HP ProLiant Server Power Management

Red Hat Enterprise Linux 6.x

Table of contents

Abstract 2
Introduction 2
HP Power Regulator 2
HP Power Capping 4
Power monitoring with HP iLO 4 5
Power capping demonstration with HP iLO 4 5
ProLiant power management with RHEL 6.x 7
Collaborative power control with RHEL 6.x 8
Idle power states (C-States) with RHEL 6.x 9
Additional RHEL 6.x power management features 9
Firmware bug issue under RHEL 6.x 11
Summary 13
For more information 14
Call to action 15
Abstract

Power management is crucial to data center power provisioning. This document provides a brief overview of the processor-based power saving features supported on HP ProLiant servers and the power management features such as Power Regulator, Power Capping and Collaborative Power Control that are embedded in the ProLiant platforms. This document also discusses how these features are used and their relationship to the Red Hat Enterprise Linux (RHEL) 6.x (6.0, 6.1, 6.2 and 6.3) operating system, including new features available with ProLiant Gen8 Intel-based servers and RHEL 6.x.

Introduction

The RHEL 6.x operating system and the HP ProLiant servers together use processor-based features to achieve better power efficiency for processors. The processor-based features include:

- Performance states (P-states) define a set of fixed operating frequencies and voltages, where P0 represents the highest operating frequency and voltage. You can save power by entering P-states with lower frequency and voltage levels. Either the platform firmware or the operating system controls the P-states.
- Power states (C-states), excluding the C0 state, represent idle states and determine the power consumed when a processor is idle. C0 is a non-idle state with higher C-states representing idle conditions with increasing power savings. The operating system controls the C-states.
- Throttle states (T-states) define a set of fixed frequency percentages which can be used to regulate the power consumption and the thermal properties of the processor. ProLiant systems may reserve the use of T-states for the system firmware.

In addition, ProLiant servers are also capable of using the various processor states to support innovative power management features that are operating system independent and are implemented in the hardware and firmware:

- HP Power Regulator provides a facility to efficiently control processor power usage and performance, either statically or dynamically depending on the mode selected.
- HP Power Capping allows an administrator to limit the power consumed by a server.
- HP Dynamic Power Capping has the additional feature of ensuring that the power limit set by an administrator is maintained by reacting to a spike in server workload more rapidly than basic HP Power Capping.

The Power Regulator and Power Capping technologies are designed to work in conjunction with each other. To make the operating system aware of Power Capping, HP provides the Collaborative Power Control technology. This is a two-way communication mechanism established between the operating system and platform firmware, and can be used by the operating system and hardware collaboratively to choose the appropriate performance level for the server. Support for this technology is present in both RHEL 6.x and on ProLiant Gen8 servers.

HP Power Regulator

HP Power Regulator is a configurable processor power usage feature which allows you to choose from several options for the server to manage P-states or to delegate control of regulating P-states to the operating system.

HP Power Regulator is implemented within the firmware on both Intel-based and AMD-based ProLiant servers.¹ ProLiant servers provide the following HP Power Regulator modes, that you can select from the ROM Based Setup Utility (RBSU) or through HP iLO 4:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP Dynamic</td>
<td>The firmware is capable of managing the P-states. However, when the Collaborative Power Control (CPC) setting is enabled in RBSU, the OS and the firmware collaborate to attain the desired frequency for a processor. When CPC is disabled, this mode allows the firmware to exclusively control the P-states of a processor to match the server load. On HP ProLiant Gen8 servers, HP Dynamic is the default mode with the CPC setting enabled.</td>
</tr>
<tr>
<td>HP Static Low</td>
<td>The firmware controls the P-states. The P-state of the processor is static and it is set to the P-state which corresponds to the lowest operating frequency supported by the processor.</td>
</tr>
</tbody>
</table>

¹ For detailed information on HP Power Regulator support across the different generations of ProLiant platforms, see: http://h18013.www1.hp.com/products/servers/management/iLO/sup_servers.html
**HP Static High**
The firmware controls the P-states. The P-state of the processor is static and it is set to P0 which corresponds to the highest operating frequency supported by the processor.

**OS Control**
The RHEL 6.x operating system controls the P-states and it manages the P-states according to the policy set by the administrator via the OS.

For the HP Static Low and HP Static High modes above, you are advised to disable CPC to ensure that the firmware has exclusive control of the P-states. CPC is located within the Advanced Power Management Options in RBSU. This causes RHEL 6.x to report in the `/var/log/messages` file and in the `dmesg` output that CPU frequency scaling is not utilized on the server.

The OS Control mode allows the ProLiant platform firmware to delegate the duty of managing P-states to the RHEL 6.x operating system.

You can adjust the Power Regulator Settings through the RBSU or the HP iLO 4 interface as shown in Figure 1. You must reboot the system to change the transitions to and from the OS Control mode but you can change the system between the other three modes dynamically.

**Figure 1:** Configuring Power Regulator and Power Capping settings via iLO 4

To adjust the CPC setting, you have to access RBSU as shown in Figure 2. Modifying this setting requires a system reboot for the setting to take effect.
Figure 2: Configuring the Collaborative Power Control (CPC) setting via RBSU

HP Power Capping

HP Power Capping satisfies data center power provisioning requirements by allowing the data center administrator to provide a power budget to a single-server or a group of servers. The ProLiant platform enforces that limit by changing the processor P-states and T-states in an operating system independent manner. HP Power Capping is independent of the HP Power Regulator setting and can occur in any setting. When server power is being capped under OS Control mode, the firmware overrides the power management instructions from the operating system for the duration of the capping.

As shown in Figure 1, you can use HP iLO 4 to configure a power cap. HP iLO displays important information about maximum available power for the power supply, the peak observed power, and the minimum observed power for the server. With this information, you can select an appropriate power cap, either by specifying the absolute maximum watts or a percentage of the maximum observed power of the server.

For more information on HP Power Capping, see:
http://www.hp.com/go/dpc

For an in-depth presentation on the HP Power Capping technology, see:
Power monitoring with HP iLO 4

HP iLO 4 supports the facility to monitor current power consumption along historical timelines. As shown in Figure 3, HP iLO 4 displays the current power consumption as well as the peak and average power consumptions for the past 24-hour and 20-minute time periods.

**Figure 3:** HP iLO 4 power meter readings for 24-hour and 20-minute time periods

For more information on the HP Integrated Lights-Out (iLO) management technology, see: www.hp.com/go/iLO

Power capping demonstration with HP iLO 4

This section demonstrates the HP Power Capping functionality by increasing workload on a ProLiant server under the RHEL 6.1 operating system. Figure 4 displays the iLO configuration setting for a server where the capping threshold is set to 180 watts. This means the maximum power consumption will be limited at approximately 180 watts. Figure 5 shows that when power capping is not set, the power consumption of a server increases when there is an increase in workload. You can also see that the maximum power was 227 watts while the minimum power was 209 watts in the past 5 minutes. The average power was 213 watts. Figure 6 demonstrates that when capping is enabled, the platform limits the power consumption to 180 watts even with an increase in workload in order to satisfy the power budget set by the user.
Figure 4: HP Power Capping threshold configuration

Figure 5: Power consumption without capping
ProLiant power management with RHEL 6.x

RHEL 6.x manages the power usage of ProLiant servers by adjusting the processor P-states when the HP Power Regulator setting in RBSU is configured in OS Control mode. Typically within the Linux operating system, a governor dictates the policy, while the actual P-state transition is accomplished by a suitable P-state driver. RHEL 6.x offers a choice of governors, each implementing a different policy ranging from userspace, which enables the user space program (cpuspeed) to directly configure the processor frequency, to performance, which selects the P-state corresponding to the highest supported frequency. The default governor is the ondemand governor, which dynamically adjusts the processor P-states to match the load on the server.

On Intel-based ProLiant platforms RHEL 6.x natively supports the Intel Demand Based Switching with Enhanced Intel SpeedStep® Technology. On AMD-based ProLiant platforms, RHEL 6.x supports AMD’s PowerNow! technology. The following table lists the P-state driver on Intel-based and AMD-based ProLiant G6, G7 or later platforms under OS Control mode.

<table>
<thead>
<tr>
<th>Processor Family</th>
<th>P-state driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Xeon®</td>
<td>acpi-cpufreq</td>
</tr>
<tr>
<td>AMD Opteron™</td>
<td>powernow-k8</td>
</tr>
</tbody>
</table>

In order for RHEL 6.x to manage the power consumption of the processor, the firmware must communicate information about the processor P-states and their associated frequencies to the OS. You can find this information in the file and directories under /sys/devices/system/cpu. Included in the RHEL 6.x media is the cpufreq-util command (installed via the cpufrequtils-007-5.el6.i686.rpm package) that provides information about the P-states of the processors in the system in a user-friendly format. When used without arguments, cpufreq-info displays information about all processor cores, including the P-state driver, the frequency range supported by the processor, the available frequency steps (which are actually the P-states), the available and current governors, and the current frequency. Example 1 shows how cpufreq-info also supports options to display information specific to a CPU.
Example 1: Output for CPU 0 in OS Control mode

```bash
# cpufreq-info -c 0

analyzing CPU 0:
   driver: acpi-cpufreq
CPUs which run at the same hardware frequency: 0
CPUs which need to have their frequency coordinated by software: 0
   maximum transition latency: 10.0 us.
   hardware limits: 1.20 GHz - 2.00 GHz
available frequency steps: 2.00 GHz, 2.00 GHz, 1.90 GHz, 1.80 GHz,
1.70 GHz, 1.60 GHz, 1.50 GHz, 1.40 GHz, 1.30 GHz, 1.20 GHz
available cpufreq governors: ondemand, userspace, performance
   current policy: frequency should be within 1.20 GHz and 2.00 GHz.
   The governor "ondemand" may decide which speed to use
   within this range.
   current CPU frequency is 1.20 GHz.
```

You can dynamically change the governor used under the OS Control mode by modifying the value in the
/sys/devices/system/cpu/cpu*/cpufreq/scaling_governor file for each CPU. RHEL 6.x provides the
`cpufreq-set` command to select the governor. For more information about the `cpufreq-util` and `cpufreq-
set` commands, refer to the RHEL 6.x man pages.

Collaborative power control with RHEL 6.x

When ProLiant servers are under OS Control mode for power management, power capping may still be imposed by the
platform without the knowledge of the operating system. First introduced on Intel-based G6 ProLiant servers and
included in all Gen8 ProLiant servers, OS Control mode enables the server and the OS to collaborate on power
management. HP provides the Collaborative Power Control (CPC) mechanism, which is capable of providing capping-
related feedback to the operating system and can collaborate with the operating system to manage the power
consumption of a server. This combination provides the quick response time of HP Dynamic Power Savings and still
provides correct processor power information to the operating system.

CPC uses the Processor Clocking Control (PCC) interface, which is an interface for coordinating processor performance
between the platform firmware and the operating system. The PCC interface, jointly developed by HP and Microsoft, is
publicly available, allowing other platform vendors the option of implementing it. For more information on PCC, see:
http://www.acpica.org/download/Processor-Clocking-Control-v1p0.pdf

Platform firmware releases for Intel-based ProLiant G6 servers from August 2009 and later include support for CPC.

When a CPC-enabled server is configured in HP Dynamic mode, the firmware does not present P-state information to the
operating system. Instead, the firmware presents the minimum and maximum frequencies the processor supports,
allowing the OS to choose any frequency within that range, rather than restricting the processor to specific P-states. If
the processor is capped at that time for any reason, then the platform firmware will inform the OS that the request could
not be accomplished due to capping. When capping is not configured, the PCC driver still continues to function in lieu of
the P-state driver. Example 2 shows a sample output for CPU 0 for a Gen8 server running under RHEL 6.1. In this
example, notice that the driver is `pcc-cpufreq`. Only the minimum and maximum frequency limits are displayed.
Unlike under OS Control, there are no preset frequency steps.

Example 2: Output for CPU 0 in HP Dynamic mode with CPC enabled

```bash
# cpufreq-info -c 0

analyzing CPU 0:
   driver: pcc-cpufreq
CPUs which run at the same hardware frequency: 0
CPUs which need to have their frequency coordinated by software: 0
   maximum transition latency: 0.00 ms.
   hardware limits: 1.20 GHz - 2.00 GHz
```

8
available cpufreq governors: ondemand, userspace, performance
current policy: frequency should be within 1.20 GHz and 2.00 GHz.
The governor "ondemand" may decide which speed to use
within this range.
current CPU frequency is 1.20 GHz.

