Developments in Linux/Intel Power Management

Len Brown
Principal Engineer
Intel Open Source Technology Center
len.brown@intel.com, lenb@kernel.org

14 October, 2018 - Nanjing, China

History

History

- 2003 *(Len Brown starts as Linux Kernel ACPI sub-system maintainer)*
  - Deploy ACPI in Linux
    - Enable fundamental system configuration
    - Start to enable “Operating System Directed Power Management (OSPM)
- 2018 *(Here we are today!)*
  - Support Government Energy Regulations
  - Support competing User Expectations
    - high-performance
    - Long battery-life
    - fanless-operation
    - Instant availability

Agenda

1. Suspend/Resume
2. Idle
3. Performance states (P-states)
4. Runtime Average Power Limiting (RAPL)
5. Power-on/boot

Suspend/Resume

Suspend/Resume Definition

1. Freeze Applications
2. Freeze Devices
3. Enable Wakeup Sources
4. Enter System Suspend State (eg. ACPI S3, or s2idle)
5. Detect Wakeup Event
6. Disable Wakeup Sources
7. Resume Devices
8. Resume Applications

Suspend/Resume Attributes

- Reliability
  - Depends highly on devices and drivers in system
- Performance
  - Latency as close to “instant-on” as possible
  - Suspend power as close to “power-off” as possible

Linux Suspend to mem/disk/freeze

```
# cat /sys/power/state
disk        Hibernate
mem         ACPI S3
freeze      s2idle

# cat /sys/power/mem_sleep
S2idle [deep]
# echo mem > /sys/power/state
```

S2idle will be DEFAULT on more systems in future!

sleepgraph.py

Project Home Page

https://01.org/suspendresume

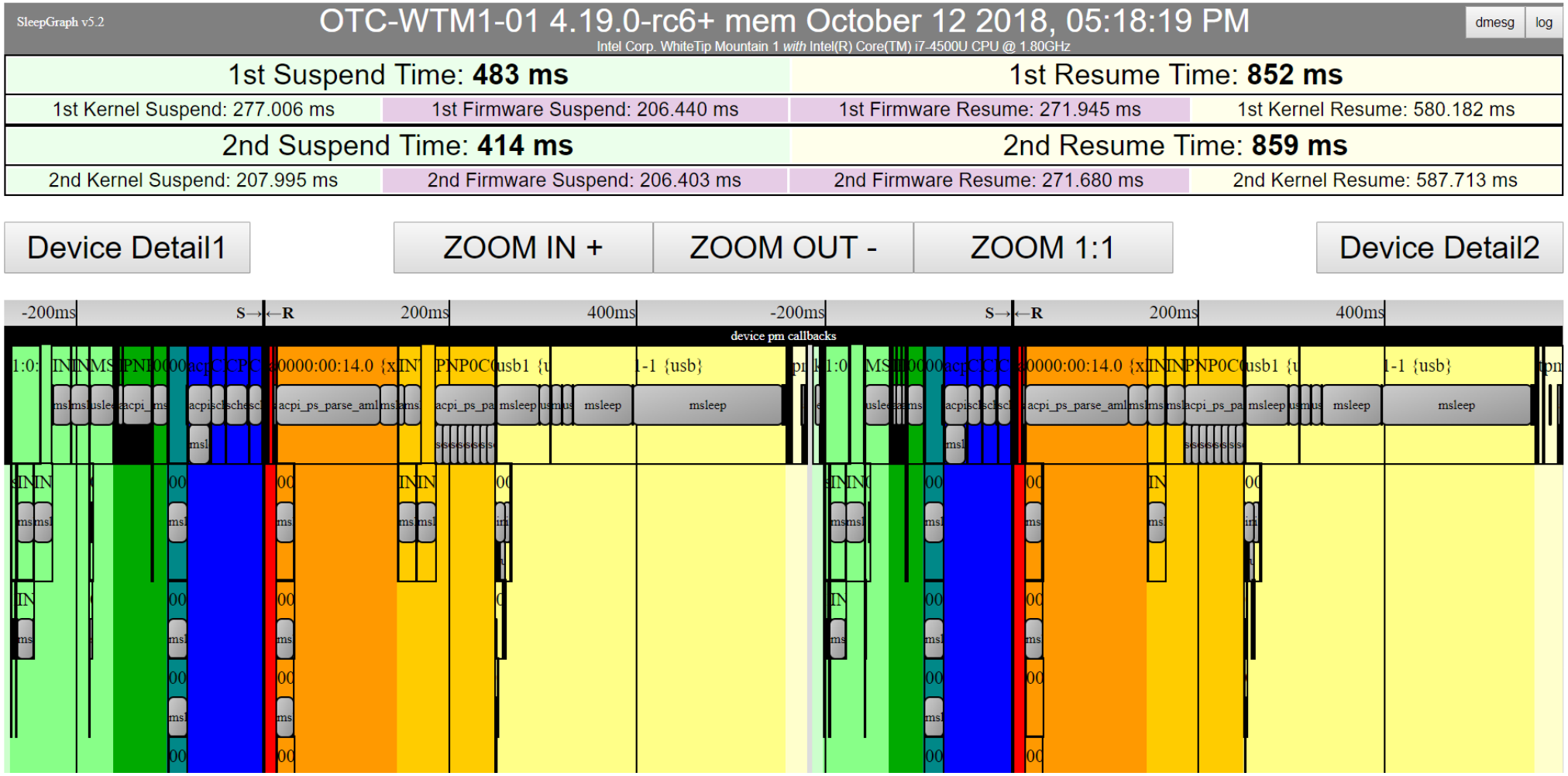
Latest Source

git://github.com/01org/suspendresume

Linux kernel source tree:

