



Contribution ID: 92

Type: **not specified**

Scheduler composability with arena-based data structures

This talk will present our progress on arena-based data structures for quickly evolving scheduler abstractions (DSQs, CPU topology).

We currently write scheduling algorithms in terms of operations on primitives provided by the kernel (BPF hash maps/arrays, CPU bitmasks, DSQs). Adding new operations to these primitives is work-intensive because it requires modifying the underlying kernel data structure. It also possibly changes the sched-ext BPF API, breaking backwards compatibility. The workaround has been to open-code this functionality in BPF, often at a performance cost. For example, finding and removing a task from a DSQ requires linearly traversing it first.

BPF arenas provide a framework for developing and evolving scheduler abstraction. BPF arenas enable writing recursive data structures directly as BPF libraries and lift the limitations imposed to developers for regular BPF code. These arena-based data structures let us introduce and evolve scheduler abstractions without kernel modifications.

This talk will go over:

- Status of arena-based data structures, both general and scheduler-specific
- Alternative DSQ designs, APIs, and implementations for arena-based data structures:
- Specific instances of arena-based abstractions
 - Flexible arena-based task queues (ATQs)
 - Representing the machine topology
 - Representing affinity groups, load balancing, and task stealing

Primary author: TSALAPATIS, Emil (Meta Platforms)

Presenter: TSALAPATIS, Emil (Meta Platforms)

Session Classification: sched_ext: The BPF extensible scheduler class MC

Track Classification: sched_ext: The BPF extensible scheduler class MC