

Power In, Dollars Out: How to Stem the Flow in the Data Center

Server Power Considerations for IT Administrators

November 17, 2008

Abstract

This document provides a comprehensive analysis of the server power landscape for information technology (IT) administrators. It explains the effect of server power usage on total cost of ownership (TCO) for IT organizations, shows the intricacies of the power-versus-performance tradeoff in the server realm, and describes in detail how the many Windows Server® system configuration parameters that IT administrators choose can affect power efficiency.

This information applies for the Windows Server 2008 operating system.

References and resources discussed here are listed at the end of this paper.

For questions or to provide feedback on this paper, contact srvpwrfb@microsoft.com

For the latest information, see:

<http://www.microsoft.com/whdc/> _____

Disclaimer: This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Document History

Date	Change
11/17/2008	Final Draft

Contents

Contents 2

Introduction 6

Background 6

Simple Power-Saving Concepts 7

 Power Efficiency Best Practices Checklist 7

 Shut Down Idle Machines 7

 Deploy Power-Efficient Hardware 8

 2.5-Inch Disk Drives 8

 Low-RPM Disk Drives 8

 Power-Efficient Processors 8

 Memory 8

 Power Supplies and Cooling Fans 8

 Remote Power Strips 8

 Use Appropriate Operating System Power Policy Configurations 9

 Tune Processor Power Management Parameters 9

 Increase Data Center Efficiency through Consolidation 9

 Turn Off Hardware-Based Power Management 9

 Use the Latest Windows Server and Service Pack Updates 9

 Minimize Unnecessary System Activity 9

Power Costs, Tradeoffs, and Analysis	10
The Increasing Cost of Data Center Power	10
Power Capacity	10
Government Oversight	10
Environmental Effect	10
The Power/Performance Tradeoff	11
Efficiency Analysis, Benchmarks, and Metrics	11
ACPI Overview	12
ACPI Processor States	13
C-States	13
P-States	13
T-States	14
System-Level Power	14
Active System Power Consumption	15
Power Consumption of Idle Systems and Components	15
Operating System Effects on Server Power	18
Windows Server 2003 vs. Windows Server 2008	18
Power Plan Selections	19
Advanced Processor Power Management Concepts	19
Performance and Idle State Transitions: Single Processor	19
Performance and Idle State Transitions: Multiple Processors	20
Configuring P-State Parameters for Increased Power Efficiency	21
Effects of PPM Policy on Power Efficiency	21
Energy Efficiency Analysis of P-State Settings	23
Driver and Application Effects on Power Efficiency	24
Maintaining Idle States	24
Interrupts	24
Timers	25
Processor Affinity	25
Measuring PPM Effectiveness	25
Preventing Problems	25
Removing Unnecessary Software	25
Turning Off or Unplugging Unnecessary Hardware	26
Using In-Box Drivers Where Possible	26
Increasing Data Center Utilization	26
Overprovisioning	26
Monitoring	26
System Consolidation	27
Multiple Roles in Windows Server 2008	27
Virtualization	27
Dynamic Provisioning	27
Virtualization and Power Management in Windows Server 2008	28
Hyper-V Architecture	28
PPM Implications	28
Performance	28
Hardware Component Effects on Power	28
Processors	28
Performance States	29
Idle States	29

Low-Voltage Processors	29
Memory	29
Memory Family.....	30
Bus Speed	30
Memory Capacity	30
Chip Density.....	30
Additional Features	30
Storage	31
Size Reduction	31
RPM Reduction	31
Solid-State Disks	31
RAID Selection	31
Network Adapters.....	32
Remote Power Strips	32
Cooling	32
Power Supply Units.....	33
Power Supply Unit Efficiency.....	33
Efficiency Programs	34
Platform Power Management and Budgeting	34
Hardware-Based Processor Power Management	34
Power Budgeting	34
Data Center Infrastructure.....	35
Resources	35
WHDC Web site white papers	35
Microsoft Tools and Web sites	35
United States Government.....	36
Organizations.....	36
Articles and Papers	37
Appendix A. Power Policy Options in Windows Server 2008.....	38
Control Panel Power Policy Options	38
Throttle State Enable.....	38
Minimum/Maximum Performance State	38
USB Selective Suspend	38
PCI Express ASPM	39
Search and Indexing Power Savings	39
Advanced PPM Performance State Options	39
Time Check	39
Increase and Decrease Time.....	39
Increase Percentage and Decrease Percentage	39
Domain Accounting Policy.....	40
Increase Policy and Decrease Policy.....	40
Advanced Idle State Options.....	40
Time Check	41
Promote and Demote Percentage.....	41
Appendix B. Changing P-State Parameters by Using Powercfg.exe	42
Appendix C. Viewing PPM Counters by Using Perfmon.exe	43
End Notes	49

Introduction

This paper presents a comprehensive analysis of the server power landscape for information technology (IT) administrators. It explains how power affects the IT budget, shows power and performance tradeoffs, and describes in detail how hardware and software can affect overall power efficiency in the data center. Administrators can use this information to make power-aware deployment and purchasing decisions, identify issues and inefficiencies on deployed systems, and maximize their organization's power efficiency.

The next few pages summarize the most important and effective approaches for saving power.

Background

This document focuses on present-day hardware components and software applications that work with the Windows Server® 2008 operating system.

The goals of this document are to:

- Provide simple, usable suggestions for increasing server power efficiency.
- Help administrators make power-conscious purchasing and usage planning decisions that can reduce power costs and reduce wasted capacity.
- Provide clear, detailed explanations of the configuration parameters on Windows Server 2008 that affect server power consumption.
- Enable administrators to identify inefficient hardware and software components and correct inefficiencies.

Calculations and estimates in the paper are based on the following assumptions:

- According to the United States Energy Information Administration, the cost of electricity in the United States is 10.88 cents per kilowatt hour (kWh), based on the average United States commercial rate as of June 2008. Readers should adjust calculations to match local electricity costs.
- A 24-hour-a-day, 7-day-a-week (24x7) server runs for 8,760 hours per year.

In addition, this paper uses the term “server” to refer only to in-box parts. External disk arrays, monitors, input devices, uninterruptible power supplies (UPSs), and other items are not included in power calculations.

Understanding the workload of a server is important to achieving any power-efficiency savings. Selecting hardware or configuration parameters without understanding the workload of the machine can lead to poor performance or power efficiency.

Gauge the utilization levels of server subsystems such as disk, network, CPU, and memory on an existing system. You can then reduce excess capacity or provision future capacity only in the subsystems where it is needed and opt for low-power parts if that is feasible.

For example, computationally intensive workloads usually do not need fast disks, large redundant array of independent disks (RAID) arrays, or quad-port 10-gigabit per-second (GBps) network adapters, but generally require as many processors as possible.

Simple Power-Saving Concepts

Some general concepts can help reduce the power footprint in most scenarios. This section describes those ideas to give administrators some good first steps for reducing power consumption before they investigate more detailed methods.

Power Efficiency Best Practices Checklist

These statements are best practices that will be described in more detail in the sections to follow.

- Shut down idle machines during off-peak times.
- Opt for 2.5-inch instead of 3.5-inch disk drives.
- Use low-revolutions-per-minute (RPM) disk drives where it is possible.
- Select power-efficient processors and memory.
- Install variable-speed fans and efficient power supplies in servers to reduce waste.
- Use remote-controlled power strips to completely eliminate electricity flow to “powered-off” servers.
- Ensure that Windows Server 2008 is configured to use the Balanced power policy.
- Tune processor power management parameters to increase efficiency by up to 10 percent.
- Consolidate workloads or combine server roles on idle and underutilized servers where it is possible.
- Turn off hardware-based power management¹.
- Deploy the latest service packs and Windows Server releases.
- Remove or shut down unnecessary roles, applications, and devices.

Shut Down Idle Machines

Workloads vary over time. Some workloads run only at specific times of the day, whereas other workloads are dynamic and user driven. By identifying consistent, long periods of nonuse, administrators can shut down servers when they are not being used. For example, backup, test, and build servers are typically idle for long periods during the day. According to “Sustainable Computing: Is It Time to Turn Off Your Servers?” on Microsoft TechNet, servers typically consume more than 50 percent of their peak instant power at idle. Turning off servers when they are not needed can save a lot of electricity.

¹ Currently, Microsoft test labs have shown hardware-based power management schemes to be less efficient than operating system-based power management. For more information, see “Hardware-Based Processor Power Management” later in this paper.

Deploy Power-Efficient Hardware

Choosing power-efficient hardware when you deploy new servers or upgrade existing servers is a simple, cost-effective way to increase power efficiency. These components can cost more up front, but you can view the operational cost savings as a return on investment (ROI). And other than the initial deployment or installation, low-power hardware incurs no additional management overhead.

2.5-Inch Disk Drives

Microsoft test data shows that power consumption for 2.5-inch disk drives is about half that of 3.5-inch disk drives that have comparable capacity and speed.

Low-RPM Disk Drives

For storage installations that have no strict latency requirements, 15,000-RPM enterprise-class disk drives may be unnecessary. If 10,000- or even 7,200-RPM disk drives can adequately satisfy performance goals, use them instead of high-RPM disk drives to significantly reduce power consumption.

Power-Efficient Processors

Power management features were built into processors for several years, and processor manufacturers are working on additional power management features for future product lines. Some processor families incorporate low-power states, whereas other processor families are designed as low-voltage parts. Either type can save significant wattage.

Memory

Memory module power consumption varies widely from one module to another. Bus speed plays a large factor in memory power consumption, but so do the density, rank, and operating voltage. On a system that has many sticks of RAM, the memory power footprint becomes a large percentage of system power and a prime target for savings.

Power Supplies and Cooling Fans

Power supplies and cooling fans are important areas to target when you want to reduce power waste in the data center. Investing in high-efficiency power supplies and variable-speed fans can reduce unnecessary power consumption by a significant percentage, which saves money in the long term and frees important capacity for other uses.

Remote Power Strips

Currently, systems that are shut down can still consume tens of watts of power. Unplugging systems is the only way to eliminate this power waste, but physically doing so might not be possible for all organizations. Remote power-control strips let administrators automate this process and can save large quantities of otherwise wasted electricity.

Use Appropriate Operating System Power Policy Configurations

A simple change that can generate power savings is to ensure that the power policy on deployed systems enables operating system power management technologies. The Balanced power policy, which is enabled by default in Windows Server 2008, is most appropriate to deliver power efficiency across the widest range of server applications. For more detail, see the section “Power Plan Selections” in this paper.

Tune Processor Power Management Parameters

Power parameter defaults in the Balanced policy are “safe” defaults that reduce the potential for power savings algorithms to negatively affect performance. We have identified a set of parameters that can increase power efficiency by up to 10 percent on some workloads. For more information, see “Configuring P-State Parameters for Increased Power Efficiency” later in this paper.

Increase Data Center Efficiency through Consolidation

Currently, the most power-efficient servers run at full utilization. Yet according to the United States Environmental Protection Agency (EPA) “Report to Congress on Server and Data Center Energy Efficiency,” production servers run at anywhere from 5- to 15-percent utilization on average. IT administrators are justifiably worried about affecting Quality-of-Service levels by increasing utilization, but in appropriate situations, machine consolidation can significantly reduce power footprint and increase efficiency.

Turn Off Hardware-Based Power Management

Some server hardware vendors implement firmware-based processor power management (PPM) features that dynamically change the processor state using a hardware interface. We believe that ideal PPM is intimately tied to the operating system’s knowledge of historical and current workload needs. In our laboratory we have seen 2- to 10-percent efficiency improvements by using operating system mechanisms instead of hardware-based implementations. We recommend that you disable any firmware power management utilities in the BIOS and let Windows Server 2008 handle PPM.

Use the Latest Windows Server and Service Pack Updates

New power management features and settings are frequently delivered in new operating system editions and service packs. Updating can provide a more power-efficient experience. The Microsoft white paper “Windows Server 2008 Power Savings” states that Windows Server 2008 is 10-percent more power efficient at equivalent utilization levels than Windows Server 2003. Keeping current on operating system releases and service packs is the best way to ensure that your Windows® servers are achieving maximum efficiency.

Minimize Unnecessary System Activity

Servers at low utilizations can reduce power consumption by entering low-power states. Unnecessary or poorly written applications and device drivers can interrupt these states and keep a system from achieving the lowest possible power

consumption. To ensure that a system maximizes usage of idle states, you should remove all nonessential roles, applications, and devices.

Power Costs, Tradeoffs, and Analysis

Before you try to optimize server power consumption, we recommend that you understand the cost motivations, tradeoffs, and analysis metrics that are involved in the process.

The Increasing Cost of Data Center Power

The increase in power consumption for IT equipment over the last decadeⁱ and the commoditization of server-class computer hardware have led to accelerated change in the dynamics of IT administration. *Electronics Cooling* magazine reports that annual utility costs just to turn on and keep a server cool are approaching the up-front cost of a server. Consider a server farm of one thousand servers whose idle power consumption is 100 watts (W). Running 24x7, this server farm costs \$95,308.80 per year *just to keep powered on*.

Server power consumption is also a primary target for cost reduction in the data center because of the multiplicative effect of cooling and infrastructure costs. A recent United States EPA report, "Report to Congress on Server and Data Center Energy Efficiency," suggests that equipment power represents only half the total electricity bill in the data center, with the other half going to support and infrastructure equipment such as air conditioning, fans, network switches, and UPSs. Each watt that is used to power a server can require an additional watt for support and cooling equipment.

