SAFARI

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

ETH zürich

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

Problem Statement

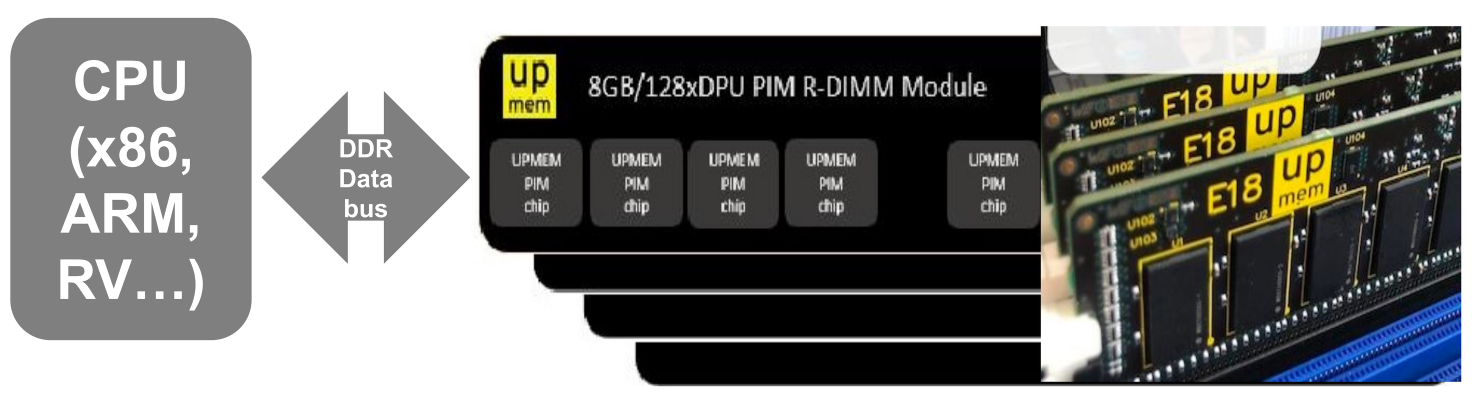
- HE enables processing on encrypted data to facilitate secure computation
- HE suffers from memory usage and data transfer bottlenecks on processor-centric chips, hampering scalability and performance
- Acceleration efforts with GPUs, FPGAs, and ASICs face ongoing challenges related to resource limitations, data transfer, and practical ASIC implementation

Our Goal

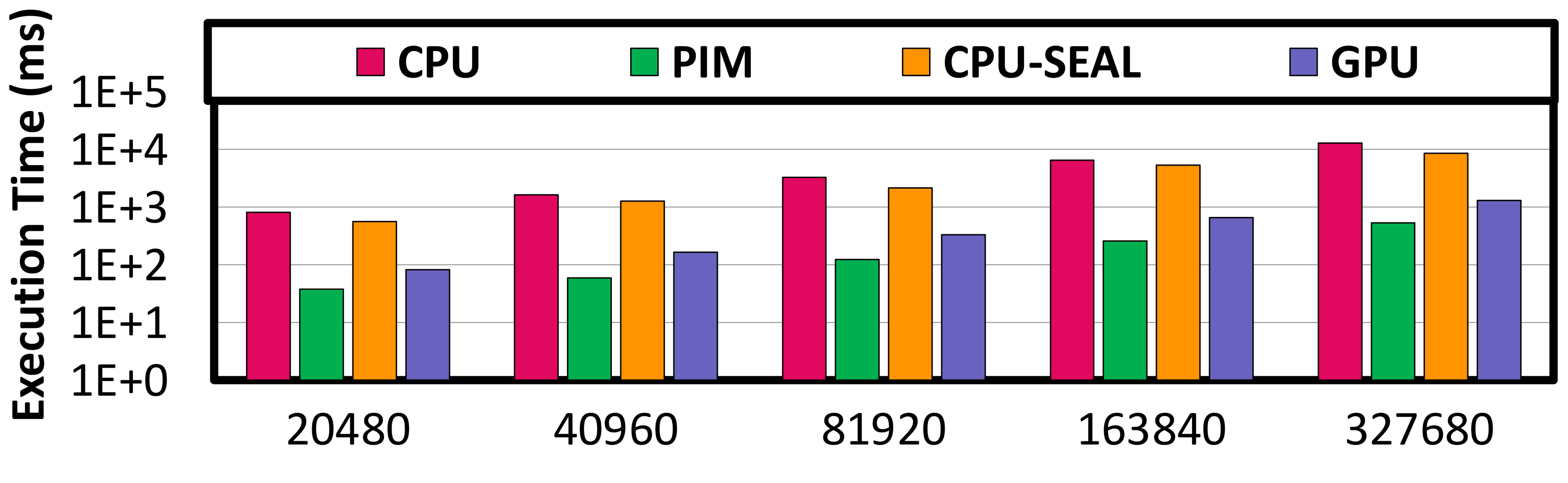
Evaluate the suitability of real-world general-purpose processing-in-memory architectures to perform homomorphic operations

Evaluation Methodology

- We offload and evaluate homomorphic operations on UPMEM PIM, Intel i5-8250U CPU, and NVIDIA A100 GPU
- Comparisons to custom CPU/GPU and optimized SEAL CPU libraries
- Microbenchmarks cover addition/multiplication with varying ciphertexts (32, 64, 128 bits)
- Statistical workloads: (i) mean (ii) variance (iii) linear regression



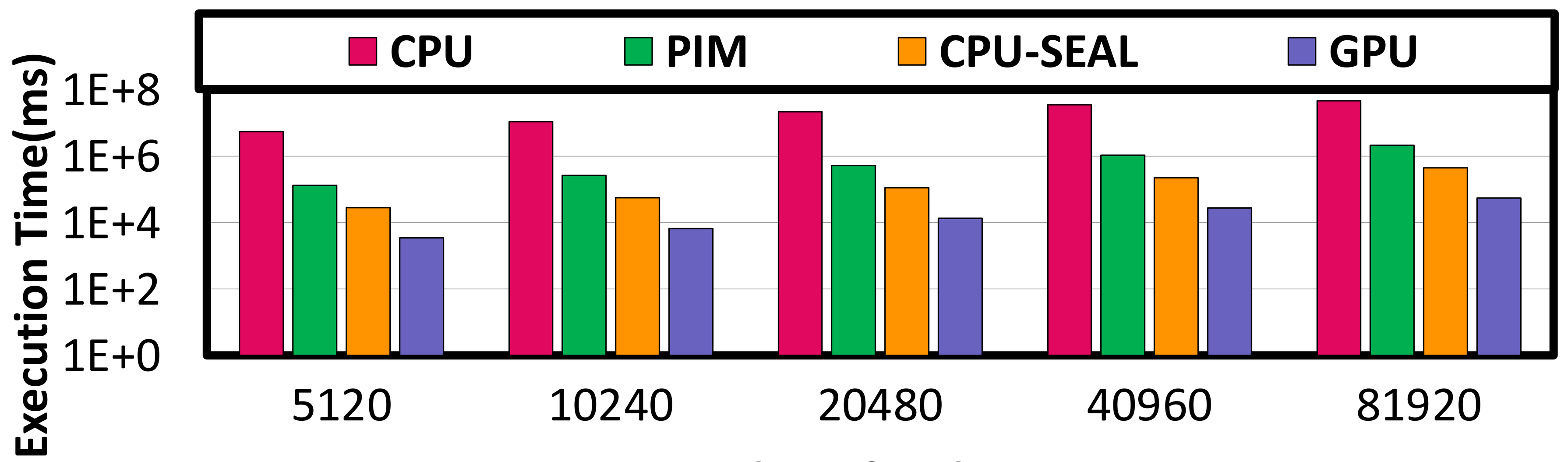
Homomorphic Vector Addition & Multiplication



PIM system is 50-100× faster than CPU and 2-15× faster than GPU in vector addition

KEY TAKEAWAY #1

UPMEM PIM excels in homomorphic addition with native 32-bit integer support, surpassing CPUs and GPUs.



