



Contribution ID: 271

Type: **not specified**

Automated Workload-Aware Scheduling with SCX to Improve Performance

Applications can greatly benefit from a workload-aware scheduling policy of worker threads that optimizes cache usage. For example, if a scheduling policy is aware of a workload's data access patterns, it can make informed decisions on how to schedule threads to cores to take advantage of cache locality. However, a key technical challenge is achieving this in a workload-agnostic manner.

The focus of our work is to automate the three-step process of (1) identifying patterns from the workload's data access profile, (2) using these patterns to craft a scheduling strategy, and (3) applying that strategy to schedule worker threads. To do this, we first use standard tools (such as perf) to collect the workload's data access pattern and represent it as a co-occurrence graph. The vertices of this graph correspond to specific components of the workload (such as collections of worker threads), and the weight of the edges denotes the degree of data-access overlap between these components. Using this structure, we apply standard optimization algorithms (such as balanced graph partitioning or clustering) to develop a scheduling policy that colocates threads with similar access patterns on the same cache level, such as L1, L2, or L3 caches.

As a proof of concept, we demonstrated this end-to-end on a synthetic workload, achieving a roughly 1.8x speedup in runtime using our approach. We are currently applying this method to a production workload and have found the early results encouraging.

Primary authors: NEWELL, Andrew (Meta); KHAN, Arif (Meta); DWIVEDI, Kumar Kartikeya (Meta); KUMAR, Neeraj (Meta); SURULIKAMU, Ravi (Meta); HEO, Tejun (Meta)

Presenter: KUMAR, Neeraj (Meta)

Session Classification: sched_ext: The BPF extensible scheduler class MC

Track Classification: sched_ext: The BPF extensible scheduler class MC