



Contribution ID: 178

Type: **not specified**

## Optimizing Chromium Low-Power Workloads on Intel Notebooks

*Monday, 13 November 2023 10:35 (20 minutes)*

At LPC 2022 we hosted an Energy Quality of Service (EQOS) API discussion. The proposed API enables user-space to inform the kernel about something it is expert in: itself. Callers do not require any knowledge of the hardware, unrelated tasks, or the internal workings of the scheduler. The session sparked a lot of follow-on discussions, with the main take-away being “okay, so prototype it and demonstrate value.”

We will present a prototype with measurements demonstrating value. A web browser, updated to invoke the EQOS API, can save significant power during video playback and video conferencing workloads, without impacting performance when required.

The EQOS API achieves this by differentiating tasks that care about maximum performance, from those that are willing to accept performance impact in the name of energy efficiency (EE tasks). The EQOS API simply tells the kernel which is which. As the entire purpose of QOS is to differentiate tasks, the kernel saves the EQOS class in `task_struct`. This prototype uses that per-task EQOS class in two ways.

First, it takes advantage of Intel’s Hardware Performance-State “Energy Performance Preference” setting, effectively transforming this hardware knob from per-CPU to per-task. This allows the hardware to aggressively use high frequency for performance tasks, while being mindful of the energy cost of frequency for EE tasks.

Second, we tell the scheduler’s ASYM-PACKING feature to continue preferring high priority CPUs for performance tasks, but to prefer low priority CPUs for EE tasks. This has the double benefit of getting the EE tasks off the limited high-performance CPUs, at the same time as retiring the EE tasks at more energy-efficient operating points.

The concept of per-task EQOS class is agnostic to the underlying hardware architecture, and the scheduler is free to use (or ignore) this hint differently on different hardware.

Unaddressed... Latency preferences are orthogonal to EQOS. Perhaps if we had a Latency-QOS hint, we could tell the scheduler when a task wants to run efficiently, but promptly.

**Primary authors:** BROWN, Len (Intel Corporation); NERI, Ricardo (Intel Corporation); Mr SHANKAR, Vaibhav (Intel Corporation)

**Presenters:** BROWN, Len (Intel Corporation); NERI, Ricardo (Intel Corporation); Mr SHANKAR, Vaibhav (Intel Corporation)

**Session Classification:** Real-time and Scheduling MC

**Track Classification:** LPC Microconference: Real-time and Scheduling MC