Table of contents

Abstract 2
What is the hpsa driver? 2
  Motivation 2
  Availability 2
  Affected devices 3
Planning for hpsa 4
  What is not changing 4
  What is changing 5
Potential impact of transitioning to hpsa 9
  Shared name space 9
  Name variability 9
  Persistent device naming 10
Deploying the hpsa driver 12
  Deploying hpsa during a new OS installation 12
  Transitioning to hpsa after the OS installation 12
  Deploying hpsa during a controller replacement 13
Conclusion 15
Issues and workarounds 15
For more information 16
Abstract

The hpsa driver is an open-source Linux device driver for HP Smart Array controllers. Red Hat and SUSE include the hpsa driver as part of their standard enterprise Linux distributions.

Unlike its predecessor, the cciss driver, the hpsa driver presents logical disk devices in the standard SCSI namespace. While this change means easier compatibility with system tools and utilities designed to work with SCSI storage, there are some differences in device naming, and other changes of which to be aware when using the hpsa driver.

This paper describes similarities and differences between hpsa and cciss drivers, defines how to determine which driver to use with particular controllers, and outlines the planning and processes for a successful deployment of the hpsa driver.

What is the hpsa driver?

The hpsa driver is an open-source device driver used with HP Smart Array SAS RAID Array controllers. SAS and SATA physical disks connect to a Smart Array controller to create RAID logical disks. The hpsa driver presents the logical disks to the operating system. The hpsa driver queues and manages I/O requests, handles retries and errors, and passes through configuration, monitoring, and management requests to the controller’s firmware.

The hpsa driver evolved from the HP Smart Array cciss driver. We have modernized and streamlined the hpsa driver for direct interaction with the standard SCSI mid-layer of Linux. The hpsa driver builds on the traditions of high performance, open source, and device compatibility that have made the cciss driver a fixture on Linux distributions for more than 12 years.

Unlike the cciss driver, the hpsa driver presents logical disk devices in the standard SCSI namespace. This change improves compatibility with system tools and utilities designed to work with SCSI storage. You should know about these differences along with differences in device naming and other changes when using the hpsa driver.

HP released the hpsa driver in 2008 for VMware, and posted to kernel.org for Linux community review in early 2009. After many refinements, the hpsa driver became a part of standard kernel.org kernels, as well as enterprise Linux releases from SUSE and Red Hat.

Motivation

The hpsa driver is a SCSI driver, while its predecessor was a block-layer driver. Hpsa combines traditional HP Smart Array driver and controller technology with the large body of Linux community development on the SCSI storage layer. Hpsa retains Smart Array reliability but enables these devices to benefit from the continuing development of the Linux SCSI layer. This results in better error handling, enhanced device management, and quick integration with standard system tools and utilities that rely on the SCSI subsystem. These benefits do not affect data or array configuration compatibility. Hpsa maintains compatibility with pre-existing storage configurations and data from systems using the cciss driver.

Availability

Hpsa is available in kernel.org distributions, and is included in industry-standard operating system (OS) vendor releases. Use of the hpsa driver also depends on the controller in use, and this selection mechanism can vary per distribution.

Kernel.org

The hpsa driver in kernel.org distributions has a well-defined mapping process that distinguishes between particular Smart Array controller models. The hpsa driver handles all newer models; the traditional HP cciss driver handles older controller models.

SUSE

The hpsa driver debuted on the SUSE Linux Enterprise Server (SLES) as an optional driver in SLES 11 SP1. For newer Smart Array controllers, there was a choice to use the standard cciss driver, or hpsa. Older Smart Array controllers require the cciss driver. HP recommends using the standard cciss driver in SLES 11 SP1, since the included hpsa driver was an early experimental version. However, ProLiant Gen8 servers require the hpsa driver, since these servers contain
newer controllers that are not compatible with the older cciss driver. When deploying SLES 11 SP1 on HP ProLiant Gen8 servers, obtain a Bootable Driver Kit from SUSE, which contains an updated hpsa driver version.1

As of SLES 11 SP2, the standard driver for newer Smart Array controllers (see Table 1) is hpsa. For HP ProLiant Gen8 servers, an updated hpsa driver is included in the SUSE Bootable Driver Kit (BDK).

Red Hat
The hpsa driver debuted on Red Hat Enterprise Linux (RHEL) 6 as the driver for all newer Smart Array controllers (see Table 1). The standard installation media includes the driver, and hpsa automatically binds to the appropriate Smart Array controller. Older Smart Array controllers continue to use the cciss driver. We recommend using hpsa only with the supported controllers as shown in Table 1.

Affected devices
As noted above, the driver used with a particular controller model varies depending on OS release. In general, the hpsa driver is used for newer controllers, and the cciss driver is used for older controllers. Table 1 lists Smart Array controller models and the corresponding drivers. Older OS releases that do not appear in the table will continue to use the cciss driver for all Smart Array controllers. Supported Smart Array controllers that do not appear in the table will continue to use the cciss driver on all supported OS releases.

Table 1. Driver support

<table>
<thead>
<tr>
<th>Smart Array controller</th>
<th>RHEL5</th>
<th>RHEL 6</th>
<th>SUSE SLES 10 SP4</th>
<th>SUSE SLES 11 SP1</th>
<th>SUSE SLES 11 SP2</th>
<th>Kernel.org</th>
</tr>
</thead>
<tbody>
<tr>
<td>P400</td>
<td>cciss</td>
<td>cciss</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss</td>
<td>cciss</td>
</tr>
<tr>
<td>P800</td>
<td>cciss</td>
<td>cciss</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss</td>
<td>cciss</td>
</tr>
<tr>
<td>P700m</td>
<td>cciss</td>
<td>cciss</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss</td>
<td>cciss</td>
</tr>
<tr>
<td>P212</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss3</td>
<td>hpsa</td>
</tr>
<tr>
<td>P410</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss3</td>
<td>hpsa</td>
</tr>
<tr>
<td>P410i</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss3</td>
<td>hpsa</td>
</tr>
<tr>
<td>P411</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss3</td>
<td>hpsa</td>
</tr>
<tr>
<td>P711</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss3</td>
<td>hpsa</td>
</tr>
<tr>
<td>P712</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss3</td>
<td>hpsa</td>
</tr>
<tr>
<td>P812</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>cciss1</td>
<td>cciss3</td>
<td>hpsa</td>
</tr>
<tr>
<td>P222</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>hpsa2</td>
<td>hpsa2</td>
<td>hpsa</td>
</tr>
<tr>
<td>P420</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>hpsa2</td>
<td>hpsa2</td>
<td>hpsa</td>
</tr>
<tr>
<td>P421</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>hpsa2</td>
<td>hpsa2</td>
<td>hpsa</td>
</tr>
<tr>
<td>P420i</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>hpsa2</td>
<td>hpsa2</td>
<td>hpsa</td>
</tr>
<tr>
<td>P220i</td>
<td>cciss</td>
<td>hpsa</td>
<td>cciss</td>
<td>hpsa2</td>
<td>hpsa2</td>
<td>hpsa</td>
</tr>
</tbody>
</table>

1 See link at end of paper.
Planning for hpsa

There are many similarities and differences between hpsa and cciss drivers. Being aware of these will help with the transition to hpsa.

What is not changing

Most aspects of Smart Array storage device operations remain the same even after a transition to the hpsa driver.

Data formats

The transition to the hpsa driver does not affect compatibility with data stored on logical disks. Data formats and array configurations are a function of the controller firmware, not the device driver software. The driver transition does not affect compatibility with array configuration metadata. The hpsa driver is compatible with array and logical disk configurations from a system running the cciss driver, because configuration data is stored on the logical disks. You generally do not need to reconfigure storage when switching to the hpsa driver.

Health monitoring

The HP Systems Management Homepage (SMH) and Simple Network Management Protocol (SNMP) storage agents fully support hpsa for health and status monitoring.

