Benchmarking Software Data Planes
Intel® Xeon® Skylake vs. Broadwell[1]

March 7th, 2019

Georgii Tkachuk
georgii.tkachuk@intel.com

Maciek Konstantynowicz
mkonstan@cisco.com

Shrikant M. Shah
shrikant.m.shah@intel.com

Table of Content

1 INTRODUCTION 5

1.1 PURPOSE 5

1.2 DOCUMENT STRUCTURE 6

2 BENCHMARKING METHODOLOGY 7

2.1 PACKET PATHS 7

2.2 METRICS 7

2.2.1 Data Plane Applications 7

2.2.2 Compute Resources 8

2.3 PERFORMANCE TESTS 12

2.3.1 Benchmarked Applications 12

2.3.2 Test Environment 13

2.3.2.1 Physical Topology 13

2.3.2.2 Server Configurations 13

2.3.2.3 Traffic Generator and Offered Load 14

3 RESULTS AND ANALYSIS 16

3.1.1 Measurements 16

***3.1.2** First Analysis 18*

3.1.2.1 Instructions-per-Packet 18

3.1.2.2 Instructions-per-Cycle 18

3.1.2.3 Cycles-per-Packet 19

3.1.2.4 Packets-per-Second Throughput 20

3.1.2.5 First Conclusions 21

3.1.3 Throughput Speedup 22

[1] Intel® Xeon® Scalable Processors (Code name Skylake), Intel® Xeon® E5 V4 Family (codename Broadwell)

3.1.3.1 Processor Core Frequency 22
3.1.3.2 Intel Hyper-Threading 22
3.1.4 Further Analysis 23
4 TOP-DOWN MICROARCHITECTURE ANALYSIS (TMA) 24
4.1 INTEL TMA OVERVIEW 24
4.2 TMA RESULTS AND INTERPRETATION 24
5 CONCLUSIONS 28
6 ACKNOWLEDGEMENTS 30
7 REFERENCES 31
8 APPENDIX: TEST ENVIRONMENT SPECIFICATION 32
8.1 SYSTEM UNDER TEST – INTEL® XEON® SKYLAKE-SP HW PLATFORM CONFIGURATION 32
8.2 SYSTEM UNDER TEST AND TESTED APPLICATIONS – INTEL® XEON® SKYLAKE-SP SOFTWARE VERSIONS 32
8.3 SKYLAKE-EP SERVER BIOS SETTINGS 33
8.4 SYSTEM UNDER TEST – INTEL® XEON® BROADWELL HW PLATFORM CONFIGURATION 34
8.5 SYSTEM UNDER TEST AND TESTED APPLICATIONS – INTEL® XEON® BROADWELL SOFTWARE VERSIONS 34
8.6 INTEL® XEON® BROADWELL SERVER BIOS SETTINGS 35
8.7 PACKET TRAFFIC GENERATOR – CONFIGURATION 36
9 INDEX: FIGURES 38
10 INDEX: TABLES 39

Space intentionally left blank.

Legal Statements from Intel Corporation

FTC Disclaimer

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 complete information visit www.intel.com/benchmarks.

Performance results are based on testing as of December 21, 2018. The platforms under test use BIOS and Kernel security patches available at the time. No product or component can be absolutely secure. Please refer to the test system configuration in *Section 8 Appendix: Test Environment Specification*.

FTC Optimization Notice

Optimization Notice: Intel's compilers and DPDK libraries may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804.

The benchmark results may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

'Mileage May Vary' Disclaimer

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 http://www.intel.com/benchmarks

Estimated Results Disclosure

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.

Dependencies Disclosure

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

Trade mark Notice

Intel, Xeon, the Intel logo, and other Intel technologies mentioned in this documents 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.

Other 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.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

Performance claims by third party:

Intel does not control or audit third-party data. You should review this content, consult other sources, and confirm whether referenced data are accurate.

1 Introduction

1.1 Purpose

Software data plane performance and efficiency are foundational properties of any NFV and cloud networking system design. They do not only impact network service economics by dictating feasible service density per unit of compute, but can also be a key enabler of new service architectures.

A good example is the cloud-native architecture based on distributed micro-services, that require meshed IP connectivity for scaled-up inter-process communication and data exchange. Those systems heavily rely on service-aware software data plane for L2 switching, IP routing, load balancing and granular in-band policies used for securing all application interactions.

Thus constructed service-mesh demands a fast software data plane to ensure immediacy of serving external requests, that in turn rely on the rapid communication between micro-services. Any indeterminism in data plane behaviour results in network impairments (packet loss, increased latency) directly impacting efficiency of network transport protocols and responsiveness of applications.

The purpose of this technical paper is to show how the advancements in both, server processor technology and data plane software, achieve new levels of deterministic performance. It builds upon data plane benchmarking, analysis techniques and tools described in [BASWDP][2] and applying them to the same set of software applications (DPDK, FD.io VPP, OVS_DPDK). The benchmarking is conducted on servers based on the latest shipping generation of Intel® Xeon® Scalable Processors (codename Skylake-SP), with test results and efficiency metrics compared to the one obtained from the previous generation on Intel® Xeon® E5 V4 family Processors (codename Broadwell). The gains of the latest processor microarchitecture and their impact on data plane performance are quantified and explained.

Note that the same performance benchmarks haven been used as those published in [BASWDP]. However, considering that DPDK and VPP software have made major improvements in the functionality and performance, the authors have chosen to use the newer versions of software for this paper (version 18.11, and 18.10 respectively). Broadwell data have been recollected, so that architecture comparison with Skylake-SP could be done with the same versions of the software. In addition, BIOS and Kernel patches were applied to the systems to protect against the security vulnerabilities such as “Spectre” and “Meltdown”.

Authors believe that this Skylake benchmarking and analysis data further prove that used testing and analysis methodology enables an effective comparison of software data plane

[2] [BASWDP] “Benchmarking and Analysis of Software Data Planes”, M.Konstantynowicz, P.Lu, S.M.Shah, December 2017, https://fd.io/wp-content/uploads/sites/34/2018/01/performance_analysis_sw_data_planes_dec21_2017.pdf.

applications and their performance across different processor generations. Furthermore it shows that sub-terabit NFV services based on native software data plane are possible.

1.2 Document Structure

The paper is organized as follows.

Section 2. Benchmarking Methodology describes packet path and logical test topology, (re-)introduces the main benchmarking metrics, highlights the main differences between the two tested versions of Intel® Xeon® processors (Skylake vs. Broadwell), and explains the physical environment setup including configurations and offered packet loads.

Section 3. Results and Analysis provides a summary of measurements and first analysis of results and benchmarking comparisons between Skylake and Broadwell processor versions.

Section 4. Top-down Microarchitecture Analysis (TMA) applies Intel's Top-down Microarchitecture Analysis to assessing processor usage efficiency and bottlenecks during the benchmarks, again with comparison between Skylake and Broadwell processor versions.

Section 5. Conclusions summarizes the findings and highlights the focus areas for future work.

Section 6. Acknowledgements

Section 7. References lists materials and publications either referred to in the paper or directly relevant to the content.

Section 8. Appendix: Test Environment Specification includes specification of benchmarked hardware and software components.

2 Benchmarking Methodology

2.1 Packet Paths

Data plane packet path benchmarked in this paper is identical to the one used in [BASWDP]. It consists of a data plane Network Function (NF) application running on a bare-metal compute host, processing and forwarding packets between the physical interfaces located on the Network Interface Cards (NICs), see *Figure 1*. Tested NF application is running in Linux user-mode, taking direct control of the NIC devices, with minimal involvement of Linux kernel in data plane operation.

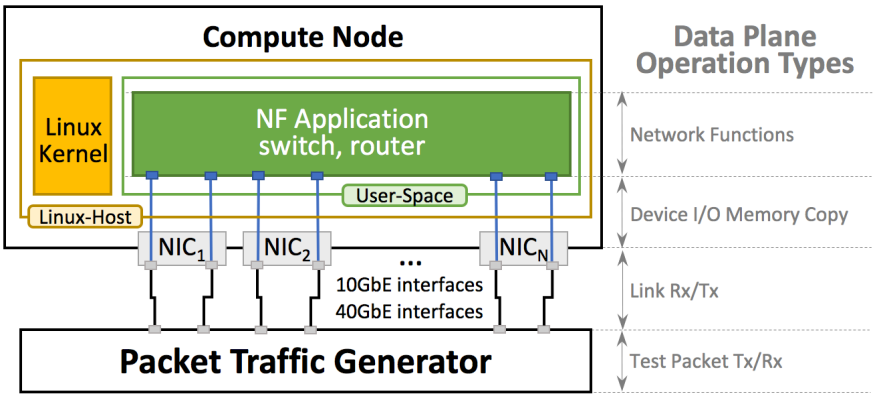

Figure 1. Baseline NF data plane benchmarking topology.

Data plane performance is benchmarked in different hardware configurations to characterize performance speed-up in scaled-up multi-thread and multi-core setup.

Presented baseline setup has two main functional parts, i) driving the physical network interface (physical device I/O, device I/O memory copy) and ii) packet processing (network functions). Both parts are present in majority of deployments, hence their performance and efficiency can be used as a baseline benchmarking reference. Other more complex NF designs involve adding virtual network interfaces (virtual I/O memory-copy) and more network functions, providing richer composite functionality, but at the same time using more compute resources. In other words, the baseline NF benchmarking data described in this paper can be treated as an upper ceiling of NF application capabilities.

2.2 Metrics

A short recap of the basic NF data plane benchmarking metrics used in this paper follows. For more complete description see [BASWDP].

2.2.1 Data Plane Applications

Performance of data plane network applications is associated with compute efficiency metrics using two main equations binding cycles per packet and packet throughput.

A single data plane centric program execution efficiency metric is proposed for benchmarking NF data plane packet processing – *#cycles/packet* (CPP):

$$CPP = \frac{\#cycles}{packet} = \frac{\#instructions}{packet} * \frac{\#cycles}{instruction}$$

Equation 1. #cycles/packet (CPP) as function of #instructions/packet (IPP) and #cycles/instruction (IPC).

Following is a formula binding the packet throughput and CPP metrics:

$$packet\_throughput\ [pps] = \frac{1}{packet\_processing\_time\ [sec]} = \frac{CPU\_freq\ [Hz]}{CPP}$$

Equation 2. Packet_throughput as function of #cycles/packet and CPU frequency.

CPP represents NF application program execution efficiency for a specific set of packet processing operations. The first contributor to CPP, Instructions-Per-Packet, usually remains constant for a given data-plane function. However, any major change in the code execution path, such as handling different protocols differently, can alter this metric. The other contributor, Cycles-Per-Instruction, can greatly vary depending on the processor architecture and operations performed. For example Cycles-Per-Instruction can go high if an application is memory latency bound or I/O bound. If CPP remains the same, packet throughput would vary linearly with the frequency. Following sections show how the CPP metric can be put to effective use for comparing network workload performance across different NF applications, packet processing scenarios and compute platforms.

2.2.2 Compute Resources

[BASWDP] describes all main compute resources critical to performance of NF data plane applications. At the top level four main compute resources matter:

1) **Processor and CPU cores** – for performing packet processing operations.
2) **Memory bandwidth** – for moving packet and lookup data, packet processing code.
3) **I/O bandwidth** – for moving packets to/from NIC interfaces.
4) **Inter-socket bandwidth** – for handling inter-socket operations.

Figure 2 and *Figure 3* depict these compute resources with associated performance counter points in logical diagrams of two-socket server based on Intel® Xeon® Broadwell and Skylake respectively.

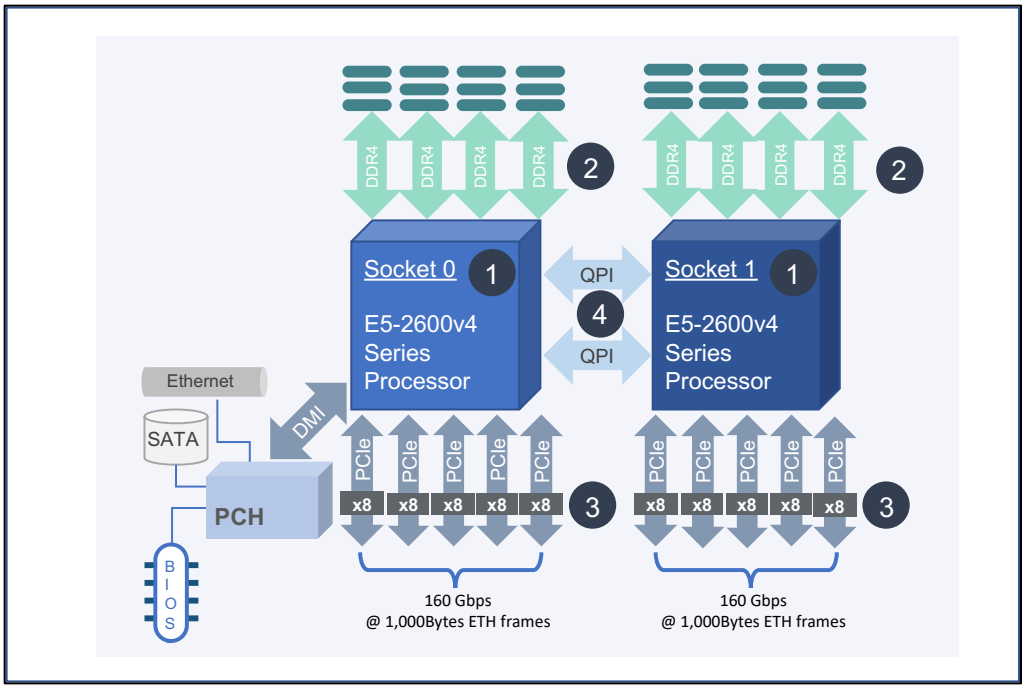

Figure 2. Main compute resources in two-socket server with Intel® Xeon® Broadwell Processors.

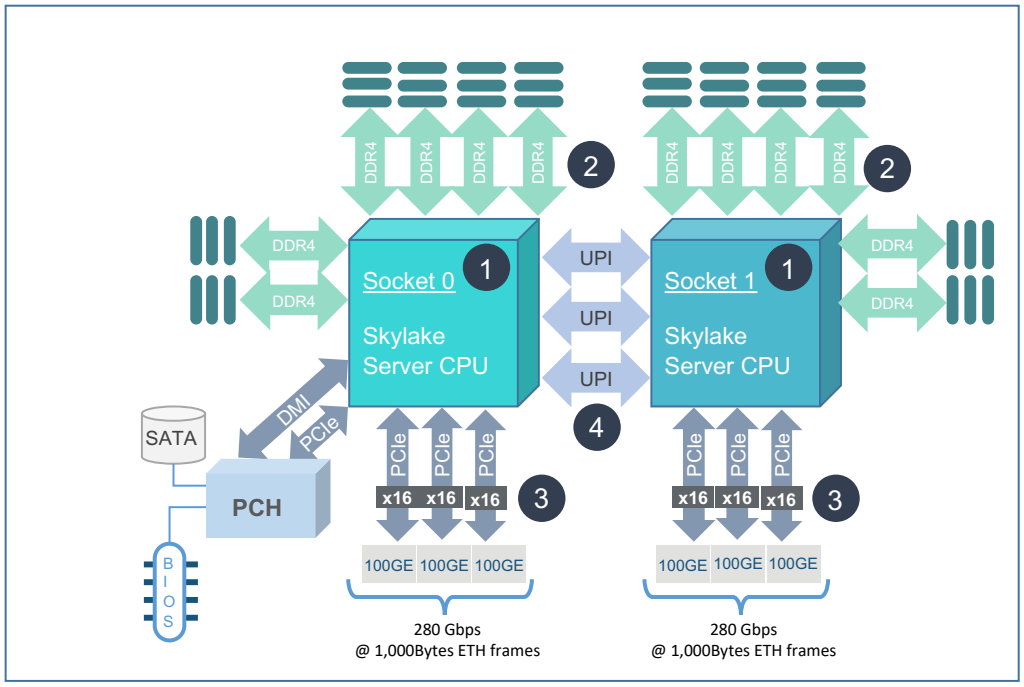

Figure 3. Main compute resources in two-socket server with Intel® Xeon® Scalable Processors.

Intel® Xeon® Skylake processors introduced a number of advancements benefiting performance of NF data plane applications, summarized below in comparison to Intel® Xeon® Broadwell processors[3]:

1) **Processor and CPU cores**
    a. Frontend: higher throughput instruction decoder, increased from 4- to 5-wide.
    b. Frontend: larger and improved branch predictor.
    c. Backend: L1 Data cache load bandwidth increased from 64Byte/cycle to 128B/c, and store bandwidth increased from 32B/c to 64 Byte/c.
    d. Backend: deeper load/store buffers, improved prefetcher.
    e. Backend: deeper out-of-order execution window to hide memory latency.
    f. Backend: improved scheduler and execution engine.
    g. Backend: Core L2 cache increased from 256K to 1MB per core.
    h. Backend: L3 cache (Last Level Cache) size is decreased from 2.5 MB to 1.375 MB per core. LLC is now non-inclusive.
    i. Uncore: topology change from ring to X-Y mesh for improved communication efficiency across up to 28 cores, Last Level Cache slices and IO Blocks.
2) **Memory bandwidth** – ~50% increase in memory bandwidth due to 1) memory channels are increased from four to six 2) support for higher speed memory from DDR4-2400 to DDR4-2666.
3) **I/O bandwidth** – increased I/O scalability from 40 to 48 lanes of PCIe Gen3. In addition, IO blocks are re-architected for delivering up to 50+% higher aggregate I/O bandwidth.
4) **Inter-socket bandwidth** – increased from 2 QPI to up to 3 UPI interfaces. Speed increased from 9.6 GTransactions/s to 10.4 GT/s per UPI interface.

Listed processor, CPU core and bandwidth related improvements in the Skylake architecture result in better IPC (instructions-per-cycle) ratio and improved generation to generation packet processing throughput for the benchmarked workloads. Although these workloads do not consume excessive memory bandwidth, improvements in the memory architecture indirectly contribute to higher performance due to better memory latencies under concurrent memory traffic.

One of the biggest improvements is the increase of network I/O bandwidth through, greatly increasing achievable network packet forwarding rate to 280 Gbps per socket.

[3] "The New Intel® Xeon® Scalable Processor (formerly Skylake-SP)", Akhilesh Kumar, https://www.hotchips.org/wp-content/uploads/hc_archives/hc29/HC29.22-Tuesday-Pub/HC29.22.90-Server-Pub/HC29.22.930-Xeon-Skylake-sp-Kumar-Intel.pdf

Although per processor socket network I/O increase, shown in *Figure 4,* has been verified in Cisco lab[4], it is not the focus of this paper.

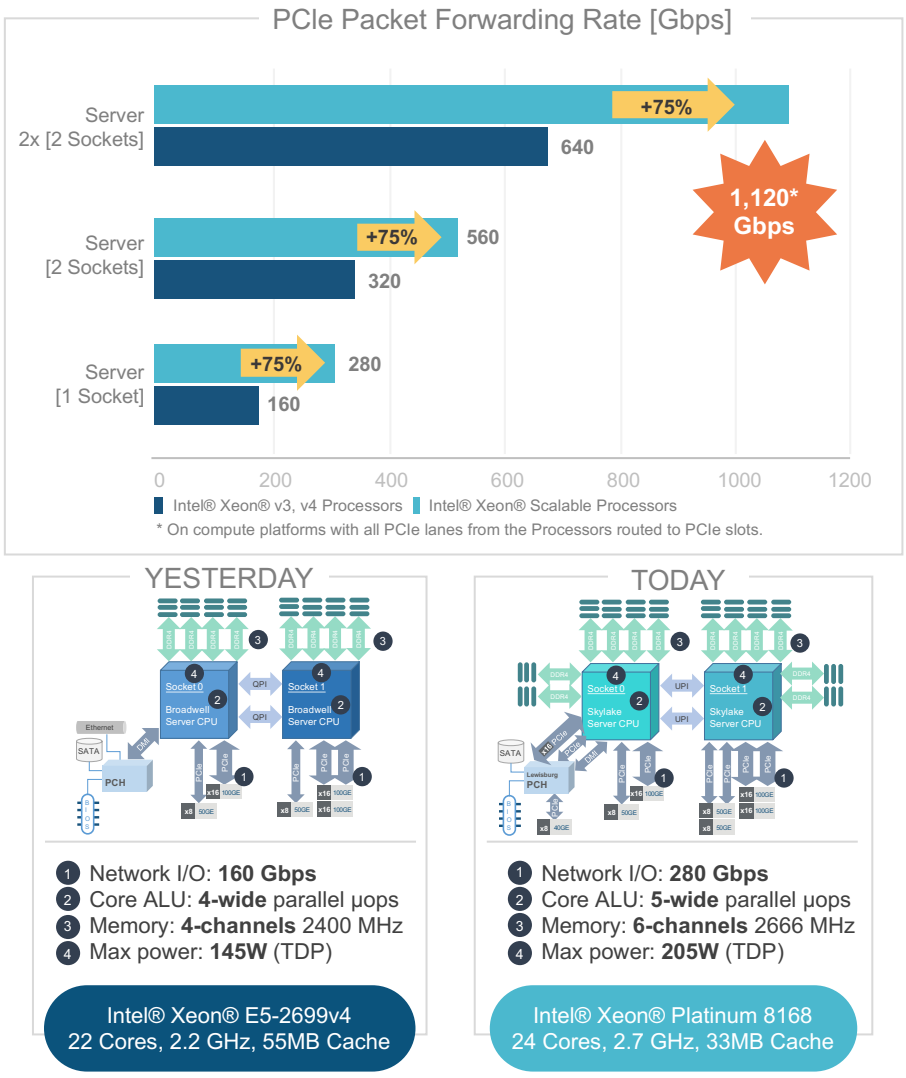

Figure 4. Increase of PCIe packet forwarding rate on Intel® Xeon® Skylake processors.

