Performance Analysis of Cryptographic Acceleration in Multicore Environment

Yashpal Dutta and Varun Sethi

Freescale Semiconductor Inc.
Noida, Uttar Pradesh-201301, India
{yashpal.dutta,varun.sethi}@freescale.com

Abstract. With the increased capability to meet processing requirements and convergence of multiple servers, Multicore platforms are getting popular in the embedded space. Seamless performance scaling is assumed by a system designer while migrating to a Multicore system. This may not be true, especially with the ever increasing cryptographic requirement of security servers in embedded space. Cryptographic computational requirements are being pushed beyond the capabilities of general purpose processors. Thus many of the advanced Multicore platforms also provide hardware cryptographic accelerators. On Multicore platforms it's possible to use the crypto accelerator in a SMP or an AMP configuration. In case of SMP configuration, OS controls the cryptographic accelerator sharing across multiple applications. In a virtualized environment the crypto accelerator can be shared across multiple guest operating systems under the supervision of the hypervisor. Hypervisor utilizes the services of an IOMMU to isolate crypto operations and data across various guest OS partitions. A proper analysis of each of the design configurations is required in order to select the best possible option while designing security server over an embedded system. The paper covers cryptographic processing for security servers on SMP Linux and in a virtualized environment (running with a hypervisor [6]).

Keywords: Cryptography, OpenSSL, Cryptodev, Multicore, Hypervisor.

1 Introduction

In an embedded environment, cryptographic processing plays a critical role. The cryptographic applications range from the ones requiring basic cryptographic operation like AES, TDES or MD5 to ones requiring complex protocol level processing like Apache Web Server[5] working with SSL/TLS protocols or Strongswan[6] providing IPSec stack working with cryptographic algorithms for VPN. The cryptographic operations generally consume a lot of CPU cycles. A single core platform becomes bottleneck in security server applications and thus there is a need of multiple servers working in a load sharing environment to meet expected processing requirements. Multicore platform helps in convergence of multiple servers in a single system. CPU may not always be a bottleneck when cryptographic

K. Singh, A.K. Awasthi, and R. Mishra (Eds.): QSHINE 2013, LNICST 115, pp. 658–667, 2013.

operations are performed in a system, but critical platform resources like cache, memory modules and platform bus may also contribute to performance challenges.

In case of a SMP operating system, hardware resources are shared across multiple user-space processes. In hypervisor controlled virtualized configuration, it is possible to both physically partition and share resources across various guest OS partitions. The resources on a server platform include CPUs, memory, Caches and IO devices including crypto accelerators. The cumulative server performance improves with optimized resource partitioning.

In this paper we discuss cryptographic processing design options available for SMP and AMP configurations on Multicore platforms. We specifically look at ways cryptographic processing can be optimally offloaded to cryptographic accelerator in each of these configurations. The paper concludes with the experimental results, pros and cons of security server running on Multicore platform under various design approaches.

2 Building Blocks

The section covers software and hardware components involved in our experiments under various configurations.

2.1 OpenSSL Library

OpenSSL[2] is an open source toolkit for SSL2.0/3.0, TLS1.0. OpenSSL is one of the standard user-space cryptography library supporting large number of general purpose symmetric ciphers operations (AES, DES, TDES etc), digests operations (MD2, MD4, MD5, SHA1 etc) and asymmetric cipher operations (RSA, DSA, DH etc). It provides interface to offload cryptographic operations to hardware accelerator with its engine interface.

Cryptodev [3] engine is one of the engine interfaces used by OpenSSL for offloading cryptographic operations to hardware accelerators. The module supports multiple hardware accelerators registered with CryptoAPI. Other engine available for offloading cryptographic operations from OpenSSL library to hardware accelerators are proprietary engine interface, AF_ALG and OCF-Linux [4].

The CryptoAPI infrastructure provides cryptographic operation handling within Linux kernel. The infrastructure includes a mechanism to register and offload cipher operation supported by a cryptographic accelerator driver. If multiple drivers support the same crypto operation, the driver with the highest priority is selected for the operation. Support exists for handling symmetric ciphers and digests in the Linux kernel. This support is available in Linux kernel version 2.6 onwards.

