
Subject: Re: [ckrm-tech] [PATCH 1/7] containers (V7): Generic container system abstracted from cpusets code

Posted by [Paul Jackson](#) on Sun, 11 Mar 2007 19:38:43 GMT

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

vatsa wrote:

> Yes, that way only the hierarchy hosting cpusets takes the hit of
> double-locking. cpuset_subsys->create/destroy can take this additional lock
> inside cpuset.c.

The primary reason for the cpuset double locking, as I recall, was because cpusets needs to access cpusets inside the memory allocator. A single, straight forward, cpuset lock failed under the following common scenario:

- 1) user does cpuset system call (writes some file below /dev/cpuset, e.g.)
- 2) kernel cpuset code locks its lock
- 3) cpuset code asks to allocate some memory for some cpuset structure
- 4) memory allocator tries to lock the cpuset lock - deadlock!

The reason that the memory allocator needs the cpuset lock is to check whether the memory nodes the current task is allowed to use have changed, due to changes in the current tasks cpuset.

A secondary reason that the cpuset code needs two locks is because the main cpuset lock is a long held, system wide lock, and various low level bits of performance critical code sometimes require quick, read-only access to cpusets.

--

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