Configuration

The tools used to configure arrays, logical disks, and controller settings do not change. Online configuration methods include Array Configuration Utility (ACU, ACU-XE), and the command-line version, hpacucli. Offline configuration methods include:

- Booting from the HP Service Pack for ProLiant (SPP) to run ACU
- Using the ROM-Based Setup Utility, called Option ROM Configuration for Arrays (ORCA), during a boot sequence
- Booting the offline Array Configuration Utility
- Using the Intelligent Provisioning utility during a boot sequence

Firmware and driver updates

Firmware and driver version maintenance will continue to be accomplished using HP Smart Components and the HP Service Pack for ProLiant (SPP). You can download the SPP from the HP website. You can obtain driver updates from the HP website as images of the driver update media and as downloads compatible with each distribution's native package management utility, such as Red Hat Package Manager (RPM). You can also download drivers from the Software Delivery

---

1 Cciss is a supported driver in SLES 11, SP1, but an early hpsa version was offered as an optional driver.

2 SLES11 SP1 Bootable Driver Kit (BDK) image from hp.com or drivers.suse.com is required for these controllers.

3 Even though SUSE in-box drivers can support hpsa with these controllers, HP recommends use of cciss driver, which is default driver for these.
Repository (SDR) at http://downloads.linux.hp.com/SDR/. The SDR allows you to use native Linux tools like yum, apt, and zypper to install HP drivers.

**Standard OS utilities**
All the usual OS-specific storage management tools work with hpsa. The administrative activities for partitioning, formatting, creating a file system, managing a logical volume, and device mapping do not change.

**Performance**
The hpsa driver’s performance is generally comparable to the cciss driver for any combination of controller, configuration, and I/O application.

**What is changing**
Hpsa presents array controller and logical disk devices differently than the cciss driver because hpsa is a SCSI lower-level driver and cciss is a block-layer driver. The hpsa driver takes advantage of standard SCSI layer namespace and error handling features, so there are some changes in device naming, numbering, and management. Figure 1 illustrates the I/O layers and the correlation of the hpsa and cciss drivers within the layers.

Figure 1: I/O layers correlate differently to cciss and hpsa drivers.

**SCSI compatibility**
The hpsa driver enables better and faster compatibility with changes in storage utilities by working directly with the standard Linux SCSI storage layer. System utilities or applications that make assumptions about storage device names being /dev/sd* are incompatible with the cciss driver but will work natively with hpsa.
Log and information files

After transitioning to the hpsa driver, log files like `/var/log/messages` contain lines tagged with “hpsa” instead of “cciss”.

Driver information files like `/proc/driver/cciss/cciss#` no longer contain controller information. Now, `/proc/scsi/scsi` or the `lsscsi -long` command provides similar information.

Use the `sysfs` file system to discover and set attributes of the driver. The `sysfs` attribute files provide information about the controller and logical disk devices. Use the command `cat <filename>` to read the values in the files.

Table 2 lists the file name and available driver attributes.

Table 2. Driver attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>File name</th>
</tr>
</thead>
</table>
| Board ID  | /sys/bus/pci/drivers/hpsa/<id>/device (controller family)  
           | /sys/bus/pci/drivers/hpsa/<id>/subsystem_device (controller model)  
           | /sys/bus/pci/drivers/hpsa/<id>/vendor  
           | /sys/bus/pci/drivers/hpsa/<id>/subsystem_vendor |
| Controller firmware version | /sys/class/scsi_disk/c:b:t:l/device/rev  
                               | c:b:t:l represents controller, bus, target, and logical unit number (LUN) |
| Driver version | /sys/bus/pci/drivers/hpsa/module/version |
| IRQ | /sys/bus/pci/drivers/hpsa/<id>/irq |
| Number of logical drives | /proc/scsi/scsi |
| Controller firmware version | /proc/scsi/scsi |
| Current queue depth | /sys/class/scsi_disk/c:b:t:l/device/queue_depth |
| Supported number of commands on controller | /sys/class/scsi_host/host[0-9]/can_queue |
| Scatter-gather entries | /sys/class/scsi_host/host[0-9]/sg_tablesize |
| Logical drive unique ID | /sys/class/scsi_disk/c:b:t:l/device/unique_id |
| Logical drive RAID type | /sys/class/scsi_disk/c:b:t:l/device/raid_level |

You can also control aspects of the driver behavior by using `sys` attribute files, such as rescan. For example, use the following command to scan for device changes:

```
echo 1 > /sys/class/scsi_host/host[0-9]/rescan (replace host[0-9] with proper host number for the controller).
```

A manual rescan is not normally required, but may be useful after hot plugging certain types of devices, like tape drives or new storage enclosures, or after other types of changes to the storage configuration, such as LUN size changes.
Device naming and numbering

Table 3 lists the driver differences that affect device naming and numbering. Figures 2 and 3 illustrate device driver numbering.

