

Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

Juan Gómez Luna, Izzat El Hajj,
Ivan Fernandez, Christina Giannoula,
Geraldo F. Oliveira, Onur Mutlu

<https://arxiv.org/pdf/2105.03814.pdf>

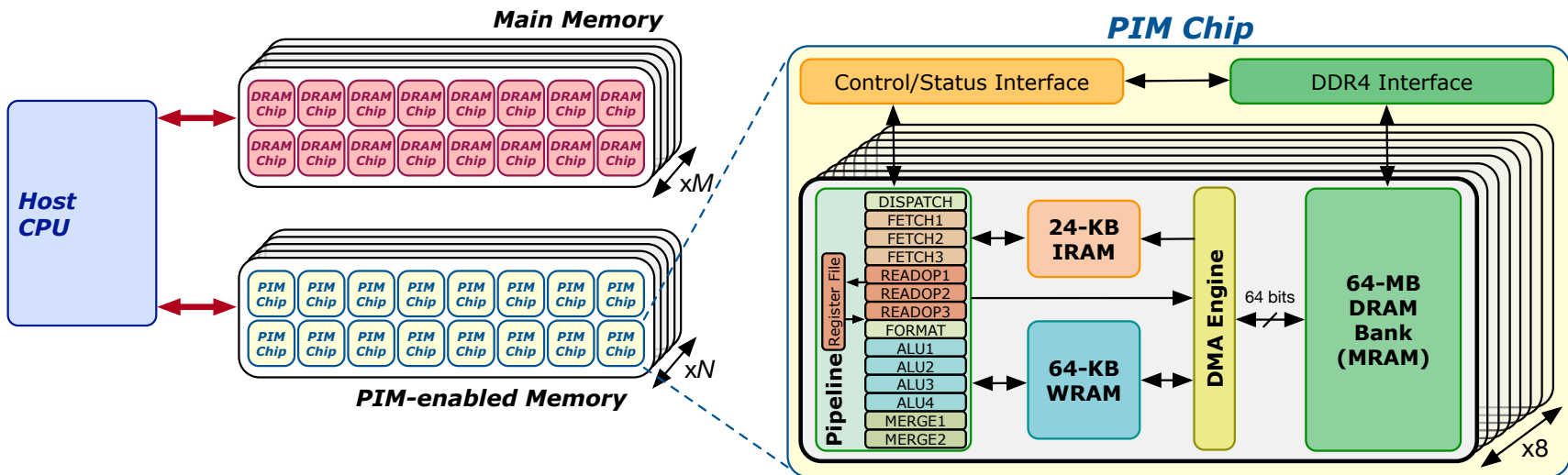
<https://github.com/CMU-SAFARI/prim-benchmarks>

