TECHNICAL BRIEF

Intel Corporation
Datacenter Network Solutions Group

Demonstrating Data Plane Performance Improvements using Enhanced Platform Awareness

Authors

Shivapriya Hiremath
Solutions Software Engineer

Tarek Radi
Lead Solution Enabling Manager

Michał Chrząszcz
Solutions Software Engineer

Table of Contents

1.0 Introduction

For the past years, many organizations have been adopting Network Functions Virtualization (NFV) to improve their business agility and reduce operational costs. Virtual Network Functions (VNF), NFV Infrastructure (NFVI), and NFV Management and Orchestration (MANO), are the main components of an NFV architecture. Together, they enable Communication Service Providers (CoSPs), Enterprises and Cloud Service Providers to consolidate and manage specialized network equipment onto industry standard, high-volume servers (SHVS). At the same time, virtualization technology and hardware advancements continue to evolve, allowing deployment of more VNFs on a single server. One of the key NFV benefits is that it enables service flexibility and scalability at lower operational costs. This can be more readily achieved if the underlying infrastructure is utilized effectively.

The rich variety of hardware features makes the SHVS based on the Intel® architecture a favorable choice for NFV deployments; however, network administrators and architects may not always know how to take full advantage of the underlying platform capabilities. One option to achieve efficient infrastructure utilization is Enhanced Platform Awareness (EPA). EPA represents a methodology and a related suite of changes across multiple layers of the orchestration stack targeting intelligent platform capability, configuration & capacity consumption. EPA features include Huge page support, NUMA topology awareness, CPU pinning, integration with OVS-DPDK, support for I/O Passthrough via SR-IOV, and many others.

This Technical Brief focuses on performance benchmarking and flavor verification steps and shows how leveraging even a few EPA features can lead to significantly improved and more predictable performance, and therefore better utilization of the underlying infrastructure.

This paper presents performance improvements achieved with several EPA features enabled while onboarding VNFs within an OpenStack* environment deployed on Intel architecture-based SHVS. The document summarizes two typical VNF onboarding test cases:

- Palo Alto Networks®* VM-Series Next-Generation Firewall* VNF spawned as a stand-alone VM
- Three VNFs—a Commercial vRouter*, a Sandvine* Policy Traffic Switch Virtual Series*, and a Palo Alto Networks® VM-Series Next-Generation Firewall VNF spawned to form a static service chain.

The results show an approximate 5x throughput improvement for the platform with a single VM, and 9x throughput gain for the platform with a service chain.

2.0 Accelerating OpenStack* Environments on Intel® Architecture

OpenStack is a leading open-source suite for cloud computing. It is capable of managing large pools of compute, storage, and networking resources throughout a data center. It provides administrators with a convenient web interface used to control and provision resources and run virtualized software.

OpenStack has an active community of users and contributors. Intel contributes to OpenStack with EPA features, enabling a better view of the underlying hardware and offering an advanced selection of tuning capabilities. EPA allows OpenStack administrators to filter platforms with specific capabilities that match particular workload requirements and to deploy NFV solutions with optimal performance and predictable characteristics.

This paper presents a use case example of a few EPA features and platform technologies selected to optimize NFV infrastructure driven by two Intel® Xeon® processors E5-2680 v3. The Intel® Xeon® processor E5 product family helps address requirements of virtualized data centers and clouds by offering leading-edge performance and advanced hardware virtualization features.

2.1 Demonstrating Platform Capabilities

As part of this demonstration, Intel engineers set up two identical single-server platforms, each running an all-in-one OpenStack Newton Installation with the same hardware configuration. To demonstrate the potential of the Intel architecture, each server was configured with a set of EPA features and platform technologies used to observe significant performance gains. These include:

- **Huge Page Support:** This EPA feature enables very large pages to improve system performance by reducing the amount of system resources required to access page table entries. Huge page support is required to allocate large memory pools to packet buffers.
- **Data Plane Development Kit (DPDK):** This set of libraries and drivers enables fast packet processing. The result of integrating DPDK with Open vSwitch is a set of DPDK-accelerated Open vSwitch network devices (netdevs) that allow packets to be processed solely in a user space. The most important advantage of DPDK-enabled netdevs is the significant acceleration of I/O traffic between the virtual switch and a connected network interface controller (NIC).
- **CPU Pinning:** This EPA feature enables the pinning of guest virtual CPUs to physical cores, and therefore ensures an application will be executed on a specific pool of cores. When enabled with CPU isolation, it ensures that other processes will not be scheduled on that pool.
- **Single-Root Input/Output Virtualization (SR-IOV):** This technology allows using one or more virtual functions (VFs) of a physical function in a VM or container, which bypasses the hypervisor to reduce overhead and improve performance.

While this paper focuses on a small set of technologies, it is important to note that even better results may be possible if other EPA features or platform technologies are enabled. For example, a security workload may achieve higher performance by taking advantage of the Intel® Trusted Execution Technology (Intel® TXT).

Please refer to the Enabling Enhanced Platform Awareness for Superior Packet Processing in OpenStack configuration guide for a comprehensive list of EPA features included in OpenStack, and the necessary steps to enable corresponding platform technologies with EPA.

