SMW HA XC Administration Guide (S-2551)

Contents

Record of Revision

S-2551 Published May 2015 Supports the release of Cray SMW High Availability Extension for SLES 11 SP3 UP02.

About This Guide

An SMW HA system is a Cray XC system with two second-generation high-end SMWs (also called *rack-mount SMWs*) that run the SUSE Linux Enterprise High Availability (SLEHA) Extension and the Cray SMW High Availability Extension for SLES 11 SP3 release package, also called the *SMW HA package*.

The intended reader of this guide is a system administrator who is familiar with operating systems derived from UNIX.

For information about installing the SMW HA system, see *SMW HA XC Installation Guide (S-0044)*.

SMW HA Overview

The Cray System Management Workstation (SMW) High Availability (HA) system supports SMW failover. An SMW HA system is a Cray XC system with two second-generation high-end SMWs (also called *rack-mount SMWs*) that run the SUSE Linux Enterprise High Availability (SLEHA) Extension and the Cray SMW High Availability Extension (SLEHA) release package. The two SMWs must be installed and configured as specified in this guide.

The SMW failover feature provides improved reliability, availability, and serviceability (RAS) of the SMW, allowing the mainframe to operate correctly and at full speed. This feature adds SMW failover, fencing, health monitoring, and failover notification. Administrators can be notified of SMW software or hardware problems in real time and be able to react by manually shutting down nodes, or allowing the software to manage the problems. In the event of a hardware failure or `rsms` daemon failure, the software will fail over to the passive SMW node, which becomes the active node. The failed node, once repaired, can be returned to the configuration as the passive node.

The SUSE Pacemaker Cluster Resource Manager (CRM) provides administration and monitoring of the SMW HA system with either a command-line interface (`crm`) and a GUI (`crm_gui`). With this interface and associated commands, the SMW administrator can display cluster status, monitor the HSS daemons (configured as cluster resources), configure automatic failover notification by email, and customize the SMW failover thresholds for each resource.

The following topics describe the unique features of the SMW HA system:

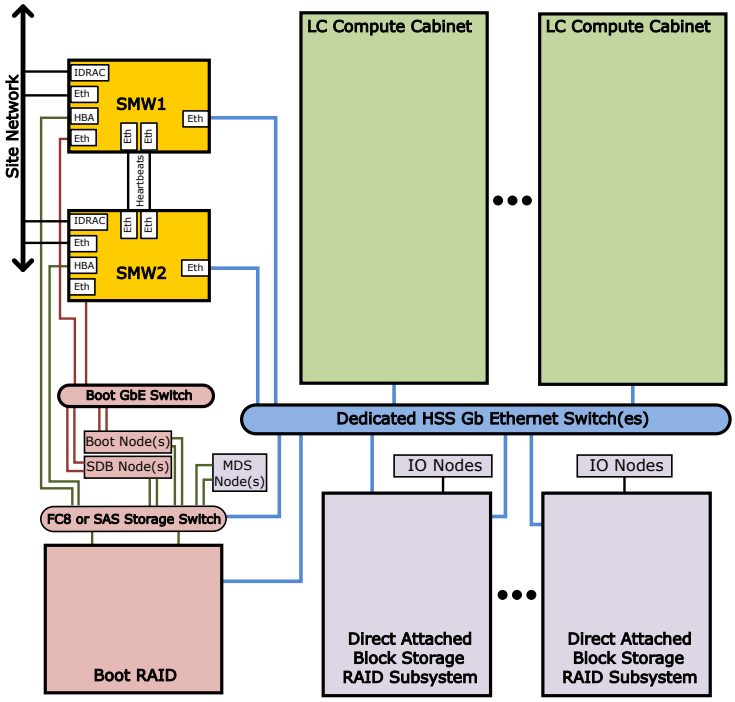
NOTE: The Pacemaker Cluster Resource Manager uses the term *node* to refer to a host in a CRM cluster. On an SMW HA system, a CRM node is an SMW, not a Cray XC compute or service node.

SMW Cluster Configuration

Both SMWs are connected to the boot RAID, and are connected to each other with heartbeat cables between the `eth2` and `eth4` ports on each SMW. The heartbeat connection monitors the health of the cluster. In addition, each SMW is connected to the boot RAID (through FC or SAS cards), to the site network through `eth0`, to the HSS network through `eth1`, and to the boot node through `eth3`. (For more information, see *Network Connections for an SMW HA System*.) An Integrated Dell™ Remote Access Controller (iDRAC) is required on both SMWs.

The following figure shows the major connections between components in an SMW HA system.

Figure 1. SMW HA Hardware Components for a Cray XC System



In a Cray SMW HA cluster, the two SMWs are configured in an active/passive configuration. This configuration lets the passive node take over the SMW functions if a software or hardware fault occurs on the active node. All HSS daemons run on the active SMW. (An additional `stonith` daemon, which monitors SMW health, runs on both SMWs.) At failover, all daemons move to the passive SMW, which then becomes the active one.

During initial installation, the first SMW that is installed and configured becomes the active SMW. The second SMW that is installed and configured becomes the passive SMW. However, either SMW can be active during normal operation. The cluster configuration does not remember which SMW was initially configured to be active.

Shared Storage on the Boot RAID

The SMW HA system uses shared disk devices on the boot RAID for data that must be highly available. The shared directories are mounted only on the active SMW. When a failover occurs, access to these directories is automatically transferred to the other SMW as part of the failover process.

> **IMPORTANT:** Because several file systems are shared between the two SMWs, an SMW HA system has a slightly increased risk for double-mount problems. Do **not** mount the CLE boot root, the shared root, or any other CLE file systems from the boot RAID on both SMWs at the same time.

The SMW HA system uses shared space on the boot RAID for the following directories:

`/var/opt/cray/disk/1` Log disk. The following directories symbolically link to the Log disk:

- `/var/opt/cray/debug`
- `/var/opt/cray/dump`
- `/var/opt/cray/log`

`/var/lib/mysql`	MySQL HSS database. Although the database is shared, the HSS database server runs on the active SMW only.
`/home`	SMW home directories.
`/var/lib/pgsql`	Power Management Database (PMDB), if on the shared boot RAID. Note that mirrored storage is preferred. For more information, see *Storage for the Power Management Database (PMDB)* on page 8.

Storage for the Power Management Database (PMDB)

The Power Management Database (PMDB) is a PostgreSQL database that contains power management data, event router file system (erfs) data, and optional System Environment Data Collections (SEDC) data. The directory `/var/lib/pgsql` is the mount point for the PMDB. On an SMW HA system, this directory should be configured to be available after failover. When a failover occurs, access to `/var/lib/pgsql` is automatically transferred to the other SMW as part of the failover process.

The following options are available for PMDB storage:

- Mirrored storage (preferred): An optional pair of disks, one in each SMW, to store PMDB data. In this configuration, the active SMW mounts `/var/lib/pgsql` as a Distributed Replicated Block Device (DRBD) device and communicates replicated writes over a private TCP/IP connection (eth5) to the passive SMW. This is the preferred PMDB configuration to ensure availability of the PMDB data without competition for I/O bandwidth to the SMW root disk or boot RAID file systems.

  For more information, see *Configure Mirrored Storage with DRBD for the PMDB* on page 38.
- Shared storage: A logical disk, configured as a LUN (Logical Unit) or logical volume on the boot RAID. The boot RAID must have sufficient space for `/var/lib/pgsql`.

  For more information, see *Shared Storage for SMW HA* and *Configure Shared Storage on the Boot RAID for the PMDB* on page 44.
- Unshared storage (not recommended): Each SMW stores an unsynchronized copy of `/var/lib/pgsql` on the local root disk. Cray strongly recommends using either mirrored storage (preferred) or shared storage. An unshared PMDB is split across both SMWs; data collected before an SMW failover will be lost or not easily accessible after failover.

Synchronized Files

For files not located on the shared storage device, the SLEHA Extension software includes the `csync2` utility to synchronize (*sync*) important files between the two SMWs. When a file changes on the active SMW, it is automatically synchronized to the passive SMW.

File synchronization is automatically configured during initial installation. The file `/etc/csync2/csync2_cray.cfg` lists the Cray-specific files and directories that must be synchronized, as well as small files that are convenient to keep in sync.

File synchronization happens in one direction only: from the active SMW to the passive SMW. If you change a synchronized file on the passive SMW, the change will not be propagated to the active SMW in the course of

normal operations and could be overwritten on the passive SMW later if there is a subsequent change to the corresponding file on the active SMW. However, if a failover occurs, the previously passive SMW becomes the active SMW. If the change is still in place, the changed file becomes a candidate for propagation to the other SMW (subject to the rules of file conflict resolution).

The `fsync` resource controls file synchronized operations. Every 100 seconds, `fsync` checks for files that need to be synchronized.

> **IMPORTANT:** If a failover occurs before a file synchronization operation has completed, it could result in the loss of the latest updates.

The `csync2` utility synchronizes the required files and directories for the SMW HA cluster, such as `/etc/passwd` and `/opt/cray/hss/*/etc/*`. For more information, see *About File Synchronization Between HA SMWs* on page 35 or examine the contents of `/etc/csync2/csync2_cray.cfg`.

Very large files are explicitly excluded from synchronization (such as `/opt/cray/hss-images/master`). The `csync2` utility is designed to synchronize small amounts of data. If `csync2` must monitor many directories or synchronize a large amount of data, it can become overloaded and failures may not be readily apparent. Cray recommends that you do not change the list of synchronized files (or add only small files); copy large files and directories manually to the other SMW. For more information, see *About File Synchronization Between HA SMWs* on page 35.

Cluster Resources

A resource is any type of service or application that is managed by the Pacemaker Cluster Resource Manager, such as a daemon or file system. In an SMW HA system, the HSS (rsms) daemons are configured as resources.

Each time a resource fails, it is automatically restarted and its failcount is raised. If the failcount exceeds the defined migration threshold for the resource, a failover occurs and management of all cluster resources migrates to the other SMW, making it the active SMW. The original SMW will no longer be allowed to run the failed resource, so no failback can occur until the resource's failcount is reset for that SMW.

> **TIP:** You can reset failcounts with the `clean_resources` or `clear_failcounts` command. For more information, see *Resources Are Stopped*.

An SMW HA system includes the following resources:

ClusterIP, ClusterIP1, ClusterIP2, ClusterIP3, and ClusterIP4	Control and monitor the Ethernet connections (`eth0`, `eth1`, `eth2`, `eth3`, and `eth4`, respectively).
ClusterMonitor	Records failcounts and failed actions in the log file `/var/log/smwha.log` at cluster startup, then clears the failure data from `crm` (for example, in the output of `crm_mon -r1`).
ClusterTimeSync	Monitors the kernel time on each SMW. If the difference is greater than 60 seconds, both SMWs are synchronized with the NTP server. If the time difference is greater than 10 hours, the time is not synchronized.
cray-syslog	Controls and monitors Lightweight Log Management (LLM).
dhcpd	Controls and monitors `dhcpd` as used by the SMW HA feature.

fsync	Provides file synchronization using `csync2`.
homedir	Mounts and unmounts the shared `/home` directory.
hss-daemons	Controls and monitors HSS daemons; corresponds to the `/etc/init.d/rsms` startup script.
ip_drbd_pgsql	Controls and monitors the Ethernet connection (`eth5`) between the two SMWs for Power Management Database (PMDB) mirrored storage using a Distributed Replicated Block Device (DRBD).
ms_drbd_pgsql	Monitors the master/slave DRBD cluster resource for PMDB mirrored storage.
Notification	Provides automatic notification email when a failover occurs.
md-fs	Mounts, unmounts, and monitors the shared MySQL database, `/var/lib/mysql`.
ml-fs	Mounts, unmounts, and monitors the shared log directory, `/var/opt/cray/disk/1`, which symbolically links to the `dump`, `install`, and `log` subdirectories in `/var/opt/cray/`.
mysqld	Controls and monitors MySQL.
pm-fs	Controls and monitors the Power Management Database (PMDB) file system, `/var/lib/pgsql`.
postgresqld	Controls and monitors the Power Management Database (PMDB) PostgreSQL server, `postgresqld`.
stonith-1 and stonith-2	Monitors the health of the other SMW. Each SMW monitors its peer and has the ability to power off that peer at failover time using the STONITH capability. STONITH failovers are used when the state of the failing SMW cannot be determined. A STONITH failover powers off the failing SMW to guarantee that the newly active SMW has exclusive access to all cluster managed resources.

Limitations of SMW Failover

The SMW HA failover feature has the following limitations:

- Both SMWs must run the same versions of SLES and SMW/HSS software.
- System administration of an SMW HA environment is more complex than administration of a system with a single SMW.
- Before using a command that interacts with the HSS daemons, wait for 30 - 60 seconds after failover to ensure that all cluster resources have started. In the first 30 seconds after failover, resources may appear to be started, then change to another state. Although you might be able to log in via the virtual IP address before this period is over, the cluster is not ready for use until all resources are fully started.

TIP: Use `crm_mon` to verify that all cluster resources have started after failover. For more information, see *crm_mon Command*.

- SMW and CLE upgrades in an HA environment require some duplication of effort, with portions of the procedure done individually to each SMW. System down-time requirements for operating system upgrades are somewhat longer as a result.
- There is no support for seamless failover (also called *double failure*) if errors occur while the system is doing error handling for another system component. If an HSS daemon or other SMW process were doing some type of error handling that got interrupted by an (unrelated) failover, when that daemon restarts on the new SMW it may not be able to resume operation where it left off and complete the recovery from the first error. In this case, even though a failover occurs, manual intervention might still be required to return the system to an operational state.
- There is no support for seamless failover during operational commands. All user commands that were started from the active SMW are terminated. These commands must be restarted on the new active SMW. The restarted commands might not start with the same internal states, if those commands do not provide persistent capabilities. An interrupted operation such as `xtbootsys`, shutdown, dump, warm-swap, or flash will need to be reissued after failover has completed and the other SMW becomes active.
- Partial migration of managed resources is not supported. For example, the SMW HA system does not support migration of individual HSS daemons or resources to the other SMW. A particular SMW is either *active*, with complete responsibility for all HSS daemons, or *passive* with no HSS daemons running.
- If both SMWs are started (powered on) at the same time, a race condition can develop that could result in one SMW being powered off via the STONITH capability. Before starting the second SMW, wait until the first SMW has completed startup and initialized all cluster resources. For more information, see *Boot an SMW HA Cluster*.
- During failover, if there is no communication between the SMW and the Cray mainframe for about 30 seconds, workload throttling can occur; therefore, auto-throttling of applications is likely while an actual SMW failover is taking place. Blades begin to auto-throttle if essential HSS daemons (`erd`, `state-manager`, or `xtnlrd`) are unavailable and lasts until those daemons resume operation on the other SMW. On a single-cabinet system, the throttled period was fairly consistent, lasting 37 seconds. The throttled period may increase for larger systems.
- Direct Attached Lustre (DAL) is not supported with the SMW HA failover release.
- If the Power Management Database (PMDB) is on local SMW disks rather than on mirrored or shared storage, PMDB data collected before an SMW failover will be lost or not easily accessible after failover.

Operational Differences on an SMW HA System

The SMW HA system includes the following operational differences from running a single SMW:

- On an SMW HA system, you must control the `rsms` daemon as `root` rather than as `crayadm`. In addition, restarting `rsms` behaves differently than on a system with a single SMW. Running `/etc/init.d/rsms restart` does not display the expected output, because the HA cluster returns immediately rather than waiting for the HSS daemons to start.

  TIP: To display the daemon status, run `/etc/init.d/rsms status`.

- Key system services (also called *resources*) are controlled by the cluster manager (see *Cluster Resources*). Do not start or stop these services individually. Instead, use cluster management tools to start and stop these services. For more information, see *About SMW HA Commands*.
- Users may notice differences in the behavior of the `find` command for the shared file systems on the boot RAID. By default, `find` does not follow symbolic links (for example, in the log file system). To follow symbolic links, use `find -L`.
- Auto-throttling of applications is likely while an actual SMW failover is taking place. Blades begin to auto-throttle if essential HSS daemons (`erd`, `state-manager`, or `xtnlrd`) are unavailable and lasts until those daemons resume operation on the other SMW. On a single-cabinet system, the throttled period was fairly consistent, lasting 37 seconds. The throttled period may increase for larger systems.
- Because several file systems are shared between the two SMWs, an SMW HA system has a slightly increased risk for double-mount problems. Do not mount the CLE boot root, the shared root, or any other CLE file systems from the boot RAID on both SMWs at the same time.
- An SMW HA system disables the automatic `fsck` for shared file systems at system start time because the checks could delay failover by several minutes or hours. Cray recommends manually checking each shared file system on a regular basis, such as during periodic maintenance. For more information, see *Check Shared File Systems Manually with fsck*.

  For additional differences, see *Limitations of SMW Failover*.

About SMW HA Commands

The SUSE Pacemaker Cluster Resource Manager (CRM) includes several administration commands to monitor and manage a cluster:

- `crm`
- `crm_gui`
- `crm_mon`
- `crm_resource`

The Cray SMW HA software includes additional commands for an SMW HA cluster:

- `clean_resources` - Cleans up all SMW failover resources on both SMWs.
- `clear_failcounts` - Resets the failcounts and failed action data for all SMW failover resources.
- `show_failcount`- Displays the failcount of a specific SMW failover resource.
- `show_failcounts`- Displays the failcounts of all SMW failover resources on both SMWs.
- `set_migration_threshold` - Sets the migration threshold for an SMW failover resource.
- `show_migration_threshold` - Displays the migration threshold for an SMW failover resource.
- `SMWHAconfig` - Configures SMW failover on both SMWs in an SMW HA cluster. Also adds and removes shared or mirrored storage.

Only the `root` user can execute the Cray SMW HA commands. These commands are included in the `ha-smw` module, which is automatically loaded when the `root` user logs in. If necessary, use the following command to load the `ha-smw` module:

```
smw1:~ # module load ha-smw
```

crm Command

The `crm` command provides a command-line interface to the SUSE Pacemaker Cluster Resource Manager (CRM). This command can be used either as an interactive shell or as a single command entered on the command line.

For example, execute the following command to display a list of all cluster resources on the system.

```
smw1:~ # crm resource show
 stonith-1 (stonith:external/ipmi): Started
 stonith-2 (stonith:external/ipmi): Started
 dhcpd (lsb:dhcpd): Started
 cray-syslog (lsb:cray-syslog): Started
 ClusterIP (ocf::heartbeat:IPaddr2): Started
 ClusterIP1 (ocf::heartbeat:IPaddr2): Started
```

```
ClusterIP2 (ocf::heartbeat:IPaddr2): Started
ClusterIP3 (ocf::heartbeat:IPaddr2): Started
ClusterIP4 (ocf::heartbeat:IPaddr2): Started
fsync (ocf::smw:fsync): Started
hss-daemons (lsb:rsms): Started
Notification (ocf::heartbeat:MailTo): Started
ClusterMonitor (ocf::smw:ClusterMonitor): Started
Resource Group: HSSGroup
    homedir (ocf::heartbeat:Filesystem): Started
    ml-fs (ocf::heartbeat:Filesystem): Started
    md-fs (ocf::heartbeat:Filesystem): Started
    mysqld (ocf::heartbeat:mysql): Started
```

To display the status of a single resource, such as `fsync`, execute the following command:

```
smw1:~ # crm resource status fsync
resource fsync is running on: smw1
```

To display the same information with the interactive interface:

```
smw1:~ # crm
crm(live)# resource
crm(live)resource# status fsync
resource fsync is running on: smw1
crm(live)resource# end
crm(live)# quit
smw1:~ #
```

> **TIP:** The `crm` command has multiple levels. Use the `help` keyword to display the commands at each level and the valid options and arguments for each command. For example, the following commands display different levels of help:
>
> - `crm help`
> - `crm resource help`
> - `crm resource failcount help`

For more information, see the `crm(8)` man page and the *SUSE Linux Enterprise High Availability Extension High Availability Guide*.

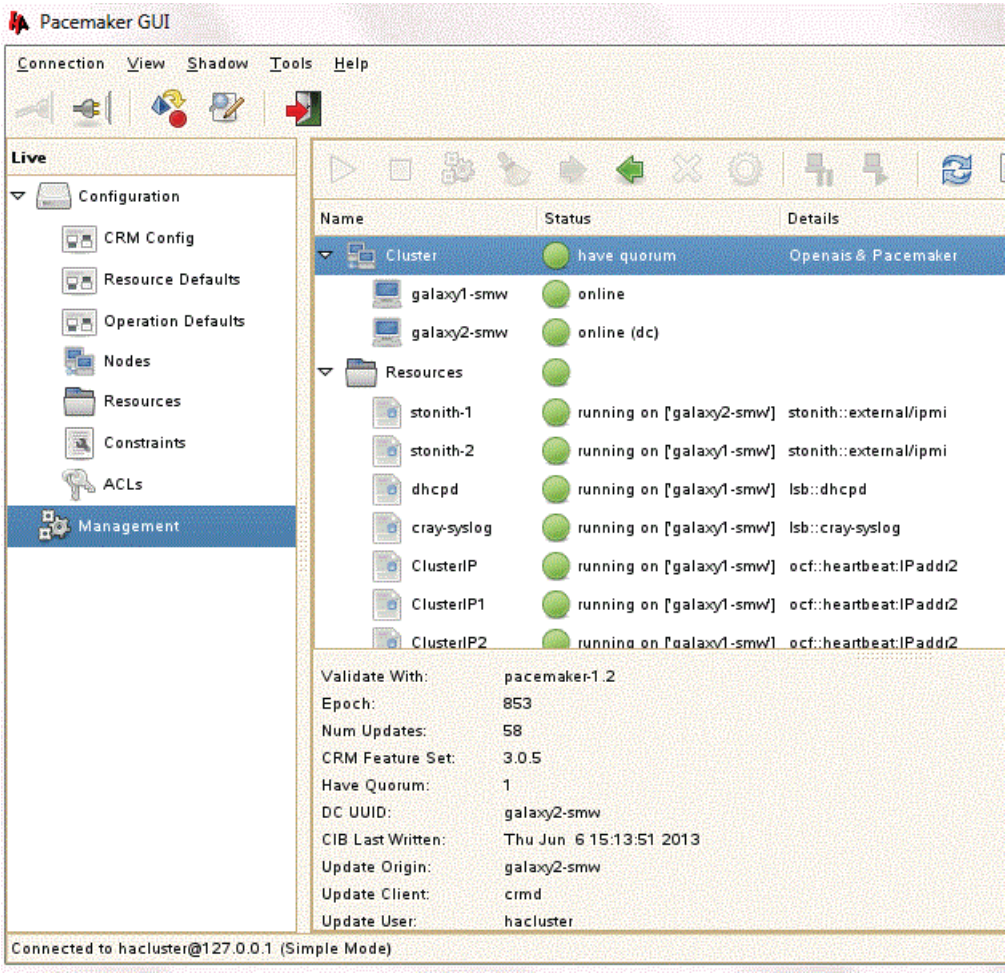
crm_gui Command

The `crm_gui` command provides a graphical interface to the SUSE Pacemaker Cluster Resource Manager (CRM).

When the `crm_gui` window opens, it is blank. Connect to the cluster with ConnectionLogin, then log in as the `hacluster` user. Use the same password as `root` on the SMW (see *Passwords For an SMW HA System*).

To display node and resource status, click on Management in the left pane.

Figure 2. Pacemaker GUI (`crm_gui`) Management Window



In the management display, a green circle marks a node or resource that is running without errors; a red circle marks an item with problems, such as an offline node or stopped resource. Click on a node or resource to display status details (including errors) in the bottom panel of the window.

IMPORTANT: Do **not** edit the resources; changing the resources configuration can break the cluster.

The management display also marks one of the nodes with `(dc)`, which stands for *designated coordinator*. This is a Pacemaker CRM concept that is not related to the SMW's current active or passive role. The active SMW is not necessarily the CRM designated coordinator.

For information on using `crm_gui`, see the *SUSE Linux Enterprise High Availability Extension High Availability Guide*.

crm_mon Command

The SUSE `crm_mon` command helps monitor cluster status and configuration. The output includes the number of nodes, host names, SMW status, the resources configured in the cluster, the current status of each resource, and any failed actions.

By default (if no options are specified), `crm_mon` runs continuously, redisplaying the cluster status every 15 seconds. To specify the number of repeats, enter a number as an option. This example displays one snapshot of cluster status.

```
smw1:~ # crm_mon -r1
Last updated: Sun Oct 26 23:54:38 2014
Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
Stack: classic openais (with plugin)
Current DC: smw1 - partition with quorum
Version: 1.1.9-2db99f1
2 Nodes configured, 2 expected votes
19 Resources configured.


Online: [ smw1 smw2 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 dhcpd  (lsb:dhcpd):    Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (lsb:cray-syslog):      Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
     postgresqld        (lsb:postgresql):       Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
```

NOTE: `crm_mon` may display different resource names, group names, or resource order on the system.

TIP: Use the `-r` option to include inactive resources.

The `crm_mon` output marks one of the nodes as the `Current DC`, which stands for *designated coordinator*. This is a Pacemaker CRM concept that is not related to the SMW's current active or passive role. The active SMW is not necessarily the CRM designated coordinator.

For more information, see the `crm_mon(8)` man page and the *SUSE Linux Enterprise High Availability Extension High Availability Guide*.

crm_resource Command

The SUSE `crm_resource` command displays resource information for the cluster (see *Cluster Resources*). You can use the `-l` (lower-case L) option to list the name of each instantiated cluster resource. For example, enter the following command as `root` on either SMW.

```
smw1:~ # crm_resource -l
ClusterIP
```

```
ClusterIP1
ClusterIP2
ClusterIP3
ClusterIP4
ClusterMonitor
Notification
dhcpd
fsync
hss-daemons
stonith-1
stonith-2
ml-fs
cray-syslog
homedir
md-fs
pm-fs
postgresqld
mysqld
```

NOTE: `crm_resource` may display different resource names or resource order on your system.

For more information, see the `crm_resource(8)` man page and the *SUSE Linux Enterprise High Availability Extension High Availability Guide*.

Cray SMW HA Cluster Commands

The Cray SMW HA software provides several commands to monitor the cluster status, clean up resource problems, and configure migration thresholds.

NOTE: You must be `root` to execute these commands. Except as noted below, all commands can be run on either the active or passive SMW.

- `show_failcounts`: Displays the failcounts of all SMW failover resources on both SMWs. This command shows the failcounts (number of failures) for all resources on both SMWs; it provides a quick way to access the failcount data for all resources in an SMW HA cluster, rather than running multiple `crm` or `crm_failcount` commands.
- `show_failcount`: Displays the failcount of a specific SMW failover resource. This command shows the failcount (number of failures) of the specified resource. This command provides a simple way to display the failcount data of a resource, rather than running the `crm` or `crm_failcount` command.
- `clear_failcounts`: Resets the failcounts and failed action data for all SMW failover resources. This command resets the resource failcounts (number of failures) and list of failed actions on both SMWs in an SMW HA cluster.

