

Herbert Poetzl <herbert@13thfloor.at> writes:

>  
> sorry for the late answer, I almost missed that one ...  
>  
> yes, that sounds like an acceptable alternative, but  
> it might give some interesting issues with references  
> to devices ... for example:  
>  
> you mount a filesystem inside a namespace, so that  
> only the guest will see it (in theory) now you somehow  
> show that in the namespace copy too (on the host system)  
> and if some task decides to go camping there (cd into  
> that) it might keep the guest from unmounting that  
> device without ever knowing why ... or do you have some  
> smart solution to that?

lazy unmount.

>> net+pid+uts  
>>  
>> Not sure about uts, but I'm pretty sure the vserver folks want the  
>> ability to enter another existing network namespace, and both vserver  
>> and openvz have asked for the ability to enter pid namespaces.  
>  
> yes, definitely, pid and network namespaces have to  
> be accessible somehow, most administrative work is  
> done this way, when the administrator also maintains  
> the guests (i.e. doesn't want to bother accessing the  
> guest via special console/ssh/logon/whatever)  
>  
>> The pid namespaces could be solved by always generating as many pids for  
>> a process as it has parent pid\_namespaces. So if I'm in /vserver1, with  
>> one pid\_namespace above me, not only my init process has an entry in the  
>> root pid\_namespace (as I think has been suggested), but all my children  
>> will also continue to have pids in the root pid\_namespace.  
>  
> yep, sounds okay to me ...  
> note, our lightweight guests do not have an init  
> process, which is perfectly fine with the above, as  
> long as the init process is not considered a special  
> handle to the pid namespace :)  
>  
>> Or, if it is ok for the pid namespace operations to be as coarse as

```
>> "kill all processes in /vserver1", then that was going to be implemented
>> using the namespace container subsystem as:
>>
>> rm -rf /container_ns/vserver1
>
> that is definitely something you do not want to make
> the general signalling solution, because typically
> we have the following scenarios:
>
> - init less (lightweight) guest
>   + a bunch of shutdown scripts are executed
>   + term/kill is sent to the processes
>   + the context is disposed
>
> - init based guest
>   + a signal is sent to init
>   + init executes the shutdown and kills off
>     the 'other' processes
>   + init finally calls reboot/halt
>   + init and the context are disposed
```

I have seen the same thing invented in a different context so this sounds like a common pattern.

```
>> Any other (a) requirements, (b) ideas for alternate pid and network
>> ns management without allowing namespace enters?
>
> entering the spaces seems most natural and quite
> essential to me, especially for administration and
> debugging purposes ...
```

Yes. But how you implement the enter need not be modifying the namespace pointer in a task\_struct/nsproxy. You can get the same user effect in other ways, which are potentially more secure.

Eric

---

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

---