The following two VNF onboarding test cases demonstrate performance capabilities of Intel architecture platforms:

- A commercial VNF spawned as a stand-alone VM (described in section 2.3)
- Three commercial VNFs spawned to form a static service chain (described in section 2.4)

These two deployments were setup on dedicated servers with Intel® Hyper-Threading Technology (Intel® HT Technology) enabled. More information on the hardware and software configurations of these two deployments is presented in sections 2.3 and 2.4.

The following networks compose the OpenStack network topology in both setups:

- The `public` flat provider network is used to reach the Internet.
- The `ixnet1` and `ixnet2` flat provider networks are directly associated with the two ports of the Intel 82599 ES 10 gigabit Ethernet Controller on system under test (SUT). They are connected to two corresponding IxVM Virtual Load Modules on the traffic generator host(described in section 2.2). The port on the `ixnet1` network acts as a “client port,” sending HTTP requests to the “server port” on the `ixnet2` network.
- The `management` network is used to manage the OpenStack services and instances. It is mapped to one of the Intel® Ethernet Controller X540-AT2 interfaces on the OpenStack node.

Figure 1 shows the setup for the test case with a single VM which was a Palo Alto Networks* VM-Series Next-Generation firewall VNF spawned in an OpenStack environment, while Figure 2 depicts the test setup with three chained VNFs.

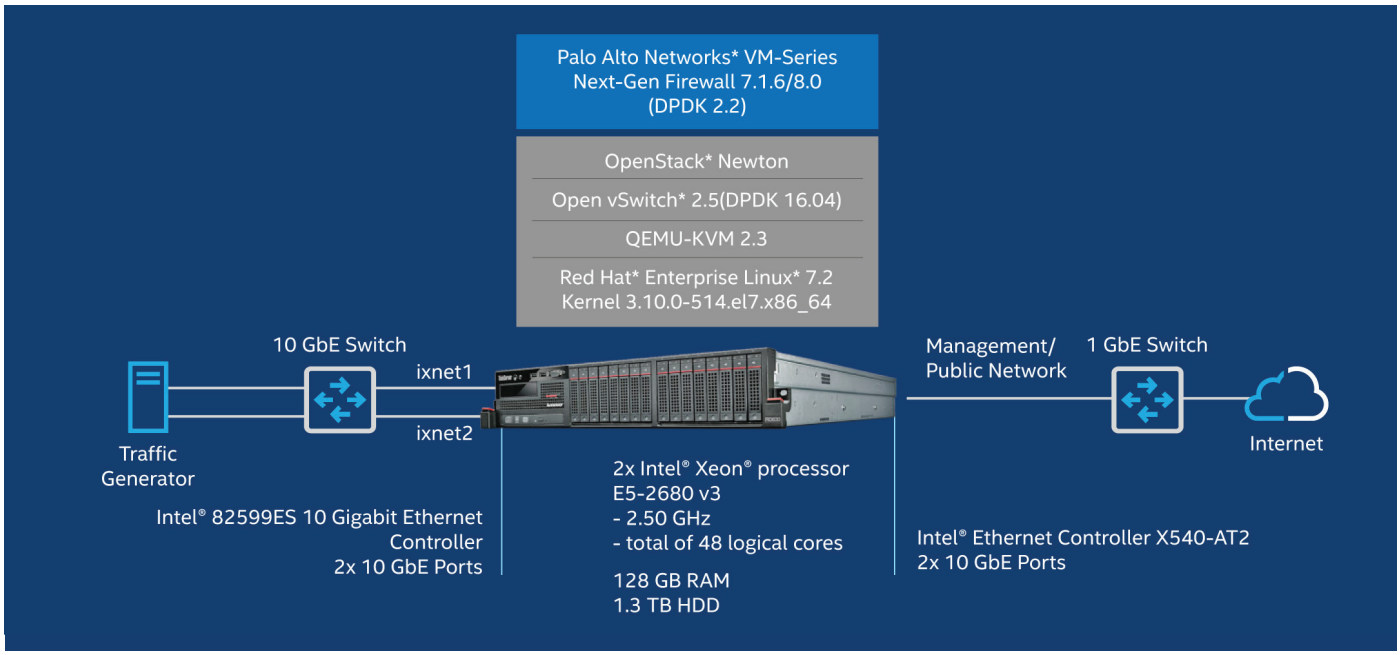

Figure 1. Test setup for the platform with a standalone virtual firewall.