  `clear_failcounts` provides a quick way to clear all failcount data, rather than running multiple `crm` or `crm_failcount` commands.
- `clean_resources`: Cleans up all SMW failover resources on both SMWs. This command sets the status of each resource to the default clean state and sets the failcount (number of failures) to 0. If some resources did not start after system boot or are marked as unclean after failover, use this command to quickly clean up all resources on both SMWs. The command `crm resource cleanup` also cleans up resources, but requires you to enter each resource name separately.

  After running `clean_resources`, wait several minutes for cluster activity to settle. You can check cluster status with the `crm_mon -r1` command.

- `set_migration_threshold`: Sets the migration threshold for an SMW failover resource. A migration threshold is defined as the maximum number of failures (the failcount) allowed for the resource. If the failcount exceeds this threshold, a failover occurs and management of all cluster resources migrates to the other SMW, making it the active SMW. By default, the migration threshold is `1000000`.
- `show_migration_threshold`: Displays the migration threshold for an SMW failover resource. A migration threshold is defined as the maximum number of failures (the failcount) allowed for a resource (any type of service or application that is managed by the Pacemaker Cluster Resource Manager, such as a daemon or file system). If the failcount exceeds this threshold, a failover occurs and management of all cluster resources migrates to the other SMW, making it the active SMW. The original SMW will no longer be allowed to run the failed resource until the resource's failcount is reset for that SMW.

  Before executing `show_migration_threshold`, you must explicitly set the migration threshold with the `set_migration_threshold` command. If the migration threshold has not been set (that is, if it has the default value), `show_migration_threshold` displays an error message.
- `SMWHAconfig`: Configures SMW failover on both SMWs in an SMW HA cluster. After installing or updating the Cray SMW HA software, execute this command on the active SMW to configure both SMWs through `ssh`.

  Execute the `SMWHAconfig` command only on the active SMW.

For more information, see the man pages for these commands.

About SMW HA Operational Differences

The administration tasks for an SMW HA system are generally the same as those for a system with a single SMW. The operational differences for an SMW HA system and HA-specific procedures are:

- *Boot an SMW HA Cluster*
- Log in to the active SMW; see *Log In to the SMW HA Cluster*
- *Identify the Active SMW*
- *Monitor the SMW HA Cluster*
- *Change Passwords on an SMW HA System*
- *Check Shared File Systems Manually with fsck*
- *Customize the SMW HA Cluster*
- *Critical Events That Cause SMW HA Failover*
- For an overview of the SMW HA commands; see *About SMW HA Commands*.

The following conventions are used to refer to the SMWs:

- The host name `smw1` specifies the currently active SMW. In examples, the prompt `smw1:~ #` shows a command that runs on this SMW.
- The host name `smw2` specifies the currently passive SMW. In examples, the prompt `smw2:~ #` shows a command that runs on this SMW.
- The host name `virtual-smw` host name specifies the virtual (active) SMW, which could be either `smw1` or `smw2`. This virtual host name was defined during initial installation.

Boot an SMW HA Cluster

IMPORTANT: When SMW HA is enabled, do not start both SMWs at the same time. Doing so can cause a race condition that could result in one SMW being powered off via the STONITH capability. Before starting the second SMW, wait until the first SMW has completed startup and initialized all cluster resources.

Follow these steps to boot or reboot both SMWs.

1. Boot `smw1` (or the SMW that you want to be active).

   Before continuing, wait until `smw1` has rejoined the cluster. After the SMW responds to a `ping` command, log into `smw1`, sleep for at least 2 minutes, then execute the `crm_mon -r1` command to verify that `smw2` is online.

   TIP: You can check the status of the SMW HA services with the `crm_mon -r1` command. For more information, see *Display SMW HA Cluster and Resource Status*.

2. Boot `smw2` (or the SMW that you want to be passive).

   Before continuing, wait until `smw2` has rejoined the cluster. After the SMW responds to a `ping` command, log into `smw2`, sleep for at least 2 minutes, then execute the `crm_mon -r1` command to verify that `smw2` is online.

3. Verify that all resources are running.

   a. Display the cluster status.

```
smw1:~ # crm_mon -r1
Last updated: Mon Oct 27 01:19:23 2014
Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
Stack: classic openais (with plugin)
Current DC: smw1 - partition with quorum
Version: 1.1.9-2db99f1
2 Nodes configured, 2 expected votes
19 Resources configured.


Online: [ smw1 smw2 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 dhcpd  (lsb:dhcpd):    Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (lsb:cray-syslog):      Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
     postgresqld        (lsb:postgresql):       Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
```

   Note that `crm_mon` may display different resource names, group names, or resource order on the system.

   b. Examine the `crm_mon` output. Verify that each resource has started by looking for `Started smw1` or `Started smw2`. Also look for any failed actions at the end of the output.

   c. If not all resources have started or if any failed actions are displayed, execute the `clean_resources` command on either SMW.

      IMPORTANT: When running the `clean_resources` command, you must be directly logged in as `root` (instead of using `su` from a `crayadm` login), because `clean_resources` terminates all non-root user sessions.

```
smw1:~ # clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=stonith-1
Cleaning resource on node=smw1 for resource=stonith-2
Cleaning resource on node=smw1 for resource=dhcpd
Cleaning resource on node=smw1 for resource=cray-syslog
```

```
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning resource on node=smw1 for resource=ClusterIP2
 ...
Cleaning resources on node smw2
Cleaning resource on node=smw2 for resource=stonith-1
Cleaning resource on node=smw2 for resource=stonith-2
 ...
Cleaning resource on node=smw2 for resource=Notification
```

After running `clean_resources`, wait several minutes for cluster activity to settle. You can check cluster status with the `crm_mon -r1` command. If the output of this command shows only a subset of the SMW HA services, wait for another minute, then check again. For more information, see the `clean_resources(8)` man page.

Log into the SMW HA Cluster

To log on to the active SMW, specify the virtual SMW host name.

Cray recommends that you always connect to the SMW cluster using the virtual host name. Avoid connecting to an SMW by specifying the actual host names, except for host-specific maintenance. In the event of a failover, all connections made using the virtual host name will be terminated. A connection to the active SMW via the actual host name could be confusing after a failover occurs, because the login session would remain open, but there is no indication that the SMW is now passive.

NOTE: This example shows the virtual host name *`virtual-smw`*. Specify the virtual host name of the SMW HA cluster.

```
remote-system% ssh root@virtual-smw
smw1:~ #
```

After you log in, the prompt displays the host name of the active SMW (in this example, `smw1`).

To log on to a specific SMW, use the actual host name of the SMW (such as `smw1` or `smw2`).

Identify the Active SMW

1. The easiest way to find the active SMW is to log in using the virtual SMW host name and look at the system prompt, as described in *Log into the SMW HA Cluster*.

2. Another way to find the active SMW is to determine where the SMW HA cluster resources are running (such as the `hss-daemons` resource).

   NOTE: One `stonith` resource runs on each SMW to monitor the other SMW. All other resources run only on the active SMW.

   As `root` on either SMW, execute the following command.

```
smw1:~ # crm_mon -r1 | grep hss-daemons
 hss-daemons    (lsb:rsms):     Started smw1
```

Display SMW HA Cluster and Resource Status

Use some or all of the following steps to check the health of the SMW HA cluster.

> **NOTE:** You must execute the CRM and Cray SMW HA commands as `root`. Unless otherwise noted, you can execute these commands on either SMW.

1. Verify that both SMWs are online.

```
smw1:~ # crm_mon -r1 | grep Online
Online: [ smw1 smw2 ]
```

2. Display the cluster status with `crm_mon`.

```
smw1:~ # crm_mon -r1
Last updated: Mon Oct 27 01:19:23 2014
Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
Stack: classic openais (with plugin)
Current DC: smw1 - partition with quorum
Version: 1.1.9-2db99f1
2 Nodes configured, 2 expected votes
19 Resources configured.


Online: [ smw1 smw2 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 dhcpd  (lsb:dhcpd):    Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (lsb:cray-syslog):      Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
     postgresqld        (lsb:postgresql):       Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1

Failed actions:
    fsync_monitor_0 (node=smw2, call=11, rc=-2, status=Timed Out):
          unknown exec error
    ml-fs_start_0 (node=smw2, call=31, rc=1, status=complete): unknown error
```

`crm_mon` may display different resource names, group names, or resource order on the system.

All resources run only on the active SMW (except for one `stonith` resource, which is a special case). In the previous example, `smw1` is the active SMW.

Failed actions can be cleared by using the `clear_failcounts` command. Any failed actions that display again indicate issues with the resources.

3. Display the status of the cluster resources.

```
smw1:~ # crm resource status
 stonith-1      (stonith:external/ipmi) Started
 stonith-2      (stonith:external/ipmi) Started
 dhcpd  (lsb:dhcpd) Started
 cray-syslog    (lsb:cray-syslog) Started
 ClusterIP      (ocf::heartbeat:IPaddr2) Started
 ClusterIP1     (ocf::heartbeat:IPaddr2) Started
 ClusterIP2     (ocf::heartbeat:IPaddr2) Started
 ClusterIP3     (ocf::heartbeat:IPaddr2) Started
 ClusterIP4     (ocf::heartbeat:IPaddr2) Started
 fsync  (ocf::smw:fsync) Started
 homedir        (ocf::heartbeat:Filesystem) Started
 hss-daemons    (lsb:rsms) Started
 Notification   (ocf::heartbeat:MailTo) Started
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem) Started
     md-fs      (ocf::heartbeat:Filesystem) Started
     mysqld     (ocf::heartbeat:mysql) Started
```

For information on restarting a stopped resource, see *Resources Are Stopped*.

4. Display failcount data for all resources.

```
smw1:~# show_failcounts
node=smw1 scope=status  name=fail-count-stonith-1 value=0
node=smw1 scope=status  name=fail-count-stonith-2 value=0
node=smw1 scope=status  name=fail-count-dhcpd value=0
node=smw1 scope=status  name=fail-count-cray-syslog value=0
node=smw1 scope=status  name=fail-count-ClusterIP value=0
   .
   .
   .
node=smw2 scope=status  name=fail-count-hss-daemons value=0
node=smw2 scope=status  name=fail-count-Notification value=0
node=smw2 scope=status  name=fail-count-ClusterMonitor value=0
node=smw2 scope=status  name=fail-count-ml-fs value=0
node=smw2 scope=status  name=fail-count-md-fs value=0
node=smw2 scope=status  name=fail-count-mysqld value=0
```

You can display the failcount data for a single resource on one SMW. This example shows the failcount data for the `fsync` resource. (Replace *smwX* with the actual SMW host name.)

```
smw1:~ # show_failcount smwX fsync
scope=status  name=fail-count-fsync value=0
```

For information on clearing the failcount values, see *Resources Are Stopped*.

5. Test file synchronization by creating a temporary file in a synchronized directory on the active SMW, then check for it on the passive SMW.
This example assumes that `smw1` is the active SMW:

```
smw1:~ # cp /etc/motd /opt/cray/hss/default/etc/my_test_file
smw1:~ # ls -l /opt/cray/hss/default/etc/my_test_file
```

```
smw1:~ # md5sum /opt/cray/hss/default/etc/my_test_file

... (wait about 2 minutes for the next file synchronization  operation to complete) ...

smw1:~ # ssh smw2
...
smw2:~ # ls -l /opt/cray/hss/default/etc/my_test_file
smw2:~ # md5sum /opt/cray/hss/default/etc/my_test_file
```

Finally, return to the active SMW to delete the test file. Within several minutes, the file will be automatically removed from the passive SMW.

Display SMW Power Status

If you are not near the SMWs to check the LEDs, you can use one of the following methods to display the power status for the SMWs:

1. As `root` on either SMW, use the `crm_mon` command to check the SMW status.

```
smw1:~ # crm_mon -r1
Last updated: Mon Oct 27 01:19:23 2014
Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
Stack: classic openais (with plugin)
Current DC: smw1 - partition with quorum
Version: 1.1.9-2db99f1
2 Nodes configured, 2 expected votes
19 Resources configured.


Online: [ smw1 smw2 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
   .
   .
   .
```

   NOTE: `crm_mon` resource names, group names, or resource order on the system.

2. As `root` on either SMW, use the `ipmitool` command to check the power status of a specific SMW.

   NOTE: Replace *smw-iDRAC-IP-addr* with the SMW's iDRAC IP address.

```
smw1:~ # /usr/bin/ipmitool -I lanplus -U root -H smw-iDRAC-IP-addr -a chassis power status
Password:
Chassis Power is on
```

   At the password prompt, enter the `root` password for the iDRAC.

   For the procedure to restore power and join the SMW to the cluster, see *An SMW Is Powered Off*.

Change SMW, iDRAC, and STONITH Passwords

1. Log on to the active SMW as `root`, using the virtual SMW host name (such as `virtual-smw`). After you have logged in successfully, the prompt displays the host name of the active SMW.

   NOTE: The examples in this procedure assume that `smw1` is the active SMW.

2. To change the SMW `root`, `hacluster`, and `stonith` passwords on the active SMW, execute the following commands on `smw1`:

   ```
   smw1:~# passwd root
   smw1:~# passwd hacluster
   smw1:~# crm resource param stonith-1 set passwd new-passwd
   smw1:~# crm resource param stonith-2 set passwd new-passwd
   ```

   IMPORTANT: The `hacluster` and `stonith` passwords must be the same as the SMW `root` password.

3. To change the SMW `root` and `hacluster` passwords on the passive SMW, execute the following commands on `smw2`:

   ```
   smw1:~# passwd root
   smw1:~# passwd hacluster
   ```

   IMPORTANT: Use the same `root` password as on `smw1`. The `hacluster` password must be the same as the `root` password.