Table 3. Device naming and numbering.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>cciss driver</th>
<th>hpsa driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver device numbering</td>
<td>/dev/cciss/c[0-9]d[0-9]p[0-9]</td>
<td>/dev/sd*</td>
</tr>
<tr>
<td></td>
<td>• c[0-9] represents a controller number.</td>
<td>* is replaced by a sequential letter for each logical disk and a sequential number for each partition in the logical disk.</td>
</tr>
<tr>
<td></td>
<td>• d[0-9] represents a logical disk number.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• p[0-9] represents the partition number, when</td>
<td></td>
</tr>
<tr>
<td></td>
<td>partitions are present.</td>
<td></td>
</tr>
<tr>
<td>Logical disk names</td>
<td>/dev/cciss/c[0-9]<em>d[0-9]</em></td>
<td>/dev/sd[a-z]*</td>
</tr>
<tr>
<td></td>
<td>• c[0-9] represents controllers, starting with 0 as first controller.</td>
<td>Controller numbers are not represented in this naming.</td>
</tr>
<tr>
<td></td>
<td>• d[0-9] represents logical disks, starting with 0 as first disk.</td>
<td>Logical disk numbers are represented using sequential letters, starting with “a.”</td>
</tr>
<tr>
<td></td>
<td>For example, /dev/cciss/c3d7 is the 4th controller using the cciss driver, the 8th logical disk.</td>
<td>Logical disk letters “wrap” to 2 characters after 26 devices.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• /dev/sdh represents 8th SCSI device.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• /dev/sdz represents the 26th SCSI device.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• /dev/sdaa represents the 27th SCSI device.</td>
</tr>
<tr>
<td>Partition names</td>
<td>/dev/cciss/c[0-9]d[0-9]p[0-9]</td>
<td>/dev/sd[a-z]<em>{0-9}</em></td>
</tr>
<tr>
<td></td>
<td>Partitions use sequential numbers, starting with 1.</td>
<td>Sequential numbers designate partitions, starting with 1.</td>
</tr>
<tr>
<td></td>
<td>For example, /dev/cciss/c3d7p4 is the fourth cciss controller, 8th logical disk, the 4th partition.</td>
<td>For example, /dev/sdh4 is the 4th partition on the 8th SCSI device.</td>
</tr>
<tr>
<td>Controller names</td>
<td>/dev/cciss/c[0-9]d0</td>
<td>/dev/sg*</td>
</tr>
<tr>
<td></td>
<td>The name of the first disk represents the actual controller. This naming for the controller exists even when there are no configured logical disks.</td>
<td>A SCSI generic node represents the controller. You can use utilities like sgutils and cciss_vol_status with these device names.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use command lsscsi -g to discover which /dev/sg* nodes represent controllers.</td>
</tr>
</tbody>
</table>
Figure 2. ccss driver device numbers include the controller number and logical disk number.

Figure 3. hpsa driver device labels include a logical disk letter.
Potential impact of transitioning to hpsa

Device name changes are the most visible difference between the cciss and hpsa drivers. Remain aware of all potential impacts of these naming differences to ensure a smooth transition.

Shared name space

Before hpsa, the names of all Smart Array devices stood apart from those of other storage devices because they appeared in a separate namespace. Now, with hpsa, Smart Array devices group with standard SCSI devices. All use standard SCSI naming as follows:

- The hpsa Smart Array device namespace is standardized /dev/sd*
- The cciss Smart Array device namespace was specialized /dev/cciss/c[0-9]d[0-9]
- A device name alone is no longer sufficient for identifying a device as a Smart Array.

Name variability

The SCSI device names are assigned sequentially in order of device discovery. If the discovery order changes due to an addition, removal, or some other change to SCSI hardware, the system assigns pre-existing Smart Array devices a new name after the next reboot. A similar issue can occur when using the cciss driver. Since the cciss device shares its namespace with other Smart Array controllers, the addition or removal of other Smart Array controllers can change the name of a cciss Smart Array device.

