Subject: Re: [PATCH] task containersv11 add tasks file interface fix for cpusets
Posted by Paul Menage on Wed, 03 Oct 2007 18:10:58 GMT
View Forum Message <> Reply to Message

On 10/3/07, Paul Jackson <pj@sgi.com> wrote:
>
> Yes, something in user space has to do it. It's part of the
> kernel-user cpuset API. If you change a cpuset's 'cpus', then
> you have to rewrite each pid in its 'tasks' file back to that
> 'tasks' file in order to get that 'cpus' change to be applied
> to the task struct cpus_allowed of each task, and thereby visible
> to the scheduler.

What's the rationale for this?

Given that later in cpusets.txt you say:

>If hotplug functionality is used
>to remove all the CPUs that are currently assigned to a cpuset,
>then the kernel will automatically update the cpus_allowed of all
>tasks attached to CPUs in that cpuset to allow all CPUs

why can't the same thing be done when changing the 'cpus' file manually.

What's wrong with, in update_cpumask(), doing a loop across all
members of the cgroup and updating their cpus_allowed fields?

The existing cpusets API is broken, since a new child can always be
forked between reading the tasks file and doing the writes.

Paul

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