Subject: Re: [PATCH 3/7] containers (V7): Add generic multi-subsystem API to containers
Posted by Paul Menage on Mon, 12 Feb 2007 18:40:52 GMT
View Forum Message <> Reply to Message

On 2/12/07, Srivatsa Vaddagiri <vatsa@in.ibm.com> wrote:
> On Mon, Feb 12, 2007 at 12:15:24AM -0800, menage@google.com wrote:
> > +/*
> > + * Call css_get() to hold a reference on the container; following a
> > + * return of 0, this container subsystem state object is guaranteed
> > + * not to be destroyed until css_put() is called on it. A non-zero
> > + * return code indicates that a reference could not be taken.
> > + *
> > + */
> > +
>
> Why can't we reuse container->count (or container_group->ref) to
> refcount the per-subsystem object attached to a container? I think
> that is how it is done for cpusets? That would make css_get/put
> unnecessary?

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.

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. 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.

Paul