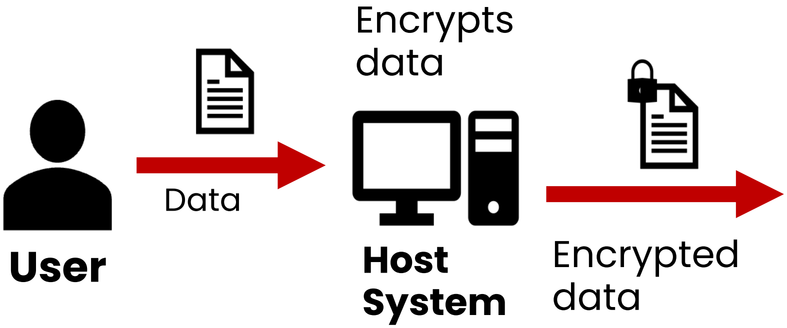
Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System

Harshita Gupta* Mayank Kabra*
Juan Gómez-Luna Konstantinos Kanellopoulos
Onur Mutlu

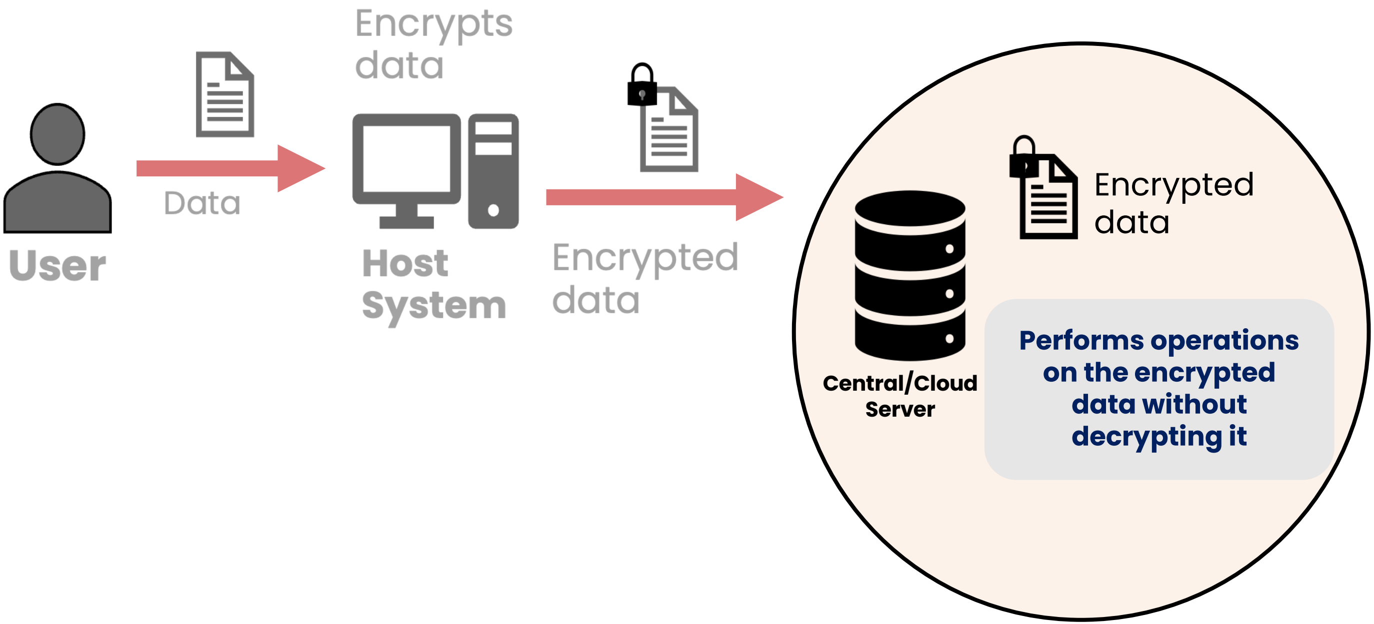
SAFARI ETH zürich

Homomorphic Encryption

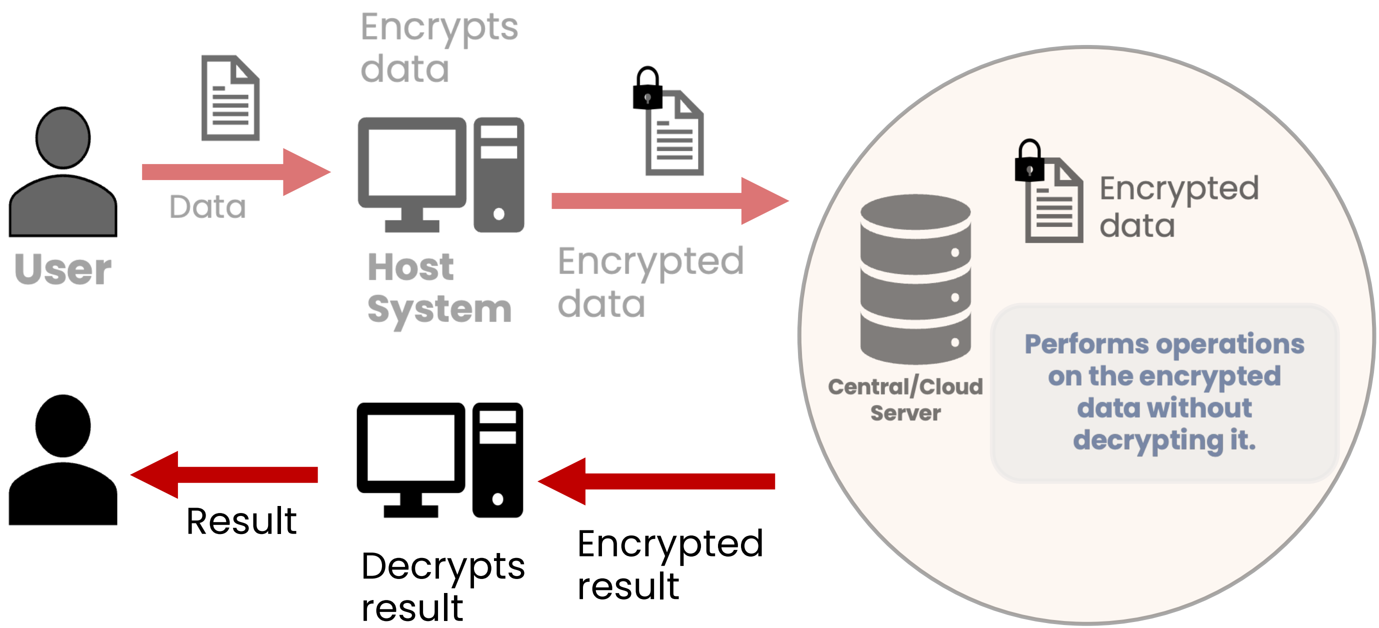
Homomorphic Encryption



Homomorphic Encryption



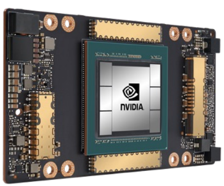
Homomorphic Encryption

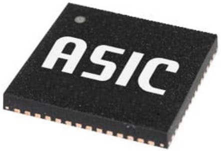


Motivation

Homomorphic operations suffer from **large memory capacity and data movement bottlenecks**

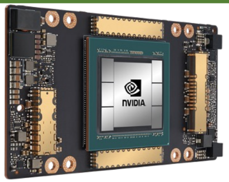
Acceleration Techniques





Motivation

These approaches face challenges in **resource limitations, data movement, and practical implementation**

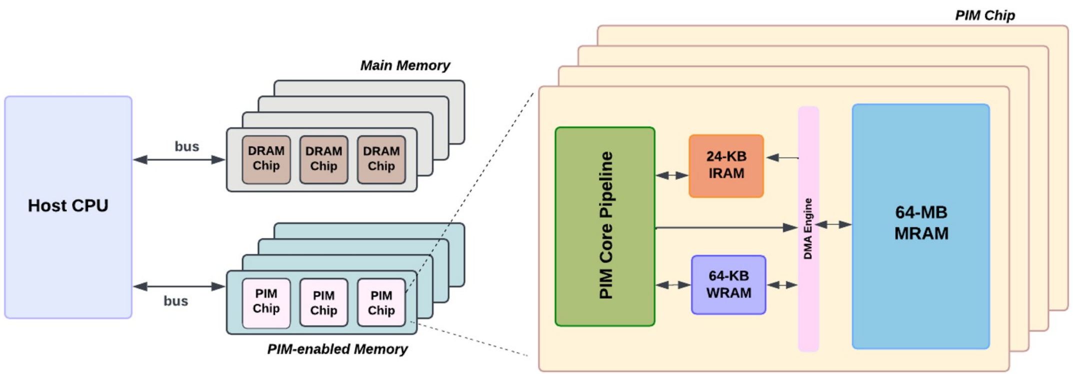


Our Goal

Evaluate the suitability of real-world general-purpose processing-in-memory (PIM) architectures to perform homomorphic operations.

Our Goal

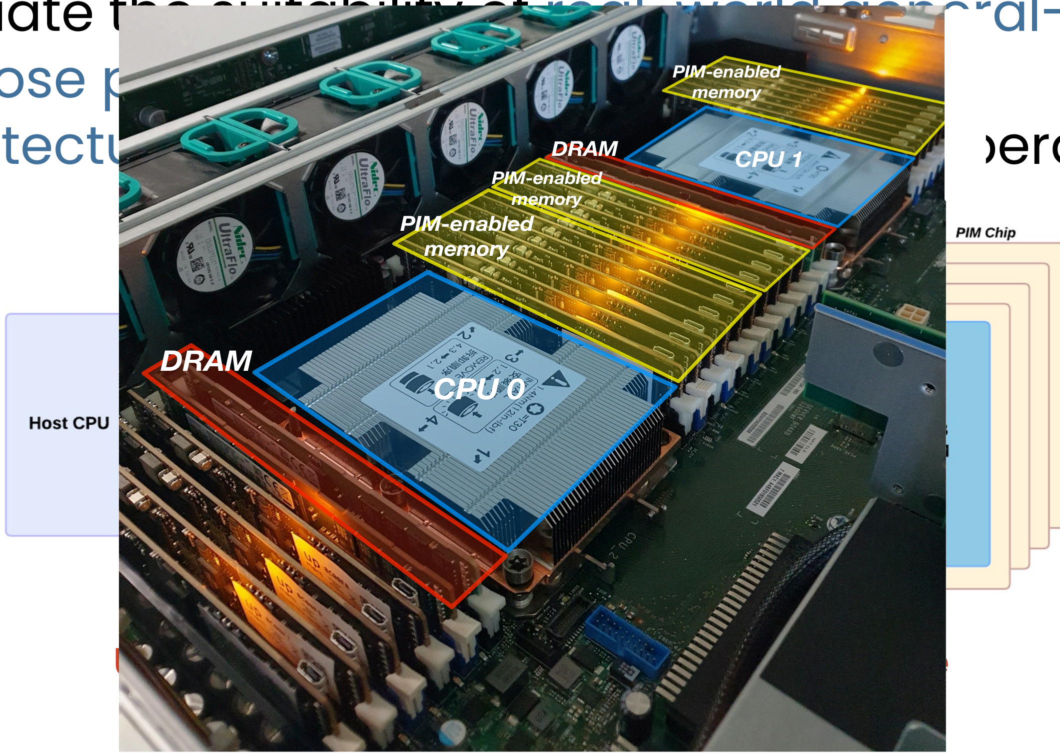
Evaluate the suitability of real-world general-purpose processing-in-memory (PIM) architectures to perform homomorphic operations.



UPMEM: First Real World PIM Architecture

Our Goal

Evaluate t… ral-
purpose p…
architectu… perations.