Idle power states (C-States) with RHEL 6.x

Processor power use at idle is a crucial factor in determining power consumption of a server when there is no workload to execute. Typically, when a processor does not have work to perform, the operating system places the processor in a halt state signified as C1. Newer generation processors support deep C-states (C2-C6, where C6 is the deepest state), allowing RHEL 6.x to take advantage of these states. The deeper the C-state, the more power consumption the processor can save. Although C-states can significantly reduce power consumption, the drawback of going to a deeper C-state is the latency associated with the time it takes for the processor to wake up from its idle state and resume executing instructions. Information about the C-states for system processors is available in /sys/devices/system/cpu/cpu*/cpuidle/state*.

Note
You can configure the server to not utilize the idle C-states by choosing the No C-states setting in RBSU.

Additional RHEL 6.x power management features

RHEL 6.x provides a comprehensive set of features for managing the power usage of ProLiant servers.

The “Green IT” features introduced in RHEL 6.0⁵ and later offer the user a range of kernel and user-space features to manage server power consumption. With the “tickless when idle” kernel feature, it is possible to reduce the number of wakeups per seconds from 1024 to typically less than 30. For instance, in Figure 7, notice that the “Wakeups-from-idle per second” is below 23. Additional tools are available to monitor the system power consumption. For example, using the PowerTOP⁴ tool (powertop-1.11-4.el6.i686.rpm), you can identify processes that are most responsible for waking a processor up from its idle state and thereby driving up power consumption. Reference the PowerTOP documentation for further reading on what the output of PowerTOP represents, and for tips and tricks on how to best tune the server for maximum power savings.

Figure 7 displays the PowerTOP V1.1 screen output on an idle 1P ProLiant DL360e Gen8 platform with Intel(R) Xeon(R) CPU E5-2420 processor and 2 GB system memory running under RHEL 6.1. The average residency in the deepest supported C-state⁴ is about 49ms. This value is due to the processor being awakened about only 23 per second times from its idle state. The output listing is for a case where the IPMI service has been stopped on the server⁵.

The latest PowerTOP v2.0⁶ — released in May, 2012 — provides many new features. For example, enhanced diagnostic capabilities are available by using the perf subsystem of the Linux kernel. The user can monitor any of five different views by selecting one of the tabs at the top of the screen: Overview, Idle stats, Frequency stats, Device stats, or Tunables. Figure 8 displays the PowerTOP v2.0 screen output for the same system hardware used in the example of Figure 7 running under RHEL 6.3.

⁵ Red Hat Enterprise Linux 6: Green Computing Features
⁶ An introduction to powertop: https://lesswatts.org/projects/powertop/
⁷ ACPI C3 actually corresponds to hardware C6 state which is the deepest C-state supported by the processors on that platform
⁸ Halting the IPMI driver results in a user losing the ability to remotely monitor the server. If the IPMI service is stopped to save power, it is possible to resume the IPMI service with the service ipmi restart command.
⁹ PowerTOP V2.0: https://01.org/powertop/blogs/ceferron/2012/powertop-v2.0-release
The May 2012 release of PowerTOP v2.0 can be compiled on RHEL 6.3; it cannot be compiled successfully on RHEL 6.0, 6.1, and 6.2 because the `libnl` library on each of these versions is out of date. An additional RPM package `pciutils-devel` must be installed on RHEL 6.3. The RPM package is not included in RHEL 6.3 media.

**Figure 7:** PowerTOP v1.11 output on an idle ProLiant DL360e Gen8 running under RHEL 6.1 with no IPMI service

<table>
<thead>
<tr>
<th>Ca</th>
<th>Avg residency</th>
<th>P-states (frequencies)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C0 (cpu running)</td>
<td>(0.2%)</td>
<td>Turbo Mode (1.0%)</td>
<td></td>
</tr>
<tr>
<td>polling</td>
<td>0.0ms (0.0%)</td>
<td>1.91 Ghz (0.0%)</td>
<td></td>
</tr>
<tr>
<td>Cl</td>
<td>0.4ms (0.1%)</td>
<td>1.80 Ghz (0.0%)</td>
<td></td>
</tr>
<tr>
<td>C3 mwait</td>
<td>48.8ms (99.7%)</td>
<td>1400 Mhz (0.0%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1200 Mhz (99.0%)</td>
<td></td>
</tr>
</tbody>
</table>

