Impact of the Intel® Data Plane Development Kit (Intel® DPDK) on Packet Throughput in Virtualized Network Elements

Using the Advantech® NeTarium*-2 ATCA System and the Intel® Data Plane Development Kit (Intel® DPDK) to achieve higher packet throughput on the Intel® Xeon® processor E5-2600 product family in virtualized network function scenarios

Introduction

Server consolidation using virtualization is a norm for IT data centers today. It is achieved by consolidating workloads running on multiple servers onto a single, virtualized server platform that can be managed centrally. The primary motivation for consolidation within IT data centers is that once multiple workloads have been consolidated onto a single virtual hardware platform, fewer physical servers will be required. This helps reduce IT management costs and lower power usage for server power and cooling by consolidating workloads running on multiple servers onto a single, virtualized server platform that can be managed centrally and decreases the amount of physical space required for server farms.

Service providers are now well aware of the benefits of virtualization, and this is leading to a paradigm shift in their future network architectures, particularly 4G. The driving force from where these changes stem is the ability of a single host server running multiple virtual machines (VMs) to now represent the equivalent of multiple physical servers. A good example of this can be found in the Evolved Packet Core (EPC), where elements described in the 3GPP specification, such as the MME, PGW, and SGW, each of which were originally represented by a physical hardware element, can now share the same hardware platform with the help of Intel® Virtualization Technology (Intel® VT).

Problem Statement

The traffic profiles of telecom and enterprise networks are quite different, as evidenced by the large disparity in average packet size, which ultimately drives different design requirements for communication infrastructure devices and IT data center servers. Communications equipment typically handles small packets, usually 64 bytes, whereas IT data center servers typically deal with an average packet size of 1K bytes. Thus, communications infrastructure devices often need to handle as many as 14.88 million packets per second versus just 1.2 million packets per second for a server in a data center. As a consequence, the typical server network interface card driver can drop a huge number of packets. Thus, Intel has delivered a paradigm shift in packet processing to solve this problem, which is called the Intel® Data Plane Development Kit (Intel® DPDK), featuring a poll mode driver.

This white paper describes the results of a joint effort between Advantech® and Intel to investigate the impact of the Intel DPDK running on the Intel® Xeon® processor E5-2600 product family in an Advantech ATCA system that implements single root I/O virtualization (SR-IOV) technology and Intel® Data Direct I/O Technology (Intel® DDIO) functions. The investigation shows the performance advantages of these technologies on Intel® processor-based hardware used for virtualized communications infrastructure applications.
Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d):

Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) enables a direct communication channel between a guest OS and a physical I/O port on the device. Each device is given a dedicated area in system memory that can be accessed only by the device and by its assigned guest OS. Intel added enhancements to facilitate memory translation and ensure protection of memory that enables a device to directly DMA to/from host memory.

One concern with direct assignment is that it has limited scalability; a physical device can only be assigned to one VM. For example, a dual port NIC allows for direct assignment to two VMs (one port per VM), yet with Intel VT-d, both ports would have to be assigned to one VM. This places a fundamental limit on the number of I/O devices that can be placed in one system.

Consider for a moment a fairly substantial server with two physical CPUs with eight cores each. Using the rule of thumb of one VM per core, there are potentially 16 VMs running at one time, so that having direct I/O assigned to each of those VMs would require 16 physical ports.

The industry recognizes the problems of alternative architectures and is developing new devices that are natively shareable. These devices replicate resources necessary for each VM to be directly connected to the I/O device so the main data movement can occur without virtual machine monitor (VMM) involvement.

SR-IOV

The PCI-SIG® Single Root I/O Virtualization (SR-IOV) and Sharing Specification defines extensions to the PCI Express® (PCIe®) specification suite that enable multiple system images or guests to directly access subset portions of physical I/O resources for performance data movement and to natively share common shared resources behind the host interface. SR-IOV device presents single or multiple physical functions (PFs), which are standard PCIe functions. Each physical function can have zero or more virtual functions (VFs), which are “light-weight” PCIe functions that have enough resources for major data movement, as well as unique requester identifiers (RIDs) to index the input/output memory management unit (IOMMU) page table for address translation.

SR-IOV extends this by enabling multiple direct communication channels for each physical I/O port on the device. The method employed by SR-IOV provides unique memory space, work queues, interrupts, and command processing for each interface exposed while utilizing underlying hardware resources. An SR-IOV device presents single or multiple physical functions (PFs), which are standard PCIe functions. Each physical function can have zero or more virtual functions (VFs), which are “light-weight” PCIe functions that have enough resources for major data movement, as well as unique requester identifiers (RIDs) to index the input/output memory management unit (IOMMU) page table for address translation.

SR-IOV specification defines a standardized mechanism to create natively shared devices.

In essence, SR-IOV is a technology that allows virtual machine platforms to bypass the hypervisor to directly access resources on the physical network interface, thus dramatically improving I/O performance. While it is quite easy to achieve a throughput of 10Gbps, and even 40Gbps, for a single stream with large packets, the objective of this investigation is to demonstrate the impact of the Intel DPDK on regular 64 byte packet sizes using SR-IOV in order to show what is possible on typical 10 GbE connections between communications infrastructure devices.

**MIC-5332**
- Two Intel® Xeon® processors E5-2600 product family
- Intel® C600 Series Chipset, a platform control hub (PCH) server class chipset with integrated SAS controller
- Eight DDR3 VLP DIMMs up to 256 GB with ECC support
- Up to four XAUI ports on Fabric interface
- Two 1000BASE-T ports on Base interface
- Three 1000BASE-T front panel ports
- One fabric mezzanine module support with front I/O support (type II)
- Two CFast / one 2.5” SSD storage device
- Fully managed, hot swappable RTM

**FMM-5001**
- PCIe®-based extension module for ATCA CPU blades & RTMs
- Based on single Intel® 82599EB 10 Gigabit Ethernet Controller
- Dual SFP+ front panel ports
- PCI Express 2.0 host interface
- Single-size fabric mezzanine module Type II
- Implicit e-keying support with FRU EEPROM for management
- Low Power <9.3W
Intel® Data Direct I/O Technology (Intel® DDIO)

Intel DDIO is a key component of Intel® Integrated I/O that increases throughput by allowing Intel® Ethernet Controllers and Server Adapters to talk directly with the processor cache. Traditional I/O transfers are first stored in main memory before they are moved to cache for processing. Then, once processing is complete, the data has to reverse its course. With Intel DDIO, Intel rearchitected the processor and dedicated a portion of cache to I/O so data transfers go directly to cache and bypass main memory. As a result, the performance of network elements can increase without having to implement main memory with greater bandwidth scalability, lower power utilization, and reduced latency. Testing methods were also devised to evaluate SR-IOV together with Intel DDIO, to investigate the effect Intel DDIO has on VMs using SR-IOV virtual network interfaces.

SR-IOV Mode Utilization in an Intel® Data Plane Development Kit (Intel® DPDK) Environment

In order to evaluate the impact of the Intel DPDK on virtual functions, Advantech performed a series of benchmark tests to measure L2 and L3 packet forwarding using SR-IOV against an increasing number of virtual functions. The same tests were then performed with core pinning and Intel DDIO enabled.

The hardware configuration selected for the test setup was a two-slot ATCA system with two Advantech MIC-5332 10G ATCA processor blades, each equipped with two Intel® Xeon® processors E5-2658. One processor blade was used as a traffic generator running the Pktgen application. The other processor blade, the device under test (DUT), was configured to run Fedora* 15, and an Intel DPDK application on an Ubuntu* Virtual Machine OS was set up to receive and forward packets. The test bed specification is detailed in Figure 1 and Table 1, and the test setup is shown in Figure 2.

Scalable Hardware Platform under Test

Advantech's MIC-5332, the ATCA blade used to perform the tests, is a dual-processor ATCA blade based on the Intel Xeon processor E5-2600 product family. It enables industry-leading performance available in an ATCA form factor with up to 20 cores and 40 threads of processing power, fast PCI Express Gen 3 lanes running at up to 8Gbps, and best-in-class virtualization support. Two Intel® QuickPath Interconnects (Intel® QPI) interfaces between the CPUs improve memory and I/O access throughput and latency when one processor needs to access the resources hosted by the other socket. Refer to the MIC-5332 block diagram in Figure 3.

The MIC-5332 provides a highly scalable platform for Advantech's Netarium* Series of ATCA Systems. Entry-level solutions can be deployed with low upfront capital investment and can grow by adding
blades when necessary. An onboard fabric mezzanine module provides connectivity to carrier Ethernet networks and is upgradeable to meet future needs. Advantech's IPMI facilitates the implementation of remote management, configuration, and upgrade services. In addition, the rear transition module's redundant storage feature provides a cost-effective and reliable subscriber database option using robust industrial grade solid-states drives (SSDs). The MIC-5332 integrated into a Netarium ATCA Chassis is designed to meet carrier grade requirements in core network solutions. Enhanced cooling, hot-swap, redundant fan trays, and dedicated RTM cooling ensures the system can work in any central office environment. With its 40Gbps-capable, full mesh backplane, single shelf management controller (ShMC), and AC or DC power options, Advantech's ATCA Systems scale from 2 to 14 slots with 20 to 240 Intel Xeon processor cores, making them ideal for packet processing in a virtual machine cluster. The number of external 10GbE network interfaces can scale from 10 (per MIC-5332 blade + RTM-5106 pair) to 120 (on 12 blades and RTM pairs) in a fully loaded system. OEMs can start with two blades and scale to higher slot counts when consolidating other network functions on a blade or in virtual machines as needed.

**Intel® Data Plane Development Kit (Intel® DPDK) with SR-IOV Performance Data**

The Intel DPDK uses the SR-IOV feature for hardware-based I/O sharing in IOV mode. Therefore, it is possible to partition SR-IOV capability on Ethernet controller NIC resources logically and expose them to a virtual machine as a separate PCI function called a “virtual function.” As a result, a NIC is logically distributed among multiple virtual machines, as shown in Figure 2, while still having global data in common to share with the physical function and other virtual functions. The PCI virtual function of the Intel® 82599 10 Gigabit Ethernet Controller (NIC) is serviced by the poll mode driver (PMD) in the Intel DPDK (igbvf or ixgbevf).

Meanwhile, the PMD also supports the physical function of the NIC for the host and the L2 switch on the Intel 82599 10 Gigabit Ethernet Controller, allowing guests to move inter-VM traffic in SR-IOV mode. A virtual function operates under its respective physical function on the same NIC Port; and therefore, it has no access to the global NIC resources that are shared between other functions for the same NIC port. A virtual function does, however, have basic access to the queue resources and control structures of the...
queues assigned to it. For global resource access, a virtual function must send a request to the physical function for that port, and the physical function operates on the global resources on behalf of the virtual function. For this out-of-band communication, an SR-IOV-enabled NIC provides a memory buffer for each virtual function, which is called a “mailbox”.

A programmer can enable a maximum of 63 virtual functions, and there must be one physical function per Intel 82599 10 Gigabit Ethernet Controller NIC port. The reasons for this are the device allows for a maximum of 128 queues per port, and a virtual/physical function has to have at least one queue pair (RX/TX). The current implementation of the Intel DPDK ixgbevf driver supports a single queue pair (RX/TX) per virtual function, and the host must have a physical function configured by the Linux* ixgbe driver (in the case of the Linux Kernel-based Virtual Machine [KVM]).

Virtual function enumeration is performed by the Linux PCI driver for a dual-port NIC. Enabled virtual functions have a Function# represented by (Bus#, Device#, Function#) in sequence starting from 0 to 3, such that:

- Virtual functions 0 and 2 belong to physical function 0 on core 0
- Virtual functions 1 and 3 belong to physical function 1 on core 1

This enumeration is an important consideration to take into account when targeting specific packets to a selected port. In the setup described in Figure 2, the Intel DPDK I2fwd-vf sample application in the Guest OS is executed with Hugepages enabled. For the expected benchmark performance, the cores from the Guest OS are pinned to the Host OS. Packets are generated by the Pktgen blade and transmitted over a 10G Ethernet link to a virtual Ethernet bridge and classifier running the virtual functions. On the software side, a VMM is a utility for virtual machine management and is used to create, start, stop, and delete virtual machines.

Test Results
In Figure 2, I2fwd-vf shown is an Intel DPDK application used as a forwarding example, which can be run with and without SR-IOV enabled. From the results shown in Table 2, the Intel DPDK with SR-IOV enabled increases L2 forwarding performance, but data throughput decreases as the number of streams increases.

