
Subject: Re: v2.6.26-rc7/cgroups: circular locking dependency
Posted by [Paul Jackson](#) on Mon, 23 Jun 2008 12:02:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

CC'd Gautham R Shenoy <ego@in.ibm.com>.

I believe that we had the locking relation between what had been `cgroup_lock` (global cgroup lock which can be held over large stretches of non-performance critical code) and `callback_mutex` (global cpuset specific lock which is held over shorter stretches of more performance critical code - though still not on really hot code paths.) One can nest `callback_mutex` inside `cgroup_lock`, but not vice versa.

The `callback_mutex` guarded some CPU masks and Node masks, which might be multi-word and hence don't change atomically. Any low level code that needs to read these these cpuset CPU and Node masks, needs to hold `callback_mutex` briefly, to keep that mask from changing while being read.

There is even a comment in `kernel/cpuset.c`, explaining how an ABBA deadlock must be avoided when calling `rebuild_sched_domains()`:

```
/*  
 * rebuild_sched_domains()  
 *  
 * ...  
 *  
 * Call with cgroup_mutex held. May take callback_mutex during  
 * call due to the kfifo_alloc() and kmalloc() calls. May nest  
 * a call to the get_online_cpus()/put_online_cpus() pair.  
 * Must not be called holding callback_mutex, because we must not  
 * call get_online_cpus() while holding callback_mutex. Elsewhere  
 * the kernel nests callback_mutex inside get_online_cpus() calls.  
 * So the reverse nesting would risk an ABBA deadlock.
```

This went into the kernel sometime around 2.6.18.

Then in October and November of 2007, Gautham R Shenoy submitted "RefCount Based Cpu Hotplug" (<http://lkml.org/lkml/2007/11/15/239>)

This added `cpu_hotplug.lock`, which at first glance seems to fit into the locking hierarchy about where `callback_mutex` did before, such as being invocable from `rebuild_sched_domains()`.

However ... the `kernel/cpuset.c` comments were not updated to describe the intended locking hierarchy as it relates to `cpu_hotplug.lock`, and it looks as if `cpu_hotplug.lock` can also be taken while invoking the hotplug callbacks, such as the one here that is handling a CPU down

event for cpusets.

Gautham ... you there?

--

I won't rest till it's the best ...
Programmer, Linux Scalability
Paul Jackson <pj@sgi.com> 1.940.382.4214

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