Intel® QuickAssist Technology (Intel® QAT) - NGINX* Performance

White Paper

Rev. 1.0

February 2023

Contents

Figures

Tables

1.0 4th Gen Intel® Xeon® Scalable Processor & Intel® QuickAssist Technology: NGINX* Performance

1.1 Abstract

Intel® QuickAssist Technology (Intel® QAT) integrates hardware acceleration of compute intensive workloads to enable significant gains in CPU efficiency, data footprint reduction, power utilization and application throughput. Accelerate bulk cryptography, public key cryptography and compression by offloading to Intel® QAT hardware.

1.2 Authors

Brian Will, Principal Engineer

Karen Shemer, Silicon Architecture Engineer

1.3 Introduction

The impact of encryption on web and data center compute resources is significant as Cloud, Edge and Enterprises look to “encrypt everything, everywhere”. This translates to increased CPU cycles, latency and therefore cost. Threats to data security are constantly evolving, and organizations must protect their web applications just as they do their other enterprise infrastructure. Intel® QAT is a new capability integrated directly into the 4th Gen Intel® Xeon® Scalable Processors, that accelerates encryption and compression to provide improved efficiency, scalability, and performance – for data in motion, at rest or in-flight. Intel® QAT used for TLS Acceleration, reduces demands on the platform and increases TLS connection rates with accelerated public key cryptography and digital signatures. Acceleration will also be a key factor in mitigating security attacks such as DDoS, by allowing the server to survive a flood of TLS connection requests.

In this paper we will show the relative gains (CPS, performance per watt) delivered by Intel® QAT for cryptographic acceleration used in the TLS protocol, through NGINX and OpenSSL as the benchmarked stack.

1.4 Usages and Applications

Hypertext Transfer Protocol Secure (HTTPS) is an internet communication protocol that protects the integrity and confidentiality of data between the user's computer and the site. Users expect a secure and private online experience when browsing online, shopping on an e-commerce platform or using a favorite banking app. All of this is done using HTTPS that uses the Transport Layer Security (TLS) protocol, which is the most widely used cryptographic protocol for protecting communications on the Internet. TLS was initially developed in the mid-1990s by Netscape Communications and has become an Internet standard under the Internet Engineering Task Force

(IETF). While TLS has been revised many times since its initial development, it remains a fundamental element of protecting data over HTTPS, which is used to secure web traffic.

The latest version of TLS 1.3 was launched in 2018 and brings significant improvements in the areas of privacy, security, and performance, by removing support for algorithms with known vulnerabilities, reducing handshake latency and improved resiliency against cross-protocol attacks.

1.5 Importance of TLS Protocol for Communications

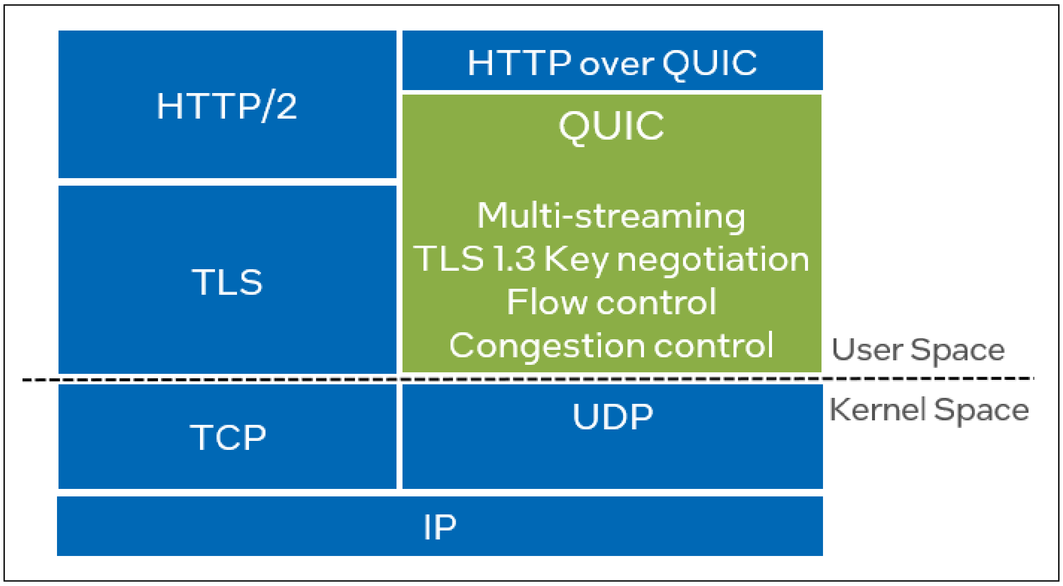
Transport Layer Security (TLS) is the most widely used security protocol for encrypting traffic on the Internet and without doubt is the backbone of Internet security today. It protects HTTP websites, emails, and other data exchanged between web browsers and servers, ensuring that no one can eavesdrop or tamper with communications. TLS is used by all of today's modern browsers and many online services expect clients to be able to use it before doing business with them.

