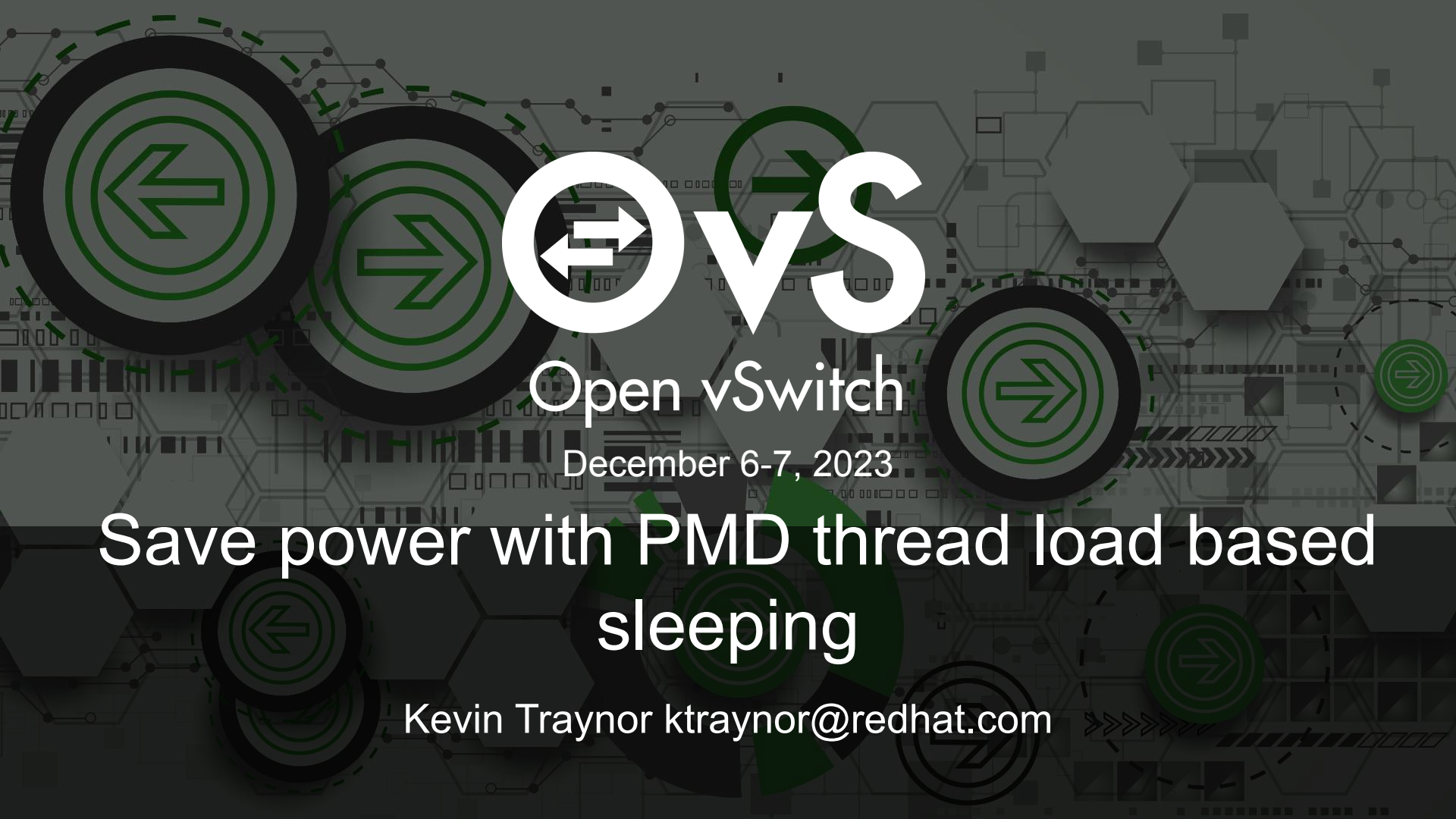


December 6-7, 2023

Save power with PMD thread load based sleeping

Kevin Traynor ktraynor@redhat.com

Introduction

- OVS userspace PMD threads
- System configuration
- Options for doing less work
- Tuning
- PMD load based sleeping operation
- Testing

OVS-DPDK PMD thread

- 1:1 on isolated core
- Datapath processing

- **Poll Rx queues**
  - Runs in a loop in userspace polling ports for packets
  - Calls Rx function for DPDK NIC and vhost ports using DPDK drivers
- **Processes packets**
  - Classification, actions, output

- Typical type of behaviour an application using DPDK
  - High throughput
  - Low latency
  - High cpu cycles usage

Stop/Slow Rx queue polling == Save power ?

- Isolate PMD thread cores
- C-states - Processor Operating States
  - C0 (full power) down to C6 (Deep sleep state)
- Enable at BIOS level

- Enable with system tuning software (tuned)
  - cpu-partitioning-powersave
  - tuned v2.20.0
- What about the other cores ?
  - e.g. 28 cores in a socket
  - 2 for OS
  - 4 for PMDs
  - 22 cores doing what ?

How stop/slow polling ?

- Sleeping
  - Sleep between polls
  - Agnostic to device type
  - Simple implementation
    - if no/low packet received sleep in polling loop
  - Wakes up when no traffic
  - Gradual and adaptive to packet rate

- NAPI
  - Change device into interrupt mode
  - Each device driver required to support interrupt mode
  - More complex implementation
    - Different OVS code needed for different device types

      i.e. DPDK NIC using Ethdev API and DPDK vhost using vhost lib API
  - Does not wakeup when no traffic
  - Binary operation - interrupt or polling
    - Threshold for enabling ?
    - Might not save power during low traffic

How long should we sleep for?

- What happens if a packet arrives during a sleep ?
  - It must wait until after the sleep

- Trade-off between longer sleep and greater wakeup packet latency
  - Sleep longer
    - Do less work when no packets => implies more power saving
    - Longer wakeup packet latency
  - Sleep shorter
    - Do more work when no packets => implies less power saving
    - Shorter wakeup packet latency

- Max sleep time tunable
  - pmd-sleep-max (pmd-maxsleep in OVS 3.1) e.g. max sleep 100 uS
    - `$ ovs-vsctl set Open_vSwitch . other_config:pmd-sleep-max=100`

- Also need to consider Processor C-State wakeup times
  - Check `/sys/devices/system/cpu/cpu8/cpuidle`
    - `cpupower -c 8 idle-info | grep -e ^C -e Latency`

```
C1:
Latency: 2
C1E:
Latency: 10
C3:
Latency: 40
C6:
Latency: 133
```

PMD load based sleeping - Low traffic rate

- `$ ovs-vsctl set Open_vSwitch . other_config:pmd-sleep-max=50`
  - poll Rx queue. Get 32 packets. Process packets. No Sleep.
  - poll Rx queue. Get 2 packets. Process packets. Sleep 1 uS.
  - poll Rx queue. Get 0 packets. ~~Process packets.~~ Sleep 2 uS
  - ...
  - poll Rx queue. Get 5 packets. Process packets. Sleep 50 uS
  - poll Rx queue. Get 5 packets. Process packets. Sleep 50 uS
  - poll Rx queue. Get 5 packets. Process packets. Sleep 50 uS
  - poll Rx queue. Get 5 packets. Process packets. Sleep 50 uS
  - poll Rx queue. Get 32 packets. Process packets. No Sleep.

Transition to max sleep

max sleep steady state

Transition out of sleep

PMD load based sleeping - Low traffic rate

- `$ ovs-vsctl set Open_vSwitch . other_config:pmd-sleep-max=100`
  - poll Rx queue. Get 32 packets. Process packets. No Sleep.
  - poll Rx queue. Get 2 packets. Process packets. Sleep 1 uS.
  - poll Rx queue. Get 0 packets. ~~Process packets.~~ Sleep 2 uS
  - ...
  - poll Rx queue. Get 10 packets. Process packets. Sleep 100 uS
  - poll Rx queue. Get 10 packets. Process packets. Sleep 100 uS
  - poll Rx queue. Get 32 packets. Process packets. No Sleep.

