Subject: Re: [PATCH 1/2] cgroup: Set CGRP_RELEASABLE when adding to a cgroup
Posted by Colin Cross on Thu, 25 Nov 2010 00:18:59 GMT
View Forum Message <> Reply to Message

On Wed, Nov 24, 2010 at 4:11 PM, Colin Cross <ccross@android.com> wrote:
> On Wed, Nov 24, 2010 at 3:54 PM, Paul Menage <menage@google.com> wrote:
>> On Tue, Nov 23, 2010 at 9:37 PM, Colin Cross <ccross@android.com> wrote:

```
>>> @@ -364,12 +372,8 @@ static void __put_css_set(struct css_set *cg, int taskexit)
>>>                 struct cgroup *cgrp = link->cgrp;
>>>                 list_del(&link->cg_link_list);
>>>                 list_del(&link->cgrp_link_list);
>>> -               if (atomic_dec_and_test(&cgrp->count) &&
>>> -                   notify_on_release(cgrp)) {
>>> -                       if (taskexit)
>>> -                               set_bit(CGRP_RELEASABLE, &cgrp->flags);
>>> +               if (atomic_dec_and_test(&cgrp->count))
>>>                         check_for_release(cgrp);
>>> -               }
```

>>
>> 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.
>

```
>>>  /* Caller must verify that the css is not for root cgroup */
>>> +void __css_get(struct cgroup_subsys_state *css, int count)
>>> +{
>>> +       atomic_add(count, &css->refcnt);
>>> +       set_bit(CGRP_RELEASABLE, &css->cgroup->flags);
>>> +}
```

>>
>> Is css_get() the right place to be putting this? It's not clear to me
>> why a subsystem taking a refcount on a cgroup's state should render it
>> releasable when it drops that refcount.
> I matched the existing behavior, __css_put sets CGRP_RELEASABLE when
> refcnt goes to 0.
>
>> Should we maybe clear the CGRP_RELEASABLE flag right before doing the
>> userspace callback?
> Actually, I think CGRP_RELEASABLE can be dropped entirely.
> 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.
Nevermind, that's not true - get_css_set does not set CGRP_RELEASABLE,

cgroup_attach_task does.

If CGRP_RELEASABLE is not cleared before the callback, the release_agent would be run once when the last task was removed from the cgroup, and then again if a task failed to be added to the empty cgroup because the task was exiting, so clearing the flag sounds like a good idea.

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