use the following steps to generate the binary configuration file and distribute the configuration to all nodes in the cluster:

- Activate the cluster lock volume group so that the lock disk can be initialized:

  ```
  # vgchange -a y /dev/vgspare
  ```

- Generate the binary configuration file and distribute it:

  ```
  # cmapplyconf -k -v -C /etc/cmcluster/cluster.ascii
  ```

  or

  ```
  # cmapplyconf -k -v -C /etc/cmcluster/cluster.ascii
  ```

- Deactivate the cluster lock volume group.

  ```
  # vgchange -a n /dev/vgspare
  ```

The `cmapplyconf` command creates a binary version of the cluster configuration file and distributes it to all nodes in the cluster. This action ensures that the contents of the file are consistent across all nodes. Note that the `cmapplyconf` command does not distribute the ASCII configuration file.

NOTE The apply will not complete unless the cluster lock volume group is activated on *only one* node before applying. There is one exception to this rule: a cluster lock had been previously configured on the same physical volume and volume group. After the configuration is applied, the cluster lock volume group must be deactivated.

Adding packages to a cluster

The value in Serviceguard comes from the automating the operation (and recovery) of software business solutions. As stated earlier, there are two styles of packages; legacy and modular. Package should not share. To start creating a package, do the following:

```
# cd /etc/cmcluster
# mkdir <pkg_name>
# cd <pkg_name>
```

To create a **modular** package, determine which modules should be incorporated in the package. If the package will be a standard failover package that does not incorporate any extra toolkits, use the following command to create a modular package configuration file:

```
# cmmakepkg –n <pkg_name> <pkg_name.config>
```

Edit the <pkg_name.config> file, uncommenting those parameters that are required to activate system resources. Additionally, it may be necessary to incorporate external_script references to the configuration file. The external script is used to start and stop applications. The Managing Serviceguard manual and a SAW document explain how to implement external scripts.

To create a **legacy** package, create the package configuration and control templates. File names are not consequential.

```
# cmmakepkg –p <pkg_name.config>
# cmmakepkg –s <pkg_name.cntl>
```

Edit the <pkg_name.config> file. Locate the parameters identified in upper case, and customize as appropriate.
Edit `<pkg_name.cntl>` and customize uppercase parameters as appropriate.
Copy the package directory to the adoptive nodes in the cluster.

After customizing the package file(s), add the package to the cluster binary:

```
# cmapplyconf –f -P <pkg_name.config>
```

Note: cmapplyconf can include both cluster ASCII and package ASCII files and build the cluster entirely with one command.

Starting the cluster

After cluster configuration is completed, the `cmruncl` command causes all nodes or specified nodes in the cluster to start their cluster daemons and form a cluster.

```
root@Node1:/# cmruncl -v
cmruncl: Validating network configuration...
cmruncl: Network validation complete
Waiting for cluster to form ..... done
Cluster successfully formed.
Check the syslog files on all nodes in the cluster to verify that no warnings
occurred during startup.
```

To start a cluster on a subset of nodes, use the `–n` option of `cmruncl`. Notice the important warning in the following example. If the –f option is not supplied, you must confirm the action (as in the example).

```
root@Node1:/# cmruncl -n Node1 -f
cmruncl: Validating network configuration...
cmruncl: Network validation complete

WARNING:
Performing this task overrides the data integrity protection normally
provided by Serviceguard.  You must be certain that no package applications
or resources are running on the other nodes in the cluster:
       Node2

To ensure this, these nodes should be rebooted (i.e. /usr/sbin/shutdown -r)
before proceeding.

Are you sure you want to continue (y/[n])? y
```

```
Waiting for cluster to form .... done
Cluster successfully formed.
Check the syslog files on all nodes in the cluster to verify that no warnings
occurred during startup.
```

When the node forms or joins a cluster, the cluster binary file is read into memory and used to govern cluster operations and the each active node starts a group of Serviceguard deamons. Example:

```
      /usr/lbin/cmcld -m -n rxh17u09 -n rxh17u07
      /usr/lbin/cmfileassistd
      /usr/lbin/cmlogd
      /usr/lbin/cmlvmd
      /usr/lbin/cmnetd
      /usr/lbin/cmresourced
      /usr/lbin/cmserviced
      /usr/lbin/cmclconfd -c
```

Starting the cluster automatically starts packages unless the package is configured with AUTO_RUN = NO.

View Cluster Status

The current cluster status, fixed configuration values or dynamic status values can be inspected using various forms of `cmviewcl`. Example output:

```
# cmviewcl -v


CLUSTER        STATUS
swrecovery       up

  NODE           STATUS       STATE
  Node1       up           running

    Cluster_Lock_LVM:
    VOLUME_GROUP          PHYSICAL_VOLUME       STATUS
    /dev/vgspare          /dev/dsk/c5t8d0       down

    Network_Parameters:
    INTERFACE    STATUS       PATH                NAME
    PRIMARY      up           0/1/2/0             lan0
    STANDBY      up           0/1/2/1             lan1

    PACKAGE        STATUS       STATE        AUTO_RUN     NODE
    sw-pkg         up           running      enabled      Node1

      Policy_Parameters:
      POLICY_NAME     CONFIGURED_VALUE
      Failover        configured_node
      Failback        manual

      Script_Parameters:
      ITEM       STATUS   MAX_RESTARTS  RESTARTS   NAME
      Service    unknown  0             0          time_not
      Node_Switching_Parameters:
      NODE_TYPE    STATUS       SWITCHING    NAME
      Primary      up           enabled      Node1 (current)
      Alternate    up           disabled     Node2

  NODE           STATUS       STATE
  Node2        up           running

    Cluster_Lock_LVM:
    VOLUME_GROUP          PHYSICAL_VOLUME       STATUS
    /dev/vgspare          /dev/dsk/c2t8d0       down

    Network_Parameters:
    INTERFACE    STATUS       PATH                NAME
    PRIMARY      up           0/1/2/0             lan0
    STANDBY      up           0/1/2/1             lan1
```

For Informations about the different states of a cluster, a node, or a package, please refer to *Managing Serviceguard*, Chapter 8: *Cluster and Package Maintenance*. The document is downloadable from http://www.hp.com/go/hpux-SG-docs

Halting the Cluster

To halt a Serviceguard cluster use the command `cmhaltcl`. This causes all nodes in the cluster to stop their cluster daemons, halting any packages and related applications in the process unless those packages are detached. This command will halt all the daemons on all currently running systems.

```
root@Node1:/# cmhaltcl -v
Disabling all packages from starting on nodes to be halted.
Disabling all packages from running on Node1.
Disabling all packages from running on Node2.
Package sw-pkg is already disabled on node Node2
Warning:  Do not modify or enable packages until the halt operation is
completed.
This operation may take some time.
Waiting for nodes to halt .... done
Successfully halted all nodes specified.
Halt operation complete.
```

The `-f` option forces cluster shutdown even if packages are running.

```
root@Node1:/# cmhaltcl -f
Disabling all packages from starting on nodes to be halted.
Warning:  Do not modify or enable packages until the halt operation is
completed.
Disabling automatic failover for failover packages to be halted.
Halting package sw-pkg
Successfully halted package sw-pkg
This operation may take some time.
Waiting for nodes to halt .... done
```

```
Successfully halted all nodes specified.
Halt operation complete.
```

If the user only wants to shutdown a subset of daemons, the `cmhaltnode` command should be used instead.

Joining a node to a running cluster

If a node is not running Serviceguard, and it's sister nodes are, it can be joined to the cluster using `cmrunnode`.

Example:

```
root@Node1:/# cmrunnode Node2

cmrunnode: Validating network configuration...
cmrunnode: Network validation complete
Waiting for nodes to join ..... done
Cluster successfully formed.
Check the syslog files on all nodes in the cluster to verify that no warnings
occurred during startup.
```

Starting a node will not cause any active packages to move to the node unless **FAILBACK_POLICY** is configured to **AUTOMATIC** in the package. However, if a package is down, has its `AUTO_RUN` and `Node_switching` enabled for that node, that package will automatically start on the joining node.

`Cmrunnode` causes a node to join a running cluster. The command will time out after the `AUTO_START_TIMEOUT` (configured in the cluster ASCII file - default 10 minutes) window expires. If `AUTOSTART_CMCLD=1` in /etc/rc.config.d/cmcluster, the node will perform a cmrunnode when it enters run level 3. If <u>all</u> nodes are executing cmrunnode within the `AUTO_START_TIMEOUT` window, a cluster will form.

Halting Serviceguard on a node

`cmhaltnode` causes a node to halt its packages and cluster daemons and remove itself from the cluster. Cmhaltnode can force packages to adoptive nodes.

```
root@Node1:/# cmhaltnode -v Node2       # halting the other node
Disabling all packages from starting on nodes to be halted.
Disabling all packages from running on Node2.
Warning:  Do not modify or enable packages until the halt operation is
completed.
Waiting for nodes to halt .... done
Successfully halted all nodes specified.
Halt operation complete.
```

If no node name is specified, the cluster daemon running on the local node will be halted and removed from the existing cluster.

Starting Packages

Ordinarily when a cluster starts, the packages will start on their **primary** configured nodes. You may need to start a package manually after it has been halted manually using the `cmrunpkg` command. This command may be run on any node within the cluster and may operate on any package within the cluster. If a node is not specified, the node on which the command is run will be used. This will result in an error if the current node is unable to run the package or is not listed as a possible owner of the package. When a package is started on a new node, the package's run script is executed with argument *start*.

```
root@Node1:/# cmrunpkg -v sw-pkg
Running package sw-pkg on node Node1
Successfully started package sw-pkg on node Node1
cmrunpkg: All specified packages are running
```

If a package startup results failure and NO_RESTART status, inspect the package log, resolve the failure cause and re-enable the node_switching and AUTO_RUN settings using cmmodpkg (see below).

Stopping Packages

To halt a high availability package use the command `cmhaltpkg`. This performs a manual halt of high availability package(s) running on Serviceguard clusters. The command may be run on any node within the cluster and may operate on any package within the cluster.

```
root@Node1:/# cmhaltpkg sw-pkg
Disabling  automatic failover for failover packages to be halted.
Halting package sw-pkg
Successfully halted package sw-pkg
One or more packages or package instances have been halted. These packages
have AUTO_RUN disabled and no new instance can start automatically. To allow
automatic start, enable AUTO_RUN via cmmodpkg -e <package_name>
cmhaltpkg: Completed successfully on all packages specified
```

Modifying Package Switching Attributes

Note the AUTO_RUN and switching values in the cmviewcl report:

```
    PACKAGE        STATUS       STATE        AUTO_RUN     NODE
    sw-pkg         up           running      enabled      Node1

--snipped---

      Node_Switching_Parameters:
      NODE_TYPE    STATUS       SWITCHING    NAME
      Primary      up           enabled      Node1  (current)
      Alternate    up           disabled     Node2
```

AUTO_RUN status identifies whether a package can run on any adoptive node.
Node_Switching SWITCHING status identifies whether a package is permitted to run on a specific node.

To enable or disable switching attributes, use the `cmmodpkg` command . The important option s for this command are –e (enable) and –d (disable).

```
Sample commands:
# cmmodpkg -e pkg1                   Enable a pkg to run on any adoptive node.
# cmmodpkg -d pkg1                   Disable a package from  running on any node.
# cmmodpkg -n Node1 -e pkg1          Enable a particular node to run a package.
# cmmodpkg -n Node1 -d pkg1          Prevent a particular node from runing a package.
```

Reconfiguring a Cluster

You can reconfigure a cluster either when it is halted or while it is still running. However, on older versions of Serviceguard, some operations can only be done when the cluster is halted. The following is consolidated from **Managing Serviceguard** editions covering A.11.16-A.11.20, however page references in the table pertain to edition 18.

Table: Online/Offline cluster configuration changes

VERSION KEY: 6 = A.11.16 7 = A.11.17 8 = A.11.18 9 = A.11.19 & A.11.20

CLUSTER PARAMETER	MODIFICATION	VERSION	CHANGE ONLINE
CLUSTER_NAME	Change	ALL	NO - The cluster must be deleted and recreated with a new cluster name.
HOSTNAME_ADDRESS_FAMILY	IPv4/ANY	9	YES
CLUSTER_LOCK_LUN	Change	9 8	YES NO - Cluster must be down
QS_HOST QS_ADDR QS_POLLING_INTERVAL QS_TIMEOUT_EXTENSION	Change	9 6,7,8	YES unless SG-CFS-pkg is running. (pg 326) NO - Cluster must be down !updated 2010.1.7
FIRST_CLUSTER_LOCK_VG FIRST_CLUSTER_LOCK_PV	Change Change	9 9 8 6,7	YES YES PROVISIONAL - see pg 361 NO
NODE_NAME	Add	ALL	YES - Member nodes must be running

	Delete	ALL	YES - even if departing node is gone.
	Change	ALL	NO - remove/re-add node with new name
NETWORK_INTERFACE	Add/Delete	8,9	YES - with qualifications
	Change from IPV4 <-> IPv6 or vice versa	9	YES
		7,8	NO
HEARTBEAT_IP	Change IP	9	PROVISIONAL - Delete and re-add
	Redesignate as STATIONARY_IP or vice versa	8,9	YES - with qualifications
		6,7	NO
Standby NIC	Add/Delete	8,9	YES - with qualifications
CAPACITY_NAME CAPACITY_VALUE	Add/Delete	9	YES - will trigger warning if the change will cause a pkg to fail.
HEARTBEAT_INTERVAL	Change	8,9	YES - except in CVM env.
		6,7	NO
NODE_TIMEOUT	Change	8	YES - except in CVM env.
		6,7	NO
MEMBER_TIMEOUT	Change	9	YES except when using CVM/CFS
AUTO_START_TIMEOUT	Change	9,8	YES except when using CVM/CFS
		6,7	NO
NETWORK_POLLING_INTERVAL	Change	8,9	YES
		6,7	NO
CONFIGURED_IO_TIMEOUT_EXTENSION	N	9	YES (definition pg 158)
NETWORK_FAILURE_DETECTION	Change	8,9	YES
NETWORK_AUTO_FAILBACK	Change	9	YES
SUBNET	Change	9	YES
IP_MONITOR	Change	9	YES
POLLING_TARGET	Change	9	YES
MAX_CONFIGURED_PACKAGES	Change	9-Jul	UNNECESSARY - leave at max!
		6	NO (affects memory usage)
WEIGHT_NAME	Change	9	YES , requires matching CAPACITY_NAME

WEIGHT_DEFAULT	Change	9	YES
USER_NAME USER_HOST USER_ROLE	Change, add remove	ALL	YES
VOLUME_GROUP	Add/Delete	ALL	YES

Updating the arbitration device (cluster lock VG/PV, LUN or Quorum Server)

Use the procedures that follow whenever you need to change the device file names of the cluster lock physical volumes – for example, when you are migrating cluster nodes to the agile addressing scheme available as of HP-UX 11i v3.

1. Locate or reconstitute the cluster configuration file using:
   ```
   $ cmgetconf cluster.ASCII
   ```
2. Run cmcheckconf to check the existing configuration:
   ```
   $ cmcheckconf cluster.ASCII
   ```
3. Modify the arbitration values or change from one type of arbitration to another (lock VG/PV to quorum server etc) in the ASCII file.
4. Run cmapplyconf to apply the configuration. (May require cluster halt):
   `$ cmapplyconf –f –C cluster.ASCII` (–f bypasses the modification query)

Reconfiguring a Halted Cluster

All versions of Serviceguard can be reconfigured when the cluster is halted. Depending on the version of Serviceguard in use, it may be necessary to halt the cluster in order to perform the cmapplyconf. Perform cmcheckconf on a modified cluster configuration ASCII file to learn whether it is necessary to halt the cluster (or see above table).

Using `-k` or `-K` option with the `cmcheckconf` and `cmapplyconf` commands can significantly reduce the response time.

Deleting Nodes from the Configuration While the Cluster is Running

Whether Serviceguard is running or not, it will allow the user to remove a node from the cluster if that node is no longer reachable (via hacl-cfg ports). Comment all NODE_NAME references for the missing node from both package and cluster configuration files, then perform cmapplyconf on all of them. If the departing node is still attached to the subnet, deactivate hacl-cfg in /etc/inetd.conf on the departing node and do ‘inetd –k ; inetd’ to restart inetd services.

NOTE Running cmapplyconf on a node that you are trying to remove from the cluster will generate an error message.

Use the following procedure to delete a node with HP-UX commands. In this example, nodes Node1, Node2 and nodeC are already configured in a running cluster named cluster1, and you are deleting node nodeC.

1. Use the following command to store a current copy of the existing cluster configuration in a temporary file:

   ```
   root@Node1:/ # cmgetconf -c cluster1 temp.ascii
   ```

2. Specify the new set of nodes to be configured (omitting nodeC) and generate a template of the new configuration:

   ```
   root@Node1:/ # cmquerycl -C cluster.ascii -c cluster1 -n Node1 –n Node2
   ```

3. Edit the file cluster.ascii to check the information about the nodes that remain in the cluster.

4. Halt the node you are going to remove (nodeC in this example):

   ```
   root@Node1:/ # cmhaltnode -f -v nodeC
   ```

5. Verify the new configuration:

   ```
   root@Node1:/ # cmcheckconf -C cluster.ascii
   ```

6. From Node1 or Node2 apply the changes to the configuration and send the new binary configuration file to all cluster nodes.:

   ```
   root@Node1:/ # cmapplyconf -C cluster.ascii
   ```

NOTE If you are attempting to remove an unreachable node that has many packages dependent on it, especially if the dependent packages use a large number of EMS resources, you may see the following message:

```
The configuration change is too large to process while the cluster is
running. Split the configuration change into multiple requests or halt the
cluster. In this situation, you must halt the cluster to remove the node.
```

Some older versions of Serviceguard do not permit the admin to change any network-related parameters while the cluster is running. A workaround to this is to remove the node from the cluster and re-add it with new network parameters and configurations (assuming the subnets match the other nodes in the cluster). In order to do this however, the node must also be removed from package configurations.

Changing the LVM Configuration While the Cluster is Running

On newer versions of Serviceguard, you can change the cluster lock volume group or physical volume configuration while the cluster is running.

If you are removing a volume group from the cluster configuration, make sure that you also modify or delete any legacy package control script or modular package configuration file that activates and deactivates this volume group. In addition, you should use the LVM vgexport command on the removed volume group from each node that will no longer be using the volume group.