Transition to max sleep

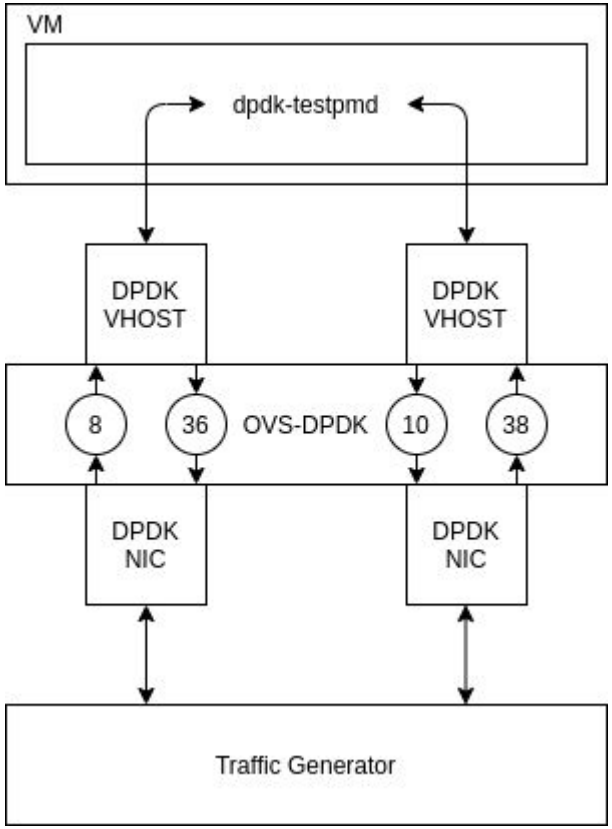
max sleep steady state

Transition out of sleep

- How long did we sleep for ?
  - `$ ovs-appctl dpif-netdev/pmd-perf-show`

```
- sleep iterations:          25249  ( 99.6 % of iterations)
Sleep time (us):           2546186  (97 us/iteration avg.)
```

Test Topology



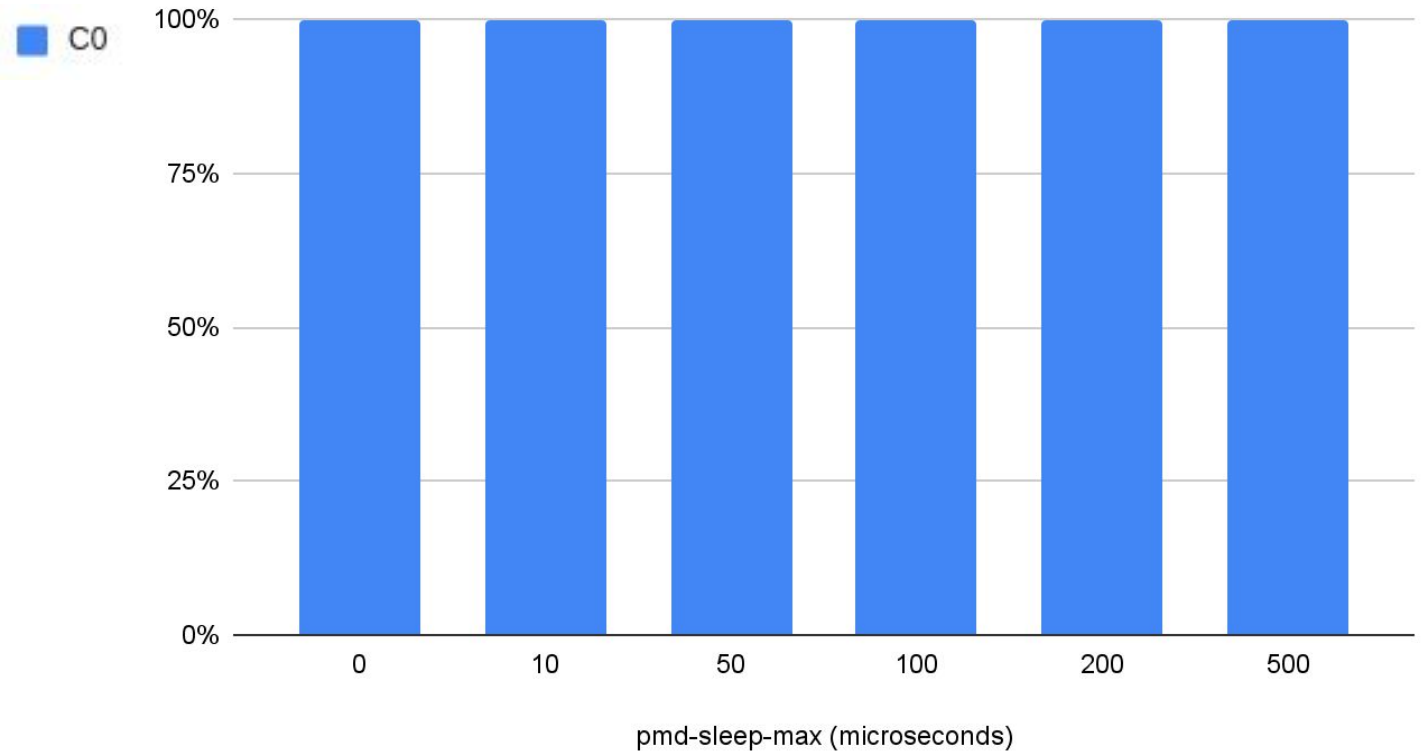
Test cases and measurement

- Test cases
  - Different traffic rates
    - Max throughput, 1 Mpps, 1 Kpps, 0 pps
  - Different max sleep times
    - 0 uS, 10 uS, 50 uS, 100 uS, 200 us, 500 uS
  - 64 byte packets

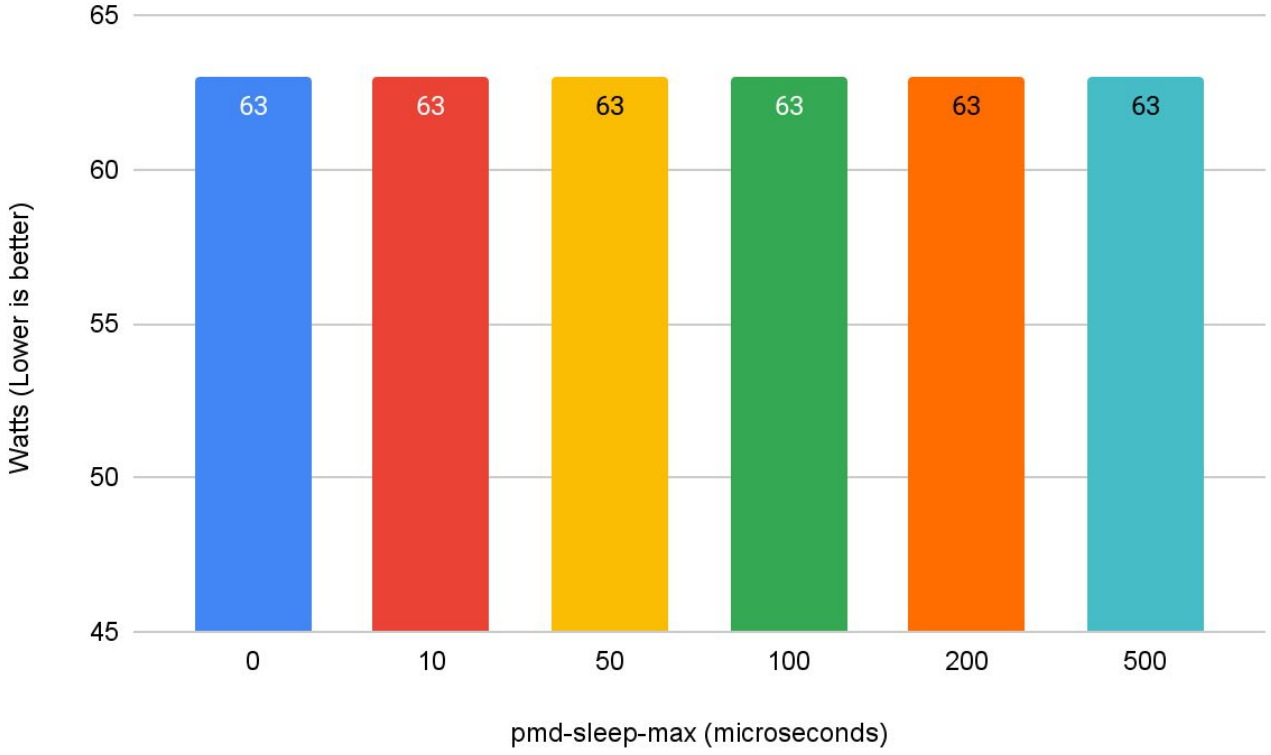
- Measurements
  - C-state
  - Power usage (Watts)
  - Wake up latencies