Power Capacity

According to the same EPA report, data center energy capacity is more often a motivation for reducing power footprint than electricity cost. Capacity expansion requires new power and cooling equipment or, in the worst case, new site construction. These costs are significantly more prohibitive than the cost of energy.

Government Oversight

Governments around the world are becoming increasingly sensitive to energy issues. The United States EPA is drafting new ENERGY STAR mandates for enterprise server power efficiency. Carbon tax programs have been instituted in countries around the world, including Sweden, the Netherlands, and New Zealand. Failure to take action now may mean high costs for your business in the future. Power-aware decision-making helps smooth your business's compliance with government mandates and may qualify you for future tax incentives.

Environmental Effect

Concern for the environment is another reason to reduce the power footprint of IT installations. A smaller power footprint increases the environmental sustainability of an organization.

The Power/Performance Tradeoff

Design tradeoffs are common in any technical endeavor.

Administrators often must consider a power/performance tradeoff when they make purchasing decisions. Consider a new machine configuration where a cheaper processor model is available at 1.8 gigahertz (GHz) and a more expensive version is available at 2.1 GHz. The performance increase might cost hundreds of additional dollars. Power efficiency involves a similar cost tradeoff—low-power parts can cost more up front, but in this scenario, the price increase is offset by savings over the life of the product.

To complicate things further, low-power parts may decrease performance in some manner. “Green” memory might have lower bus speeds and throughput, whereas low-power disks might have reduced capacity or increased latency. These performance penalties can reduce the useful lifespan and versatility of a system or component.

Rarely is absolute power savings possible without complex trade-offs.

Efficiency Analysis, Benchmarks, and Metrics

Defining a standard of measurement for server systems is important. Many benchmark measurements seek to quantify only the maximum amount of work that a particular configuration can perform. In terms of power consumption, this information is not as valuable. “Power efficiency” is a much more useful metric. Power efficiency is the ratio of work that is done over time to the power that is consumed over time.

In equation form, this is given as the following:

$$\text{Power Efficiency} = \frac{\frac{\text{Units of Work Done}}{\text{Time Unit}}}{\frac{\text{Units of Power Consumed}}{\text{Time Unit}}}$$

Over a specific unit of time, a server system can do a specific amount of computational work. This work is known as its *throughput*. Over the same unit of time, the server consumes a particular quantity of power. The time units are the same (generally seconds), so units can be eliminated from the equation. Because power measurements are generally measured in watts, the equation reduces to the following:

$$\text{Power Efficiency} = \frac{\text{Units of Work Done}}{\text{Watts Of Power Consumed}}$$

Two general mechanisms improve power efficiency:

- Delivering more performance at the same power level.
- Delivering equivalent performance at a lower power level.

The analysis in this paper focuses largely on the second mechanism.

Generating a “load line” for a system configuration is useful. You can do this by measuring the power consumption of a system while throughput varies across the utilization range of a system, from idle up to 100-percent utilization. Figure 1 is an example of the load line concept.

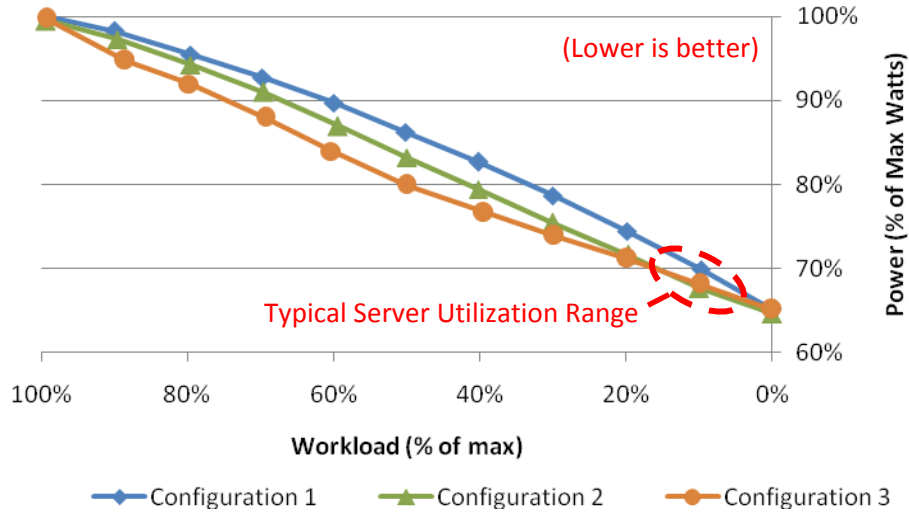


Figure 1. Load line example

Figure 1 is a graph of the power consumption of three system configurations across their load lines. Configuration 3 is the most power-efficient configuration at all points except at 10-percent utilization, where configuration 2 is more power efficient. Configuration 1 has the highest power consumption across the load line, which makes it the least power efficient. All configurations can achieve the same level of peak throughput. By using the load line approach, we can evaluate and compare the performance and energy consumption of different configurations.

The load line is a good measurement approach because measuring power only when hardware is fully utilized does not reflect real-world usage. Production servers on average run at much lower utilizations, typically in the range of 5- to 15-percent for nonvirtualized servers.

You can use several benchmarks to collect load line data. SPECpower from the Standard Performance Evaluation Corporation (SPEC) is one such benchmark. Internal file and Web server workloads can be used. Also, you can modify several traditional server-class benchmarks to vary their workload. We have had success with this approach by using the TPC-E benchmark from Transaction Processing Performance Council (TPC).

ACPI Overview

The Advanced Configuration and Power Interface (ACPI) benefits both hardware and operating system designers by clearly defining an interface between the operating system software and hardware to be used for power management purposes. Without going into details about the data structures or communication mechanisms involved,

this section describes some power management states that exist in practice because of ACPI. For more information, see “Resources” at the end of this paper.

ACPI Processor States

The processor has traditionally consumed the most power in a server, which makes it a great candidate for power-efficiency optimizations. To add detail and flexibility for processor power management, ACPI defines three sets of states for processors.

You can find comprehensive Microsoft information about PPM in “Processor Power Management in Windows Vista and Windows Server 2008,” on the WHDC Web site.

C-States

C-states define incremental levels of processor idle, from C0 (active) to Cn (lowest power idle). ACPI specification 2.0 and later versions do not specify a maximum number of C-states, so the terminology Cn is used to refer to the highest numbered, lowest-power idle state that a processor supports.

C0 is the state at which a processor executes instructions, whereas C1 and greater are nonoperational idle states. C1, C2,...,Cn are by definition sequentially lower power (and higher latency) idle states. For example, a processor at C1 idle requires a short time to return to C0. A processor at C2 idle requires more time than C1, but will draw less power. A transition from idle state C1 to a deeper idle state C2 is often called *promotion*, while a transition from a deep idle state like C3 to state C2 is often called *demotion*.

Implementation

C-state implementations generally involve shutting down successively larger areas of the processor floor plan as deeper idle states are entered. A simplified theoretical model might be as follows: the first idle state might involve turning off power to the execution units. No work must be done, so this silicon is already unused, and turning off these components eliminates leakage current. The execution units can be restored to operation very quickly, because the processor’s execution context has not changed.

The next idle state might involve a shutdown of the processor’s first-level cache. By shutting down these caches, significant power can be saved. Caches also have a hierarchical construction, which lets the processor shut down increasingly larger on-die caches at each successive idle state and possibly even the entire socket.

P-States

Processors in operation (state C0) can transition between multiple performance states, or P-states. P-states define incremental levels of processor performance, from P0 (most performant) to Pn (least performant). The ACPI specification does not specify a maximum number of P-states, so Pn is used to refer to the highest numbered, lowest performant P-state that a processor supports.

Each successively higher numbered P-state consumes less power than the previous P-state. Processors can dynamically switch between these states during operation to provide only as much computational capacity as is necessary, which saves power during periods of low usage.

Figure 2 shows a hypothetical set of six P-states that would be available to a processor. Note that the maximum P-state (P0) has the highest frequency, while successively higher numbered P-states reduce in frequency. In this case, the minimum P-state is P5, so the terms Pn and P5 would be interchangeable.

State	Freq (MHz)	Percent	Type
0	2800	100	Performance
1	2520	90	Performance
2	2380	85	Performance
3	2100	75	Performance
4	1680	60	Performance
5	1400	50	Performance

← Maximum Processor State

← Minimum Processor State

↑ DBS ↓

Figure 2. Illustration of P-state number and corresponding frequency

Implementation

Because of transistor physics, you can reduce power consumption on a processor by reducing the frequency or by reducing the operating voltage. Therefore, P-states are implemented as reductions of processor frequency, voltage, or both. ⁱⁱ

T-States

You can use a set of throttle states—or T-states—to reduce the power consumption and processor performance by scaling back the quantity of clock cycles by a percentage value. Assuming one operation per clock cycle, a 2.5-GHz processor at a 50-percent T-state would perform 1.25 billion operations instead of 2.5 billion operations.

Implementation

When a processor enters a T-state, the processor's clock signal is masked for a specific time period. This eliminates the power that is necessary for clock distribution throughout the processor. It also prevents the processor from doing computational work or changing state, which eliminates the processor transistors' dynamic power consumption.

System-Level Power

Servers can differ in size, throughput, and component makeup, but generally they tend to exhibit similar power consumption trends. This section describes those trends and gives a high-level perspective on the power consumption of server-class machines.

Active System Power Consumption

All IT administrators should understand how power consumption varies as machine utilization changes. Figure 3 was generated by measuring system power consumption as a scalable workload (the TPC-E benchmark workload) was ramped up.

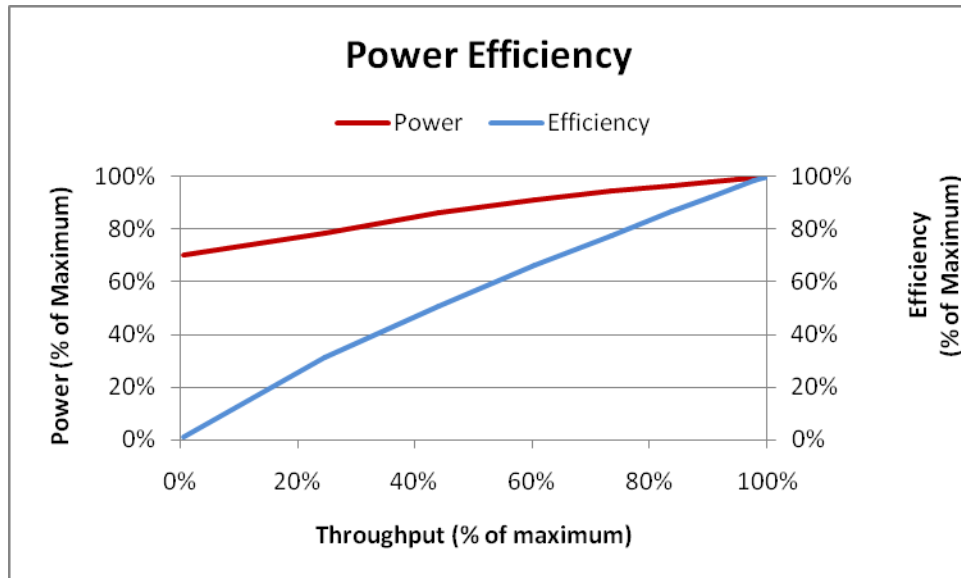


Figure 3. Power and power efficiency versus throughput (in transactions per second)

The top line on the graph represents power consumption as a percentage of maximum power. As shown here, idle power consumption is approximately 65 percent of maximum, even though no work was being done. This is not anomalous behavior. It is common for server idle power to be approximately two-thirds of fully utilized system power consumption.

The bottom line tracks efficiency relative to the maximum, given in transactions per watt. The important takeaway is that an idle server is 0-percent efficient, and the most efficient server (currently) is one at full utilization.

Both findings are strong rationale for increasing overall utilization in the data center, especially through machine consolidation. Combining two boxes into one eliminates the over-60-percent idle power overhead that is necessary to keep one of the boxes running and moves the remaining server further along its power efficiency curve.

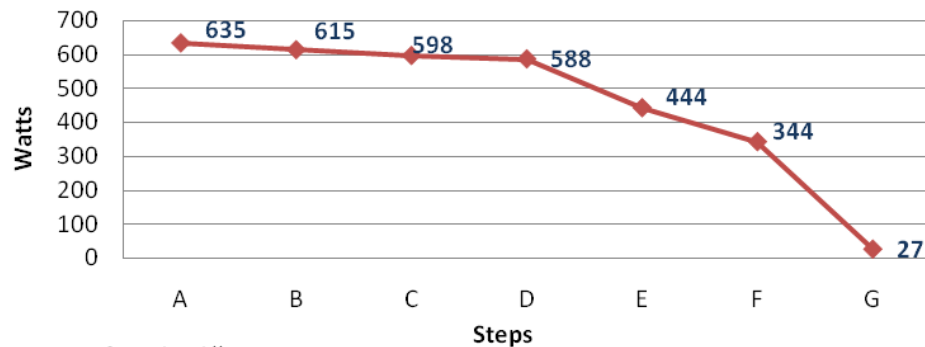
Power Consumption of Idle Systems and Components

In a Microsoft test lab, we connected a dual-socket, dual-core server that was shipped in 2005 (Server A) and a quad-socket, quad-core server that was shipped in 2008 (Server B) to power meters at the wall. We measured power consumption for the servers at idle and as components were removed. Components installed in each system are shown in Table 1.

