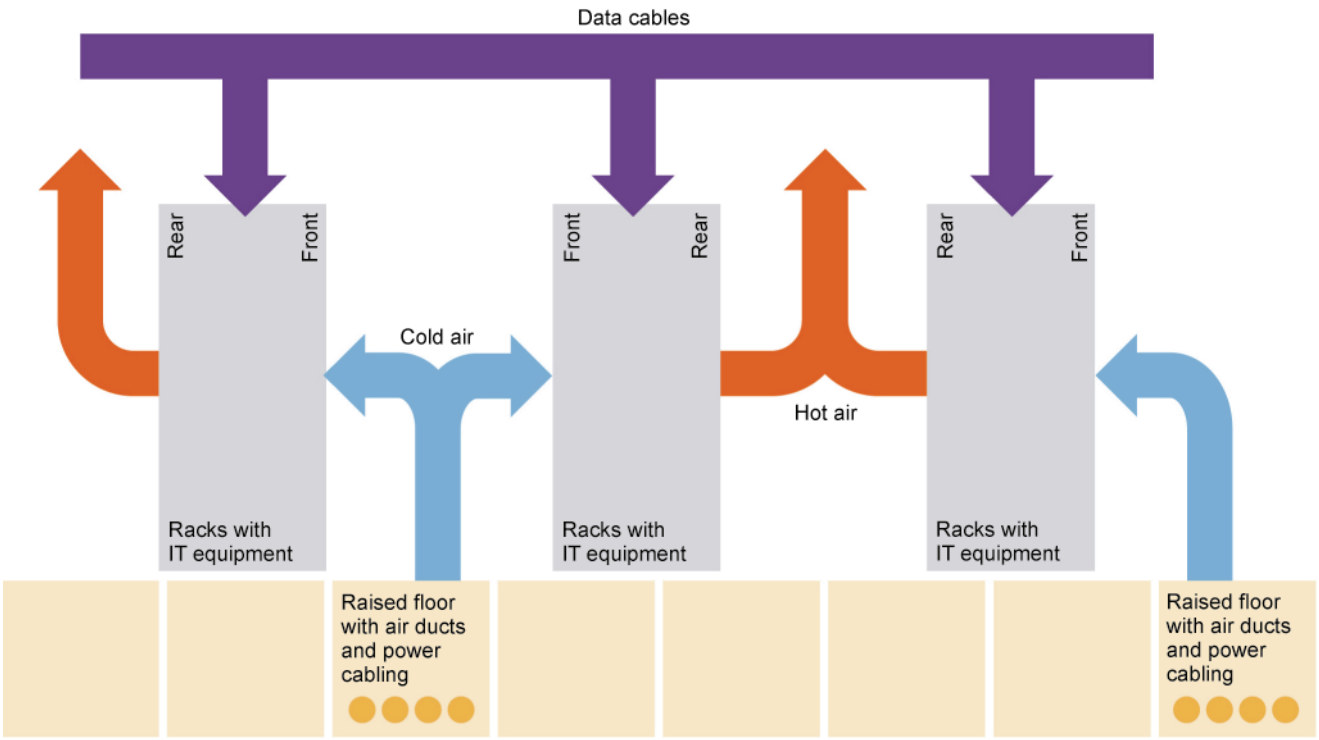
Green Sustainable Data Centres

Operating Systems & Virtualisation



This course is produced under the authority of e-Infranet: http://e-infranet.eu/

Course team
prof. dr. Colin Pattinson, Leeds Beckett University (United Kingdom),
course chairman and author of Chapter 1 and 7
prof. dr. Ilmars Slaidins, Riga Technical University (Latvia*),*
assessment material development: Study Guide
dr. Anda Counotte, Open Universiteit (The Netherlands),
distance learning material development, editor- in-chief
dr. Paulo Carreira, IST, Universidade de Lisboa(Portugal),
author of Chapter 8
Damian Dalton, MSc, University College Dublin (Ireland),
author of Chapter 5 and 6
Johan De Gelas, MSc, University College of West Flanders (Belgium),
author of Chapter 3 and 4
dr. César Gómez-Martin, CénitS - Supercomputing Center and
University of Extremadura (Spain),
author of Checklist Data Centre Audit
Joona Tolonen, MSc, Kajaani University of Applied Sciences (Finland),
author of Chapter 2

