Subject: L2 network namespaces + macvlan performances
Posted by Benjamin Thery on Fri, 06 Jul 2007 16:48:15 GMT
View Forum Message <> Reply to Message

Following a discussion we had at OLS concerning L2 network namespace
performances and how the new macvlan driver could potentially improve
them, I've ported the macvlan patchset on top of Eric's net namespace
patchset on 2.6.22-rc4-mm2.

A little bit of history:

Some months ago, when we ran some performance tests (using netperf)
on net namespace, we observed the following things:

Using 'etun', the virtual ethernet tunnel driver, and IP routes
from inside a network namespace,

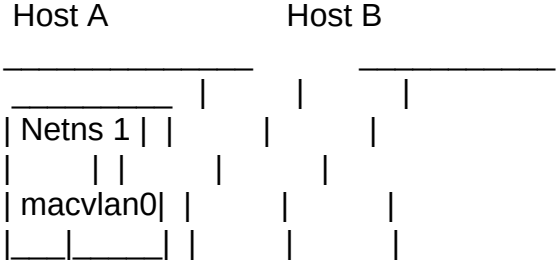
- The throughput is the same as the "normal" case(*)
  (* normal case: no namespace, using physical adapters).
  No regression. Good.

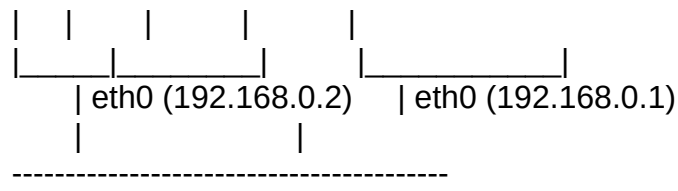
- But the CPU load increases a lot. Bad.
  The reasons are:
 - All checksums are done in software. No hardware offloading.
 - Every TCP packets going through the etun devices are
  duplicated in ip_forward() before we decrease the ttl.
        (packets are routed between both ends of etun)

We also made some testing with bridges, and obtained the same results:
 CPU load increase:
 - No hardware offloading
 - Packets are duplicated somewhere in the bridge+netfilter
        code (can't remember where right now)

This time, I've replaced the etun interface by the new macvlan,
which should benefits from the hardware offloading capabilities of the
physical adapter and suppress the forwarding stuff.

My test setup is:





- netperf runs on host A
- netserver runs on host B
- Adapters speed is 1GB/s

On this setup I ran the following netperf tests: TCP_STREAM,
TCP_MAERTS, TCP_RR, UDP_STREAM, UDP_RR.

Between the "normal" case and the "net namespace + macvlan" case,
results are about the same for both the throughput and the local CPU
load for the following test types: TCP_MAERTS, TCP_RR, UDP_STREAM, UDP_RR.

macvlan looks like a very good candidate for network namespace in
these cases.

But, with the TCP_STREAM test, I observed the CPU load is about the
same (that's what we wanted) but the throughput decreases by about 5%:
from 850MB/s down to 810MB/s.
I haven't investigated yet why the throughput decrease in the case.
Does it come from my setup, from macvlan additional treatments, other?
I don't know yet

Attached to this email you'll find the raw netperf outputs for the
three cases:

- netperf through a physical adapter, no namespace:
 netperf-results-2.6.22-rc4-mm2-netns1-vanilla.txt
- netperf through etun, inside a namespace:
 netperf-results-2.6.22-rc4-mm2-netns1-using-etun.txt
- netperf through macvlan, inside a namespace:
 netperf-results-2.6.22-rc4-mm2-netns1-using-macvlan.txt

macvlan looks promising.

Regards,
Benjamin

--
B e n j a m i n T h e r y - BULL/DT/Open Software R&D

http://www.bull.com

NETPERF RESULTS: the "normal" case :
==========
No network namespace, traffic goes through real 1GB/s physical adapters.

TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Recv Send Send Utilization Service Demand
Socket Socket Message Elapsed Send Recv Send Recv
Size Size Size Time Throughput local remote local remote
bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB

87380 16384 1400 20.03 857.39 6.39 9.75 2.444 3.727

TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Recv Send Send Utilization Service Demand
Socket Socket Message Elapsed Send Recv Send Recv
Size Size Size Time Throughput local remote local remote
bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB

87380 16384 87380 20.03 763.15 4.75 10.33 2.038 4.434

TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Local /Remote
Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
Send Recv Size Size Time Rate local remote local remote
bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr

16384 87380 1 1 20.00 12594.24 4.16 6.06 13.212 19.231
16384 87380

UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Socket Message Elapsed Messages CPU Service
Size Size Time Okay Errors Throughput Util Demand
bytes bytes secs # # 10^6bits/sec % SS us/KB

110592 1400 20.00 1701653 0 952.9 6.84 2.354
107520 20.00 1701647 952.9 9.66 3.321

UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Local /Remote
Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
Send Recv Size Size Time Rate local remote local remote
bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr

110592 110592 1 1 20.00 13789.92 3.82 6.16 11.087 17.855
107520 107520

NETPERF RESULTS: the etun case :
==========
netperf is ran from a network namespace,
traffic goes through etun adapters.

TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Recv Send Send Utilization Service Demand
Socket Socket Message Elapsed Send Recv Send Recv
Size Size Size Time Throughput local remote local remote
bytes bytes bytes secs. 10^6bits/s % S % U us/KB us/KB

87380 16384 1400 40.02 840.64 12.89 -1.00 5.025 -1.000

TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Recv Send Send Utilization Service Demand
Socket Socket Message Elapsed Send Recv Send Recv
Size Size Size Time Throughput local remote local remote
bytes bytes bytes secs. 10^6bits/s % S % U us/KB us/KB

87380 16384 87380 40.03 763.30 6.29 -1.00 2.701 -1.000

TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Local /Remote
Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem

```
Send   Recv   Size    Size   Time    Rate     local  remote local   remote
bytes  bytes  bytes   bytes  secs.   per sec  % S    % U    us/Tr   us/Tr

16384  87380  1       1      40.00   12230.34  4.64   -1.00  15.167  -1.000
16384  87380
-------------------------------------------------


-------------------------------------------------
UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Socket  Message  Elapsed      Messages                CPU      Service
Size    Size     Time         Okay Errors   Throughput  Util     Demand
bytes   bytes    secs            #      #   10^6bits/sec % SU     us/KB

110592    1400   40.00     12981742      0     3634.7     25.64    8.801
107520           40.00      3409123             954.5     -1.00    -1.000

-------------------------------------------------


-------------------------------------------------
UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Local /Remote
Socket Size   Request Resp.  Elapsed Trans.   CPU    CPU    S.dem   S.dem
Send   Recv   Size    Size   Time    Rate     local  remote local   remote
bytes  bytes  bytes   bytes  secs.   per sec  % S    % U    us/Tr   us/Tr

110592 110592 1       1      40.00   13385.96   4.22   -1.00  12.658  -1.000
107520 107520
-------------------------------------------------


NETPERF RESULTS: the "normal" case :
===================================
netperf is ran from a network namespace,
traffic goes through a macvlan adapter.

-------------------------------------------------
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0
AF_INET : +/-2.5% @ 95% conf.
Recv   Send    Send                          Utilization       Service Demand
Socket Socket  Message  Elapsed              Send     Recv     Send    Recv
Size   Size    Size     Time     Throughput  local    remote   local   remote
bytes  bytes   bytes    secs.    10^6bits/s  % S      % S      us/KB   us/KB

 87380  16384   1400    20.03       817.40   7.26     12.96    2.912   5.200
-------------------------------------------------
```

```
------------------------------------------------------
TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0
AF_INET : +/-2.5% @ 95% conf.
Recv   Send    Send                          Utilization       Service Demand
Socket Socket  Message  Elapsed              Send     Recv     Send    Recv
Size   Size    Size     Time     Throughput  local    remote   local   remote
bytes  bytes   bytes    secs.    10^6bits/s  % S      % S      us/KB   us/KB

 87380  16384  87380    20.03      763.33   4.95     10.32    2.127   4.429
------------------------------------------------------


------------------------------------------------------
TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Local /Remote
Socket Size   Request Resp.  Elapsed Trans.   CPU    CPU    S.dem   S.dem
Send   Recv   Size    Size   Time    Rate     local  remote local   remote
bytes  bytes  bytes   bytes  secs.   per sec  % S    % S    us/Tr   us/Tr

16384  87380  1       1      20.00   12448.36  4.34   6.21   13.950  19.939
16384  87380
------------------------------------------------------


------------------------------------------------------
UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Socket  Message  Elapsed      Messages                   CPU      Service
Size    Size     Time         Okay Errors   Throughput   Util     Demand
bytes   bytes    secs            #      #   10^6bits/sec % SS     us/KB

110592    1400   20.00     1704200      0      954.3     7.11     2.440
107520           20.00     1704194             954.3     9.66     3.318

------------------------------------------------------


------------------------------------------------------
UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
Local /Remote
Socket Size   Request Resp.  Elapsed Trans.   CPU    CPU    S.dem   S.dem
Send   Recv   Size    Size   Time    Rate     local  remote local   remote
bytes  bytes  bytes   bytes  secs.   per sec  % S    % S    us/Tr   us/Tr

110592 110592 1       1      20.00   13751.49   3.98   6.09   11.625  17.788
107520 107520
------------------------------------------------------
```

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

Subject: Re: L2 network namespaces + macvlan performances
Posted by Daniel Lezcano on Sat, 07 Jul 2007 11:39:32 GMT
View Forum Message <> Reply to Message

Benjamin Thery wrote:
> Following a discussion we had at OLS concerning L2 network namespace
> performances and how the new macvlan driver could potentially improve
> them, I've ported the macvlan patchset on top of Eric's net namespace
> patchset on 2.6.22-rc4-mm2.
>
> A little bit of history:
>
> Some months ago, when we ran some performance tests (using netperf)
> on net namespace, we observed the following things:
>
> Using 'etun', the virtual ethernet tunnel driver, and IP routes
> from inside a network namespace,
>
> - The throughput is the same as the "normal" case(*)
> (* normal case: no namespace, using physical adapters).
> No regression. Good.
>
> - But the CPU load increases a lot. Bad.
> The reasons are:
> - All checksums are done in software. No hardware offloading.
> - Every TCP packets going through the etun devices are
> duplicated in ip_forward() before we decrease the ttl.
> (packets are routed between both ends of etun)
>
> We also made some testing with bridges, and obtained the same results:
> CPU load increase:
> - No hardware offloading
> - Packets are duplicated somewhere in the bridge+netfilter
> code (can't remember where right now)
>
>
> This time, I've replaced the etun interface by the new macvlan,
> which should benefits from the hardware offloading capabilities of the
> physical adapter and suppress the forwarding stuff.
>
> My test setup is:
>

> Host A   Host B
> _________________   _____________
> | __________ | | |
> | | Netns 1 | | | |
> | | | | | |
> | | macvlan0| | | |
> | |___|_____| | | |
> | | | | |
> |_____|_______| |___________|
> | eth0 (192.168.0.2) | eth0 (192.168.0.1)
> | |
> --
> macvlan0 (192.168.0.3)
>
> - netperf runs on host A
> - netserver runs on host B
> - Adapters speed is 1GB/s
>
> On this setup I ran the following netperf tests: TCP_STREAM, TCP_MAERTS,
> TCP_RR, UDP_STREAM, UDP_RR.
>
> Between the "normal" case and the "net namespace + macvlan" case,
> results are about the same for both the throughput and the local CPU
> load for the following test types: TCP_MAERTS, TCP_RR, UDP_STREAM, UDP_RR.
>
> macvlan looks like a very good candidate for network namespace in these
> cases.
>
> But, with the TCP_STREAM test, I observed the CPU load is about the
> same (that's what we wanted) but the throughput decreases by about 5%:
> from 850MB/s down to 810MB/s.
> I haven't investigated yet why the throughput decrease in the case.
> Does it come from my setup, from macvlan additional treatments, other? I
> don't know yet
>
> Attached to this email you'll find the raw netperf outputs for the three
> cases:
>
> - netperf through a physical adapter, no namespace:
>   netperf-results-2.6.22-rc4-mm2-netns1-vanilla.txt
> - netperf through etun, inside a namespace:
>   netperf-results-2.6.22-rc4-mm2-netns1-using-etun.txt
> - netperf through macvlan, inside a namespace:
>   netperf-results-2.6.22-rc4-mm2-netns1-using-macvlan.txt
>
>
> macvlan looks promising.
>

> Regards,
> Benjamin

Very interesting.
Thank you very much Benjamin for investigating this.
I will update the http://lxc.sf.net web site with your description and
results.