Using Serviceguard Commands to Change the LVM Configuration While the Cluster is Running

From the LVM's cluster, follow these steps:

1. Use the cmgetconf command to store a copy of the cluster's existing cluster configuration in a temporary file. For example: `cmgetconf cluster.ascii`
2. Edit the file `clconfig.ascii` to add or delete volume groups.
3. Use the cmcheckconf command to verify the new configuration.
4. Use the cmapplyconf command to apply the changes to the configuration and send the new configuration file to all cluster nodes.

NOTE If the volume group that you are deleting from the cluster is currently activated by a package, the configuration will be changed but the deletion will not take effect until the package is halted; thereafter, the package will no longer be able to run without further modification, such as removing the volume group from the package control script.

Reconfiguring a Package

The cluster can be either halted or running during package reconfiguration. The types of changes that can be made and the times when they take effect depend on whether the package is running or not. If you reconfigure a package while it is running, it is possible that the package could fail later, even if the `cmapplyconf` succeeded. For example, consider a package with two volume groups. When this package started, it activated both volume groups. While the package is running, you could change its configuration to list only one of the volume groups, and `cmapplyconf` would succeed. If you issue `cmhaltpkg` command, however, the halt would fail. The modified package would not deactivate both of the volume groups that it had activated at startup, because it would only see the one volume group in its current configuration file.

Reconfiguring a Package on a Running Cluster

You can reconfigure a package while the cluster is running, and in some cases you can reconfigure the package while the package itself is running. You can do this in Serviceguard Manager, or use Serviceguard commands.

To modify the package with Serviceguard commands, use the following procedure (pkg1 is used as an example):

1. Halt the package if necessary:

```
root@Node1:/ # cmhaltpkg sw-pkg
```

2. If it is not already available, you can obtain a copy of the package's ASCII configuration file by using the cmgetconf command, specifying the package name.

   ```
   root@Node1:/ # cmgetconf -p sw-pkg sw-pkg.ascii
   ```

3. Edit the ASCII package configuration file.

4. Verify your changes as follows:

   ```
   root@Node1:/ # cmcheckconf -v -P sw-pkg.ascii
   ```

5. Distribute your changes to all nodes:

   ```
   root@Node1:/ # cmapplyconf -v -P sw-pkg.ascii
   ```

6. Copy the package control script to all nodes that can run the package.

Reconfiguring a Package on a Halted Cluster

You can also make permanent changes in package configuration while the cluster is not running. Use the same steps as mentioned above.

Adding a Package on a Running Cluster

You can create a new package and add it to the cluster configuration while the cluster is up and while other packages are running. The number of packages you can add is subject to the value of *Maximum Configured Packages* in the cluster configuration file. To create a package configuration template see section “Create configuration templates” above.

For example, to use Serviceguard commands to verify the configuration of newly created *pkg1* on a running cluster:

```
root@Node1:/ # cmcheckconf -P /etc/cmcluster/sw-pkg/sw-pkg.conf
```

Use a command such as the following to distribute the new package configuration to all nodes in the cluster:

```
root@Node1:/ # cmapplyconf -P /etc/cmcluster/sw-pkg/sw-pkg.conf
```

Remember to copy the control script to the /etc/cmcluster/pkg1 directory on all nodes that can run the package.

Deleting a Package on a Running Cluster

Serviceguard will not allow you to delete a package if any other package is dependent on it. To check for dependencies, use the cmviewcl -v –l package command. System multi-node packages cannot be deleted froma running cluster.

The following example halts the failover package *mypkg* and removes the package configuration from the cluster:

```
root@Node1:/# cmhaltpkg sw-pkg

root@Node1:/# cmdeleteconf -p sw-pkg
```

The command prompts for a verification before deleting the files unless you use the -f option. The directory /etc/cmcluster/mypkg is not deleted by this command.

Allowable Package States During Reconfiguration

All nodes in the cluster must be powered up and accessible when making configuration changes. Refer to Table below to determine whether or not the package may be running while you implement a particular kind of change. Note that for all of the following cases the cluster may be running, and also packages other than the one being reconfigured may be running.

Table: Types of Changes to Packages

PACKAGE-SPECIFIC CHANGES

(modular package configuration uses lower case parameters)

PACKAGE PARAMETER	MODIFICATION	VERSION	CHANGE ONLINE
PACKAGE_NAME	Add	ALL	YES
	Delete	ALL	YES - pkg must be halted
	Change	ALL	NO - must delete/recreate pkg
PACKAGE_TYPE	Change	6,7	NO - Only FAILOVER is user-definable
		8	NO - MULTI_NODE now allowed
		9	YES - MULTI_NODE now allowed
NODE_NAME	ADD/REMOVE	ALL	YES - pkg must be halted
	SWAP	9	YES - pkg may be running
AUTO_RUN	OFF/ON	ALL	YES
NODE_FAIL_FAST_ENABLED	YES/NO	ALL	YES
RUN_SCRIPT	Change	ALL	NO
HALT_SCRIPT	Change	ALL	NO
RUN_SCRIPT_TIMEOUT	Change	ALL	YES
HALT_SCRIPT_TIMEOUT	Change	ALL	YES

SUCCESSOR_HALT_TIMEOUT	Change	7-9	YES
SCRIPT_LOG_FILE	Change	7-9	NO - pkg must be halted
FAILOVER_POLICY	Change	ALL	YES
FAILBACK_POLICY	Change	ALL	YES
PRIORITY	Change	7-9	YES
DEPENDENCY_NAME \ DEPENDENCY_CONDITION } DEPENDENCY_LOCATION /	Add	7-9	YES, if FAILOVER_POLICY != MIN_PACKAGE_NODE
weight_name	Change	9	YES, requires matching CAPACITY and WEIGHT parameters in the cluster configuration
weight_value	Change	9	YES, must not exceed CAPACITY or pkg will halt
LOCAL_LAN_FAILOVER_ALLOWED (aka NET_SWITCHING_ENABLED)	Change	ALL	YES
MONITORED_SUBNET	Add/Mod/Delete	8	NO - pkg must be halted
(or SUBNET)	Add/Delete	9	YES - subnet must exist on nodes
	Change from IPv4<->IPv6	9	Change from YES
MONITORED_SUBNET_ACCESS	Add/Mod/Delete	8	NO
		9	YES
CLUSTER_INTERCONNECT_SUBNET	Add/Mod/Delete	8,9	NO - for multi-node SGeRAC pkgs only
ip_subnet ip_subnet_node ip_address	Add/Mod/Delete	9	YES - subnet must exist on nodes

SERVICE_NAME SERVICE_FAIL_FAST_ENABLED SERVICE_HALT_TIMEOUT	Add/Delete	6-9	NO (legacy pkg format only)
service_name service_cmd service_restart service_fail_fast_enabled service_halt_timeoutk	Add/Delete	8,9	YES (modular package format only)
RESOURCE_NAME RESOURCE_POLLING_INTERVAL \ RESOURCE_START / RESOURCE_UP_VALUE /	Add/Delete	6-8	NO - halt pkg first. YES - change must not cause pkg to halt NO - DEFERRED resources require pkg halt
	Change	6-7	NO
	Change	9	YES - unless resource_up_value would cause package to fail.
STORAGE_GROUP	Add/Delete	6-8	NO
USER_NAME USER_HOST USER_ROLE	Add/Mod/Delete	7-9	YES

Other changes unique to modular package configuration files
CAUTION: In general it is safe to add system resources to a package during package operation, but unsafe to modify or remove them, as attempting to do so may fail due to business of the resource, causing the package to halt
Documentation warns about this possibility.

PACKAGE PARAMETER	**MODIFICATION**	**VERSION**	**CHANGE ONLINE**
script_log_file	Modify	8,9	NO
log_level	Modify	8,9	YES
vg	Add	8,9	YES
cvm_dg	Add	8,9	YES
vxvm_dg	Add	8,9	YES

fs_name	Add	8,9	YES
fs_directory	Add	8,9	YES
fs_type	Add	8,9	YES
fs_mount_opt	Add	8,9	YES
fs_umount_opt	Add	8,9	YES
fs_umount_opt	Modify/Remove	8,9	YES
fs_fsck_opt	Add/Delete	8,9	YES
pev_	Add/Delete	8,9	YES
external_pre_script	Add/Delete	8,9	YES
external_script	Add/Delete	8,9	YES

Deleting or modifying the following parameters may cause the package to halt if the dependent application is still running.

vg	Delete	8,9	NO - may halt application
cvm_dg	Delete	8,9	NO
vxvm_dg	Delete	8,9	NO
fs_name	Modify/Remove	8,9	NO
fs_directory	Modify/Remove	8,9	NO
fs_type	Modify/Remove	8,9	NO
fs_mount_opt	Modify/Remove	8,9	NO

Troubleshooting Serviceguard Problems

This chapter will cover the following sections to assist you in troubleshooting various kinds of Serviceguard Problems:

- General Troubleshooting Commands
- Logfiles, increase logging level, debug different Serviceguard Commands and Daemons
- Solving Problems like Package control script hangs or Node and Network Failures and others
- Tools

Please keep in mind: Since your cluster is unique, there are no cookbook solutions to all possible problems. But if you apply these checks and commands and work your way through the log files, you will be successful in identifying and solving problems. If not, do not hesitate to contact your HP Support.

General Troubleshooting Commands

cmcheckconf

cmcheckconf can be used to troubleshoot your cluster just as it was used to verify the configuration. The following example shows the commands used to verify the existing cluster configuration on Node1 and Node2:

```
# cmquerycl -v -C /etc/cmcluster/verify.ascii -n Node1 –n Node2
# cmcheckconf -v -C /etc/cmcluster/verify.ascii
```

The `cmcheckconf` command checks:

- The network addresses and connections.
- The cluster lock disk connectivity.
- The validity of configuration parameters of the cluster and packages for: The uniqueness of names; the existence and permission of scripts.
- It does not check: The correct setup of the power circuits; the correctness of the package configuration script.

cmscancl

cmscancl is a diagnostic script that saves it's output file in /tmp/scancl.out. It displays information about all the nodes in a cluster in a structured report that allows you to compare such items as IP addresses or subnets, physical volume names for disks, and other node-specific items for all nodes in the cluster.

cmscancl needs a root-enabled .rhosts file on all nodes to scan all nodes successfully. Without it, the command can only collect information on the local node.
The following are the types of configuration data that cmscancl displays for each node:

Table: Data Displayed by the `cmscancl` **Command**

Description	**Source of Data**
LAN device configuration and status	`lanscan` command
network status and interfaces	`netstat` command
file systems	`mount` command
LVM configuration	`/etc/lvmtab` file
LVM physical volume group data	`/etc/lvmpvg` file
link level connectivity for all links	`linkloop` command
binary configuration file	`cmviewconf` command

cmviewconf/cmviewcl –v –f line

`cmviewconf` (preA.11.18) `and cmviewcl -v -f line` (A.11.18 and newer) allows you to examine the cluster binary file, even when the cluster is not running.

Serviceguard Logs

- Serviceguard daemons log their actions to /var/adm/syslog/syslog.log. If nothing is found in this file, check if syslogd still works as expected ('logger testing syslogd').
- Package start and stop actions are logged in the package log which is typically located in the package directory or the /var/adm/cmcluster/logs/ (modular packages). Modular packages offer the **script_log_file** parameter to identify a different path/file.
- Unless DSAU is configured (see next item), it may be necessary to inspect all system's syslog.log and/or package logs to understand the actual events preceding a problem.
- If Distributed Systems Administration Utilities (DSAU) is configured to consolidate system logs, the syslogs may be located on another system. Otherwise, cluster level events are saved locally in the normal location.
- DSAU offers following functions: Configuration synchronization, Log consolidation, Command fan-out
- With consolidated logging, you can examine a single log that contains entries from all systems (e.g. Cluster Nodes) in your configuration.

For more details please check out the document: *Managing Systems and Workgroups* (A Guide for HP-UX System administrators, http://docs.hp.com -> choose OS -> Choose System Administration). Link as in July 2007: http://www.docs.hp.com/en/B2355-90950/B2355-90950.pdf.

Debug logging

Package debug logging

Package start/stop issues are captured in the package log. Legacy packages log their operations to a log file in the same directory as the package control script. Modular packages have a parameter that allows the admin to identify the destination of the package log. Typically, it will be either the package directory or /var/adm/cmcluster/logs/.

Modular packages have adjustable log levels:
Level 0 : user visible informative messages
Level 1 : slightly more detail user visible informative messages
Level 2 : messages provide logic flow
Level 3 : messages provide detailed data structure information
Level 4 : debugging information that provides detailed data
Level 5 : function call flow

To change the log level, edit the package configuration file and set the log_level parameter, then run cmapplyconf on the file, then start/stop the package and check the package log.

Legacy package control scripts do not offer a log_level feature, so if the package log does not contain sufficient detail, insert a '**set –x**' near the top of the package control script, and test the package startup or shutdown to capture specific actions in the package log to identify package run/halt failures. (No cmapplyconf is necessary).

Cluster-level logging

With the advent of a new cluster manager engine in A.11.18, Serviceguard daemons and commands offer different debug logging mechanisms. The following is taken from this WTEC page:
http://teams3.sharepoint.hp.com/teams/esssupport/InsideESSSupport/InsideWTEC/HAProducts/Pages/sg_debug_logging.aspx

Debug logging of cmcld (SG A.11.19 and later)

From SG A.11.19 onwards you can start logging for cmcld by adding the following lines in the /etc/cmcluster.conf and do a kill -SIGHUP <pid>.

```
CMCLD_LOG_FILE=/var/adm/cmcluster/cmcld.log
CMCLD_LOG_LEVEL=6
```

You can also enable debug logging for a specific module only by setting in the same file:

CMCLD_*{module mnemonic}*_LOG_LEVEL=6

Ex: For the Utility Daemon use CMCLD_UTL_LOG_LEVEL

where *{module mnemonic}* is one of the following:

ATN = Auto Trans
CDB = Configuration Database
CLM = Cluster Management
CMD = Commands
CML = SAF Distributed Lock Service
CMP = Config/Status Proxy
CMS = SAF Cluster Membership Service
CNF = Configuration
COM = Communication Server
CSV = Command Server
DFA = Disk File Access
DLK = Disk Lock
DLM = Distributed Lock Management
EVT = Cluster Management Event
GMD = Group Membership Server
GMS = Group Membership Service
LOC = Local Communication
LVM = Logical Volume Management
MDV = Module Version
MSG = Messaging
NET = Network Management
PKG = Package Management
PLE = Placement Engine
QSM = Quorum Service
REM = Remote Communication
RES = EMS Resources

SDB = Status Database
SEC = Security Service
SES = Sessions
SRV = Service Management
STA = Status Database API
SYN = Synchronization
UNK = Unknown
UTD = Utility Daemon

Serviceguard A.11.18 and older versions use the internal function cl_log() to log messages, warnings, notices and errors. A Serviceguard message is classified by the module that issued the message, by the reason (category) the message is being logged, and by the level of detail of information.

These are the different logging categories:

CAT	SG internal category	syslog(3C) Level
INT	LOG_INTERNAL	LOG_INFO
EXT PER	LOG_EXTERNAL LOG_PERIODIC	LOG_NOTICE LOG_INFO
XER	LOG_EXT_ERROR	LOG_ERR
ERR	LOG_INT_ERROR	LOG_ERR
DTH	LOG_DEATH	LOG_EMERGTRC
TRC	LOG_TRACE	LOG_INFO

Serviceguard can be divided into several modules. Each message has its origin in one of the following modules.

MOD	SG Module
CLM	Cluster management
PKG	Package management
NET	Network interface
SRV	Service monitoring
LOC	Local communications
REM	Remote communications
SDB	Status database
SYN	Synchronization
CSV	Command server
COM	Communications server
DLM	Distributed lock manager
GMS	OPS group membership service
LVM	Shared LVM
UTL	General support
CDB	Configuration database

STA	Status database API
DEV	Storage devices
ATS	Shared tape device
QSM	Quorum Devices

The cmsetlog command (obsolete with SG A.11.19)
The cmsetlog command enables users to obtain a more verbose output of cmcld. This is extremely useful if a problem should be reproduced. Cmsetlog allows to set the log level and to restrict logging to specific categories and modules. Cmsetlog is used to enable and disable debug logging. cmsetlog only works if the Serviceguard cluster is already running and can therefore not be used to obtain debug logs if the cluster startup fails. Refer to the '-T' option below.

Warning: `cmsetlog` should only be used for debugging purposes and all debug logging should be disabled once the debugging has been finished.

`cmsetlog` is located in the `/usr/contrib/bin` directory, to stress that: this program is basically an unsupported tool that should be used only by HP Support or by customer on request by an HP Support Engineer.

Enable cmcld debug logging with cmsetlog (obsolete with SG A.11.19)
`cmsetlog` can enable the debug logging for `cmcld` only on the local node. Note that `cmsetlog` operates only on the local daemon. It must be run on each node you wish to change logging on.

Note: With high log levels greater 3, the log files can **grow very fast** and can **fill up** the filesystem. Use `cmsetlog -h` to display usage information.

Here are various examples that show how cmsetlog can be used to obtain debug logging:

```
# cmsetlog 5
```

Log cmcld debug information to syslog.log at log level 5 for all categories (except PER (frequent actions)) and modules.

```
# cmsetlog -f /tmp/SG.log
```

Redirect the Serviceguard cmcld logging in the file /tmp/SG.log. No further logs go into syslog.log. This is recommended for the verbose log levels greater 3.

```
# cmsetlog -M NET -M REM 6
```

Use the most verbose log level for cmcld, but restrict logging to the modules 'network interface' and 'remote communications'.

```
# cmsetlog -M NET -C PER -C ERR -C XER -C INT -C EXT -C DTH -C TRC 6
```

The most complete log that is possible.

```
# cmsetlog -C PER -C ERR -C XER -C INT -C EXT -C DTH -C TRC 6
```

Disable cmcld debug logging with cmsetlog (obsolete with SG A.11.19)
The debug logging is automatically stopped and reset to default once the cluster halted.

To reset the debug logging to default modules, categories and loglevel on a running cluster, simply use the command

```
# cmsetlog -r
```

If the '-f <file>' option has been used with cmsetlog to redirect logging to another file than syslog.log, you have also to issue

```
# cmsetlog -s
```

to direct logging to syslog.log again.

Debug logging of SG commands using the '-T' option
For some commands like cmruncl and cmrunnode there is an additional option -F <file> that allows to write the debug logs into a file. This option is not supported by cmcheckconf, cmapplyconf, cmquerycl and others.