Table 1. Components in 2005 and 2008 Test Systems

Component	Server A (shipped in 2005)	Server B (shipped in 2008)
Processors	2 dual-core processors	4 quad-core processors
Memory	32 2-GB Double Data Rate (DDR) PC-2100 dual inline memory modules (DIMMs)	32 4-GB DDR-2 PC-5300 DIMMs
Network	1 PCI-X network adapter (single-port 1-GB)	2 PCI-E network adapters (quad-port 1 GB)
Disk Controllers	4 PCI-X host bus adapters—HBAs (dual-port)	2 PCI-E HBA's (2x dual port)
Disks	4 36-GB, 15,000-RPM 3.5-inch SCSI	4 72GB, 15k RPM 2.5-inch Serial Attached SCSI (SAS)

In some cases, it was necessary to extrapolate power consumption for components that were required to keep the system running. For example, the system must have one processor and one memory stick. The power measurements for Server B are shown in Figure 4.



- Step A: All components.
 Step B: Removed 3x72-GB 15,000 2.5-inch SAS disks.
 Step C: Removed 2x quad-port 1-GBps NICs.
 Step D: Removed 2x dual-port HBA cards.
 Step E: Removed 16x4-GB memory DIMMs (on daughterboards).
 Step F: Removed 8x4-GB memory DIMMs (on motherboard).
 Step G: Shut down the server.

Figure 4. Server B system power during device removal experiment

The following are important points of note:

- The processors on this machine could not be removed because of the fragile connection points. Instead, we obtained the idle power data from the manufacturer's technical specification sheet for the product.
- The memory power that the sticks consumed on the daughterboards (9 W) differs from that consumed on the motherboard (12.5 watts). We do not have an explanation for this behavior, so we assume that it is related to the platform design.
- The system consumed 27 watts of power when the server was fully powered off. This may seem excessive, but this is not anomalous behavior. We found the other system consumed 30 watts when powered off. Unplugging systems is currently the only way to eliminate this power waste; administrators can automate this process if remote controlled power strips are deployed.

The remaining power that is consumed when all possible components were removed or otherwise accounted for was added to an “others” section. Consumers of power in this category include power supply, motherboard and chipsets, fans, and other miscellaneous items.

With all components in place, idle power consumption was measured at 568 watts on the 2005 system and 635 watts on the 2008 system. Figure 5 and Figure 6 detail the power consumption of individual components on these systems relative to overall system consumption.

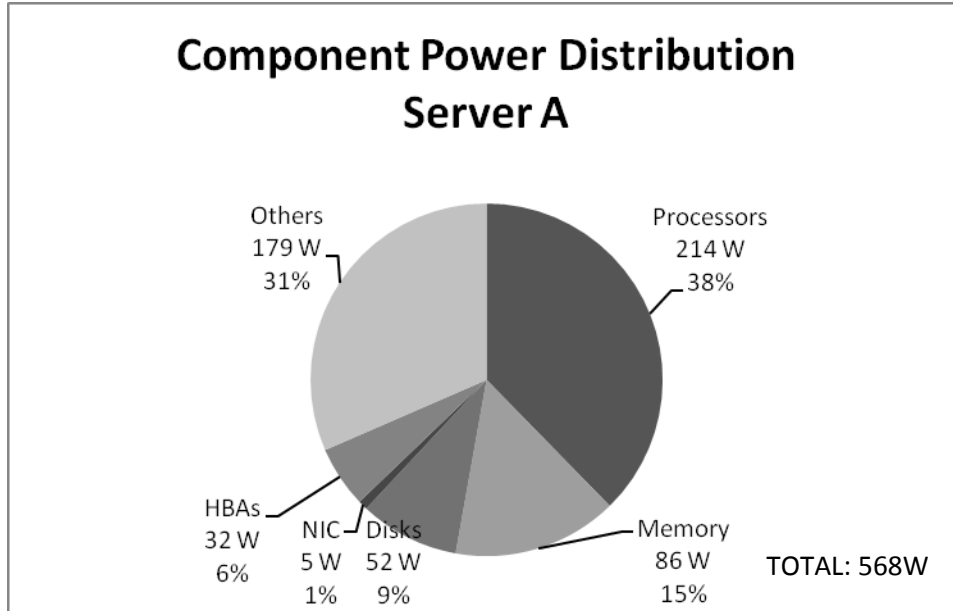


Figure 5. Component power distribution, 2005 two-socket dual-core server

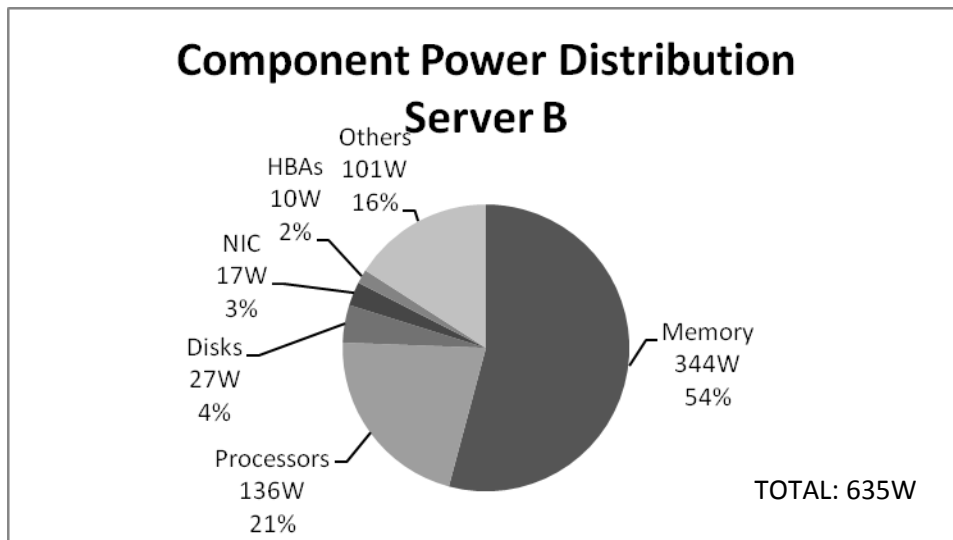


Figure 6. Component power distribution, 2008 four-socket quad-core server

It is immediately obvious that processors are not necessarily the largest consumers of power in the system. In a modern server, 32 sticks of memory can easily consume as much or more power than other system components at idle.

The cause of the increase in memory power usage is explained in more detail in "Hardware Component Effects on Power" later in this paper, but generally a doubling in bus speed or capacity can double the power consumption. Because both bus speed and capacity increased between systems, the fourfold increase in power consumption is understandable.

Unfortunately, the processors in the 2008 machine could not be removed. From manufacturer specifications, we estimated the idle power consumption to be around 136 watts. In this scenario, even though the newer system has more cores, the advanced PPM in the newer processor family saves 78 watts at idle.

The 48-percent decrease in disk power consumption is because of the switch from 3.5-inch to 2.5-inch disk drives. For the small quantity of disks in this system, the absolute savings is only 5 percent of overall system power, but for large disk arrays the savings can accumulate quickly.

The network adapter and HBA power variation is due more to quantity and feature set than to any power optimizations.

The large quantities of power that are attributed to the "others" category demonstrates the significant effect that frequently ignored items like power supplies and fans have on power consumption.

Operating System Effects on Server Power

Power efficiency cannot be achieved by hardware alone. The operating system plays an important role in increasing a system's power efficiency. This section details how the Windows Server 2008 operating system contributes to server power efficiency.

Windows Server 2003 vs. Windows Server 2008

According to "Windows Server 2008 Power Savings," Windows Server 2008 delivered 20-percent more throughput than Windows Server 2003 and achieved power savings of up to 10 percent at comparable throughput levels. Such savings are possible because of new power management features that were introduced in Windows Server 2008, most of which are specifically related to ACPI PPM support.

These additional features include the following:

- Support for all ACPI 3.0 objects.
- Multiprocessor PPM support.
- Support for throttling on processors that do not have P-states.
- Better in-box driver support.
- Various algorithmic improvements.

Some of these improvements are discussed in this section. For more information, see "Processor Power Management in Windows Vista and Windows Server 2008" in "Resources."

Power Plan Selections

Windows Server 2008 has three power plans that can be accessed through the Control Panel **Power Options** applet:

- **High Performance**

This power policy has few power saving features enabled. Parameters heavily favor performance over power, so processors and devices constantly consume maximum power. This might be appropriate for a machine where minimizing latency is critically important.

- **Balanced (Default)**

This power policy enables most power management features. Under this policy, Windows determines an optimal state for the processors and devices that delivers necessary throughput while maximizing power savings. The Balanced policy has several configurable options from Control Panel and even more are available through in-box tools such as powercfg.exe. These parameters are covered in detail in "Appendix A: Power Policy Options in Windows Server 2008."

- **Power Saver**

This power policy restricts processors to their lowest available power state (Pn) and takes the most advantage of low power device states. Pn minimizes the power consumption of the processors when they are doing useful work. However, this option also limits the throughput of a system. Workloads might experience Quality-of-Service degradations in the Power Saver plan.

If saving power is at all a concern, you should choose between the Balanced and Power Saver mode. Balanced mode takes the best advantage of operating system power management by striking the most optimal balance between power and performance.

Advanced Processor Power Management Concepts

Processor power management algorithms in Windows Server 2008 are complex. However, PPM is an important part of server power management. Understanding the concepts that are involved can help you understand how tuning PPM parameters can increase efficiency and how poorly designed hardware, drivers, or application configurations can reduce the effectiveness of power management.

Performance and Idle State Transitions: Single Processor

Workloads on servers are transient. Workload patterns change throughout the day and are subject to instantaneous spikes. Windows Server 2008 uses ACPI performance and idle states to match available computational resources to the current system demand.

Under the Balanced power policy that was described earlier, the machine reevaluates processor performance states at 100-millisecond (ms) intervals, which is called a time check. At each time check, the operating system determines the appropriate P-state or C-state for the processor for the next time interval.ⁱⁱⁱ

Performance and Idle State Transitions: Multiple Processors

In today’s market, single-core, single-processor servers are rare. PPM becomes much more complex on multicore processors and multisocket servers.^{iv} To demonstrate how P-states and C-states change as system utilization changes, we tracked the cumulative time that a multiprocessor system’s cores spent in various P-states and the C1 idle state on a machine in which a workload was incrementally decreased every 10 minutes. We used this information to create Figure 7.

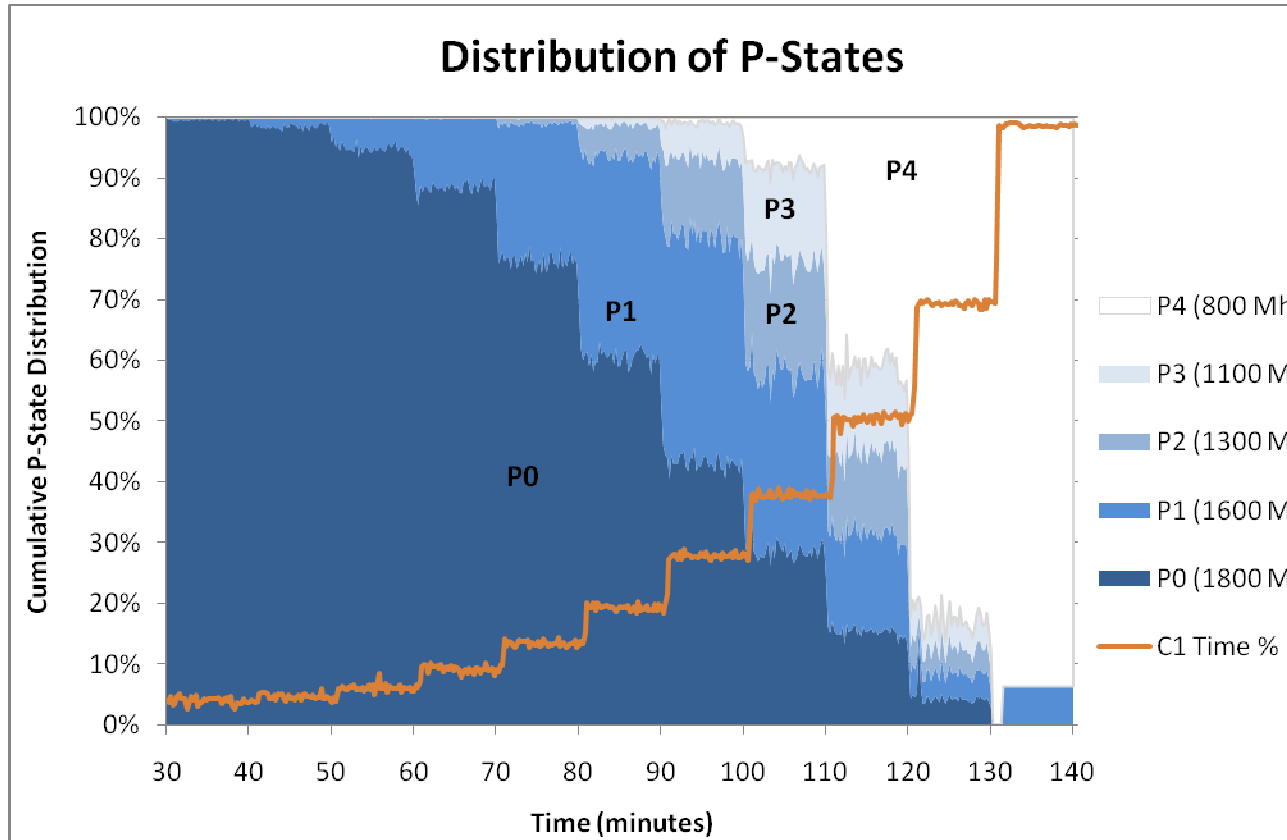


Figure 7. Distribution of P- and C1-state as workload decreases over time

As Figure 6 shows, minutes 30 through 40 are typical of full machine utilization. All processors spend almost all their time in performance state P0, and C1 residency is at a minimum. As system utilization decreases, processors are distributed into higher numbered P-states and idle state C1. Minutes 130 through 140 represent the idle scenario and, as expected, we see almost all the time is spent in C1 and P4.

Note that from minutes 60 on, processors enter the C1 state without necessarily entering higher P-states. This is because performance and idle states do not have a dependent relationship. A processor is not required to go through Pn to enter an idle state, and P-states are considered irrelevant to a processor in an idle state.

Configuring P-State Parameters for Increased Power Efficiency

Administrators who want to adjust the power/performance balance of power management but do not want to go to the extremes of High Performance and Power Saver plans can reconfigure the Balanced power plan's parameter defaults. Empirically measuring the effects of these changes currently requires benchmarks and external hardware, which may not be available to all administrators. For that reason, we have determined a set of parameters that you can use to further increase energy efficiency with minor performance cost.

Effects of PPM Policy on Power Efficiency

Windows Server includes several PPM configuration parameters that you can use to adjust the behavior of the PPM engine. "Appendix A. Power Policy Options in Windows Server 2008" describes these policies in detail.

The defaults for these parameters are tuned to deliver excellent power efficiency for most systems and workloads out of the box. However, these are "safe" defaults. They balance performance and power efficiency. Default settings are shown in Table 2. Note that "P-state increase" in this context refers to a transition to a lower numbered, more performant P-state, whereas "P-state decrease" refers to a transition to a higher numbered, less performant P-state. Looking back to Figure 2, an increase would mean moving upward in the chart whereas a decrease would mean moving downward.

Table 2. Default P-State Parameter Settings in Windows Server 2008

Name	Default	Description
Time Check	100 ms	The time interval at which the operating system considers a change of the current P-state.
Increase Time	100 ms	The minimum time period that must expire before considering a P-state increase.
Decrease Time	300 ms	The minimum time period that must expire before considering a P-state decrease.
Increase Percentage	30%	The utilization percentage ¹ that the CPU must exceed to increase P-state.
Decrease Percentage	50%	The utilization percentage that the CPU must be below to decrease P-state
Domain Accounting Policy	0 (On)	Determines how the kernel power manager accumulates idle time. Settings: 0 (On): idle time is accumulated only when all processors in a C- state domain ² are idle. 1 (Off): idle time is accumulated and P-states are calculated for each processor without regard to any other processor in the domain.

Name	Default	Description
Increase Policy	IDEAL (0)	Determines how P-state transition decisions are made. Settings: IDEAL (0): calculates the target P-state based only on processor utilization and then finds a nearby available P-state on the system. SINGLE (1): calculates an ideal P-state but only increases or decreases by one P-state per time check interval. ROCKET (2): transitions to the highest P-state available on increase or lowest P-state available on decrease
Decrease Policy	SINGLE (1)	

¹The utilization percentage referenced here is not the same as the CPU usage counter in the Task Manager tool. Without going into more details, this setting is best optimized through empirical experimentation.

²A “state domain” is a dependency between different processor cores or packages on a server. Often, processor designs require that if one core is at a particular P- or C-state, the other cores or packages in the domain must also be at the same P- or C-state. The hardware notifies the operating system of this dependency by establishing a domain through the ACPI interface.

Important: Modifying any of these parameters changes the behavior of performance state handling from the out-of-box experience. Before you deploy to production servers, validate the effects of any changes in a test environment.

Notice in Table 2 that the decrease time default is larger than the increase time default. This setting favors P-state increases over decreases. The default increase and decrease percentage settings of 30 and 50 percent, respectively, lean to P-state increases as well. The default domain accounting policy requires all processors to be idle before they accumulate idle time, which trends processors in each package to lower numbered P-states. The increase and decrease policy defaults favor lower-numbered P-states as well.

To tune the machine for more aggressive power savings, we suggest reducing the decrease time to 100 ms to match the increase time, changing the increase and decrease policies to favor P-state decrease, and switching the domain accounting policy to 0 (off). We left the increase and decrease percentages as their defaults to ensure that the system PPM parameters were not completely biased toward power savings and to reduce negative performance consequences.

Table 3. Default and New PPM Parameter Values

Setting	Default value	New value
Time Check	100 ms	100 ms
Increase Time	100 ms	100 ms
Decrease Time	300 ms	100 ms
Increase Percentage	30 %	30 %
Decrease Percentage	50 %	50 %
Domain Accounting Policy	0 (On)	1 (Off)
Increase Policy	0 (Ideal)	1 (Single)
Decrease Policy	1 (Single)	0 (Ideal)

These parameters can be set only by using the powercfg.exe command-line tool, which is installed by default to the Windows\System32 folder on Windows Server 2008. The commands to change the P-state settings by using powercfg.exe are given in Appendix B.

Energy Efficiency Analysis of P-State Settings

To test the efficiency of these new power settings (henceforth called “Aggressive” settings), we performed a set of experiments on a four-socket quad-core server. Table 4 gives the system configuration.

Table 4. Four-Socket Quad-Core Server Configuration

System configuration	
Processors	4 quad-core 2.9-GHz
Memory	32 4-GB FB-DDR2 667-MHz DIMMs
Disk	4 72-GB, 15,000 SCSI
Network adapter	2 1-Gbps

We ran the SPECpower benchmark with both the default settings and the Aggressive power saving settings. Figure 8 and Figure 9 show the power usage and power efficiency across different SPECpower workload levels. The Aggressive settings exhibit better power efficiency than the default settings at most load levels. The maximum power saving is achieved at 60-percent workload level on this configuration with approximately 10-percent improvement in power efficiency when it is compared to the default setting. There is a negligible reduction in overall throughput at utilization levels above 97 percent.

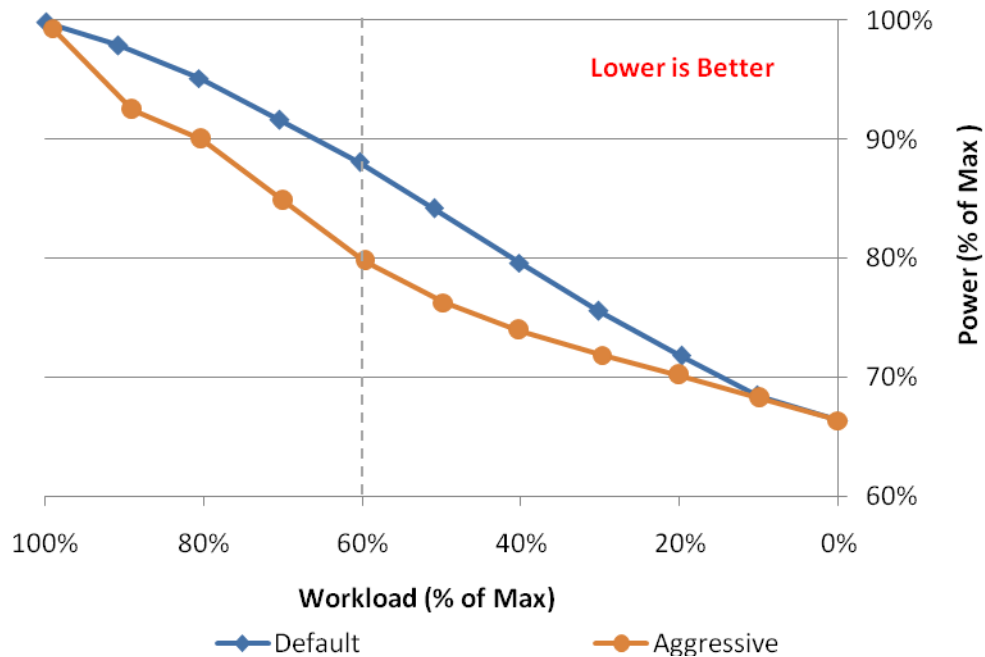


Figure 8. System power across varying SPECpower load levels

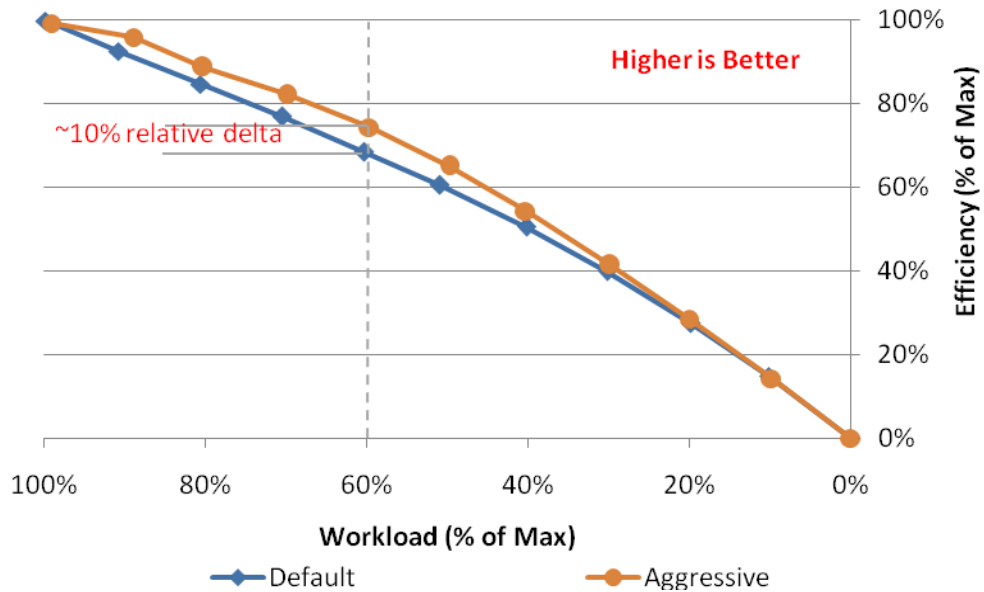


Figure 9. System power efficiency across varying SPECPower load levels

Note that these settings were tested on commodity servers that have the SPECPower workload. Other machine and workload types may see reduced benefit. Before deploying in your production environment, be sure to test any parameter changes.

Driver and Application Effects on Power Efficiency

Achieving excellent power efficiency is a delicate task that requires cooperation from all software elements, not just the operating system. Third-party drivers and applications on a system can affect power efficiency, as this section describes.

Maintaining Idle States

Dropping a processor into an idle state has associated latency and performance costs.^v To achieve real power savings, processors must enter the lowest possible idle states and remain there for long periods of time.^{vi}

Therefore, a key strategy for power efficiency is to maintain these low-power states. Unfortunately, accomplishing this is difficult. Any piece of hardware or software, including drivers and user-mode applications, can break idle state residency and ruin the effectiveness of power management.

Interrupts

Frequent interrupts are detrimental to power management. Processors must service hardware interrupts quickly regardless of the state they are in, so interrupts pull a processor out of an idle state.

Most devices use interrupts to perform I/O operations.^{vii} Every time that a computer receives a network packet or a keystroke, the processor receives an interrupt. Although this is usually desirable, unnecessary interrupts from third-party device drivers are identified as a common disruptive system activity for PPM.

Timers

At a predetermined interval on any computer system, a hardware interrupt—which is colloquially called a *timer tick*—is generated to provide clock functionality for the system. Because this is a hardware interrupt, the timer is also a guaranteed “wake up” for any idle processors that receive the timer tick. The default for a Windows machine is one timer tick every 15.6 ms.

Applications can adjust the tick rate by using a Windows API. User-mode applications can reduce the tick rate to 1 ms. Short timer intervals adversely affect power management by reducing the chances that processors remain in low-power states long enough to realize nontrivial power savings.

Processor Affinity

To increase performance, *interrupt affinity*—or a tendency to interrupt a specific processor—can be set by using the IntPolicy tool. For more information about this tool, see “Interrupt-Affinity Policy Tool” on the WHDC Web site. However, the PPM engine cannot see interrupt affinity and might target affinitized processors for entry into idle states. Constant interrupts to these processors reduces the chances of idle state power savings.

For software applications, you can configure *thread affinity*—or a preference for code to be run on a particular processor. Setting thread affinities to a particular processor can result in better cache hit performance and therefore better application throughput, but may also have harmful effects on power. For example, if you use affinities to spread out work among all processors on a system, throughput might increase but less time will be spent in idle states than a scenario in which all work is affinitized to one socket or core.

You should consider your goals for power and performance and configure thread affinity appropriately.

Measuring PPM Effectiveness

A simple approach to diagnosing bad driver and application behavior is to examine the processor idle and C1 residency time and the interrupt-per-second count when the system is at or near idle. You can do this by using the Windows Performance Monitor tool (Perfmon.exe), which is included in Windows. A walkthrough of this tool is given in Appendix C.

