Subject: Re: [patch 2/6] [Network namespace] Network device sharing by view
Posted by Herbert Poetzl on Wed, 28 Jun 2006 17:18:37 GMT
View Forum Message <> Reply to Message

On Wed, Jun 28, 2006 at 09:36:40AM -0600, Eric W. Biederman wrote:
> Herbert Poetzl <herbert@13thfloor.at> writes:
>
> > On Wed, Jun 28, 2006 at 06:31:05PM +1200, Sam Vilain wrote:
> >> Eric W. Biederman wrote:
> >> > Have a few more network interfaces for a layer 2 solution
> >> > is fundamental. Believing without proof and after arguments
> >> > to the contrary that you have not contradicted that a layer 2
> >> > solution is inherently slower is non-productive. Arguing
> >> > that a layer 2 only solution most prove itself on guest to guest
> >> > communication is also non-productive.
> >> >
> >>
> >> Yes, it does break what some people consider to be a sanity
> >> condition when you don't have loopback anymore within a guest. I
> >> once experimented with using 127.* addresses for per-guest loopback
> >> devices with vserver to fix this, but that couldn't work without
> >> fixing glibc to not make assumptions deep in the bowels of the
> >> resolver. I logged a fault with gnu.org and you can guess where it
> >> went :-).
> >
> > this is what the lo* patches address, by providing
> > the required loopback isolation and providing lo
> > inside a guest (i.e. it looks and feels like a
> > normal system, except that you cannot modify the
> > interfaces from inside)
>
> Ok. This is new. How do you talk between guests now?

> Before those patches it was through IP addresses on the loopback
> interface as I recall.

no, that was probably your assumption, the IPs are
assigned (in a perfectly normal way) to the interfaces
(e.g. eth0 carries some IPs for guest A and B, eth1
carries others for guest C). the way the linux network
stack works, local addresses (i.e. those of A,B and C)
will automatically communicate via loopback (as they
are local) while outbound traffic will use the proper
interface (nothing is changed here)

the difference in the lo patches is, that we allow to
use the 'localhost' ip range (127.x.x.x) by isolating
traffic (in this range) on the loopback interface

(which typically allows to have 127.0.0.1 and lo
visible inside a guest)

> >> > With a guest with 4 IPs
> >> > 10.0.0.1 192.168.0.1 172.16.0.1 127.0.0.1
> >> > How do you make INADDR_ANY work with just filtering at bind time?
> >> >
> >>
> >> It used to just bind to the first one. Don't know if it still does.
> >
> > no, it _alway_ binds to INADDR_ANY and checks
> > against other sockets (in the same context)
> > comparing the lists of assigned IPs (the subset)
> >
> > so all checks happen at bind/connect time and
> > always against the set of IPs, only exception is
> > a performance optimization we do for single IP
> > guests (where INADDR_ANY gets rewritten to the
> > single IP)
>
> What is the mechanism there?
>
> My rough extrapolation says this mechanism causes problems when
> migrating between machines.

that might be, as we do not consider migration such
important as other folks do :)

> In particular it sounds like only one process can bind to *:80, even
> if it is only allowed to accept connections from a subset of those
> IPs.

no, guest A,B and C can all bind to *:80 and coexist
quite fine, given that they do not have any IP in
the intersection of their subsets (which is checked
at bind time)

> So if on another machine I bound something to *:80 and only allowed to
> use a different set of IPs and then attempted to migrate it, the
> migration would fail because I could not restart the application,
> with all of it's layer 3 resources.

actually I do not see why, unless the destination
has a conflict on the ip subset, in which case you
would end up with a migrated, but not working guest :)

> To be clear I assume when I migrate I always take my IP address or
> addresses with me.

that's fine, the only requirement would be that the host has a superset of the IP addresses used by the guests ...

HTC,
Herbert

> Eric