Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Daniel Lezcano on Tue, 12 Dec 2006 13:59:31 GMT
View Forum Message <> Reply to Message

Dmitry Mishin wrote:
> Temporary code to debug and play with pass-through device.
> Create device pair by
> modprobe veth
> echo 'add veth1 0:1:2:3:4:1 eth0 0:1:2:3:4:2' >/proc/net/veth_ctl
> and your shell will appear into a new namespace with `eth0' device.
> Configure device in this namespace
> ip l s eth0 up
> ip a a 1.2.3.4/24 dev eth0
> and in the root namespace
> ip l s veth1 up
> ip a a 1.2.3.1/24 dev veth1
> to establish a communication channel between root namespace and the newly
> created one.
>
> Code is done by Andrey Savochkin and ported by me over Cedric'c patchset
>
> Signed-off-by: Dmitry Mishin <dim@openvz.org>
>

[...]

>
> --- linux-2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
> +++ linux-2.6.19-rc6-mm2/include/linux/net_namespace.h
> @@ -24,6 +24,9 @@ struct net_namespace {
> int fib4_trie_last_dflt;
> #endif
> unsigned int hash;
> + struct net_namespace *parent;
> + struct list_head child_list, sibling_list;
> + unsigned int id;
> };

Why do yo need to have a child list and sibling list ?

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-throughdevice
Posted by Mishin Dmitry on Tue, 12 Dec 2006 14:04:20 GMT
View Forum Message <> Reply to Message

On Tuesday 12 December 2006 16:59, Daniel Lezcano wrote:

> Dmitry Mishin wrote:
> > Temporary code to debug and play with pass-through device.
> > Create device pair by
> > modprobe veth
> > echo 'add veth1 0:1:2:3:4:1 eth0 0:1:2:3:4:2' >/proc/net/veth_ctl
> > and your shell will appear into a new namespace with `eth0' device.
> > Configure device in this namespace
> > ip l s eth0 up
> > ip a a 1.2.3.4/24 dev eth0
> > and in the root namespace
> > ip l s veth1 up
> > ip a a 1.2.3.1/24 dev veth1
> > to establish a communication channel between root namespace and the newly
> > created one.
> >
> > Code is done by Andrey Savochkin and ported by me over Cedric'c patchset
> >
> > Signed-off-by: Dmitry Mishin <dim@openvz.org>
> >
>
> [...]
>
> >
> > --- linux-2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
> > +++ linux-2.6.19-rc6-mm2/include/linux/net_namespace.h
> > @@ -24,6 +24,9 @@ struct net_namespace {
> >   int   fib4_trie_last_dflt;
> > #endif
> >   unsigned int hash;
> > + struct net_namespace *parent;
> > + struct list_head child_list, sibling_list;
> > + unsigned int id;
> > };
>
> Why do yo need to have a child list and sibling list ?
Because of the level2<->level3 hierarchy, for example.

--
Thanks,
Dmitry.

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Daniel Lezcano on Tue, 12 Dec 2006 14:10:21 GMT
View Forum Message <> Reply to Message

Dmitry Mishin wrote:

> On Tuesday 12 December 2006 16:59, Daniel Lezcano wrote:
>> Dmitry Mishin wrote:
>>> Temporary code to debug and play with pass-through device.
>>> Create device pair by
>>> modprobe veth
>>> echo 'add veth1 0:1:2:3:4:1 eth0 0:1:2:3:4:2' >/proc/net/veth_ctl
>>> and your shell will appear into a new namespace with `eth0' device.
>>> Configure device in this namespace
>>> ip l s eth0 up
>>> ip a a 1.2.3.4/24 dev eth0
>>> and in the root namespace
>>> ip l s veth1 up
>>> ip a a 1.2.3.1/24 dev veth1
>>> to establish a communication channel between root namespace and the newly
>>> created one.
>>>
>>> Code is done by Andrey Savochkin and ported by me over Cedric'c patchset
>>>
>>> Signed-off-by: Dmitry Mishin <dim@openvz.org>
>>>
>> [...]
>>
>>> --- linux-2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
>>> +++ linux-2.6.19-rc6-mm2/include/linux/net_namespace.h
>>> @@ -24,6 +24,9 @@ struct net_namespace {
>>>   int   fib4_trie_last_dflt;
>>> #endif
>>>   unsigned int hash;
>>> + struct net_namespace *parent;
>>> + struct list_head child_list, sibling_list;
>>> + unsigned int id;
>>> };
>> Why do yo need to have a child list and sibling list ?
> Because of the level2<->level3 hierarchy, for example.

This hierarchy doesn't exist with ns->parent ? Do you have an example
when the hierarchy should be used ? I mean when we need to browse from
l2 -> l3 ?

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Mishin Dmitry on Tue, 12 Dec 2006 14:12:55 GMT
View Forum Message <> Reply to Message

On Tuesday 12 December 2006 17:10, Daniel Lezcano wrote:
> Dmitry Mishin wrote:
> > On Tuesday 12 December 2006 16:59, Daniel Lezcano wrote:

> >> Dmitry Mishin wrote:
> >>> Temporary code to debug and play with pass-through device.
> >>> Create device pair by
> >>> modprobe veth
> >>> echo 'add veth1 0:1:2:3:4:1 eth0 0:1:2:3:4:2' >/proc/net/veth_ctl
> >>> and your shell will appear into a new namespace with `eth0' device.
> >>> Configure device in this namespace
> >>> ip l s eth0 up
> >>> ip a a 1.2.3.4/24 dev eth0
> >>> and in the root namespace
> >>> ip l s veth1 up
> >>> ip a a 1.2.3.1/24 dev veth1
> >>> to establish a communication channel between root namespace and the newly
> >>> created one.
> >>>
> >>> Code is done by Andrey Savochkin and ported by me over Cedric'c patchset
> >>>
> >>> Signed-off-by: Dmitry Mishin <dim@openvz.org>
> >>>
> >> [...]
> >>
> >>> --- linux-2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
> >>> +++ linux-2.6.19-rc6-mm2/include/linux/net_namespace.h
> >>> @@ -24,6 +24,9 @@ struct net_namespace {
> >>>   int   fib4_trie_last_dflt;
> >>> #endif
> >>>   unsigned int hash;
> >>> + struct net_namespace *parent;
> >>> + struct list_head child_list, sibling_list;
> >>> + unsigned int id;
> >>> };
> >> Why do yo need to have a child list and sibling list ?
> > Because of the level2<->level3 hierarchy, for example.
>
> This hierarchy doesn't exist with ns->parent ? Do you have an example
> when the hierarchy should be used ? I mean when we need to browse from
> l2 -> l3 ?
For example, to check that new ifaddr is already used by child l3 namespace.

--
Thanks,
Dmitry.

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Daniel Lezcano on Tue, 12 Dec 2006 14:19:56 GMT

Dmitry Mishin wrote:

>>>> Why do yo need to have a child list and sibling list ?
>>> Because of the level2<->level3 hierarchy, for example.
>> This hierarchy doesn't exist with ns->parent ? Do you have an example
>> when the hierarchy should be used ? I mean when we need to browse from
>> l2 -> l3 ?
> For example, to check that new ifaddr is already used by child l3 namespace.

The devinet isolation does already do that, you can not add a new ifaddr
if it already exists. Do you have another example ?

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-throughdevice
Posted by Mishin Dmitry on Tue, 12 Dec 2006 14:26:43 GMT

On Tuesday 12 December 2006 17:19, Daniel Lezcano wrote:
> Dmitry Mishin wrote:
>
> >>>> Why do yo need to have a child list and sibling list ?
> >>> Because of the level2<->level3 hierarchy, for example.
> >> This hierarchy doesn't exist with ns->parent ? Do you have an example
> >> when the hierarchy should be used ? I mean when we need to browse from
> >> l2 -> l3 ?
> > For example, to check that new ifaddr is already used by child l3 namespace.
>
> The devinet isolation does already do that, you can not add a new ifaddr
> if it already exists. Do you have another example ?
Could devinet isolation provide ifaddrs list with namespaces?
What will be with child namespaces if you decide to destroy parent namespace?
If we decide to destroy them, than how we could get their list?
It is a question of flexibility and easy management.
Why do you want to remove this code?

