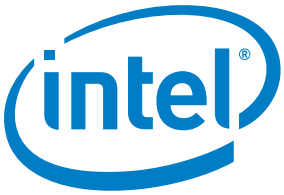


Telco/Cloud Enablement for 2nd Generation Intel® Xeon® Scalable platform – Intel® QuickAssist Technology

Application Note

April 2019

Document Number: 338929-001

Contents

Figures

Tables

Revision History

Date	Revision	Description
April 2019	001	Initial release.

1.0 Introduction

Prior generations of Intel® Xeon® platforms offered Intel® QuickAssist Technology either as part of Intel® Communications Chipset or via PCIe cards. For both the Intel® Xeon® Scalable platform and the 2nd generation Intel® Xeon® Scalable platform (formerly codenamed Purley Refresh), Intel® QuickAssist Technology is integrated as part of the server chipset. Compute-intensive workloads (compression and crypto) commonly used by many data center workloads now have the option to scale through software processing and/or with the use of Intel® QuickAssist Technology hardware acceleration.

This paper describes the Intel® QuickAssist Technology capabilities, examines the telecoms use cases that can potentially take advantage of the technology, and describes how these can be exposed to the customer through software enablement.

This document is part of the Network Transformation Experience Kit, which is available at: https://networkbuilders.intel.com/

1.1 Terminology

Table 1. Terminology

Term	Description
DPDK	Data Plane Development Kit
HCI	Hyper-Converged Infrastructure
NFV	Network Functions Virtualization
PCH	Platform Controller Hub
PF	Physical Function
SDI	Software Defined Infrastructure
SHVS	Standard High Volume Server
SKU	Stock Keeping Unit
SLA	Service Level Agreement
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VF	Virtual Function

Term	Description
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function

1.2 Reference Documents

Table 2. Reference Documents

Document	Location
Intel® QuickAssist Technology Overview	Overview: www.intel.com/content/www/us/en/embedded/technology/quickassist/overview.html
Intel® QuickAssist Technology Software Package	01.org: https://01.org/packet-processing/intel%C2%AE-quickassist-technology-drivers-and-patches?wapkw=quickassist
Intel Device Plugins for Kubernetes Application Note	https://builders.intel.com/docs/networkbuilders/intel-device-plugins-for-kubernetes-appnote.pdf

2.0 Intel® QuickAssist Technology

2.1 Overview

A few compute-intensive workloads are common to many data center infrastructure functions involving networking, storage, and compute. These common compute-intensive workloads include:

- **Symmetric cryptography**, also known as *bulk cryptography*, is used to secure data in flight or data at rest, thereby providing data confidentiality, integrity, and authentication. This is used in networking in applications such as IPSec gateways and SSL/TLS applications, including secure web servers, SSL proxies, load balancers, Application Delivery Controllers, and so forth. It is also useful in storage applications.
- **Asymmetric cryptography**, also known as *public key cryptography*, is typically used to perform a key exchange between parties that then use the derived keys to perform the bulk cryptography later. For example, the SSL/TLS protocols perform one or more public key cryptographic operations as part of an SSL handshake at the beginning of each new connection.
- **Compression** is used to reduce the size of data in flight or data at rest, thereby saving on the cost and/or latency to transmit the data over the network, or to read the data from, or write the data to, a storage device.

All of these workloads can be performed in software on a CPU. However, encrypting or compressing large quantities of data at line rate on a network function can consume significant CPU cycles. Public key cryptography is even more compute-intensive—a single RSA private key operation, using 2048-bit keys, can consume a million CPU cycles. A web server may have to perform thousands or even tens of thousands of such operations per second, consuming many CPU cores just to perform the cryptography, over and above the cores required to run the rest of the networking stack and application.

Intel® QuickAssist Technology offers dedicated hardware resources to accelerate the processing of these CPU-intensive workloads. By accelerating these compute-intensive workloads with a dedicated accelerator, they can be run at lower power, lower cost, and in a smaller area than the corresponding CPU cores required if they were to be run in software. Leveraging acceleration from Intel® QuickAssist Technology can free up CPU cycles or entire CPU cores that can then be used for value added applications, such as running virtual network functions (VNFs) and services. This enables users to choose a platform with the same number of cores but, by adding Intel® QuickAssist Technology, they can:

- Achieve a higher throughput for packet processing-intensive workloads and/or
- Run more VNFs

Intel® QuickAssist Technology is exposed to software as a PCI device. It can be included in a platform in different ways, for example, via a plugin card or by integration in a chipset or in a system-on-a-chip form factor. The next section describes how it is integrated in the 2nd generation Intel® Xeon® Scalable platform, in particular.

This paper describes Intel® QuickAssist Technology capabilities available with the 2nd generation Intel® Xeon® Scalable platform. It also describes how hardware acceleration of cryptography and compression, such as that provided by Intel® QuickAssist Technology, is applicable in an environment that uses Network Function Virtualization (NFV), Software Defined Infrastructure (SDI), or Hyper-Converged Infrastructure (HCI). This paper also demonstrates the performance benefit offered by Intel® QuickAssist Technology in the processing of IPSec traffic through the open source vector packet processing (VPP) IPSec implementation. Finally, the need for the Virtual Infrastructure Manager (such as OpenStack*) to be aware of such platform capabilities is addressed. This paper shows how Intel® QuickAssist Technology can be managed today using OpenStack*, and proposes enhancements to this to expose the specific services offered, including the functional capabilities and performance capacity thereof, and to allow service assurance via enforcement of service-level agreements.

