Subject: Re: Re: Hang with fair cgroup scheduler (reproducer is attached.)
Posted by Steven Rostedt on Mon, 17 Dec 2007 17:58:21 GMT
View Forum Message <> Reply to Message

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)

Note: the 'T' should be a 'D' because my logdev didn't add the change that
-rt does (adding a 'M' state).

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).

>
> swapper (idle)    --> softirq-timer (RT)
> softirq-timer (RT) --> softirq-rcu (RT)
> softirq-rcu(RT)  --> picks up se == 0 for SCHED_NORMAL upon scheduling
> out ---> OOPS
>
> 'hackbench' was of SCHED_NORMAL upon scheduling _in_, and it's of RT
> type (prio: 49 and schedule() --> put_prev_task_rt()) upon scheduling
> _out_.
>
> Unless you run some modified version of 'hackbench', it doesn't chenge
> scheduling classes... so maybe a lifted prio is a consequence of the
> resource contention with some RT task ?

Yes. Which means it could be an spinlock, mutex, semaphore or RCU read
lock. But since it is in the TASK_UNINTERRUPTIBLE state, I'm willing to
bet this is a mutex (or converted spinlock).

>
> This 'hackbench' was the last SCHED_NORMAL task to run on this CPU...
> so however this NORMAL -> RT transition happened, it might leave a
> sched_fair's runqueue corrupted...

Could very well have. The PI uses task_setprio (aka. rt_mutex_setprio) to

raise the priority.  I'll start looking there.

>
> (Will try to look more when time allows).

Thanks, I'll probably spend the rest of the day on this.

-- Steve