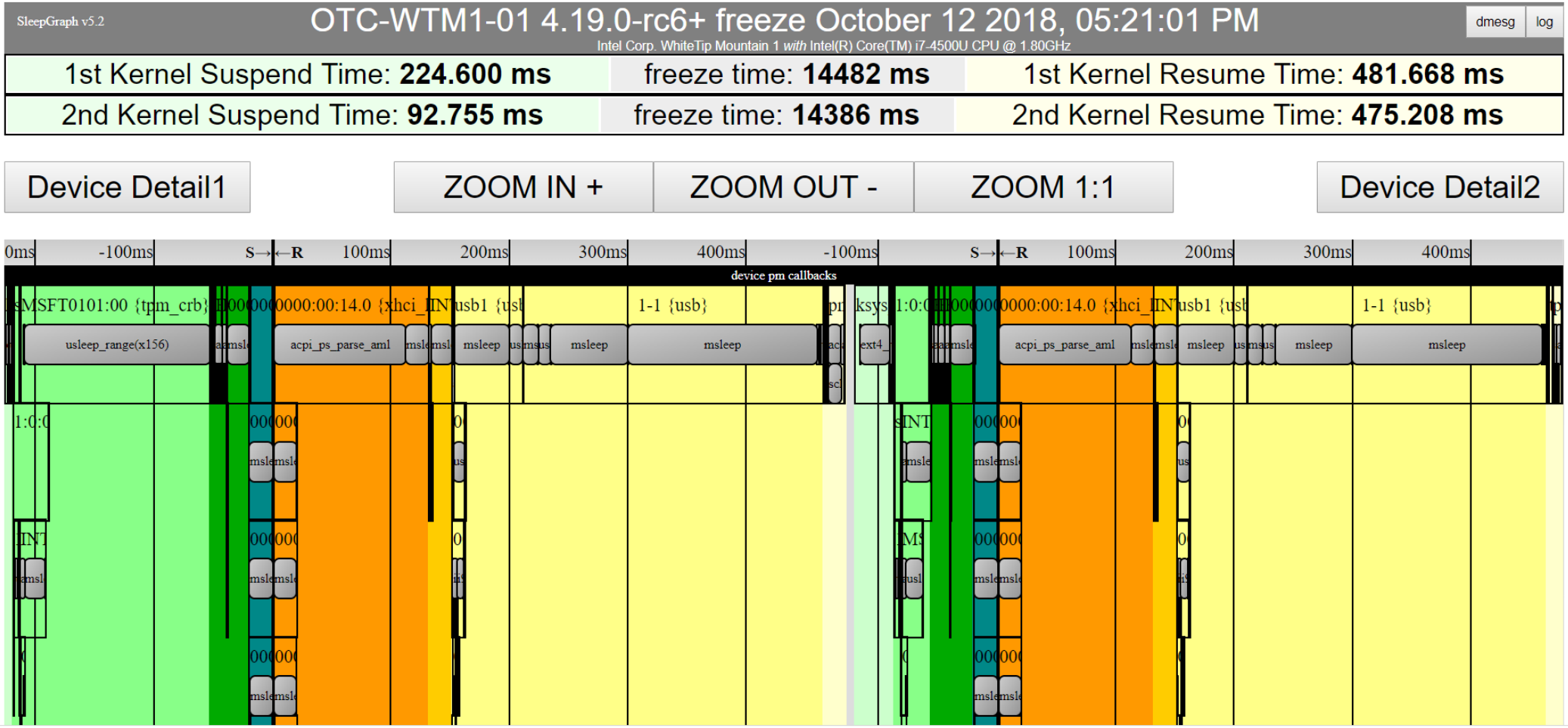
tools/power/pm-graph/

Sleepgraph example: mem (ACPI S3)



HTML HERE

Sleepgraph example: s2idle (x2)



HTML HERE

Low Power Idle System State

- *S2idle* is used to accesses system “Low Power Idle” state
- Prerequisites:
  - Processors in deepest idle state
  - Devices in OFF state
- Indicators

```
/sys/devices/system/cpu/cpuidle/
    low_power_idle_cpu_residency_us
    low_power_idle_system_residency_us system has LPI

# turbostat --show CPU%LPI,SYS%LPI
```

Suspend/resume reliability & endurance

Host	Mode	Results		Suspend Time	Resume Time	Worst Suspend Devices	Worst Resume Devices
otcpl-lenovo-yoga-13ISK2	mem	PASS HANG	2486/2510 (99.0%) 24/2510 (1.0%)	Max=1250.735 Med=559.965 Min=477.237	Max=5760.666 Med=1484.438 Min=1139.812	◦ 0000:00:02.0 {i915} (x2475) ◦ hdaudioC0D0 {snd_hda_codec_realtek} (x8) ◦ hdaudioC0D2 {snd_hda_codec_hdmi} (x2) ◦ phy0 {ieee80211} (x1)	◦ AX88772 [1-2] {usb} (x2183) ◦ 8087:0a2b [1-7] {usb} (x173) ◦ Lenovo EasyCamera [1-6] {usb} (x95) ◦ hdaudioC0D0 {snd_hda_codec_realtek} (x30) ◦ ITE8396:00 [i2c-ITE8396:00] {i2c_hid} (x4) ◦ hdaudioC0D2 {snd_hda_codec_hdmi} (x1)

Issues found	Count	html
[Firmware Bug]: TSC ADJUST differs: CPU0 0 --> -32376011. Restoring	1 times	html

Host	Mode	Results		Suspend Time	Resume Time	Worst Suspend Devices	Worst Resume Devices
otcpl-lenovo-yoga-13ISK2	freeze	PASS HANG FAIL(suspend)	2691/2696 (99.8%) 4/2696 (0.1%) 1/2696 (0.0%)	Max=1598.902 Med=458.608 Min=372.120	Max=3272.332 Med=725.360 Min=640.744	◦ 0000:00:02.0 {i915} (x2665) ◦ hdaudioC0D0 {snd_hda_codec_realtek} (x13) ◦ phy0 {ieee80211} (x6) ◦ 0000:03:00.0 {nvme} (x5) ◦ freeze_processes (x1)	◦ phy0 {ieee80211} (x2437) ◦ hdaudioC0D0 {snd_hda_codec_realtek} (x245) ◦ 0000:00:16.0 {mei_me} (x6) ◦ pm_notifier_call_chain (x3) ◦ ITE8396:00 [i2c-ITE8396:00] {i2c_hid} (x1)

Issues found	Count	html
WARNING: CPU: 3 PID: 7876 at drivers/net/wireless/intel/iwlwifi/mvm/scan.c:1786 iwl_mvm_rx_umac_scan_complete_notif+0x1d8/0x220 [iwlmvm]	6 times	html
WARNING: CPU: 2 PID: 1783 at drivers/net/wireless/intel/iwlwifi/mvm/mac80211.c:1253 __iwl_mvm_mac_stop+0x143/0x150 [iwlmvm]	3 times	html

HTML HERE

Idle


洛神花
金银花
25/50克
野生天麻
35/50克
雪菊
25/50克
三七花
48/50克
黑玛卡
98/50克
甘草
10/50克
野生
紫玛卡
68/50
米菊
25/50克
柠檬干
15/50克
黄芪
15/50克

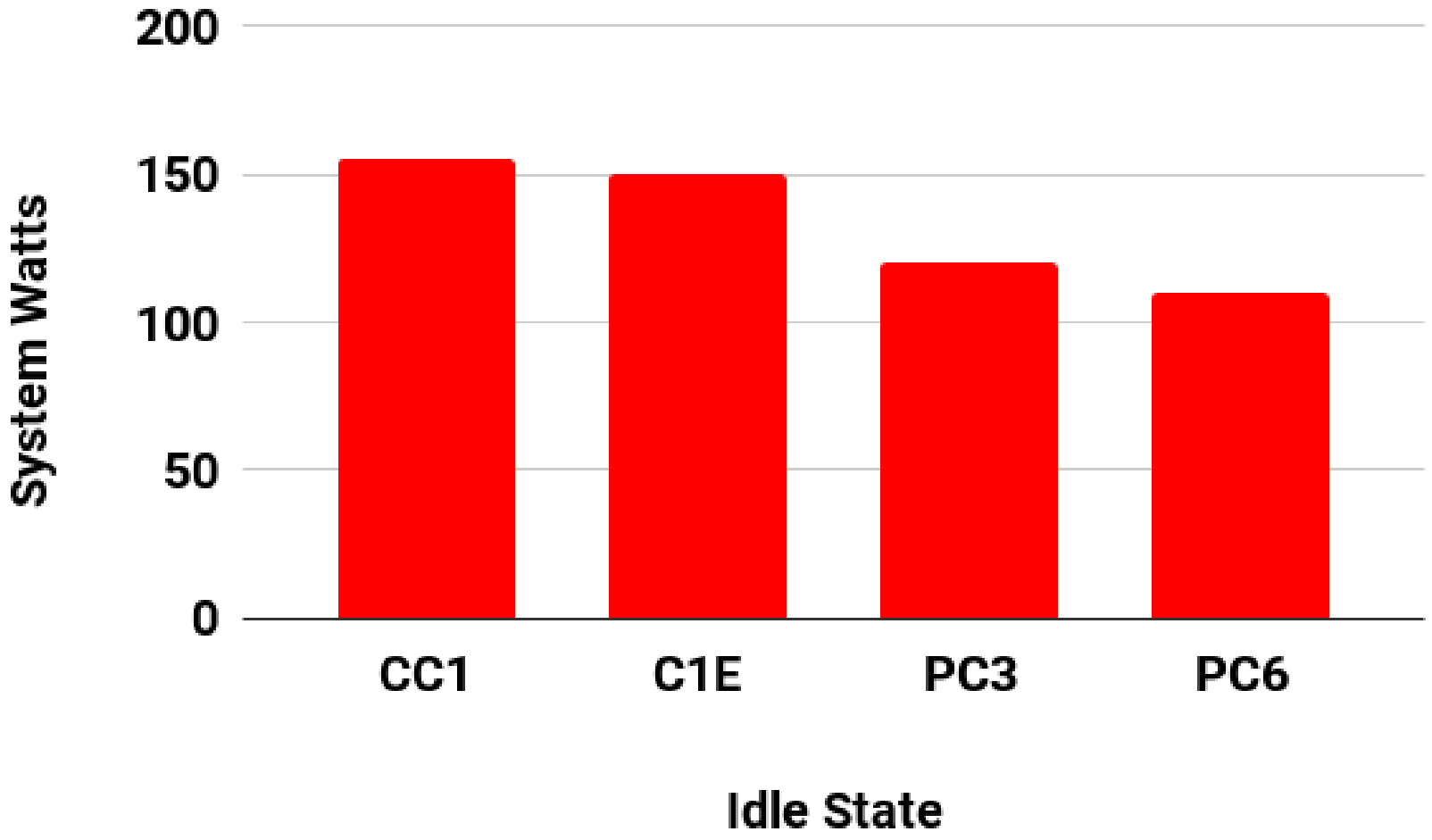
Idle Definition

1. Linux Scheduler has no runnable tasks
2. Linux Scheduler calls cpuidle sub-system
3. Cpuidle sub-system governor selects idle state
4. Cpuidle driver enters idle state
5. No instructions execute during idle (frequency = 0 MHz)
6. Break out of idle event occurs (eg. HW interrupt)
7. Return to scheduler for look for work to do

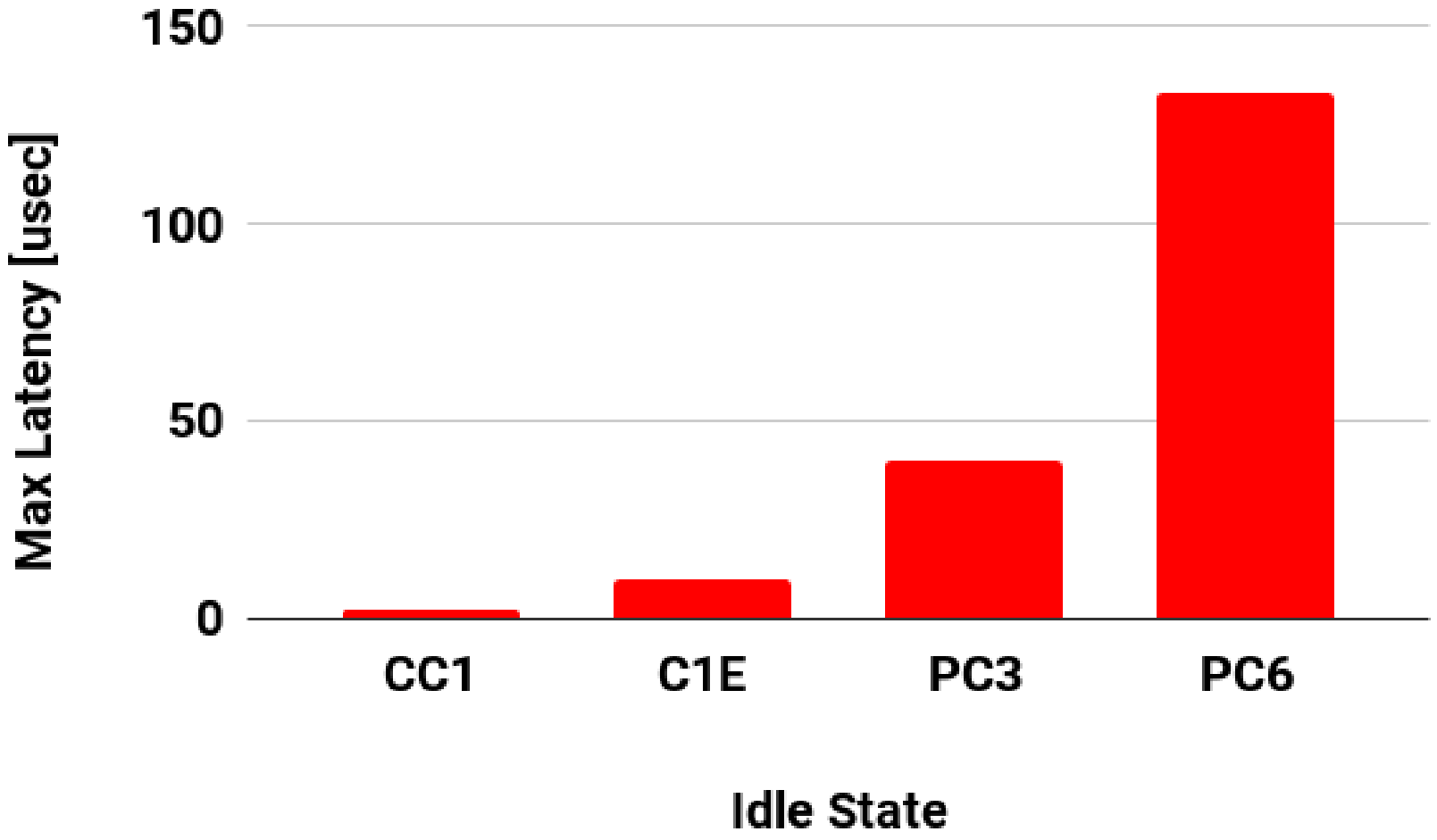
Linux cpuidle sub-system

Governor:	menu	ladder
Driver:	intel_idle	acpi_idle

Example Benefit of Deeper HW Idle States



Example Cost of Deeper HW Idle States



Guarantee Low Latency by limiting C-states

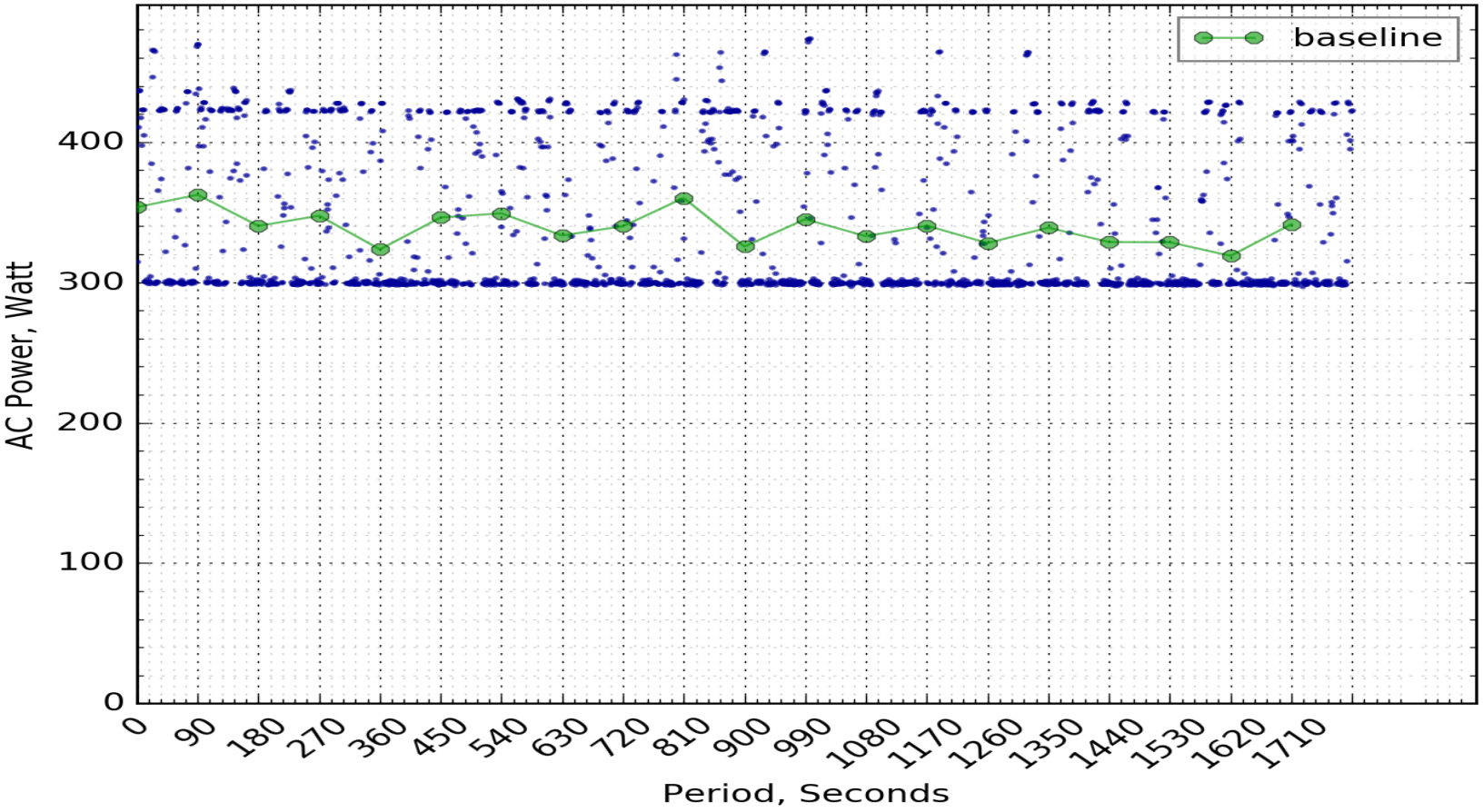
1. PM_QOS
   See pm_qos_interface.txt
2. Command-line
   Native: “intel_idle.max_cstate=N”
   ACPI: “Processor.max_cstate=N”
3. Sysfs
