


# Platform-based Power Management and Linux

Bdale Garbee, HP  
Naga Chumbalkar, HP

© Copyright 2010 Hewlett-Packard Development Company, L.P.





"Power saved is power  
generated"

Anonymous

# Agenda

- More upstream, less maintenance
- BIOS-based P-state management
- Platform-based Power Capping
- Processor Clocking Control (PCC) interface
- Collaboration between OS and platform

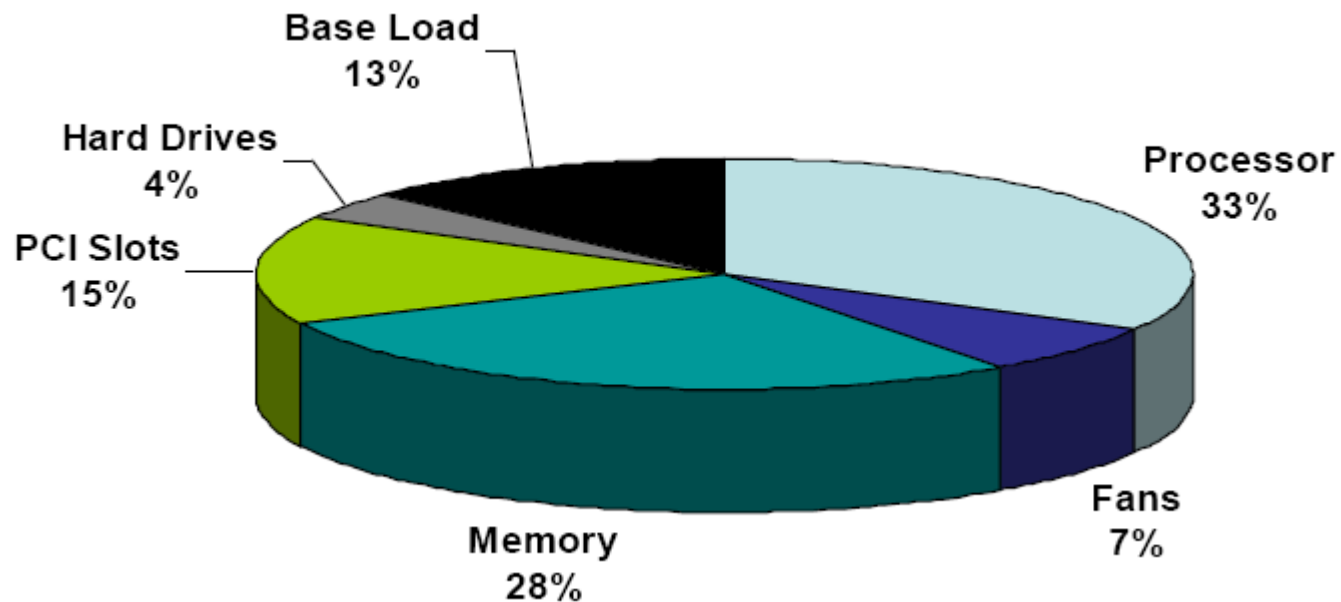


# More upstream, less maintenance

- Previously closed-source drivers are now upstream
- `hpilo` : Channel interface driver to communicate with the management processor
- `hpwdt` : Watchdog timer. Also sources NMI to inform user what HW component is at fault



# Typical Server Power Usage

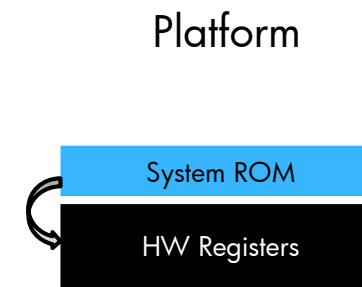
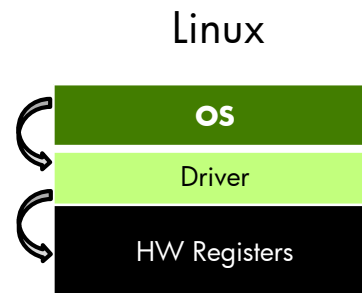


# Managing P-states

	P-State Technology	HP ProLiant
Intel	Enhanced SpeedStep	DL380 G6
AMD	PowerNow!	DL585 G6

## Intel Xeon X5570 Processor

P-State	Frequency
P0	2.92 GHz
P1	2.79 GHz
P2	2.66 GHz
P3	2.53 GHz
P4	2.39 GHz
P5	2.26 GHz
P6	2.13 GHz
P7	2.00 GHz
P8	1.86 GHz
P9	1.73 GHz
P10	1.60 GHz



