NVIDIA NICs Performance Report with DPDK 23.11

Rev 1.0

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

List of Figures

List of Tables

Document History

Table 1: Document History

Version	Date	Description of Change
1.0	15-Jan-2024	Initial report release

1 About this Report

The purpose of this document is to provide packet rate performance data for NVIDIA® Network Interface Cards (NICs - ConnectX®-6 Lx, ConnectX®-6 Dx, ConnectX®-7) and Data Processing Unit (BlueField-2, BlueField-3) that has been 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 References

NVIDIA MLX5 Ethernet Driver documentation - https://doc.dpdk.org/guides/nics/mlx5.html

1.3 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
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

Note: NVIDIA acquired Mellanox Technologies in 2020, resulting in the transition of former Mellanox trademarks, such as BlueField and ConnectX, to become NVIDIA's trademarks. Please be aware that certain elements within the DPDK documentation, code, pictures, and titles in this document may still refer to Mellanox as part of the product name. Kindly note that any mention of Mellanox NIC, DPU, or OFED in this document are now associated with products offered exclusively by NVIDIA.

2 Test Description

2.1 Hardware Components

The following hardware components are used in the test setup:

- One of the following servers:
  - HPE® ProLiant DL380 Gen10 Server
  - HPE® ProLiant DL380 Gen10 Plus Server
  - HPE® ProLiant DL380 Gen11 Server
  - AMD Corporation®: QUARTZ Server
- One of the followings NICs, SmartNICs or DPUs:
  - NVIDIA ConnectX-6 Lx, ConnectX-6 Dx, ConnectX-7 Network Interface Cards (NICs), BlueField-2 Data Processing Unit (DPU), and BlueField-3 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 9.20EA) 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 9.20EA) is used with the IXIA packet generator.

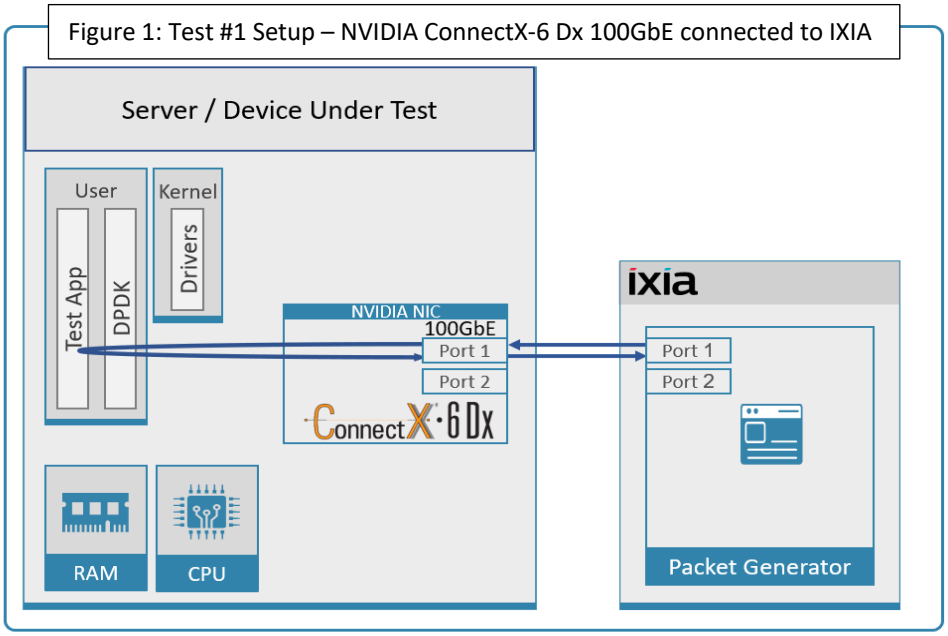
3 Test#1 NVIDIA ConnectX-6Dx 100GbE PCIe Gen4 Throughput at Zero Packet Loss (1x 100GbE)

Table 3:Test #1 Setup

Item	Description
Test #1	NVIDIA ConnectX-6 Dx 100GbE Dual-Port PCIe Gen 4 Throughput at zero packet loss
Server	HPE ProLiant DL380 Gen10 Plus
CPU	Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz 40 CPU cores * 2 NUMA nodes
RAM	512GB: 32 * 16GB DIMMs * 2 NUMA nodes @ 3200MHz
BIOS	BIOS Revision: 1.42
NIC	One MCX623106AN-CDAT ConnectX-6 Dx EN adapter card; 100GbE; Dual-port QSFP56; PCIe 4.0/3.0 x16;
Operating System	Ubuntu 20.04.2 LTS (Focal Fossa)
Kernel Version	5.4.0-164-generic.x86_64
GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC firmware version	22.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
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 NVIDIA 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 1: Test #1 Setup – NVIDIA ConnectX-6 Dx 100GbE connected to IXIA



3.1 Test Settings

Table 4: Test #1 Settings

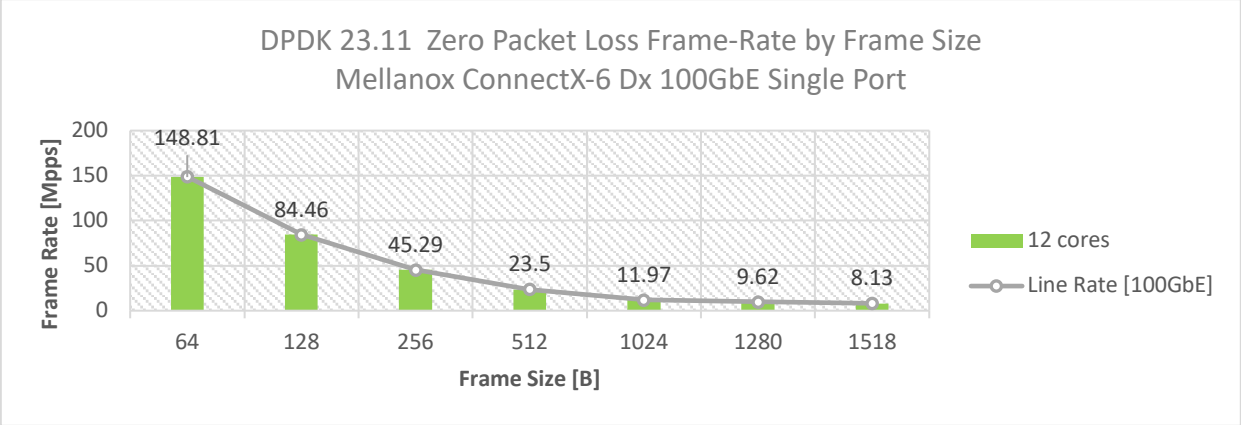
Item	Description
BIOS	Select Workload Profile = “Low Latency”; See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 plus low latency”
BOOT Settings	ro isolcpus=0-39 nohz_full=0-39 rcu_nocbs=0-39 intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G hugepages=80 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable rcu_nocb_poll audit=0
DPDK Settings	Compile DPDK using: meson <build> -Dexamples=l3fwd ; ninja -C <build> 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	./build/examples/dpdk-l3fwd -c 0xfff000000000 -n 4 -a 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

3.2 Test Results

Table 5: Test #1 Results – NVIDIA 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 2: Test #1 Results – NVIDIA ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss



4 Test#2 NVIDIA ConnectX-6Dx 100GbE PCIe Gen4 Single Core Performance (2x 100GbE)

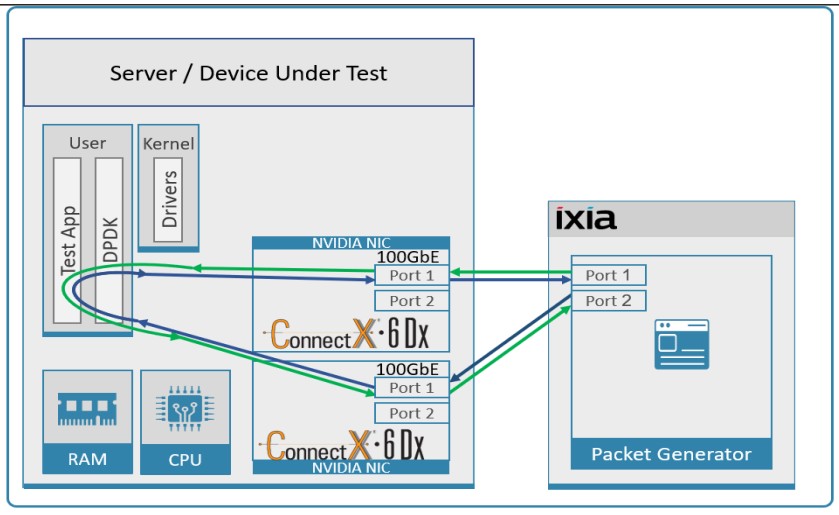
Table 6: Test #2 Setup

Item	Description
Test #2	NVIDIA ConnectX-6Dx 100GbE PCI Gen4 Single Core Performance
Server	HPE ProLiant DL380 Gen10 Plus
CPU	Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz 40 CPU cores * 2 NUMA nodes
RAM	512GB: 32 * 16GB DIMMs * 2 NUMA nodes @ 3200MHz
BIOS	BIOS Revision: 1.42
NIC	Two MCX623106AN-CDAT ConnectX-6 Dx EN adapter cards; 100GbE; Dual-port QSFP56; PCIe 4.0/3.0 x16;
Operating System	Ubuntu 20.04.2 LTS (Focal Fossa)
Kernel Version	5.4.0-164-generic.x86_64
GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC firmware version	22.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
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 NVIDIA 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 3: Test #2 Setup – Two NVIDIA ConnectX-6 Dx 100GbE connected to IXIA



4.1 Test Settings

Table 7: Test #2 Settings

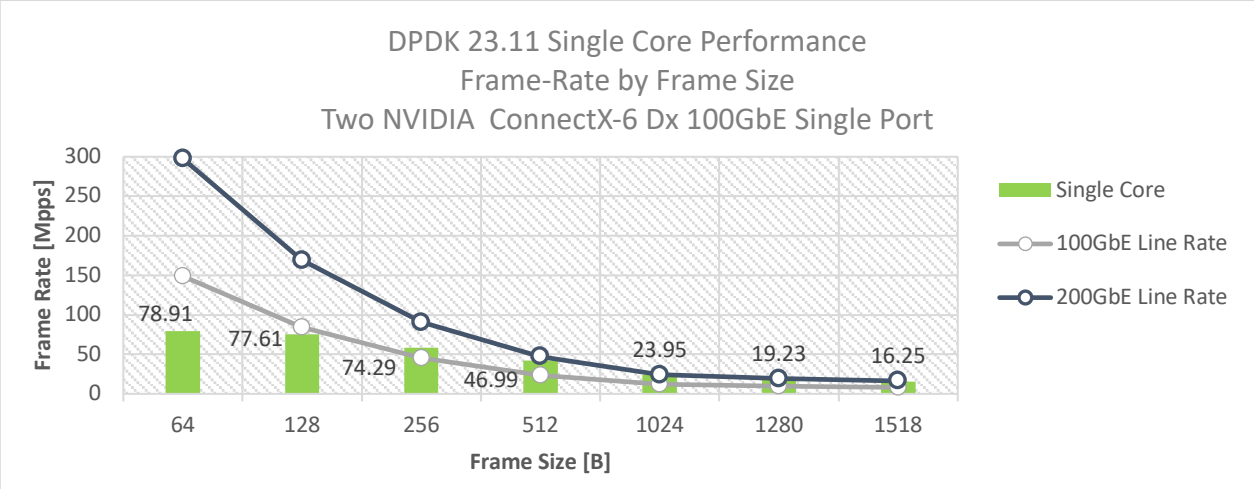
Item	Description
BIOS	Select Workload Profile = “Low Latency”; See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 plus low latency”
BOOT Settings	ro isolcpus=0-39 nohz_full=0-39 rcu_nocbs=0-39 intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G hugepages=80 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable rcu_nocb_poll audit=0
DPDK Settings	Compile DPDK using: meson <build> ; ninja -C <build> During testing, testpmd was given real-time scheduling priority.
Command Line	./build/app/dpdk-testpmd -c 0xc000000000 -n 4 -a 0000:2b:00.1 -a 0000:0f:00.1 --socket-mem=8192,0 -- --port-numa-config=0,0,1,0 --socket-num=0 --burst=64 --txd=1024 --rxd=1024 --mbcache=512 --rxq=1 --txq=1 --nb-cores=1 -i -a --rss-udp --record-core-cycles --record-burst-stats
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

4.2 Test Results

Table 8: Test #2 Results – NVIDIA 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	78.91	297.62	148.81	40.407	25
128	77.61	168.92	84.46	79.475	25
256	74.29	90.58	45.29	152.150	25
512	46.99	46.99	23.50	192.474	23
1024	23.95	23.95	11.97	196.162	23
1280	19.23	19.23	9.62	196.916	24
1518	16.25	16.25	8.13	197.393	24

Figure 4: Test #2 Results – NVIDIA ConnectX-6Dx 100GbE Single Core Performance



5 Test#3 NVIDIA ConnectX-6 Dx 100GbE PCIe Gen4 Throughput at Zero Packet Loss (2x 100GbE)

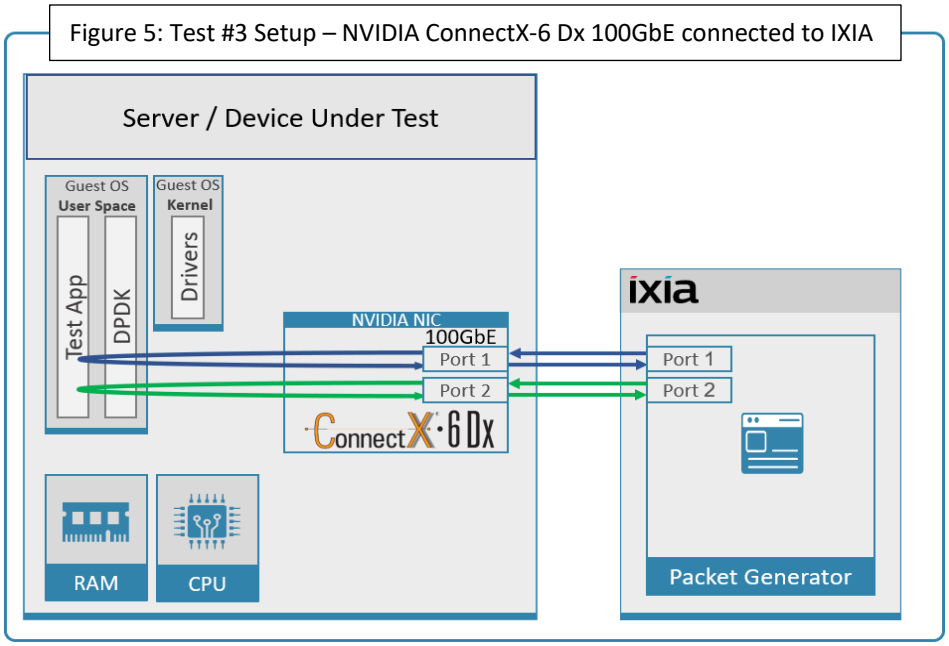
Table 9: Test #3 Setup

Item	Description
Test #3	NVIDIA ConnectX-6 Dx 100GbE Dual-Port PCIe Gen 4 Throughput at zero packet loss
Server	HPE ProLiant DL380 Gen10 Plus
CPU	Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz 40 CPU cores * 2 NUMA nodes
RAM	512GB: 32 * 16GB DIMMs * 2 NUMA nodes @ 3200MHz
BIOS	BIOS Revision: 1.42
NIC	Two MCX623106AN-CDAT ConnectX-6 Dx EN adapter card; 100GbE; Dual-port QSFP56; PCIe 4.0 x16 ; No Crypto
Operating System	Ubuntu 20.04.2 LTS (Focal Fossa)
Kernel Version	5.4.0-164-generic.x86_64
GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC firmware version	22.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
Test Configuration	1 NIC, 2 port used on NIC; each port has 8 queues assigned to it, 1 queue per logical core for a total of 16 logical cores. Each port receives a stream of 8192 IP flows from the IXIA

The Device Under Test (DUT) is made up of the Dell server and the NVIDIA ConnectX-6 Dx Dual-Port NIC (both ports are used in this test). The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-6 Dx NIC ports.

The ConnectX-6 Dx data traffic is passed via PCIe Gen 4 bus 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 5: Test #3 Setup – NVIDIA ConnectX-6 Dx 100GbE connected to IXIA



5.1 Test Settings

Table 10: Test #3 Settings

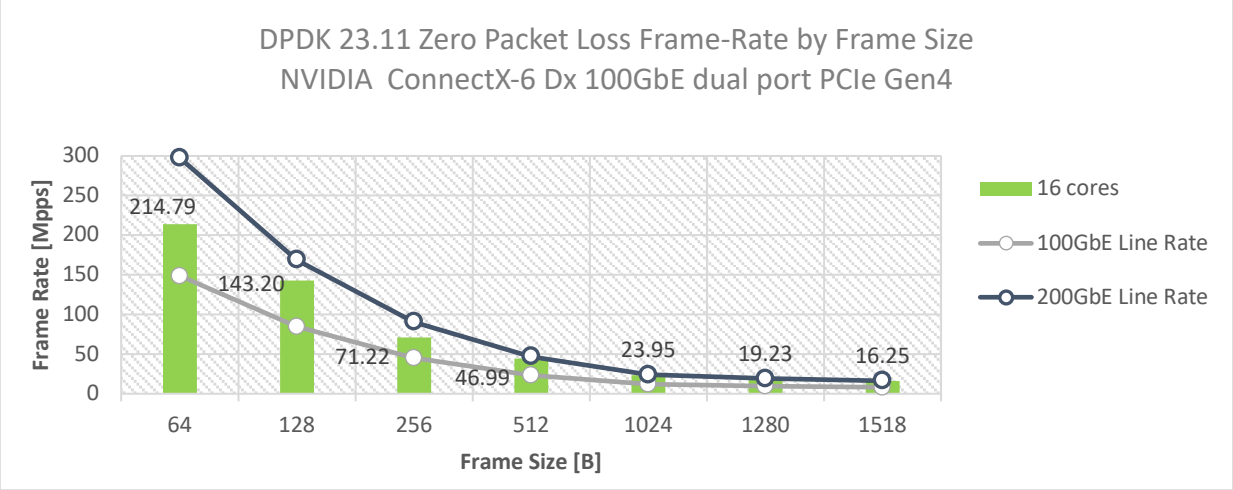
Item	Description
BIOS	Select Workload Profile = “Low Latency”; See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 plus low latency”
BOOT Settings	ro isolcpus=0-39 nohz_full=0-39 rcu_nocbs=0-39 intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G hugepages=80 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable rcu_nocb_poll audit=0
DPDK Settings	Compile DPDK using: meson <build> -Dexamples=l3fwd && ninja -C <build> 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	/build/examples/dpdk-l3fwd -c 0xffff0000000000000000 -n 4 --socket-mem=0,4096 -a 0000:84:00.0,mprq_en=1,rxqs_min_mprq=1,mprq_log_stride_num=9,txq_inline_mpw=128,rxq_pkt_pad_en=1 -a 0000:84:00.1,mprq_en=1,rxqs_min_mprq=1,mprq_log_stride_num=9,txq_inline_mpw=128,rxq_pkt_pad_en=1 -- -p 0x3 -P --config='(0,0,79),(0,1,78),(0,2,77),(0,3,76),(0,4,75),(0,5,74),(0,6,73),(0,7,72),(1,0,71),(1,1,70),(1,2,69),(1,3,68),(1,4,67),(1,5,66),(1,6,65),(1,7,64)' --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 $netdev rx off tx off" (for both ports) b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Change PCI MaxReadReq to 4096B 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= 5BCD " f) Set CQE COMPRESSION to “AGGRESSIVE”: mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1 g) Set PCI write ordering: mlxconfig -d $PORT_PCI_ADDRESS set PCI_WR_ORDERING=1 h) Disable Linux real-time throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us i) Disable auto neg for both ports: ethtool -s $PORT_PCI_ADDRESS autoneg off speed 100000

5.2 Test Results

Table 11: Test #3 Results – NVIDIA ConnectX-6 Dx 100GbE Dual-Port PCIe Gen4 Zero Packet Loss Throughput

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [200G] (Mpps)	Line Rate [100G] (Mpps)	% Line Rate
64	214.79	297.62	148.81	72.17
128	143.20	168.92	84.46	84.75
256	71.22	90.58	45.29	78.64
512	46.99	46.99	23.50	100.00
1024	23.95	23.95	11.97	100.00
1280	19.23	19.23	9.62	100.00
1518	16.25	16.25	8.13	100.00

Figure 6: Test #3 Results – NVIDIA ConnectX-6 Dx 100GbE Dual-Port PCIe Gen4 Throughput at Zero Packet Loss



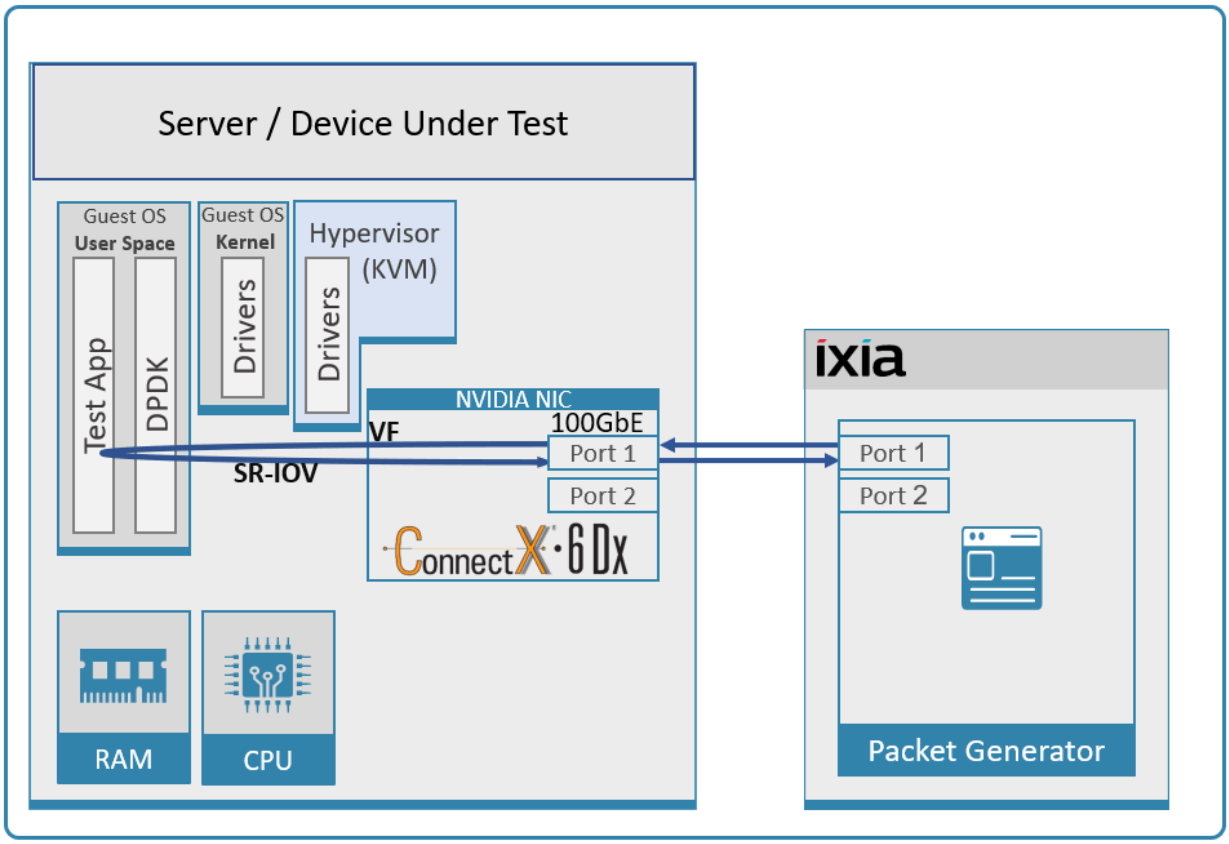
6 Test#4 NVIDIA ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss (1x 100GbE) using SR-IOV over KVM Hypervisor

Table 12: Test #4 Setup

Item	Description
Test #4	NVIDIA 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 8380 CPU @ 2.30GHz 40 CPU cores * 2 NUMA nodes
RAM	512GB: 32 * 16GB DIMMs * 2 NUMA nodes @ 3200MHz
BIOS	BIOS Revision: 1.42
NIC	One MCX623106AN-CDAT ConnectX-6 Dx EN adapter card; 100GbE; Dual-port QSFP56; PCIe 4.0/3.0 x16;
Hypervisor	Ubuntu 20.04.2 LTS (Focal Fossa)
Hypervisor Kernel Version	5.4.0-164-generic.x86_64
Hypervisor Mellanox Driver	MLNX_OFED_LINUX- 23.07-0.5.0.0
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- 23.07-0.5.0.0
Mellanox NIC firmware version	22.38.1002
DPDK version	23.11
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 NVIDIA 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 7 - Test #4 Setup – NVIDIA ConnectX-6 Dx 100GbE connected to IXIA using KVM SR-IOV



6.1 Test Settings

Table 13: Test #4 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"), ip link 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

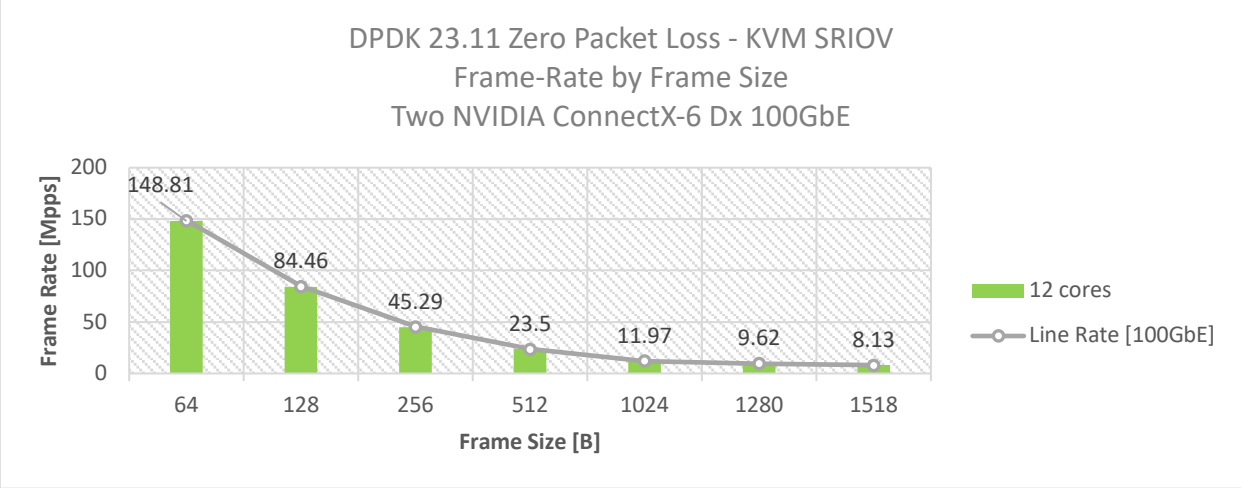
Item	Description
DPDK Settings on Guest OS	Compile DPDK using: meson <build> -Dexamples=l3fwd ; ninja -C <build> During testing, l3fwd was given real-time scheduling priority.
L3fwd settings on Guest OS	Updated values /l3fwd/l3fwd.h: #define RTE_TEST_RX_DESC_DEFAULT 2048 #define RTE_TEST_TX_DESC_DEFAULT 2048 #define MAX_PKT_BURST 64
Command Line on Guest OS	./build/examples/dpdk-l3fwd -c 0x3ffc00 -n 4 -a 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

6.2 Test Results

Table 14: Test #4 Results – NVIDIA 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	148.81	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 8 - Test #4 Results – NVIDIA ConnectX-6 Dx 100GbE Throughput at Zero Packet Loss using KVM SR-IOV



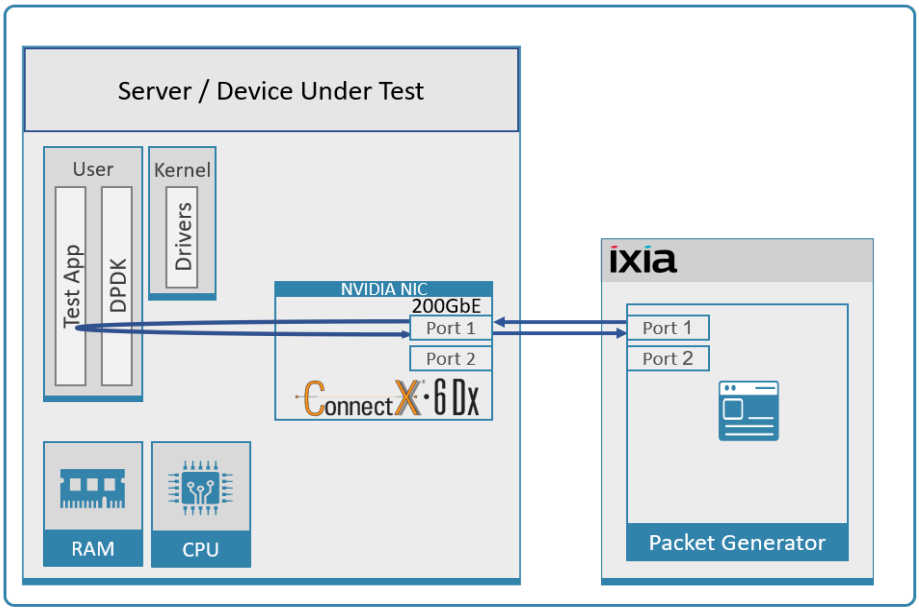
7 Test#5 NVIDIA ConnectX-6 Dx 200GbE PCIe Gen4 Throughput at Zero Packet Loss (1x 200GbE)

Table 15: Test #5 Setup

Item	Description
Test #5	NVIDIA ConnectX-6 Dx 200GbE single-port PCIe Gen4 throughput at zero packet loss
Server	HPE ProLiant DL380 Gen10 Plus
CPU	Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz 40 CPU cores * 2 NUMA nodes
RAM	512GB: 16 * 32GB DIMMs @ 3200MHz
BIOS	BIOS Revision: 1.42
NIC	One MCX623105AN-VDAT ConnectX-6 Dx EN adapter card, 200GbE, Single-port QSFP56, PCIe 4.0 x16, No Crypto
Operating System	Ubuntu 20.04.2 LTS (Focal Fossa)
Kernel Version	5.4.0-164-generic.x86_64
GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC firmware version	22.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
Test Configuration	1 NIC, 1 port used on NIC; Port has 16 queues assigned to it, 1 queue per logical core for a total of 16 logical cores. Each port receives a stream of 8192 IP flows from the IXIA

The Device Under Test (DUT) is made up of the Dell server and the NVIDIA ConnectX-6 Dx Single-Port NIC . The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-6 Dx NIC port.
The ConnectX-6 Dx data traffic is passed via PCIe Gen 4 bus 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 9 - Test #5 Setup – NVIDIA ConnectX-6 Dx 200GbE connected to IXIA



7.1 Test Settings

Table 16: Test #5 Settings

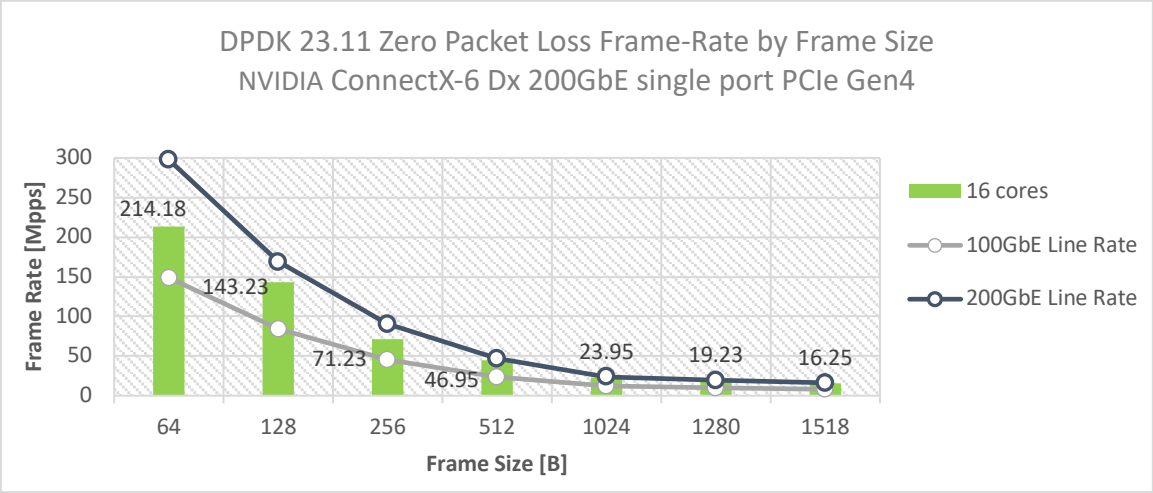
Item	Description
BIOS	Select Workload Profile = “Low Latency”; See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 plus low latency”
BOOT Settings	ro isolcpus=40-79 nohz_full=40-79 rcu_nocbs=40-79 intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G hugepages=80 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable rcu_nocb_poll audit=0
DPDK Settings	Compile DPDK using: meson <build> ; ninja -C <build> 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	/build/examples//dpdk-l3fwd -c 0xffff0000000000000000 -n 4 --socket-mem=0,4096 -a 0000:a2:00.0,mprq_en=1,rxqs_min_mprq=1,mprq_log_stride_num=9,txq_inline_mpw=128,rxq_pkt_pad_en=1 -- -p 0x1 -P -- config='(0,0,79),(0,1,78),(0,2,77),(0,3,76),(0,4,75),(0,5,74),(0,6,73),(0,7,72),(0,8,71),(0,9,70),(0,10,69),(0,11,68),(0,12,67),(0,13,66),(0,14,65),(0,15,64)' --eth-dest=0,00:52:11:22:33:10
Other optimizations	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" (for both ports) b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Change PCI MaxReadReq to 4096B 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= 5BCD " f) Set CQE COMPRESSION to “AGGRESSIVE”: mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1 g) Set PCI write ordering: mlxconfig -d $PORT_PCI_ADDRESS set PCI_WR_ORDERING=1 h) Disable Linux real-time throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us i) Disable auto neg for both ports: ethtool -s $PORT_PCI_ADDRESS autoneg off speed 200000

