Tutorial on Networking and Power Management

Jesse Brandeburg

Agenda

- What is power management
- Terms and acronyms
- Why it matters
- Platforms and Power Management
- Measurement tools and examples
- Controls and Methods
- Cpupower example
- Effects, side-effects, and gremlins
- Previous Works
- Lots of thoughts
- Call to action
- Similar Links

What is power management?

Use less power

What are you willing to sacrifice?

Latency? Throughput?

Think ahead

CPU

Reduce or stop cycles of the CPU (C-state)

Reduce the frequency of the CPU (P-state)

RAM

Frequency changes, more or fewer DIMMS

Uncore

Reduce or stop cycles of the uncore (PC-state)

Stops DMA

Adapter(s)

Device state (D0, D3)

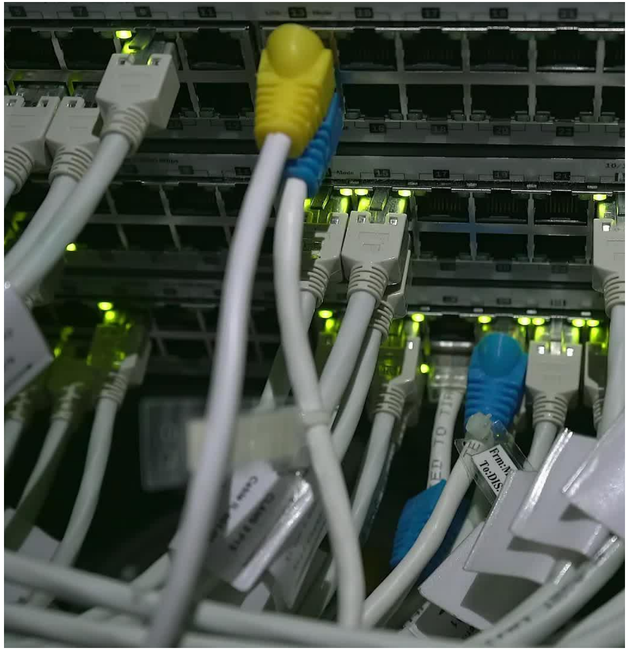
Energy Efficient Ethernet (EEE)

PCI Express power management (ASPM)

Link State (link down, reduce speed)

Terms and Acronyms

- CPU – I hope you know this one
    - C-state
        - Core state – running or one of the various sleep states which take a certain time to wake from each state
    - P-state
        - Frequency management
- ASPM
    - Active-State Power Management – PCI Express power down link when no traffic
- EEE (802.3az) – also “Green Ethernet”
    - Energy Efficient Ethernet – power down transmitter when “idle”
- Uncore
    - PC-state: Package C-state: CPU+uncore’s own sleep states
    - Usually contains the memory controller and DMA controller logic, among other things



Why it matters

- Hypothetical
  - Data center with 10,000 servers
  - 48 port switches (ToR)
  - Save 10 watts per server, per hour
  - 10wH * 10,000 = 100,000 wH aka 100kwH
  - * 24 hours = 240kwH per day
  - US range (2023) 0.084 $/kwH to 0.20 $/kwH, Oregon commercial rate $0.131 [1]
  - 240 kwH * 0.131 = 26.2 dollars / day * 365 days
  - $9,563 USD a year

[1] Electricity Rates by State (October 2023) | ChooseEnergy.com®

Insights on Networking and Power

High speed ethernet is the only asynchronously driven (by surprise receive traffic) high speed I/O device

Platforms and Power Management

- Servers are waaaay different than laptops
- Servers are big power consumers
  - Power supplies (yep, they **use** power, not just supply it)
  - Big processors
  - Lots of RAM
  - Plug in cards (I/O, Ethernet)
  - Lasers
  - Fans
  - (potentially) Lots of storage devices
- 500 to 1,200+ watts per server

Measurement Tools and Examples

- turbostat
- Intel PTAT tool (Intel Design Center)
- GNOME power manager (client)
- PowerTOP (client)
- External power measurement (for example Kill-a-watt, Watts Up, many data center power distribution systems)

Control and Methods

- Kernel
  - cpufreq subsystem
  - Power aware scheduler
- cpupower
  - cpupower idle-info
  - cpupower idle-set --help
  - cpupower frequency-info
  - cpupower frequency-set --help
- sysfs
  - /sys/devices/system/cpu/cpu1/cpuidle/state2/name == C1E
  - /sys/devices/system/cpu/cpu1/cpufreq/
- Scripts
  - https://github.com/VitorRamos/cpufreq

Cpupower example

- What do I have?

```
cpupower idle-info
```

- Change power state management to reduce latency, but don’t poll

```
cpupower idle-set –D10
   □(my system idle package watts[1] went from 67 watts to 137
   watts)
```

- What does it do?
  - Sets CPU maximum wake time to 10us
  - Self selects correct C-state to honor above limit

Effects, side-effects, and gremlins

- Lots of times, optimizing for power means sacrificing
  - Latency – it goes up
  - Throughput – it might go down, or cause RTT to go up (possibly need for bufferbloat)
  - Responsiveness upon initial request

- The past Best-Known Methods (BKM)
  - Just turn off power management!
    - Continuous 1,000+ watt usage (oops)
  - Let's poll!
    - Uses a LOT of CPU, therefore lots of power
  - Draconian
    - Thermal limiting the platform or CPU (don't get hot!)
    - /dev/cpu_dma_latency (whole platform! One setting)
- BIOS Settings!

Previous work

- Reduce power by using RSS table modification in real-time to scale queues, and sleep CPUs
  - Brandeburg / Creeley – netdev 0x15 [1]

[1] Netdev 0x15 - Dynamic Interface Power Management (PowerMAN)

Lots of thoughts

- How do we help the networking stack give more feedback to the scheduler, power manager?
- Can the **stack** keep a CPU awake "a little longer" when the networking stack is expecting more traffic?
  - Power aware stack
- Busy poll (as a side effect of polling) keeps the CPU awake by polling from kernel to driver, is there a more granular option, or use mwait somehow?
- Should we consider an extra property of a "queue" the power policy of that queue?
- Kernel is missing granular driver-available per-CPU policy for power, today only has userspace /dev/cpu_dma_latency which affects all CPUs, and cpu power limits and c-state limits
- Scheduler delay of 1ms is much too long for 100Gb/s + ethernet

Call to Action

Working group to drive net-stack power awareness?

Meet monthly

Curate ideas {publish}

Create list of tasks {publish}

Prioritize tasks

Create some patches from tasks and send to list

Lets try! Want to help?

Contact jesse.brandeburg@intel.com

or mail to net-power@netdevconf.info

Cool similar links

- Redhat
  - Chapter 14. Importance of power management Red Hat Enterprise Linux 9 | Red Hat Customer Portal
- DPDK power management
  - 56. Power Management — Data Plane Development Kit 23.11.0-rc1 documentation (dpdk.org)