   # echo 1 > /sys/devices/system/cpu/cpu0/cpuidle/state3/disable
4. BIOS SETUP
   BIOS can limit Package C-states
   BIOS changes to ACPI idle tables are ignored by Linux intel_idle driver

** n.b. disabling C-states may hurt max turbo performance*

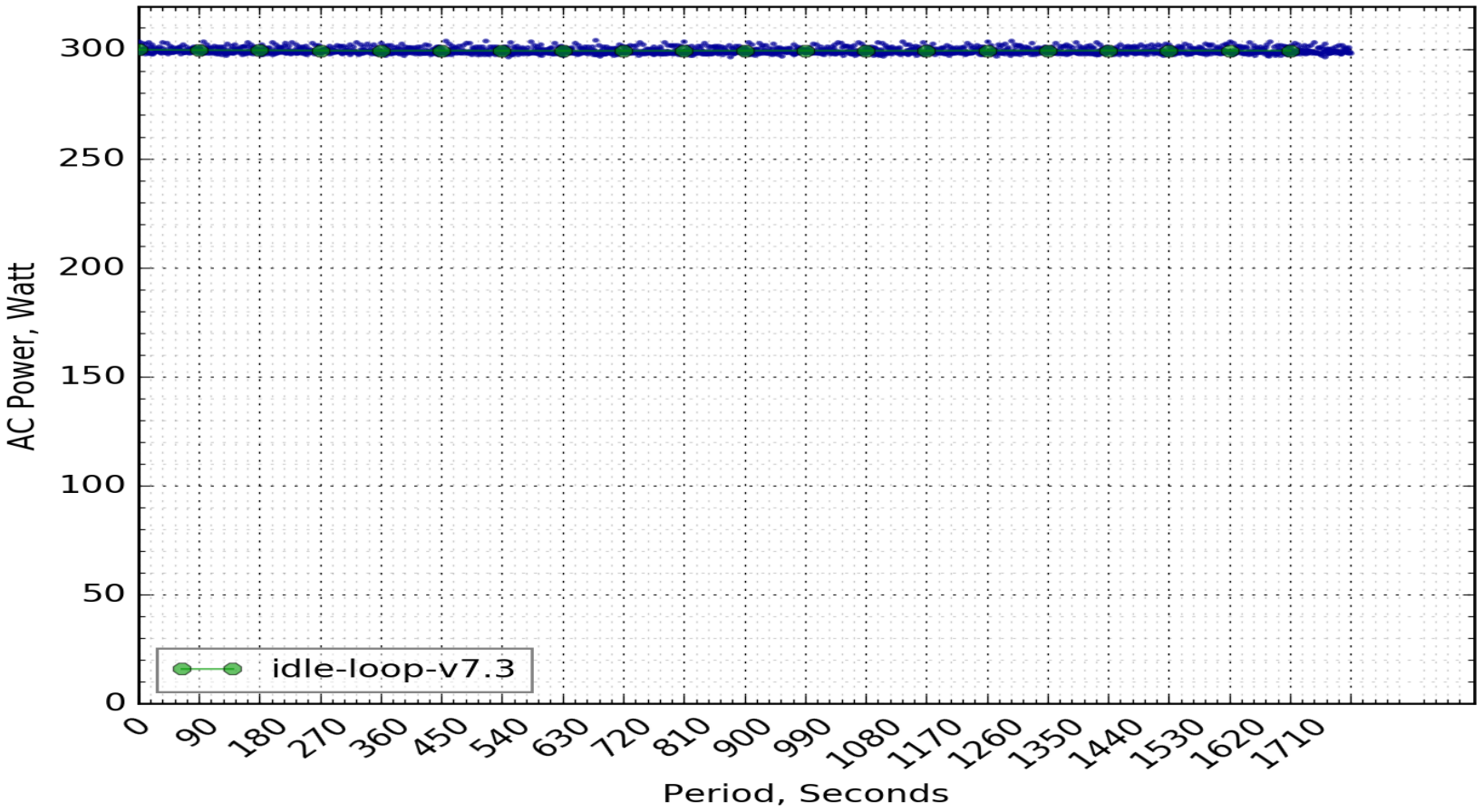
SW noise Impact on average idle power

- Powertop(8) utility to discover noisey wake source
- Fix bugs in noisy applications
- Fix bugs in noisy drivers

