

Optimizing Chromium Low-Power Workloads on Intel Notebooks

Len Brown, Ricardo Neri, Vaibhav Shankar - Intel

LPC 2022:

"Linux needs a Scheduler QOS API -- and it isn't nice(2)" [Video](#) [PDF](#)

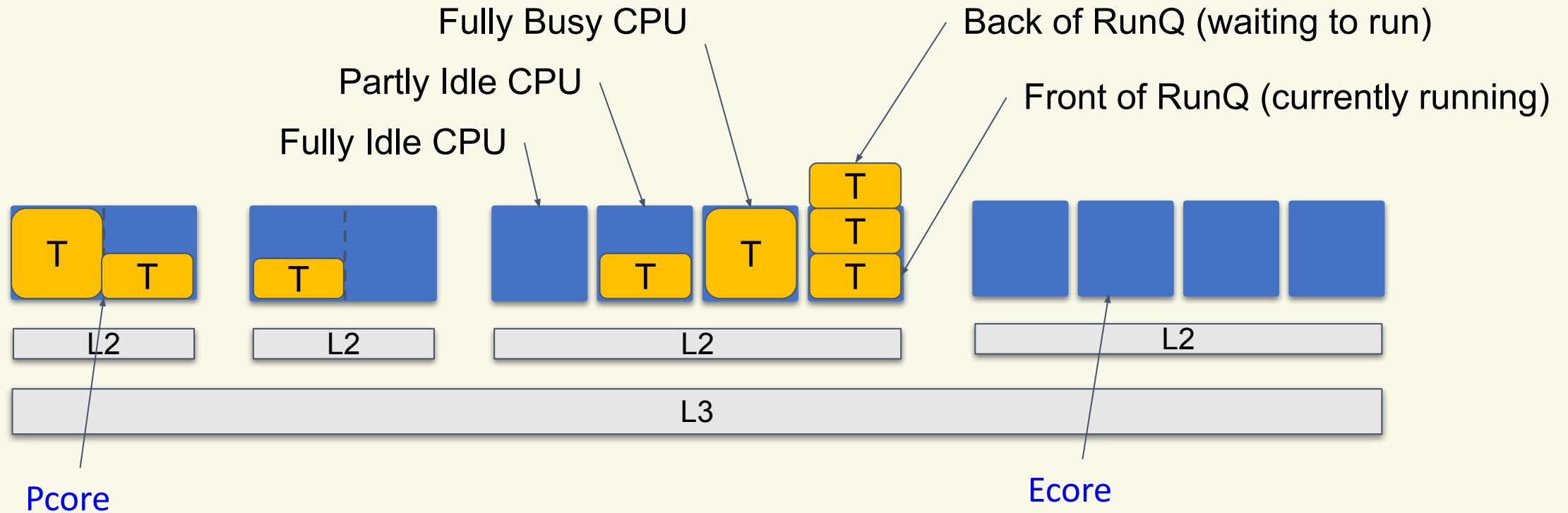
Lively discussion in the session, and more in the hallways.

Vincent: *"Demonstrate value"*

Acknowledgement

- Jianlin "Johnny" Qiu, Zheda Chen
- Srinivas Pandruvada, Rui Zhang, Noor Mubeen
- Tim C. Chen, Yu C. Chen, Rafael Wysocki

Linux Tasks on Intel Hybrid Topology

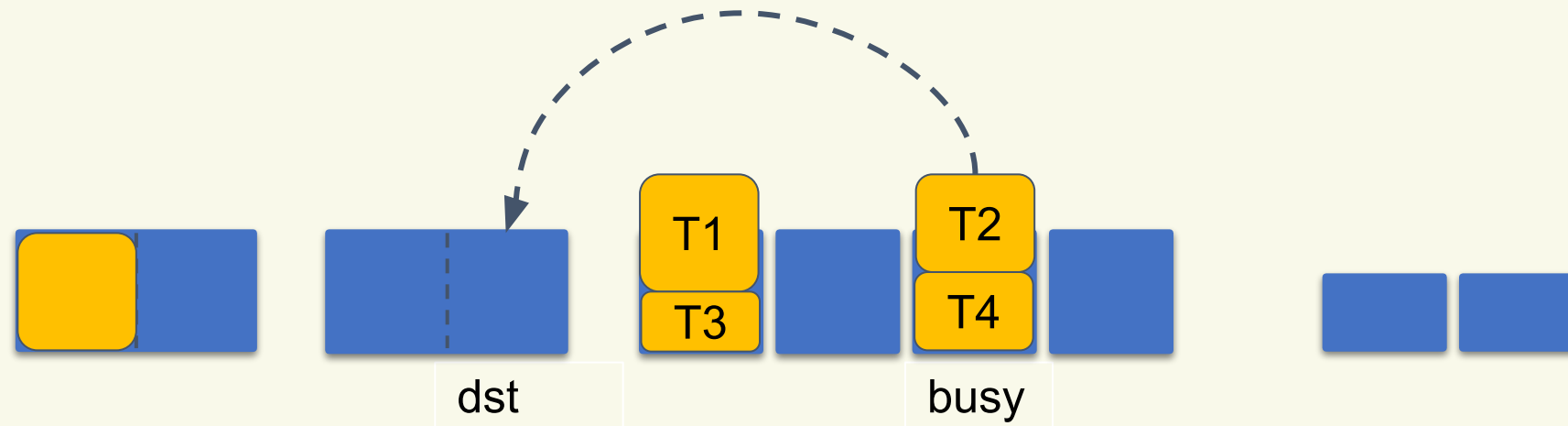


Uniform Instruction Set on all CPUs

ITMT: Idle Load Balance (Ecore -> Pcore)

When: some CPUs fully idle ($\#task \leq \#CPU$)

What: CPU "dst" enters idle, searches for "busy" to offload

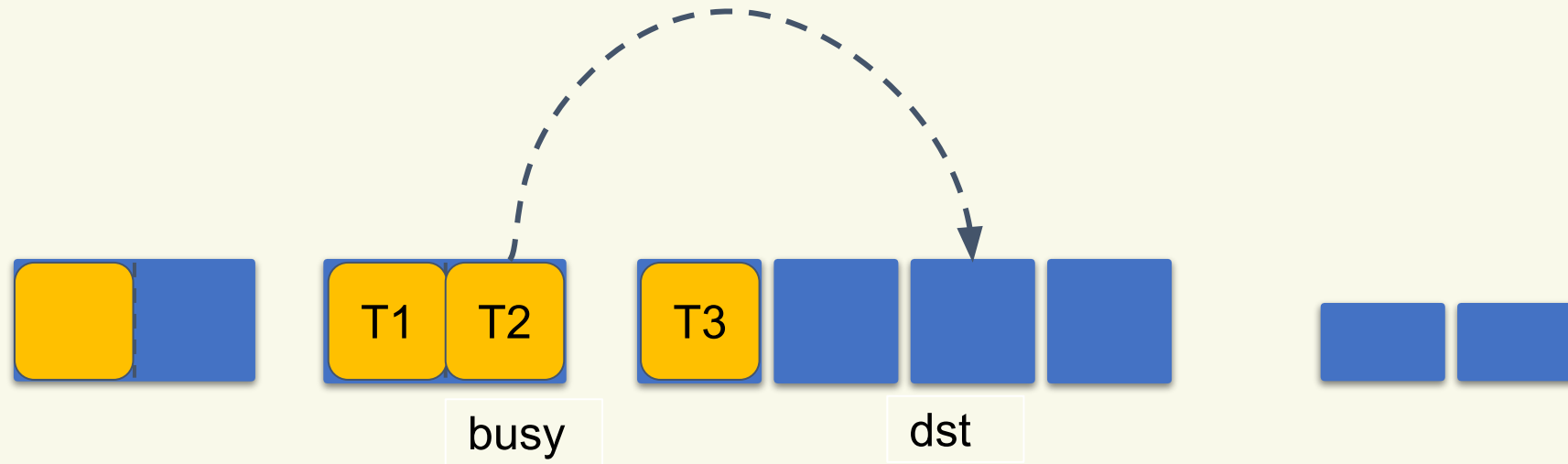


ITMT: fully-idle Pcore pulls from Ecore

ITMT: Idle Load Balance (HT -> Ecore)

When: some CPUs fully idle ($\#task \leq \#CPU$)

What: CPU "dst" enters idle, searches for "busy" to offload



ITMT: Ecore pull from busy-HT, always

Energy Performance Preference (EPP)

Hardware P-States (HWP) is enabled by intel_pstate driver.

Per-CPU MSR.HWP.REQ.EPP is architectural HW interface but...