7.2 Test Results

Table 17: Test #5 Results – NVIDIA ConnectX-6 Dx 200GbE single port PCIe Gen4 Throughput at Zero Packet Loss

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [200G] (Mpps)	Line Rate [100G] (Mpps)	% Line Rate
64	214.18	297.62	148.81	71.96
128	143.23	168.92	84.46	84.79
256	71.23	90.58	45.29	78.64
512	46.95	46.99	23.50	99.9
1024	23.95	23.95	11.97	100
1280	19.23	19.23	9.62	100
1518	16.25	16.25	8.13	100

Figure 10 - Test #5 Results - NVIDIA ConnectX-6 Dx 200GbE single port PCIe Gen4 Throughput at Zero Packet Loss



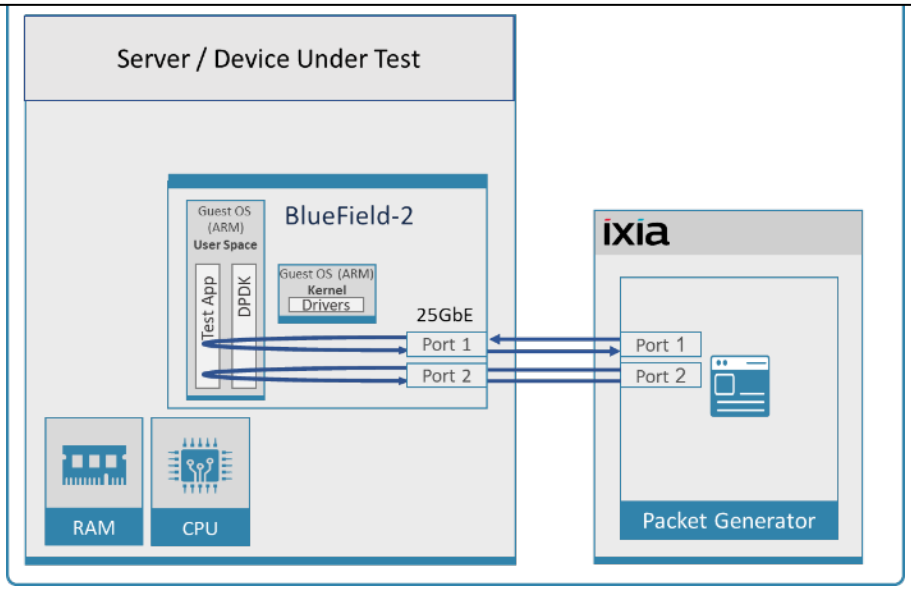
8 Test#6 NVIDIA BlueField-2 25GbE Throughput at Zero Packet Loss (2x 25GbE)

Table 18: Test #6 Setup

Item	Description
Test #6	NVIDIA BlueField-2 25GbE Dual-Port Throughput at zero packet loss
Server	HPE ProLiant DL380 Gen10
Data Processing Unit (DPU)	One MBF2H332A-AEEOT_A1 BlueField-2 P-Series SmartNIC; 25GbE; Dual-port SFP56; PCIe Gen3/4 x8
DPU hosted CPUs	BlueField-2 A1 A72 @2.5GHz , 8 Cores-Processor
DPU RAM	DDR On-board Memory 16GB
DPU BIOS	U30 rev. 1.36 (02/15/2018)
Operating System	DOCA_2.2.0_BSP_4.2.0_Ubuntu_20.04-2.23-07
DPU Kernel Version	5.4.0-1049-bluefield, aarch64
DPU GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC/DPU firmware version	24.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
Test Configuration	1 NIC/DPU, 2 ports; Each port receives a stream of 7500 UDP flows from the IXIA 1 queue assigned per logical core with a total of 2,4 and 8 logical cores

The Device Under Test (DUT) is made up of the HPE server and one NVIDIA BlueField-2 25GbE DPU utilizing two ports. It is connected to the IXIA packet generator which generates traffic towards both ports of the BlueField-2 25GbE DPU.
NVIDIA BlueField-2 25GbE data traffic is passed through DPDK to the test application **testpmd** that is running on the ARM cores (**embedded in the DPU**) and is redirected to the opposite direction using the second port. IXIA measures throughput and packet loss. The test measured the results while using 1,2,4,6 or 7 ARM cores.

Figure 11 -Test #6 Setup – NVIDIA BlueField-2 25GbE Dual-Port connected to IXIA



8.1 Test Settings

Table 19: Test #6 Settings

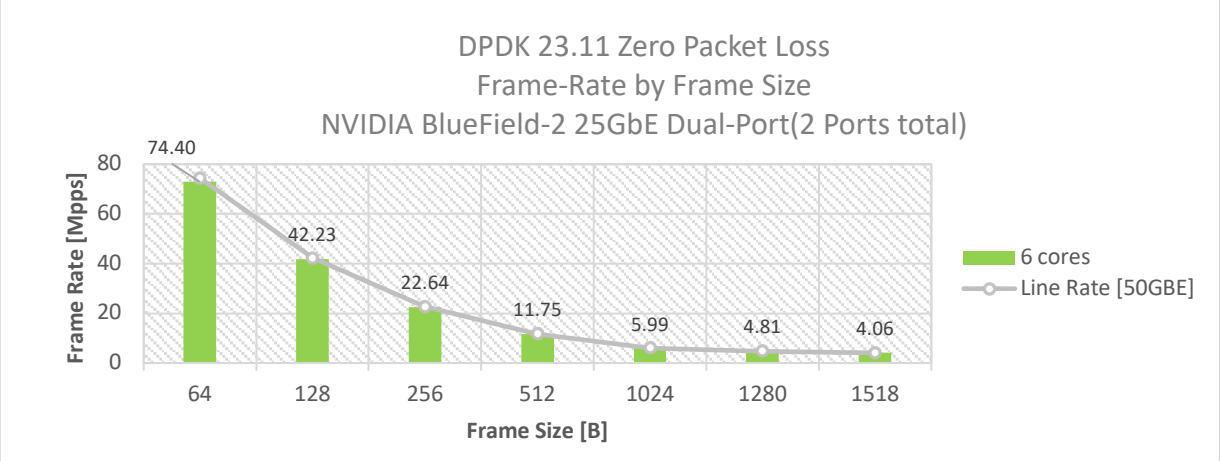
Item	Description
BIOS	1) Workload Profile = “Low Latency”; 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 low latency”
DPU BOOT Settings	ro crashkernel=auto console=ttyAMA1 console=hvc0 console=ttyAMA0 earlycon=pl011,0x01000000 earlycon=pl011,0x01800000 modprobe.blacklist=mlx5_core,mlx5_ib isolcpus=1-7 nohz_full=1-7 rcu_nocbs=1-7
DPDK Settings	Compile DPDK using: meson <build> ; ninja -C <build>
Command Lines	1 Core: /build/app/dpdk-testpmd -c 0x5 --master-lcore=0 -n 4 -w 03:00.0 -w 03:00.1 --socket-mem=1024 -- --burst=64 --txq=1 --rxq=1 --rxd=1024 --txd=1024 --mbcache=512 --nb-cores=1 -i -a --rss-udp --port-topology=loop 2 Cores: /build/app/dpdk-testpmd -c 0x15 --master-lcore=0 -n 4 -w 03:00.0 -w 03:00.1 --socket-mem=1024 -- --burst=64 --txq=2 --rxq=2 --rxd=1024 --txd=1024 --mbcache=512 --nb-cores=2 -i -a --rss-udp --port-topology=loop 4 Cores: /build/app/dpdk-testpmd -c 0xab --master-lcore=0 -n 4 -w 03:00.0 -w 03:00.1 --socket-mem=1024 -- --burst=64 --txq=4 --rxq=4 --rxd=1024 --txd=1024 --mbcache=512 --nb-cores=4 -i -a --rss-udp --port-topology=loop 6 Cores: /build/app/dpdk-testpmd -c 0x7f --master-lcore=0 -n 4 -w 03:00.0 -w 03:00.1 --socket-mem=1024 -- --burst=64 --txq=6 --rxq=6 --rxd=1024 --txd=1024 --mbcache=512 --nb-cores=6 -i -a --rss-udp --port-topology=loop
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=3900" 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 20: Test #6 Results – NVIDIA BlueField-2 25GbE Dual-Port Throughput at Zero Packet Loss

Frame Size (Bytes)	Line Rate [50G] (Mpps)	Frame Rate (Mpps)				Line rate % (6 Cores)
		1 Core	2 Cores	4 Cores	6 Cores	
64	74.40	24.58	46.47	73.93	74.40	100.00
128	42.23	23.79	42.12	42.21	42.23	100.00
256	22.64	22.52	22.62	22.64	22.64	100.00
512	11.75	11.75	11.75	11.75	11.75	100.00
1024	5.99	5.99	5.99	5.99	5.99	100.00
1280	4.81	4.81	4.81	4.81	4.81	100.00
1518	4.06	4.06	4.06	4.06	4.06	100.00

Figure 12 - Test #6 Results – NVIDIA BlueField-2 25GbE Dual-Port Throughput at Zero Packet Loss



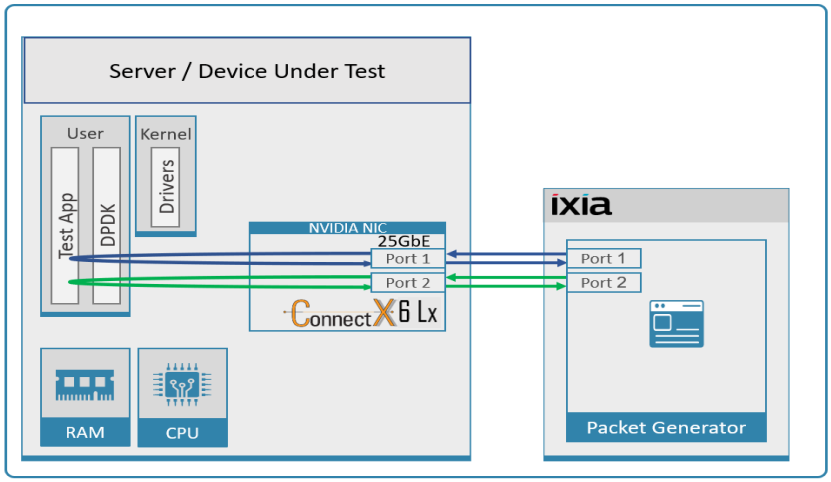
9 Test#7 NVIDIA ConnectX-6 Lx 25GbE Throughput at Zero Packet Loss (2x 25GbE)

Table 21: Test #7 Setup

Item	Description
Test #7	NVIDIA ConnectX-6 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 MCX631102AN-ADAT, ConnectX-6 Lx EN adapter card, 25GbE, Dual-port SFP28, PCIe 4.0 x8, No Crypto
Operating System	Ubuntu 20.04.2 LTS (Focal Fossa)
Kernel Version	5.4.0-164-generic.x86_64
GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC firmware version	26.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
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 NVIDIA ConnectX-6 Lx Dual-Port NIC. The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-6 Lx NIC.
The ConnectX-6 Lx 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 13 - Test #7 Setup – NVIDIA ConnectX-6 Lx 25GbE Dual-Port connected to IXIA



