**Network Performance Tuning**

In order to achieve the highest bandwidth from high-speed network devices, (40Gb and faster), it is highly likely that you will need to perform some tuning. The following steps were taken to tune the test system. These steps were performed on a system with 100Gb network devices and are provided only as an example.

**References**

* <https://srcc.stanford.edu/100g-network-adapter-tuning>
* <https://community.mellanox.com/docs/DOC-2496>
* <https://linuxconfig.org/how-to-enable-jumbo-frames-in-linux>
* <https://fasterdata.es.net/>
* <https://www.mellanox.com/related-docs/prod_software/Mellanox_EN_for_Linux_Release_Notes_v4_4-1_0_1_0.pdf>

**Please Note:** The following actions were all performed as the "*root*" user. Most or all of these steps will require you to have root access. This can be achieved either by running each command via "*sudo*" or by first switching to the root account completely by using "*sudo su -*

Additionally, not all tuning steps are supported on all systems or devices. For example, the NUMA steps only work on systems that support NUMA. Setting "*MaxReadReq*" comes directly from tuning suggestions for Mellanox 100Gb adapters and thus may not be supported on adapters from other vendors.

**Tuning Steps**

1. Find the device's NUMA node
2. root@sys-6029u-trt:~# cat /sys/class/net/enp94s0f0/device/numa\_node
3. 0
4. root@sys-6029u-trt:~# cat /sys/class/net/enp94s0f1/device/numa\_node
5. 0
6. Find which CPU(s) the nodes are associated with:
7. root@sys-6029u-trt:~# lscpu |grep NUMA
8. NUMA node(s):          2
9. NUMA node0 CPU(s):     0-17,36-53
10. NUMA node1 CPU(s):     18-35,54-71

\* this lets us set affinity to keep the iperf processes close to the CPU

1. Check current CPU frequencies:
2. root@sys-6029u-trt:~# grep -E '^cpu MHz' /proc/cpuinfo
3. cpu MHz         : 1000.000
4. Set governor on all CPUs (Requires install of cpufrequtils)
5. for x in `seq 0 71`;do
6. cpufreq-set -r -g performance -c $x
7. done
8. Check frequencies again
9. grep -E '^cpu MHz' /proc/cpuinfo
10. cpu MHz         : 2301.000

Note they should now be at or above the CPU max.

1. Make sure the card is in the right slot (100Gb cards should show speed of 8GT/s and Width of 16x otherwise the PCIe slot it’s in can’t handle the throughput)
2. lspci -s 04:00.0 -vvv | grep Speed
3. LnkCap: Port #0, Speed 8GT/s, Width x16, ASPM not supported, Exit Latency L0s unlimited, L1 unlimited
4. LnkSta: Speed 8GT/s, Width x16, TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
5. Check the MaxReadRequest using the PCI address of each port (you have to set this per port)
6. lspci -s 04:00.0 -vvv | grep MaxReadReq
7. MaxPayload 256 bytes, MaxReadReq 512 bytes
8. setpci -s 04:00.0 68.w
9. 2936
10. Set Max Read Request to the upper limit
11. setpci -s 04:00.0 68.w=5936
12. lspci -s 04:00.0 -vvv | grep MaxReadReq
13. MaxPayload 256 bytes, MaxReadReq 4096 bytes
14. Set buffer to 512M Buffers
15. root@sys-6029u-trt:~# sysctl net.core.rmem\_max=563870912
16. net.core.rmem\_max = 563870912
17. root@sys-6029u-trt:~# sysctl net.core.wmem\_max=563870912
18. net.core.wmem\_max = 563870912
19. Increase linux autotuning TCP Buffer limits to 256MB
20. root@sys-6029u-trt:~# sudo sysctl net.ipv4.tcp\_rmem="4096 87380 268435456"
21. net.ipv4.tcp\_rmem = 4096 87380 268435456
22. root@sys-6029u-trt:~# sudo sysctl net.ipv4.tcp\_wmem="4096 87380 268435456"
23. net.ipv4.tcp\_wmem = 4096 87380 268435456
24. Set max\_backlog to 300K
25. root@sys-6029u-trt:~# sysctl net.core.netdev\_max\_backlog=300000
26. net.core.netdev\_max\_backlog = 300000
27. Don't cache ssthresh from previous connection
28. # sysctl net.ipv4.tcp\_no\_metrics\_save=1
29. net.ipv4.tcp\_no\_metrics\_save = 1
30. Explicitly set htcp as the congestion control. You could also set this to 'bbr'.
31. # sysctl net.ipv4.tcp\_congestion\_control=htcp
32. net.ipv4.tcp\_congestion\_control = htcp
33. If you are using Jumbo Frames, also set this
34. # sysctl net.ipv4.tcp\_mtu\_probing=1
35. net.ipv4.tcp\_mtu\_probing = 1
36. Set default qdisc to fq
37. # sysctl net.core.default\_qdisc=fq
38. net.core.default\_qdisc = fq
39. NIC tweaks:
    1. Turn on Large Receive Offload:
    2. # ethtool -K enp216s0f0 lro on
    3. # ethtool -K enp216s0f1 lro on
    4. Set txqueuelen buffer higher:
    5. # ifconfig enp216s0f0 txqueuelen 20000
    6. # ifconfig enp216s0f1 txqueuelen 20000
    7. Enable jumbo frames:
    8. # ip link set enp216s0f0 mtu 9000
    9. # ip link set enp216s0f1 mtu 9000
    10. On some Mellanox ConnectX-4/ConnectX-5 cards, you may still only see around 60Gb/s on what should be a 100Gb/s device. This may be due to an issue with Adaptive RX being in use alongside hardware LRO.
    11. # ethtool -C adaptive-rx off
    12. # ethtool -C rx-usecs 8 rx-frames 128

