Subject: Re: [Patch 1/3] Miscellaneous container fixes
Posted by Paul Jackson on Fri, 01 Dec 2006 20:31:34 GMT
View Forum Message <> Reply to Message

Paul M wrote:
> Ah - this may be the lockup that PaulJ hit.

Yes - looks like this fixes it. Thanks, Srivatsa.

And with that fix, it becomes obvious how to reproduce this problem:

```
 mount -t cpuset cpuset /dev/cpuset # if not already mounted
 cd /dev/cpuset
 mkdir foo
 echo 1 > foo/cpu_exclusive
 rmdir foo    # hangs ...
```

However ...

Read the comment in kernel/cpuset.c for the routine cpuset_destroy().
It explains that update_flag() is called where it is (turning off
the cpu_exclusive flag, if it was set), to avoid the calling sequence:

```
  cpuset_destroy->update_flag->update_cpu_domains->lock_cpu_hotplug
```

while holding the callback_mutex, as that could ABBA deadlock with the
CPU hotplug code.

But with this container based rewrite of cpusets, it now seems that
cpuset_destroy -is- called holding the callback_mutex (though I don't
see any mention of that in the cpuset_destroy comment ;), so it would
seem that we once again are at risk for this ABBA deadlock.

I also notice that the comment for container_lock() in the file
kernel/container.c only mentions its use in the oom code. That is
no longer the only, or even primary, user of this lock routine.
The kernel/cpuset.c code uses it frequently (without comment ;),
and I wouldn't be surprised to see other future controllers calling
container_lock() as well.

Looks like its time to update those comments, and think about what
was written there before, as that might catch a bug or two, such as
the one Srivatsa just fixed for us.

Most of those long locking comments in kernel/cpuset.c are there
for a reason - recording the results of a lesson learned in the
school of hard knocks.

--

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