Subject: Re: [PATCH] task containersv11 add tasks file interface fix for cpusets
Posted by David Rientjes on Sun, 07 Oct 2007 06:13:10 GMT
View Forum Message <> Reply to Message

On Sat, 6 Oct 2007, Paul Jackson wrote:

> > struct cgroup_iter it;
> > struct task_struct *p, **tasks;
> > int i = 0;
> >
> > cgroup_iter_start(cs->css.cgroup, &it);
> > while ((p = cgroup_iter_next(cs->css.cgroup, &it))) {
> > get_task_struct(p);
> > tasks[i++] = p;
> > }
> > cgroup_iter_end(cs->css.cgroup, &it);
>
> Hmmm ... guess I'd have to loop over the cgroup twice, once to count
> them (the 'count' field is not claimed to be accurate) and then again,
> after I've kmalloc'd the tasks[] array, filling in the tasks[] array.
>

Use the struct task_struct **tasks above. Any call here should only be about 300 bytes into the stack so, with 8K stacks on x86_64, you should be able to store pointers to a little under 1,000 tasks before corrupting it.

Or, if you think there's going to be more than 1,000 tasks in this cgroup, just do an arbitrary number of them and then reuse your stack before calling set_cpus_allowed() and put_task_struct() on all tasks you've collected so far; then finally call cgroup_iter_end() when you're done with all of them. Use the task_struct usage counter as your locking so that tasks don't prematurely exit or deallocate before you migrate them.

Keep in mind that simply iterating over the tasks and using set_cpus_allowed() isn't good enough not only because you're holding a spinlock but also because the task can exit during the call, unless you're using some sort of cgroups locking I'm not aware of that prevents that. My solution prevents that with {get,put}_task_struct().

The current update_cpu_domains(), or -mm's equivalent of it with this patchset, cannot call lock_cpu_hotplug() with callback_mutex held, so it'll need to be unlocked after all the tasks have been put. If you're preempted here before update_cpu_domains(), attach_task() will take care of the cpus_allowed reassignment.

David

_____
Containers mailing list