DPDK Intel NIC Performance Environment Information - IXGBE

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 a software traffic generator , named Trex, which will control NIC to transmit packets and determines the throughput at the tester side.

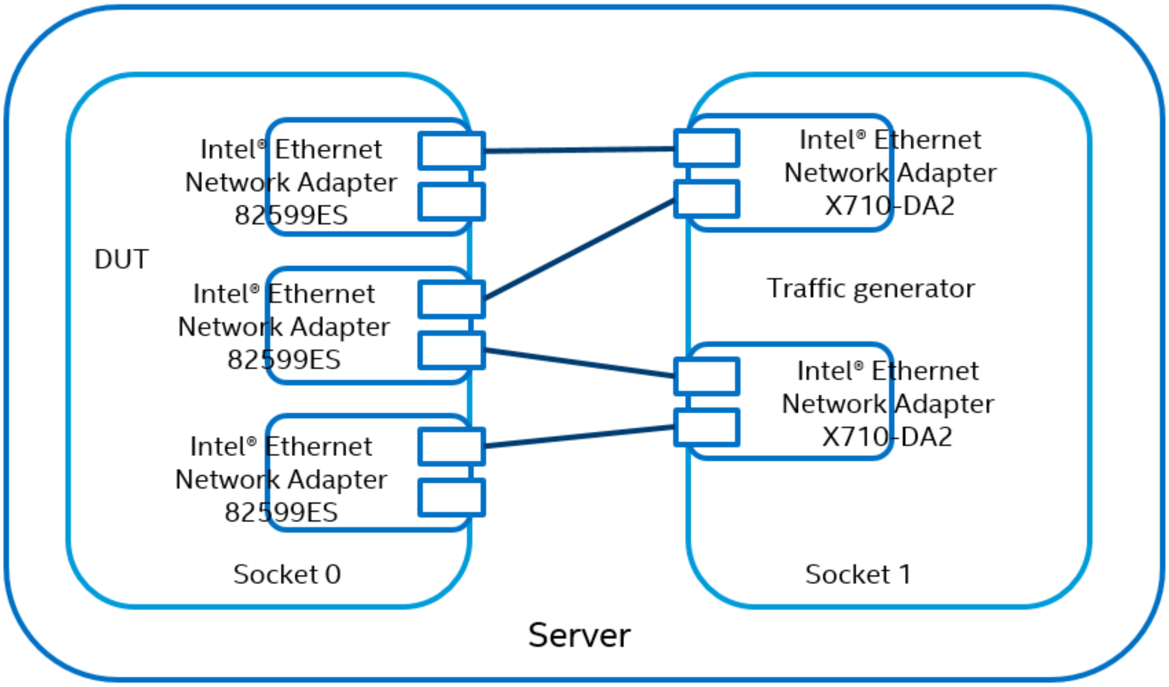
DPDK Testpmd application: Documentation may be found at http://www.dpdk.org/doc/guides/testpmd_app_ug/index.html.

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).

Single core test case: Used to check the maximum IO throughput for a single core. The case requires at least 4x 10GbE ports for the ixgbe DPDK Poll Mode Driver (PMD) and 2x 40GbE ports for the i40e DPDK PMD. In this case, we don't calculate the packet loss rate but record the average throughput within 60 seconds.

More details about the DPDK setup and BIOS information can be found at http://dpdk.org/doc/guides/linux_gsg/nic_perf_intel_platform.html

Setup topology: Below is setup topology for the performance test. Generally, NIC ports in DUT connect to the traffic generator ports directly. And we will send bi-directional traffic from traffic generators and calculate the aggregated throughputs at the traffic generator side for the overall throughput. Please note the NIC port connections to traffic generator would vary from case to case.


Intel® Ethernet Network Adapter 82599ES
DUT
Intel® Ethernet Network Adapter 82599ES
Intel® Ethernet Network Adapter 82599ES
Socket 0
Intel® Ethernet Network Adapter X710-DA2
Traffic generator
Intel® Ethernet Network Adapter X710-DA2
Socket 1
Server

DUT and traffic generator on the same server, need 4x 10G to get the max performance

Figure1. DPDK ixgbe test setup

Hardware & Software Ingredients

Item	Description
Server Platform	Supermicro X11DPG-QT http://www.supermicro.com/products/motherboard/Xeon/C620/X11DPG-QT.cfm
CPU	Intel® Xeon® Platinum 8160 Processor (33M Cache, 2.10 GHz) https://ark.intel.com/products/120501/Intel-Xeon-Platinum-8160-Processor-33M-Cache-2_10-GHz Number of cores 48, Number of threads 96.
Memory	Total 98304 MBs over 12 channels @ 2666 MHz
PCIe	4 x PCIe Gen3 x8 slots
NICs	3 x Intel ® Ethernet Converged Network Adapter 82599ES (4X10G)
Operating System	Ubuntu 17.10
BIOS	2.0b
Linux kernel version	4.13.0-43-generic
GCC version	gcc version 7.2.0 (Ubuntu 7.2.0-8ubuntu3)

Boot and BIOS settings

Item	Description
Boot settings	`hugepagesz=1G hugepages=8 panic=30 isolcpus=1-22,25-46 rcu_nocbs=1-22,25-46 default_hugepagesz=1G intel_pstate=disable nmi_watchdog=0 audit=0 nosoftlockup processor.max_cstate=1 intel_idle.max_cstate=1 hpet=disable mce=off numa_balancing=disable` Note: nohz_full and rcu_nocbs is to disable Linux system interrupts, and it's important for zero-packet loss test. Generally, 1G huge pages are used for performance test.
BIOS	CPU Power and Performance Policy <Performance> CPU C-state Disabled CPU P-state Disabled Enhanced Intel® Speedstep® Tech Disabled Turbo Boost Disabled

Test Case – ixgbe PMD Single core performance

Item	Description
Test Case	Single core performance for ixgbe PMD
NIC	3 x Intel ® Ethernet Converged Network Adapter 82599ES
Topology	Figure1
Driver	ixgbe
Device ID	0x10fb
Device Firmware	0x61bf0001, 0x18f10001
Test configuration	3 NIC cards attached to the first processor, only 1st port used of 2 NICs, and 2 ports used of 1 NIC, total 4 ports. Each port has 1 queue assigned. All queues are assigned to the same logical core. The first core defined is for the CLI of the Testpmd example due to the –i argument. Ixgbe Vector PMD and IO forwarding is used for the test. Test duration is 60 seconds.

Command line	`./x86_64-native-linuxapp-gcc/app/testpmd -c 0x6 -n 4 -w 0000:05:00.1 -w 0000:07:00.1 -- -i  --rxq=1 --txq=1 --portmask=0x3 --txd=512 --rxd=512`
	`testpmd>start`
	Note: txd/rxd will be changed case by case.

Combination covered in case:

frame_size (bytes)	txd/rxd (descriptors)	throughput Difference (Mpps)
64	128	
64	512	
64	2048	