Evaluation Methodology

①

Evaluation of **homomorphic addition and multiplication** on UPMEM PIM system

Evaluation Methodology

① Evaluation of **homomorphic addition and multiplication** on UPMEM PIM system

↓

② Evaluation with statistical workloads **(mean, variance, linear regression)**

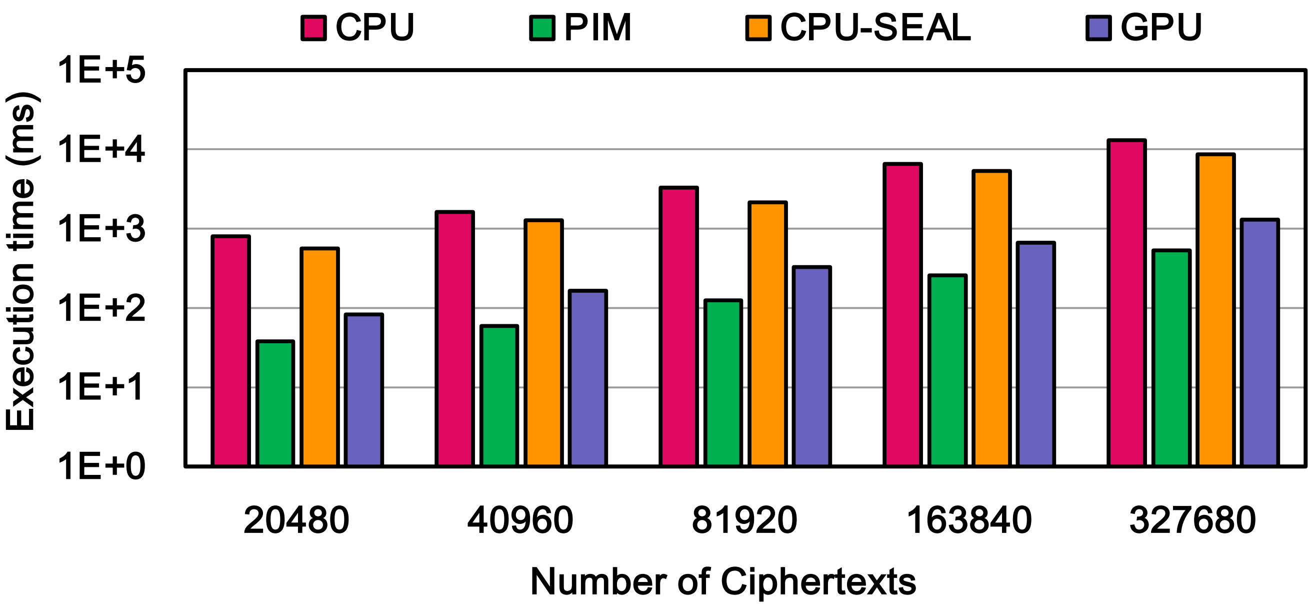
Evaluation Methodology

① Evaluation of **homomorphic addition and multiplication** on UPMEM PIM system

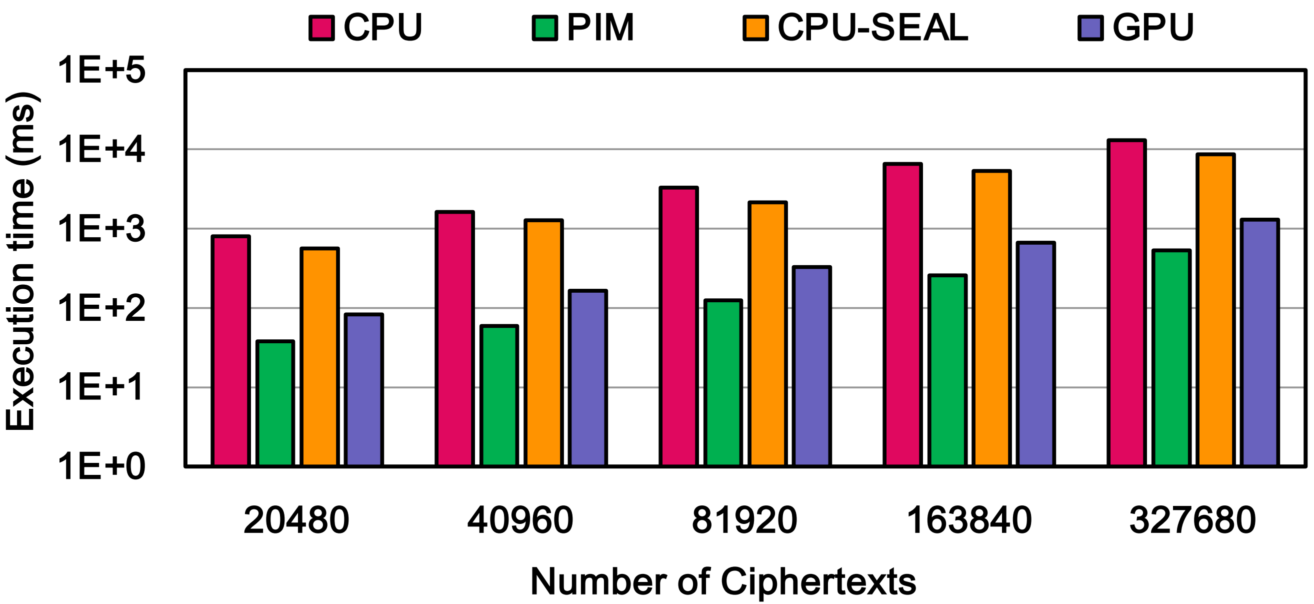
② Evaluation with statistical workloads **(mean, variance, linear regression)**

