Subject: Re: [PATCH 0/5] Make nicer CONFIG_NET_NS=n case code
Posted by ebiederm on Thu, 01 Nov 2007 00:58:28 GMT
View Forum Message <> Reply to Message

David Miller <davem@davemloft.net> writes:

> From: Eric Dumazet <dada1@cosmosbay.com>
> Date: Wed, 31 Oct 2007 23:40:59 +0100
>

>> > Eric Dumazet <dada1@cosmosbay.com> writes:
>> >
>> >
>> >> Definitly wanted here. Thank you.
>> >> One more refcounting on each socket creation/deletion was expensive.
>> >
>> > Really? Have you actually measured that? If the overhead is
>> > measurable and expensive we may want to look at per cpu counters or
>> > something like that. So far I don't have any numbers that say any
>> > of the network namespace work inherently has any overhead.
>>
>> It seems that on some old opterons (two 246 for example),
>> "if (atomic_dec_and_test(&net->count))" is rather expensive yes :(
>
> P4 chips are generally very poor at mispredicted branches and
> atomics. So every atomic you remove from the socket paths
> gives a noticable improvement on them.

Interesting.

> Network device reference counting is such a stupid problem. There has
> to be a way to get rid of it on the packet side.
>
> I think we could get rid of all of the device refcounting from packets
> if we:
>
> 1) Formalize "SKB roots". This is every place a packet
>   could sit in the transmit path.

Yes there are very few of these, and I think they are generally
in interrupt or at least bottom half context aren't they?

I think the OpenVz version of network namespaces may have
already identified all of these.

> 2) On device unregister:
>
> a) wait for RCU quiesce period

> b) stop_machine_run(skb_walk_roots, netdev, NR_CPUS);

RCU sounds sufficient but possibly overkill to achieve what
we need to do here.

>   skb_walk_roots is a function that walks all the places in
>   #1, rewriting the packet to point to loopback or whatever
>   instead of 'netdev' which we are trying to unregister.
>
> This gives us two things.
>
> First, we no longer would need to rectount net devices
> for packet references.
>
> Second, we have a debugging framework for all those dreaded SKB leaks
> that keep devices from being unloadable. As we walk the roots
> we'll see where all packets referencing a device actually are.

Sounds quite useful. Grrr. The brain cache locality that gets us
to rewrite things while we are refactoring them to have more
functionality.... It just keeps the problem from being straigh
forward ;)

Eric