Cryptographic accelerator driver initializes the hardware accelerator and registers supported cipher operations to CryptoAPI infrastructure in Linux kernel. Cryptographic operation is only requested by hardware accelerator if operation is registered by driver with CryptoAPI. Figure 1 below shows various layers involved in OpenSSL cryptographic processing in our experiments.

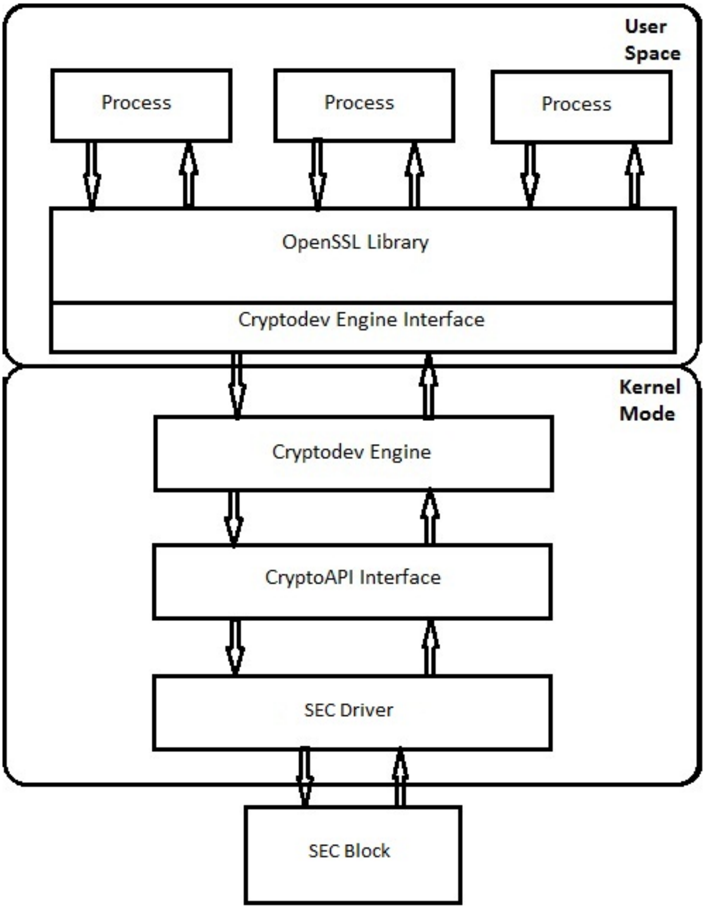

Fig. 1. OpenSSL Library with Cryptodev Engine

2.2 Cryptographic Acceleration

A cryptographic hardware accelerator offloads actual math operation related to cryptographic operation. The accelerator may implement common cryptographic processing including symmetric Cipher operations (e.g. AES, DES, TDES etc), Digest operations (MD5, SHA1, SHA-256 etc), Public key Ciphers (RSA, DSA, DH etc), Protocol Offloading (e.g. IPSec, SSL, TLS, DTLS etc) and random number generations.

For this paper, we used Freescale Multicore platform with integrated security accelerator called SEC. Figure 2 shows high level block diagram of SEC block. Parallel sub-blocks processing multiple cryptographic Job's in parallel help in scaling performance. The Job Queue controller unit checks whether the cryptographic jobs can be processed in parallel. The decision depends on whether there is inter-dependency among two cryptographic jobs. This is true for situations like multi-pass hashing on a sequence of buffers requiring init, update and final-state processing.

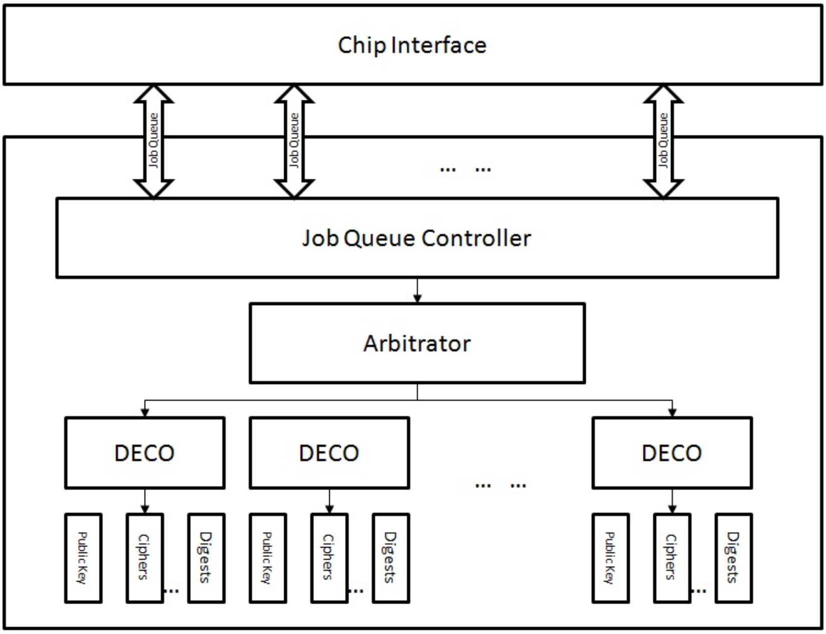

Fig. 2. Freescale SEC Block

2.3 Hypervisor

Freescale Embedded Hypervisor [6] is used to run OS in supervised configuration. This provides easy porting of an OS to hypervisor with minimal performance impact.

3 Cryptographic Configuration Options

3.1 SMP Configuration

In SMP configuration, all the processing threads use cryptographic support provided by cryptographic accelerator. The system resources like system bus and caches are used without any partitioning [8]. Thus, a lower priority security process can starve high priority process by consuming shared system resources. E.g. cached data corresponding to a high priority flow may get evicted by cache line corresponding to a low priority flow. Such issues can be mitigated by proper QoS implementation at hardware level to handle high priority crypto requests before lower priority process. Figure 3 shows cryptographic accelerator access by multiple processing running under SMP configuration.

The performance of a cryptographic process is impacted by presence of other processes under SMP configuration which can impact scheduling of the processes on CPUs. Proper configuration of scheduling parameters of various processing running on CPU helps mitigate performance impact.

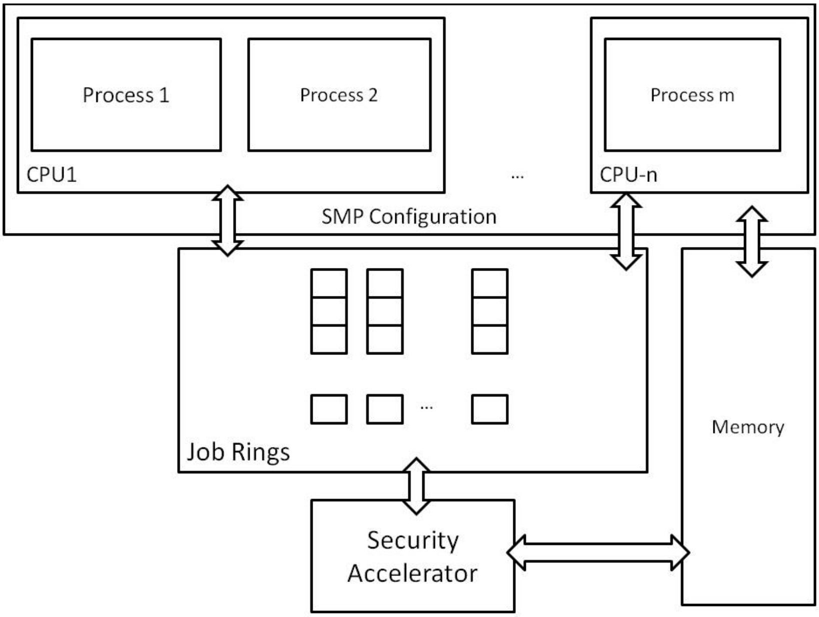

Fig. 3. Security accelerator sharing in SMP Configuration

3.2 Virtualized Configuration

In a virtualized environment the cryptographic accelerator can be shared across multiple guest operating systems under the supervision of the hypervisor. Hypervisor utilizes the services of an IOMMU to isolate cryptographic operations and data across various guest operating system partitions. IOMMU ensures that unless memory is shared, no two partitions can access each other's memory. An attempt to access memory of other guest partition is checked and violation is raised by IOMMU.

The partition virtualized environment offloads its cryptographic acceleration requirements with the notion that it owns the block. If system resources could be partitioned among guest partitions, the issues like cacheline eviction by unrelated partition are avoided and thus reduce interference among partitions. Figure 4 shows a virtualized configuration under hypervisor control.

One issue with such a partitioning is that the global configuration space associated with security block must be owned by one partition which can work as a control partition for the platform. Assumption made by all partitions offloading their security operation to security block is that security block is already initialized by control partition.

Access to hypervisor controlled resources like MMU or interrupt controllers can lead to significant overhead, thus impacting performance of the cryptographic applications.

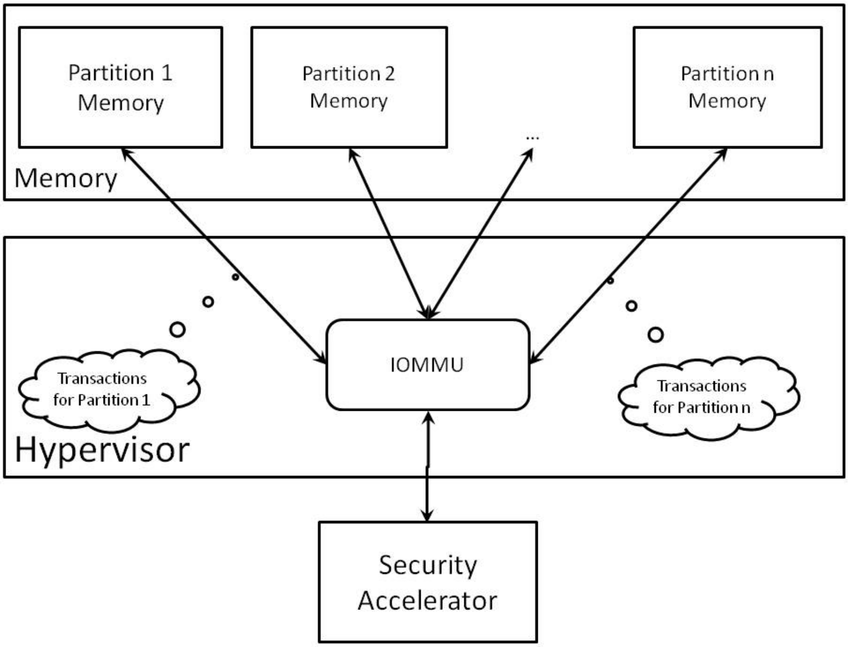

Fig. 4. Virtualization of Security hardware accelerator

4 Experimental Setup

We performed our experiments on Freescale QorIQ P4080 Multicore platform. This platform has eight e500mc cores each running at 1.5GHz. Each core has 32KB L1 instruction and data cache and a 128 KB unified backside L2 cache. The platform has a shared 2MB platform cache. For our experiments, we used a system with 4GB DDR with DDR bank interleaving. The crypto accelerator block used is the Freescale security block IP.

4.1 Setup for the SMP Configuration

The experiments were performed on Linux kernel version 3.0.48. OpenSSL version 1.0.1c and cryptodev version 1.5 were used for the experiments.

4.2 Setup for the AMP configuration

The virtualized setup consisted of lightweight baremetal executive guest partitions running on Freescale embedded hypervisor.

5 Performance Results

5.1 Results on Native SMP Configuration

The results are obtained using speed test built-in OpenSSL with and without cryptodev engine. Results show that performance gain varies for different size of buffer and number of parallel threads. The results in figure 5 shows MD5 and SHA1 performance comparison for offloaded operation against software based cryptographic implementation with single thread of speed test.

