Design of Secure Silo with SGX

Masahide Fukuyama
Keio University
t20703mf@sfc.keio.ac.jp

Masahiro Tanaka
Keio University
masa16.tanaka@keio.jp

Hideyuki Kawashima
Keio University
river@sfc.keio.ac.jp

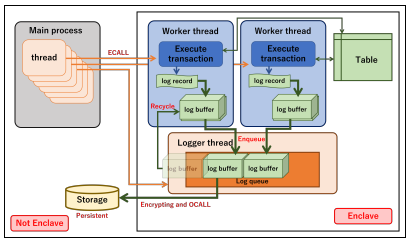

Figure 1. Overview of eSilo

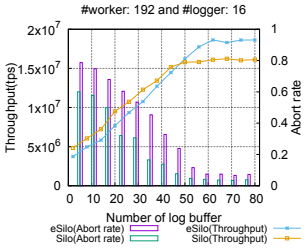

Figure 2. Varying # log buffers

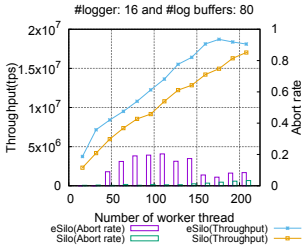

Figure 3. Varying # worker threads

We propose **eSilo**, an extension of the Silo [6] that is the basis of modern high-performance transaction processing systems with a security property using SGX. Vanilla Silo is an in-memory database system and it stores all the records in memory, thus it cannot protect against attacks from adversaries. Our eSilo places all of the records including sensitive data on an enclave, and thus performs all the transaction processing inside the enclave. Thus eSilo provides security property for database users.

eSilo consists of worker threads and logger threads. The worker thread executes transactions and saves generated log records to a log buffer. The logger thread receives the log buffer and writes its contents to the storage for persistence. A worker thread rotates multiple log buffers to avoid blocking. We used SGX SDK [5] and *tlibc/tlibcxx*, which are compatible with the standard libraries to apply SGX specification (i.e., the library and system call limitation).

We implemented eSilo system and evaluated its performance using a many-core machine with 224 cores. We also implemented a vanilla Silo system without enclaves. These implementations are publicly available on GitHub [1, 3]. The evaluation machine has four Intel Xeon Platinum 8176 2.10 GHz, 512GB DRAM, Micron 5200 ECO 480GB SSD, and Ubuntu 20.04.4 LTS. The number of data access operations (i.e., read or write) in a transaction is 10, the workload is similar to YCSB-A. The ratio of read and write operations is 50% and 50% respectively. The number of records in the database is 1 million. The data access skew is set to 0, which is under a uniform distribution and provides low contentions between transactions. Enclave was executed in the "simulation mode", which is the same way adopted in the EnclaveDB work [4].

Based on these conditions, we measured the effect of the number of log buffers and worker threads on performance. Figure. 2 shows the result of varying log buffers. eSilo peaked at 18.64 Mtps with 62 log buffers and Silo peaked 16.23 Mtps with 68 log buffers. Figure. 3 shows the result of varying worker threads. eSilo peaked at 18.71 Mtps with 176 worker threads. Silo peaked at 17.03 Mtps with 208 worker threads. eSilo exceeds Silo's throughput because of the difference in library performance. We also measured the search performance using an index and an iterator for each library. tlibcxx was found to be about three times faster than STL library for searches using iterators. The experimental program is publicly available on GitHub. [2]

In this study, we designed and implemented a secure transaction processing system based on the enclave and Silo. We observed a maximum of 18.71 million tps on 176 worker threads and 16 logger threads. Silo observed a maximum of 17.03 million tps on 208 worker threads and 16 logger threads, which is 9% lower compared to that of eSilo.

There are two future works for this study. The first is the detection of log tampering. This is provided in EnclaveDB, while the method for this function with parallel logging is not yet studied. The second is coping with the faults of TSC. Since SGX does not support time systems like TSC that depends on the CPU clock. Thus, Silo does not work properly if an adversary controls the CPU clock, which should be avoided.

References

[1] Masahide Fukuyama. 2022. Code of eSilo. https://github.com/Noxy3301/enclaveSilo
[2] Masahide Fukuyama. 2022. Code of tlibcxx vector experiment. https://github.com/Noxy3301/tlibcxx_vector_test
[3] Masahide Fukuyama. 2022. Code of vanilla Silo. https://github.com/Noxy3301/silo_minimum
[4] Christian Priebe, Kapil Vaswani, and Manuel Costa. 2018. EnclaveDB: A secure database using SGX. In *IEEE S&P*. 264–278.
[5] SGXSDK 2022. https://download.01.org/intel-sgx/latest/linux-latest/distro/ubuntu20.04-server/sgx_linux_x64_sdk_2.17.100.3.bin
[6] Stephen Tu, Wenting Zheng, Eddie Kohler, Barbara Liskov, and Samuel Madden. 2013. Speedy transactions in multicore in-memory databases. In *SOSP*. 18–32.