- Tools
  - `pcm-power`
  - `cpupower`
  - `powerstat`
  - `powertop`

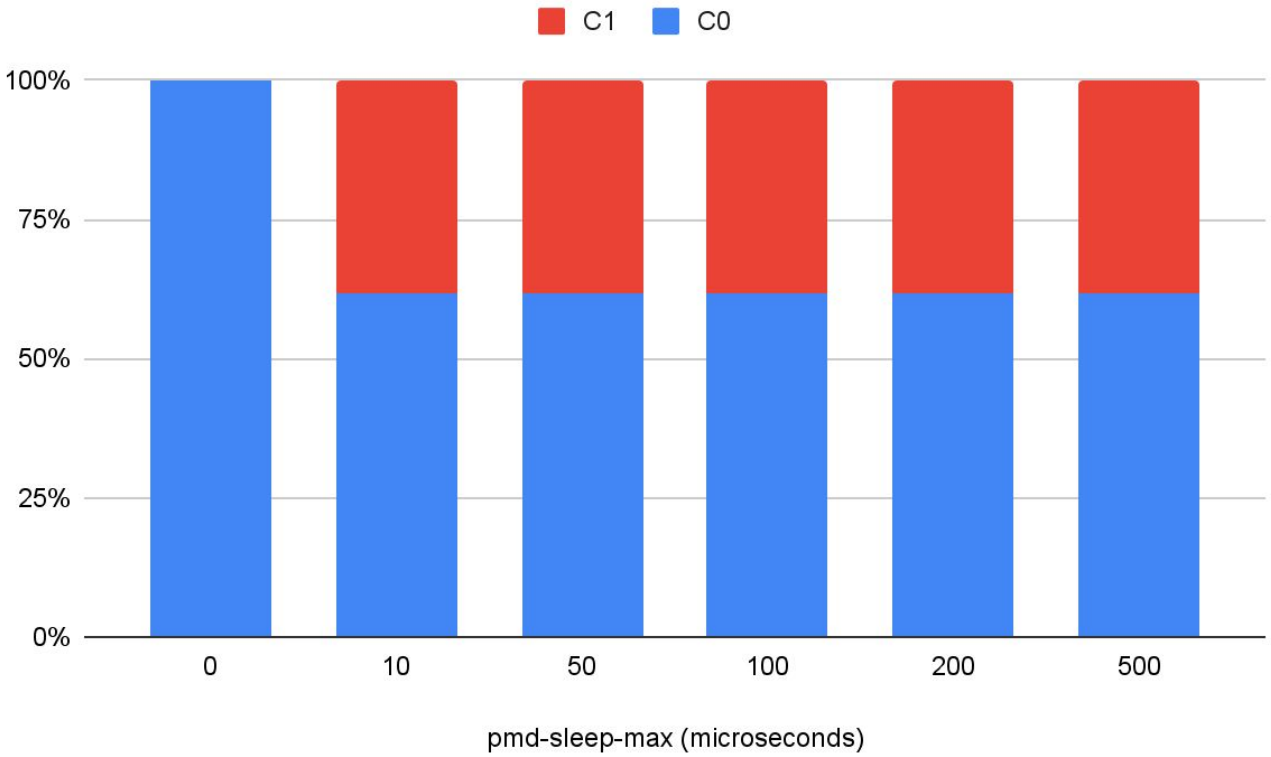
Max throughput - Processor C-States



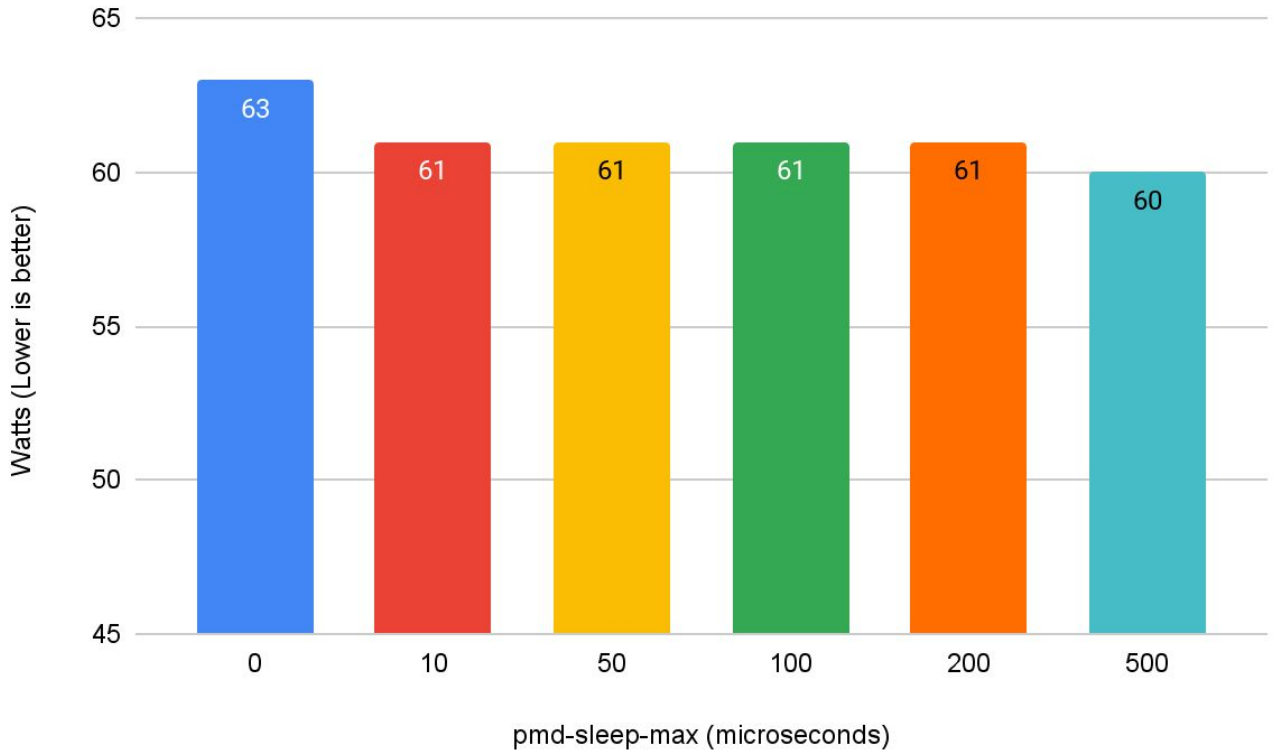
Max throughput - Power consumption



