Contribution ID: **122** Type: **not specified**

Core Scheduling feature Upstreaming Plans

Tuesday, 25 August 2020 07:00 (50 minutes)

As a follow up to the OSPM discussion, we would like to discuss the upstreaming plans.
As per the OSPM, the work left to be done were:
1. Documentation.
2. Cross cpu, vruntime comparison logic for CFS tasks.
3. Kernel protection from sibling during Syscall and interrupts.
4. Load balancing fixes.
5. API and usage.
6. Hotplug fixes.
7. Other fixes and code cleanup.

Now, v6 is released and we have made good progress. Documentation is mostly done and code has been cleaned up as per discussion at OSPM. Kernel protection from siblings during syscall and interrupts is also complete (pending posting and review). Hotplug fixes are also ready(pending posting and review). We need to work on vruntime comparison and load balancing fixes. Also API needs to be finalized.

The plan that we propose is to have a phased upstreaming approach. The code now could be considered almost feature complete, but with some known bugs. We propose to upstream the current code after a thorough review and then work on the known bugs aiming to get them in shortly there after:
1. Vruntime comparison.
2. Load balancing fixes.
3. Uperf regression reported by Aubrey (ksoftirqd getting force-idled).

The feature will be default-disabled on SMT systems and will be marked as experimental until all these known issues are fixed.

API is the other major thing. We have couple of different API proposals during OSPM/in mailing list, but did not reach a consensus:
1. Coresched specific cgroup.
2. prctl/sched_setattr.
3. Sysfs interfaces.
4. Auto tagging based on process properties(user, group, VM etc).
5. Trusted cookie value (We can make 0 as default, and auto tag everything on fork.

The current API of cpu cgroups might not be worth upstreaming. We could either have the first phase go in without any API(not usable without out of tree patch) and then get API in soon after, or have a simple auto tagging interface(all tasks/processes under a separate tag, etc) in the first phase.

So, we propose 4 sessions:
1. Discuss vruntime priority comparison.
2. Discuss load balancer issue.
3. Discuss API.
4. Discuss upstreaming.

I agree to abide by the anti-harassment policy

I agree

Primary authors: DESFOSSEZ, Julien (DigitalOcean); FERNANDES, Joel; REMANAN PILLAI, Vineeth (DigitalOcean)

Presenters: DESFOSSEZ, Julien (DigitalOcean); FERNANDES, Joel; REMANAN PILLAI, Vineeth (DigitalOcean)

Session Classification: Scheduler MC

Track Classification: Scheduler MC