Executive Summary

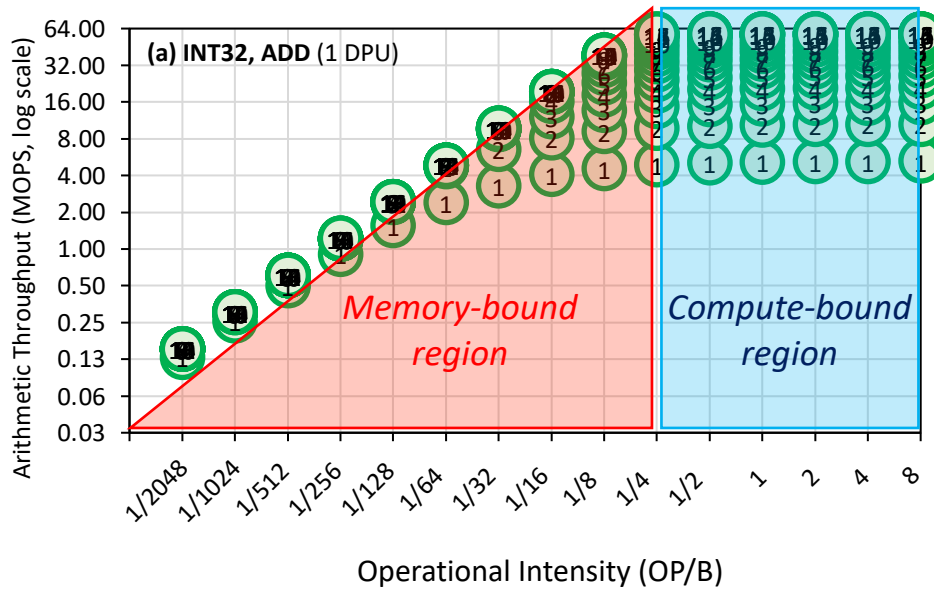
- **Data movement** between memory/storage units and compute units is a major contributor to execution time and energy consumption
- **Processing-in-Memory** (PIM) is a paradigm that can tackle the *data movement bottleneck*
 - Though explored for +50 years, technology challenges prevented the successful materialization
- UPMEM has designed and fabricated **the first publicly-available real-world PIM architecture**
 - DDR4 chips embedding in-order multithreaded DRAM Processing Units (DPUs)
- Our work:
 - **Introduction** to UPMEM programming model and PIM architecture
 - **Microbenchmark-based characterization** of the DPU
 - Benchmarking and **workload suitability** study
- Main contributions:
 - Comprehensive **characterization and analysis of the first commercially-available PIM architecture**
 - **PrIM** (Processing-In-Memory) benchmarks:
 - 16 workloads that are memory-bound in conventional processor-centric systems
 - Strong and weak scaling characteristics
 - Comparison to **state-of-the-art CPU and GPU**
- Takeaways:
 - Workload characteristics for **PIM suitability**
 - **Programming** recommendations
 - Suggestions and hints for **hardware and architecture designers** of future PIM systems
 - **PrIM**: (a) programming samples, (b) evaluation and comparison of current and future PIM systems

System Organization

- A UPMEM DIMM contains 8 or 16 chips
 - Thus, 1 or 2 ranks of 8 chips each
- Inside each PIM chip there are:
 - 8 64MB banks per chip: Main RAM (MRAM) banks
 - 8 DRAM Processing Units (DPUs) in each chip, 64 DPUs per rank



Key Takeaway 1

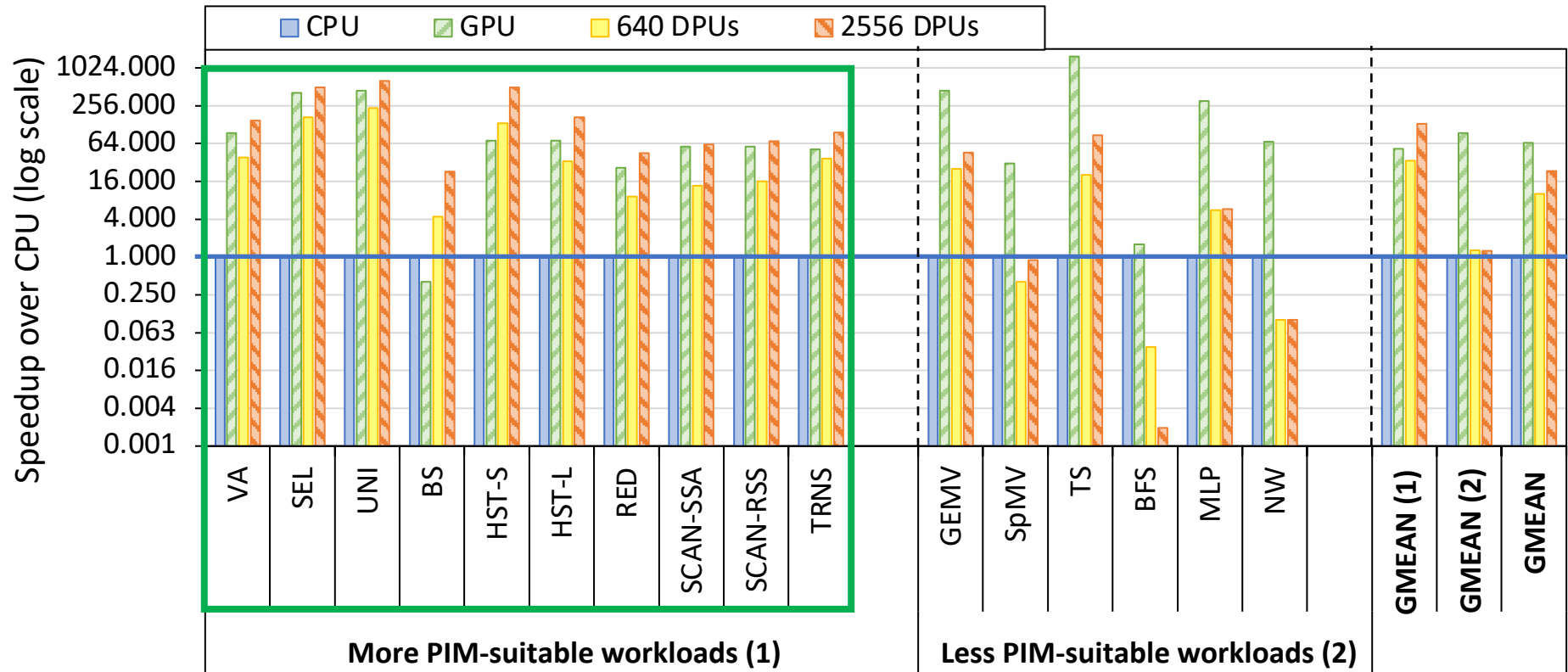


The throughput saturation point is as low as $\frac{1}{4}$ OP/B, i.e., 1 integer addition per every 32-bit element fetched

KEY TAKEAWAY 1

The UPMEM PIM architecture is fundamentally compute bound. As a result, the most suitable workloads are memory-bound.