# Power Capping

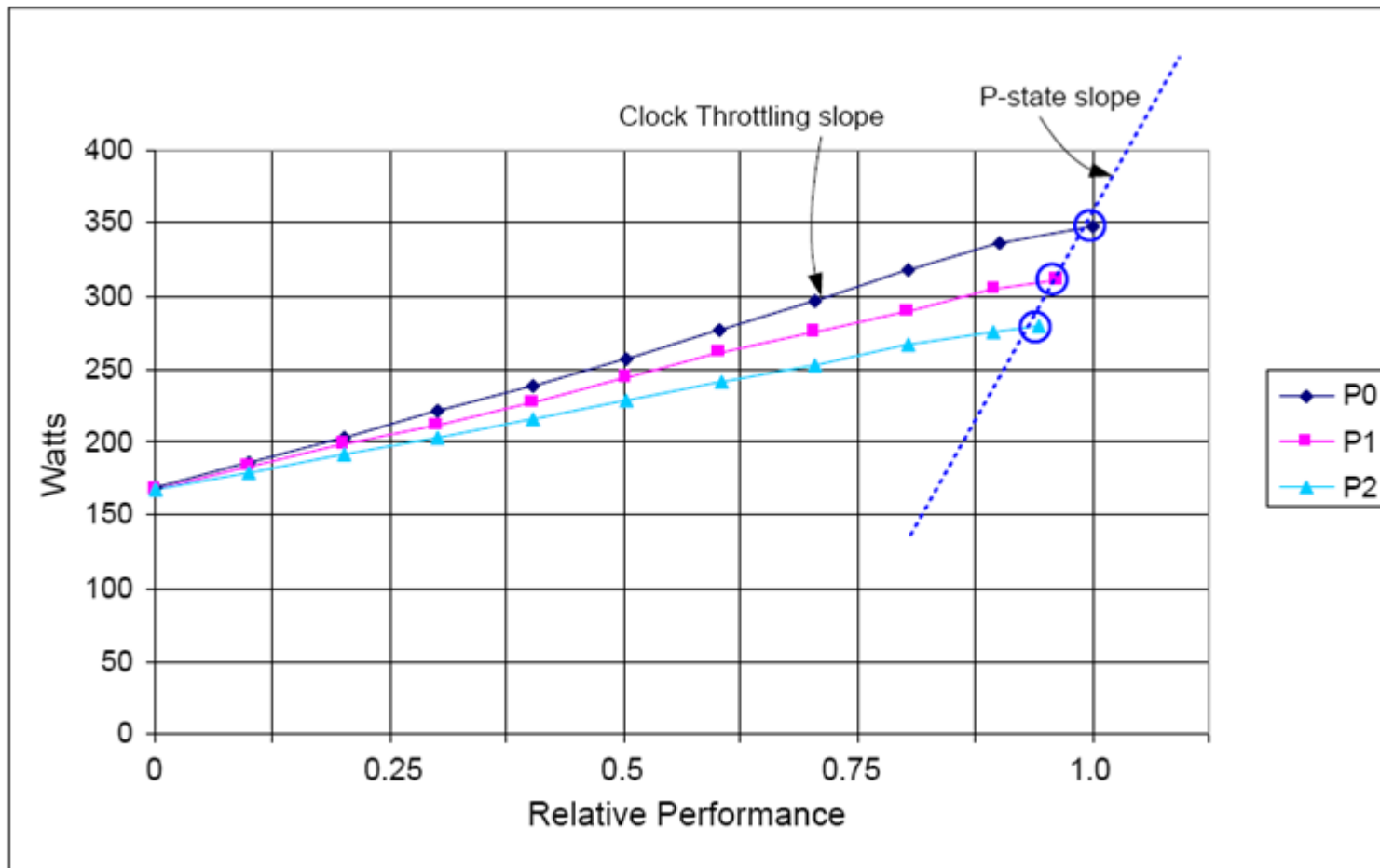
- Limit the power a server can use
- Implemented in the firmware, hardware
- Independent of the OS

## Data center advantages:

- Aids in provisioning cooling infrastructure
- Protects electrical circuit-breaker from tripping
- Reclaims trapped power (example on another slide)



# Power vs. Performance





# Power Capping - Example

HP ProLiant DL380 G5	Faceplate	"capped"	Total Power Consumed
6	400 W		2400 W
6		300 W	1800 W
8		300 W	2400 W

Platform guarantees that the power will be "capped" at 300W per server

Reclaim trapped power, and allocate two more servers!



