Regaining Lost Cycles with HotCalls: A Fast Interface for SGX Secure Enclaves

Ofir Weisse Valeria Bertacco Todd Austin
University of Michigan
{oweisse,vale,austin}@umich.edu

Cloud computing allows lowering the cost of computation and storage, outsourcing the acquisition and maintenance to a third party. Using hardware and software under the control of a third party implies substantial trust: trust that the service provider will not snoop on the data on its servers and will not tamper with the execution flow. Even if the cloud provider can be trusted not to actively snoop or tamper with processed data, there is implicit trust in the operating system, the virtual machine manager, and the firmware (BIOS & System Management Mode code - SMM). A compromise in the security of any of these, by means of remote attack or rogue employee tampering with the hardware, leads to the compromise of the information and execution on the cloud.

In 2015, Intel released Skylake micro-architecture, the first x86 production processor featuring a secure execution technology - Software Guard Extensions (SGX). This technology allows secure execution in user-space (ring 3) in a container called a secure-enclave which is shielded from the OS, VMM, and SMM. Ideally, no vulnerability or intentionally malicious code in any of these layers should compromise the confidentiality or the integrity of the secure-enclave. No probing of physical buses outside the processor chip should compromise the security, as the memory is encrypted as well.

This work gives the first taxonomy of the operations involved in using the SGX framework and their costs in cycles. Through these observations, we offer a performance boosting alternative interface to interact with secure-enclaves. We found that the overhead of calling a secure-enclave function takes between 8,600-17,000 cycles (depending on cache state), compared to 150 cycles for a regular OS syscall, and compared to 1300 cycles for a hyper-call in a KVM virtualization solution. We also found that the mechanism allowing secure code to interact with the application or OS outside the enclave incurs between 8,200-17,000 cycles (depending on cache state). This is a 54x-113x degradation in performance compared to OS calls.

The overhead of SGX-related calls becomes a significant bottleneck in applications with high system call frequency. For instance, a database application serving 200,000 requests per second (*e.g., Memcached*) requires at least 200,000 system calls to send the responses on the network. According to our measurements, each call consumes at least 8,200 cycles, totaling 1,640 million cycles. On a 4 GHz core, this amounts to 41% of the core time spent on merely facilitating the calls, without doing any actual work. Our evaluation of non-trivial applications shows that this is not a hypothetical problem.

Identifying that context switches used for facilitating system calls are a major bottleneck in SGX applications, we design and implement *HotCalls* - an alternative interface for calling enclave functions and requesting system calls by the enclave. *HotCalls* are based on a spin-lock synchronization mechanism, and provide more than an order of magnitude speedup. Compared to the standard SGX SDK framework, *HotCalls* cost only 620 cycles in most cases, a 13-27x improvement.

We evaluate the performance of three non-trivial applications within SGX: *OpenVPN* (encrypted tunnel), *Memcached* (memory based database), and *Lighttpd* (fast HTTP server), using a straightforward approach to port them into SGX secure-enclaves. We show that using *HotCalls* it is possible to improve throughput by 2.6-3.7x and reduce the applications' response latency by 62-74%.

To summarize, we make the following contributions:

- Identify and analyze fundamental operations in SGX technology which have major performance implications. We provide the first comprehensive evaluation of the latency of each such operation, by designing and running a set of micro-benchmarks. Based on the micro-benchmarks' results, we offer best practices for using SGX when performance is just as important as security.
- Leveraging the insights from the micro-benchmarks, we design and implement an alternative calling-interface to SGX, *HotCalls*, for communication between secure-enclaves and untrusted code, which is 13-27x faster than the existing mechanism provided by the SGX SDK. Source code of *HotCalls* is available at online (will be up soon).
- We evaluate the benefit of *HotCalls* on widely used applications: *OpenVPN*, *Memcached*, and *Lighttpd*, showing that the throughput of these applications can be improved by a factor of 2.6-3.7x and the response latency can be reduced by 62-74%.