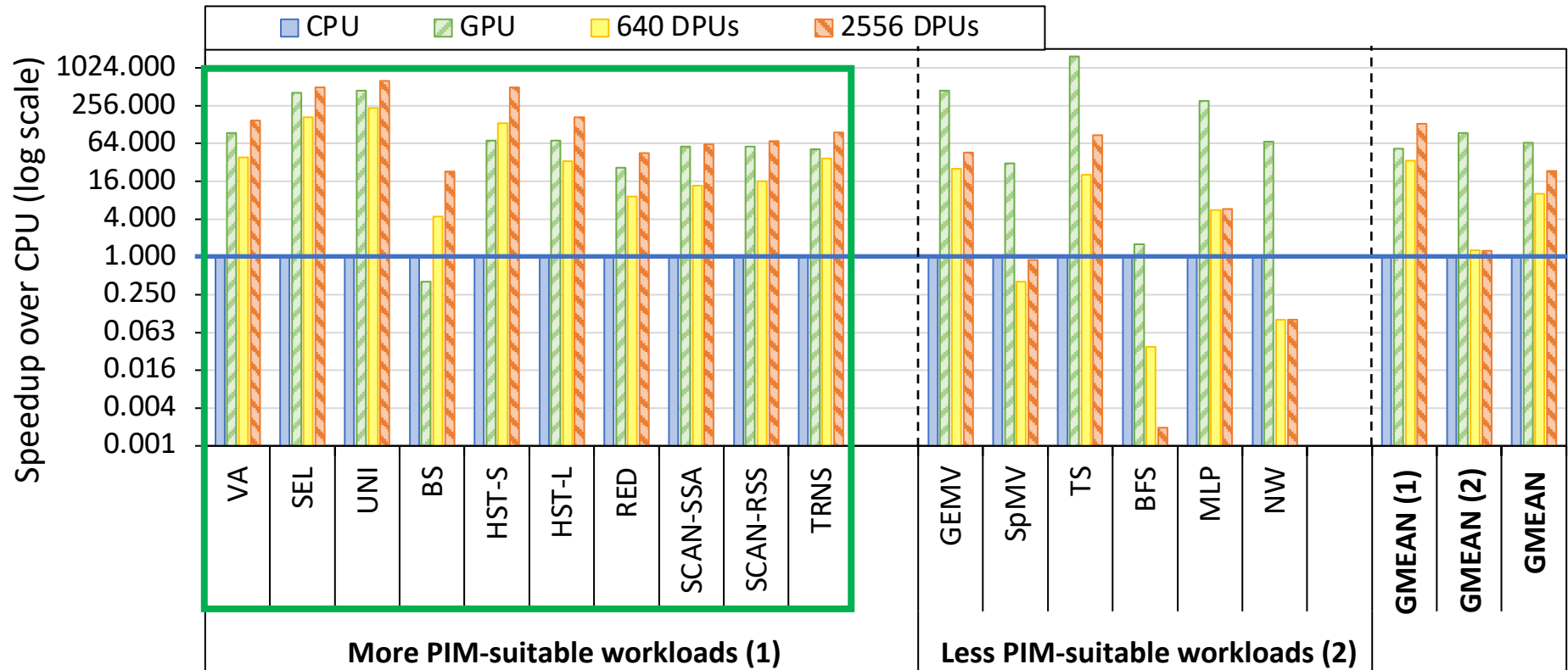
Key Takeaway 2



KEY TAKEAWAY 2

The most well-suited workloads for the UPMEM PIM architecture use no arithmetic operations or use only simple operations (e.g., bitwise operations and integer addition/subtraction).

Key Takeaway 3



KEY TAKEAWAY 3

The most well-suited workloads for the UPMEM PIM architecture require little or no communication across DPUs (inter-DPU communication).

Key Takeaway 4

KEY TAKEAWAY 4

- UPMEM-based PIM systems **outperform state-of-the-art CPUs in terms of performance and energy efficiency on most of PrIM benchmarks.**
- UPMEM-based PIM systems **outperform state-of-the-art GPUs on a majority of PrIM benchmarks**, and the outlook is even more positive for future PIM systems.
- UPMEM-based PIM systems are **more energy-efficient than state-of-the-art CPUs and GPUs on workloads that they provide performance improvements** over the CPUs and the GPUs.

Understanding a Modern PIM Architecture

Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

Juan Gómez-Luna¹ Izzat El Hajj² Ivan Fernandez^{1,3} Christina Giannoula^{1,4}
Geraldo F. Oliveira¹ Onur Mutlu¹

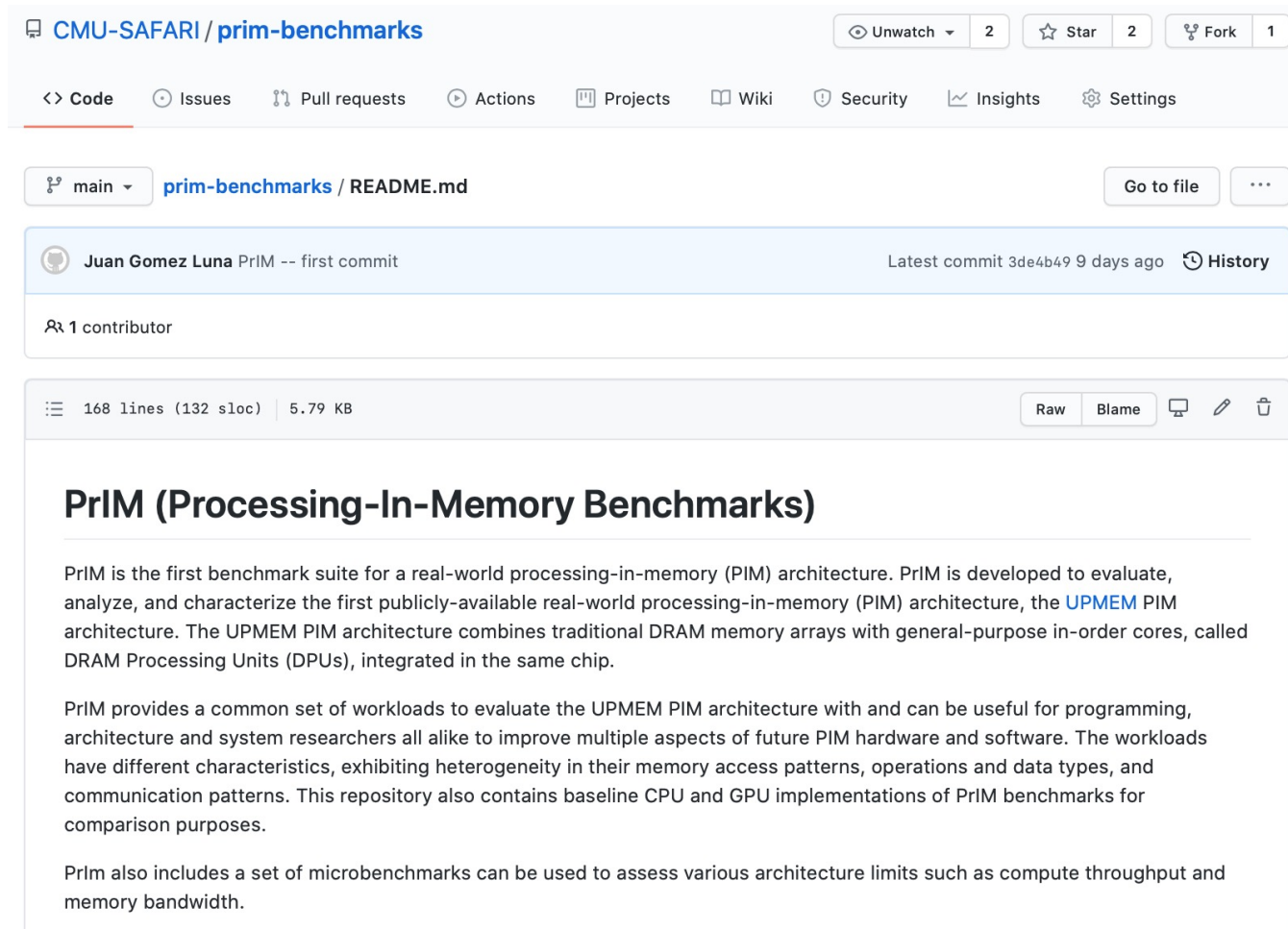
¹ETH Zürich ²American University of Beirut ³University of Malaga ⁴National Technical University of Athens

<https://arxiv.org/pdf/2105.03814.pdf>

<https://github.com/CMU-SAFARI/prim-benchmarks>

PrIM Repository

- All microbenchmarks, benchmarks, and scripts
- <https://github.com/CMU-SAFARI/prim-benchmarks>



CMU-SAFARI / prim-benchmarks

Unwatch 2 Star 2 Fork 1

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main prim-benchmarks / README.md Go to file

Juan Gomez Luna PrIM -- first commit Latest commit 3de4b49 9 days ago History

1 contributor

168 lines (132 sloc) 5.79 KB Raw Blame

PrIM (Processing-In-Memory Benchmarks)

PrIM is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publicly-available real-world processing-in-memory (PIM) architecture, the [UPMEM PIM](#) architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM Processing Units (DPUs), integrated in the same chip.

PrIM provides a common set of workloads to evaluate the UPMEM PIM architecture with and can be useful for programming, architecture and system researchers all alike to improve multiple aspects of future PIM hardware and software. The workloads have different characteristics, exhibiting heterogeneity in their memory access patterns, operations and data types, and communication patterns. This repository also contains baseline CPU and GPU implementations of PrIM benchmarks for comparison purposes.

Prim also includes a set of microbenchmarks can be used to assess various architecture limits such as compute throughput and memory bandwidth.

Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

Juan Gómez Luna, Izzat El Hajj,
Ivan Fernandez, Christina Giannoula,
Geraldo F. Oliveira, Onur Mutlu

el1goluj@gmail.com

<https://arxiv.org/pdf/2105.03814.pdf>

<https://github.com/CMU-SAFARI/prim-benchmarks>