Subject: namespace acceptance process. bad news
Posted by den on Wed, 05 Dec 2007 10:42:42 GMT
View Forum Message <> Reply to Message

Hello, All!

We are completely bite to ground with the current Eric's patchset today
by Dave Miller. flowi tagging considered wrong. The same opinion has
been received from Alexey Kuznetsov :(

So, it seems that we can't push this approach.

Daniel, Benjamin, should I merge your code to our git after this news or
we should stop a bit and think? We have talked on OLS that if Dave stop
us with current approach we could try global context as in OpenVz.

I think I'll code this a bit and see a reaction, but we need to have
some agreement here :)

Regards,
Den

Subject: Re: namespace acceptance process. bad news
Posted by Daniel Lezcano on Wed, 05 Dec 2007 10:58:57 GMT
View Forum Message <> Reply to Message

Denis V. Lunev wrote:
> Hello, All!
>
> We are completely bite to ground with the current Eric's patchset today
> by Dave Miller. flowi tagging considered wrong. The same opinion has
> been received from Alexey Kuznetsov :(
>
> So, it seems that we can't push this approach.

Argh !

>
> Daniel, Benjamin, should I merge your code to our git after this news or
> we should stop a bit and think? We have talked on OLS that if Dave stop
> us with current approach we could try global context as in OpenVz.

IMHO, doing netns switching has no sense now we are so far in the netns
implementation.

> I think I'll code this a bit and see a reaction, but we need to have
> some agreement here :)

I am more inclined to think about how to handle this problem before
doing anything.

Let's try to understand why flowi tagging is considered wrong first.

Alexey seems to disagree with this approach, is it possible to elaborate
a little bit ?

Subject: Re: namespace acceptance process. bad news
Posted by den on Wed, 05 Dec 2007 11:22:49 GMT
View Forum Message <> Reply to Message

Daniel Lezcano wrote:
> Denis V. Lunev wrote:
>> Hello, All!
>>
>> We are completely bite to ground with the current Eric's patchset today
>> by Dave Miller. flowi tagging considered wrong. The same opinion has
>> been received from Alexey Kuznetsov :(
>>
>> So, it seems that we can't push this approach.
>
> Argh !
>
>>
>> Daniel, Benjamin, should I merge your code to our git after this news or
>> we should stop a bit and think? We have talked on OLS that if Dave stop
>> us with current approach we could try global context as in OpenVz.
>
> IMHO, doing netns switching has no sense now we are so far in the netns
> implementation.
>
>> I think I'll code this a bit and see a reaction, but we need to have
>> some agreement here :)
>
> I am more inclined to think about how to handle this problem before
> doing anything.
>
> Let's try to understand why flowi tagging is considered wrong first.
>
> Alexey seems to disagree with this approach, is it possible to elaborate
> a little bit ?
>
>
Here is a quote from Miller:

| I'm not applying this, it's going to have a negative impact on routing
| performance.
|
| It also changes the semantics of the flowi object in a way I very
| much dislike, in that there is now non-clobberable state in there.
|
| Previously only addressing identifying objects were present in the
| flow, you could use it any context, and there were no pointer
| dereferencing or object references from this thing. It was very
| simple.
|
| That is no longer the case after your patch and I don't want us
| to go down this path.
|
| Please find another way to implement this.

flowi marking is a way to deliver the namespace into the routing code,
as far as I can understand the implementation.

Regards,
 Den

Subject: Re: namespace acceptance process. bad news
Posted by ebiederm on Wed, 05 Dec 2007 11:52:30 GMT
View Forum Message <> Reply to Message

"Denis V. Lunev" <den@sw.ru> writes:

> Daniel Lezcano wrote:
>> Denis V. Lunev wrote:
>>> Hello, All!
>>>
>>> We are completely bite to ground with the current Eric's patchset today
>>> by Dave Miller. flowi tagging considered wrong. The same opinion has
>>> been received from Alexey Kuznetsov :(
>>>
>>> So, it seems that we can't push this approach.
>>
>> Argh !
>>
>>>
>>> Daniel, Benjamin, should I merge your code to our git after this news or
>>> we should stop a bit and think? We have talked on OLS that if Dave stop
>>> us with current approach we could try global context as in OpenVz.
>>
>> IMHO, doing netns switching has no sense now we are so far in the netns
>> implementation.

