Is it Memory-Centric or Communication-Centric Computing?

John Kim
School of Electrical Engineering
KAIST
jjk12@kaist.edu

KAIST Korea Advanced Institute of Science and Technology

Data ~~Memory~~-Centric Computing

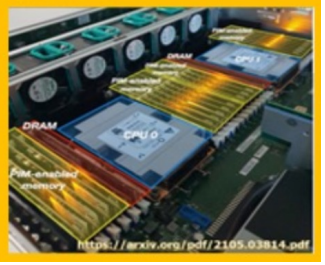
PPoPP 2025 - Tutorial on Memory-Centric Computing Systems

March 1st, Las Vegas, Nevada, USA

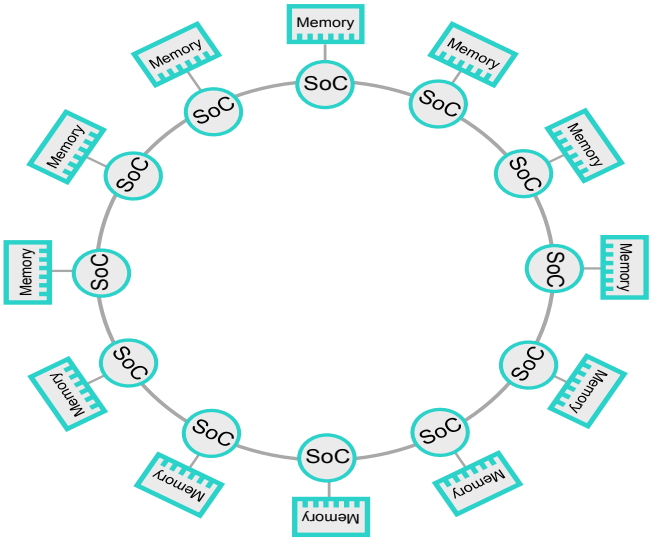
Organizers: Geraldo F. Oliveira, Dr. Mohammad Sadrosadati, Ataberk Olgun, Professor Onur Mutlu

Program: https://events.safari.ethz.ch/ppopp25-memorycentric-tutorial/

- Overview of PIM | PIM taxonomy
- PIM in memory & storage
- Real-world PNM systems
- PUM for bulk bitwise operations
- Programming techniques & tools
- Infrastructures for PIM Research
- Research challenges & opportunities



Memory-Centric (Memory-Driven) Computing



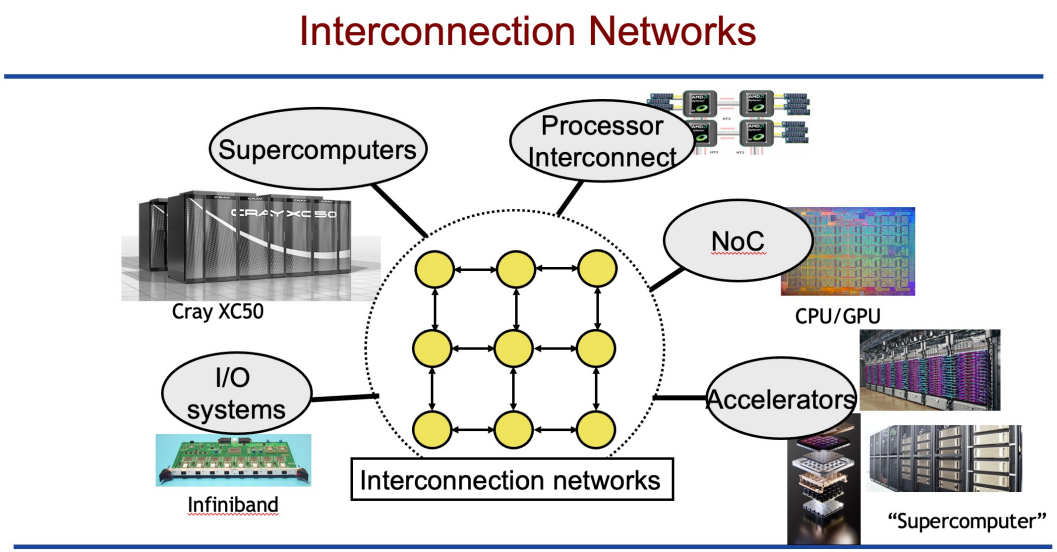
From Processor-Centric Computing

[image from Keeton FAST'17 Keynote]

What is Communication-Centric?

what is communication-centric computing?

Communication-centric computing is a paradigm where the focus is on optimizing data movement and communication between computing elements rather than just processing power. This approach is particularly relevant in large-scale distributed systems, cloud computing, edge computing, and high-performance computing (HPC), where efficient data exchange is crucial for performance.



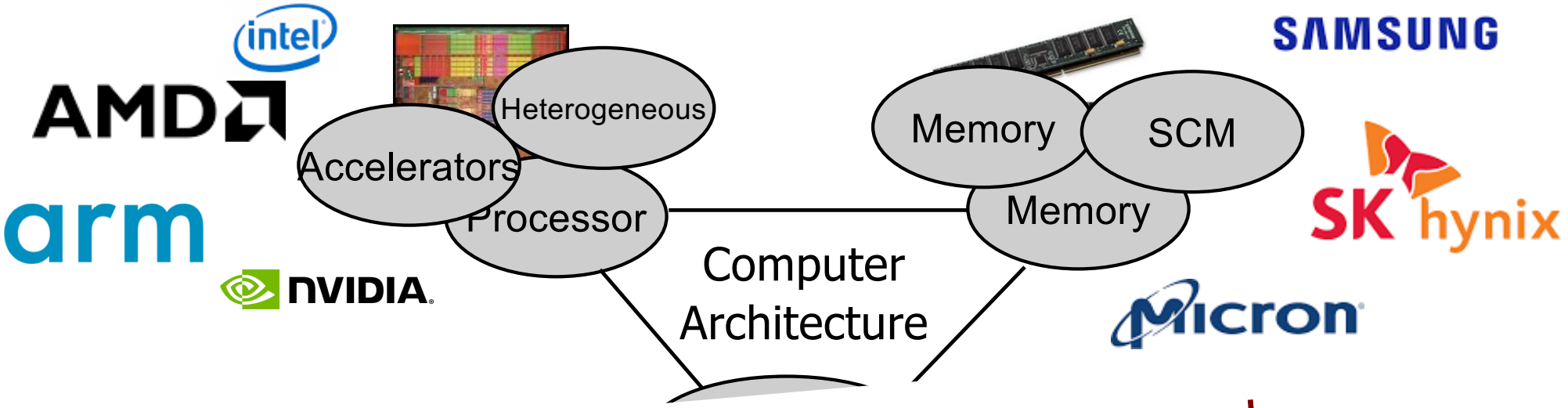
My definition:

0. Communication is always involved

1. When communication or data movement becomes a problem

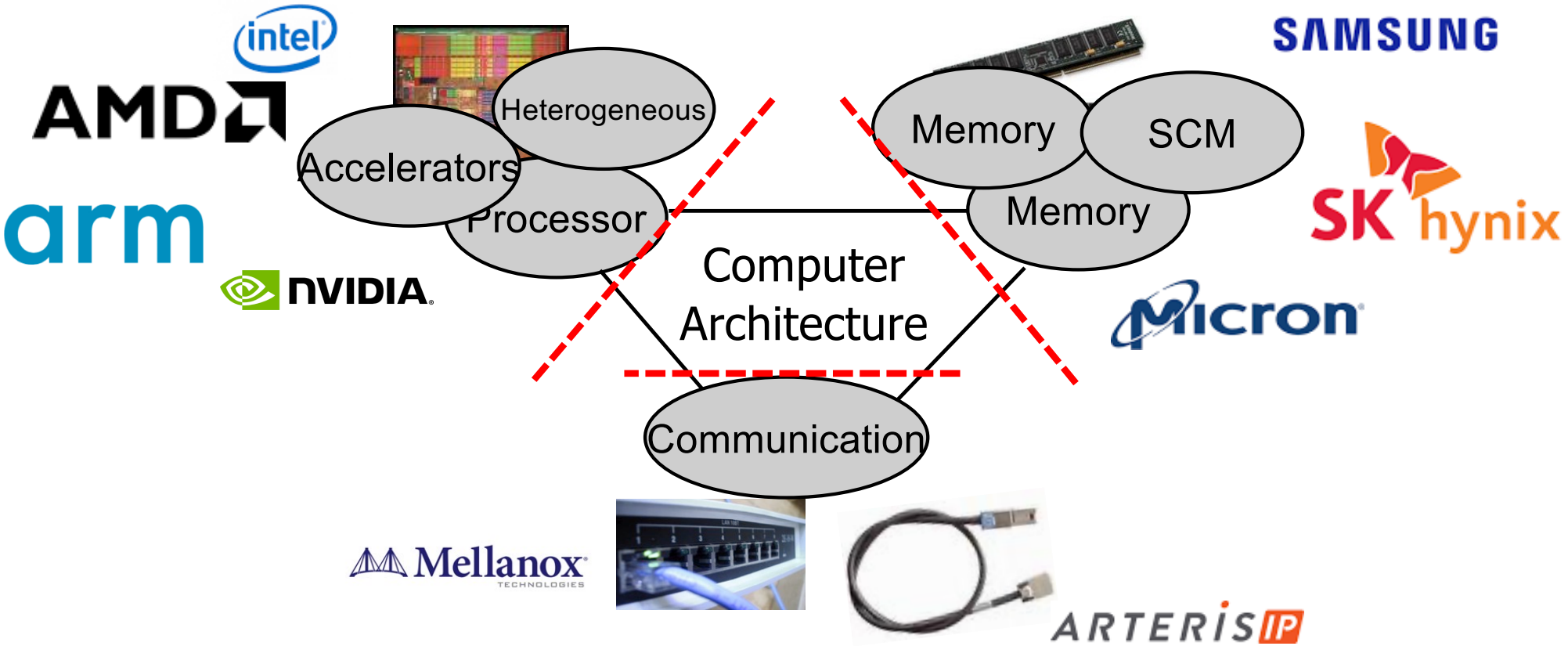
*2. **Exposed** data movement*

Computer Systems & Architecture

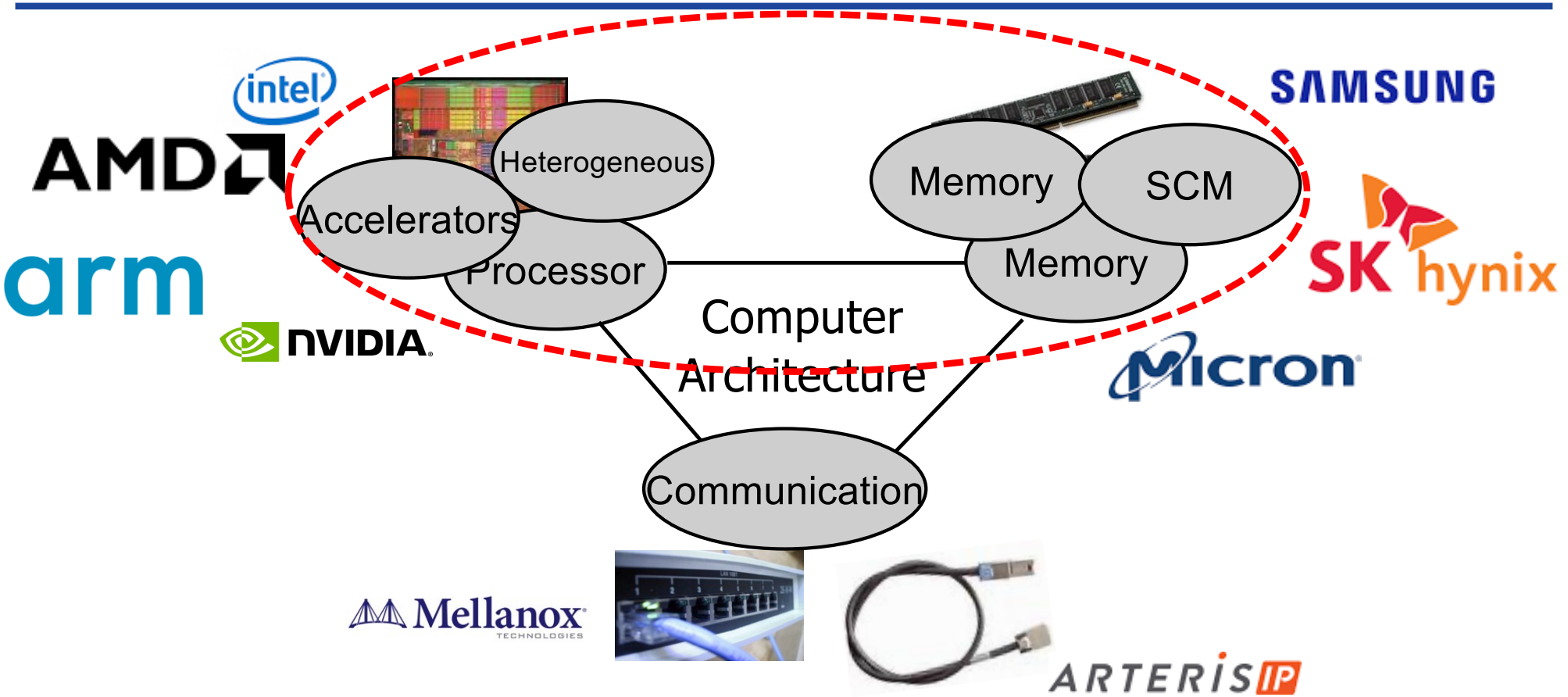


Interconnect : Moving bits around

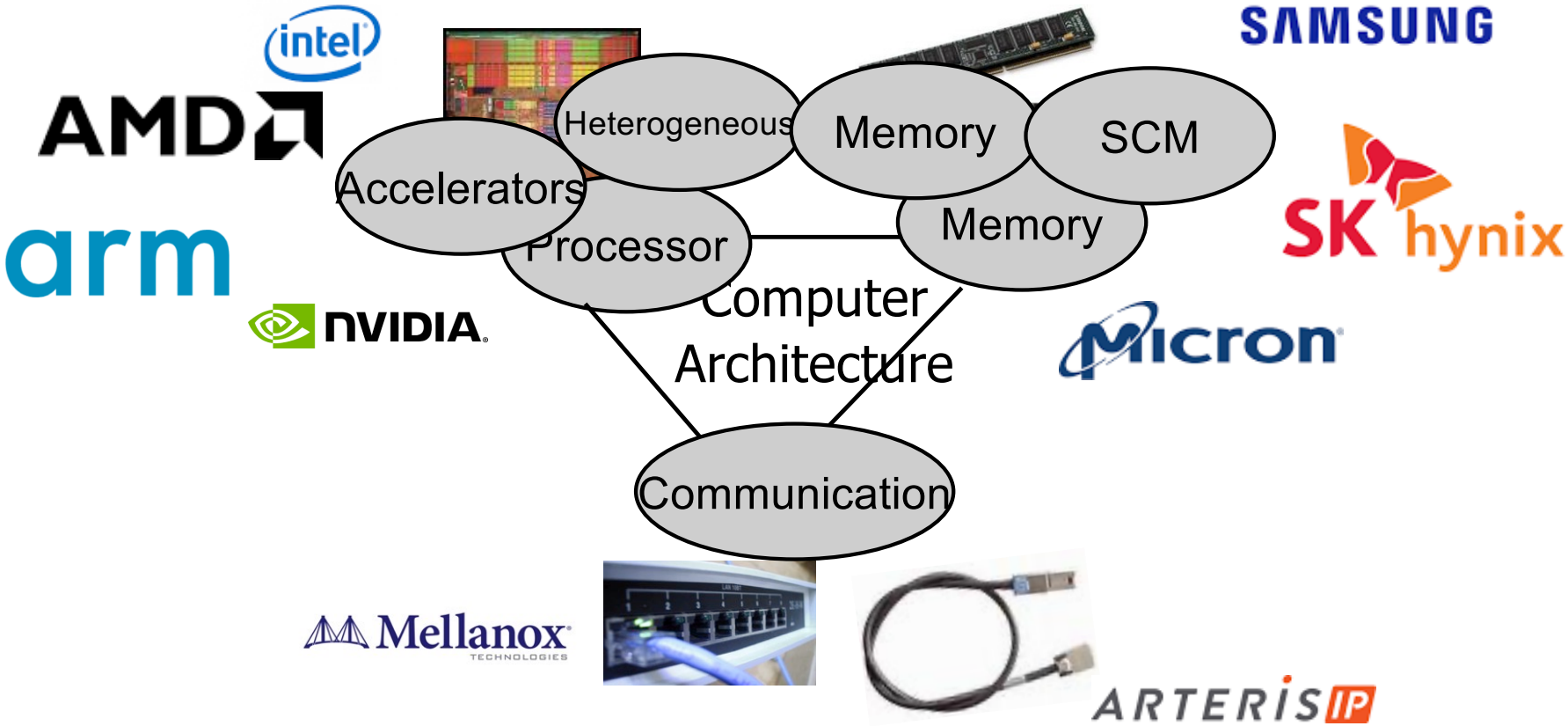
Computer Systems & Architecture



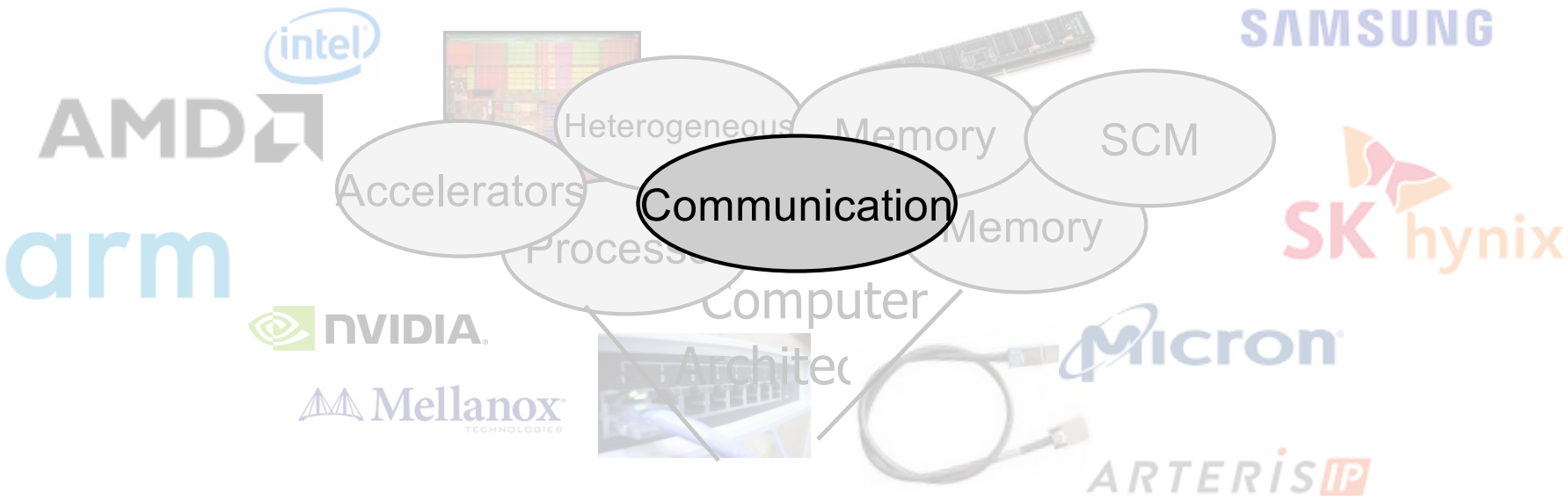
Co-design System Architecture



Blurry of the boundary



Blurrying of the boundary

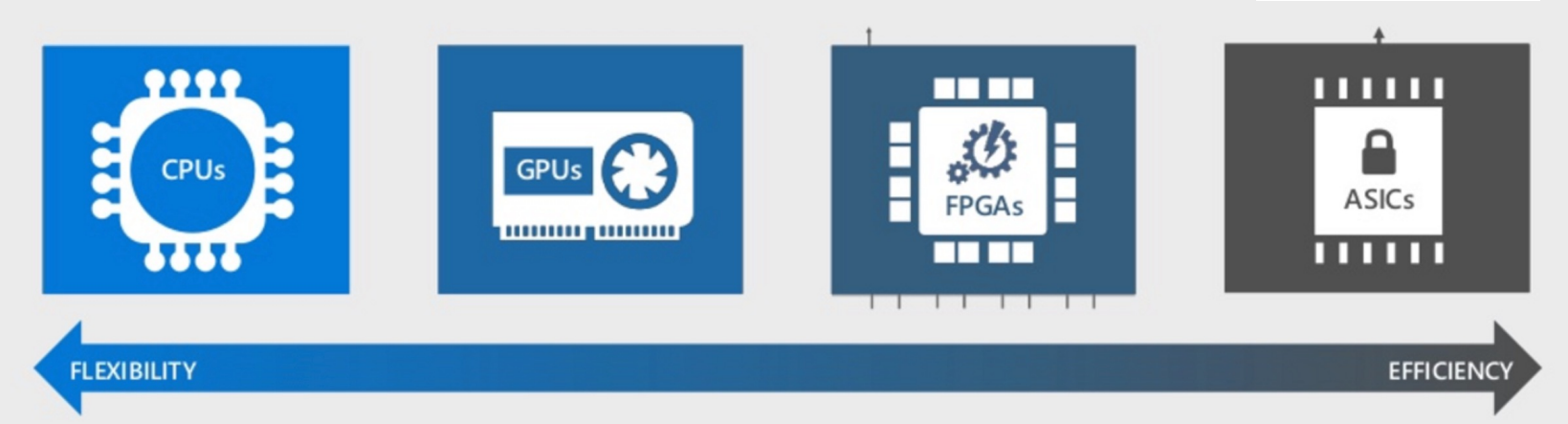


Today's Talk

- Introduction/Background
- Domain-specific architecture & beyond
- Minimizing data movement : Unified memory system
- Optimizing data movement : Domain-specific (PIM) interconnect
- Summary

Domain-Specific Architectures

Domain-Specific Arch



[Microsoft]

A New Golden Age for Computer Architecture:

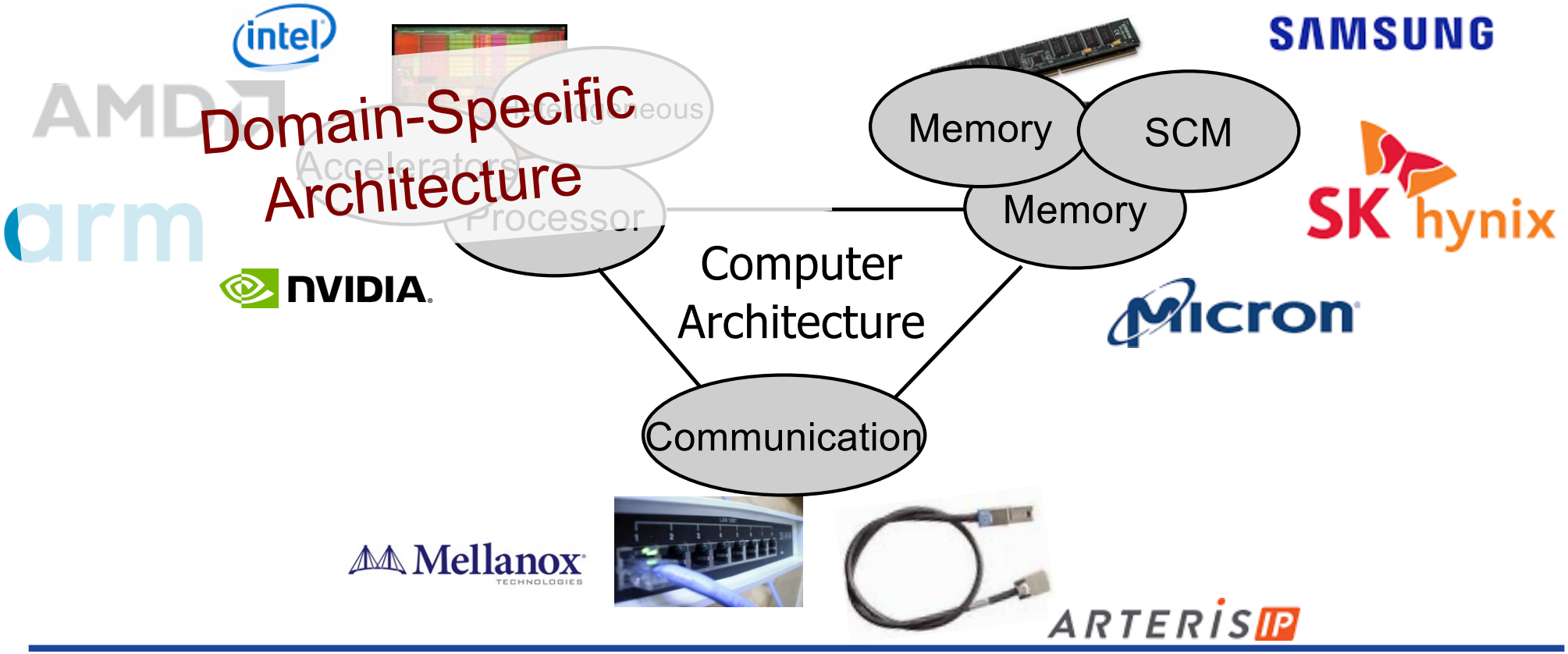
Domain-Specific Hardware/Software Co-Design, Enhanced Security, Open Instruc[illegible] Sets, and Agile Chip Development

John Hennessy and David Patterson

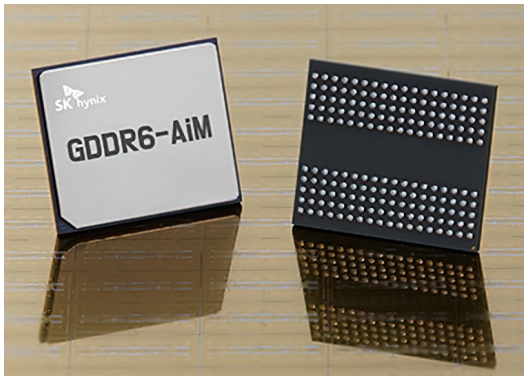
June 4, 2018

HW-centric
- Only path left is *Domain Specific Architectures*
- Just do a few tasks, but extremely well

Computer Systems & Architecture



Domain-Specific Memory

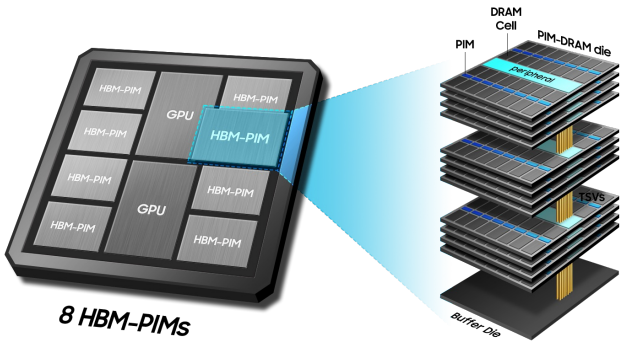


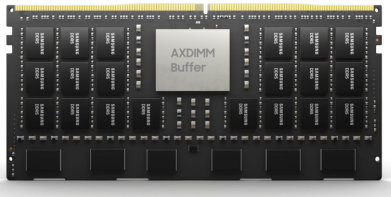
SK Hynix AiM

Processing in Memory	
Memory-centric Computing with SK Hynix's Domain-Specific Memory	Yongkee Kwon, SK Hynix
Samsung AI-cluster system with HBM-PIM and CXL-based Processing-near-Memory for transformer-based LLMs	Jin Hyun Kim, Samsung

Hot Chips 2023 Program

Samsung HBM-PIM

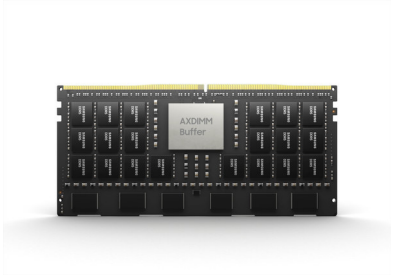




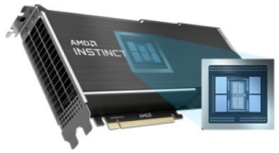
Samsung AXDIMM

Domain-Specific Memory

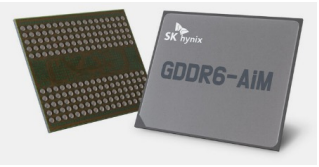
UPMEM PIM-DIMM



AxDIMM
(Samsung)



HBM-PIM
(Samsung)



GDDR6-AiM
(SK hynix)

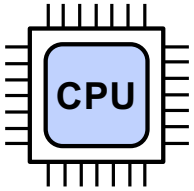
Flexibility → Efficiency

Domain-Specific *Networks*

General-Purpose Networks

Ethernet

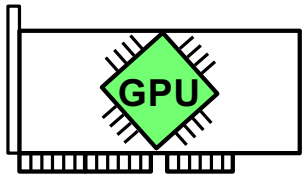
Infiniband (IB)



Cray/HPE Slingshot

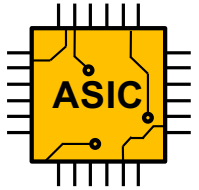
IBM BlueGene/Q

NVIDIA NVLink/NVSwitch



Domain-Specific Networks

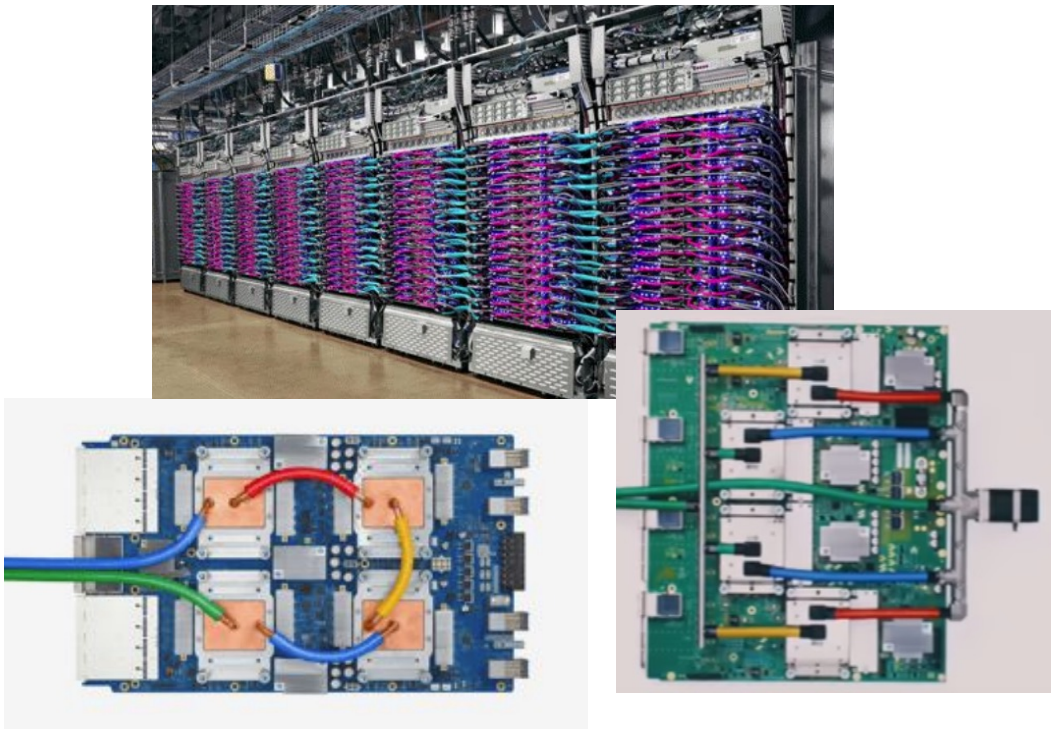
Groq Scale-out TSP
Google TPU
DE Shaw Anton
Tesla Dojo



Flexibility ⟷ Efficiency

[Abts & Kim Hotl'23, IEEE Micro'24]

Domain-Specific Supercomputer/Systems



DOI:10.1145/3360307

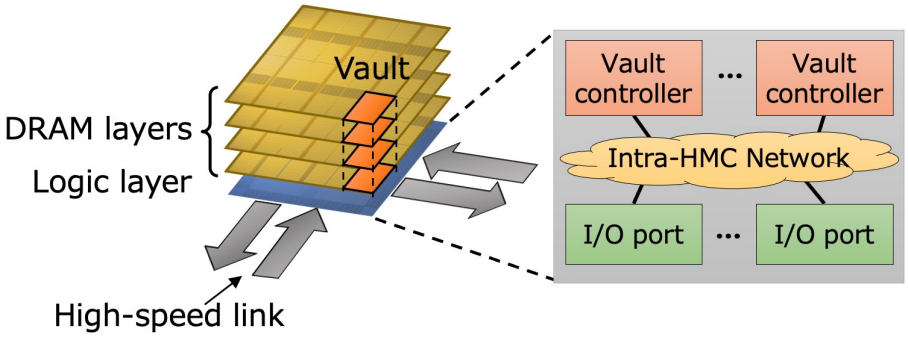
Google's TPU supercomputers train deep neural networks 50x faster than general-purpose supercomputers running a high-performance computing benchmark.

BY NORMAN P. JOUPPI, DOE HYUN YOON, GEORGE KURIAN, SHENG LI, NISHANT PATIL, JAMES LAUDON, CLIFF YOUNG, AND DAVID PATTERSON

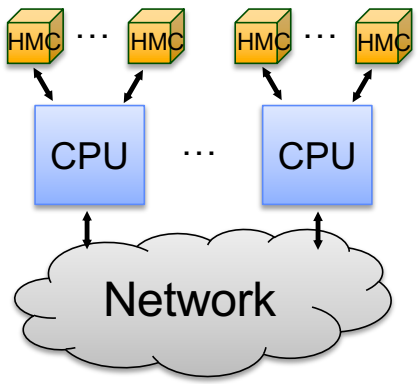
A Domain-Specific Supercomputer for Training Deep Neural Networks

How can you minimize data movement in domain-specific systems/architectures?

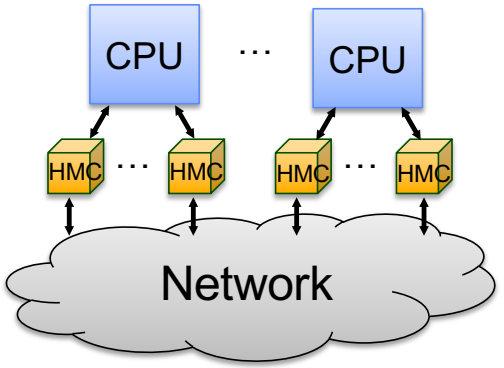
Memory-Centric Network or Memory-Centric Computing?



Hybrid-Memory Cube (HMC)



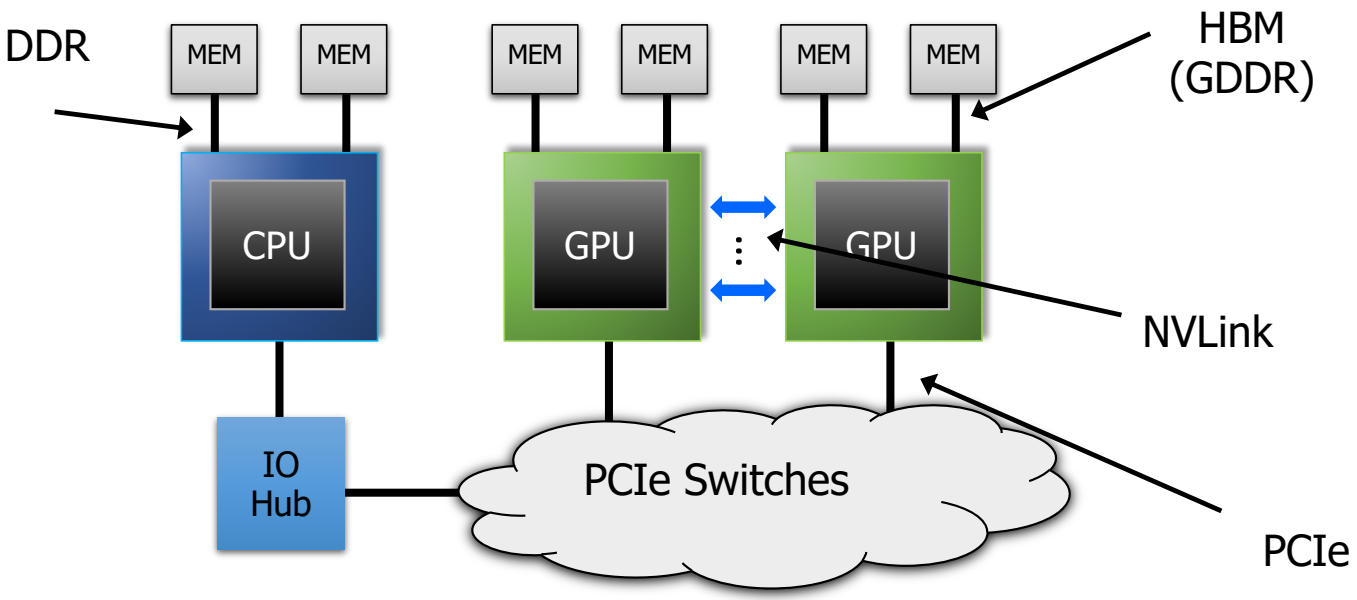
Processor-centric Network



Memory-centric Network

[Kim et al PACT'13]

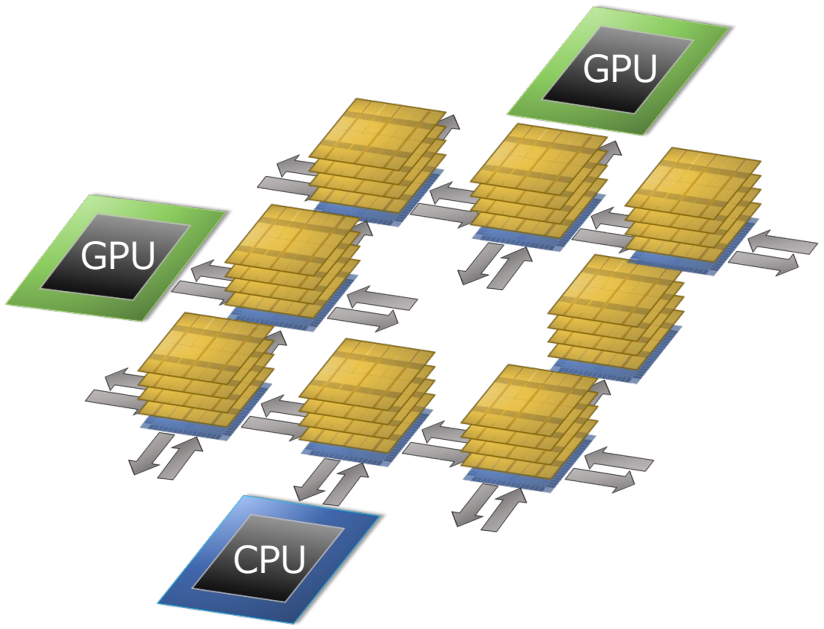
Many interfaces in Multi-GPU system



- Multiple interconnect interfaces
- Multiple steps of data movement:
  CPU Mem -> CPU -> PCIe->GPU->GPU Mem
- Can we try to minimize the number of interfaces?
- Can we minimize the amount of data movement?

[Kim et al. PACT'13, MICRO'14]

Memory-centric Network



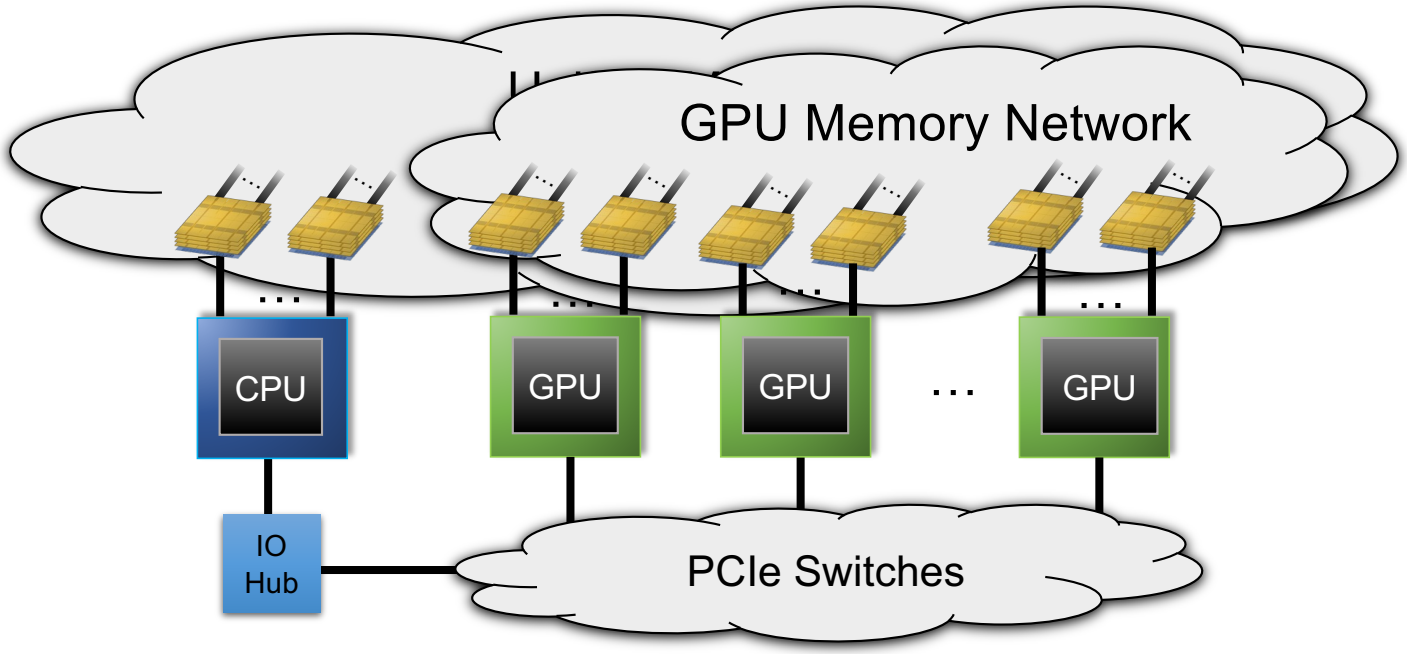
- *Unified memory* system that is shared between CPU and GPU
- No data copy between CPU and GPU
- Memory-centric organization

- Challenges?
  - Potentially longer latency to "remote" memory ➔ migration/NUMA
  - Is there such a standard? ➔ CXL perhaps?

[Kim et al MICRO'14]

Memory-Centric Network in Heterogeneous Systems

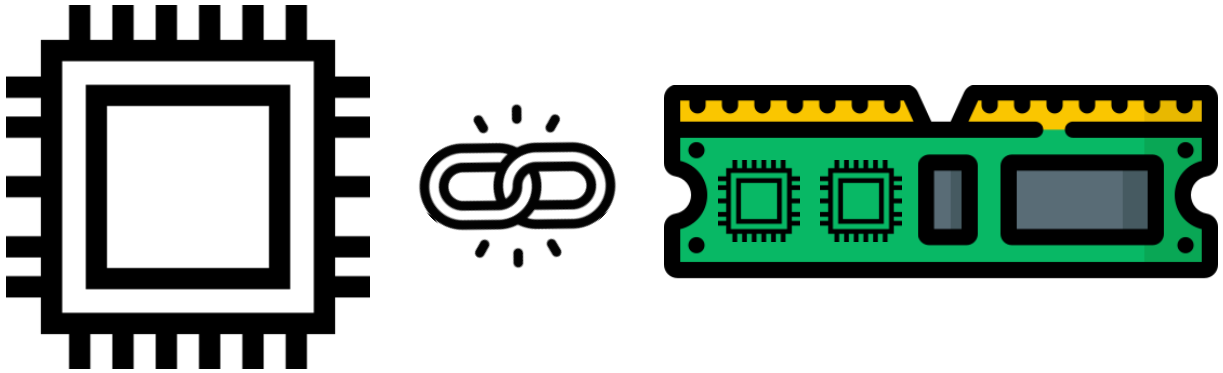
- 3 types of networks
  - GPU memory network
  - CPU memory network
  - CPU-GPU network
- Unified memory network that is shared by all compute endpoints



