Mellanox NIC's Performance Report with DPDK 20.08

Rev 1.1

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation NVIDIA makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer Terms of Sale NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

For the complete and most updated list of Mellanox trademarks, visit http://www.mellanox.com/page/trademarks.

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051

http://www.nvidia.com

Table of Contents

1 About this Report 5

1.1 Target Audience 5

1.2 Terms and Conventions 5

2 Test Description 6

2.1 Hardware Components 6

2.2 Zero Packet Loss Test 6

2.3 Zero Packet Loss over SR-IOV Test 6

2.4 Single Core Performance Test 6

3 Test#1 Mellanox ConnectX-4 Lx 25GbE Throughput at Zero Packet Loss (2x 25GbE) 7

3.1 Test Settings 8

3.2 Test Results 9

4 Test#2 Mellanox ConnectX-5 25GbE Throughput at Zero Packet Loss (2x 25GbE) 10

4.1 Test Settings 11

4.2 Test Results 12

5 Test#3 Mellanox ConnectX-5 25GbE Single Core Performance (2x 25GbE) 13

5.1 Test Settings 14

5.2 Test Results 15

6 Test#4 Mellanox ConnectX-5 25GbE Throughput at Zero Packet Loss (2x 25GbE) using SR-IOV over VMware ESXi 6.7 16

6.1 Test Settings 17

6.2 Test Results 18

7 Test#5 Mellanox ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss (1x 100GbE) 19

7.1 Test Settings 20

7.2 Test Results 21

8 Test#6 Mellanox ConnectX-6Dx 100GbE Single Core Performance (2x 100GbE) 22

8.1 Test Settings 23

8.2 Test Results 24

9 Test#7 Mellanox ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss (1x 100GbE) using SR-IOV over KVM Hypervisor 25

9.1 Test Settings 27

9.2 Test Results 29

Document History

Table 1 - Document History

Version	Date	Description of Change
1.0	28-Sep-2020	Initial report release
1.1	25-Nov-2020	Added BlueField-2 testing and results

1 About this Report

The purpose of this report is to provide packet rate performance data for Mellanox ConnectX-4 Lx, ConnectX-5, ConnectX-6 Dx Network Interface Cards (NICs) and BlueField-2 Data Processing Unit (DPU) achieved with the specified Data Plane Development Kit (DPDK) release. The report provides the measured packet rate performance as well as the hardware layout, procedures, and configurations for replicating these tests.

The document does not cover all network speeds available with the ConnectX or BlueField family of NICs / DPUs and is intended as a general reference of achievable performance for the specified DPDK release.

1.1 Target Audience

This document is intended for engineers implementing applications with DPDK to guide and help achieving optimal performance.

1.2 Terms and Conventions

The following terms, abbreviations, and acronyms are used in this document.

Table 2 - Terms, Abbreviations and Acronyms

Term	Description
DPU	Data Processing Unit
DUT	Device Under Test
IXIA	
MPPS	Million Packets Per Seconds
PPS	Packets Per Second
OFED	OpenFabrics Enterprise Distribution; An open-source software for RDMA & kernel bypass. Read more on Mellanox OFED here.
SR-IOV	Single Root IO Virtualization
ZPL	Zero Packet Loss

2 Test Description

2.1 Hardware Components

The following hardware components are used in the test setup:

- HPE® ProLiant DL380 Gen10 Server
- Mellanox ConnectX-4 Lx, ConnectX-5,ConnectX-6 Dx Network Interface Cards (NICs) and BlueField-2 Data Processing Unit (DPU)
- IXIA® XM12 packet generator

2.2 Zero Packet Loss Test

Zero Packet Loss tests utilize **l3fwd** (http://www.dpdk.org/doc/guides/sample_app_ug/l3_forward.html) as the test application for testing maximum throughput with zero packet loss at various frame sizes based on RFC2544 https://tools.ietf.org/html/rfc2544.

The packet generator transmits a specified frame rate towards the Device Under Test (DUT) and counts the received frame rate sent back from the DUT. Throughput is determined with the maximum achievable transmit frame rate and is equal to the received frame rate i.e. zero packet loss.

- Duration for each test is 60 seconds.
- Traffic of 8192 IP flows is generated per port.
- IxNetwork (Version 9.00EA) is used with the IXIA packet generator.

2.3 Zero Packet Loss over SR-IOV Test

The test is conducted similarly to the bare-metal zero packet loss test with the distinction of having the DPDK application running in a Guest OS inside a VM utilizing SR-IOV virtual function.

2.4 Single Core Performance Test

Single Core performance tests utilize **testpmd** (http://www.dpdk.org/doc/guides/testpmd_app_ug), for testing the max throughput while using a single CPU core. The duration of the test is 60 seconds and the average throughput that is recorded during that time is used as the result of the test.

- Duration for each test is 60 seconds.
- Traffic of 8192 UDP flows is generated per port.
- IxNetwork (Version 9.00EA) is used with the IXIA packet generator.

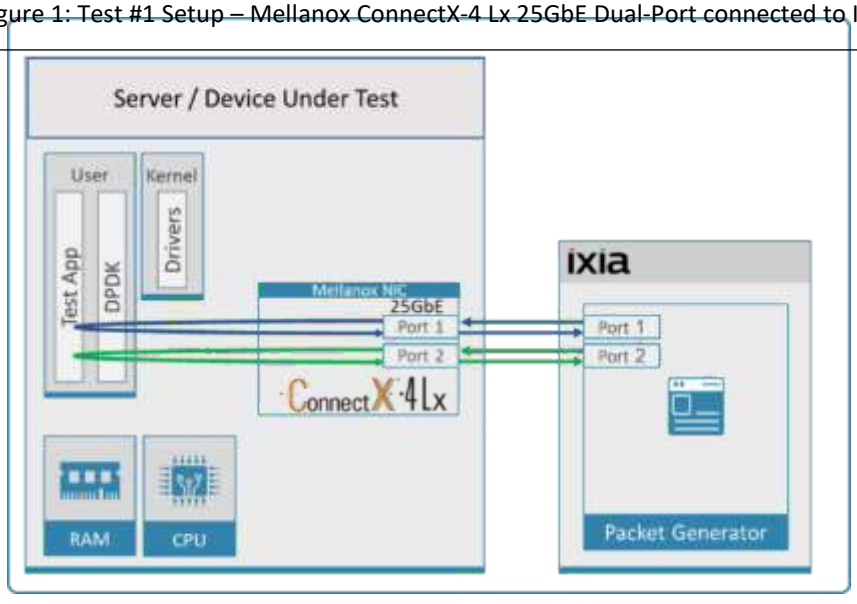
3 Test#1 Mellanox ConnectX-4 Lx 25GbE Throughput at Zero Packet Loss (2x 25GbE)

Table 3: Test #1 Setup

Item	Description
Test #1	Mellanox ConnectX-4 Lx 25GbE Dual-Port Throughput at zero packet loss
Server	HPE ProLiant DL380 Gen10
CPU	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz 24 CPU cores * 2 NUMA nodes
RAM	384GB: 6 * 32GB DIMMs * 2 NUMA nodes @ 2666MHz
BIOS	U30 rev. 1.36 (02/15/2018)
NIC	One MCX4121A-ACAT - ConnectX-4 Lx network interface card 25GbE dual-port SFP28; PCIe3.0 x8; ROHS R6
Operating System	Red Hat Enterprise Linux Server release 7.7 (Maipo)
Kernel Version	3.10.0-1062.el7.x86_64
GCC version	4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
Mellanox NIC firmware version	14.29.0332
Mellanox OFED driver version	MLNX_OFED_LINUX-5.1-0.6.6.0
DPDK version	20.08
Test Configuration	1 NIC, 2 ports used on the NIC. Each port receives a stream of 8192 IP flows from the IXIA Each port has 4 queues assigned for a total of 8 queues 1 queue assigned per logical core with a total of 8 logical cores

The Device Under Test (DUT) is made up of the HPE server and the Mellanox ConnectX-4 Lx Dual-Port NIC. The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-4 Lx NIC.
The ConnectX-4 Lx data traffic is passed through DPDK to the test application **l3fwd** and is redirected to the opposite direction on the opposing port. IXIA measures throughput and packet loss.

