

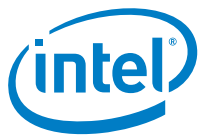
# DPDK Vhost/Virtio Performance Report

## Release 20.05

---

**Test Date:** May 20 2020

**Author:** Intel DPDK Validation Team



## Revision History

---

Date	Revision	Comment
May 20 2020	1.0	Initial document for release



# Contents

---

Audience and Purpose.....	4
Test setup: .....	4
Intel® Xeon® Processor Platinum 8180 (39.5M Cache, 2.50 GHz) .....	7
Hardware & Software Ingredients.....	7
Test Case 1 – RFC2544 zero packet loss test for Split ring Vhost/Virtio PVP Mergeable...	8
Test Case 2 – RFC2544 zero packet loss test for Packed ring Vhost/Virtio PVP Mergeable	10
Test Case 3 – DPDK VM2VM iperf performance test .....	12

## Audience and Purpose

---

The primary audience for this test report are architects and engineers implementing the Data Plane Development Kit (DPDK). This report provides information on packet processing performance testing for the specified DPDK release on Intel® architecture. The initial report may be viewed as the baseline for future releases and provides system configuration and test cases based on DPDK examples.

The purpose of reporting these tests is not to imply a single “correct” approach, but rather to provide a baseline of well-tested configurations and procedures with reproducible results. This will help guide architects and engineers who are evaluating and implementing DPDK solutions on Intel® architecture and can assist in achieving optimal system performance.

## Test setup:

---

The device under test (DUT) consists of a system with an Intel® architecture motherboard populated with the following;

- A single or dual processor and PCH chip, except for System on Chip (SoC) cases
- DRAM memory size and frequency (normally single DIMM per channel)
- Specific Intel Network Interface Cards (NICs)
- BIOS settings noting those that updated from the basic settings
- DPDK build configuration settings, and commands used for tests

Connected to the DUT is an IXIA\*, a hardware test and simulation platform to generate packet traffic to the DUT ports and determine the throughput at the tester side. The IXIA is used to implement RFC2544 on the DUT.

Benchmarking a DPDK system requires knowledge of networking technologies including knowledge of network protocols and hands-on experience with relevant open-source software, such as Linux\*, and the DPDK. Engineers also need benchmarking and debugging skills, as well as a good understanding of the device-under-test (DUT) across compute and networking domains.

**DPDK Testpmd Test Case:** Documentation may be found at [http://www.dpdk.org/doc/guides/testpmd\\_app\\_uq/index.html](http://www.dpdk.org/doc/guides/testpmd_app_uq/index.html).

The testpmd application can be used to test the DPDK in a packet forwarding mode and also to access NIC hardware features. Note in the Testpmd example if the `-i` argument is used, the first core is used for the command language interface (CLI).

**RFC2544 Zero packet loss test case:** Used to determine the DUT throughput as defined in RFC1242(<https://www.ietf.org/rfc/rfc1242.txt>). Note RFC6201 <https://www.ietf.org/rfc/rfc6201.txt> has updated RFC2544 and RFC1242. Please check the link for more details. In this report, RFC2544 test uses DPDK testpmd as test application.

Procedure: Send a specific number of frames at a specific rate through the DUT and then count the frames that are transmitted by the DUT. If the count of offered frames is not equal to the count of

---

received frames, the rate of the offered stream is reduced and the test is rerun. The throughput is the fastest rate at which the count of test frames transmitted by the DUT is equal to the number of test frames sent to it by the test equipment.

**DPDK Phy-VM-Phy(PVP) RFC2544 test case:**

This test setup is shown in Figure1. The traffic is generated by Ixia running RFC2544(IxNetwork\* 8.12 with 0 packet loss, and the duration for each round is 60 seconds). The flow is one fixed flow. In this test setup, one port(40G) of Intel® Ethernet Converged Network Adapter XL710-QDA2 is used to inject traffic to Vhost/virtio. The case is to measure vhost/virtio system forwarding throughput, and the theoretical system forwarding throughput is 40 Gbps. Both Vhost and Virtio is DPDK polling mode driver. The flow is as below: IXIA→NIC port0→Vhost-user0→Virtio→Vhost-user0→NIC port0→IXIA.

