Subject: Re: [RFC][PATCH] Do not set /proc inode->pid for non-pid-related inodes
Posted by serue on Tue, 20 Mar 2007 14:58:12 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Dave Hansen <hansendc@us.ibm.com> writes:
>
> > On Mon, 2007-03-19 at 20:04 -0600, Eric W. Biederman wrote:
> >> Dave Hansen <hansendc@us.ibm.com> writes:
> >> Regardless I would like to see a little farther down on
> >> how we test to see if the pid namespace is alive and how we
> >> make these functions do nothing if it has died.
> >
> > That shouldn't be too hard. We have access to the superblock pretty
> > much everywhere, and we now store the pid_namespace in there (with some
> > patches I posted earlier).
>
> Sounds right. I don't think my original version had that. Which
> changes the rules a little bit.
>
> >> I would also
> >> like to see how we perform the appropriate lookups by pid namespace.
> >
> > What do you mean?
>
> proc_pid_readdir ... next_tgid().

next_tgid() is simple enough - we can always use current->pid_ns to find
the next pidnr.

The only hitch, as mentioned earlier, is how do we find the first task.
Currently task 1 is statically stored as the first inode, and as Dave
mentioned we can't do that now, because we dont' know of any one task
which will outlive the pid_ns.

> >> Basically I want to see how we finish up multiple namespace support
> >> for /proc before we start with the micro optimizations.
> >
> > Serge was tracking down some weird /proc issues and noticed that we
> > expect a pid_nr==1 for the pid namespace as long as it has a /proc
> > around. That is an assumption doesn't always hold now.
>
> Maybe. It really depends on how we define a namespace exiting.
> That must be in the lxc tree.
>
> There should be no code in the -mm or in Linus's tree that has
> that property.

True.

> While I'm not categorically opposed to supporting things like that it
> but it is something for which we need to tread very carefully because
> it is an extension of current semantics. I can't think of any weird
> semantics right now but for something user visible we will have to
> support indefinitely I don't see a reason to rush into it either.

Except that unless we mandate that pid1 in any namespace can't exit, and
put that feature off until later, we can't not address it.

> >> I'm fairly certain this patch causes us to do the wrong thing when
> >> the pid namespace exits, and I don't see much gain except for the
> >> death of find_get_pid.
> >
> > In the default, mainline case, it shouldn't be a problem at all. We
> > don't have the init pid namespace exiting.
>
> True but we are getting close. And it is about time we worked up
> patches for that so our conversations can become less theoretical.

Yes I really hope a patchset goes out today.

> > Shouldn't the lifetime of things under a /proc mount be tied to the life
> > of the mount, and not to the pid_namespace it is tied to? It seems
> > relatively sane to me to have a /proc empty of all processes, but still
> > have /proc/cpuinfo even if all of its processes are gone.
>
> That is what is implemented. When the pid namespace goes away there
> are no more pid directories, and the /proc/self symlink goes away.
> But everything else remains.
>
> If you look proc_root_readdir is not affected when the pid namespace
> goes away. Just proc_pid_readdir.
>
> Everything in fs/proc/base.c is about pid files in one way or another.
>
> > pid_delete_dentry() looks like the remaining place that really cares.
> > It would be pretty easy to have it check the pid namespace.
>
> Sure although it also needs the pid check for files that have it as
> the process can go away sooner.
>
> Eric
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers

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