Figure 1: Test #1 Setup – Mellanox ConnectX-4 Lx 25GbE Dual-Port connected to IXIA



3.1 Test Settings

Table 4: Test #1 Settings

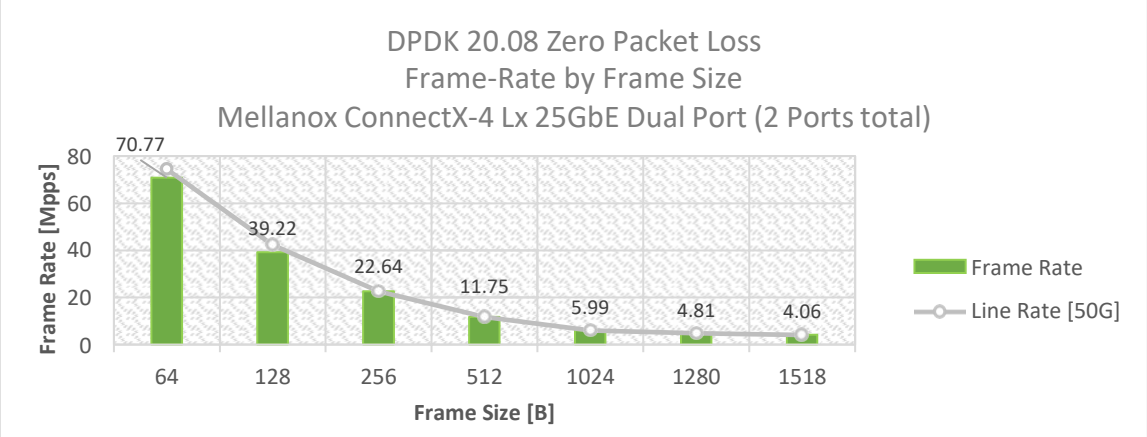
Item	Description
BIOS	1) Workload Profile = Low Latency"; 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) See Configuring and tuning HPE ProLiant Servers for low-latency applications": hpe.com > Search DL380 gen10 low latency"
BOOT Settings	isolcpus=24-47 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable nohz_full=24-47 rcu_nocbs=24-47 rcu_nocb_poll default_hugepagesz=1G hugepagesz=1G hugepages=64 audit=0 nosoftlockup
DPDK Settings	Enable mlx5 PMD before compiling DPDK: In .config file generated by "make config", set: "CONFIG_RTE_LIBRTE_MLX5_PMD=y" During testing, l3fwd was given real-time scheduling priority.
L3fwd settings	Updated values /l3fwd/l3fwd.h: #define RTE_TEST_RX_DESC_DEFAULT 4096 #define RTE_TEST_TX_DESC_DEFAULT 4096 #define MAX_PKT_BURST 64
Command Line	./examples/l3fwd/build/app/l3fwd -c 0xff0000000000 -n 4 -w d8:00.0,txq_inline=200,txq_mpw_en=1 -w d8:00.1,txq_inline=200,txq_mpw_en=1 --socket-mem=0,8192 -- -p 0x3 -P --config='(0,0,47),(0,1,46),(0,2,45),(0,3,44),(1,0,43),(1,1,42),(1,2,41),(1,3,40)' --eth-dest=0,00:52:11:22:33:10 --eth-dest=1,00:52:11:22:33:20
Other optimizations	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Change PCI MaxReadReq to 1024B for each port of each NIC: Run "setpci -s $PORT_PCI_ADDRESS 68.w", it will return 4 digits ABCD --> Run "setpci -s $PORT_PCI_ADDRESS 68.w=3BCD" f) Set CQE COMPRESSION to AGGRESSIVE": mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1 G) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us

3.2 Test Results

Table 5: Test #1 Results – Mellanox ConnectX-4 Lx 25GbE Dual-Port Throughput at Zero Packet Loss

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [50G] (Mpps)	% Line Rate
64	70.77	74.4	94.83
128	39.22	42.23	92.88
256	22.64	22.64	100
512	11.75	11.75	100
1024	5.99	5.99	100
1280	4.81	4.81	100
1518	4.06	4.06	100

Figure 2: Test #1 Results – Mellanox ConnectX-4 Lx 25GbE Dual-Port Throughput at Zero Packet Loss



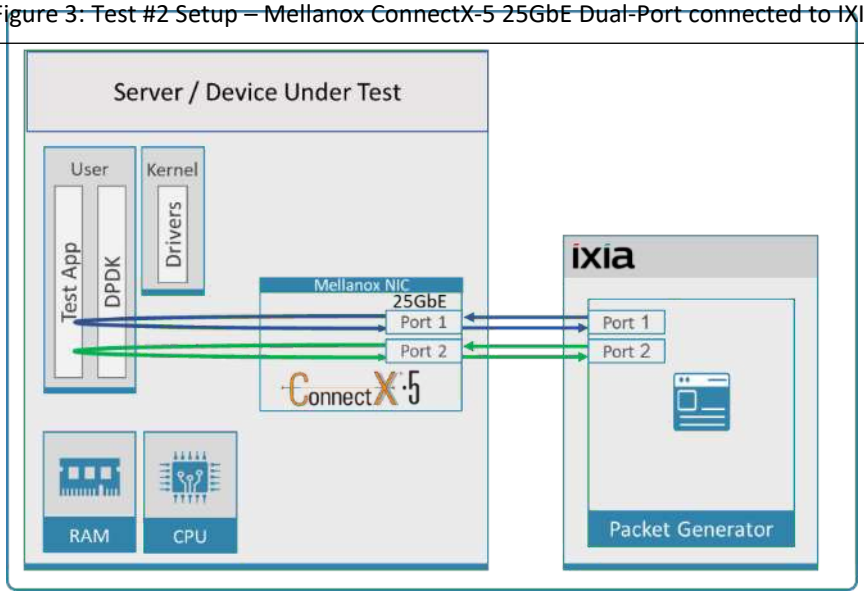
4 Test#2 Mellanox ConnectX-5 25GbE Throughput at Zero Packet Loss (2x 25GbE)

Table 6: Test #2 Setup

Item	Description
Test #2	Mellanox ConnectX-5 25GbE Dual-Port Throughput at zero packet loss
Server	HPE ProLiant DL380 Gen10
CPU	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz 24 CPU cores * 2 NUMA nodes
RAM	384GB: 6 * 32GB DIMMs * 2 NUMA nodes @ 2666MHz
BIOS	U30 rev. 1.36 (02/15/2018)
NIC	One MCX512A-ACAT ConnectX-5 EN network interface card; 10/25GbE dual-port SFP28; PCIe3.0 x8; tall bracket; ROHS R6
Operating System	Red Hat Enterprise Linux Server release 7.7 (Maipo)
Kernel Version	3.10.0-1062.el7.x86_64
GCC version	4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
Mellanox NIC firmware version	16.28.1002
Mellanox OFED driver version	MLNX_OFED_LINUX-5.1-0.6.6.0
DPDK version	20.08
Test Configuration	1 NIC, 2 ports; Each port receives a stream of 8192 IP flows from the IXIA Each port has 4 queues assigned for a total of 8 queues 1 queue assigned per logical core with a total of 8 logical cores

The Device Under Test (DUT) is made up of the HPE server and the Mellanox ConnectX-5 Dual-Port NIC. The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-5 NIC.
The ConnectX-5 data traffic is passed through DPDK to the test application **l3fwd** and is redirected to the opposite direction on the same port. IXIA measures throughput and packet loss.

