Linux Plumbers Conference 2025

Contribution ID: **57** Type: **not specified**

Optimizing GPU bound workloads with sched_ext via scx_layered

Optimizing GPU bound workloads with sched_ext via scx_layered

In this talk, I will discuss how to optimize GPU bound workloads through the use of the sched_ext scheduler, scx_layered and how API changes could make make this simpler.

I will use a well understood open source GPU benchmark job (something like mnist or resnet) and a common cpu-bound open source workload (something like compiling the linux kernel, or chromium) to demonstrate how workload-customized scheduling policies, such as those which scx_layered enables the use of, can be leveraged to optimize workload run time and system resource utilization.

After this brief overview, I will highlight the challenges of optimizing workloads such as this encountered while working on scx_layered, and ways in which improved APIs and/or tooling could simplify use cases such as this.

Some particulars I would like to discuss with kernel developers are the following:

- If there could be a better way to confirm verifiability of scheduler code across a range of kernels/hardware types other than running scheduler code on hardware/kernel combinations.
- If there could be APIs enabling easier association of TIDs with GPU devices or NUMA nodes.
- If there could be an API kernel side enabling setting mempolicy from the scheduler.

Primary author: SOMARU, pat

Presenter: SOMARU, pat

Session Classification: sched_ext: The BPF extensible scheduler class MC

Track Classification: sched_ext: The BPF extensible scheduler class MC