Subject: Re: [RFC] [PATCH 0/3] Add group fairness to CFS
Posted by Srivatsa Vaddagiri on Fri, 25 May 2007 07:59:36 GMT
View Forum Message <> Reply to Message

On Wed, May 23, 2007 at 08:32:52PM +0200, Ingo Molnar wrote:
> > Here's an attempt to extend CFS (v13) to be fair at a group level,
> > rather than just at task level. The patch is in a very premature state
> > (passes simple tests, smp load balance not supported yet) at this
> > point. I am sending it out early to know if this is a good direction
> > to proceed.
>
> cool patch! :-)

Thanks!

> > 1. This patch reuses CFS core to achieve fairness at group level also.
> >
> > To make this possible, CFS core has been abstracted to deal with
> > generic schedulable "entities" (tasks, users etc).
>
> yeah, i like this alot.
>
> The "struct sched_entity" abstraction looks very clean, and that's the
> main thing that matters: it allows for a design that will only cost us
> performance if group scheduling is desired.
>
> If you could do a -v14 port and at least add minimal SMP support: i.e.
> it shouldnt crash on SMP, but otherwise no extra load-balancing logic is
> needed for the first cut - then i could try to pick all these core
> changes up for -v15. (I'll let you know about any other thoughts/details
> when i do the integration.)

Sure ..I will work on a -v14 port. I would like to target for something which:

1. doesn't break performance/functionality of existing CFS scheduler
-if- CONFIG_FAIR_USER_SCHEDULER is disabled. This also means load balance should work as it works today when the config option is disabled.

Do you recommend a set of tests that I need to run to ensure there is no regression? I know that there is a bunch of scheduler tests floating around on lkml ..Just need to dig to them (or if someone has all these tests handy on a website, I will download from that site!)

2. Provides fairness at group (user) level at the cost of missing load balance functionaility (missing until I get around to work on it that is).

> kernel builds dont really push scheduling micro-costs, rather try
> something like 'hackbench.c' to measure that. (kernel builds are of
> course one of our primary benchmarks.)

sure i will try that on my next version.

--
Regards,
vatsa

_____

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