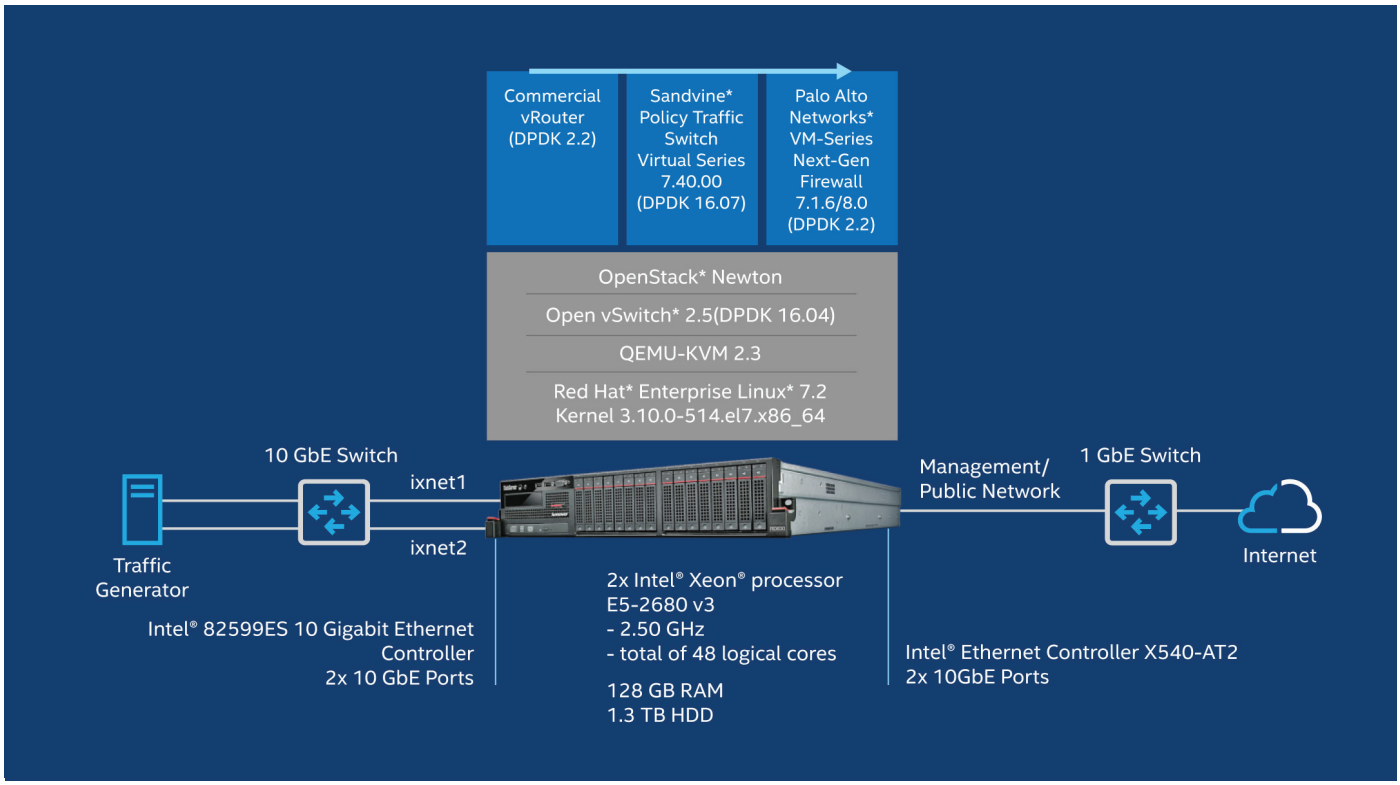

Figure 2. Test setup for the platform with three chained VNFs.

2.2 Traffic Generation

Ixia* IxLoad Virtual Edition* Tier-3 10G was deployed on a separate server for HTTP traffic generation. IxLoad Virtual Edition is released in the form of IxVM Virtual Chassis and IxVM Virtual Load Module VMs running over QEMU-KVM and CentOS* 7.0.

In this test setup, the traffic generator consists of three VMs: an IxVM Linux Chassis and two IxVM Virtual Modules, as depicted in Figure 3. Each VM is allocated 4,096 MB of RAM and two virtual CPUs pinned to physical cores on the host.

The IxVM Linux Chassis connects to the `management` network for external access, and internally with the two IxVM Virtual Load Modules for backplane communication between the VMs.

Both IxVM Virtual Load Modules have two virtual ports connected via a `test` virtual switch. Ports denoted as eth0 are used to send traffic, while eth1 ports receive traffic. Thus, each of the VMs participates in traffic generation.

On the host, the two ports (NIC 0 and NIC 1) connect to the system-under-test (SUT). The NIC 0 sends the traffic via the `ixnet1` network, whereas NIC 1 on the `ixnet2` network is used for traffic reception.

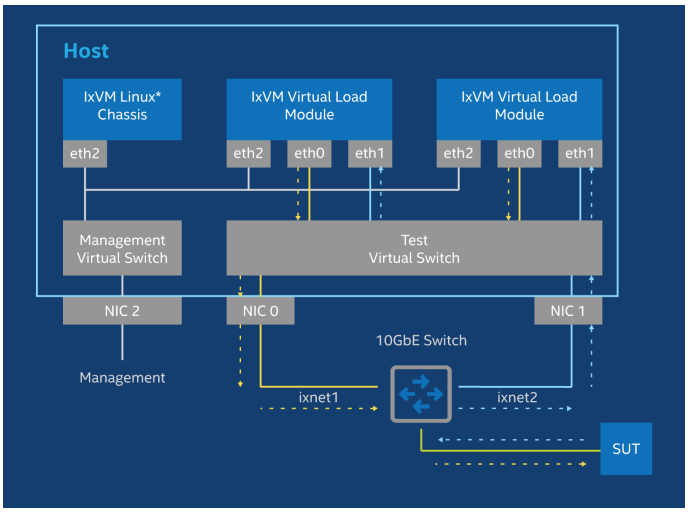

Figure 3. Overview on the traffic generator host.

2.3 Test Setup #1: Standalone VNF

Figure 1 shows a setup for the test case with a single VM, which is a Palo Alto Networks VM-Series Next-Generation firewall VNF spawned in an OpenStack environment. OpenStack Newton was installed on the same server using Packstack*'s all-in-one deployment option with both controller and compute nodes in the same server. The figure also presents the hardware and software configuration of the platform. The baseline configuration includes the Open vSwitch and Palo Alto Networks Next-Generation Firewall VNF (without DPDK integration). The software components (DPDK versions) in the best performing configuration are indicated in parentheses, which includes DPDK optimized Palo Alto Networks Next-Generation Firewall VNF.

