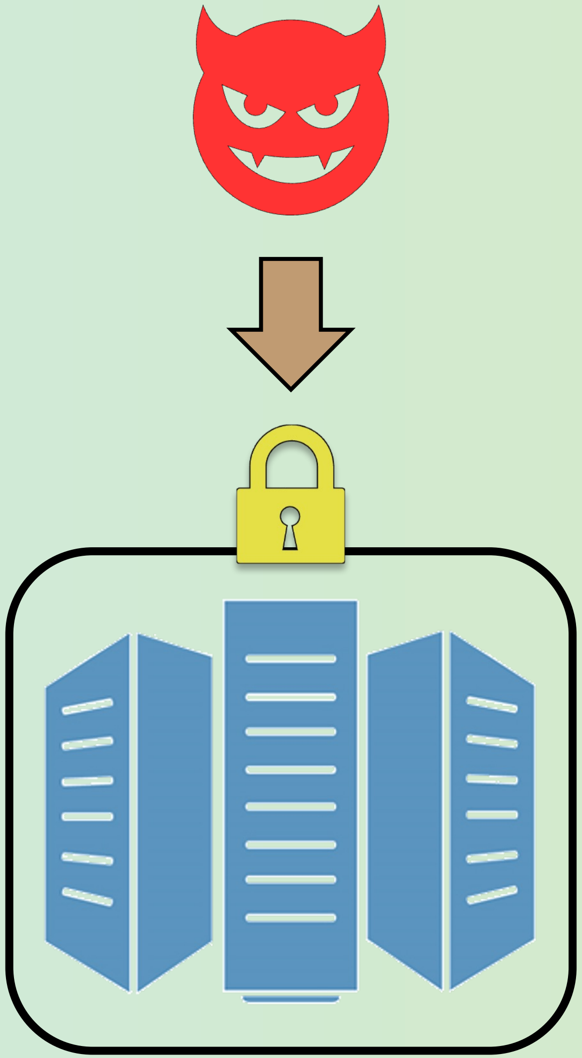
Analyzing the Performance Impact of HPC Workloads with Gramine+SGX on 3rd Generation Xeon Scalable Processors

Shinobu Miwa (The University of Electro-Communications, Japan)

Shin'ichiro Matsuo (Georgetown University, USA)

Confidential Computing

- Data analytic workloads are becoming one of the main applications running on modern supercomputing systems
  - E.g.) Deep learning, graph analysis, etc.
- Some data analysts often need to deal with very sensitive data provided by third parties as **confidential**
- Today's supercomputers are unsuitable for confidential computing
  - Privileged users can access any user data
- For sensitive data analysis, supercomputers need to protect user data even from privileged adversaries



Intel SGX (Software Guard Extensions)

- Strongly isolated execution environment for multi-user systems
- Enable a user application to be executed as a secure container (called *enclave*) having its own memory region (called *enclave memory*)
- Support hardware-assisted data encryption and access control
- Can protect user data from various adversary including malicious operating systems and hypervisors
- Some cloud systems already support confidential computing with SGX

SGX in HPC

- At present, there is no production system that provides a computing service with SGX

- SGX had a few problems before

☹ **Limited size of enclave memory**

  - The upper limit was only 128MB in the Coffee Lake or previous architecture
  - SGX showed substantial performance degradation for many HPC workloads due to the frequent memory swapping

☹ **Poor programmability**

  - Users need code modification to exploit SGX, but it is impractical for some HPC workloads relying on third-party software
  - Some frameworks that enable the execution of unmodified applications on SGX had been developed, but they were immature

Innovation of SGX Architecture and Frameworks

- **3rd generation Xeon scalable processors** (from 2021)
  - Great increase in the size of enclave memory (up to 1TB)
  - Improvement of memory access latency by eliminating a Merkle tree

©Intel

- **Gramine** (from 2021)
  - First production-ready version of an SGX library OS
  - Enable the execution of various applications on SGX without code modification

GRAMINE

©Gramine

Research Objectives and Contributions

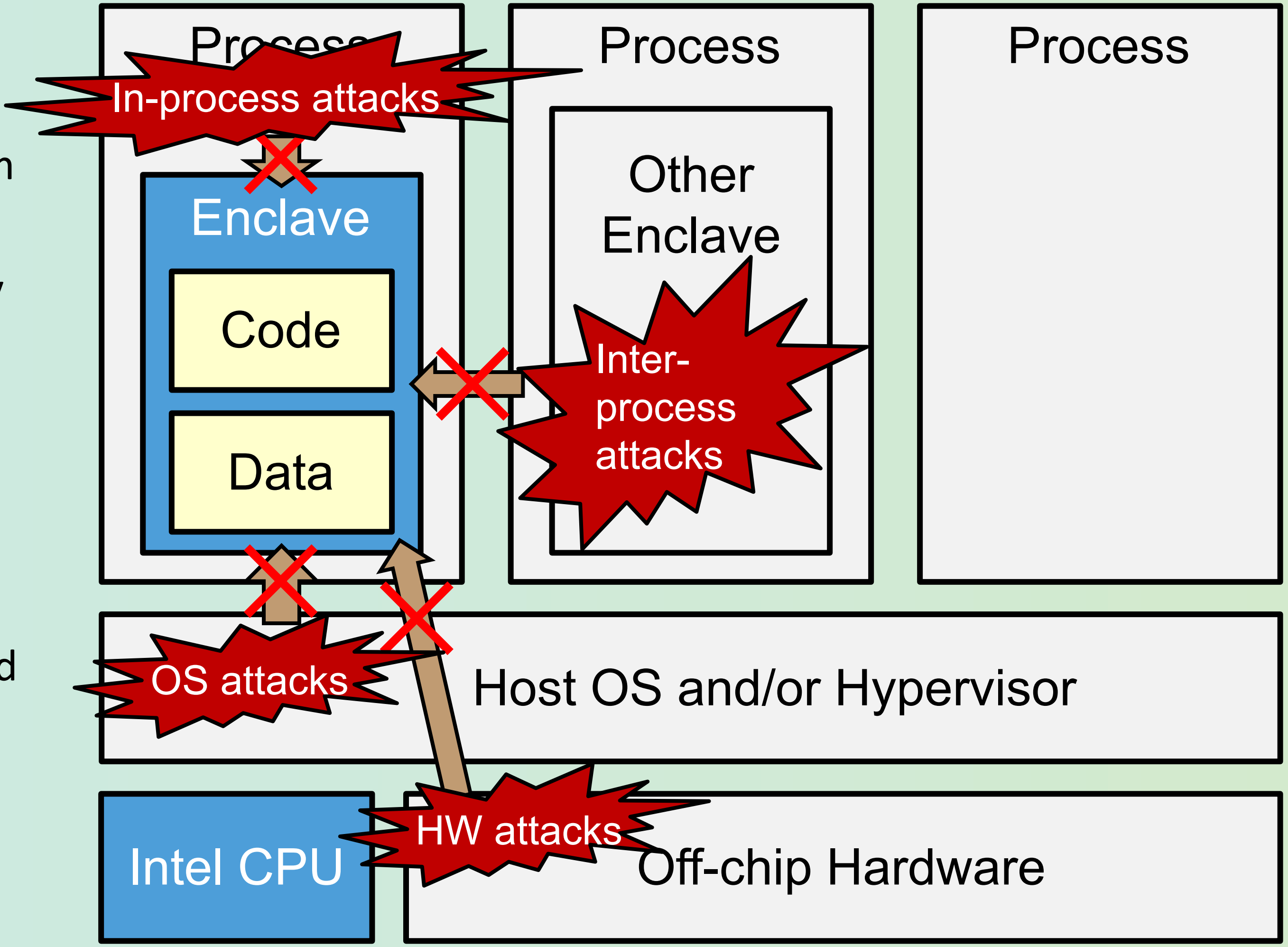
- Our goal is to uncover the performance impact of HPC workloads with Gramine+SGX on 3rd gen Xeon CPUs
  - Unlike many previous studies (e.g., [1,2]), this work provides the first performance analysis on the combination of Gramine and a 3rd gen Xeon processor

- Main findings
  - The impact of Gramine+SGX on both core performance and memory bandwidth is small (a slowdown of up to **1.1x**)
  - The impact of Gramine+SGX on memory latency is a bit large (a slowdown of up to **2.7x**)
  - Gramine+SGX greatly improves the performance of HPC workloads (slowdowns of **1.5x** on average and up to **4.4x**)

[1] A . Akam, et al.: Performance Analysis of Scientific Computing Workloads on General Purpose TEEs , IPDPS, 2021
[2] M. El-Hindi, et al.: Benchmarking the Second Generation of Intel SGX Hardware, DaMoN, 2022

Overview of SGX

- Enclave
  - Secure computing environment isolated from normal execution environment
  - Store code and data into an enclave memory with encryption
  - Enclave memory access from the other programs (including OSes and hypervisors) is prohibited by memory controllers
- Restrictions on SGX applications
  - The components to be executed on SGX need to be written at the application code level explicitly
  - Many libraries that rely on system calls (e.g., *glibc*) can not be executed within an enclave



[SGX's threat model]

Gramine (called Graphene-SGX previously)

- Lightweight library OS that enables the execution of unmodified applications on SGX
  - Offer a minimum set of system calls
  - Delegate unsupported system calls to the host OS
- Can pull any libraries into an enclave with the integrity check
- All gramine users need to do is to write and sign *manifests*

```
[loader]
entrypoint = "file:/usr/lib/x86_64-linux-gnu/gramine/libsysdb.so"

[libos]
entrypoint = "/cblas-dgemm“

[fs]
mounts = [
  { path = "/lib",
    uri = "file:/usr/lib/x86_64-linux-gnu/gramine/runtime/glibc" },
  { path = "/lib/x86_64-linux-gnu",
    uri = "file:/lib/x86_64-linux-gnu" },
  { path = "/lib64", uri = "file:/lib64" },
  { path = "/cblas-dgemm", uri = "file:cblas-dgemm" },
]

[sgx]
trusted_files = [
  { uri = "file:/usr/lib/x86_64-linux-gnu/gramine/libsysdb.so" },
  { uri = "file:cblas-dgemm" },
  { uri = "file:/usr/lib/x86_64-linux-gnu/gramine/runtime/glibc/" },
  { uri = "file:/lib/x86_64-linux-gnu/" },
  { uri = "file:/lib64/" },
]
```

[An example of a manifest]

Experimental System

- A single compute node composed of a 1-socket 3rd gen Xeon CPU
- The performance analysis of Gramine+SGX on multiple nodes will be performed in the future

[System configuration]

Name	Remarks
CPU	1x Xeon Gold 5317 (12C24T, 3.0GHz) L1D: 48KB, L2C: 1.25MB, LLC (shared): 18MB
Memory	64GB DDR4-3200 (32GB enclave memory)
Host OS	Ubuntu-20.04
Library OS	Gramine-1.4

Benchmark Programs

- **Microbenchmarks**
  ➡ assessment of impact on some basic computation patterns
- **HPC benchmarks**
  ➡ assessment of impact on HPC workloads
- Software infrastructure
  - C/C++/Fortran: gcc/g++/gfortran-9.4.0 or icc-2021.9.0
  - PyTorch: Python-3.8 and PyTorch-2.0.1

[Microbenchmarks]

Name	Remarks
GEMM	Matrix multiplication (**double**, single, and bfloat16 precision)
STREAM	STREAM benchmark
LATENCY	Linked-list traversal

[HPC benchmarks]

Groups	Name	Remarks
NPB	**bt, cg, ep, is, sp, ua**	NAS parallel benchmarks
GAPBS	**bc, bfs, cc, cc_sv, pr, pr_spmv**	Graph workloads
Modern-HPC	**Kripke**	3D particle transport
	LULESH	Shock hydrodynamics
	LightGBM	Gradient decision tree
PyTorch	**MNIST**	CNN training for MNIST
	TIME	LSTM training for time sequence
	WORD	Transformer training for Wikitext-2

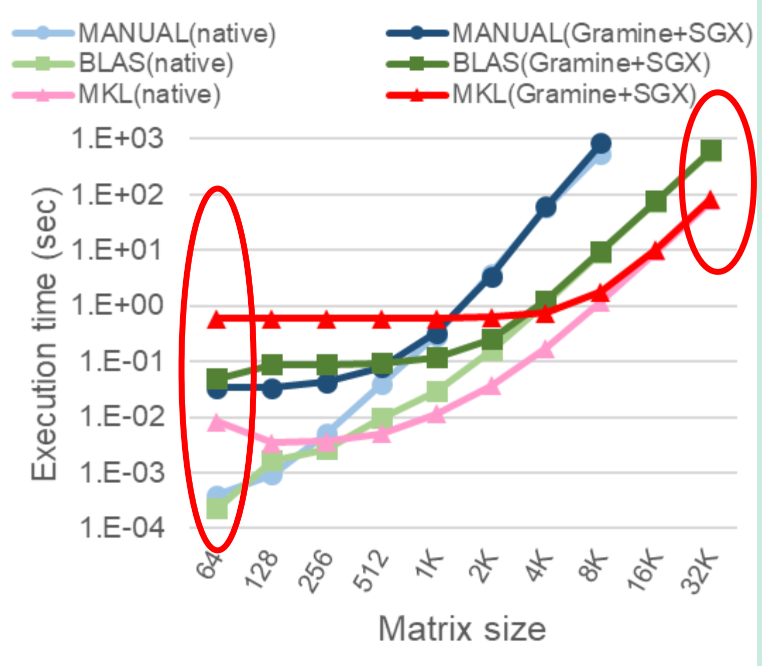
Used in [1] (NPB, GAPBS, Modern-HPC)

[1] A . Akam, et al.: Performance Analysis of Scientific Computing Workloads on General Purpose TEEs , IPDPS, 2021