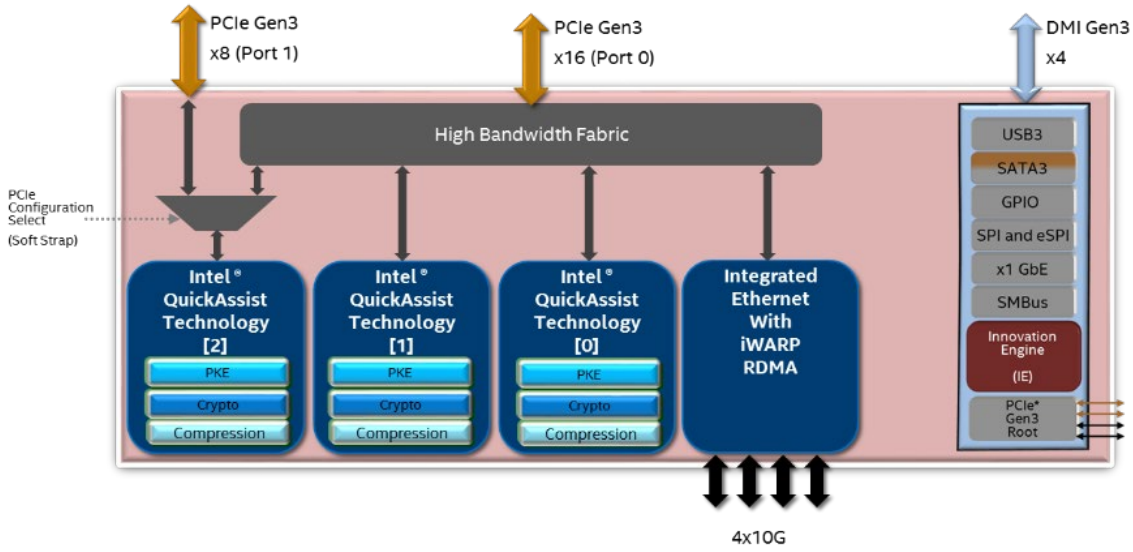
2.2 Demystifying Intel® QuickAssist Technology

The 2nd generation Intel® Xeon® Scalable platform incorporates the second generation of Intel® QuickAssist Technology and allows for different Intel® QuickAssist Technology deployment models:

- The Intel® C620 series chipset (formerly codenamed Lewisburg) has SKUs that include integrated Intel® QuickAssist Technology. In this case, in addition to connecting the CPU and the chipset via DMI, the platform should also connect some number of PCIe lanes between the CPU and the x16 upstream port, and optionally to the x8 upstream port.
- Alternatively, a PCIe plugin card with Intel® QuickAssist Technology, such as the Intel® QuickAssist Adapter 8960/8970, may be plugged into an available PCIe slot.

The Intel® C620 series chipset is illustrated in Figure 1. Parallel PCIe 3.0 x8 and x16 lanes provide up to 24 lanes of connectivity with the platform CPU. 16 of these lanes are shared between Intel® QuickAssist Technology and the on-chip 4x10 Gbps Ethernet interfaces. The optional x8 port can be used for one of the three Intel® QuickAssist Technology endpoints.

Figure 1. High-Level Schematic of Intel® C620 series chipset with Integrated Intel® QuickAssist Technology



Capabilities available with this second generation device include:

- **Symmetric cryptography.** In excess of 100 Gbps (up to 110 Gbps) of IPSec/SSL cryptography on a single device. It supports various ciphers, hash, and authentication functions, and authenticated encryption algorithms, in various modes, as shown in Table 3, Table 4, and Table 5. It can also perform chained cipher and authentication algorithms in a single request to hardware, as used by cryptographic protocols, including IPSec and SSL/TLS.
- **Public key cryptography.** Up to 100 K operations per second of RSA2K decryption. It supports various public key cryptography algorithms for encryption/decryption, digital signature generation/verification, and key exchange, as shown in Table 6.
- **Lossless Data Compression.** Up to 100 Gbps of data compression, and over 150 Gbps of decompression. It supports the deflate algorithm (LZ77 with either static or dynamic Huffman encoding), at various compression levels.

Table 3. Cipher Algorithms Supported

Cipher Algorithm	Mode
Null	
ARC4	
AES (key sizes 128, 192, 256)	ECB, CBC, CTR, XTS
DES	ECB, CBC
3DES	ECB, CBC, CTR
Kasumi	F8
Snow3G	UEA2

Cipher Algorithm	Mode
ZUC[1]	EEA3

Table 4. Hash/Authentication Algorithms Supported

Hash/Authentication Algorithm	Mode
MD5	Plain, Nested, HMAC
SHA1	Plain, Nested, HMAC
SHA-2 (output sizes 224, 256, 384, 512)	Plain, Nested, HMAC
SHA-3 (output size 256 only)[1]	Plain, HMAC
AES (key sizes 128, 192, 256)	XCBC-MAC CBC-MAC CMAC (key size 128 bit only) GMAC
Kasumi	F9
Snow3G	UIA2
ZUC[1]	EIA3

Table 5. Authenticated Encryption (AEAD) Algorithms Supported

Authenticated Encryption Algorithm	Mode
AES (key sizes 128, 192, 256)	GCM CCM (key size 128 bit only)

Table 6. Public Key Cryptography Algorithms

Algorithm	Operations	Parameter Sizes
RSA	• Sign, Verify • Encrypt, Decrypt • Generate Private Key	• Modulus Size (in bits): 512, 1024, 1536, 2048, 3072, 4096
DSA	• Sign, Verify • Generate Public Key (Y) • Generate Parameters (P, G)	• N: 160; L: 1024 • N: 224; L: 2048 • N: 256; L: 2048, 3072
Diffie-Hellman (DH)	• Generate Public Value (phase 1) • Generate Shared Secret (phase 2)	• Bit length of prime p: 768,1024,1536,2048,3072,4096
Large Number Arithmetic	• Modular Exponentiation • Modular Inversion	• Modulus Size: 1-4096 • Exponent and Base Size: 0-4096
ECDSA	• Sign, Verify	• Prime and Binary Fields

NOTE: [1]ZUC and SHA-3 are new algorithms in the second generation of Intel® QuickAssist Technology.