Note, this only applies to Mellanox cards. See Issue #1241056 in the [driver release notes](https://www.mellanox.com/related-docs/prod_software/Mellanox_EN_for_Linux_Release_Notes_v4_4-1_0_1_0.pdf).

1. Turn off irqbalance:
2. # systemctl stop irqbalance
3. # systemctl status irqbalance |grep Active
4. Active: inactive (dead) since Tue 2018-04-17 20:58:16 UTC; 23s ago

**For testing with iperf3**

*Note, the example below shows tests run over 60 seconds. Actual certification testing requires a test run of 1 hour per port. Thus for certification testing you would need to use* "-t 3600" *rather than* "-t 60".

On the iperf target server, start 4 iperf3 daemons on different ports, pinned to NUMA Node 0 cores (see #2 above)

# iperf3 -sD -B 172.16.21.2 -p5101 -A0

# iperf3 -sD -B 172.16.21.2 -p5102 -A14

# iperf3 -sD -B 172.16.21.2 -p5103 -A36

# iperf3 -sD -B 172.16.21.2 -p5104 -A52

Note we're using -A to ensure each process is on a CPU core on the same NUMA node that our 100Gb NIC is attached to. On the System Under Test, kick off four iperf3 processes, one for each remote port. Please note that this is for example only. It is easier and neater to perform this using the tool "parallel" as noted in the next section.

$ iperf3 -c 172.16.21.1 -O 15 -t 60 -p 5101 -R -i 60 -T s1 & iperf3 -c 172.16.21.1 -O 15 -t 60 -p 5102 -R -i 60 -T s2 & iperf3 -c 172.16.21.1 -O 15 -t 60 -p 5103 -R -i 60 -T s3 & iperf3 -c 172.16.21.1 -O 15 -t 60 -p 5104 -R -i 60 -T s4&

This is abbreviated output

s4:  [ ID] Interval           Transfer Bandwidth    Retr

s4:  [ 4]   0.00-60.00  sec 161 GBytes  23.1 Gbits/sec 18726             sender

s4:  [ 4]   0.00-60.00  sec 161 GBytes  23.1 Gbits/sec              receiver

s4:

s4:  iperf Done.

s3:  [ ID] Interval           Transfer Bandwidth

s3:  [ 4]   0.00-60.00  sec 160 GBytes  22.9 Gbits/sec

s3:  - - - - - - - - - - - - - - - - - - - - - - - - -

s3:  [ ID] Interval           Transfer Bandwidth    Retr

s3:  [ 4]   0.00-60.00  sec 160 GBytes  22.9 Gbits/sec 16953             sender

s3:  [ 4]   0.00-60.00  sec 160 GBytes  22.9 Gbits/sec              receiver

s3:

s3:  iperf Done.

s2:  [ ID] Interval           Transfer Bandwidth

s2:  [ 4]   0.00-60.00  sec 163 GBytes  23.3 Gbits/sec

s2:  - - - - - - - - - - - - - - - - - - - - - - - - -

s2:  [ ID] Interval           Transfer Bandwidth    Retr

s2:  [ 4]   0.00-60.00  sec 163 GBytes  23.3 Gbits/sec 17582             sender

s2:  [ 4]   0.00-60.00  sec 163 GBytes  23.3 Gbits/sec              receiver

s1:  [ ID] Interval           Transfer Bandwidth

s2:

s2:  iperf Done.

s1:  [ 4]   0.00-60.00  sec 159 GBytes  22.7 Gbits/sec

s1:  - - - - - - - - - - - - - - - - - - - - - - - - -

s1:  [ ID] Interval           Transfer Bandwidth    Retr

s1:  [ 4]   0.00-60.00  sec 159 GBytes  22.7 Gbits/sec 17869             sender

s1:  [ 4]   0.00-60.00  sec 159 GBytes  22.7 Gbits/sec              receiver

The average bandwidth over 60 seconds for all four threads adds up to 92Gb/s.

