
Subject: Re: [PATCH 0/7] containers (V7): Generic Process Containers

Posted by [ebiederm](#) on Tue, 20 Feb 2007 19:29:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Paul Menage" <menage@google.com> writes:

> What are you defining here as "everything"? If you mean "all things
> that could be applied to a segregated group of processes such as a
> virtual server", then "container" seems like a good name for my
> patches, since it allows you to aggregate namespaces, resource
> control, other virtualization, etc.

The aggregation happens at the task struct when you apply different pieces to a task we don't need a structure to allow aggregation.

> Sam said "the NSProxy *is* the container". You appear to be planning
> to have some namespaces, possibly not aggregated within the nsproxy
> (pid namespace?) but are you planning to have some higher-level
> "container" object that aggregates the nsproxy and the other
> namespaces?

No. A reverse mapping is not needed and is not interesting.
As long as I can walk all processes and ask what namespace are you in I don't care.

> If so, what is it? Does it track process membership, etc?
> What's the userspace API? In what fundamental ways would it be
> different from my generic containers patches with Serge Hallyn's
> namespace subsystem. (If these questions are answered already by
> designs or code, I'd be happy to be pointed at them).

> I guess what it comes down to, is why is an aggregation of namespaces
> suitable for the name "container", when an aggregation of namespaces
> and other resource controllers isn't?

To me at least the interesting part of your work is not the aggregation portion. But the infrastructure for building the different process groups.

You seem to be calling your groups of processes lumped together for one purpose or another a container.

> What do you think might be a better name for the generic process
> groups that I'm pushing? As I said, I'm happy to do a simple
> search/replace on my code to give a different name if that turned out
> to be the gating factor to getting it merged. But I'd be inclined to
> leave that decision up to Andrew/Linus.

As we have previously been using the term container is like application. The end product that the user sees built out of individual kernel provided components, but not something that exists in the kernel as such.

We need a term for the non-aggregated case for the individual process groups we build this out of in your infrastructure. Because you clearly have more than one kind of process group a set of processes can belong to.

>> > But there's a lot of common ground between these different approaches,
>> > and potential for synergy, so the point of this patch set is to
>> > provide a unification point for all of them, and a stepping stone for
>> > other new resource controllers and process control modules.

>>

>> For the case of namespaces I don't see how your code makes things
>> better. I do not see a real problem that you are solving.

>

> I'm trying to solve the problem that lots of different folks
> (including us) are trying to do things that aggregate multiple process
> into some kind of constrained group, and are all trying to use
> different and incompatible ways of grouping/tracking those processes.

>

> I agree that namespaces fit slightly less well into this model than
> some other subsystems like resource management. But by integrating
> with it you'd get automatic access to all the various different
> resource controller work that's being done.

I don't need to integrate with it to get access to the resource controller work. The resource controller work applies to a set of processes. I have a set of processes. I just need to apply the resource controllers to my set of processes.

At least when we don't talk about the names for these things it is that simple. Now global names I have issues with because almost always the global names fall into their own category, but that is a different story.

Now I have some half backed ideas that might be useful for providing a better abstraction. But it requires setting down and looking at the problem in detail, and understanding what people are trying to accomplish with these things they are building. What subset of process groups do you find interesting.

All that is necessary to have a group of processes do something in an unnamed fashion is to hang a pointer off of the task_struct. That's easy. You are adding a lot more to that, and there is

enough abstraction I haven't been able to easily work my way through the infrastructure to what is really going on. Admittedly I haven't looked closely yet.

Eric

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
