Subject: Re: [RFC] network namespaces
Posted by Daniel Lezcano on Tue, 05 Sep 2006 13:34:40 GMT
View Forum Message <> Reply to Message

Hi all,

>> This complete separation of namespaces is very useful for at least two
>> purposes:
>> - allowing users to create and manage by their own various tunnels and
>> VPNs, and
>> - enabling easier and more straightforward live migration of groups of
>> processes with their environment.
>
>
> I conceptually prefer this approach, but I seem to recall there were
> actual problems in using this for checkpoint/restart of lightweight
> (application) containers. Performance aside, are there any reasons why
> this approach would be problematic for c/r?

I agree with this approach too, separated namespaces is the best way to
identify the network ressources for a specific container.

> I'm afraid Daniel may be on vacation, and don't know who else other than
> Eric might have thoughts on this.

Yes, I was in "vacation", but I am back :)

>>2. People expressed concerns that complete separation of namespaces
>> may introduce an undesired overhead in certain usage scenarios.
>> The overhead comes from packets traversing input path, then output path,
>> then input path again in the destination namespace if root namespace
>> acts as a router.

Yes, performance is probably one issue.

My concerns was for layer 2 / layer 3 virtualization. I agree a layer 2
isolation/virtualization is the best for the "system container".
But there is another family of container called "application container",
it is not a system which is run inside a container but only the
application. If you want to run a oracle database inside a container,
you can run it inside an application container without launching <init>
and all the services.

This family of containers are used too for HPC (high performance
computing) and for distributed checkpoint/restart. The cluster runs
hundred of jobs, spawning them on different hosts inside an application
container. Usually the jobs communicates with broadcast and multicast.
Application containers does not care of having different MAC address and

rely on a layer 3 approach.

Are application containers comfortable with a layer 2 virtualization ? I don't think so, because several jobs running inside the same host communicate via broadcast/multicast between them and between other jobs running on different hosts. The IP consumption is a problem too: 1 container == 2 IP (one for the root namespace/ one for the container), multiplicated with the number of jobs. Furthermore, lot of jobs == lot of virtual devices.

However, after a discussion with Kirill at the OLS, it appears we can merge the layer 2 and 3 approaches if the level of network virtualization is tunable and we can choose layer 2 or layer 3 when doing the "unshare". The determination of the namespace for the incoming traffic can be done with an specific iptable module as a first step. While looking at the network namespace patches, it appears that the TCP/UDP part is **very** similar at what is needed for a layer 3 approach.

Any thoughts ?

Daniel