Algorithm	Operations	Parameter Sizes
ECDH	• ECDH Point Multiply • EC-CDH Point Multiply	• Supports NIST standard curves P224, P256, P384, and P521 (and cofactors 1, 2, and 4, for ECDH) • Supports ANY Weierstrass curves where max(log2(p), log2(n)+log2(h)) <= 512, where p = modulus, n = order of the curve and h = cofactor; curve parameters are passed via the API
EC	• Point Multiply, Point Verify	
SM2	• Point Multiply, Generator Multiply, Point Verify • Sign, Verify • Encrypt, Decrypt • Key Exchange (phases 1 and 2)	
EC25519	• Point Multiply • Point Verify	• ECDH using Curve25519

Comparison with performance capabilities of previous generations of Intel® QuickAssist Technology is illustrated in Figure 2, where “Lewisburg” indicates the Intel® C620 series chipset.

Note that Cave Creek (8920) and Coleto Creek (8955) are both considered to be the first generation of Intel® QuickAssist Technology.

Figure 2. Gen-to-Gen Comparison of Intel® QuickAssist Technology Capabilities

In Figure 2, the X axis shows throughput in Gbps for compression, SSL, and IPSec. For RSAs 2K, the X axis shows kilo operations / second (kops/s).

3.0 Utility/Use Case

3.1 Application of Technology

Intel® QuickAssist Technology has many potential uses in Network Functions Virtualization (NFV), telecommunications, cloud, and storage, including Software Defined Infrastructure (SDI), Software Defined Storage (SDS), and Hyper-Converged Infrastructure (HCI).

Applications include those requiring IPSec or SSL termination or high rates of compression or decompression, and for protecting Service Function Chaining (SFC) traffic between compute nodes. Inclusion of an Intel® QuickAssist Technology-enabled PCI device on a Standard High-Volume Server (SHVS) allows encryption/decryption jobs to run on the device, freeing up server compute resources and accelerating the security and compression functions.

Many network functions can take advantage of Intel® QuickAssist Technology's cryptographic acceleration capabilities. Data, control, and management traffic is increasingly being encrypted and authenticated to protect it as it traverses public networks, including in NFV-hosted scenarios. In different parts of the network, traffic is protected using one of IPSec, SSL/TLS, or a handful of other cryptographic protocols. As a result, many network functions need to decrypt inbound traffic and encrypt outbound traffic for various reasons, including:

- As a core function, such as in the case of an IPSec Security Gateway, SSL reverse proxy, or secure web server
- To perform its core function when the traffic is encrypted, such as in the case of a firewall, intrusion detection/prevention system, anti-malware, and so forth
- As a result of its location in the network, for instance, Application Delivery Controllers tend to sit at the trust boundary between the local and wide area networks

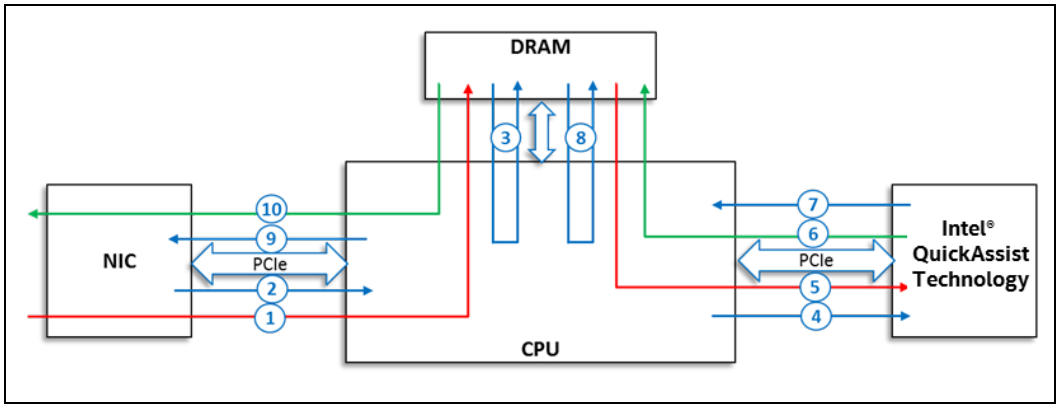
As noted earlier, each new connection (that is, IPSec tunnel or SSL/TLS session) typically begins with a key-exchange phase, where the communicating parties agree on the session keys. This involves the use of public key cryptography algorithms, such as RSA, ECDH, and others. Depending on the rate at which new connections are set up, accelerating these using Intel® QuickAssist Technology can free up significant CPU cycles. It can also help mitigate certain known side-channel attacks against software-based implementations of these algorithms.

Similarly, many network, storage, and compute functions can take advantage of Intel® QuickAssist Technology's compression acceleration capabilities. In many places in the network, data is also compressed, to save network bandwidth and/or to reduce the latency of the data as it traverses the network. Network functions such as content delivery networks, WAN optimization functions, and application delivery controllers generally use techniques including compression, as well as other techniques such as caching and data deduplication, to reduce the amount of data to be transmitted and the associated network latency. Storage applications also use compression to reduce storage requirements (and associated latency).