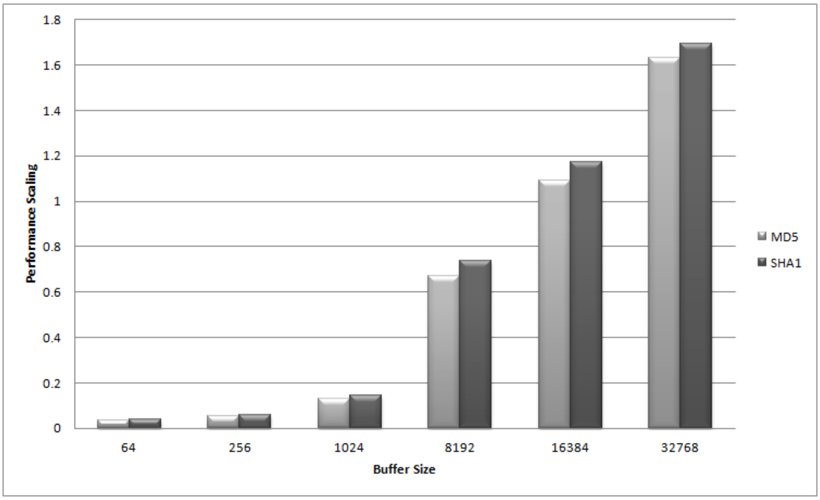

Fig. 5. MD5 and SHA1 Performance Scaling

Figure 6 shows performance scaling results for AES-CBC and DES-CBC cryptographic operations. From experimental results in graphs, we can see that cost of offloading gives less CPU bandwidth saving for small sized packet than large sized packets.

In presence of multiple accelerator sub-blocks capable of performing parallel processing of different and unrelated cryptographic operations, the performance of application threads can be improved.

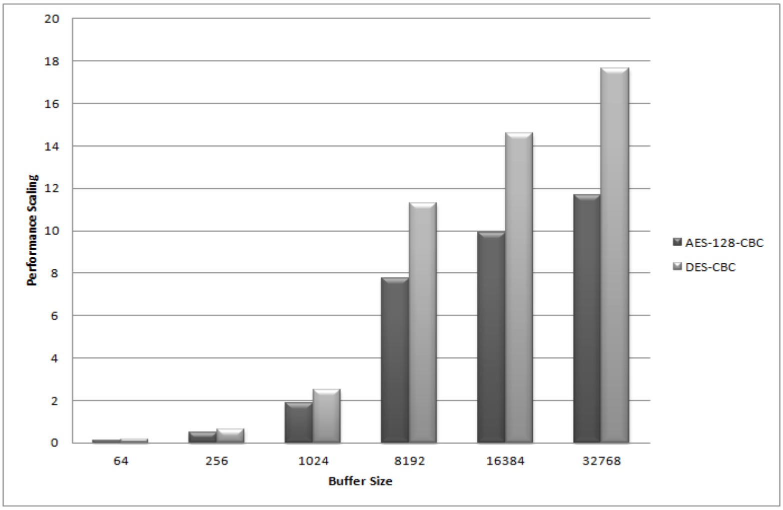

Fig. 6. Performance scaling of single threaded AES-128-CBC and DES-CBC

Figure 7 above shows performance scaling with security application threads running in parallel natively on Multicore platform.