4. To change the iDRAC passwords, see *Manage System Software for the Cray Linux Environment (S-2393)*.

   IMPORTANT: The iDRAC passwords must be the same as the SMW `root` password.

Check the fsck Status of Shared File Systems

An SMW HA system disables the automatic `fsck` for shared file systems at system start time because the checks could delay failover by several minutes or hours. Cray recommends manually checking each shared file system on a regular basis, such as during periodic maintenance.

1. Determine the `/dev/sd` name for each shared file system.

   ```
   smw1:~ # df
        Filesystem     1K-blocks      Used Available Use% Mounted on
        /dev/sda2      120811676  82225412  32449332  72% /
        udev            16433608       756  16432852   1% /dev
        tmpfs           16433608     37560  16396048   1% /dev/shm
        /dev/sdo       483807768 197536596 261695172  44% /var/opt/cray/disk/1
        /dev/sdp       100791728  66682228  28989500  70% /home
        /dev/sdq       100791728    484632  95187096   1% /var/lib/mysql
        /dev/sdr        30237648    692540  28009108   3% /var/lib/pgsql
   ```

2. Check the mount count and last-checked date of each shared file system, as in this example.

```
smw1:~ # tune2fs -l /dev/sdo | egrep -i "mount|check"
Last mounted on:          <not available>
Default mount options:    (none)
Last mount time:          Wed Feb 26 16:33:10 2014
Mount count:              352
Maximum mount count:      38
Last checked:             Sun Jul 21 15:57:55 2013
Check interval:           15552000 (6 months)
Next check after:         Fri Jan 17 14:57:55 2014
```

3. If the mount count exceeds the maximum mount count, schedule time to manually check the file system with `fsck`.

Critical Events That Cause SMW HA Failover

IMPORTANT: When an SMW HA failover occurs, also see *Limitations of SMW Failover* on page 10.

The following critical events cause a failover from the active SMW to the passive SMW:

- Hardware fault on the active SMW.
- Lost heartbeat between the two SMWs.
- Kernel fault (panic) on the active SMW.
- Failed resource (HSS daemon or cluster service). If a resource stops, the cluster manager automatically restarts it and increments the failcount by 1. When the failcount exceeds the migration threshold (by default, `1,000,000`), a failover occurs.

The failover type (STONITH or non-STONITH) depends upon whether the newly active SMW can determine the health of the failing SMW. A STONITH failover occurs only if there is no other way for the new SMW to ensure the integrity of the cluster.

- In the case of STONITH failover, the original SMW is powered off (via the STONITH capability) if it is not already off. This guarantees that file synchronization is stopped and the failed SMW no longer holds any cluster-managed resources so that the new SMW will have exclusive access to those resources.
- In the case of non-STONITH failover, the original SMW is still powered up. In addition:
  - HSS daemons are stopped on the original SMW.
  - Lightweight Log Manager (LLM) logging to shared disk is stopped.
  - File synchronization (`csync2`) between SMWs is stopped.
  - The shared storage versions of `/home`, `/var/opt/cray/disk/1`, and `/var/lib/mysql` are unmounted on the original SMW.
  - Network connections using the `eth0`, `eth1`, `eth2`, `eth3`, and `eth4` virtual IP addresses are dropped and those interfaces begin accepting connections to their actual IP addresses only.

For both types of failover, the following actions then occur on the new SMW:

- The `eth0`, `eth1`, `eth2`, `eth3`, `eth4`, and `eth5` (optional) interfaces begin accepting connections using the virtual IP addresses in addition to their actual IP addresses.
- The shared storage versions of `/home`, `/var/opt/cray/disk/1`, `/var/lib/mysql`, and `/var/lib/pgsql` are mounted on the new SMW.
- File synchronization (`csync2`) between SMWs usually resumes (depending on the reason for failover).
- LLM logging to the shared disk resumes.
- The HSS database (MySQL) is started on the original SMW.
- HSS daemons are started on the new SMW (including, if necessary, any `xtbootsys`-initiated daemons).
- Failcounts and failed actions are written to the log file `/var/log/smw.ha` on the newly active SMW.

Restore Normal Operations After SMW Failover

While a failover is automatic, adding the failed SMW back into the cluster requires manual intervention to identify the reason for failover, take corrective action if needed, and return the failed SMW to an online state. Another failover (that is, a "failback" to the originally active SMW) is not possible until the failed SMW returns to online status and its failcounts are cleared so that it is eligible to run all cluster resources.

1. Identify and fix the problems that caused the failover (such as a hardware fault, kernel panic, or HSS daemon issues). Use the following methods to help diagnose problems:

   a. Examine the log file `/var/log/smwha.log` on the new active SMW. For more information, see *Examine the SMW HA Log File to Determine SMW Failover Cause*.

   b. Execute the `show_failcounts` command and note any resources with non-zero failcounts.

   c. From the active SMW, examine `/var/opt/cray/log/smwmessages-`*yyyymmdd* for relevant messages.

   d. Examine the failing SMW for additional clues.

   e. For a non-STONITH failover: In most cases, the failing SMW will still be running; additional clues may be available in `dmesg` or via other commands.

   f. For a STONITH failover: The failing SMW will be powered off. Before powering it back on, place it into standby mode so that it does not automatically try to rejoin the cluster at startup before ensuring that the node is healthy. For more information, see *Restart Stopped Resources*.

2. Log on to the failing SMW (either from the console or remotely by using the actual host name). Identify the reason for the failure and take corrective action as needed. This might include administrative actions such as freeing space on a file system that has filled up or hardware actions such as replacing a failing component.

3. After the SMW is ready to rejoin the cluster, run the `clean_resources` command as described in *Restart Stopped Resources*. This command also resets all failcounts to zero.

   After running `clean_resources`, wait several minutes for cluster activity to settle. You can check cluster status with the `crm_mon -r1` command.

4. Return the SMW to online status as the passive SMW.

   NOTE: Replace *smw2* with the host name of the failed SMW.

   ```
   smw1:~ # crm node online smw2
   ```

5. If the boot node mounts any SMW directories, and passwordless access between the boot node and SMW is not configured, the mount point on the boot node to the SMW is stale. To refresh the mount point:

   a. Log into the boot node.

   b. Unmount then remount the SMW directories.

c. Restart `bnd`.

```
boot:~ # /etc/init.d/bnd restart
```

Perform a Manual Failover

1. As `root` on the active SMW, put the active SMW into standby mode. This command forces a failover, which stops all resources on the active SMW and moves them to the passive SMW.

   ```
   smw1:~ # crm node standby smw1
   ```

   At this point, the other SMW (`smw2`) is now the active SMW.

2. Bring the previously active SMW (`smw1`) online as the passive SMW.

   ```
   smw1:~ # crm node online smw1
   ```

3. Check the cluster status.

   ```
   smw1:~ # crm_mon -r1
   Last updated: Mon Oct 27 01:19:23 2014
   Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
   Stack: classic openais (with plugin)
   Current DC: smw1 - partition with quorum
   Version: 1.1.9-2db99f1
   2 Nodes configured, 2 expected votes
   19 Resources configured.


   Online: [ smw1 smw2 ]

    ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
    Notification   (ocf::heartbeat:MailTo):        Started smw1
    dhcpd  (lsb:dhcpd):    Started smw1
    fsync  (ocf::smw:fsync):       Started smw1
    hss-daemons    (lsb:rsms):     Started smw1
    stonith-1      (stonith:external/ipmi):        Started smw2
    stonith-2      (stonith:external/ipmi):        Started smw1
    Resource Group: HSSGroup
        ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
        cray-syslog        (lsb:cray-syslog):      Started smw1
        homedir    (ocf::heartbeat:Filesystem):    Started smw1
        md-fs      (ocf::heartbeat:Filesystem):    Started smw1
        pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
        postgresqld        (lsb:postgresql):       Started smw1
        mysqld     (ocf::heartbeat:mysql): Started smw1
   ```

   NOTE: `crm_mon` may display different resource names, group names, or resource order on the system.

4. If `crm_mon` shows resource problems, see the troubleshooting tips in *Resources Are Stopped*.

Examine the SMW HA Log File to Determine SMW Failover Cause

The log file `/var/log/smwha.log` contains cluster status and resource failure data that can help determine the cause of a failover. At system startup (such as after a failover), the `ClusterMonitor` resource records failcounts and failed actions in the log file, then clears this failure information from `crm` (for example, in the output of `crm_mon -1`).

NOTE: The log file `/var/log/smwha.log` is not shared; entries are recorded only on the active SMW.

This example shows the format of entries in the log file.

```
************************************************************
smw1 acted as active SMW at Wed Jul 23 08:45:42 CDT 2014
************************************************************
node=smw1 scope=status  name=fail-count-stonith-1 value=0
node=smw1 scope=status  name=fail-count-stonith-2 value=0
node=smw1 scope=status  name=fail-count-dhcpd value=0
node=smw1 scope=status  name=fail-count-ClusterIP value=0
node=smw1 scope=status  name=fail-count-ClusterIP1 value=0
node=smw1 scope=status  name=fail-count-ClusterIP2 value=0
node=smw1 scope=status  name=fail-count-ClusterIP3 value=0
node=smw1 scope=status  name=fail-count-ClusterIP4 value=0
node=smw1 scope=status  name=fail-count-fsync value=0
node=smw1 scope=status  name=fail-count-hss-daemons value=24
node=smw1 scope=status  name=fail-count-Notification value=0
node=smw1 scope=status  name=fail-count-ClusterMonitor value=0
node=smw1 scope=status  name=fail-count-ml-fs value=0
node=smw1 scope=status  name=fail-count-cray-syslog value=0
node=smw1 scope=status  name=fail-count-homedir value=0
node=smw1 scope=status  name=fail-count-md-fs value=0
node=smw1 scope=status  name=fail-count-pm-fs value=0
node=smw1 scope=status  name=fail-count-postgresqld value=0
node=smw1 scope=status  name=fail-count-mysqld value=0
node=smw2 scope=status  name=fail-count-stonith-1 value=0
node=smw2 scope=status  name=fail-count-stonith-2 value=0
node=smw2 scope=status  name=fail-count-dhcpd value=0
node=smw2 scope=status  name=fail-count-ClusterIP value=0
node=smw2 scope=status  name=fail-count-ClusterIP1 value=0
node=smw2 scope=status  name=fail-count-ClusterIP2 value=0
node=smw2 scope=status  name=fail-count-ClusterIP3 value=0
node=smw2 scope=status  name=fail-count-ClusterIP4 value=0
node=smw2 scope=status  name=fail-count-fsync value=0
node=smw2 scope=status  name=fail-count-hss-daemons value=0
node=smw2 scope=status  name=fail-count-Notification value=0
node=smw2 scope=status  name=fail-count-ClusterMonitor value=0
node=smw2 scope=status  name=fail-count-ml-fs value=0
node=smw2 scope=status  name=fail-count-cray-syslog value=0
node=smw2 scope=status  name=fail-count-homedir value=0
node=smw2 scope=status  name=fail-count-md-fs value=0
```

```
node=smw2 scope=status  name=fail-count-pm-fs value=0
node=smw2 scope=status  name=fail-count-postgresqld value=0
node=smw2 scope=status  name=fail-count-mysqld value=0

smw1:~ # crm status
Last updated: Wed Jul 23 08:45:48 2014
Last change: Wed Jul 23 08:45:09 2014 by root via crm_shadow on smw1
Current DC: smw1 - partition with quorum
2 Nodes configured, 2 expected votes
19 Resources configured.


Online: [ smw1 smw2 ]

 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 dhcpd  (lsb:dhcpd):    Started smw1
 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (lsb:cray-syslog):      Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
     postgresqld        (lsb:postgresql):       Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
```

Customize the SMW HA Cluster

The SMW HA system is configured during installation. You can customize the system by changing the failover notification address, resource migration threshold, and list of synchronized files.

When customizing the SMW HA system, follow these basic rules:

- Do not change the cluster configuration, except for the migration threshold (maximum failcount value). You can set the migration threshold for each resource by using the `set_migration_threshold` command. For more information, see *Cray SMW HA Cluster Commands*.
- Do not attempt to migrate a single resource. All resources must migrate as a group. For more information, see *Cluster Resources*.
- Do not change the system list of synchronized files. You can define which local (site-specific) files are synchronized or excluded from synchronization, but do not add large files or directories to the local list of synchronized files. For more information, see *About File Synchronization Between HA SMWs*.

Change Failover Notification

Prerequisites

Failover notification requires email to be configured on both SMWs. For information about configuring email, see *http://www.postfix.org/BASIC_CONFIGURATION_README.html*.

The SMW HA software includes a `Notification` resource that automatically sends email when a failover occurs.

You can configure failover notification either during initial installation or after the HA system is installed and running.

> **NOTE:** Only one email address is allowed. To send notifications to multiple addresses, create a group email alias that includes all necessary email addresses.

1. As `root` on either SMW, execute the following `crm resource` command.

```
smw1:~ # crm resource param Notification set email address@thedomain.com
```

2. Verify the setting.

```
smw1:~ # crm resource param Notification show email
address@thedomain.com
```

About File Synchronization Between HA SMWs

For files not located on the shared storage device, the SLEHA Extension software includes the `csync2` utility to synchronize (*sync*) important files between the two SMWs. When a file changes on the active SMW, it is automatically synchronized to the passive SMW.

File synchronization happens in one direction only: from the active SMW to the passive SMW. If you change a synchronized file on the passive SMW, the change will not be propagated to the active SMW in the course of normal operations and could be overwritten on the passive SMW later if there is a subsequent change to the corresponding file on the active SMW. However, if a failover occurs, the previously passive SMW becomes the active SMW. If the change is still in place, the changed file becomes a candidate for propagation to the other SMW (subject to the rules of file conflict resolution).

Monitor the fsync Resource

1. Check the status of the `fsync` resource by executing the following command as `root` on either SMW:

   ```
   smw1:~ # crm_mon -r1 | grep fsync
    fsync  (ocf::smw:fsync):       Started smw1
   ```

   The `fsync` resource controls file synchronized operations. Every 100 seconds, `fsync` checks for files that need to be synchronized. If `fsync` stops, no file synchronization occurs.

2. If `fsync` is stopped, display the failcount data for this resource. The status `Stopped` is usually caused by exceeding the failcount for a resource.

   ```
   smw1:~ # show_failcounts | grep fsync
   node=smw1 scope=status  name=fail-count-fsync value=13
   node=smw2 scope=status  name=fail-count-fsync value=0
   ```

3. If necessary, clear the failcount data for the `fsync` resource.

   ```
   smw1:~ # clear_failcounts

   Clearing failcount on node smw1
   Clearing failcount on node=smw1 for resource=stonith-1
   Clearing failcount on node=smw1 for resource=stonith-2
   Clearing failcount on node=smw1 for resource=dhcpd
   Clearing failcount on node=smw1 for resource=cray-syslog
   Clearing failcount on node=smw1 for resource=ClusterIP
    .
    .
    .
   Clearing failcount on node=smw2 for resource=hss-daemons
   Clearing failcount on node=smw2 for resource=Notification
   Clearing failcount on node=smw2 for resource=ClusterMonitor
   Clearing failcount on node=smw2 for resource=ml-fs
   Clearing failcount on node=smw2 for resource=md-fs
   Clearing failcount on node=smw2 for resource=mysqld
   ```

4. After all failcounts have been cleared, the resource should be up and running. Check the cluster status again to verify that the resource has been restarted.

```
smw1:~ # crm resource status
 stonith-1      (stonith:external/ipmi) Started
 stonith-2      (stonith:external/ipmi) Started
 dhcpd  (lsb:dhcpd) Started
 cray-syslog    (lsb:cray-syslog) Started
 ClusterIP      (ocf::heartbeat:IPaddr2) Started
 ClusterIP1     (ocf::heartbeat:IPaddr2) Started
 ClusterIP2     (ocf::heartbeat:IPaddr2) Started
 ClusterIP3     (ocf::heartbeat:IPaddr2) Started
 ClusterIP4     (ocf::heartbeat:IPaddr2) Started
 fsync  (ocf::smw:fsync) Started
 homedir        (ocf::heartbeat:Filesystem) Started
 hss-daemons    (lsb:rsms) Started
 Notification   (ocf::heartbeat:MailTo) Started
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem) Started
     md-fs      (ocf::heartbeat:Filesystem) Started
     mysqld     (ocf::heartbeat:mysql) Started
```

5. If not all resources have started, execute the `clean_resources` command.

```
smw1:~ # clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=stonith-1
Cleaning resource on node=smw1 for resource=stonith-2
Cleaning resource on node=smw1 for resource=dhcpd
Cleaning resource on node=smw1 for resource=cray-syslog
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning resource on node=smw1 for resource=ClusterIP2
 ...
Cleaning resources on node smw2
Cleaning resource on node=smw2 for resource=stonith-1
Cleaning resource on node=smw2 for resource=stonith-2
 ...
Cleaning resource on node=smw2 for resource=Notification
Cleaning resource on node=smw2 for resource=ClusterMonitor
```

NOTE: After running `clean_resources`, wait several minutes for cluster activity to settle. You can check cluster status with the `crm_mon -r1` command.

Add Site-specific Files to the Synchronization List

The file `/etc/csync2/csync2_cray.cfg` specifies the Cray-specific files and directories that must be synchronized, as well as small files that are convenient to keep in sync.

IMPORTANT: The `csync2` utility is designed to synchronize small amounts of data. If `csync2` must monitor many directories or synchronize a large amount of data, it can become overloaded and failures may not be readily apparent. Cray recommends that you add only small files to `/etc/csync2/csync2_cray.cfg`. For example, do not synchronize the following files or directories:

- `/home`
- `/home/crayadm/.ssh/authorized_keys`
- `/opt/xt-images` (Cray boot images are very large)

- `/tmp/SEDC_FILES`, if SEDC does not use the PMDB
- `/etc/hosts`
- Very large files

TIP: You can use `scp` to copy a large, static file to the passive SMW, as in this example:

```
smw1:~ # scp -pr /path/file smw2:/path/file
```

For directories and files that may change during the copy operation, you can use the `rsync` command.

1. For each file or directory on the active SMW that you want to synchronize, ensure that the parent directory exists on the passive SMW. In some cases, you must either manually create directories on the passive SMW or copy the directory structure from the active SMW. With either method, be sure that owner, group, and permissions are maintained, because `csync2` can be sensitive to mismatches.

2. Edit the file `/etc/csync2/csync2_cray.cfg` as `root` on the active SMW.

3. To add a file or directory, add the full path (one entry per line) to `/etc/csync2/csync2_cray.cfg`. Comments in this file explain how to make changes.

   IMPORTANT: For a symbolic link, only the link itself is synchronized, not the content (destination) of the symbolic link.

4. Save your changes and exit the editor.

   The `fsync` resource will synchronize the additional files and directories the next time it runs.

Set the Migration Threshold for a Resource

The `set_migration_threshold` command sets the migration threshold for a resource in an SMW HA cluster. A migration threshold is defined as the maximum number of failures (the failcount) allowed for the resource. If the failcount exceeds this threshold, a failover occurs and management of all cluster resources migrates to the other SMW, making it the active SMW. By default, the migration threshold is `1,000,000`.

IMPORTANT: Cray recommends that you either leave migration thresholds at the default values or set them to a very high value until you have experience with SMW HA operation. Migration threshold settings that are too low could cause the resource to be ineligible to run if the failcount exceeds that value on both SMWs. If lower settings are used, Cray recommends that you monitor failcounts regularly for trends and clear the failcount values as appropriate. Otherwise, transient errors over time could push failcount values beyond the migration threshold, which could lead to one of the following scenarios:

- Failovers could be triggered by a transient error condition that might otherwise have been handled by a less disruptive mechanism.
- Failovers might not be possible because both SMWs have exceeded the migration threshold.

NOTE: Execute these commands as `root` on either SMW.

1. Determine the resource name. To display a list of resource names, execute the `crm_resource` command.

```
smw1:~ # crm_resource -l
```

2. Use the `set_migration_threshold` command to change the migration threshold for a resource.

   NOTE: For *`resource`*, specify a resource name. For *`value`*, specify an integer in the range of `0 – 1000000`.

   ```
   smw1:~ # set_migration_threshold resource value
   ```

3. Verify the change.

   ```
   smw1:~ # show_migration_threshold resource
   ```

   For more information, see the `set_migration_threshold(8)` man page.

Configure PMDB Storage

Choose one of these options to configure shared storage for the Power Management Database (PMDB).

- *Configure Mirrored Storage with DRBD for the PMDB* on page 38. Mirrored storage (preferred): An optional pair of disks, one in each SMW, to store PMDB data. In this configuration, the active SMW mounts `/var/lib/pgsql` as a Distributed Replicated Block Device (DRBD) device and communicates replicated writes over a private TCP/IP connection (eth5) to the passive SMW. This is the preferred PMDB configuration to ensure availability of the PMDB data without competition for I/O bandwidth to the SMW root disk or boot RAID file systems.
- *Configure Shared Storage on the Boot RAID for the PMDB* on page 44. Shared storage: A logical disk, configured as a LUN (Logical Unit) or logical volume on the boot RAID. The boot RAID must have sufficient space for `/var/lib/pgsql`.

Cray strongly recommends using either mirrored storage (preferred) or shared storage. An unshared PMDB is split across both SMWs; data collected before an SMW failover will be lost or not easily accessible after failover. For more information, see *Storage for the Power Management Database (PMDB)* on page 8.

Configure Mirrored Storage with DRBD for the PMDB

Prerequisites

IMPORTANT:

If mirrored storage becomes available after the PMDB has been configured for shared storage, use the procedure *Migrate PMDB Data from the Boot RAID to Mirrored Storage* on page 49 instead of this procedure.

Before beginning this procedure:

- Ensure that the SMW HA software is correctly configured and that the HA cluster is running correctly.
- Plan sufficient time for this procedure. Transfering the Power Management Database (PMDB) to a 1 TB disk requires about 10 hours. The SMW HA cluster should be in maintenance mode until the synchronization operation completes. The Cray system (compute and service nodes) can remain up and can run jobs during this period.
- Check `/etc/fstab` to ensure that there is no entry for `phy3`.

- If upgrading or updating the SMW HA system, ensure that the following RPMs are installed on both SMWs and that the version number is 8.4.4 or higher:

  ```
  drbd-bash-completion-8.4.4-0.22.9
  drbd-kmp-default-8.4.4_3.0.101_0.15-0.22.7
  drbd-udev-8.4.4-0.22.9
  drbd-utils-8.4.4-0.22.9
  drbd-pacemaker-8.4.4-0.22.9
  drbd-xen-8.4.4-0.22.9
  drbd-8.4.4-0.22.9
  ```

  If necessary, install or update any missing RPMs with "`zypper install drbd`".

Mirrored storage (preferred): An optional pair of disks, one in each SMW, to store PMDB data. In this configuration, the active SMW mounts `/var/lib/pgsql` as a Distributed Replicated Block Device (DRBD) device and communicates replicated writes over a private TCP/IP connection (eth5) to the passive SMW. This is the preferred PMDB configuration to ensure availability of the PMDB data without competition for I/O bandwidth to the SMW root disk or boot RAID file systems.

This procedure configures the network for DRBD, configures the DRBD disks, and transfers the PMDB data from local disk to the mirrored DRBD disks.

1. Add eth5 to the network files.

a. Log in as root on the first SMW (*smw1*).

```
workstation> ssh root@smw1
```

b. On *smw1*, create the file `/etc/sysconfig/network/ifcfg-eth5` and add the following contents.

```
BOOTPROTO='static'
IPADDR='10.5.1.2/16'
NAME='eth5 SMW HA DRBD Network'
PREFIXLEN='16'
STARTMODE='auto'
USERCONTROL='no'
```

c. In a separate terminal session, log in as root on the other SMW (*smw2*).

```
workstation> ssh root@smw2
```

d. On *smw2*, create the file `/etc/sysconfig/network/ifcfg-eth5` and add the following contents.

```
BOOTPROTO='static'
IPADDR='10.5.1.3/16'
NAME='eth5 SMW HA DRBD Network'
PREFIXLEN='16'
STARTMODE='auto'
USERCONTROL='no'
```

2. Reinitialize the eth5 interface on both SMWs.

```
smw1:~# ifdown eth5; sleep 1; ifup eth5
```

```
smw2:~# ifdown eth5; sleep 1; ifup eth5
```

3. Verify the IP addresses from *smw1*.

```
smw1:~# ping -c3 10.5.1.3
```

4. Configure the firewall to allow eth5 as an internal connection on both SMWs.

   a. Edit the file `/etc/sysconfig/SuSEfirewall2` on both *smw1* and *smw2*.

   b. Locate the line containing the `FW_DEV_INT` variable.

   c. If necessary, add `eth5` to the end of the `FW_DEV_INT` line.

```
FW_DEV_INT="eth1 eth2 eth3 eth4 eth5 lo"
```

   d. Save your changes and exit the editor on both SMWs.

5. Reinitialize the IP tables by executing the `/sbin/SuSEfirewall2` command on both SMWs.

```
smw1:~# /sbin/SuSEfirewall2
```

```
smw2:~# /sbin/SuSEfirewall2
```

6. On the active SMW only, add the new DRDB disk to the SMW HA configuration.

   NOTE: The following examples assume that *smw1* is the active SMW.

   a. Verify that the device exists on both SMWs.

```
smw1:~# ls -l /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1
ls -l /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1
```

```
smw2:~# ls -l /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1
ls -l /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1
```

   b. Determine if the dedicated disk for the PMDB must be formatted. In this procedure, this disk is referred to as PMDISK.

      NOTE: If the PMDISK is already correctly formatted, skip to step *6.f* on page 41.

      This procedure assumes that a disk drive is available for use as a dedicated drive for the PMDB. The drive should be physically located within the rack-mount SMW at slot 4. The drive should be of the specification 1 TB 7.2K RPM SATA 3Gbps 2.5in HotPlug Hard Drive 342-1998, per the SMW Bill of Materials. The device for PMDISK is `/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0`.

   c. Verify that the PMDISK is inserted into the SMW.

```
smw:#fdisk -l \
/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0

Disk /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0:
1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders, total 1953525168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xffdfd1e1

Device Boot      Start         End      Blocks   Id  System
```

d. Create a new primary partition for the PMDISK, and write it to the partition table. If there are any existing partitions on this disk, manually delete them first.

```
smw:#fdisk \
/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)p
Partition number (1-4, default 1): 1
First sector (2048-1953525167, default 2048): [press return]
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-1953525167, default 1953525167): [press
return]
Using default value 1953525167
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

e. Verify that the partition has been created. This should be device `/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1`

```
smw:#fdisk -l \
/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0

Disk /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0:
1000.2 GB, 1000204886016 bytes
81 heads, 63 sectors/track, 382818 cylinders, total 1953525168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xffdfd1e1

                         Device Boot      Start         End      Blocks   Id
System
/dev/disk/by-path/. . .-lun-0-part1       2048  1953525167   976761560   83
Linux
```

f. Navigate to the directory containing the `SMWHAconfig` command.

