Simulating Trusted Execution Environments in gem5

Ayaz Akram, Venkatesh Akella, Sean Peisert, and Jason Lowe-Power
University of California Davis, and Lawrence Berkeley National Lab

Cycle-level architectural simulation of Trusted Execution Environments (*TEEs*) can enable extensive design space exploration of these secure architectures. The existing architectural simulators do not provide any support for such studies. In this work, we focus on *gem5*[1], and a *RISC-V* based open-source *TEE*, *Keystone*[2], to open new avenues for designing and studying *TEEs*. The architectural simulations are useful to iterate on high-level architectural tradeoffs before focusing on an RTL of the chosen design. With this simulation support, it is easy to pick a design and analyze how it applies to different architectures.

gem5 is one of the most popular computer architecture research simulation frameworks with its multitude of hardware models and rich support for full system simulations. Currently, the *RISC-V* ecosystem provides support to perform functional/RTL level simulation of *RISCV-TEEs* using *QEMU* or *FireSim*[3], however, there is no tool/simulator available to do high-level architectural/microarchitectural studies of *RISC-V TEEs* at a cycle level (for an early design space exploration). Researchers have to rely on analytical modeling for their studies involving *Keystone*.

Keystone is proposed as a customizable and modular *TEE*, which allows fine-grained *TCB* (trusted compute base) configuration. *Keystone* relies on a *RISC-V* feature, *PMP* (physical memory protection), to provide memory protection. Recently *gem5's* full system support has been extended and M-mode (RISC-V's most privileged execution mode) support has been improved which allows (unmodified) *RISC-V* Linux kernel booting on *gem5*. We further extended this support to add *RISC-V PMP* feature in *gem5* which enabled running *Keystone's Security Monitor* (M-mode software which enforces all security guarantees) on *gem5*. *Keystone's SM* is shipped as a part of both *BBL*[4], and *OpenSBI*[5], bootloaders. We tested both of these bootloaders with *SM* on *gem5*. We further set-up all *Keystone* components for simulation on *gem5* and performed different tests.

Evaluation: We rely on the following actions for the functional validation of *Keystone* implementation in *gem5*: 1) We performed physical memory access checks using Linux utilities to test working of *PMP*, which passes successfully. 2) We successfully ran primary *Keystone* tests, which apart from performing some basic functionality tests, check if an enclave access is violated or not. 3) Finally, we tested workloads used in *Keystone's EuroSys* paper which also work successfully.

Apart from functional validation, we are concerned about the performance validation of *Keystone's gem5* implementation as well. Towards this goal, we performed some experiments and collected performance numbers for *Keystone* benchmarks on *gem5* and compared them with the performance numbers published in the original *Keystone's* paper at *EuroSys*. We show that the *Keystone* simulations on gem5 exhibit similar performance numbers/trends as in the *EuroSys* paper. Figure 1 shows a comparison of the slowdown experienced from the trusted execution on two *gem5* CPU models, and the slowdown numbers taken from *Keystone's EuroSys* paper.

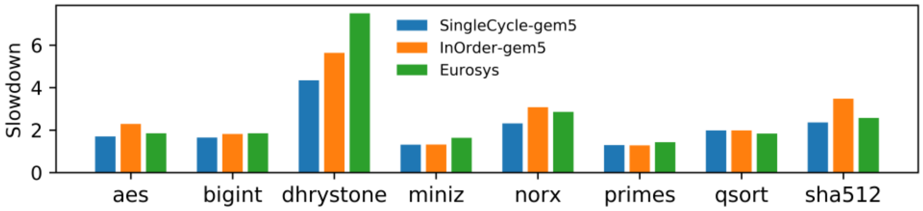

Figure 1. Comparison of *Keystone* slowdown on *gem5* and *EuroSys* paper (native execution). This slowdown includes enclave creation and management time as well.

In future, this work will allow us to do large design space exploration with different microarchitectures (in order, out of order cores, small caches, large caches, one core, many cores). Microarchitecture can impact both performance and security. This work can enable deep introspection into microarchitectural behavior from both performance and security's perspective. Moreover, we can easily study high-level changes to the secure hardware mechanisms (e.g., adding hardware encryption and integrity support).

[1] Lowe-Power et al., "The gem5 simulator: Version 20.0+. arXiv preprint arXiv:2007.03152 (2020).
[2] Lee et al., "Keystone: An open framework for architecting trusted execution environments", in EuroSys 2020, pp. 1—16.
[3] Karandikar et al., "Firesim: Fpga-accelerated cycle-exact scale-out system simulation in the public cloud.," in ISCA, 2018, pp. 29—42.
[4] https://github.com/riscv/riscv-pk
[5] https://github.com/riscv/opensbi