> ---
>
> NETPERF RESULTS: the "normal" case :
> ==========
> No network namespace, traffic goes through real 1GB/s physical adapters.
>
> ---
> TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port
0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB
>
> 87380 16384 1400 20.03 857.39 6.39 9.75 2.444 3.727
> ---
>
> ---
> TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port
0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB
>
> 87380 16384 87380 20.03 763.15 4.75 10.33 2.038 4.434
> ---
>
> ---
> TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr
>
> 16384 87380 1 1 20.00 12594.24 4.16 6.06 13.212 19.231
> 16384 87380
> ---

>
> ---
> UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Socket Message Elapsed   Messages   CPU   Service
> Size   Size   Time   Okay Errors   Throughput   Util   Demand
> bytes   bytes   secs   #   #   10^6bits/sec % SS   us/KB
>
> 110592   1400   20.00   1701653   0   952.9   6.84   2.354
> 107520   20.00   1701647   952.9   9.66   3.321
>
> ---
>
> ---
> UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size   Request Resp. Elapsed Trans.   CPU   CPU   S.dem   S.dem
> Send   Recv   Size   Size   Time   Rate   local remote local   remote
> bytes bytes bytes   bytes secs.   per sec % S   % S   us/Tr   us/Tr
>
> 110592 110592 1   1   20.00   13789.92   3.82   6.16   11.087 17.855
> 107520 107520
> ---
>
>
>
> --
>
> NETPERF RESULTS: the etun case :
> ==================================
> netperf is ran from a network namespace,
> traffic goes through etun adapters.
>
> ---
> TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Recv   Send   Send   Utilization   Service Demand
> Socket Socket Message Elapsed   Send   Recv   Send   Recv
> Size   Size   Size   Time   Throughput local   remote   local   remote
> bytes bytes   bytes   secs.   10^6bits/s % S   % U   us/KB   us/KB
>
> 87380 16384   1400   40.02   840.64   12.89   -1.00   5.025   -1.000
> ---
>
> ---
> TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.

> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % U us/KB us/KB
>
> 87380 16384 87380 40.03 763.30 6.29 -1.00 2.701 -1.000
> ---
>
> ---
> TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % U us/Tr us/Tr
>
> 16384 87380 1 1 40.00 12230.34 4.64 -1.00 15.167 -1.000
> 16384 87380
> ---
>
> ---
> UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Socket Message Elapsed Messages CPU Service
> Size Size Time Okay Errors Throughput Util Demand
> bytes bytes secs # # 10^6bits/sec % SU us/KB
>
> 110592 1400 40.00 12981742 0 3634.7 25.64 8.801
> 107520 40.00 3409123 954.5 -1.00 -1.000
>
> ---
>
> ---
> UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % U us/Tr us/Tr
>
> 110592 110592 1 1 40.00 13385.96 4.22 -1.00 12.658 -1.000
> 107520 107520
> ---
>
>
>
> ---
>

> NETPERF RESULTS: the "normal" case :
> ==========
> netperf is ran from a network namespace,
> traffic goes through a macvlan adapter.
>
> ----------
> TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB
>
> 87380 16384 1400 20.03 817.40 7.26 12.96 2.912 5.200
> ----------
>
> ----------
> TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB
>
> 87380 16384 87380 20.03 763.33 4.95 10.32 2.127 4.429
> ----------
>
> ----------
> TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr
>
> 16384 87380 1 1 20.00 12448.36 4.34 6.21 13.950 19.939
> 16384 87380
> ----------
>
> ----------
> UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Socket Message Elapsed Messages CPU Service
> Size Size Time Okay Errors Throughput Util Demand
> bytes bytes secs # # 10^6bits/sec % SS us/KB
>
> 110592 1400 20.00 1704200 0 954.3 7.11 2.440
> 107520 20.00 1704194 954.3 9.66 3.318

>
> ---
>
> ---
> UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr
>
> 110592 110592 1 1 20.00 13751.49 3.98 6.09 11.625 17.788
> 107520 107520
> ---
>

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

---

Subject: Re: L2 network namespaces + macvlan performances
Posted by Herbert Poetzl on Mon, 09 Jul 2007 11:55:04 GMT
View Forum Message <> Reply to Message

On Fri, Jul 06, 2007 at 06:48:15PM +0200, Benjamin Thery wrote:
> Following a discussion we had at OLS concerning L2 network namespace
> performances and how the new macvlan driver could potentially improve
> them, I've ported the macvlan patchset on top of Eric's net namespace
> patchset on 2.6.22-rc4-mm2.
>
> A little bit of history:
>
> Some months ago, when we ran some performance tests (using netperf)
> on net namespace, we observed the following things:
>
> Using 'etun', the virtual ethernet tunnel driver, and IP routes
> from inside a network namespace,
>
> - The throughput is the same as the "normal" case(*)
> (* normal case: no namespace, using physical adapters).
> No regression. Good.
>
> - But the CPU load increases a lot. Bad.
> The reasons are:
> - All checksums are done in software. No hardware offloading.
> - Every TCP packets going through the etun devices are