```
smw1:~# cd /opt/cray/ha-smw/default/hainst
```

g. Execute `SMWHAconfig` to add the DRBD disk. For *`disk-device`*, specify the disk ID of the disk backing the DRBD disk, using either the by-name or by-path format for the device name.
On a rack-mount Dell PowerEdge R815 SMW, the DRBD disk is a partition on the disk in slot 4; for example, `/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1`.

```
smw1:~# ./SMWHAconfig --add_disk=pm-fs --device=/dev/drbd_r0 --directory=/var/lib/pgsql \
--pm_disk_name=/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1
```

7. Reboot the active SMW (*smw1*) and wait for it to boot completely.

8. Reboot the other SMW (*smw2*) and wait for it to boot completely.

9. Correct the permissions for the `/var/lib/pgsql` file on the active SMW.

```
smw1:~# chown postgres:postgres /var/lib/pgsql
smw1:~# chmod 750 /var/lib/pgsql
```

10. Put the SMW HA cluster into maintenance mode while waiting for the DRBD sync operation to complete. When *smw1* and *smw2* rejoin the cluster after rebooting, the primary DRBD disk (in *smw1*) synchronizes data to the secondary disk (in *smw2*). DRBD operates at the device level to synchronize the entire contents of the PMDB disk. A full initial sychronization takes a long time, regardless of the size of the PMDB. The time to synchronize a 1 TB external DRBD disk is approximately 10 hours. The Cray system (service and compute nodes) can be booted and can run jobs during this period.

    IMPORTANT:

    Cray strongly recommends putting the SMW HA cluster into maintenance mode to prevent any failover during the sync operation. If a failover were to occur during this period, the newly-active SMW could have an incomplete copy of PMDB data.

    a. Put the SMW HA cluster into maintenance mode on *smw1*.

```
smw1:~# crm configure property maintenance-mode=true 2> /dev/null
```

    b. Check the status of the DRBD sync operation with either `rcdrbd status` or `cat /proc/drbd`. The `rcdrbd` output is easier to read, but `/proc/drbd` contains more status information and includes an estimate of time to completion.

```
smw1:~# rcdrbd status
drbd driver loaded OK; device status:
version: 8.4.4 (api:1/proto:86-101)
GIT-hash: 599f286440bd633d15d5ff985204aff4bccffadd build by phil@fat-tyre,
2013-10-11 16:42:48
m:res  cs          ro                 ds                     p
mounted         fstype
0:r0   SyncSource  Primary/Secondary  UpToDate/Inconsistent  C  /var/lib/
pgsql  ext3
...    sync'ed:    72.7%              (252512/922140)M
```

```
smw1:~# cat /proc/drbd
version: 8.4.4 (api:1/proto:86-101)
GIT-hash: 599f286440bd633d15d5ff985204aff4bccffadd build by phil@fat-tyre,
2013-10-11 16:42:48
 0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r-----
    ns:695805444 nr:12508 dw:1808112 dr:694131606 al:171 bm:43068 lo:0 pe:2
ua:0 ap:0 ep:1 wo:f oos:260636656
        [=============>......] sync'ed: 72.4% (254524/922140)M
        finish: 2:21:07 speed: 30,768 (29,720) K/sec
```

    For an explanation of the status information in `/proc/drbd`, see the DRDB User's Guide at linbit.com: *http://drbd.linbit.com/users-guide/ch-admin.html#s-proc-drbd*.

11. When the DRBD sync operation finishes, bring the HA cluster out of maintenance mode on *smw1*.

```
smw1:~# crm configure property maintenance-mode=false 2> /dev/null
```

12. Examine the output of `crm status` to ensure that the `ip_drbd_pgsql` is started on *smw1* and that the `Masters` and `Slaves` entries for `ms_drbd_pgsql` display the SMW host names (*smw1* and *smw2*).

```
smw1:~# crm status
Last updated: Thu Jan 22 18:40:21 2015
Last change: Thu Jan 22 11:51:36 2015 by hacluster via crmd on smw1
Stack: classic openais (with plugin)
Current DC: smw1 - partition with quorum
Version: 1.1.11-3ca8c3b
2 Nodes configured, 2 expected votes
23 Resources configured


Online: [ smw1 smw2 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
   .
   .
   .
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (lsb:cray-syslog):      Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
     postgresqld        (lsb:postgresql):       Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
 ip_drbd_pgsql  (ocf::heartbeat:IPaddr2):       Started smw1
 Master/Slave Set: ms_drbd_pgsql [drbd_pgsql]
     Masters: [ smw1 ]
     Slaves: [ smw2 ]
```

Remove the Mirrored Storage Disk for the PMDB

Cray recommends using mirrored storage with the Distributed Replicated Block Device (DRBD) for the power management database (PMDB). However, if mirrored storage must be disabled, use this procedure to remove the mirrored storage disk from the SMW HA configuration.

1. Log in as root on the first SMW (*smw1*).

   ```
   smw1:~# ssh root@smw1
   ```

2. Navigate to the directory containing the `SMWHAconfig` command.

   ```
   smw1:~# cd /opt/cray/ha-smw/default/hainst
   ```

3. Execute `SMWHAconfig` to remove the DRBD disk.

   ```
   smw1:~# ./SMWHAconfig --remove_disk=pm-fs
   ```

4. Stop the DRBD service on both *smw1* and *smw2*.

   ```
   smw1:~# rcdrbd stop
   ```

   ```
   smw2:~# rcdrbd stop
   ```

5. Reboot the active SMW (*smw1*) and wait for it to boot completely.

6. Reboot the other SMW (*smw2*) and wait for it to boot completely.

7. Examine the output of `crm status` to ensure that there are no entries for `postgresqld`, `ip_drbd_pgsql`, and the `Masters` and `Slaves` resources.

```
smw1:~# crm status
Last updated: Mon Jan 26 14:20:16 2015
Last change: Thu Jan 15 10:44:11 2015
Stack: corosync
Current DC: smw2 (167903490) - partition with quorum
Version: 1.1.12-ad083a8
2 Nodes configured
18 Resources configured


Online: [ smw2 smw1 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
    .
    .
    .
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (systemd:llmrd.service):        Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
```

Configure Shared Storage on the Boot RAID for the PMDB

Prerequisites

The SMW HA system can be configured to store the Power Management Database (PMDB) on shared storage, a logical disk configured as a LUN (Logical Unit) or logical volume on the boot RAID.

> **IMPORTANT:** Cray strongly recommends using mirrored storage, if available, for the PMDB; for more information, see *Storage for the Power Management Database (PMDB)* on page 8. To move the PMDB from shared storage to mirrored storage, see *Migrate PMDB Data from the Boot RAID to Mirrored Storage* on page 49.

Before beginning this procedure:

- Ensure that the boot RAID contains a LUN for the PMDB with sufficient space for the data. Use the following command to check the size of `/var/lib/pgsql` on the local disk:

```
smw1:~ # du -hs /var/lib/pgsql
```

- Check that the boot RAID is connected.
- Ensure that the SMW HA software is correctly configured and that the HA cluster is running correctly.
- To capture typescript output from this procedure, do not use a typescript session running directly on the SMW. To save the output of this procedure, use the `script` command to start the typescript session on your local workstation before logging into the SMW, as in this example:

```
workstation> script -af my_output_file
Script started, file is my_output_file
workstation> ssh crayadm@smw1
```

Use this procedure to configure the RAID disk and transfer the power management data base (PMDB) to the power management disk on the shared boot RAID.

1. Shut down the Cray system by typing the following command as `crayadm` on the active SMW (`smw1`).

   ```
   crayadm@smw1:~>xtbootsys -s last -a auto.xtshutdown
   ```

2. Log into the active SMW as `root`, either at the console or by using the actual (not virtual) host name.

   IMPORTANT: You must log in directly as `root`. Do not use `su` from a different SMW account such as `crayadm`.

3. Change to the directory containing the `SMWHAconfig` command.

   ```
   smw1:~ # cd /opt/cray/ha-smw/default/hainst
   ```

4. Use the `SMWHAconfig` command to move the PMDB and configure the required HA resources. In the following command, replace *scsi-xxxxxxxxx* with the persistent device name for the PMDB directory on the boot RAID.

   ```
   smw1:~ # ./SMWHAconfig --add_disk=pm-fs \
   --device=/dev/disk/by-id/scsi-xxxxxxxxx --directory=/var/lib/pgsql
   ```

   This command mounts the PMDB directory (`/var/lib/pgsql`) to the boot RAID, copies the PMDB data, and configures the HA resources `pm-fs` and `postgresqld`.

5. Reboot `smw1` and wait for the reboot to finish.

   ```
   smw1:~ # reboot
   ```

   Before continuing, wait until `smw1` has rejoined the cluster. After the SMW responds to a `ping` command, log into `smw1`, sleep for at least 2 minutes, then execute the `crm_mon -r1` command to verify that `smw1` is online.

6. Reboot `smw2` and wait for the reboot to finish.

   ```
   smw2:~ # reboot
   ```

   Before continuing, wait until `smw2` has rejoined the cluster. After the SMW responds to a `ping` command, log into `smw2`, sleep for at least 2 minutes, then execute the `crm_mon -r1` command to verify that `smw2` is online.

7. Verify that all resources are running.

   a. Display the cluster status.

      ```
      smw1:~ # crm_mon -r1
      Last updated: Mon Oct 27 01:19:23 2014
      Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
      Stack: classic openais (with plugin)
      Current DC: smw1 - partition with quorum
      Version: 1.1.9-2db99f1
      2 Nodes configured, 2 expected votes
      19 Resources configured.
      ```

```
Online: [ smw1 smw2 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 dhcpd  (lsb:dhcpd):    Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (lsb:cray-syslog):      Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
     postgresqld        (lsb:postgresql):       Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
```

Note that `crm_mon` may display different resource names, group names, or resource order on the system.

b. Examine the `crm_mon` output. Verify that each resource has started by looking for `Started smw1` or `Started smw2`. Also look for any failed actions at the end of the output.

c. If not all resources have started or if any failed actions are displayed, execute the `clean_resources` command on either SMW.

> **IMPORTANT:** When running the `clean_resources` command, you must be directly logged in as `root` (instead of using `su` from a `crayadm` login), because `clean_resources` terminates all non-root user sessions.

```
smw1:~ # clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=stonith-1
Cleaning resource on node=smw1 for resource=stonith-2
Cleaning resource on node=smw1 for resource=dhcpd
Cleaning resource on node=smw1 for resource=cray-syslog
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning resource on node=smw1 for resource=ClusterIP2
 ...
Cleaning resources on node smw2
Cleaning resource on node=smw2 for resource=stonith-1
Cleaning resource on node=smw2 for resource=stonith-2
 ...
Cleaning resource on node=smw2 for resource=Notification
```

After running `clean_resources`, wait several minutes for cluster activity to settle. You can check cluster status with the `crm_mon -r1` command. If the output of this command shows only a subset of the SMW HA services, wait for another minute, then check again. For more information, see the `clean_resources(8)` man page.

8. Verify that the Power Management Database is on the boot RAID and that the required PMDB resources are running.

a. Examine the log file `/opt/cray/ha-smw/default/hainst/SMWHAconfig.out` to verify that the Power Management Database disk appears in the `Cluster RAID Disks` section (at the end of the file), as in this example.

```
---------------- Cluster RAID Disks -----------------
07-07 20:47 INFO   MYSQL Database disk = /dev/disk/by-id/scsi-360080e5..xxx
07-07 20:47 INFO   Log disk            = /dev/disk/by-id/scsi-360080e5..xxx
07-07 20:47 INFO   /home disk          = /dev/disk/by-id/scsi-360080e5..xxx
07-07 20:47 INFO   PM database disk    = /dev/disk/by-id/scsi-360080e5..xxx
07-07 20:47 INFO   ***************** Ending of HA software add_disk *****************
```

b. Ensure that the power management file system is mounted by checking for `/var/lib/pgsql` in the output of the `df` command.

```
smw1:~ # df
     Filesystem     1K-blocks      Used Available Use% Mounted on
     /dev/sda2      120811676  82225412  32449332  72% /
     udev            16433608       756  16432852   1% /dev
     tmpfs           16433608     37560  16396048   1% /dev/shm
     /dev/sdo       483807768 197536596 261695172  44% /var/opt/cray/disk/1
     /dev/sdp       100791728  66682228  28989500  70% /home
     /dev/sdq       100791728    484632  95187096   1% /var/lib/mysql
     /dev/sdr        30237648    692540  28009108   3% /var/lib/pgsql
```

c. Check the output of `crm_mon` to ensure that the `pm-fs` and `postgresqld` resources are running.

```
smw1:~ # crm_mon -r1
Last updated: Mon Oct 27 01:19:23 2014
Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
Stack: classic openais (with plugin)
Current DC: smw1 - partition with quorum
Version: 1.1.9-2db99f1
2 Nodes configured, 2 expected votes
19 Resources configured.


Online: [ smw1 smw2 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 dhcpd  (lsb:dhcpd):    Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (lsb:cray-syslog):      Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
     postgresqld        (lsb:postgresql):       Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
```

Move the PMDB Off the Shared Boot RAID

Use these steps to move the PMDB directory, `/var/lib/pgsql`, from the shared boot RAID to local disk on both SMWs.

> **NOTE:** During this procedure, do not use a typescript session running directly on the SMW. To save the output of this procedure, use the `script` command to start the typescript session on your local workstation before logging into the SMW.
>
> **IMPORTANT:** If the Power Management Database (PMDB) is on local SMW disks rather than on mirrored or shared storage, PMDB data collected before an SMW failover will be lost or not easily accessible after failover.

1. Log into the active SMW as `root`.

   > **IMPORTANT:** You must log in directly as `root` (via `ssh`). Do not use `su` from a different SMW account such as `crayadm`.

2. Change to the directory containing the `SMWHAconfig` command.

   ```
   smw1:~ # cd /opt/cray/ha-smw/default/hainst
   ```

3. Use the `SMWHAconfig` command to remove the PM database disk from the boot RAID.

   ```
   smw1:~ # ./SMWHAconfig --remove_disk=pm-fs
   ```

   This command moves the PMDB directory, `/var/lib/pgsql`, to the original location on the active SMW (smw1) and removes the power management resources (`pm-fs` and `postgresqld`) from the HA configuration.

   > **IMPORTANT:** This command does not copy the PMDB data from the boot RAID.

4. Reboot the active SMW and wait for it to boot completely.

   Before continuing, wait until `smw1` has rejoined the cluster. After the SMW responds to a `ping` command, log into `smw1`, sleep for at least 2 minutes, then execute the `crm_mon -r1` command to verify that `smw2` is online and all resources have started.

5. Reboot the passive SMW and wait for it to boot completely.

   Before continuing, wait until `smw1` has rejoined the cluster. After the SMW responds to a `ping` command, log into `smw1`, sleep for at least 2 minutes, then execute the `crm_mon -r1` command to verify that `smw2` is online and all resources have started.

6. Verify that the Power Management Database is not on the shared boot RAID and that the PMDB resources are not running.

   a. Examine the log file `/opt/cray/ha-smw/default/hainst/SMWHAconfig.out` to verify that the Power Management Database disk does not appear in the `Cluster RAID Disks` section (at the end of the file), as in this example.

      ```
      ---------------- Cluster RAID Disks -----------------
      07-07 20:47 INFO     MYSQL Database disk = /dev/disk/by-id/
      scsi-360080e500023bff6000006b3515d9bdf
      07-07 20:47 INFO     Log disk            = /dev/disk/by-id/
      ```

```
scsi-360080e500023bff6000006b1515d9bc9
07-07 20:47 INFO     /home disk          = /dev/disk/by-id/
scsi-360080e500023bff6000006b5515d9c01
07-07 20:47 INFO
***************** Ending of HA software add_disk *****************
```

b. Check the output of `crm_mon` to ensure that the `pm-fs` and `postgresqld` resources are not running.

```
smw1:~ # crm_mon -r1
Last updated: Mon Oct 27 01:19:23 2014
Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
Stack: classic openais (with plugin)
Current DC: smw1 - partition with quorum
Version: 1.1.9-2db99f1
2 Nodes configured, 2 expected votes
19 Resources configured.


Online: [ smw1 smw2 ]

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP2     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP3     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterIP4     (ocf::heartbeat:IPaddr2):       Started smw1
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
 Notification   (ocf::heartbeat:MailTo):        Started smw1
 dhcpd  (lsb:dhcpd):    Started smw1
 fsync  (ocf::smw:fsync):       Started smw1
 hss-daemons    (lsb:rsms):     Started smw1
 stonith-1      (stonith:external/ipmi):        Started smw2
 stonith-2      (stonith:external/ipmi):        Started smw1
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem):    Started smw1
     cray-syslog        (lsb:cray-syslog):      Started smw1
     homedir    (ocf::heartbeat:Filesystem):    Started smw1
     md-fs      (ocf::heartbeat:Filesystem):    Started smw1
     pm-fs      (ocf::heartbeat:Filesystem):    Started smw1
     postgresqld        (lsb:postgresql):       Started smw1
     mysqld     (ocf::heartbeat:mysql): Started smw1
```

7. If necessary, manually copy the PMDB data from the shared boot RAID to the original location of the PMDB for each SMW.

8. If the original location for the PMDB was not on local disk, you can use the `xtmvpmdb` command to move the PMDB to another location such as a dedicated disk. For more information, see the `xtmvpmdb(8)` man page and *Monitoring and Managing Power Consumption on the Cray XC System* (S-0043).

Migrate PMDB Data from the Boot RAID to Mirrored Storage

Prerequisites

Before beginning this procedure:

- Ensure that the mirrored PMDB disk has been configured as specified in *Configure Mirrored Storage with DRBD for the PMDB* on page 38.

- Identify the device name of the boot RAID partition containin the Power Management Database (PDMB).

Use the following procedure to move the PMDB data from shared storage on the boot RAID to the mirrored storage on the DRBD disk.

1. Log into the active SMW as `root`.
2. Put the cluster in maintenance mode.

   ```
   smw1:~# crm configure property maintenance-mode=true 2> /dev/null
   ```

3. Stop `rsms`.

   ```
   smw1:~# rsms stop
   ```

4. Stop `postgresql`.

   ```
   smw1:~# /etc/init.d/postgresql stop
   ```

5. Mount the boot RAID partition previously used by the PMDB.

   ```
   smw1:~# mount boot_RAID_partition /mnt/pgsql_tmp
   ```

6. Back up the existing copy of `/var/lib/pgsql`, if possible.

   ```
   smw1:~# cp -pr /var/lib/pgsql /var/lib/pgsql-backup
   ```

7. Remove the existing contents of `/var/lib/pgsql` on the mirrored disk.

   ```
   smw1:~# rm -rf /var/lib/pgsql/*
   ```

8. Copy the PMDB contents from the boot RAID partition to `/var/lib/pgsql`.

   ```
   smw1:~# cp -pr /mnt/pgsql_tmp/* /var/lib/pgsql
   ```

9. Start `postgresql`.

   ```
   smw1:~# /etc/init.d/postgresql start
   ```

10. Check the `postgresql` status.

    ```
    smw1:~# /etc/init.d/postgresql status
    Checking for PostgreSQL
    9.1.12:                                                    running
    ```

11. Start `rsms`.

    ```
    smw1:~# rsms start
    ```

12. Inspect the status of the `rsms` daemons and the contents of `/var/opt/cray/log/power_management-`*YYYYMMDD*, where *YYYYMMDD* is today's date. If `xtpmd` is running and no database errors are noted, the transfer went properly.

```
smw1:~# rsms status
cluster is in maintenance mode and daemons are not under cluster control
Checking for RSMS service:
erd..                                                          running
Checking for RSMS service:
erdh..                                                         running
Checking for RSMS service:
sm..                                                           running
Checking for RSMS service:
nm..                                                           running
Checking for RSMS service:
bm..                                                           running
Checking for RSMS service:
sedc_manager..                                                 running
Checking for RSMS service:
cm..                                                           running
Checking for RSMS service:
xtpmd..                                                        running
Checking for RSMS service:
erfsd..                                                        running
Checking for RSMS service:
xtremoted..                                                    running
```

13. If the `rsms` status is good, remove the backup of `/var/lib/pgsql`.

14. Wait for the PMDB to sync completely. A full initial sychronization takes a long time, regardless of the size of the PMDB. The time to synchronize a 1 TB external DRBD disk is approximately 10 hours.
Check the status of the DRBD sync operation with either `rcdrbd status` or `cat /proc/drbd`. The `rcdrbd` output is easier to read, but `/proc/drbd` contains more status information and includes an estimate of time to completion.

```
smw1:~# rcdrbd status
drbd driver loaded OK; device status:
version: 8.4.4 (api:1/proto:86-101)
GIT-hash: 599f286440bd633d15d5ff985204aff4bccffadd build by phil@fat-tyre,
2013-10-11 16:42:48
m:res  cs          ro                 ds                     p  mounted
fstype
0:r0   SyncSource  Primary/Secondary  UpToDate/Inconsistent  C  /var/lib/pgsql
ext3
...    sync'ed:    72.7%              (252512/922140)M
```

```
smw1:~# cat /proc/drbd
version: 8.4.4 (api:1/proto:86-101)
GIT-hash: 599f286440bd633d15d5ff985204aff4bccffadd build by phil@fat-tyre,
2013-10-11 16:42:48
 0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r-----
    ns:695805444 nr:12508 dw:1808112 dr:694131606 al:171 bm:43068 lo:0 pe:2 ua:
0 ap:0 ep:1 wo:f oos:260636656
        [=============>......] sync'ed: 72.4% (254524/922140)M
        finish: 2:21:07 speed: 30,768 (29,720) K/sec
```

15. Take the cluster out of maintenance mode.

```
smw1:~# crm configure property maintenance-mode=false 2 > /dev/null
```

Troubleshooting an SMW HA System

The following procedures describe how to troubleshoot issues on an SMW HA system.

Restart Stopped Resources

Use this procedure on either the active or passive SMW. Execute the commands in this procedure as `root`.

1. Use the following commands to check the status of cluster resources:
   - Execute the `crm_gui` command, then check the management display (click on Management in the left pane) to verify that all resources are marked with green circles. For more information, see *crm_gui Command*.
   - Execute the following command as `root` on either SMW.

```
smw1:~ # crm resource status
 stonith-1       (stonith:external/ipmi) Stopped
 stonith-2       (stonith:external/ipmi) Started
 dhcpd  (lsb:dhcpd) Started
 cray-syslog     (lsb:cray-syslog) Started
 ClusterIP       (ocf::heartbeat:IPaddr2) Started
 ClusterIP1      (ocf::heartbeat:IPaddr2) Started
 ClusterIP2      (ocf::heartbeat:IPaddr2) Started
 ClusterIP3      (ocf::heartbeat:IPaddr2) Started
 ClusterIP4      (ocf::heartbeat:IPaddr2) Started
 fsync  (ocf::smw:fsync) Started
 homedir         (ocf::heartbeat:Filesystem) Started
 hss-daemons     (lsb:rsms) Started
 Notification    (ocf::heartbeat:MailTo) Stopped
 ClusterMonitor (ocf::smw:ClusterMonitor):       Started smw1
 Resource Group: HSSGroup
     ml-fs       (ocf::heartbeat:Filesystem) Started
     md-fs       (ocf::heartbeat:Filesystem) Started
     mysqld      (ocf::heartbeat:mysql) Started
```

   The status `Stopped` is usually caused by exceeding the failcount for a resource.

2. Display the failcount data for all resources.

```
smw1:~# show_failcounts
node=smw1 scope=status  name=fail-count-stonith-1 value=0
node=smw1 scope=status  name=fail-count-stonith-2 value=0
node=smw1 scope=status  name=fail-count-dhcpd value=0
node=smw1 scope=status  name=fail-count-cray-syslog value=0
...
```

You can also use the `show_failcount` command to display the failcount data for a single resource on the specified SMW.

NOTE: Replace *smw* with the SMW host name.

```
smw1:~ # show_failcount smw fsync
```

3. Clear the failcounts and return all values to zero.

```
smw1:~ # clear_failcounts

Clearing failcount on node smw1
Clearing failcount on node=smw1 for resource=stonith-1
Clearing failcount on node=smw1 for resource=stonith-2
Clearing failcount on node=smw1 for resource=dhcpd
Clearing failcount on node=smw1 for resource=cray-syslog
Clearing failcount on node=smw1 for resource=ClusterIP
 .
 .
 .
Clearing failcount on node=smw2 for resource=hss-daemons
Clearing failcount on node=smw2 for resource=Notification
Clearing failcount on node=smw2 for resource=ClusterMonitor
Clearing failcount on node=smw2 for resource=ml-fs
Clearing failcount on node=smw2 for resource=md-fs
Clearing failcount on node=smw2 for resource=mysqld
```

4. After all failcounts have been cleared, the resource should be up and running. Check the cluster status again to verify that the resource has been restarted.

```
smw1:~ # crm resource status
 stonith-1      (stonith:external/ipmi) Started
 stonith-2      (stonith:external/ipmi) Started
 dhcpd  (lsb:dhcpd) Started
 cray-syslog    (lsb:cray-syslog) Started
 ClusterIP      (ocf::heartbeat:IPaddr2) Started
 ClusterIP1     (ocf::heartbeat:IPaddr2) Started
 ClusterIP2     (ocf::heartbeat:IPaddr2) Started
 ClusterIP3     (ocf::heartbeat:IPaddr2) Started
 ClusterIP4     (ocf::heartbeat:IPaddr2) Started
 fsync  (ocf::smw:fsync) Started
 homedir        (ocf::heartbeat:Filesystem) Started
 hss-daemons    (lsb:rsms) Started
 Notification   (ocf::heartbeat:MailTo) Started
 ClusterMonitor (ocf::smw:ClusterMonitor):      Started
 Resource Group: HSSGroup
     ml-fs      (ocf::heartbeat:Filesystem) Started
     md-fs      (ocf::heartbeat:Filesystem) Started
     mysqld     (ocf::heartbeat:mysql) Started
```

5. If not all resources have started, execute the `clean_resources` command.

```
smw1:~ # clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=stonith-1
Cleaning resource on node=smw1 for resource=stonith-2
Cleaning resource on node=smw1 for resource=dhcpd
Cleaning resource on node=smw1 for resource=cray-syslog
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning resource on node=smw1 for resource=ClusterIP2
 ...
Cleaning resources on node smw2
Cleaning resource on node=smw2 for resource=stonith-1
Cleaning resource on node=smw2 for resource=stonith-2
 ...
Cleaning resource on node=smw2 for resource=Notification
Cleaning resource on node=smw2 for resource=ClusterMonitor
```

After running `clean_resources`, wait several minutes for cluster activity to settle, then check cluster status with the `crm_mon -r1` command.

Return an SMW to the HA Cluster After It Has Been Powered Off

1. As `root` on either SMW, check the SMW status with the `crm_mon` command.