**Running multiple iperf3 instances with parallel**

Parallel is a tool that will allow you to run multiple commands at the same time. The following outlines how we used parallel to do some testing with iperf3.

1. Ensure that parallel is installed.
2. # sudo apt-get -y install parallel
3. Create a file that looks like this:
4. # cat commands.txt
5. iperf3 -c 172.16.21.1 -O 15 -t 30 -p 5101 -R -i 60 -T s1
6. iperf3 -c 172.16.21.1 -O 15 -t 30 -p 5102 -R -i 60 -T s2
7. iperf3 -c 172.16.21.1 -O 15 -t 30 -p 5103 -R -i 60 -T s3
8. iperf3 -c 172.16.21.1 -O 15 -t 30 -p 5104 -R -i 60 -T s4
9. Execute the commands like this:
10. # parallel -a commands.txt |tee -a 100Gb-Port0.log
11. When using programs that use GNU Parallel to process data for publication please cite:
12. O. Tange (2011): GNU Parallel - The Command-Line Power Tool,
13. ;login: The USENIX Magazine, February 2011:42-47.
14. This helps funding further development; and it won't cost you a cent.
15. Or you can get GNU Parallel without this requirement by paying 10000 EUR.
16. To silence this citation notice run 'parallel --bibtex' once or use '--no-notice'.
17. s1:  Connecting to host 172.16.21.1, port 5101
18. s1:  Reverse mode, remote host 172.16.21.1 is sending
19. s1:  [ 4] local 172.16.21.11 port 47762 connected to 172.16.21.1 port 5101
20. s1:  [ ID] Interval           Transfer Bandwidth
21. s1:  [ 4]   0.00-30.00  sec 74.4 GBytes  21.3 Gbits/sec
22. s1:  - - - - - - - - - - - - - - - - - - - - - - - - -
23. s1:  [ ID] Interval           Transfer Bandwidth    Retr
24. s1:  [ 4]   0.00-30.00  sec 74.5 GBytes  21.3 Gbits/sec 39793             sender
25. s1:  [ 4]   0.00-30.00  sec 74.4 GBytes  21.3 Gbits/sec              receiver
26. s1:
27. s1:  iperf Done.
28. s2:  Connecting to host 172.16.21.1, port 5102
29. s2:  Reverse mode, remote host 172.16.21.1 is sending
30. s2:  [ 4] local 172.16.21.11 port 33354 connected to 172.16.21.1 port 5102
31. s2:  [ ID] Interval           Transfer Bandwidth
32. s2:  [ 4]   0.00-30.00  sec 79.6 GBytes  22.8 Gbits/sec
33. s2:  - - - - - - - - - - - - - - - - - - - - - - - - -
34. s2:  [ ID] Interval           Transfer Bandwidth    Retr
35. s2:  [ 4]   0.00-30.00  sec 79.7 GBytes  22.8 Gbits/sec 43638             sender
36. s2:  [ 4]   0.00-30.00  sec 79.6 GBytes  22.8 Gbits/sec              receiver
37. s2:
38. s2:  iperf Done.
39. s3:  Connecting to host 172.16.21.1, port 5103
40. s3:  Reverse mode, remote host 172.16.21.1 is sending
41. s3:  [ 4] local 172.16.21.11 port 57094 connected to 172.16.21.1 port 5103
42. s3:  [ ID] Interval           Transfer Bandwidth
43. s3:  [ 4]   0.00-30.00  sec 75.3 GBytes  21.6 Gbits/sec
44. s3:  - - - - - - - - - - - - - - - - - - - - - - - - -
45. s3:  [ ID] Interval           Transfer Bandwidth    Retr
46. s3:  [ 4]   0.00-30.00  sec 75.4 GBytes  21.6 Gbits/sec 41230             sender
47. s3:  [ 4]   0.00-30.00  sec 75.3 GBytes  21.6 Gbits/sec              receiver
48. s3:
49. s3:  iperf Done.
50. s4:  Connecting to host 172.16.21.1, port 5104
51. s4:  Reverse mode, remote host 172.16.21.1 is sending
52. s4:  [ 4] local 172.16.21.11 port 59674 connected to 172.16.21.1 port 5104
53. s4:  [ ID] Interval           Transfer Bandwidth
54. s4:  [ 4]   0.00-30.00  sec 75.7 GBytes  21.7 Gbits/sec
55. s4:  - - - - - - - - - - - - - - - - - - - - - - - - -
56. s4:  [ ID] Interval           Transfer Bandwidth    Retr
57. s4:  [ 4]   0.00-30.00  sec 75.8 GBytes  21.7 Gbits/sec 41177             sender
58. s4:  [ 4]   0.00-30.00  sec 75.7 GBytes  21.7 Gbits/sec              receiver
59. s4:
60. s4:  iperf Done.