In the standalone VM setup, the firewall VNF connects to the `ixnet1, ixnet2` and the `management` networks. The HTTP traffic from the client port on the traffic generator enters the virtual firewall on `ixnet1` and reaches the server port on the traffic generator on `ixnet2`. The HTTP reply traffic flows in the opposite direction, reaching the virtual firewall on `ixnet2` and then returning to the traffic generator on the client port on `ixnet1`.

2.4 Test Setup #2: VNFs in a Static Service Chain

Figure 2 presents the setup for the second scenario configured in the same way as the previous one. The setup includes three VNFs, a virtual router from a 3rd party vendor, virtual network policy controller from Sandvine, and virtual next-generation firewall from Palo Alto Networks, manually configured to forward traffic within the service chain in the given order.

In the chained setup, the flow becomes more complex because apart from the Host-VM communication, there is also inter-VM communication. The HTTP traffic from the client port on the traffic generator enters the virtual router on `ixnet1`. Then, the router forwards the traffic to the virtual DPI, which consequently forwards the traffic to the virtual firewall. The Firewall VNF forwards the traffic to the traffic generator on `ixnet2` network. The HTTP reply traffic flows in the opposite direction, from the virtual firewall on `ixnet2`, then reaching the virtual DPI and virtual router, and finally returning to the traffic generator on the client port on `ixnet1`.

3.0 Test Methodology

We have considered the benchmarking methodology for firewall performance as given in RFC3511.

We have benchmarked the two setups using HTTP traffic of the following configuration:

- Content-type field of the HTTP header: `text/html`
- Transaction sizes: 1MB for the Throughput metric, and 1B for the other 2 metrics

The performance metrics have been gathered at the traffic generator to characterize end-to-end performance of each deployment. Table 1 presents these metrics and the corresponding workload characteristics configured in the traffic generator.

Metric	Objective	Workload Configuration in the Traffic Generator
Throughput	To measure throughput performance of the deployment	• HTTP 1.1 • Maximal possible transactions per second • 21 concurrent connections per user • 1048576-byte GET • 1048576-byte TCP Send/Receive Buffer
Connections per second	To measure how well the deployment can accept and service new connections	• HTTP 1.0 • 21 concurrent connections per user • 1-byte GET
Maximum Concurrent connections	To measure deployment scalability via number of sustained TCP connections	• HTTP 1.1 • Maximum possible transactions per second • 21 concurrent connections per user • 1-byte GET

Table 1. Configuration of testing workload in the traffic generator.

3.1 Optimizations

The two presented deployments were optimized by enabling specific EPA features incrementally, and the measurements were gathered after a tuning method was applied. Although multiple tuning variants were tested, this technical brief focuses only on the two particular configuration profiles (see Table 2 for configuration details):

- **Baseline Configuration:** Initial configuration of the setup (i.e., without EPA features or platform technologies enabled)
- **Best Configuration:** Final configuration of the setup (i.e., resulting in the highest obtained dataplane performance)

	Baseline Configuration	Best Configuration
Intel® Hyper-Threading Technology (Intel® HT Technology)	Enabled	Enabled
CPU Core Isolation	No	Yes, isolated cores were allocated to Open vSwitch, PMD threads, and VMs.
CPU Pinning	No	Yes, isolated cores were pinned to specific VMs. Note the sibling threads were not used for scheduling VMs
Memory Page Size	4 KB	1 GB (huge pages)
Virtual Switch	Open vSwitch	OVS-DPDK (i.e., DPDK at NFVI level)
Number of poll mode driver (PMD) Threads	N/A	Setup #1: 1 PMD thread pinned to a physical core Setup #2: 4 PMD threads pinned to four physical cores
DPDK Support at VNF Layer	No	Yes (see Table 3 for DPDK versions used)

Table 2. Details of the baseline and best configuration profiles.

Table 3 shows the resources allocated to each VNF as well as the DPDK versions used by each VNF (this is separate from the DPDK used at the NFVI layer). Figure 4 shows the CPU layout of the host machines. Sibling cores exist since Hyperthreading was enabled.

VNF	Virtual CPUs	NUMA Node	Memory	DPDK version
Setup #1: Standalone VM				
Palo Alto Networks* VM-Series Next-Gen Firewall 8.0	8 (Cores 0-7)	0	16 GB	2.2
Setup #2: Chained VNFs				
Commercial vRouter	2 (Cores 12-13)	1	4 GB	2.2
Sandvine* Policy Traffic Switch Virtual Series 7.40.00	4 (Cores 14-17)	1	8 GB	16.07
Palo Alto Networks* VM-Series Next-Gen Firewall 8.0	8 (Cores 0-7)	0	16 GB	2.2

Table 3. Resources allocated to VNFs in the best configuration profiles.

