Subject: Re: pspace child_reaper
Posted by serue on Tue, 29 Aug 2006 15:54:37 GMT
View Forum Message <> Reply to Message

Quoting Cedric Le Goater (clg@fr.ibm.com):
> Eric W. Biederman wrote:
> > Cedric Le Goater <clg@fr.ibm.com> writes:
> >
> >> Hello All,
> >>
> >> Eric, in your initial proof of concept on the pid namespace, you were
> >> defining a child_reaper per pid namespace.
> >>
> >> IMO, we can't use init_task as a child_reaper in a pid namespace because we
> >> will have pid collision which might result in a breakage of the init_task.
> >
> > The kernel doesn't use init_task (The idle thread) once it starts
> > init. Reaping children is the job of pid == 1.
>
> agree.
>
> >> Here are some questions on the model you intended to follow :
> >>
> >> Do you think we should have a child_reaper task per container ?
> > We have an init per container so yes.
>
> hmm, have we always ? what if i don't start an /sbin/init process in my
> newly created pid namespace or container. where do I collect all the SIGCHLD ?
>
> >> Could we use a kthread to do the job ?
> > Definitely not.
>
> why ?
>
> >> Could that kthread be global to all pid namespace ?
> >
> > Makes no sense.
>
> if you don't have an init per container, we need to find someone for the job.
>
> >> Any completely different idea on the topic ?
> > Init reaps the children, and I believe there are parts of user space
> > that depend on this. We shouldn't change that semantic.

What parts of userspace actually depend on this?

As far as I can tell, the closest we get is /sbin/init wanting to know
when children it spawned die so it can restart this. Clearly that will

happen anyway, even if there is one global reaper.

> IMHO, the only semantic i see is in the kernel, which needs someone to take
> care of sigchld. /sbin/init is a very good candidate bc it collects sigchld
> anyway and discards the ones it doesn't know about.

But don't confuse /sbin/init with the reaper. /sbin/init will only know about pids in it's own container, so if we do need to call to userspace for every task which the reaper hears about, then we will need a per-container reaper. But if that's not the case (and i haven't seen where it is), then userspace is simply not involved, and so one global reaper suffices, since it will know not about pid_ts but about struct pids == (container,pid_t).

-serge

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers