Subject: Re: Re: Hang with fair cgroup scheduler (reproducer is attached.)
Posted by Dmitry Adamushko on Mon, 17 Dec 2007 22:52:37 GMT
View Forum Message <> Reply to Message

[trimmed the cc' list]

On 17/12/2007, Steven Rostedt <rostedt@goodmis.org> wrote:
>
> On Mon, 17 Dec 2007, Dmitry Adamushko wrote:
>
> >
> > It may be related, maybe not. One 'abnormal' thing (at least, it
> > occurs only once in this log. Should be checked wheather it happens
> > when the system works fine) is that a few iterations before the oops
> > happens we observe the following pattern:
> >
> > CPU=2 [94359.651930] hackbench:1932(120:120:120:T) -->>
> > hackbench:1591(120:120:120)
> >
> > CPU=2 [94359.651980] hackbench:1591(49:120:120:T) -->> swapper:0(140:120:140)
>
> Thanks for noticing. The -rt patch has more priority inheritance
> situations than vanilla kernel (sleeping spinlocks or semaphors, and even
> the Preempt RCU Boost logic).

One more thing is that we don't actually see a point where that
'hackbench' gets its priority lifted.

It was scheduled in as a NORMAL task and scheduled out as a RT one.
i.e. the task got its prio elevated while it was running... a
contention with the task on another CPU?

anyway, i.e. this task must have 'p->se.on_rq == 1' and I'd expect to
see "switched_to_rt" message somewhere in between... hmm?
(check_class_changed() shoud have been called in task_setprio()).

btw., we do see one 'switched_from_rt --> switched_to_fair' case for
another 'hackbench' on CPU #0... according to traces, this one might
get a prio lifted while sleeping (it got scheduled in as a RT task).

>
> -- Steve
>

--
Best regards,

Dmitry Adamushko