Preventing Problems

Identifying and determining the cause of some power management problems can require some advanced analysis, which administrators likely lack the time to perform. Instead, this section gives some simple approaches that can still be effective at finding and preventing problems.

Removing Unnecessary Software

A pragmatic administrator can reduce inefficient behavior by removing all unnecessary server roles and applications from a system. Less unnecessary code executing on a system means more power efficiency.

Turning Off or Unplugging Unnecessary Hardware

Removing devices from the system and turning off unnecessary chipset features in the BIOS means less drivers are loaded on the system. Repeating from the previous section, less code executing on a system means less potential for bad behavior to occur.

Using In-Box Drivers Where Possible

In-box drivers are part of Windows Server 2008 and are tested to ensure compatibility with power management schemes. However, Microsoft does not necessarily verify third-party drivers, which can lack power management features or generate excessive interrupt traffic.

Poorly written USB device drivers are a frequent offender in this category. Some third-party USB peripheral and controller drivers can generate enough interrupt activity that processors cannot enter states C2 or lower, even when the system is otherwise completely idle.

Using in-box drivers where possible reduces the risk that a driver will compromise system power efficiency through lack of support or bad behavior.

Increasing Data Center Utilization

Idle server power overhead is expensive. It requires significant power but gives your organization zero computational output in exchange. Reducing machine count and increasing deployed system utilization levels better amortizes this cost and increases overall data center efficiency.

Overprovisioning

Data centers are generally overprovisioned and underutilized. The most convincing explanation is the tendency for IT administrators to be risk averse. These IT professionals might provision as much hardware as is necessary to meet service level agreements continuously during the day, even if this means most of the boxes are at low utilization or idle most of the time.

Another cause of overprovisioning is lack of monitoring. Almost one-third of the IT administrators surveyed for one Forrester report did not know their current systems' utilizations.

Monitoring

Monitoring system utilization is easy to do and can help you identify candidate areas for power savings. Even though it does not paint a comprehensive picture of overall system utilization, a good starting point is to measure CPU and disk load through the Performance Monitor in Windows Server 2008. An example of CPU monitoring is given in "Appendix C: Viewing PPM Counters by Using Perfmon.exe."

System Consolidation

If servers are poorly utilized, you can sometimes consolidate deployed systems onto fewer physical machines to eliminate the significant overhead of idle server power. This approach frees capacity, increases overall efficiency, and saves your organization money.

Consolidating deployed systems onto fewer physical machines is possible in several ways, from least to most complex:

- Multirole servers
- Virtualization
- Dynamic provisioning

Multiple Roles in Windows Server 2008

The simplest way to consolidate is to install multiple roles in Windows Server 2008 or add server application software to an existing machine.

The key to successfully using this strategy is to combine roles that have complementary workload resource requirements. For example, a payroll system that is used only at night and a business application that is used only during the day might peacefully coexist on a single machine. Combining low-usage or complimentary applications on existing hardware can be a simple and effective approach to saving power.

Virtualization

Much like the multirole approach, virtualization lets the workloads of underutilized server machines be consolidated onto a smaller number of better utilized machines. Fewer physical machines can lead to reduced costs through lower hardware and energy costs and reduced management overhead.

As described in “Virtualization and Power Management in Windows Server 2008” later in this paper, PPM might not be as effective on virtualized systems. However, the savings from eliminating idle power overhead are usually greater, and virtualization has other value propositions.

Dynamic Provisioning

A third solution to overprovisioning is to dynamically allocate systems as required. Some enterprise data centers have developed automated solutions to achieve a pseudo-dynamically provisioned environment.

To save some power in this manner, you are not required to invest large amounts of money. For systems that have specific, fixed usage periods, such as nightly enterprise backup or build servers, you can implement scripts to turn off the machines and turn them on at certain times.

For applications that require many servers to meet peak demand, high-usage patterns might exist during business hours but systems remain near idle for the rest of the day. If your enterprise has a good understanding of the load patterns, you can develop automation to shut down some excess capacity during off-peak hours.

Virtualization and Power Management in Windows Server 2008

When you install the Hyper-V role on Windows Server 2008, the dynamics of power management change, especially when you are running multiple virtual machines (VMs). This section details how virtualization can affect power management.

Hyper-V Architecture

The operating system architecture of a system that is running the Hyper-V role on Windows Server 2008 is very different from a typical Windows Server 2008 installation. When this role is installed, a hypervisor sits between the partitions and physical hardware. Only the root partition accesses physical hardware. Any present VMs do not see physical processors and memory. Instead, the root partition presents system resources to the VMs as virtual devices.

PPM Implications

PPM on a server with the Hyper-V role installed has one subtle difference from native Windows Server installations. Multiple VMs can exist, each with its own PPM engine.

Hyper-V resolves this problem by allowing the root partition to control the power management policies for the entire system. Any VM power policy settings have no effect. This is a side-effect of a virtualized environment in which VMs do not interact with physical hardware. For example, VMs might have power management code in the form of ACPI-enabled processor drivers. However, the processor that is exposed to a VM is generic and hides all ACPI features such as P- and C-states, so PPM is not enabled in the VM.

Performance

Performance tuning can improve not only system responsiveness but power as well. Minimizing background work such as synthetic I/O and timer ticks for VMs reduces interrupt traffic and ensures that PPM effects are maximized. You should follow the performance tuning guide steps for virtualized systems in “Performance Tuning Guidelines for Windows Server 2008” on the WHDC Web site.

Hardware Component Effects on Power

Each hardware component in a server consumes power. Administrators should understand the optimizations that are available within each component and select efficient components to maximize power efficiency at a system level.

Processors

Modern processors include advanced power management features such as performance and idle states that operating systems can take advantage of to save power. Some processors take a different approach and utilize low operating voltages to increase power efficiency. Regardless of the specific implementation, choosing power-efficient processors for your data center is a good investment.

Performance States

According to the Green Grid's article "Five Ways to Reduce Data Center Power Consumption," enabling processor performance states on server platforms can reduce power consumption up to 20 percent.

Processor performance states can vary in number and implementation for each processor generation. For example, Intel's Xeon line of processors includes four P-states on the Harpertown family versus six P-states on the Tigerton, as shown in Table 5.

Table 5. P-State Support in Harpertown and Tigerton Processor Models

Processor family	Release date	P-state support
Harpertown	September 18, 2007	P0 @ 3,165-MHz (100%); P1 @ 2,666-MHz (84%); P2 @ 2,332-MHz (73%); P3 @ 1,999-MHz (63%)
Tigerton	September 5, 2007	P0 @ 2,931-MHz (100%); P1 @ 2,665-MHz (90%); P2 @ 2,398-MHz (81%); P3 @ 2,132,MHz (72%); P4 @ 1,865-MHz (63%); P5 @ 1,599-MHz (54%)

These processors were only chosen to exemplify P-state differences between processor lines. More P-states do not always guarantee increased efficiency, especially when you compare processors from two architectures or families. Benchmarks or manufacturer's datasheets are usually necessary to determine this information.

Idle States

Idle states present the best opportunity for overall power savings on a processor. The latest quad-core processors might see as much as a 60-watt difference between C0 and Cn power. Again, availability is specific to processor families and architectures, so refer to datasheets or benchmark tests to determine coverage and efficiency.

Low-Voltage Processors

A different variation of power-efficient processor also exists on the market today. These processors, known as "low voltage" or "ultra-low voltage" are exactly what their name implies. Some of these processors are designed to be power efficient without using performance states. Again, you should review datasheets and use benchmarks if you want to compare processors.

Memory

Memory is becoming an increasingly important element of server power consumption, especially in servers that have fully populated memory banks. As the section "Power Consumption of Idle Systems and Components" demonstrated, on a server in our lab, memory required 54 percent of total system power.

You must consider several parameters when you optimize memory purchases for power:

- Memory family (DDR, DDR2, DDR3)
- Bus speed
- Memory capacity

- Chip density
- Additional features

Memory Family

Generally, if all else is held equal, newer RAM families consume less power than older families because of new features that promote power efficiency. As long as bus speed, capacity, and density are the same, a double data rate 2 (DDR2) RAM module—or physical stick of memory—should consume less power than a DDR1 module.^{viii}

Bus Speed

Bus clock frequencies generally double each generation. Even within the same generation, bus speed increases correspond to a significant increase in power consumption. Generally, doubling bus speed doubles the power consumption.^{ix}

Memory Capacity

RAM capacity is the well-understood “size” (that is, 512 MB, 2 GB) of a module. Generally, higher capacity RAM consumes more power.^x

Chip Density

Dynamic RAM (DRAM) chip density refers to how much data can be stored in each chip. Each memory module has multiple chips. Generally, the higher the chip density, the lower the overall power consumption. If overall capacity is held constant, doubling the density of the memory chips reduces the number of chips on the module by half. This significantly reduces the power consumption.^{xi}

Additional Features

Some memory modules have additional features such as buffering stages and reduced chip counts. These features have a significant effect on memory power consumption.

Fully Buffered DIMMS

Instead of writing to memory directly, fully buffered DIMMs (FBDIMMs) introduce an intermediate buffering stage between the memory controller and the memory modules. This buffering stage can improve reliability and throughput, but has an associated increase in latency, and the power consumption of such DIMMs is generally double that of a comparable DDR2 module.^{xii} If you do not require the additional reliability or throughput, consider a platform that does not use FBDIMMs.

Low Chip Count and Low-Voltage Memory

Low chip count (LCC) and low-voltage (LV) memory are recent innovations for DDR2 and FBDIMM memory technologies that reduce the number of chips by half and reduce the operating voltage. According to manufacturers, these modules can reduce power consumption by 20 to 40 percent over standard modules.

Storage

You can achieve significant power reductions by selecting power-efficient hard disks for your data center. Although the absolute power savings per disk may seem small, quantities of hard disks that are deployed in a data center can easily number in the thousands. At this scale, choosing power-efficient hard disks can save many kilowatts of power.

Size Reduction

The simplest way to save power for storage is to migrate from 3.5-inch to 2.5-inch disk drives. This is one of the few scenarios in which saving power does not generally lead to a decrease in performance, and cost effects can be reduced by the long-term savings that the new platform provides.

The reduction in platter size and weight means that the disk can use smaller actuators and more power-efficient motors, which results in a large power savings. In our labs, 2.5-inch drives consume 50 percent less power than 3.5-inch drives that have similar rotational speed.

In addition, the performance of 2.5-inch drives is generally equal to or better than that of 3.5-inch drives, with a few caveats for specific workloads and data layouts. The complicating factor is that the sheer area of a 3.5-inch drive allows for higher capacity per spindle, which 2.5-inch drives have not yet achieved. Currently, it may be unfeasibly expensive or impossible to use 2.5-inch drives to build low-power, high-capacity installations.

RPM Reduction

Enterprise-class, 15,000-RPM disks offer the highest performance from rotational media disk drives, but they are not always necessary. In many scenarios, you can deploy lower RPM disk drives (such as 10,000 or even 7,200 RPM) in a storage array and still accomplish your business goals. If large sequential reads and writes are not part of your workload, this power optimization has a small effect on performance.^{xiii}

Solid-State Disks

Solid state disks (SSDs) have no magnetic platters to rotate, eliminating power costs for disk motors and actuators—which are the most significant items on the power budget for rotational drives. Although client rotational drives spin their media down at idle periods to save power, enterprise drives do not. Therefore, SSDs generally use much less power than hard disk drives (HDDs). However, SSD technology is changing rapidly. If you are considering SSDs for your IT environment, you must consider not only their power usage, but also their expected lifetime and how much your application workload can take advantage of solid-state storage technology.

RAID Selection

In the past, power has not generally been a factor in choosing a RAID setup. However, power capacity can influence this decision. If your infrastructure is approaching its maximum power capacity, the power budget of large RAID arrays can make expansion of existing RAID setups or purchase of new setups impossible.

If replacing existing 3.5-inch with 2.5-inch disks of identical capacity is not possible and your organization is at its power capacity limit, one way to free some capacity is to change your RAID setup. For example, you might move from a high degree of reliability and performance (RAID 10) to a lower degree (RAID 5). Of course, you should evaluate if the performance, redundancy, and availability of different RAID setups can still achieve your business requirements. For detailed information on this tradeoffs, see “Disk Subsystem Performance Analysis for Windows” on the WHDC Web site.

Network Adapters

To minimize network adapter power consumption, purchase only as much capacity as you need. If a server has low utilization or does not require a large amount of bandwidth, purchasing the highest throughput network adapters is unnecessary and consumes more power. Typical power consumption in our lab for a 1-GBPS PCIe network adapter was 10 watts, whereas a 10-GBps PCIe card consumed about 17 watts.

Another thing to consider is the number of ports on the card. If four connections are required, a single four-port card consumes significantly less power than four individual single-port cards. Current quad-port 1-GBps cards consume around 17 watts of power, whereas four 1-GBps single-port cards typically consume almost 40 watts.

Remote Power Strips

As demonstrated in “Power Consumption of Idle Systems and Components” earlier in this paper, servers that have been shut down may still consume 27 to 30 watts of power. Physically unplugging the servers is one way to fix this problem, but administrators likely lack the resources to unplug and plug in servers each day.

Power strips that can be controlled remotely by networking or other means let administrators automate this process. Again, this introduces an up-front cost that may be recouped over time.