It was observed that running more virtual functions on a single core degrades the performance from 13.3 to 9.1 Mpps since adding virtual functions increases the workload on the core.

From the data collected, a single core running two virtual functions yields similar performance as two cores running four virtual functions, whereby each core handles two separate virtual functions. Likewise, similar performance was measured for two cores running two virtual functions and four cores running four virtual functions, with each individual core assigned one virtual function.

Based on the two stream data, a single core running two virtual functions delivers 9.1 Mpps, compared to the 7.8 Mpps from two cores running two virtual functions, where each virtual function is pinned to its own core. The reason for the decrease in performance is that a single core keeps packet data in its L1/L2/L3 caches, but processing packets for different virtual functions on a two core system requires packets to be transmitted from L1 down to L2 and to L3, which incurs extra latency. Hence, using a single core to process four virtual functions provides better performance than the other configurations.

In short, having a VM dedicated for packet I/O while the other VMs process other workloads returns the best results. This allows the threads to share cache lines on multiple levels (first-, second-, and last-level cache) and minimizes the need for cache fill operations used for cache prefetching. However, binding an application to a single core produces the best results when all the threads are accessing the same cached data.

Impact of Intel® Data Plane Development Kit (Intel® DPDK) on Packet Throughput in Virtualized Network Elements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>L2 Forwarding (Mpps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Data Plane Development Kit (Intel® DPDK) Native</td>
<td></td>
</tr>
<tr>
<td>1 stream to VF0</td>
<td>13.3 13.2</td>
</tr>
<tr>
<td>2 streams to VF0 and VF1</td>
<td>9.1 9.2</td>
</tr>
<tr>
<td>4 streams to VF0,1,2,3</td>
<td>- - - -</td>
</tr>
</tbody>
</table>

Table 2: Performance Data for the Intel® Data Plane Development Kit (Intel® DPDK) with SR-IOV Enabled for L2 Forwarding

In the setup described in Figure 2, the Intel DPDK I2fwd-vf sample application in the Guest OS is executed with Hugepages enabled. For the expected benchmark performance, the cores from the Guest OS are pinned to the Host OS. Packets are generated by the Pktgen blade and transmitted over a 10G Ethernet link to a virtual Ethernet bridge and classifier running the virtual functions. On the software side, a VMM is a utility for virtual machine management and is used to create, start, stop, and delete virtual machines.

Test Results
In Figure 2, I2fwd-vf shown is an Intel DPDK application used as a forwarding example, which can be run with and without SR-IOV enabled. From the results shown in Table 2, the Intel DPDK with SR-IOV enabled increases L2 forwarding performance, but data throughput decreases as the number of streams increases.

It was observed that running more virtual functions on a single core degrades the performance from 13.3 to 9.1 Mpps since adding virtual functions increases the workload on the core.

From the data collected, a single core running two virtual functions yields similar performance as two cores running four virtual functions, whereby each core handles two separate virtual functions. Likewise, similar performance was measured for two cores running two virtual functions and four cores running four virtual functions, with each individual core assigned one virtual function.

Based on the two stream data, a single core running two virtual functions delivers 9.1 Mpps, compared to the 7.8 Mpps from two cores running two virtual functions, where each virtual function is pinned to its own core. The reason for the decrease in performance is that a single core keeps packet data in its L1/L2/L3 caches, but processing packets for different virtual functions on a two core system requires packets to be transmitted from L1 down to L2 and to L3, which incurs extra latency. Hence, using a single core to process four virtual functions provides better performance than the other configurations.

In short, having a VM dedicated for packet I/O while the other VMs process other workloads returns the best results. This allows the threads to share cache lines on multiple levels (first-, second-, and last-level cache) and minimizes the need for cache fill operations used for cache prefetching. However, binding an application to a single core produces the best results when all the threads are accessing the same cached data.

Impact of Intel® Data Plane Development Kit (Intel® DPDK) on Packet Throughput in Virtualized Network Elements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>L2 Forwarding (Mpps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Data Plane Development Kit (Intel® DPDK) Native</td>
<td></td>
</tr>
<tr>
<td>1 stream to VF0</td>
<td>13.3 13.2</td>
</tr>
<tr>
<td>2 streams to VF0 and VF1</td>
<td>9.1 9.2</td>
</tr>
<tr>
<td>4 streams to VF0,1,2,3</td>
<td>- - - -</td>
</tr>
</tbody>
</table>

Table 2: Performance Data for the Intel® Data Plane Development Kit (Intel® DPDK) with SR-IOV Enabled for L2 Forwarding

In the setup described in Figure 2, the Intel DPDK I2fwd-vf sample application in the Guest OS is executed with Hugepages enabled. For the expected benchmark performance, the cores from the Guest OS are pinned to the Host OS. Packets are generated by the Pktgen blade and transmitted over a 10G Ethernet link to a virtual Ethernet bridge and classifier running the virtual functions. On the software side, a VMM is a utility for virtual machine management and is used to create, start, stop, and delete virtual machines.

Test Results
In Figure 2, I2fwd-vf shown is an Intel DPDK application used as a forwarding example, which can be run with and without SR-IOV enabled. From the results shown in Table 2, the Intel DPDK with SR-IOV enabled increases L2 forwarding performance, but data throughput decreases as the number of streams increases.

It was observed that running more virtual functions on a single core degrades the performance from 13.3 to 9.1 Mpps since adding virtual functions increases the workload on the core.

From the data collected, a single core running two virtual functions yields similar performance as two cores running four virtual functions, whereby each core handles two separate virtual functions. Likewise, similar performance was measured for two cores running two virtual functions and four cores running four virtual functions, with each individual core assigned one virtual function.

Based on the two stream data, a single core running two virtual functions delivers 9.1 Mpps, compared to the 7.8 Mpps from two cores running two virtual functions, where each virtual function is pinned to its own core. The reason for the decrease in performance is that a single core keeps packet data in its L1/L2/L3 caches, but processing packets for different virtual functions on a two core system requires packets to be transmitted from L1 down to L2 and to L3, which incurs extra latency. Hence, using a single core to process four virtual functions provides better performance than the other configurations.

In short, having a VM dedicated for packet I/O while the other VMs process other workloads returns the best results. This allows the threads to share cache lines on multiple levels (first-, second-, and last-level cache) and minimizes the need for cache fill operations used for cache prefetching. However, binding an application to a single core produces the best results when all the threads are accessing the same cached data.
Intel® Data Plane Development Kit (Intel® DPDK) with Core Pinning and Intel® Data Direct I/O Technology (Intel® DDIO) Versus Non-Intel DDIO Performance Data

Table 3 shows the impact of Intel DDIO on throughput performance. Using the setup shown in Figure 4, two cores running two virtual functions show similar results as four cores running four virtual functions, whereby each individual core is assigned one virtual function. Using the Intel DPDK with SR-IOV for L3 forwarding performance, there is no improvement in performance by adding more pairs of cores/virtual functions, and the data throughput decreases as the number of streams increases. When processing two streams, VF0 and VF3 have lower throughput than VF0 and VF1 since the data streaming has to go through a different core. Without processor pinning, the system can dispatch virtual processors to any physical processor in the system and can move the Intel DPDK to different cores, which causes the performance degradation, as shown in Table 4. Processor pinning ensures the system only dispatches a virtual processor to one or more specified physical processors. For example, a virtual processor can be pinned to a single physical processor, to the physical processors on the same core, or to the physical processors on the same socket.

From Figure 5, it can be seen that the Intel DPDK does a very good job in optimizing 64 bytes packet performance, although enabling/disabling Intel DDIO has no impact on performance. It can therefore be concluded that Intel DDIO does not improve the performance in non-virtualized environments. However, there is significant improvement with Intel DDIO enabled in a virtualized environment, as shown in Table 4. Intel DDIO is new technology available in the Intel Xeon processor E5-2600 product family, which is an improvement compared to prior generations of processors.

**Performance Data**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Single Core 4 VF ports (1 core, 4 ports)</th>
<th>Dual Core 4 VF ports (1 core, 2 ports)</th>
<th>Quad Core 4 VF ports (1 core, 1 port)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Intel DDIO</td>
<td>non-Intel DDIO</td>
<td>Intel DDIO</td>
</tr>
<tr>
<td>Intel® data Plane Development Kit (Intel® DPDK)</td>
<td>13.5</td>
<td>12.8</td>
<td>13.2</td>
</tr>
<tr>
<td>with 1 stream to VF0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intel DPDK with 2 streams to VF0</td>
<td>13.4</td>
<td>12.6</td>
<td>-</td>
</tr>
<tr>
<td>Intel DPDK with 2 streams to VF0 and VF1</td>
<td>8.9</td>
<td>8.8</td>
<td>9.0</td>
</tr>
<tr>
<td>Intel DPDK with 2 streams to VF0 and VF3</td>
<td>-</td>
<td>-</td>
<td>7.9</td>
</tr>
<tr>
<td>Intel DPDK with 4 streams to VF0, 1, 2, 3</td>
<td>8.3</td>
<td>8.1</td>
<td>7.4</td>
</tr>
</tbody>
</table>

Table 3: Performance Data for Intel® Data Direct I/O Technology (Intel® DDIO) Versus Non-Intel DDIO
Impact of Intel® Data Plane Development Kit (Intel® DPDK) on Packet Throughput in Virtualized Network Elements

PERFORMANCE DATA

Table 4: Performance Data for Intel® Data Direct I/O Technology (Intel® DDIO) Enabled With and Without Core Pinning

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Single Core (1 core, 4 ports)</th>
<th>Dual Core (1 core, 2 ports)</th>
<th>Quad Core (1 core, 1 port)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Data Plane Development Kit (Intel® DPDK) with 1 stream to VF0</td>
<td>13.5</td>
<td>13.2</td>
<td>13.2</td>
</tr>
<tr>
<td>Intel DPDK with 2 streams to VF0</td>
<td>13.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Intel DPDK with 2 streams to VF0 and VF1</td>
<td>8.9</td>
<td>8.5</td>
<td>9.0</td>
</tr>
<tr>
<td>Intel DPDK with 2 streams to VF0 and VF3</td>
<td>-</td>
<td>7.9</td>
<td>7.9</td>
</tr>
<tr>
<td>Intel DPDK with 4 streams to VF0, 1, 2, 3</td>
<td>8.3</td>
<td>8.0</td>
<td>7.4</td>
</tr>
</tbody>
</table>

Table: Performance Data for Intel® Data Direct I/O Technology (Intel® DDIO) Enabled With and Without Core Pinning

Packet Throughput

Configuration:
L3 Forwarding Benchmark
Intel® Data Plane Development Kit (Intel® DPDK) rte-20110211
Rose City CRB, 8x4GB DDR3-1333MHz
1xSNB-EP 8C B0, 2.0GHz
Linux® 2.6.33.6
4x Intel® Ethernet Server Bypass Adapter X520-SR2,
ECG Labs, February 2011

Figure 5: Intel® Data Direct I/O Technology (Intel® DDIO) Versus Non-Intel DDIO Across Packet Sizes
Results & Analysis

The greater the number of entries in the input/output translation lookaside buffer (IOTLB), the less likely it is that caching a new entry in the IOTLB will cause an eviction of another entry that will be used later. Due to the limited number of entries in the IOTLB to support the large I/O transaction rate of 14.88 Mpps (64 bytes), the caching of a new entry in the IOTLB will always cause an eviction of another entry.

Enlarging the IOTLB can be done by either increasing the number of sets or increasing the IOTLB associativity. However, since streaming DMA mappings are only cached for a short while and have spatial locality, IOTLB thrashing is less likely to occur. It should be noted, increasing the number of IOTLB entries increases the complexity and cost of implementation in hardware.

Conclusion

In the data center, server consolidation using virtualization yields many benefits. However, to realize these benefits with communications infrastructure devices, servers supporting virtualized network functions must be able to handle very high packet count per second since packet sizes are small, typically 64 bytes. Using the Intel DPDK with SR-IOV, testing indicates improved performance for both L2 and L3 forwarding. There is also performance degradation when the number of streams increases since adding virtual functions increases the workload on the same core. Promisingly, the best performance is achieved using single cores to process four virtual functions at a time.

The Intel DPK delivers significant performance increases for 64 byte packets, whereas enabling/disabling Intel DDIO results in no specific performance gains in a non-virtualized environment. There is significant improvement in packet throughput performance, however, when Intel DDIO is enabled in a virtualized environment. Overall, the performance data shows that virtualization on the Intel DPDK has good potential.