Program direction
prof. dr. Colin Pattinson, Leeds Beckett University (United Kingdom),
prof. dr. Ilmars Slaidins, Riga Technical University (Latvia)
dr. Anda Counotte, Open Universiteit (The Netherlands)

Hosting and Lay-out
http://portal.ou.nl/web/green-sustainable-data-centres
Arnold van der Leer, MSc
Maria Wienbröker-Kampermann
Open Universiteit in the Netherlands

First edition 2014

Operating Systems & Virtualisation

Introduction *1*

Core of Study *1*

1 Energy Reduction via the Operating System *1*
1.1 Power Management: aTtrade-Off *1*
1.2 ACPI: the Power Steering Standard *2*
1.2.1 Device States *4*
1.2.2 Understanding ACPI: Processors *4*
1.3 Beyond ACPI: Hardware Power Management *5*
1.3.1 Separate Power Planes *5*
1.3.2 Clock Gating at the CPU Block Level *6*
1.3.3 Power Gating *6*
2 Energy Reduction by Consolidation and Virtualisation of Servers *7*
2.1 Server Virtualisation: Introduction *7*
2.2 The role of Server Virtualisation: Dynamic Resource Scheduling in a Cluster *8*
3 IT Equipment Power Saving Strategy *9*

Summary *10*

Literature *11*

Self-Assesment *11*

Model Answers *12*

1 Answers to Reflection Questions *12*
2 Answers to Self-Assesment Questions *12*

Chapter 4

Operating Systems & Virtualisation

Johan De Gelas
University College of West Flanders

INTRODUCTION

In the preceding chapter we discussed the different models and components of IT equipment with respect to energy saving. In this chapter we take a look *inside* the equipment: how can power management and consolidation of servers contribute to decrease the energy consumption and what is the best strategy?
Energy usage of IT can be reduced by influencing the operating system of a computer or server, or by server virtualisation.

LEARNING OBJECTIVES
After you studied this chapter we expect that you are able to
- understand the basic workings of OS (ACPI) power management
- understand how to configure power management
- understand which features of virtualisation software influence power consumption
- reduce power inside the rack without breaking SLAs
- have an understanding of the impact of virtualisation on the energy efficiency of the IT equipment.

Study hints
The purpose of this chapter is to understand how power usage of IT equipment is controlled.
The workload is 6 hours.

CORE OF STUDY

1 Energy Reduction via the Operating System

The energy of the operating system can be reduced by influencing the ACPI system states or by hardware power management.

1.1 POWER MANAGEMENT: A TRADE-OFF

Your power efficiency efforts will be a trade-off between availability, performance and power.

Performance per watt

Total energy consumption and *Performance per watt* are important metrics for IT equipment energy efficiency. Performance per watt is the rate of computation that can be delivered by a computer for every watt of power consumed. However, power efficiency of IT equipment is not necessarily a high priority for most datacenters. The most important priorities for the datacenter administrator are typically:
1 Availability, the amount of time that users can interact with the system.
2 Response time, the waiting time users perceive.

These two important demands from the customers can quickly clash with the need to keep the energy consumption, carbon footprint and ultimately the power bills low.

REFLECTION 1

What is the ultimate goal of a datacenter? How does that effect power management?

A clear illustration of this, is the fact that servers based upon the (ultra) low energy Atom and ARM CPUs are currently relegated to a niche market. The reason is that, in most applications those servers can't offer low enough response times.

So you first want to attain a certain level of performance and/or availability. In some cases these levels will be contractually agreed upon, in the form of Service Level Agreement (SLA). So your power efficiency efforts will be a *trade-off between availability, performance and power*.

