



Host Power Management in VMware vSphere® 5.5

Performance Study

TECHNICAL WHITE PAPER

Table of Contents

Introduction	3
Power Management BIOS Settings	3
Host Power Management in ESXi 5.5	5
Relationship between HPM and DPM	5
HPM Power Policy Options in ESXi 5.5	5
Defining Your Own Power Policy in ESXi	7
Esxtop and HPM	10
Performance Impact of Turbo Mode with C-states	12
Power Savings Evaluation	13
Experimental Environment and Results	13
SPECpower_ssj2008 Results	15
SPECvirt_sc2010 Results	16
WxWidgets Compilation Workload	16
Best Practices	18
References	18

Introduction

Recent increases in density and computing needs in datacenters are straining power and cooling capacities, dictating the need for host power management. Managing the energy consumed can lower operating costs.

vSphere Host Power Management (HPM) is a technique that saves energy by placing certain parts of a computer system or device into a reduced power state when the system or device is inactive or does not need to run at maximum speed. The term *host power management* is not the same as vSphere Distributed Power Management (DPM) [1], which redistributes virtual machines among physical hosts in a cluster to enable some hosts to be powered off completely. Host power management saves energy on hosts that are powered on. It can be used either alone or in combination with DPM.

vSphere handles power management by utilizing Advanced Configuration and Power Interface (ACPI) performance and power states. In VMware vSphere® 5.0, the default power management policy was based on dynamic voltage and frequency scaling (DVFS). This technology utilizes the processor's performance states and allows some power to be saved by running the processor at a lower frequency and voltage. However, beginning in VMware vSphere 5.5, the default HPM policy uses deep halt states (C-states) in addition to DVFS to significantly increase power savings over previous releases while still maintaining good performance.

Power Management BIOS Settings

VMware ESXi™ includes a full range of host power management capabilities. These can save power when an ESXi host is not fully utilized. You should configure your BIOS settings to allow ESXi the most flexibility in using the power management features offered by your hardware and then make your power management choices within ESXi. The following are a few important points to keep in mind:

- By setting power management or Power Profile in the BIOS to “OS control mode” or the equivalent, you can enable ESXi to control some of the CPU power-saving features directly (for example, performance states or DVFS). For example, to achieve this on an HP ProLiant DL580 G7 system, go to **the Power Management Options→HP Power Profile** and then select **Custom**. Then go to **Power Management Options→HP Power Regulator** and select **OS Control Mode** as shown in [Figure 1](#). On a Dell PowerEdge R710 system, go to the **Power Management** option and set the **Power Management** sub-option to **OS Control**. Even if you don't intend to use these power-saving features, ESXi provides a convenient way to manage them. On most systems, the default setting is BIOS-controlled power management. With that setting, ESXi won't be able to manage power, instead it will be managed by the BIOS firmware.
- C1E is a hardware-managed state: When ESXi puts the CPU into the C1 state, the CPU hardware can determine, based on its own criteria, to deepen the state to C1E. Availability of the C1E halt state typically provides a reduction in power consumption, with little or no impact on performance.
- C-states deeper than C1/C1E (typically C3 and/or C6 on Intel and AMD) are managed by software and enable further power savings. You should enable all C-states and the deepest C-state in the BIOS to get the best performance per watt. [Figure 2](#) shows how to enable deep C-states for an HP ProLiant DL580 G7 Intel system. This gives you the flexibility to use vSphere host power management to control their use.
- When “Turbo Boost” or “Turbo Core” is enabled, C1E and deep halt states (for example, C3 and C6 on Intel) can sometimes even increase the performance of certain lightly-threaded workloads (workloads that leave some hardware threads idle). Therefore, you should enable C1E and deep C-states in the BIOS. However, for a very few multithreaded workloads that are highly sensitive to I/O latency, C-states can reduce performance. In these cases, you might obtain better performance by disabling them in the BIOS. Also, C1E and deep C-states implementation can be different for different processor vendors and generations, so your results may vary.
- Some systems have Processor Clocking Control (PCC) technology, which enables ESXi to manage power on the host system indirectly, in cooperation with the BIOS. This setting is usually located under the Advanced Power Management options in the BIOS of supported HP systems and is usually called Collaborative Power

Control. It is enabled by default in the BIOS. With this technology, ESXi does not manage P-states directly. It instead cooperates with the BIOS to determine the processor clock rate. This feature was turned on by default only in ESXi for 5.0 GA up until 5.0 U2 and has been disabled in ESXi for stability reasons. You should not re-enable it [7].