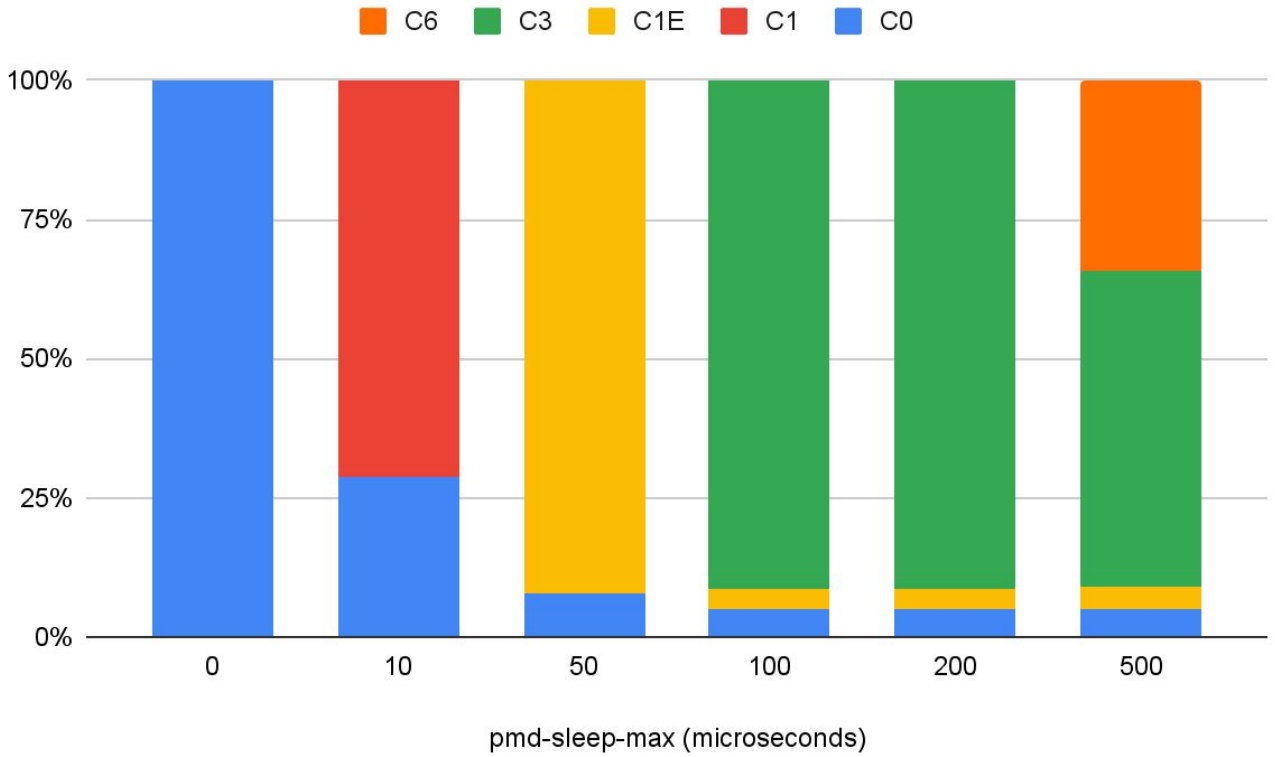
1 Mpps - Processor C-States



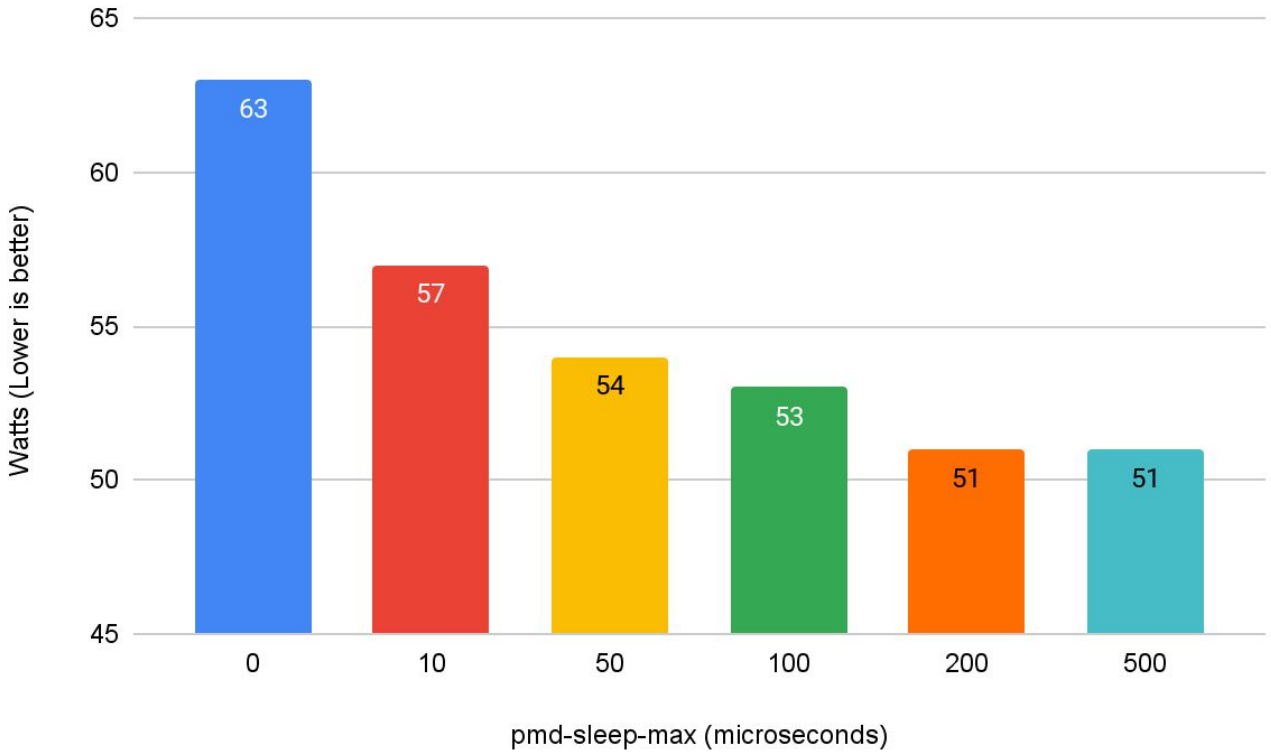
1 Mpps - Power consumption



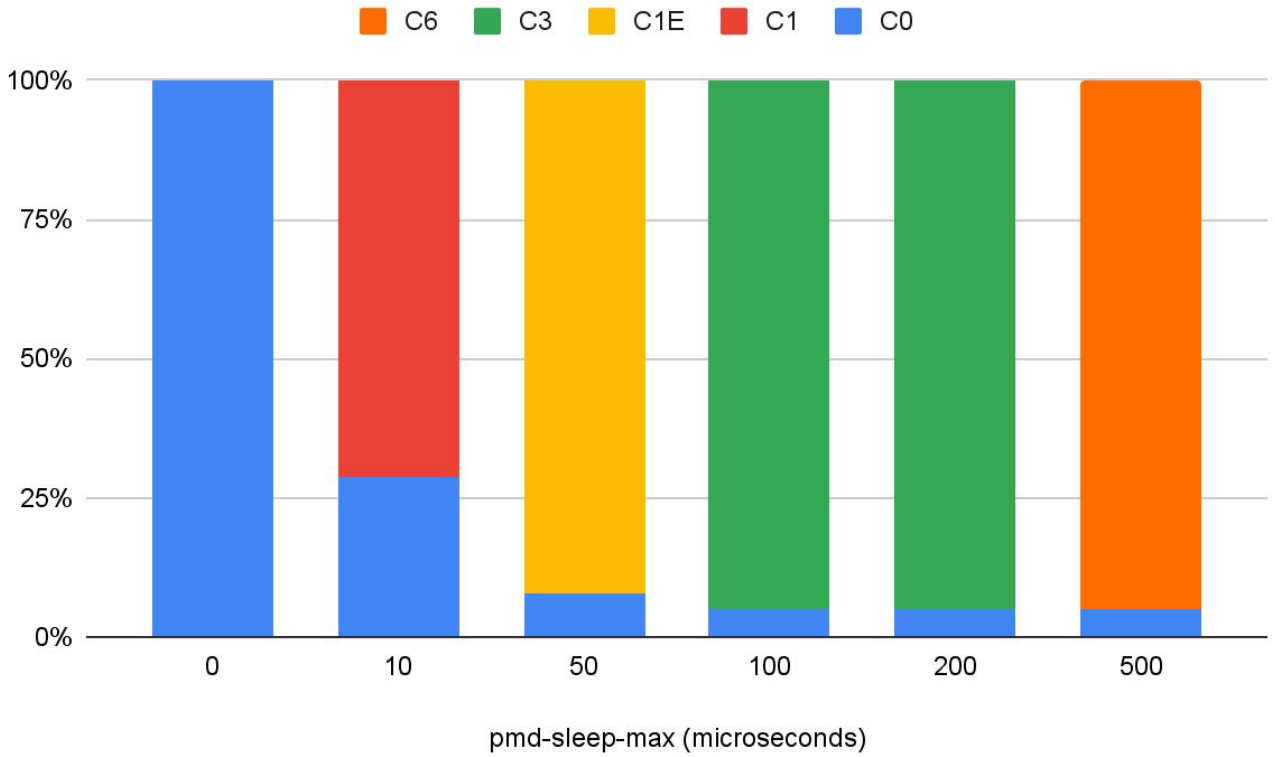
1 Kpps - Processor C-States



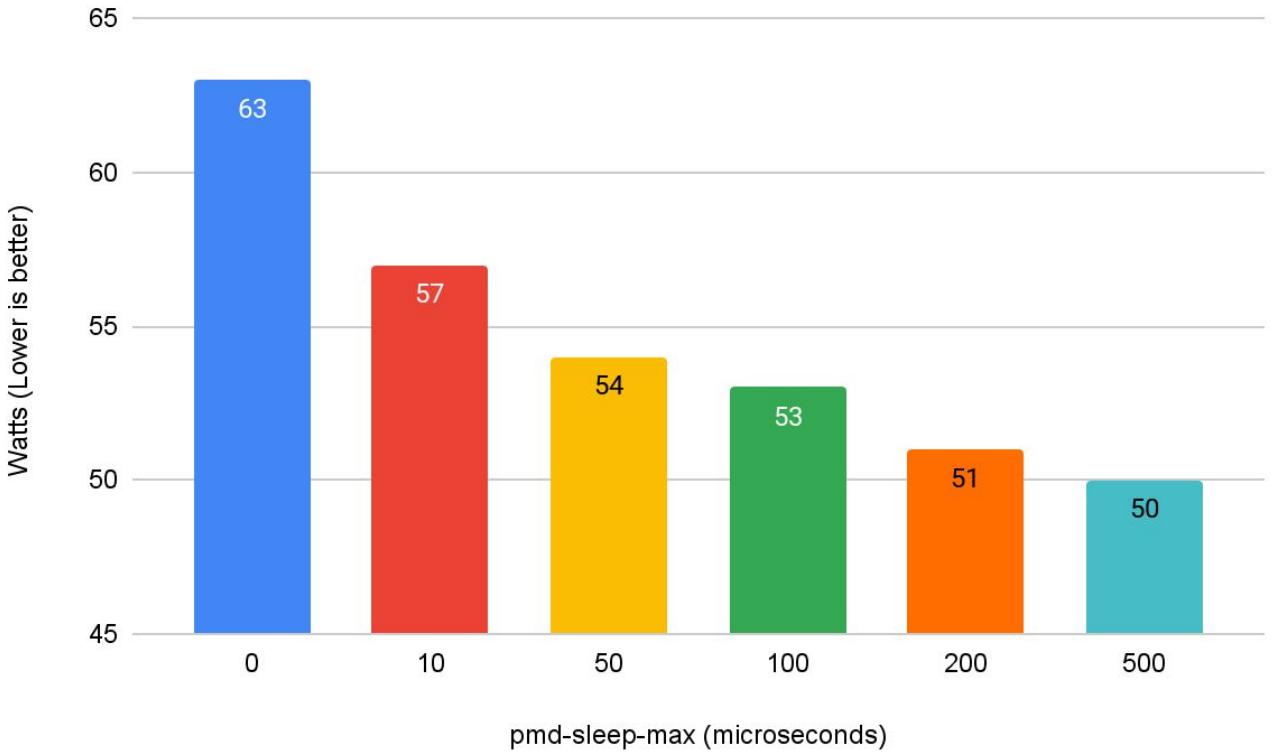