Adding or changing other SCSI devices may change names of existing hpsa devices. Only changes to Smart Array controllers can change names of existing cciss devices.

For example:

A server uses cciss driver with one Smart Array controller:

```
/dev/cciss/c0d0 Existing Smart Array Controller
```

If you add a new Smart Array controller, the system may discover it before the older controller on the next boot up. The new controller would claim the first device name, c0d0, forcing the system to rename the older controller’s device:

```
/dev/cciss/c0d0 New Smart Array controller
/dev/cciss/c1d0 Existing Smart Array controller
```

Similarly, a server using the hpsa driver with one Smart Array controller and one logical disk:

```
/dev/sda Existing Smart Array controller
```

If you add another SCSI device to the system and then reboot, the system may discover the new SCSI device before discovering the Smart Array controller. The Smart Array device name could change:

```
/dev/sda New SCSI device
/dev/sdb Existing Smart Array controller
```

To avoid this problem, use the device name persistence features of the Linux udev system. Udev enables alternate device naming mechanisms, avoiding the need to employ specific device names in configuration files. This enhances your system’s stability.

Name transitions during OS changes

Changing the operating system on a server could alter the way it presents Smart Array devices. This is because the new OS may change the driver used with a previously installed Smart Array controller. Complete the following tasks before you upgrade or install a new OS on a server:

- Determine which Smart Array controllers are installed in the server.
- Using Table 1, determine which, if any, of these controllers will use the hpsa driver after installation of the new OS.
For example:

You install RHEL 6 on a server with P400 and P212 controllers.

P400 controller uses the cciss driver on RHEL 6, according to Table 1.
P212 controller uses the hpsa driver, according to Table 1.

Any logical disks on the P212 will switch to hpsa-style naming as described in “Device naming and numbering” after hpsa activation. The /dev/cciss/c[0-9]d[0-9] names for P212 logical disks will change to /dev/sd* names.

**Transitions due to controller replacement**

Since hpsa is associated with particular controllers, replacing a Smart Array controller with a different model could result in a change of logical disk-device names. The new controller may force the use of a different device driver.

For example:

You install RHEL 6 on a server with a P800 controller using the cciss driver.

You replace the P800 controller with a P812 controller.

P812 controller will use the hpsa driver, as noted in Table 1.

Any logical disks on the P812 will switch to hpsa-style naming as described in Device naming and numbering after hpsa activation.

Similarly, if you replace a controller using the hpsa driver with a model that only supports the cciss driver, the device names will switch from /dev/sd* naming to /dev/cciss/c[0-9]d[0-9] names.

**Transitions between pre- and post-boot environments**

Pre-boot environments may detect Smart Array devices with the cciss driver and use this information to establish configuration files for installation. This could cause a problem if the post-install environment uses the hpsa driver instead of cciss for the installed controller.

For example, some automated installation utilities boot a temporary Linux environment on the server to be installed. Then they perform device discovery to build configuration files for the OS install. If the device discovery environment uses the cciss driver, the configuration files will contain cciss device names that become invalid during the OS install, resulting in “device not found” errors if the OS uses the hpsa driver.

You must update these types of pre-boot environments and dynamic discovery tools to use the hpsa driver on certain controllers. Likewise, scripts or utilities that assume all Smart Array controllers will have /dev/cciss/... device names need updates to account for hpsa.

**Persistent device naming**

The persistent device naming method reduces the potential for problems that could occur when you change from the cciss driver to the hpsa driver:

Unique identifiers on the logical volumes or partitions serve as the basis for persistent device names. The system either generates the persistent device names automatically, or in the case of “by-label”, the user assigns them. The identifiers are as follows:

- **by-label**—Allows the administrator to assign a meaningful name to each volume.
- **by-uuid**—The file system generates a unique identifier as the name.
- **by-id**—The storage system generates a unique identifier when the logical disk is created. The name includes the driver type (scsi or cciss).

