Subject: Re: [PATCH] Hookup group-scheduler with task container infrastructure
Posted by Dmitry Adamushko on Mon, 10 Sep 2007 22:28:51 GMT
View Forum Message <> Reply to Message

On 10/09/2007, Srivatsa Vaddagiri <vatsa@linux.vnet.ibm.com> wrote:
> On Mon, Sep 10, 2007 at 10:22:59AM -0700, Andrew Morton wrote:
> > objection ;) "cpuctlr" isn't memorable. Kernel code is write-rarely,
> > read-often. "cpu_controller", please. The extra typing is worth it ;)
>
> Ok! Here's the modified patch (against 2.6.23-rc4-mm1).
>
> [...]
>
> +/* change task's runqueue when it moves between groups */
> +static void sched_move_task(struct container_subsys *ss, struct container *cont,
> + struct container *old_cont, struct task_struct *tsk)
> +{
> + int on_rq;
> + unsigned long flags;
> + struct rq *rq;
> +
> + rq = task_rq_lock(tsk, &flags);
> +

I guess, update_rq_clock(rq) should be placed here.

humm... do you really need deactivate/activate_task() here? 'rq' and
p->se.load.weight stay unchanged so enqueue/dequeue_task() would do a
job, no?
Although, I might be missing (definitely, as this part is not
completely clear to me yet) something important at this late hour :)

> + on_rq = tsk->se.on_rq;
> + if (on_rq)
> + deactivate_task(rq, tsk, 0);
> +
> + if (unlikely(rq->curr == tsk) && tsk->sched_class == &fair_sched_class)
> + tsk->sched_class->put_prev_task(rq, tsk);
> +
> + set_task_cfs_rq(tsk);
> +
> + if (on_rq)
> + activate_task(rq, tsk, 0);
> +
> + if (unlikely(rq->curr == tsk) && tsk->sched_class == &fair_sched_class)
> + tsk->sched_class->set_curr_task(rq);
> +

```
> +       task_rq_unlock(rq, &flags);
> +}
> +
```

--
Best regards,
Dmitry Adamushko

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