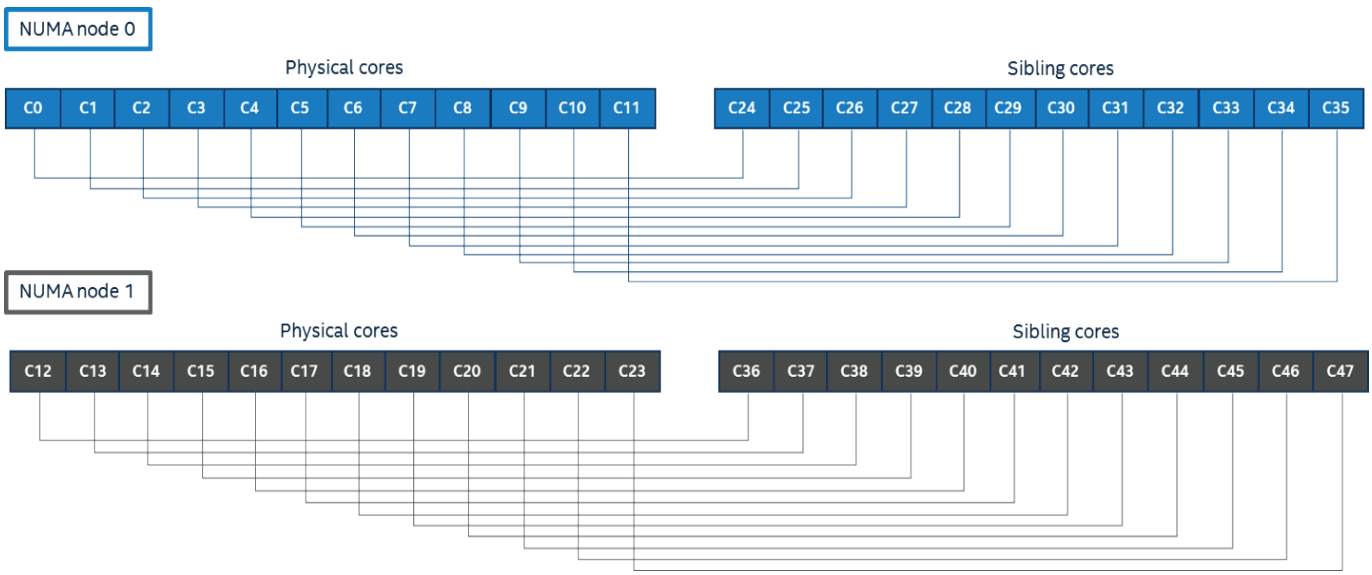

Figure 4. CPU Layout of the host machines.

3.2 Optimizations with SR-IOV

An alternative optimization method uses Single-Root Input-Output Virtualization (SR-IOV). In this case, the firewall VNF is spawned in OpenStack with an SR-IOV interface that enables the allocation of VFs to the VM from the PCIe* device, such as a network adapter. In this configuration the VM does not have a virtio interface, and a virtual switch is not used for communication.

To use SR-IOV, the Input-Output Memory Management Unit (IOMMU) and Intel® Virtualization Technology for Directed I/O (Intel® VT-d) must be enabled in the kernel. To learn more about enabling SR-IOV technology, refer to the Enabling Enhanced Platform Awareness for Superior Packet Processing in OpenStack document.

In this configuration, the device drivers for the NICs are configured to enable the NIC VFs on the host. OpenStack Networking requires the SR-IOV modular layer 2 (ML2) mechanism driver to be enabled. Since the firewall VNF used supports only the MacVTap interface instead of direct port, it is necessary to create the SR-IOV port with `vnic _ type` as `macvtap` and to boot the instance with that port assigned.

4.0 Results

The throughput performance with standalone VM (Setup #1) is presented in Figure 5. Compared to the baseline profile (solid blue line), the throughput is 2x higher when applying 1 GB huge pages, isolating and pinning CPU cores to virtual cores at VM, and enabling DPDK-accelerated Open vSwitch (dashed grey line).

Even better performance is possible when DPDK is enabled in the virtual firewall; this configuration (upper solid black line) achieves approximately five times greater performance compared to the baseline configuration profile. This demonstrates the advantage of DPDK's polling technique of packet processing in the user space. The VNF is able to process more packets this way, in turn yielding higher throughput.

