Subject: Re: net namespace plans for 2.6.25 (was Re: Pid namespaces problems)
Posted by Daniel Lezcano on Thu, 08 Nov 2007 14:09:36 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov wrote:
> Daniel Lezcano wrote:
>> Denis V. Lunev wrote:
>> > Daniel Lezcano wrote:
>> >> Denis V. Lunev wrote:
>> >>> Daniel Lezcano wrote:
>> >>>
>> >>>> * the first one is the locking of the network namespace list by
>> >>>> rtnl_lock, so from the timer callback we can not browse the network
>> >>>> namespace list to check the age of the routes. It is a problem I would
>> >>>> like to talk with Denis if he has time
>> >>> From my point of view, the situation is clear. The timer should be
>> >>> per/namespace. The situation is completely different as one in IPv4.
>> >> We thought to make a timer per namespace for ipv6, but we are a little
>> >> afraid for the performances when there will be a lot of containers.
>> >> Anyway, we can do a timer per namespace and optimize that later. I will
>> >> cook a new patch to take into account that for the next week.
>> >
>> > IMHO not a problem. tcp_write_timer is per/socket timer. If this works
>> > efficiently, per/namespace one will work also.
>>
>> That's right, this is a good argument. By the way, the amount of work to
>> be done in the tcp_write_timer is perhaps smaller than the one done in
>> the ipv6 routing age check, no ? Anyway, I'm not against a timer per
>> namespace in this case, I already did a try before rolling back to a
>> for_each_net in the gc timer, that changes a little the API, but nothing
>
> We can easily make the netns list rcu protected to address this issue.
> If you're interested, I can prepare a patch tomorrow.

Sure, I'm interested :)

Benjamin and I, we thought about using a rcu to avoid to use a timer per
namespace in ipv6 but we faced to the problem with rtnl_unlock function
when the network namespace is protected with the rtnl_lock/rtnl_unlock.
In the function rtnl_unlock (not the one in net-2.6 but the one which is
in netns49), there is loop, for_each_net, in this loop, we do
rtnl_unlock, call sk_data_ready and take the lock again. If we are in
rcu protected model, this loop will take a lock (one time just before
sk_data_ready and one time in the sk_data_ready function). As far as I
understand with rcu, we should not block inside a rcu_read_lock, right ?

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