Trade-off between availability, performance and power

Most professionals understand the concept of availability very well, but are less concerned about response times. But if your customers get frustrated with the high response times they *will* quit. Case closed. And customers are easily frustrated. "Would users prefer 10 search results in 0.4 seconds or 25 results in 0.9 seconds?" That is a question Google asked (Stross, 2008). They found out to their surprise that a significant number of users got bored and moved on if they had to wait 0.9 seconds.

Not everyone has an application like Google, but most applications now run inside a virtualized machine and share the hardware resources with tens, sometimes hundreds of machines. Extra performance and RAM space is turned into more servers per physical server, or business efficiency. So it is very important not to forget how demanding we all are as customers when we are browsing and searching.

1.2 ACPI: THE POWER STEERING STANDARD

Advanced Configuration and Power Interface ((ACPI)

Power management in modern IT equipment starts with *ACPI, the Advanced Configuration and Power Interface*. In 1996, the three most influential companies in the PC world (Intel, HP, and Microsoft) together with Toshiba and Phoenix standardized power management by presenting the ACPI Specification.

ACPI defines which registers a piece of hardware should have available, and what information the BIOS/firmware should offer: these are the red pieces in Figure 1.

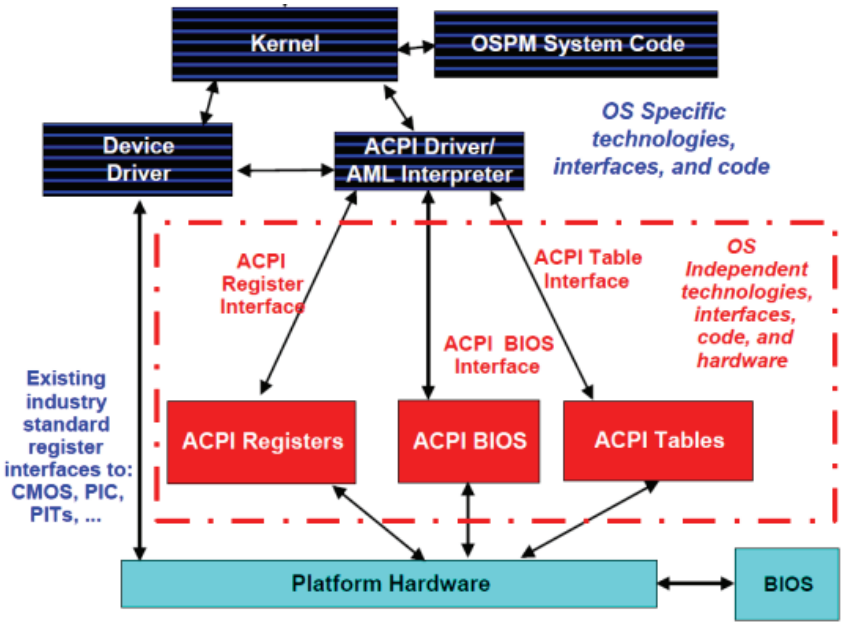


FIGURE 1 ACPI overview

The most important information can be found in the ACPI tables, which describe the capabilities of the different devices of the platform. So an error in the firmware can result in a serious increase in energy consumption.
Once the kernel has read and interpreted them, the role of the BIOS is over. This is in sharp contrast with the power management (APM) systems that we used throughout the 80s and 90s, where for example CPU power management was completely controlled by the BIOS.
The basic idea behind ACPI based power management is that unused/less used devices should be put into lower *power states*. You can even place the entire system in a low-power state (sleeping state) when possible.

Power states

ACPI system states

The *ACPI system states* are probably the best known ACPI states:

S0 – *S0* Working

S1 – *S1* processor idle and in low power state but still getting power, RAM powered

S2 – *S2* Processor in a deep sleep, RAM powered, most devices in lower power

S3 – *S3* CPU in a deep sleep, RAM still getting power, devices in the lowest power states, also known as 'Standby'

S4 – *S4* RAM no longer powered, disk contains an image of the RAM contents, also known as 'Hibernate'

S5 – *S5* is the soft power off

We will describe the ACPI system states using the most popular implementations; the standards are actually a bit vague or flexible if you like. You can find more details in the latest ACPI specification [1].
The steering of the ACPI based power management happens inside the power management component of the operating system kernel. The kernel power manager handles the devices' power policy, calculates and commands the required processor power state transitions, and so on. Of course, a kernel component does not have to know every specific detail of each different device.
To better understand how ACPI works, you have to understand how processors regulate power. Processor states are the most complex. Before we explain them, let us first take a look at the simpler more general device states which are applicable to all devices including USB, network controllers and so on.