1 Kpps - Power Consumption



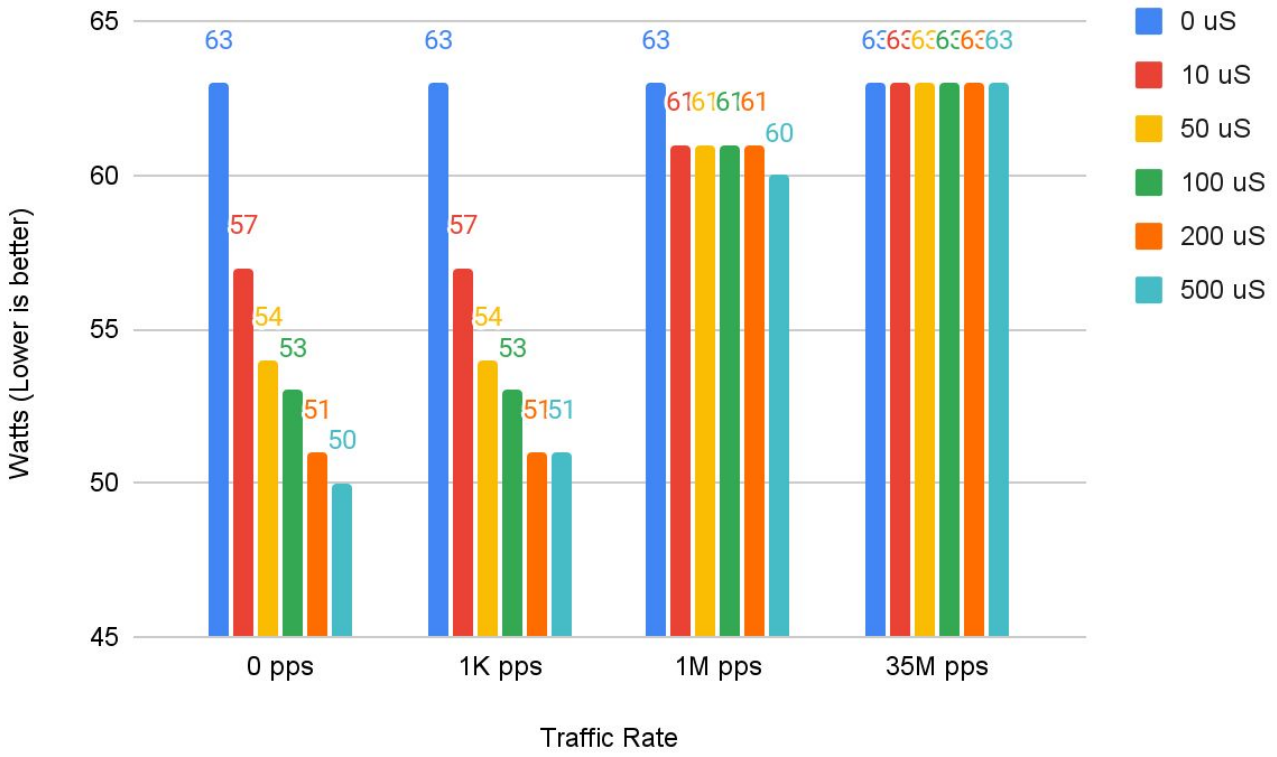
0 pps - Processor C-States



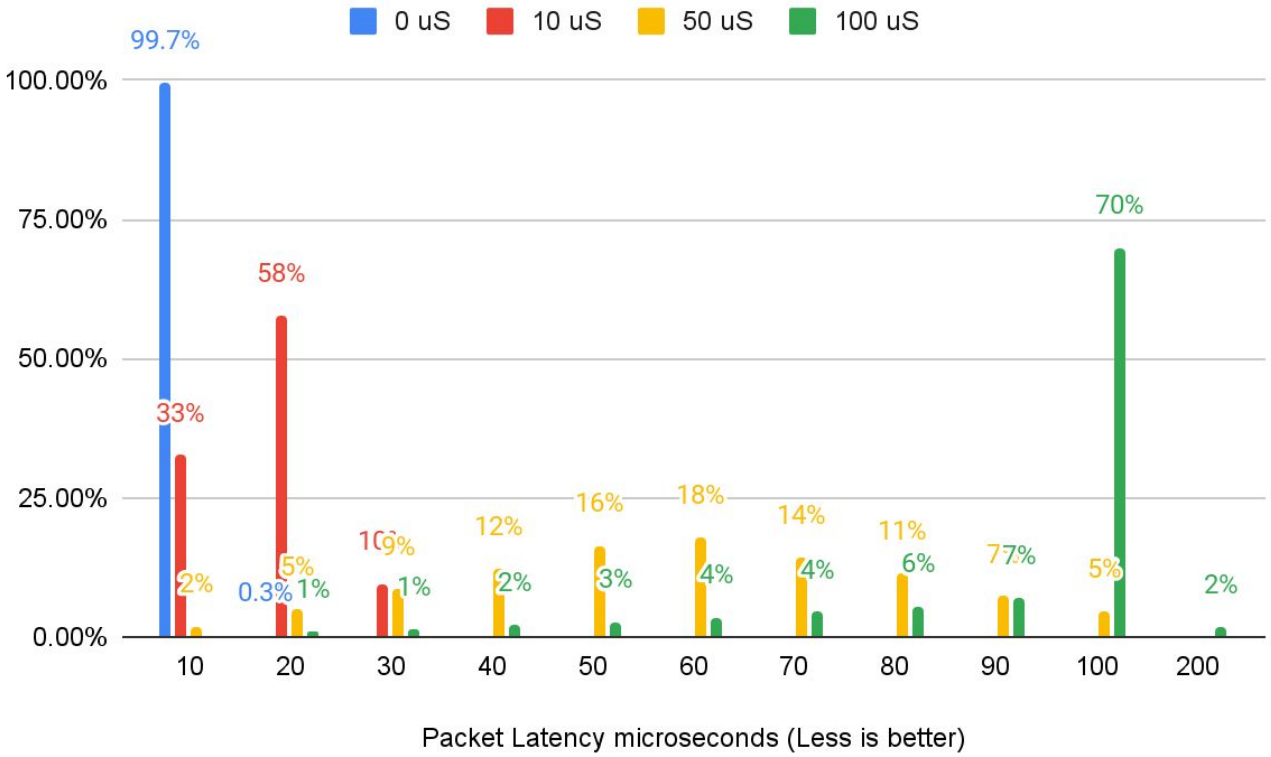
0 pps - Power Consumption



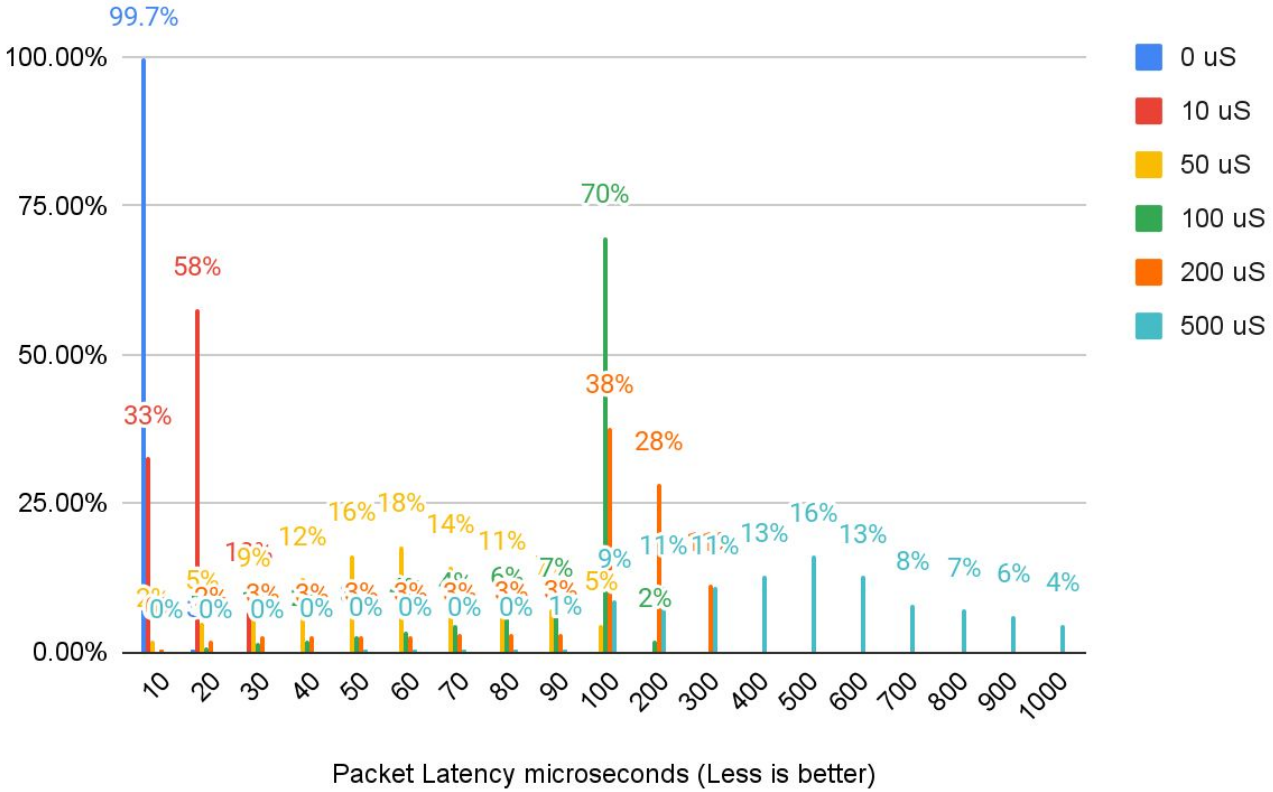