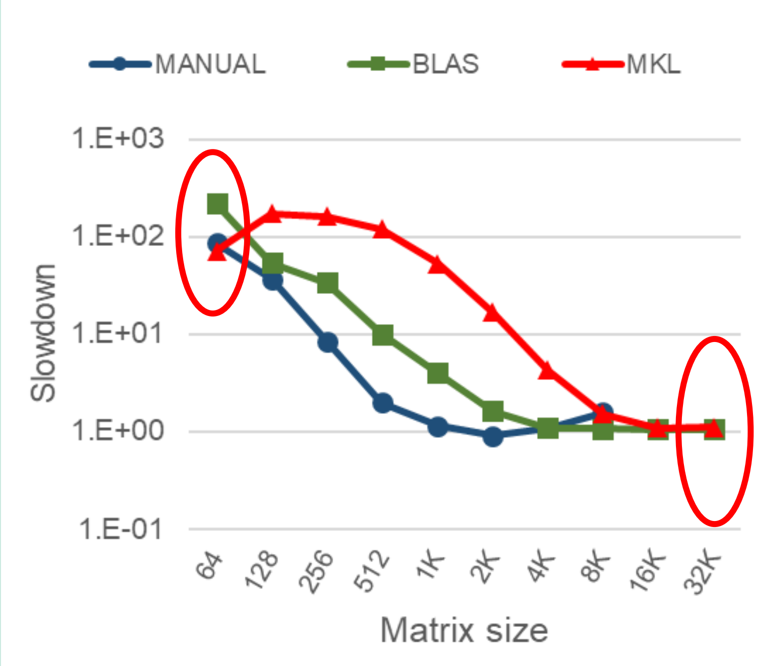
Impact on Arithmetic Operations (DGEMM)

- When matrix size is small, Gramine+SGX shows large slowdown compared to native Linux due to the impact of system call emulation
- The performance gap between Gramine+SGX and native Linux becomes negligible (1.05-1.10x) as the matrix size (i.e., the number of arithmetic operations) increases

➡ Impact of Gramine+SGX on the performance of arithmetic operations is small!



[Execution time]



[Slowdown]

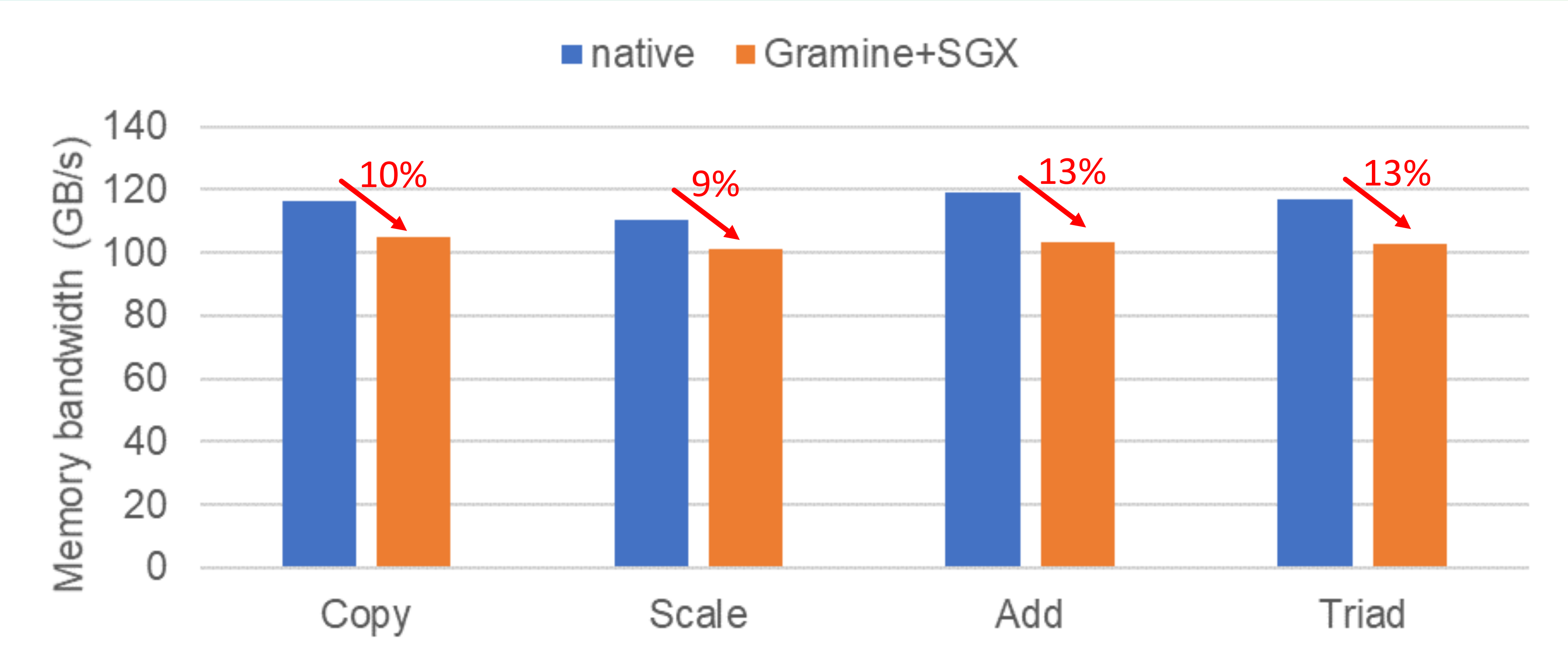
[DGEMM kernels]

Name	Remarks
MANUAL	OpenMP+ auto-vectoriation
BLAS	OpenBLAS-0.3.8
MKL	Intel MKL-2023.1.0

Impact on Memory Bandwidth

- Gramine+SGX shows a reduction of 9-13% in memory bandwidth

➡ Will be acceptable for many HPC users!

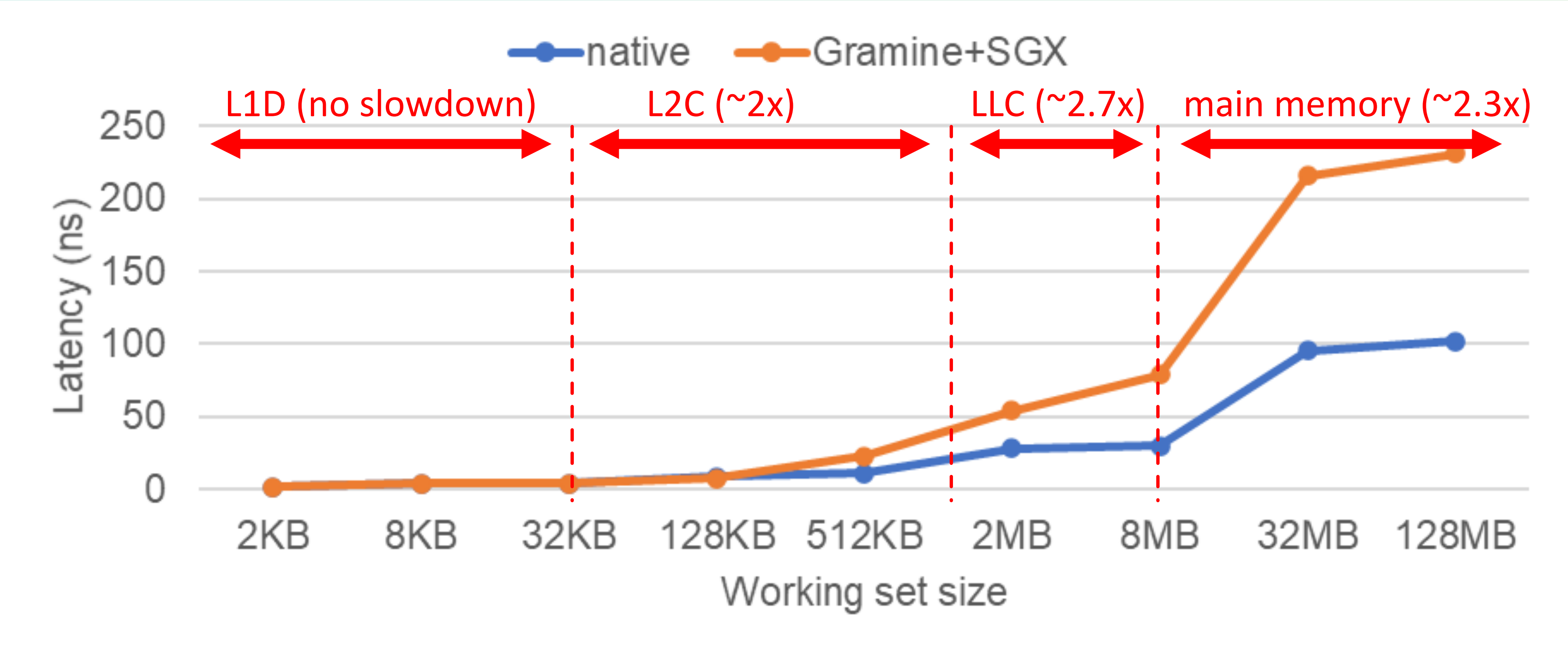


[Performance of STREAM]

Impact on Memory Latency

- Gramine+SGX shows an increase of up to 2.7x in memory latency

➡ Some optimizations (e.g., software pipelining) will be needed!



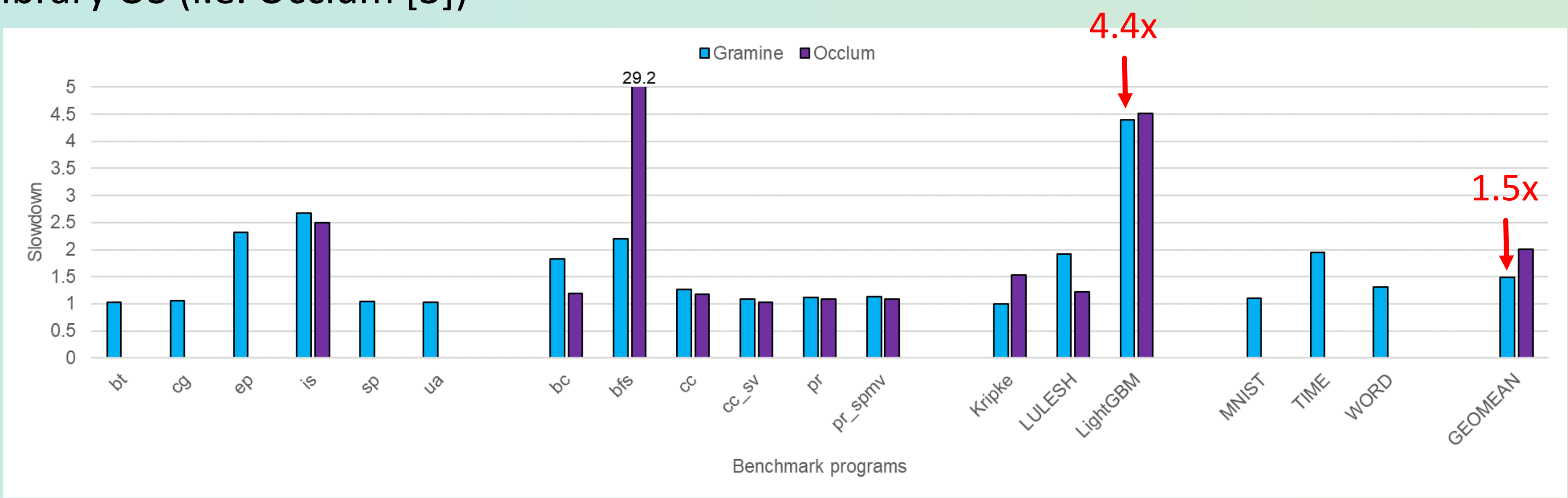
[Performance of LATENCY]

Impact on HPC Benchmarks

- The slowdown caused by Gramine+SGX is 1.5x on average and up to 4.4x

➡ Gramine and 3rd gen Xeon CPU improve the performance overhead of SGX by one or two orders of magnitude compared to the combination of the former generation SGX framework and CPU [1]!

- Gramine covers a wider range of applications and has a more stable performance than the other SGX library OS (i.e. Occlum [3])

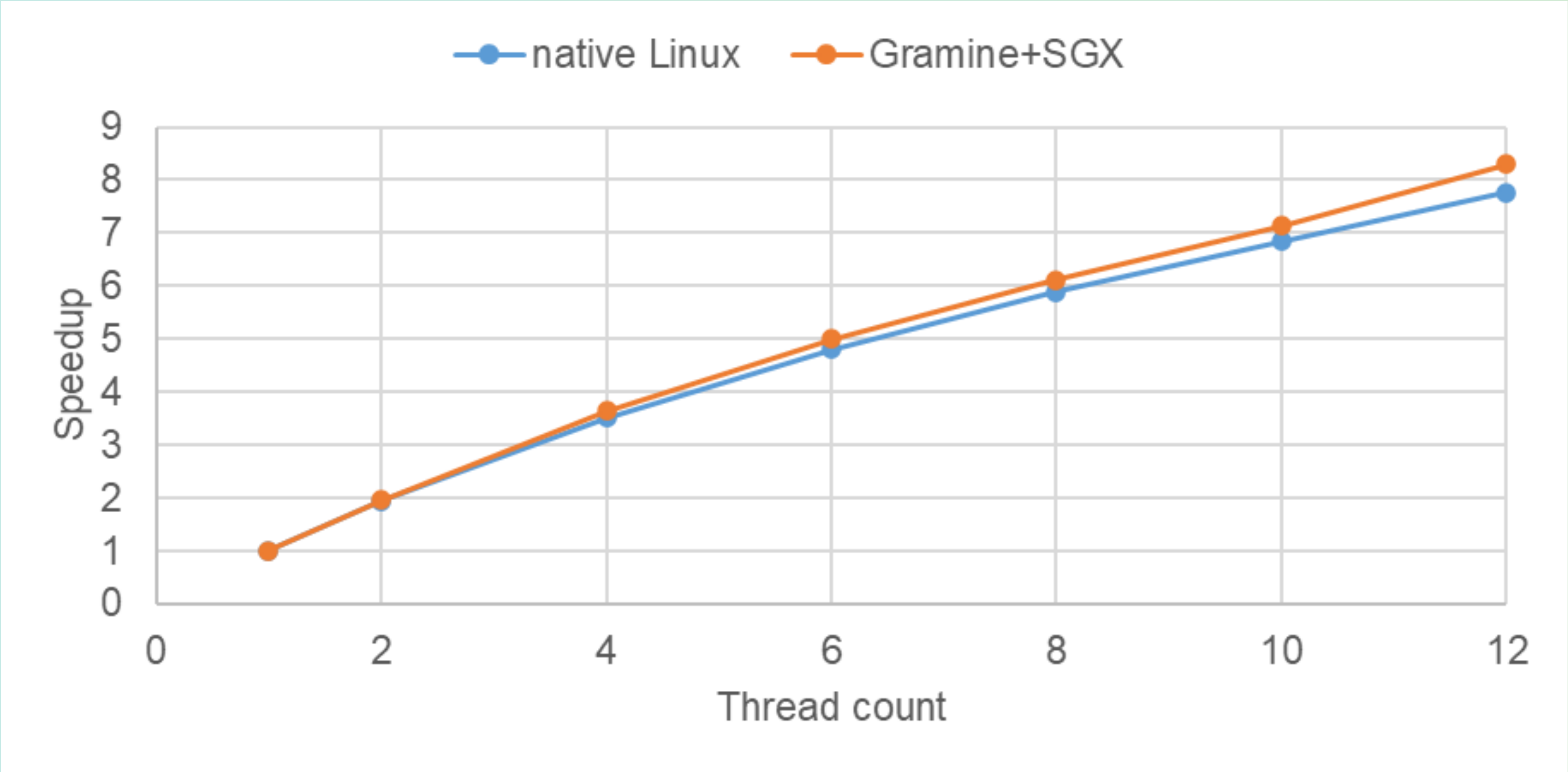


[Performance of HPC benchmarks]

[3] Y. Shen, et al.: Occlum: Secure and Efficient Multitasking Inside a Single Enclave of Intel SGX, ASPLOS, 2020

Scalability

- Gramine+SGX shows almost same scalability as native Linux



[Scalability of *cg*]

Discussion

- The previous work [1] lists three reasons why SGX is inappropriate to HPC
  1. **Limited size of enclave memory**
  2. **Poor scalability**
  3. **Poor programmability**

Items	Previous work [1]	This work	Where does it come from?
Enclave memory size	128MB ➡	32GB	Improvement of SGX architecture
Scalability	1.4x at 6 threads (for cg) ➡	4.8x at 6 threads (for cg)	Improvement of SGX architecture
Programmability	Require code modification for some apps ➡	Do not require code modification for any apps	Improvement of SGX frameworks

SGX is almost ready for use in HPC!

[1] A . Akam, et al.: Performance Analysis of Scientific Computing Workloads on General Purpose TEEs , IPDPS, 2021

Conclusions and Future Work

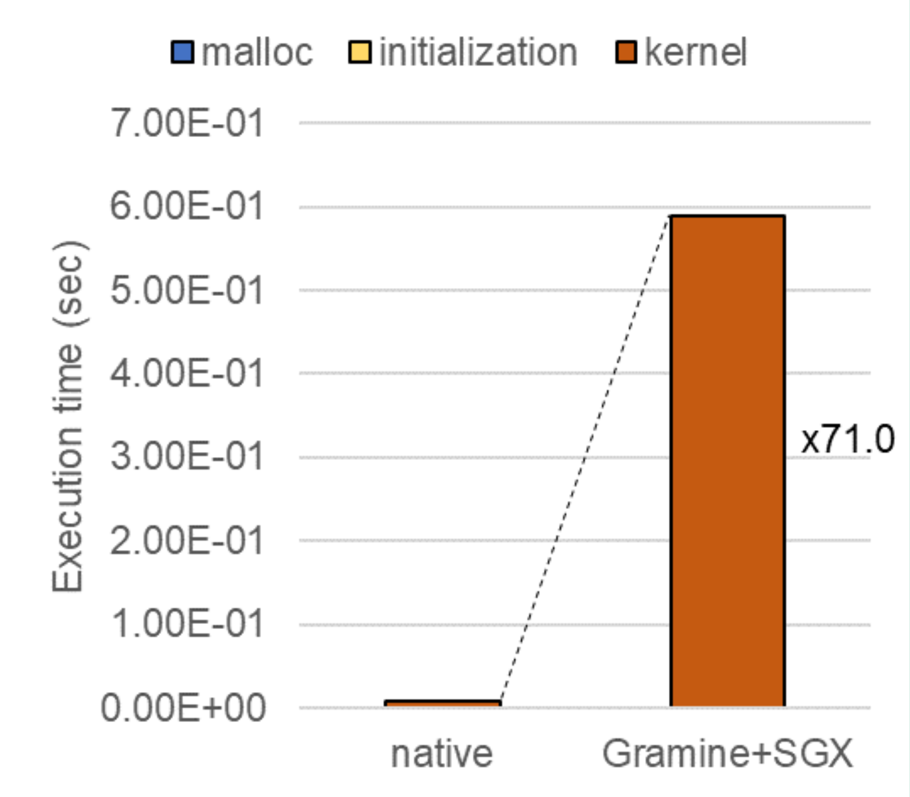
- Conclusions
  - Performed the first performance analysis of HPC workloads with Gramine+SGX on a 3rd gen Xeon CPU
    - A performance overhead of 4-170% (in arithmetic and memory operations)
    - A slowdown of 1.5x on average and up to 4.4x (for HPC workloads)
  - SGX has a bright future in the field of HPC

- Future work
  - Analyze the impact of Gramine+SGX on network performance
  - Develop some techniques that optimize the performance of parallel SGX applications

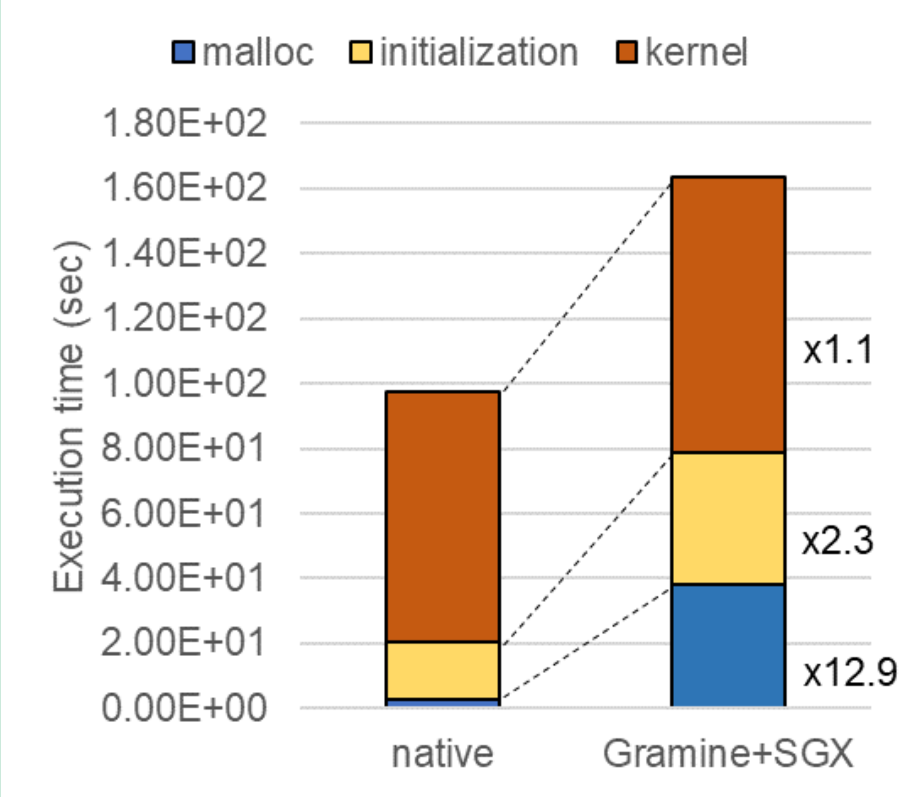
Thank you!

Impact on Overall Performance (DGEMM, MKL)

- Small matrices: the slowdown of the parallel region (*kernel*) is dominant
- Large matrices: the slowdown of the serial regions (*malloc* and *initialization*) is dominant



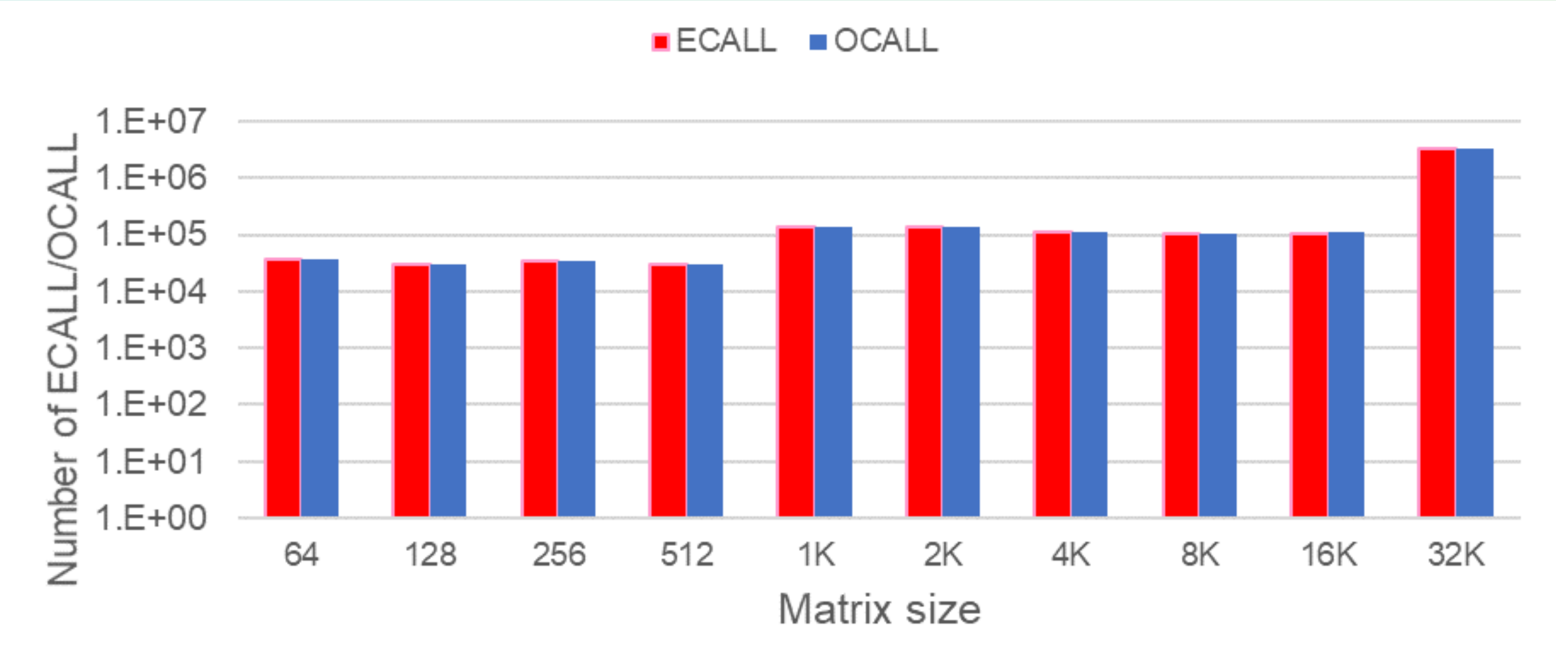
[Matrix size = 64]



[Matrix size = 32K]

Impact of Switching Environment (DGEMM, MKL)

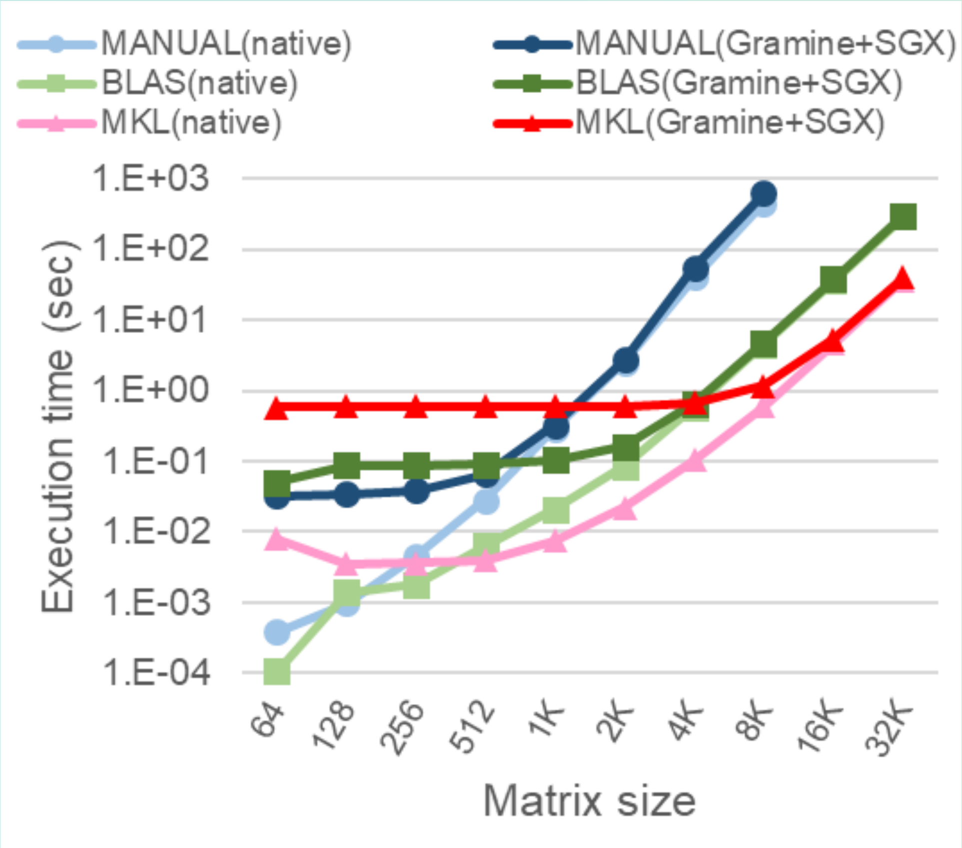
- Switching between SGX and normal execution environment (i.e., the execution of ECALL and OCALL) is known as one of the main sources of the limited performance of SGX
- In contrast to the slowdown, ECALL and OCALL counts increase as the matrix size increases
  ⇨ Little impact on the performance!



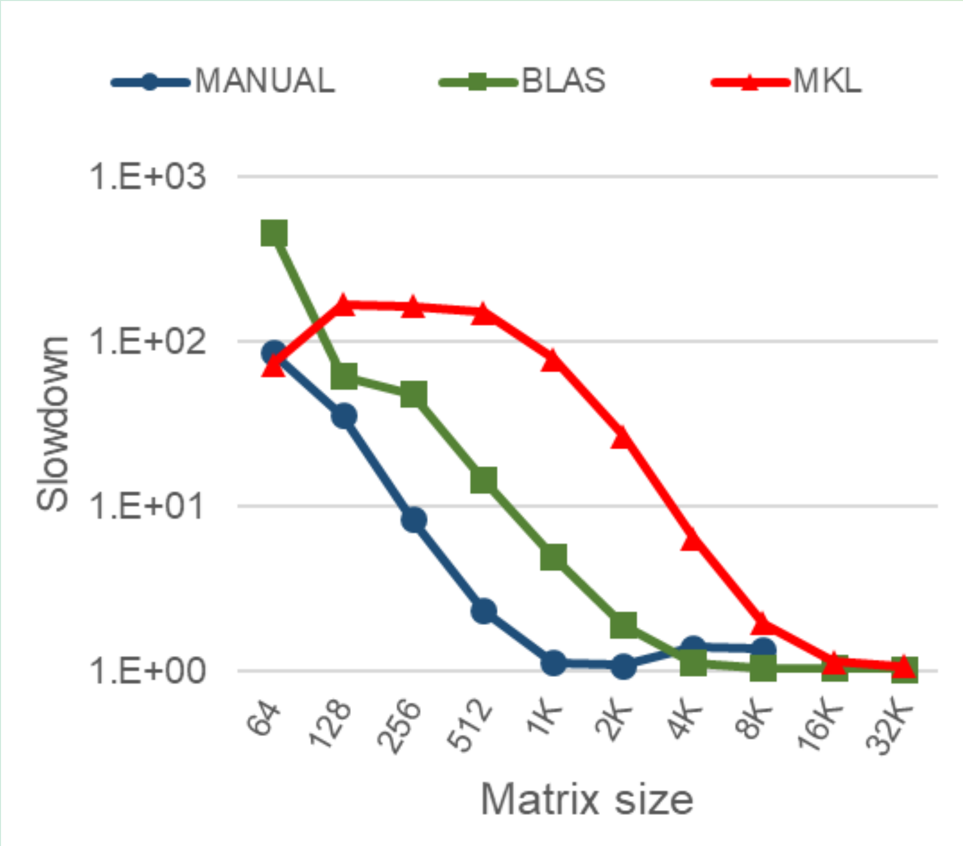
[ECALL and OCALL counts]

Impact on Arithmetic Operations (SGEMM)

- Gramine+SGX shows a slowdown of 1.05-1.08x at a matrix size of 32K



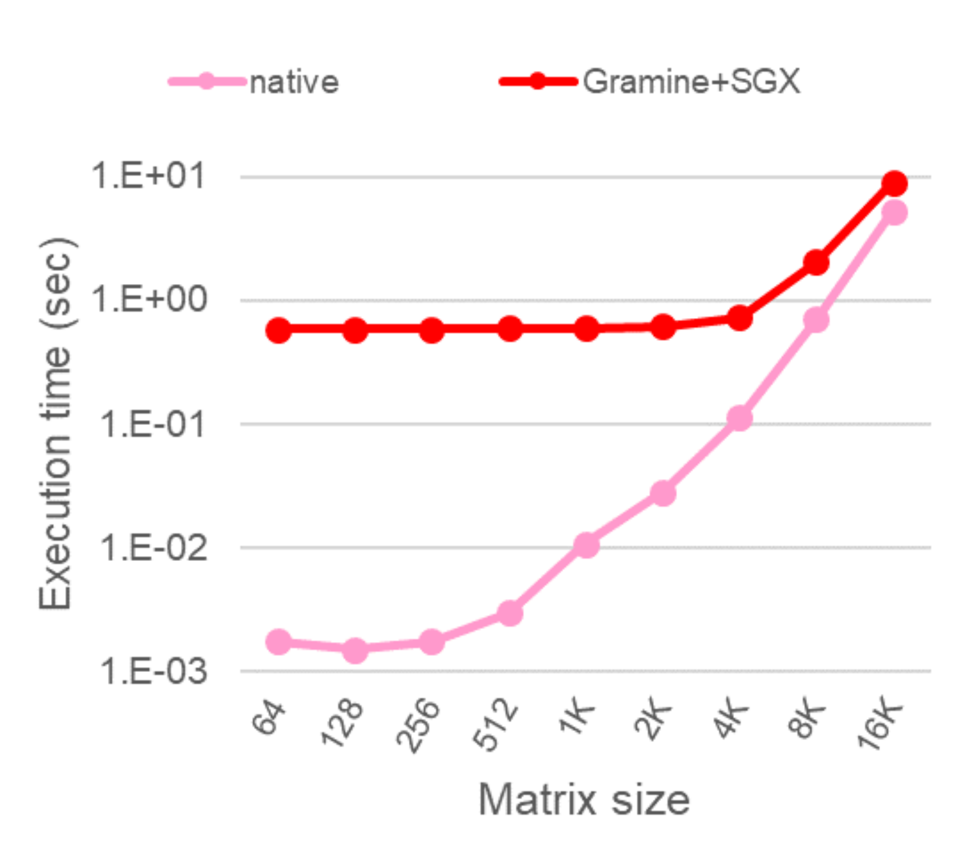
[Execution time]



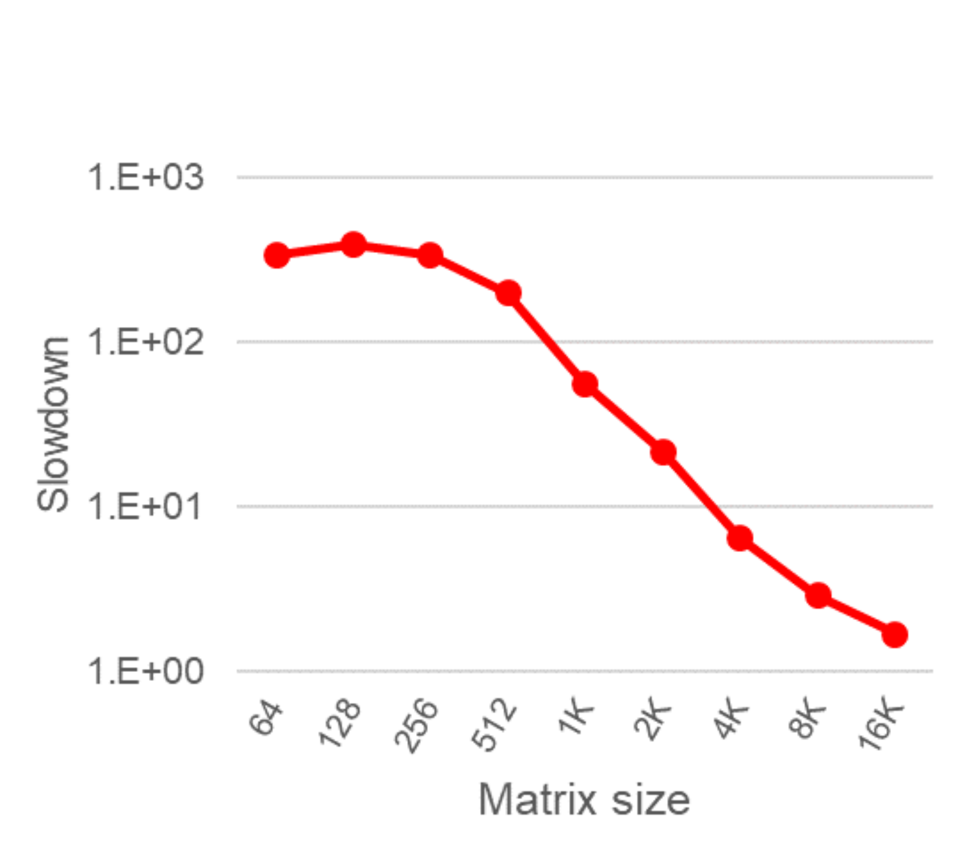
[Slowdown]

Impact on Arithmetic Operations (bfloat16 GEMM)

- Gramine+SGX shows a slowdown of 1.7x at a matrix size of 16K



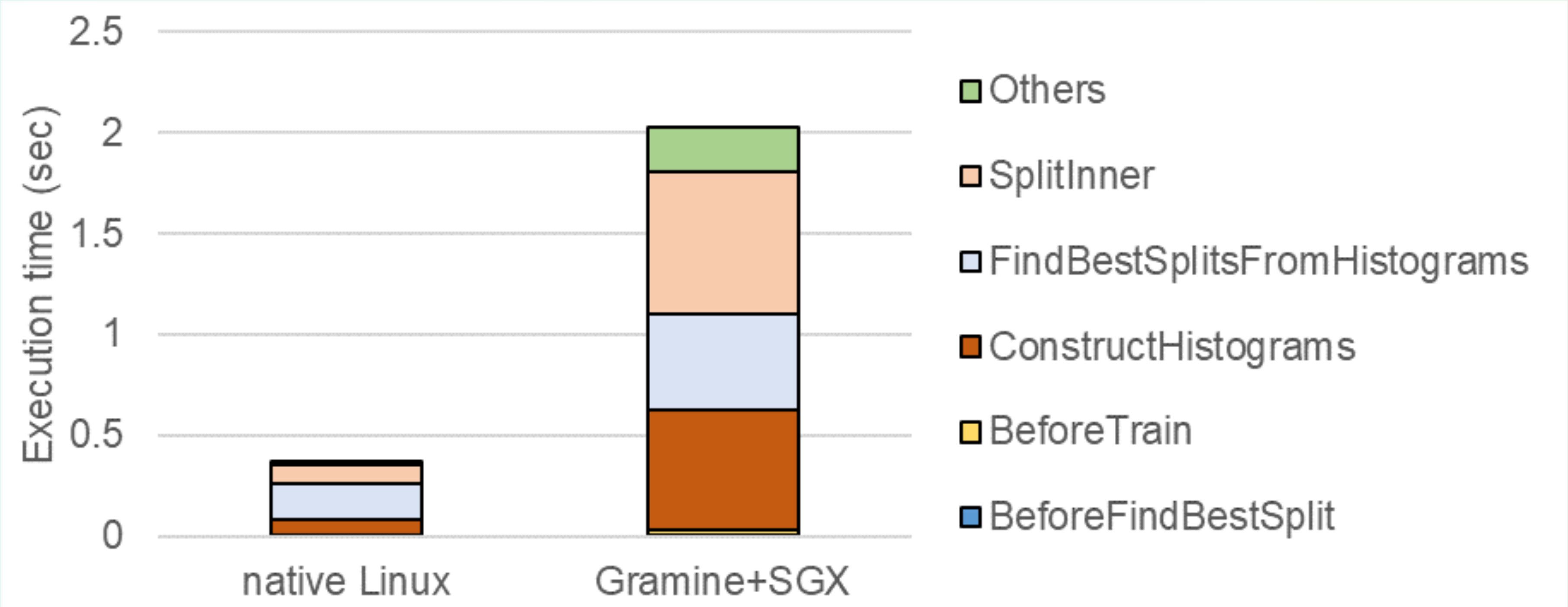
[Execution time]



[Slowdown]

Breakdown of LightGBM Execution Time

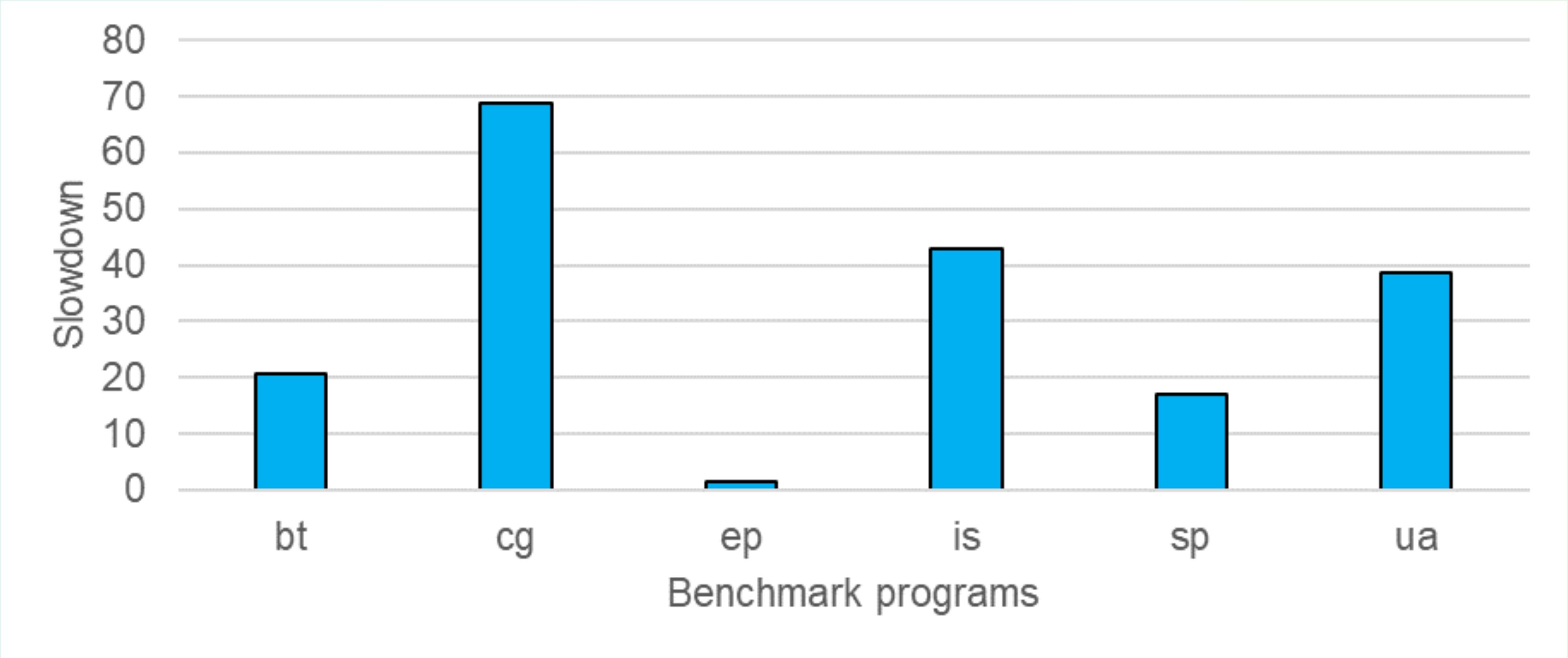
- *ConstructHistgrams* and *SplitInner* are largely influenced by Gramine+SGX (slowdowns of 7.5x and 7.6x)
- These functions frequently invoke many system calls such as *memset,* which need to be emulated by Gramine



[Breakdown of execution time]

Impact of SGX Architectural Improvement

- We executed NPB also on Core i7-9700 (Coffee Lake), which has a 128MB enclave memory
- The slowdown caused by Gramine+SGX is significant due to a number of memory swaps



[Performance of NPB on Core i7-9700]