

---

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

Posted by [Sam Vilain](#) on Tue, 13 Feb 2007 00:30:05 GMT

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

---

Paul Menage wrote:

>> I know I'm a bit out of touch, but AUI the NSProxy *\*is\** the container.  
>> We decided a long time ago that a container was basically just a set of  
>> namespaces, which includes all of the subsystems you mention.  
>>  
> You may have done that, but the CKRM/ResGroups independently decided a  
> long time ago that the fundamental unit was the resource class, and  
> the OpenVZ folks decided that the fundamental unit was the  
> BeanCounter, and the CPUSet folks decided that the fundamental unit  
> was the CPUSet, etc ... :-)  
>

There was a consensus that emerged that resulted in the nsproxy being included in the kernel. That is why I used "we". What was included varies greatly from the patch I put forward, as I mentioned.

You missed the point of what I was trying to say. Let me try again.

Originally I too thought that in order to begin on the path of virtualisation merging we would just make a simple container/vserver/whatever structure and hang everything off that.

I'll now say the same thing that was said before of my patch, that I don't think that adding the containers structure inside the kernel adds anything interesting or useful. In fact, I'd go further to say that very thing you think is a useful abstraction is locking yourself into inflexibility. This is what Eric recognised early on and eventually brought me around to the idea of too.

So, we have the current implementation - individual subsystems are virtualised, and it is the subsystems that you can see and work with. You can group together your virtualisation decisions and call that a container, great.

Ask yourself this - what do you need the container structure for so badly, that virtualising the individual resources does not provide for? You don't need it to copy the namespaces of another process ("enter") and you don't need it for checkpoint/migration. What does it mean to make a new container? You're making a "nothing" namespace for some as-yet-unspecified grouping of tasks. That's a great idea for a set of tightly integrated userland utilities to simplify the presentation to the admin, but I don't see why you need to enshrine this in the kernel. Certainly not for any of the other patches in your set as far as I can see.

> 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.  
>

That precisely echos my sentiment on my submissions of some 12 months ago.

Sam.

---

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

---