Max sleep time vs. packet rate matrix



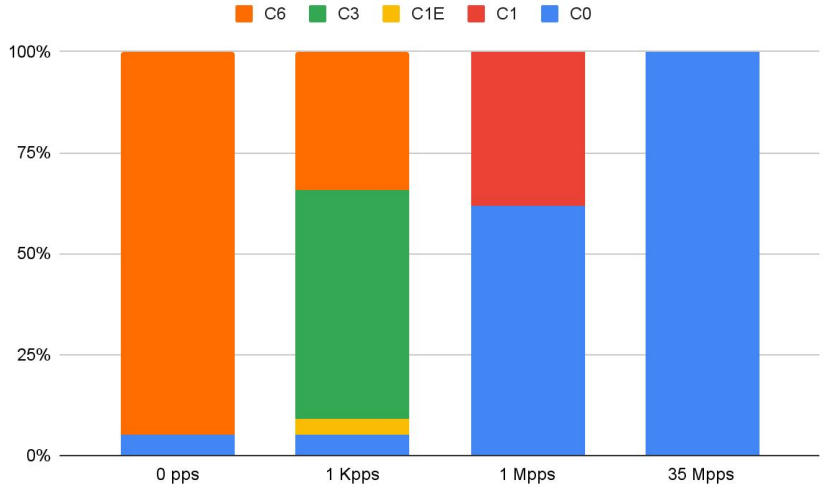
Wakeup packet latency - limited range

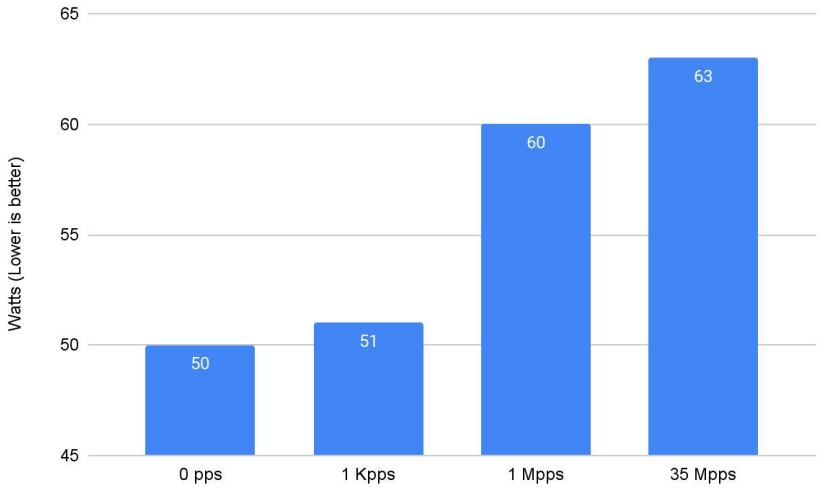


Wakeup packet latency - full range

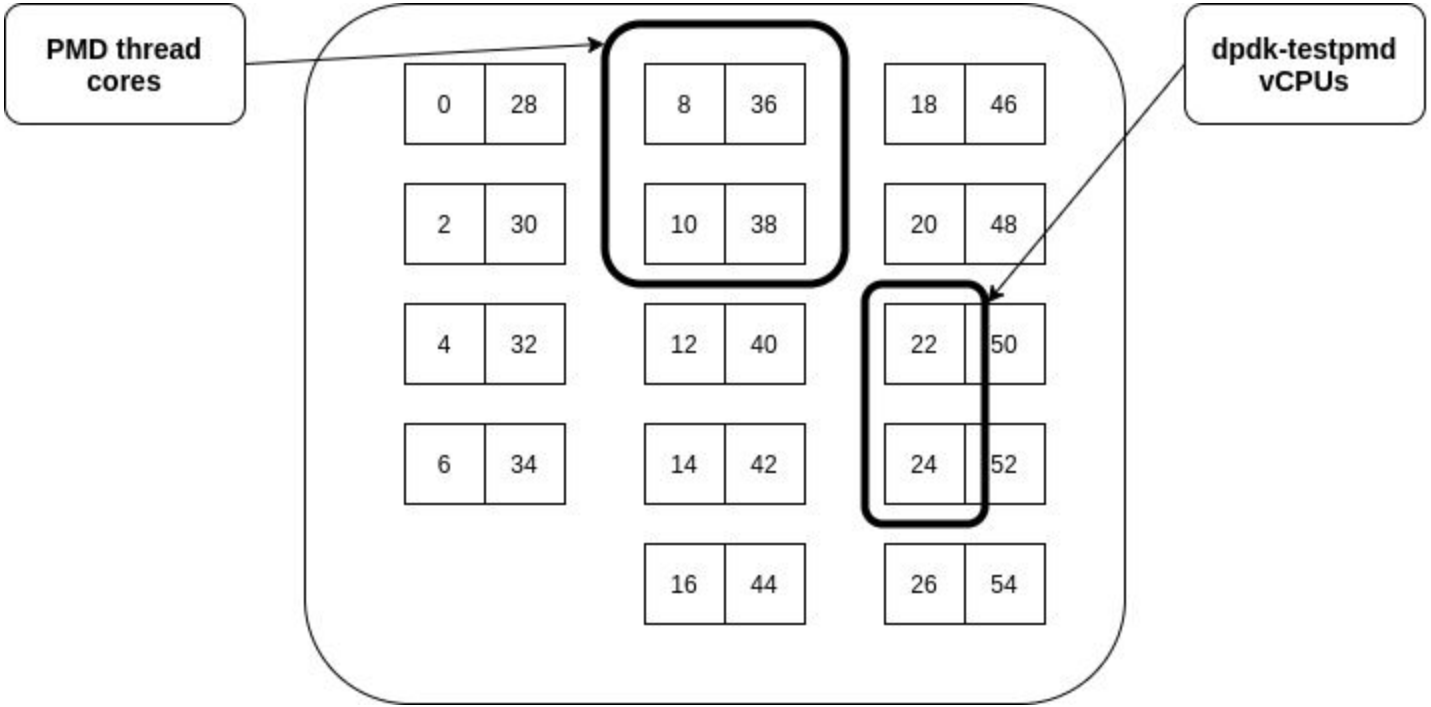