```
smw1:~ # crm_mon -r1
============
Last updated: Mon Jul 15 15:32:58 2013
Last change: Wed Jun 26 11:35:09 2013 by root via crm_attribute on smw1
Stack: openais
Current DC: smw1 - partition WITHOUT quorum
Version: 1.1.6-b988976485d15cb702c9307df55512d323831a5e
2 Nodes configured, 2 expected votes
16 Resources configured.
============

Online: [ smw1 ]
OFFLINE: [ smw2 ]

 stonith-2      (stonith:external/ipmi):        Started smw1
 dhcpd  (lsb:dhcpd):    Started smw1
...
```

   NOTE: `crm_mon` may display different resource names, group names, or resource order on the system.

2. Determine the cause of the problem and resolve it before continuing with this procedure.

3. On the active SMW, put the passive SMW into standby mode.

   NOTE: Replace *smw2* with the host name of the passive SMW.

```
smw1:~ # crm node standby smw2
```

4. Check the power status of the passive SMW.

   NOTE: Replace *smw2-iDRAC-IP-addr* with the passive SMW's iDRAC IP address.

   ```
   smw1:~ # /usr/bin/ipmitool -I lanplus -U root -H smw2-iDRAC-IP-addr -a chassis
   power status
   Password:
   Chassis Power is off
   ```

   At the `Password:` prompt, enter the `root` password for the iDRAC.

5. If the power status is `off`, use the following command to turn power on.

   NOTE: Replace *smw2-iDRAC-IP-addr* with the passive SMW's iDRAC IP address.

   ```
   smw1:~ # /usr/bin/ipmitool -I lanplus -U root -H smw2-iDRAC-IP-addr -a chassis
   power on
   ```

6. Verify the changed power status.

   NOTE: Replace *smw2-iDRAC-IP-addr* with the passive SMW's iDRAC IP address.

   ```
   smw1:~ # /usr/bin/ipmitool -I lanplus -U root -H smw2-iDRAC-IP-addr -a chassis
   power status
   Password:
   Chassis Power is on
   ```

   At the `Password:` prompt, enter the `root` password for the iDRAC.

7. Wait for the SMW to reboot.

   Before continuing, wait until the SMW has rejoined the cluster. After the SMW responds to a `ping` command, log into the SMW, sleep for at least 2 minutes, then execute the `crm_mon -r1` command to verify that the active SMW is online.

8. Join the passive SMW to the cluster.

   NOTE: Replace *smw2* with the host name of the passive SMW.

   ```
   smw1:~ # crm node online smw2
   ```

9. Verify that all resources are running.

   a. Display the cluster status.

   ```
   smw1:~ # crm_mon -r1
   Last updated: Mon Oct 27 01:19:23 2014
   Last change: Thu Oct 23 15:15:04 2014 by root via crm_attribute on smw2
   Stack: classic openais (with plugin)
   Current DC: smw1 - partition with quorum
   Version: 1.1.9-2db99f1
   2 Nodes configured, 2 expected votes
   19 Resources configured.


   Online: [ smw1 smw2 ]

    ClusterIP      (ocf::heartbeat:IPaddr2):       Started smw1
    ClusterIP1     (ocf::heartbeat:IPaddr2):       Started smw1
   ```

```
ClusterIP2      (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP3      (ocf::heartbeat:IPaddr2):       Started smw1
ClusterIP4      (ocf::heartbeat:IPaddr2):       Started smw1
ClusterMonitor (ocf::smw:ClusterMonitor):       Started smw1
Notification    (ocf::heartbeat:MailTo):        Started smw1
dhcpd  (lsb:dhcpd):    Started smw1
fsync  (ocf::smw:fsync):        Started smw1
hss-daemons     (lsb:rsms):     Started smw1
stonith-1       (stonith:external/ipmi):        Started smw2
stonith-2       (stonith:external/ipmi):        Started smw1
Resource Group: HSSGroup
    ml-fs       (ocf::heartbeat:Filesystem):    Started smw1
    cray-syslog         (lsb:cray-syslog):      Started smw1
    homedir     (ocf::heartbeat:Filesystem):    Started smw1
    md-fs       (ocf::heartbeat:Filesystem):    Started smw1
    pm-fs       (ocf::heartbeat:Filesystem):    Started smw1
    postgresqld         (lsb:postgresql):       Started smw1
    mysqld      (ocf::heartbeat:mysql): Started smw1
```

Note that `crm_mon` may display different resource names, group names, or resource order on the system.

b. Examine the `crm_mon` output. Verify that each resource has started by looking for `Started smw1` or `Started smw2`. Also look for any failed actions at the end of the output.

c. If not all resources have started or if any failed actions are displayed, execute the `clean_resources` command on either SMW.

> **IMPORTANT:** When running the `clean_resources` command, you must be directly logged in as `root` (instead of using `su` from a `crayadm` login), because `clean_resources` terminates all non-root user sessions.

```
smw1:~ # clean_resources
Cleaning resources on node smw1
Cleaning resource on node=smw1 for resource=stonith-1
Cleaning resource on node=smw1 for resource=stonith-2
Cleaning resource on node=smw1 for resource=dhcpd
Cleaning resource on node=smw1 for resource=cray-syslog
Cleaning resource on node=smw1 for resource=ClusterIP
Cleaning resource on node=smw1 for resource=ClusterIP1
Cleaning resource on node=smw1 for resource=ClusterIP2
 ...
Cleaning resources on node smw2
Cleaning resource on node=smw2 for resource=stonith-1
Cleaning resource on node=smw2 for resource=stonith-2
 ...
Cleaning resource on node=smw2 for resource=Notification
```

After running `clean_resources`, wait several minutes for cluster activity to settle. You can check cluster status with the `crm_mon -r1` command. If the output of this command shows only a subset of the SMW HA services, wait for another minute, then check again. For more information, see the `clean_resources(8)` man page.

Cluster Manager Repeatedly Kills an SMW

If the cluster manager repeatedly kills one or both SMWs with the STONITH capability, it usually means that the cluster has lost the heartbeat because of a communication issue. In this situation, check that the `eth2` and `eth4`

cables are connected correctly on each SMW. For more information, see *Network Connections for an SMW HA System*.

Clear an HSS Lock After Failover Occurs During a Mainframe Boot

1. As `crayadm` on the active SMW, determine the lock ID.

```
crayadm@smw1:~> xtcli lock show
Network topology: class 2
======== SM Session Info ========
:3:s0:  mtoken=0
session id:     1
time       :    Sat Feb  2 11:22:16 2013
target type:    rt_node
members:        c0-0
---------------------------------
```

   In this example, the line "`:3:s0: mtoken=0`" indicates that service number 3 (boot manager) holds a lock. The lock ID is shown in the line "`session id: 1`", indicating a lock ID of `1`.

2. On the active SMW, manually clear the lock.

   NOTE: Replace *id-number* with the actual lock ID.

```
crayadm@smw1:~> xtcli lock -u id-number
Network topology: class 2
```

3. Verify that the lock has been cleared.

```
crayadm@smw1:~> xtcli lock show
Network topology: class 2
======== SM Session Info ========
No session found in the SM.
```

4. If the lock remains in place, log on to the active SMW as `root` and restart the RSMS service.

   NOTE: Replace *smw1* with the host name of the active SMW.

```
crayadm@smw1:~> ssh root@smw1
Password:
...
smw1:~ # /etc/init.d/rsms restart
```

5. Ensure that CLE is not running (that is, the boot node is not partially or fully booted) before running `xtbootsys` again.

```
crayadm@smw:~> ping boot
```

Recover System Settings After Failover During Discovery

1. Restore the previously saved HSS database, as described in the NOTES section of the `xtdiscover(8)` man page. The recovery procedure is the same as that for a system with a single SMW.

2. Rerun `xtdiscover`.

Check File Synchronization and Stop Extra corosync Processes

1. Check the current `/var/opt/cray/log/smwmessages-`*timestamp* file for the following error (or other `fsync` errors).

   ```
   While syncing file /etc/corosync/corosync.conf:
   ERROR from peer hex-14: File is also marked dirty here!
   Finished with 1 errors.
   ```

   If no explanation can be found in the log file, continue with the following steps.

2. Check for the `corosync` process on each SMW by executing the following `ps` command on each SMW.

   ```
   smw1:~ # ps h -C corosync
   10840 ?        Ssl    3:45 /usr/sbin/corosync
   smw1:~ # ssh smw2
   ...
   smw2:~ # ps h -C corosync
    7621 ?        Ssl    2:44 /usr/sbin/corosync
   ```

   Each SMW must have one (and only one) `corosync` process. The remaining steps describe how to stop extra `corosync` processes.

3. If you see multiple `corosync` processes, stop the OpenAIS service on both SMWs.

   IMPORTANT: Stopping OpenAIS is likely to trigger a failover.

   ```
   smw1:~ # /etc/init.d/openais stop
   Stopping OpenAIS/corosync daemon (corosync): 1
   .2
   .3
   .4
   .5
   .6
   .7
   .8
   .9
   .10
   .11
   .done OK

   smw2:~ # /etc/init.d/openais stop
   Stopping OpenAIS/corosync daemon (corosync): 1
   .2
   .3
   .4
   .5
   ```

```
.6
.7
.8
.9
.10
.11
.done OK
```

4. Verify that `corosync` is no longer running on either SMW.

```
smw1:~ # ps h -C corosync

smw2:~ # ps h -C corosync
```

5. If `corosync` is still running on either SMW, use the `killall` command to kill the process manually.

```
smw1:~ # killall -9 corosync

smw2:~ # killall -9 corosync
```

6. Once no `corosync` processes are running on either SMW, restart OpenAIS on both SMWs simultaneously.

```
smw1:~ # /etc/init.d/openais start
Starting OpenAIS/Corosync daemon (corosync): starting... OK

smw2:~ # /etc/init.d/openais start
Starting OpenAIS/Corosync daemon (corosync): starting... OK
```

Migrate PMDB Data from Mirrored Storage to the Boot RAID

Prerequisites

NOTE: Mirrored storage with the Distributed Replicated Block Device (DRBD) is preferred for the Power Management Database (PMDB). Do not move the PMDB to the boot RAID unless no other option is available.

Before beginning this procedure:

- Do not disable DRBD in the SMW HA configuration or physically remove the mirrored disks from the SMWs. The mirrored DRBD storage must remain accessible for this procedure.
- Identify temporary storage with sufficient space for `/var/lib/pgsql`, such as the root partition on the SMW or an external storage device.

Use the following procedure to move the PMDB data from mirrored storage to shared storage on the boot RAID.

1. As `crayadm`, shut down the Cray system.

```
crayadm@smw1:~> xtbootsys -s last -a auto.xtshutdown
```

2. Log into the active SMW as `root`.

3. Put the cluster in maintenance mode.

```
smw1:~# crm configure property maintenance-mode=true 2> /dev/null
```

4. Stop `rsms`.

```
smw1:~# rsms stop
```

5. Stop `postgresql`.

```
smw1:~ # /etc/init.d/postgresql stop
```

6. Copy the contents of `/var/lib/pgsql` to temporary storage, preserving permissions and ownership. The following example shows how to copy `/var/lib/pgsql` to a directory on the root partition on the SMW (for example, `/pgsql_tmp`).

```
smw1:~# mkdir /pgsql_tmp
smw1:~# cp -pr /var/lib/pgsql/* /pgsql_tmp
smw1:~# ls -l /pgsql_tmp/
total 12
drwx------ 14 postgres postgres 4096 Mar 13 12:14 data
-rw-r--r--  1 postgres postgres 1224 Feb  4 17:57 initlog
drwx------  2 root     root     4096 Feb  6 14:35 lost+found
```

7. Remove the DRBD disk from the SMW HA configuration as described in *Remove the Mirrored Storage Disk for the PMDB* on page 43.

8. Add the RAID disk to the SMW HA configuration as described in *Configure Shared Storage on the Boot RAID for the PMDB* on page 44.

9. Put the cluster in maintenance mode.

```
smw1:~# crm configure property maintenance-mode=true 2> /dev/null
```

10. Stop `rsms`.

```
smw1:~# rsms stop
```

11. Stop `postgresql`.

```
smw1:~# /etc/init.d/postgresql stop
```

12. Remove the existing contents of `/var/lib/pgsql` on the boot RAID.

```
smw1:~# rm -rf /var/lib/pgsql/*
```

13. Copy the contents of `/var/lib/pgsql` from the temporary location (see step *6* on page 60) to the boot RAID partition. The following example assumes that `/pgsql_tmp` was used as temporary storage.

```
smw1:~# cp -rp /pgsql_tmp/* /var/lib/pgsql
```

14. Start `postgresql`.

```
smw1:~# /etc/init.d/postgresql start
```

15. Check the `postgresql` status.

```
smw1:~# /etc/init.d/postgresql status
Checking for PostgreSQL
9.1.12:                                                    running
```

16. Start `rsms`.

```
smw1:~# rsms start
```

17. Inspect the status of the `rsms` daemons and the contents of `/var/opt/cray/log/power_management-`*YYYYMMDD*, where *YYYYMMDD* is today's date. If `xtpmd` is running and no database errors are noted, the transfer went properly.

```
smw1:~# rsms status
cluster is in maintenance mode and daemons are not under cluster control
Checking for RSMS service:
erd..                                                      running
Checking for RSMS service:
erdh..                                                     running
Checking for RSMS service:
sm..                                                       running
Checking for RSMS service:
nm..                                                       running
Checking for RSMS service:
bm..                                                       running
Checking for RSMS service:
sedc_manager..                                             running
Checking for RSMS service:
cm..                                                       running
Checking for RSMS service:
xtpmd..                                                    running
Checking for RSMS service:
erfsd..                                                    running
Checking for RSMS service:
xtremoted..                                                running
```

18. Take the cluster out of maintenance mode.

```
smw1:~# crm configure property maintenance-mode=false 2 > /dev/null
```