SOLUTION BRIEF
Intel® QuickAssist Technology

Accelerating OpenSSL* Using Intel® QuickAssist Technology

Modifications to open source implementation yield dramatic performance improvement.

Wider Use of Encryption

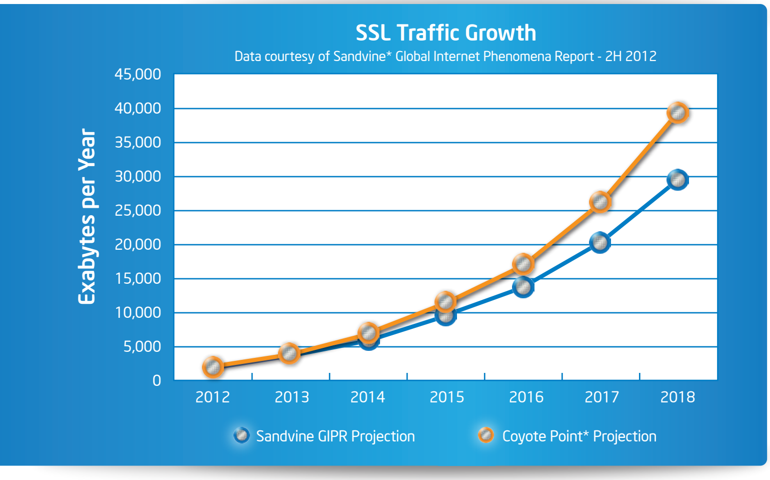
Historically, the demand for secure data transmissions over the Internet was driven primarily by institutions conducting e-commerce and banking transactions. Today, the volume of secured communications is skyrocketing, as personal information of all sorts is being encrypted by applications like Gmail*, Twitter*, and Facebook* using the HTTPS protocol. As a result, servers in data centers, telecom networks, and enterprises are expected to handle increasing amounts of traffic using the Secure Sockets Layer (SSL) protocol, increasing compute requirements. Industry forecasts, shown in **Figure 1**, suggest SSL traffic could increase by as much as 16-fold from 2012 to 2018.[1]

Open Source Optimization

The OpenSSL project[2] provides an open source implementation of the SSL/TLS[3] protocols and is a commonly deployed library for SSL/TLS world-wide. The SSL/TLS protocols consist of two phases: an initial session-initiation/handshake and a bulk data transfer.

Over time, the implementation has been modified to increase performance, including contributions by Intel that speed up the cryptographic workloads in both phases. OpenSSL performance can be further improved with Intel QuickAssist Technology, which is supported by select Intel architecture platforms.

Equipment manufacturers who utilize the current version of OpenSSL will benefit from the optimizations already integrated in the release, as well as the straightforward installation process. Not surprisingly, contributors to OpenSSL continue to work to make the implementation more efficient, and one of these efforts is referred to as Asynchronous OpenSSL, which is expected to be delivered as a patch in a development branch (i.e., parallel development project) with the potential of many-fold performance increases.



SSL Traffic Growth

Data courtesy of Sandvine* Global Internet Phenomena Report - 2H 2012

Figure 1. Projected Growth in SSL Traffic 2012 to 2018

This solution brief discusses two models for increasing the performance of OpenSSL* on Intel® architecture, and is one in a series of five briefs describing how to maximize the benefits from Intel® QuickAssist Technology. Please see the Resources section for links to the series.

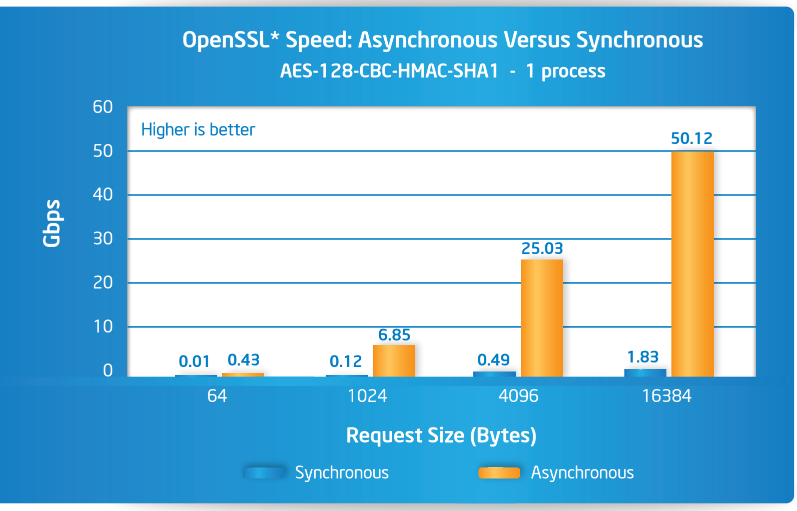


Figure 2. OpenSSL* Speed: Asynchronous Versus Synchronous[3]

Asynchronous OpenSSL*

The standard release of OpenSSL is serial in nature, meaning it handles one connection within one context. From the point of view of cryptographic operations, the release is based on a synchronous/ blocking programming model. A major limitation is throughput can be scaled higher only by adding more threads (i.e., processes) to take advantage of core parallelization, but this will also increase context management overhead.

Asynchronous OpenSSL is a non-blocking approach that supports a parallel-processing model at the cryptographic level for SSL/TLS protocols, which in turn allows for other types of optimizations. Although the concept of a non-blocking mode of operation is not new within OpenSSL, it previously had not been applied to cryptographic transforms. This capability allows cryptographic transforms to be processed on dedicated hardware engines or on separate logical cores, thereby allowing the protocol stack and applications to run other tasks unencumbered. Asynchronous operation also enables single-threaded applications to efficiently handle multiple SSL connections because individual flows are not blocked when using hardware acceleration.

Two major benefits of asynchronous OpenSSL are increased single-flow throughput, leading to maximum performance, and fewer contexts, thus reducing context management overhead.

Performance Results

Preliminary test results demonstrate that asynchronous OpenSSL running on an Intel architecture platform with Intel QuickAssist Technology is significantly faster than the standard (synchronous) release. **Figure 2** compares the OpenSSL speed for the two versions at various packet sizes running on one process. These measurements are indicative of the throughput during the bulk data transfer phase.

For 64 byte requests, asynchronous operation is about 49 times faster than synchronous, and over 7.5 times faster for 16,384 byte requests.[4] When the number of processes is increased to two and four (**Figure 3**), the difference in performance decreases somewhat, with asynchronous being 20 and 2 times faster than the standard release for 64 byte and 16,384 byte requests, respectively.

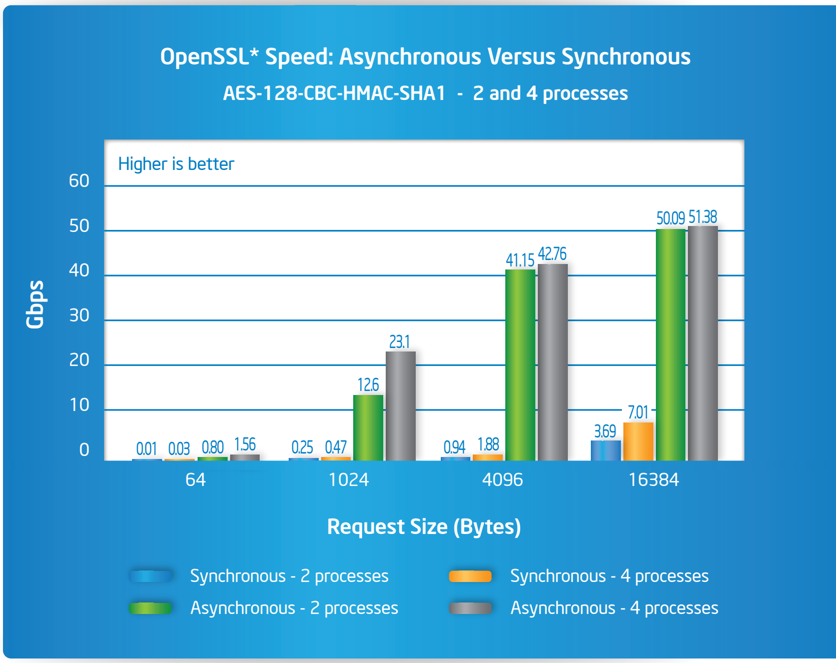


Figure 3. OpenSSL* Speed: Asynchronous Versus Synchronous With Two and Four Processes[3]

Figure 4 shows the number of RSA 2028 bit sign operations per second for asynchronous and synchronous OpenSSL. This operation is used during the initial session-initiation/handshake phase. The key finding from this test is asymmetric cryptography produces full throughput with only one process (i.e., context), whereas the synchronous version delivers significantly less throughput with four processes. Consequently, synchronous OpenSSL may require significantly more cores than asynchronous OpenSSL to achieve the same performance.

Simplifying Accelerator Integration

With more and more traffic being encrypted, servers and security appliances will rely more heavily on accelerators to offload cryptography workloads. For this reason, Intel is working with the OpenSSL Software Foundation to optimize its implementation for use with hardware accelerators, such as those provided by Intel QuickAssist Technology. In addition, the findings from this effort will be used to optimize the performance of proprietary SSL/TLS-based solutions running on Intel QuickAssist Technology-enabled platforms.

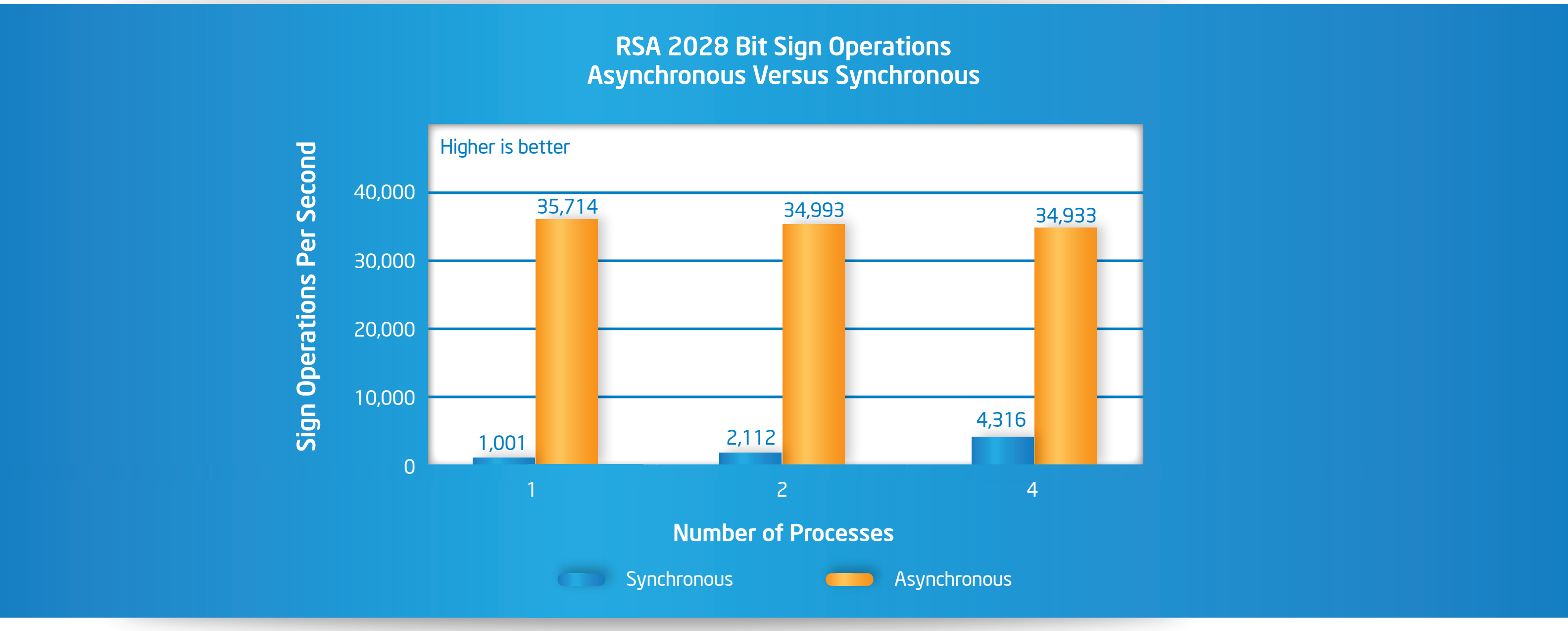


Figure 4. RSA 2028 Bit Sign Operations: Asynchronous Versus Synchronous[3]

Resources

Solution Brief Series: Intel® QuickAssist Technology

Part 1: Integrated Cryptographic and Compression Accelerators on Intel® Architecture Platforms

Part 2: Bridging Open Source Applications and Intel® QuickAssist Technology Acceleration

Part 3: Accelerating OpenSSL* Using Intel® QuickAssist Technology

Part 4: Accelerating Hadoop* Applications Using Intel® QuickAssist Technology

Part 5: Scaling Acceleration Capacity from 5 to 50 Gbps Intel® QuickAssist Technology

Appendix A: Platform Configuration

Hardware Platform Configuration	
Processor	Dual socket Intel® Xeon® processor E5-2680 v2
Chipset	Intel® Communications Chipset 8950 (Intel® DH8920 PCH)
Memory	DDR3 1333MHz
Operating System	Linux* version 3.1.0-7.fc16.x86_64
Compiler	gcc version 4.6.2 20111027 (Red Hat 4.6.2-1) (GCC)

For For more information About Intel QuickAssist Technology, visit
http://www.intel.com/content/www/us/en/io/quickassist-technology/quickassist-technology-developer.html

[1] Global Internet Phenomena Report: Sandvine,2012

[2] OpenSSL: http://www.openssl.org

[3] The TLS protocol: http://www.ietf.org/rfc/rfc2246.txt

[4] Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel® products as measured by those tests. Any difference in system hardware or software design or configuration, as well as system use patterns including wireless connectivity, may affect actual test results and ratings.

Printed in USA   MS/VC/1113   Order No. 329877-001US