Subject: Re: [PATCH] cgroup: Convert synchronize_rcu to call_rcu in cgroup_attach_task
Posted by Colin Cross on Wed, 24 Nov 2010 02:10:58 GMT
View Forum Message <> Reply to Message

On Tue, Nov 23, 2010 at 6:06 PM, Li Zefan <lizf@cn.fujitsu.com> wrote:
> Paul Menage wrote:
>> On Sun, Nov 21, 2010 at 8:06 PM, Colin Cross <ccross@android.com> wrote:
>>> The synchronize_rcu call in cgroup_attach_task can be very
>>> expensive. All fastpath accesses to task->cgroups that expect
>>> task->cgroups not to change already use task_lock() or
>>> cgroup_lock() to protect against updates, and, in cgroup.c,
>>> only the CGROUP_DEBUG files have RCU read-side critical
>>> sections.
>>
>> I definitely agree with the goal of using lighter-weight
>> synchronization than the current synchronize_rcu() call. However,
>> there are definitely some subtleties to worry about in this code.
>>
>> One of the reasons originally for the current synchronization was to
>> avoid the case of calling subsystem destroy() callbacks while there
>> could still be threads with RCU references to the subsystem state. The
>> fact that synchronize_rcu() was called within a cgroup_mutex critical
>> section meant that an rmdir (or any other significant cgrooup
>> management action) couldn't possibly start until any RCU read sections
>> were done.
>>
>> I suspect that when we moved a lot of the cgroup teardown code from
>> cgroup_rmdir() to cgroup_diput() (which also has a synchronize_rcu()
>> call in it) this restriction could have been eased, but I think I left
>> it as it was mostly out of paranoia that I was missing/forgetting some
>> crucial reason for keeping it in place.
>>
>> I'd suggest trying the following approach, which I suspect is similar
>> to what you were suggesting in your last email
>>
>> 1) make find_existing_css_set ignore css_set objects with a zero refcount
>> 2) change __put_css_set to be simply
>>
>> if (atomic_dec_and_test(&cg->refcount)) {
>>   call_rcu(&cg->rcu_head, free_css_set_rcu);
>> }
>
> If we do this, it's not anymore safe to use get_css_set(), which just
> increments the refcount without checking if it's zero.

I used an alternate approach, removing the css_set from the hash table
in put_css_set, but delaying the deletion to free_css_set_rcu. That

way, nothing can get another reference to the css_set to call
get_css_set on.

_____

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