Kernel level AES Acceleration using GPUs

Ahmed Mousa
Outline

• Problem definition
• Motivation
• Objective
• Approach
• Related Work
• References
Problem Definition

• Often times encryption is not the main focus of an application, rather it is something that the application provides as part of its service, an example of this is OpenSSL [2].

• Problem Definition:
  • Performing AES Encryption is a computationally expensive operation[1], the process of encrypting and decrypting data can take up a large percentage of the CPU's time which may affect the whole system’s performance.
  • Increasing the bit count of our keys/encryption may even slow down performance more to a non sustainable level, although this is vital nowadays since with the new hardware accelerators available, cracking of current means of encryption even using brute force became feasible in some cases.
Motives: Why is acceleration is needed for cryptography?

• Better security mainly for data transmission and encryption (since we can use AES 192 or AES 256 instead of AES 128 at the accelerated solution)
• Better performing web servers (Enhancing SSL speed)
• Virtual Private Networks (Accelerating IPsec VPNs)
• Storage Area Networks (Encryption of data transmission)
• Pay TV (Securing pay TV through tamper resistant service)
Objective

• Alleviate the amount of work done by the CPU by offloading AES encryption to GPUs, since the process of encrypting and decrypting data can take up a large percentage of the CPU's time.

• Implement the accelerated algorithm on GPUs at the kernel level layer in order to avoid overheads from user space and provide abstraction to the services using this acceleration.
Approach

• Leverage the usage of modern GPU frameworks in providing an accelerated algorithm using the full capabilities of modern GPUs.
• Optimize memory transfers between GPU memory layers and CPU ram to minimize overheads.
• Auto-tune the accelerated parameters based on GPU model and architecture.
Approach

- Integrate the GPU as a driver abstracted in the OCF (OpenBSD Cryptographic Framework) providing an accelerated version of the AES algorithm in it (and other algorithms in future work).
Related Work


• They show that AES, a popular example of a symmetric-key function, can be competitive with the CPU on recent GPUs and outperform on contemporary GPUs.
Related Work

• 2) GPU accelerated cryptography as an OS service, Harrison, O., & Waldron, J., Springer Transactions on Computational Science XI (2010).

• This paper investigates the integration of GPU accelerated cryptographic algorithms with an established service virtualization layer within the Linux kernel, the OCF-Linux framework
Related Work


• The authors here acknowledges that a traditional AES CUDA implementation does not specify efficient techniques to utilize the GPU parallelism. As a result, in this research, they evaluate the recent proposed parallel AES implementations over GPU
References