>   duplicated in ip_forward() before we decrease the ttl.
>   (packets are routed between both ends of etun)
>
> We also made some testing with bridges, and obtained the same results:
> CPU load increase:
> - No hardware offloading
> - Packets are duplicated somewhere in the bridge+netfilter
>   code (can't remember where right now)
>
>
> This time, I've replaced the etun interface by the new macvlan,
> which should benefits from the hardware offloading capabilities of the
> physical adapter and suppress the forwarding stuff.
>
> My test setup is:
>
>   Host A   Host B
> _________________   _____________
> | ___________ | | |
> | | Netns 1 | | | |
> | | | | | |
> | | macvlan0| | | |
> | |___|_____| | | |
> | | | | |
> |_____|________| |___________|
> | eth0 (192.168.0.2) | eth0 (192.168.0.1)
> | |
> --
> macvlan0 (192.168.0.3)
>
> - netperf runs on host A
> - netserver runs on host B
> - Adapters speed is 1GB/s
>
> On this setup I ran the following netperf tests: TCP_STREAM,
> TCP_MAERTS, TCP_RR, UDP_STREAM, UDP_RR.
>
> Between the "normal" case and the "net namespace + macvlan" case,
> results are about the same for both the throughput and the local CPU
> load for the following test types: TCP_MAERTS, TCP_RR, UDP_STREAM, UDP_RR.
>
> macvlan looks like a very good candidate for network namespace in
> these cases.
>
> But, with the TCP_STREAM test, I observed the CPU load is about the
> same (that's what we wanted) but the throughput decreases by about 5%:
> from 850MB/s down to 810MB/s.
> I haven't investigated yet why the throughput decrease in the case.

> Does it come from my setup, from macvlan additional treatments, other?
> I don't know yet
>
> Attached to this email you'll find the raw netperf outputs for the
> three cases:
>
> - netperf through a physical adapter, no namespace:
> netperf-results-2.6.22-rc4-mm2-netns1-vanilla.txt
> - netperf through etun, inside a namespace:
> netperf-results-2.6.22-rc4-mm2-netns1-using-etun.txt
> - netperf through macvlan, inside a namespace:
> netperf-results-2.6.22-rc4-mm2-netns1-using-macvlan.txt
>
> macvlan looks promising.

nice, any performance tests for multiple network
spaces sharing the same eth0 (with different macvlans)?
how does that compare to IP isolation performance wise?

TIA,
Herbert

> Regards,
> Benjamin
>
> --
> B e n j a m i n T h e r y - BULL/DT/Open Software R&D
>
> http://www.bull.com

> NETPERF RESULTS: the "normal" case :
> ====================================
> No network namespace, traffic goes through real 1GB/s physical adapters.
>
> --
> TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port
0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB
>
> 87380 16384 1400 20.03 857.39 6.39 9.75 2.444 3.727
> --
>
> --
> TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port
0 AF_INET : +/-2.5% @ 95% conf.

> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB
>
> 87380 16384 87380 20.03 763.15 4.75 10.33 2.038 4.434
> ---
>
> ---
> TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr
>
> 16384 87380 1 1 20.00 12594.24 4.16 6.06 13.212 19.231
> 16384 87380
> ---
>
> ---
> UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Socket Message Elapsed Messages CPU Service
> Size Size Time Okay Errors Throughput Util Demand
> bytes bytes secs # # 10^6bits/sec % SS us/KB
>
> 110592 1400 20.00 1701653 0 952.9 6.84 2.354
> 107520 20.00 1701647 952.9 9.66 3.321
>
> ---
>
> ---
> UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr
>
> 110592 110592 1 1 20.00 13789.92 3.82 6.16 11.087 17.855
> 107520 107520
> ---
>

> NETPERF RESULTS: the etun case :
> ==========
> netperf is ran from a network namespace,

> traffic goes through etun adapters.
>
> --
> TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % U us/KB us/KB
>
> 87380 16384 1400 40.02 840.64 12.89 -1.00 5.025 -1.000
> --
>
> --
> TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % U us/KB us/KB
>
> 87380 16384 87380 40.03 763.30 6.29 -1.00 2.701 -1.000
> --
>
> --
> TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % U us/Tr us/Tr
>
> 16384 87380 1 1 40.00 12230.34 4.64 -1.00 15.167 -1.000
> 16384 87380
> --
>
> --
> UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Socket Message Elapsed Messages CPU Service
> Size Size Time Okay Errors Throughput Util Demand
> bytes bytes secs # # 10^6bits/sec % SU us/KB
>
> 110592 1400 40.00 12981742 0 3634.7 25.64 8.801
> 107520 40.00 3409123 954.5 -1.00 -1.000
>
> --
>