Figure 3: Test #2 Setup – Mellanox ConnectX-5 25GbE Dual-Port connected to IXIA



4.1 Test Settings

Table 7: Test #2 Settings

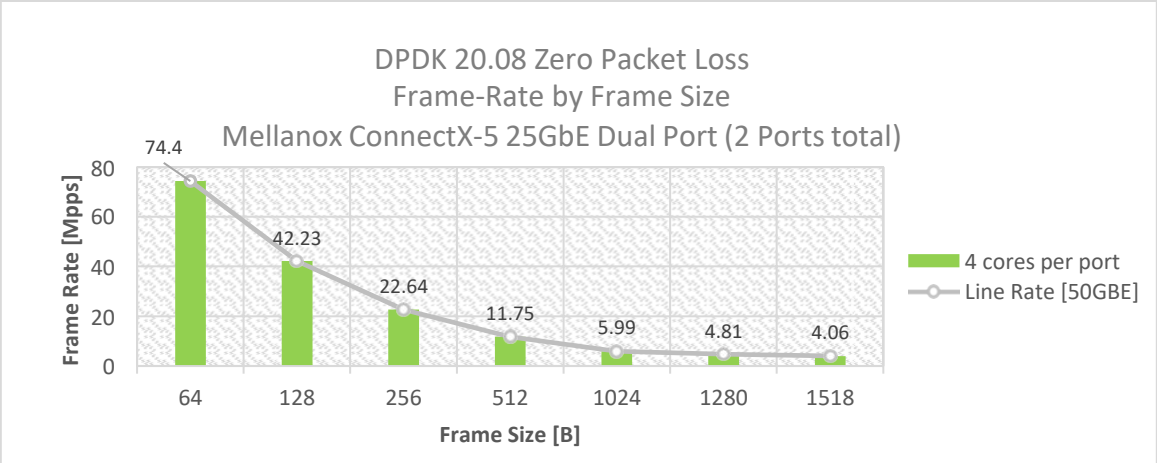
Item	Description
BIOS	1) Workload Profile = Low Latency"; 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) See Configuring and tuning HPE ProLiant Servers for low-latency applications": hpe.com > Search DL380 gen10 low latency"
BOOT Settings	isolcpus=24-47 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable nohz_full=24-47 rcu_nocbs=24-47 rcu_nocb_poll default_hugepagesz=1G hugepagesz=1G hugepages=64 audit=0 nosoftlockup
DPDK Settings	Enable mlx5 PMD before compiling DPDK: In .config file generated by "make config", set: "CONFIG_RTE_LIBRTE_MLX5_PMD=y" During testing, l3fwd was given real-time scheduling priority.
L3fwd settings	Updated values /l3fwd/l3fwd.h: #define RTE_TEST_RX_DESC_DEFAULT 4096 #define RTE_TEST_TX_DESC_DEFAULT 4096 #define MAX_PKT_BURST 64
Command Line	./examples/l3fwd/build/app/l3fwd -c 0xff0000000000 -n 4 -w d8:00.0,mprq_en=1,rxqs_min_mprq=1 -w d8:00.1,mprq_en=1,rxqs_min_mprq=1 --socket-mem=0,8192 -- -p 0x3 -P --config='(0,0,47),(0,1,46),(0,2,45),(0,3,44),(1,0,43),(1,1,42),(1,2,41),(1,3,40)' --eth-dest=0,00:52:11:22:33:10 --eth-dest=1,00:52:11:22:33:20
Other optimizations	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Change PCI MaxReadReq to 1024B for each port of each NIC: Run "setpci -s $PORT_PCI_ADDRESS 68.w", it will return 4 digits ABCD --> Run "setpci -s $PORT_PCI_ADDRESS 68.w=3936" f) Set CQE COMPRESSION to AGGRESSIVE": mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1 g) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us

4.2 Test Results

Table 8: Test #2 Results – Mellanox ConnectX-5 25GbE Dual-Port Throughput at Zero Packet Loss

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [50G] (Mpps)	% Line Rate
64	74.40	74.40	100.00
128	42.23	42.23	100.00
256	22.64	22.64	100.00
512	11.75	11.75	100.00
1024	5.99	5.99	100.00
1280	4.81	4.81	100.00
1518	4.06	4.06	100.00

Figure 4: Test #2 Results – Mellanox ConnectX-5 25GbE Dual-Port Throughput at Zero Packet Loss



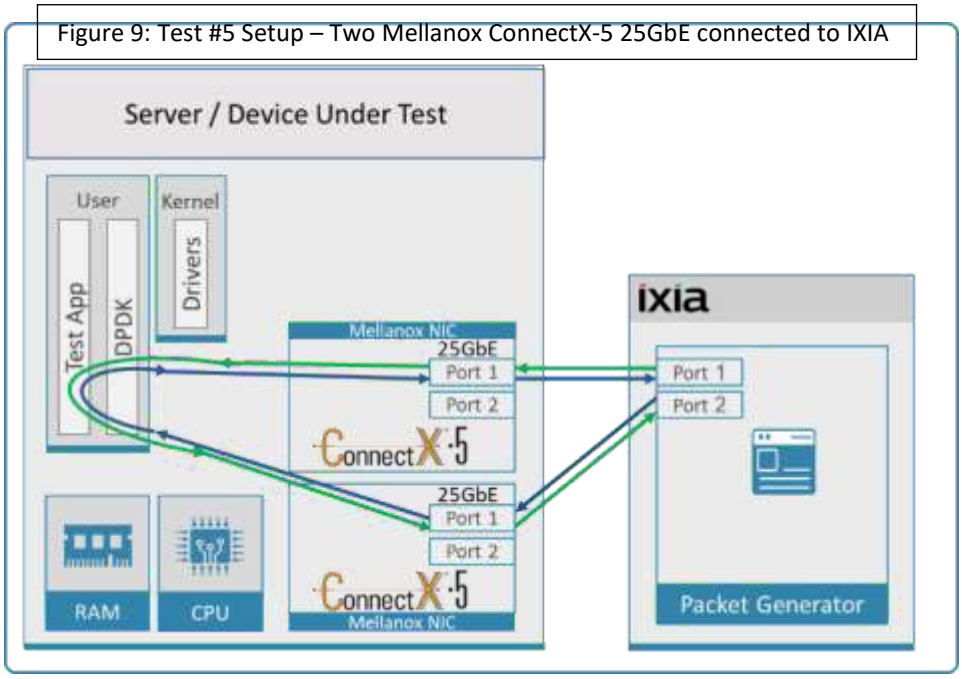
5 Test#3 Mellanox ConnectX-5 25GbE Single Core Performance (2x 25GbE)

Table 15: Test #5 Setup

Item	Description
Test #5	Mellanox ConnectX-5 25GbE Single Core Performance
Server	HPE ProLiant DL380 Gen10
CPU	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz 24 CPU cores * 2 NUMA nodes
RAM	384GB: 6 * 32GB DIMMs * 2 NUMA nodes @ 2666MHz
BIOS	U30 rev. 1.36 (02/15/2018)
NIC	Two MCX512A-ACA ConnectX-5 EN network interface cards; 10/25GbE dual-port SFP28; PCIe3.0 x8; tall bracket; ROHS R6
Operating System	Red Hat Enterprise Linux Server release 7.7 (Maipo)
Kernel Version	3.10.0-1062.el7.x86_64
GCC version	4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
Mellanox NIC firmware version	16.28.1002
Mellanox OFED driver version	MLNX_OFED_LINUX-5.1-0.6.6.0
DPDK version	20.08
Test Configuration	2 NICs; 1 port used on each. Each port receives a stream of 8192 UDP flows from the IXIA Each port has 1 queue assigned, a total of two queues for two ports, and both queues are assigned to the same single logical core.

The Device Under Test (DUT) is made up of the HPE server and two Mellanox ConnectX-5 25GbE NICs utilizing one port each. The DUT is connected to the IXIA packet generator which generates traffic towards the first port of both ConnectX-5 25GbE NICs.
The ConnectX-5 25GbE data traffic is passed through DPDK to the test application **testpmd** and is redirected to the opposite direction on the opposing NIC's port. IXIA measures throughput and packet loss.