--
Thanks,
Dmitry.

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Vlad Yasevich on Tue, 12 Dec 2006 14:52:11 GMT

Dmitry Mishin wrote:
> On Tuesday 12 December 2006 17:19, Daniel Lezcano wrote:
>> Dmitry Mishin wrote:
>>
>>>>>> Why do yo need to have a child list and sibling list ?
>>>>> Because of the level2<->level3 hierarchy, for example.
>>>> This hierarchy doesn't exist with ns->parent ? Do you have an example
>>>> when the hierarchy should be used ? I mean when we need to browse from
>>>> l2 -> l3 ?
>>> For example, to check that new ifaddr is already used by child l3 namespace.
>> The devinet isolation does already do that, you can not add a new ifaddr
>> if it already exists. Do you have another example ?
> Could devinet isolation provide ifaddrs list with namespaces?

I hope the answer is yes... It seems to me that we do way to many lookups like this:

```
+ rcu_read_lock();
+ in_dev = __in_dev_get_rcu(dev);
+ if (!in_dev)
+  goto no_in_dev;
+
+ for_ifa(in_dev) {
```

in the proposed L3 code.

> What will be with child namespaces if you decide to destroy parent namespace?
> If we decide to destroy them, than how we could get their list?

I think they should be destroyed as well. This is where the child_list will/should be used.

However, I don't see a need for sibling_list until interface migration is done.

-vlad

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Daniel Lezcano on Tue, 12 Dec 2006 15:50:50 GMT
View Forum Message <> Reply to Message

Dmitry Mishin wrote:
> On Tuesday 12 December 2006 17:19, Daniel Lezcano wrote:
>> Dmitry Mishin wrote:
>>

>>>>>> Why do yo need to have a child list and sibling list ?
>>>>> Because of the level2<->level3 hierarchy, for example.
>>>> This hierarchy doesn't exist with ns->parent ? Do you have an example
>>>> when the hierarchy should be used ? I mean when we need to browse from
>>>> l2 -> l3 ?
>>> For example, to check that new ifaddr is already used by child l3 namespace.
>> The devinet isolation does already do that, you can not add a new ifaddr
>> if it already exists. Do you have another example ?
> Could devinet isolation provide ifaddrs list with namespaces?
> What will be with child namespaces if you decide to destroy parent namespace?
> If we decide to destroy them, than how we could get their list?
> It is a question of flexibility and easy management.
> Why do you want to remove this code?

I don't want to especially remove this code, I just want to understand
what it does and why. If it appears to be useless, let's remove it, if
it appears to be useful, let's keep it.

By the way, what is the meaning on destroying the namespaces directly,
is it not the kref mechanism which needs to do that ? For example, if
you create a l2 namespace and after you create l3 namespaces. You want
to destroy the l2 namespace, the l2 namespace should stay "zombie" until
all the l3 namespaces exit. If you need to wipe out all the namespaces,
you should destroy all the related namespaces' ressources, like killing
all processes inside it. The namespaces will "put" their respective kref
and will trigger the freeing of the ressources.

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Herbert Poetzl on Wed, 13 Dec 2006 07:18:15 GMT
View Forum Message <> Reply to Message

On Tue, Dec 12, 2006 at 04:50:50PM +0100, Daniel Lezcano wrote:
> Dmitry Mishin wrote:
> > On Tuesday 12 December 2006 17:19, Daniel Lezcano wrote:
> >> Dmitry Mishin wrote:
> >>
> >>>>>>> Why do yo need to have a child list and sibling list ?
> >>>>> Because of the level2<->level3 hierarchy, for example.
> >>>> This hierarchy doesn't exist with ns->parent ? Do you have an example
> >>>> when the hierarchy should be used ? I mean when we need to browse from
> >>>> l2 -> l3 ?