1.2.1 *Device States*

– D0 is the operating state.
– D1 and D2 are intermediate power-states whose definition varies by device.
– D3 Off has the device powered off and unresponsive to its bus.
CPUs are a special kind of 'device' as we will see further.

1.2.2 *Understanding ACPI: Processors*

There are two processor states:
1 P-states
2 C-states.

P-states
Clock speed
Voltage

P-states are described as performance states; each P-state corresponds with a certain *clock speed* and *voltage*. P-states could also be called processing states: contrary to C-states, a core in a P-state is actively processing instructions.
With the exception of C0, C-states are sleep/idle states: there is no processing whatsoever. What follows is a quick overview of the ACPI standard C-states. The ACPI standard only defines four CPU pwer states from C0 to C3:
1 C0 is the state where the P-state transitions happen: the CPU is processing.
2 C1 halts the CPU. There is no processing, but the CPU's own hardware management determines whether there will be any significant power savings. All ACPI compliant CPUs must have a C1 state.
3 C2 is optional, also known as 'stop clock'. While most CPUs stop 'a few' clock signals in C1, most clocks are stopped in C2.
4 C3 is also known as 'sleep', or completely stop all clocks in the CPU.

The actual result of each ACPI C-state is not defined. It depends on the power management hardware that is available on the platform and the CPU. Most CPUs of the past 10 years support an Enhanced C1 state (C1E) which is not described in the ACPI states. This C1E is entered automatically if the CPU stays in C1 for a while.

[1] http://www.acpi.info/spec.htm

Modern CPUs, produced in 2008 or later, will not only stop the clock in C3, but also move to 'deeper C4/C5/C6/C7 sleep states'. The C1E, C4, C5, and C6 states are only known to the hardware; the operating system sees them as ACPI C2 or C3.

1.3 BEYOND ACPI: HARDWARE POWER MANAGEMENT

Build-in hardware power management

So it is clear that processors go beyond the ACPI specifications by using *build-in hardware power management*. That is not surprising. First of all, when a server is under load, most of the power is consumed by the CPU. Secondly, hardware can regulate power in a few tens of nanoseconds, while operating systems need milliseconds. As a result it pays off to invest in special power management hardware.

Intel x86 CPUs

The most popular CPUs in the datacenter are based upon Intel's x86 ISA. To keep the scope manageable, we will focus on the *Intel x86 CPUs*.

Since the introduction of the PIII mobile at the end of the past century, both CPUs have been using Dynamic Frequency and Voltage Scaling (DFVS). DFVS has been marketed as 'PowerNow!', 'SpeedStep' and many other names. In a multi-core CPU, the simplest implementation is that all the cores will clock at the clock speed of highest loaded core. A certain clock speed requires a corresponding core voltage, so all cores also use the same voltage.

As we have seen, the different clockspeed/voltage steps are called P-states or performance states. Each P-state corresponds thus with a certain clock speed and voltage.

More advanced implementations of DVFS have been around since 2006-2007. For example, the AMD K10h family (aka 'Barcelona') in 2007, introduced Dynamic Frequency Scaling per Core. Each core runs at its own clock.

The effect of this on performance/watt is not a complete success story: power is linear with frequency, and some OS schedulers will always try to 'load balance' across the cores to avoid having one core get hot (which increases leakage, see further). As a result the power savings are relatively small, and the lag in transitioning from one P-state to another reduces performance.

AMD Opterons typically support 4-5 P-states. The Opteron 'Shanghai' 2389 in this test supports 2.9, 2.3, 1.7 and 0.8 GHz. The six-core Opteron 2435 supports 2.6, 2.1, 1.7, 1.4 and 0.8 GHz. [2]

1.3.1 *Separate Power Planes*

Systems On a Chip (SoCs)

Uncore

The current server CPUs are more than just processors: they are *Systems On a Chip (SoCs)*. They are typically divided in cores – the actual processing cores – and uncore. *Uncore* comprises the I/O interfaces (PCIe for example), the memory controller, interconnects, the last level cache.

Separate power planes for core and uncore provide several benefits. The first benefit is that the cores can go to a sleep (C-state) while the memory controller is still working for another external device (e.g. via DMA). Another advantage is that the CPU/SoC is able to run the uncore out of sync with the cores. In other words, the uncore can run at different (mostly lower) voltage and frequencies than the cores. This lowers power significantly, while performance only decreases slightly. Overall, performance/watt is clearly increased.

1.3.2 *Clock Gating at the CPU Block Level*

The clock signal is a synchronization signal that is propagated throughout the entire chip if necessary. However, in many applications not all components of the chip need to be active. For example, contrary to HPC code, most server applications do not perform a lot of floating point calculations. Therefore, it is interesting to disable the clock signal to the *Floating Point Unit, FPU*. Most modern CPUs are able to block the clock signals to some components if they are not used.

Floating Point Unit, FPU

A good illustration of this is that the highest power numbers are measured by floating point intensive benchmarks like LINPAC; typical server benchmarks based on databases or web servers do not even come close. LINPAC needs 20-25% more power than integer based benchmarks, despite the fact that in both situations the operating system reports utilization to be 100%.

The reason is that floating point intensive benchmarks use almost every component of the CPU, while most integer intensive code leaves the FPU alone.

Clock gating reduces power by 20 to 40% according to some publications (Biegel, 2000).

1.3.3 *Power Gating*

It is important to note that a processor or any semi-conductor component for that matter consumes power in two ways: static and dynamic. The *dynamic* part is the result of the transistors switching while processing data. The static part is also called leakage. Power Leakage happens as a part of the current, which is supposed to make transistors switch leaks away in the substrate and finally in the ground. There are several leakage currents, but the two most important ones are the *gate oxide tunnelling* current and *sub-threshold leakage* (Hillman, 2004).

Dynamic
Static

Gate oxide tunnelling

Sub-threshold leakage

Clock gating only reduces the dynamic power. Therefore, most modern server CPU (for example, the Xeon 5500 and later) go one step further: power gating. Power gating shuts off current to all blocks that can go 'to sleep'. As a result, power gating reduces both dynamic and leakage power.

In summary, modern CPUs have separate hardware circuitry that goes beyond the ACPI standard. By using

– Using deeper C-states controlled by CPU internal circuitry
– Separate power planes
– Clock gating
– Power gating.

Modern CPUs achieve significant power savings.

2 Energy Reduction by Consolidation and Virtualisation of Servers

In Chapter 3 we discussed that a large server is more energy efficient than a small server and that higher utilization of a server means more performance per watt. Therefore tasks for servers have to be combined. Consolidation means to combine things. In the case of server consolidation, many small physical servers are replaced by one larger physical server to increase the utilization of costly hardware resources such as CPU. When the small physical servers use different operating systems, it is still possible to run them on one server. This is called server virtualization. The large servers is the host for several small virtual machines.

2.1 SERVER VIRTUALISATION: INTRODUCTION

Traditional 'native' Operating systems already offer some form of 'virtualisation'. A modern OS already virtualizes quite a few resources: memory, disks, and CPUs for example. For example, while there may only be only 4GB RAM in a 32-bit server, each of the tens of running applications is given the illusion that they can use the full 2GB (or 3GB) user-mode address space. There might only be three disks in a RAID-5 array available, but as you have created 10 volumes (or LUNs), it appears as if there are 10 disks in the machine. Although there might only be two CPUs in the server, you get the impression that five actively running applications are all working in parallel at full speed.