Linux-4.16 Idle Power Spike Issue



Linux-4.17 Idle Power Spike Issue FIXED



Performance States (P-States)

Processor Performance States (P-states)

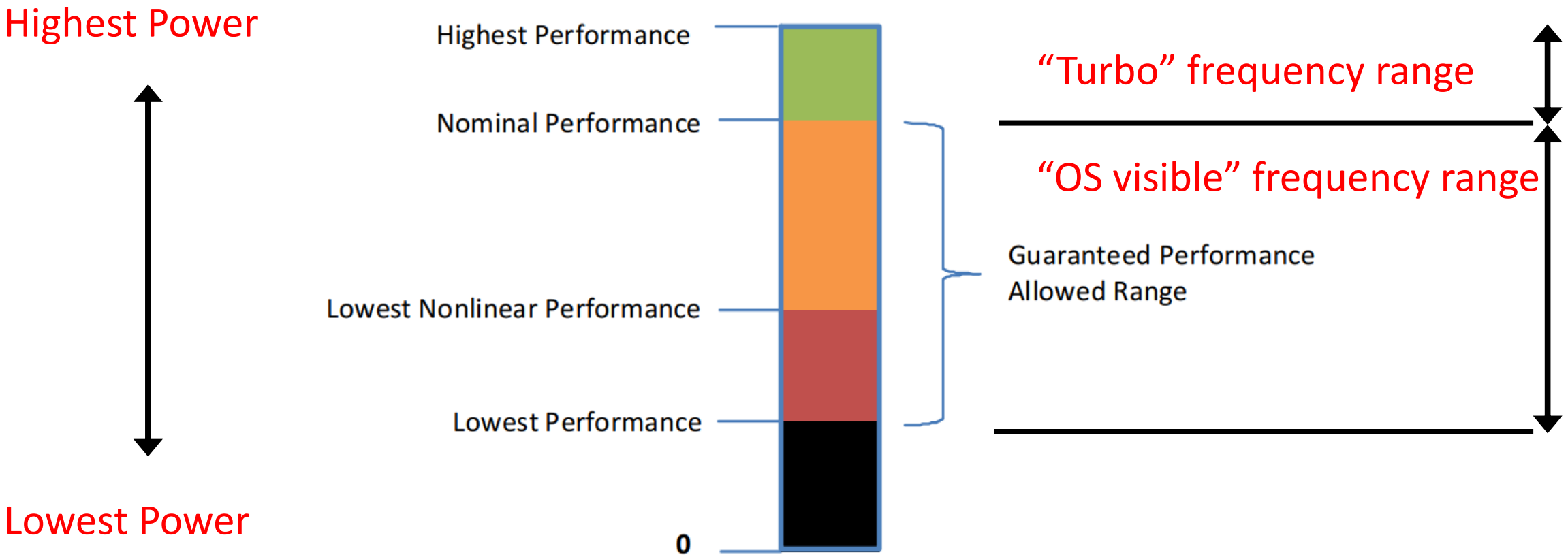


**Diagram from ACPI Specification*

Linux cpufreq sub-system

Governor: performance | powersave | ondemand | schedutil

Driver: intel_pstate | acpi_cpufreq

Linux P-state Administration with intel-pstate

HWP mode:

enabled by-default

unless “intel_pstate=no_hwp”

Manage with energy_performance_preference

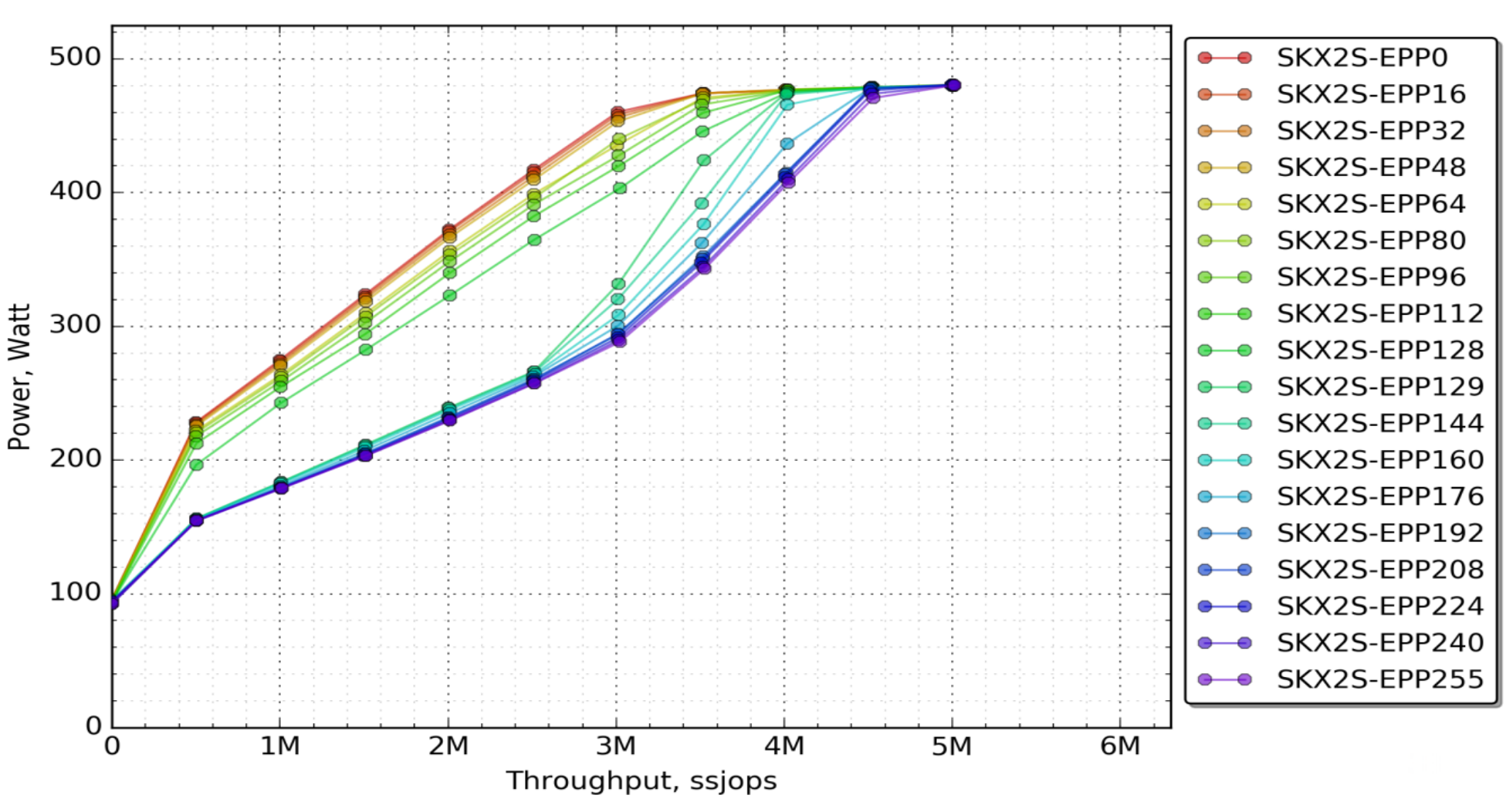
Software mode:

built-in SW governor active active as “powersave”

Turbo opt-out:

/sys/devices/system/cpu/intel_pstate/no_turbo: 0

HWP: Energy-Performance-Preference (EPP)



How to set EPP from Linux

/sys/devices/…/cpu*/cpufreq/energy_performance_preference

performance
balance_performance [default]
balance_power
power

Or use x86_energy_performance_policy(8)

Per-Core P-states (PCPS)

- Independent frequency and Voltage for every core

But…

- Thread Migration Risks fooling governors, impacting performance
  - Producer/consumer is worst-case

RAPL – Runtime Average Power Limiting

RAPL – Runtime Average Power Limiting

- Exposes Internal energy counter

```
# turbostat -i 1 -S -quiet -show PkgWatt
PkgWatt
18.44
18.20
…
```

n.b. counter is most accurate at high power

RAPL – Runtime Average Power Limiting

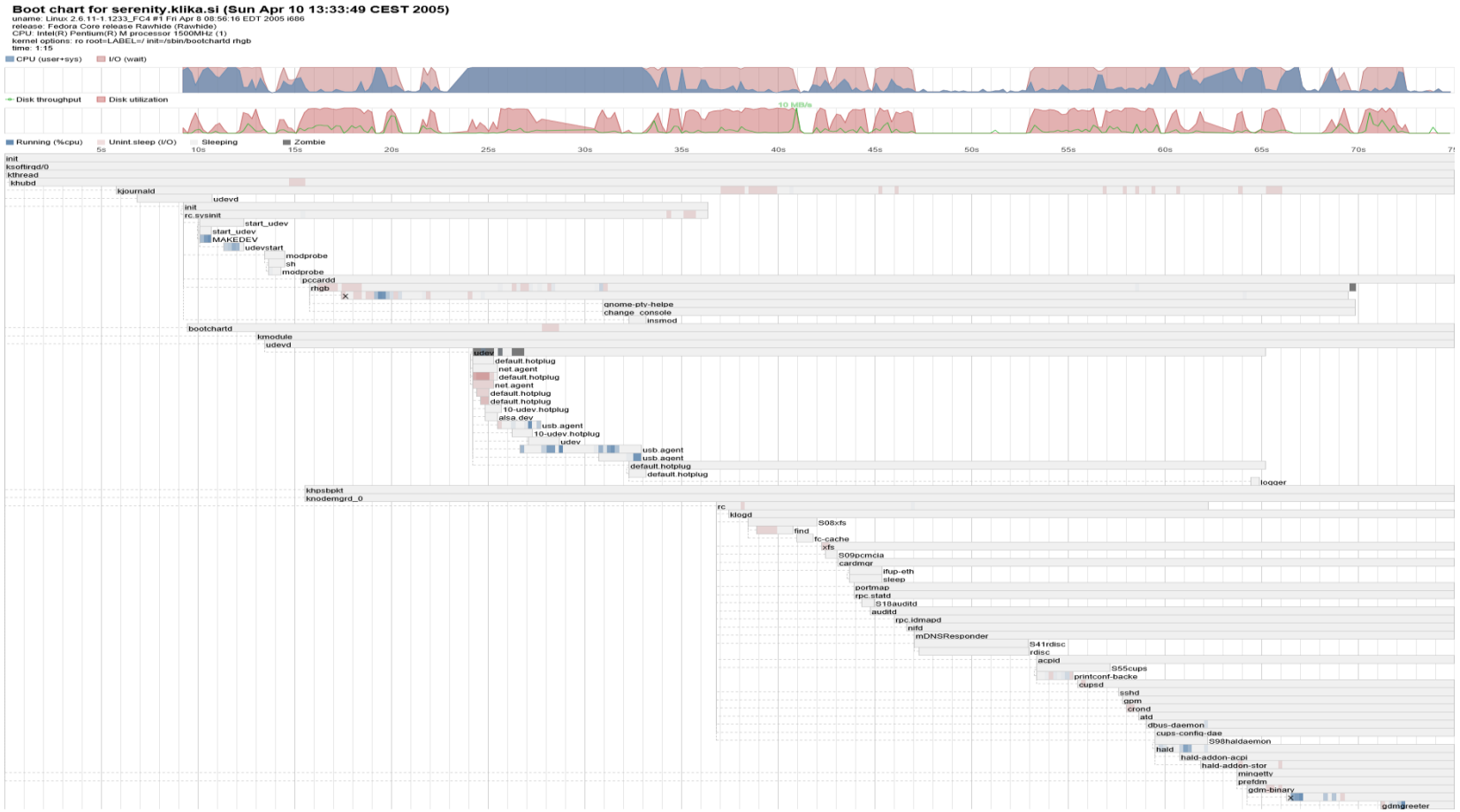
- RAPL can limit SOC to TDP and below
  - Uses least impact on performance to lower power consumption
- /sys/class/powecap/intel-rapl
  - Used by thermal daemon in response to temperature limits