Figure 9: Test #5 Setup – Two Mellanox ConnectX-5 25GbE connected to IXIA



5.1 Test Settings

Table 16: Test #5 Settings

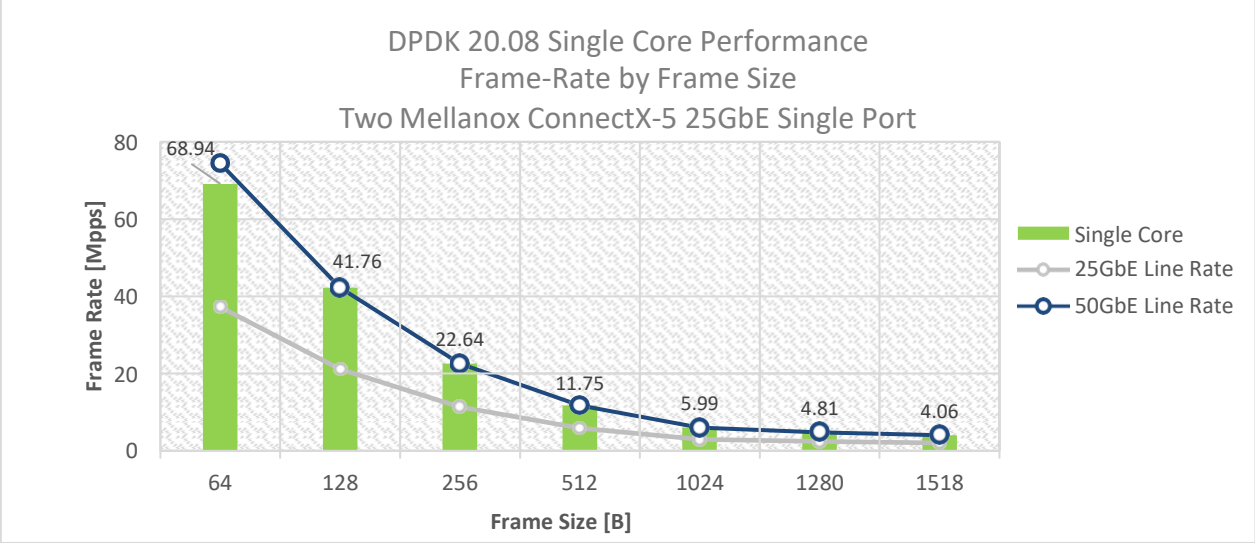
Item	Description
BIOS	1) Workload Profile = Low Latency” 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) See Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search DL380 gen10 low latency”
BOOT Settings	isolcpus=24-47 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable nohz_full=24-47 rcu_nocbs=24-47 rcu_nocb_poll default_hugepagesz=1G hugepagesz=1G hugepages=64 audit=0 nosoftlockup
DPDK Settings	Enable mlx5 PMD before compiling DPDK: In .config file generated by "make config", set: "CONFIG_RTE_LIBRTE_MLX5_PMD=y" set: "CONFIG_RTE_TEST_PMD_RECORD_CORE_CYCLES=y" During testing, testpmd was given real-time scheduling priority.
Command Line	/build/app/testpmd -c 0x30000000000 -n 4 -w d8:00.0 -w d9:00.0 --socket-mem=0,8192 -----port-numa-config=0,1,1,1 --socket-num=1 --burst=64 --txd=1024 --rxd=1024 --mbcache=512 --rxq=1 --txq=1 --nb-cores=1 -i -a --rss-udp --no-numa --disable-crc-strip
Other optimizations	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Change PCI MaxReadReq to 1024B for each port of each NIC: Run "setpci -s $PORT_PCI_ADDRESS 68.w", it will return 4 digits ABCD --> Run "setpci -s $PORT_PCI_ADDRESS 68.w=3BCD" f) Set CQE COMPRESSION to AGGRESSIVE”: mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1 g) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us

5.2 Test Results

Table 17: Test #5 Results – Mellanox ConnectX-5 25GbE Single Core Performance

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [25G] (Mpps)	Line Rate [50G] (Mpps)	Throughput (Gbps)	CPU Cycles per packet NOTE: Lower is Better
64	68.94	37.2	74.4	35.30	34
128	41.76	21.11	42.23	42.767	32
256	22.64	11.32	22.64	46.369	32
512	11.75	5.87	11.75	48.12	32
1024	5.99	2.99	5.99	49.037	37
1280	4.81	2.4	4.81	49.226	33
1518	4.06	2.03	4.06	49.342	37

Figure 10: Test #5 Results – Mellanox ConnectX-5 25GbE Single Core Performance



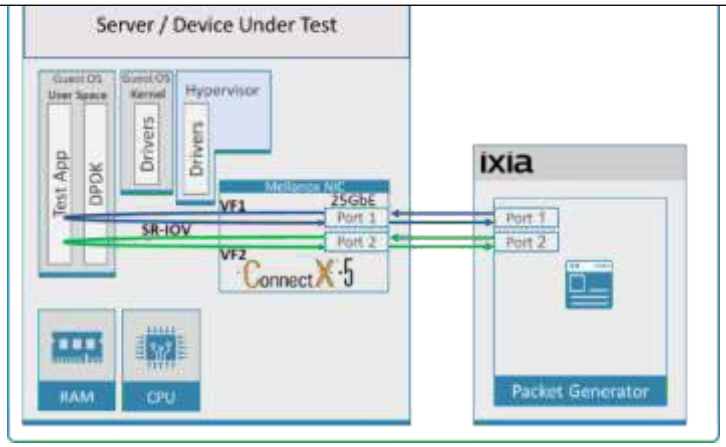
6 Test#4 Mellanox ConnectX-5 25GbE Throughput at Zero Packet Loss (2x 25GbE) using SR-IOV over VMware ESXi 6.7

Table 18: Test #6 Setup

Item	Description
Test #6	Mellanox ConnectX-5 25GbE Dual-Port Throughput at zero packet loss SRIOV over VMware ESXi 6.7U3
Server	HPE ProLiant DL380 Gen10
CPU	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz 24 CPU cores * 2 NUMA nodes
RAM	384GB: 6 * 32GB DIMMs * 2 NUMA nodes @ 2666MHz
BIOS	U30 rev. 1.36 (02/15/2018)
NIC	One MCX512A-ACAT ConnectX-5 EN network interface card; 10/25GbE dual-port SFP28; PCIe3.0 x8; tall bracket; ROHS R6
Hypervisor	VMware ESXi 6.7U3
Hypervisor Build	VMware-ESXi-6.7.0-Update3-15160138-HPE-Gen9plus-670.U3.10.5.5.25-Mar2020.iso
Hypervisor Mellanox Driver	MLNX-NATIVE-ESX-ConnectX-4-5_4.17.70.1
Guest Operating System	Red Hat Enterprise Linux Server release 7.7 (Maipo)
Guest Kernel Version	3.10.0-1062.el7.x86_64
Guest GCC version	4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
Guest Mellanox OFED driver version	MLNX_OFED_LINUX-5.1-0.6.6.0
Mellanox NIC firmware version	16.28.1002
DPDK version	20.08
Test Configuration	1 NIC, 2 ports with 1 VF per port (SR-IOV); Each port receives a stream of 8192 IP flows from the IXIA Each VF (SR-IOV) has 4 queues assigned for a total of 8 queues 1 queue assigned per logical core with a total of 8 logical cores.

The Device Under Test (DUT) is made up of the HPE server and the Mellanox ConnectX-5 NIC with dual-port. The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-5 NIC. The ConnectX-5 data traffic is passed to VF1 (SR-IOV assigned to Port1) and VF2 (SR-IOV assigned to Port2) to VM running over ESXi 6.5 hypervisor. VM runs **l3fwd** over DPDK and is redirects traffic to the opposite direction on the same VF/port. IXIA measures throughput and packet loss.