Following sections describe measured performance and efficiency improvements between Intel® Xeon® Broadwell and Skylake based servers for tested NF data plane applications, quantifying the impact of micro-architecture changes.

[4] 3min video clip, "FD.io: A Universal Terabit Network Dataplane", https://www.youtube.com/watch?v=aLJ0XLeV3V4 (Intel does not control or audit third-party data. You should review this content, consult other sources, and confirm whether referenced data are accurate.)

2.3 Performance Tests

2.3.1 Benchmarked Applications

Network data plane applications benchmarked for this paper are the same as the ones used in [BASWDP]. The same source software code versions got used, compiled to respective Intel® Xeon® processor micro-architectures.

Tested applications and benchmarked configurations are listed in increasing level of packet processing complexity in *Table 1*.

Idx	Application Name	Application Type	Benchmarked Configuration
1	EEMBC CoreMark®[5]	Compute benchmark	Runs computations in L1 core cache. Not a data plane applications, used here as a reference for compute efficiency.
2	DPDK Testpmd[6]	DPDK example	Baseline L2 packet looping, point-to-point.
3	DPDK L3Fwd	DPDK example	Baseline IPv4 forwarding, /8 entries.
4	FD.io VPP[7]	NF application	vSwitch with L2 port patch, point-to-point cross-connect.
5	FD.io VPP	NF application	vSwitch MAC learning and switching.
6	OVS-DPDK[8]	NF application	vSwitch with L2 port cross-connect, point-to-point.
7	FD.io VPP	NF application	vSwitch with IPv4 routing, /32 entries.

Table 1. NF data plane applications benchmarked in this paper.

First benchmark is chosen to compare pure compute performance against the rest of benchmarks having I/O as well.

Benchmarks 2. and 3. cover basic packet processing operations covering both I/O and compute aspects of the system. The packet processing functionalities increase with each benchmark order, and so do compute requirements.

Last four benchmarks, listed as 4. to 7. cover performance of a virtual switch, one of the most important ingredients in NF infrastructure. Virtual switch applications are tested in

[5] EEMBC CoreMark - http://www.eembc.org/index.php.

[6] DPDK testpmd - http://dpdk.org/doc/guides/testpmd_app_ug/index.html.

[7] FDio VPP – Fast Data IO packet processing platform, docs: https://wiki.fd.io/view/VPP, code: https://git.fd.io/vpp/.

[8] OVS-DPDK - https://software.intel.com/en-us/articles/open-vswitch-with-dpdk-overview.

L2 switching and IPv4 routing configurations, covering both different implementations and various packet switching scenarios.

2.3.2 Test Environment

2.3.2.1 Physical Topology

All benchmarking of x86 server with Intel® Xeon® Skylake processor has been conducted using a simple two-node topology with server System Under Test node and Ixia® Packet Traffic Generator node. Physical test topology is shown in *Figure 5*.

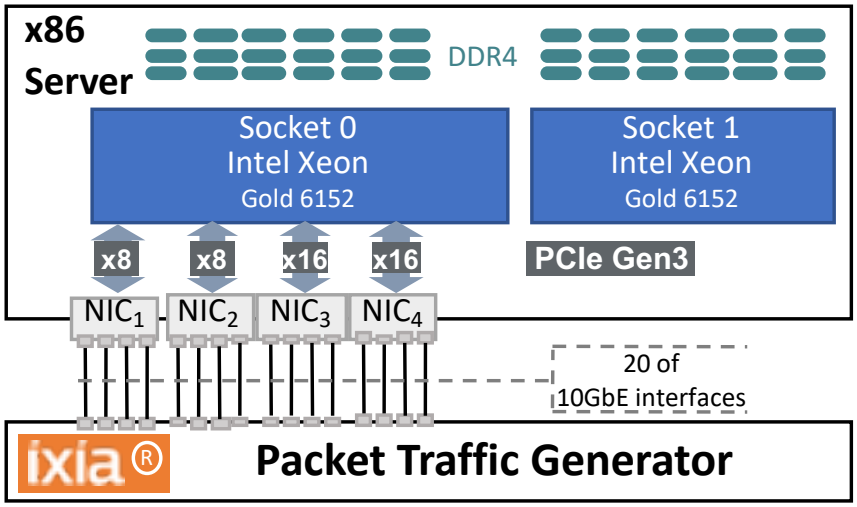

Figure 5. Physical Test Topology.

2.3.2.2 Server Configurations

Server with Intel® Xeon® Gold 6152 (formerly known as Skylake-SP) sku has been chosen for performing the tests and comparison with Intel® Xeon® Broadwell(E5-2699v4) tests described in [BASWDP].

Idx	Core Frequency	Core Density	Intel® Xeon® Processor Model
Server1	2.10 GHz	22C	Xeon Gold 6152 30.25MB 140W

Table 2. Benchmarked server processor specification.

The choice of the Skylake processor model was based on the similarities in the core count, frequency, and TDP range with E5-2699v4 (22 C, 2.2 GHz, 145W).

All applications run in user-mode on Linux.

The exact compute server specifications in terms of used Hardware[9] and Operating System[10] have been provided in *Section 8. Appendix: Test Environment Specification*.

[9] Hardware – Supermicro® server with Intel® Xeon® processors and Intel® NICs X710-DA4 4p10GE.

[10] Operating System – Linux 18.04 LTS (kernel version 4.15.0-36-generic).

NF applications' data planes are benchmarked while running on a single physical CPU core, followed by multi-core tests to measure performance speed-up when adding CPU core resources. In order to stay within the known PCIe Slot limits of the system, following multi-core and multi-10GbE port combinations have been chosen.

Number of 10 GbE ports used per test	
# of cores used / Benchmarked Workload	1 core
DPDK-Testpmd L2 Loop	4
DPDK-L3Fwd IPv4 Forwarding	4
VPP L2 Patch Cross-Connect	2
VPP L2 MAC Switching	2
OVS-DPDK L2 Cross-Connect	2
VPP IPv4 Routing	2

Table 3. Benchmark test variations for listed software applications.

Similarly to configuration used in [BASWDP], two main network I/O bottlenecks drove above choices are: i) 14.88 Mpps 10GbE linerate for 64B Ethernet frames, and ii) ~35.8 Mpps frame forwarding rate limit per used NIC cards (Intel® X710-DA4 4p10GbE). PCI Gen3 x8 and x16 slots' bandwidth has not been identified as a bottleneck in any of the benchmarks reported in this paper.

All tests are executed without and with hardware Symmetric Multi-Threading[11] using Intel® Hyper-Threading, with consistent mappings of threads to physical cores to 10GbE ports.

All tests are executed using CPU cores located on a single socket and using single NUMA node resources.

2.3.2.3 Traffic Generator and Offered Load

Ixia®[12] packet traffic generator was used for all tests. Purpose developed automation test tools used Ixia Python API for controlling the traffic generator and to ensure consistent execution across multiple test iterations.

Similarly to [BASWDP], configured network I/O packet load for the L2 tests involved 3,125 distinct (source, destination) MAC flows generated per interface, and highest scale of 50,000 flows for 16 of 10GbE interfaces. Each IPv4 test involved 62,500 distinct (source, destination) IPv4 flows per interface, and highest scale of 1,000,000 IPv4 flows. All flows were configured with 64Byte Ethernet L2 frame size.

[11] Symmetric Multi-Threading (SMT) – hardware-based parallel execution of independent threads to better utilize micro-architecture resources of CPU core.

[12] Other names and brands may be claimed as the property of others.

Details of packet traffic generator configuration have been provided in *Section 8. Appendix: Test Environment Specification*.

3 Results and Analysis

3.1.1 Measurements

Following tables show the test results for benchmarked NF applications including all identified high-level performance and efficiency metrics: i) Throughput *#packets/sec* [Mpps], ii) *#instructions/packet* (IPP), iii) *#instructions/cycle* (IPC) and iv) resulting *#cycles/packet* (CPP). EEMBC CoreMark® benchmark results are listed for comparison of CPU core usage metrics, more specifically *#instructions/cycle.* All results are presented in the same was as in [BASWDP]. However this time the focus is on comparing Intel® Xeon® Skylake vs. Broadwell performance and efficiency metrics.

All benchmarked NF applications focus on packet header processing, hence all benchmarks were conducted with smallest possible Ethernet frame size (64B) in order to stress compute resources and their interactions. Benchmarks were run on a single physical core with Intel Hyper-Threading disabled (marked as *noHT*) and enabled (marked as *HT*) to demonstrate the key performance and efficiency differences.

Summary results for tested processor Intel® Xeon® Skylake Gold 6152 2.1 GHz (*Table 4*) are compared with processor Intel® Xeon® Broadwell E5-2699v4 2.2 GHz (*Table 5*), with relative changes between the two (*Table 6*).

Benchmarked Workload	Throughput [Mpps]		#instructions /packet		#instructions /cycle		#cycles /packet	
Dedicated 1 physical core with =>	**noHT**	**HT**	**noHT**	**HT**	**noHT**	**HT**	**noHT**	**HT**
CoreMark [Relative to CMPS ref*]	0.99	1.35	n/a	n/a	2.54	3.44	n/a	n/a
DPDK-Testpmd L2 Loop	54.6	59.5	82	93	2.13	2.64	38	35
DPDK-L3Fwd IPv4 Forwarding	32.3	38.4	134	135	2.06	2.46	65	55
VPP L2 Patch Cross-Connect	23.0	28.1	223	224	2.45	2.99	91	75
VPP L2 MAC Switching	8.3	9.5	599	644	2.37	2.91	253	221
OVS-DPDK L2 Cross-Connect	7.2	10.9	539	500	1.85	2.59	292	193
VPP IPv4 Routing	12.8	14.8	425	438	2.59	3.09	164	142
*CoreMarkPerSecond reference value - score in the reference configuration: E5-2699v4, 1 Core noHT.								

Table 4. Benchmark measurements on Intel® Xeon® Skylake Gold-6152 2.1 GHz.