This is useful especially if the node is currently not running as a cluster member, because cmsetlog wouldn't work in this case. Here are some examples:

```
# cmruncl -v -T 5 -F /tmp/cmruncl.out
```

The execution of above command gives debug level 5 output in the file /tmp/cmruncl.out for cluster startup on all cluster nodes.

```
# cmquerycl -n Node1 -Node2 -T 3 >cmquery.debug
```

The output of cmquerycl debug logs at log level 3 are written to the local file cmquery.debug.

cmcheckdisk debug logging
`cmcheckdisk` is used on SG/LX to check the status of disks. The following debug options are available:

```
Usage: cmcheckdisk [-h, --help] [-v, --version]
 [-c, --config-file <cfg_file>]
 [-f, --log-file <log_file>]
 [-l, --log-level <1-7>]
 <disk_path>
```

Debug logging of cmlvmd (SG A.11.17 on HPUX and later)
You can start logging for `cmsrvassistd` by adding the following lines in the `/etc/cmcluster.conf` and restarting Serviceguard on the node.

```
CMLVMD_LOG_FILE=/var/adm/cmcluster/cmsrvassistd.log
CMLVMD_LOG_LEVEL=5
```

Note: Do not try to enable `cmlvmd` debug logging online by changing `cmcluster.conf` and then sending SIGHUP to `cmlvmd` PID. This will cause the node to perform a TOC! Starting with PHSS_35427 for SG A.11.17 `cmlvmd` ignores the signal SIGHUP.

Debug logging of cmsrvassistd (SG A.11.17 on HPUX and later)
You can start logging for cmsrvassistd by adding the following lines in the `/etc/cmcluster.conf` and do a `kill -SIGHUP <pid>`.

```
CMSRVASSISTD_LOG_FILE=/var/adm/cmcluster/cmsrvassistd.log
CMSRVASSISTD_LOG_LEVEL=5
```

Debug logging of cmvxd (SG A.11.17 on HPUX and later)
You can start logging for cmvxd by adding the following lines in the `/etc/cmcluster.conf` and do a `kill -SIGHUP <pid>`.

```
CMVXD_LOG_FILE=/var/adm/cmcluster/cmvxd.log
CMVXD_LOG_LEVEL=5
```

Because `cmvxd` talks to the Veritas daemon `vxfend` it might be of interest that the logfile of `vxfend` is at `/var/VRTSvcs/log/vxfend_A.log`.

Debug logging of cmvxpingd (SG A.11.17 on HPUX and later)
You can start logging for `cmvxpingd` by adding the following lines in the `/etc/cmcluster.conf` and do a `kill -SIGHUP <pid>`.

```
CMVXPINGD_LOG_FILE=/var/adm/cmcluster/cmvxpingd.log
CMVXPINGD_LOG_LEVEL=5
```

Debug logging of cmnetassistd (only SG A.11.17 on HPUX)
You can start logging for `cmnetassistd` by adding the following lines in the `/etc/cmcluster.conf` and do a `kill -SIGHUP <pid>`.

```
CMNETASSISTD_LOG_FILE=/var/adm/cmcluster/cmnetassistd.log
CMNETASSISTD_LOG_LEVEL=5
```

Debug logging of cmproxyd (SG A.11.17 on HPUX and later)
You can also start logging for `cmproxyd` by adding the following lines in the `/etc/cmcluster.conf` and do a `kill -SIGHUP <pid>`.

```
CMPROXYD_LOG_FILE=/var/adm/cmcluster/cmproxyd.log
CMPROXYD_LOG_LEVEL=5
```

Debug logging of cmsnmpd (SG A.11.17 on HPUX and later)
You can also start logging for `cmsnmpd` by adding the following lines in the `/etc/cmcluster.conf` and restarting `cmsnmpd`. Debug logging for `cmsnmpd` cannot be enabled online.

```
CMSNMPD_LOG_FILE=/var/adm/cmcluster/cmsnmpd.log
CMSNMPD_LOG_LEVEL=5
```

Debug logging of cmclconfd on HPUX (up to SG A.11.17)
The '-T' option described above can also be used to instrument the cmclconfd. This daemon is used to gather and send configuration data from the local and the remote nodes and is therefore started with many Serviceguard commands. To enable cmclconfd debug logging the following has to be done.

Modify the following lines in the file /etc/inetd.conf file

```
hacl-cfg dgram udp wait root /usr/lbin/cmclconfd cmclconfd -p
hacl-cfg stream tcp nowait root /usr/lbin/cmclconfd cmclconfd -c
```

to

```
hacl-cfg dgram udp wait root /usr/lbin/cmclconfd cmclconfd -p -T 5
hacl-cfg stream tcp nowait root /usr/lbin/cmclconfd cmclconfd -c -T 5
```

and run `# inetd -c`

The logs will go into syslog.log by default.

Starting with SG A.11.13 there is also an "-L" option that allows to redirect the cmclconfd logging to a different file than syslog.log. To enable debug logging with log level 5 and to redirect the logging to /tmp/cmclconfd.log, the above lines need to be modified to

```
hacl-cfg dgram udp wait root /usr/lbin/cmclconfd cmclconfd -p -T 5 -L
/tmp/cmclconfd_udp.log
hacl-cfg stream tcp nowait root /usr/lbin/cmclconfd cmclconfd -c -T 5 -L
/tmp/cmclconfd_tcp.log
```

Make this change on any node where debug logging is needed.

To disable the logging of the cmclconfd, undo the changes in /etc/inetd.conf and run 'inetd -c' again.

Flight Recorder

Serviceguard A.11.15 introduced a new feature called the Flight Recorder, which continuously maintains a detailed log of the most recent Serviceguard activities and stores that data in an accessible file if a failure occurs. Please refer to the Flight Recorder manual at http://haweb.ind.hp.com/Support/misc/FlightRec_v5.pdf (HP internal) for details.

But here are the important steps to obtain the logs:

Obtaining Flight Recorder Logs manually (SG A.11.15 and later on HPUX)

Flight recorder files are saved to /var/adm/cmcluster/frdump.cmcld.x, where x is incremented from 0 to 9. The flight recorder will re-use (overwrite) frdump.cmcld.0 when it closes frdump.cmcld.9.

Format a dump file to make a readable outfile using this example:

```
$ /usr/contrib/bin/cmfmtfr frdump.cmcld.0 > /tmp/frdump0_formatted
```

Use this syntax to identify when the flight recorder file was dumped:

```
$ grep Dumped /tmp/frdump0.formatted
Dumped time: 2011/04/25 07:59:49
```

Obtaining Flight Recorder Logs from an HPUX crashdump (SG A.11.15 and later) using q4

To retrieve SGFR log from a system crash dump, use `q4`, the HP-UX crash dump analyzer. Follow these steps:

1. Run `q4` on the latest system crash dump stored in `/var/adm/crash/`.

2. Load the `cmfr.pl` script (provided with SG in `/usr/contrib/lib/Q4)` in q4, using the include command.

   ```
   q4> include cmfr.pl
   ```

3. While in `q4`, execute `DumpFRB` to extract the SGFR log buffer from the crash dump and puts it into an SGFR binary file, dumpfile.

   ```
   q4> run DumpFRBin dumpfr
   ```

4. Quit `q4`.

5. Apply the `cmfmtfr` command to convert the SGFR binary file, dumpfile, into readable form. The output goes to standard output.

   ```
   # /usr/contrib/bin/cmfmtfr dumpfr
   ```

One can also extract the Flight Recorder logs from a SG TOC dump (or INIT on IA64) by using crashinfo version 3.48 or later. Simply run

```
# crashinfo -sgfr > fr.out
```

For more informations about crashinfo see the section titled ‘Tools’

Obtaining Flight Recorder Logs from a cmcld core file (SG A.11.15 and later on HPUX)

1. Run the `cmcorefr` command thus:

   ```
   # /usr/contrib/bin/cmcorefr -o dumpfr corefile
   ```

`-o <dumpfile>` extracts the SGFR log buffer from the core file. This extracts a SGFR log buffer from the core, and it outputs a SGFR binary file.

2. Run the `cmfmtfr` command to convert the SGFR binary file, dumpfile, into readable form. The output goes to standard output.

```
# /usr/contrib/bin/cmfmtfr dumpfr > dumpfr_formatted
```

Before Logging Serviceguard case

Download the sginfo script from ftp://hpcu:Toolbox1@ftp.usa.hp.com/sginfo (authored by Ross Benight) and run it on a node in the cluster. It collects most all logs and configuration files needed to help L2 and the experts identify failure causes. Normally, an ftp account on ftp.usa.hp.com is needed to get the sginfo collection data to HP. Also provide the following details

- Problem description including commands used, or customer goal:
  - Date and time the problem first occurred (critical!):
  - Error messages:
  - Is the problem repeatable?
  - System changes that may contribute to problem:
  - Problem occurs on multiple clusters or packages?
- Is this a new implementation?
- What troubleshooting and research did you do before elevating the issue?

Solving Serviceguard Problems

Problems with Serviceguard may be of several types. The following is a list of common categories of problem:

- Serviceguard command hangs.
- Cluster re-formations
- Serviceguard TOCs
- System administration errors.
- Package Control Script hangs.
- Package movement errors.
- Problems with VxVM disk groups.
- Node and network failures.
- Quorum Server problems.

- Cluster Lock initialization

Serviceguard Command hangs

Serviceguard uses network messages to other nodes when SG commands involve other nodes. Many Serviceguard commands, including cmviewcl, depend on name resolution services to match the IP of the SG network message to a valid nodename. Nsswitch.conf should use `/etc/nsswitch.files` as a basis, and point *hosts:* to *files* (/etc/hosts) before DNS, to avoid a possible inability to contact the DNS server, which may produce a command hang. All permanent IPs on each node must be listed in /etc/hosts and must be aliased to the nodename (See Managing Serviceguard: Configuring Name Resolution).

Use nslookup to learn how nodenames are resolved. Example:

```
# nslookup Node1

Name Server: server1.cup.hp.com <- DNS is searched first! ☹
Address: 15.13.168.63
Name: Node1.cup.hp.com
Address: 15.13.172.229

What you should see:

# nslookup d7
Using /etc/hosts on:  rxg16u09  <--  Good!

looking up FILES  <-- Good!
Name:    rxg16u07.aqn.gsc.mvlabs.corp.hp.com
Address:  10.226.72.13
Aliases:  rxg16u07, d7, dump7
```

If the output of either command does not include the correct IP address of the node, then check your name resolution services further.

For common Serviceguard Command Problems please refer also to the Section: Common Issues

Cluster Re-formations

Cluster re-formations is logged in syslog.log. Examples:

```
Apr 24 14:07:11 etldts01 cmcld[4966]: 3 nodes have formed a new cluster, sequence #1
Feb 12 23:49:41 tcenh254 cmcld[2436]: 2 nodes have formed a new cluster, sequence #5
Sep 24 18:23:56 sdhn5452 cmcld[4122]: 1 nodes have formed a new cluster, sequence #22
```

Re-formation may occur for the following reasons:

- A node has left or joined the cluster. Serviceguard operates a periodic heartbeat between current members of the cluster, to insure all packages have owners. When nodes leave or join the cluster, heartbeat targeting must be updated, by way of a cluster reformation.

- Heartbeat generation and transmission is delayed by increased kernel activity. The default NODE_TIMEOUT (pre-A.11.19) is often too small. A sign of this is when the 'sequence #' value skyrocket in the syslog.log. Increase the default value of NODE_TIMEOUT (pre-A.11.19) from 2 seconds to 8 seconds or add 10 seconds to MEMBER_TIMEOUT (A.11.19 and newer) in the cluster ASCII configuration file and cmapplyconf the file. If the problem persists, look for syslog.log messages indicating cmcld has not run for several seconds and treat according to SAW documents.

- Excessive network traffic on heartbeat LANs. Created a dedicated heartbeat LAN and change all STATIONARY_IP parameters in the cluster ASCII configuration file to HEARTBEAT_IP and cmapplyconf the file. This allows Serviceguard to use all networks to transmit heartbeat, in the event of a problematic NIC.

- An overloaded system, with too much total I/O and network traffic. Performance analysis may be in order.

- An improperly configured network, for example, one with a very large routing table.

Serviceguard TOC

Serviceguard can invoke an HP-UX TOC (Transfer of Control), to halt the O/S and transfer control to SPU microcode responsible for saving a kernel crash dump. It is not a graceful shutdown because Serviceguard must insure integrity of disk data. The TOC vector is used when Serviceguard does not reset a kernel safety countdown timer which it normally does periodically. If cmcld does not keep on advancing the safety timer, the system clock will eventually over take the safety timer. Once the system clock is equal to or beyond the safety timer, we say that the safety timer expires. To ensure that the node will stop its HA services once the safety timer expires, the node triggers a Serviceguard TOC to take itself out of the cluster. So in essence it is not cmcld that initiates the TOC, it is cmcld that prevents the TOC from happening.

Before initiating the TOC the following message is logged to the kernel's message buffer and to the system's console:

```
Serviceguard: Unable to maintain contact with cmcld daemon.
Performing TOC to ensure data integrity.
```

Beginning with HP-UX 11.22 this kind of information is also logged to the dumps INDEX and to /etc/shutdownlog to make it easier to tell that a TOC was initiated by Serviceguard.
The /etc/shutdownlog will be loaded when the system boots, and show the panic message when dump is saved to /var/adm/crash/crash.N (N increases with savecrash).

```
18:23 Thu Apr 24 2003. Reboot after panic: SafetyTimer expired, ...
```

In a very few cases, an attempt is first made to reboot the system prior to the TOC. If the reboot is able to complete before the safety timer expires, then the TOC will not take place. In either case, packages are able to move quickly to another node.

The following may cause cmcld to cease to reset the safety timer:

1. cmcld is not given CPU time to reset the timer (system hang)
2. A crucial package such as SG-CFS-pkg has failed. The admin can identify others by of setting failfast=enabled in the cluster binary (via the package configuration file).
   Clues such as these appear in OLDsyslog.log:
   ```
   Oct 20 22:32:40 ndhdbp6 vmunix: Halting ndhdbp6 to preserve data integrity
   Oct 20 22:32:40 ndhdbp6 vmunix: Reason: A crucial package failed
   ```
3. The node is refused the cluster lock (or quorum server or lock LUN) during an unexpected cluster reformation. Example in OLDsyslog:
   ```
   Dec 28 14:13:34 kuikka cmcld[3043]: Cluster lock was denied. Lock was obtained by another node.
   ```
4. The node unexpectedly finds itself in a minority of nodes able to communicate with one another (see cluster formation protocol).
5. A shutdown does not halt the cluster daemons, and 'killall' kills cmcld before terminating the relationship with the safety timer. Evidence of this is found in the OLDsyslog.log. Example last line in OLDsyslog.log:
   ```
   Dec  8 17:02:39 uifxp42p syslogd: going down on signal 15
   ```
   The dump will show that cmcld is not on the process list.
   Admins should use cmhaltcl (or cmhaltnode –f) to halt Serviceguard daemons before performing shutdown (per the Managing Serviceguard manual).
6. Veritas (Symantec) cluster file system (available with Serviceguard) uses a different heartbeat which if impaired, will also force a system TOC.

As in item 4, inspect the bottom of OLDsyslog.log that preceded the TOC dump, and the dump, to find cause of the TOC. The INDEX file of the dump identifies when the TOC was performed. Example:

```
dumptime  1323381767 Thu Dec   8 17:02:47 EST 2011
```

System Administration Errors

There are a number of errors you can make when configuring Serviceguard that will not show up when you start the cluster. Your cluster can be running, and everything appears to be fine, until there is a hardware or software failure and control of your packages is not transferred to another node as you would have expected.

These are errors caused specifically by errors in the cluster configuration file and package configuration scripts.

Examples of these errors include:

- Volume groups not defined on adoptive node.
- Mount point does not exist on adoptive node.
- Network errors on adoptive node (configuration errors).
- User information not correct on adoptive node.

You can use the following commands to check the status of your disks:

- `bdf` - to see if your package's volume group is mounted.
- `vgdisplay` - to see if all volumes are present.
- `lvdisplay -v` - to see if the mirrors are synchronized.
- `strings /etc/lvmtab` - to ensure that the configuration is correct.
- `ioscan -fnC disk` - to see physical disks.
- `diskinfo -v /dev/rdsk/cxtydz` - to display information about a disk.
- `lssf /dev/dsk/*` - to check LV and paths.
- `vxdg` list - to list VERITAS disk groups.
- `vxprint` - to show VERITAS disk group details.

Packages

As of version A.11.18, there are now two styles of packages; modular and legacy. Legacy packages were created using cmmakepkg –p to create a package configuration file, and cmmakepkg –s, to create a package control script. The admin then customized parameters in both, and customer_defined_[run|halt]_cmds functions in the control script to make the package control critical business application startup and shutdown. A 2-file format is problematic when the admin forgets to copy modifications to the control script to adoptive nodes. The Serviceguard lab also decided to make package control features modular to also incorporate package enhancements, ECMT toolkit and metrocluster modules more easily. So the result was that cmmakepkg was modified to generate a single package configuration file that consolidates all parameters into the single configuration file. The admin need only customize the one file and cmapplyconf it. All generic control scripts are now placed in /etc/cmcluster/scripts/. Modular package contents are recorded in the cluster binary. Use 'cmviewcl –v –f line' to look at it.

What happens during package start

Modular packages identify order-of-activity in package configuration file. Example:

```
operation_sequence            $SGCONF/scripts/sg/external_pre.sh
operation_sequence            $SGCONF/scripts/sg/volume_group.sh
operation_sequence            $SGCONF/scripts/sg/filesystem.sh
operation_sequence            $SGCONF/scripts/sg/package_ip.sh
operation_sequence            $SGCONF/scripts/sg/external.sh
operation_sequence            $SGCONF/scripts/sg/service.sh
operation_sequence            $SGCONF/scripts/sg/resource.sh
```

Legacy package control scripts generally operate using the same order of operation.

If toolkits are embedded in the package configuration, their entry points will also be listed.
Example listing of a SGeSAP modular package configuration file:

```
operation_sequence            $SGCONF/scripts/sg/external_pre.sh
operation_sequence            $SGCONF/scripts/sg/volume_group.sh
operation_sequence            $SGCONF/scripts/sg/filesystem.sh
operation_sequence            $SGCONF/scripts/sg/package_ip.sh
operation_sequence            $SGCONF/scripts/sgesap/infra_pre.sh
operation_sequence            $SGCONF/scripts/sgesap/sapextinstance_pre.sh
operation_sequence            $SGCONF/scripts/sgesap/livecache.sh
operation_sequence            $SGCONF/scripts/sgesap/dbinstance.sh
operation_sequence            $SGCONF/scripts/sgesap/sapinstance.sh
operation_sequence            $SGCONF/scripts/sgesap/sapextinstance_post.sh
operation_sequence            $SGCONF/scripts/sgesap/infra_post.sh
operation_sequence            $SGCONF/scripts/sg/external.sh
operation_sequence            $SGCONF/scripts/sg/service.sh
operation_sequence            $SGCONF/scripts/sg/resource.sh
operation_sequence            $SGCONF/scripts/sgesap/mdm.sh
```

Note the sgesap modules are interspersed among standard Serviceguard modules by the sgesap toolkit module during package configuration creation.

