Subject: Re: [PATCH 4/9] pid: Generalize task_active_pid_ns
Posted by ebiederm on Thu, 13 Dec 2007 16:22:39 GMT
View Forum Message <> Reply to Message

Oleg Nesterov <oleg@tv-sign.ru> writes:

> Sorry for the delay, and sorry, can't read this series carefully now.
> A couple of question though.
>
> On 12/12, Eric W. Biederman wrote:
>>
>> Currently task_active_pid_ns is not safe to call after a
>> task becomes a zombie and exit_task_namespaces is called,
>> as nsproxy becomes NULL. By reading the pid namespace from
>> the pid of the task we can trivially solve this problem
>
> Confused. If the task becomes a zombie, we can't assume it has a valid
> ->pids[].pid. The parent can release us as soon as exit_notify() drops
> tasklist.

Where this really matters is in the signal sending code. By the time
I have acquired sighand lock I know release_task has executed and
thus my ->pids[].pid is valid, because __exit_signal has not completed
and thus __unhash_process has not yet run.

When release_task_gets called we are EXIT_DEAD unhashed and unfindable
and I don't care. I do however care about finding my pid namespace
as long as the task is on hashed.

So as long as we have tasklist_lock or sighand lock and we can find
the task we are good.

What this allows me to do (as seen later in the patchset) is to send
to call pid_nr_ns and deliver a signal to a task group without caring
if the element of the task group I am talking to is a zombie or not.
Which is rather important when the task group leader has exited
and a zombie, but yet it is the task all of the signals are sent
to and group_send_siginfo is called on.

Eric

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers