
Subject: Re: [PATCH] task containersv11 add tasks file interface fix for cpusets
Posted by [Paul Jackson](#) on Fri, 12 Oct 2007 01:23:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

> Since we know that the tasks are not running, and that we have
> exclusive access to the tasks in the control group we can take action
> as if we were the actual tasks themselves. Which should simplify
> locking.

The Big Kernel Lock (BKL), born again, as a Medium Sized Cgroup Lock ?

This only simplifies things if it enables us to remove finer grain locking, but some finer grain locking is essential for performance on higher processor count systems (which if Intel and AMD have their way, will be just about any system bigger than a cell phone.)

There is no escaping actually having to think about these things, and clearly understand and document what locks what. Locks don't just guard code sections, and they don't just guard particular data items.

Rather, in my view, they ensure certain invariants on your data. Non-atomic code sections that depend on, or modify, data subject to such invariants must be done while holding the appropriate lock. One is guaranteed not to see the invariant violated while holding the lock, nor to expose others to temporary violations of the invariant done while locked.

In this case, we have multiple copies of cpumasks in task structs and cpusets that must honor the invariant that they are equal or subsets of, `cpu_online_map`. Changes to the set of online CPUs must hold some lock, apparently `sched_hotcpu_mutex`, until all those cpumasks are adjusted to once again honor, this invariant.

--

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>