Table 6. Annual Remote Power Strip Energy Savings

Strip size	Energy savings ¹ (W)	Annual energy savings ² (kWh)	Annual savings (\$)
4 outlet	120	612.8	66.67
8 outlets	240	1225.7	133.35
16 outlets	480	2451.4	266.71

¹ Assumes 30-watt savings per outlet.

² Assumes servers are powered off for 14 hours per day.

Cooling

A commonly disregarded or forgotten element of server power consumption is on-board fans. Most servers use multiple fans in box to generate the required airflow.

Variable-speed fans let the platform reduce the fan RPM rate when the server is not under peak load. The paper “Data center TCO benefits of reduced system airflow” submitted to the IThERM conference found that fans in some 1U rack-mounted servers consume 15 to 20 percent of overall system power. This can translate to a

significant power saving during off-peak periods in the data center. Variable-speed fans typically require additional support from the platform, so this may not be a simple power savings option for existing installations.

Power Supply Units

Power supply units (PSUs) perform an AC/DC conversion. This process has built-in inefficiencies, so much so that PSUs have received increased attention from industry and government.

Power Supply Unit Efficiency

PSUs have two key measures of efficiency:

- Power factor
- Ratio of input/output power

Power factor measures how in-phase the input voltage and current waveforms are. Ensure that your power supplies use *active* power factor correction (PFC) to reduce inefficiency.

The ratio of input-to-output power is the key determinant of power supply quality. Power supplies must be less than 100-percent efficient in this category because of the conversion from AC to DC power. With the recent change in power trends, ENERGY STAR standards, and the institution of the 80 PLUS program, new designs are coming out that raise the overall efficiency significantly.

Whereas default supplies were typically about 70-percent efficient, *The Register* reports that new models from vendors such as Dell, SGI, and Rackable are over 86-percent efficient across the entire load spectrum. This translates to a significant percentage reduction in power waste and excess heat in your data center—a real candidate for cost savings.

Consider Table 7 **Error! Reference source not found.**, which details the effects of power supply efficiency on a 12-U machine that requires 500 watts of power.

Table 7. Effects of Power-Supply Efficiency on 12-U Server Consuming 500 W of Power 24x7

Efficiency	Output power (W)	Required input power (W)	Waste power (W)	Annual waste power cost (\$)
70 (default)	500	714	214	203.96
80 (near 80 plus Bronze)	500	625	125	119.14
85 (80 plus silver)	500	588	88	83.87
90 (above 80 plus gold)	500	555	55	52.42

On a 70-percent efficient power supply, 714 watts of wall power is required to supply 500 watts to the system. The additional 214 watts is waste power, much of which is converted into heat and requires additional costs for cooling and airflow infrastructure. A 90-percent efficient supply requires only 555 watts, a saving of 159 watts for one machine. If this is a 24x7 server, you will realize annual savings of over \$151 on your power bill. If cooling costs are added, this might be as high as \$200 or more. For a farm of 1,000 such servers, efficient power supplies could save your organization \$200,000 a year.

Efficiency Programs

Government programs such as ENERGY STAR in the United States and industry certification programs such as 80 PLUS are working to test and ensure the efficiency of power supplies and server systems. These organizations publish results that might help you find the most ideal power supply for your particular configuration.

You can find more information on these programs in “Resources.”

Platform Power Management and Budgeting

Innovations in data center power management such as power budgeting and hardware-based PPM can change how systems behave. This section outlines these two technologies and points out potential side-effects for administrators.

Hardware-Based Processor Power Management

Original equipment manufacturers (OEMs) have devised firmware features to dynamically manage processor P-states. These devices adjust P-states many thousands of times per second. However, these devices lack access to information such as usage history, system I/O patterns, and other workload details that are visible to the operating system.

Operating systems can use this data to predict future computational power needs and set P-states accordingly. We believe that the operating system can deliver the most efficient PPM possible. By using the strategies in this paper, you can configure Windows Server 2008 to achieve efficiency improvements over hardware-based PPM. In our laboratory, we have seen 2- to 10-percent efficiency improvements over hardware-based implementations at different points on the load line.

One side-effect of hardware-based PPM is the potential for resource conflicts. For example, some hardware PPM schemes require the use of processor performance counters. System management and monitoring tools and third-party applications that rely on these counters cannot function if they are being used by hardware PPM.

Power Budgeting

Data center management tools are beginning to include power budgeting and monitoring tools. Power budgeting is a process that lets administrators set power limits, or caps, on data center components as small as a rack or a single server.

Implementations for enforcing power caps differ. Many power budgeting solutions developers use proprietary mechanisms to throttle back CPU power consumption. Generally, hardware on the systems or racks that are approaching their power budget lower CPU P-states or enable processor clock throttling. When machines exceed their budget, the power budgeting system asserts thermal emergency pins on the processor, which causes the CPU to drastically reduce its throughput.

Use caution if power budgeting systems in your data center use these mechanisms. The Windows Server 2008 power management algorithms throttle down P-states to save power when a system is at low utilization and throttle up to P0 for maximum performance when the system is under load. A power budgeting system does the opposite—when machines are under load (which increases their power usage), the

management software drops available computing capacity. Removing computing resources when these resources are in demand can adversely affect response times and Quality of Service.

For more information about correctly implementing and deploying power budgeting, see “Recommendations for Power Budgeting with Windows Server.”

Data Center Infrastructure

Data center real estate, high-capacity power and HVAC equipment, network routing and UPS systems, maintenance for all these items, and the power that is required to operate them can easily equal or exceed the costs of computing equipment and computing equipment power year over year.

You can directly reduce these costs by using innovative management technologies and intelligent planning. One such tool is the Microsoft Assessment and Planning (MAP) Toolkit. For a link, see “Resources.”

External resources also exist. Many companies specialize in planning for infrastructure efficiency, and we are a participating member of the Green Grid consortium, a group that is dedicated to improving energy efficiency in data centers.

For links to Web sites that can provide more information, see “Resources.”

Resources

WHDC Web site white papers

Disk Subsystem Performance Analysis for Windows

http://www.microsoft.com/whdc/device/storage/subsys_perf.msp

Interrupt-Affinity Policy Tool

<http://www.microsoft.com/whdc/system/sysperf/intpolicy.msp>

Performance Tuning Guidelines for Windows Server 2008

http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.msp

Processor Power Management in Windows Vista and Windows Server 2008

<http://www.microsoft.com/whdc/system/pnppwr/powermgmt/ProcPowerMgmt.msp>

Recommendations for Power Budgeting with Windows Server

http://www.microsoft.com/whdc/system/pnppwr/powermgmt/Svr_PowerBudget.msp

Windows Server 2008 Power Savings

<http://www.microsoft.com/downloads/details.aspx?FamilyID=61d493fd-855d-4719-8662-3a40ba3a0a5c&displaylang=en>

Microsoft Tools and Web sites

Microsoft Environmental Sustainability Webpage

<http://www.microsoft.com/environment>

Microsoft Assessment and Planning Toolkit

<http://www.microsoft.com/downloads/details.aspx?FamilyID=67240b76-3148-4e49-943d-4d9ea7f77730&DisplayLang=en>

United States Government

Energy Information Administration, "Average Retail Price of Electricity to Ultimate Customers by End-Use Sector, by State"

http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_a.html

"ENERGY STAR Server Power Supply Specification Draft"

http://www.energystar.gov/ia/partners/prod_development/new_specs/downloads/Draft1_Server_Spec.pdf

Environmental Protection Agency (EPA), "Report to Congress on Server and Data Center Energy Efficiency," August 2, 2007

http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Report_Exec_Summary_Final.pdf

EPA, "New Product Specifications in Development: Enterprise Servers"

http://www.energystar.gov/index.cfm?c=new_specs.enterprise_servers

Organizations

80 PLUS Program

<http://www.80plus.org>

ACPI Specification, Revision 3.0b

<http://www.acpi.info/spec.htm>

Applied Power Electronic Conference (APEC)

Details on power supply efficiency testing, metrics, power supply design standards and proposed ENERGY STAR requirements

http://www.apec-conf.org/2006/APEC_2006_SP4_2.pdf

Forrester research, "Firms Still Struggle To Predict Capacity Utilization Accurately," by Laura Koetzle, 17 May 2006

http://www.bmc.com/USA/Corporate/attachments/Forrester_Capacity_Utilization_Survey_May.pdf

Standard Performance Evaluation Corporation, "SPECpower_ssj2008"

http://www.spec.org/power_ssj2008/

The Green Grid, "Five Ways to Reduce Data Center Power Consumption," by Mark Blackburn, 2008

http://www.thegreengrid.org/gg_content/White_Paper_7_-_Five_Ways_to_Save_Power.pdf

Transaction Processing Performance Council, "TPC Benchmark E (TPC-E)"

<http://www.tpc.org/tpce/default.asp>

Articles and Papers

***Electronics Cooling*, “In the Data Center, Power and Cooling Costs More than the IT Equipment It Supports,” by Christian Belady, February 2008**

<http://electronics-cooling.com/articles/2007/feb/a3/>

“Data center TCO benefits of reduced system airflow,” by C. G. Malone, W. Vinson, and C. E. Bash, *Thermal and Thermomechanical Phenomena in Electronic Systems*. IThERM 2008 11th Intersociety Conference, 28-31 May, 2008, pp. 1199-1202

<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4544397&isnumber=4544243>

***TechNet Magazine*, “Sustainable Computing: Is It Time to Turn Off Your Servers?” by Dave Ohara**

<http://technet.microsoft.com/en-us/magazine/cc700341.aspx>

***The Register*, “Dell develops ultra-efficient server power supplies,” by [Austin Modine](#)**

http://www.theregister.co.uk/2008/06/25/dell_80_plus_gold_power_supply/

Appendix A. Power Policy Options in Windows Server 2008

The available power policy options in Windows Server 2008 are both numerous and complex. Descriptions of these options follow.

Control Panel Power Policy Options

Windows Server 2008 exposes a basic set of PPM options in the Control Panel **Power Options** application as listed in Table A-1, with default settings given for the Balanced power policy.

Table A-1. Control Panel PPM Options

Description	Name	Default	Range	
			Minimum	Maximum
PPM Parameters	Throttle State Enable	*	Off	On
	Minimum Processor Performance State	5%	5%	100%
	Maximum Processor Performance State	100%	5%	100%
Non-PPM Parameters	USB Selective Suspend*	Disabled	Disabled	Enabled
	PCI Express ASPM*	Moderate	Off	Maximum
	Search and Indexing Power Savings*	High Performance	Power Saver	High Performance

Items marked with an asterisk () may not appear by default.

Throttle State Enable

The throttle state enable option enables clock throttling or T-states. This option is available only for legacy processors that do not support P-states. It is turned on by default for such processors. Otherwise, this option is disabled.

Minimum/Maximum Performance State

This percentage represents the minimum and maximum performance state (as a percentage of maximum processor throughput) that the PPM algorithm considers. The effects on power and performance are self evident.

High-performance and power-saver plans adjust this parameter to keep processors at the highest or lowest possible P-state at all times.

USB Selective Suspend

This option appears on configurations that support USB and enable selective suspend. Selective suspend lets the USB I/O system enter a low-power state and wake only when it is necessary, which saves system power.

However, poorly written USB drivers can wreak havoc with PPM. If USB support is not necessary, we recommend disabling USB support in the BIOS instead of using USB selective suspend to save power.

PCI Express ASPM

Active state power management (ASPM) lets PCIe devices save power by throttling back the I/O bus bandwidth. If power is an important concern, increasing this setting to maximum can save some power.

Search and Indexing Power Savings

This setting is visible if the Windows Search Service is installed as a Server Role through the File Services role in Server Manager. This setting controls how active the indexing service is at any time. If this role is installed and power savings is a concern, we recommend setting this parameter to Power Saver.

Advanced PPM Performance State Options

Performance state transitions are most significantly affected by the eight parameters in Table A-2 and Table A-3. **Error! Reference source not found.** These parameters affect the performance state change frequency, tendency to change, and degree of change, as summarized on the left.

Important: Modifying any of these parameters changes the behavior of performance state handling from the out-of-box experience. Before you deploy these to production servers, validate the effects of any changes in a test environment.

Table A-2. Default Settings for PPM Performance State Parameters

Description	Name	Default	Range	
			Minimum	Maximum
How Frequent	Time Check	100ms	1 ms	5,000 ms
	Increase Time	100 ms	5%	1,000 ms
	Decrease Time	300 ms	5%	1,000 ms
Change P-state or not	Increase Percentage	30%	Disabled	80%
	Decrease Percentage	50%	Off	80%
	Domain Accounting Policy	On (0)	On (0)	Off (1)

Time Check

The time check parameter determines how often the system considers a change in performance state. Increasing this parameter leads to less frequent P-state management, whereas decreasing it leads to more frequent management.

Although the P-state code has a fairly small footprint, tuning this value to very low values can adversely affect performance.^{xiv}

Increase and Decrease Time

Increase and decrease time are the time intervals at which the system considers increasing or decreasing P-state. Effectively, these parameters control when code paths for increase and decrease decisions are made. The concepts, caveats, and warnings that apply to time check also apply here.

Increase Percentage and Decrease Percentage

Increase percentage and decrease percentage define utilization “boundaries” that must be crossed to raise or lower P-state. For example, if the increase percentage is

30 percent, when the operating system considers a P-state increase, any processor utilization greater than 30 percent triggers a transition to a lower numbered P-state.