EPP numeric Value is Model Specific

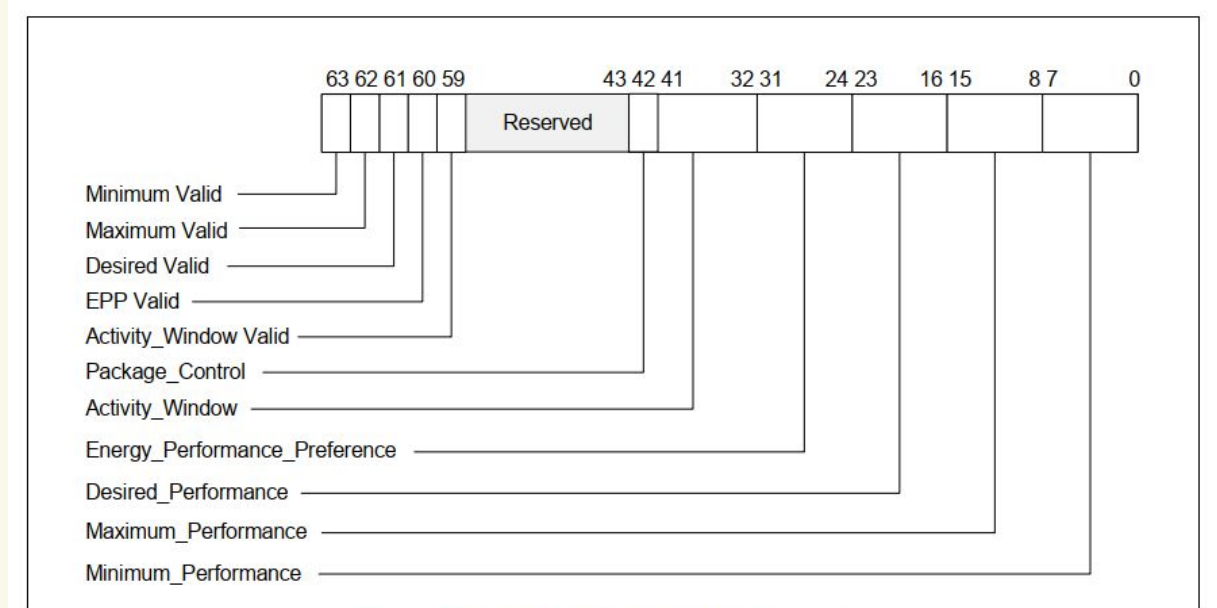


Figure 15-6. IA32_HWP_REQUEST Register



Linux intel_pstate driver maps Policy to EPP

```
/sys/devices/system/cpu/cpu0/cpufreq$ grep . energy*
```

```
energy_performance_available_preferences:
```

```
default
```

```
performance
```

```
balance_performance
```

```
balance_power
```

```
power
```

```
energy_performance_preference:
```

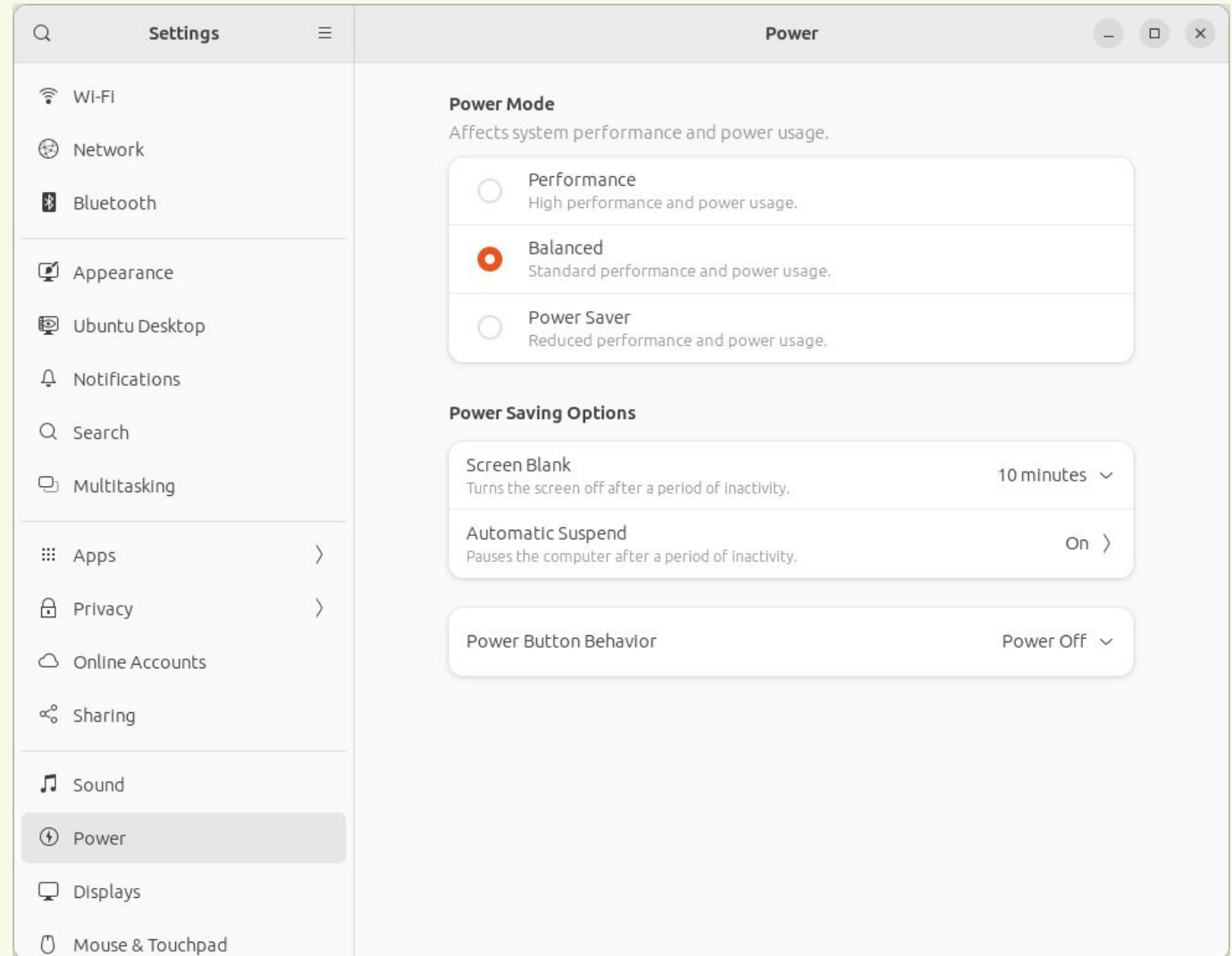
```
balance_performance
```



Linux Power Settings

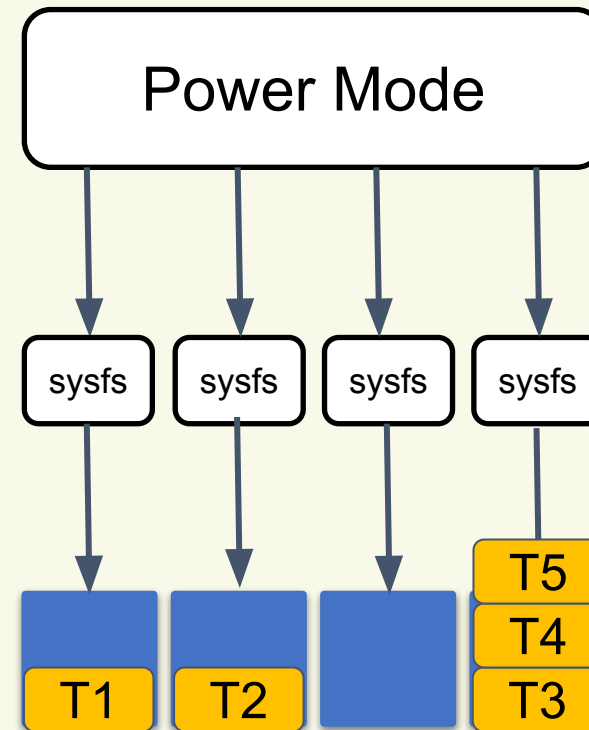
Power Mode
sets EPP
on all CPUs

(to the same static value)