Benchmarked Workload	Throughput [Mpps]		#instructions /packet		#instructions /cycle		#cycles /packet	
Dedicated 1 physical core with =>	**noHT**	**HT**	**noHT**	**HT**	**noHT**	**HT**	**noHT**	**HT**
CoreMark [Relative to CMPS ref*]	1.00	1.33	n/a	n/a	2.43	3.34	n/a	n/a
DPDK-Testpmd L2 Loop	44.8	58.3	88	89	1.80	2.35	49	38
DPDK-L3Fwd IPv4 Forwarding	27.8	36.1	138	138	1.74	2.27	79	61
VPP L2 Patch Cross-Connect	19.3	23.3	208	218	1.83	2.30	114	94
VPP L2 MAC Switching	7.7	9.1	585	605	2.05	2.50	286	242
OVS-DPDK L2 Cross-connect	7.3	10.1	537	513	1.78	2.35	301	218
VPP IPv4 Routing	11.8	13.5	415	416	2.23	2.55	186	163
*CoreMarkPerSecond reference value - score in the reference configuration: E5-2699v4, 1 Core noHT.								

Table 5. Benchmark measurements on Intel® Xeon® Broadwell E5-2699v4 2.2 GHz.

Benchmarked Workload	Throughput [Mpps]		#instructions /packet		#instructions /cycle		#cycles /packet	
Dedicated 1 physical core with =>	**noHT**	**HT**	**noHT**	**HT**	**noHT**	**HT**	**noHT**	**HT**
CoreMark [Relative to CMPS ref*]	-1%	1%	n/a	n/a	4%	3%	n/a	n/a
DPDK-Testpmd L2 Loop	22%	2%	-7%	5%	18%	12%	-22%	-7%
DPDK-L3Fwd IPv4 Forwarding	16%	6%	-3%	-3%	18%	9%	-18%	-10%
VPP L2 Patch Cross-Connect	19%	21%	7%	3%	34%	30%	-20%	-21%
VPP L2 MAC Switching	8%	4%	2%	6%	16%	16%	-11%	-9%
OVS-DPDK L2 Cross-Connect	-1%	8%	0%	-3%	4%	10%	-3%	-12%
VPP IPv4 Routing	8%	10%	2%	5%	16%	21%	-12%	-13%
*CoreMarkPerSecond reference value - score in the reference configuration: E5-2699v4, 1 Core noHT.								

Table 6. Intel® Xeon® Skylake Gold 6152 2.1 GHz performance and other statistics relative to Intel® Xeon® Broadwell E5-2699v4 2.2 GHz.

Top-level observations comparing benchmark measurements of Skylake vs. Broadwell:

- Throughput (PPS): PPS gain improvements range from 8 to 22% for noHT and 2 to 21% for HT cases. Even though Skylake is running at lower frequency than Broadwell (100 Mhz or ~5%), PPS is consistently high due to better IPC for all workloads. However, OVS-DPDK shows small negative delta indicating only a minimal improvement at less or around the level of processor core frequency difference. DPDK-Testpmd with HT exhibits minimal improvement as the performance is reaching 4x10GbE line rate limit (59.5 Mpps).
- #Instructions/packet (IPP, Lower is better): This metric is indirectly calculated from PPS, and #instructions/cycle metric mentioned below. As expected, IPP remains the same for noHT mode, small decrease for HT. Note that negative value means Skylake executes less instructions per packet.
- #Instructions/cycle (IPC): IPC is measured from processors' performance monitoring counters and is indicative of execution efficiency of an architecture. This metric improved from 4% to 34% due to number of microarchitecture enhancements such has better frontend design, increased load/store bandwidth, efficient execution engines, and larger L2 cache. OVS-DPDK exhibits smallest IPC gain as compared to other workloads.
- #Cycles/packet (CPP, lower is better): This metric is calculated from the throughput and the core frequency. Similarly to throughput (as it is a dependent variable), reduction improvements range from -22%% to -3% for noHT setup, and -21% to -7% for HT setup - mainly resulting from improved #instructions/cycle metric. Two workloads show very small improvements obscured by ~5% frequency difference: i) DPDK-Testpmd and ii) OVS-DPDK.

3.1.2 First Analysis

Here are the initial observations of measured baseline performance and efficiency metrics, with special attention paid to the changes between tested Broadwell and Skylake processors.

3.1.2.1 Instructions-per-Packet

Instructions-per-Packet metric (IPP) depends on the number and type of packet processing operations required to realize a specific network function (or set of network functions), and how optimally they are programmed. The simpler the function, the smaller number of instructions per packet. This can be clearly seen in *Figure 6*, where simple applications/configurations show much lower instruction count compared to MAC switching or IPv4 routing.

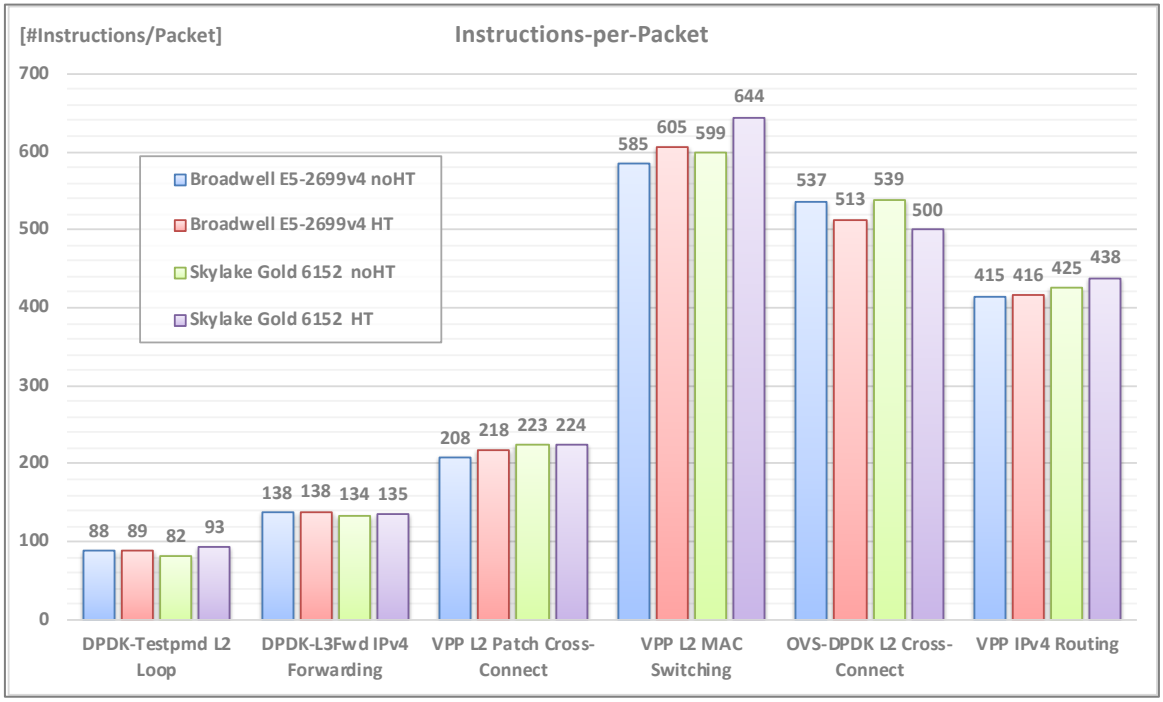

Figure 6. Number of instructions per packet for benchmarked applications.

From above benchmark results one can glean that number of instructions per packet does not really depend on processor microarchitecture. Due to the fact that all applications rely on polling of packets, some small variation in instructions-per-packet is expected.

3.1.2.2 Instructions-per-Cycle

Instructions-per-Cycle (IPC) is usually the first efficiency metric to analyse. The most common underlying reason behind the low value (i.e. below 2) is CPU core waiting for the data from various levels of cache or system memory. This especially applies to memory and I/O intensive programs like NF data planes as seen for some applications running in noHT setup, as seen in *Figure 7*.

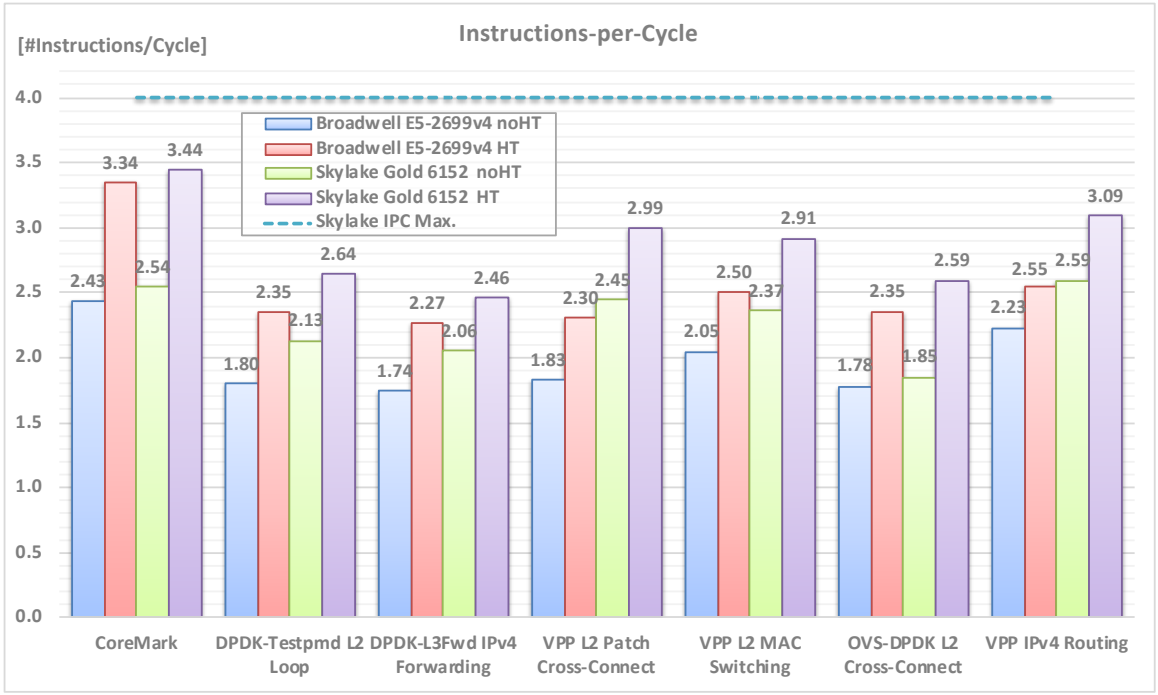

Figure 7. Number of instructions per core clock cycle for benchmarked applications.

From all benchmarked workloads, **CoreMark** scores the highest IPC of **3.44 with HT** (2.54 with noHT) on Skylake, as expected. The closest to CoreMark is **VPP IPv4 Routing** with IPC scores of **3.09** (2.59), with theoretical maximum being **4** per figure. The higher IPC indicates that the software do aggressive prefetching to bring data into L1/L2 caches keeping it ready for further processing. This relative scoring is very much aligned with what was observed on Broadwell per [performance_analysis_bdx_paper].

Comparing IPC results on Skylake vs. Broadwell processors, IPC increased for all applications, but the gain levels vary:

- Coremark: **3% with HT** (~ 4% with noHT).
- DPDK applications: **9% . .12%** (18%).
- OVS-DPDK: **10%** (4%).
- VPP configurations: **16% .. 30%** (16% .. 34%) - best gain among benchmarked applications.

3.1.2.3 Cycles-per-Packet

Cycles-per-Packet metric (CPP) is the direct measure of time spent by compute machine in processing a packet. Clearly software optimization techniques has been applied to all NF applications tested, as all of them measure good CPP values. CPP results for Skylake and Broadwell processors are presented in *Figure 8*.

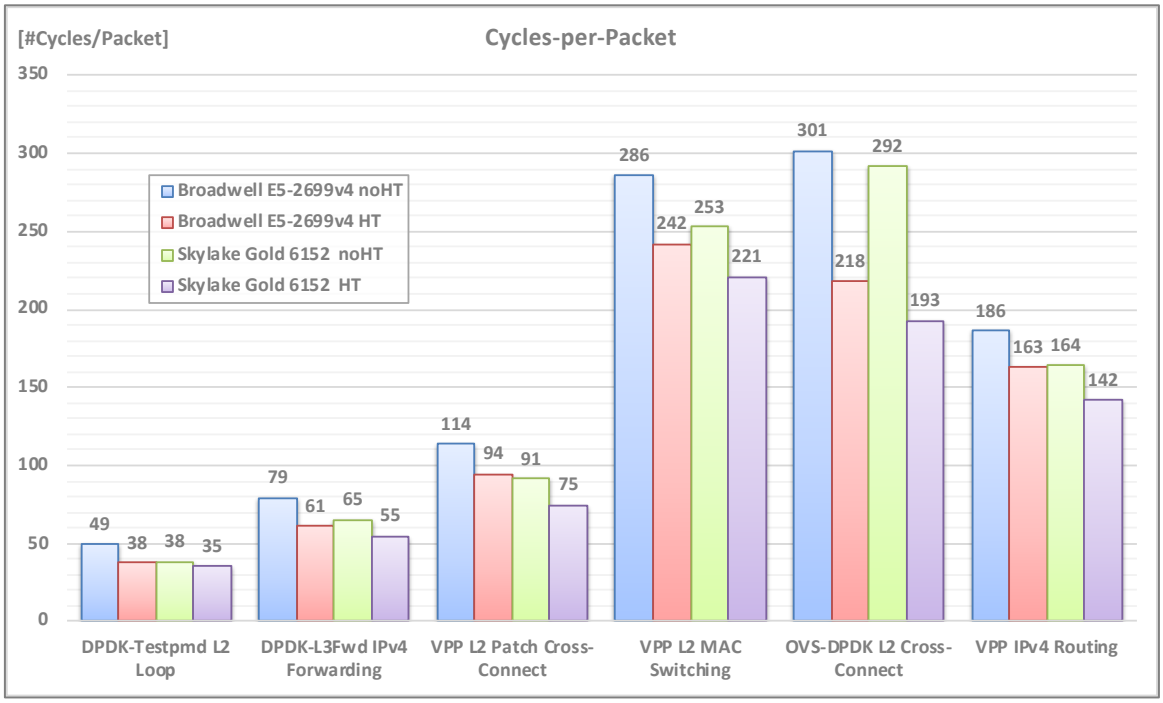

Figure 8. Number of core clock cycles per packet for benchmarked applications.

Comparing CPP results on Skylake vs. Broadwell tested processors, CPP decreased for all applications, but the reduction levels vary:

- DPDK applications: **-10% .. -7%** (-22% .. -18%).
- OVS-DPDK: **-12%** (-3%).
- VPP configurations: **-21% .. -9%** (-20% .. -11%) - best reduction among benchmarked applications.

3.1.2.4 Packets-per-Second Throughput

Measured packet throughput values [Mpps] are inversely proportional to reported CPP values, therefore the same observations noted for CPP equally apply here. Results for tested Skylake and Broadwell processors are presented in *Figure 9*.

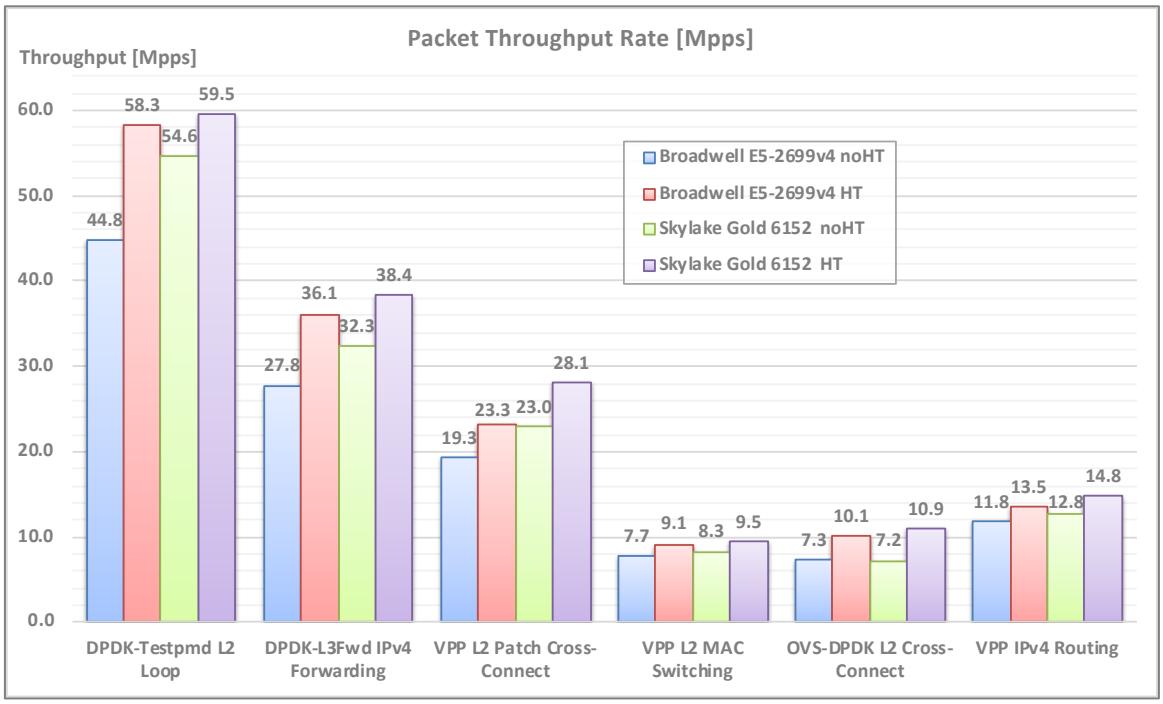

Figure 9. Packet Throughput Rate for benchmarked applications with a single core.

Reported packet throughput [Mpps] values are measured directly using Ixia® traffic generator. Comparing packet throughput results on Skylake vs. Broadwell tested processors, despite bit lower core frequency (2.1GHz vs. 2.2GHz respectively, throughput increased for all applications, but the gain levels vary:

- DPDK applications: **2% .. 6%** (16% .. 22%).
- OVS-DPDK: **8%** (-1%).
- VPP configurations: **4% .. 21%** (8% .. 19%) - best gain among benchmarked applications.

3.1.2.5 First Conclusions

From reported performance data and the initial observations, it is clear that all tested NF applications gained performance by running on Skylake processor compared to Broadwell. The amount of gain varies considerably, and based on the data it is clear that FD.io VPP configurations excelled in all measured and derived performance and efficiency metrics. FD.io VPP configurations consume significant processing power processing packet, therefore benefitting more from improvements in processor compute microarchitecture.

3.1.3 Throughput Speedup

3.1.3.1 Processor Core Frequency

One expects performance to proportionally scale up or down with processor core frequency. However this may not apply to processors with different micro-architectures, as the gains of micro-architecture enhancements may exceed gains/losses due to frequency change. And this is exactly what the packet throughput comparison between Skylake processor Intel® Xeon® Skylake Gold 6152 2.1 GHz and Broadwell processor Intel® Xeon® E5-2699v4 2.2 GHz shows in *Figure 10*.

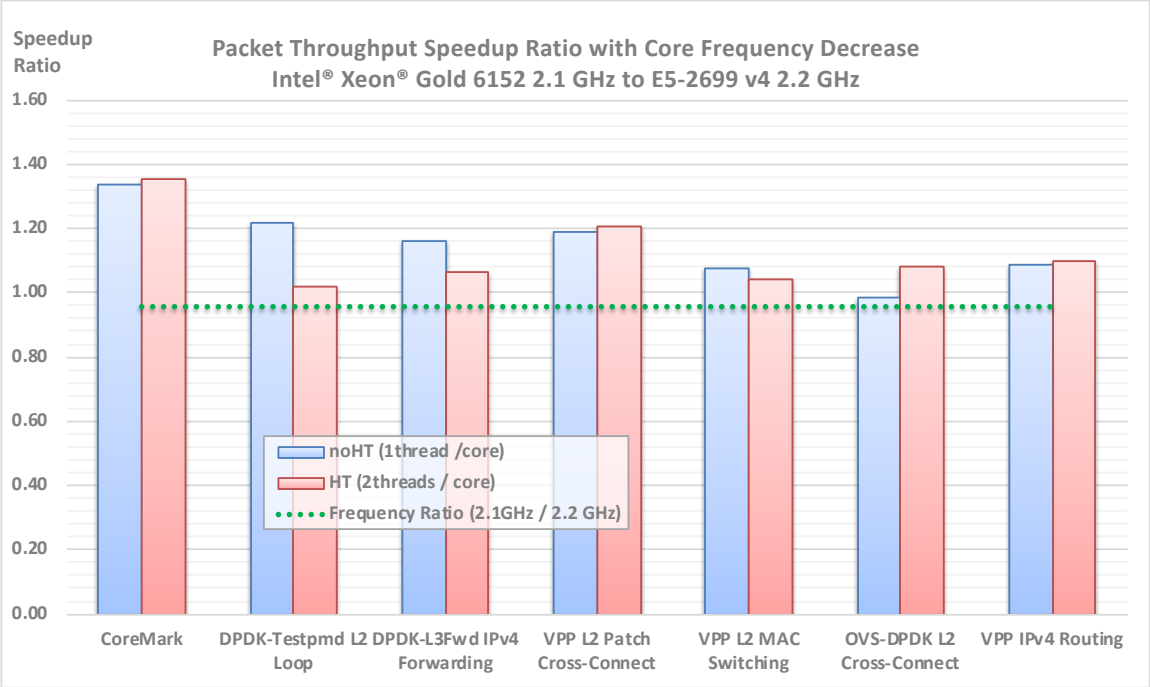

Figure 10. Packet throughput speedup with core frequency decrease.