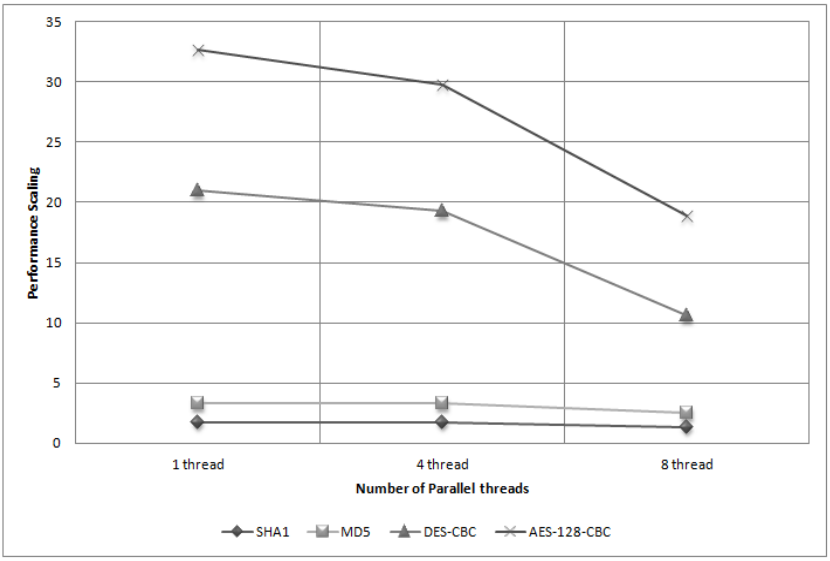

Fig. 7. Performance scaling with multiple threads

5.2 Results on Supervised AMP partition

Figure 8 below shows performance scaling with a proprietary security application running under Freescale Hypervisor. The application exercises security accelerator for IPSec protocol processing. Performance scales almost linearly for small sized frames but for large size frame, the performance scales for 3 partitions and additional cores/partition don't give performance benefit due to bandwidth limitation of hardware accelerator.

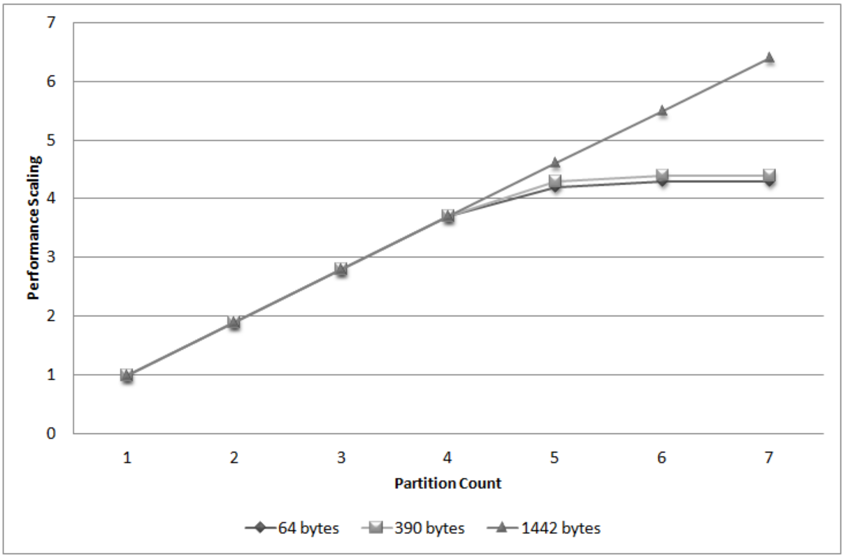

Fig. 8. IPSec Performance scaling of Virtualized Security Block

6 Conclusion

On Multicore platforms, there are multiple options available for Security Server design. Convergence of multiple security servers under SMP and virtualized configuration is possible. The sections above looked at both configurations on Freescale Multicore platforms. The paper also shared performance scaling of OpenSSL library on Multicore platform with and without hardware acceleration. It could be seen that performance scaling stops beyond a number of CPUs with hardware accelerator. This is the point where CPUs were generating more cryptographic operation requests for hardware accelerator than it could handle and there is a need for congestion support to avoid overloading on cryptographic acceleration [8]. Virtualization of cryptographic accelerator helps in isolation and security of application domains running in separate partitions. This separate traffic domain in virtualization for a partition reduces performance impact on traffic running on other partition which is lacking in SMP Linux.

References

1. Freescale P4080 QorIQ processor: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=P4080
2. OpenSSL project: http://www.openssl.org/
3. Cryptodev engine: http://home.gna.org/cryptodev-linux/
4. OCF-Linux project: http://ocf-linux.sourceforge.net/
5. Apache Web Server: http://www.apache.org
6. Freescale Embedded Hypervisor: http://cache.freescale.com/files/32bit/doc/white_paper/EMBEDDED_HYPERVISOR.pdf?fsrch=1&sr=2
7. Strongswan project: http://www.strongswan.org/
8. Dutta, Y., Malik, S.: Hemant Agrawal: Multicore Development Challenges in Embedded Space, ARM TechCon-2012 (2012)