Subject: Re: [PATCH 4/4 (resent) net-2.6.25][UNIX] Make the unix sysctl tables per-namespace
Posted by ebiederm on Sat, 01 Dec 2007 19:32:02 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov <xemul@openvz.org> writes:

>>> But I gotta say this struct/file is going to be enormous. It's also
>>> one of those files that causes everything to get recompiled. Maybe
>>> we ought to make a rule that each subsystem only gets to have at most
>>> one entry in it :)
>>>
>>> Thanks,
>>
>> Good point, thanks. We'll start thinking in that direction. Right now it
>> is not finally cursed with all staff around.
>
> Agree, the point is good :) but it has one pitfall :(
>
> Look, now we make _one_ dereference to get any net->xxx variable
> (sysctl, list head, lock, etc). When we force each subsystem
> has it's "private" pointer on this, we'll make them take _two_
> dereferences. Before the whole net namespace stuff started we
> made _zero_ dereferences :) This may tell upon the performance.
>
> I'm not claiming that this is the major case against this idea,
> but when developing this idea, I think we should keep that fact
> in ming and pay good attention to performance regressions.

Currently in my proof of concept tree I am at 65 variables and 648 bytes.
This includes patches that are largely complete for ipv4. In number
of variables this is about half of the current struct net_device,
so I think the usage looks managable.

I agree that both performance and size are significant concerns,
and that is essentially why struct net has the structure it does
today.

I print the size of struct net out at boot, we have to actually look
at struct net when we make changes, so I don't think size bloat
is going to happen unnoticed.

By keeping the size below PAGE_SIZE, and keeping the number of
variables per network subsystem few and small we should be ok.

The fact that changing struct net causes the core of
the networking stack to recompile is an added bonus
that should also discourage people from playing with it to

much.

My recommendation is to keep an eye on struct net and if what we
are doing there becomes a problem address it then.

Eric

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