In all cases gains of processor micro-architecture enhancements from Broadwell to Skylake cover for a small frequency decrease (2.1GHz / 2.2GHz = 0.95), with biggest gains observed for VPP with up to 1.20 speedup.

3.1.3.2 Intel Hyper-Threading

The performance change between Hyper-Thread and non-Hyper-Thread setups highly depends on the characteristics of the programs running on each thread. *Figure 11* shows the noHT-to-HT speedup for benchmarked applications.

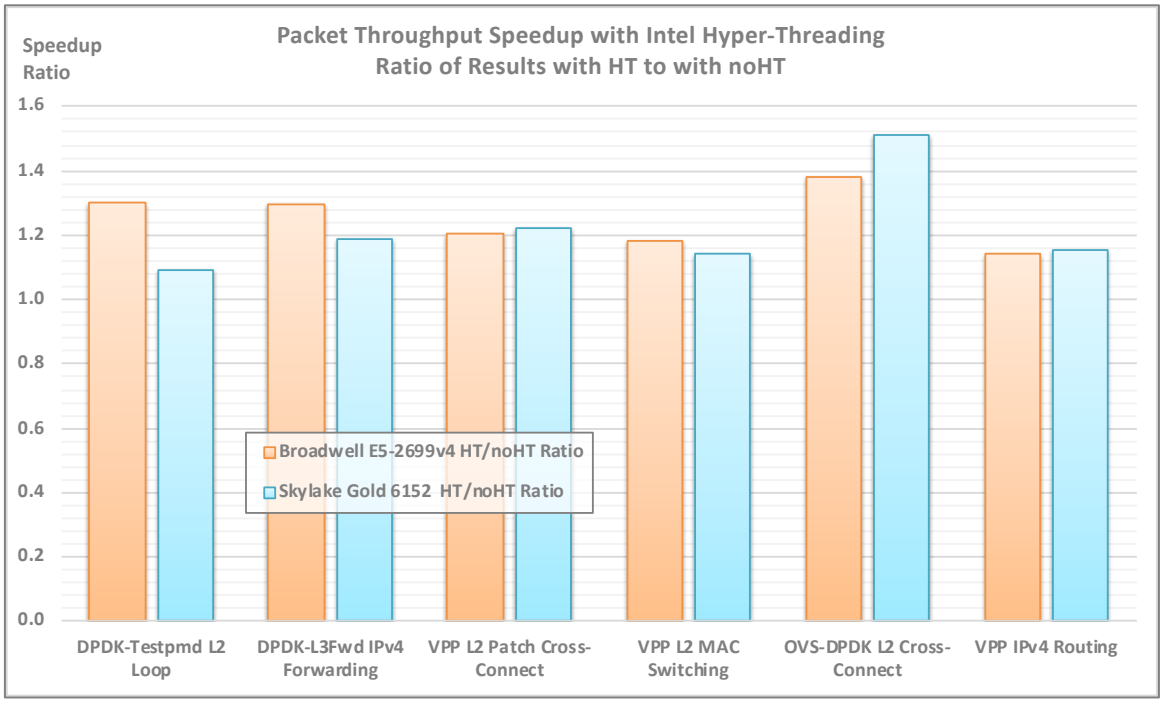

Figure 11. Packet throughput speedup with Intel Hyper-Threading.

Interestingly, HT/noHT throughput speedup is similar between Skylake and Broadwell for all tested applications, except the DPDK-Testpmd case, where throughput is throttled by 4x10GbE link rate of 59.5 Mpps.

In all cases HT/noHT speedup is within 1.2 .. 1.4 range.

Relative gain with HyperThreading enabled (HT/noHT ratio) varies for different workloads and can be explained for each processor architecture as follows:

- Broadwell: Single thread performance with noHT is lower due to Backend latency bound and HT mode recovers some of that performance by better hiding the Backend latency.
- Skylake: Single thread performance is better than on Broadwell due to Backend architecture improvements and Skylake Frontend architecture enhancements improve packet throughput even further in HT mode.

3.1.4 Further Analysis

Further analysis of performance test results and associated collected hardware performance counters data require deeper understanding of modern processor CPU micro-architecture and a well-defined interpretation approach for analysing underlying compute resource utilization and hotspots limiting program execution performance.

A good overview description of monitoring counters, their usage as well as list of performance monitoring tools with usage examples are provided in [BASWDP]. They equally apply to tested Skylake processors.

Further analysis for Skylake processors is provided with Intel Top-down Micro-architecture Analysis in the next section.

4 Top-down Microarchitecture Analysis (TMA)

4.1 Intel TMA Overview

Intel Top-down Microarchitecture Analysis[13] (TMA) has been developed and successfully applied to address the problem analysing and optimizing applications' performance without having to know increasing processor microarchitecture complexities and huge volumes of measurement data produced by performance tools. Top-down Microarchitecture Analysis aims to simplify performance analysis and eliminate any "guess work" from analysing performance counters.

For more details of how TMA applies to NF benchmarking please refer to [BASWDP].

Intel® Xeon® Skylake processors further extended performance monitoring unit (PMU) counters coverage and increased their accuracy making TMA accurate for Intel Hyper-Threading scenarios[14].

4.2 TMA Results and Interpretation

Graphical summaries of TMA measurements for NF applications benchmarked on Intel® Xeon® Skylake processors has been provided in *Figure 12* (HT) and *Figure 14* (noHT), and for visual comparison on Broadwell processors in *Figure* 13 (HT) and *Figure 15* (noHT) Figure 14.

[13] Top-down Microarchitecture Analysis through Linux perf and toplev tools, Haifa: C++ Meetup, 14th March 2018, http://www.cs.technion.ac.il/~erangi/TMA_using_Linux_perf__Ahmad_Yasin.pdf.

[14] Inside 6th-Generation Intel Core: New Microarchitecture Code-Named Skylake, 2017 IEEE, https://ieeexplore.ieee.org/document/7924286/.

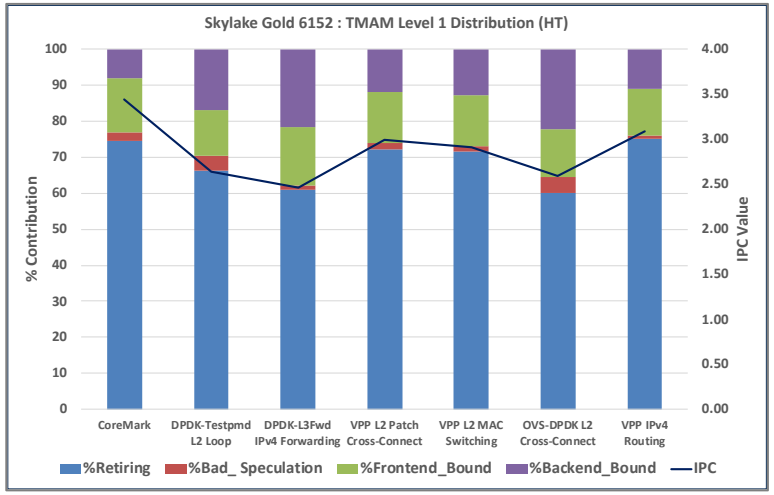

Figure 12. TMA Level-1 Metrics: Xeon Skylake with HT.

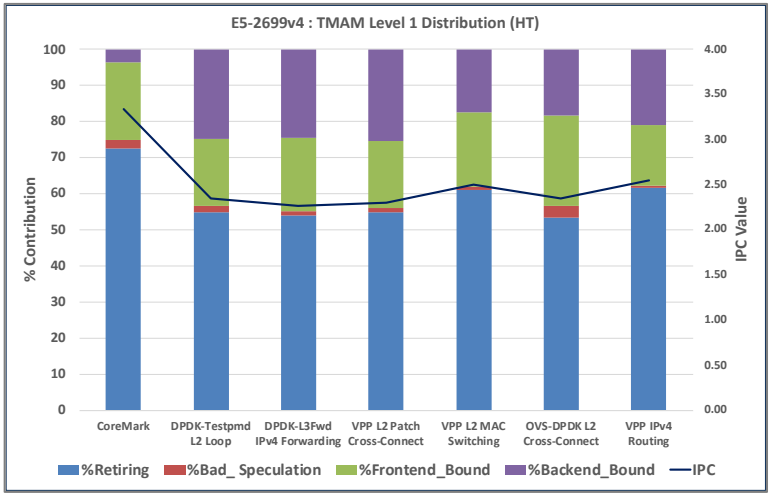

Figure 13. TMA Level-1 Metrics: Xeon Broadwell with HT.

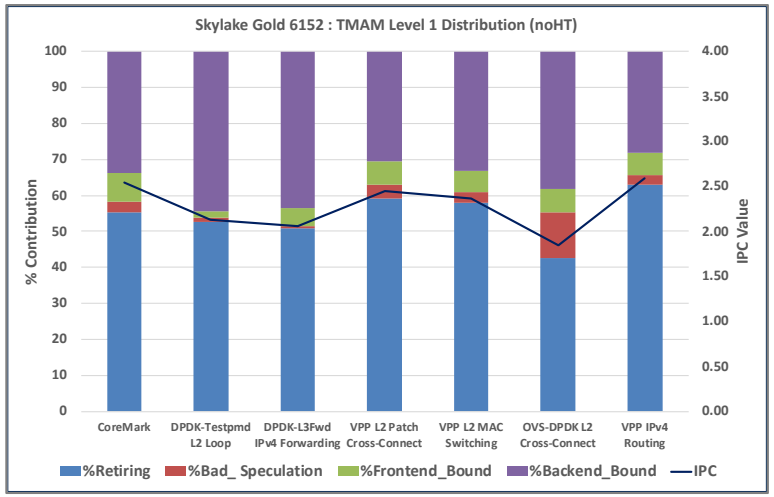

Figure 14. TMA Level-1 Metrics: Xeon Skylake with noHT.

