Subject: Re: Re: Hang with fair cgroup scheduler (reproducer is attached.)
Posted by Dmitry Adamushko on Mon, 17 Dec 2007 10:23:08 GMT
View Forum Message <> Reply to Message

On 17/12/2007, Steven Rostedt <rostedt@goodmis.org> wrote:
>
> Here's a little snippet of where things went wrong.
>
> [94359.652019] cpu:3 (hackbench:1658) pick_next_task_fair:1036 nr_running=1
> [94359.652020] cpu:3 (hackbench:1658) pick_next_entity:625 se=ffff810009020800
> [94359.652021] cpu:0 (hackbench:1473) put_prev_entity:631
> [94359.652022] cpu:3 (hackbench:1658) pick_next_entity:625 se=ffff81003906b5a8
> [94359.652022] cpu:2 (softirq-timer/2:32) put_prev_task_rt:283
> [94359.652023] cpu:0 (hackbench:1473) put_prev_entity:631
> >>>> IN LOGDEV SWITCH <<<< cpu:3
> CPU=3 [94359.652023] hackbench:1658(120:120:120:D) -->> hackbench:1586(120:120:120)
> >>>> IN LOGDEV SWITCH <<<< cpu:2
> CPU=2 [94359.652024] softirq-timer/2:32(49:115:49:D) -->> softirq-rcu/2:39(49:115:49)
> >>>> IN LOGDEV SWITCH <<<< cpu:0
> CPU=0 [94359.652025] hackbench:1473(120:120:120:R) -->> hackbench:1591(49:120:120)
> [94359.652029] cpu:3 (hackbench:1586) put_prev_entity:631
> [94359.652030] cpu:2 (softirq-rcu/2:39) move_tasks:2472
> [94359.652030] cpu:3 (hackbench:1586) put_prev_entity:631
> [94359.652032] cpu:3 (hackbench:1586) pick_next_task_fair:1036 nr_running=1
> [94359.652033] cpu:3 (hackbench:1586) pick_next_entity:625 se=ffff810009020800
> [94359.652034] cpu:3 (hackbench:1586) pick_next_entity:625 se=ffff810014c7ab18
> [94359.652034] cpu:2 (softirq-rcu/2:39) put_prev_task_rt:283
> >>>> IN LOGDEV SWITCH <<<< cpu:3
> CPU=3 [94359.652035] hackbench:1586(120:120:120:T) -->> hackbench:1623(120:120:120)
> [94359.652036] cpu:2 (softirq-rcu/2:39) pick_next_task_fair:1036 nr_running=1
> [94359.652038] cpu:2 (softirq-rcu/2:39) pick_next_entity:625 se=0000000000000000
>
> I see that softirq-rcu on cpu 2 started doing a move_tasks, when it got to
> the state where nr_running returned 1 and the se from pick_next_entity was
> NULL.

move_task() is likely to be run from schedule() --> idle_balance() ,
for our case it means that 'softirq-rcu' (which is a RT task) went to
sleep and it was the last task on this CPU (rq->nr_running == 0 -->
idle_balance() was triggered).

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)

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 ?

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...

(Will try to look more when time allows).

>
> Thanks,
>
> -- Steve

--
Best regards,
Dmitry Adamushko

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