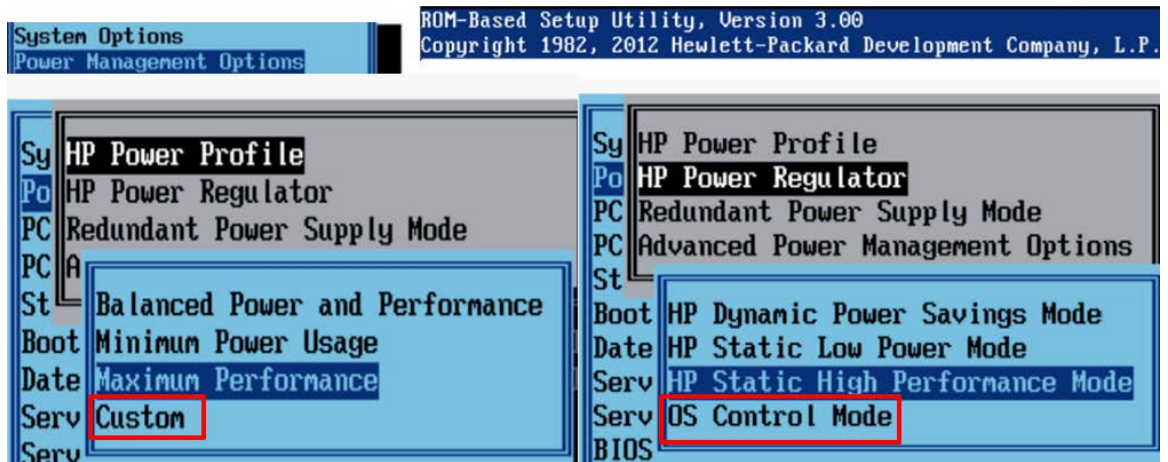


Figure 1. HP ProLiant DL580 G7 BIOS options for DVFS

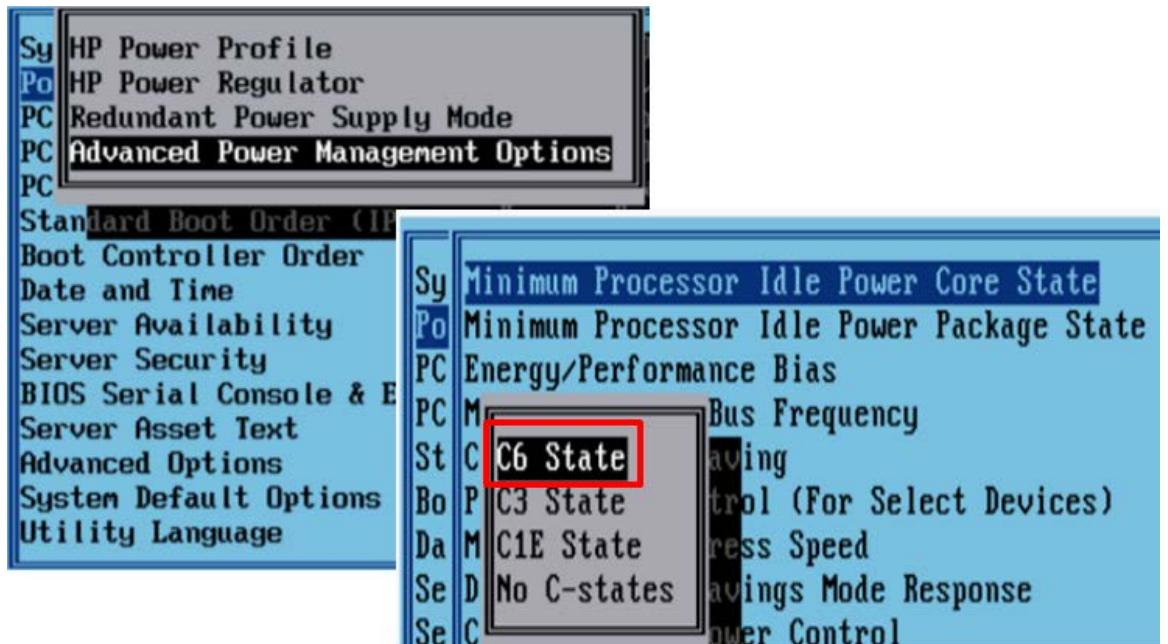


Figure 2. HP ProLiant DL580 G7 BIOS options for selecting C-states

Host Power Management in ESXi 5.5

Relationship between HPM and DPM

vSphere HPM in ESXi 5.5 is designed to reduce the power consumption and to get the optimal performance per watt of ESXi hosts. vSphere DPM approaches the energy efficiency problem from a different angle. It tries to completely power off hosts by consolidating running virtual machines onto the smallest possible set of hosts on a timescale of five minutes or more. HPM, in contrast, attempts to reduce power consumption per host, on a timescale of milliseconds. HPM and DPM are complementary; each one can save power. And they work well together, saving more power than when either is used alone. For more information about DPM, see *VMware Distributed Power Management Concepts and Use* [2].

HPM Power Policy Options in ESXi 5.5

ESXi 5.5 offers four different power policies that are based on using the processor's ACPI performance states, also known as P-states, and the processor's ACPI power states, also known as C-states. P-states can be used to save power when the workloads running on the system do not require full CPU capacity. C-states can help save energy only when CPUs have significant idle time; for example, when the CPU is waiting for an I/O to complete. ESXi 5.5 offers the following power policy options:

- **High Performance:** This power policy maximizes performance, using no power management features. It keeps CPUs in the highest P-state at all times. It uses only the top two C-states (running and halted), not any of the deep states (for example, C3 and C6). High performance is the default power policy for ESXi releases prior to 5.0.
- **Balanced:** This power policy is designed to reduce host power consumption while having little or no impact on performance. This is the default power policy since version 5.0. ESXi has used P-states in the Balanced power policy since 5.0. Beginning in ESXi 5.5, we now also use deep C-states (greater than C1) in the Balanced power policy. Formerly, when a CPU was idle, it would always enter C1. Now ESXi chooses a suitable deep C-state depending on its estimate of when the CPU will next need to wake up.
- **Low Power:** This power policy is designed to save substantially more power than the Balanced policy by making the P-state and C-state selection algorithms more aggressive, at the risk of reduced performance.
- **Custom:** This power policy starts out the same as Balanced, but it allows individual parameters to be modified.

To change power policies using the vSphere Client:

1. Select the host from the inventory and click the **Manage** tab and then the **Settings** tab as shown in [Figure 3](#).
2. In the left pane under **Hardware**, select **Power Management**.
3. Click **Edit** on the right side of the screen.
4. The Edit Power Policy Settings dialog box appears, as shown in [Figure 4](#).

