
Contribution ID: **424** Type: **not specified**

# Limiting Memory Use of Userspace Per-CPU Data Structures in Containers

*Thursday, 19 September 2024 18:15 (15 minutes)*

- New machines with 512+ hardware threads (and thus logical CPUs) bring interesting challenges for user-space per-CPU data structures due to their large memory use.
- The RSEQ per-memory-map concurrency IDs (upstreamed in Linux v6.3) allow indexing user-space memory based on indexes derived from the number of concurrently running threads,
- I plan to apply the same concept to IPC namespace,
- This provides memory use upper bound when limiting containers with cpusets (e.g. cpuset: 0-31),
- It does not work when limiting containers that have many threads with time slices (e.g. cpu.max 2000 1000),
- Cpusets are far from ideal to describe the constraints in a cloud-native way:
- those are bound to the machine topology,
- hard to compose containers expressed with cpuset constraints,
- tricky with big.LITTLE, p-core/e-core CPUs.
- Use-cases: userspace tracing ring buffers, userspace memory allocators (e.g. tcmalloc), statistics counters.
- Discuss proposal: Introduce a new cpu.max.concurrency interface file to the cpu controller, which defines the maximum number of concurrently running threads for the cgroup. Track the number of CPUs concurrently used by the cgroup. Extend the scheduler to prevent migration when the number of concurrently used CPUs is above the maximum threshold.

**Primary author:** DESNOYERS, Mathieu (EfficiOS Inc.)

**Presenter:** DESNOYERS, Mathieu (EfficiOS Inc.)

**Session Classification:** Containers and checkpoint/restore MC