The primary aim of TLS is to provide privacy, integrity, and authenticity using certificates, between clients and servers. TLS runs in the application layer and is composed of three protocols, the Handshake, Record and Alert protocols. Of these operations the TLS handshake consumes the most CPU cycles per byte due to the costly Public Key Exchange (PKE) calculations involved. Performance of TLS connections are influenced by network latency, bandwidth, as well as key size, algorithm, and cipher suite.

1.5.1 Emerging Alternative: QUIC

QUIC is a new transport layer network protocol that provides security, reliability, and speed.

Figure 1. HTTPS Stack with QUIC



The protocol relies on User Datagram Protocol (UDP), which eliminates many of the handshakes required when using TLS over TCP for HTTPS support, which takes significant cycles and increased connection establishment latency. This in turn is making your download and latency times faster. Google first tested it in 2013 in its Chrome browser before other browsers and services. After many years of development and testing, in 2021 the Internet Engineering Task Force (IETF) released QUIC as a standard.

1.5.2 Web Server/Proxy/Load Balancer

NGINX* is a lightweight, high-performance HTTP and reverse proxy/web server based on a Berkeley Source Distribution (BSD)-like license. It also provides the following services:

- Internet Message Access Protocol (IMAP)
- Post Office Protocol Version 3 (POP3)
- Simple Mail Transfer Protocol (SMTP)

Since the release of the first version in 2004, its market penetration rate has increased year by year, and it has been widely applied in many front-line Internet companies and IT enterprises. The NGINX architecture design is very flexible, with a small and simple kernel containing core modules, basic modules, and tripartite modules. It collaborates with modules through file static mapping and configurable instructions, highlighting significant advantages of high performance, high concurrency, and low memory in various application scenarios such as HTTP proxy, static and dynamic separation, load balancing, virtual host, reverse proxy, cache acceleration, authorized access, and others.

1.5.3 Optimizations

Intel® Crypto-New Instructions (Intel® Crypto-NI) is a new encryption instruction set that improves on the Advanced Encryption Standard (AES) algorithm and accelerates the encryption of data in the Intel® Xeon® processor family and the Intel® Atom™ processor family.

Comprised of several new instructions, Intel® AES-NI gives your IT environment faster, more affordable data protection, and greater security; making pervasive encryption feasible in areas where previously it was not.

By implementing some intensive sub-steps of the AES algorithm into the hardware, Intel® AES-NI accelerates execution of the AES application.

For years Intel has partnered with the open-source community to offer new optimizations, such as the asynchronous infrastructure, which was added into OpenSSL-1.1.0 and enables cryptographic operations to execute asynchronously with respect to the stack and application. Generically, the infrastructure could be applied to any asynchronous operations that might occur, but currently only encompasses cryptographic operations executed within the engine framework.

Asynchronous operations are initiated and consumed (via events/polling) by the main program but will occur in parallel to those operations. Figure 2 and Figure 3 are an illustration of the shift in execution.

Figure 2. Asynchronous Execution

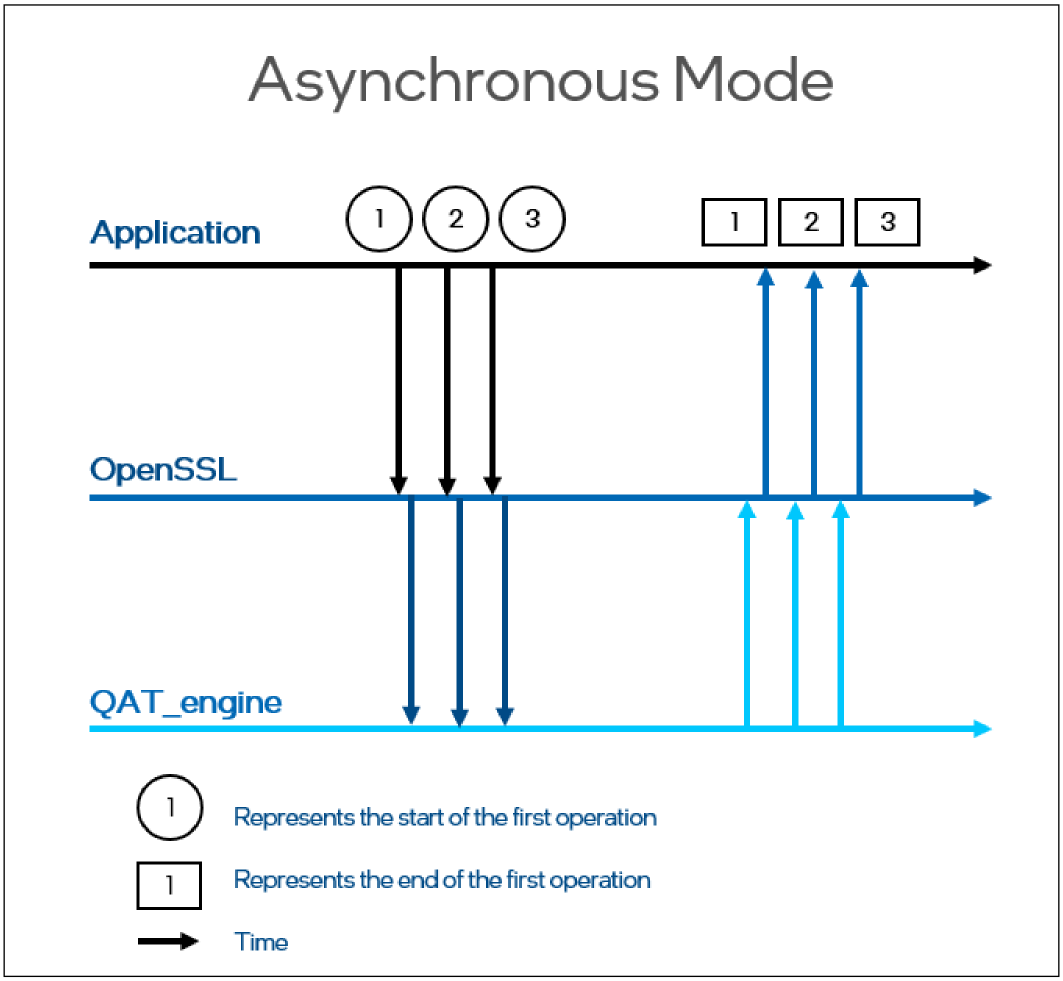
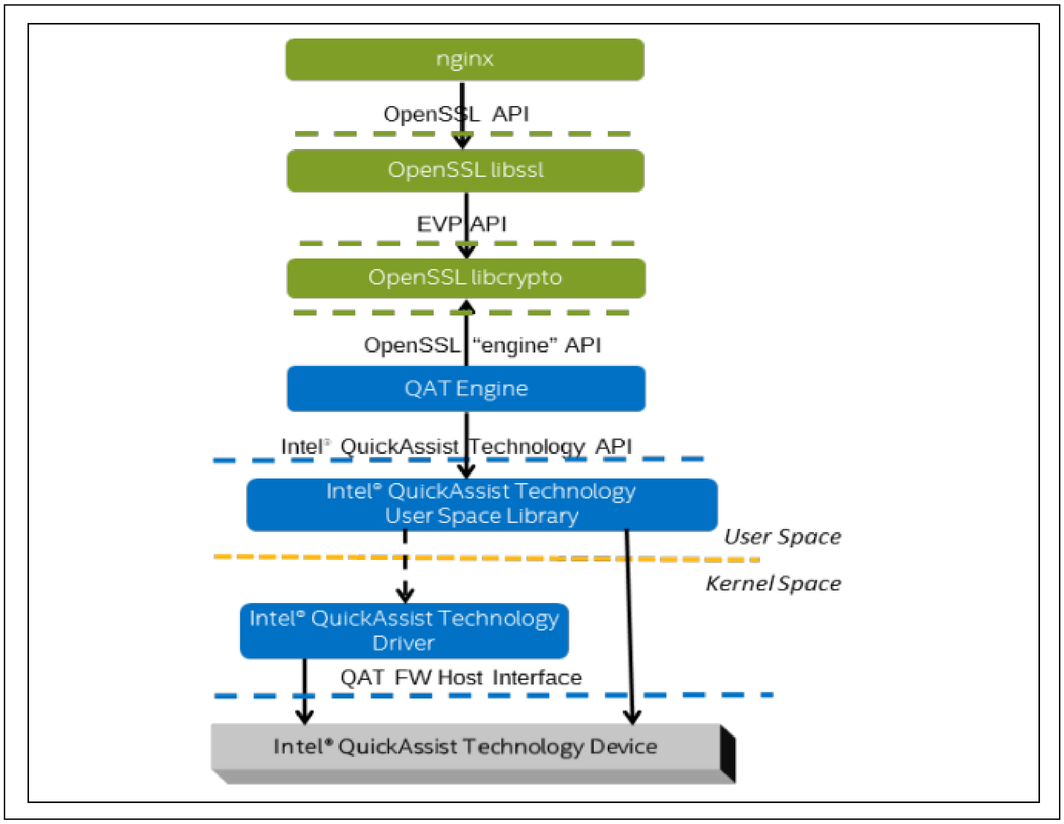

