Subject: Network namespaces without isolation
Posted by Andreas B Aaen on Wed, 02 Jul 2008 07:18:50 GMT
View Forum Message <> Reply to Message

Hi,

I am looking into the network namespace implementation because I need an IP stack that is capable of talking with a number of separate IP nets with possible overlapping IP adresses. My connection to each separate IP-net is through a tunnel e.g. a VLAN interface.

A special application will then be able to listen to traffic on all the nets through a socket option SO_NS that sets the namespace to talk/listen to for a particular socket. For this to work network namespaces needs to be indexed. It would also be very handy if the configuration can be made without a clone() call.
Something like:

ip ns add ns 1
ip link set eth0.42 ns 1
ip addr add 192.168.50.4/24 dev eth0.42 ns 1

It would be fairly ok if this possibility to set up interfaces on other namespaces only is possible from the default namespace.

It would also be nice to be able to see the network statistics from all the namespaces through the proc filesystem at least in an uncloned (isolated) namespace.

So you would be able to see the network statistics in /proc/net/ns/<index>/

It should be said that we have an implementation of all this already, but NOT based on network namespaces and for elder kernels. We don't want to forward port this, put instead add a few features to the network namespace implementation to be able to fullfill the requirement of our application:
talk to a number of IP networks with possible overlapping IP addresses.

Regards,
--
Andreas Bach Aaen   System Developer, M. Sc.
Tieto Enator A/S   tel: +45 89 38 51 00
Skanderborgvej 232   fax: +45 89 38 51 01
8260 Viby J   Denmark   andreas.aaen@tietoenator.com

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: Re: Network namespaces without isolation
Posted by ebiederm on Fri, 04 Jul 2008 09:52:08 GMT

View Forum Message <> Reply to Message

Andreas B Aaen <andreas.aaen@tietoenator.com> writes:

> Hi,
>
> I am looking into the network namespace implementation because I need an IP
> stack that is capable of talking with a number of separate IP nets with
> possible overlapping IP adresses. My connection to each separate IP-net is
> through a tunnel e.g. a VLAN interface.
>
> A special application will then be able to listen to traffic on all the nets
> through a socket option SO_NS that sets the namespace to talk/listen to for a
> particular socket. For this to work network namespaces needs to be indexed.

Answering part of your question. As currently designed you can use multiple
network namespaces in a single task, and you can place each vlan interface
in different network namespace. However the current model is most cumbersome
for doing so.

You can use unshare instead of clone which is a little easier.

A socket option sounds like a nice idea.

The two challenges are what names to use to refer to network namespaces
and how to get network namespaces to persist.

There have been a number of discussions about identifiers none of which
have led to any sort of agreement. One of the goals in the design is
that we don't introduce new global identifiers allowing us to ultimately
have nested containers.

So far we have been referring to namespaces indirectly by the pids of the
processes which are using them.

> It would also be nice to be able to see the network statistics from all the
> namespaces through the proc filesystem at least in an uncloned (isolated)
> namespace.

Currently this is possible by looking at /proc/<pid>/net.

> So you would be able to see the network statistics in /proc/net/ns/<index>/

One of the things we have tried to do is to keep the number of new interfaces
to a minimum.

> It should be said that we have an implementation of all this already, but NOT

> based on network namespaces and for elder kernels. We don't want to forward
> port this, put instead add a few features to the network namespace
> implementation to be able to fullfill the requirement of our application:
> talk to a number of IP networks with possible overlapping IP addresses.

If we can work out the details on how to do that cleanly it seems totally reasonable to enhance network namespaces in that direction. You are not the first to express those kind of requirements, and probably won't be the last.

Eric

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: Network namespaces without isolation
Posted by Andreas B Aaen on Fri, 04 Jul 2008 15:07:28 GMT
View Forum Message <> Reply to Message

On Friday 04 July 2008 11:52, Eric W. Biederman wrote:
> Andreas B Aaen <andreas.aaen@tietoenator.com> writes:

> Answering part of your question. As currently designed you can use
> multiple network namespaces in a single task, and you can place each vlan
> interface in different network namespace. However the current model is
> most cumbersome for doing so.
>
> You can use unshare instead of clone which is a little easier.

How do you actually use multiple name spaces in the current implementation in the same task if you refer to them indirectly through pids?
So if I need 500 network namespaces then I need to fork 500 processes.

> A socket option sounds like a nice idea.

And quite easy to implement except for the handling of which network namespaces you should be allowed to talk to.

> The two challenges are what names to use to refer to network namespaces
> and how to get network namespaces to persist.

