Subject: Re: [PATCH] task containersv11 add tasks file interface fix for cpusets
Posted by Paul Jackson on Sat, 06 Oct 2007 08:24:37 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> What was wrong with my suggestion from a couple of emails back? Adding
> the following in cpuset_attach():
>
> struct cgroup_iter it;
> struct task_struct *p;
> cgroup_iter_start(cs->css.cgroup, &it);
> while ((p = cgroup_iter_next(cs->css.cgroup, &it)))
> set_cpus_allowed(p, cs->cpus_allowed);
> cgroup_iter_end(cs->css.cgroup, &it);

This isn't working for me.

The key kernel routine for updating a tasks cpus_allowed
cannot be called while holding a spinlock.

But the above loop holds a spinlock, css_set_lock, between
the cgroup_iter_start and the cgroup_iter_end.

I end up generating complaints of:
 BUG: scheduling while atomic
when I invoke the set_cpus_allowed() above.

Should css_set_lock be a mutex? Locking changes like that
can be risky.

Or perhaps there should be another callback, called only by
attach() requests back to the same group. Likely cpusets would
be the only subsystem interested in plugging that callback.

That, or my original patch, which calls the attach routine
even if re-attaching to the current cgroup ...

--
 I won't rest till it's the best ...
 Programmer, Linux Scalability
 Paul Jackson <pj@sgi.com> 1.925.600.0401

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