Subject: Re: [PATCH 0/7] containers (V7): Generic Process Containers
Posted by Sam Vilain on Tue, 20 Feb 2007 22:58:13 GMT

View Forum Message <> Reply to Message

Paul Menage wrote:
>> The term "segregated group of processes" is too vague. Segregated for
>> what? What is the kernel supposed to do with this information?
>>
> The generic part of the kernel just keeps track of the fact that
> they're segregated (and their children, etc).
>
> It's the clients of this subsystem (virtual servers, resource
> controllers) that can decide to give different per-process behaviour
> based on the membership of various groups.
>

So those clients can use helper functions to use their own namespaces.
If they happen to group things in the same way they they'll all end up
using the same nsproxy.

>> Did you like the names I came up with in my original reply?
>>
>> - CPUset namespace for CPU partitioning
>> - Resource namespaces:
>>   - cpusched namespace for CPU
>>   - ulimit namespace for memory
>>   - quota namespace for disk space
>>   - io namespace for disk activity
>>   - etc
>>
>
> This is a strange abuse of the term "namespace".
> http://en.wikipedia.org/wiki/Namespace_%28computer_science%29
>
> For the virtual server work that you're doing, namespace is a good term:
>
> pids name processes, hence a pid namespace lets you have multiple
> distinct mappings from pids to processes
> filenames name files, so a filename (or mount) namespace lets you have
> multiple distinct mappings from filenames to files.
>
> For resource QoS control, it doesn't really make sense to talk about
> namespaces. We're not virtualizing resources to rename them for
> different virtual servers, we're limiting the quality of access to the
> resources.
>
> But the semantics of the term "namespace" notwithstanding, you're
> equating a virtual server namespace (pid, ipc, uts, mounts, etc) with

> a resource controller (memory, I/O, etc) in terms of their place in a
> hierarchy, which is something I agree with. All of these subsystems
> can be considered to be units that can be associated with groups of
> processes; the ultimate grouping of processes is something that we're
> both ultimately referring to as a container.
>

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.

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. 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.

We talked about naming a bit before, see http://lkml.org/lkml/2006/3/21/403 and possibly other threads. So, 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.

Sam.