Acceleration of these workloads through the use of Intel® QuickAssist Technology can improve the performance of virtual customer premise equipment (vCPE), virtual evolved packet core (vEPC), and GiLAN deployments, among others, where routing, firewall, WAN optimization, and so forth are core functions. Whether implemented in a single large VNF or by service chaining of several smaller VNFs, the application of Intel® QuickAssist Technology to accelerate compute-intensive workloads is relevant and advantageous.

Figure 3 shows a sample data path for IPSec lookaside acceleration, used, for example, in a virtual firewall or vRouter application. In this example, packets flow between the Ethernet ports (which may or may not be the integrated ports on the Intel® C620 series chipset), the CPU, platform memory, and the Intel® QuickAssist Technology device (such as on the Intel® C620 series chipset). Actions performed at each stage in the data path are described below.

Figure 3. Flow of Packets in the Intel® QuickAssist Technology IPSec Data Path



An encrypted packet arrives at the network interface and copied to memory.

1. A packet descriptor is sent to the CPU. This can generate an interrupt, or the CPU may poll.
2. The CPU processes the packet descriptor and header. On detecting an IPSec-encrypted packet, it looks up the Security Association, and then creates a request descriptor to send to the Intel® QuickAssist Technology device.

3. The CPU enqueues the Intel® QuickAssist Technology request descriptor to the accelerator.
4. The accelerator fetches the encrypted data from memory, and performs decryption and authentication of the MAC.
5. The accelerator DMAs the cleartext back to DRAM.
6. A response is sent to the CPU. This can be configured to generate an interrupt, or again, the CPU can poll for responses.
7. The CPU retrieves the packet to perform whatever additional layer 3 through 7 functionality is required, based on the application. The packet is written back to memory.
8. The Ethernet device is notified that a packet is ready for egress transmission.
9. The Ethernet device fetches the packet from DRAM and transmits the cleartext packet.

3.2 IPSec Application Instantiation

Demonstration of Intel® QuickAssist Technology capabilities on the 2nd generation Intel® Xeon® Scalable platform for an NFV-relevant use case is facilitated by integrating the Data Plane Development Kit (DPDK) Cryptodev library into the Vector Packet Processing (VPP) project within FD.io, resulting in a feature-rich, high performing IPSec data plane.

The 2nd generation Intel® Xeon® Scalable platform with integrated Intel® QuickAssist Technology hardware acceleration is the foundation for an IPSec solution scaling to over 100 Gbps per CPU socket.

The DPDK Cryptodev library offers a consistent framework and API, managing and supporting multiple software and hardware implementations on a given platform. DPDK Cryptodev automatically performs HW/SW load balancing and scales to fully use the cryptographic capability of the 2nd generation Intel® Xeon® Scalable platform. The Cryptodev framework currently supports Cipher, Authentication, chained Cipher/Authentication, and AEAD symmetric Crypto operations.

VPP is built on top of DPDK poll mode drivers (PMD) and ring buffer libraries, architected with parallel processing (use of graph nodes) to reduce the number of CPU cache misses, to achieve unprecedented switching and routing performance on standard high volume Intel based servers.

VPP provides a commercial-ready IPSec solution with address resolution protocol (ARP), security association database (SADB) locking, Internet key exchange (IKE), plus extended sequence number (ESN) and anti-replay support showing scalable performance with the use of software only, hardware crypto acceleration only, or a combination of the two.

Figure 4 presents a diagram of the solution components.

Figure 4. VPP IPSec Solution Components

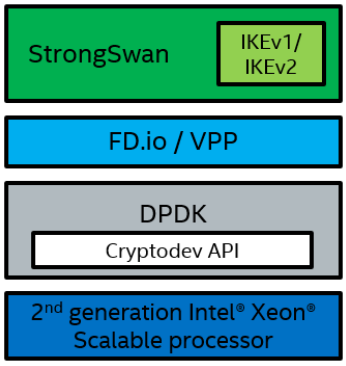
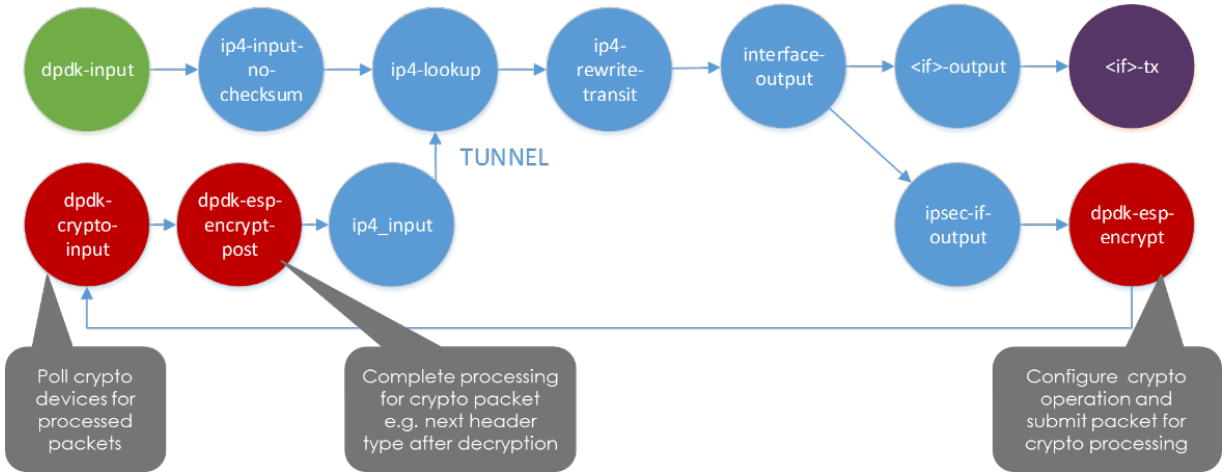


Figure 5 shows the additional graph nodes to accommodate hardware and software IPSec acceleration in VPP through the integration of the DPDK Cryptodev library.

Figure 5. VPP Graph Nodes Accommodating HW and SW IPSec Acceleration in VPP



3.3 Relevant Benchmarks

Demonstration of Intel® QuickAssist Technology capabilities on the 2nd generation Intel® Xeon® Scalable platformfor an NFV-relevant use case is facilitated by use of VPP IPSec, which takes advantage of the DPDK Cryptodev API as described previously.

Figure 6 shows the benchmark setup. A traffic generator sends 60 Gbps of plain text traffic (1420 byte packet sizes were used) to the first device under test (DUT). The network interface card was the Intel® X710, presenting four 10 Gbps Ethernet ports. DUT1 runs one instance of VPP IPSec and encrypts the plain text traffic either by HW acceleration with Intel® QuickAssist Technology or by using the SW AES_NI multi-buffer library (AES_128_GCM is the algorithm used in this case).

The encrypted traffic is sent via Intel® X710 to the second device under test, where another instance of VPP IPSec decrypts the data (again, either using Intel® QuickAssist Technology or AES_NI). The resulting plain text traffic is returned to the traffic generator for throughput analysis. The demonstration runs bidirectionally to enable >100 Gbps of input traffic to each DUT.

In this setup, each 10 Gbps port in the NIC gets its own security flow, and the keys are fixed, not negotiated. Hyperthreading is enabled on both DUTs. The AES_128_GCM algorithm is used in IPSec encryption/decryption here.

Figure 6. Intel® QuickAssist Technology Acceleration of IPSec Encryption/Decryption—VPP IPSec Benchmark Setup

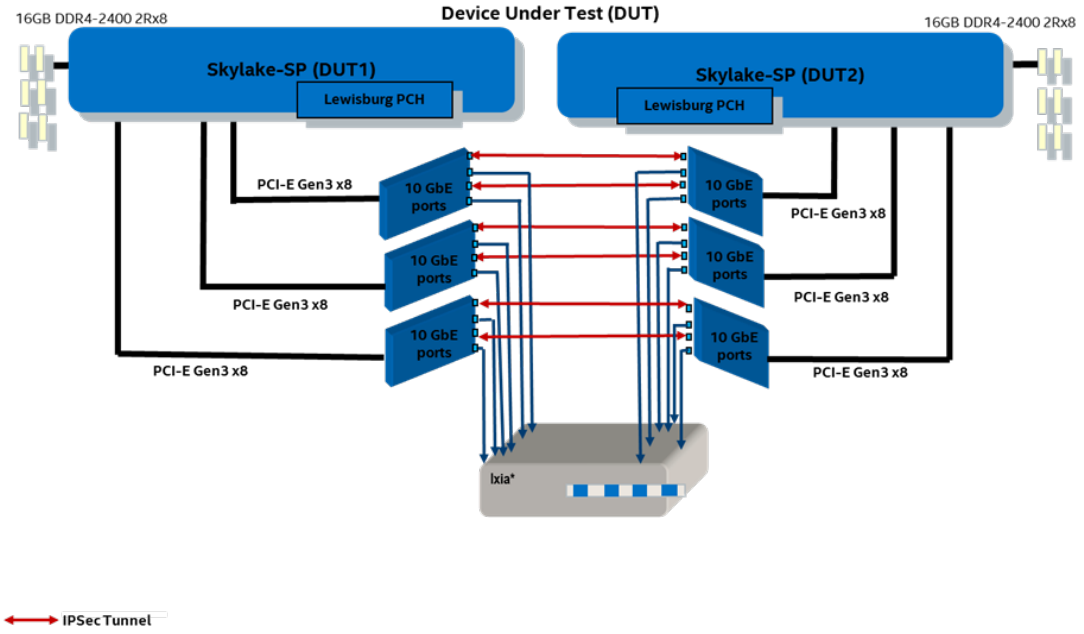
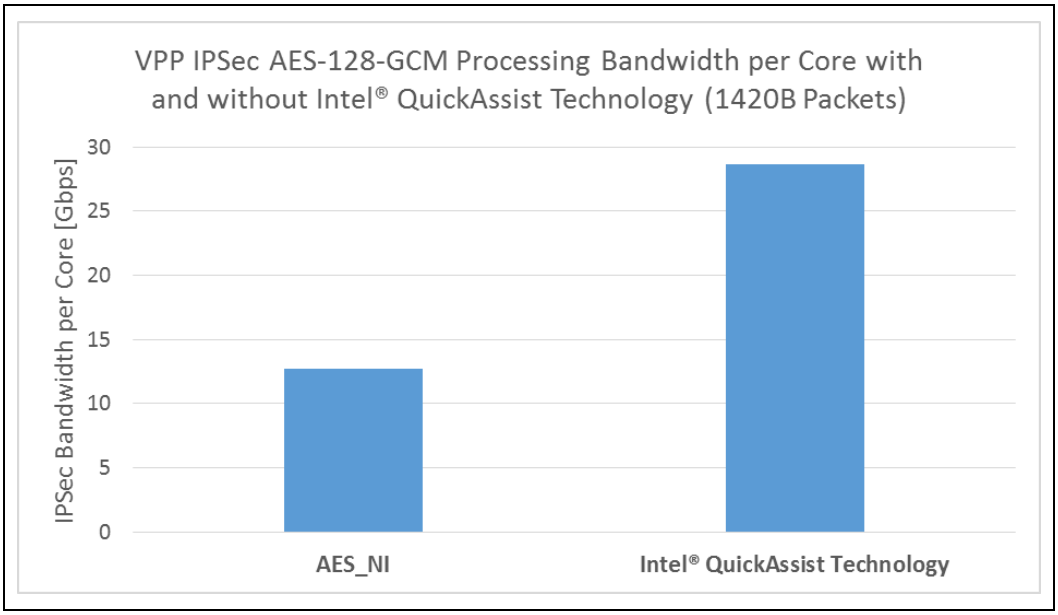


Figure 7 shows the results from this benchmark test. The graph presents the VPP IPSec processing throughput per physical CPU core for traffic encryption/decryption using either Intel® QuickAssist Technology or with the AES_NI multi-buffer library.

Figure 7. VPP IPSec Benchmark Result—Comparison of Per Core IPSec Throughput with Intel® QuickAssist Technology Acceleration and AES_NI Processing



With the SW-based AES_NI multi-buffer library implementation, the per core IPSec processing achievable through the VPP software stack is 12.75 Gbps, therefore requiring 8 physical CPU cores to achieve 100 Gbps total throughput. By leveraging Intel® QuickAssist Technology's HW acceleration, the per core bandwidth increases to 28.7 Gbps, resulting in a 3.5 CPU core requirement to achieve 100 Gbps total throughput. This 2.25x performance improvement—made possible by Intel® QuickAssist Technology—frees up compute capacity on the platform to run more VNFs.

Note:

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
Performance results are based on testing as of 3/20/2019 and may not reflect all publicly available security updates. See configuration disclosure in Appendix A for details. No product or component can be absolutely secure.

4.0 Enablement

Intel® QuickAssist Technology can be enabled and orchestrated in both OpenStack* and Kubernetes*. This section discusses both options.

4.1 Using OpenStack* with Intel® QAT

Monitoring and provision of Intel® QuickAssist Technology resources, especially in an NFV environment, can be managed through OpenStack*.

4.1.1 OpenStack* Base Capability

Since the Havana* release, OpenStack* has had the ability to see PCI devices on a platform and pass through such devices to a Virtual Machine (VM). By filtering for PCI devices that match the Vendor ID and Device ID of Intel® QuickAssist Technology devices, the OpenStack* controller can be aware that a given platform is Intel® QuickAssist Technology enabled, and can place Virtual Network Functions (VNFs) that require acceleration on these platforms. It can also assign Virtual Functions (VFs) to the corresponding VMs; this is managed by extending the metadata (also known as *extra specs*) of a "flavor" to specify that a VF should be assigned.

This existing capability is useful in identifying Intel® QuickAssist Technology enabled platforms, but it has some limitations, as follows:

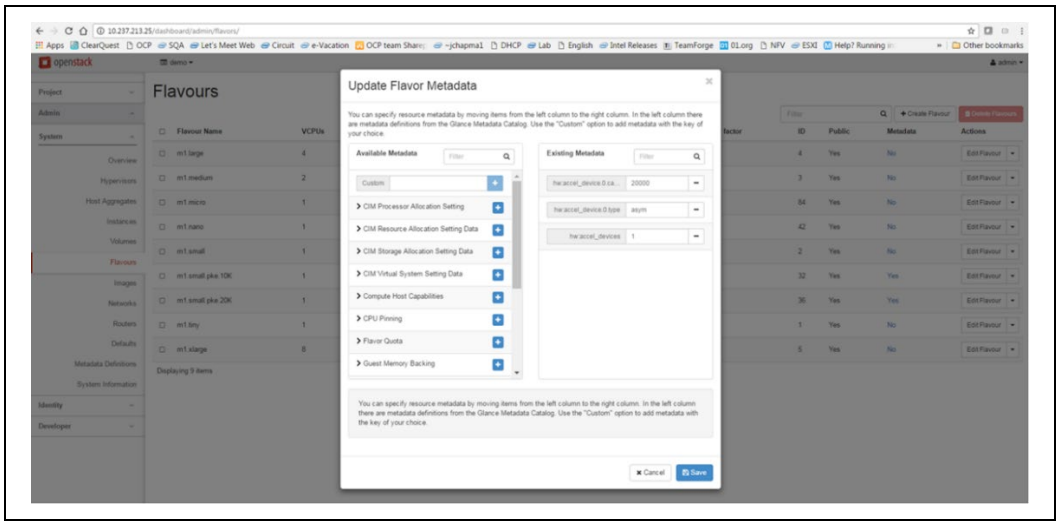
- No information is shared about the specific services offered by the Intel® QuickAssist Technology device. Most Intel® QuickAssist Technology devices offer all three services (symmetric crypto, public key crypto, and compression), but there may be SKUs of certain Intel® QuickAssist Technology devices that offer only a subset of these services.
- No information is shared about the capacities of the services, as measured in units of, say, throughput of symmetric crypto for a given algorithm and packet size, or the rate of operations per second of public key crypto for a given algorithm and key size. Again, these capacities may vary with SKU.
- There is no way to assign a given subset of the capacity to a specified VF or VM. Currently, by default, Intel® QuickAssist Technology devices share their capacity across all VFs using a Weighted Round Robin (WRR) scheme.

4.1.2 OpenStack* Enhanced Capability

Intel has implemented a proof of concept of the following enhancements to the basic capability described in the previous section:

- For each Intel® QuickAssist Technology enabled device on a platform, the platform reports to the OpenStack* controller the specific services offered by that device and the capacity of each.
- The OpenStack* controller flavor metadata has been extended to include a specification of the service and the capacity required (for example, 20 Gbps of symmetric crypto, or 5000 operations/second of public key crypto), as illustrated in Figure 8. When this flavor is instantiated, a VM will be launched with the number of vCPUs and amount of memory and storage specified by the flavor, and an Intel® QuickAssist Technology VF will be passed through to that VM (per the base capability above). In addition, the OpenStack* controller will request the platform to offer the specified capacity to that VF.