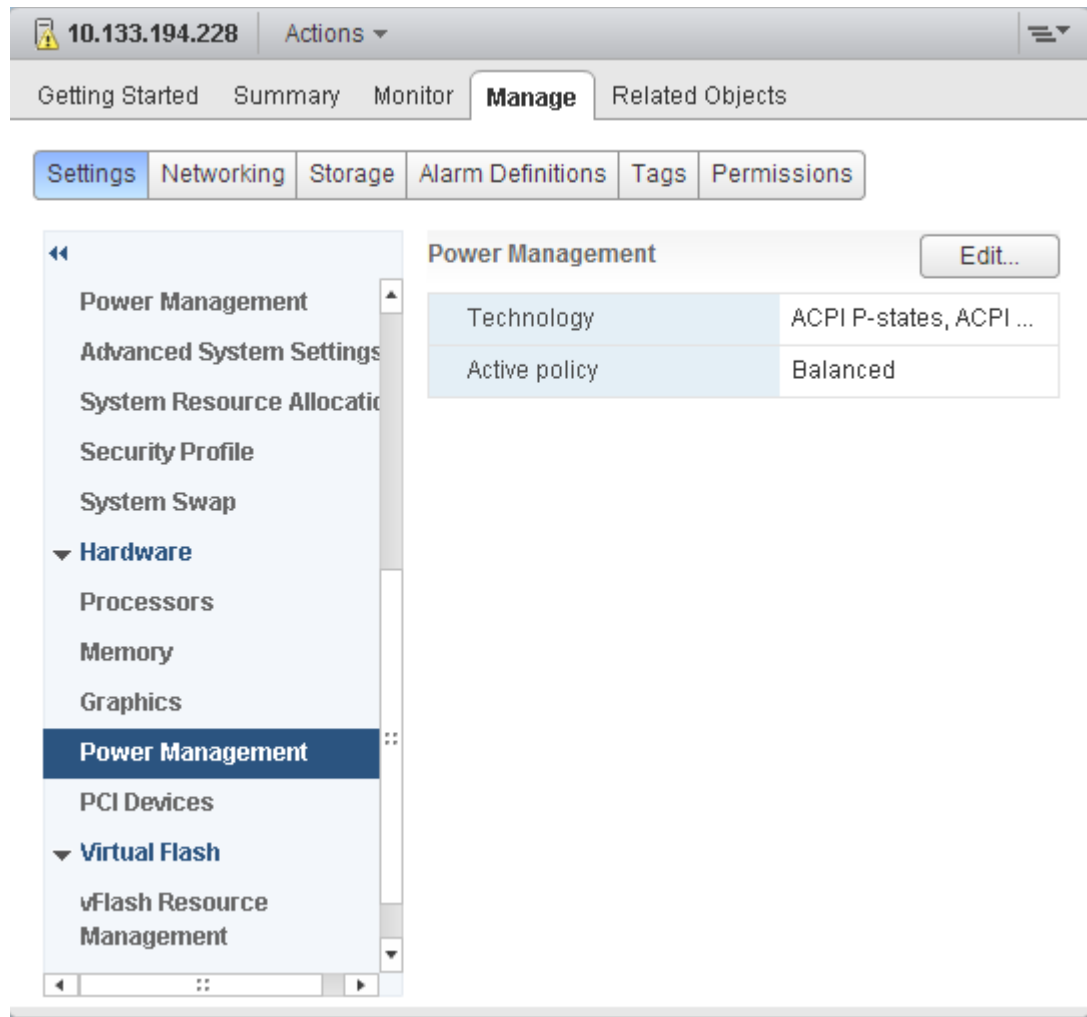


Figure 3. Power Management view from the vSphere Web Client

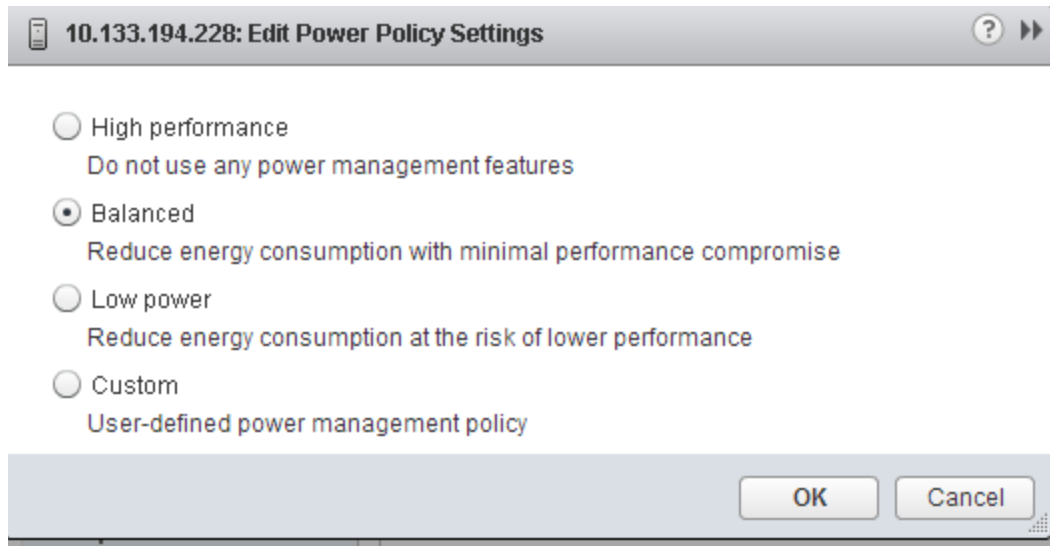


Figure 4. Power policy settings

Defining Your Own Power Policy in ESXi

You can define your own policy by selecting **Custom**. When you use the custom policy for host power management, ESXi bases its power management policy on the values of several advanced configuration parameters (see [Figure 5](#)).

1. Select the host from the inventory and click the **Manage** tab and then **Settings** tab.
2. Under **System**, select **Advanced System Settings**.
3. In the right pane, you can edit the power management parameters that affect the **Custom** policy.

Power management parameters that affect the custom policy have descriptions that begin with the phrase, "In custom policy." All other power parameters affect all power management policies. [Table 1](#) describes these parameters.

10.133.194.228 Actions

Getting Started Summary Monitor **Manage** Related Objects

Settings Networking Storage Alarm Definitions Tags Permissions

System
Licensing
Host profile
Time Configuration
Authentication Services
Power Management
Advanced System Settings
System Resource Allocation
Security Profile
System Swap
Hardware
Processors
Memory
Graphics

Advanced System Settings

Filter

Name	Value	Description
Power.CStateMaxLatency	500	In Custom ...
Power.CStatePredictionC...	110...	In Custom ...
Power.CStateResidency...	5	In Custom ...
Power.ChargeMemoryPct	20	Percentage ...
Power.MaxCpuLoad	60	In Custom ...
Power.MaxFreqPct	100	In Custom ...
Power.MinFreqPct	0	In Custom ...
Power.PerfBias	6	In Custom ...
Power.PerfBiasEnable	1	Use Perfor...
Power.TimerHz	100	In Custom ...
Power.UseCStates	1	In Custom ...
Power.UsePStates	1	In Custom ...

1050 items

Figure 5. Advanced power management options for custom policy

PARAMETER	DESCRIPTION
Power.UsePStates	Use ACPI P-states to save power when the processor is busy.
Power.MaxCpuLoad	Use P-states to save power on a CPU only when the CPU is busy for less than the given percentage of real time.
Power.MinFreqPct	Do not use any P-states slower than the given percentage of full CPU speed.
Power.TimerHz	Controls how many times per second ESXi reevaluates which P-state each CPU should be in.
Power.UseCStates	Use deep ACPI C-states (C2 or below) when the processor is idle.
Power.CStateMaxLatency	Do not use C-states whose latency is greater than this value.
Power.CStateResidencyCoef	When a CPU becomes idle, choose the deepest C-state whose latency multiplied by this value is less than the host's prediction of how long the CPU will remain idle. Larger values make ESXi more conservative about using deep C-states; smaller values are more aggressive.
Power.CStatePredictionCoef	A parameter in the ESXi algorithm for predicting how long a CPU that becomes idle will remain idle. Changing this value is not recommended.
Power.PerfBias	Performance Energy Bias Hint (Intel only). Sets an MSR on Intel processors to an Intel-recommended value. Intel recommends 0 for high performance, 6 for balanced, and 15 for low power. Other values are undefined.

