Subject: Re: [PATCH 3/7] containers (V7): Add generic multi-subsystem API to containers
Posted by Srivatsa Vaddagiri on Tue, 13 Feb 2007 13:19:45 GMT
View Forum Message <> Reply to Message

On Mon, Feb 12, 2007 at 10:40:52AM -0800, Paul Menage wrote:
> I did consider that approach at one point. The reason I rejected it
> was that then container->count would no longer even vaguely represent
> the number of processes in a container. Now that we have the
> container_group object, we have to use that for counting the number of
> processes in a container anyway, so that objection goes away.

Yep.

> However, I think it's important to be able to provide some kind of a
> reference count that subsystems can grab (e.g. to store a reference in
> a non-task object such as a file struct) without taking manage_mutex
> or callback_mutex (since that would be excessively heavyweight) but
> which can still be "frozen" at zero at the point when you're trying to
> destroy a container.

Well, we already bump up reference count in fork() w/o grabbing those
mutexes don't we? Also if rmdir() sees container->count to be zero, then
it means no task is attached to the container. How will then a function
like bc_file_charge() bump up the reference count to such a container
(presuming it wanted to do so w/o manage/callback mutexes -and- that the
container pointer in bc_file_charge is derived from some task in
that container). I think it is safe to bump up container->count in
bc_file_charge w/o grabbing manage/callback mutexes.

> Additionally, having it per subsystem will be
> important for when we implement arbitrary binding/unbinding of
> subsystems from hierarchies - at that point we need to be able know
> which subsystems have external reference counts, and hence aren't
> removeable.

Are you talking about (un)bind of subsystem to/from hierararchies that
have non-zero containers in them? That sounds very icky. Anyway that
doesnt seem to be supported in current patches.

Basically I felt we should defer introducing css_get/put until we find a good
user for it, (and bc_file_(un)charge dont seem to be good users of it-
see above).

--
Regards,