Figure 8. Horizon GUI Showing the Metadata for a Flavor

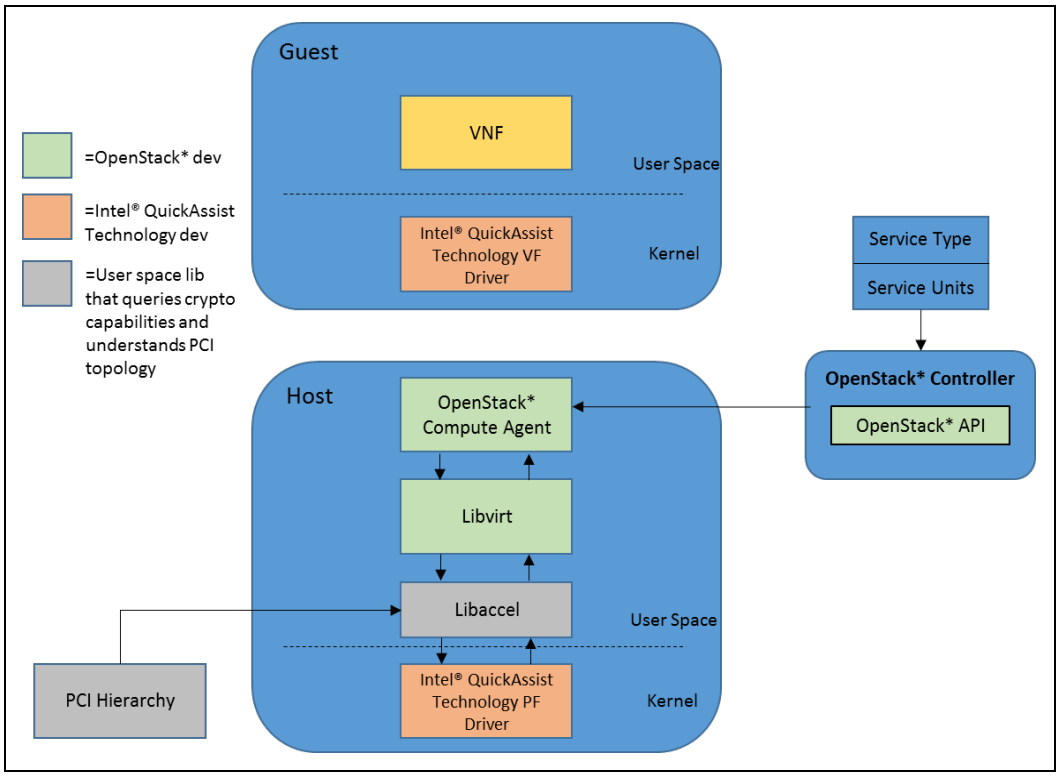


In turn, this requires that the Intel® QuickAssist Technology device can offer and enforce Service Level Agreements (SLAs). As part of this proof of concept, an SLA enforcement mechanism was implemented on the Intel® QuickAssist Technology device.

In this way, an application that requires acceleration services (for example, a virtual firewall requiring IPSec encryption and decryption) can be automatically placed on nodes in the compute pool that can service the acceleration requirement. Furthermore, the acceleration resource is offered with a service-level agreement (SLA) that aims to ensure a specified capacity of the service (for example, 20 Gbps of symmetric cryptography).

Figure 9 illustrates the software stack.

Figure 9. Software Stack for OpenStack* Enhanced Platform Awareness of Intel® QuickAssist Technology



The Intel® QuickAssist Technology Physical Function (PF) driver in the Linux* kernel publishes information regarding the available capacity of Intel® QuickAssist Technology resources. The information published in this proof of concept includes:

- The services offered by an Intel® QuickAssist Technology device—one or more of symmetric cryptography, public key cryptography, and data compression.
- For each service, the total device capacity and the available capacity. (The available capacity may be less than the total capacity if some of the capacity is already allocated to guests, or if it is managed locally, rather than by the orchestrator.) The capacity is reported in the following units:
  - **Symmetric crypto** measured in throughput of a chained cipher and hash operation using the algorithm AES-128-CBC + SHA256-HMAC, at a packet size of 1 KB. Throughput for different algorithms, buffer sizes, and so forth may differ.
  - **Public key crypto** measured in the rate of operations per second of an RSA decrypt operation using Chinese Remainder Theorem, at a key size of 2048 bits. Throughput for different algorithms, key sizes, and so forth may differ.
  - **Compression** measured in throughput of a compress operation using the deflate algorithm at level 1, stateless, with dynamic Huffman, at a packet size of 64 KB. Throughput for different levels, buffer sizes, and so forth may differ.

- The number of device SLAs available. (This may be limited by device hardware, in some cases.)

On a compute node, this information is passed from the PF driver to a user space library, *libaccel*, which also understands the PCI bandwidth availability for the connection with the Intel® QuickAssist Technology device. This combined information is abstracted by libvirt and collected by the OpenStack* Nova agent on the compute node.

In a system of multiple compute nodes with all, some, or none containing Intel® QuickAssist Technology devices, the OpenStack* controller collects this acceleration capacity information from all nodes in the compute pool and thus understands the instantaneous system-wide acceleration capabilities. A user or application requesting a specific capacity of acceleration service will trigger the controller to reserve the specified acceleration capacity on a compute node with sufficient capacity, associate that with a particular accelerator Virtual Function, and assign that VF to the Virtual Machine.

