



Contribution ID: 87

Type: **not specified**

scx_chaos: Pushing the sched_ext API to schedule badly

sched_ext has guardrails in kernel and lots of examples in BPF for how to schedule tasks effectively. We use sensible defaults for idle tracking, NUMA aware masks, and prevent you losing track of tasks in BPF. But what happens when you try to schedule badly?

scx_chaos builds on top of scx_p2dq, another sched_ext scheduler. It adds options for introducing delays, randomly decreasing CPU frequency, inverting mutexes, and slowing down kprobes. This helps with hunting down race conditions or points of contention, and reproducing the conditions that create them. In this MC session we will discuss, with examples:

- What classes of bugs can we fix in the kernel with a chaotic scheduler?
- What can we learn from building a scheduler on top of another?
- What makes writing chaotic features in sched_ext challenging?
- Can the kernel selftests survive scx_chaos?

Primary author: HILLION, Jake (Meta)

Presenter: HILLION, Jake (Meta)

Session Classification: sched_ext: The BPF extensible scheduler class MC

Track Classification: sched_ext: The BPF extensible scheduler class MC