500us sleep - Something suspicious?

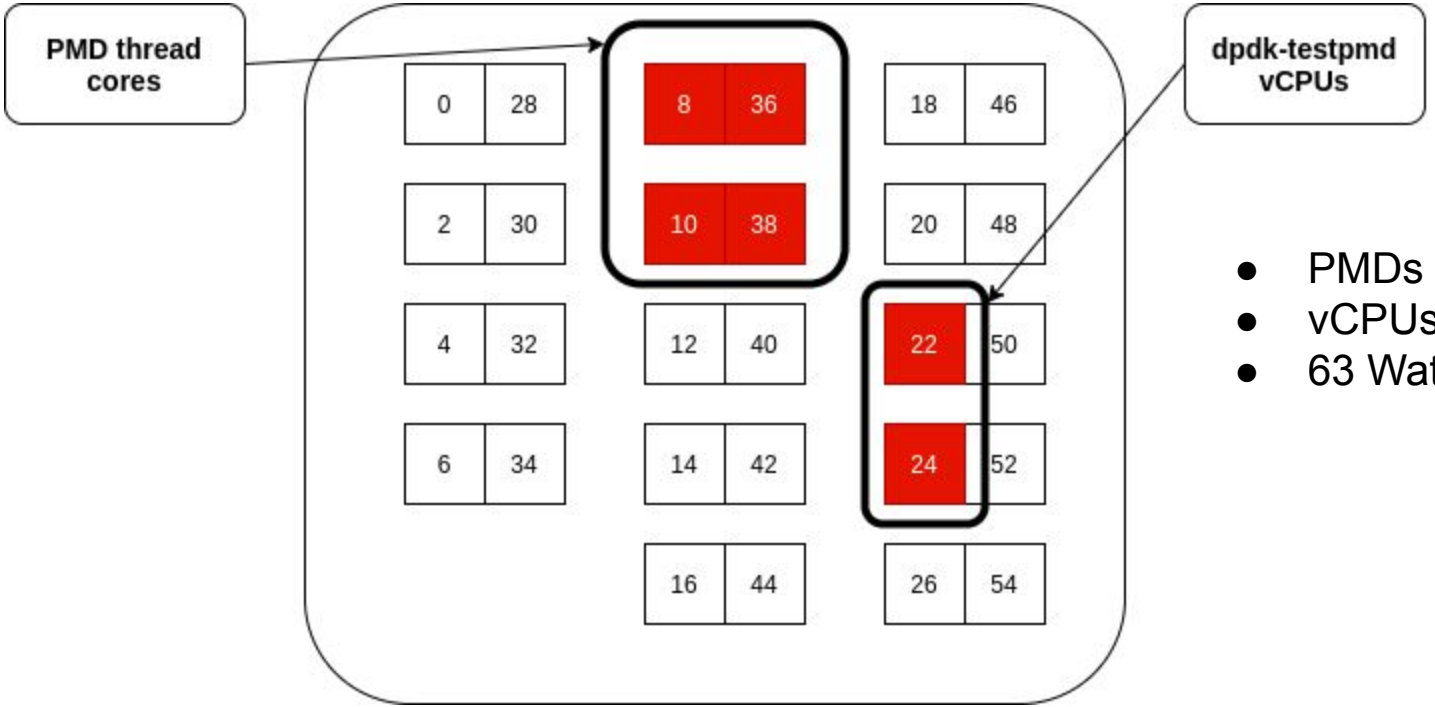




DUT Cores

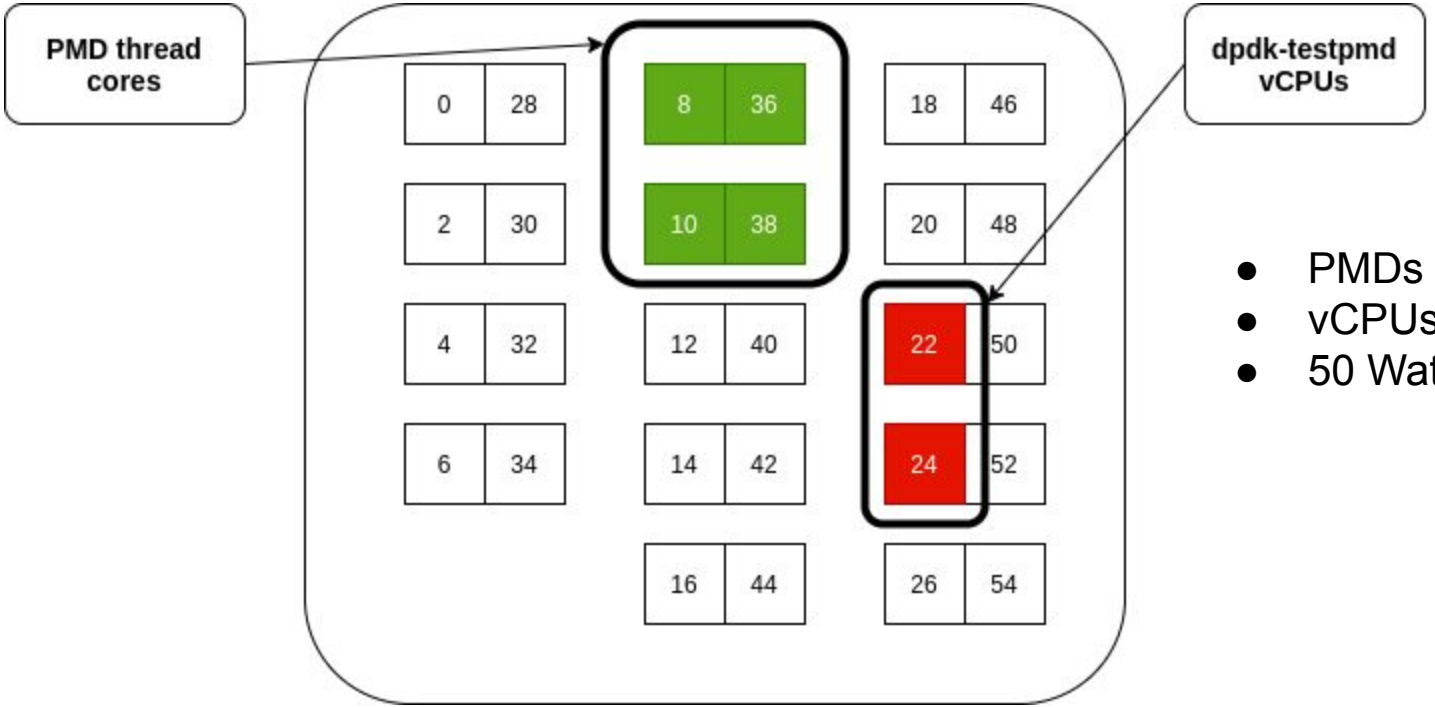

PMD thread cores
dpdk-testpmd vCPUs
0
28
8
36
18
46
2
30
10
38
20
48
4
32
12
40
22
50
6
34
14
42
24
52
16
44
26
54

