Subject: Re: [RFC][PATCH 0/6] Add group fairness to CFS - v1
Posted by Srivatsa Vaddagiri on Tue, 12 Jun 2007 10:56:37 GMT
View Forum Message <> Reply to Message

[resending ..my earlier reply doesn't seem to have made it to lkml]

On Tue, Jun 12, 2007 at 08:26:12AM +0200, Ingo Molnar wrote:
> > So where's this precise stats based calculation of cpu_load?
>
> but there's a change in the interpretation of bit 6:
>
```
> -	if (!(sysctl_sched_features & 64)) {
> -		this_load = this_rq->raw_weighted_load;
> +	if (sysctl_sched_features & 64) {
> +		this_load = this_rq->lrq.raw_weighted_load;
```
>
> the update of the cpu_load[] value is timer interrupt driven, but the
> _value_ that is sampled is not. [...]

Ah ..ok. Should have realized it earlier. Thanks for the education, but:

> Previously we used ->raw_weighted_load
> (at whatever value it happened to be at the moment the timer irq hit the
> system), now we basically use a load derived from the fair-time passed
> since the last scheduler tick. [...]

Isn't that biasing the overall cpu load to be dependent on SCHED_NORMAL
task load (afaics update_curr_rt doesn't update fair_clock at all)?

What if a CPU had just real-time tasks and no SCHED_NORMAL/BATCH tasks?
Would the cpu_load be seen to be very low?

[Dmitry's proposal for a per-class update_load() callback seems to be a
good thing in this regard]

> > Just to be clear, by container patches, I am referring to "process"
> > container patches from Paul Menage [1]. They aren't necessarily tied
> > to "virtualization-related" container support in -mm tree, although I
> > believe that "virtualization-related" container patches will make use
> > of the same "process-related" container patches for their
> > task-grouping requirements. Phew ..we need better names!
>
> i'd still like to hear back from Kirill & co whether this framework is
> flexible enough for their work (OpenVZ, etc.) too.

sure .. i would love to hear their feedback as well on the overall
approach of these patches, which is:

1. Using Paul Menage's process container patches as the basis of
   task-grouping functionaility. I think there is enough consensus
   on this already

(more importantly)

2. Using CFS core to achieve fairness at higher hierarchical levels
   (including at a container level). It would be nice to reuse much
   of the CFS logic which is driving fairness between tasks currently.

3. Using smpnice mechanism for SMP load-balance between CPUs
   (also largely based on what is there currently in CFS). Basic idea behind
   this is described at http://lkml.org/lkml/2007/5/25/146

Kirill/Herbert/Eric?

--
Regards,
vatsa

--
Regards,
vatsa

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers