PID-Comm: Fast and Flexible Collective Communication Framework for Commodity Processing-in-DIMM Devices

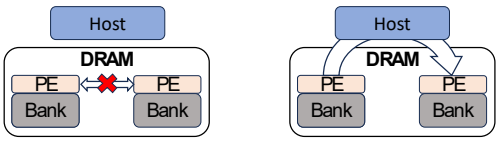
ISCA 2024

Si Ung Noh*, Junguk Hong*, Chaemin Lim, Seongyeon Park, Jeehyun Kim, Hanjun Kim, Youngsok Kim, and Jinho Lee

* Equal Contribution

Introduction

- **Processing in Memory (PIM)** is the solution for memory bottlenecks
  - **Inter-PE (Processing Element) communication** bottlenecks performance



- How can we **minimize communication overhead** for optimal performance?
  ➔ **PID-Comm!** The **1st collective communication library for PIM**

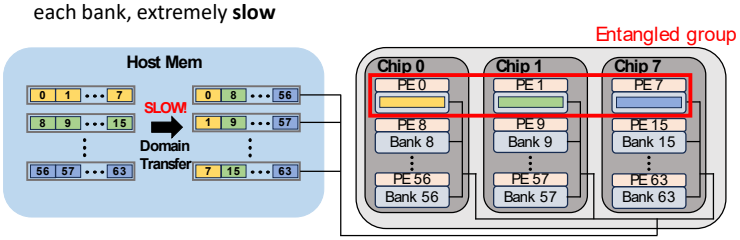
Background

1. PIM-Enabled DIMMs (UPMEM DIMMs)

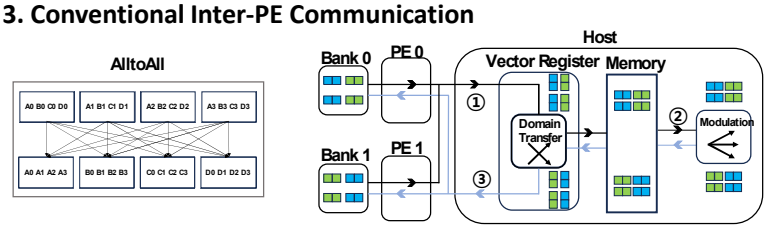
- PIM cores implemented on dual in-line memory modules
- Shares the same hierarchy as DDR4 DRAMs

2. Entangled Groups and Domain Transfers

- **Entangled Group :** A set of banks accessed together in a rank
- A 8B data is split into 1B segments and spread to an entangled group
- **Domain Transfer** : Reordering of data segments to place a full 8B word to each bank, extremely **slow**



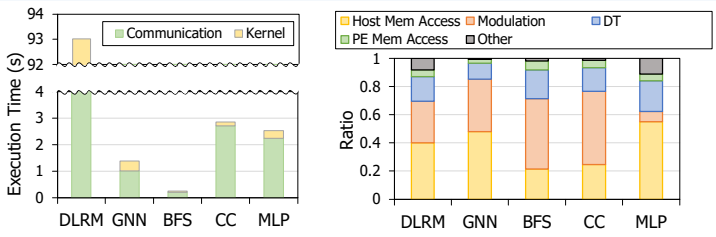
3. Conventional Inter-PE Communication



- Conventional inter-PE communication example targeting **collective communications**

① Data is sent from the banks to the vector registers and domain transferred
② Data is stored in host memory and modulated by the host
③ Result is domain transferred in the vector registers and sent to each bank

Motivation



- **Inefficient inter-PE communication** causes performance degradation

➔ Communication takes an average of 76% of total execution time
➔ Roots from **unnecessary host access, modulation, and domain transfer**

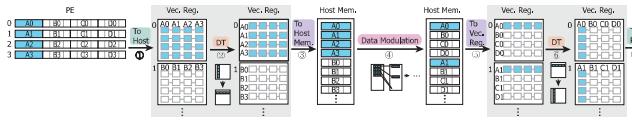
	Multi-Instance Communication	Performance	Supported Primitives							
			AA	RS	AG	AR	Sc	Ga	Re	Br
UPMEM SDK	Not Supported	Not Optimized					✓	✓		✓
SimplePIM	Not Supported	Not Optimized			✓	✓	✓	✓		✓
PID-Comm	**Supported**	**Optimized**	✓	✓	✓	✓	✓	✓	✓	✓

- **Lack of a flexible communication model**

➔ No prior approach supports a full set of collective primitives
➔ Lack of support for multi-instance communication

PID-Comm Design

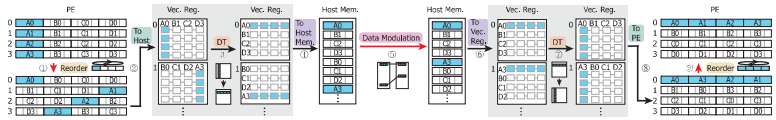
0. Baseline (AlltoAll)



1. Inter-PE Communication Optimization Techniques

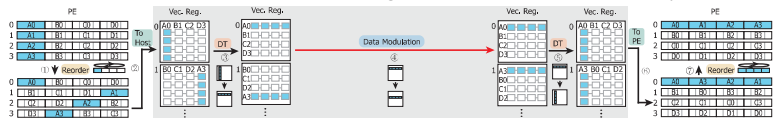
1-1. PE-Assisted Reordering

- **Reorder data inside PEs** to enhance data locality



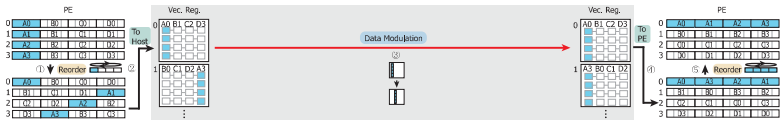
1-2. In-Register Modulation

- Modulate data **within the vector register** to eliminate host memory access



1-3. Cross-Domain Modulation

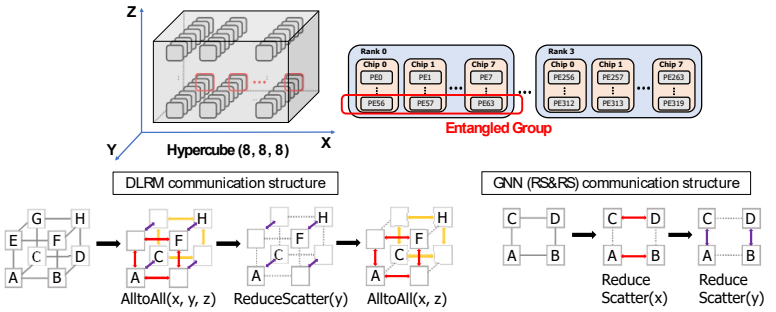
- Replace inefficient procedures with **light bitwise rotation**



2. Flexible Inter-PE Communication Framework

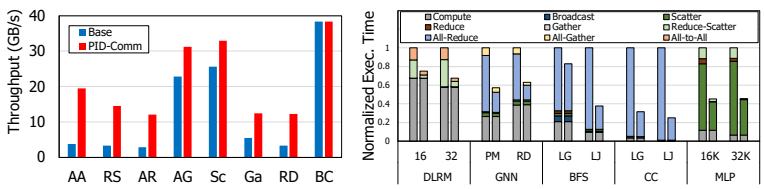
2-1. Virtual Hypercube Communication Model

- Form a virtual hypercube of PEs and **map the virtual hypercube to physical banks** for communication among any **N-dimensional partial cubes**



Evaluation

- Environment: Four PIM-enabled DIMMs (1024 PEs) / Intel Xeon Gold 5215 CPU
- **2.83x** geomean speedup against the best PIM baseline
- **1.99x** geomean speedup compared to the conventional PIM implementation across five benchmark applications



Conclusion

- PID-Comm is the **1st collective communication library for PIM devices**
- Provides micro-level optimizations and a flexible hypercube communication model
- **2.83x** geomean speedup compared to the state-of-the-art PIM baseline

Check out our github repo!