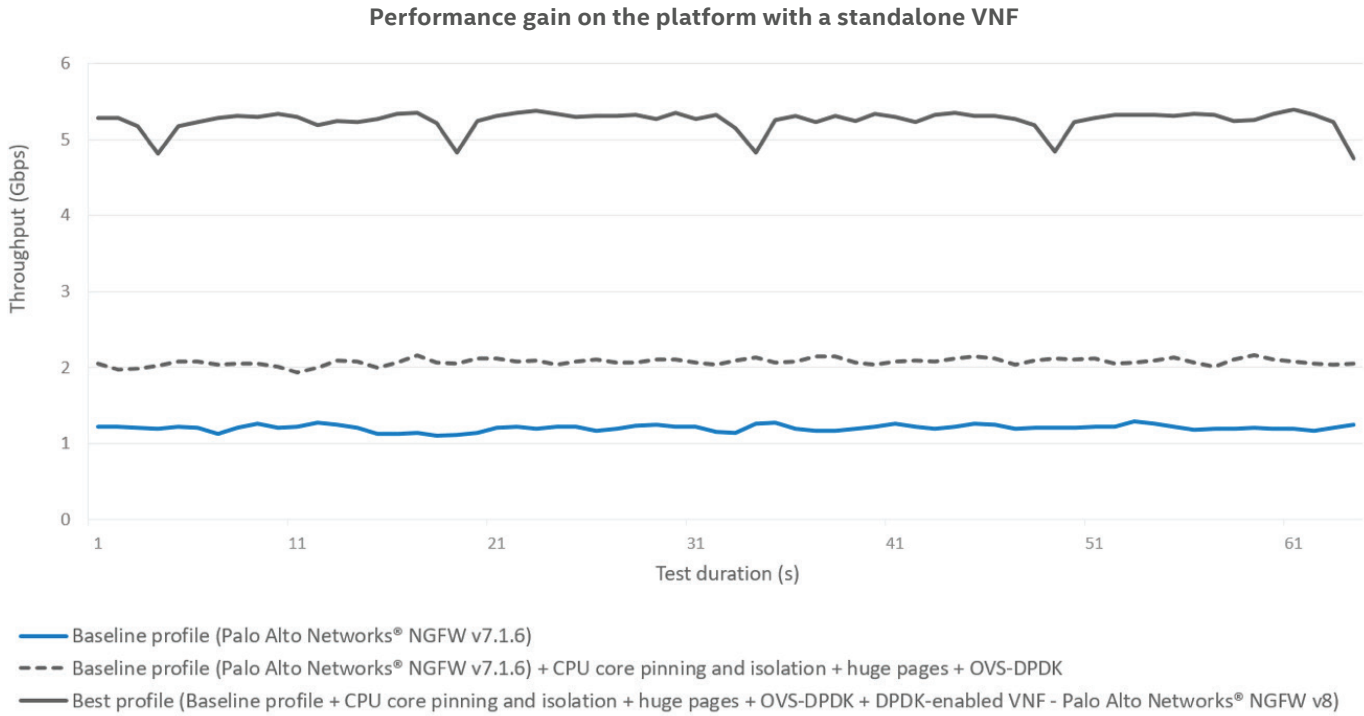

Figure 5. Impact of performance optimizations on the throughput at the platform with a standalone VNF (Setup #1).

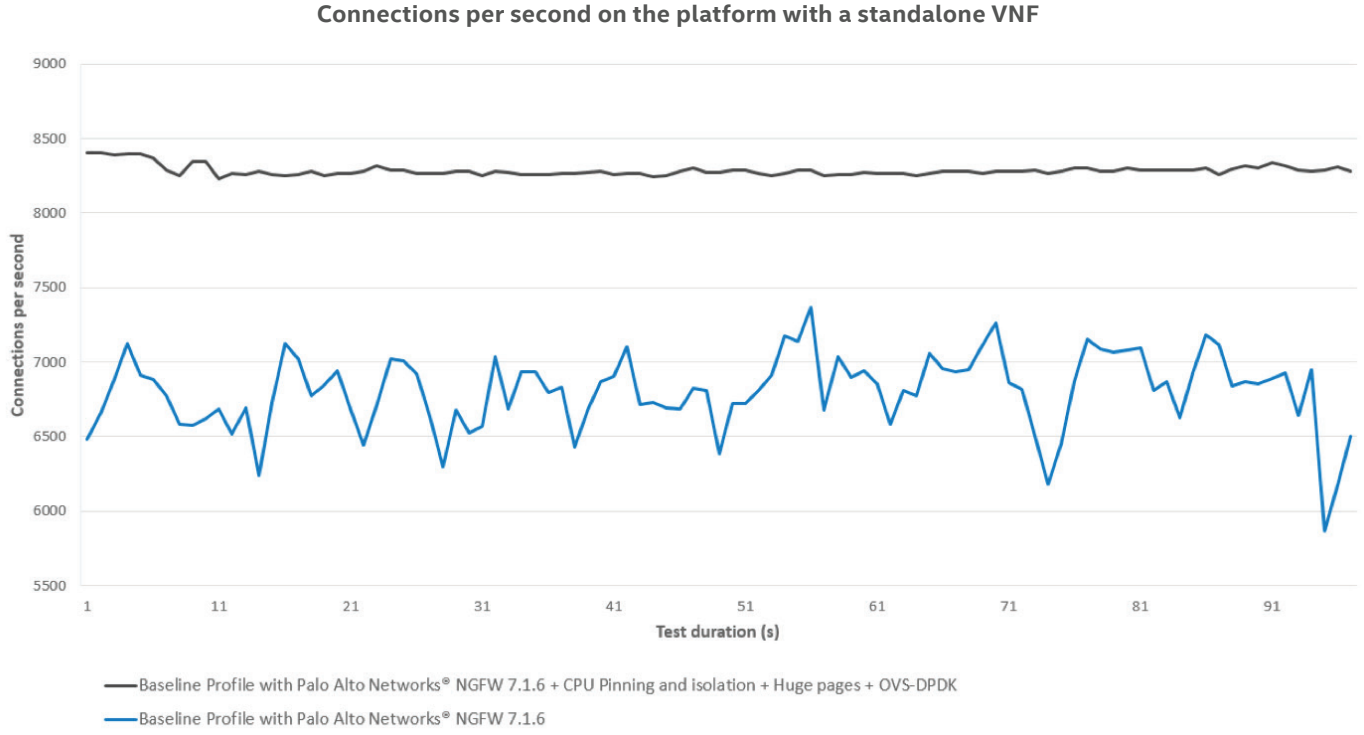

Figure 6. Impact of performance optimizations on consistency and stability of the performance at the platform with a standalone VNF (Setup #1).

In Figure 6, the baseline profile (solid blue line) was not stable. This was because CPU pinning was not enabled. Later, when CPU pinning was enabled (black line), performance was a lot more consistent and stable.