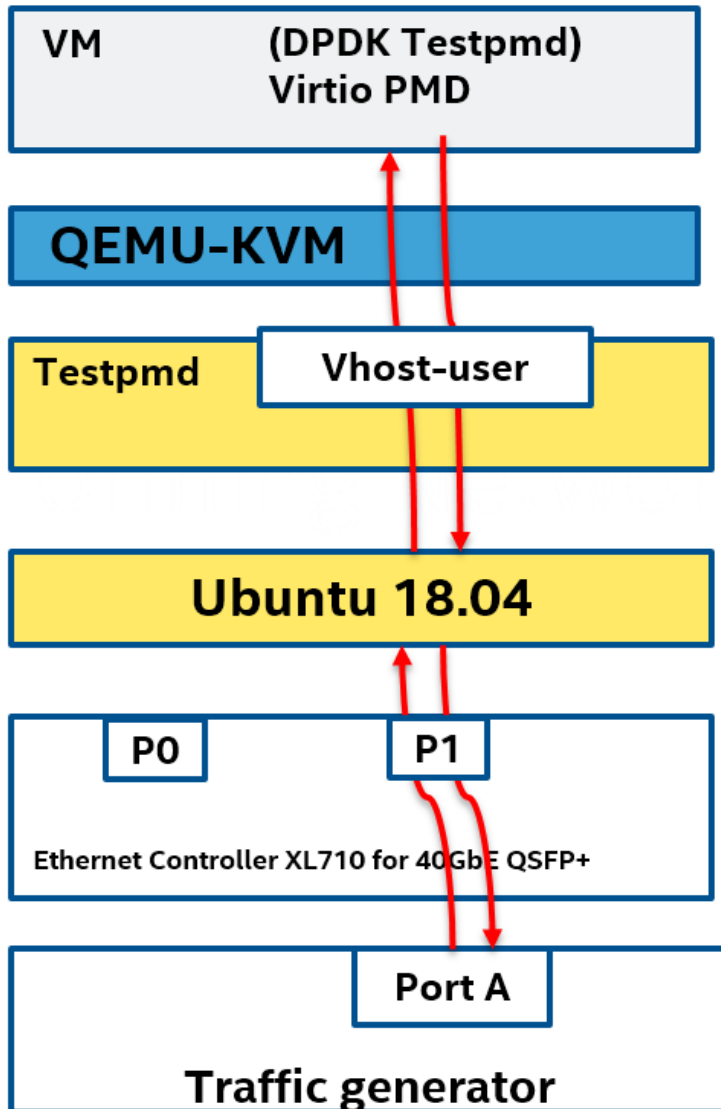


Figure1. DPDK PVP test setup

**DPDK Vhost VM to VM iperf test case:**

This test setup is as shown in Figure2. iperf is the TCP/UDP/SCTP network bandwidth measurement tool. Iperf performance test is widely used in the industry. In this case, Vhost is using DPDK polling mode driver, Virtio is using Linux kernel driver. The test case is to measure DPDK vhost PMD's capability for supporting the maximum TCP bandwidth with virtio-net device. The flow is as below: virtio-net1 → vhost-user0 → vhost-user1 → virtio-net2.

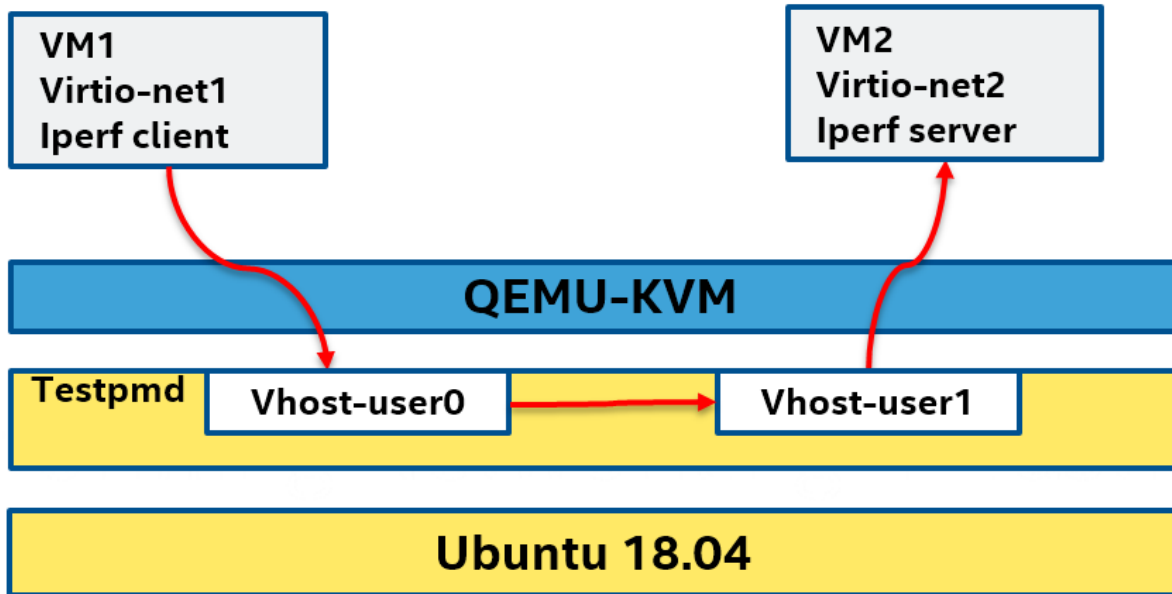


Figure2. DPDK VM2VM iperf test setup



# Intel® Xeon® Processor Platinum 8180 (39.5M Cache, 2.50 GHz)

## Hardware & Software Ingredients

Item	Description
Server Platform	Intel® Server Board S2600GZ Intel® Server Board S2600GZ Family
CPU	Intel(R) Xeon(R) Platinum 8180 (39.5M L3 Cache, 2.50 GHz) Number of cores 56, Number of threads 112.
Memory	Total 96GB over 8 channels, DDR4 @2666 Mhz
PCIe	1 x PCIe Gen3 x8
NICs	Intel® Ethernet Converged Network Adapter XL710-QDA2 (2x40G)
BIOS	SE5C620.86B.01.00.0013
Microcode version	0x2000043
Host Operating System	Ubuntu 18.04 LTS
Host Linux kernel version	4.15.0-20-generic
Host GCC version	gcc (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0
Host DPDK version	20.05
Guest Operating System	Ubuntu 16.04 LTS
Guest GCC version	gcc (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609
Guest DPDK version	20.05
Guest Linux kernel version	4.4.0-62-generic

### Boot and BIOS settings

Item	Description
Host Boot Settings	hugepagesz=1G hugepages=140 default_hugepagesz=1G isolcpus=1-60 intel_iommu=on iommu=pt nohz_full=1-60 rcu_nocbs=1-60 intel_pstate=disable numa_balancing=disable maybe-ubiquity  <b>Note: nohz_full and rcu_nocbs is to disable Linux* kernel interrupts, and it's important for zero-packet loss test. Generally, 1G huge pages are used for performance test.</b>
VM Boot Settings	hugepagesz=2M hugepages=512 isolcpus=1-2 nohz_full=1-2 rcu_nocbs=1-2
BIOS	CPU Power and Performance Policy <Performance> CPU C-state Disabled CPU P-state Disabled Enhanced Intel® Speedstep® Tech Disabled Turbo Boost Disabled
Host Real Time Settings	echo -1 > /proc/sys/kernel/sched_rt_period_us echo -1 > /proc/sys/kernel/sched_rt_runtime_us echo 10 > /proc/sys/vm/stat_interval echo 0 > /proc/sys/kernel/watchdog_thresh

VM Real Time Settings	<pre>echo 0 &gt; /proc/sys/kernel/watchdog echo 0 &gt; /proc/sys/kernel/nmi_watchdog echo -1 &gt; /proc/sys/kernel/sched_rt_period_us echo -1 &gt; /proc/sys/kernel/sched_rt_runtime_us</pre>
-----------------------	---

