Subject: Re: [PATCH 1/2] cgroup: Set CGRP_RELEASABLE when adding to a cgroup
Posted by Colin Cross on Fri, 17 Dec 2010 01:12:42 GMT
View Forum Message <> Reply to Message

On Thu, Dec 16, 2010 at 4:54 PM, Paul Menage <menage@google.com> wrote:
> On Thu, Dec 2, 2010 at 7:07 PM, Colin Cross <ccross@android.com> wrote:
>>> 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.
>
> I don't quite follow your argument here. Are you saying that the
> problem is that you could end up spawning a release agent for a cgroup
> that was no longer releasable since it now had a process in it again?
> If so, then I don't think that's a problem - spurious release agent
> invocations for non-empty cgroups will always happen occasionally due
> to races between the kernel and userspace. But a failed move of a task
> into a previously-empty cgroup shouldn't trigger the agent.

No, if you add a new process to the group while check_for_release, the
bit could get set by the add for the new process, then cleared by the
concurrently running check_for_release. The release agent would be
spawned with a process in the group, which is fine, but when
RELEASABLE bit would be clear. When the new process was removed,
check_for_release would not call the release agent at all.

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