Per-CPU EPP - Limited Utility

1. If all tasks have same need
2. If diverse task are bound



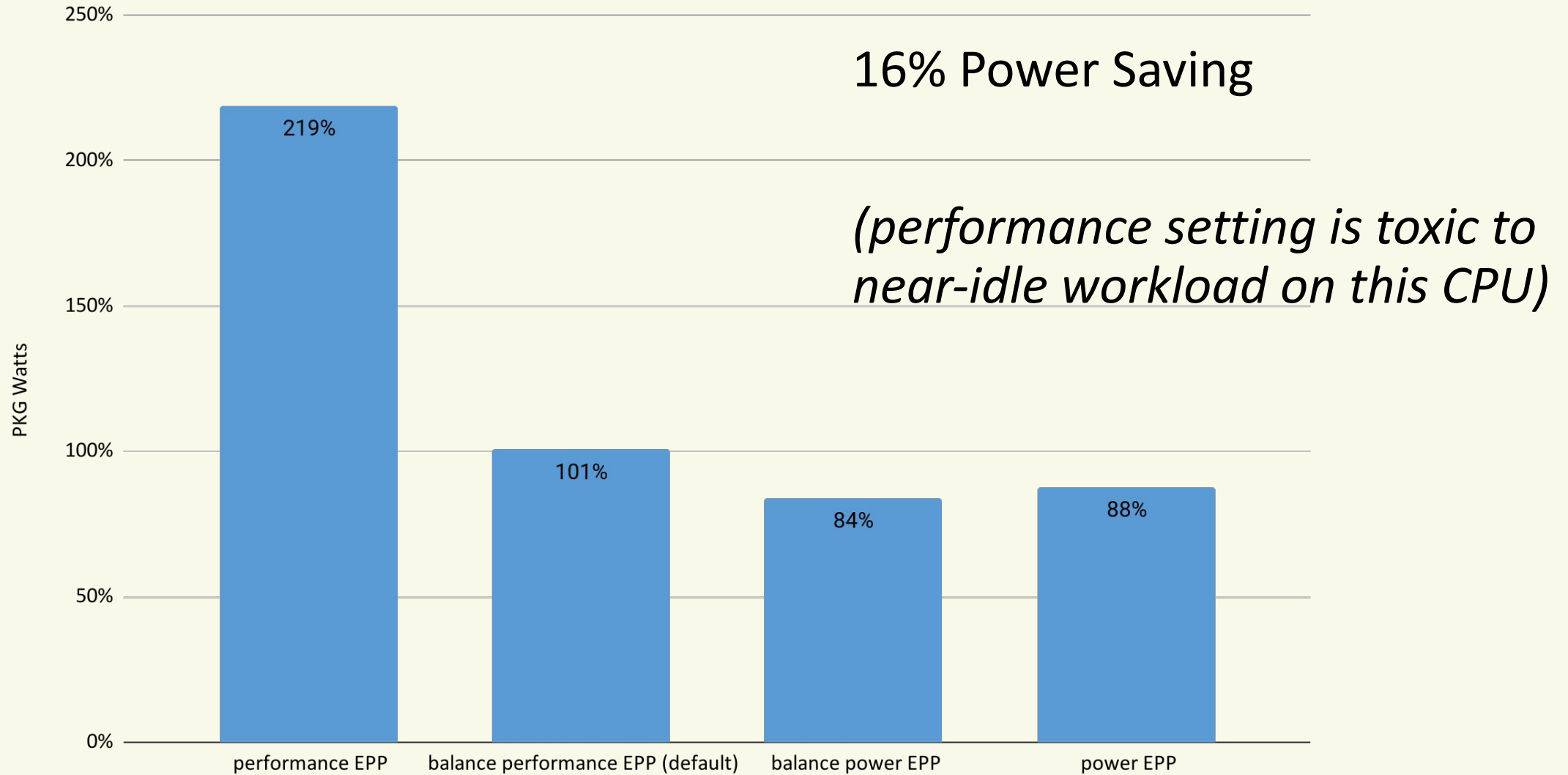
Chrome Browser Video Streaming Example

YouTube streaming 1080p 24fps video on chrome browser

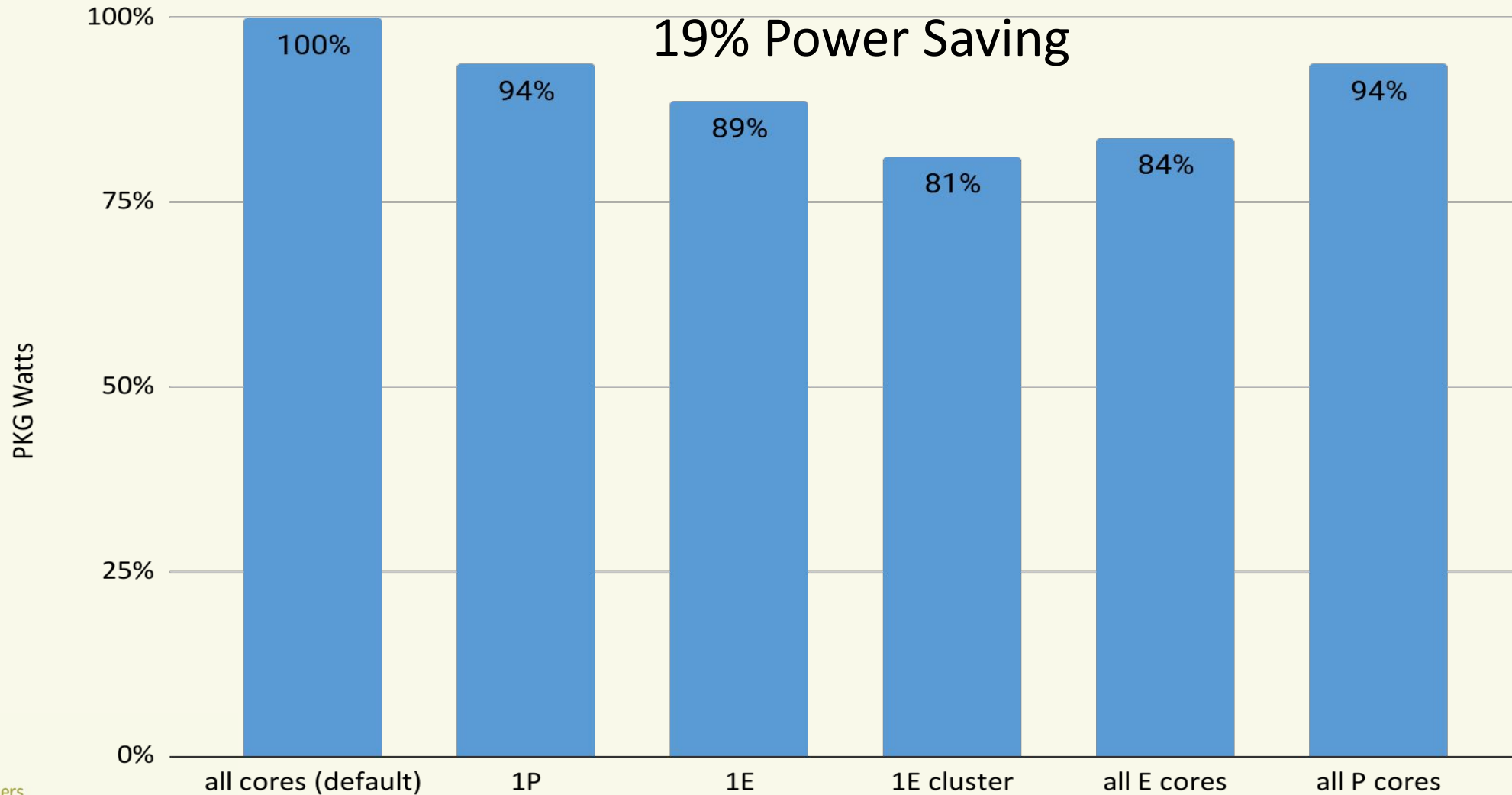
System: Intel Core i7-12700H 6+8 (45W)

1x Ecore can retire this workload w/o performance impact

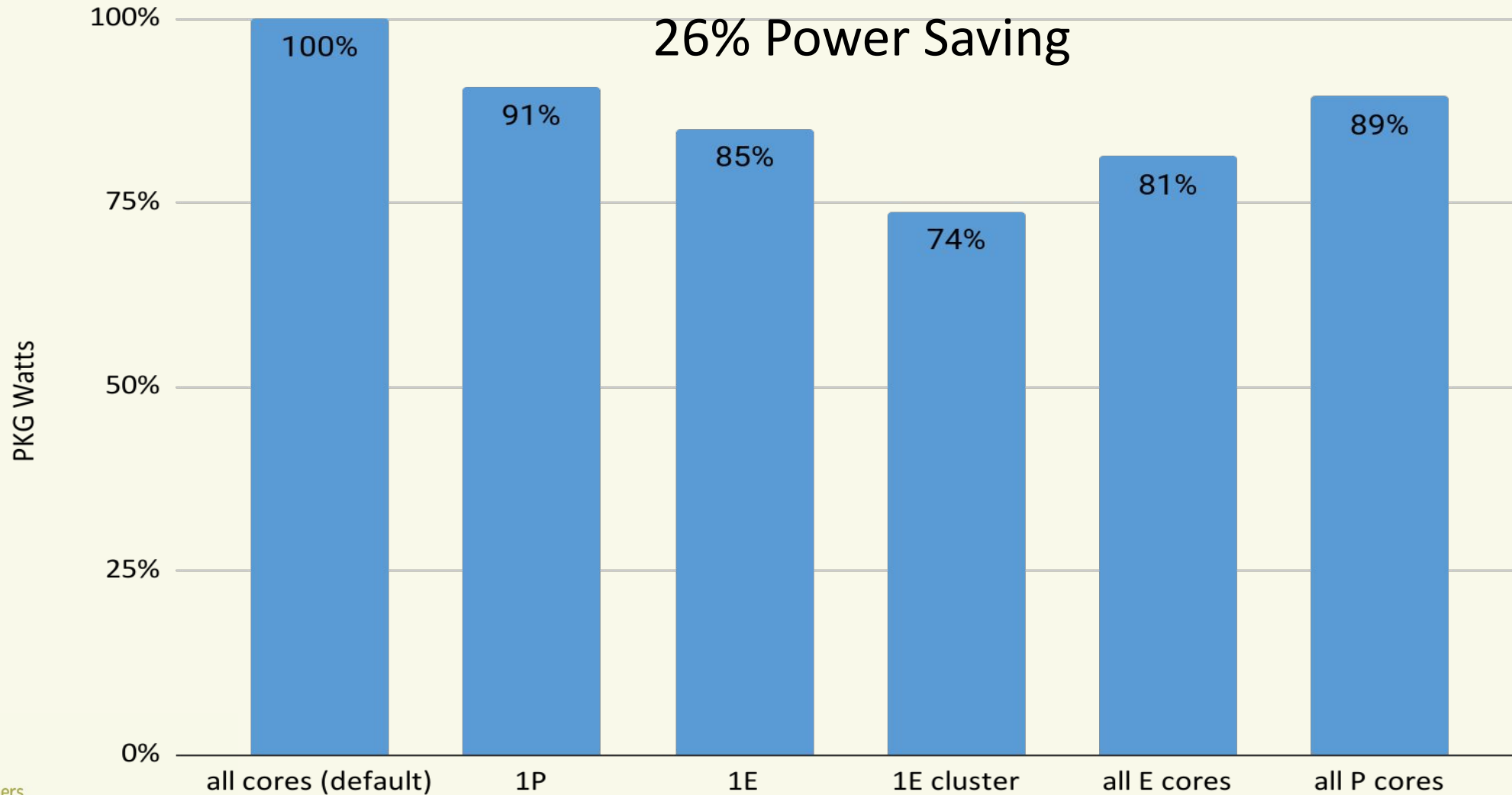
Video Streaming - All-CPU EPP Impact



Video Streaming - Task Binding Impact



Video Streaming - EPP + Task Binding Impact



Chromium Video Conferencing Example

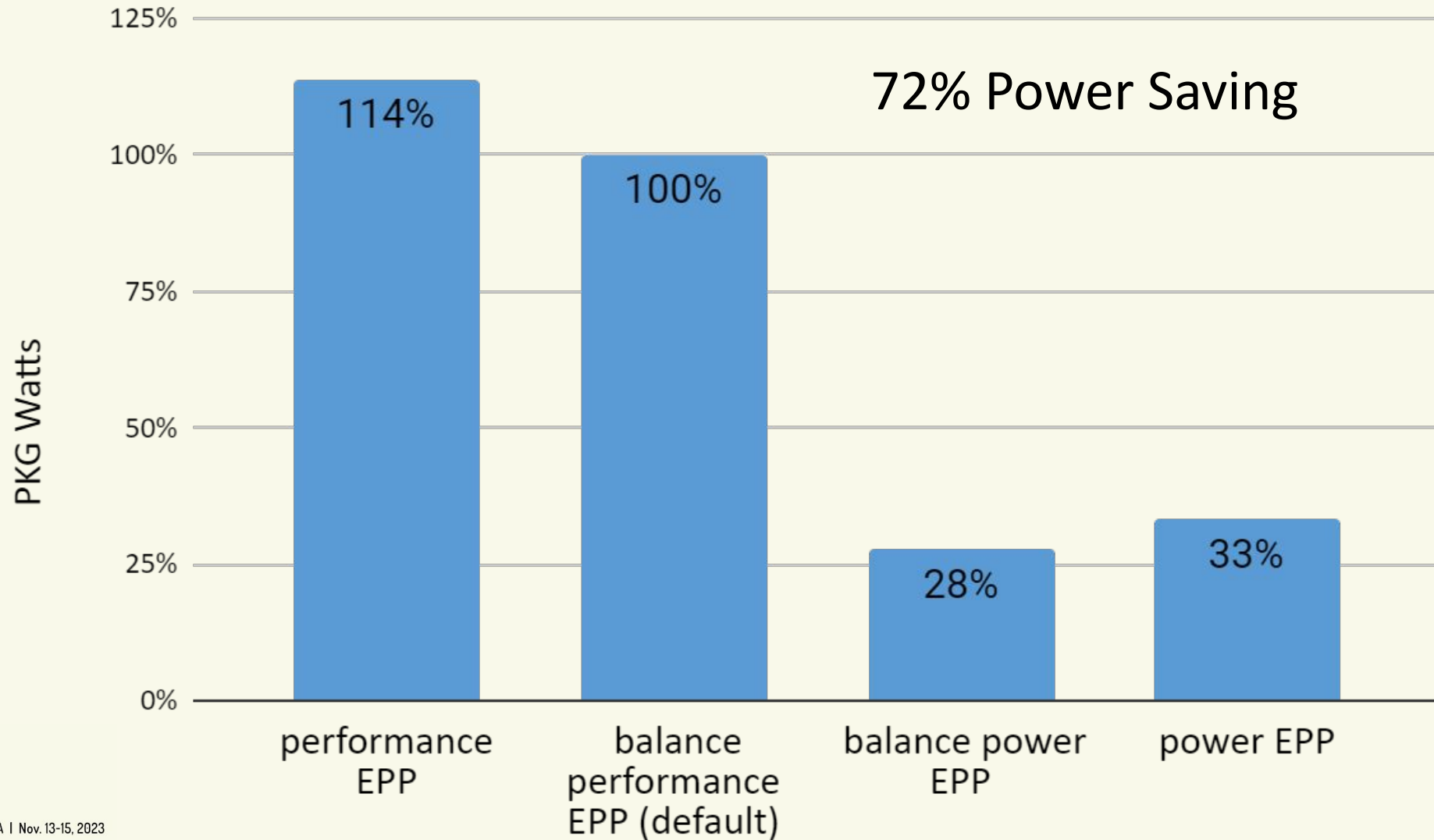
Google Video conference 2 party call

1080p 30fps (SW enc/dec)

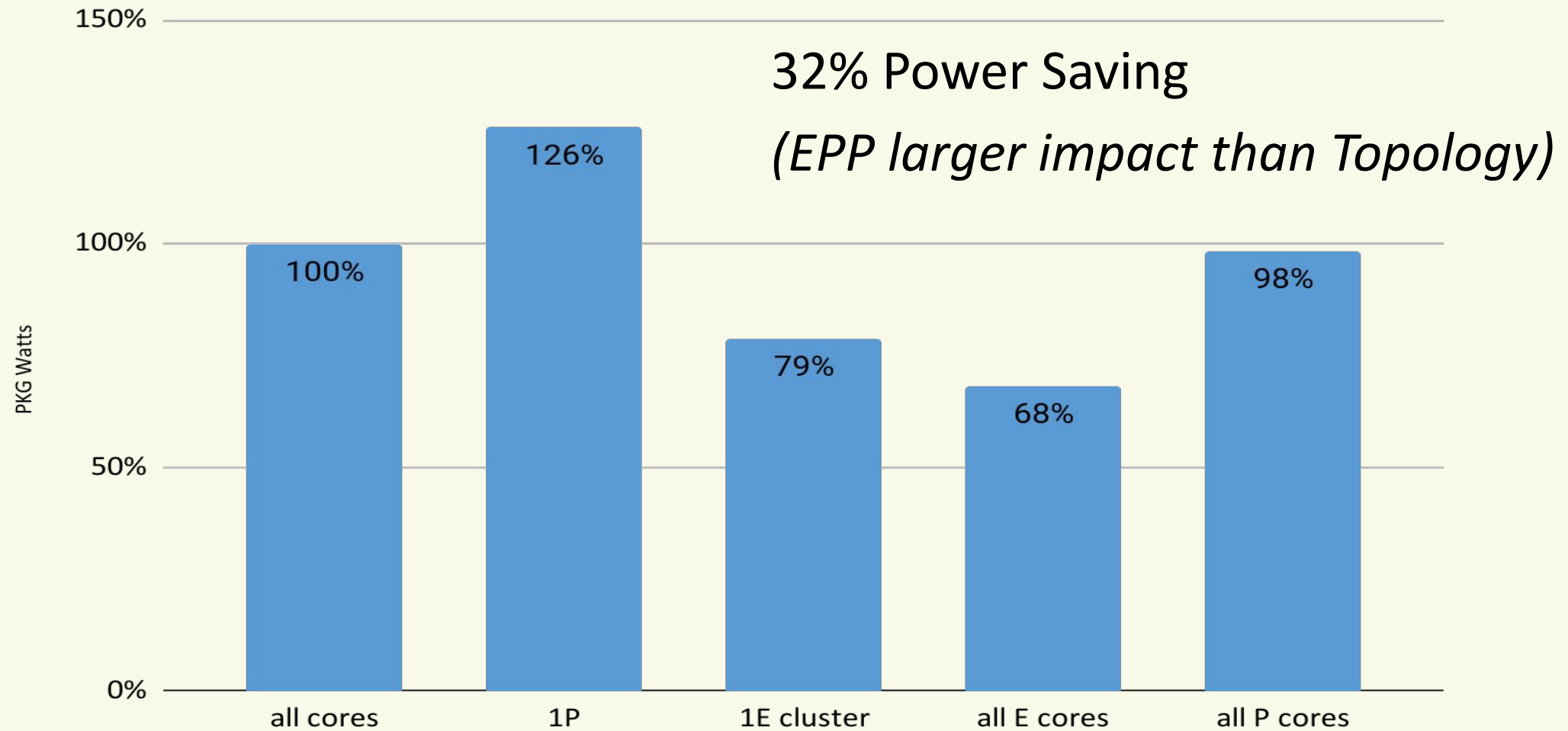
System: Intel Core i7-12700H 6+8 (45W)

> 1x Ecore required to retire workload w/o performance impact

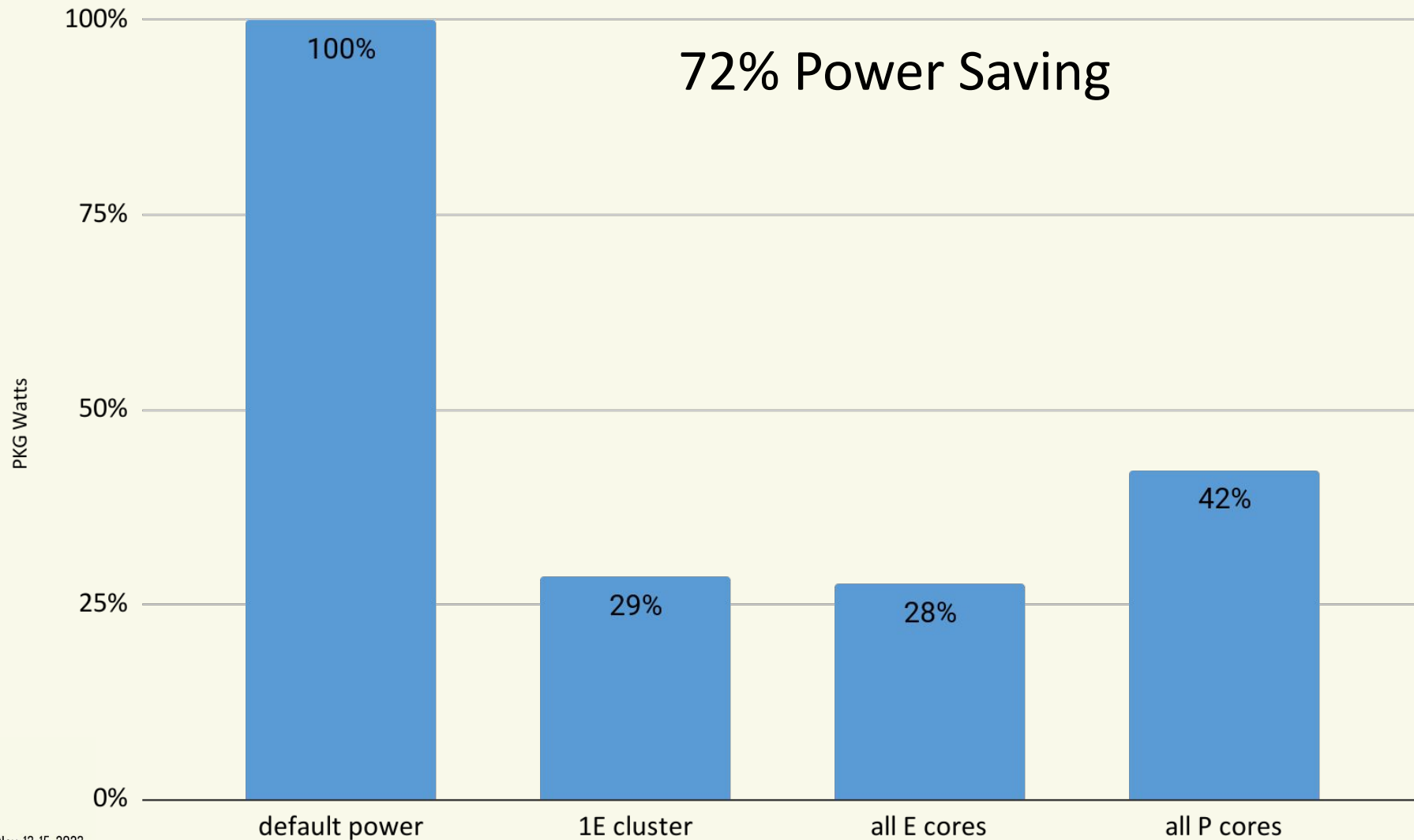
Video Conferencing - All-CPU EPP Impact



Video Conferencing - Task Binding Impact



Video Conferencing - All-CPU EPP + Binding



Is system wide orchestration the answer?

Sometimes yes.

For near system-wide idle.

When no risk to performance experience, can choose powersave

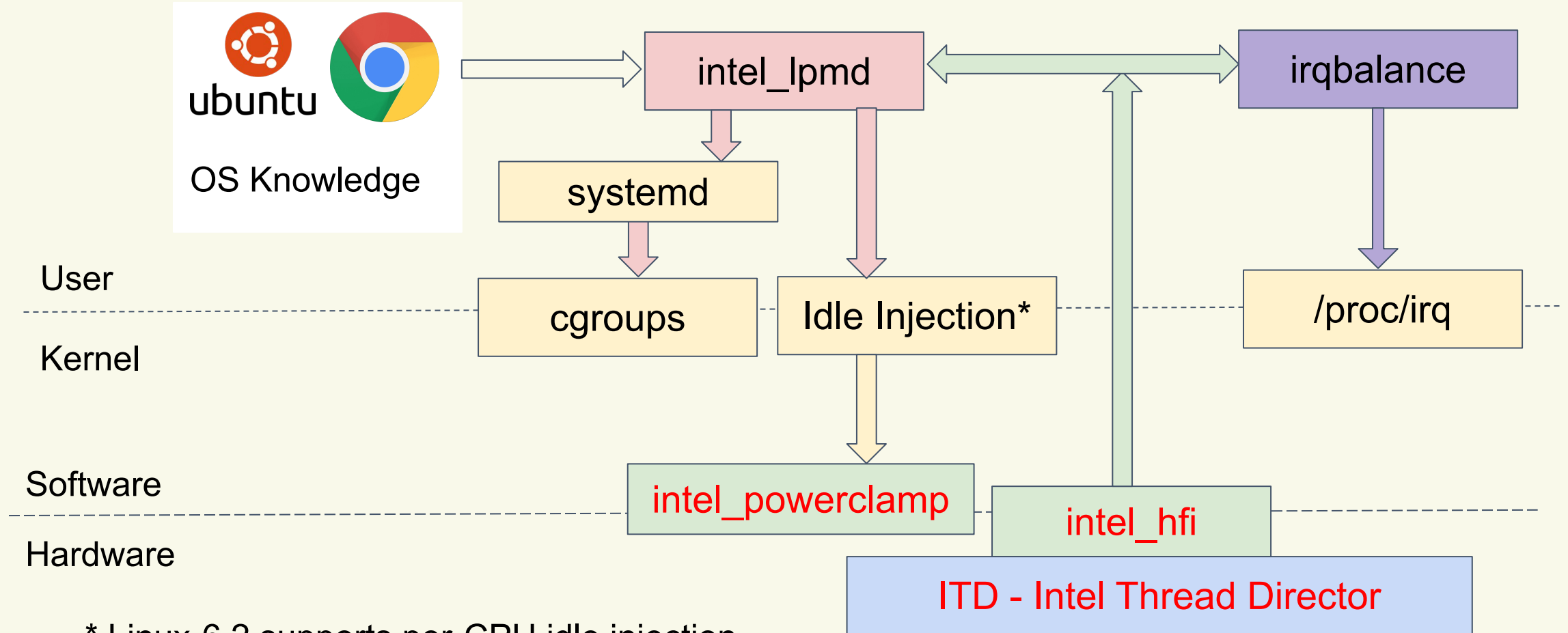
Low Power Mode (eg. Meteor Lake)

1. No tasks should run on Compute Die CPUs
2. No IRQs should wake Compute Die CPUs
3. How much work can "fit" on the LP CPUs?



