Subject: Re: [ckrm-tech] [PATCH 1/7] containers (V7): Generic container system abstracted from cpusets code
Posted by Srivatsa Vaddagiri on Sun, 25 Mar 2007 00:38:29 GMT
View Forum Message <> Reply to Message

On Sat, Mar 24, 2007 at 12:25:59PM -0700, Paul Jackson wrote:

```
> > P.S : cpuset.c checks for PF_EXITING twice in attach_task(), while this
> > patch seems to be checking only once. Is that fine?
>
> I think the cpuset code is ok, because, as you note, it locks the task,
> picks off the cpuset pointer, and then checks a second time that the
> task still does not have PF_EXITING set:
```

Well afaics, PF_EXITING is set for the exiting task w/o taking any lock, which makes this racy always.

```
> In the kernel/cpuset.c code for attach_task():
>
>        task_lock(tsk);
>        oldcs = tsk->cpuset;
>        /*
>         * After getting 'oldcs' cpuset ptr, be sure still not exiting.
>         * If 'oldcs' might be the top_cpuset due to the_top_cpuset_hack
>         * then fail this attach_task(), to avoid breaking top_cpuset.count.
>         */
>        if (tsk->flags & PF_EXITING) {
```

What if PF_EXITING is set after this check? If that happens then,

```
>                task_unlock(tsk);
>                mutex_unlock(&callback_mutex);
>                put_task_struct(tsk);
>                return -ESRCH;
>        }
```

the following code becomes racy with cpuset_exit() ...

```
        atomic_inc(&cs->count);
        rcu_assign_pointer(tsk->cpuset, cs);
        task_unlock(tsk);
```

--
Regards,
vatsa