Standard Serviceguard package startup in general consists of several steps, summarized in the next section (using legacy package control script function identifiers)

- **activate_volume_group** All volume groups of the package are activated with exclusive option (as specified in the package control script). The message Activation mode requested for the volume group conflicts with configured mode appears if the volume group is not flagged as cluster aware (`vgchange -c y <VG>`). The command `cmapplyconf` sets this flag autmatically for all cluster aware VGs listed in the cluster ASCII file and it clears it from all VGs that are missing (if no `-k` option is used).

- **check_and_mount** The file systems are checked by fsck and are then mounted, using the mount options specified in the package control script. Missing (No such file or directory) or busy (already mounted, is busy, or allowable number of mount points exceeded) mount points make package start fail in this stage. Use fuser <directory> to find responsible processes.

- **add_ip_address** The Serviceguard command cmmodnet is used to configure all relocatable IP addresses associated with this package. The command can be also used from command line, e.g. `cmmodnet -a -I 192.10.10.120 192.10.10.0`. Extreme caution should be exercised when executing this command outside the context of the package control script.

- **customer_defined_run_cmds** This is the place where the customer's HA application is started. Failures analysis in this area should be started from the application side. Commenting out the faulty command may be a good strategy to get a minimum trouble-shooting environment running (otherwise the complete package start fails, causing all file systems to be umounted, etc). In modular packages, this section is handled by the external.sh operation.

- **start_services** The Serviceguard command `cmrunserv` is used to start the package's services. These are typically shell scripts monitoring the health of the HA application. An exiting service script means Monitoring Failed, causing the corresponding package to failover to an adoptive node. The cmrunserv and cmhaltserv commands must not be used manually from the command line.

- **start_resources** The Serviceguard command `cmstartres` is used to start monitoring of deferred EMS resources.

What happens during package stop

Stopping of a package performs in general the reverse order of steps as described above. If a package halt fails, cleanup may be needed to get the system back to a defined "package halted state" manually. This includes:

- Package services are stopped.
- The application is halted.
- All relocatable IP adresses are de-configured. If not done automatically, this can be done manually using e.g. `cmmodnet -r -i 192.10.10.120 192.10.10.` or `ifconfig lan0:2 0.0.0.0`
- All package filesystems are umounted. All processes keeping them busy need to be terminated first (the *'lsof'* command may be useful to identify processes not killed by fuser). If the filesystem was exported via NFS it may be required to kill/restart the `rpc.statd` and `rpc.lockd` processes also.
- Deactivate all VGs of the package (`vgchange -a n VG`). This is only possible if all filesystems were successfully umounted before. Otherwise this fails with a Device busy error.

Package Control Script Hangs or Failures

When a `RUN_SCRIPT_TIMEOUT` or `HALT_SCRIPT_TIMEOUT` value is set, and the control script hangs, causing the timeout to be exceeded, Serviceguard kills the script and marks the package "Halted." Similarly, when a package control script fails, Serviceguard kills the script and marks the package "Halted." In both cases, the following also take place:

- Control of a failover package will not be transferred.
- The run or halt instructions may not run to completion.

- AUTO_RUN (automatic package switching) will be disabled (important!).
- The current node will be disabled from running the package.
- Following such a failure, since the control script is terminated, some of the package's resources may be left activated. Specifically:
- Volume groups may be left active.
- File systems may still be mounted.
- IP addresses may still be installed.
- Services may still be running.

In this kind of situation, Serviceguard will not restart the package without manual intervention. You must clean up manually before restarting the package. Use the following steps as guidelines:

1. Perform application-specific cleanup. Any application-specific actions the control script might have taken should be undone to ensure successfully starting the package on an alternate node. This might include such things as shutting down application processes, removing lock files, and removing temporary files.

2. Ensure that package IP addresses are removed from the system. This step is accomplished via the `cmmodnet(1M)` command. First determine which package IP addresses are installed by inspecting the output resulting from running the command `netstat -in`. If any of the IP addresses specified in the package control script appear in the netstat output under the "Address" column for IPv4 or the "Address/Prefix" column for IPv6, use `cmmodnet` to remove them:

```
# cmmodnet -r -i <ip-address> <subnet>  (or ifconfig lanX:Y 0.0.0.0)
```

   where `<ip-address>` is the address in the "Address" or the "Address/Prefix column and `<subnet>` is the corresponding entry in the "Network" column for IPv4, or the prefix (which can be derived from the IPV6 address) for IPv6.

3. Ensure that package volume groups are deactivated. First unmount any package logical volumes which are being used for filesystems. This is determined by inspecting the output resulting from running the command `bdf -l`. If any package logical volumes, as specified by the LV[] array variables in the package control script, appear under the "Filesystem" column, use umount to unmount them:

```
# fuser -ku <logical-volume>
# umount <logical-volume>
```

   Next, deactivate the package volume groups. These are specified by
   the VG[n] array entries in the package control script, or 'vg' in the package configuration file. Identify VGs associates with a modular packages with 'cmviewcl –v –f line | grep vg'.

```
# vgchange -a n <volume-group>
```

4. Finally, re-enable the package for switching.

```
# cmmodpkg -e <package-name>
```

If after cleaning up the node on which the timeout occurred it is desirable to have that node as an alternate for running the package, remember to re-enable the package to run on the node:

```
# cmmodpkg -e -n <node-name> <package-name>
```

Package Movement Errors

Package fail to move to an adoptive node for these reasons:

- The initial package halt failed
- Package startup occurred on the target node, but something failed during package startup.
- Package startup is not allowed on the target node

The first two causes require inspection of the package log and maybe the syslog. The last cause requires inspection of the AUTO_RUN and Node_Switching flags, which requires '*cmviewcl –v –p <pkg_name>*'. Enable AUTO_RUN using '*cmmodpkg –e <pkg_name>*' and enable Node_switching using '*cmmodpkg –e –n <nodename> <pkg_name>*'

Network issues

Here we will first cover Node and Network Failures which are normal action of Serviceguard and then discuss how to troubleshoot "real" network Problems.

Node and Network Failures

These failures cause Serviceguard to transfer control of a package to another node. This is the normal action of Serviceguard, but you have to be able to recognize when a transfer has taken place and decide to leave the cluster in its current condition or to restore it to its original condition.

Possible node failures can be caused by the following conditions:

- HPMC - (High Priority Machine Check) is a system panic caused by a hardware error.
- TOC – admin manually forced a transfer of control to the TOC/dump vector in microcode.
- Panics – kernel malfunction, generating a memory dump
- Hangs – Serviceguard will trigger a TOC.

- Power failures – not documented in /etc/shutdownlog! Check MP logs.

In the event of a TOC, a system dump is performed on the failed node and numerous messages are also displayed on the console.

You can use the following commands to check the status of your network and subnets:

- `netstat -in` - to display LAN status and check to see if the package IP is assigned to a NIC.
- `cmviewcl -v [-n node] – to see if a standby NIC has been invoked.`
- `arp -a` - to check the arp tables.
- `lanadmin` - to display, test, and reset the LAN cards.

How to track down Networking Problems

While discovering the network (e.g. during `cmcheckconf` or `cmapplyconf`), Serviceguard records 'bridged networks'. Interfaces that can communicate on the link level are associated to the same brigded net. Bridged networks are revealed using the older `cmviewconf` or newer method:

```
root@rxg15u20:root# cmviewcl -v -f line | grep bridge
node:rxg15u18|interface:lan0|bridged_net=1
node:rxg15u18|interface:lan3|bridged_net=2
node:rxg15u18|interface:lan1|bridged_net=2
node:rxg15u20|interface:lan0|bridged_net=1
node:rxg15u20|interface:lan3|bridged_net=2
node:rxg15u20|interface:lan1|bridged_net=2
```

Network interfaces with the same 'bridged net ID' are on the same bridged net. A good starting point for troubleshooting networking problems is `cmquerycl –l net –v`. To check the link level connectivity between interfaces one may use the `linkloop(1M)` command. The `cmscancl(1M)` script does this checking automatically for all interfaces on all nodes of the cluster.

Example:

```
# lanscan
Hardware Station        Crd Hdw   Net-Interface  NM  MAC       HP-DLPI DLPI
Path     Address        In# State NamePPA        ID  Type      Support Mjr#
0/1/2/0  0x0017A4AB10D1 0   UP    lan0 snap0     1   ETHER     Yes     119
0/1/2/1  0x0017A4AB10D0 1   UP    lan1 snap1     2   ETHER     Yes     119
```

The following linkloop command checks the local link level connectivity from lan2 to lan1. Please note that you need to specify the outgoing PPA (NMID for 10.x) with the –i option.

```
# linkloop -i 2 0x00108318AFED
Link connectivity to LAN station: 0x00108318AFED
--- OK
```

Sometimes Serviceguard's discovery does not match the results achieved with `linkloop`, because it uses a slightly differerent method. The HP unsupported tool `dlpiping` (More information about dlpiping can be found under subsection "Tools") can be used in such cases. The `dlpiping` tool uses exactly the same communcation mechanism that Serviceguard uses. Therefore the results are more representative than simply using `linkloop`.

To test connectivity from local PPA 1 to local PPA 2:

```
# dlpiping 2 1
Bound PPA 2, Ethernet, address 0x00108318afee
Bound PPA 1, Ethernet, address 0x00108318afed
Send from PPA 2 to PPA 1 0x00108318afedaa080009167f
Send from PPA 1 to PPA 2 0x00108318afeeaa080009167f
Recv from 0x00108318afedaa080009167f on PPA 2 0x00108318afeeaa080009167f
Recv from 0x00108318afeeaa080009167f on PPA 1 0x00108318afedaa080009167f
```

To test connectivity from local PPA 1 to remote PPA 2:

```
# dlpiping -n <remote node> 2 1
Bound PPA 2, Ethernet, address 0x00108318afee
Bound remote PPA 1, Ethernet, address 0x00108318cff8
Send from PPA 2 to remote PPA 1 0x00108318cff8aa080009167f
Send from remote PPA 1 to PPA 2 0x00108318afeeaa080009167f
Recv from 0x00108318cff8aa080009167f on PPA 2 0x00108318afeeaa080009167f
Recv from 0x00108318afeeaa080009167f on remote PPA 1
0x00108318cff8aa080009167f
```

Tools

crashinfo

`crashinfo` is a tool for helping with coredump analysis, in particular system hangs. `crashinfo` prints helpful notes and warnings intended for less experienced dump readers, and also things that may be easily overlooked by more experienced users such as the psw Q-bit being off. It also tries to be "smart" with certain types of panics. For example, with spinlock deadlock panics `crashinfo` will attempt to find and print the lock structure, and the stack trace of the lock holder. Other examples are for DPF and MPF panics it pulls the trapping address etc from the `save_state`, decodes the instruction, and depending on the trapping address calls the `kmeminfo` -virtual function. When we are very low on memory, or in the case of a `kalloc` panic, we again call `kmeminfo` routines to display the kernel memory usage.

More information can be found on
http://teams3.sharepoint.hp.com/teams/esssupport/InsideESSSupport/InsideWTEC/HPUX-

KERNEL/tools/downloads.htm (HP internal)

dlpiping
http://teams3.sharepoint.hp.com/teams/esssupport/InsideESSSupport/InsideWTEC/HA/Pages/Tools.aspx **(HP internal).**

`Dlpiping` is an unsupported program to help troubleshoot ServiceGuard problems where errors are reported such as non uniform network connections.

The program sends messages at a link level to check link level connectivity. The advantage that this program has over `linkloop` is that it sends real messages with the same sap and snap binding used by `cmclconfd` and `cmcld`, whereas `linkloop` sends special test packets.

Since real messages are being sent and received the program needs to have a component running on the system sending and receiving the messages. When connectivity is being checked between interfaces on a single system this is done via a single instance of the program. When connectivity is being checked between 2 nodes the program starts another copy of itself on the remote node after copying itself into `/tmp` on the remote node. This is handled transparently within the program and relies on tcp, remsh and rcp working.

Here are the key features of `dlpiping`:

- sends messages matching the size and binding of `cmclconfd` or `cmcld`
- reports network type
- reports original mac address if address downloaded
- checks returned message size
- supports Ethernet, 100VG, Token Ring, FDDI and Fibre Channel
- controllable message timeout

More information can be found at this website:
http://teams3.sharepoint.hp.com/teams/esssupport/InsideESSSupport/InsideWTEC/HA/Pages/Tools-dlpiping.aspx (HP internal).

Common Issues
This chapter will give us a short overview of the most common Serviceguard issues, divided in the following groups:

- LVM related problems
- Networking related Problems
- Problems with VxVM Disk Groups
- Further Problems
- Intermittent Cluster Reformations
- Cluster daemon abort with possible node TOC
- Serviceguard Command Problems

Most of them are taken from the daily support work. Hopefully they will help you in finding and solving your problems.

LVM related Problems

The predominant causes of LVM problems are incorrect content of /etc/lvmtab (or lvmtab_p) on each node, or incorrect activation mode assigned to a given volume group.

Example 1
If a disk is missing from lvmtab on one node, cmcheckconf may fail with:

```
ERROR: Volume group vgA is configured differently on node Node1 than on node
Node2.
```

If an admin adds a disk to a VG on one node, the lvmtab files on the other nodes become out of date until the admin resolves the problem. The fastest way is to use a map file containing the VGID, then using vgexport and vgimport to re-load the volume group LUN list to lvmtab.

The message may take on a warning mode instead, if only Pvlinks are missing from lvmtab on a given node.

Example 2
If an admin fails to add a VOLUME_GROUP parameter to the cluster ASCII file and attempts to start a package that attempts to activate that VG, the package log will record something like this:

```
Error: vgchange:  Activation mode requested for the volume group "/dev/vg03"
conflicts with configured mode.
```

Example 3
Errors: Volume group vgA currently belongs to another cluster. First cluster lock volume group vgA belongs to another cluster. Second cluster lock volume group vgB belongs to another cluster.

The LVM header of that volume group contains the cluster ID of a different cluster. If you are sure that your cluster should own these disks you can perform

```
# vgchange -c n <VG>
```

to clear the ID. This command should be performed on node on witch the VG is active.

Example 4

```
Warning: The disk at XXX on node Node1 does not have an ID.
```

Serviceguard prints warning messages for all disks that do not contain a valid header. This is normal and can be ignored for all disks that are not part of a LVM volume group or VxVM disk group.

Example 5

```
Error: Unable to determine a unique identifier for physical volume vgA on
node Node1.
```

An attempt to read a valid PVID from the listed device failed. Check if the device is readable and contains a PVID ≠ 0.

Example 6

```
Error: Unable to recv initialize VG message to %s: %s
```

Symptom: `cmcheckconf/cmapplyconf` hangs or needs very long to complete.

When performing cmcheckconf/cmapplyconf a `cmclconfd` helper process is launched on each node for gathering configuration information. Most likely reasons for hangs or long runtimes are `cmclconfd` processes being blocked while trying to access disk devices. By default `cmclconfd` expects every claimed disk that is visible in ioscan to be readable without problems. This assumption is not true in some cases, especially for devices associated to physical data replication (ContinousAccess/XP, BusinessCopy/XP, EMC SRDF, EMC Timefinder). Making the devices accessible (e.g. by splitting pairs) should solve related problems.

Use e.g. `ps -el | grep cmclconfd` on every cluster node. The `cmclconfd`'s Open Files screen in GlancePlus or the public domain tool `lsof` should show you what device we are waiting for. Another workaround is to temporarily remove all VOLUME_GROUP statements (except for cluster lock volume groups) form the cluster ASCII file (comment them out by prepending a '#' sign) and use the `-K` option with cmcheckconf or cmapplyconf.

Example 7

```
Error: Found two volume groups with the same name vgA but different ids

Warning: The volume group vgA is activated on more than one node
```

This problem typically occurs when Serviceguard considers "privat" volumes groups to be "shared", which happens often to the root volume group `/dev/vg00`. Serviceguard uses LVM's VGID as unique differentiator to find shared volume groups. In general there are two scenarios why Serviceguard may get misleading information about volume groups, complaining about their configuration:

- The systems were cloned, meaning that their private disks were copied using tools like `dd(1)`. This causes the VGIDs to be identical in their LVM headers, although they should be unique.

- The VGID in the `/etc/lvmtab` file does not match the VGID of attached disks. Copying the lvmtab file to another system might cause this problem; `vgscan(1M)` can be used to fix the file.

Example 8

Symptom: Some non-shared LVM volume groups are not automatically activated during bootup.

If the admin customizes the `/sbin/lvmrc` script to avoid activating cluster-aware VGs at boot time (to avoid sending error messages like those in example 2 to the console), the admin may forget to list non-shared volume groups in the file, particularly if adding some later.

In all reality, it is easier leave the file generic and simply ignore the boot-time error messages than it is to try to remember this file when adding VGs.

Networking related Problems

Networking problems can be devided into 2 classes

- Serviceguard commands fail
- Network connectivity fails

Serviceguard commands fail

Serviceguard commands are processed by cmclconfd, which is activated by inetd. Cmclconfd is also tied to identd, a daemon included with the sendmail software. If identd cannot match the packet IP to an an authorized hostname, the command is aborted. The common solutions are:

- Load /etc/cmcluster/cmclnodelist on each node
- Insure /etc/inetd.conf has correct hacl-cfg lines.
```
hacl-probe  stream  tcp    nowait  root  /opt/cmom/lbin/cmomd
/opt/cmom/lbin/cmomd -i -f /var/opt/cmom/cmomd.log -r /var/opt/cmom
hacl-cfg    dgram   udp    wait    root  /usr/lbin/cmclconfd cmclconfd -p
hacl-cfg    stream  tcp    nowait  root  /usr/lbin/cmclconfd cmclconfd -c -i
```
- Insure /etc/hosts identifies all IPs hosted by each node, and alias all IPs to the host nodename.
- Insure /etc/nsswitch.conf searches *files* before *dns*.
- Insure /var/adm/inetd.sec does not deny hacl-cfg or registrar for the affected nodes
- Insure /etc/opt/sec_mgmt/bastille/config does not prohibit hacl activity
- Insure the random number generator driver is loaded and /dev/*random files exist.
- kill the 'cmclconfd –p' if any and reset inetd (inetd –k ; inetd).

Examples follow:

Example 1

```
Error: Unable to determine the nodes on the current cluster.

Error: Unable to communicate with node Node1.
```

Serviceguard's `cmclconfd` does authorization checks before servicing Serviceguard commands. The `cmclconfd` daemon is activated by `inetd`, so it is a good idea to also check `/etc/inetd.conf`, `/etc/services` and `/var/adm/inetd.sec`. Correct `/etc/inetd.conf` entries look like this:

```
hacl-cfg dgram udp wait root /usr/lbin/cmclconfd cmclconfd –p
hacl-cfg stream tcp nowait root /usr/lbin/cmclconfd cmclconfd –c
```

These are the valid `/etc/services` entries:

```
clvm-cfg 1476/tcp # HA LVM configuration hacl-hb 5300/tcp # High Availability
(HA

hacl-gs 5301/tcp # HA Cluster General Services
```

```
hacl-cfg 5302/tcp # HA Cluster TCP configuration
hacl-cfg 5302/udp # HA Cluster UDP configuration
hacl-probe 5303/tcp # HA Cluster TCP probe
hacl-probe 5303/udp # HA Cluster UDP probe
hacl-local 5304/tcp # HA Cluster Commands
hacl-test 5305/tcp # HA Cluster Test
hacl-dlm 5408/tcp # HA Cluster distributed lock manager
```

Check this from every node to every other node of the cluster:

```
# telnet localhost hacl-cfg

# telnet <own hostname> hacl-cfg

# telnet <other hostname> hacl-cfg
```

The telnet commands should simply hang, which is the normal behaviour if you reach the remote `cmclconfd`. Messages like "`connection refused`" or "`hacl-cfg: bad port number`" indicate a problem.

Example 2

```
Error: Node Node1 is refusing Serviceguard communication.
Please make sure that the proper security access is configured on node
Node1 through either file-based access (pre-A. %s version) or role-based
access (version A.11.16 or higher) and/or that the host name lookup on node
Node2 resolves the IP address correctly. cmcheckconf: Failed to gather
configuration information
```

The reason for this error message could be:

- The file cmclnodelist is incorrect
- network issue

For troubleshooting try to add a + in the cmclnodelist file.

If there is a network issue:

Check the network: nslookup, nsquery, telnet
Check the lan configuration

- `Error: `**`Non-uniform connections detected`**

This error is usually the result of a misconfiguration of a network component. The dominant errors are mismatching speed/duplex of a LAN interface, sometimes even assigning the same link level address to more than one interface. The problem occurs while Serviceguard probes the IP or link level connectivity between all interfaces that are part of the cluster configuration. The probing result needs to be always symmetrical. Traffic needs to pass either in both or no direction!

Please note that often linkloop(1M) is not able to catch such problems since its checking differs from Serviceguard's. Please refer to the tool dlpiping. More information about the tool can be found under Troubleshooting.

- `Error:` **`Network interface lanX on node Node1 couldn't talk to itself.`**

This message is usually a result of a failed LAN interface check, either on link level. Each interface being part of the cluster configuration needs to be either configured (UP and ping'able through its IP address) or unconfigured (unplumbed standby interface, without having any IP adress).

If the interface is supposed be configured with HEARTBEAT_IP or STATIONARY_IP you should check with e.g. ping(1M) if it is up and running. Otherwise, for standby interfaces, you should check with ifconfig(1M) if it is really unplumbed. It should return no such interface.

It's also important that every interface is able to communicate on link level. Verify this for each reported interface using the linkloop(1M) command:

```
# linkloop –i <PPA> <MAC address>
```

The –i option specifies the lan interface as outgoing interface.

Please refer to the tool dlpiping. More information about the tool can be found under Troubleshooting

Note: Some Ignite revisions are known to recover a bogus "0.0.0.0" configuration for unused interfaces to /etc/rc.config.d/netconf. Remove that entry and unplumb the interface as described above.

- `Error: Detected a partition of IP subnet X.X.X.X.`

The error indicates that some lan interfaces are unable to talk to each other on IP level, although they should be able to do so. Verify if the configuration information in the cluster ASCII file is correct. Check the network's physical connections. A failure with the linkloop command means there is no connectivity on link level, maybe because some network component such as a switch does not pass that type of traffic.

- DLPI errors: Serviceguard cluster daemon (cmcld) may abort with error messages: "`Unable to send DLPI message, Interrupted system call`" followed by "`Aborting! Failed to send over DLPI`"

Serviceguard uses DLPI (Datalink Provider Interface) to perform network polling in order to check the health of the lan cards in the cluster. Link level packets are sent and received which allows cmcld to gather statistical information to ensure data is being transmitted and received and to check for errors returned by the network drivers. Discussing all possible DLPI error conditions would exceed this document's scope. It is usually best practice to check affected interfaces with tools like linkloop(1M) and lanadmin(1M). In the past many of those problems were tracked down to defective hardware components. Some of those problem were caused by Serviceguard defects, so it is also advisable to install a current Serviceguard patch to address known problems in this area.

• syslog message indicates DLPI message corruption.
Possible messages include:

```
cmcld: DLPI message too small (X < Y + Z). Ignoring the message.
cmcld: DLPI message checksum incorrect upon completion. Group, X, seems to be
corrupting the message.
cmcld: DLPI message not sent because oversized (X + Y > Z).
cmcld: Length of DLPI header (X) is too small. Ignoring the message.
cmcld: DLPI message too small (X < Y + Z). Ignoring the message.
cmcld: DLPI message too big (X + Y). Ignoring the message.
cmcld: The comm_link message length is inconsistant. (X < Y + Z).
cmcld: DLPI message checksum incorrect upon receipt from X. Ignoring the
message.
```

Beginning with Serviceguard A.10.12 DPLI traffic is protected against corruption using checksums. Earlier revisions could abort under such conditions resulting in a node TOC. The affected hardware (interfaces, network switch or hub, other hardware on that bridged net, etc.) should be checked and replaced if needed. See also HA Newsletter #15 (HP internal) for details.

• Problem: Crossover cables for a Heartbeat LAN of 2-node clusters

When either LAN card fails, or the crossover cable is disconnected, both LAN cards go down. This is because the electrical signals necessary for the cards to determine that a valid LAN connection exists are not present. The result is that since both nodes appear to have a bad LAN card, Serviceguard may TOC the wrong node. On multi-speed cards, such as 10/100Base-T, the cards should not negotiate which speed will be used when the system boots up. Otherwise, if only one system is booted and the remote system is down, the negotiation would fail, and the card would not be enabled at all. So when the second node eventually comes up, it's LAN would also be down.
For the reasons listed above, it is not recommend (but supported) to use crossover cables for Serviceguard configurations. Auto-negotiation should be disabled by setting a fixed speed/duplex configuration.

Problems with VxVM Disk Groups

This section describes some approaches to solving problems that may occur with VxVM disk groups in a cluster environment. For most problems, it is helpful to use the vxdg list command to display the disk groups currently imported on a specific node. Also, you should consult the package control script log files for messages associated with importing and deporting disk groups on particular nodes.

Force Import and Deport After Node Failure

After certain failures, packages configured with VxVM disk groups will fail to start, and the following error will be seen in the package log file:

```
vxdg: Error gd_01 may still be imported on Node1
ERROR: Function check_dg failed
```

This can happen if a package is running on a node which then fails before the package control script can deport the disk group. In these cases, the host name of the node that had failed is still written on the disk group header.

When the package starts up on another node in the cluster, a series of messages is printed in the package log file, as in the following example (the hostname of the failed system is Node1, and the disk group is dg_01):

```
check_dg: Error dg_01 may still be imported on Node1
```

To correct this situation, logon to node1 And execute the following command:

```
vxdg deport dg_01
```

Once dg_01 has been deported from Node1, this package may be restarted via either cmmodpkg(1M) or cmrunpkg(1M).
In the event that Node1 is either powered offor unable to boot, then dg_01 must be force imported.

******************* WARNING*************************
The use of force import can lead to data corruption if Node1 is still running and has dg_01 imported. It is imperative to positively determine that Node1 is not running prior to performing the force import. See -C option on vxdg(1M).
****

To force import dg_01, execute the following commands on the local system:

```
vxdg -tfC import $dg
vxdg deport $dg
```

Follow the instructions in the message to use the force import option (-C) to allow the current node to import the disk group. Then deport the disk group, after which it can be used again by the package. Example:

```
vxdg -tfC import dg_01
vxdg deport dg_01
```

The force import will clear the host name currently written on the disks in the disk group, after which you can deport the disk group without error so it can then be imported by a package running on a different node.

Note: This force import procedure should only be used when you are certain the disk is not currently being accessed by another node. If you force import a disk that is already being accessed on another node, data corruption can result.

Further Problems

In this section are described further uncategorized very well known Serviceguard problems.

• Sendmail version

Some companies such as AT&T install an unsupported version of sendmail. Sendmail hosts identd, which Serviceguard uses to validate the host from which Serviceguard commands are run. If the customer uses an unsupported version of sendmail, edit the /etc/inetd.conf file and modify this hacl-cfg line thus:

```
hacl-cfg    stream  tcp    nowait  root  /usr/lbin/cmclconfd cmclconfd -c -i
```

Then restart inetd:

```
$ inetd -k ; inetd
```

• Problem: **HP-UX Strong Random Number Generator**

Serviceguard A.11.16 and the latest patches for A.11.15 include functionality which uses the HP-UX Strong Random Number Generator on HP-UX 11i if this is loaded.

Unfortunately some versions of the HP-UX Strong Random Number Generator prior to version B.11.11.07 have a defective startup script which can result in the wrong major number for the random number device files /dev/random and /dev/urandom. This can result in a number of different symptoms which can be difficult to troubleshoot. Serviceguard commands can hang, fail or take a very long time to work often with no apparent errors in syslog.log or other external symptoms.

Any Serviceguard clusters which have the HP-UX Strong Random Number Generator software loaded should ensure they have version B.11.11.07 or newer. This can be checked by looking at the version of the KRNG11i bundle:

```
# swlist -l bundle | grep KRNG
KRNG11i      B.11.11.09   HP-UX 11.11 Strong Random Number Generator
```

If you have a cluster which is behaving erratically after installing Serviceguard A.11.16 or the patches shown above and the HP-UX Strong Random Number Generator is loaded you should check that the /dev/urandom device file has the correct major number. The driver given by the major number should be the "rng" driver. This can be verified by checking the devices files and the kernel major number:

```
# ls -l /dev/*random
cr--r--r--   1 bin      bin     62 0x000000 Nov 19 18:09 /dev/random
cr--r--r--   1 bin      bin     62 0x000001 Nov 19 18:09 /dev/urandom
# lsdev | grep 62
        62          -1         rng             pseudo
```

If the driver shown by lsdev grep'ing for the major number of the
/dev/urandom device file is not "rng" then Serviceguard will not
function correctly. Note that the actual major number can vary since
this number is dynamically allocated.

If the major number does not match, the device files should be corrected
or deleted before Serviceguard is started until a newer version of the
HP-UX Strong Random Number Generator is loaded. If /dev/urandom is not
present Serviceguard uses an alternate mechanism for generating random
numbers.

- Problem: **cmclconfd[####]: cmclconfd running with weak security (identd disabled)** in the syslog.log file.

When a system logs the above messages one or several conditions have been meet:

1) The new security feature has been disabled/bypassed using the /etc/inetd.conf file.

2) All the nodes in the configured cluster do not have the Serviceguard Security patch installed properly.
3) All the nodes have not successfully formed a complete cluster
4) Optional: The ‘-i’ option has been added to the `hacl-cfg cmclconfd -c` line in /etc/inetd.conf.

- cmruncl : **Cluster did not form. Check the syslog file for information.**

The following is the output from cmruncl:

```
# cmruncl -v
cmruncl  : Validating network configuration...
Gathering configuration information .......... Done
cmruncl  : Network validation complete
Successfully started $SGLBIN/cmcld on Node1.
Successfully started $SGLBIN/cmcld on Node2.
cmruncl  : Waiting for cluster to form...........
cmruncl  : Cluster did not form.  Check the syslog file for information.
```

The following is info from "/var/adm/syslog/syslog.log":

```
Apr  2 11:38:55 db1 cmclconfd[####]: The ServiceGuard
```

```
daemon, /usr/lbin/cmcld [####], died upon receiving signal number 11.
```

The kernel parameter maxssiz was set too low. Change maxssiz back to its previous setting.

• cmcld: **WARNING: Cluster lock on disk /dev/dsk/cXtYdZ is missing. Until is fixed, a single failure could cause all nodes in the cluster to crash.**

This event has been known to be caused by the following:
a. During the most recent cluster configuration, the cluster lock VG was active in vgchange -a y on one of the adoptive nodes in the cluster.
b. The cluster lock disk was replaced or moved to a different disk.

Insure the cluster lock VG is listed in /etc/lvmtab and that the cluster binary file uses the correct device special file (cmapplyconf).

• Problems with SAP-Package-Start

For DEBUGGING purposes, the following steps can be used to start Serviceguard in Debug mode:

1. If the package is running, run "cmhaltpkg -v <pkgname>".
2. Run the following unix command on the ServiceGuard nodes:

   touch /etc/cmcluster/<SID>/debug

3. Next, enter:
   cmrunpkg -n <failover_node> -v <pkgname>

   At this point, the package should start without starting the DataBase or SAP.

4. Now, just specify the SAP startup script and observe. This method avoids executing Serviceguard SAP script.

Intermittent Cluster Reformations

• Problem: **Intermittent cluster reformations with possible node TOC**

If a node does not receive a heartbeat from a remote node within the NODE_TIMEOUT interval, then that node will be timed out. At that time all cluster nodes enter the cluster reformation process. If the heartbeat interval is one second, and the node timeout interval is two seconds, it takes two consecutive missed heartbeats to cause the node to time out, and a cluster reformation to start. Cluster reformation involves informing all nodes of the reformation (including the node which missed the heartbeat), voting for a new cluster coordinator, and reforming the cluster (the new cluster is based upon the number of nodes which responded during the reformation. Note, if the node which missed the heartbeat is able to respond during the reformation, then the reformation will end up with the same number of nodes in the cluster and your packages will not be effected).

Example: Node Node1 is missing heartbeats from Node2. It starts reformation and goes for the cluster lock disk, before Node2 comes back:

```
Aug 5 16:08:29 Node1 cmcld: Timed out node Node2. It may have failed.
Aug 5 16:08:29 Node1 cmcld: Attempting to form a new cluster
Aug 5 16:08:36 Node1 cmcld: Obtaining Cluster Lock
Aug 5 16:08:36 Node1 vmunix: SCSI: Reset requested from above --
lbolt:25093597, bus:0
Aug 5 16:08:37 Node1 vmunix: SCSI: Resetting SCSI -- lbolt:25093697, bus:0
Aug 5 16:08:37 Node1 vmunix: SCSI: Reset detected -- lbolt:25093697, bus:0
Aug 5 16:08:54 Node1 cmcld: Attempting to adjust cluster membership
Aug 5 16:08:56 Node1 cmcld: Enabling safety time protection
Aug 5 16:08:56 Node1 cmcld: Clearing Cluster Lock
Aug 5 16:08:57 Node1 cmcld: 2 nodes have formed a new cluster, sequence #7
Aug 5 16:08:57 Node1 cmcld: The new active cluster membership is:
Node1(id=1), Node2(id=2)
```

Often the factory-default setting for the cluster parameter NODE_TIMEOUT of 2 seconds (2000000 microseconds) is too small for many configurations. The general recommendation is to set NODE_TIMEOUT in the range of 5000000-8000000 microseconds.

• Problem: **SCSI reset messages logged to syslog and the kernel's message buffer**

If Serviceguard performs a cluster reformation, SCSI reset messages appear in the dmesg output and syslog.log, if the reformation requires a race to the cluster lock disk. To ensure that the SCSI bus is available for the server to grab the lock, a reset is performed, (please note the "Reset requested from above" message), e.g.:

```
SCSI: Reset requested from above -- lbolt: 400804081, bus: 0 SCSI: Resetting
SCSI -- lbolt: 400804381, bus: 0 SCSI: Reset detected -- lbolt: 400804381,
bus: 0
```

The SCSI messages are only seen when the cluster lock disk is handled by the sdisk driver. The disc3 driver does not output information when a SCSI reset is issued or detected.

• Error: **The local node Node1 appears to belong to a different cluster.**

The node's /etc/cmcluster/cmclconfig file already contains a configuration with a different cluster ID in it. You should use cmdeleteconf to remove that configuration first. As a last resort you can remove the file from that node.

Cluster daemon abort with possible node TOC

(See the section titled Serviceguard TOC)

• Active cmcld aborts with syslog messages like:

```
cmcld: Aborting!
cmcld: Service Guard Aborting!
cmcld: Aborting Serviceguard Daemon to preserve data integrity.
```

These messages are logged by cmcld before it actively aborts due to some fatal error condition, that may be also part of the error message. Typically the syslog.log looks similar to this:

```
Aug 5 11:05:31 Node1 cmcld: Aborting: cl_rwlock.c 1030 (reader/writer lock
not locked)
Aug 5 11:05:35 Node1 cmlvmd: Could not read messages from /usr/lbin/cmcld:
Software caused connection abort
Aug 5 11:05:35 Node1 cmlvmd: CLVMD exiting
Aug 5 11:05:35 Node1 cmsrvassistd[8688]: The cluster daemon aborted our
connection.
Aug 5 11:05:35 Node1 cmsrvassistd[8688]: Lost connection with Serviceguard
cluster daemon (cmcld): Software caused connection abort
Aug 5 11:05:35 Node1 cmtaped[8691]: The cluster daemon aborted our
connection.
Aug 5 11:05:35 Node1 cmtaped[8691]: cmtaped terminating. (ATS 1.14)
…
```

The exact error message should be checked against known problems. Often also a cmcld core file is written to the directory /var/adm/cmcluster, which may be helpful for further investigation.