DUT Cores - pmd-sleep-max=0



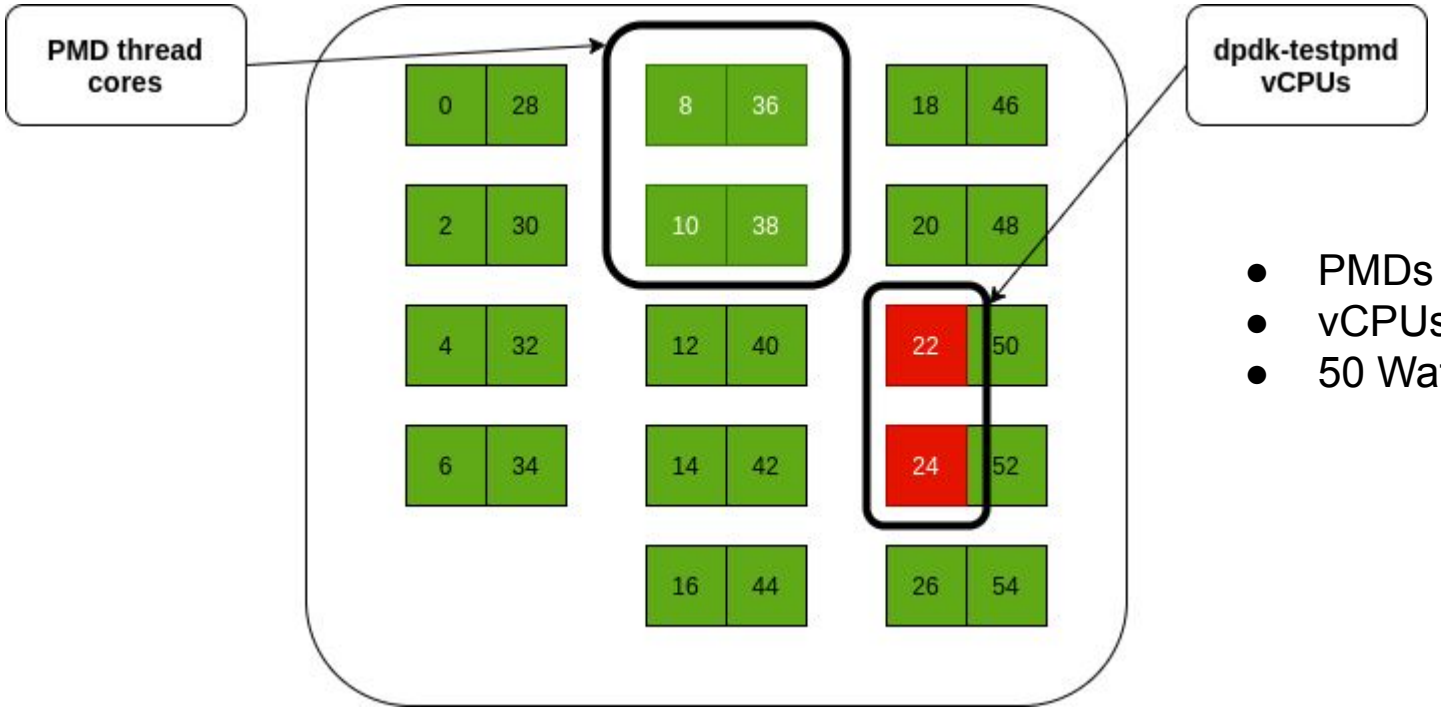
- PMDs - in use
- vCPUs - in use
- 63 Watts

DUT Cores - pmd-sleep-max=500



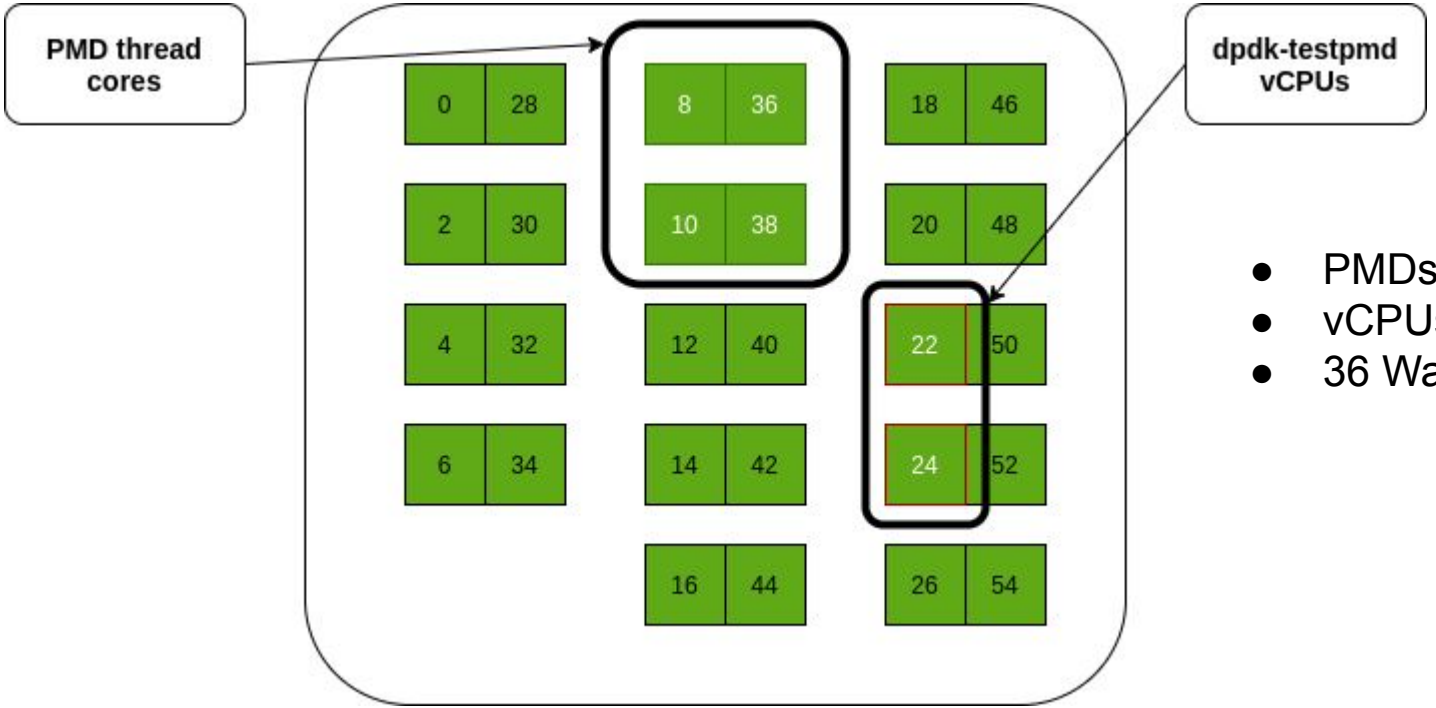
- PMDs - sleeps
- vCPUs - in use
- 50 Watts

DUT Cores - pmd-sleep-max=500



- PMDs - sleeps
- vCPUs - in use
- 50 Watts

DUT Cores - pmd-sleep-max=500 & no testpmd



- PMDs - sleeps
- vCPUs - stopped
- 36 Watts

Summary

- PMD load based sleeping feature is available in OVS 3.2
- Experimental in OVS 3.1

- Trade off between max sleep time and power saving (under zero/lowest load)

- Sleep time adapts to traffic rate

- Transition gradually into longer sleeps
- Transition quickly back to full power

- System configuration matters
- Other cores matter