Decrease percentage works in the opposite way. If the decrease percentage is 50 percent, when the operating system considers a P-state decrease, any processor utilization that is below 50 percent triggers a transition to a higher numbered P-state.

Domain Accounting Policy

When this setting is at the default of 0, all processors in a domain must be idle before the engine considers lowering performance states.^{xv}

If you change this to 1, each logical processor receives its own count. Although all processors in a domain still must transition together, the net effect is that higher numbered P-states are entered more frequently, which potentially saves power.

Table A-3. P-State Increase and Decrease Parameter Defaults

Description	Name	Default	Available options
How to Change	Increase Policy	IDEAL (0)	IDEAL (0)
			SINGLE (1)
			ROCKET (2)
	Decrease Policy	SINGLE (1)	IDEAL (0)
			SINGLE (1)
			ROCKET (2)

Increase Policy and Decrease Policy

When the operating system decides to make a P-state transition, it need not transition a processor to the next highest or lowest P-state—any P-state in the supported set may be chosen. Increase and decrease policy lets administrators control how transitions between P-states occur.

The ideal policy calculates a target P-state out of the entire set of P-states that the system exposes. Under ideal conditions, transitions from any P-state to any P-state are possible.

The single policy allows increases or decreases by just one P-state per time check interval.

If a P-state transition is considered necessary under the rocket policy, the processor always transitions to the highest or lowest currently available P-state, P₀, or P_n. Intermediate P-states are ignored.

Advanced Idle State Options

Idle state PPM is controlled by a different subset of parameters. The idle state parameters given in Table A-4 control how often the operating system moves processors between idle states.

Table A-4. PPM Idle State Parameters

Description	Name	Default	Range	
			Minimum	Maximum
Entry Idle (promote only)	C1 Time Check	50 ms	1 ms	200 ms
	C1 Promote Percent	20%	1%	100%
Transition Idle	C2 Time Check	50 ms	1 ms	200 ms

			Range	
(promote or demote)	C2 Demote Percent	20%	1%	100%
	C2 Promote Percent	40%	1%	100%
Deep Idle (demote only)	C3 Time Check	50 ms	1 ms	200 ms
	C3 Demote Percent	20%	1%	100%

Time Check

The time check parameter determines how often a change in idle state is considered. Unlike P-states, during each time check both promotion and demotion are considered. Increasing this parameter leads to less frequent C-state management, whereas decreasing it leads to more frequent management.

Promote and Demote Percentage

These percentages resemble the increase and decrease percentages for performance states. However, instead of measuring processor utilization, these percentages refer to how idle the operating system was over the previous time check interval.

Appendix B. Changing P-State Parameters by Using Powercfg.exe

Important: Use of these command parameters changes the behavior of performance state handling from the out-of-box experience. Before you deploy these command parameters, validate the effects of any changes in a test environment.

Get the current binary dataset that represents the power setting definition for P-states:

```
powercfg /getpossiblevalue sub_processor procperf 1
Type: BINARY
Value: 640864000000A0860100E09304001E00000032000000
```

The parameter values for this dataset can be shown with the decode command:

```
powercfg /ppmperf /decode 640864000000A0860100E09304001E00000032000000
Busy Adjust Threshold: 100
Time Check: 100
Increase Time: 100000
Decrease Time: 300000
Increase Percent: 30
Decrease Percent: 50
Domain Accounting Policy: 0
Increase Policy: 0
Decrease Policy: 1
```

To change individual parameter values to match the “Aggressive” settings that were described in this paper, use the following command:

```
powercfg /ppmperf /encode base
640864000000A0860100E09304001E00000032000000 /decreasetime 100000
/domaaccountingpolicy 1 /increasepolicy 1 /decreasepolicy 0
640364000000a0860100a08601001e00000032000000
```

Apply the new dataset by using the **setpossiblevalue** command:

```
powercfg /setpossiblevalue /sub_processor /procperf 2 binary
640364000000a0860100a08601001e00000032000000
```

Finally, use the **setactive** command to enable the new parameter set (no reboot is necessary):

```
powercfg /setactive scheme_balanced
```

Appendix C. Viewing PPM Counters by Using Perfmon.exe

This walkthrough guides you through the process of adding PPM-related performance counters to the Performance Monitor (Perfmon) tool in Windows Server 2008. It also demonstrates how the counters change under different machine loads and power policies. You can use this information to identify and fix issues that affect power management on your servers.

First, open the Performance Monitor tool by opening a new command windows and typing "perfmon". Under **Monitoring Tools**, double-click **Performance Monitor**.

Performance Monitor initially shows a graph of % Processor Time. Clear this counter by right-clicking the graph window and selecting **Clear All Counters**.

Next, right-click the graph window and select **Add Counters**. After the counter sets finish loading, locate the **Processor Information** performance counter group in the **Available Counters** frame and expand the group by clicking the group name.

Next, add the following counters to the performance monitor. You can press and hold the Ctrl key to select multiple counters at a time:

- % C1 Time
- % C2 Time
- % C3 Time
- % Idle time
- % of Maximum Frequency
- Interrupts / Sec
- Processor Frequency

After you have selected these counters, a set of instances appear in the lower-left pane that are labeled **Instances of selected object**. Each logical processor generally counts as one instance for processor performance counters, as well as an overall **Total** instance. For each counter, select the **All Instances** option, and then click the **Add** button below the pane to add the counters to Performance Monitor.

When you are finished, your screen should look similar to Figure C-1.

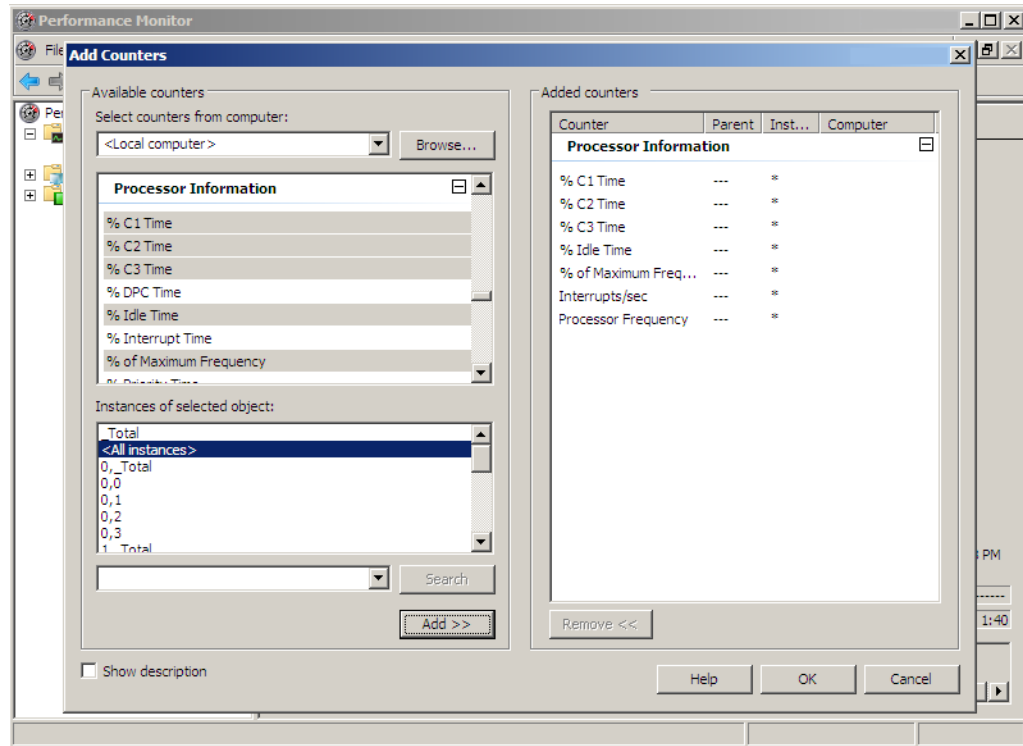


Figure C-1. Adding selected instances of the Processor Information performance counters

To return to the performance monitor graph with your new counters loaded, click OK.

The next step is to configure this data to a more easily read format. You can do this in three ways:

- Click the **Change Graph Type** button two times.
- or-
- Use the drop-down arrow to the right of the **Change Graph Type** button, and then select **Report**.
- or-
- Press Ctrl-G two times to change the graph format to **Report**.

The drop-down menu location is shown in Figure C-2.

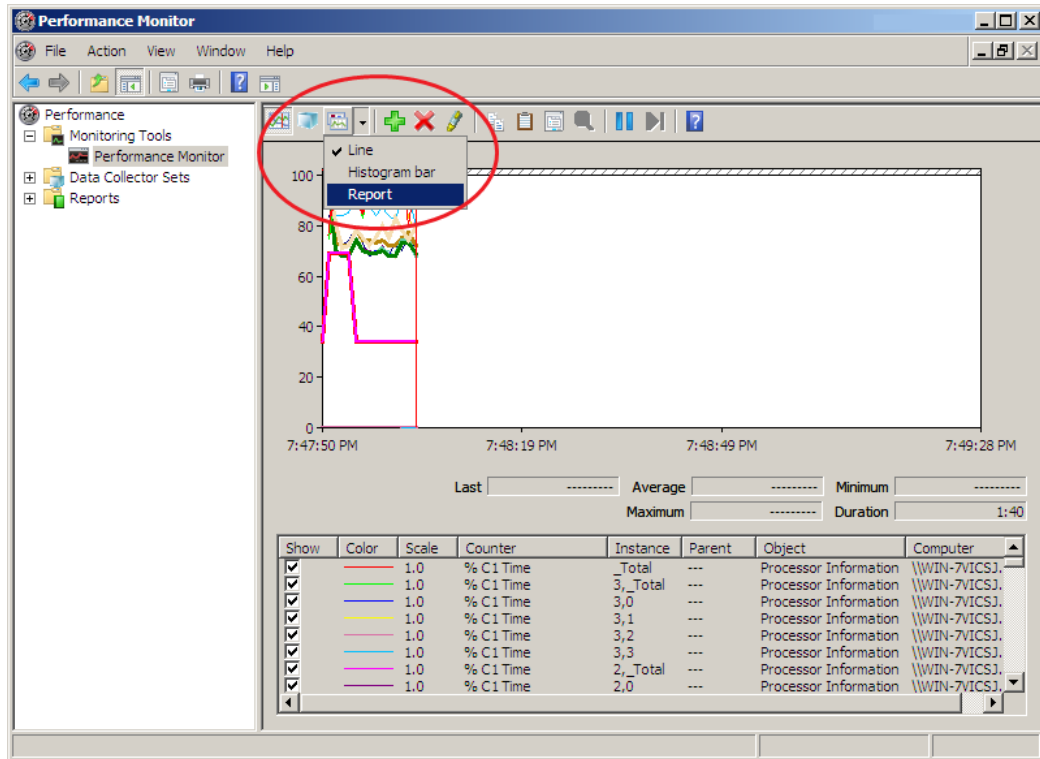


Figure C-2. Changing the performance monitor to Report mode

In **Report** format, you now see easy-to-read tables of data for your system. First, we will review what each row and column of the table means. Each row corresponds to a particular performance counter, which is given in the left-most column.

The columns correspond to individual logical processors, processor packages, or totals. In this example, the machine that was used to generate these screenshots has 4 processor sockets. Each processor socket has 4 processor cores, for a total of 16 logical processors.

The naming convention for these processors in Performance Monitor is <socket>,<core>. As shown in the rightmost two columns of Figure C-3, the column titled 0,0 contains the counter data for socket 0, core 0, while 0,1 contains the counter data for socket 0, core 1. This convention continues for a sixteen logical processor configuration up to socket 3, core 3.

In Performance Monitor, special *Total* instances also exist for each socket and for all logical processors overall. The 0,_Total column represents the counter totals for all of the cores on socket 0, while 3,_Total represents the counter totals for the cores on socket 3. The second column in Figure C-3, which is labeled “_Total,” contains the counter totals for all 16 cores on the server.

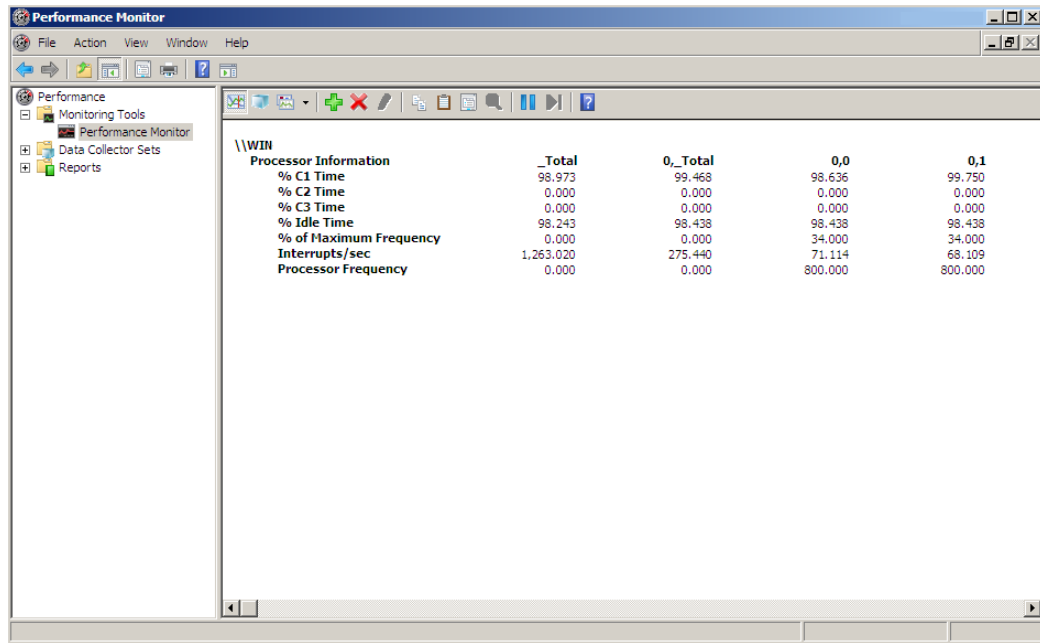


Figure C-3. Processor Information counters active on a 16- processor system in Balanced mode

The `_Total` column is an average (for the percentage-based counters and Processor Frequency counter) or a sum (for the Interrupts/Sec counter) of the values across all processors in the system. Each column to the right of the `_Total` column represents a logical processor, so the `_Total` column is an average or a sum of all these columns.

A simple example of this behavior can be seen in the Interrupts/sec counter. The `_total` interrupts per second counter counts all interrupts on the system, whereas the individual processor columns show the number of interrupts per second that occurred on each logical processor.

Perfmon counters fluctuate with the load on your system. As load increases, you should see drastically different values in some of these counters. These counters also indicate whether multimedia timers are enabled. As an example, examine the counters for a sixteen core server at idle with the Balanced power policy selected and no multimedia timer, given in Figure C-3.

The system is spending 98 percent of its time at idle and approximately 99 percent of the time in idle state C1. Cores 0,0 and 0,1 are at 34 percent of their maximum frequency, a mere 800 Hz. Interrupt count is low—approximately 300 per socket, 75 per processor, for a total of 1,263 per second.

Now compare that to the same system with the High Performance power policy and a 1-ms multimedia timer enabled, as given in Figure C-4. No work is actually being done by the system; the % Idle Time and % C1 Time counters are still at 99 percent. Yet the % of maximum frequency for Cores 0,0 and 0,1 jumped to 100 percent and the frequency increased to 2,300 MHz. Interrupt counts have skyrocketed to over 1,000 per processor, for a total of 16,555 per second across the system.

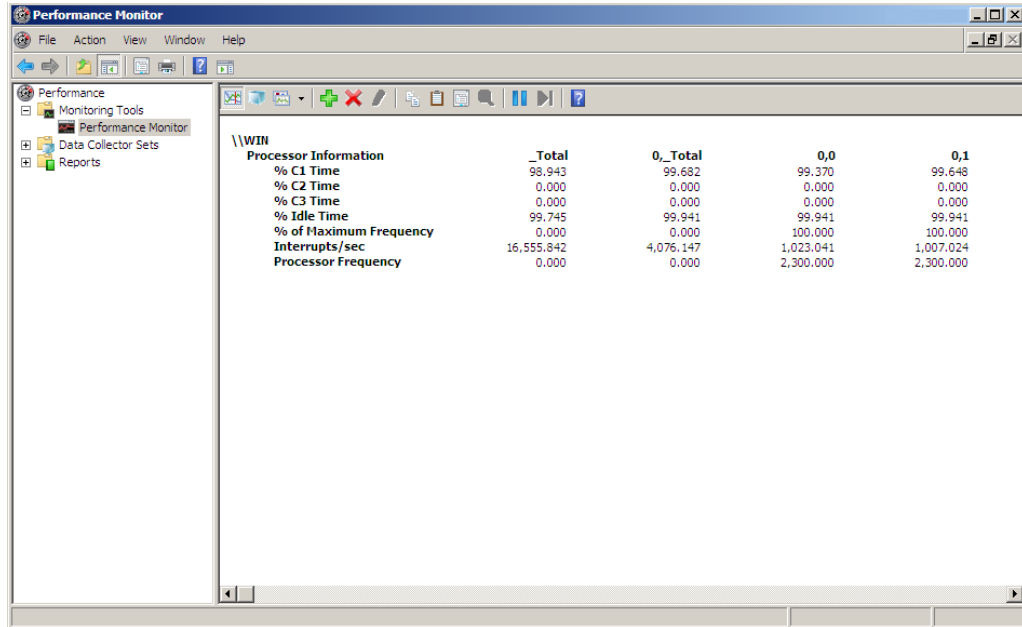


Figure C-4. Counter values with High Performance power policy and a multimedia timer enabled

A system with a 15.68-ms timer resolution has a minimum 63 interrupts per second per processor. Expect a system at or near idle to have more than this amount, generally somewhere in the range of 150 to 400 per second per processor. Although your results will vary, the key point is that at idle, interrupt counts of over 1,000 interrupts per second per processor are typically generated by a 1-ms application timer or a misbehaving driver. Remember that systems under moderate or heavy load can easily generate thousands of interrupts per second per processor.

We reiterate that these changes in Interrupts/sec and % of Maximum Frequency were not caused by any change in workload. In both scenarios, these servers were almost completely idle. This highlights the importance of correct configuration and system monitoring to ensure that your servers are achieving maximum efficiency.

The final scenario we will demonstrate is how the counters change when workloads are introduced onto the system. We ran a small application that allocated memory from the system in a tight loop and captured the performance counter changes in Figure C-5.

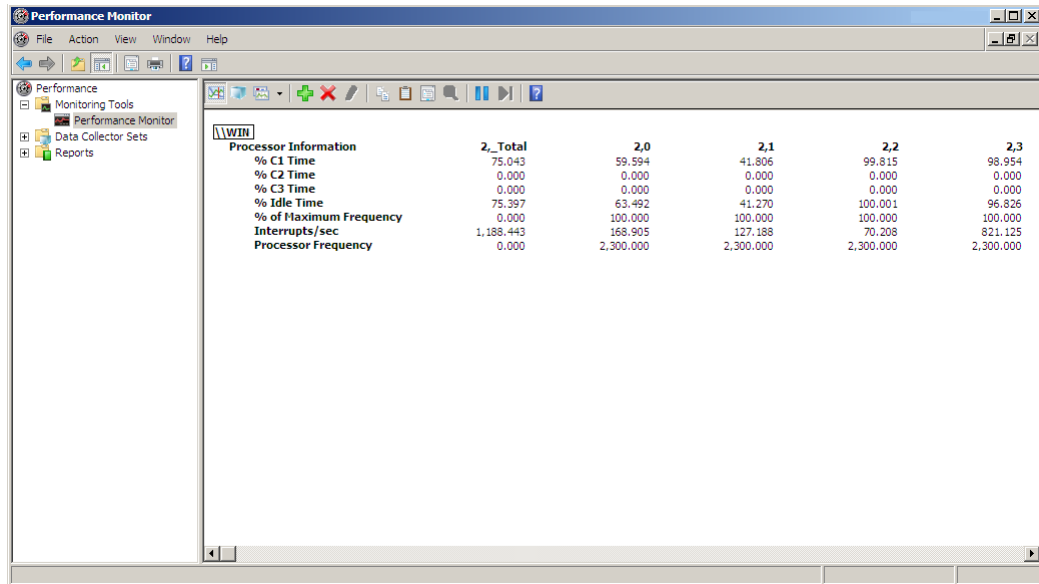


Figure C-5. Counter values during a memory allocation workload

The counter data for the socket (column 2,_Total) shows that the %C1 Time and %Idle Time counters have reduced significantly under this workload from 99 to 75 percent. The PPM engine also raised the processor frequency to 100 percent to handle the load. The number of interrupts per second on the socket increased from their idle value of 300 to over 1,000, but this is still far from the 4,000 interrupts per socket that were caused by the 1-ms timer.

You may notice that the counters shown in Figure C-5 were not from socket 0, but instead from socket 2. You might also notice that cores 2,2 and 2,3 have a much higher idle time than cores 2,0 and 2,1. This demonstrates how workloads are not always evenly distributed among all logical processors on a system. Often, one processor in a socket is utilized while others are almost completely idle. It is best to use the _Total column to obtain an accurate picture of your overall server CPU utilization.

End Notes

ⁱ The first transistor circuits were built by using passive components that consumed power while the device was inactive. The complementary metal-oxide semiconductor (CMOS) transistor eliminated this static power consumption from digital circuits many years ago, which reduced transistor power consumption to that required for switching events. As transistor size passed the 180-nanometer (nm) mark, leakage currents reintroduced static power consumption. The result has been a substantial year-to-year increase in power consumption and density.

ⁱⁱ For example, if P0 is 2,500 MHz at 1.8 volts, P1 could be 2,200 MHz at 1.8 volts, 2,500 MHz at 1.75 volts, or 2,200 MHz at 1.75 volts. All of these are valid performance reductions. Frequency is exposed to the operating system through a performance counter, so it is relatively easy to determine whether a reduction in performance state affects frequency or voltage.

ⁱⁱⁱ The operating system can do this effectively by using the historic CPU utilization. The operating system uses a predictive model that is based on historic utilization data to calculate the expected throughput that is necessary for the next time interval and chooses to increase, decrease, or maintain P-state or possibly enter an idle state accordingly.

^{iv} The combinatorial complexity of the problem becomes apparent after a bit of thought. If you have four processors that can do 400 cycles of possible work each over a specified unit of time at full throughput, which P-state and C-state combinations lead to the most power efficient setup for calculating 200 operations? 400 operations? 1000? Also consider that servers do not know exactly how much work they must perform in the next time quantum. The algorithm must consider historic workload data and make an accurate prediction.

^v Instruction and data caches are not essential for correct execution, so the data in them can be discarded without affecting functionality. However, these caches improve performance significantly, so the system suffers a large performance loss if they must be refilled during execution. An alternative option is to write the cache data to the next highest level of cache memory, shut down, and read the data back in when the system resumes from the idle state. Of course, this significantly increases the latency of a state transition.

^{vi} The rationale for this is as follows: If the processor enters a low idle state and is called back to C0 immediately, this causes a net efficiency loss for the platform. However, if the core enters C1 and remains there for tens of milliseconds, the power savings can outweigh any performance loss, which results in a net efficiency gain.

^{vii} Some drivers and devices employ a technique called *polling* to carry out I/O operations. Rather than stop execution and let the processor enter an idle state, drivers that poll use a tight loop to repeatedly check if I/O has completed. Although this does not require a later interrupt that can potentially wake a processor from an idle state, polling requires the processor to remain active and consume power during the entire operation. The best approach to use depends on the latency of the I/O request and can employ both strategies.

^{viii} Original DDR ran at a nominal voltage of 2.5 volts. DDR2 lowered this to 1.8 volts, and DDR3 specifies a nominal voltage of only 1.5 volts. Although the voltage drop alone saves power, each generation of DDR RAM also includes power management features that previous generations lacked. DDR2 introduced self-refresh states, lower activate and standby power consumption, and 4-bit prefetching, which allowed it to save 65 percent on power

consumption during its highest active operating condition. DDR3 introduces 8-bit prefetching and dual-gate transistors to lower leakage current.

^{ix} For example, according to our test results, a 2-GB DDR2 stick of PC2-4200 RAM that has a bus speed of 533 MHz consumes 12.3 watts of power. A 2-GB DDR2 stick of PC2-8500 RAM (1,066-MHz bus speed) that has the same density consumes 23.8 watts, which is almost double the amount.

^x This is true if all other parameters are held equal. Doubling the capacity at the same density requires doubling the number of chips on the module and approximately doubles the power consumption. However, increases in capacity can be offset by an accompanying increase in chip density, and in some cases, this can actually result in power savings. For example, our test results show a 1-GB stick of PC2-8300 memory that has a chip density of 512 Mbit consumes 12.1 watts of power whereas a 2-GB stick of the same RAM speed that has a 1-Gbit chip density consumes 11.0 watts.

^{xi} For example, a 2-GB DIMM with a chip density of 512 Mbit requires 36 memory chips. According to our test results, a module that has a bus speed of 1,066 MHz consumes 23.8 watts. A module with 1-Gbit chip density requires 18 chips and only consumes 10.9 watts.

^{xii} The buffering stage in an FBDIMM can act as signal conditioning and error correction logic, which increases overall reliability. This stage also lets a memory controller issue reads and writes in parallel, which increases memory bandwidth. However, instead of directly reading from or writing to memory, the memory controller writes to the buffer, which then writes to memory, which adds an extra amount of latency to each request.

^{xiii} In random access scenarios, our tests have shown that the additional rotational latency from stepping down to a lower RPM class (for example, from 15,000 to 10,000 or from 10,000 to 7,200) can be as low as 1ms.

^{xiv} While relatively small sections of code (such as 10,000 instructions) have a negligible impact on a processor that can perform billions of instructions per second, when this code is run thousands of times per second, suddenly the number of instructions executed is nontrivial. This issue is called *code path overhead*. Although the PPM engine code is relatively small, a time check value of 1 ms requires the code to execute 1,000 times per second, which can affect performance.

^{xv} To reduce complexity and cost, some hardware manufactures share resources such as memory controller, clocks, and power distribution planes between processor cores and sockets. Generally, this means that the processors that share these resources remain at the same performance and idle states. This requirement is called a *dependency domain*, which is referred to by some parameters as simply a *domain*.