• Node TOC after tuning TCP parameters using ndd(1M):

```
Oct 17 10:10:02 Node1 cmlvmd: Could not read messages from /usr/lbin/cmcld:
Connection reset by peer
Oct 17 10:10:02 Node1 cmlvmd: CLVMD exiting
...
Oct 17 10:10:08 Node1 vmunix: Halting Node1 to preserve data integrity
Oct 17 10:10:08 Node1 vmunix: Reason: LVM daemon failed
```

It is officially unsupported to change the TCP settings on a Serviceguard system. The specific ndd(1M) parameters involved here are tcp_keepalive_interval and tcp_ip_abort_interval. The same applies for the 10.X nettune(1M) parameters tcp_keepstart, tcp_keepfreq and tcp_keepstop. The reason is that Serviceguard needs all ports to behave the same way, and if ndd(1M) is run after starting the cluster, then some ports will use the original TCP behavior, while all other ports established after running ndd(1M) use the new behavior. It has been found that usually no problems arise if the ndd(1M) tuning is done before the cluster is started, but nevertheless, this is neither tested nor supported.

Serviceguard Command Problems

Sometimes Serviceguard commands fail and log messages that indicate that either the node is not configured into the cluster, that the binary configuration file misses or other basic problems. The commands that are usually affected by this are:

cmapplyconf
cmcheckconf
cmcp
cmdeleteconf
cmexec
cmgetconf
cmhaltcl
cmhaltnode
cmhaltpkg
cmmodpkg
cmquerycl
cmruncl
cmrunnode
cmrunpkg
cmviewcl
cmviewconf

Symptoms

Typical examples:

```
# cmviewcl
cmviewcl : Cannot view the cluster configuration.
Either this node is not configured in a cluster, user doesn't have
access to view the cluster configuration, or there is some obstacle
to viewing the configuration. Check the syslog file for more information.
For a list of possible causes, see the Serviceguard manual for cmviewcl.

# cmviewconf
cmviewconf: Unable to get cluster configuration information.
Unable to open communications to configuration daemon: Connection refused
Unable to connect to configuration database.

# cmviewconf
cmviewconf: Either binary file does not exist, or the user doesn't
have access to view the cluster configuration.

# cmhaltpkg pkgA
cmhaltpkg : Unable to open handle to local cluster
Either no cluster configuration file exists, the file is corrupted,
cmclconfd is unable to run, or user root on node nero
doesn't have access to view the configuration.

# cmviewcl
```

```
CLUSTER STATUS
alwayson up
Failed to get dlm configuration.
```

The above mentioned commands have in common that they access the Serviceguard configuration daemon cmclconfd to collect information for them. Basically if the Serviceguard commands do not get a reply from the daemon they will log messages similar to those above. There are numerous causes why a reply is not returned to the command. Ruling them out one after another will usually resolve the problem.

If you have a problem with one of those commands you can follow the steps bellow, to determine the reason for that.

1. Do you try to run a Serviceguard command on a node **not** having Servicegurad configured and that accesses a node that **is** running in a cluster? E.g. you run cmcheckconf to add a node that is not currently member of the cluster and you run the cmcheckconf on the node that is to be added? For Serviceguard A.11.16 and later you should run the command on the node that has Serviceguard already configured. Otherwise you would need to change the Role Based Access Policies in the cluster.ascii of the existing cluster to allow external nodes to modify the configuration.

2. /usr/lbin/cmclconfd exists and has appropriate execution rights. It's cksum and what string match what is documented (e.g. in patch texts for Serviceguard).

   ```
   # ls -l /usr/lbin/cmclconfd
   -r-xr--r-- 1 bin bin 3725848 Mar 14 2005 /usr/lbin/cmclconfd
   ```

3. The hacl-cfg/tcp and hacl-cfg/udp ports are listed in /etc/services. In a NIS environment the command "ypcat services" list the ports.

   ```
   # grep hacl-cfg /etc/services
   hacl-cfg 5302/tcp # HA Cluster TCP configuration
   hacl-cfg 5302/udp # HA Cluster UDP configuration
   ```

4. The hacl-cfg/tcp and hacl-cfg/udp ports are listed in /etc/inetd.conf on HP-UX as

   ```
   hacl-cfg dgram udp wait root /usr/lbin/cmclconfd cmclconfd -p
   hacl-cfg stream tcp nowait root /usr/lbin/cmclconfd cmclconfd -c
   ```

5. inetd is running and it registered the ports correctly when it was restarted last

   ```
   # grep hacl-cfg /var/adm/syslog/syslog.log
   ```

```
Aug 22 14:25:30 nero inetd[980]: hacl-cfg/udp: Added service, server
/usr/lbin/cmclconfd
Aug 22 14:25:30 nero inetd[980]: hacl-cfg/tcp: Added service, server
/usr/lbin/cmclconfd
```

Netstat -an shows that inetd is listening on hacl-cfg/tcp.

```
# netstat -an | grep 5302 | grep LISTEN
tcp 0 0 *.5302 *.* LISTEN
```

6. Make sure that /var/adm/inetd.sec does not deny access for cluster nodes to hacl-cfg ports.

7. Enable inetd connection logging (inetd -l) to verify that a local Serviceguard command (e.g. cmviewconf) connects to inetd which in turn starts the cmclconfd server process.

```
# tail -f /var/adm/syslog.log &
# inetd -l
# Apr 20 14:57:14 nero inetd[1189]: Connection logging enabled

# cmviewconf > /dev/null 2>&1
Apr 20 14:57:47 nero inetd[1395]: hacl-cfg/tcp: Connection from
localhost (127.0.0.1) at Fri Apr 20 14:57:47 2007
Apr 20 14:57:48 nero inetd[1396]: ident/tcp: Connection from localhost
(127.0.0.1) at Fri Apr 20 14:57:47 2007
```

If there are no messages for connection logging, check the next step or consider restarting inetd.

8. Firewall software (like IP Filter) must not disallow access to hacl-cfg ports from and to all other cluster nodes on ***all IP addresses the cluster nodes can potentially talk on***.

9. Make sure to list all cluster IP addresses in /etc/hosts. Make sure the cluster IP addresses are listed at the top of the file. Use an /etc/nsswitch.conf file of the form:

```
hosts:              files              [NOTFOUND=continue]              dns
```

10. For Serviceguard versions SG A.11.15 and earlier: Make sure either /etc/cmcluster/cmclnodelist or if this file does not exist $HOME/.rhosts contain the IP addresses of all cluster nodes and of all subnets the nodes can potentially talk on (not only those configured in the cluster binary /etc/cmcluster/cmclconfig). **Make sure these Serviceguard versions use most recent patch levels.** If you use $HOME/.rhosts make

sure to list the cluster IP addresses at the top of the file.

11. For Serviceguard versions SG A.11.16 and later:

- If there is no /etc/cmcluster/cmclconfig file: Make sure /etc/cmcluster/cmclnodelist contains the IP addresses of all cluster nodes and of all subnets the cluster nodes can potentially talk on.
- If there is /etc/cmcluster/cmclconfig already: Make sure /etc/hosts resolves the IP addresses of all cluster nodes and of all subnets the cluster nodes can potentially talk on. Also make sure that each of these IP addresses has an alias that matches the hostname of the host that owns the IP. Below is an example /etc/hosts file:

```
127.0.0.1       localhost       loopback
16.53.34.95     Node1.bgr.hp.com Node1
16.53.34.100    Node2.bgr.hp.com Node2
```

12. In recent versions of Serviceguard, cmclconfd -c (and cmomd) makes use of the identd service. Make sure access to the port 113/tcp is possible (listed as identd or auth port in /etc/services or NIS). To verify that identd works correctly, use the following test

1) Choose a connection from netstat –an

```
# netstat -an | more
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address
(state)
[..]
tcp        0      0  127.0.0.1.60934        127.0.0.1.5304
ESTABLISHED
```

2) telnet to the identd port and enter the two port numbers, separated by comma.

```
# telnet localhost 113
Trying...
Connected to localhost.
Escape character is '^]'.
60934, 5304           <---- enter port numbers here
60934 , 5304 : USERID : UNIX :root   <----- identd’s reply
Connection closed by foreign host.
```

To disabled identd usage on HP-UX add the '-i' option to the line of cmclconfd -c and cmomd in /etc/inetd.conf on all cluster nodes and run inetd -c.

13. For versions of Serviceguard running on HP-UX 11.11 that make use of the security enhancements, make sure you have version 2.7.4 or later of identd installed. Check by doing:

```
# what /usr/lbin/identd
usr/lbin/identd:
$Revision identd 2.7.4 (PHNE_26305) $
```

If the version is not sufficient you need to update to a later version of sendmail.Also make sure that you run ARPA patch PHNE_31247 for HPUX 11.11 or later or PHNE_24715 on HPUX 11.00.

14. If the Strong Random Number Generator is installed make sure you run version B.11.11.07 or later.

```
# swlist -l bundle | grep KRNG
KRNG11i B.11.11.09 HP-UX 11.11 Strong Random Number Generator
```

Also make sure the /dev/random and /dev/unrandom device files have a major number matching the "rng" driver:

```
# ls -l /dev/*random
cr--r--r-- 1 bin bin 62 0x000000 Nov 19 18:09 /dev/random
cr--r--r-- 1 bin bin 62 0x000001 Nov 19 18:09 /dev/urandom
# lsdev | grep 62
62 -1 rng pseudo
```

15.

a. Make sure that the number of Serviceguard commands running in parallel is not too high. Often users run cmviewcl in shell scripts to automate status monitoring. This can lead to problems when the requests cannot be answered quickly enough. You can check this by determining how long **cmclconfd -p** (not the ones with -c!) is running already (use ps -ef). If cmclconfd -p runs for a long time already (days or even weeks) this is an indicaton that many Serviceguard commands are executed on the cluster (not only the local node) in parallel. Also high CPU usage of cmclconfd -p and/or cmcld could be an indicator for this (run top(1m) to verify). Read HA Products Newsletter #50 (HP internal), 1st article. Also read HA Products Newsletter #59 (HP internal), 3rd article, for a known problem caused by Openview Operations agents.

b. If the CPU usage of cmclconfd -p is low and there is no indication of many SERVICEGUARD commands being executed, but cmclconfd -p already runs for a long time, it might be that cmclconfd is hung. To kill it and to get a core file of cmclconfd, kill it with signal SIGABRT (kill -SIGABRT <pid_of_cmclconfd_-p>). cmclconfd would automatically be restarted by the next Serviceguard command requesting it.

16. Read and adhere to the Special Installation Instructions of the Serviceguard patch you are using.

Cluster Quorum to Prevent Split-Brain Syndrome (See the section titled “Quorum Rules and Cluster Arbitration Device”)

In general, the algorithm for cluster re-formation requires a **cluster quorum** of a strict majority (that is, more than 50%) of the nodes previously running. If both halves (exactly 50%) of a previously running cluster were allowed to re-form, there would be a **split-brain** situation in which two instances of the same cluster were running. In a split-brain scenario, different incarnations of an application could end up simultaneously accessing the same disks. Serviceguard’s quorum requirement is designed to prevent a split-brain situation.

Cluster Lock

The prevention of split-brain syndrome is guaranteed by the use of a tie-breaker to choose between the two equal-sized node groups, allowing one group to form the cluster and forcing the other group to shut down. This tie-breaker is known as a **cluster lock**. The cluster lock is implemented by means of a **lock disk, lock LUN** or a **quorum server**. The cluster lock is used as a tie-breaker only for situations in which a running cluster fails and, as Serviceguard attempts to form a new cluster, the cluster is split into two sub-clusters of equal size. If you have a two node cluster and the communications are lost between these two nodes, the node that obtains the cluster lock will take over the cluster and the other node will halt or perform a TOC. Without a cluster lock, a failure of either node in the cluster will cause the other node, and therefore the cluster, to halt. Note also that if the cluster lock fails during an attempt to acquire it, the cluster will halt.

Lock Requirements

The following table gives an overview which type of cluster has to or must not use a cluster lock and which type of cluster lock is supported.

Type of Cluster	**Cluster lock**	**Type of Cluster Lock**	
		Lock Disk	**Quorum Server**
1 node cluster	Not required	N/A	N/A
2 node cluster	Required	Ok	Ok
3 and 4 node cluster	Recommended	Ok	Ok
> 4 node cluster	SCSI limited to 4 controllers, so no VG	Not Ok	Ok

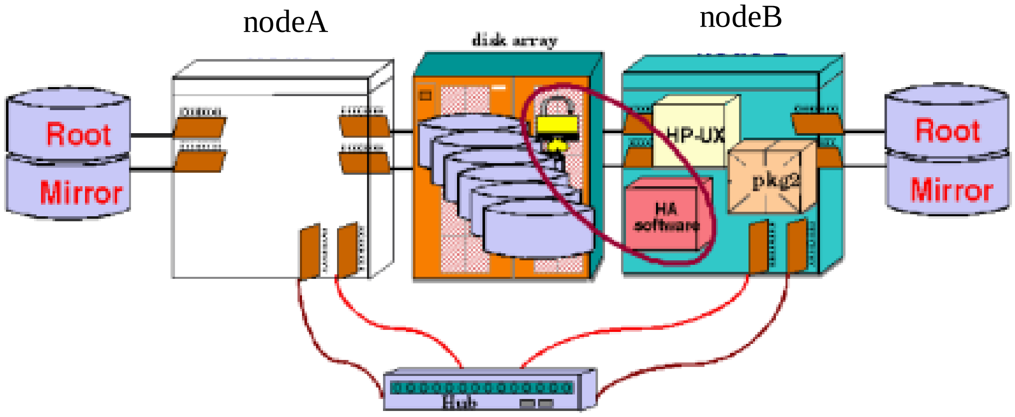
You may consider using no cluster lock with configurations of three or more nodes, although the decision should be affected by the fact that any cluster may require tie-breaking.

For Lock Disks you can choose between options—a single or dual lock disk—for further details refer to the Managing Serviceguard manual.

Use of an LVM Lock Disk as the Cluster Lock

Specifying a Lock Disk
The lock must be accessible to all nodes and must be powered separately from the nodes
To create a lock disk, enter the lock disk information following the cluster name. The lock disk must be in an LVM volume group that is accessible to all the nodes in the cluster.

Lock Disk Operation
When a node obtains the cluster lock, this area is marked so that other nodes will recognize the lock as "taken."
The lock disk is not dedicated for use as the cluster lock; the disk can be employed as part of a normal volume group with user data on it. The cluster lock volume group and physical volume names are identified in the cluster configuration file.
The operation of the lock disk is shown in Figure below.



Setting up the cluster lock disk
The default FIRST_CLUSTER_LOCK_VG and FIRST_CLUSTER_LOCK_PV supplied in the ASCII template (See "Useful Procedures and Commands", Creating configuration templates) created with cmquerycl are the volume group and physical volume name of a disk connected to all cluster nodes; if there is more than one, the disk is chosen on the basis of minimum failover time calculations. You should ensure that this disk meets your power wiring requirements. If necessary, choose a disk powered by a circuit which powers *fewer* than half the nodes in the cluster.

To display the failover times of disks, use the cmquerycl command, specifying all the nodes in the cluster.The output of the command lists the disks connected to each node together with the re-formation time
associated with each.

```
root@Node1:/# cmquerycl -v -n Node1 -n Node2
```

cmquerycl will not print out the re-formation time for a volume group that currently belongs to a cluster. If you want cmquerycl to print the re-formation time for a volume group, run vgchange -c n <*vg name*> to clear the cluster ID from the volume group. After you are done, do not forget to run vgchange -c y *vgname* to re-write the cluster ID back to the volume group; for example:
vgchange -c y /dev/vglock
If your configuration requires you to configure a second cluster lock, enter the following parameters in the cluster configuration file:

```
SECOND_CLUSTER_LOCK_VG /dev/volume-group
SECOND_CLUSTER_LOCK_PV /dev/dsk/block-special-file
```

where the */dev/volume-group* is the name of the second volume group and *block-special-file* is the physical volume name of a lock disk in the chosen volume group. These lines should be added for each node; for example:

```
SECOND_CLUSTER_LOCK_VG /dev/vglock
SECOND_CLUSTER_LOCK_PV /dev/dsk/c4t0d0
```

or (if using agile addressing;):

```
SECOND_CLUSTER_LOCK_VG /dev/vglock
SECOND_CLUSTER_LOCK_PV /dev/disk/disk100
```

Showing Cluster Lock Disk Status

With the Serviceguard version 11.17 comes new functionality to the cmviewcl command. It shows the status of the cluster lock volume group.

```
root@Node1:/etc/cmcluster/sw# cmviewcl -v

CLUSTER        STATUS
swrecovery       up

  NODE           STATUS       STATE
  Node1       up           running

    Cluster_Lock_LVM:
    VOLUME_GROUP         PHYSICAL_VOLUME       STATUS
    /dev/vgspare         /dev/dsk/c5t8d0       down

    …

  NODE           STATUS       STATE
  Node2       up           running

    Cluster_Lock_LVM:
    VOLUME_GROUP         PHYSICAL_VOLUME       STATUS
    /dev/vgspare         /dev/dsk/c2t8d0       down
```

…

Backing Up Cluster Lock Disk Information

After you configure the cluster and create the cluster lock volume group and physical volume, you should create a backup of the volume group configuration data on each lock volume group.

```
# vgcfgbackup <vg_lock>
```

Replacing a failed Cluster Lock Disk

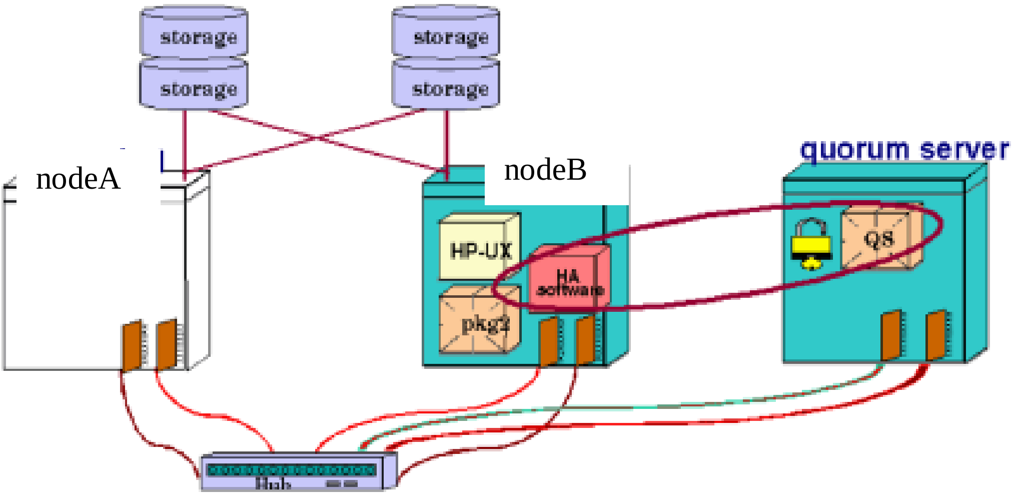
See ”Useful Procedures and Commands”, Reconfiguring a Cluster

Use of the Quorum Server as the Cluster Lock

A quorum server can be used in clusters of any size or for multiple clusters.