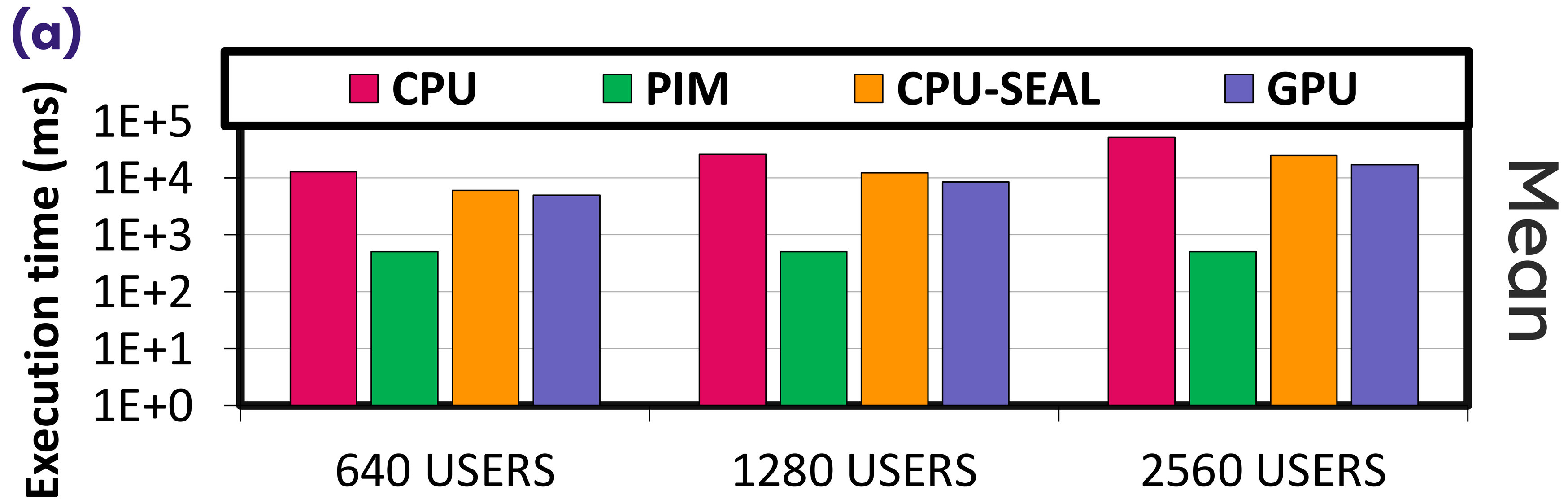
PIM outperforms custom CPU library by 40-50× in vector multiplication but lags 10-15× behind GPU

KEY TAKEAWAY #2

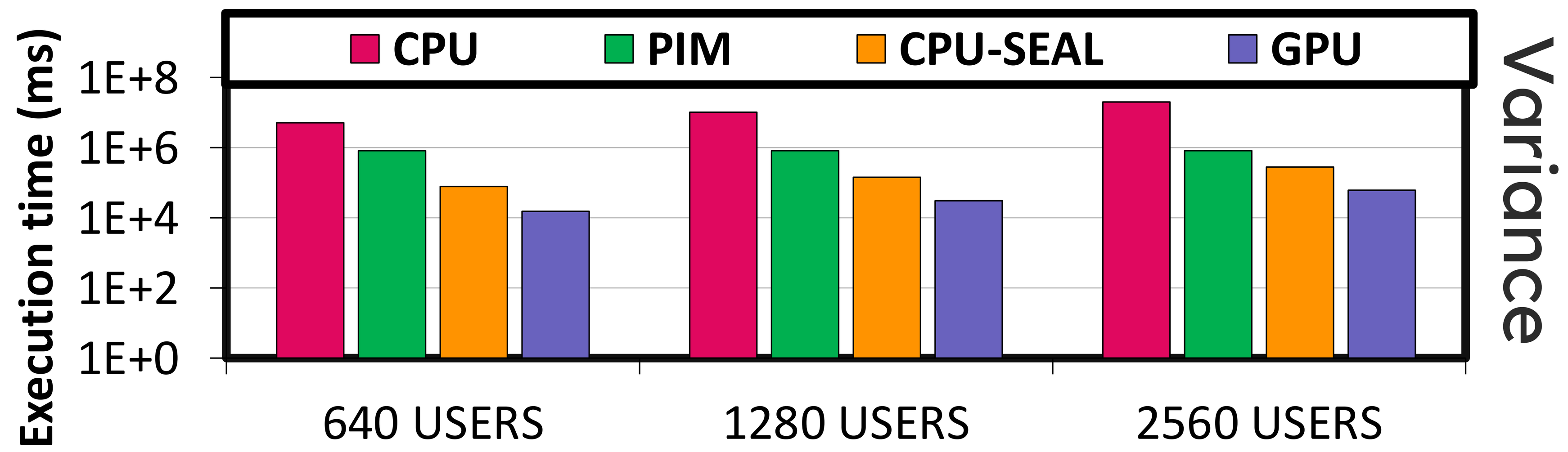
PIM performance for homomorphic multiplication lags due to the absence of native 32-bit integer multiplication support, but future PIM systems may outperform CPUs and GPUs.

Statistical Workloads

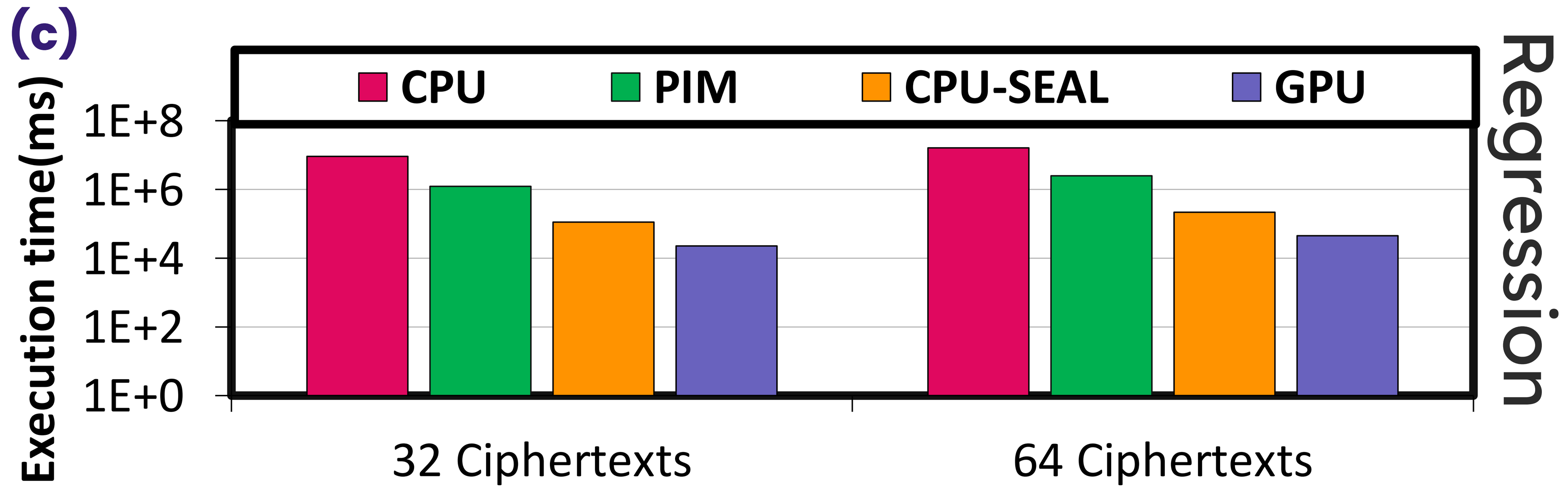
(a)



(b)



(c)



For statistical operations, PIM achieves 30× to 300× improvement over CPU and 10× to 30× over GPU

KEY TAKEAWAY #3

PIM's computational power scales with memory capacity via more memory banks and cores.