# OS vs. Platform

## OS

Kernel

Single-server  
centric

Server-level  
capping



## Platform

System ROM, BMC/Enclosure FW

Enclosure-level view

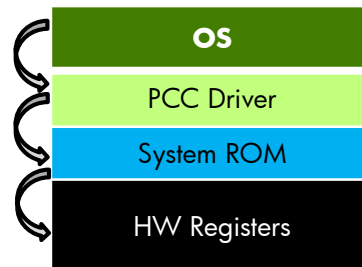
Group-level power capping



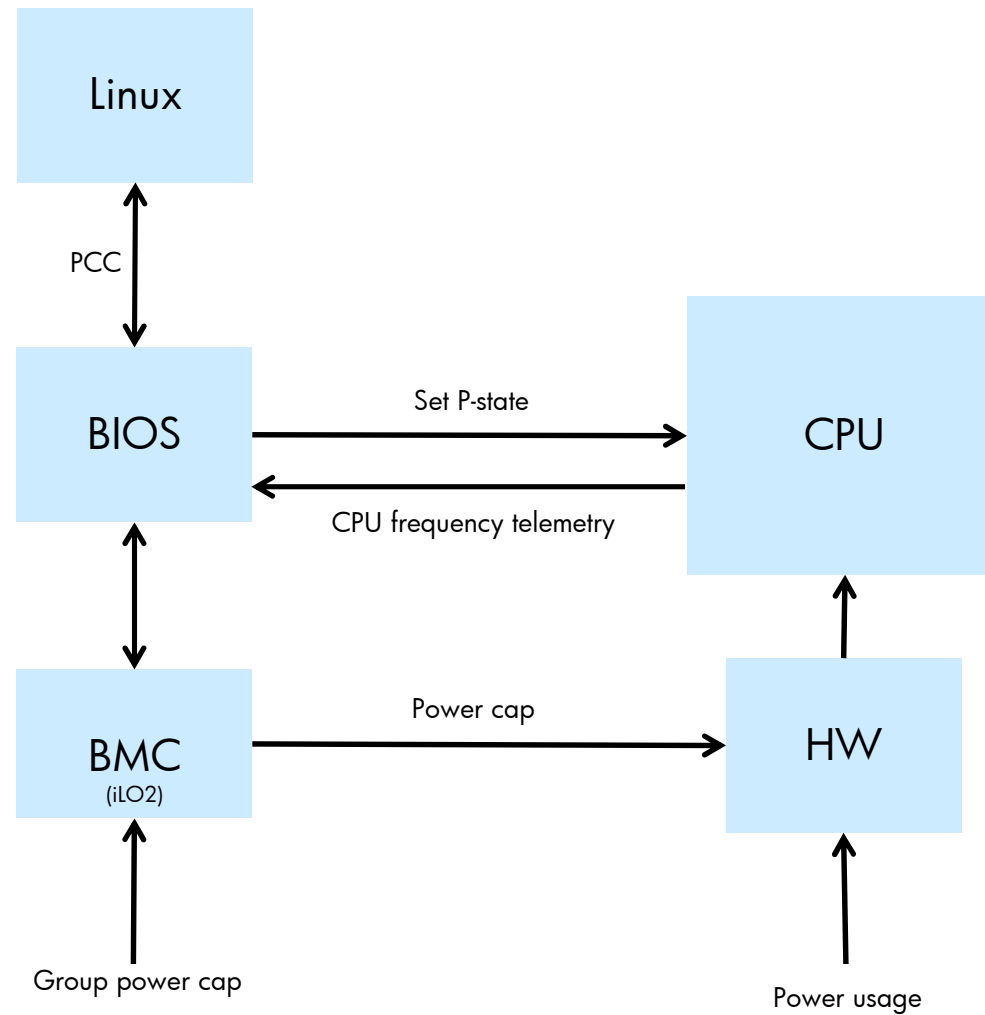
# Processor Clocking Control (PCC)

- Interface between BIOS and OS
- Feedback to the OS to become “capping” aware

Linux with PCC support



# Interaction between Linux and platform



# PCC usage

- PCC consists of a shared memory region that can be accessed by OS and BIOS
- OS computes required target frequency for a logical CPU based on workload
- PCC driver sends commands to the BIOS via input/output buffers to honor the OS's request
- Platform will achieve that frequency. However, if power budget conditions are in place, platform will achieve the best possible frequency