Figure 7 shows throughput of the Setup #1 with an SR-IOV interface enabled in the VNF yielding a ~3x improvement over the baseline. In this case, the packets are directed from the physical network interface to the virtual function in the VM without the overhead of a virtual switch in between but through the MacVTap abstraction layer.

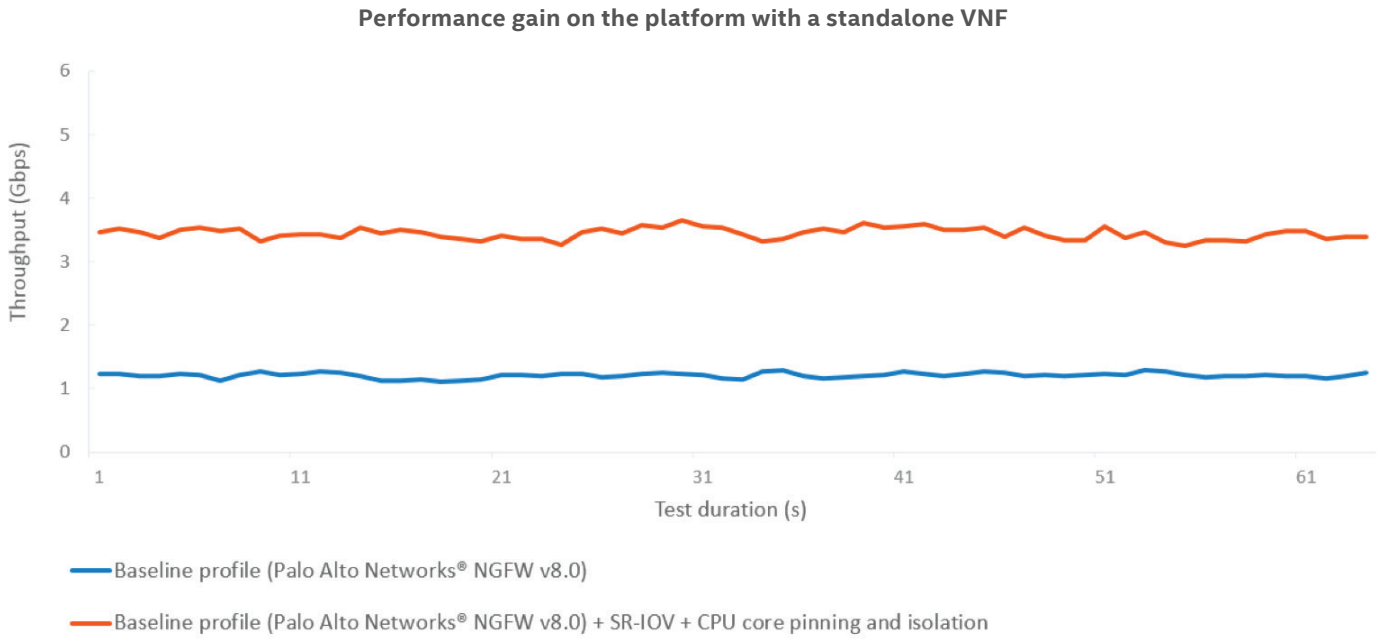

Figure 7. Impact of SR-IOV and core pinning and isolation on the throughput at the platform with a standalone VNF (Setup #1).

Figure 8 shows the performance difference between the baseline profile (solid blue line) and the best profile (solid grey line). This was on Setup #2 which contained a static chain of three VNFs. When allocating four PMD threads to the OVS-DPDK at the NFVI layer, a performance boost of about nine times was measured.

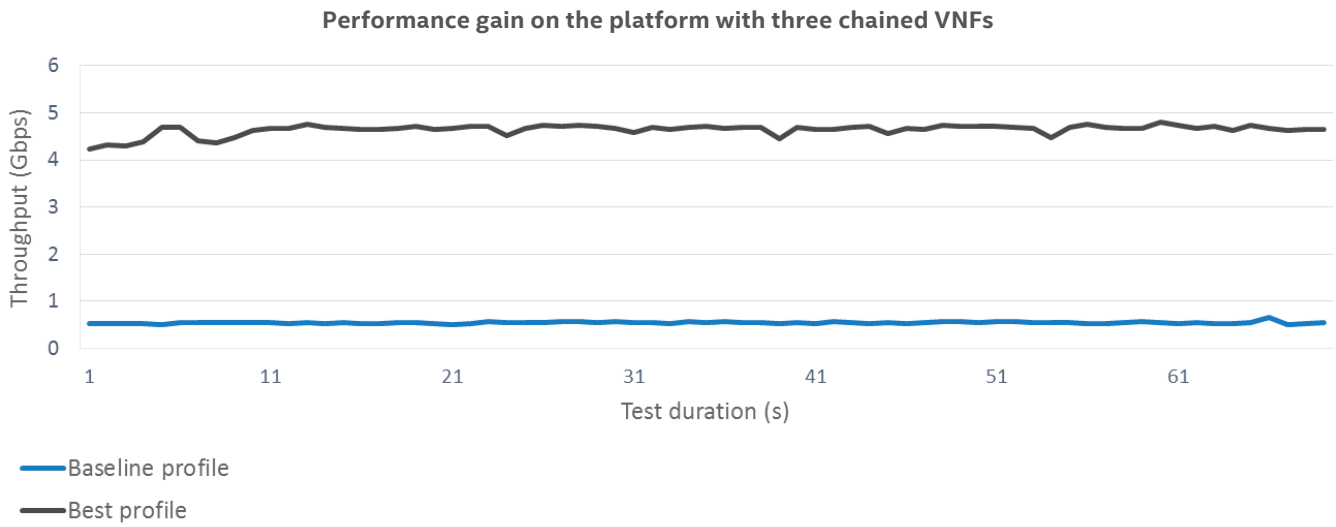

Figure 8. Impact of performance optimizations on the throughput at the platform with three chained VNFs (Setup #2).

