



QuickAssist Technology to Accelerate



Feb. 2015

ABSTRACT

NGINX (pronounced, "engine-x") is considered to be the fastest growing web server worldwide [1]. It is an open source project backed by commercial source branch, with pretty good software design (asynchronous single thread) and features (server, reverse proxy, load balancer, etc.). NGINX is both feature rich and powerful. It exhibits ease of use to the administrator, along with a better than average performance figures. In fact, there are many "switches", "levers", "slides" and "button" one can push and pull, to tune up and optimize server's performance [2] to best fit a given platform. However, tuning NGINX operation it only half of optimization that can be implemented, for SSL crypto computation will always wait patiently around the corner to come and consume considerable CPU power; especially when it comes to asymmetric encryption, crypto key exchange or key signing. Looking further ahead, things will only get more serious, when SPDY will evolve to be the de-facto standard web protocol. Considerably larger portion of data will traversed encrypted through the net, and SSL handshakes and re-handshakes will be more and more ubiquitous.

Keywords

NGINX; SSL Offload, Acceleration, SPDY, man-in-the-middle

1. INTRODUCTION

The primary goal of the benchmarks and tests described herein is to exhibit and demonstrate how NGINX HTTPS service is **increased** and **optimized** in performance, through the use of Intel® Coletto Creek 8955 acceleration chip set. The specific areas where an acceleration is expected are demonstrated. Finally, an analysis is reached regarding the measured capacity improvement, against the **cost of acceleration and offload**.

A "near real life" data of SSL acceleration options is required more and more, mainly among application vendors whose application is expected to massively deal with encryption, facing upcoming HTTP/2.0. Formerly considered an attack, "man-in-the-middle" increasingly becomes the mode of operation for more and more lawful interception operators.

Nominal benchmark results are the first reference for assessment of how far can we further go to optimize real life like test setup. These nominal results for the current Intel® Coletto Creek SKUs (the 8950 and 8955) are presented in table 1

In near real life scenario, however, the figures described in Table 1, translate to more abstract gauges. The RSA operations rate translates to **SSL handshakes or connections per second**, while the bulk crypto figure translates to **SSL or HTTPS bandwidth**.

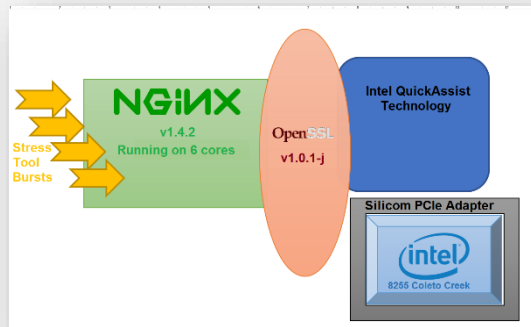
Table 1 – Intel® Coletto Creek Nominal Performance

#	Intel® Coletto Creek SKU	RSA 1K ops/sec*	RSA 2K ops/sec*	AES128 Crypto**
1	8950	165K	35K	50Gpbs
2	8955	190K	40K	50Gpbs

* Asymmetric cryptography

** Symmetric cryptography

Figure 1 - Software Components Integration



Cost of Offload

Operating an offload engine require transferring data to and from the accelerator. Indeed, DMA operation over PCIe bus, where the accelerator card is the DMA master, relive large part of buffer management off the host CPU; but still, CPU cycles needs to be vested to manage this path.

Offloading a task off CPU to a sub engine, therefore, is beneficial if such use lowers significantly the CPU cycles that are spent for the cryptography task; relieving valuable CPU cycles for business logic processing. A manual note to the GNU/Linux 'top' utility was used in this test to determine CPU usage during loads, with software or with hardware offload engine.

Once series of test is completed and both SSL handshake and bulk crypto is benchmarked, it would be interesting to see if both type of offload exhibit same gain in terms of gain and benefit.

3. TESTS

3.1 Outline

The following setup was used for the tests.

Server	Supermicro X9DRD-7LN4F, CPU 2xE5-2670 v2, 128Gb RAM
OS	Fedora16 x86_64 kernel 3.0.1
QuickAssist	QAT1.6.L.1.0.9-22
OpenSSL	1.0.1h
NGINX	1.4.2, patch nginx-1.4.2-005
Adapter	Silicom PE3iSCO3

Stress traffic was generated with a Spirent layer 7 packet generator.