Figure 11: Test #6 Setup – Mellanox ConnectX-5 25GbE connected to IXIA using ESXi SR-IOV



6.1 Test Settings

Table 19: Test#6 Settings

Item	Description
BIOS	1) Workload Profile = Low Latency"; 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) 3) Change "Workload Profile" to "Custom" 4) Change VT-x, VT-d and SR-IOV from "Disabled" to "Enabled". See Configuring and tuning HPE ProLiant Servers for low-latency applications": hpe.com > Search DL380 gen10 low latency"
BOOT Settings Guest OS	isolcpus=0-22 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable idle=poll nohz_full=0-22 rcu_nocbs=0-22 rcu_nocb_poll default_hugepagesz=1G hugepagesz=1G hugepages=16 nosoftlockup
Hypervisor settings	1) Enable SRIOV via NIC configuration tool: (requires installation of mft-tools) /opt/mellanox/bin/mlxconfig -d <PCI ID> set NUM_OF_VFS=2 SRIOV_EN=1 CQE_COMPRESSION=1 reboot 2) Install Driver esxcli software vib install -MLNX-NATIVE-ESX-ConnectX-4-5_4.17.70.1-1OEM.670.0.0.8169922.zip reboot esxcfg-module -s 'max_vfs=1,1,1,1,1,1,1,1 supported_num_ports=8' nmlx5_core reboot 3) Virtual Hardware Configuration: CPU": 23 Cores per Socket" : 1 Sockets = 23) or 23 (Socket = 1) Hardware virtualization": enabled Scheduling Affinity": 25-47 CPU/MMU Virtualization": Hardware CPU and MMU" RAM": 32768 MB Reservation": 32768 MB Reserve all guest memory All locked)": enabled VM options > Advanced > Configuration Parameters" > Edit Configuration" : Add parameter: numa.nodeAffinity = 1 4) Create virtual switch: Networking>Virtual Switches>Add standard virtual switch>Switch_SRIOV_1> Uplink : select vmnicXXXX matching the card under test 5) Add port group to Switch_SRIOV_XX (VLAN=0): Networking>Port groups>Add port group>SRIOV_PG1>Switch_SRIOV_XX 6) Add 2xSRIOV network adapters to VM (same settings for both ports): Select correct port group created previously (SRIOV_PG1) Adapter Type: SR-IOV passthrough Physical function: select pci for the portX of the card under the test
DPDK Settings on Guest OS	Enable mlx5 PMD before compiling DPDK: In .config file generated by "make config", set: "CONFIG_RTE_LIBRTE_MLX5_PMD=y" During testing, l3fwd was given real-time scheduling priority.

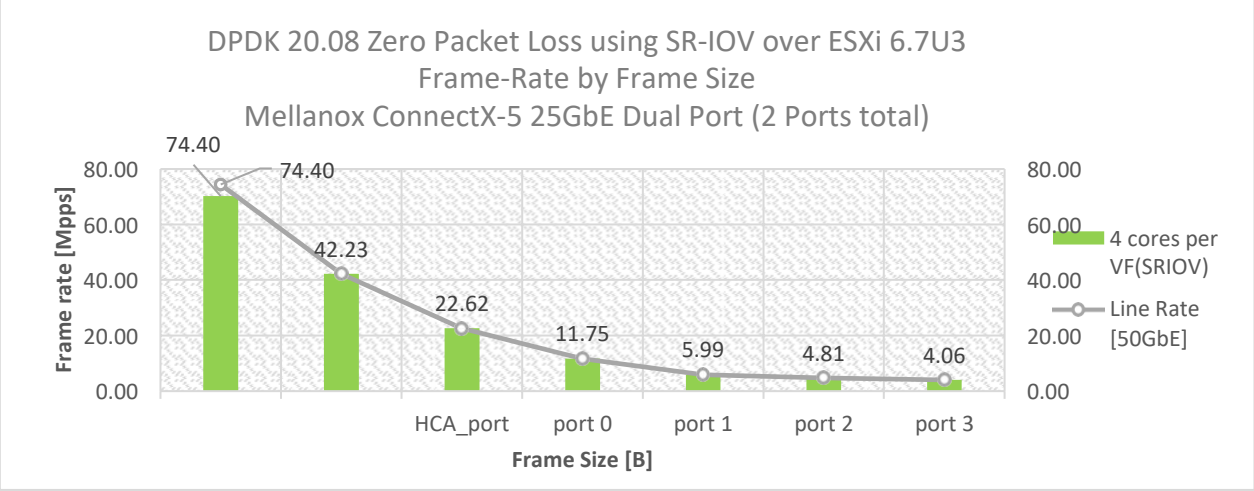
Item	Description
L3fwd settings on Guest OS	Updated values /l3fwd/l3fwd.h: #define RTE_TEST_RX_DESC_DEFAULT 2048 #define RTE_TEST_TX_DESC_DEFAULT 2048 #define MAX_PKT_BURST 64
Command Line on Guest OS	./examples/l3fwd/build/app/l3fwd -c 0x7f8000 -n 4 -w 13:00.0,mprq_en=1,rxqs_min_mprq=1 -w 1b:00.0,mprq_en=1,rxqs_min_mprq=1 --socket-mem=8192 -- -p 0x3 -P --config='(0,0,22),(0,1,21),(0,2,20),(0,3,19),(1,0,18),(1,1,17),(1,2,16),(1,3,15)' --eth-dest=0,00:52:11:22:33:10 --eth-dest=1,00:52:11:22:33:20
Other optimizations on Guest OS	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us

6.2 Test Results

Table 20: Test#6 Results – Mellanox ConnectX-5 25GbE Throughput at Zero Packet Loss using ESXi SR-IOV

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [50G] (Mpps)	% Line Rate
64	74.4	74.4	100.00
128	42.23	42.23	100.00
256	22.62	22.64	100.00
512	11.75	11.75	100.00
1024	5.99	5.99	100.00
1280	4.81	4.81	100.00
1518	4.06	4.06	100.00

Figure 12: Test#6 Results – Mellanox ConnectX-5 25GbE Throughput at Zero Packet Loss using ESXi SR-IOV



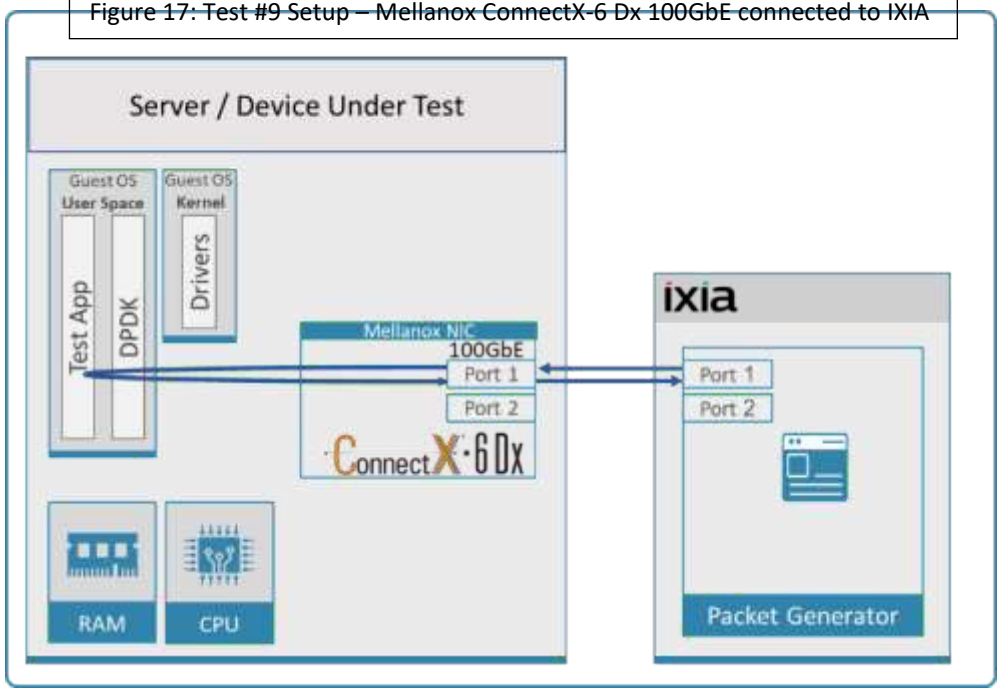
7 Test#5 Mellanox ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss (1x 100GbE)

Table 27: Test #9 Setup

Item	Description
Test #9	Mellanox ConnectX-6 Dx 100GbE Throughput at zero packet loss
Server	HPE ProLiant DL380 Gen10
CPU	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz 24 CPU cores * 2 NUMA nodes
RAM	384GB: 6 * 32GB DIMMs * 2 NUMA nodes @ 2666MHz
BIOS	U30 rev. 1.36 (02/15/2018)
NIC	One MCX623106AN-CDAT ConnectX-6 Dx EN adapter card; 100GbE; Dual-port QSFP56; PCIe 4.0/3.0 x16;
Operating System	Red Hat Enterprise Linux Server release 7.7 (Maipo)
Kernel Version	3.10.0-1062.el7.x86_64
GCC version	4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
Mellanox NIC firmware version	22.28.1002
Mellanox OFED driver version	MLNX_OFED_LINUX-5.1-0.6.6.0
DPDK version	20.08
Test Configuration	1 NIC, 1 port used on NIC; Port has 12 queues assigned to it, 1 queue per logical core for a total of 12 logical cores. Each port receives a stream of 8192 IP flows from the IXIA

The Device Under Test (DUT) is made up of the HPE server and the Mellanox ConnectX-6 Dx Dual-Port NIC (only the first port is used in this test). The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-6Dx NIC.
The ConnectX-6Dx data traffic is passed through DPDK to the test application **l3fwd** and is redirected to the opposite direction on the same port. IXIA measures throughput and packet loss.