Asynchronous Mode
Application
1
2
3
1
2
3
OpenSSL
QAT_engine
1
Represents the start of the first operation
1
Represents the end of the first operation
Time

Figure 3. Intel® QuickAssist Technology Stack Diagram


nginx
OpenSSL API
OpenSSL libssl
EVP API
OpenSSL libcrypto
OpenSSL "engine" API
QAT Engine
Intel® QuickAssist Technology API
Intel® QuickAssist Technology User Space Library
User Space
Kernel Space
Intel® QuickAssist Technology Driver
QAT FW Host Interface
Intel® QuickAssist Technology Device

Intel® QAT accelerates symmetric and asymmetric encryption as well as lossless compression in hardware, offloading compute-intensive operations to Intel® QAT accelerator. This translates to significant gains in CPU efficiency, data footprint reduction, power utilization and application throughput. Intel® QAT Gen 4 hardware is integrated in the 4th Gen Intel® Xeon® Scalable processors and offers higher throughput, improved compression ratio, concurrent compression and decompression, support for additional standards and features including Acceleration Interface Architecture (AIA), Shared Virtual Memory (SVM), Scalable IO Virtualization (SIOV), RSA 2k + P256, and RSA 8k support. Significant gen-over-gen performance gains include Bulk Crypto: 200-400Gbs, and PKE: 80 kops.

The Intel® QuickAssist Technology accelerator is accessed through a device driver in kernel space and a library in user space. Cryptographic services are provided to OpenSSL through the standard engine (provider in OpenSSL-3.0) framework. This engine (Figure 3) builds on top of the user space library, interfacing with the Intel® QAT API, which allows it to be used across Intel® QAT generations without modification. This layering and integration into the OpenSSL framework allow for seamless usage by applications.

1.6 One Board Running Pre-Production Intel® Xeon® Scalable System Configuration

1.6.1 Benchmarks & Key Performance Indicators

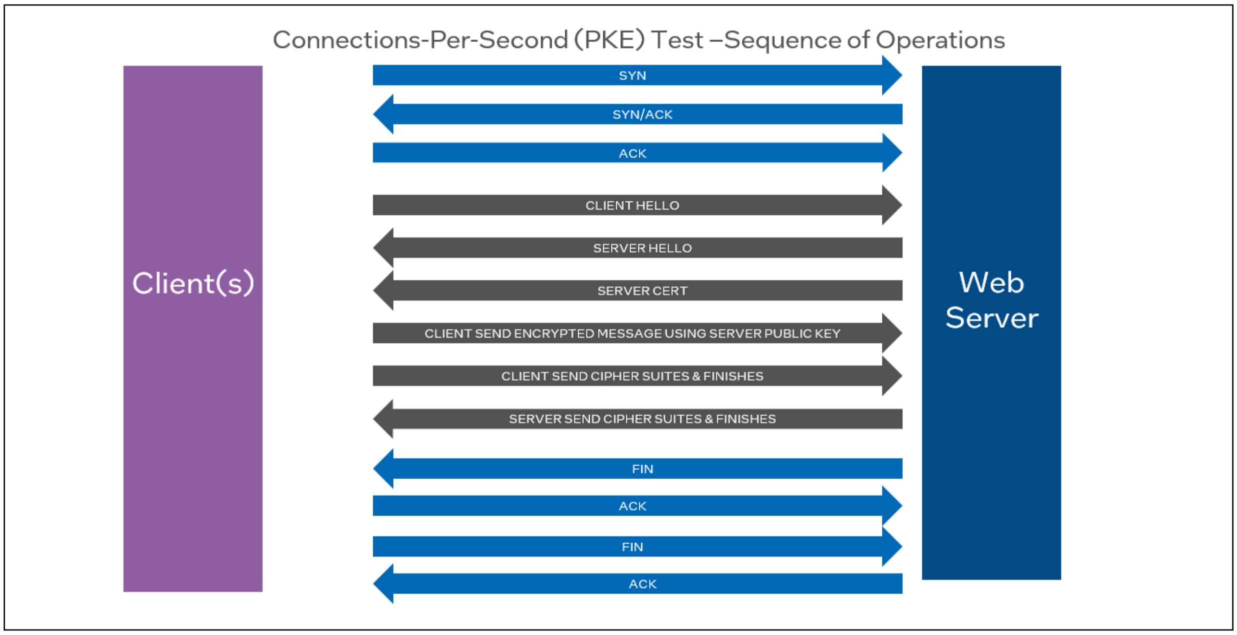
1.6.1.1 Connections per Second (CPS)

Clients send HTTPS connection requests without requesting data. This will utilize Key Exchange + Certificate Authentication exercising the TLS-1.3 handshake only with no data transfer.

1.6.2 Test Setup

For this test a pre-production Intel® Xeon® Platinum 8470N is used. The board is interconnected to a Cisco Nexus C3232C router with a total possible aggregated link bandwidth of 600 GbE (6x100GbE links).

Figure 4. Sequence of Operations for Public Key Encryption Test


Connections-Per-Second (PKE) Test –Sequence of Operations
Client(s)
SYN
SYN/ACK
ACK
CLIENT HELLO
SERVER HELLO
SERVER CERT
CLIENT SEND ENCRYPTED MESSAGE USING SERVER PUBLIC KEY
CLIENT SEND CIPHER SUITES & FINISHES
SERVER SEND CIPHER SUITES & FINISHES
FIN
ACK
FIN
ACK
Web Server

The following tables show the test setup used for this benchmarking.

Table 1. Hardware System Configuration

Component	Description
CPU	Pre-production Intel® Xeon® Platinum 8470N
Memory	16*32 GB DDR4, 4800 MT/s
Hard Drive	Intel® SSDSC2BB240G4
Ethernet Adapter	6x Intel® Ethernet Network Adapter E810-CQDA2 (ice 1.8.3, firmware 3.20)
Intel® QAT Hardware	Intel® QAT Gen 4
Turbo	Disabled

Table 2. Software System Configuration

Component	Description
Operating System	Ubuntu 22.04 LTS
Kernel	5.15.0-27-generic
NGINX	Async NGINX 0.4.7
OpenSSL	OpenSSL 1.1.1o
Intel® QAT driver	QAT 20.L.0.9.0-00023
Intel® QAT Engine for OpenSSL	QAT Engine v0.6.12
Intel® Multi-Buffer Crypto for IPsec Library	Intel IPsec MB v1.2
Intel® Integrated Performance Primitives (Intel® IPP) Crypto library	IPP Crypto ippcp_2021.5

For each client, 1000 outstanding connection requests are created with OpenSSL* s_time on the Client. The requests run both continuously and simultaneously for 400 seconds. Multiple client machines are used to ensure that there are no limitations from the Client side. Each Client process establishes a secure connection, exits gracefully, and sends a new request to establish a secure connection. At the end of the 400-Second run, the Connections per Second from each process is summed up and reported on the screen.

One (1) worker process is allocated for every hyper-thread in the nginx.conf. Example: 2C4T has 4 worker_processes

Each test is task set to cores and hyperthreads for consistency of results. Example – 2C4T: taskset –c 1-2,65-66 ./nginx –c nginx.conf

The SSL record size is 16K, Ethernet MTU sizes on Clients and Server are kept at 1500, and the TCP Segmentation Offload is On.

Figure 5. Hardware Configuration

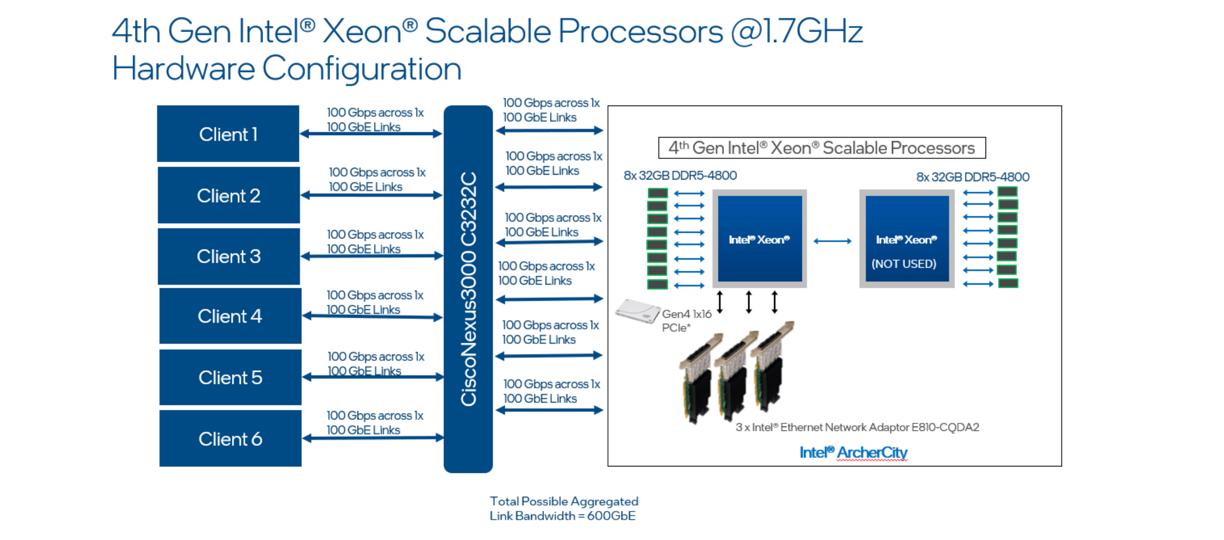
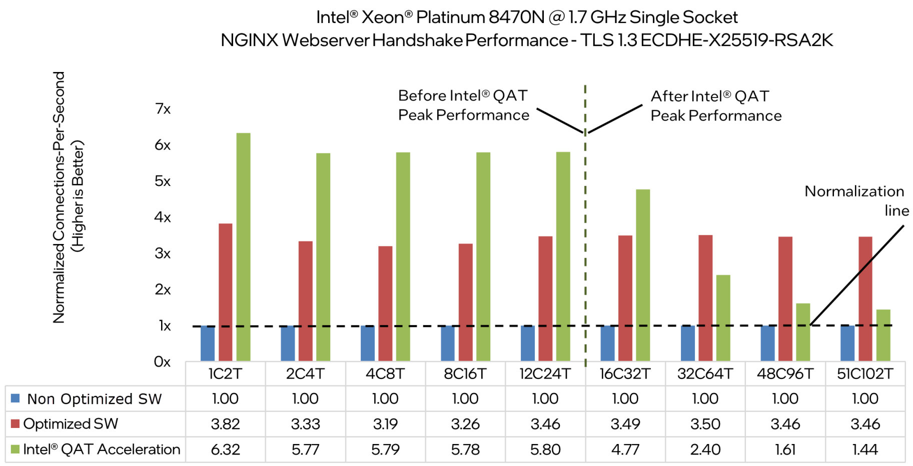