## Test Case 1 – RFC2544 zero packet loss test for Split ring Vhost/Virtio PVP Mergeable

Item	Description
Test Case	RFC2544 zero packet loss test for Split ring Vhost/Virtio PVP Mergeable
NIC	Intel® Ethernet Converged Network Adapter XL710-QDA2 (2x40G)
Driver	i40e DPDK PMD
Test Configuration	<p>Test tool: IxNetwork 8.40.1400.5 EA</p> <p><b>Qemu Version: 2.11.1, Qemu above 2.8 can support change Vring size using qemu command.</b></p> <p><b>Vring size: 1024</b>, the max Vring size Qemu support</p> <p>Hugepage size: 1G</p> <p>Virtio Mergeable: On</p> <p>Forward Mode: testpmd mac forward</p> <p><b>Vhost: 1 queue 1 logic core</b></p> <p><b>Virtio: 1 queue 1 logic core</b></p> <p><b>Totally 2 logic cores from 2 physical cores are used.</b></p>
Flow Configuration	1 Flow with fixed source and destination IP.
Test Step	<ol style="list-style-type: none"> <li>1. Bind one 40G NIC port to vfio-pci</li> <li>2. Launch vhost: <pre>chrt -f 95 ./x86_64-native-linuxapp-gcc/app/testpmd -l 29-30 -n 4 --file-prefix=vhost\ --vdev 'eth_vhost0,iface=vhost-net,queues=1' -- -i --txd=1024 --rx=1024 --nb-cores=1 testpmd&gt;set fwd mac testpmd&gt;start</pre> </li> <li>3. Launch VM : <pre>taskset -c 31,32,33 qemu-system-x86_64 \ -name us-vhost-vm1 -cpu host -enable-kvm -m 2048 \ -object memory-backend-file,id=mem,size=2048M,mem-path=/mnt/huge,share=on \ -numa node,memdev=mem -mem-prealloc \ -smp cores=2,sockets=1 -drive file=/home/osimg/ubuntu16.img \ -chardev socket,id=char0,path=./vhost-net \ -monitor unix:/tmp/vm2_monitor.sock,server,nowait -net nic,vlan=2,macaddr=00:00:00:08:e8:aa,addr=1f \ -net user,vlan=2,hostfwd=tcp:127.0.0.1:6002-:22 \</pre> </li> </ol>





	<pre>-netdev type=vhost-user,id=mynet1,chardev=char0,vhostforce \ -device virtio-net- pci,mac=52:54:00:00:00:01,netdev=mynet1,mrg_rxbuf=on,rx_queue_size=1 024,tx_queue_size=1024 -vnc :10 --monitor stdio</pre> <p>4. Use monitor to bind vcpu with physical cpu on host machine: qemu monitor: info cpus #check pid taskset -cp 32 xxx #xxx is the pid number taskset -cp 33 xxx</p> <p>5. Bind vdev to igb_uio and launch virtio in VM: ./x86_64-native-linuxapp-gcc/app/testpmd -c 0x3 -n 4 -- -i --txd=1024 -- rxq=1024 testpmd&gt;set fwd mac testpmd&gt;start</p>
--	--

**Test Result:**

Packet Size(Bytes)	Throughput(Mpps)	Line Rate[40G](Mpps)
64	7.61	59.52
128	6.64	33.783
256	5.41	18.116
512	3.73	9.398
1024	2.78	4.789
1280	2.48	3.846
1518	2.23	3.251

