Subject: Re: Network virtualization/isolation
Posted by ebiederm on Mon, 04 Dec 2006 12:15:00 GMT
View Forum Message <> Reply to Message

jamal <hadi@cyberus.ca> writes:

> I have removed the Re: just to add some freshness to the discussion
>
> So i read quickly the rest of the discussions. I was almost suprised to
> find that i agree with Eric on a lot of opinions (we also agree that
> vindaloo is good for you i guess);->
> The two issues that stood out for me (in addition to what i already said
> below):
>
> 1) the solution must ease the migration of containers; i didnt see
> anything about migrating them to another host across a network, but i
> assume that this is a given.

It is mostly a given. It is a goal for some of us and not for others.
Containers are a necessary first step to getting migration and checkpoint/restart
assistance from the kernel.

> 2) the socket level bind/accept filtering with multiple IPs. From
> reading what Herbert has, it seems they have figured a clever way to
> optimize this path albeit some challenges (speacial casing for raw
> filters) etc.
>
> I am wondering if one was to use the two level muxing of the socket
> layer, how much more performance improvement the above scheme provides
> for #2?

I don't follow this question.

> Consider the case of L2 where by the time the packet hits the socket
> layer on incoming, the VE is already known; in such a case, the lookup
> would be very cheap. The advantage being you get rid of the speacial
> casing altogether. I dont see any issues with binds per multiple IPs etc
> using such a technique.
>
> For the case of #1 above, wouldnt it be also easier if the tables for
> netdevices, PIDs etc were per VE (using the 2 level mux)?

Generally yes. s/VE/namespace/. There is a case with hash tables where
it seems saner to add an additional entry because hash it is hard to dynamically
allocate a hash table, (because they need something large then a
single page allocation). But for everything else yes it makes things
much easier if you have a per namespace data structure.

A practical question is can we replace hash tables with some variant of trie or radix-tree and not take a performance hit. Given the better scaling of tress to different workload sizes if we can use them so much the better. Especially because a per namespace split gives us a lot of good properties.

> In any case, folks, i hope i am not treading on anyones toes; i know
> each one of you has implemented and has users and i am trying to be as
> neutral as i can (but clearly biased;->).

Well we rather expect to bash heads until we can come up with something we all can agree on with the people who more regularly have to maintain the code. The discussions so far have largely been warm ups, to actually doing something.

Getting feedback from people who regularly work with the networking stack is appreciated.

Eric