Exactly. In my current proof of concept implementation indexed/named network namespaces are created through an extended netlink interface instead of the clone/unshare calls. Delete of the namespaces are also through a netlink interface. E.g.:

ip netns add 1
(adds a network namespace with the "name" index 1)
ip netns del 1
(deletes it again)

> There have been a number of discussions about identifiers none of which
> have led to any sort of agreement. One of the goals in the design is
> that we don't introduce new global identifiers allowing us to ultimately
> have nested containers.

In this case this means that the index' should be a namespace of itself just
like pids. It seems to be overkill. At lest for my purpose.

> So far we have been referring to namespaces indirectly by the pids of the
> processes which are using them.

Right. And with namespaces into namespaces and usage of pid namespaces you
could have two different namespaces named with the same numerical value of
pid.

> > It would also be nice to be able to see the network statistics from all
> > the namespaces through the proc filesystem at least in an uncloned
> > (isolated) namespace.
>
> Currently this is possible by looking at /proc/<pid>/net.

Which was what lead me to the question of how you can have more name spaces in
a single task with the current implementation.

> > So you would be able to see the network statistics in
> > /proc/net/ns/<index>/

Or maybe this should have been /proc/<pid>/net/<index>/ ?

> One of the things we have tried to do is to keep the number of new
> interfaces to a minimum.

Sure.

> If we can work out the details on how to do that cleanly it seems totally
> reasonable to enhance network namespaces in that direction. You are not
> the first to express those kind of requirements, and probably won't be the
> last.

So it seems that we need to restart the naming discussion.

Regards,
--

Andreas Bach Aaen System Developer, M. Sc.
Tieto Enator A/S tel: +45 89 38 51 00
Skanderborgvej 232 fax: +45 89 38 51 01
8260 Viby J Denmark andreas.aaen@tietoenator.com

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: Network namespaces without isolation
Posted by ebiederm on Fri, 04 Jul 2008 22:45:38 GMT
View Forum Message <> Reply to Message

Andreas B Aaen <andreas.aaen@tietoenator.com> writes:
>
> How do you actually use multiple name spaces in the current implementation in
> the same task if you refer to them indirectly through pids?
> So if I need 500 network namespaces then I need to fork 500 processes.

Currently sockets are explicitly tagged with the namespace they are in.
And you can pass sockets between processes. The only case we have where
we need to refer to a namespace other then our current one is when
passing network devices between namespaces.

ip link set eth0 netns <pid>.

>> A socket option sounds like a nice idea.
>
> And quite easy to implement except for the handling of which network
> namespaces you should be allowed to talk to.

Yes.

>> There have been a number of discussions about identifiers none of which
>> have led to any sort of agreement. One of the goals in the design is
>> that we don't introduce new global identifiers allowing us to ultimately
>> have nested containers.
>
> In this case this means that the index' should be a namespace of itself just
> like pids. It seems to be overkill. At lest for my purpose.

Well the index should be in a namespace and hopefully an existing namespace.

>> So far we have been referring to namespaces indirectly by the pids of the
>> processes which are using them.
>
> Right. And with namespaces into namespaces and usage of pid namespaces you

> could have two different namespaces named with the same numerical value of
> pid.

I each pid namespaces the pid values are unique, and since the pid namespace is hierarchical with all pids showing up in the initial pid namespace.

>> > It would also be nice to be able to see the network statistics from all
>> > the namespaces through the proc filesystem at least in an uncloned
>> > (isolated) namespace.
>>
>> Currently this is possible by looking at /proc/<pid>/net.
>
> Which was what lead me to the question of how you can have more name spaces in
> a single task with the current implementation.

Well you can't look at them very well.

>> > So you would be able to see the network statistics in
>> > /proc/net/ns/<index>/
>
> Or maybe this should have been /proc/<pid>/net/<index>/ ?

I'm not certain. If I can figure out how to break /proc/net into it's own filesystem it gets easier to manage. Despite being possible I haven't figured out how to do that cleanly yet.

>> If we can work out the details on how to do that cleanly it seems totally
>> reasonable to enhance network namespaces in that direction. You are not
>> the first to express those kind of requirements, and probably won't be the
>> last.
>
> So it seems that we need to restart the naming discussion.

I had the clear realization a while back that if we had done this cleanly everything would have been in the mount namespace from the start.

Since we have the problem of when to free a network namespace as well as how to open a socket I have relatively simple suggestion. Create a netnsfs. By default have that filesystem refer to the mounters current network namespace or if passed -o create generate a new one.

Opening a socket is a bit trickier to map because sockets have the triple of options. But I expect you could open a socket like:
open("/path/to/netnsfs/mount/inet;stream;tcp", O_RDWR | O_CREAT, 0777);

I suppose a socket option that takes the path to the mount point might also work. Just generating the socket cleanly the first time seems better though.

With a netnsfs you could make the lifetime of the namespace the lifetime of the mount.
Which you could make automatically reap when you process or set of processes go away
by using a new mount namespace to hold it all in.

Eric