Figure 17: Test #9 Setup – Mellanox ConnectX-6 Dx 100GbE connected to IXIA



7.1 Test Settings

Table 28: Test #9 Settings

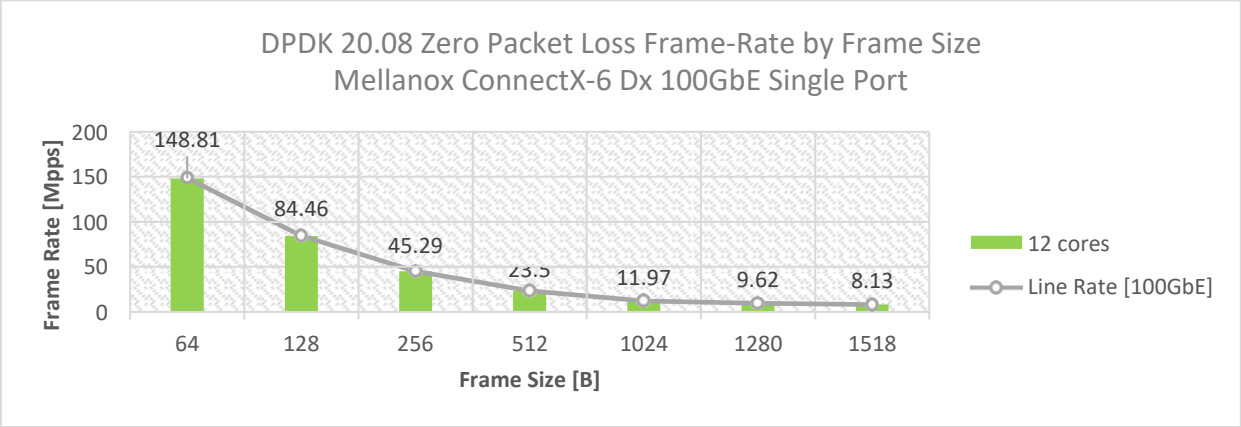
Item	Description
BIOS	1) Workload Profile = Low Latency"; 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) See Configuring and tuning HPE ProLiant Servers for low-latency applications": hpe.com > Search DL380 gen10 low latency"
BOOT Settings	isolcpus=24-47 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable nohz_full=24-47 rcu_nocbs=24-47 rcu_nocb_poll default_hugepagesz=1G hugepagesz=1G hugepages=64 audit=0 nosoflockup
DPDK Settings	Enable mlx5 PMD before compiling DPDK: In .config file generated by "make config", set: "CONFIG_RTE_LIBRTE_MLX5_PMD=y" During testing, l3fwd was given real-time scheduling priority.
L3fwd settings	Updated values /l3fwd/l3fwd.h: #define RTE_TEST_RX_DESC_DEFAULT 4096 #define RTE_TEST_TX_DESC_DEFAULT 4096 #define MAX_PKT_BURST 64
Command Line	./examples/l3fwd/build/app/l3fwd -c 0xfff000000000 -n 4 -w 0000:af:00.0,mprq_en=1,mprq_log_stride_num=8 --socket-mem=0,8192 -- -p 0x1 -P --config='(0,0,47),(0,1,46),(0,2,45),(0,3,44),(0,4,43),(0,5,42),(0,6,41),(0,7,40),(0,8,39),(0,9,38),(0,10,37),(0,11,36)' --eth-dest=0,00:52:11:22:33:10
Other optimizations	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Change PCI MaxReadReq to 1024B for each port of each NIC: Run "setpci -s $PORT_PCI_ADDRESS 68.w", it will return 4 digits ABCD --> Run "setpci -s $PORT_PCI_ADDRESS 68.w=3BCD" f) Set CQE COMPRESSION to AGGRESSIVE": mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1 g) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us

7.2 Test Results

Table 29: Test #9 Results –Mellanox ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [100G] (Mpps)	% Line Rate
64	148.81	148.81	100.00
128	84.46	84.46	100.00
256	45.29	45.29	100.00
512	23.50	23.50	100.00
1024	11.97	11.97	100.00
1280	9.62	9.62	100.00
1518	8.13	8.13	100.00

Figure 18: Test #9 Results – Mellanox ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss



8 Test#6 Mellanox ConnectX-6Dx 100GbE Single Core Performance (2x 100GbE)

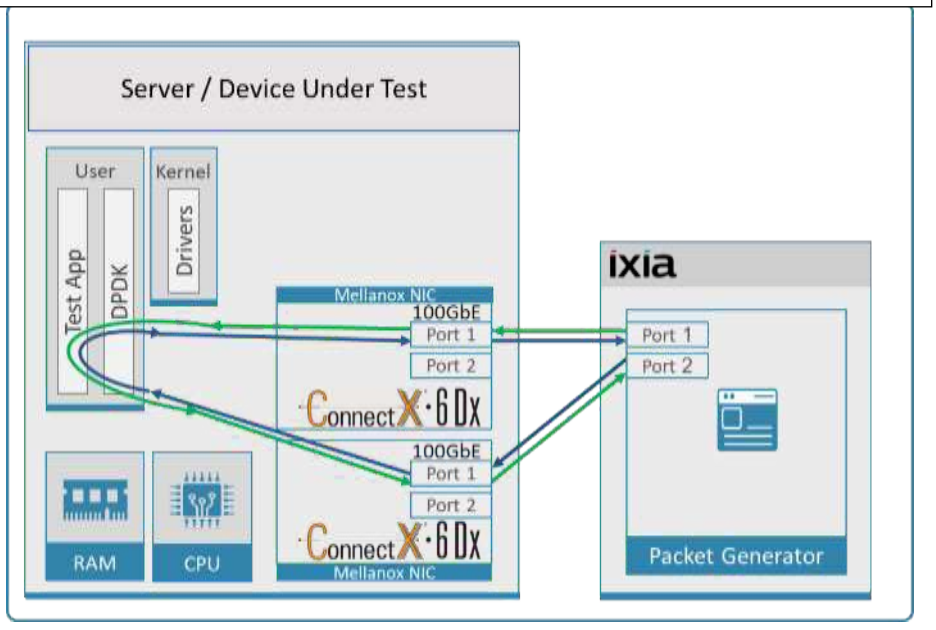
Table 30: Test #10 Setup

Item	Description
Test #10	Mellanox ConnectX-6Dx 100GbE Single Core Performance
Server	HPE ProLiant DL380 Gen10
CPU	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz; 24 CPU cores * 2 NUMA nodes
RAM	384GB: 6 * 32GB DIMMs * 2 NUMA nodes @ 2666MHz
BIOS	U30 rev. 1.36 (02/15/2018)
NIC	Two MCX623106AN-CDAT ConnectX-6 Dx EN adapter cards; 100GbE; Dual-port QSFP56; PCIe 4.0/3.0 x16;
Operating System	Red Hat Enterprise Linux Server release 7.7 (Maipo)
Kernel Version	3.10.0-1062.el7.x86_64
GCC version	4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
Mellanox NIC firmware version	22.28.1002
Mellanox OFED driver version	MLNX_OFED_LINUX-5.1-0.6.6.0
DPDK version	20.08
Test Configuration	2 NICs; 1 port used on each. Each port receives a stream of 8192 UDP flows from the IXIA Each port has 1 queue assigned, a total of two queues for two ports, and both queues are assigned to the same single logical core.

The Device Under Test (DUT) is made up of the HPE server and two Mellanox ConnectX-6 Dx 100GbE NICs utilizing one port each. The DUT is connected to the IXIA packet generator which generates traffic towards the first port of both ConnectX-6 Dx 100GbE NICs.

The ConnectX-6 Dx 100GbE data traffic is passed through DPDK to the test application **testpmd** and is redirected to the opposite direction on the opposing NIC's port. IXIA measures throughput and packet loss.

Figure 19: Test #10 Setup – Two Mellanox ConnectX-6 Dx 100GbE connected to IXIA



8.1 Test Settings

Table 31: Test #10 Settings

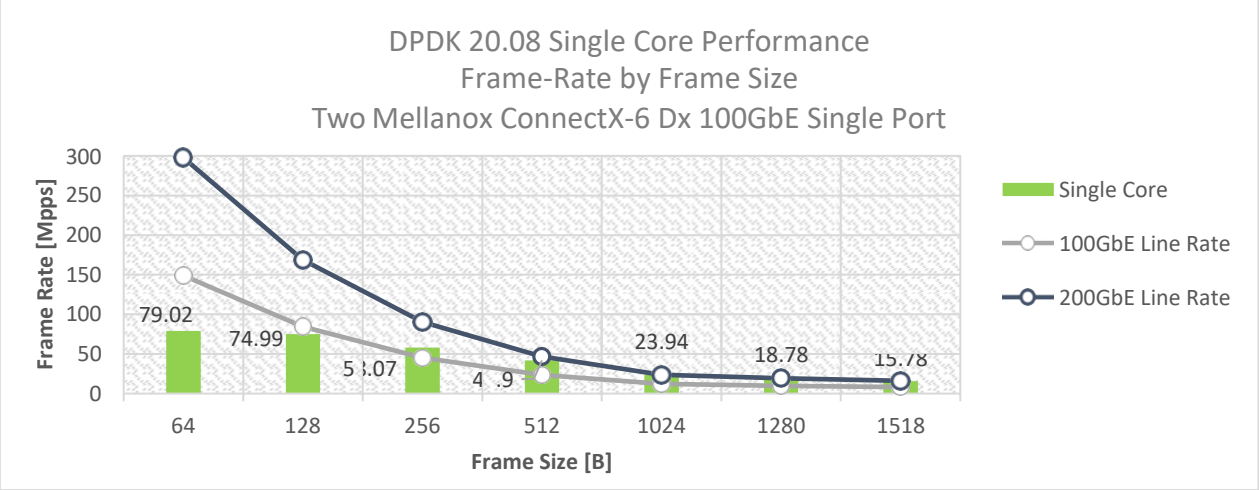
Item	Description
BIOS	1) Workload Profile = Low Latency” 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) See Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search DL380 gen10 low latency”
BOOT Settings	isolcpus=24-47 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable nohz_full=24-47 rcu_nocbs=24-47 rcu_nocb_poll default_hugepagesz=1G hugepagesz=1G hugepages=64 audit=0 nosoftlockup
DPDK Settings	Enable mlx5 PMD before compiling DPDK: In .config file generated by "make config", set: "CONFIG_RTE_LIBRTE_MLX5_PMD=y" set: "CONFIG_RTE_TEST_PMD_RECORD_CORE_CYCLES=y" During testing, testpmd was given real-time scheduling priority.
Command Line	. /build/app/testpmd -c 0x110000000000 -n 4 -w 86:00.0,decap_en=0 -w af:00.0,decap_en=0 --socket-mem=0,8192 ---- port-numa-config=0,1,1,1 --socket-num=1 --burst=64 --txd=1024 --rxd=1024 --mbcache=512 --rxq=1 --txq=1 --nb-cores=1 -i -a --rss-udp --no-numa --disable-crc-strip
Other optimizations	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMAnode: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Change PCI MaxReadReq to 1024B for each port of each NIC: Run "setpci -s $PORT_PCI_ADDRESS 68.w", it will return 4 digits ABCD -->Run "setpci -s $PORT_PCI_ADDRESS 68.w=3BCD" f) Set CQE COMPRESSION to AGGRESSIVE”: mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1 g) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us

8.2 Test Results

Table 32: Test #10 Results – Mellanox ConnectX-6 Dx 100GbE Single Core Performance

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [200G] (Mpps)	Line Rate [100G] (Mpps)	Throughput (Gbps)	CPU Cycles per packet NOTE: Lower is Better
64	78.73	297.62	148.81	40.459	33
128	74.99	168.92	84.46	76.789	33
256	58.07	90.58	45.29	118.923	30
512	41.9	46.99	23.50	171.623	31
1024	23.94	23.95	11.97	196.131	33
1280	18.78	19.23	9.62	192.342	32
1518	15.78	16.25	8.13	191.638	34

Figure 20: Test #10 Results – Mellanox ConnectX-6Dx 100GbE Single Core Performance



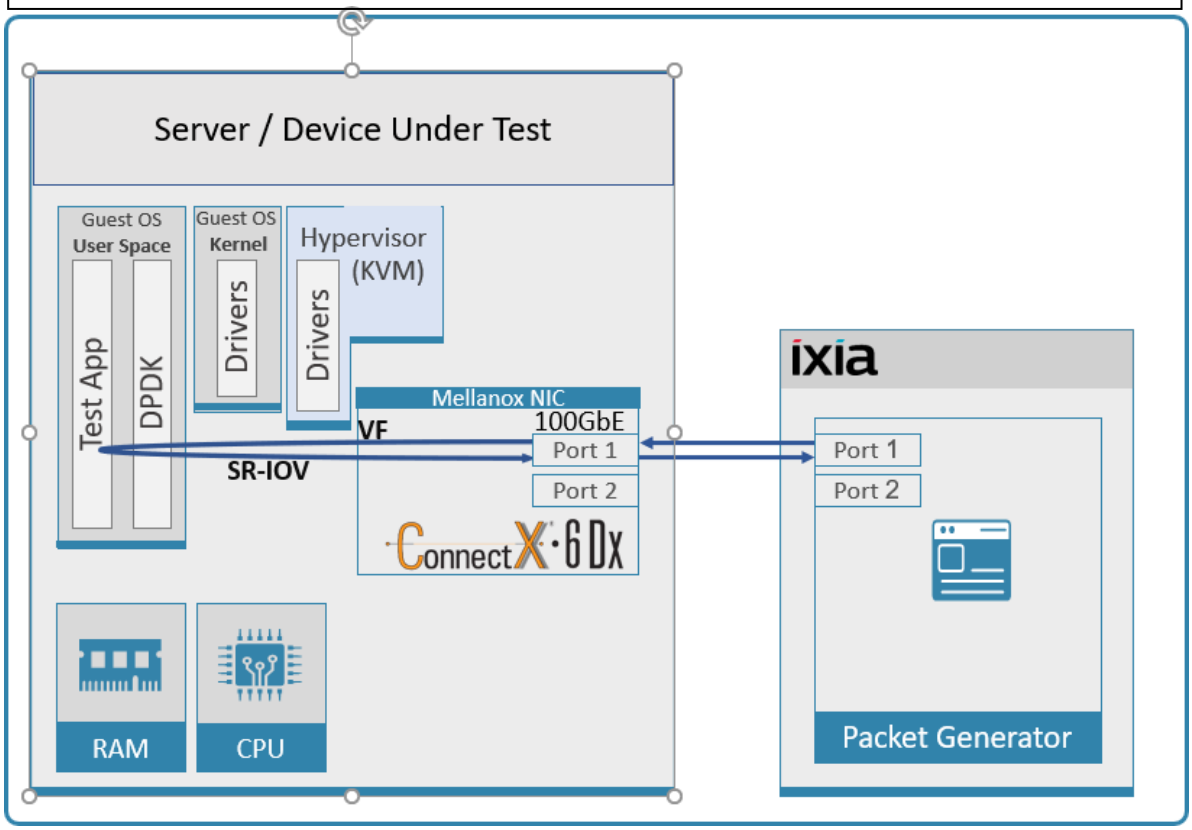
9 Test#7 Mellanox ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss (1x 100GbE) using SR-IOV over KVM Hypervisor

Table 36 - Test #12 Setup

Item	Description
Test #12	Mellanox ConnectX-6 Dx 100GbE Throughput at zero packet loss using SR-IOV over KVM Hypervisor
Server	HPE ProLiant DL380 Gen10
CPU	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz 24 CPU cores * 2 NUMA nodes
RAM	384GB: 6 * 32GB DIMMs * 2 NUMA nodes @ 2666MHz
BIOS	U30 rev. 1.36 (02/15/2018)
NIC	One MCX623106AN-CDAT ConnectX-6 Dx EN adapter card; 100GbE; Dual-port QSFP56; PCIe 4.0/3.0 x16;
Hypervisor	Red Hat Enterprise Linux Server release 7.7 (Maipo) QEMU emulator version 1.5.3 (qemu-kvm-1.5.3-156.el7)
Hypervisor Kernel Version	3.10.0-1062.el7.x86_64
Hypervisor Mellanox Driver	MLNX_OFED_LINUX-5.1-0.6.6.0
Guest Operating System	Red Hat Enterprise Linux Server release 7.7 (Maipo)
Guest Kernel Version	3.10.0-1062.el7.x86_64
Guest GCC version	4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
Guest Mellanox OFED driver version	MLNX_OFED_LINUX-5.1-0.6.6.0
Mellanox NIC firmware version	22.27.2008
DPDK version	20.08
Test Configuration	1 NIC, 1 port over 1 VF (SR-IOV); VF has 12 queues assigned to it, 1 queue per logical core for a total of 12 logical cores. Each physical port receives a stream of 8192 IP flows from the IXIA directed to VF assigned to Guest OS.

The Device Under Test (DUT) is made up of the HPE server and the Mellanox ConnectX-6 Dx NIC with a dual- port (only first port used in this test) running Red Hat Enterprise Linux Server with qemu-KVM managed via libvirt, Guest OS running DPDK is based on Red Hat Enterprise Linux Server as well. The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-6 Dx NIC. The ConnectX-6 Dx data traffic is passed through a virtual function (VF/SR-IOV) to DPDK running on the Guest OS, to the test application **l3fwd** and is redirected to the opposite direction on the same port. IXIA measures throughput and packet loss.

Figure 23 - Test #12 Setup – Mellanox ConnectX-6 Dx 100GbE connected to IXIA using KVM SR-IOV



9.1 Test Settings

Table 37 - Test #12 Settings

Item	Description
BIOS	1) Workload Profile = Low Latency"; 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) 3) Change "Workload Profile" to "Custom" 4) Change VT-x, VT-d and SR-IOV from "Disabled" to "Enabled". See Configuring and tuning HPE ProLiant Servers for low-latency applications": hpe.com > Search DL380 gen10 low latency"
Hypervisor BOOT Settings	isolcpus=24-47 intel_idle.max_cstate=0 processor.max_cstate=0 nohz_full=24-47 rcu_nocbs=24-47 intel_pstate=disable default_hugepagesz=1G hugepagesz=1G hugepages=70 audit=0 nosoftlockup intel_iommu=on iommu=pt rcu_nocb_poll

Hypervisor settings	1) Enable SRIOV via NIC configuration tool: (requires installation of mft-tools)
	mlxconfig -d /dev/mst/mt4121_pciconf1 set NUM_OF_VFS=1 SRIOV_EN=1 CQE_COMPRESSION=1
	echo 1 > /sys/class/net/***ens5f0***/device/sriov_numvfs
	
	2) Assign VF
	HCA_netintf=***ens5f0*** #assign a VF to the DUT device
	VF_PCI_address="0000:af:00.2" #VF PCI address
	echo $VF_PCI_address > /sys/bus/pci/drivers/mlx5_core/unbind
	modprobe vfio-pci
	echo "$(cat /sys/bus/pci/devices/$VF_PCI_address/vendor) $(cat /sys/bus/pci/devices/$VF_PCI_address/device)" > /sys/bus/pci/drivers/vfio-pci/new_id
	# Now the VF may be assigned to Guest (passthrough) with libvirt virt-manager.
	
	3) Setting VF MAC - use the command below (find out the vf-index from "ip link show"), iplink set <<PF NIC interface>> <vf index> mac <MAC Address> : (mac is random)
	ip link set $HCA_netintf vf 0 mac 00:52:11:22:33:42
	
	4) VM tuning: vcpupin and memory backing from hugepages:
	To persistently configure vcpu pinning and memory backing, add the below config to the VM's XML config before starting the VM. Add the following two elements to the XML: <cputune> and <memoryBacking> and also increase the number of cpus and memory: virsh edit <vmID> (to get vmID use - virsh list --all)
	Example xml configuration: (change "nodeset" and "cpuset" attributes to suit the local NUMA node in your setup)
	<domain type='kvm' id='1'>
	<name>perf-dpdk-01-005-RH-7.4</name>
	<uuid>06f283fc-fd76-4411-8b6a-72fe94f50376</uuid>
	<memory unit='KiB'>33554432</memory>
	<currentMemory unit='KiB'>33554432</currentMemory>
	<memoryBacking>
	<hugepages>
	<page size='1048576' unit='KiB' nodeset='0'/>
	</hugepages>
	<nosharepages/>

Item	Description
	<locked/> </memoryBacking> <vcpu placement='static'>23</vcpu> <cputune> <vcpupin vcpu='0' cpuset='24'/> <vcpupin vcpu='1' cpuset='25'/> <vcpupin vcpu='2' cpuset='26'/> <vcpupin vcpu='3' cpuset='27'/> <vcpupin vcpu='4' cpuset='28'/> <vcpupin vcpu='5' cpuset='29'/> <vcpupin vcpu='6' cpuset='30'/> <vcpupin vcpu='7' cpuset='31'/> <vcpupin vcpu='8' cpuset='32'/> <vcpupin vcpu='9' cpuset='33'/> <vcpupin vcpu='10' cpuset='34'/> <vcpupin vcpu='11' cpuset='35'/> <vcpupin vcpu='12' cpuset='36'/> <vcpupin vcpu='13' cpuset='37'/> <vcpupin vcpu='14' cpuset='38'/> <vcpupin vcpu='15' cpuset='39'/> <vcpupin vcpu='16' cpuset='40'/> <vcpupin vcpu='17' cpuset='41'/> <vcpupin vcpu='18' cpuset='42'/> <vcpupin vcpu='19' cpuset='43'/> <vcpupin vcpu='20' cpuset='44'/> <vcpupin vcpu='21' cpuset='45'/> <vcpupin vcpu='22' cpuset='46'/> </cputune>
Other optimizations on Hypervisor	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Change PCI MaxReadReq to 1024B for each port of each NIC: Run "setpci -s $PORT_PCI_ADDRESS 68.w", it will return 4 digits ABCD --> Run "setpci -s $PORT_PCI_ADDRESS 68.w=3BCD" f) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us
Guest BOOT Settings	isolcpus=0-22 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable idle=poll nohz_full=0-22 rcu_nocbs=0-22 rcu_nocb_poll default_hugepagesz=1G hugepagesz=1G hugepages=16 nosoftlockup
Other optimizations on Guest OS	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us

Item	Description
DPDK Settings on Guest OS	Enable mlx5 PMD before compiling DPDK: In .config file generated by "make config", set: "CONFIG_RTE_LIBRTE_MLX5_PMD=y" During testing, l3fwd was given real-time scheduling priority.
L3fwd settings on Guest OS	Updated values /l3fwd/l3fwd.h: #define RTE_TEST_RX_DESC_DEFAULT 2048 #define RTE_TEST_TX_DESC_DEFAULT 2048 #define MAX_PKT_BURST 64
Command Line on Guest OS	./examples/l3fwd/build/app/l3fwd -c 0x3ffc00 -n 4 -w 00:07.0,mprq_en=1,rxqs_min_mprq=1,mprq_log_stride_num=8 --socket-mem=8192 -- -p 0x1 -P -- config='(0,0,21),(0,1,20),(0,2,19),(0,3,18),(0,4,17),(0,5,16),(0,6,15),(0,7,14),(0,8,13),(0,9,12),(0,10,11),(0,11,10)' --eth-dest=0,00:52:11:22:33:10

9.2 Test Results

Table 38 - Test #12 Results – Mellanox ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss using KVM SR-IOV

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [100G] (Mpps)	% Line Rate
64	148.66	148.81	99.91
128	84.46	84.46	100
256	45.29	45.29	100
512	23.50	23.50	100
1024	11.97	11.97	100
1280	9.62	9.62	100
1518	8.13	8.13	100

Figure 24 - Test #12 Results – Mellanox ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss using KVM SR-IOV