Subject: Re: [PATCH 1/4] Virtualization/containers: introduction
Posted by ebiederm on Tue, 07 Feb 2006 15:42:12 GMT
View Forum Message <> Reply to Message

Sam Vilain <sam@vilain.net> writes:

```
> Rik van Riel wrote:
>> On Mon, 6 Feb 2006, Eric W. Biederman wrote:
>>
>>> We are never going to form a consensus if all of the people doing
>>> implementations don't talk.
>> Speaking of which - it would be interesting to get Kirill's
>> comments on Eric's patchset ;)
>> Once we know what's good and bad about both patchsets, we'll
>> be a lot closer to knowing what exactly should go upstream.
>
> Let's compare approaches of patchsets before the patchsets themselves.
>
> It seems to be, should we:
>
>    A) make a general form of virtualising PIDs, and hope this assists
>       later virtualisation efforts (Eric's patch)
>
>    B) make a general form of containers/jails/vservers/vpses, and layer
>       PID virtualisation on top of it somewhere (as in openvz, vserver)
>
> I can't think of any real use cases where you would specifically want A)
> without B).
```

You misrepresent my approach.

First there is a huge commonality in the code bases between the different implementations and I have already gotten preliminary acceptance from the vserver developers, that my approach is sane. The major difference is what user interface does the kernel export, and I posted my user interface.

What user interface to export is a debate worth having.

For a lot of things getting the details just so is very important to long term maintainability and it is not my impression that anyone has done that yet.

Second I am not trying to just implement a form of virtualizing PIDs. Heck I don't intend to virtualize anything. The kernel has already virtualized everything I require. I want to implement multiple

instances of the current kernel global namespaces. All I want is
to be able to use the same name twice in user space and not have
a conflict.

Beyond getting multiple instance of all of the kernel namespaces
(which is the hard requirement for migration) my approach is to
see what is needed for projects like vserver and vps and see how
their needs can be met as well.

I disagree with a struct container simply because I do not see what
value it happens to bring to the table. I have yet to see a problem
that it solves that I have not solved yet.

In addition I depart from vserver and other implementations in another
regard. It is my impression a lot of their work has been done so
those projects are maintainable outside of the kernel, which makes
sense as that is where those code bases live. But I don't think that
gives the best solution for an in kernel implementation, which is
what we are implementing.

So far I have succeeded in communicating with both the IBM and
vserver developers. Hopefully I can do the same with Kirill Korotaev
and the OpenVz team.   I think my implementation stands up to
criticism. But expect surprises in the way I solve a number of
problems.

I suspect I will find similar surprises in the OpenVz code.

Time to do some more research I guess.

Eric