However, 'supervisor' operating systems isolate the applications weakly by giving each process a well-defined memory space, separating data from instructions. At the same time, processes share the same files, may have access to some shared memory, and share the same OS configuration. In many situations, this kind of isolation was and is not sufficient. One process that takes up 100% of the CPU time may slow the other applications to snail speed for example, despite the fact that modern OS use preemptive multitasking. In case of pure hardware virtualisation, you will have completely separate virtual servers with their own OS (guest OS), and communication is only possible via a virtual network.

Since 2005, x86 vendors have been increasing the core count of their CPUs quickly. At the same time, the memory capacity has increased exponentially. A typical server in 2005 could support about 32 GB and up to 4 cores. Anno 2013, most 2-socket systems could support up to 768 GB, and core counts of 8 to 16 cores per socket are common.

The result is that non-virtualized servers are only utilized at a fraction of their total computing capacity. Running several virtual machines (VMs) on a single physical host improved hardware utilization. A host machine running at 60% load instead of 10% consumes a few tens of percent more, but considerably less than the many servers it can replace. The reason is the inability of servers to be 'Energy Proportional', i.e. there is no linear relationship between the load and the energy consumption. Especially at *low utilization*, servers are very *energy inefficient*. The processors are among the most energy proportional components, but the rest of the server consumes quite a bit of energy at low loads (Feng, 2013).

Low utilization
Energy inefficient

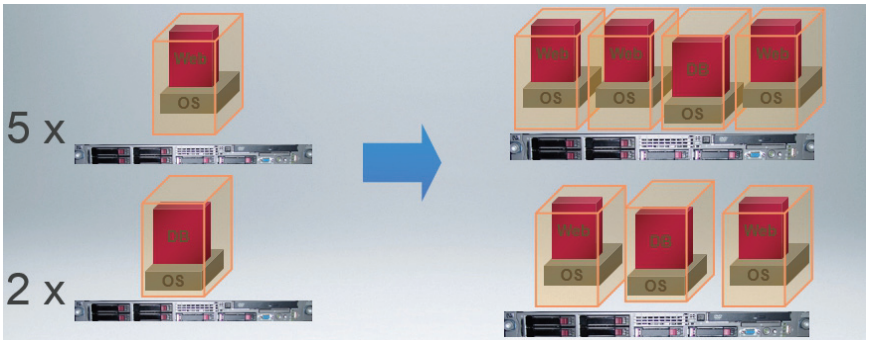


FIGURE 2 Consolidation: from one application per server to one virtual machine (VM) per application

Instance

Muliple instances on top of one physical machine

Consolidation

An *instance* is a combination of an operating system and one or more applications (for example, a database software). In many cases, an instance represents one software service. The 'strong isolation' feature of a 'Hypervisor' makes it possible to run *multiple instances on top of one physical machine*, see Figure 2. This is called *consolidation*, and consolidation can result in enormous energy savings.

Anywhere from 5 to 100 (older) physical servers running one instance can be consolidated into 1 (or 2 for availability) modern host machine. One modern host server that is utilized at a higher level will consume a certain percentage (a few tens) more than a machine that is less utilized, but in total that one host will replace 5 to 100 hosts. As a result, the total energy bill of virtual host can be a fraction of running all instances on a physical machine.

2.2 THE ROLE OF SERVER VIRTUALISATION: DYNAMIC RESOURCE SCHEDULING IN A CLUSTER

More advanced VM capabilities such as cloning, template-based deployment and live migration made managing VMs easier. Combining these new technologies with a cluster of servers paved the way for new usage models, with the side effect of saving even more energy.

Cluster schedulers

Cluster schedulers, such as VMware's Distributed Resource Scheduler (DRS), manages the allocation of physical resources to a certain number of virtual machines deployed in a cluster of servers, all running a hypervisor. By intelligently mapping VMs to a cluster of servers instead of one host, it is possible to get much better energy proportionality than what is possible with one host. In case of low load, intelligent cluster schedulers map VMs to other hosts and shuts unnecessary servers down (ACPI S5). If hardware resources demand goes beyond a certain threshold, the cluster scheduler wakes them up again if necessary.

