Remote Charging in the CPU Controller

Linux Plumbers / Scheduler MC
20 September 2021

Daniel Jordan <daniel.m.jordan@oracle.com>

The setup

- CPU-intensive kernel threads generally escape CPU controller limits

- Use cases
  1. padata multithreaded jobs (VFIO page pinning, struct page init for memory hotplug, page clearing)
  2. async memory reclaim (kswapd, cswapd)
  3. net rx (softirq handler)
  4. kworkers on Android, but no specifics

The setup

- Requirements
  - Shouldn’t throttle high priority work of reclaim and net rx
  - Don’t know the task group ahead of time for net rx

- “Obvious” but unworkable solutions
  - cgroup-aware workqueues
  - Kernel path for `CLONE_INTO_CGROUP`

The idea

Focused on weight now, bandwidth to come

- Incur debt with interface for kernel threads
  - kthread runtime accounted to the target task group and all its parents

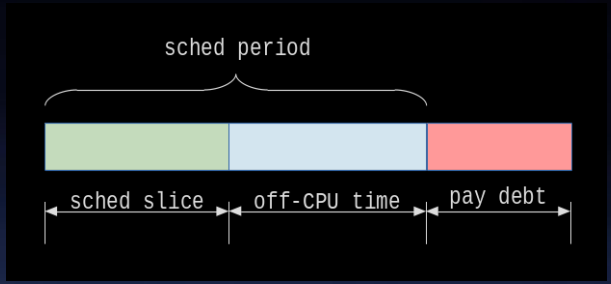
```
void cpu_remote_charge_begin();

void cpu_remote_charge_account(struct cgroup *cgroup);
void cpu_remote_charge_end(struct cgroup *cgroup);
```

- Pay debt by increasing vruntime of group entities
  - `vruntime` increased in `put_prev_entity()` for entity going off-CPU
  - Competing entities on the `cfs_rq` are required for debt to matter
  - Forgive debt if the system is idle

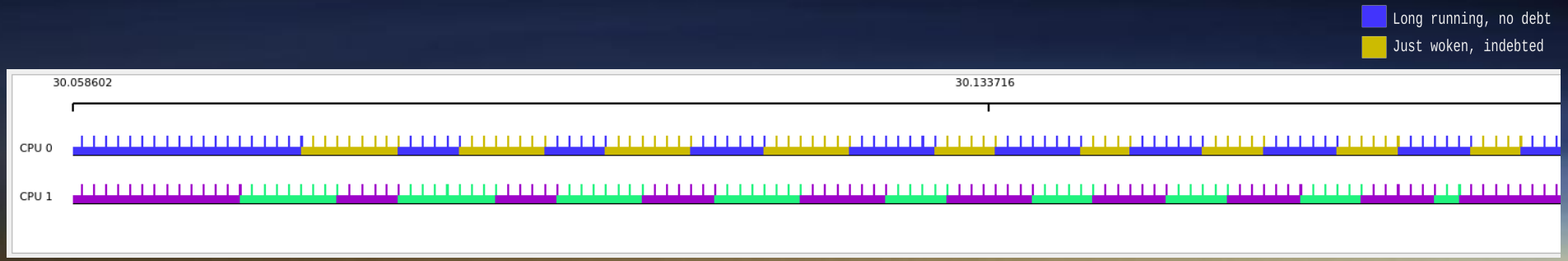
The idea

- Save any unpaid debt
  - On dequeue of an indebted group entity
  - On dequeue where `cfs_rq→nr_running` falls to 1 (no more competition)
  - When an indebted entity goes on-CPU (`set_next_entity()`)



Current status

- Done
  - Prototype that incurs, pays, and saves debt
  - Correctness testing on synthetic CPU-bound loads



- TODO
  - Correctness testing on realistic loads
  - Evaluate performance impact on the scheduler
  - Add support for CFS bandwidth
  - Debt forgiveness

Discussion points

- Under what conditions should debt be incurred?
  - Incorrect to incur on a sufficiently idle system, yet global state hard to come by in CFS
- Where should debt be stored?
  - Globally in `struct task_group`
  - Locally in `struct sched_entity`
- Limitations of using vruntime to pay debt
  - Debt can’t be paid if a task group doesn’t compete
  - Debts cancel if indebted groups aren’t scheduled with debt-free groups

Remote Charging in the CPU Controller

Linux Plumbers / Scheduler MC
20 September 2021

Daniel Jordan <daniel.m.jordan@oracle.com>