> >>> For example, to check that new ifaddr is already used by child l3 namespace.
> >> The devinet isolation does already do that, you can not add a new ifaddr
> >> if it already exists. Do you have another example ?
> > Could devinet isolation provide ifaddrs list with namespaces?
> > What will be with child namespaces if you decide to destroy parent namespace?
> > If we decide to destroy them, than how we could get their list?
> > It is a question of flexibility and easy management.
> > Why do you want to remove this code?
>
> I don't want to especially remove this code, I just want to understand
> what it does and why. If it appears to be useless, let's remove it, if
> it appears to be useful, let's keep it.
>
> By the way, what is the meaning on destroying the namespaces directly,
> is it not the kref mechanism which needs to do that ? For example, if
> you create a l2 namespace and after you create l3 namespaces. You want
> to destroy the l2 namespace, the l2 namespace should stay "zombie" until
> all the l3 namespaces exit. If you need to wipe out all the namespaces,
> you should destroy all the related namespaces' ressources, like killing
> all processes inside it. The namespaces will "put" their respective kref
> and will trigger the freeing of the ressources.

networking (mostly sockets) will probably require
some mechanism to 'zap' them, ignoring the defined
timeouts. otherwise the spaces could hang around
for quite a while waiting for some response, which
might never come ...

but that should not be _that_ important right now

best,
Herbert

> _____
> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Daniel Lezcano on Wed, 13 Dec 2006 09:36:25 GMT
View Forum Message <> Reply to Message

Herbert Poetzl wrote:
> On Tue, Dec 12, 2006 at 04:50:50PM +0100, Daniel Lezcano wrote:
>> Dmitry Mishin wrote:
>>> On Tuesday 12 December 2006 17:19, Daniel Lezcano wrote:
>>>> Dmitry Mishin wrote:
>>>>
>>>>>>>>> Why do yo need to have a child list and sibling list ?
>>>>>>>>> Because of the level2<->level3 hierarchy, for example.
>>>>>>>>> This hierarchy doesn't exist with ns->parent ? Do you have an example
>>>>>>>>> when the hierarchy should be used ? I mean when we need to browse from
>>>>>>>>> l2 -> l3 ?
>>>>>>> For example, to check that new ifaddr is already used by child l3 namespace.
>>>>>> The devinet isolation does already do that, you can not add a new ifaddr
>>>>>> if it already exists. Do you have another example ?
>>>>> Could devinet isolation provide ifaddrs list with namespaces?
>>>>> What will be with child namespaces if you decide to destroy parent namespace?
>>>>> If we decide to destroy them, than how we could get their list?
>>>>> It is a question of flexibility and easy management.
>>>>> Why do you want to remove this code?
>> I don't want to especially remove this code, I just want to understand
>> what it does and why. If it appears to be useless, let's remove it, if
>> it appears to be useful, let's keep it.
>>
>> By the way, what is the meaning on destroying the namespaces directly,
>> is it not the kref mechanism which needs to do that ? For example, if
>> you create a l2 namespace and after you create l3 namespaces. You want
>> to destroy the l2 namespace, the l2 namespace should stay "zombie" until
>> all the l3 namespaces exit. If you need to wipe out all the namespaces,
>> you should destroy all the related namespaces' ressources, like killing
>> all processes inside it. The namespaces will "put" their respective kref
>> and will trigger the freeing of the ressources.
>
> networking (mostly sockets) will probably require
> some mechanism to 'zap' them, ignoring the defined
> timeouts. otherwise the spaces could hang around
> for quite a while waiting for some response, which
> might never come ...

Yes, exact. We will need a specific socket cleanup by namespace in order
 to do network migration. This is the only case I see to 'zap' the sockets.
The sockets should never be flushed in other cases. For example, you
launch an application into a network namespace, it sends 10MB to a peer
and exits. The network namespace should stay "alive" until all orphans
sockets have flushed their buffers to the peer. This behavior is
perfectly handled by the kref mechanism because sock_release will "put"
the network namespace and that will trigger the network namespace
destruction.

> but that should not be _that_ important right now

I think this should be addressed later for the network checkpoint/restart.

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers