Subject: Re: [RFC][PATCH 0/6] Add group fairness to CFS - v1
Posted by Srivatsa Vaddagiri on Tue, 12 Jun 2007 05:50:24 GMT
View Forum Message <> Reply to Message

On Mon, Jun 11, 2007 at 09:37:35PM +0200, Ingo Molnar wrote:
> > Patches 1-3 introduce the essential changes in CFS core to support
> > this concept. They rework existing code w/o any (intended!) change in
> > functionality.
>
> i currently have these 3 patches applied to the CFS queue and it's
> looking pretty good so far! If it continues to be problem-free i'll
> release them as part of -v17, just to check that they truly have no bad
> side-effects (they shouldnt). Then #4 can go into -v18.

ok. I am also most concerned about not upsetting current performance of
CFS when CONFIG_FAIR_GROUP_SCHED is turned off. Staging these patches in
incremental versions of CFS is a good idea to test that.

> i've attached my current -v17 tree - it should apply mostly cleanly
> ontop of the -mm queue (with a minor number of fixups). Could you
> refactor the remaining 3 patches ontop of this base? There's some
> rejects in the last 3 patches due to the update_load_fair() change.

sure, i will rework them on this -v17 snapshot.

> > Patch 4 fixes some bad interaction between SCHED_RT and SCHED_NORMAL
> > tasks in current CFS.
>
> btw., the plan here is to turn off 'bit 0' in sched_features: i.e. to
> use the precise statistics to calculate lrq->cpu_load[], not the
> timer-irq-sampled imprecise statistics. Dmitry has fixed a couple of
> bugs in it that made it not work too well in previous CFS versions, but
> now we are ready to turn it on for -v17. (indeed in my tree it's already
> turned on - i.e. sched_features defaults to '14')

On Mon, Jun 11, 2007 at 09:39:31PM +0200, Ingo Molnar wrote:
> i mean bit 6, value 64. I flipped around its meaning in -v17-rc4, so the
> new precise stats code there is now default-enabled - making SMP
> load-balancing more accurate.

I must be missing something here. AFAICS, cpu_load calculation still is
timer-interrupt driven in the -v17 snapshot you sent me. Besides, there
is no change in default value of bit 6 b/n v16 and v17:

```
-unsigned int sysctl_sched_features __read_mostly = 1 | 2 | 4 | 8 | 0 | 0;
+unsigned int sysctl_sched_features __read_mostly = 0 | 2 | 4 | 8 | 0 | 0;
```

So where's this precise stats based calculation of cpu_load?

Anyway, do you agree that splitting the cpu_load/nr_running fields so that:

rq->nr_running   = total count of -all- tasks in runqueue
rq->raw_weighted_load   = total weight of -all- tasks in runqueue
rq->lrq.nr_running   = total count of SCHED_NORMAL/BATCH tasks in runqueue
rq->lrq.raw_weighted_load = total weight of SCHED_NORMAL/BATCH tasks in runqueue

is a good thing to avoid SCHED_RT<->SCHED_NORMAL/BATCH mixup (as accomplished in Patch #4)?

If you don't agree, then I will make this split dependent on
CONFIG_FAIR_GROUP_SCHED

> > Patch 5 introduces basic changes in CFS core to support group
> > fairness.
> >
> > Patch 6 hooks up scheduler with container patches in mm (as an
> > interface for task-grouping functionality).

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!

> ok. Kirill, how do you like Srivatsa's current approach? Would be nice
> to kill two birds with the same stone, if possible :-)

One thing the current patches don't support is the notion of virtual cpus (which Kirill and other "virtualization-related" container folks would perhaps want). IMHO, the current patches can still be usefull for containers to load balance between those virtual cpus (as and when it is introduced).

> you'll get the best hackbench results by using SCHED_BATCH:
>
>   chrt -b 0 ./hackbench 10

thanks for this tip. Will try out and let you know how it fares for me.

> or indeed increasing the runtime_limit would work too.

References:

1. https://lists.linux-foundation.org/pipermail/containers/2007-May/005261.html

--
Regards,
vatsa