5.0 Summary

Enabling EPA features and other platform technologies significantly improves performance. In the presented platform configurations, the maximum performance was achieved by enabling DPDK in both NFVI and VNFs along with enabling 1 GB huge pages and pinning virtual CPUs on the guest to physical cores on the host. The achieved gain was approximately equal to 5x in the case of a standalone VNF setup, and approximately 9x for the platform with three chained VNFs.

The enabled EPA features and platform technologies also increased the number of connections per second. In case of the standalone VNF deployment, this number increased by up to 15%. However, the maximum concurrent connections measurement was found to be a function of memory that was allotted to the VNF. Hence, there was no impact of EPA features and platform technologies on that KPI.

Apart from offering performance gains, EPA also offers increased efficiency by delivering more capabilities with existing resources leading to better total cost of ownership. EPA ensures that the intended workload runs on the right hardware with matching compute, storage, network, and security capabilities.

The performance results presented in this paper do not represent the absolute maximum possible and better performance can be achieved by enabling other relevant EPA features based on the requirement of the VNF or the Network Service. For example, in order to improve security, cloud subscribers may want their workloads to be executed on verified compute platforms. In OpenStack, it is possible to designate groups of trusted hosts to form pools of trusted computes which is an EPA feature. The trusted hosts are supported with hardware-based security features, such as Intel® TXT, and 3rd party attestation servers.

With EPA in Openstack, operators have additional levels of control and configuration available to them. EPA features are systematically added to OpenStack, exposing new platform capabilities to build powerful and deterministic network performance on industry standard, high-volume servers.

6.0 Next Steps

- To learn more about the technologies mentioned in this paper, please follow the links in the document.
- To learn more about Intel's technology for NFV, attend the courses available in the Intel® Network Builders University at https://networkbuilders.intel.com/university.
- To learn more about Intel® Network Builders partners for NFV products, visit https://networkbuilders.intel.com/solutionscatalog.
- To get a higher performance from your NFV systems, consider using DPDK in both your NFVI and VNF layers.
- To get a higher return on investment from your NFV systems, consider leveraging Enhanced Platform Awareness throughout the 3 elements of your NFV architecture: your orchestration layer, your VNF layer and your NFVI layer.

Appendix A: References

Title	Reference
Benchmarking methodology for firewall performance	https://tools.ietf.org/html/rfc3511
Considerations for SR-IOV and NFV Solutions	https://www.youtube.com/watch?v=6UUFWZs-Sck
DPDK	http://dpdk.org/doc/guides/
Enabling Enhanced Platform Awareness for Superior Packet Processing in OpenStack*	https://builders.intel.com/docs/networkbuilders/EPA_Enablement_Guide_V2.pdf
OpenStack Enhanced Platform Awareness	https://networkbuilders.intel.com/docs/OpenStack_EPA.pdf
OVS-DPDK	https://01.org/openstack/blogs/stephenfin/2016/enabling-ovs-dpdk-openstack
Palo Alto Networks* VM-Series	https://www.paloaltonetworks.com/products/secure-the-network/virtualized-next-generation-firewall/vm-series
Perspective on VNF Onboarding on Intel® Architecture in an NFVI Environment	https://builders.intel.com/docs/Perspective-on-VNF-Onboarding-on-Intel-Architecture-in-an-NFVI-Environment.pdf
Sandvine* Virtual Series	https://www.sandvine.com/platform/sandvine-virtual-series.html
SR-IOV for NFV Solutions	http://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/sr-iov-nfv-tech-brief.pdf

Appendix B: Abbreviations

Abbreviation	Description
DPDK	Data Plane Development Kit
DPI	Deep Packet Inspection
EPA	Enhanced Platform Awareness
Intel® HT Technology	Intel® Hyper-Threading Technology
Intel® TXT	Intel® Trusted Execution Technology
IOMMU	Input-Output Memory Management Unit
MANO	Management and Orchestration
NFV	Network Functions Virtualization
NFVI	NFV Infrastructure
NGFW	Next-Generation Firewall
NIC	Network Interface Controller
OVS-DPDK	DPDK-Accelerated Open vSwitch
PCIe	Peripheral Component Interconnect Express
PF	Physical Function
PMD	Poll Mode Driver
SHVS	Standard High Volume Server
SR-IOV	Single-Root Input-Output Virtualization
SUT	System-under-Test
VF	Virtual Function
VM	Virtual Machine
VNF	Virtual Network Function

Legal Information

By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below.

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. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

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.

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. For more information go to www.intel.com/benchmarks.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel processors of the same SKU may vary in frequency or power as a result of natural variability in the production process.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice. Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel does not control or audit third-party web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel, the Intel logo, Xeon, and others are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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

© 2017 Intel Corporation Printed in USA 336517-001US ♻ Please Recycle 0917/HB/ICMCSW/PDF