DPDK Vhost/Virtio Performance Report Release 18.02

Test Date: March 8 2018

Author: Intel DPDK Validation Team

Revision History

Date	Revision	Comment
Mar 8st, 2018	1.0	Initial document for release

Contents

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_ug/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

* Other names and brands may be claimed as the property of others.

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.

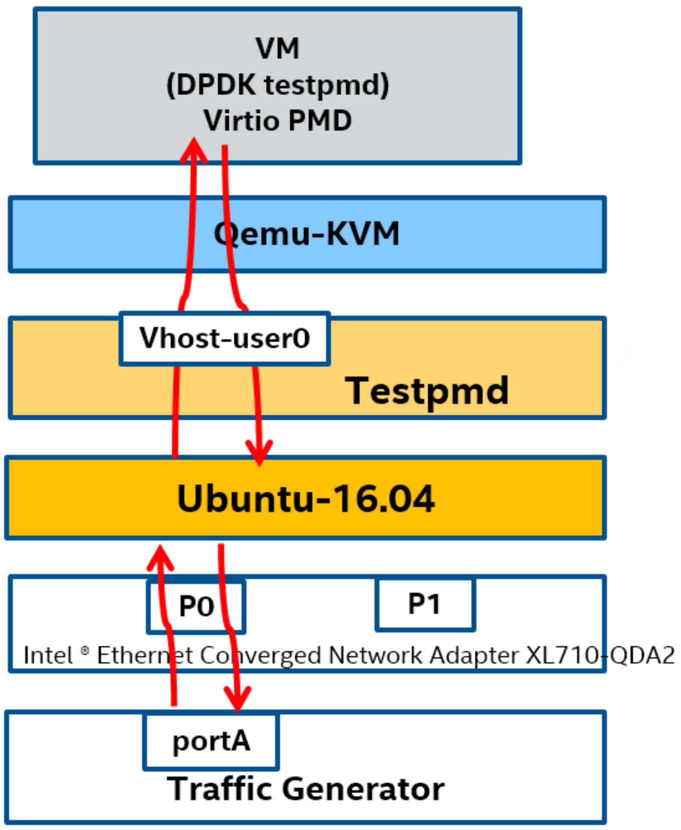

Figure1. DPDK PVP test setup

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.

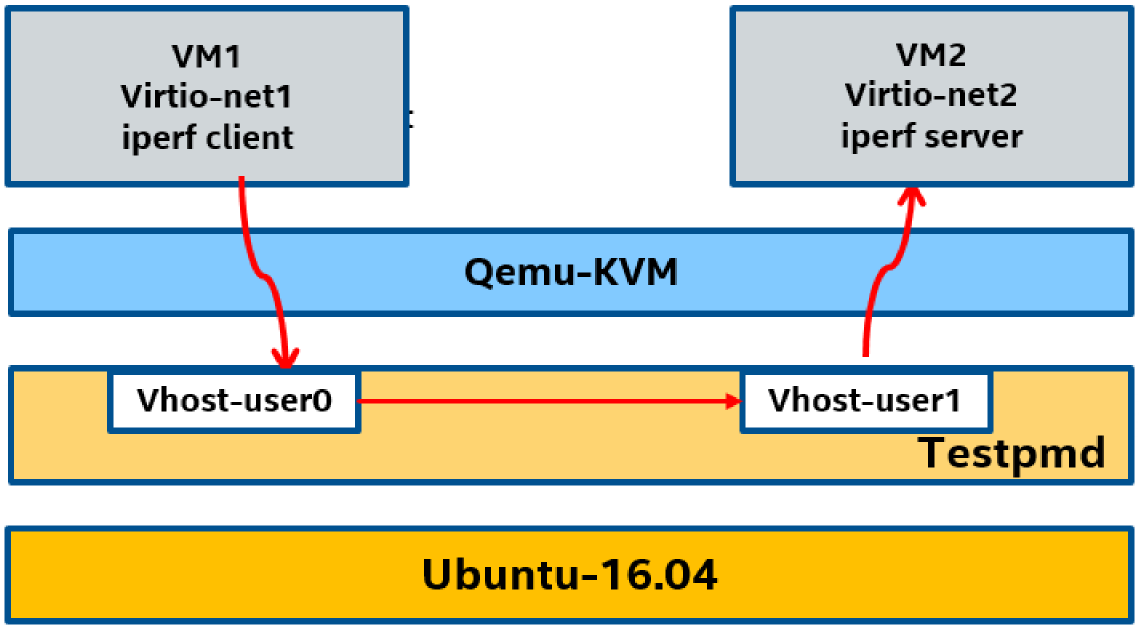

Figure2. DPDK VM2VM iperf test setup

Intel® Xeon® Processor Platinum 8160 (33M Cache, 2.10 GHz)

