Subject: Re: [ckrm-tech] [PATCH 1/7] containers (V7): Generic container system abstracted from cpusets code
Posted by Paul Jackson on Sun, 25 Mar 2007 04:45:50 GMT
View Forum Message <> Reply to Message

vatsa wrote:
> Now consider:

Nice work - thanks. Yes, both an extra cpuset count and a negative cpuset count are bad news, opening the door to the usual catastrophes.

Would you like the honor of submitting the patch to add a task_lock to cpuset_exit()? If you do, be sure to fix, or at least remove, the cpuset_exit comment lines:

```
 * We don't need to task_lock() this reference to tsk->cpuset,
 * because tsk is already marked PF_EXITING, so attach_task() won't
 * mess with it, or task is a failed fork, never visible to attach_task.
```

I guess that taking task_lock() in cpuset_exit() should not be a serious performance issue. It's taking a spinlock that is in the current exiting tasks task struct, so it should be a cache hot memory line and a rarely contested lock.

And I guess I've not see this race in real life, as one side of it has to execute quite a bit of code in the task exit path, from when it sets PF_EXITING until it gets into the cpuset_exit() call, while the other side does the three lines:

```
 if (tsk->flags & PF_EXITING) ...
 atomic_inc(&cs->count);
 rcu_assign_pointer(tsk->cpuset, cs);
```

So, in real life, this would be a difficult race to trigger.

Thanks for finding this.

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