The instruction from the OpenStack* controller (service type, quantity of service) is passed to the Nova* agent on the compute node and through libvirt and libaccel to effect the changes required in the Intel® QuickAssist Technology PF driver. Once the service is provisioned, the published capacity of the Intel® QuickAssist Technology service is decremented by the appropriate amount and the controller information is updated to reflect this.

Separate security workloads can run concurrently on a single accelerator with no interference or crosstalk, through SR-IOV-based virtualization of the hardware. The proof of concept implementation supports the establishment of up to 12 SLAs across the three Intel® QuickAssist Technology endpoints on an Intel® C620 series chipset, allowing for sharing of the resources by up to 12 VMs on the platform.

The solution demonstrates improved performance and scalability of NFV solutions where the time-critical processing of secure transmission protocols is required, ultimately leading to improved performance, efficiency, and density of secure NFV applications.

4.2 Using Kubernetes* with Intel® QAT

The Kubernetes* device plugin framework provides a vendor-independent solution for hardware devices. Intel has developed a set of device plugins, which includes the Intel® QuickAssist Technology (Intel® QAT) plugin. The plugins comply with the Kubernetes* device plugin framework and allow users to request and consume hardware devices across Kubernetes clusters.

The Intel® QAT device plugin for Kubernetes supports Intel® QuickAssist adapters and includes an example scenario that uses the Data Plane Development Kit (DPDK) drivers.

An additional demo that executes an Intel® QAT accelerated OpenSSL* workload with the Kata Containers runtime is also available.

For more information, refer to:

- Detailed plugin configuration and deployment instructions: https://github.com/intel/intel-device-plugins-for-kubernetes/blob/master/README.md
- Plugin build, test, and DPDK demo details: https://github.com/intel/intel-device-plugins-for-kubernetes/blob/master/cmd/qat_plugin/README.md
- OpenSSL demo details: https://github.com/intel/intel-device-plugins-for-kubernetes/tree/master/demo

5.0 Summary

This paper outlines the features and capabilities of Intel® QuickAssist Technology (Intel® QAT) in the 2nd generation Intel® Xeon® Scalable platform. It describes NFV use cases, with a particular focus on VNFs that use IPSec-encrypted traffic. The performance benefits are showcased through the demonstration of VPP IPSec packet encryption/decryption, enabled through the underlying DPDK Cryptodev API. VPP IPSec benchmarking demonstrates a 2.25x per core performance improvement for IPSec processing of 1420B packets with Intel® QAT, as compared to the AES_128_GCM multi-buffer library implementation (28.7 Gbps per core with Intel® QAT, 12.75 Gbps per core with AES_NI). >100 Gbps IPSec processing on a single socket is achievable in both approaches.

Finally, the current status of orchestration-awareness of acceleration devices is discussed, as well as Intel's proof of concept activities to enable enhanced platform awareness of acceleration devices and intelligent placement of VNFs requiring acceleration service.

In summary, the following benefits are highlighted in this document.

- Increased Intel® QAT throughput capacity on PCH, up to 110 Gbps of IPSec/SSL cryptography, up to 100 Kop/s of RSA2K public key crypto, up to 100 Gbps of compression, and 165 Gbps of decompression for deflate (LZ77).
- Intel® QuickAssist Technology is available in a server chipset—Intel® C620 series chipset (formerly codenamed Lewisburg PCH).
- DPDK Cryptodev software supports the 2nd generation Intel® Xeon® Scalable platform by combining a best in class software crypto implementation in addition to fully taking advantage of the integrated Intel® QuickAssist Technology hardware acceleration.
- Demonstration of a performant open source IPSec solution stack with significant CPU core saving made possible by virtue of Intel® QuickAssist Technology (>100 Gbps of IPSec encryption/decryption for 1420B packets on VPP, using Intel® QuickAssist Technology acceleration).
- Demonstration of an OpenStack* enhanced platform awareness proof of concept, enabling intelligent placement of VNFs that require acceleration onto platforms with the capacity to service the request.

Appendix A Test Information

A.1 Test Setup

This section specifies the hardware and software configurations for the VPP IPSec benchmark test described in Section 3.3.

A.1.1 Hardware

The tests described in this paper were carried out using a 2nd generation Intel® Xeon® Scalable platform. Details are outlined below.

Platform	2nd generation Intel® Xeon® Scalable platform
CPU	Intel® Xeon® Gold 6230 CPU @ 2.10 GHz
Chipset	Intel® C620 series chipset
No of CPU	1
Cores per CPU	20
L3 cache (total)	39424 K
QPI/DMI	Auto
PCIe	SLOT1 PCI-E 3.0 X8 Speed (FVL X710-DA4) SLOT2 PCI-E 3.0 X16 (with a passive x16 to 2x8 Switch) Speed (2xFVL X710-DA4)
IXIA/DUT Connectivity	Six 10 G ports to each DUT, six 10 G ports back to back
MEMORY	Samsung* Part # M393A1G43DB1-CRC00, DDR4-2400 @ 2400 MHz, 8GB RDIMM, 1 DIMM per Channel, 4 Channels per Socket, 32 GB Total
NIC	3 x Intel® Ethernet Controller X710 (Quad Port FVL Card)
BIOS	PLYXCRB1.86B.0568.D10.1901032132
Acceleration Card	Integrated Intel® C620 series chipset PCH (in x24 mode unless specified otherwise)

A.1.2 Software

The software versions for IPSec benchmarking used are indicated below.

Host OS	Ubuntu* 14.10
Kernel version	3.16.0-23-generic
Other	VPP IPSec
VPP version	17.01 (commit id: abd98b2c88ec127c38ff804a0c2f2a6d6f018830) + 'migration from MB library to ISA-L' patch (only affects Software GCM performance)