Subject: Re: L2 network namespace benchmarking
Posted by ebiederm on Wed, 28 Mar 2007 11:52:31 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <daniel.lezcano@free.fr> writes:

> Eric W. Biederman wrote:
>> Daniel Lezcano <dlezcano@fr.ibm.com> writes:
>>
>>> 3. General observations
>>> ----------------------
>>>
>>> The objective to have no performances degrations, when the network
>>> namespace is off in the kernel, is reached in both solutions.
>>>
>>> When the network is used outside the container and the network
>>> namespace are compiled in, there is no performance degradations.
>>>
>>> Eric's patchset allows to move network devices between namespaces and
>>> this is clearly a good feature, missing in the Dmitry's patchset. This
>>> feature helps us to see that the network namespace code does not add
>>> overhead when using directly the physical network device into the
>>> container.
>>
>> Assuming these results are not contradicted this says that the extra
>> dereference where we need it does not add measurable to the overhead
>> in the Linus network stack. Performance wise this should be good
>> enough to allow merging the code into the linux kernel, as it does
>> not measurably affect networking when we do not have multiple
>> containers in use.
>
> I have a few questions about merging code into the linux kernel.
>
> * How do you plan to do that ?
One small comprehensible piece at a time.

Basically some variant of etun should not be a problem to merge
then I have to get some part of the network namespace code merged,
and the concept accepted.

Once the basic acceptance occurs it just becomes a long slog of
merging more and more patches.

> * When do you expect to have the network namespace into mainline ?
My current goal is to finish my rebase against 2.6.linus_lastest in
the next couple of days after having figured out how to deal with sysfs.

I have been doing reviewing in more code then I know what to do with,
and fighting some very strange bugs during the stabilization window.
Which has kept me from doing additional development. Plus I have
had a cold.

> * Are Dave Miller and Alexey Kuznetov aware of the network namespace ?
Aware yes, reviewed not yet. I believe Alexey is a little more
familiar with the OpenVZ work. The high level concepts still apply.

> * Did they saw your patchset or ever know it exists ?
Yes.

> * Do you have any feedbacks from netdev about the network namespace ?
Not really. Except that Dave Miller wanted to review what I posted
last time but the timing was bad and he failed to get around to it.

>> To be fully satisfactory how we get the packets to the namespace
>> still appears to need work.
>>
>> We have overhead in routing. That may simply be the cost of
>> performing routing or there may be some optimizations opportunities
>> there.
>> We have about the same overhead when performing bridging which I
>> actually find more surprising, as the bridging code should involve
>> less packet handling.
>
> Yep. I will try to figure out what is happening.

Thanks.

>> Ideally we can optimize the bridge code or something equivalent to
>> it so that we can take one look at the destination mac address and
>> know which network namespace we should be in. Potentially moving this
>> work to hardware when the hardware supports multiple queues.
>>
>> If we can get the overhead out of the routing code that would be
>> tremendous. However I think it may be more realistic to get the
>> overhead out of the ethernet bridging code where we know we don't need
>> to modify the packet.
>
> The routing was optimized for the loopback, no ? Why can't we do the same for
> the etun device ?

I have no problem with it if we can use valid optimizations. Avoiding a
packet copy when the packet is marked as having a second copy somewhere
else does not sound like a valid optimization to me.

Routing through both network namespaces so that we can set up a dst

cache entry that takes you to the final destination I am will to
working with. Perhaps something that hits this piece of the etun driver,
so we don't have to make a second set of routing decisions.

```
 if (skb->dst)
  skb->dst = dst_pop(skb->dst); /* Allow for smart routing */
```

tcpdump at any phase of the process should be able to do the right thing.

Mostly I care right now in that it is interesting to know where the
performance overhead is coming from. Unless it is something of a
merge stopper I don't much care about how we are going to fix it yet,
especially if it is only cross network namespace traffic.

If I read the results right it took a 32bit machine from AMD with
a gigabit interface before you could measure a throughput difference.
That isn't shabby for a non-optimized code path.

Eric

_____