Evaluating and Optimizing Hardware Enclaves for Overloaded Systems

Demetrius Billey, demeb@nmsu.edu;
Naveed UL Mustafa, num@nmsu.edu
Computer Science Department, New Mexico State University

Abstract

This study aims at understanding Security Service Engines (SSE) in managing system call overheads for applications running in hardware enclaves. SSE pairs each enclave with a lightweight responder core to handle system calls with objective of reducing cost of OS-interaction. Static coupling of responder cores with enclaves can lead to resource underutilization and performance degradation in overloaded systems. To address these challenges, this work intends to measure the SSE's efficiency by running workloads in an overloaded system, analyze both the results and SSE approach to identify performance bottlenecks and then propose design optimizations to achieve performance scalability.

Introduction

- **Secure Process Technologies**: Use hardware enclaves to protect programs on untrusted remote servers.
- **OS Interaction**: Programs within enclaves must often interact with the operating system, which involves costly transitions (exiting and re-entering the enclave) [1].
- **Problem**: Frequent system calls between the enclave and the OS can result in performance inefficiencies.

Background

- **Exit-less Approach**: Researchers proposed spawning a responder thread to handle system calls, avoiding enclave exits.
- **Drawbacks of Exit-less Approach**:
  - **Core Workload**: Responder threads increase the workload on available cores, which can degrade performance if there are not enough cores to support multiple enclaves and responder threads.
  - **Polling Overhead**: Responder threads incur overhead while waiting for system call requests, which can further impact performance.

Hypothesis

"As enclaves are statically coupled with SSEs, the execution time of workloads is expected to increase significantly in an overloaded system".

Overloaded system: When no more responder threads can be launched due to fix number of responder cores (aka SSEs)

Related Work

- **Security Service Engine (SSE) [2]**: Proposes using a dedicated lightweight core, called SSE, to run responder threads.
- **Design**:
  - Each enclave core is paired with a corresponding *responder core* to handle system calls efficiently.
  - Enclave core and SSEs are *statically* coupled.

Project Progress

- Investigated SSE architecture and Redis for system call optimization.
- Implemented a 4-million entry Redis cache to reduce system call overhead.
- Measured system calls in a virtual machine using 'strace'.
- Developed and analyzed Hashtable and inter-process communication (IPC) C programs.
- Presented research findings at URCAS.

Difficulties Faced Throughout the Study

- Failed to install MIT Graphite which was the simulator of first choice.
- Could not get source-code for SSE simulation from the authors of SSE [2].
- gem5 is a huge code base which has a steep learning curve.

References

[1] Orenbach M, Lifshits P, Minkin M, Silberstein M. Eleos: ExitLess OS services for SGX enclaves. InProceedings of the Twelfth European Conference on Computer Systems 2017 Apr 23 (pp. 238-253).

[2] Nye J, Ali U, Khan O. SSE: Security Service Engines to Scale Enclave Parallelism for System Interactive Applications. In2024 International Symposium on Secure and Private Execution Environment Design (SEED) 2024 May 16 (pp. 84-95). IEEE.

[3] Binkert, Nathan, et al. "The gem5 simulator." ACM SIGARCH computer architecture news 39.2 (2011): 1-7.

[4] Miller JE, Kasture H, Kurian G, Gruenwald C, Beckmann N, Celio C, Eastep J, Agarwal A. Graphite: A distributed parallel simulator for multicores. InHPCA-16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture 2010 Jan 9 (pp. 1-12). IEEE.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Numbers CNS-2137791, HRD-1834620.
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.