4th Gen Intel® Xeon® Scalable Processors @1.7GHz
Hardware Configuration
Client 1
Client 2
Client 3
Client 4
Client 5
Client 6
100 Gbps across 1x 100 GbE Links
CiscoNexus3000 C3232C
4th Gen Intel® Xeon® Scalable Processors
8x 32GB DDR5-4800
8x 32GB DDR5-4800
Intel® Xeon®
Intel® Xeon® (NOT USED)
Gen4 1x16 PCIe*
3 x Intel® Ethernet Network Adaptor E810-CQDA2
Intel® ArcherCity
Total Possible Aggregated Link Bandwidth = 600GbE

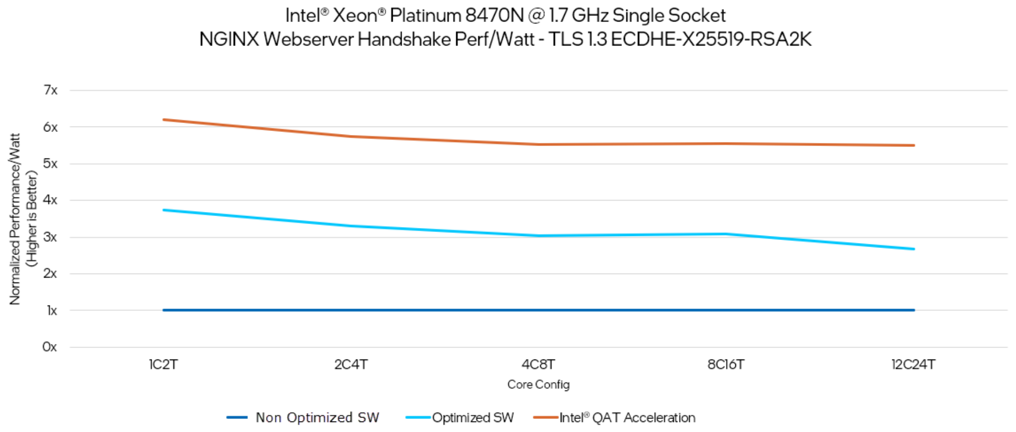
Figure 6. NGINX Webserver Handshake Performance



	1C2T	2C4T	4C8T	8C16T	12C24T	16C32T	32C64T	48C96T	51C102T
🟦 Non Optimized SW	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
🟥 Optimized SW	3.82	3.33	3.19	3.26	3.46	3.49	3.50	3.46	3.46
🟩 Intel® QAT Acceleration	6.32	5.77	5.79	5.78	5.80	4.77	2.40	1.61	1.44

As Figure 6 demonstrates, offloading ECDHE-X25519-RSA2K to Intel® QAT provides the highest performance per core. That is, Intel® QAT provides up to 6.3x higher performance than default software stack for 1-core, 2-threads, and up to 1.5x higher than using the optimized software stack for 12-core, 24 threads. It must be noted that Intel® QAT has a peak performance, and it will be hitting this point for ECDHE-X25519-RSA2K by 16 cores. Once Intel® QAT reaches its maximum performance per device, its performance will plateau.

Figure 7. NGINX Webserver Handshake Performance per Watt

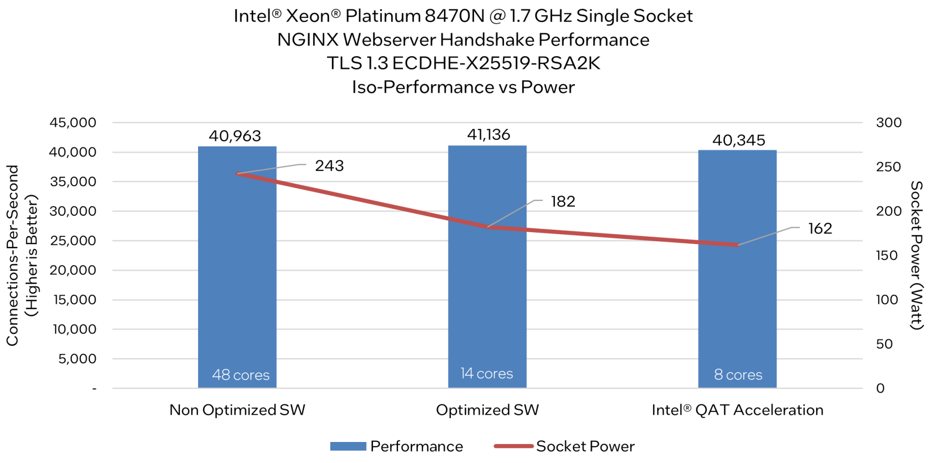


NOTE

Optimized SW 14C28T performance and power is an estimation to highlight performance around 40k connections-per-second. This estimate was extrapolated using average performance per core and power per core from collected data.

Socket power was captured using EMON, which is a low-level command-line tool that provides the ability to profile application and system performance. The tool leverages counters from hardware Performance Monitoring Units (PMUs) to collect performance monitoring events. Figure 8 was created by calculating performance per watt, from socket power, and normalizing the resulting perf/watt by dividing by default software stack's result. Looking at the normalized performance per watt, Intel® QAT delivers up to 6.3x higher performance compared to default software stack for 1-core, 2 threads, and up to 1.5x higher than using the optimized software stack for 12-core, 24 threads. Keep in mind, this is a focused look showing performance per watt prior to Intel® QAT hitting its saturation point. To summarize, Intel® QAT provides higher performance with lower power consumption.

Figure 8. NGINX Webserver Handshake Iso-Performance vs Power



NOTE

Optimized SW 14C28T performance and power is an estimation to highlight performance around 40k connections-per-second. This estimate was extrapolated using average performance per core and power per core from collected data.

Intel® QAT having a higher performance per core and performance per watt results in real world core and power savings. This is demonstrated in Figure 8. Intel® QAT can achieve around 40k connections-per-second with 6 fewer cores and 20 W fewer than optimized software.

1.7 Results

Test data shows that on the 4th Gen Intel® Xeon® Scalable Processor family, integrated Intel® QAT has proven significant performance advantages over non-optimized software solutions, with NGINX connections-per-second performance up to 6.3x using TLS 1.3 ECDHEX25519-RSA2K for 1-core, 2-thread. Additionally, by enabling these optimizations through a single engine/provider into OpenSSL, the overall solution can attain full platform level performance by leveraging the best combination of implementations available.

Because Intel® QAT has a higher perf/watt than both default configuration (non-optimized software and with optimized software), it can achieve better performance, while reducing total cost of ownership (TCO) and the need to scale up datacenter clusters to meet demand, while also providing power savings of up to 20W.

1.8 Conclusions

The 4th Gen Intel Xeon® Scalable Processor family provides significant performance gains per core, and TCO benefits when utilizing Intel® QAT, for TLS connection establishment. Performance is delivered using a mainstream release of OpenSSL and extensions to NGINX to allow for asynchronous processing of TLS handshake operations, plugging in accelerated cryptographic implementations through a standard provider/engine into OpenSSL.

As shown, this solution delivers up to 6.3x times better TLS-1.3 handshake performance when compared to the OpenSSL software implementation for the key exchange ECDHE-X25519-RSA-2K.

Legal and Disclaimers

Performance varies by use, configuration and other factors. Learn more on the Performance Index Site.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software, or service activation.