Secure Compute-VM: Secure Big Data Processing with SGX and Compute Accelerators

Seehwan Yoo
Dankook University
seehwan.yoo@dankook.ac.kr

Hyunik Kim
Dankook University
eternity13@dankook.ac.kr

Joongheon Kim
Chung-Ang University
joongheon@cau.ac.kr

ABSTRACT

This paper considers secure big data processing. With the architectural support, big data processing with compute accelerators can efficiently preserve privacy concern.

CCS CONCEPTS

• **Security and privacy → Virtualization and security**;

KEYWORDS

SGX secure enclave; deep learning

ACM Reference Format:
Seehwan Yoo, Hyunik Kim, and Joongheon Kim. 2018. Secure Compute-VM: Secure Big Data Processing with SGX and Compute Accelerators. In *3rd Workshop on System Software for Trusted Execution (SysTEX '18), October 15, 2018, Toronto, ON, Canada.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3268935.3268945

1 INTRODUCTION

. Big-data, deep learning, blockchain are emerging applications on contemporary computer systems, which make a system more compute-intensive than ever before. To meet the high demands of computing power, additional compute accelerators such as DSP, GPU, and manycore devices have been actively utilized. For example, NVidia's GTX 1080Ti GPU includes more than a thousand (1,152) processing cores, which makes it popular to mining new bitcoins.

One of flexible and feasible ways to utilize GPU for versatile applications is to make virtual cores instead of physical cores. Recent GPU virtualization [8, 13] allows virtual machines to use GPUs, preserving logical separation among different workloads.

Some of big data and machine learning applications deal with user privacy, in order to analyze user shopping patterns [5], personal geological locations [9], or genome information [14]. Thus, trusted execution environment, TEE, are considered for securing data in big-data platforms. This paper focuses on the utilization of SGX enclave and compute accelerator for securing big-data processing.

2 CHALLENGES AND PROBLEMS

Intel SGX [4] extends Intel64 ISA so that the processor embeds the safe-guard application areas. For the SGX-enabled DRAM region, or SGX enclave, all memory contents are encrypted by the key inside CPU, and it never expose plain-text to the outside of the SGX enclave. In addition, for every memory read/write operation on the SGX enclave region is monitored by the CPU, so that any external memory access cannot break the integrity of the SGX enclave. There are some study to put privacy-sensitve users' big data into the SGX enclave [7, 11, 12], but there are some problems when the enclave usage comes with the GPU processing.

Unfortunately, modern GPUs do not support SGX, thus it cannot properly decrypt the SGX enclave's memory contents. In addition, the SGX uses a specific DRAM region within main memory, called Processor Reserved Memory (PRM) at which external peripheral device, such as GPU, cannot access. In fact, GPU device usually has several gygabytes of internal DRAM, and uses DMA to move data from/to main memory. Note that SGX does not allow DMA into an SGX enclave region. Namely, privacy-sensitive user data should be handled inside SGX enclave; however, it GPU cannot directly access it. Therefore, it is generally difficult to secure data processing with GPU.

3 PROPOSED APPROACH

Most big data processing consists of several stages for handling data. In particular, deep learning framework consists of multiple 'inference layers', to extract meaningful information from raw data. We observe that the raw data could bear some privacy-sensitive information; However, the data in the intermediate processing stages are much less privacy-sensitive than the initial raw data because each layer conducts irreversible complex matrix calculation to find unknown relationships among the given inputs.

To securely handle the big data, specifically with deep neural networks, we propose a secure compute-VM. First of all, the secure compute-VM initiates the privacy-preserving stage. In the stage, CPU begins the early learning stage with raw data located inside the SGX enclave, which can efficiently hide privacy-related data from external environments. Once CPU begins the learning stage, the raw data flows into the first inference layer. CPU stops processing when the correlation between the original raw data from the intermediate one is small enough (i.e., we cannot infer the original data). Then, the secure compute-VM exports data to the non-enclave region.

Secondly, the secure compute-VM performs the enlightening stage. In this stage, the secure compute-VM delegates middle layers processing to the compute accelerators (e.g., GPU or Xeon-Phi). Here, the secure compute-VM can leverage numerous cores, parallelize complex data processing, and never lose privacy control from the secure enclave.

SysTEX '18, October 15, 2018, Toronto, ON, Canada

ACM ISBN 978-1-4503-5998-6/18/10…$15.00
https://doi.org/10.1145/3268935.3268945

4 RELATED WORK

Cloud computing with secure enclave has been addressed in several research results, such as Haven [3], graphine-sgx [15] and SCONE [2]. Haven was one of first approaches to run fully functional OS (Windows) with SGX enclave. Graphine-sgx pursues the adoption of unmodified Linux applications, and also fully functional multi-process Linux within SGX enclave. SCONE cleverly combines docker container with SGX so that the service distribution and update could be made in an easy and secure way. Each user can create secure enclave as a docker container, which guarantees the confidentiality of user data. To minimize the enclave enter/exit overheads, SCONE proposes an asynchronous event queue for handling Linux system calls. Some other approach [7] presents SGX-enabled learning framework, but it does not support any GPU nor manycore devices, thus the learning efficiency could be much improved when we use them.

5 THREAT MODELS AND DEFENSES

Our threat model considers the confidentiality attack for the privacy-sensitive raw input data. SGX assumes untrusted privileged software such as OS, and hypervisor, therefore, an attacker is assumed to run code and is able to access main memory, outside the SGX enclave.

Although SGX enclave has some known side channel at the micro-architecture and OS level [6, 10, 16], we assume SGX enclave is securely protected by hardware, and there is no side channel because the enclave is involved only at the initial stage (preprocessing) among the multiple layered computation stages. The raw data is temporarily available on the cache of the running cores. However, the attacker is difficult to mount cache-timing attack on the deep learning data because vector size could be various, and the size is usually much smaller than the cacheline size, therefore it is very hard to exactly identify the cached item.

Note that an attacker can create a new VM, however we assume a GPU device is separately allocated to a virtual machine. In addition, memory access within the secure compute-VM is restricted within the internal DRAM of GPU device. Therefore, another virtual GPU devices cannot mount cross-VM attacks.

6 PROOF OF CONCEPTS

To separate the distinct processing stages, we leverages tensorflow framework [1]. The tensorflow framework supports device placement, which places code/data in specific compute device. So far, we manually identify the proper separating point between the two stages, and place stages on CPU and GPU, respectively. Then, we put the CPU processing inside the SGX enclave using SCONE container. Because we set the second stage to be placed on the GPU, the tensorflow framework will gather data from the containerized job, and exports it to the other memory region. In the compute-VM, the GPU executes the learning model and then completes the rest of the learning process.

In our preliminary experiments with MNIST test benchmark which is widely used in the literature, the execution time is presented in Figure 1. In the figure, the execution times of CPU only, GPU only, and our secure compute-VM are presented. Our secure compute-VM has longer execution time than the time of GPU only case, but it is shorter than the execution time of CPU only case. That is due to the acceleration from GPU. During the execution, the utilization of the GPU device is about 40% in GPU-only case. Our secure compute-VM consumes 15% of GPU utilization, so it can reduce the execution time as much as the GPU execution proportion. Notice in the Figure, time_user has been decreased almost by half.

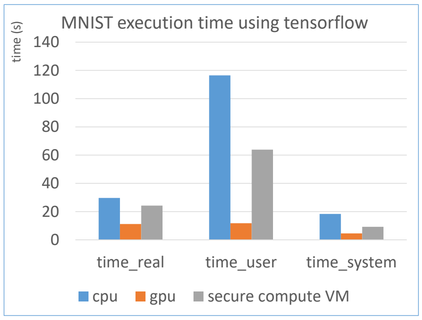

Figure 1: MNIST execution time comparison

7 CONCLUSION AND FUTURE WORK

Big data processing with machine learning has privacy concerns. To secure big data processing, we present a secure compute-VM, which leverages SGX enclave as well as compute-accelerator, GPU. Its GPU processing enables efficient deep neural network with multiple inference layers, and its SGX enclave provides a trusted execution environment that preserves the privacy of user data.

In this work, we focus on GPU as compute accelarator, but Xeon-Phi and FPGA could also be considered. In addition, more deep-dive performance evaluation with multiple benchmarks is left as future work.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-scale Machine Learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16)*. USENIX Association, Berkeley, CA, USA, 265–283. http://dl.acm.org/citation.cfm?id=3026877.3026899

[2] Sergei Arnautov, Bohdan Trach, Franz Gregor, Thomas Knauth, Andre Martin, Christian Priebe, Joshua Lind, Divya Muthukumaran, Dan O'Keeffe, Mark L. Stillwell, David Goltzsche, David Eyers, Rüdiger Kapitza, Peter Pietzuch, and Christof Fetzer. 2016. SCONE: Secure Linux Containers with Intel SGX. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16)*. USENIX Association, Berkeley, CA, USA, 689–703. http://dl.acm.org/citation.cfm?id=3026877.3026930

[3] Andrew Baumann, Marcus Peinado, and Galen Hunt. 2015. Shielding Applications from an Untrusted Cloud with Haven. *ACM Trans. Comput. Syst.* 33, 3, Article 8 (Aug. 2015), 26 pages. https://doi.org/10.1145/2799647

[4] Victor Costan and Srinivas Devadas. 2016. Intel SGX Explained. Cryptology ePrint Archive, Report 2016/086. https://eprint.iacr.org/2016/086.

[5] Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. 2015. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science* 347, 6221 (2015), 536–539. https://doi.org/10.1126/science.1256297 arXiv:http://science.sciencemag.org/content/347/6221/536.full.pdf

[6] Johannes Götzfried, Moritz Eckert, Sebastian Schinzel, and Tilo Müller. 2017. Cache Attacks on Intel SGX. In *Proceedings of the 10th European Workshop on Systems Security (EuroSec'17)*. ACM, New York, NY, USA, Article 2, 6 pages. https://doi.org/10.1145/3065913.3065915

[7] Tyler Hunt, Congzheng Song, Reza Shokri, Vitaly Shmatikov, and Emmett Witchel. 2018. Chiron: Privacy-preserving Machine Learning as a Service. *CoRR* abs/1803.05961 (2018). arXiv:1803.05961 http://arxiv.org/abs/1803.05961

[8] Shinpei Kato, Michael McThrow, Carlos Maltzahn, and Scott Brandt. 2012. Gdev: First-class GPU Resource Management in the Operating System. In *Proceedings of the 2012 USENIX Conference on Annual Technical Conference (USENIX ATC'12)*.

USENIX Association, Berkeley, CA, USA, 37–37. http://dl.acm.org/citation.cfm?id=2342821.2342858

[9] Rob Kitchin. 2013. Big data and human geography: Opportunities, challenges and risks. *Dialogues in Human Geography* 3, 3 (2013), 262–267. https://doi.org/10.1177/2043820613513388 arXiv:https://doi.org/10.1177/2043820613513388

[10] Ahmad Moghimi, Gorka Irazoqui, and Thomas Eisenbarth. 2017. CacheZoom: How SGX Amplifies The Power of Cache Attacks. *CoRR* abs/1703.06986 (2017). arXiv:1703.06986 http://arxiv.org/abs/1703.06986

[11] C. Priebe, K. Vaswani, and M. Costa. 2018. EnclaveDB: A Secure Database using SGX. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 405–419. https://doi.org/10.1109/SP.2018.00025

[12] Felix Schuster, Manuel Costa, Cédric Fournet, Christos Gkantsidis, Marcus Peinado, Gloria Mainar-Ruiz, and Mark Russinovich. 2015. VC3: Trustworthy Data Analytics in the Cloud Using SGX. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP '15)*. IEEE Computer Society, Washington, DC, USA, 38–54. https://doi.org/10.1109/SP.2015.10

[13] Lin Shi, Hao Chen, Jianhua Sun, and Kenli Li. 2012. vCUDA: GPU-Accelerated High-Performance Computing in Virtual Machines. *IEEE Trans. Comput.* 61, 6 (June 2012), 804–816. https://doi.org/10.1109/TC.2011.112

[14] Zachary D. Stephens, Skylar Y. Lee, Faraz Faghri, Roy H. Campbell, Chengxiang Zhai, Miles J. Efron, Ravishankar Iyer, Michael C. Schatz, Saurabh Sinha, and Gene E. Robinson. 2015. Big Data: Astronomical or Genomical? *PLOS Biology* 13, 7 (07 2015), 1–11. https://doi.org/10.1371/journal.pbio.1002195

[15] Chia-Che Tsai, Donald E. Porter, and Mona Vij. 2017. Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX. In *Proceedings of the 2017 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC '17)*. USENIX Association, Berkeley, CA, USA, 645–658. http://dl.acm.org/citation.cfm?id=3154690.3154752

[16] Jo Van Bulck, Frank Piessens, and Raoul Strackx. 2017. SGX-Step: A Practical Attack Framework for Precise Enclave Execution Control. In *Proceedings of the 2Nd Workshop on System Software for Trusted Execution (SysTEX'17)*. ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3152701.3152706