9.1 Test Settings

Table 22: Test #7 Settings

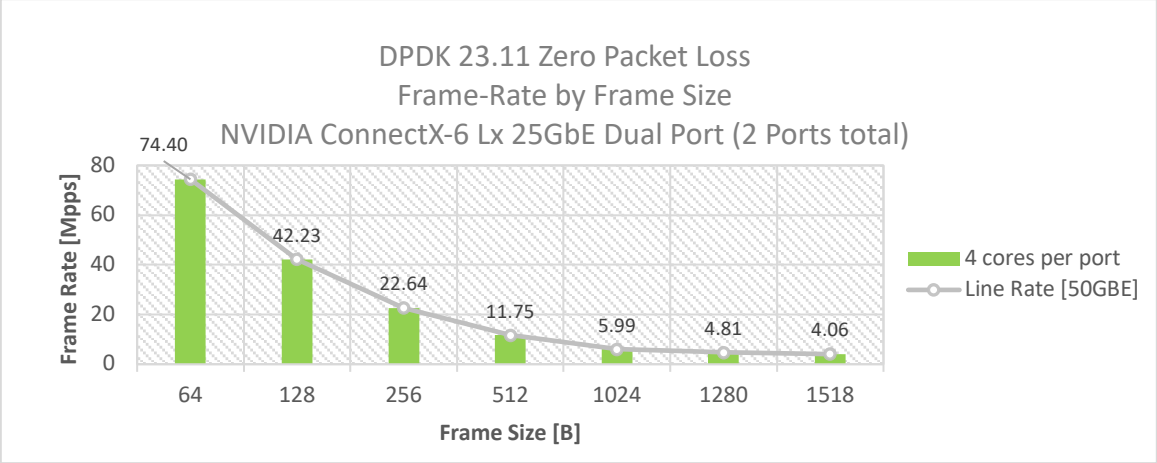
Item	Description
BIOS	1) Workload Profile = “Low Latency”; 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 low latency”
BOOT Settings	isolcpus=0-23 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable nohz_full=0-23 rcu_nocbs=0-23 rcu_nocb_poll default_hugepagesz=1G hugepagesz=1G hugepages=64 audit=0 nosoftlockup idle=poll
DPDK Settings	Compile DPDK using: meson <build> -Dexamples=l3fwd ; ninja -C <build> 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	./build/examples/dpdk-l3fwd -c 0xff0000 -n 4 -a 12:00.0,mprq_en=1,rxqs_min_mprq=1 -a 12:00.1,mprq_en=1,rxqs_min_mprq=1 --socket-mem=8192 -- -p 0x3 -P --config='(0,0,23),(0,1,22),(0,2,21),(0,3,20),(1,0,19),(1,1,18),(1,2,17),(1,3,16)' --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

9.2 Test Results

Table 23: Test #7 Results – NVIDIA ConnectX-6 Lx 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 14 - Test #7 Results – NVIDIA ConnectX-6 Lx 25GbE Dual-Port Throughput at Zero Packet Loss



10 Test#8 NVIDIA ConnectX-6 Lx 25GbE Single Core Performance (2x 25GbE)

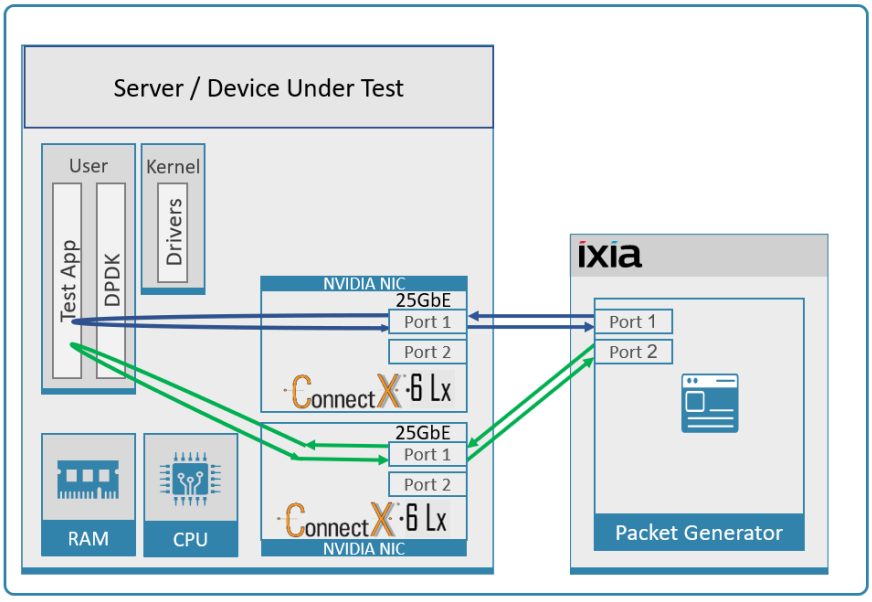
Table 24: Test #8 Setup

Item	Description
Test #8	NVIDIA ConnectX-6 Lx 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 MCX631102AN-ADAT, ConnectX-6 Lx EN adapter card, 25GbE, Dual-port SFP28, PCIe 4.0 x8, No Crypto
Operating System	Ubuntu 20.04.2 LTS (Focal Fossa)
Kernel Version	5.4.0-164-generic.x86_64
GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC firmware version	26.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
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 NVIDIA ConnectX-6 Lx 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-6 Lx 25GbE NICs.

The ConnectX-6 LX 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 15: Test #8 Setup – Two NVIDIA ConnectX-6 Lx 25GbE connected to IXIA



10.1 Test Settings

Table 25: Test #8 Settings

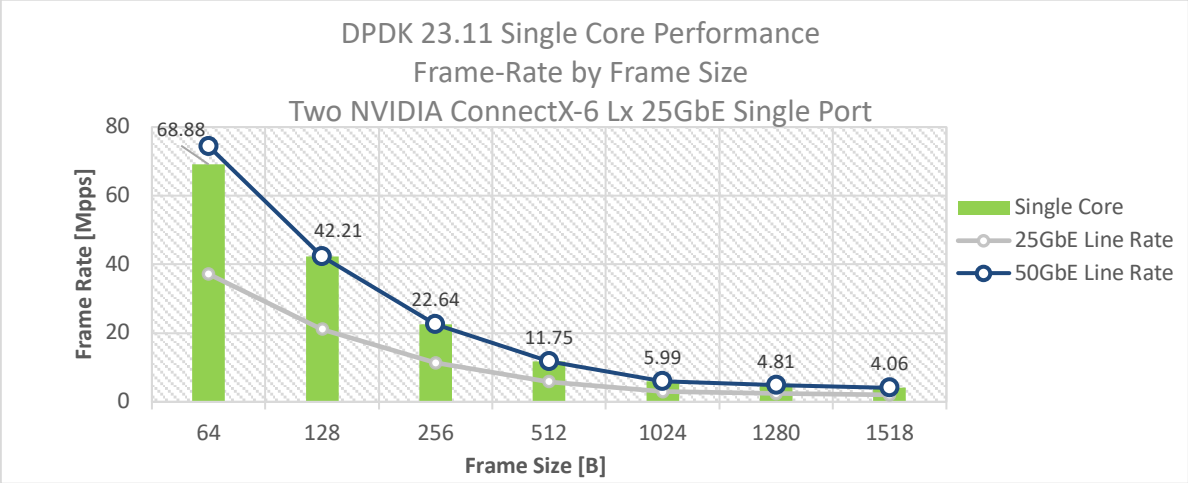
Item	Description
BIOS	1) Workload Profile = “Low Latency” 2) Jitter Control = Manual, 3400. (Setting turbo boost frequency to 3.4 GHz) See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 low latency”
BOOT Settings	isolcpus=24-47 intel_idle.max_cstate=0 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	Compile DPDK using: meson <build> ; ninja -C <build> During testing, testpmd was given real-time scheduling priority.
Command Line	./build/app/dpdk-testpmd -c 0x30000000000 -n 4 -a d8:00.0 -a 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 --disable-crc-strip --record-core-cycles --record-burst-stats
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

10.2 Test Results

Table 26: Test #8 Results – NVIDIA ConnectX-6 Lx 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	68.88	37.2	74.4	33.299	25
128	42.21	21.11	42.23	43.222	27
256	22.64	11.32	22.64	46.371	27
512	11.75	5.87	11.75	48.115	24
1024	5.99	2.99	5.99	49.037	25
1280	4.81	2.4	4.81	49.226	27
1518	4.06	2.03	4.06	49.345	28

Figure 16: Test #8 Results – NVIDIA ConnectX-6 Lx 25GbE Single Core Performance



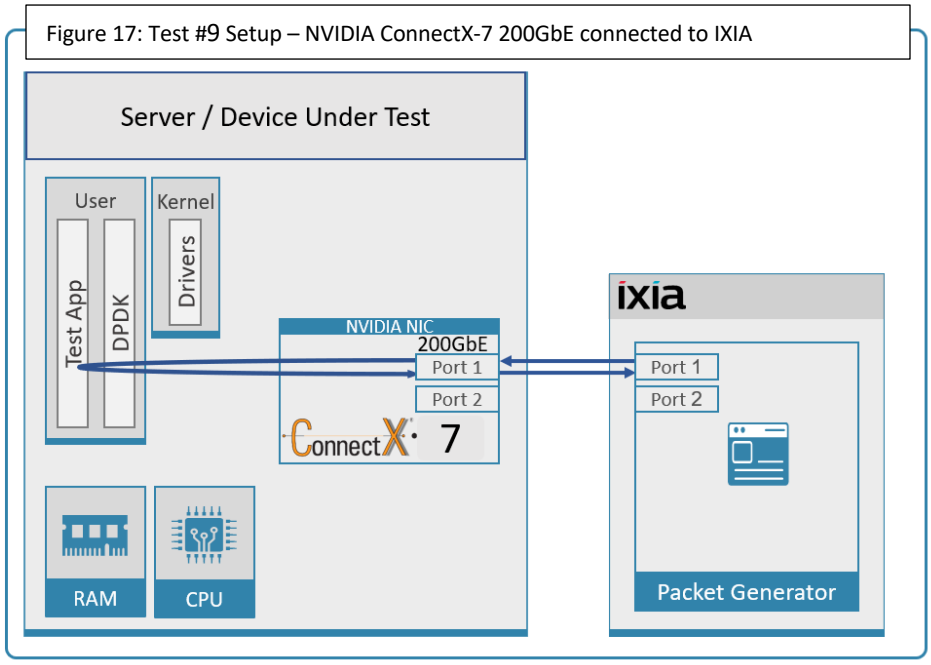
11 Test#9 NVIDIA ConnectX-7 200GbE Throughput at Zero Packet Loss (1x 200GbE)

Table 27: Test #9 Setup

Item	Description
Test #9	NVIDIA ConnectX-7 200GbE dual-port throughput at zero packet loss
Server	HPE ProLiant DL380 Gen10 Plus
CPU	Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz 40 CPU cores * 2 NUMA nodes
RAM	512GB: 16 * 32GB DIMMs @ 3200MHz
BIOS	BIOS Revision: 1.42
NIC	One MCX713106AEHEA_QP1 NVIDIA ConnectX-7 VPI adapter card, 200GbE HDR, Dual-port QSFP, PCIe 5.0 x16
Operating System	Ubuntu 20.04.2 LTS (Focal Fossa)
Kernel Version	5.4.0-164-generic.x86_64
GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC firmware version	28.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
Test Configuration	1 NIC, 1 port used on NIC; Port has 16 queues assigned to it, 1 queue per logical core for a total of 16 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 NVIDIA ConnectX-7 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-7 NIC. The ConnectX-7 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 – NVIDIA ConnectX-7 200GbE connected to IXIA



11.1 Test Settings

Table 28: Test #9 Settings

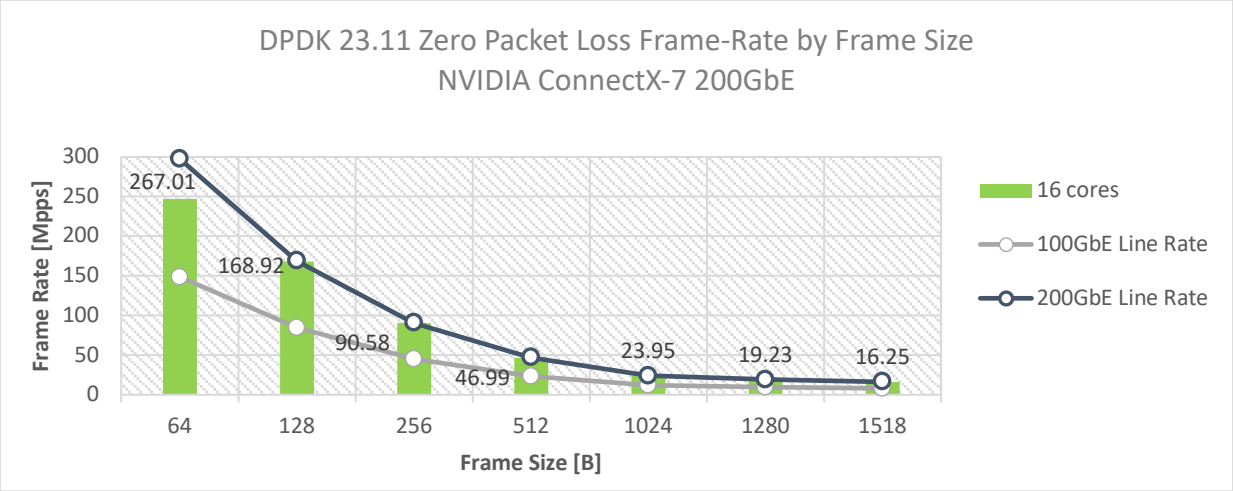
Item	Description
BIOS	Select Workload Profile = “Low Latency”; See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 plus low latency”
BOOT Settings	isolcpus=40-79 nohz_full=40-79 rcu_nocbs=40-79 intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G hugepages=80 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable rcu_nocb_poll audit=0
DPDK Settings	Compile DPDK using: meson <build> -Dexamples=l3fwd ; ninja -C <build>
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 #define NB_SOCKETS 8
Command Line	chrt -r 99 /dpdk/build/examples//dpdk-l3fwd -c 0xfff00000000000000000 -n 6 --socket-mem=0,4096 -a 0000:84:00.0,mprq_en=1,rxqs_min_mprq=1,mprq_log_stride_num=9,txq_inline_mpw=128,rxq_pkt_pad_en=1 -- -p 0x3 -P --config='(0,0,79),(0,1,78),(0,2,77),(0,3,76),(0,4,75),(0,5,74),(0,6,73),(0,7,72),(0,8,71),(0,9,70),(0,10,69),(0,11,68),(0,12,67),(0,13,66),(0,14,65),(0,15,64)' --eth-dest=0,00:52:11:22:33:10
Other optimizations	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" (for both ports) b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0" c) Move all IRQs to far NUMA node: "IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" d) Disable irqbalance: "systemctl stop irqbalance" e) Set CQE COMPRESSION to “AGGRESSIVE”: mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1 f) Remove DUT ports from DHCP Network management: “nmcli dev set $netdev managed no” (for both ports) g) Disable Linux realtime throttling: echo -1 > /proc/sys/kernel/sched_rt_runtime_us

11.2 Test Results

Table 29: Test #9 Results – NVIDIA ConnectX-7 200GbE Throughput at Zero Packet

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [200G] (Mpps)	% Line Rate
64	267.01	297.62	89.72
128	168.92	168.92	100
256	90.58	90.58	100
512	46.99	46.99	100
1024	23.94	23.95	100
1280	19.23	19.23	100
1518	16.25	16.25	100

Figure 18: Test #9 Results – NVIDIA ConnectX-7 200GbE Throughput at Zero Packet



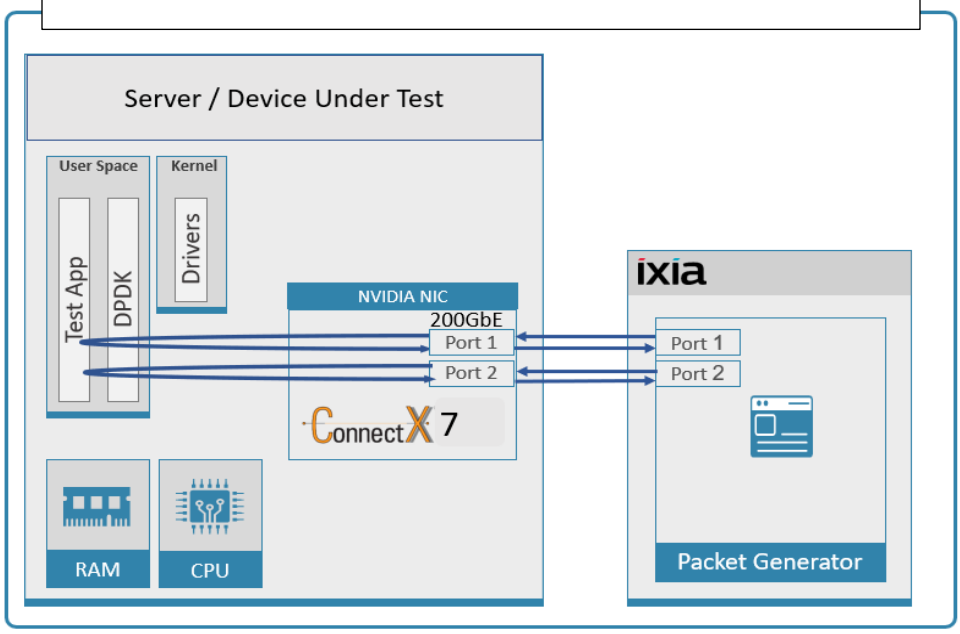
12 Test#10 NVIDIA ConnectX-7 200GbE PCIe Gen5 Throughput at Zero Packet Loss (2x 200GbE)

Table 30: Test #10 Setup

Item	Description
Test #10	NVIDIA ConnectX-7 200GbE PCIe Gen5 dual-port throughput at zero packet loss
Server	AMD Corporation: QUARTZ
CPU	AMD EPYC 9654 96-Core Processor @ 2.40GHz 96 CPU cores * 2 NUMA nodes
RAM	512GB: 32 * 32GB DIMMs @ 4800MHz
BIOS	AMD RQZ1006C, Revision: 5.25
NIC	One MCX713106AEHEA_QP1 NVIDIA ConnectX-7 VPI adapter card, 200GbE HDR, Dual-port QSFP, PCIe 5.0 x16
Operating System	Red Hat Enterprise Linux 8.5 (Ootpa)
Kernel Version	4.18.0-348.el8.x86_64
GCC version	gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-3)
Mellanox NIC firmware version	28.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
Test Configuration	1 NIC, 2 ports used on NIC; each port has 8 queues assigned to it, 1 queue per logical core for a total of 16 logical cores. Each port receives a stream of 8192 IP flows from the IXIA

The Device Under Test (DUT) is made up of the AMD server and the NVIDIA ConnectX-7 Dual-Port NIC (both NIC ports are used in this test). The DUT is connected to the IXIA packet generator which generates traffic towards the ConnectX-7 NIC. The ConnectX-7 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 19:: Test #10 Setup – NVIDIA ConnectX-7 200GbE PCIe Gen5 connected to IXIA



12.1 Test Settings

Table 31: Test #10 Settings

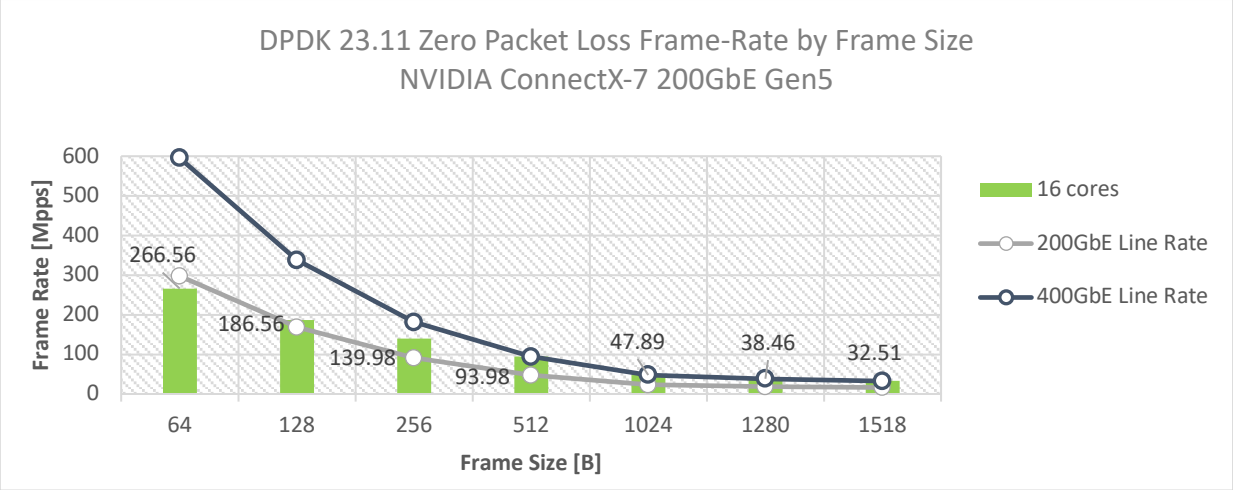
Item	Description
BIOS	Out of the Box BIOS configurations
BOOT Settings	amd_iommu=on iommu=pt pci=realloc processor.max_cstate=0 default_hugepagesz=1G hugepagesz=1G hugepages=16 isolcpus=1-16 nohz_full=1-16 rcu_nocbs=1-16
DPDK Settings	Compile DPDK using: meson <build> ; ninja -C <build>
L3FWD settings	Updated values /l3fwd/l3fwd.h: #define RTE_TEST_RX_DESC_DEFAULT 8192 #define RTE_TEST_TX_DESC_DEFAULT 8192 #define MAX_PKT_BURST 64
Command Line	./examples/dpdk-l3fwd -l 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 -n 6 --socket-mem=4096 -a 01:00.0,mprq_en=1,rxqs_min_mprq=1,mprq_log_stride_num=9,txq_inline_mpw=128,rxq_pkt_pad_en=1 -a 01:00.1,mprq_en=1,rxqs_min_mprq=1,mprq_log_stride_num=9,txq_inline_mpw=128,rxq_pkt_pad_en=1 -- -p 0x3 -P -- config='(0,0,1),(0,1,2),(0,2,3),(0,3,4),(0,4,5),(0,5,6),(0,6,7),(0,7,8),(1,0,9),(1,1,10),(1,2,11),(1,3,12),(1,4,13),(1,5,14),(1,6,15),(1,7,16)' --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 $netdev1 rx off tx off ethtool -A $netdev2 rx off tx off b) Set PCI MaxReadReq to 1024B setpci -s $Port0_PCI_address 68.w=5957 setpci -s $Port1_PCI_address 68.w=5957 c) Memory optimization : sysctl -w vm.zone_reclaim_mode=0 sysctl -w vm.swappiness=0 d) Stop irqbalance : systemctl stop irqbalance e) Disable Linux realtime throttling : echo -1 > /proc/sys/kernel/sched_rt_runtime_us f) Move all IRQs to far NUMA node: IRQBALANCE_BANNED_CPUS=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 irqbalance --oneshot g) Remove DUT ports from DHCP Network management : nmcli dev set $netdev1 managed no nmcli dev set $netdev2 managed no

12.2 Test Results

Table 32: Test #10 Results – NVIDIA ConnectX-7 200GbE PCIe Gen5 dual port Throughput at Zero Packet

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [400G] (Mpps)	% Line Rate
64	266.56	595.24	44.78
128	186.56	337.84	55.22
256	139.98	181.16	77.27
512	93.98	93.98	100
1024	47.89	47.89	100
1280	38.46	38.46	100
1518	32.51	32.51	100

Figure 20: Test #10 Results – NVIDIA ConnectX-7 200GbE PCIe Gen5 dual port Throughput at Zero Packet



13 Test#11 NVIDIA ConnectX-7 200GbE Single Core Performance (2x100GbE)

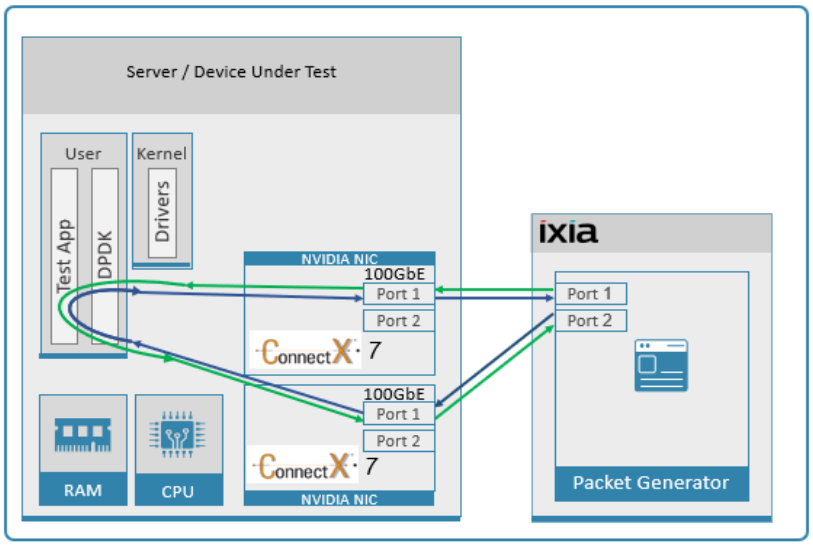
Table 33: Test #11 Setup

Item	Description
Test #11	NVIDIA ConnectX-7 200GbE Single Core Performance (2x100GbE)
Server	HPE ProLiant DL380 Gen10 Plus
CPU	Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz 40 CPU cores * 2 NUMA nodes
RAM	512GB: 32 * 16GB DIMMs * 2 NUMA nodes @ 3200MHz
BIOS	BIOS Revision: 1.42
NIC	Two MCX713106AEHEA_QP1 NVIDIA ConnectX-7 VPI adapter cards, 200GbE HDR, Dual-port QSFP, PCIe 5.0 x16
Operating System	Ubuntu 20.04.2 LTS (Focal Fossa)
Kernel Version	5.4.0-164-generic.x86_64
GCC version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Mellanox NIC firmware version	28.38.1002
Mellanox OFED driver version	MLNX_OFED_LINUX- 23.07-0.5.0.0
DPDK version	23.11
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 NVIDIA ConnectX-7 200GbE NICs utilizing one port each. The DUT is connected to the IXIA packet generator which generates traffic towards the first port of both ConnectX-7 200GbE NICs.

The ConnectX-7 200GbE 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 21: Test#11 Setup - Two NVIDIA ConnectX-7 2x100GbE connected to IXIA



13.1 Test Settings

Table 34: Test #11 Settings:

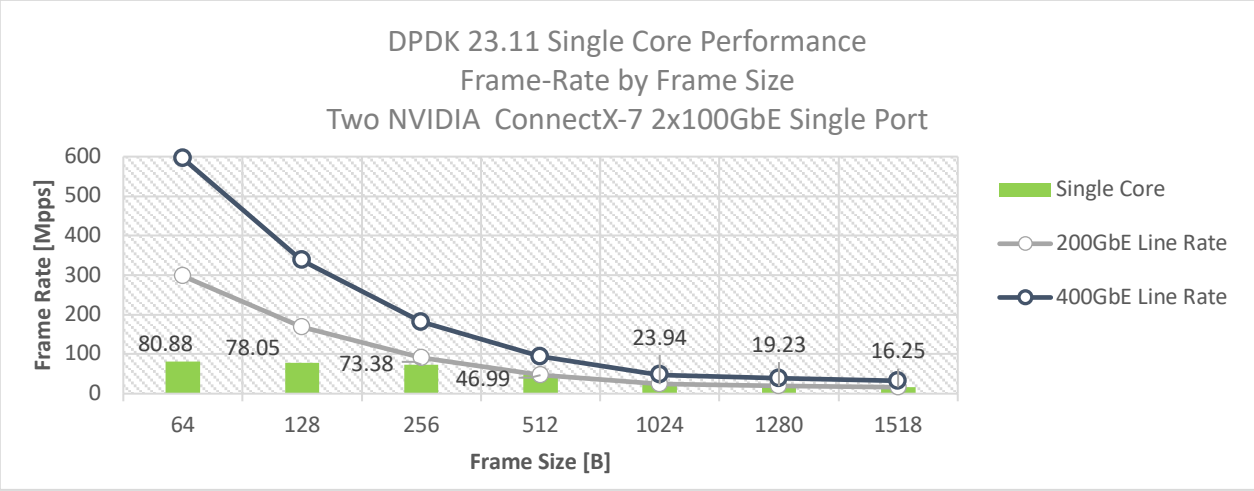
Item	Description
BIOS	Select Workload Profile = “Low Latency”; See “Configuring and tuning HPE ProLiant Servers for low-latency applications”: hpe.com > Search “DL380 gen10 plus low latency”
BOOT Settings	ro isolcpus=40-79 nohz_full=40-79 rcu_nocbs=40-79 intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G hugepages=80 intel_idle.max_cstate=0 processor.max_cstate=0 intel_pstate=disable rcu_nocb_poll audit=0
DPDK Settings	Compile DPDK using: meson <build> ; ninja -C <build> During testing, testpmd was given real-time scheduling priority.
Command Line	./build/app/dpdk-testpmd -c 0xc0000000000000000000 -n 4 -a 0000:84:00.0 -a 0000:a2:00.0 --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 --record-core-cycles --record-burst-stats
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

13.2 Test Results

Table 35: Test #11 Results – NVIDIA ConnectX-7 2x100GbE Single Core Performance

Frame Size (Bytes)	Frame Rate [2x100G] (Mpps)	Throughput [2x100G] (Gbps)	Line Rate [200G]	Line Rate %	CPU Cycles per packet [100G]
64	80.88	41.293	297.62	27.18	29
128	78.05	79.672	168.92	46.21	28
256	73.38	150.892	90.58	81.02	26
512	46.99	192.466	46.99	100	23
1024	23.94	196.155	23.95	100	23
1280	19.23	196.905	19.23	100	23
1518	16.25	197.386	16.25	100	24

Figure 22: Test #11 Results – NVIDIA ConnectX-7 2x100GbE Single Core Performance



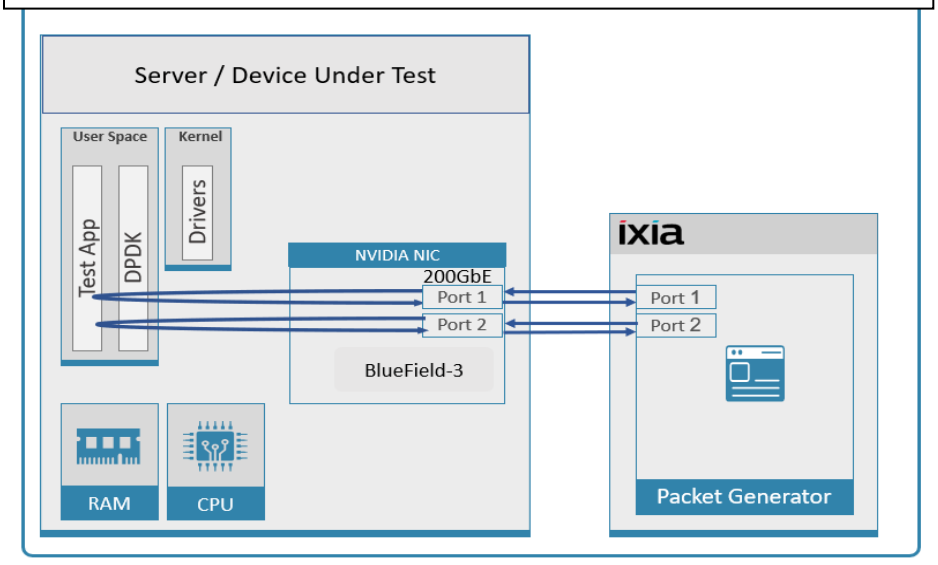
14 Test#12 NVIDIA BlueField-3 DPU 200GbE Throughput at Zero Packet Loss (2x 200GbE) – NIC Mode

Table 36: Test #12 Setup

Item	Description
Test #12	NVIDIA BlueField-3 DPU 200GbE Throughput at Zero Packet Loss (2x 200GbE) – NIC Mode
Server	AMD Corporation: QUARTZ
CPU	AMD EPYC 9654 96-Core Processor @ 2.40GHz 96 CPU cores * 2 NUMA nodes
RAM	512GB: 32 * 32GB DIMMs * 2 NUMA nodes @ 4800MHz
BIOS	AMD RQZ1006C, Revision: 5.25
Data Processing Unit (DPU)	Nvidia BlueField-3 BF3220 P-Series DPU 200GbE/NDR200 dual-port QSFP112; PCIe Gen5.0 x16 FHHL with x16 PCIe extension option; Crypto Enabled; SB Disabled 32GB on-board DDR; integrated BMC; Tall Bracket; IPN QP
Host Operating System	Red Hat Enterprise Linux 8.5 (Ootpa)
Host Kernel Version	4.18.0-348.el8.x86_64
Host GCC version	gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-3)
DPU BFB image	DOCA_2.2.1_BSP_4.2.2_Ubuntu_22.04-12.23-09
Mellanox NIC firmware version	32.38.3056
Mellanox OFED driver version	MLNX_OFED_LINUX-23.07-0.5.0.0
DPDK version	23.11
Test Configuration	1 NIC, 2 port used on NIC; each port has 8 queues assigned to it, 1 queue per logical core for a total of 16 logical cores. Each port receives a stream of 8192 IP flows from the IXIA

The Device Under Test (DUT) is made up of the AMD server and the NVIDIA BlueField-3® DPU . The DUT is connected to the IXIA packet generator which generates traffic towards the BlueField-3® DPU.
The BlueField-3® 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, the BlueField-3® DPU is set to run on DPU NIC mode.