Power-on/boot

Power-on/Boot

- Power-on/Boot performance is a power-management feature!
  - Off-state enjoys minimal power consumption
  - Off-state is measured in Government Regulations
  - Faster poweron/boot allows more use of off-state
    - Eg. Data center can add capacity on-demand if boot is fast enough
- Key Challenge: Spinning storage vs. power-cycle
  - Poor performance
  - Reliability concerns

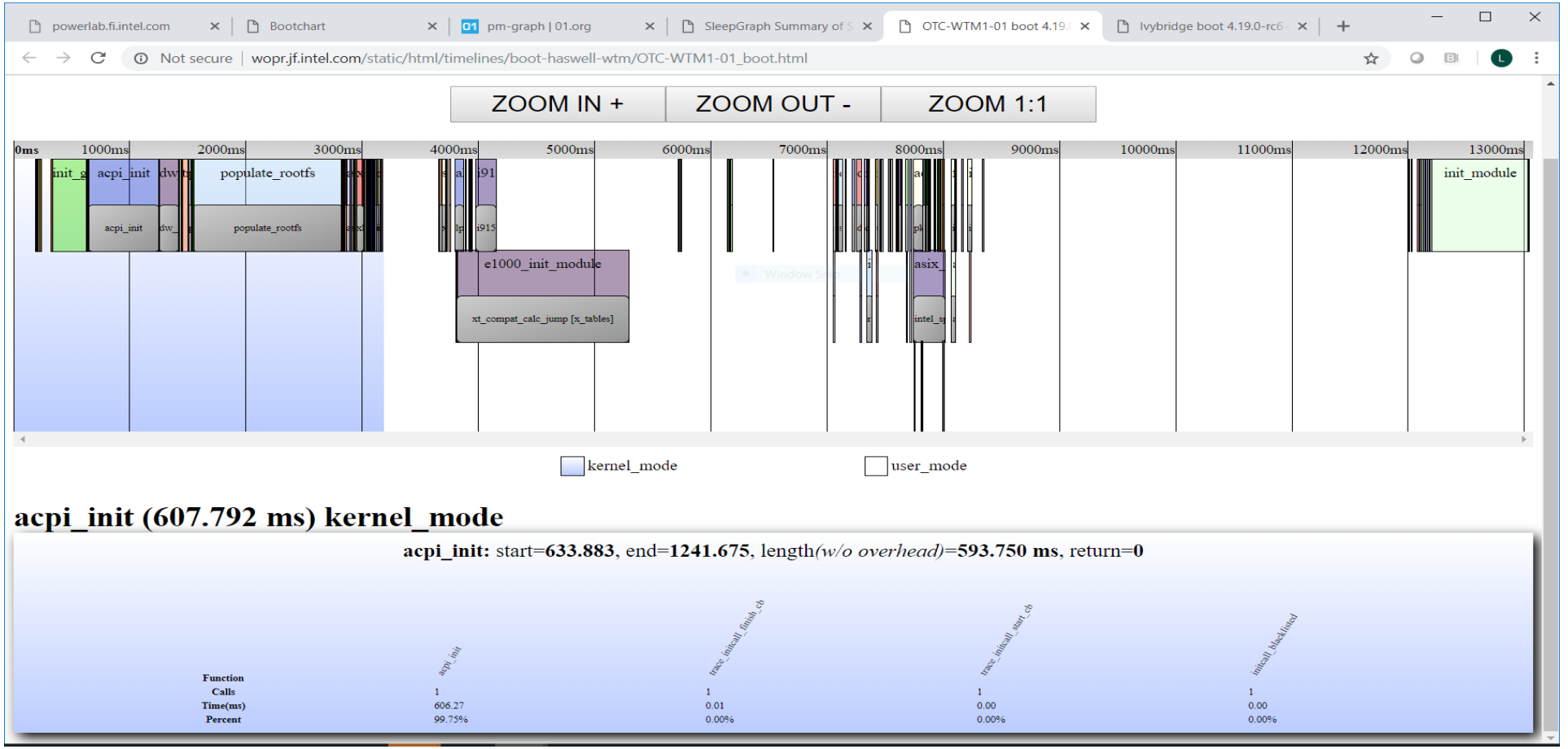
http://www.bootchart.org/



Kernel initcalls
user-space tasks

bootgraph.py

Bootgraphy.py produces HTML
Add kprobes and ftrace
Replaces bootgraph.pl, which produced SVG



Call To Action

- Expect Power Linux/Intel Power & Performance to be Great!
- Speak up when it is not!

  For upstream Linux Kernel, file bugs here:

  https://bugzilla.kernel.org

  (Product: Power Management)

Questions?