Subject: Re: Containers: css_put() dilemma
Posted by Balbir Singh on Thu, 19 Jul 2007 03:44:28 GMT
View Forum Message <> Reply to Message

On 7/19/07, Paul Menage <menage@google.com> wrote:
> On 7/17/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
> >
> > Thinking out loud again, can we add can_destroy() callbacks?
> >
>
> What would the exact semantics of such a callback be?
>
> Since for proper interaction with release agents we need the subsystem
> to notify the framework when a subsystem object becomes releasable
> (currently as part of css_put()), what would a can_destroy() callback
> let you do that you couldn't do just by taking an extra css refcount
> to prevent destruction and releasing that refcount to allow
> destruction?

I was thinking along those lines before you mentioned that the next version of css_put() will not block. The advantage I see of can_destory() is that it allows subsystems to do their own reference counting and decide whether they are ready to be deleted or not. The other advantage I see is that it can act like a prepare to be deleted phase for the controller, the controller might decide to take some action in the can_destroy() phase, like the memory controller could decide to start reclaiming all the remaining page cache pages.

BTW, do you know upfront as to when the next set of container enhancement patches will be ready? The css_put() issue is blocking us currently.

Balbir

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