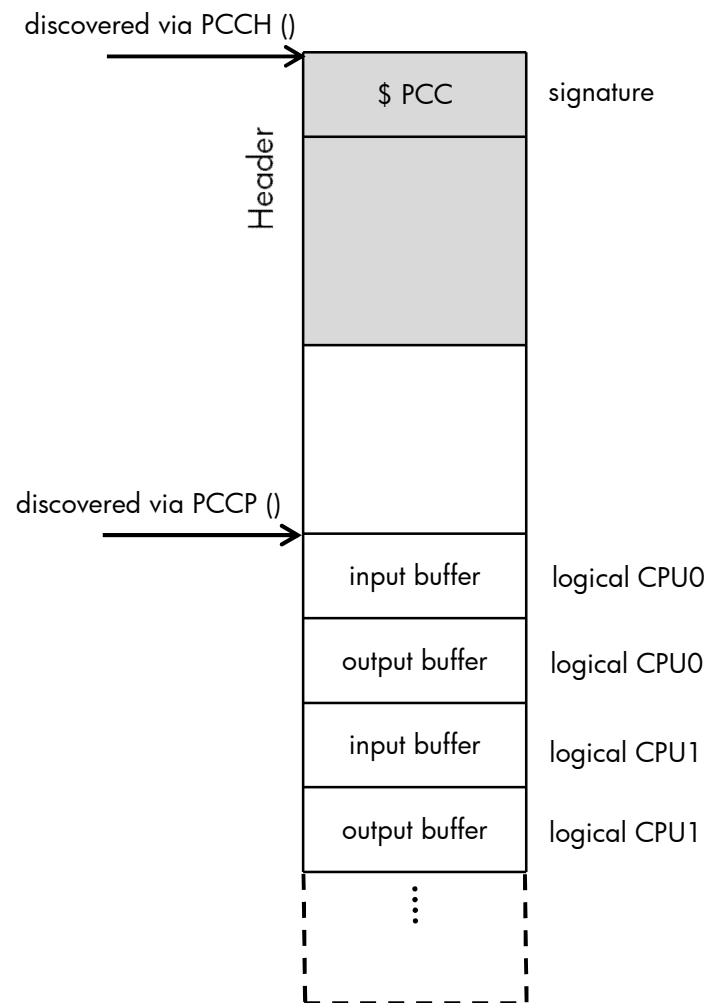


# PCC discovery

- BIOS will expose PCC only if OS is capable
- OS can evaluate ACPI PCCH() to discover shared memory area location
- OS can evaluate ACPI PCCP() to discover input/output buffer location for each logical CPU

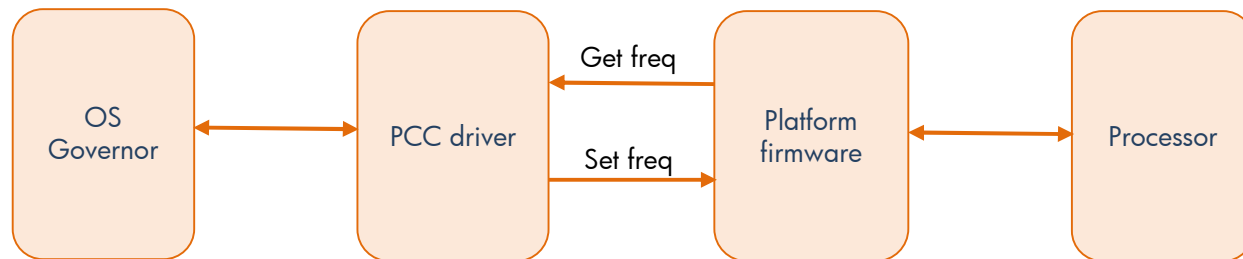


# Shared Memory Region



# PCC Commands

- Get Average Frequency
- Set Desired Frequency





# PCC is not proprietary

The Processor Clocking Control (PCC) Specification is open:

<http://www.acpica.org/download/Processor-Clocking-Control-v1p0.pdf>

PCC driver is open source, and is upstream (2.6.34)

<http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=arch/x86/kernel/cpu/cpufreq/pcc-cpufreq.c;h=ce7cde713e7174e8293b552b332f968946a46904;hb=250541fca717a5c9b0d3710e737b2ca32ebb6fbc>

Driver is not tied to any specific platform vendor

Driver will load on any platform that supports the PCC specification



# Collaborate and Innovate

PCC allows the platform to innovate  
(e.g.) power capping, enclosure-level group capping

Allows the OS to innovate  
(e.g.) better methods to react to changing workload,  
decide what CPU on what NUMA should be at what speed

No one entity (platform, OS) has a complete picture of the individual server and the data center.



Outcomes that matter.

