

---

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.

hummm... 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>

---