Table 1. Power management parameters for Custom policy

Esxtop and HPM

The esxtop power screen (obtained by pressing the p key) shows power information for the ESXi host. The following are some details of interest, shown in [Figure 6](#):

- Power usage (second row): This is obtained from a sensor in the host's power supply. If the machine does not have a power sensor, or if ESXi is not capable of reading the sensor on this specific type of machine, zero is shown.
- Frequency of each P-state (third row): The screenshot was obtained on an idle ESXi host, which had an Intel® Xeon® E7 4850 @ 2.0GHz with 40 logical processors.
- Which P-states are voted for by the ESXi P-state algorithm (%P0-%P8 in the table heading): The screenshot shows nine different P-states, P0 to P8; P8 has the lowest clock frequency. Because the system is idle, all the cores are in the slowest P-state. On current systems, there are P-state dependencies across logical processors. Each logical processor can vote for a different P-state, but the entire core or entire socket must be in the same P-state, so the hardware selects the shallowest state that any of its logical processors are currently voting for.
- Which deep C-states are voted for by the ESXi C-state algorithm (%C2-%C3 in the table heading): In this display, the available C-states are numbered consecutively without gaps, so C2 corresponds to Intel's C3 deep state and C3 corresponds to Intel's C6 deep C-state.
- %USED and %UTIL: Some of the esxtop counters can be affected by HPM. For example, %UTIL will show higher values for the same workload if the CPU is operating at a lower frequency. %UTIL is the percentage of real time that the logical CPU has been non-idle; %USED is the percentage of its nominal speed that the logical CPU has been operating at (with idle time counted as operating at zero speed). So %USED can be greater than 100 if the CPU has been in turbo mode, and it can be less than %UTIL in P-states slower than P0. This phenomenon can also happen when the BIOS is handling power management instead of ESXi. Be careful in interpreting these values when HPM is enabled (BIOS managed or ESXi managed). For more information, refer to the VMware vSphere Blog post, "What is the difference between PCPU Used and PCPU Utilized?" [\[6\]](#)

```
2:09:01am up 5:49, 569 worlds, 4 VMs, 16 vCPUs; CPU load average: 0
Power Usage: 401W, Power Cap: N/A
PSTATE MHZ: 1996 1995 1862 1729 1596 1463 1330 1197 1064
```

CPU	%USED	%UTIL	%CO	%C1	%C2	%C3	%P0	%P1	%P2	%P3	%P4	%P5	%P6	%P7	%P8
0	8.9	9.4	10	39	8	43	48	0	0	0	0	0	1	0	50
1	3.2	3.5	4	15	6	76	24	0	0	1	2	2	1	2	69
2	11.1	11.9	12	45	6	37	61	0	0	0	0	0	0	0	39
3	0.1	100.0	0	1	4	95	3	0	0	0	0	1	1	1	93
4	11.9	13.0	13	56	5	27	78	0	0	1	0	0	0	0	22
5	2.2	1.8	2	18	3	77	18	0	0	0	1	0	1	1	78
6	6.3	7.2	7	74	5	14	50	0	0	0	0	0	0	0	50
7	0.0	100.0	0	1	2	97	0	0	0	0	0	0	1	0	99
8	12.3	13.3	13	34	4	50	57	0	0	0	0	0	0	0	43
9	1.0	1.3	1	18	9	72	20	0	0	1	0	1	2	2	73
10	6.8	7.6	8	49	4	39	48	0	0	0	0	0	0	1	51
11	2.1	2.4	2	11	7	79	16	0	0	0	0	0	1	1	82
12	2.0	2.4	2	79	7	12	10	0	0	0	0	0	0	0	90
13	0.0	100.0	0	2	2	96	0	0	0	1	0	0	0	0	99
14	6.3	6.7	6	42	5	47	37	0	0	0	0	0	1	0	61
15	3.3	3.4	4	12	4	80	17	0	0	0	0	0	1	1	81
16	0.6	0.7	1	41	8	50	0	0	0	0	0	0	0	0	100
17	0.0	0.1	0	1	2	97	0	0	0	0	0	0	0	0	100
18	2.8	4.8	5	94	1	0	22	0	0	0	0	0	0	0	78
19	0.0	100.0	0	2	6	92	0	0	0	0	0	0	0	0	100
20	14.2	15.5	16	64	8	13	76	0	0	0	0	0	0	0	24
21	0.0	100.0	0	1	5	94	5	0	0	2	0	1	2	1	90
22	5.0	5.6	6	33	5	56	42	0	0	0	0	0	0	0	58
23	0.0	100.0	0	0	5	95	1	0	0	0	0	0	1	2	96
24	5.7	6.5	7	66	7	21	50	0	0	0	0	0	0	0	50
25	0.1	100.0	0	1	4	95	0	0	1	0	1	0	0	1	97
26	13.6	14.5	14	62	8	15	70	0	0	0	0	0	0	0	30
27	0.0	100.0	0	2	3	95	0	0	0	0	0	0	2	1	96
28	8.3	9.1	9	58	6	27	40	0	0	0	0	0	0	0	59
29	0.0	100.0	0	1	7	93	1	0	0	0	0	2	1	1	95
30	7.5	8.1	8	64	8	20	38	0	0	0	0	0	0	0	62
31	0.0	100.0	0	1	3	96	1	0	0	0	0	0	0	2	97
32	14.4	15.4	16	66	5	13	77	0	0	0	0	0	0	0	23
33	0.1	0.3	0	1	3	96	1	0	0	1	0	1	1	2	94
34	3.9	4.1	4	46	9	41	35	0	0	0	0	0	0	0	65
35	0.0	100.0	0	0	5	95	1	0	0	0	0	0	1	0	98
36	3.8	4.1	4	50	11	35	13	0	0	0	0	0	0	0	87
37	0.0	100.0	0	0	7	92	1	0	0	1	0	1	0	1	96
38	3.8	4.2	4	70	8	18	24	0	0	0	0	0	0	0	76
39	0.0	100.0	0	1	5	95	1	0	0	0	0	1	0	1	97

Figure 6. Power usage and different processor and power states in esxtop

Some vsi nodes can be queried to see how much actual time is spent in the deep C-states. For example, by using

```
vsish -e get /power/pcpu/<pcpu_num>/cres
```

ESXi can also compute an estimate of power usage for each virtual machine (in fact, for each process or world running on the ESXi host).

To view the per-virtual machine power consumption in esxtop, press c to show the CPU screen, and then enable the power column on this screen by typing f, J, and then Enter.

Figure 7 shows three virtual machines (VM1, VM2, and VM3) running on ESXi. VM1 and VM3 are not running any workload, which is why the power usage is low: 1W and 4W respectively. VM2 is running some workload and therefore is showing a power consumption of 112W. You will see a substantial amount of power charged to the idle process. This is the baseline power that is used by the machine even when no virtual machines are running.

