Subject: Re: A strange behavior of sched_fair
Posted by akpm on Fri, 29 Feb 2008 07:08:20 GMT
View Forum Message <> Reply to Message

(cc containers list)

On Wed, 27 Feb 2008 17:51:35 -0500 Kei Tokunaga <tokunaga.keiich@jp.fujitsu.com> wrote:

> Hi Ingo,
>
> I am playing around with sched_fair and cgroup, and it seems like
> I hit a possible bug. Could you also check if that is a bug?
>
> Description of behavior:
>   Start a cpu-bound task (t1), attach it to a cgroup (cgA), and let the
>   task to run for a while (e.g. several tens of seconds or a couple of
>   minutes would be adequate.) Then, start another cpu-bound task (t2)
>   and attach it to cgA in the way described in "Steps to Reproduce" section.
>   You will see t1 does not get run for a while.
>   (The tasks may not have to be cpu-bound, but it is easier to see the
>   behavior using cpu-bound tasks.)
>
> How reproducible:
>   Always.
>
> Environments where I saw the behavior:
>   2.6.25-rc3 with resource management functions enabled on ia64 box.
>
> Steps to Reproduce:
>   # mkdir /dev/cgroup
>   # mount -t cgroup -ocpuset,cpu cpu /dev/cgroup
>   # mkdir /dev/cgroup/{a,b}
>   # echo 0 > /dev/cgroup/a/cpuset.cpus
>   # echo 0 > /dev/cgroup/b/cpuset.cpus
>   # echo 1 > /dev/cgroup/a/cpuset.mems
>   # echo 1 > /dev/cgroup/b/cpuset.mems
>   # echo $$ > /dev/cgroup/b/tasks
>   # ./a.out & echo $! > /dev/cgroup/a/tasks (a.out is just a for-loop program)
>   [Wait for several tens of seconds or a couple of minutes.]
>   # ./a.out2 & echo $! > /dev/cgroup/a/tasks (a.out2 is just a for-loop program)
>   [You will see a.out does not get run for a while by running top command.]
>
> Additional Info:
>   a.out2 needs to be started from the shell of cgroup-b in order to
>   reproduce the problem (, unless the system is UP.) Starting a.out2
>   in the manner, se->vruntime (or something to create the se->vruntime)
>   of a.out2 seems to be initialized to a small value, compared to the
>   value of a.out. And the fair scheduler only runs a.out2 until the

>   se->vruntime catches up with the se->vruntime of a.out.
>
> Thanks,
> Kei
> --
> Kei Tokunaga
> Fujitsu (Red Hat On-site Partner)

_____

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