Hardware & Software Ingredients

Item	Description
Server Platform	Intel® Server Board S2600GZ Intel® Server Board S2600GZ Family
CPU	Intel(R) Xeon(R) Platinum 8160 (33M L3 Cache, 2.10 GHz) Number of cores 48, Number of threads 96.
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.01.00.0456
Host Operating System	Ubuntu 16.04 LTS
Host Linux kernel version	4.4.0-21-generic
Host GCC version	gcc (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609
Host DPDK version	18.02
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	18.02
Guest Linux kernel version	4.4.0-62-generic

Boot and BIOS settings

Item	Description
Host Boot Settings	`hugepagesz=1G hugepages=24 default_hugepagesz=1G isolcpus=1-40 intel_iommu=on nohz_full=1-40 rcu_nocbs=1-40 iommu=pt` **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.**
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`

```
# realtime setup
host_isolcpus=10,11,12,13,14

# Disable watchdogs to reduce overhead
echo 0 > /proc/sys/kernel/watchdog
echo 0 > /proc/sys/kernel/nmi_watchdog

# Change RT priority of ksoftirqd and rcuc kernel threads on
isolated CPUs
i=0
for c in `echo $host_isolcpus | sed 's/,/ /g'` ; do
   tid=`pgrep -a ksoftirq | grep "ksoftirqd/${c}$" | cut -d ' ' -f
1`
   chrt -fp 2 ${tid}
   tid=`pgrep -a rcuc | grep "rcuc/${c}$" | cut -d ' ' -f 1`
   chrt -fp 3 ${tid}
   cpu[$i]=${c}
   i=`expr $i + 1`
done

# Change RT priority of rcub kernel threads
for tid in `pgrep -a rcub | cut -d ' ' -f 1` ; do
   chrt -fp 3 ${tid}
done

# no interrupt will routed to the isolated CPUs
for irq in /proc/irq/* ; do
        if [ -d ${irq} ] && ! grep - ${irq}/smp_affinity_list >
/dev/null ; then
                al=`cat ${irq}/smp_affinity_list`
                if [[ ${cpu[*]} =~ ${al} ]] ; then
                        echo 0 > ${irq}/smp_affinity_list
                        printf ${irq}
                        printf "\n"
                fi
        fi
done
```

VM Real Time Settings

```
set_irq_affinity () {

    echo 0 > /proc/irq/${1}/smp_affinity_list }


echo 0 > /proc/sys/kernel/watchdog
echo 0 > /proc/sys/kernel/nmi_watchdog

for irq in `cat /proc/interrupts | grep virtio | cut -d ':' -f 1`
; do

set_irq_affinity ${irq}
done

echo -1 > /proc/sys/kernel/sched_rt_period_us
echo -1 > /proc/sys/kernel/sched_rt_runtime_us
```

Test Case 1 – DPDK PVP RFC2544 zero packet loss test

Item	Description
Test Case	RFC2544 zero packet loss test for Vhost/Virtio PVP Mergeable
NIC	Intel® Ethernet Converged Network Adapter XL710-QDA2 (2x40G)
Driver	i40e DPDK PMD
Test Configuration	Test tool: IxNetwork 8.12.1053.5 EA **Qemu Version: 2.8.0, Qemu above 2.8 can support change Vring size using qemu command.** **Vring size : 1024 ,** the max Vring size Qemu support Hugepage size : 1G Virtio Mergeable: On Forward Mode: testpmd mac forward **Vhost : 1 queue 1 logic core** **Virtio: 1 queue 1 logic core** **Totally 2 logic cores from 2 physical cores are used.**
Flow Configuration	1 Flow with fixed source and destination IP.

Test Step	1. Bind one 40G NIC port to igb_uio
	
	2. Launch Vhost:
	`./x86_64-native-linuxapp-gcc/app/testpmd -l 10-11 -n 4 --socket-mem 256,2048 --vdev 'eth_vhost0,iface=vhost-net,queues=1'  -- -i --txd=1024 --rxd=1024 --nb-cores=1`
	`testpmd>set fwd mac`
	`testpmd>start`
	
	3. Launch VM :
	`chrt -f 95 taskset -c 12,13,14 qemu_2.8/bin/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=3,sockets=1 -drive file=/home/osimg/ubuntu16.img  \`
	`-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,rx_queue_size=1024 \`
	`-netdev tap,id=ipvm1,ifname=tap3,script=/etc/qemu-ifup \`
	`-device rtl8139,netdev=ipvm1,id=net0,mac=00:00:00:00:12:01 -localtime -vnc :10 -daemonize`
	
	4. Launch Virtio in VM:
	`./x86_64-native-linuxapp-gcc/app/testpmd -c 0x6 -n 4 -- -i --txd=1024 --rxd=1024 --tx-offloads=0x0 --enable-hw-vlan-strip`
	`testpmd>set fwd mac`
	`testpmd>start`

Test Result:

Packet Size(Bytes)	Throughput(Mpps)	Line rate%
64	6.12	10.28
128	5.47	16.18
256	4.99	27.59
512	3.76	40.06
1024	2.60	54.27
1280	2.28	59.30
1518	2.10	64.62

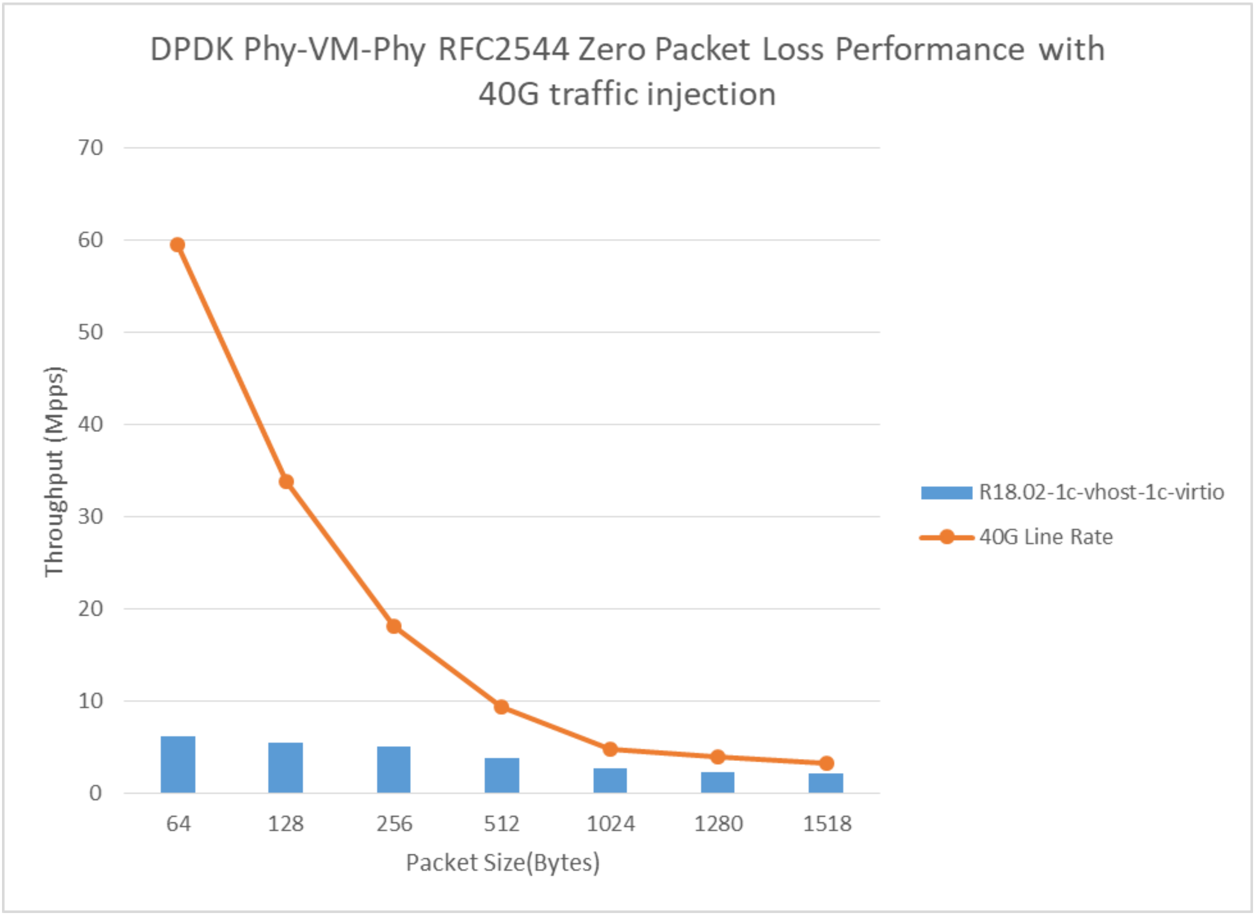

Figure3. DPDK PVP RFC2544 performance with 1core for vhost-user and 1core for virtio

Test Case 2 – DPDK VM2VM iperf performance test

Item	Description
Test Case	virtio-net performance test for VM2VM
Test configuration	**Qemu Version : 2.6** Hugepage size : 1G

	dequeue-zero-copy: Enabled Iperf version: 2.0.5
Core settings	**1 core for vhost-user, and assign each VM 1 core**