The quorum server process runs on a machine *outside of the cluster for which it is providing quorum services*. The quorum server listens to connection requests from the Serviceguard nodes on a known port. The server maintains a special area in memory for each cluster, and when a node obtains the cluster lock, this area is marked so that other nodes will recognize the lock as “taken.” If communications are lost between two equal-sized groups of nodes, the group that obtains the lock from the Quorum Server will take over the cluster and the other nodes will perform a TOC. Without a cluster lock, a failure of either group of nodes will cause the other group, and therefore the cluster, to halt. Note also that if the quorum server is not available during an attempt to access it, the cluster will halt. The operation of the quorum server is shown in the Figure below.

Quorum Server Operation



Setting up the Quorum Server

The quorum server software, must be installed on a system other than the nodes on which your cluster will be running. The actual Quorum Server Version is 3.0

NOTE It is recommended that this node is in the same subnet as the clusters for which it is providing services. This will help prevent any network delays which could affect quorum server

operation. If you use a different subnet, you may experience network delays which may cause quorum server timeouts. To prevent these timeouts, you can use the *QS_TIMEOUT_EXTENSION* parameter in the cluster configuration file to increase the quorum server timeout interval.
The Quorum Server 3.0 allows cluster nodes to communicate with the QS on an alternate subnet.
For more information please refer to: http://docs.hp.com/en/B8467-90041/B8467-90041.pdf

If the network used to connect to the quorum server is a cluster heartbeat network, ensure that at least one other network is also a heartbeat network so that both quorum server and heartbeat communication are not likely to fail at the same time.

Installing the Quorum Server
The quorum server executable file, qs, is installed in the /usr/lbin directory. When the installation is complete, you need to create an authorization file on the server where the QS will be running to allow specific host systems to obtain quorum services. The *required* pathname for this file is /etc/cmcluster/qs_authfile. Add to the file the names of all cluster nodes that will access cluster services from this quorum server. Use one line per node, as in the following example:
Node1.localdomain.com
Node2.localdomain.com

To allow access by all nodes, enter the plus character (+) on its own line.

Specifying a Quorum Server
To specify a quorum server instead of a lock disk, use the -q option of the cmquerycl command, specifying a Quorum Server host server. Example:

```
root@Node1:/ # cmquerycl -n Node1 -n Node2 -q qshost
```

Running the Quorum Server
The quorum server must be running during the following cluster operations:

- when the cmquerycl command is issued.
- when the cmapplyconf command is issued.
- when there is a cluster re-formation.

By default, quorum server run-time messages go to stdout and stderr. It is suggested that you create a directory /var/adm/qs, then redirect stdout and stderr to a file in this directory, for example, /var/adm/qs/qs.log.
On a single system, configure the quorum server to start up any time the system on which it is installed restarts or reboots. Do this by creating an entry like the following in the /etc/inittab file:

```
qs:345:respawn:/usr/lbin/qs >> /var/adm/qs/qs.log 2>&1
```

Start the quorum server as follows:

```
root@QS:/ # init q
```

When the command is complete, the prompt appears. Verify the quorum server is running by checking the qs.log file.

```
root@QS:/ # cat /var/adm/qs/qs.log
```

The log should contain entries like the following indicating the quorum server has started:

```
Oct 04 12:25:06:0:main:Starting Quorum Server
Oct 04 12:25:09:0:main:Server is up and waiting for connections at port 1238
```

Replacing a Failed Quorum Server System

When a quorum server fails or becomes unavailable to the clusters it is providing quorum services for, this will not cause a failure on any cluster. However, the loss of the quorum server increases the vulnerability of the client clusters to TOC if split-brain failure occurs. Use the following procedure to replace a defective quorum server system. If you use this procedure, you do not need to change the configuration of any cluster nodes.

1. Remove the old quorum server system from the network.
2. Set up the new system and configure it with the old quorum server's IP address and hostname.
3. Install and configure the quorum server software on the new system.Be sure to include in the new QS authorization file (/etc/cmcluster/qs_authfile) all of the nodes that were configured for the old quorum server. Refer to the qs(1) man page for details about configuring the QS authorization file.
4. Start the quorum server as follows:
   - Edit the /etc/inittab file to add the quorum server entries.
   - Use the init q command to run the quorum server.

   Refer to the qs(1) man page for more details.
5. All nodes in all clusters that were using the old quorum server will connect to the new quorum server. Use the cmviewcl -v command from any cluster that is using the quorum server to verify that the nodes in that cluster have connected to the QS.
6. The quorum server log file on the new quorum server will show a log message like the following for each cluster that uses the quorum server: Request for lock /sg/<*ClusterName*> succeeded. New lock owners: N1, N2
7. To check that the quorum server has been correctly configured and to verify the connectivity of a node to the quorum server, you can execute the following command from your cluster nodes as follows:

```
root@Node1:/ # cmquerycl -q qhost -n Node1 -n Node2 ...
```

The command will output an error message if the specified nodes cannot communicate with the quorum server.
NOTE While the old quorum server is down and the new one is being set up,
these things can happen:

- These three commands will not work:

cmquerycl -q, cmapplyconf -C, and cmcheckconf -C

• If there is a node or network failure that creates a 50-50 membership split, the quorum server will not be available as a tie-breaker, and the cluster will fail

Types of Volume Managers

Serviceguard allows a choice of volume managers for data storage:
• HP-UX Logical Volume Manager (LVM) and (optionally)
Mirrordisk/UX
• VERITAS Volume Manager for HP-UX (VxVM)
• VERITAS Cluster Volume Manager for HP-UX (CVM)

Supported Volume Managers

For a list of supported volume managers on HPUX and Serviceguard versions, refer to the **HP Serviceguard Solutions Storage Support Matrix** at http://www.hp.com/go/sgstoragesupport.

HP-UX Logical Volume Manager

Logical Volume Manager (LVM) is the default storage management product on HP-UX.
The Serviceguard cluster lock disk is configured using a disk configured in an LVM volume group. LVM continues to be supported on HP-UX single systems and on Serviceguard clusters.

LVM VG activation modes

Volume groups that are private to a particular server activate using the *standard* activation mode using ‘vgchange –a y <vg>’.
In order to protect the data on volume groups that may be shuffled amongst Serviceguard nodes when a failover package (one that runs only on one node at a time) is moved, *exclusive* activation mode, implemented using ‘vgchange –a e <vg>’ is required. For those volume groups activated on multiple systems simultaneously to implement SGeRAC raw-data volume access, *shared* activation mode is implemented using ‘vgchange –a s <vg>’.
To set activation modes:
Standard: vgchange –c n <vg>
Exclusive: cmapplyconf where VG is identified in the cluster ASCII file with a VOLUME_GROUP parameter or vgchange –c y <vg> (NOTE: cmcld and cmlvmd must be running). The latter method is sometimes used if cmapplyconf rejects the disk type.
Shared: With the cluster running: ‘vgchange –c y –S y <vg>’.

Co-mounting file systems

Logical volumes whose volume groups are activated in standard mode or shared mode cannot be mounted on multiple systems concurrently without risking a system panic because HPUX LVM provides no means of informing other systems of in-memory file system inode table modifications. Serviceguard with VxVM-based CFS is the only supported means to concurrently mount a file system to multiple servers.

VERITAS Volume Manager (VxVM)

VxVM provides us the basic functionality to manage the physical disk space in order to create or/and to divide it into virtual disk volumes which are transparently presented as a physical devices used by the operating system and different applications.

VxVM can be used in clusters that:
- are of any size, up to 16 nodes.
- require a fast cluster startup time.
- do not require shared storage group activation. (required with CFS)
- do not have all nodes cabled to all disks. (required with CFS)
- need to use software RAID mirroring or striped mirroring.
- have multiple heartbeat subnets configured.

Cluster Volume Manager

The CVM allows VxVM disk groups to be activated on many nodes at the same time, which also gives the opportunity to choose between different modes. So it would be possible to allow read-only access to a node while another has a read-write access.

CVM is bundled with SGeRAC and beginning with A.11.17, certain Serviceguard Storage Management Suite (SMS) bundles that include cluster file system enablement.

CVM imposes a “Uniform Shared Storage” model. All systems must be connected to the same disk sets for a given disk group. Any system unable to see the entire set of physical disks for a given disk group cannot import the group. If a node loses contact with a specific disk, CVM excludes the node from participating in the use of that disk.

CVM 3.5 supported only single heartbeat network with standby LAN card(s) and single heartbeat network with APA and worked with SGeRAC.
Dual (multiple) heartbeat networks and single heartbeat network with standby LAN card(s) are the minimum recommended configurations for CVM 4.1, used by SMS A.01.0x for A.11.17. CVM 5.0 is matched to A.11.18 and SMS A.02.00.
VxFS5.0 features:
- Extent-based space management that maps files up to 1 terabyte in size
- Fast recovery from system crashes using the intent log to track recent file system
metadata updates
- Online administration that allows file systems to be extended and defragmented
while they are in use

SLVM Single Node Online volume Re-configuration

Rarely referenced in customer support cases, the SLVM Single Node Online volume Re-configuration (SNOR) feature allows changing the configuration of an active shared volume group in a cluster by deactivating the VG on all but one node, and on that node, changing the volume groups activation mode to exclusive (vgchange –a e –x ..) so that LVM modifications can occur. After modifications, the activation mode is changed back to shared-mode (vgchange –a s –x ..).and other nodes are then allowed to start their packages that activate the VG in shared mode

- More information about SNOR you can find at: SLVM Online Volume Reconfiguration http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01914684/c01914684.pdf

Comparison of Volume Managers

The following table summarizes some of the advantages and disadvantages of the volume managers that are currently available.

Product	Advantages	Tradeoffs
Logical Volume Manager (LVM)	• Software is provided with all versions of HP-UX. • Provides up to 3-way mirroring using optional Mirrordisk/UX software. • Dynamic multipathing (DMP) is active by default as of HP-UX 11i v3. • Supports exclusive activation as well as read-only activation from multiple nodes • Can be used to configure a cluster lock disk • Supports multiple heartbeat subnets; the one with the faster failover time is used to re-form the cluster.	• Lacks flexibility and extended features of VxVM
Mirrordisk/UX	• Software mirroring • free on 11i v3	• Lacks extended features of VxVM/CVM
Shared Logical Volume Manager (SLVM)	• Provided free with SGeRAC for multi-node access to RAC data • Supports up to 16 nodes in shared read/write mode for each cluster • Supports exclusive activation • Supports multiple heartbeat subnets. • Online node configuration with activated shared volume groups (using specific SLVM kernel and Serviceguard revisions)	• Lacks the flexibility and extended features of VxVM. • Limited mirroring support
Base-VxVM	• Software is supplied free with HP-UX 11i releases. • Java-based administration through graphical user interface. • Striping (RAID-0) support. • Concatenation. • Online resizing of volumes. • Supports multiple heartbeat subnets.	• Cannot be used for a cluster lock • root/boot disk supported only on VxVM 3.5 or later, on HP-UX 11i • Supports only exclusive read or write activation • Package delays are possible, due to lengthy vxdg import at the time the package is started or failed over
VERITAS Volume Manager Full VxVM product B9116AA (VxVM 3.5) B9116BA (11.31 VxVM 4.1) B9116CA (11.23 VxVM 5.0) B9116CB (11.31 VxVM 5.00.01-.03)	• Disk group configuration from any node. • DMP for active/active storage devices. • Supports exclusive activation. • Hot relocation and unrelocation of failed subdisks	• Requires purchase of additional license • Cannot be used for a cluster lock • Using the disk as a root/boot disk is only supported for VxVM 3.5 or later, when installed on HP-UX 11i.

B9116DB (11.31 VxVM 5.01.01) B9116EB (11.31 VxVM 5.10)	• Supports up to 32 plexes per volume • RAID 1+0 mirrored stripes • RAID 1 mirroring • RAID 5 • RAID 0+1 striped mirrors • Supports multiple heartbeat subnets, which could reduce cluster reformation time.	• Does not support activation on multiple nodes in either shared mode or read-only mode • May cause delay at package startup time due to lengthy vxdg import
VERITAS Cluster Volume Manager – B9117AA (CVM 3.5) B9117BA (11.23 CVM 4.1-4.1.01) B9117CA (11.23 CVM 5.0 SMS A.02.00-.01) B9117CB (11.31 CVM 5.0 SMS A.02.01) B9117DB (11.31 CVM 5.0.1 SMS A.03.00) B9117EB (11.31 CVM 5.1 SMS A.04.00)	• Online volume propagation propagation. • Supports cluster shareable disk groups. • Package startup time is faster than with VxVM. • Supports shared activation. • Supports true exclusive activation. • Supports activation in different modes on different nodes at the same time • CVM version 4.1 supports the VERITAS Cluster File System (CFS)	• Disk groups must be configured on a master node • Full CVM can only be used with up to 8 cluster nodes. CFS can also been used with up to 8 nodes. • Cluster startup may be slower than with VxVM • Requires purchase of additional license • No support for striped mirrors or RAID 5 • Version 3.5 supports only a single heartbeat subnet (Version 4.1 supports more than one heartbeat) • CVM requires all nodes to have connectivity to the shared disk groups

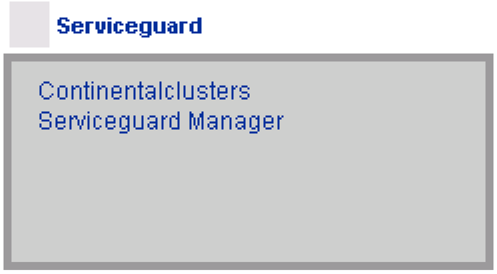
Recommendations and further Informations

Since Serviceguard for RAC configurations may be complex to configure and maintain, **it is strongly recommended that you use Hewlett-Packard's high availability consulting services to ensure a smooth installation and rollout.** Please contact your HP representative to inquire about high availability consulting. In addition, you should work with your HP representative to ensure that you have the latest firmware revisions for disk drives, disk controllers, LAN controllers, and other hardware.
This is an extract of the release notes and release notes are subject to change without further notice. Please check http://www.hp.com/go/hpux-serviceguard-docs for the latest version of the Serviceguard Extension for RAC manual and release notes.

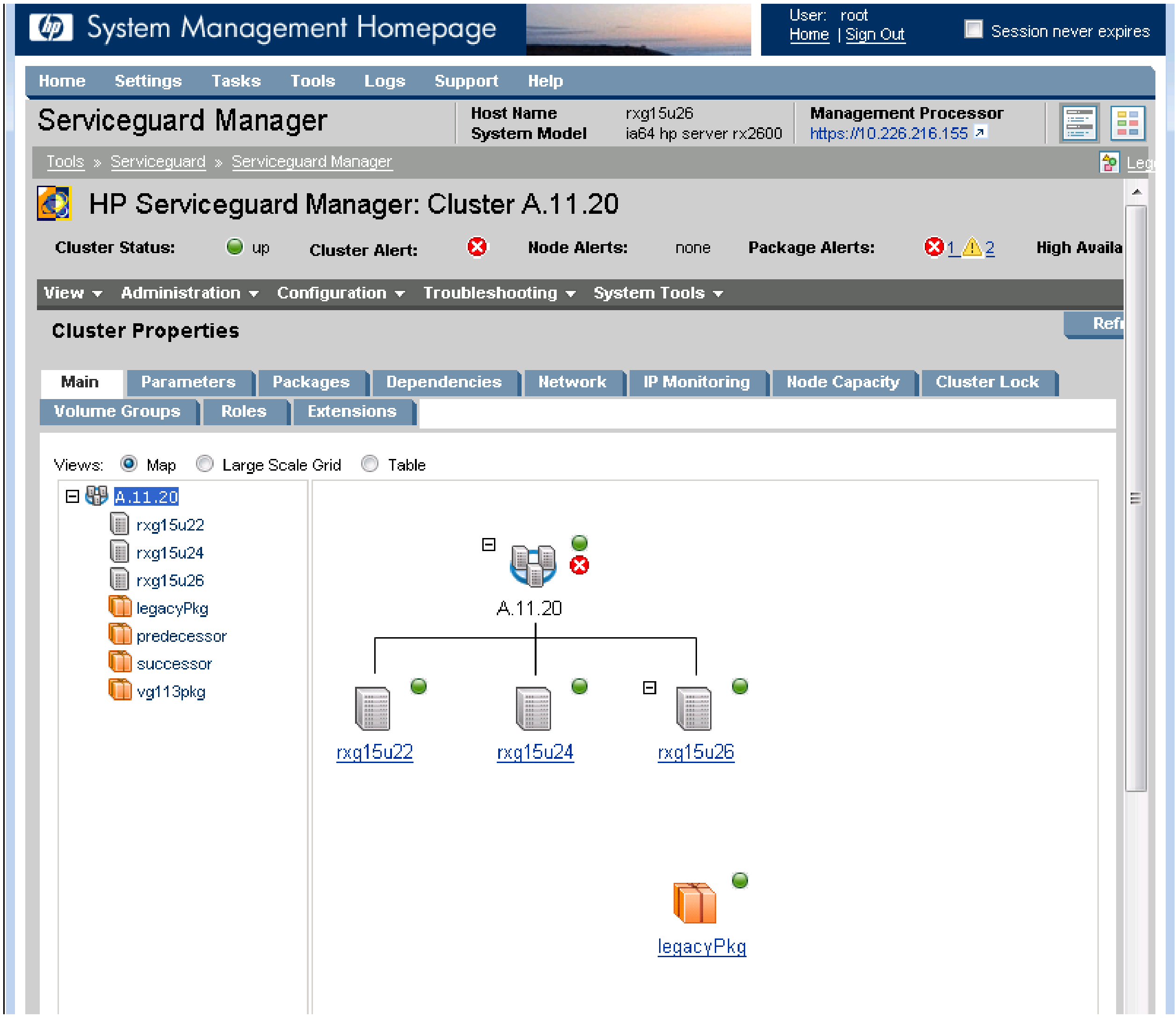
Serviceguard Manager

Serviceguard Manager is the graphical user interface for Serviceguard configuration and management. The current version of Serviceguard runs as an API to System Management Homepage – a system administration webpage.

