Subject: Re: [PATCH 0/7] containers (V7): Generic Process Containers
Posted by Paul Menage on Tue, 20 Feb 2007 23:28:07 GMT
View Forum Message <> Reply to Message

On 2/20/07, Sam Vilain <sam@vilain.net> wrote:
>
> I don't necessarily agree with the 'heirarchy' bit. It doesn't have to
> be so segregated. But I think we already covered that in this thread.

OK, but it's much easier to use a hierarchical system as a flat system (just don't create children) than a flat system as a hierarchical system. And others do seem to want a hierarchy for their process groupings.

>
> I agree with the comment on the abuse of the term "namespace", though
> consider that it has already been abused with the term IPC namespaces.

That doesn't seem like an abuse to me at all - you're controlling what IPC object a given name (shm_id, sem_id or msg_id) refers to for any given group of processes.

> We have for some time been using it to refer to groupable entities
> within the kernel that are associated with tasks, even if they don't
> involve named entities that clash within a particular domain. But there
> is always an entity and a domain, and that is the key point I'm trying
> to make - the features you are putting forward are no different to the
> examples that we made specifically for the purpose of setting the
> standard for further features to follow.

They're very similar, I agree. An important difference is that things like pid/mount namespaces are simply ways of controlling the visibility of existing unix concepts, such as processes or filesystems. You don't need additional configuration to make them useful, as unix already has standard ways of manipulating them.

Things like resource controllers typically require additional configuration to control how much resources each group of processes can consume, etc. Also, it appears to be much more common to want to move tasks between different resource controllers than to move them between different namespaces. (And in order to configure and move, you need to be able to name)

The configuration, naming and movement are the features that my container patch provides on top of the features that nsproxy provides for namespaces.

> anyway, feel free to flog this old dead horse and suggest different

> terms. We've all had long enough to think about it since so maybe it's
> worth it, but with any new term it should be really darned clear that
> they're essentially the same thing as namespaces, or otherwise be really
> likable.

What you're calling a "namespace", I'm calling a "subsystem state".
Essentially they're the same thing. The important thing that generic
containers provide is a standard way to manipulate "subsystem states"
or "namespaces".

Paul

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