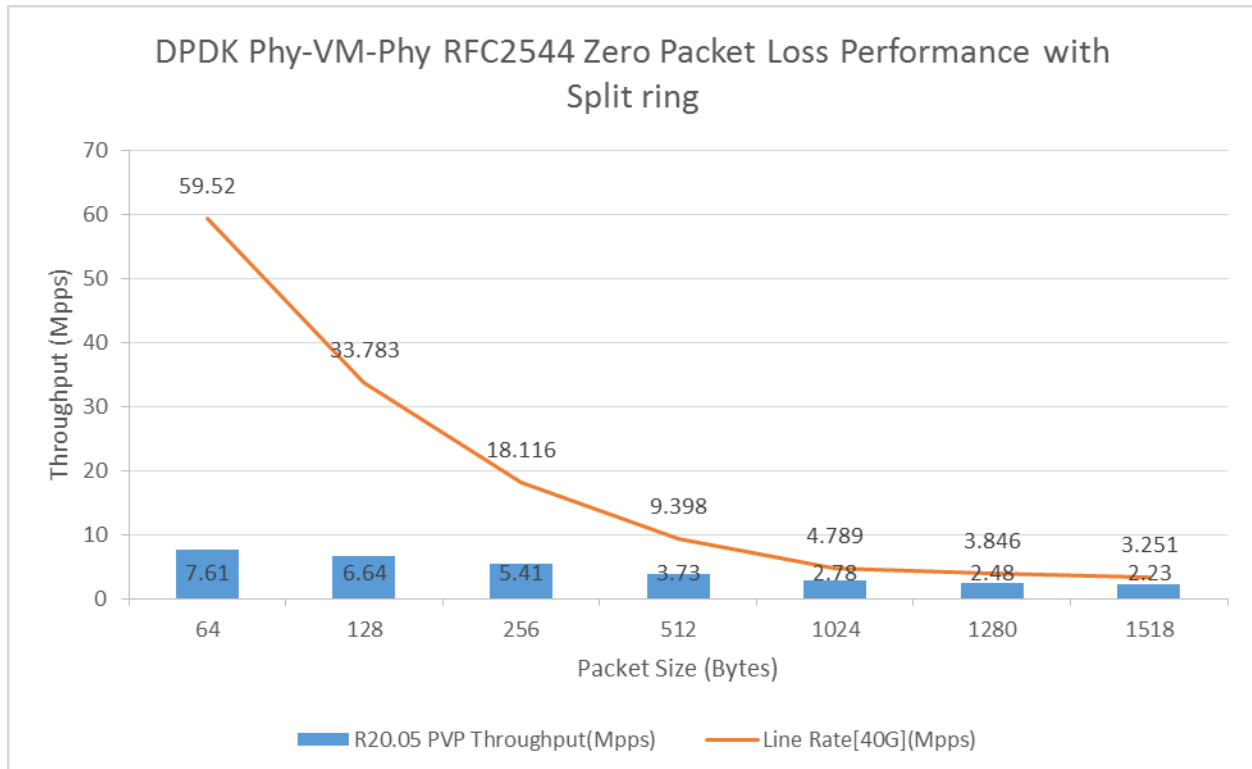


Figure3. RFC2544 zero packet loss performance with Split ring Vhost/Virtio PVP Mergeable

## Test Case 2 – RFC2544 zero packet loss test for Packed ring Vhost/Virtio PVP Mergeable

Item	Description
Test Case	RFC2544 zero packet loss test for Packed ring Vhost/Virtio PVP Mergeable
NIC	Intel® Ethernet Converged Network Adapter XL710-QDA2 (2x40G)
Driver	i40e DPDK PMD
Test Configuration	<p>Test tool: IxNetwork 8.40.1400.5 EA</p> <p><b>Qemu Version: qemu-4.2.0.</b></p> <p><b>Vring size: 1024</b>, the max Vring size Qemu support</p> <p>Hugepage size: 1G</p> <p>Virtio Mergeable: On</p> <p>Forward Mode: testpmd mac forward</p> <p><b>Vhost: 1 queue 1 logic core</b></p> <p><b>Virtio: 1 queue 1 logic core</b></p> <p><b>Totally 2 logic cores from 2 physical cores are used.</b></p>
Flow Configuration	1 Flow with fixed source and destination IP.
Test Step	<ol style="list-style-type: none"> <li>1. Bind one 40G NIC port to vfio-pci</li> <li>2. Launch vhost:  <pre>chrt -f 95 ./x86_64-native-linuxapp-gcc/app/testpmd -l 29-30 -n 4 --file-prefix=vhost\ --vdev 'eth_vhost0,iface=vhost-net,queues=1' -- -i --txd=1024 --rxd=1024 --nb-cores=1 testpmd&gt;set fwd mac testpmd&gt;start</pre> </li> <li>3. Launch VM :  <pre>taskset -c 31,32,33 qemu-system-x86_64 \ -name us-vhost-vm1 -cpu host -enable-kvm -m 2048 \ -object memory-backend-file,id=mem,size=2048M,mem-path=/mnt/huge,share=on \ -numa node,memdev=mem -mem-prealloc \ -smp cores=2,sockets=1 -drive file=/home/osimg/ubuntu16.img \ -chardev socket,id=char0,path=./vhost-net \ -monitor unix:/tmp/vm2_monitor.sock,server,nowait -device e1000,netdev=nttsip1 \ -netdev user,id=nttsip1,hostfwd=tcp:127.0.0.1:6002-:22 \ -netdev type=vhost-user,id=mynet1,chardev=char0,vhostforce \ -device virtio-net-pci,mac=52:54:00:00:00:01,netdev=mynet1,mrg_rxbuf=on,rx_queue_size=1024,tx_queue_size=1024,packed=on -vnc :10 --monitor stdio</pre> </li> <li>4. Use monitor to bind vcpu with physical cpu on host machine:  <pre>qemu monitor: info cpus #check pid</pre> </li> </ol>



	<pre>taskset -cp 32 xxx #xxx is the pid number taskset -cp 33 xxx  5. Bind vdev to igb_uio and launch virtio in VM: ./x86_64-native-linuxapp-gcc/app/testpmd -c 0x3 -n 4 -- -i --txd=1024 -- rxd=1024 testpmd&gt;set fwd mac testpmd&gt;start</pre>
--	---

Test Result:

Packet Size(Bytes)	Throughput(Mpps)	Line Rate[40G](Mpps)
64	8.77	59.52
128	8.27	33.783
256	6.57	18.116
512	4.93	9.398
1024	3.31	4.789
1280	2.99	3.846
1518	2.77	3.251

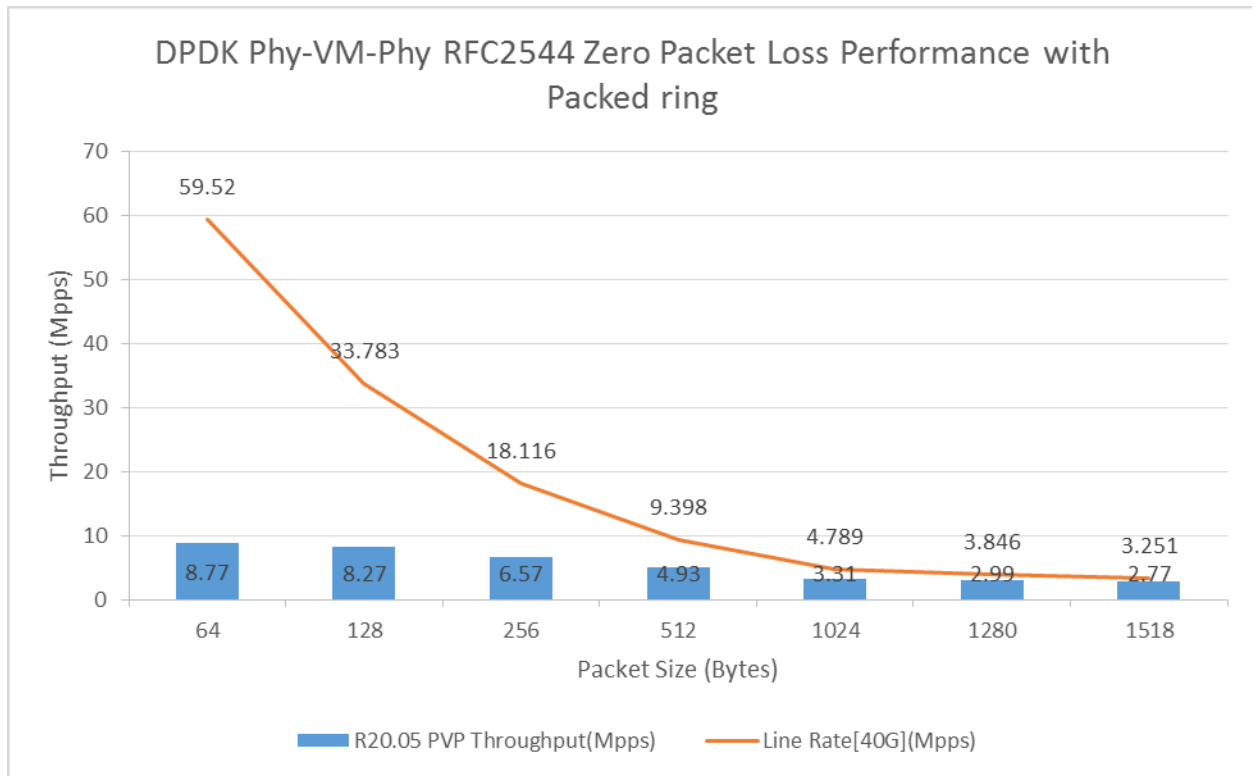


Figure4. RFC2544 zero packet loss performance with Packed ring Vhost/Virtio PVP Mergeable