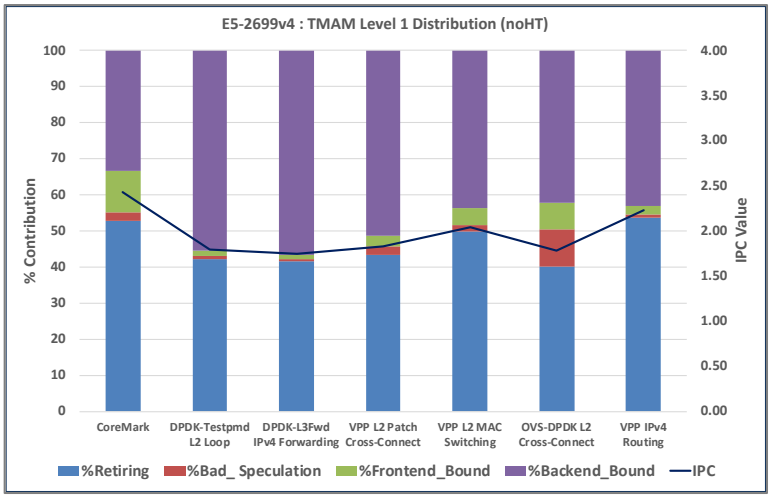

Figure 15. TMA Level-1 Metrics: Xeon Broadwell with noHT.

Core Pipeline Slots	Not Stalled				Stalled			
TMAM Level-1 Metrics	%Retiring		%Bad_ Speculation		%Frontend_Bound		%Backend_Bound	
Processor Mode: noHT, HT	noHT	HT	noHT	HT	noHT	HT	noHT	HT
CoreMark	55.2	74.6	3.0	2.2	8.1	15.2	33.7	8.0
DPDK-Testpmd L2 Loop	52.6	66.4	1.3	3.9	1.7	12.8	44.4	16.9
DPDK-L3Fwd IPv4 Forwarding	51.0	61.1	0.5	1.2	4.9	16.0	43.6	21.7
VPP L2 Patch Cross-Connect	59.0	72.2	3.9	1.6	6.6	14.2	30.6	11.9
VPP L2 MAC Switching	58.0	71.7	2.9	1.5	6.0	14.1	33.1	12.7
OVS-DPDK L2 Cross-Connect	42.7	60.1	12.7	4.3	6.3	13.4	38.3	22.3
VPP IPv4 Routing	63.0	75.2	2.5	0.8	6.2	12.9	28.3	11.1

Table 7. Intel® Xeon® Skylake Gold 6152 2.1 GHz TMAM Level-1 metrics.

Core Pipeline Slots	Not Stalled				Stalled			
TMAM Level-1 Metrics	%Retiring		%Bad_ Speculation		%Frontend_Bound		%Backend_Bound	
Processor Mode: noHT, HT	noHT	HT	noHT	HT	noHT	HT	noHT	HT
CoreMark	52.9	72.6	2.4	2.2	11.3	21.7	33.4	3.5
DPDK-Testpmd L2 Loop	42.1	54.9	1.2	1.8	1.4	18.5	55.4	24.8
DPDK-L3Fwd IPv4 Forwarding	41.5	54.1	0.6	1.0	1.3	20.4	56.6	24.6
VPP L2 Patch Cross-Connect	43.4	54.9	2.4	1.1	2.9	18.8	51.3	25.2
VPP L2 MAC Switching	49.9	61.1	1.7	0.9	4.7	20.7	43.7	17.3
OVS-DPDK L2 Cross-Connect	40.1	53.3	10.3	3.4	7.5	24.8	42.1	18.4
VPP IPv4 Routing	53.7	61.5	1.0	0.6	2.3	16.8	43.0	21.1

Table 8. Intel® Xeon® E5-2699v4 2.2 GHz TMAM Level-1 metrics.

Core Pipeline Slots	Not Stalled				Stalled			
TMAM Level-1 Metrics Change from Bdx to Skx	%Retiring (+) change is		%Bad_ Speculation (-) change is		%Frontend_Bound (-) change is		%Backend_Bound (-) change is	
Processor Mode: noHT, HT	noHT	HT	noHT	HT	noHT	HT	noHT	HT
CoreMark	2.33	2.02	0.57	-0.01	-3.25	-6.46	0.35	4.45
DPDK-Testpmd L2 Loop	10.53	11.48	0.14	2.14	0.31	-5.66	-10.98	-7.96
DPDK-L3Fwd IPv4 Forwarding	9.45	6.96	-0.07	0.24	3.63	-4.32	-13.01	-2.88
VPP L2 Patch Cross-Connect	15.64	17.34	1.43	0.51	3.66	-4.57	-20.73	-13.28
VPP L2 MAC Switching	8.13	10.65	1.16	0.58	1.25	-6.61	-10.54	-4.62
OVS-DPDK L2 Cross-Connect	2.60	6.80	2.40	0.84	-1.14	-11.45	-3.86	3.81
VPP IPv4 Routing	9.30	13.69	1.49	0.18	3.94	-3.94	-14.73	-9.93

Table 9. Intel® Xeon® Skylake Gold 6152 2.1 GHz TMAM Level-1 metrics incremental(+)/decremental(-) change from Intel® Xeon® E5-2699v4 2.2 GHz.

Comparing TMA Level-1 Skylake vs. Broadwell, specifically for more performant HT scenario as it uses the processor hardware in optimal manner for workloads tested:

1. **%Retiring:** reflects the ratio of core pipeline slots the µOps are successfully executed and retired, relative to the maximum possible, higher value is better. Higher value of metric results in higher IPC. Note that gain in this metric means reduction in one or more TMAM-L1 metrics.
    a. CoreMark: Both Skylake and Broadwell attains very high %Retiring ratio (~73% .. ~75%) and achieve about 19 points gain with HT vs. noHT.
    b. DPDK applications: This metric is up to ~11 points better for Skylake. However, having better value does not necessarily result in proportional

packet processing rate, as busy polling operation may artificially make this metric better without doing meaningful work.

c. OVS-DPDK: This metric is marginally better on Skylake indicating only a marginal improvement in performance.

d. VPP configurations: This metric is 17 points higher for L2 Cross-Connect and 13 points higher for IPv4 Routing, yielding 20% and 10% higher throughput respectively. These major gains come from the reduction in %Backend_bound metric (-13 and -3 points respectively) due to a number of Skylake Backend improvements including increased L1 bandwidth, deeper load/store buffers, and larger L2 cache caching large routing table.

2. **%Bad_Speculation:** represents the ratio of core pipeline slots pre-fetching and executing non-useful operations, lower value is better.

   This metric remains almost the same for both the architectures across all workloads.

3. **%Frontend_Bound:** captures the ratio of core pipeline slots the Frontend fails to supply the pipeline at full capacity, while there are no Backend stalls, lower value is better.

   This metric is improved in Skylake for all the workloads running in HT mode. Skylake microarchitecture frontend enhancements such as higher throughput instruction decoder and improved branch predictor make this metric better**.**

   a. CoreMark: Skylake has this metric reduced by more than 6 points, improving %Retiring metric.

   b. DPDK applications: Metric lower by ~5 points due to Frontend improvements.

   c. OVS-DPDK: Metric lower by 11 points due to Frontend improvements.

   d. VPP configurations: Metric lower by 4 to 6 points due to Frontend improvements.

4. **%Backend_Bound:** represents the ratio of core pipeline slots the µOps are not delivered from µOp queue to the pipeline due to Backend being out of resources to accept them, lower value is better.

   This metric for Skylake shows improvements backend improvements for all cases on networking workloads with exception of OVS-DPDK with HT.

   a. CoreMark: Skylake has this metric is increased by 4 points indicating that with frontend getting more efficient, the bottleneck has shifted to backend from the Broadwell architecture.

   b. DPDK applications: Metric lower by 3 to ~8 points.

   c. OVS-DPDK: Metric higher by 3 points.

   d. VPP configurations: Metric lower by 4 to 14 points due to a large number of Backend improvements.

5 Conclusions

There is no doubt that software data plane performance and efficiency are foundational properties underpinning any NFV and cloud networking system design. High performance communication matters, especially in the emerging scaled-out and scaled-up cloud-native micro-services deployments. Benchmarking and analysing this performance is the key to only understanding and comparing the key performance and efficiency metrics. But equally important is identifying the next set of hotspots in the hardware/software stack in a methodical and consistent manner, and providing feedback to the industry and involve communities to address them. And then verifying progress by re-running the benchmarks.

Continuing from [BASWDP], this paper applied the same benchmarking and analysis methodology to the same set of software applications on newer Intel Xeon Skylake processors, to compare performance and efficiency, and to quantify the gains of the Skylake processor architecture improvements.

Presented and discussed benchmarking data points out, that in addition to increasing per processor I/O throughput from 160 Gbps to 300 Gbps, number of other Skylake processor improvements across Frontend, Backend, Uncore blocks result in substantial software execution efficiency gains and higher data plane performance.

These gains are significant for multiple reasons. The main generic one is that software native data planes naturally and inherently benefit from the "Moore's law", gaining from every processor generation providing higher density of logic gates. Benchmark data shows that! But digging deeper, one finds that this increased gate capacity is distributed across a number of processor blocks, not only increasing the raw compute execution engine capacity, but also addressing bottlenecks in the in-order pre-processing (Frontend processing), out-of-order parallel execution (Backend processing) and storage units (Backend cache hierarchy). And thanks to increased market demand for SDN and NFV applications, also increasing I/O bandwidth capacity, directly benefiting networking and packet processing designs.

Identifying bottlenecks in a complex hardware/software stack is not easy due to a sheer volume of technologies involved, depth of the stack and very tight timing constraints often preventing the tools to provide a good view of the actual run-time behaviour. This paper, together with its pre-sequel, proves that Intel TMA (Top-down Microarchitecture Analysis) method combined with Intel PMU technology addresses this challenge, yielding reliable data. Authors of this paper found TMA and associated tools (i.e. pmu_tools[15]) an extremely useful for understanding the behaviour, performance and efficiency of benchmarked software data plane applications, validating TMA applicability in the SDN/NFV space. Furthermore, TMA did also provide a good insight into the run-time workings of the processor architecture itself and its blocks, enabling fairly straightforward interpretation of run-time processor hardware performance data collected during benchmarks and correlating those measurements to what is expected, quantifying the impact of processor and CPU core hardware improvements.

[15] Linux PMU-tools, https://github.com/andikleen/pmu-tools.

Authors hope that technical community with vested interest in SDN/NFV/Cloud-native network technologies find this paper of interest. They will be delighted to get any feedback about if and how this work benefits the real world designs and applications, and what should be added to become more relevant to the actual SDN/NFV/Cloud-native networking designs and deployments. Authors do have plans for a sequel paper/report addressing multi-core scaled-up scenarios and adding virtual interfaces (e.g. virtio/vhost-user for VMs, FD.io memif for Containers) involving memory copy operations. It is also authors' desire to continue the benchmarking comparisons for newer versions of both software data plane applications and processors, as they become available.

6 Acknowledgements

The authors would like to acknowledge Patrick Lu (former employee of Intel Corporation) for his contributions. They also thank Ray Kinsella and Kannan Ramia Babu (both Intel Corporation), Dave Barach and Jerome Tollet (both Cisco), Marco Varlese (SUSE), for their reviews and helpful for providing feedback for this paper.

