Subject: Re: [PATCH 1/4] Virtualization/containers: introduction
Posted by ebiederm on Tue, 07 Feb 2006 17:20:46 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

```
>>>I can't think of any real use cases where you would specifically want A)
>>>without B).
>
>> You misrepresent my approach.
> [...]
>
>> 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.
> if you want not virtualize anything, what is this discussion about? :)
> can you provide an URL to your sources? you makes lot's of statements about that
> your network virtualization solution is better/more complete, so I'd like to see
> your solution in whole rather than only words.
> Probably this will help.
```

Sure.

I think it is more an accident of time, and the fact that I am quite
proud of where I am at. You quite likely have improved things in openvz
since last I looked as well. Currently my code quality is only a proof
of concept but the tree is below.

git://git.kernel.org/pub/scm/linux/kernel/git/ebiederm/linux -2.6-ns.git/

Basically the implementation appears to the user as a separate instance
of the network stack. Since the tree was experimental the path is
absolutely horrible. I am in the process of cleaning and redoing all
of that now in preparation for kernel inclusion.

But I suspect I am in the lead as no one else had noticed the ipv6
reference counting bugs.

```
>> 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.
> again, source would help to understand your solution and problem you solved and
> not solved yet.
```

Above. But at least with pids it has all been posted on the mailing list.

I think I have solved most of the code structural issues and the big
kernel API issues. A lot of the little things I have not gotten to yet
as I figured it was best approached later.

>> 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.
> These soltuions are in kernel implementations actually.

Sorry in/out in this context I was referring to the stock linux kernel.
As soon as I had a viable proof of concept I began working to get
my code merged.

Eric