Figure 2 - Silicom PE3iSCO2 with Intel® ColettoCreek 8950 Chip Set

3.2 Results – SSL/TLS Handshakes

Let's start with the bottom line. Intel® 8955 Coletto Creek chip set brings considerable added value under real life scenario. This fact, however has to be further detailed. The method of the test that was carried out herein, was to test a software only implementation, against an implementation that incorporates the Intel® 8955 Coletto Creek chip set. However, while NGINX as a software only implementation exhibited fair load balance and distribution across all incorporated CPU cores (maximum of 8 cores), same NGINX setup, but with Intel® 8955 Coletto Creek chip set as an offload crypto engine did not exhibit same fair load balance and distribution. Nevertheless, the Coletto Creek operation has brought:

- a. **Significant performance improvements**
- b. **Significant CPU relief**

The test still has a length to cover, but even under far from ideal conditions, the power of the Intel® 8955 could be demonstrated.

4. Where to go from here

The target is to get **close to the nominal capability** of the acceleration engine as brought herein in Table 1. There is a lot to cover down the road. Tests will be continued with more efficient NGINX setup, to show that even with tuned software implementation, and offload engine for encryption still brings considerable value, especially for the heavy lifting tasks, of the asymmetric cryptography.

A fair estimate, based on several other tests, points at 10,000 RSA 2K SSL/TLS handshakes per second as a cap number for typical user space software implementation on a server similar to the one that was tested in this test; with all core screaming 100% utilization.

Therefore, based on the results brought herein, in Table 2, specifically from the CPU relief that was enabled by the offload engine, it is fair to estimate that in more optimal application set up, the above number of 10K handshakes, could be doubled, and perhaps even more.

Table 2 - Interim Results for SSL/TLS Handshakes Benchmarks

1 x GET https://193.0.0.1/1.html	RSA 2K key	RSA 4K key
		AES128-SHA
QuickAssist with Intel® 8955	21,376	4,130
	CPU usage: CPU 0-7: 80%; CPU 8-19: 0%	
Software only	4,866	2,072
	CPU usage: CPU 0-19: 100%	

We need to further check core affinity, QPI buss avoidance, no question. But even with such non optimal setup, Intel® 8955 Coletto Creek chip set has great value:

- 1) It literally doubles SSL/TLS handshake rate with 2k and 4k keys;
- 2) It relieves considerable amount of CPU cycles.

And when considering that the relieved CPU cycles are of an Ivy bridge Intel® microarchitecture running at 2.5 GHz, the added value of the accelerator immediately appear.

3.3 Results – Bandwidth

Yet to be completed next. More optimal setup would be configured for meaningful tests results.

To gain better results out of the setup, the following areas are to be more thoroughly observed:

- 1) Incoming traffic **load balancing** across CPU cores and core affinity scheme should be tightened;
- 2) Not using HTTP request methods at all, for instance, not sending GET request at all. Later on a 1 byte resource request would be added in controlled manner.
- 3) Two threads per core with and without CPU hyper threading enablement would be tested.
- 4) TCP tuning would be revisited.

In a more optimized setup, a bandwidth tests would be carried out as well.

Further down the road, same tests would be expanded to ECC cryptography. Moreover, dual chip adapters, as well as quad chip PCIe adapters, that are already available by Silicom, would be tested, to demonstrate linear scalability.

REFERENCES

[1] NGINX Now Powers 146 Million Websites, Launches Version 1.6 and 1.7 Of Its Web Server (Apr, 24th, 2014)

<http://techcrunch.com/2014/04/24/nginx-now-powers-146-million-websites-launches-versions-1-6-and-1-7-of-its-web-server/>

[2] Tuning NGINX for Performance (Oct. 10th, 2014)

<http://nginx.com/blog/tuning-nginx/>

[3] NGINX SSL Performance (July, 2014)

<http://nginx.com/wp-content/uploads/2014/07/NGINX-SSL-Performance.pdf>