## Test Case 3 – DPDK VM2VM iperf performance test

Item	Description
Test Case	virtio-net performance test for VM2VM
Nic	Intel® Ethernet Converged Network Adapter X710-DA4 (4x10G)
Driver	i40e DPDK PMD
Test configuration	<p><b>Qemu Version : 2.11.1</b>  Hugepage size : 1G  <b>dequeue-zero-copy: Enabled</b>  Iperf version: 2.0.5</p>
Core settings	<b>1 core for vhost-user, and assign each VM 1 core</b>
Test step	<ol style="list-style-type: none"> <li>1. Bind one 10G NIC port to vfio-pci.</li> <li>2. Launch vhost :  <pre>./examples/vhost/build/vhost-switch -c 0x1c00 -n 4 -- -p 0x1 --mergeable 1 --vm2vm 1 --tso 1 --tx-csum 1 --dequeue-zero-copy --socket-file ./vhost-net --socket-file ./vhost-net1</pre> </li> <li>3. Launch VM1 and run iperf server:  <pre>taskset -c 13 /home/qemu-install/qemu-2.11/bin/qemu-system-x86_64 -name us-vhost-vm1 -cpu host -enable-kvm -m 4096 -object memory-backend-file,id=mem,size=4096M,mem-path=/mnt/huge,share=on -numa node,memdev=mem -mem-prealloc -smp cores=1,sockets=1 -drive file=/home/osimg/ubuntu16.img -monitor unix:/tmp/vm2_monitor.sock,server,nowait -net nic,vlan=2,macaddr=00:00:00:08:e8:aa,addr=1f -net user,vlan=2,hostfwd=tcp:127.0.0.1:6004-:22 -chardev socket,id=char0,path=./vhost-net -netdev type=vhost-user,id=mynet1,chardev=char0,vhostforce -device virtio-net-pci,mac=52:54:00:00:00:01,netdev=mynet1,mrg_rxbuf=on,csum=on,gso=on,guest_csum=on,host_tso4=on,guest_tso4=on,guest_ecn=on \ -vnc :10 -daemonize</pre> <p>In VM1:  <pre>ifconfig ens4 1.1.1.2 arp -s 1.1.1.8 52:54:00:00:00:02 iperf -s -i 1</pre> </p> </li> <li>4. Launch VM2 and run iperf client:  <pre>taskset -c 15 /home/qemu-install/qemu-2.11/bin/qemu-system-x86_64 -name us-vhost-vm2 -cpu host -enable-kvm -m 4096 -object memory-backend-file,id=mem,size=4096M,mem-path=/mnt/huge,share=on -numa node,memdev=mem -mem-prealloc -smp cores=1,sockets=1 -drive file=/home/osimg/ubuntu16-2.img -monitor unix:/tmp/vm2_monitor.sock,server,nowait -net</pre> </li> </ol>



	<pre>nic,vlan=2,macaddr=00:00:00:08:e8:aa,addr=1f -net user,vlan=2,hostfwd=tcp:127.0.0.1:6005-:22 -chardev socket,id=char1,path=./vhost-net1 -netdev type=vhost- user,id=mynet2,chardev=char1,vhostforce -device virtio-net- pci,mac=52:54:00:00:00:02,netdev=mynet2,mrg_rxbuf=on,csum=on,gso=on, guest_csum=on,host_tso4=on,guest_tso4=on,guest_ecn=on \ -vnc :11 -daemonize  In VM2: ifconfig ens4 1.1.1.8 arp -s 1.1.1.2 52:54:00:00:00:01 iperf -c 1.1.1.2 -i 1 -t 60</pre>
--	---

**Test Result:**

Throughput with vhost dequeue zero-copy	<b>43.4Gb/s</b>
--	-----------------



## DISCLAIMERS

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit [www.intel.com/benchmarks](http://www.intel.com/benchmarks).

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Performance results are based on testing as of December 6st 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

For more information go to <http://www.intel.com/performance>

Intel® AES-NI requires a computer system with an AES-NI enabled processor, as well as non-Intel software to execute the instructions in the correct sequence. AES-NI is available on select Intel® processors. For availability, consult your reseller or system manufacturer. **For more information, see <http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni/>**

Copyright © 2020 Intel Corporation. All rights reserved.