Subject: Re: [PATCH 4/4] The control group itself
Posted by serue on Thu, 14 Feb 2008 17:18:57 GMT
View Forum Message <> Reply to Message

Quoting Paul Jackson (pj@sgi.com):
> Serge wrote:
> > Paul (actually both Menage and Jackson :) do you have an opinion on
> > this? Are there sites which eg do 'chown -R some_user_id /cgroup/cpusets/'
> > to have some non-root user be able to dole out cpusets? Is there any
> > way it would be ok to have cgroup_file_write() check for CAP_SYS_ADMIN?
>
> I don't know what my users actually do here ... I'm a couple layers
> removed from that reality. But certainly I've recommended that they
> sometimes do things like having the batch scheduler chown the files
> of each jobs cpuset to the uid of the user running that job, so that
> the job can manipulate its own cpuset allocate resources in finer
> detail.
>
> One of the more elaborate ways of doing this nests a pair of cpusets,
> with the parent owned by the batch scheduler confining the child
> owned by the individual job. The job can actually do things like
> write its own cpus and mems files, but is confined by the parent
> cpuset to only specify cpus and mems assigned to that job.
>
> As to how this affects your question ... I'm not sure. However I
> suspect that an added requirement for CAP_SYS_ADMIN would cause
> breakage and not be a good idea.

Yeah, I guess a more general mechanism to couple a user namespace's
connection to a mount is the right way to go. If we can just specify
that root in this namespace is not root in that namespace (or any other
userid we've chowned the files to), we've got what we need.

thanks,
-serge

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