Figure 23: Test #12 Setup – NVIDIA BlueField-3 2x200GbE connected to IXIA



14.1 Test Settings

Table 37: Test #12 Setup

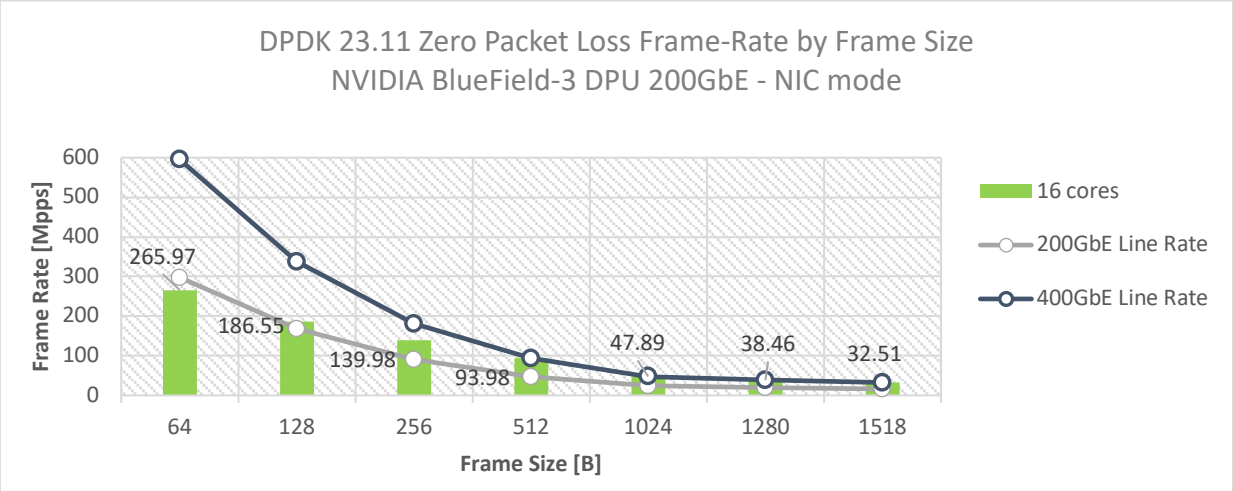
Item	Description
BIOS	Out of the Box BIOS configurations
BOOT Settings	amd_iommu=on iommu=pt pci=realloc processor.max_cstate=0 default_hugepagesz=1G hugepagesz=1G hugepages=16 isolcpus=1-16 nohz_full=1-16 rcu_nocbs=1-16
DPDK Settings	Compile DPDK using: meson <build> ; ninja -C <build>
L3FWD settings	Updated values /l3fwd/l3fwd.h: #define RTE_TEST_RX_DESC_DEFAULT 8192 #define RTE_TEST_TX_DESC_DEFAULT 8192 #define MAX_PKT_BURST 64
Command Line	./examples/dpdk-l3fwd -c 0xffff -n 6 --socket-mem=4096 -a 21:00.0,mprq_en=1,rxqs_min_mprq=1,txq_inline_mpw=128,rxq_pkt_pad_en=1,mprq_log_stride_num=9 -a 21:00.1,mprq_en=1,rxqs_min_mprq=1,txq_inline_mpw=128,rxq_pkt_pad_en=1,mprq_log_stride_num=9 -- -p 0x3 -P --config='(0,0,15),(0,1,14),(0,2,13),(0,3,12),(0,4,11),(0,5,10),(0,6,9),(0,7,8),(1,0,7),(1,1,6),(1,2,5),(1,3,4),(1,4,3),(1,5,2),(1,6,1),(1,7,0)' --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 $netdev1 rx off tx off ethtool -A $netdev2 rx off tx off b) Set PCI MaxReadReq to 1024B : setpci -s $Port0_PCI_address 68.w=5957 setpci -s $Port1_PCI_address 68.w=5957 c) Memory optimizations: sysctl -w vm.zone_reclaim_mode=0 sysctl -w vm.swappiness=0 d) Stop irqbalance : systemctl stop irqbalance e) Disable Linux realtime throttling : echo -1 > /proc/sys/kernel/sched_rt_runtime_us f) Remove DUT ports from DHCP Network management nmcli dev set $netdev1 managed no nmcli dev set $netdev2 managed no

14.2 Test Results

Table 38: Test #12 Results – NVIDIA BlueField-3 DPU 200GbE Throughput at Zero Packet Loss (2x 200GbE) – NIC Mode

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [400G] (Mpps)	% Line Rate
64	265.97	595.24	44.68
128	186.55	337.84	55.22
256	139.98	181.16	77.27
512	93.98	93.98	100
1024	47.89	47.89	100
1280	38.46	38.46	100
1518	32.51	32.51	100

Figure 24: Test #12 Results – NVIDIA BlueField-3 DPU 200GbE Throughput at Zero Packet Loss (2x 200GbE) – NIC Mode



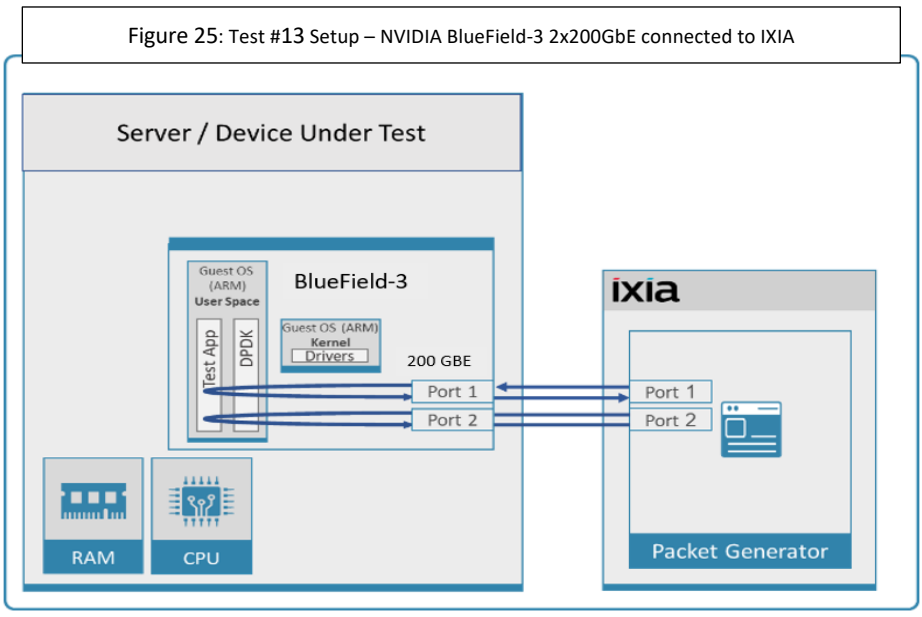
15 Test#13 NVIDIA BlueField-3 DPU 200GbE Throughput at Zero Packet Loss (2x 200GbE) – DPU Mode

Table 39: Test #13 Setup

Item	Description
Test #13	NVIDIA BlueField-3 DPU 200GbE Throughput at zero packet loss 2x 200GbE – DPU mode
Server	HPE ProLiant DL380 Gen11
Data Processing Unit (DPU)	Nvidia BlueField-3 BF3220 P-Series DPU 200GbE/NDR200 dual-port QSFP112; PCIe Gen5.0 x16 FHHL with x16 PCIe extension option; Crypto Enabled; SB Disabled 32GB on-board DDR; integrated BMC; Tall Bracket; IPN QP
DPU hosted CPUs	Cortex-A78AE, 16 Cores
DPU RAM	DDR On-board Memory 32GB
DPU BIOS	4.2.2.12958 rev 3.0
Operating System	DOCA_2.2.1_BSP_4.2.2_Ubuntu_22.04-12.23-09
DPU Kernel Version	5.15.0-1022-bluefield, aarch64
DPU GCC version	gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Mellanox NIC/DPU firmware version	32.38.3056
Mellanox OFED driver version	MLNX_OFED_LINUX-23.07-0.5.0.0
DPDK version	23.11
Test Configuration	1 NIC/DPU, 2 ports. the ports receive a stream of 8192 IP flows from the IXIA 1 queue assigned per two logical cores with a total of 12 logical cores

The Device Under Test (DUT) is made up of the HPE server and one NVIDIA BlueField-3 200GbE DPU utilizing two ports. It is connected to the IXIA packet generator which generates traffic towards the ports of the BlueField-3 200GbE DPU.
NVIDIA BlueField-3 200GbE data traffic is passed through DPDK to the test application **testpmd** that is running on the ARM cores (embedded in the DPU) and is redirected to the opposite direction using the same ports. IXIA measures throughput and packet loss. The test measured the results while using 12 ARM cores.

Figure 25: Test #13 Setup – NVIDIA BlueField-3 2x200GbE connected to IXIA



15.1 Test Settings

Table 40: Test #13 Settings

Item	Description
BOOT Settings	BOOT_IMAGE=/boot/vmlinuz-5.15.0-1022-bluefield root=UUID=a89dca85-f54c-4da0-95fb-fb6c7f54c0ad ro console=hvc0 console=ttyAMA0 earlycon=pl011,0x13010000 fixrtc net.ifnames=0 biosdevname=0 iommu.passthrough=1 isolcpus=4-15 nohz_full=4-15 console=tty1 console=ttyS0
DPDK Settings	Compile DPDK using: meson <build> ; ninja -C <build>
Command Line	/root/dpdk/build/app/dpdk-testpmd-l 3-15 --main-lcore=3 -n 4 --socket-mem=2048 -a03:00.0,txq_inline_mpw=128,mprq_en=1,rxqs_min_mprq=1,rxq_pkt_pad_en=1 -a03:00.1,txq_inline_mpw=128,mprq_en=1,rxqs_min_mprq=1,rxq_pkt_pad_en=1 ----burst=64 --mbcache=512 --txd=1024 --rxd=1024 --rxq=6 --txq=6 --nb-cores=12--forward-mode=io -i -a --total-num-mbufs=300000

Other optimizations	On ARM DPU :
	
	a) Flow Control OFF: "ethtool -A $netdev rx off tx off" (for both ports)
	b) Memory optimizations: "sysctl -w vm.zone_reclaim_mode=0"; "sysctl -w vm.swappiness=0"
	c) Disable irqbalance: "systemctl stop irqbalance"
	d) Set CQE COMPRESSION to “AGGRESSIVE”:
	mlxconfig -d $PORT_PCI_ADDRESS set CQE_COMPRESSION=1
	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 $PORT_PCI_ADDRESS 68.w=3910"
	f) Stop the below services :
	systemctl stop openvswitch; systemctl stop set_emu_param;
	/usr/share/openvswitch/scripts/ovs-ctl stop
	g) Delete subfunctions if any : mlnx-sf -a delete -i *SF name*
	h) Delete TC rules if present: tc qdisc del dev *dev* ingress
	i) Reserve hugepages: sysctl -w vm.nr_hugepages=4096;
	j) switch to legacy mode:
	for pci in 0000:03:00.0 0000:03:00.1; do
	devlink dev eswitch set pci/$pci mode legacy;
	done

15.2 Test Results

Table 41: Test #13 Results – NVIDIA BlueField-3 DPU 200GbE Zero Packet Loss Throughput – DPU mode

Frame Size (Bytes)	Frame Rate (Mpps)	Line Rate [400G] (Mpps)	% Line Rate
64	235.78	595.24	39.61
128	219.84	337.84	65.07
256	139.98	181.16	77.27
512	93.98	93.98	100
1024	47.89	47.89	100
1280	38.46	38.46	100
1518	32.51	32.51	100

Figure 26: Test #13 Results – NVIDIA BlueField-3 DPU 200GbE Throughput at Zero Packet Loss (2x 200GbE) – DPU Mode