To start Serviceguard Manager
Set `HPWS_APACHE_START=1` in `/etc/rc.config.d/hpws_apacheconf`
Read the file into the shell: `. /etc/rc.config.d/hpws_apacheconf`
Start web services: `/sbin/init.d/hpws_apache start`
Open a browser on the server network and load: http://<hostname>:2301
Select Tools (in the top bar) -> and at the bottom of the result, locate and click Serviceguard (see below):



Click on Serviceguard Manager.


System Management Homepage
User: root
Home | Sign Out
☐ Session never expires
Home Settings Tasks Tools Logs Support Help
Serviceguard Manager
Host Name rxg15u26
System Model ia64 hp server rx2600
Management Processor https://10.226.216.155
Tools » Serviceguard » Serviceguard Manager
HP Serviceguard Manager: Cluster A.11.20
Cluster Status: ● up
Cluster Alert:
Node Alerts: none
Package Alerts: 1 2
High Availa
View Administration Configuration Troubleshooting System Tools
Cluster Properties
Refr
Main Parameters Packages Dependencies Network IP Monitoring Node Capacity Cluster Lock
Volume Groups Roles Extensions
Views: ● Map ○ Large Scale Grid ○ Table
A.11.20
rxg15u22
rxg15u24
rxg15u26
legacyPkg
predecessor
successor
vg113pkg
A.11.20
rxg15u22
rxg15u24
rxg15u26
legacyPkg

From there you can do the following tasks:

- **Monitoring Clusters with Serviceguard Manager**
  You can see all the clusters the server can reach, or you can list specific clusters. You can also see all the unused nodes on the subnet - that is, all the Serviceguard nodes that are not currently configured in a cluster. Note that SMH has a timeout, unless the session is set to NEVER EXPIRE (upper right corner of the GUI page).

- **Administering Clusters with Serviceguard Manager**
  You can administer clusters, nodes, and packages if access control policies permit:
  - – Cluster: halt, run
  - – Cluster nodes: halt, run
  - – Package: halt, run, move from one node to another, reset node- and package-switching flags

- **Configuring Clusters with Serviceguard Manager**
  You can configure clusters and packages. You must have root (UID=0) access to the cluster nodes.

Since Serviceguard A.11.17.01, Serviceguard Manager is available in two forms, a standalone utility and the new SMH-based utility.

- **Old:** as an **independent** management application running on an HP-UX, Linux, or Windows system. This utility is not used on current software.

- **New:** as a **"plug-in"** to the System Management Homepage (SMH).
  SMH is a web-based graphical user interface (GUI) that replaces SAM as the system administration GUI as of HP-UX 11i v3 (but you can still run the SAM terminal)

Note: Future Development will be done for the "plug-in" version and no longer for the independent management application.

Check the latest Release Notes for your version of Serviceguard and the Supportmatrix for up-to-date information here:
http://www.docs.hp.com -> High Availability -> Serviceguard Manager

Serviceguard Management Application

Serviceguard Management Application is the old, independent form of Serviceguard Manager.

How to get it? (obsolete – not available with current versions of Serviceguard)

How to start it?

To start the Serviceguard management application

- on a Unix or Linux management station, use the *sgmgr* command. You can enter the options on the command line, or in a dialog box after the GUI opens. For command syntax and options, enter man sgmgr on the command line.
- on a Windows management station, double-click the icon on your desktop. To see or change the actual command used, right click the icon and choose Properties. See Help -> Troubleshooting for command syntax and options.

How to use it?

To open a saved "snapshot" cluster file, specify a filename with the .sgm extension; you must have view permission on the file and its directory.

To see "live" clusters, from a management station, connect to a Serviceguard node's Cluster Object Manager (COM) daemon. (The COM is automatically installed with Serviceguard.) This node becomes the session server. It goes out over its subnets, and establishes connections with the COMs on other Serviceguard nodes. The session server relays commands from the management station to the target nodes, and relays the target nodes' configuration and status data back to the management station. It also relays operation messages from the target nodes to the management station.

If the Session Server node is running Serviceguard versions A.11.13 through A.11.17, the Serviceguard Security Patch must be installed and enabled before the Session Server node can connect to cluster nodes running A.11.17.

To connect, you need to specify a valid username and password from the session server's /etc/passwd file. List the cluster or clusters you want to see. Click "unused nodes" to see nodes that are not currently configured into a cluster, but do have Serviceguard installed.
For the session server to get information from a cluster, the target cluster must allow it access. The target node will resolve the session server's hostname through /etc/hosts or DNS. Access method and non-root roles changed in Serviceguard Version A.11.16:

- In clusters with Serviceguard version A.11.16 and later, the cluster configuration file or a package configuration file, must have an Access Control Policy that specifies this triplet: the intended user, the COM server's hostname, and a role of at least Monitor.
- In earlier versions of Serviceguard, the /etc/cmcluster/cmclnodelist file must have this pair listed: COM server's host_node, and user root.

For more informations and also some screenshots, have a look in the Release Notes:
http://www.docs.hp.com -> High Availability -> Serviceguard Manager

Serviceguard Manager plug-in

Serviceguard Manager plug-in is the new form of Serviceguard Manager, a plug-in to the System Management Homepage (SMH). It's available with Serviceguard A.11.17.01 and later.

How to get it?
The plug-in version does not require installation; see the Serviceguard Release Notes for more information.
https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=B8325BA

How to start it?

To start the Serviceguard Manager plug-in in your web browser from the System Management Homepage, click on the link to Serviceguard Cluster or a particular cluster. Then select a cluster, node, or package, and use the drop-down menus below the "Serviceguard Manager" banner to navigate to the task you need to do**.**

Example for the URL of the System Management Homepage:
http://node1.domain.hp.com:2301

Patch information

Patch Requirements (external Documentation)

To find the required patches for Serviceguard, check the Release Notes of your Serviceguard version on http://www.hp.com/go/hpux-SG-docs; particularly the section named: *"Patches and Fixes in this Version"*

For required patches for the Serviceguard Storage Management Suite check here: http://www.hp.com/go/sgsms/patches

Download of Patches

To download individual patches go to http://patch-hub.corp.hp.com/wtec/catalog/ or the public HP Solution Center:
http://hp.com/go/hpsc → Patch Database → Find individual Patches

If the patch is older and a replacement was already released, you will see it in the result list.

ITRC Patch Set for Serviceguard

HP provides two additional services in the ITRC Patch Database which help you to patch your Serviceguard System:

1. *"run a patch assessment"*
   Login to the ITRC and go to:
   https://h20566.www2.hp.com/portal/site/hpsc/patch/home/ → run a patch assessment

   In the assessment profile you are able to specify *"Application specific patch sets"*.
   A patch set for Serviceguard is electible.
   For more details please refer to the documenation on the Patch Assessment page.

2. *"find HP-UX patches in patch set"*
   Login to the ITRC and go to:
   http://hp.com/go/hpsc → Patch Database → find HP-UX patches in a patch set
   Choose as Product Serviceguard, choose your OS-Version and your Patchstrategy and click on search. You can add the patches for your Serviceguard Version(s) to your

selected patch list.

HP-Internal Documentation

Recommended Patches for ACSL Products
http://haweb.ind.hp.com/Support/RecmdPatches.html

Serviceguard Daemons

• /usr/lbin/cmclconfd	—Serviceguard Configuration Daemon
• /usr/lbin/cmcld	—Serviceguard Cluster Daemon
• /usr/lbin/cmfileassistd	—Serviceguard File Management daemon
• /usr/lbin/cmlogd	—Serviceguard Syslog Log Daemon
• /usr/lbin/cmlvmd	—Cluster Logical Volume Manager Daemon
• /opt/cmom/lbin/cmomd	—Cluster Object Manager Daemon
• /usr/lbin/cmsnmpd	—Cluster SNMP subagent (optionally running)
• /usr/lbin/cmserviced	—Serviceguard Service Assistant Daemon
• /usr/lbin/qs	—Serviceguard Quorum Server Daemon
• /usr/lbin/cmnetd	—Serviceguard Network Manager daemon.
• /usr/lbin/cmvxd	—Serviceguard-to-Veritas

Each of these daemons logs to the `/var/adm/syslog/syslog.log` file
except for `/opt/cmom/lbin/cmomd`, which logs to
`/var/opt/cmom/cmomd.log`.
The quorum server runs outside the cluster. By default, it logs to the standard output, and it is suggested you redirect output to a file named `/var/adm/qs/qs.log`.

For more Informations about Logging see Section *"Troubleshooting"*.

CFS Components, Commands and Daemons

CFS Components

The HP Serviceguard Storage Management Suite (SMS) offers additional components for interfacing with the VERITAS Cluster File System. SMS documents are posted at:

- Storage Management Suite (Legacy Versions) (http://www.hp.com/go/hpux-SG-SMS-previous-docs)
- Storage Management Suite A.03.xx (http://www.hp.com/go/hpux-SG-SMS-A03-docs)
- HP Serviceguard Storage Management Suite A.04.xx (with 5.1 SP1 Veritas)

VERITAS CFS components operate directly over Ethernet networks that connect the nodes within a cluster. Redundant networks are required to avoid single points of failure.
The VERITAS CFS components are:

- GAB (Group Membership Services/Atomic Broadcast) – When VERITAS Cluster Volume Manager (CVM) 4.1 or VERITAS Cluster File System (CFS) is deployed as part of the Serviceguard Storage Managment Suite bundles, the file /etc/gabtab is automatically configured and maintained by Serviceguard. GAB provides membership and messaging for CVM and the CFS. GAB membership also provides orderly startup and shutdown of the cluster file system.

- LLT (Low Latency Transport) - When VERITAS CVM or CFS is deployed as part of the Serviceguard Storage Managment Suite bundles, the LLT files /etc/llthosts and /etc/llttab are automatically configured and maintained by Serviceguard. LLT provides kernel-to-kernel communications and monitors etwork communications for CFS.

For more informations about CVM please refer to the section "*Type of Volume Managers*".

CFS Commands

Requires selected HP Serviceguard Storage Management Suite Bundle

`cfscluster`	• Configure or unconfigure SG-CFS-pkg, the system multi-node package used for clusters that use the VERITAS Cluster File System. • Start or stop the CVM package for the CFS. • Get the status of the SG-CFS-pkg package.
`cfsdgadm`	• Display the status of CFS disk groups. • Add shared disk groups to a VERITAS Cluster File System CFS cluster configuration, or remove existing CFS disk groups from the configuration. Serviceguard automatically creates the multi-node package SG-CFS-DG-id# to regulate the disk groups. This package has a dependency on the SG-CFS-pkg created by cfscluster command. • Activate or de-activate the shared disk groups, clusterwide or on specified node(s).
`cfsmntadm`	Add, delete, modify, or set policy on mounted filesystems in a VERITAS Cluster File System (CFS) cluster.
`cfsmount` `cfsumount`	Mount or unmount a VERITAS Cluster File System.
`cmgetpkgenv`	Allows the VERITAS Cluster File System (CFS) admin packages to retrieve their configured values from the context of the package control script.

CFS Daemons

`vxfend`	the I/O fencing daemon	implements a quorum-type functionality for the VERITAS Cluster File System
`cmvxd`	Serviceguard-to-VERITAS Membership Coordination daemon. (Only present when VERITAS CFS is installed.)	coordinates the membership information between Serviceguard and VERITAS' Clustered File System product
`cmvxping`	Serviceguard-to-VERITAS	Activates certain subsystems of the

	Activation daemon. (Only present when VERITAS CFS is installed.)	VERITAS Clustered File System product

Serviceguard Related Files

Serviceguard uses a special file, /etc/cmcluster.conf, to define the locations for configuration and log files within the HP-UX filesystem.

Note: Do **not** edit this /etc/cmcluster.conf configuration file unless you need to cause debug logging to occur and after completing troubleshooting, revert to the standard format to prevent filling a file system with a huge log file.

The following locations are defined in the file:

```
################### cmcluster.conf ###############
# Highly Available Cluster file locations
# This file must not be edited
############################################################
SGCONF=/etc/cmcluster
SGSBIN=/usr/sbin
SGLBIN=/usr/lbin
SGLIB=/usr/lib
SGRUN=/var/adm/cmcluster
SGAUTOSTART=/etc/rc.config.d/cmcluster
SGFFLOC=/opt/cmcluster/cmff
CMSNMPD_LOG_FILE=/var/adm/SGsnmpsuba.log
```

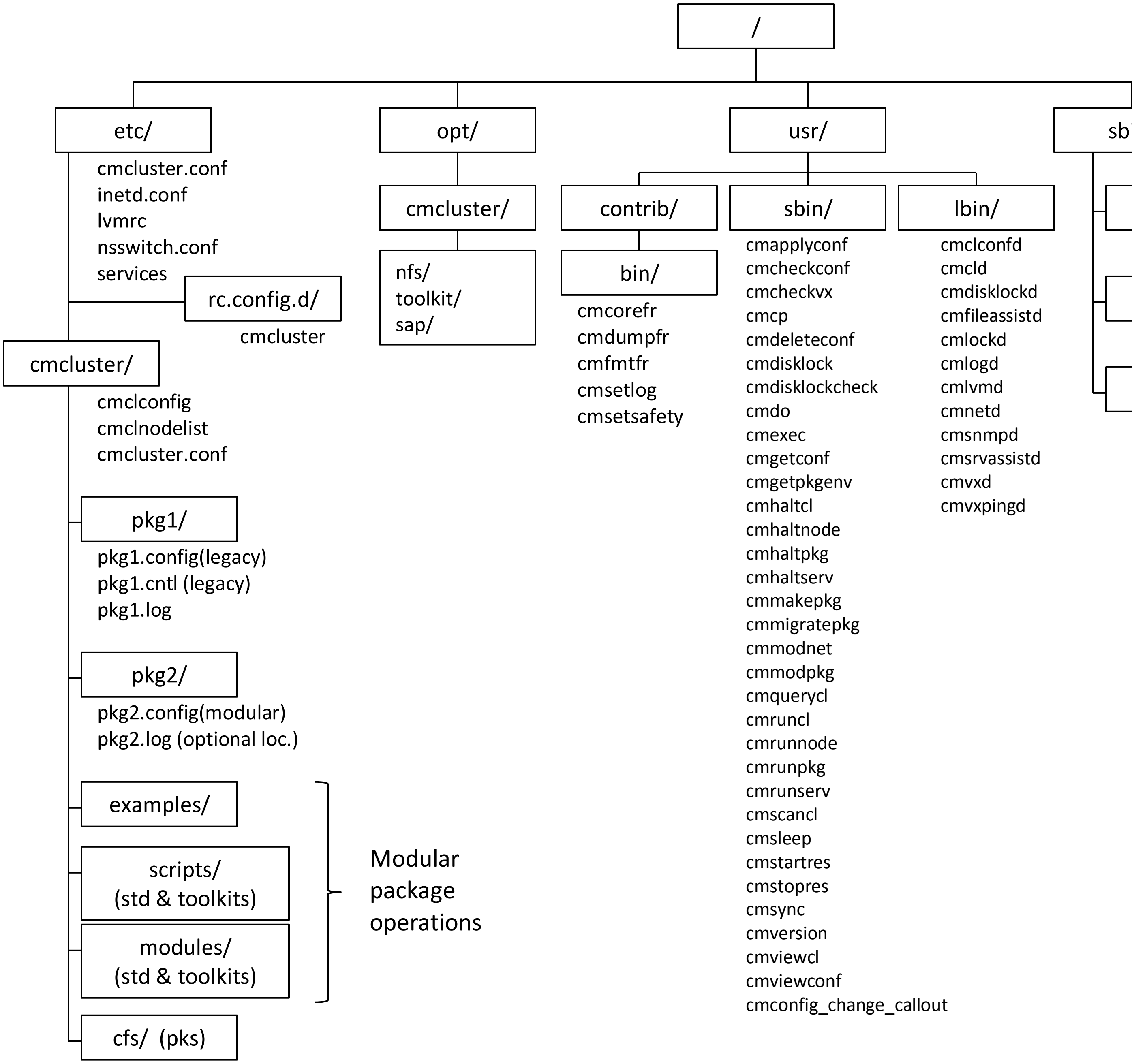
Only if for any reason these variables are not defined on your system, **then** source the file /etc/cmcluster.conf in your login profile for user root. For example, you can add this line to root's .profile file:
. /etc/cmcluster.conf

Debug Logging
Use the following link to learn how to set debug logging for various versions of Serviceguard commands and daemons:
http://teams3.sharepoint.hp.com/teams/esssupport/InsideESSSupport/InsideWTEC/HAProducts/Pages/sg_debug_logging.aspx

The following tree shows the location of common Serviceguard Files. Special files for SGeRAC, SGeSAP, HA-NFS, CFS, and others are not included.
/opt/cmcluster contains the various toolkit files, if installed.

For more informations about the files, please refer to
Managing Serviceguard, see Chapter 5: *Understanding Where Files Are Located* .
The document is downloadable from http://www.hp.com/go/hpux-serviceguard-docs

Serviceguard file location



Additional Information

External Technical Documentation

This URL is the starting point for all Documentation for High Availability Products:

http://www.hp.com/go/hpux-serviceguard-docs

Additionally a list of direct links for the most interesting parts of Serviceguard on HPUX:

- Manuals and Release Notes for ServiceGuard
  http://www.hp.com/go/hpux-SG-docs
- Manuals and Release Notes for Serviceguard Extension for Real Application Cluster
  http://www.hp.com/go/hpux-SGeRAC-docs
- Manuals and Release Notes for Serviceguard Extention for SAP
  http://www.hp.com/go/hpux-SGeSAP-docs
- Manuals and Release Notes for Serviceguard Manager
  http://www.hp.com/go/hpux-SGManager-docs
- Release Specific Information

  » HP-UX 11i v1 (must use URL embedded in this line)
  » HP-UX 11i v2
  » HP-UX 11i v3

HP-Internal Technical Documentation

- ACSL Division, Support Team
  http://haweb.ind.hp.com/Support (HP internal)
- HPUX HA WTEC
  http://teams3.sharepoint.hp.com/teams/esssupport/InsideESSSupport/InsideWTEC/HAProducts/Pages/Default.aspx (HP internal)
- HPUX HA WTEC Newsletter
  http://teams3.sharepoint.hp.com/teams/esssupport/InsideESSSupport/InsideWTEC/HA/Pages/Newsletter.aspx (HP internal)
- HPVR Assist Room (L2): **L2 UX Sys Interrupt**

External Training

http://www.hp.com/education/ -> HP-UX -> HA Curriculum path

Internal Training

- http://grow.hp.com (HP internal) - search for serviceguard
- ACSL High Availability Clusters Support and Training
  http://haweb.ind.hp.com/ATC/Training/training.html (HP internal)