Test step	**1. Launch Vhost with :**
	`./examples/vhost/build/vhost-switch -c 0x1c00 -n 4 --socket-mem 2048,2048 -- -p 0x1 --mergeable 1 --vm2vm 1 2 --tso 1 --tx-csum 1 --dequeue-zero-copy --socket-file ./vhost-net --socket-file ./vhost-net1`
	
	2. Launch VM1 and run iperf server:
	`taskset -c 13 qemu-system-x86_64 -name vm0 -enable-kvm -chardev socket,path=/tmp/vm0_qga0.sock,server,nowait,id=vm0_qga0 -device virtio-serial -device virtserialport,chardev=vm0_qga0,name=org.qemu.guest_agent.0 -daemonize -monitor unix:/tmp/vm0_monitor.sock,server,nowait -net nic,vlan=0,macaddr=00:00:00:d9:19:f3,addr=1f -net user,vlan=0,hostfwd=tcp:localhost:6062-:22 -chardev socket,id=char0,path=./vhost-net -netdev type=vhost-user,id=netdev0,chardev=char0,vhostforce -device virtio-net-pci,netdev=netdev0,mac=52:54:00:00:00:01 -cpu host -smp 1 -m 4096 -object memory-backend-file,id=mem,size=4096M,mem-path=/mnt/huge,share=on -numa node,memdev=mem -mem-prealloc -drive file=/home/osimg/ubuntu16.img -vnc :4`
	
	In VM1:
	`ifconfig ens3 1.1.1.2`
	`arp -s 1.1.1.8 52:54:00:00:00:02`
	`iperf -s -i 1`
	
	3. Launch VM2 and run iperf client:
	`taskset -c 15 qemu-system-x86_64 -name vm1 -enable-kvm -chardev socket,path=/tmp/vm1_qga0.sock,server,nowait,id=vm1_qga0 -device virtio-serial -device virtserialport,chardev=vm1_qga0,name=org.qemu.guest_agent.0 -daemonize -monitor unix:/tmp/vm1_monitor.sock,server,nowait -net nic,vlan=0,macaddr=00:00:00:e2:a5:99,addr=1f -net user,vlan=0,hostfwd=tcp:localhost:6090-:22 -chardev socket,id=char0,path=./vhost-net1 -netdev type=vhost-user,id=netdev0,chardev=char0,vhostforce -device virtio-net-pci,netdev=netdev0,mac=52:54:00:00:00:02 -cpu host -smp 1 -m 4096 -object memory-backend-file,id=mem,size=4096M,mem-path=/mnt/huge,share=on -numa node,memdev=mem -mem-prealloc -drive file=/home/osimg/ubuntu16-2.img -vnc :5`
	
	In VM2:
	`ifconfig ens3 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`

Test Result:

Throughput with vhost dequeue zero-copy	**44.6 Gb/s**

Note: In previous version before 18.02, if we enable the dequeue zerocopy, the payload of the packet will be corrupted. For example, scp file between VMs will be failed. In DPDK v18.02, this issue is fixed. This issue fix will bring performance drop. On Intel Intel(R) Xeon(R) CPU E5-2699 v4 which was used in DPDK 17.11 performance report, the VM2VM iperf performance will drop from 44.1 to 38.0 Gbps. In this DPDK 18.02 performance report, as we change the CPU from Broadwell to the latest Intel processor, so the performance still be above 44 Gbps. But actually, there is performance drop here. Please pay attention on this.

For more info, please refer to following commit info:

Author:	Junjie Chen <junjie.j.chen@intel.com>
Author date:	1 month ago (1/17/2018 11:45:53 PM)
Committer:	Ferruh Yigit <ferruh.yigit@intel.com>
Commit date:	1 month ago (1/21/2018 10:51:52 PM)
Commit hash:	3ebd930588b7847906e08e2645a35761a90abf2a
Parent(s):	7365504f77

vhost: fix mbuf free

dequeue zero copy change buf_addr and buf_iova of mbuf, and return
to mbuf pool without restore them, it breaks vm memory if others allocate
mbuf from same pool since mbuf reset doesn't reset buf_addr and buf_iova.

Fixes: b0a985d1f340 ("vhost: add dequeue zero copy")
Cc: stable@dpdk.org

Signed-off-by: Junjie Chen <junjie.j.chen@intel.com>
Reviewed-by: Maxime Coquelin <maxime.coquelin@redhat.com>
Acked-by: Yuanhan Liu <yliu@fridaylinux.org>

Contained in no branch
Contained in tags: v18.02, v18.02-rc4, v18.02-rc3, v18.02-rc2, v18.02-rc1

§