Subject: Re: [PATCH v3] SUNRPC: protect service sockets lists during per-net shutdown

Posted by bfields on Mon, 20 Aug 2012 16:58:52 GMT

View Forum Message <> Reply to Message

On Mon, Aug 20, 2012 at 07:11:00PM +0400, Stanislav Kinsbursky wrote:

> >On Mon, Aug 20, 2012 at 03:05:49PM +0400, Stanislav Kinsbursky wrote:

> >>>Looking back at this:
> >>>
> >>> - adding the sv_lock looks like the right thing to do anyway
> >>>   independent of containers, because svc_age_temp_xprts may
> >>>   still be running.
> >>>
> >>> - I'm increasingly unhappy about sharing rpc servers between
> >>>   network namespaces. Everything would be easier to understand
> >>>   if they were independent. Can we figure out how to do that?
> >>>
> >>
> >>Could you, please, elaborate on your your unhappiness?
> >
> >It seems like you're having to do a lot of work on each individual rpc
> >server (callback server, lockd, etc.) to make per-net startup/shutdown
> >work. And then we still don't have it quite right (see the shutdown
> >races).)
> >
> >In general whenever we have the opportunity to have entirely separate
> >data structures, I'd expect that to simplify things: it should eliminate
> >some locking and reference-counting issues.
> >
>
> Agreed. But current solution still looks like the easies way to me
> to implement desired functionality.
>
> >>I.e. I don't like it too. But the problem here, is that rpc server
> >>is tied with kernel threads creation and destruction. And these
> >>threads can be only a part of initial pid namespace (because we have
> >>only one kthreadd). And we decided do not create new kernel thread
> >>per container when were discussing the problem last time.
> >
> >There really should be some way to create a kernel thread in a specific
> >namespace, shouldn't there?
> >
>
>
> Kthreads support in a container is rather a "political" problem,
> than an implementation problem.

Is there a mail thread somewhere with a summary of the objections?

> Currently, when you call kthread_create(), you add new job to
> kthreadd queue. Kthreadd is unique, starts right after init and
> lives in global initial environment. So, any kthread inherits
> namespaces from it.
> Of course, we can start one kthread per environment and change it's
> root or even network namespace in kthread function. But pid
> namespace of this kthread will remain global.

OK. But the current implementation will leave all the server threads in
the initial pid namespace, too.

> It looks like not a big problem, when we shutdown kthread by some
> variable. But what about killable nfsd kthreads?

And we're stuck with that problem either way too, aren't we?

> 1) We can't kill them from nested pid namespace.
> 2) How we will differ nfsd kthreads in initial pid namespace?

I have to admit for my purposes I don't care too much about pid
namespaces or about signalling server threads. It'd be nice to get
those things right but it wouldn't bother me that much not to.

Another stupid idea: can we do our own implementation of something like
kthreadd just for the purpose of starting rpc server threads? It
doesn't seem that complicated.

--b.

> In OpenVZ we have kthreadd per pid hamespace and it allows us to
> create kthreads (and thus services) per pid namespace.