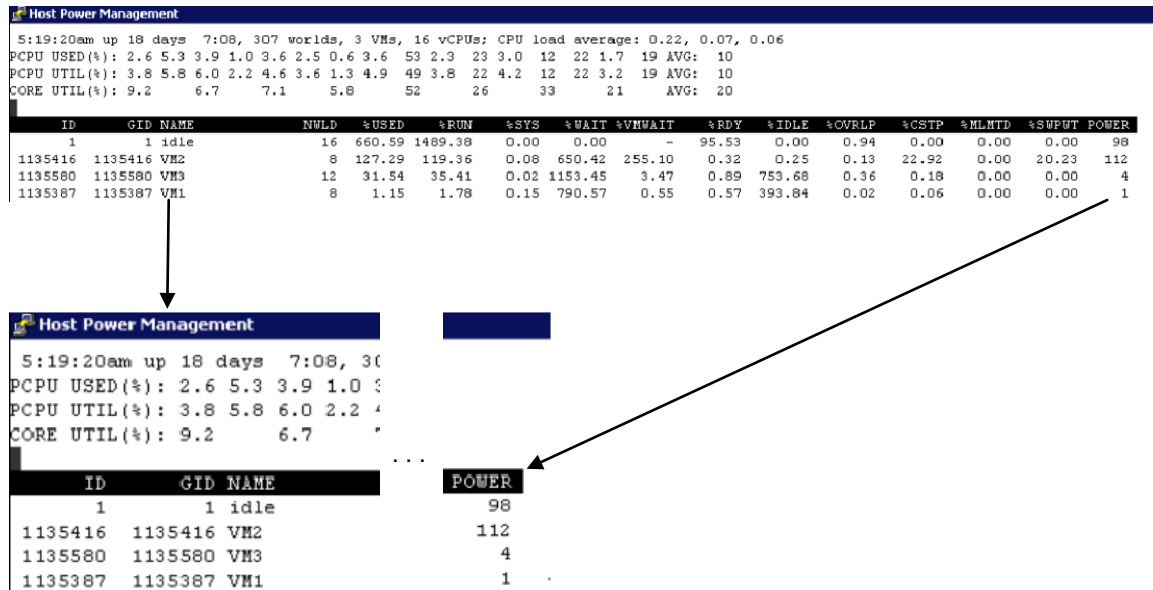


Figure 7. Power usage estimation per virtual machine in esxtop

Performance Impact of Turbo Mode with C-States

We also evaluated the impact of Turbo mode on performance with and without C-states (C1E and deep C-states) enabled. We used a single vCPU 4GB Windows Server 2008 R2 VM with SPECjbb2005 using JDK 1.7 HotSpot JVM. The system used was HP ProLiant DL580 G7 2 X Intel Xeon Processors E7 4850 @ 2.0GHz with 256GB RAM.

Figure 8 shows the impact of Turbo mode on this benchmark with three different settings of C-states: (a) Disabled, (b) with C1E enabled, and (c) deep C-states. The baseline configuration had Turbo mode off and the benchmark scores were compared with the various cases of Turbo mode.

The figure shows that the biggest performance gain with Turbo mode is obtained with deep C-states enabled. An 11.2% increase in throughput is observed due to Turbo mode when deep C-states are enabled. Since Turbo mode is designed to operate within predetermined TDC, TDP, and thermal constraints, deep C-states puts some of the inactive cores in a package in low power mode, giving enough room to increase the frequency of the active cores. Up to a 2.35GHz clock frequency was observed with this setting. (The maximum Turbo frequency on this processor was 2.4GHz.) Without deep C-states enabled, Turbo mode yields only about a 2.5% improvement in performance.

* Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

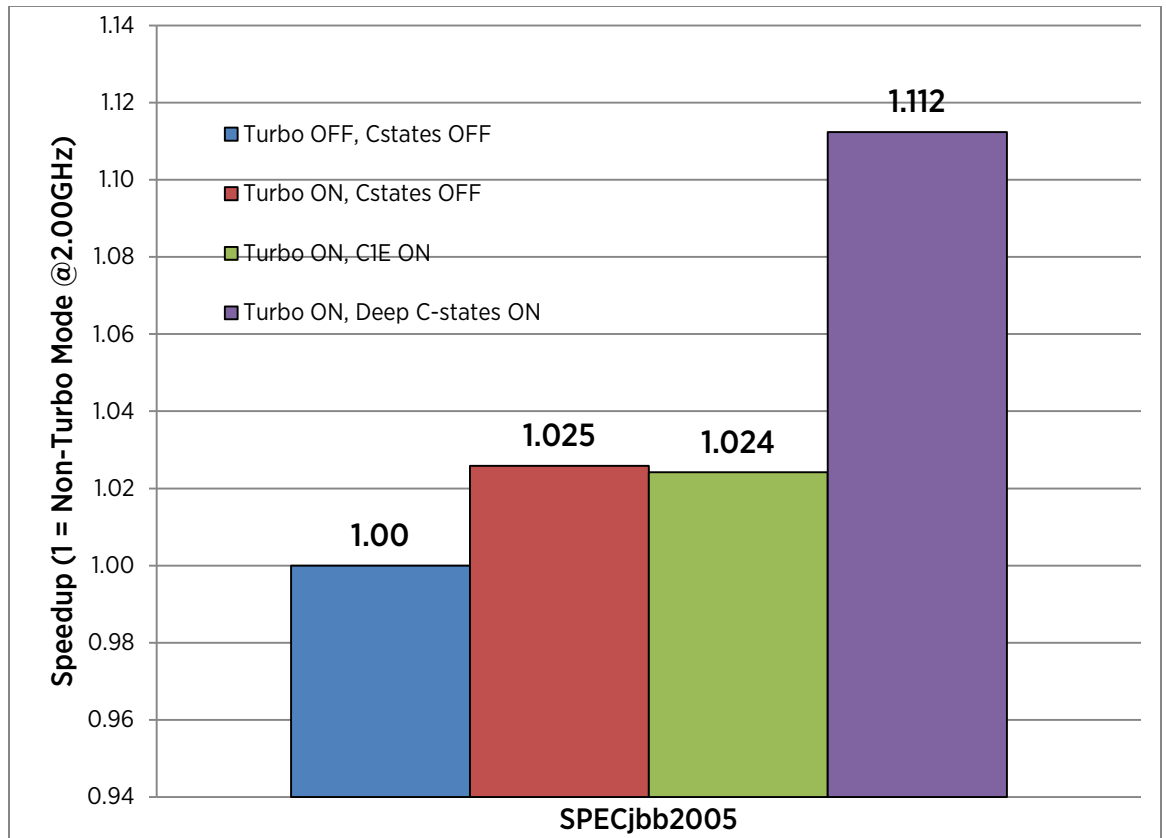


Figure 8. Impact of Turbo Mode and C-states on SPECjbb2005 workload

Power Savings Evaluation

In this section, we used various benchmarks including a micro-benchmark, SPECpower_ssj2008, and SPECvirt_sc2010 to evaluate power savings on different systems.

We used a micro-benchmark CPU-load generator, which simulates application loads running at different levels.

Experimental Environment and Results

ESXi 5.5 was installed on an HP ProLiant DL580 G7 with 2 X Intel Xeon Processor E7 4850 @ 2.0GHz with 256GB RAM system. All the power-related BIOS settings were set per recommendations in the above section. A GW Instek GPM-8212-RS232 power meter was used to measure system power consumption.

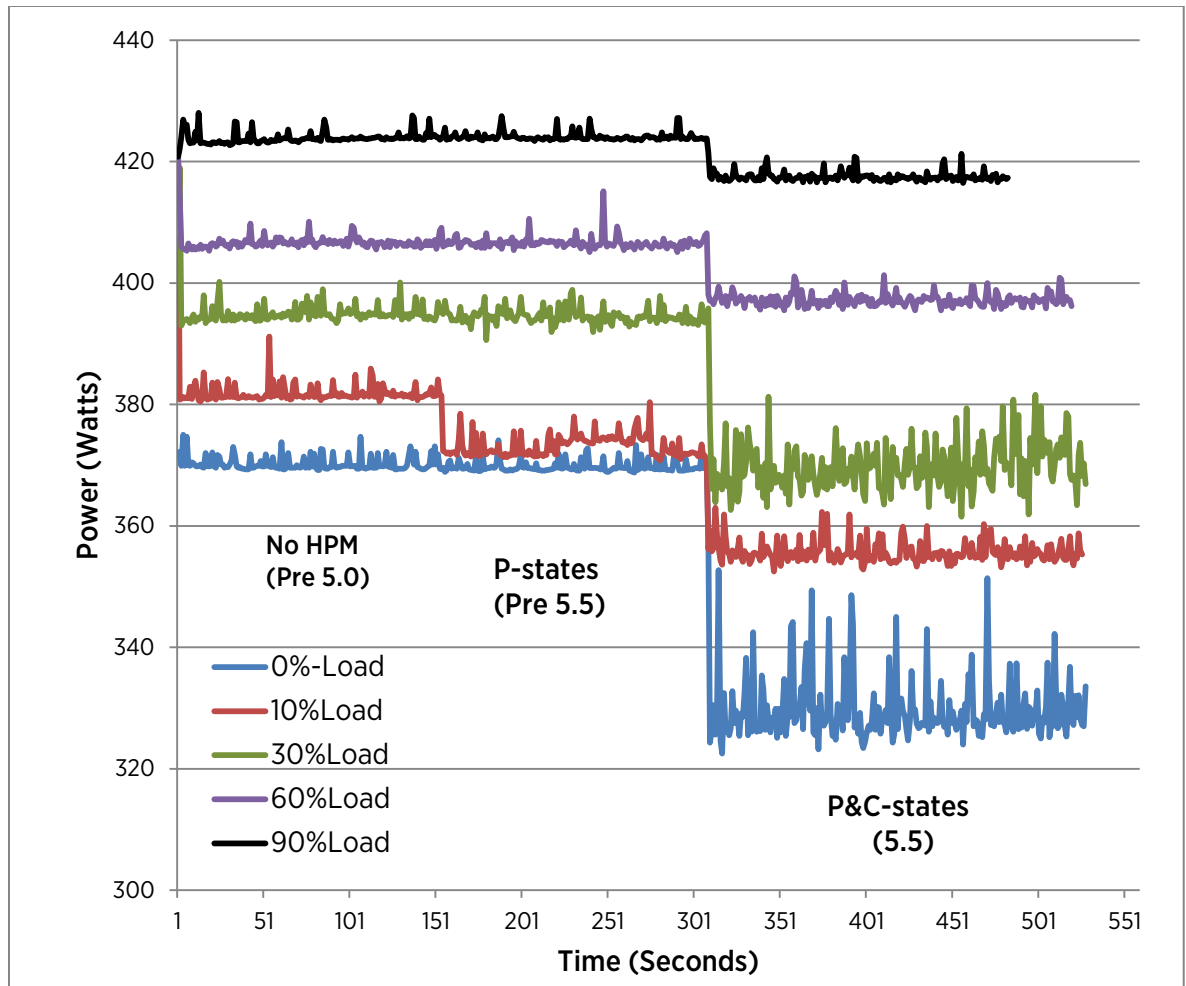


Figure 9. Power consumption at different load levels under different configurations

Four different load configurations are shown in Figure 9. Load-0% is no load, load-10% means the workload utilizes 10% of the CPU ("AVG PCPU UTIL (%)” counter in esxtop), and so on. The graph shows three different regions, *No HPM*, *P-States (Pre 5.1)* and *P&C-States (5.5)*. In the *No HPM* case there is no power management and we selected the High Performance policy. In the *P-states*, we just enabled P-states using the Custom policy. In the *P&C-states*, we use the default Balanced policy of ESXi. This graph shows how we have improved power consumption going from the ESXi pre 5.0 version to ESXi 5.5 using different processor power saving technologies including ACPI P-states and ACPI C-states.

- At load-0%, there is no power savings by going from *No HPM*(pre 5.0) to *P-states*(pre 5.5), because C1E is already putting the cores in a lower frequency state. However, about 11% power is saved when the low policy is enabled because of deep C-states.
- For load-30%, there is about 6.5% power savings from the balanced policy.
- More power savings can be achieved if you switch to Low power policy

SPECpower_ssj2008 Results

We also used SPECpower_ssj2008 to evaluate power savings. This industry standard SPEC benchmark evaluates the power and performance characteristics of server class and multi-node computers [4].

We evaluated the power advantages of using Balanced mode in ESXi 5.5 versus using High Performance mode. The host operating system was installed on a server platform using two Intel Xeon Processors E5-2680 with 16 logical processors on each CPU, giving a total of 32 logical processors with 64GB of physical memory on the platform. We used a Microsoft Windows® Server 2008 Enterprise Edition R2 with SP1 installation as the virtual machine with 16 vCPUs with 60GB vRAM to run the benchmark.

We found that the peak score difference between using High Performance mode and Balanced mode was only about 1.4%. This difference manifests itself down all the way through the load line. This indicates that there was no performance degradation when using the default Balanced mode. Figure 10 shows the percentage power savings for each load-point by using Balanced mode compared to High Performance mode.[†] There is significant power savings in the lower sections of the load line (under 50% load). As CPU activity increases, the power savings are less apparent beyond a 70% load.

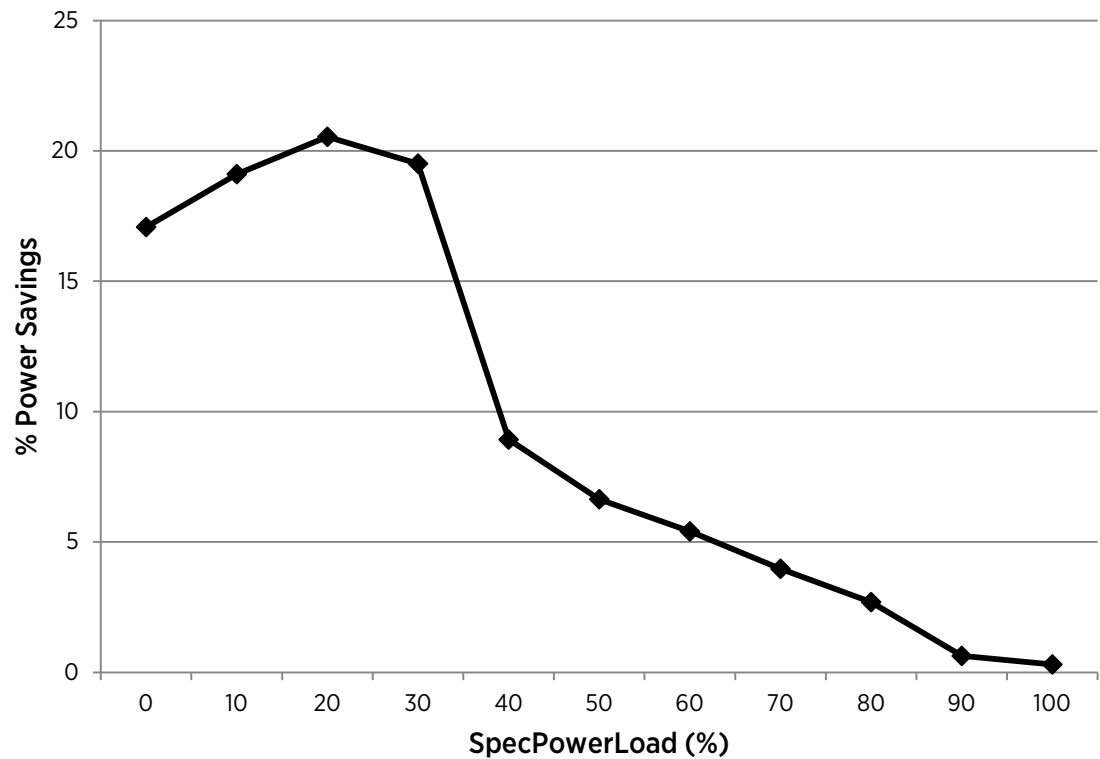


Figure 10. SPECPower_ssj2008 power consumption at different load levels

[†] Instead of reporting SPEC scores in the standard way, we show the workload percentage run on the X axis and the power savings on the Y axis.

SPECvirt_sc2010 Results

We also used SPECvirt_sc2010, which is a SPEC benchmark that evaluates the performance of datacenter servers used in virtualized server consolidation [5]. We evaluated power savings using the same methodology as in the SPECPower benchmark and reported the results in a similar way[†]. In this case, load is increased in units of SPECvirt tiles. We used Linux virtual machines to execute the SPECvirt_sc2010 workload on ESXi 5.5. We observed under 1% run-to-run throughput difference at each load point between High Performance and Balanced mode. As can be seen in Figure 11, there are significant power savings at lower load points. There are no significant power savings beyond a 30% load. 0% load is an idle measurement with no virtual machines.

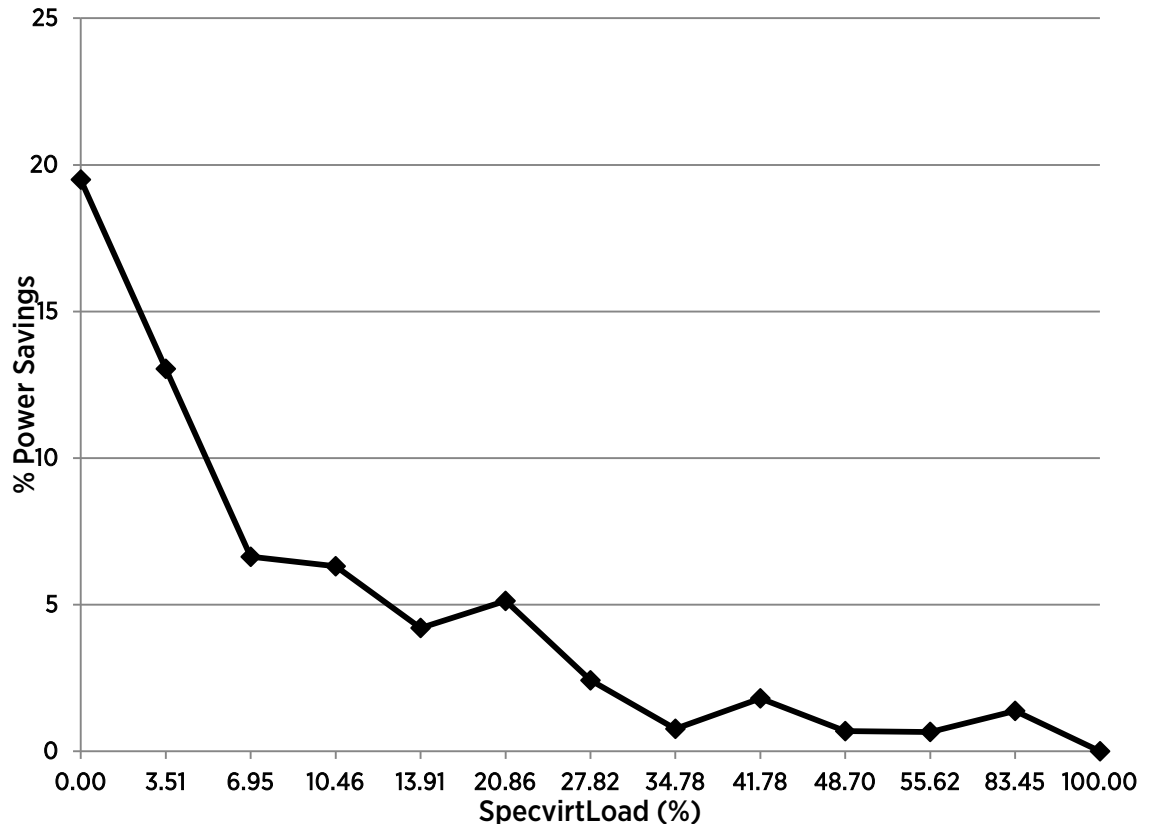


Figure 11. SPECvirt_sc2010 power consumption at different load levels

WxWidgets Compilation Workload

We also evaluated compiling wxWidgets workload, which is a cross platform GUI library [8]. The hardware was a four socket Intel Xeon E5-4650 2.7GHz processor with 512GB RAM.

We ran two similar 10 vCPU, 10GB Windows Server 2008 Enterprise virtual machines in two configurations: High Performance and Balanced. We ran four iterations of the compile workload. Figure 12 shows two phases: compiling and clean up. The compile phase has high CPU utilization and consumed more power compared to the clean up phase. The four iterations show up as crests and troughs. During the compile phase, we save about 10%

[†] This is not standard per SPEC, which specifies that the results be shown with the number of tiles on the X axis and throughput on the Y axis.

power in Balanced mode compared to High Performance mode. In the end when the workloads finish, both virtual machines are idle and we save about 28% power.

