Subject: Re: [PATCH 1/2] cgroup: Set CGRP_RELEASABLE when adding to a cgroup

Posted by Colin Cross on Fri, 03 Dec 2010 03:07:12 GMT

View Forum Message <> Reply to Message

On Wed, Nov 24, 2010 at 4:21 PM, Paul Menage <menage@google.com> wrote:
> On Wed, Nov 24, 2010 at 4:11 PM, Colin Cross <ccross@android.com> wrote:
>>>
>>> We seem to have lost some notify_on_release() checks - maybe move that
>>> to check_for_release()?
>> check_for_release immediately calls cgroup_is_releasable, which checks
>> for the same bit as notify_on_release. There's no need for
>> CGRP_RELEASABLE to depend on notify_on_release, or to check
>> notify_on_release before calling check_for_release.
>
> OK.
>
>> I matched the existing behavior, __css_put sets CGRP_RELEASABLE when
>> refcnt goes to 0.
>>
>
> Ah, we do appear to have had that behaviour for a while. I don't
> remember the justification for it at this point :-)
>
>> check_for_release is only called from __css_put, cgroup_rmdir, and
>> __put_css_set (or free_css_set_work after my second patch). Those all
>> imply that __css_get, get_css_set, or cgroup_create have been
>> previously called, which are the functions that set CGRP_RELEASABLE.
>
> Not in one case - if we create a new cgroup and try to move a thread
> into it, but the thread is exiting as we move it, we'll call
> put_css_set() on the new css_set, which will drop the refcount on the
> target cgroup back to 0. We wouldn't want the auto-release
> notification to kick in in that situation, I think.

Clearing the CGRP_RELEASABLE bit any time after the tests in
check_for_release introduces a race if __css_get is called between the
check and clearing the bit - the cgroup will have an entry, but the
bit will not be set. Without additional locking in __css_get, I don't
see any way to safely clear CGRP_RELEASABLE.

_____

Containers mailing list
Containers@lists.linux-foundation.org
 https://lists.linux-foundation.org/mailman/listinfo/containe rs