
Subject: Re: Containers: css_put() dilemma
Posted by [Balbir Singh](#) on Tue, 17 Jul 2007 10:28:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul (??) Menage wrote:

> On 7/17/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

>> >

```
>> > >      mutex_lock(&container_mutex);
>> > >      set_bit(CONT_RELEASABLE, &cont->flags);
>> > >-      if (atomic_dec_and_test(&css->refcnt)) {
>> > >-          check_for_release(cont);
>> > >-      }
>> > >+      check_for_release(cont);
>> > >      mutex_unlock(&container_mutex);
>> > >
```

>

> I think that this isn't safe as it stands, without a synchronize_rcu()
> in container_diput() prior to the kfree(). Also, it will break if
> anyone tries to use a release agent on a hierarchy that has your
> memory controller bound to it.
>

Isn't the code functionally the same as before? We still do atomic_test_and_dec() as before. We still set_bit() CONT_RELEASABLE, we take the container_mutex and check_for_release(). I am not sure I understand what changed?

Could you please elaborate as to why using a release agent is broken when the memory controller is attached to it? I am not quite sure why we need the synchronize_rcu() either in container_diput().

> Paul

>

> Containers mailing list

> Containers@lists.linux-foundation.org

> <https://lists.linux-foundation.org/mailman/listinfo/containers>

--

Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

Containers mailing list

Containers@lists.linux-foundation.org

