Subject: Re: NET namespace locking seems broken to me
Posted by den on Fri, 21 Sep 2007 07:27:14 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> "Denis V. Lunev" <den@sw.ru> writes:
>
>> Hello, Eric!
>>
>> Current locking in mainstream seems broken to me.
>
> Thanks. After looking at this I concur.
>
>> 1. struct net->list is manipulated under double net_mutex/net_list_mutex
>
> Yes. Making iteration safe if we hold only one of those.
>
>> 2. net_list_mutex has been taken only in cleanup_net/net_ns_init inside
>> net_mutes and seems pointless now
>
> And in rtnl_unlock (although that isn't upstream just yet).
> It looks like I forgot to call net_lock in some of my later
> insertions of for_each_net.
>
> Certainly it looks like too many locks.
>
> Thinking.
>
> net_mutex appears to be there to serial the addition/removal of
> subsystems/modules and the creation/destruction of network namespaces.
>
> net_list_mutex is just there to serialize operations on the list of
> namespaces.
>
> I'm trying to see if there is something that implies a nesting of:
> net_mutex, rtnl, net_list_mutex.
>
> Although it is no longer an issue now that I am making fewer locks
> per network namespace.
>
> I am remembering that there was something keeping from using the rtnl.
>
>> 3. for_each_net (iterating against net_namespace_list) is called from
>>   a) register_netdevice_notifier/__rtnl_link_unregister
>
> Yes this is fishy, and probably needs to be fixed.
>
>>   b) register_pernet_operations/unregister_pernet_operations

>>   In the case b) the situation is sane, i.e. net_mutex is held while in
>> the case b) we held rtnl_only
>>
>> So, this does not look good to me for now.
>> How to cure this situation? I think that we can drop all locks for now
>> and perform all operations under rtnl only. In the other case we must
>> decide now should we make rtnl inner or outer for net_mutex.
>
> Ok. I have found an important case. loopback.

May be it will be better to move this in netdev_run_todo to cleanup
locking. I am not sure right now.

Basically, there are 4 (four) locks after the patch:
- dev_base_lock
- rtnl
- net_list_mutex
- net_mutex

Too many for me :)

> We must hold net_mutex when we are calling all of the .init routines.
> The loopback code calls register_netdev which grabs rtnl.
>
> - So we have net_mutex must be outside of rtnl.
>
> We have to do for_each_net in rtnl_unlock so we can find all of the
> rtnl netlink sockets and sk_data_ready aka rtnetlink_rcv which takes
> the rtnl_lock.
>
> - So net_list_lock should be taken outside of rtnl_lock.
>
> We take net_list_mutex in rtnl_unlock() but not under rtnl_mutex. And
> rtnl_unlock is called inside of net_mutex, so we can't use net_mutex.
>
> - So we need both net_list_lock and net_mutex.
>
> Therefore it looks like we just need to take net_lock() outside of
> rtnl_lock() in register_netdevice_notifier.
>
>> >From my point of view net_mutex should be taken inside rtnl lock and we
>> must add it now into list manipulation routines.
>
> I think that is where I started and I failed miserably. The per
> network namespace instances of the rtnl socket look to make that
> impossible.

Why do we need them? The only case is that we want absence of some

protocols/layers inside different namespaces. We have the only rtnl
socket in OpenVZ

>> Plz point me to my mistake in logic :)
>
> Does what I said sound reasonable now.
>
> Thanks for spotting the missing lock by the way.
>
> You want to cook up the patch to fix register_netdevice_notifier?

I am trying this now.

Regards,
Den