> ---
> UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % U us/Tr us/Tr
>
> 110592 110592 1 1 40.00 13385.96 4.22 -1.00 12.658 -1.000
> 107520 107520
> ---
>

> NETPERF RESULTS: the "normal" case :
> ==================================
> netperf is ran from a network namespace,
> traffic goes through a macvlan adapter.
>
> ---
> TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB
>
> 87380 16384 1400 20.03 817.40 7.26 12.96 2.912 5.200
> ---
>
> ---
> TCP MAERTS TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Recv Send Send Utilization Service Demand
> Socket Socket Message Elapsed Send Recv Send Recv
> Size Size Size Time Throughput local remote local remote
> bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB
>
> 87380 16384 87380 20.03 763.33 4.95 10.32 2.127 4.429
> ---
>
> ---
> TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1 (192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
> Send Recv Size Size Time Rate local remote local remote
> bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr
>

> 16384 87380 1   1   20.00   12448.36 4.34   6.21   13.950 19.939
> 16384 87380
> ---
>
> ---
> UDP UNIDIRECTIONAL SEND TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Socket Message Elapsed   Messages   CPU   Service
> Size   Size   Time   Okay Errors   Throughput   Util   Demand
> bytes   bytes   secs   #   #   10^6bits/sec % SS   us/KB
>
> 110592   1400   20.00   1704200   0   954.3   7.11   2.440
> 107520   20.00   1704194   954.3   9.66   3.318
>
> ---
>
> ---
> UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.76.1
(192.168.76.1) port 0 AF_INET : +/-2.5% @ 95% conf.
> Local /Remote
> Socket Size   Request Resp. Elapsed Trans.   CPU   CPU   S.dem   S.dem
> Send   Recv   Size   Size   Time   Rate   local remote local   remote
> bytes bytes bytes   bytes secs.   per sec % S   % S   us/Tr   us/Tr
>
> 110592 110592 1   1   20.00   13751.49   3.98   6.09   11.625 17.788
> 107520 107520
> ---
>

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

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

---

Subject: Re: L2 network namespaces + macvlan performances
Posted by Rick Jones on Mon, 09 Jul 2007 16:59:06 GMT
View Forum Message <> Reply to Message

> Between the "normal" case and the "net namespace + macvlan" case,
> results are about the same for both the throughput and the local CPU
> load for the following test types: TCP_MAERTS, TCP_RR, UDP_STREAM, UDP_RR.
>
> macvlan looks like a very good candidate for network namespace in these

> cases.
>
> But, with the TCP_STREAM test, I observed the CPU load is about the
> same (that's what we wanted) but the throughput decreases by about 5%:
> from 850MB/s down to 810MB/s.
> I haven't investigated yet why the throughput decrease in the case.
> Does it come from my setup, from macvlan additional treatments, other? I
> don't know yet

Given that your "normal" case doesn't hit link-rate on the TCP_STREAM, but it does with UDP_STREAM, it could be that there isn't quite enough TCP window available, particularly given it seems the default settings for sockets/windows are in use. You might try your normal case with the test-specific -S and -s options to increase the socket buffer size:

netperf -H 192.168.76.1 -i 30,3 -l 20 -t TCP_STREAM -- -m 1400 -S 128K -S 128K

and see if that gets you link-rate. One other possibility there is the use of the 1400 byte send - that probably doesn't interact terribly well with TSO. Also, it isn't (?) likely the MSS for the connection, which you can have reported by adding a "-v 2" to the global options. You could/should then use the MSS in a subsequent test, or perhaps better still use a rather larger send size for TCP_STREAM|TCP_MAERTS - I myself for no particular reason tend to use either 32KB or 64KB as the send size in the netperf TCP_STREAM tests I run.

A final WAG - that the 1400 byte send size interacted poorly with the Nagle algorithm since it was a sub-MSS send. When Nagle is involved, things can be very timing-sensitive, change the timing ever so slightly and you can have a rather larger change in throughput. That could be dealt-with either with the larger send sizes mentioned above, or by adding a test-specific -D option to set TCP_NODELAY.

happy benchmarking,

rick jones

_____

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