Extensive evaluations show that clusters with non-trivial periods of lowered demand, significant power savings are possible (Ajay Gulati, 2012).

Cluster schedulers use Wake-on-LAN or IPMI to wake up servers when amount of CPU power or memory capacity is required.

3 IT Equipment Power Saving Strategy

Power savings inside the rack

The general strategy for *power savings inside the rack* of your datacenter should now be clear:
1 Make sure that your hardware makes the best use of the power saving techniques
2 Use virtualisation, consolidation and cluster techniques as much as you can.

Practical steps

How do you turn this strategy into *practical steps*? First of all, ACPI starts with the right firmware settings. Firmware is software that is programmed into hardware. Firmware updates disrupt the current operations (as the servers has to be shutdown) but if the updates contain important ACPI update, there is good chance they are worth the trouble.

Secondly you should enable all C-state capabilities in your firmware. As you have learned, they are essential for power saving.

Natively

Thirdly, the datacenter should be scanned for '*natively*' (i.e. running on a physical machine) running applications and a maximum of servers should be virtualized. Fourthly, you should try to make good use of some cluster scheduling software. For example, if you run VMware vSphere, you check out the guide to using Distributed Power Management.[2]

Lastly, if an application needs to run on a 'native OS' for whatever reason, we should try to make the best use of power manager inside the OS.
For applications that are highly latency sensitive such as databases, Operating system driven DVFS should not be used.
Transitioning from one P-state to another takes a while, especially if you scale up. Each transition wastes energy and decreases the performance of the core.

REFLECTION 2
Why does a P-state transition waste energy and performance? When will it be beneficial and when not?

The OS power manager has to predict whether or not the process will need more processing power soon or not. As a result the OS transitions are a lot slower than the hardware. Only when the application is rather predictable, P-state transitions are almost always saving power. Examples are most HPC applications: they run at a certain load for a long time and when the result is calculated, the load is gone. As a result it is very easy for the OS to predict the load and thus save energy: few P-states transitions are necessary.

C-states rarely have this problem: although the power manager of the OS regulates them, the hardware circuits of modern CPUs will override them if necessary. Modern CPUs that use power gating to make cores to go to sleep can thus save much more power than what is possible with DVFS.

[2] http://www.vmware.com/files/pdf/Distributed-Power-Management-vSphere.pdf

An example is presented in Figure 3. On 64-bit Windows servers at least two power configurations, called 'power plans', are available.

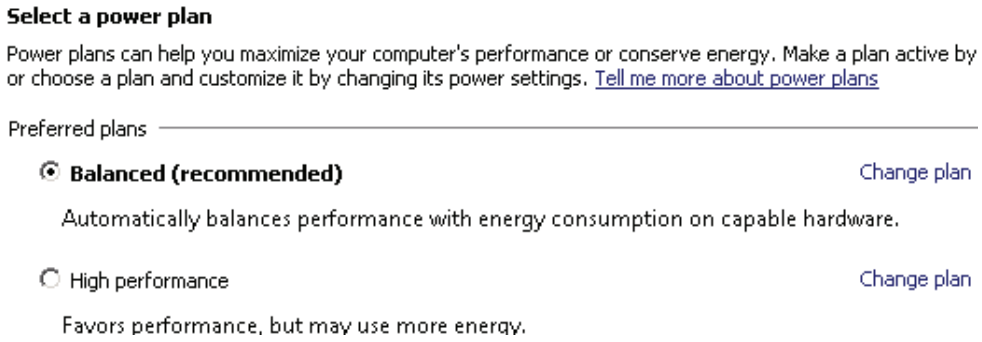


FIGURE 3 Windows ACPI based 'Power Plans'

High performance disables P-states but not C-states. So depending on the your performance requirements, you will need to set up the 'power plan'. As both power plans are using C-states, do not expect the power savings in balanced to be very high.

SUMMARY

ACPI is the standard technology that is necessary for power management. Through the use of ACPI compliant hardware, ACPI tables in the firmware that describe the capabilities of that hardware and an ACPI aware power manager in the OS kernel, it is possible to save a lot of energy when your software is not running at maximum performance.

It is good to understand the capabilities of the hardware and the limitations of the power manager (software) of the hardware. The OS software has to predict the necessary performance and if the predictions do not correspond to reality, power can be wasted and or performance can be lost.

Therefore, a good understanding and thus monitoring of your applications is necessary.

Consolidation of hardware servers into virtual machines on powerful hosts can be an excellent way to reduce the energy consumption. In a later stadium, energy proportionality can be greatly improved by using a form of cluster management/scheduling software.

Literature

Ajay Gulati, A. H. (2012). *VMware Distributed Resource Management: Design, Implementation, and Lessons Learned.* VMware.

Biegel, F. E. (2000). Power Reduction through RTL Clock Gating. *SNUG (Synopsis User Group) Conference.* San Jose: SNUG.

Feng, B. S.-c. (2013). Towards Energy-Proportional Computing for Enterprise-Class Server Workloads. *International Conference on Performance Engineering (ICPE);.* Prague: Virginia Tech.

Hillman, D. (2004). Implementing Power Management IP forDynamic and Static Power Reduction in Configurable Microprocessors using the Galaxy Design Platform at 130nm. *SNUG (Synopsis User Group).* San Jose: SNUG.

Stross, R. (2008). *'Planet Google': One Company's Audacious Plan to Organize Everything.* New York: Free Press.

SELF–ASSESSMENT

1 What does ACPI mean?

2 What is the relation between ACPI, the firmware and the OS?

3 In Windows we have the settings 'sleep' and 'hibernate' for your pc. To what ACPI state do they correspond?

4 Why are P-states much less effective than C-states?

5 Why is consolidation through virtualisation so beneficial for power saving?

6 How does virtualisation cluster scheduling improve energy proportionality?

MODEL ANSWERS

1 Answers to Reflection Questions

1 The ultimate goal of a data centre is the same as for the enterprise as a whole: serve as many (internal or external) customers as possible with the best service at the lowest cost. In case of most IT customers, the best service will translate into the necessary availability and being able to serve up request with a response time that users find acceptable.

Attaining a certain level of performance and availability is the first priority. At that point you want the lowest power consumption possible, so it is power efficiency at a certain performance/availability level that you are after, not the best performance/Watt ratio.

2 Suppose that the OS decides that the CPU can clock down to a lower P-state. Just a few milliseconds later, a running process requires a lot more performance. The result is that the voltage must be increased and this takes a while. During that time, the CPU is wasting more power than it should: processing is suspended for a small time and the clock speed cannot increase unless the higher voltage is reached and is stable enough.

If this scenario is repeated a lot, the small power *savings* of going to a lower P-state will be overshadowed by the power *losses* of scaling quickly back up to a higher clock and voltage. It is important to understand that each voltage increase results in a small period where power is wasted without any processing happening. The same problem is true for entering a C-state: enter it too quickly and performance is lowered as it takes some time to wake that core up again.

So a P-state transition will be *beneficial* if the power saving of running at lower P-state will be higher than the power wasted through the transition. P-state changes must not be too frequent and the transition time should be as low as possible.

2 Answers to Self-Assessment Questions

1 Advanced Configuration and Power Interface.

2 The ACPI tables inside the firmware, describe the capabilities of the different devices of the platform. Once the Power manager inside the kernel has read and interpreted them, the role of the BIOS is over. See also Figure 1.

3 System state S3 and S4.

4 P-states are mostly regulated by the OS which is much slower. The impact of P-states is limited to reducing the active power of the core. C-states are regulated by hardware mostly and reduce both the active and passive power to almost zero. So it is more beneficial to run one core at its highest P-state and put one in a deep sleep than to run two at a lower P-state.

5 Because running many servers at low load consumes much more power than a few servers at high(er) load. Servers are not energy proportional: a server running idle still consumes a lot of power.

6 By placing unnecessary servers in system state S5. In other words, servers are no long running idle, but are completely shut off.