LP Mode: Compute die is 100% idle, all work runs on SOC Die

Low Power Mode Policy in User-Space



* Linux-6.2 supports per-CPU idle injection



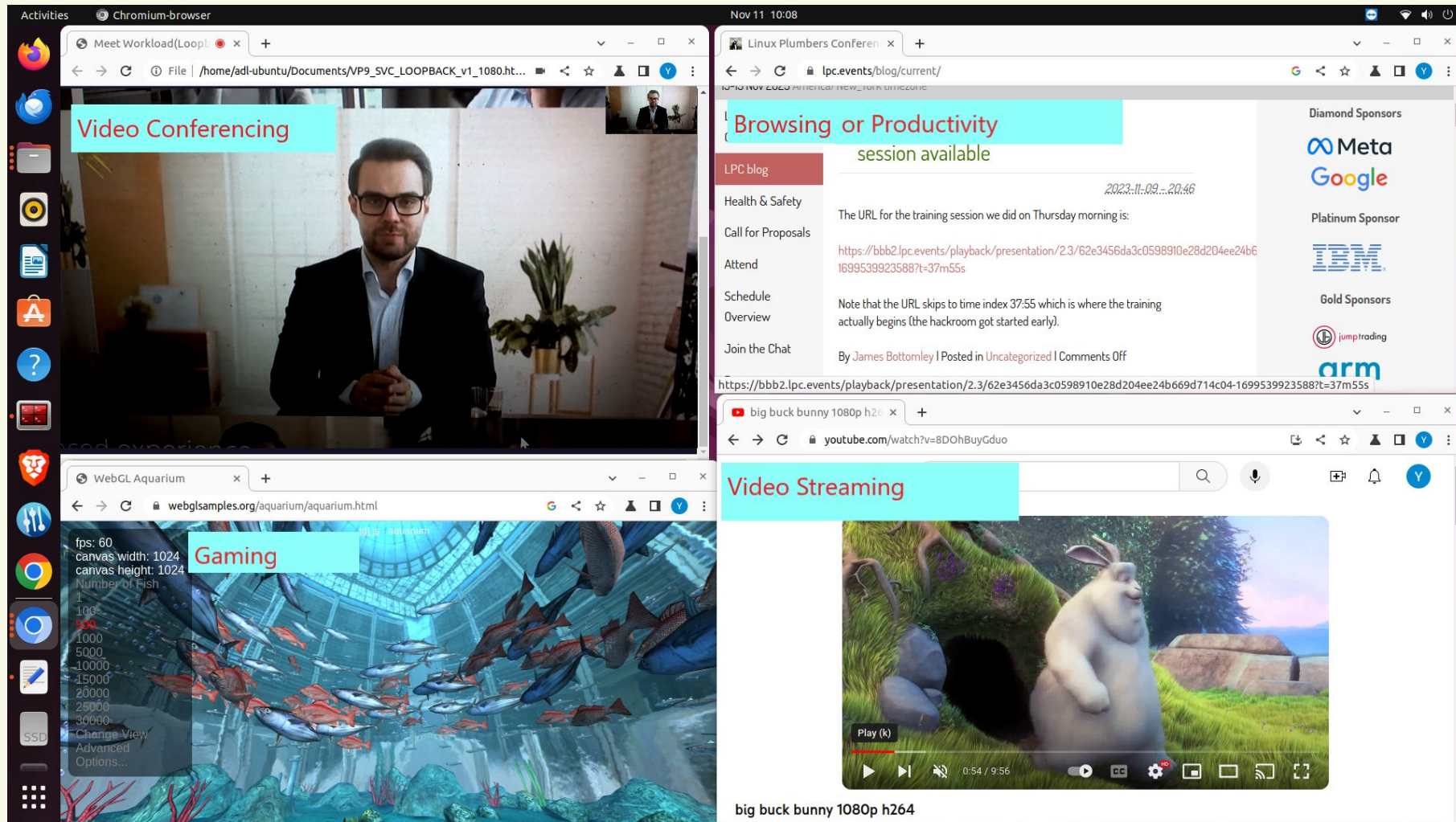
Is system wide orchestration the answer?

Sometimes no.

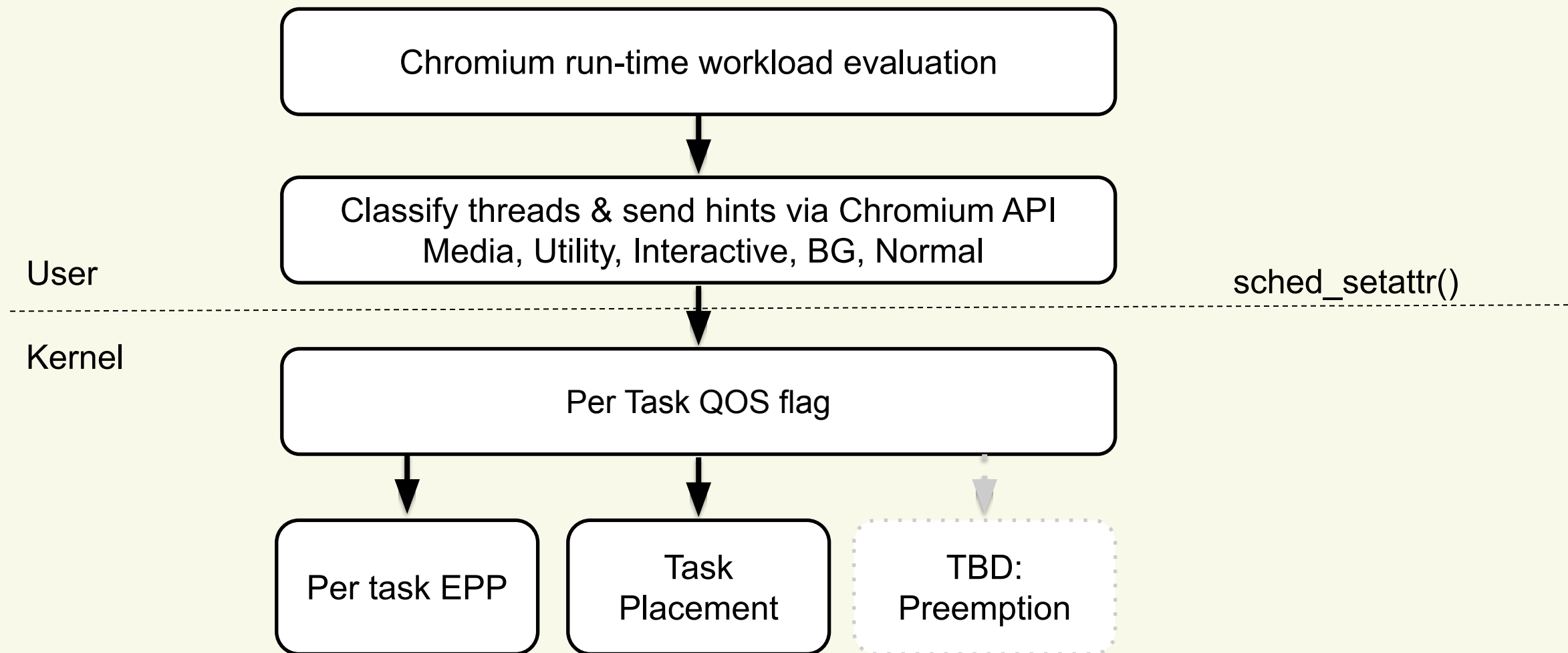
When diversity of task needs, especially within a process.

When load high enough that power-bias degrades performance.

Simultaneous Diverse Task requirements



Browser Requests per-task Power vs. Perf Policy

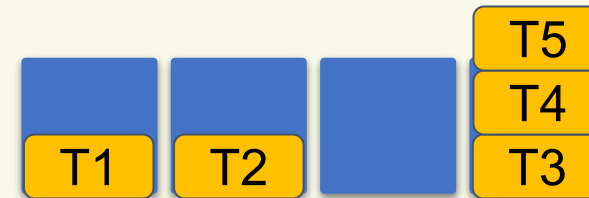


EQOS via per-task EPP

New QOS member in `task_struct`

Change EPP of CPU upon `sched_set_attr()` and context switch

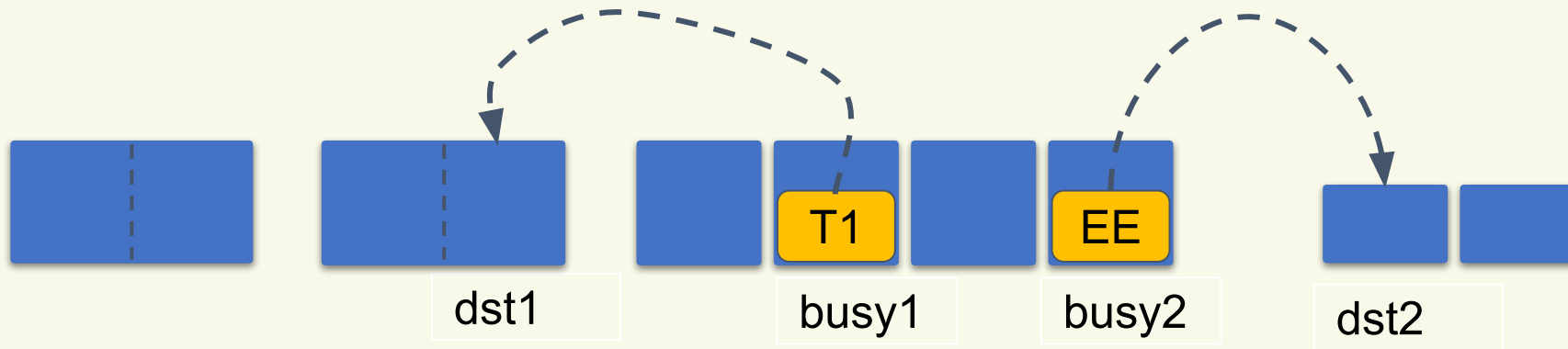
Yield to use of legacy `sysfs` interface



EQOS via Task Placement

When: partially idle

What: Default tasks prefer high priority, EE tasks prefer low priority

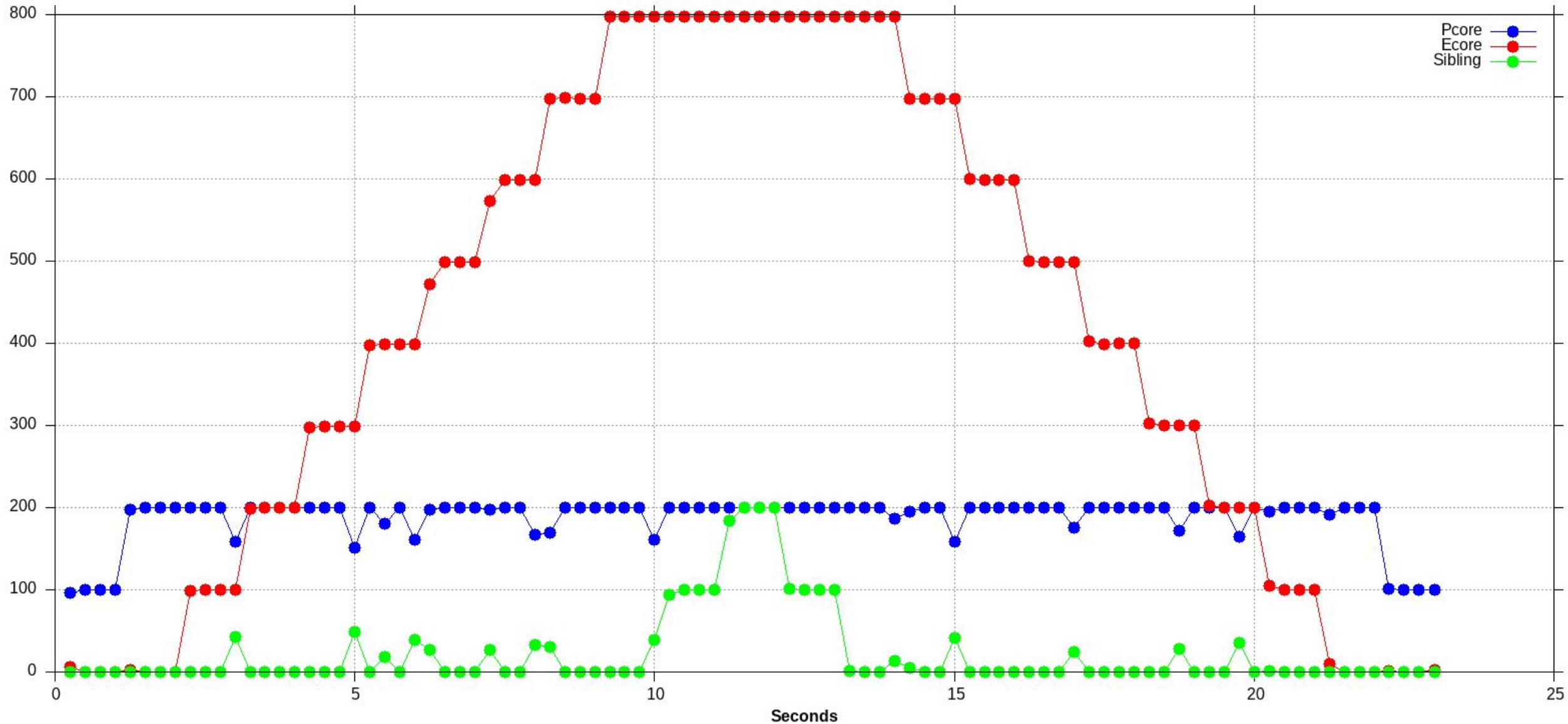


Utilization by CPU-Type

GETCPU@100

ADL-P_4 4xP@4.7 8xE@3.5 9W

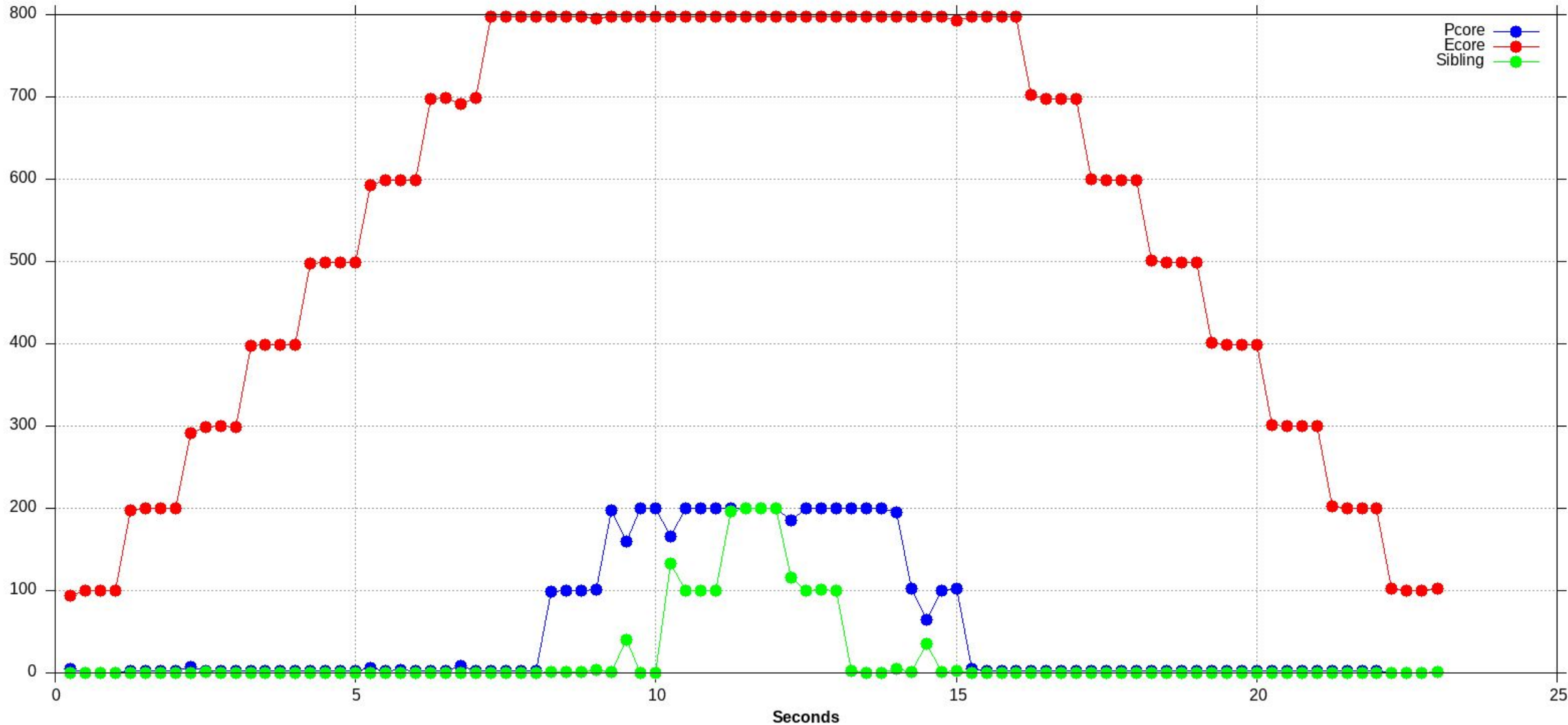
Pcore
Ecore
Sibling



Utilization by CPU-Type

GETCPU-ME@100

ADL-P_4 4xP@4.7 8xE@3.5 9W

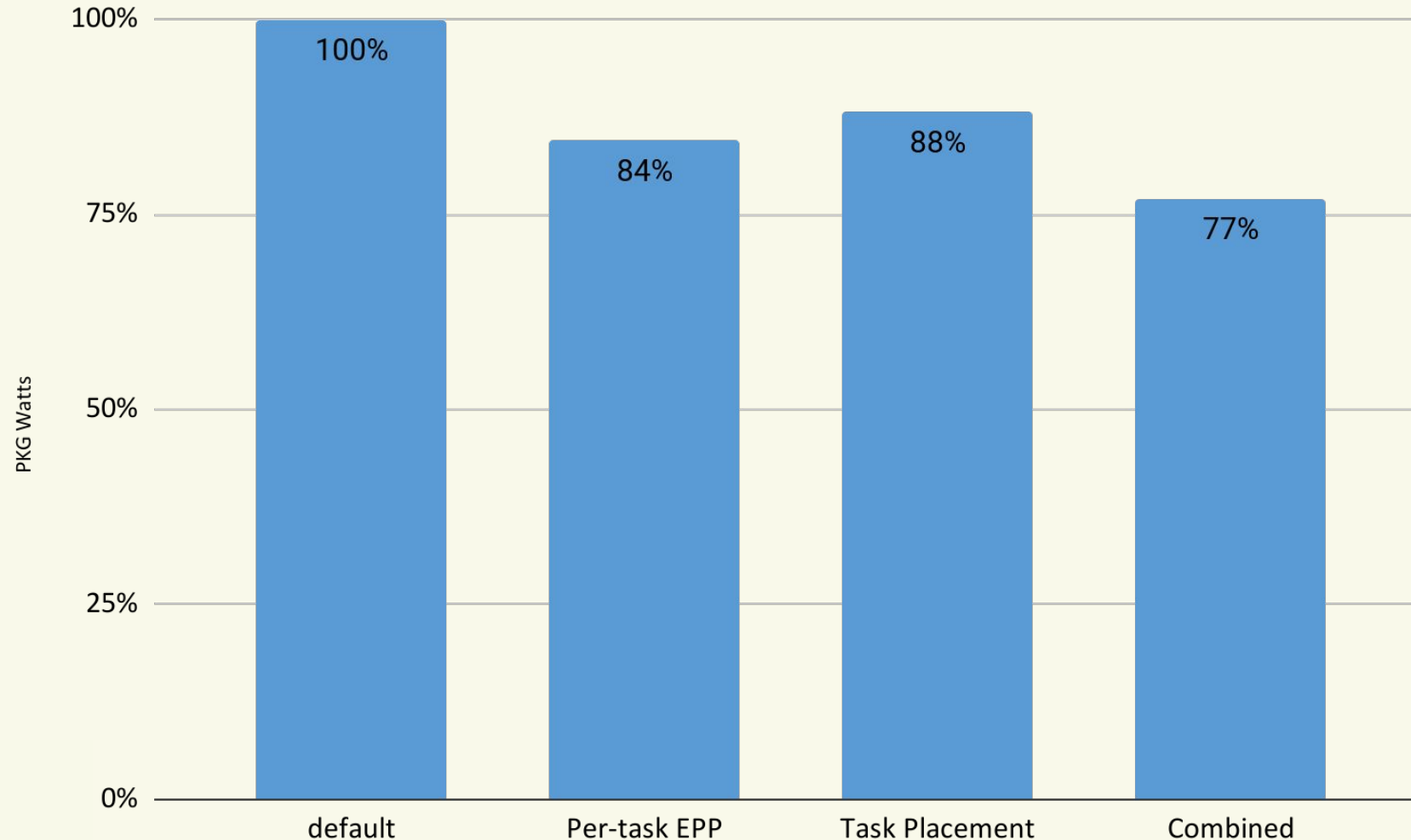


Chromium Demo using EQOS API

Statically assigned just Chromium's Render task to:
EQOS_BALANCE_EFFICIENCY

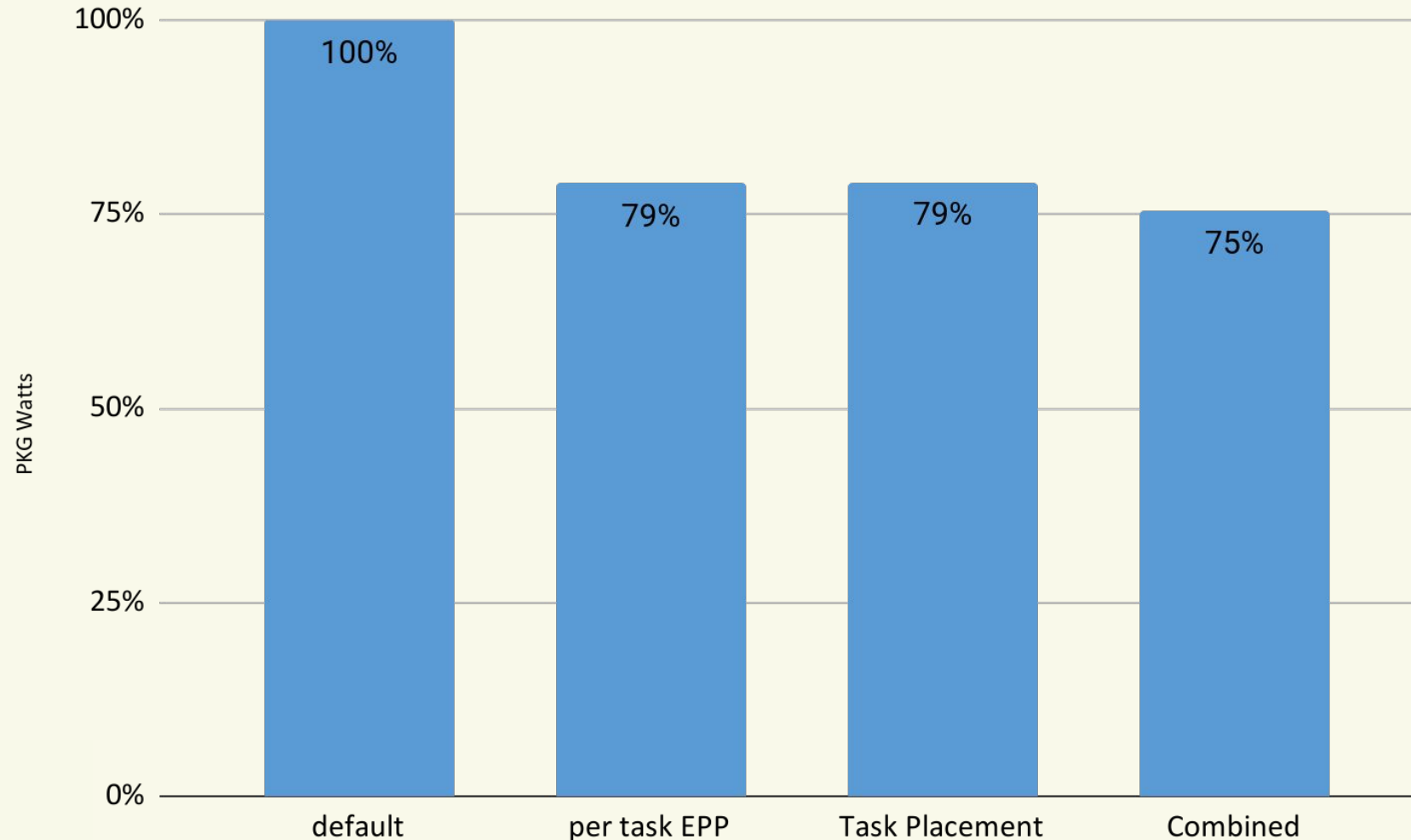
Chromium Video Streaming using EQOS API

23% Power Saving



Chromium Video Conf with EQOS API

25% Power Saving



Yet to Do

Chromium statically assigns QOS per-task type

Better done dynamically based on user experience
(eg. don't opt into EE if you are dropping frames)

Preemption can benefit both Perf and EE tasks.

Orthogonal to EQOS, likely complementary

Limitation: EQOS_LB depends on ILB

if no idle, then no ILB

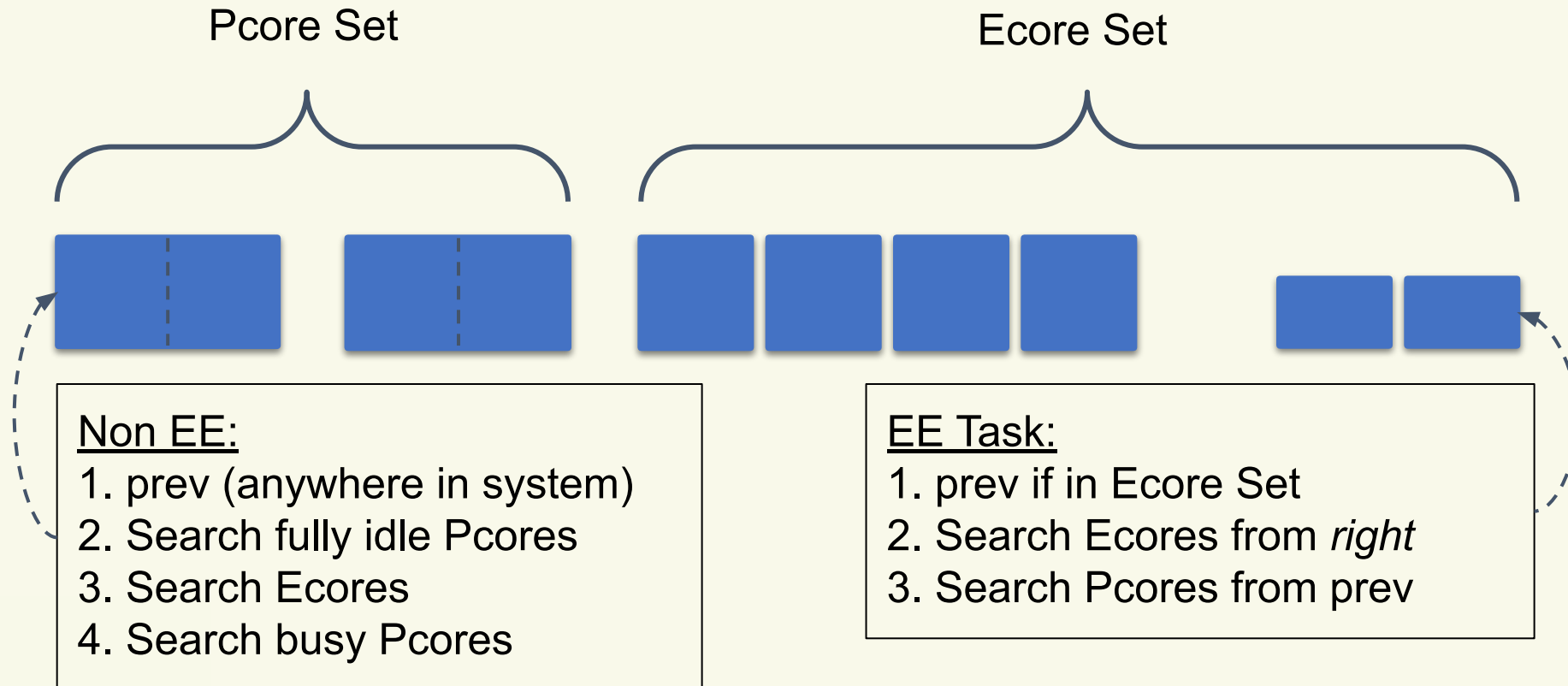
periodic LB

Short running tasks missed by ILB

Tasks could wake up on preferred CPU from the start
hook into the wakeup path

EQOS: Wake-up Idle CPU Search [RFC]

- Pcore Set (includes ITBM 3.0 "favored" cores)
- Ecore Set (includes Lcores)



Code is Here



<https://github.com/ricardon/linux/tree/rneri/eqos-demo>