>>
>>> I think I'll code this a bit and see a reaction, but we need to have
>>> some agreement here :)
>>
>> I am more inclined to think about how to handle this problem before
>> doing anything.
>>
>> Let's try to understand why flowi tagging is considered wrong first.
>>
>> Alexey seems to disagree with this approach, is it possible to elaborate
>> a little bit ?
>>
>>
> Here is a quote from Miller:
>
> | I'm not applying this, it's going to have a negative impact on routing
> | performance.
> |
> | It also changes the semantics of the flowi object in a way I very
> | much dislike, in that there is now non-clobberable state in there.
> |
> | Previously only addressing identifying objects were present in the
> | flow, you could use it any context, and there were no pointer
> | dereferencing or object references from this thing. It was very
> | simple.
> |
> | That is no longer the case after your patch and I don't want us
> | to go down this path.
> |
> | Please find another way to implement this.
>
> flowi marking is a way to deliver the namespace into the routing code,
> as far as I can understand the implementation.

Ok. Sounds like a reasonable technical objection that we need to look at,
and it is pretty significant. I need to look at this and sleep on it
before I can address this.

Eric

Subject: Re: namespace acceptance process. bad news
Posted by Benjamin Thery on Wed, 05 Dec 2007 12:31:00 GMT
View Forum Message <> Reply to Message

Daniel Lezcano wrote:
> Denis V. Lunev wrote:
>> Hello, All!

>>
>> We are completely bite to ground with the current Eric's patchset today
>> by Dave Miller. flowi tagging considered wrong. The same opinion has
>> been received from Alexey Kuznetsov :(
>>
>> So, it seems that we can't push this approach.
>
> Argh !

Re-Argh...

>> Daniel, Benjamin, should I merge your code to our git after this news or
>> we should stop a bit and think?

Um, on a pratical point of view, I think it could be good to merge the
IPv6 patchset in your git, (if there aren't too much conflicts and if it
doesn't take too long), to store it somewhere and be able to use it as
a reference.

I also tend to think that we should think a bit more about the issue
raised by Dave and try to find an alternative solution (if needed)
before dropping the current model for handling netns.

Benjamin

>> We have talked on OLS that if Dave stop
>> us with current approach we could try global context as in OpenVz.
>
> IMHO, doing netns switching has no sense now we are so far in the netns
> implementation.
>
>> I think I'll code this a bit and see a reaction, but we need to have
>> some agreement here :)
>
> I am more inclined to think about how to handle this problem before
> doing anything.
>
> Let's try to understand why flowi tagging is considered wrong first.
>
> Alexey seems to disagree with this approach, is it possible to elaborate
> a little bit ?
>
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers
>

--
B e n j a m i n T h e r y - BULL/DT/Open Software R&D

http://www.bull.com

_____

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

---

Subject: Re: namespace acceptance process. bad news
Posted by Alexey Kuznetsov on Wed, 05 Dec 2007 12:33:55 GMT
View Forum Message <> Reply to Message

Hello!

> Alexey seems to disagree with this approach, is it possible to elaborate
> a little bit ?

My first reaction was exactly the same as David's one. Exactly. :-)

flowi structure was invented to be both easily initialized/disposed
as a local variable and copied/stored in various caches as a key.

If it has some reference inside, it becomes really ugly.

But it is the first reaction. I guess you do not have much of choice.
The only alternative is to add an additional argument to functions
taking flowi, which is even uglier.

So, it looks like netns still have to go to flowi, but functions copying
flowi (in route.c/flow.c/whatever) should not use raw memcpy to store this
and must remember that saving flowi is possible only when refcnt to netns
is held somewhere.

Alexey

Subject: Re: namespace acceptance process. bad news
Posted by den on Wed, 05 Dec 2007 12:40:45 GMT
View Forum Message <> Reply to Message

Alexey Kuznetsov wrote:
> Hello!
>
>> Alexey seems to disagree with this approach, is it possible to elaborate

>> a little bit ?
>
> My first reaction was exactly the same as David's one. Exactly. :-)
>
> flowi structure was invented to be both easily initialized/disposed
> as a local variable and copied/stored in various caches as a key.
>
> If it has some reference inside, it becomes really ugly.
>
> But it is the first reaction. I guess you do not have much of choice.
> The only alternative is to add an additional argument to functions
> taking flowi, which is even uglier.
>
> So, it looks like netns still have to go to flowi, but functions copying
> flowi (in route.c/flow.c/whatever) should not use raw memcpy to store this
> and must remember that saving flowi is possible only when refcnt to netns
> is held somewhere.

flowi does not take the ref. You will not :)

Regards,
Den

Subject: Re: namespace acceptance process. bad news
Posted by Daniel Lezcano on Wed, 05 Dec 2007 13:20:10 GMT
View Forum Message <> Reply to Message

Alexey Kuznetsov wrote:
> Hello!
>
>> Alexey seems to disagree with this approach, is it possible to elaborate
>> a little bit ?
>
> My first reaction was exactly the same as David's one. Exactly. :-)
>
> flowi structure was invented to be both easily initialized/disposed
> as a local variable and copied/stored in various caches as a key.
>
> If it has some reference inside, it becomes really ugly.
>
> But it is the first reaction. I guess you do not have much of choice.
> The only alternative is to add an additional argument to functions
> taking flowi, which is even uglier.
>
> So, it looks like netns still have to go to flowi, but functions copying
> flowi (in route.c/flow.c/whatever) should not use raw memcpy to store this
> and must remember that saving flowi is possible only when refcnt to netns

> is held somewhere.
>
> Alexey

Thanks Alexey for your analysis.

There is no refcount for netns held because it is used as an identifier.
We can perhaps make it clear by changing the field fl_net by:

struct net *fl_net => unsigned long fl_net_key;

In this case, we must track all places where we reused fl_net as a pointer to retrieve the netns like in route.c, fib_hash.c or fib_rules.c because in this case we must held a reference. So the functions will probably take a new netns parameter or pick the netns pointer from somewhere else.

Subject: Re: namespace acceptance process. bad news
Posted by ebiederm on Wed, 05 Dec 2007 22:21:30 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> Alexey Kuznetsov wrote:
>> Hello!
>>
>>> Alexey seems to disagree with this approach, is it possible to elaborate
>>> a little bit ?
>>
>> My first reaction was exactly the same as David's one. Exactly. :-)
>>
>> flowi structure was invented to be both easily initialized/disposed
>> as a local variable and copied/stored in various caches as a key.
>>
>> If it has some reference inside, it becomes really ugly.
>>
>> But it is the first reaction. I guess you do not have much of choice.
>> The only alternative is to add an additional argument to functions
>> taking flowi, which is even uglier.
>>
>> So, it looks like netns still have to go to flowi, but functions copying
>> flowi (in route.c/flow.c/whatever) should not use raw memcpy to store this
>> and must remember that saving flowi is possible only when refcnt to netns
>> is held somewhere.
>>
>> Alexey
>

> Thanks Alexey for your analysis.
>
> There is no refcount for netns held because it is used as an identifier. We can
> perhaps make it clear by changing the field fl_net by:
>
> struct net *fl_net => unsigned long fl_net_key;

> In this case, we must track all places where we reused fl_net as a pointer to
> retrieve the netns like in route.c, fib_hash.c or fib_rules.c because in this
> case we must held a reference. So the functions will probably take a new netns
> parameter or pick the netns pointer from somewhere else.

I did a quick grep for the places we actually use fl_net, and we barely
examine it so I don't expect there will be to much pain.

Several of the references work against the routing table entry and
we can just put a struct net reference in rtable. (The hold_net and release_net is
just for sanity checking).

```
net/ipv4/icmp.c:                        dev = dev_get_by_index(rt->fl.fl_net, rt->fl.iif);
net/ipv4/route.c:       peer = inet_getpeer(rt->fl.fl_net, rt->rt_dst, create);
net/ipv4/route.c:                               hold_net(rt->fl.fl_net);
net/ipv4/route.c:       release_net(rt->fl.fl_net);
net/ipv4/route.c:       rth->fl.fl_net  = hold_net(oldflp->fl_net);
net/ipv4/route.c:                       rth->fl.fl_net == flp->fl_net &&
```

The rest look like we will have to examine in detail.

```
net/ipv4/fib_rules.c:   if ((tbl = fib_get_table(flp->fl_net, rule->table)) == NULL)
net/ipv4/fib_rules.c:           if ((tb = fib_get_table(flp->fl_net, res->r->table)) != NULL)
net/ipv4/route.c:               if (r->fl.fl_net != st->p.net)
include/net/ip_fib.h:   struct net *net = flp->fl_net;
include/net/ip_fib.h:   struct net *net = flp->fl_net;
net/ipv4/fib_hash.c:    struct net *net = flp->fl_net;
net/ipv4/fib_rules.c:   struct net *net = flp->fl_net;
net/ipv4/fib_trie.c:    struct net *net = flp->fl_net;
net/ipv4/icmp.c:        net = rt->fl.fl_net;
net/ipv4/route.c:               fl1->fl_net == fl2->fl_net;
net/ipv4/route.c:       struct net *net = oldflp->fl_net;
```

But it is a small enough list it shouldn't take an insanely long time to look at.

Eric