Subject: Re: Re: Hang with fair cgroup scheduler (reproducer is attached.)
Posted by Dmitry Adamushko on Fri, 14 Dec 2007 19:51:28 GMT
View Forum Message <> Reply to Message

```
> [ ... ]
>
>  [<a0000001002e0480>] rb_erase+0x300/0x7e0
>  [<a000000100076290>] __dequeue_entity+0x70/0xa0
>  [<a000000100076300>] set_next_entity+0x40/0xa0
>  [<a0000001000763a0>] set_curr_task_fair+0x40/0xa0
>  [<a000000100078d90>] sched_move_task+0x2d0/0x340
>  [<a000000100078e20>] cpu_cgroup_attach+0x20/0x40
>
> [ ... ]
```

argh... it's a consequence of the 'current is not kept within the tree" indeed.

When sched_move_task() is called for the 'current' (running on another CPU), we get the following:

```
...
	running = task_running(rq, tsk);
	on_rq = tsk->se.on_rq;

	if (on_rq) {
		dequeue_task(rq, tsk, 0);
		if (unlikely(running))
			tsk->sched_class->put_prev_task(rq, tsk);
	}
```

[1] tsk->sched_class->put_prev_task() actually _inserts_ 'tsk' back into the cfs_rq of its _old_ group :

```
	set_task_cfs_rq(tsk, task_cpu(tsk));
```

[2] now task.se->cfs_rq gets changed

```
	if (on_rq) {
		if (unlikely(running))
			tsk->sched_class->set_curr_task(rq);
```

[3] and now, tsk->sched_class->set_curr_task(rq) _removes_ the 'current' from the tree... but this tree belongs to the _new_ group (the task is still within the 'old_group->cfs_rq->rb_tree') ---> oops!

```
		enqueue_task(rq, tsk, 0);
	}
```

Anyway, I have to admit that this problem is a consequence of the
special-case treatment for the 'current' by
'dequeue/enqueue_task()'... it makes the interface less transparent
indeed.

/me thinking on how to get it fixed (e.g. set_task_cfs_rq() might take
care of it) or just get this special-case issue removed (have to check
whether we lose anything in this case)... sigh.

--
Best regards,
Dmitry Adamushko