Subject: Re: [PATCH 0/7] Generic Process Containers (+ ResGroups/BeanCounters)
Posted by Paul Jackson on Thu, 30 Nov 2006 07:32:29 GMT
View Forum Message <> Reply to Message

I got a chance to build and test this patch set, to see if it behaved like I expected cpusets to behave, on an ia64 SN2 Altix system.

Two details - otherwise looked good. I continue to like this approach.

The two details are (1) /proc/<pid>/cpuset not configured by default if CPUSETS configured, and (2) a locking bug wedging tasks trying to rmdir a cpuset off the notify_on_release hook.

1) I had to enable CONFIG_PROC_PID_CPUSET. I used the following one line change to do this. I am willing to consider, in due time, phasing out such legacy cpuset support. But so long as it is small stuff that is not getting in anyone's way, I think we should take our sweet time about doing so -- as in a year or two after marking it deprecated or some such. No sense deciding that matter now; keep the current cpuset API working throughout any transitition to container based cpusets, then revisit the question of whether to deprecate and eventually remove these kernel API details, later on, after the major reconstruction dust settles.
In general, we try to avoid removing kernel API's, especially if they are happily being used and working and not causing anyone grief.

```
============================ begin ============================
--- 2.6.19-rc5.orig/init/Kconfig 2006-11-29 21:14:48.071114833 -0800
+++ 2.6.19-rc5/init/Kconfig 2006-11-29 22:19:02.015166048 -0800
@@ -268,6 +268,7 @@ config CPUSETS
 config PROC_PID_CPUSET
  bool "Include legacy /proc/<pid>/cpuset file"
  depends on CPUSETS
+ default y if CPUSETS

 config CONTAINER_CPUACCT
  bool "Simple CPU accounting container subsystem"
============================ end ============================
```

2) I wedged the kernel on the container_lock, doing a removal of a cpuset using notify_on_release.

Right now, that test system has the following two tasks, wedged:

```
============================ begin ============================
F S UID   PID PPID C PRI NI ADDR SZ  WCHAN  STIME TTY  TIME     CMD
0 S root 4992   34 0  71 -5 -   380   wait   22:51 ?   00:00:00 /bin/sh /sbin/cpuset_release_agent
/cpuset_test_tree
0 D root 4994 4992 0  72 -5 -   200 contai   22:51 ?   00:00:00 rmdir /dev/cpuset//cpuset_test_tree
============================ end ============================
```

I had a cpuset called /cpuset_test_tree, and some sub-cpusets below it. I marked it 'notify_on_release' and then removed all tasks from it, and then removed the child cpusets that it had. Removing that last child cpuset presumably triggered the above callout to /sbin/cpuset_release_agent, which called rmdir.

That wait address (from /proc/4994/stat) in hex is a0000001000f1060, and my System.map has the two lines:

```
a0000001000f1040 T container_lock
a0000001000f1360 T container_manage_unlock
```

So it is wedged in container_lock.

I have subsequently also wedged an 'ls' command trying to scan this /dev/cpuset directory, waiting in the kernel routine vfs_readdir (not surprising, given that I'm in the middle of doing a rmdir on that directory.)

If you don't immediately see the problem, I can go back and get a kernel stack trace or whatever else you need.

This lockup occurred the first, and thus far only, time that I tried to use notify_on_release to rmdir a cpuset. So I presume it is an easy failure for me to reproduce.

--

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