Wakeup: from idle per second: 22.8 interval: 15.0s

No ACPI power usage estimate available

Top causes for wakeups:
- 65.3% (94.6) `<kernel core>`: hrtimer_start_range_ns (tick_sched_timer)
- 9.1% (13.1) `<kernel core>`: hrtimer_start (tick_sched_timer)
- 9.1% (13.1) `<kernel core>`: hrtimer_start_range_ns (hrtimer_wakeup)
- 2.0% (4.0) `<kernel core>`: usb_hcd_pollrh_status (rh_timer_func)
- 1.7% (2.5) `<interrupt>`: eth3-TxRx-0
- 1.7% (2.4) `<interrupt>`: uhci_hcd:usb3, hpiolo
- 1.4% (2.0) `<kernel core>`: clocksource_watchdog (clocksource_watchdog)
- 1.1% (1.6) `<interrupt>`: ahci
- 1.1% (1.6) events/6: worker_thread (i915_watchdog)
- 0.8% (1.1) `hpsmhid`: hrtimer_start_range_ns (hrtimer_wakeup)
- 0.7% (1.0) `hp-asrd`: hrtimer_start_range_ns (hrtimer_wakeup)
- 0.4% (0.6) `<kernel core>`: __enqueue_rt Entity (sched_rt_period_timer)
- 0.4% (0.5) `<interrupt>`: eth0-TxRx-0
- 0.4% (0.5) `<interrupt>`: eth1-TxRx-0
- 0.4% (0.5) `<interrupt>`: eth2-TxRx-0
- 0.3% (0.5) events/11: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/10: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/9: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/8: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/7: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/6: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/5: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/4: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/3: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/2: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/1: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) events/0: worker_thread (i915_watchdog)
- 0.3% (0.5) events/0: queue_delayed_work (delayed_work_timer_fn)
- 0.3% (0.5) cmahostd: __nf_conntrack_confirm (death_by_timeout)

**Note**
You can see the P-states (frequencies) information in the PowerTOP output only when the HP Power Regulator is configured to OS Control mode.

Figure 8: PowerTOP v2.0 output on an idle ProLiant DL360e Gen8 running under RHEL 6.3 with no IPMI service

Firmware bug issue under RHEL 6.x

When booting into RHEL 6.x, a firmware bug message will be logged as shown in Figure 9. The Linux kernel reports that BIOS occupies the MSR (Model-specific register) address 0x38d. The address represents the fixed-function performance counter control register (IA32_FIXED_CTR_CTRL) defined in the Intel specification. The value 0x330 indicates that the fixed-function performance counter registers (IA32_FIXED_CTR1 and IA32_FIXED_CTR2) are enabled for all ring levels.

This firmware bug message does not impact the `perf` performance counter subsystem because the subsystem is executed continuously (as you can see by the message "Intel PMU driver." in Figure 9), which means the subsystem still works properly. To support the performance monitoring functionality, the subsystem uses the general-purpose performance counters instead of the fixed-function performance counters. This approach avoids the MSR conflict. The `perf` utility in RHEL 6.x provides performance monitoring by selecting the monitored events. For details about performance monitoring events, see the Intel specification identified previously.

Example 3 shows an example of output from the `perf` utility. The command in the example specifies three events to be monitored: UnHalted Core Cycles (r003c), UnHalted Reference Cycles (r013c), and Instruction retired (r00c0). The example proves that the `perf` subsystem still works correctly even if BIOS occupied the fixed-function performance counters. In this example, the `stress` utility is used to make the processor as busy as possible.

**Example 3: Output from `perf` utility**

```bash
# perf stat -e r003c -e r013c -e r00c0 stress --cpu 50 --timeout 120s
stress: info: [5693] dispatching hogs: 50 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5693] successful run completed in 120s

Performance counter stats for 'stress --cpu 50 --timeout 120s':

2,715,777,736,549 r003c
 142,982,564,816 r013c
3,031,663,572,695 r00c0

120.004665335 seconds time elapsed
```

If the customer is still concerned about the fixed-function performance counters being occupied by BIOS, the customer can disable the “Processor Power and Utilization Monitoring” option in RBSU by following these steps (the GUI is shown in Figure 10):

1. Press CTRL-A.
2. Select **Service Options**.
3. Disable the **Processor Power and Utilization Monitoring** option.
4. Reboot the system.
5. After rebooting, the firmware bug message will no longer appear.

