



Contribution ID: 84

Type: **not specified**

Can ProxyExec and sched_ext get along?

Proxy Execution provides a generalized form of priority inheritance, which leaves mutex-blocked tasks on the run-queue. Then if the scheduler tries to run a mutex-blocked task, it will instead run the mutex owner on the blocked task's behalf, so the mutex can be released.

In order for this to work, we introduced the idea of split contexts (scheduler and execution), tracking both the task that was initially selected to run (the donor - scheduler context), and the task that actually runs (the execution context).

However, running tasks on behalf of other tasks has the potential to confuse sched_ext schedulers. So currently they are mutually exclusive via Kconfig.

I'm currently looking into what promises are made to sched_ext schedulers, and what might be needed to ensure sched_ext schedulers can be used along with Proxy Execution. At the talk, we'll discuss what I have found and hopefully potential solutions.

Discussion Bait: I'm very much hoping that by merging sched_ext, and exposing its interfaces to BPF, we haven't effectively frozen the scheduler's pre-Proxy Exec behavior as a user ABI.

Primary author: STULTZ, John (Google)

Presenter: STULTZ, John (Google)

Session Classification: sched_ext: The BPF extensible scheduler class MC

Track Classification: sched_ext: The BPF extensible scheduler class MC