7 References

[1] [BASWDP] “Benchmarking and Analysis of Software Data Planes”, Maciek Konstantynowicz, Patrick Lu, Shrikant M. Shah, December 2017, https://fd.io/wp-content/uploads/sites/34/2018/01/performance_analysis_sw_data_planes_dec21_2017.pdf.

[2] “The New Intel® Xeon® Scalable Processor (formerly Skylake-SP)”, Akhilesh Kumar, https://www.hotchips.org/wp-content/uploads/hc_archives/hc29/HC29.22-Tuesday-Pub/HC29.22.90-Server-Pub/HC29.22.930-Xeon-Skylake-sp-Kumar-Intel.pdf

[3] Video clip, "FD.io: A Universal Terabit Network Dataplane", https://www.youtube.com/watch?v=aLJ0XLeV3V4

[4] EEMBC CoreMark - http://www.eembc.org/index.php.

[5] DPDK testpmd - http://dpdk.org/doc/guides/testpmd_app_ug/index.html.

[6] FDio VPP – Fast Data IO packet processing platform, docs: https://wiki.fd.io/view/VPP, code: https://git.fd.io/vpp/.

[7] OVS-DPDK - https://software.intel.com/en-us/articles/open-vswitch-with-dpdk-overview.

[8] Top-down Microarchitecture Analysis through Linux perf and toplev tools, Haifa: C++ Meetup, 14th March 2018, http://www.cs.technion.ac.il/~erangi/TMA_using_Linux_perf__Ahmad_Yasin.pdf.

[9] Inside 6th-Generation Intel Core: New Microarchitecture Code-Named Skylake, 2017 IEEE, https://ieeexplore.ieee.org/document/7924286/.

[10] Linux PMU-tools, https://github.com/andikleen/pmu-tools.

8 Appendix: Test Environment Specification

8.1 System Under Test – Intel® Xeon® Skylake-SP HW Platform Configuration

Mother Board	Intel® Purely Customer reference board
Processor	Intel® Xeon® Gold 6152, Dual Socket configuration
Memory	DDR4-2666, 1 DIMM per channel, 6 Channels for each socket
BIOS Version	PLYDCRB1.86B.0155.R08.1806130538,06/13/2018, Microcode 0x200004d
Network Cards	X710-DA4 quad 10 Gbe Port cards, 2 cards total

8.2 System Under Test and Tested Applications – Intel® Xeon® Skylake-SP Software Versions

Linux OS Distribution	Ubuntu 18.04.1 LTS x86_64
Kernel Version	4.15.0-36-generic
Fortville firmware version	fw 6.0.48442 api 1.7 nvm 6.01 0x80003484 1.1747.0
DPDK Version	DPDK v18.11
VPP Version	v18.10-release
QEMU version	2.11.1
OVS version	2.10.1
Guest OS and kernel	Ubuntu 16.04.1 LTS x86_64

8.3 Skylake-EP Server BIOS Settings

Menu (Advanced)	BIOS Submenu Items	BIOS Settings Used for the tests	BIOS Default
CPU Configuration:	Hyper-Threading (ALL)	Disable	Enable
Socket Configuration -> Advanced Power Management Configuration -> CPU P State Control	SpeedStep (Pstates)	Disable	Enable
	Turbo Mode	Disable	Enable
	Energy Efficient Turbo	Disable	Enable
Socket Configuration -> Advanced Power Management Configuration -> Hardware PM State Control	Hardware P-States	Disable	Native Mode
Socket Configuration -> Advanced Power Management Configuration -> CPU C State Control	Autonomous Core C-State	Disable	Enable
	CPU C6 Report	Disable	Enable
	Enhanced Halt State (C1E)	Enable	Enable
Socket Configuration -> Advanced Power Management Configuration -> Package C State Control	Package C State	<C0/C1 state>	Auto
Socket Configuration -> Advanced Power Management Configuration -> CPU – Advanced PM Tuning	Energy Perf BIAS -> Power Performance Tuning	<BIOS Controls EPB>	<OS Controls EPB>
	Energy Perf BIAS -> ENERGY_PERF_BIAS_CFG mode	Performance	Balanced Performance
Socket Configuration -> IIO Configuration	PCIe ASPM	Disable	Enable
	Intel VT for Directed I/O (VT-d)	Disable	Enable
Socket Configuration -> UPI Configuration	Link L0 P	Disable	Enable
	Link L1	Disable	Enable

Socket Configuration -> Memory Configuration	Enforce POR	Disable	Auto
	Memory Map -> IMC Interleaving	2-way Interleave	Auto

8.4 System Under Test – Intel® Xeon® Broadwell HW Platform Configuration

Mother Board	SuperMicro® X10DRX
Processor	Intel® Xeon® E5-2699v4, Dual Socket configuration
Memory	DDR4-2400, 1 DIMM per channel, 4 Channels for each socket
BIOS Version	3.0a, 02/08/2018, Microcode: 0x200004d
Network Cards	X710-DA4 quad 10 Gbe Port cards, 2 cards total

8.5 System Under Test and Tested Applications – Intel® Xeon® Broadwell Software Versions

Linux OS Distribution	Ubuntu 18.04.1 LTS x86_64
Kernel Version	4.15.0-36-generic
Fortville firmware version	fw 6.0.48442 api 1.7 nvm 6.01 0x80003484 1.1747.0
DPDK Version	DPDK v18.11
VPP Version	v18.10-release
QEMU version	2.11.1
OVS version	2.10.1
Guest OS and kernel	Ubuntu 16.04.1 LTS x86_64

8.6 Intel® Xeon® Broadwell Server BIOS Settings

Menu (Advanced)	BIOS Submenu Items	BIOS Settings Used for the tests	BIOS Default
CPU Configuration: Advanced Power Management Configuration	Hyper-Threading (ALL)	Disable	Enable
	Power Technology	Disable	Custom
	Energy Performance Tuning	Disable	Enable
	Energy Performance BIAS Setting	Performance	Enable
	Energy Efficient Turbo	Disable	Enable
-> CPU P State Control	EIST (P-States)	Disable	Enable
	Turbo Mode	Disable	Enable
	P-State Coordination	HW_ALL	HW_ALL
-> CPU C State Control	Package C State Limit	[C0/C1 State]	[C6 (Retention)]
	CPU C3 Report	Disable	Enable
	CPU C6 Report	Disable	Enable
	Enhanced Halt State (C1E)	Disable	Enable
Chipset Configuration: North Bridge -> IIO Configuration	EV DFX Features	Enable	Disable
	Intel VT for Directed I/O (VT-d)	Disable	Enable
Chipset -> North Bridge -> QPI Configuration	Link L0 P	Disable	Enable
	Link L1	Disable	Enable
	COD Enable	Disable	Auto
	Early Snoop	Disable	Auto

	Isoc Mode	Disable	Disable
-> North Bridge ->Memory Configuration	Enforce POR	Disable	Auto
	Memory Frequency	2400	Auto
	DRAM RAPL Baseline	Disable	Auto
	A7 Mode	Enable	Enable
-> South Bridge	EHCI Hand-off	Disable	Auto
	USB3.0 Support	Disable	Enable
PCIe/PCI/PnP Configuration	ASPM	Disable	Enable
	Onboard LAN 1 OPROM	Disable	PXE

8.7 Packet Traffic Generator – Configuration

Traffic Generator	Ixia® Traffic Generator

Throughput Test	Ixia® Quick Test: throughput rate search for finding zero-frame loss packet throughput in compliance with RFC 2544
Search algorithm	Binary search.
Starting condition	100% of link rate.
Stopping condition	Search finds the <0.01% loss rate packet throughput and exceeds minimum rate change value.
Number of test trials per each search step	8.
Test trial duration	20 seconds.
Allowed packet loss	<0.01%.
Minimum rate change value	0.1 Mpps.

Test	Ixia packet flow definitions
All L2 Ethernet tests	3,125 distinct flows transmitted per interface.

	Each distinct flow with unique tuple of (Source_MAC_Address, Destination_MAC_Address).
All L3 IPv4 tests	62,500 distinct flows transmitted per interface.
	Each distinct flow with unique tuple of (Source_IPv4_Address, Destination_ IPv4_Address).
Common to all tests	Both packet header source and destination address fields incremented pairwise by 1 in a packet-by-packet sequence.
	Continuous packet flows at fixed rate, with packets equally spaced in time, no bursts.
	Single Ethernet frame size of 64B including Ethernet FCS, smallest standard Ethernet frame possible with IPv4 payload.

9 Index: Figures

Figure 1. Baseline NF data plane benchmarking topology. 7

Figure 2. Main compute resources in two-socket server with Intel® Xeon® Broadwell Processors. 9

Figure 3. Main compute resources in two-socket server with Intel® Xeon® Scalable Processors. 9

Figure 4. Increase of PCIe packet forwarding rate on Intel® Xeon® Skylake processors. 11

Figure 5. Physical Test Topology. 13

Figure 6. Number of instructions per packet for benchmarked applications. 18

Figure 7. Number of instructions per core clock cycle for benchmarked applications. ... 19

Figure 8. Number of core clock cycles per packet for benchmarked applications..... 20

Figure 9. Packet Throughput Rate for benchmarked applications with a single core. 21

Figure 10. Packet throughput speedup with core frequency decrease. 22

Figure 11. Packet throughput speedup with Intel Hyper-Threading. 23

Figure 12. TMA Level-1 Metrics: Xeon Skylake with HT. Figure 13. TMA Level-1 Metrics: Xeon Broadwell with HT. 25

Figure 14. TMA Level-1 Metrics: Xeon Skylake with noHT. Figure 15. TMA Level-1 Metrics: Xeon Broadwell with noHT. 25

10 Index: Tables

Table 1. NF data plane applications benchmarked in this paper. 12

Table 2. Benchmarked server processor specification. 13

Table 3. Benchmark test variations for listed software applications. 14

Table 4. Benchmark measurements on Intel® Xeon® Skylake Gold-6152 2.1 GHz. 16

Table 5. Benchmark measurements on Intel® Xeon® Broadwell E5-2699v4 2.2 GHz. . 17

Table 6. Intel® Xeon® Skylake Gold 6152 2.1 GHz performance and other statistics relative to Intel® Xeon® Broadwell E5-2699v4 2.2 GHz. 17

Table 7. Intel® Xeon® Skylake Gold 6152 2.1 GHz TMAM Level-1 metrics. 26

Table 8. Intel® Xeon® E5-2699v4 2.2 GHz TMAM Level-1 metrics. 26

Table 9. Intel® Xeon® Skylake Gold 6152 2.1 GHz TMAM Level-1 metrics incremental(+)/decremental(-) change from Intel® Xeon® E5-2699v4 2.2 GHz. 26

END OF DOCUMENT