③ Comparison with custom **CPU and GPU** libraries and an optimized **CPU library (SEAL)**

Evaluation: Homomorphic Addition



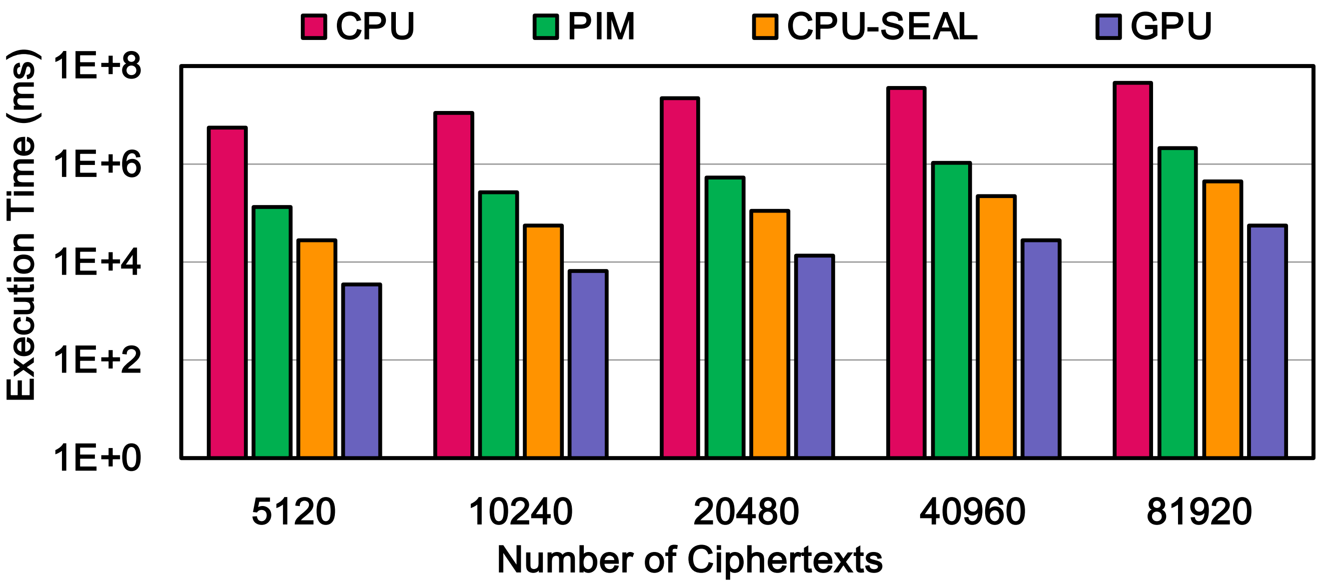
Evaluation: Homomorphic Addition



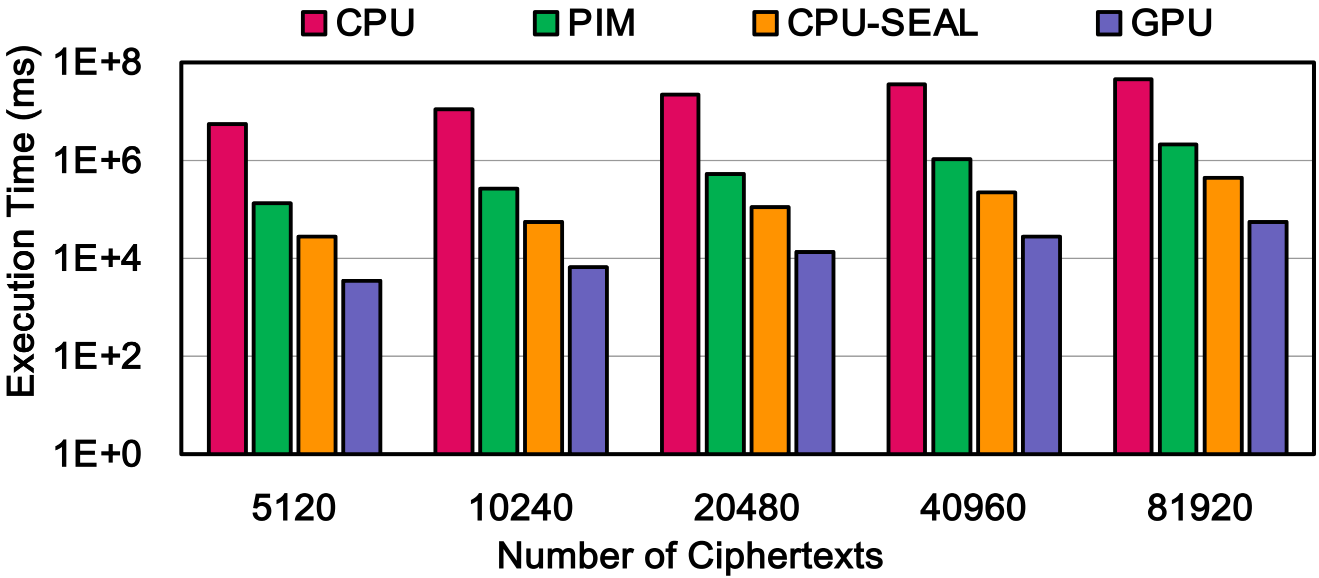
50 – 100× speedup provided by PIM over CPU

2 – 15× speedup over GPU

Evaluation: Homomorphic Multiplication

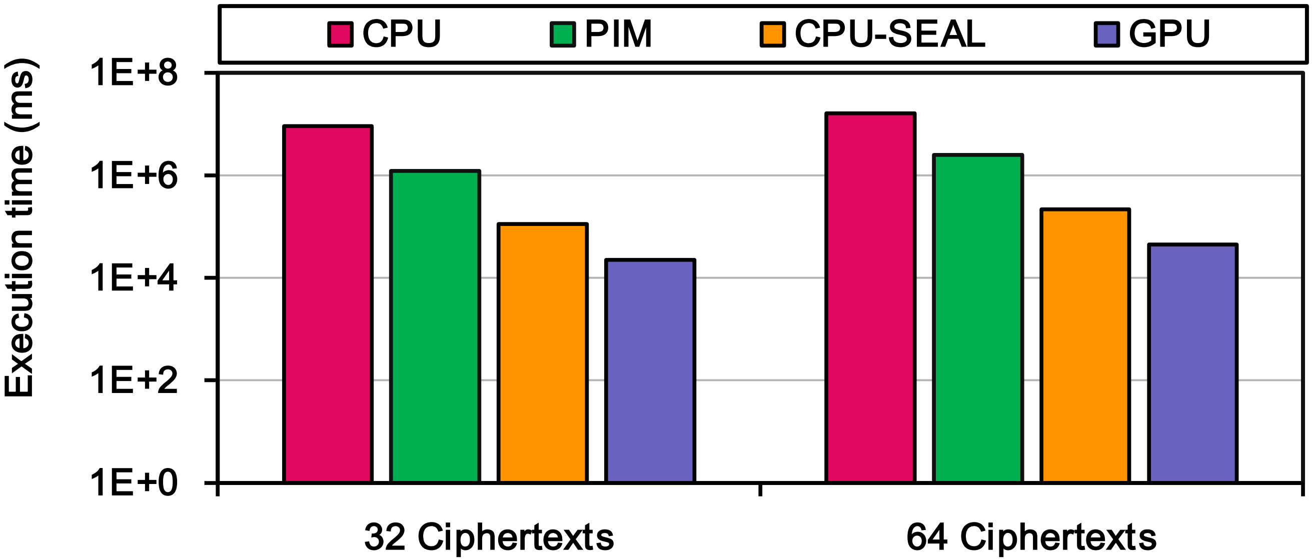


Evaluation: Homomorphic Multiplication

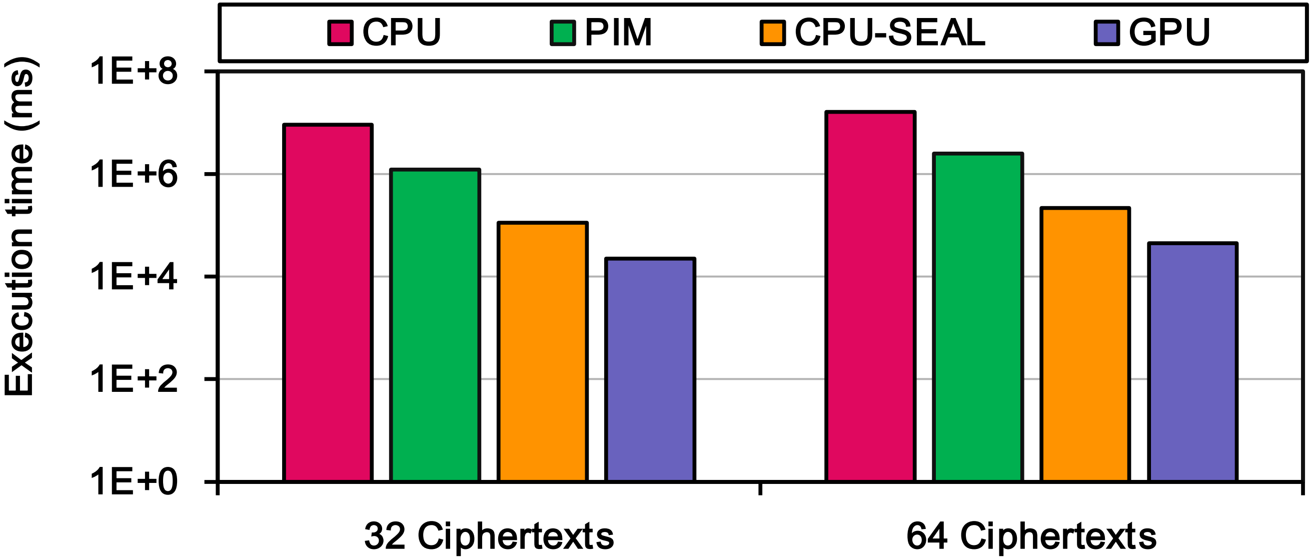


PIM lags 10 – 15× behind the GPU due to the lack of native multiplication support

Evaluation: Linear Regression



Evaluation: Linear Regression



PIM is 6.4–7.5x faster than the custom CPU implementation

CPU–SEAL and GPU are faster than PIM

Key Takeaways

①

UPMEM PIM system natively supports 32-bit integer addition and outperforms CPU and GPU for homomorphic ***addition***

Key Takeaways

① **UPMEM PIM system natively supports 32-bit integer addition and outperforms** CPU and GPU for homomorphic ***addition***

② The **lack of native support for 32-bit integer multiplication hampers the performance** of PIM for homomorphic ***multiplication***.

Key Takeaways

① **UPMEM PIM system natively supports 32-bit integer addition and outperforms** CPU and GPU for homomorphic ***addition***

② The **lack of native support for 32-bit integer multiplication hampers the performance** of PIM for homomorphic ***multiplication***.

③ The **computational power of PIM scales with memory capacity** via the addition of more memory banks and PIM cores

Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System

Harshita Gupta* Mayank Kabra*
Juan Gómez-Luna Konstantinos Kanellopoulos
Onur Mutlu

SAFARI ETH zürich