Instead of using actual device node names like /dev/cciss/c0d0 or /dev/sda, persistent names are used in configuration files to avoid the problems that can arise if the device node name changes.
Names can change for several reasons, including:

- The number of controllers changes.
- The order of discovery of devices changes.
- The number of configured logical disks changes.
- The driver used with a controller changes.

Most modern distributions use one of the persistent naming methods as the default mounting method to insulate the system from most device name changes. The `by-id` persistent name is not effective for the type of change that occurs during the transition from cciss to hpsa driver. This is because the cciss driver’s name is actually embedded in the cciss `by-id` device names, as seen in Figure 4.

You can run system utilities like the YaST2 partitioner module to change the mounting method of each logical disk or partition to use one of the persistent device naming methods.

Figure 4 illustrates the replacement of a P400 controller using the cciss driver with a P410 controller using the hpsa driver, and demonstrates how persistent names remain consistent across driver types. Although the OS device presentation changes with controller or driver replacement, persistent device names remain consistent through the transition.

Figure 4. Persistent device names remain consistent.
Deploying the hpsa driver

Ideally, you should begin the transition to the hpsa driver during an OS installation. You also can make the change later in certain cases, such as when changing Smart Array controller hardware.

Deploying hpsa during a new OS installation

The optimum time for deploying hpsa occurs during the installation of a new operating system. This avoids nearly all issues with device name changes.

Preparing for the installation

Smart Array devices may appear as /dev/sd*.

Installation is simpler if you limit the configuration to storage devices necessary for the install.

RHEL 6

RHEL 6 installations are the simplest case for hpsa deployment. If you install an hpsa-supported controller in the server, the controller will automatically begin using the hpsa driver during the installation. If you need the cciss driver instead, select and install one of the cciss-supported controllers as shown in Table 1.

SLES 11 SP1

ProLiant Gen8 servers and controllers require the use of and updated hpsa driver, available in a Bootable Driver Kit (BDK) package from SUSE*. Use the BDK to replace the older, experimental hpsa driver (version 2.0.2-1) shipped with SLES 11 SP1, which should not be used for any controllers.

For ProLiant servers before Gen8, use the cciss driver for all supported Smart Array controllers.

Transitioning to hpsa after the OS installation

HP recommends deploying hpsa only during OS installations.

In some environments, a post-installation transition from cciss to hpsa can be complicated. But you can still transition to the hpsa driver if you mount the affected devices using the persistent naming method and if you do not use any of the devices as the boot device.

---

* See link at end of paper.
Figure 5 illustrates the OS device presentation changes that occur when transitioning to hpsa after the OS installation.

Figure 5. OS device presentation changes with driver replacement

Pre-transition tasks
Before transitioning from the cciss driver to the hpsa driver, complete the following processes:

1. Schedule downtime for the transition.
2. Perform and verify a complete system backup.
3. Record physical and logical storage configuration and device mapping.
4. Identify the devices affected by the driver transition.
5. Ensure that none of the affected devices is a boot device.
6. Mount the affected devices using a persistent device naming method, either by-label, or by-uuid.
7. Save a copy of the OS configuration files for future reference.

Post-transition tasks
1. Reboot the server.
2. Verify all appropriate Smart Array devices are accounted for by /dev/sd* names.
3. Perform a full system backup.
4. Verify the backup media.

Deploying hpsa during a controller replacement
Replacing a Smart Array controller with a different model may require you to change drivers. Since newer controllers use hpsa, and older models use the cciss driver, you may need to switch to the hpsa driver when you replace a Smart Array controller with a newer model. Even though the data on disk format and the array configuration is entirely compatible between the controllers, you need to anticipate the potential device name changes and plan for them as part of the replacement process. For example, if a server is running RHEL 6, and you need to replace a P400 controller with a P410 controller, the driver used with the controller will need to change from cciss to hpsa.
Figure 6 illustrates the potential device name changes that may occur due to a controller replacement.