---

Summary

HP ProLiant servers are capable of saving power both when the server is under load and when the server is idle. The processor-based power management features supported in the hardware are enabled by the firmware automatically, and are used in close coordination between the firmware and the RHEL 6.x operating system. Typically, you do not have to activate these features; they are already enabled by default.

Several innovative features provide advanced power saving and budgeting features on HP ProLiant servers. These features include the HP Power Regulator, HP Power Capping, HP Dynamic Power Capping, and Collaborative Power Control.
For more information

For additional information, refer to the resources listed below.

<table>
<thead>
<tr>
<th>Resource description</th>
<th>Web address</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP ProLiant Gen8</td>
<td><a href="http://www.hp.com/go/proliantgen8">http://www.hp.com/go/proliantgen8</a></td>
</tr>
<tr>
<td>HP Linux documentation</td>
<td><a href="http://www.hp.com/go/linux-docs">http://www.hp.com/go/linux-docs</a>, select HP Linux Server Management Software</td>
</tr>
<tr>
<td>Power Regulator</td>
<td><a href="http://www.hp.com/servers/power-regulator">http://www.hp.com/servers/power-regulator</a></td>
</tr>
<tr>
<td>Enhanced Intel SpeedStep ® Technology and Demand Based Switching on Linux</td>
<td><a href="http://softwarecommunity.intel.com/articles/eng/1611.htm">http://softwarecommunity.intel.com/articles/eng/1611.htm</a></td>
</tr>
<tr>
<td>Linux cpufreq kernel documentation</td>
<td><a href="http://lxr.linux.no/linux+v2.6.32/Documentation/cpu-freq/">http://lxr.linux.no/linux+v2.6.32/Documentation/cpu-freq/</a></td>
</tr>
<tr>
<td>Linux cpuidle kernel documentation</td>
<td><a href="http://lxr.linux.no/linux+v2.6.32/Documentation/cpuidle/">http://lxr.linux.no/linux+v2.6.32/Documentation/cpuidle/</a></td>
</tr>
<tr>
<td>Intelligent Platform Management Interface (IPMI)</td>
<td><a href="http://www.intel.com/design/servers/ipmi/">http://www.intel.com/design/servers/ipmi/</a></td>
</tr>
<tr>
<td>An introduction to powerTOP</td>
<td><a href="https://lesswatts.org/projects/powertop/">https://lesswatts.org/projects/powertop/</a></td>
</tr>
<tr>
<td>PowerTOP V2.0</td>
<td><a href="https://01.org/powertop/blogs/ceferron/2012/powertop-v2.0-release">https://01.org/powertop/blogs/ceferron/2012/powertop-v2.0-release</a></td>
</tr>
<tr>
<td>LessWatts</td>
<td><a href="http://www.lesswatts.org/">http://www.lesswatts.org/</a></td>
</tr>
<tr>
<td>RHEL on HP servers</td>
<td><a href="http://www.acpica.org/download/Processor-Clocking-Control-v1p0.pdf">http://www.acpica.org/download/Processor-Clocking-Control-v1p0.pdf</a></td>
</tr>
<tr>
<td>Processor Clocking Control Interface Specification</td>
<td><a href="http://lxr.linux.no/linux+v2.6.34/Documentation/cpu-freq/pcc-cpufreq.txt">http://lxr.linux.no/linux+v2.6.34/Documentation/cpu-freq/pcc-cpufreq.txt</a></td>
</tr>
<tr>
<td>Kernel documentation on the Linux PCC implementation</td>
<td><a href="https://events.linuxfoundation.org/slides/lfcs2010_garbee.pdf">https://events.linuxfoundation.org/slides/lfcs2010_garbee.pdf</a></td>
</tr>
</tbody>
</table>
Call to action

Send comments about this document to: TechCom@HP.com

Get connected

hp.com/go/getconnected

Current HP driver, support, and security alerts delivered directly to your desktop

© Copyright 2012 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft is a U.S. registered trademarks of Microsoft Corporation.

5900-2532, Created November 2012