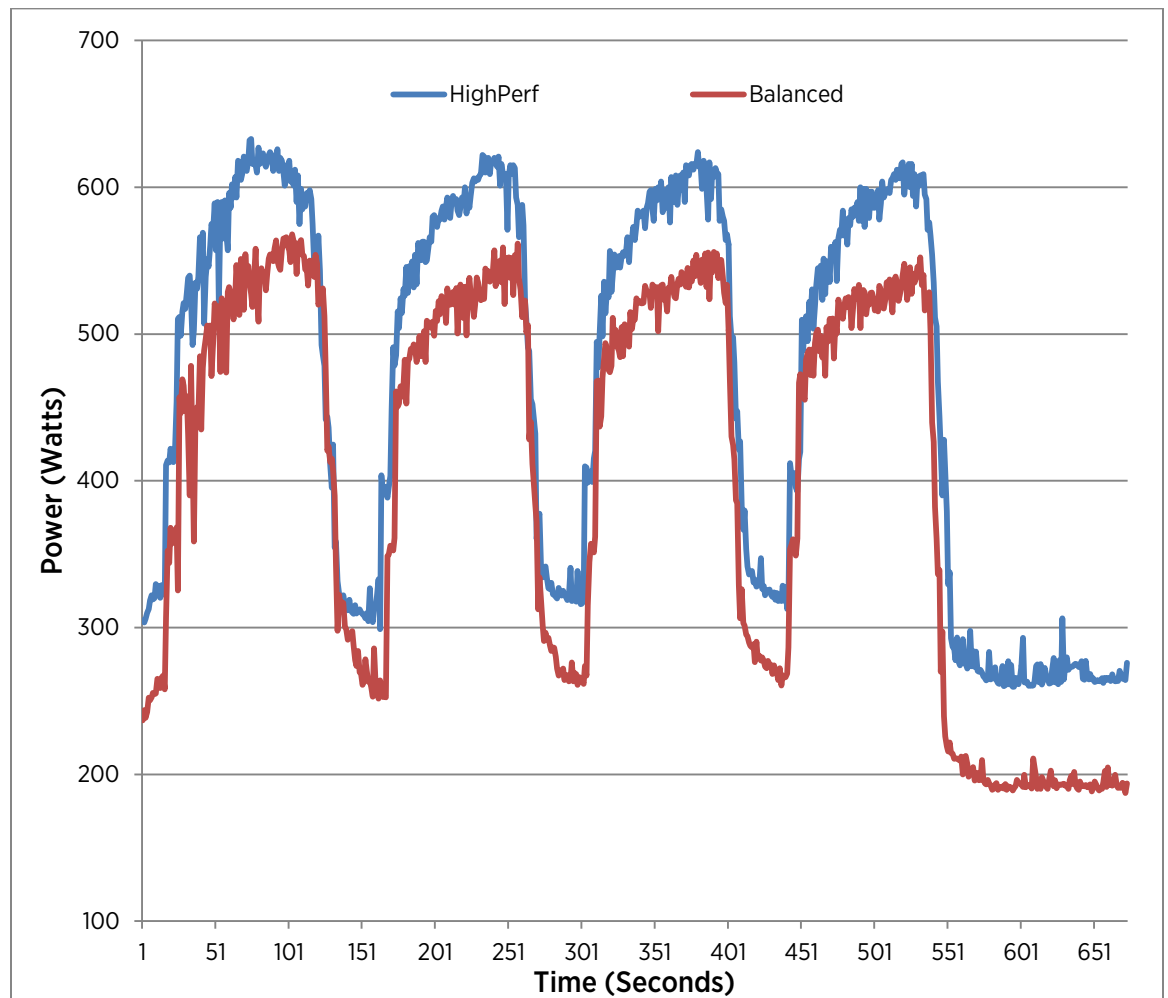


Figure 12. wxWidgets compile Workload with High perf and Balanced Mode

We also ran many other workloads with different configurations (for example, Exchange 2010, DVD Store, VDI workloads from View Planner), showing little to no regression with decent power savings.

Best Practices

Based on the power management concepts and performance evaluation results presented in the previous sections, the following are some best practices to adopt when using HPM:

- Configure your BIOS settings to allow ESXi the most flexibility in using the power management features offered by your hardware. Select the OS Control mode under the BIOS power management options. Enable all C-states and the deepest C-state in the BIOS.
- To achieve the best performance per watt for most workloads, leave the power policy setting at the default, which is Balanced.
- Turn Turbo mode on with all C-states enabled to get the maximum power and performance benefit.
- For applications that require maximum performance, switch the power policy to High Performance. This includes latency-sensitive applications that must execute within strict constraints on response time. For maximum power savings, switch the power policy to Low Power.

References

1. *VMware Distributed Power Management Concepts and Use*. VMware, Inc., 2010.
<http://www.vmware.com/files/pdf/Distributed-Power-Management-vSphere.pdf>.
2. *Advanced Configuration and Power Interface Specification*, Revision 5.0. Hewlett-Packard Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies Ltd., Toshiba Corporation, 2011.
<http://www.acpi.info/>.
3. *vSphere Resource Management Guide*. VMware, Inc., 2012.
<http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-50-resource-management-guide.pdf>.
4. "SPECpower_ssj2008." Standard Performance Evaluation Corporation, March 22, 2013.
http://www.spec.org/power_ssj2008/
5. "SPECvirt_sc2010." Standard Performance Evaluation Corporation, May 23, 2011.
http://www.spec.org/virt_sc2010/
6. Achtemichuk, Mark. "What is the difference between PCPU Used and PCPU Utilized?" *VMware vSphere Blog*, Jan. 4, 2013. <http://blogs.vmware.com/vsphere/2013/01/what-is-the-difference-between-pcpu-used-and-pcpu-utilized.html>
7. "ESXi 5.0 host experiences a purple diagnostic screen with the errors 'Failed to ack TLB invalidate' or 'no heartbeat' on HP servers with PCC support." VMware Knowledge Base. VMware, Inc., May 7, 2013.
<http://kb.vmware.com/kb/2000091>
8. *wxWidgets Cross-Platform GUI Library*. <http://www.wxwidgets.org/>

About the Author

Qasim Ali works at VMware with the Performance Engineering team. He earned his PhD from Purdue University in August 2009. His interests include Power management, CPU and memory resource management, and monitor performance.

Acknowledgments

The author thanks all members of the VMware Performance Engineering team who contributed to and reviewed this paper. He also thanks Tim Mann from the vSphere Power Management Development team for providing technical background for this paper. He would also like to thank Raghunathan Srinivasan at Intel Corporation for helping with the experimental section.