Pre-replacement tasks
Replacing a Smart Array controller requires a level of planning identical to the previously discussed case of activating the hpsa driver, since replacement can force a driver transition. We recommend these steps:

1. Schedule downtime for replacing the controller.
2. Follow the instructions for the installation and replacement of controllers contained in the HP Smart Array Controllers for HP ProLiant Servers User Guide.
3. Perform and verify a complete system backup.
4. Record physical and logical storage configuration and device mapping.
5. Identify the devices a driver transition would affect during the replacement. Verify that you are mounting the affected devices using a persistent device naming method, either by-label, or by-uuid, and that the devices are NOT required to boot the server.
6. Save copies of relevant OS configuration files for future reference.
7. For SLES 11 SP1 with Gen8 servers: obtain and install new hpsa version in a Bootable Driver Kit from SUSE.\(^5\)

Performing the replacement

1. Shut down the server.
2. Disconnect internal or external cables connecting to the old controller, noting the connection ports.
3. Remove the old Smart Array controller.
4. Install the new Smart Array controller, preferably in the same expansion slot.
5. Reconnect the internal and external cables to ports on the new controller that correspond to the connections for the previous controller.

\(^5\) See link at end of paper.
Post-replacement tasks
1. Reboot the server.
2. Verify all expected Smart Array devices are accounted for by /dev/sd* names.
3. Perform a full system backup.
4. Verify backup media.

Conclusion

The new Linux Smart Array driver, hpsa, improves error handling, device management, and integration with standard system tools and utilities. Keep in mind possible changes to device names when implementing hpsa. You can achieve a smooth transition to hpsa by implementing persistent names, which assigns unique identifiers to logical volumes and partitions. By planning for the deployment of hpsa and understanding the changes that occur, you can implement the hpsa driver and enjoy its benefits.

Issues and workarounds

Table 3 provides a list of the possible issues and solutions that could occur during the upgrade.

Table 3. Solutions are available for these upgrade issues

<table>
<thead>
<tr>
<th>Details</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hpsa driver did not load during Red Hat RHEL6 installation</td>
<td>Depending on the controller type, RHEL6 uses either cciss or hpsa for Smart Array controllers. For information on supported drivers, see Table 1. Select and install a controller that uses hpsa on RHEL6.</td>
</tr>
<tr>
<td>Hpsa driver did not load during SUSE SLES 11 SP1 installation</td>
<td>HP recommends that the default cciss driver be used on SLES 11 SP1 except with ProLiant Gen8 controllers, which support only hpsa on that operating system (P220i, P222, P420, P421, P420i, P822, P721m)</td>
</tr>
<tr>
<td>Unable to determine which controller is providing a particular /dev/sd* device.</td>
<td>Use lsscsi -long command to display details about each device.</td>
</tr>
<tr>
<td>Smart Array logical volumes are no longer visible after activating hpsa driver.</td>
<td>Hpsa uses different device naming for devices. See Device naming and numbering.</td>
</tr>
<tr>
<td>I have a dependency on cciss driver, or cciss-style device names.</td>
<td>Use a cciss-compatible controller, as specified in Table 1.</td>
</tr>
</tbody>
</table>
Logical drive size expansion is not being recognized or updated in the operating system.

Solution
The cciss driver communicated directly with the block layer to inform it of size changes in the underlying logical disk devices. With hpsa driver, one must execute a manual refresh command to alert the SCSI layer that size information has changed.

For example, after increasing the size of a logical volume using Array Configuration Utility, use the following command on RHEL 6 to force a rescan of the device size:

```
    echo 1 > /sys/class/scsi_disk/C:B:T:L/device/rescan
```
(replace C:B:T:L with the controller, bus, target, and LUN number of the affected device)

You can use the command `cat /proc/partitions` to view the size information before and after to verify the rescan process.

For more information
Visit the URLs listed below if you need additional information.

<table>
<thead>
<tr>
<th>Resource description</th>
<th>Web address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link to Bootable Driver Kit containing new hpsa driver version for use with ProLiant Gen8 servers</td>
<td><a href="http://drivers.suse.com/hp/HP-ProLiant-Gen8">http://drivers.suse.com/hp/HP-ProLiant-Gen8</a></td>
</tr>
<tr>
<td>Kernel.org</td>
<td><a href="http://www.kernel.org">www.kernel.org</a></td>
</tr>
</tbody>
</table>

Get connected

hp.com/go/getconnected
Current HP driver, support, and security alerts delivered directly to your desktop