[Kim et al MICRO'14]

Domain-Specific *System* Architecture

- Domain-specific accelerator (NPU) + domain-specific memory (PIM) for end-to-end LLM inference
- NPU : Matrix-Matrix Multiplications
- PIM : Matrix-Vector Multiplications



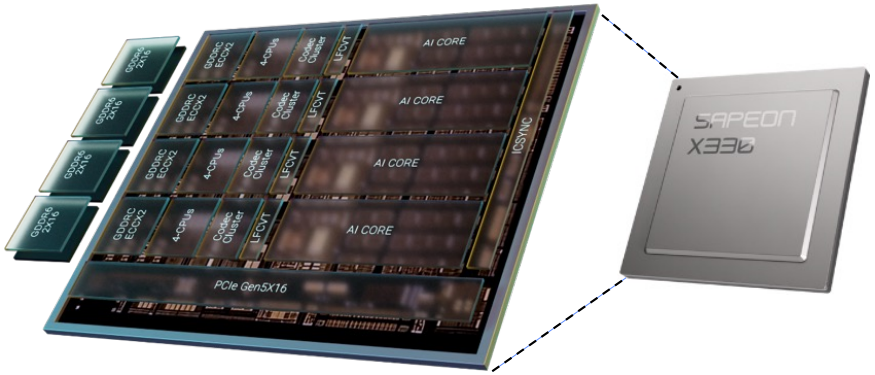
Domain-specific accelerator

Domain-specific memory

[Seo et al ASPLOS'24]

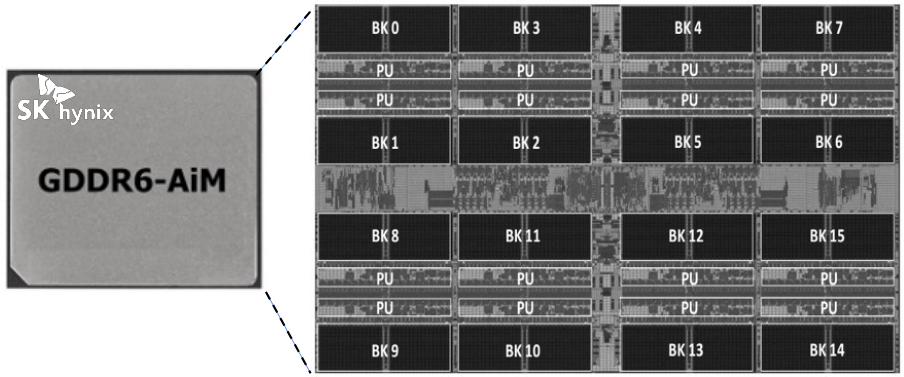
Domain-Specific System Architecture

SAPEON's NPU (x330)



- Datacenter-targeted
- Power-cost efficient

SK Hynix's AiM

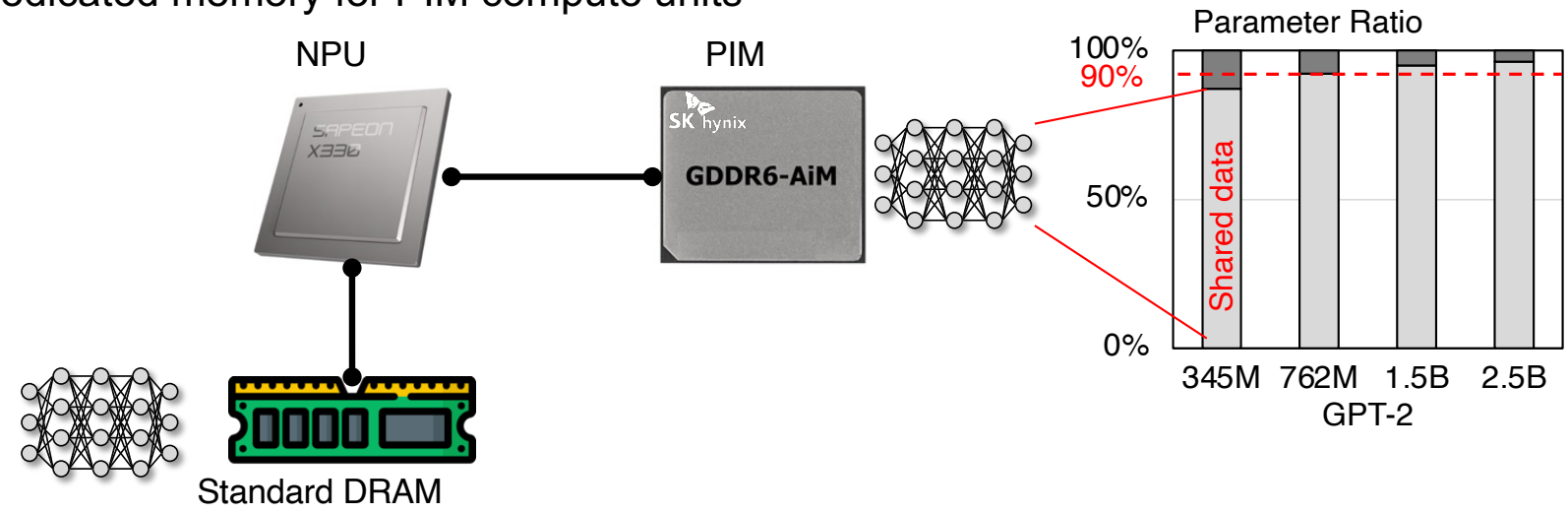


- Based on commodity DRAM
- True all-bank parallelism

[Seo et al ASPLOS'24]

Previous PIM Systems with ***Partitioned*** Memory

- Dedicated memory for host (NPU)
- Separate dedicated memory for PIM compute units



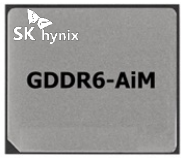
☺ Maximized parallelism between NPU and PIM

☹ Inefficient memory usage

[Seo et al ASPLOS'24]

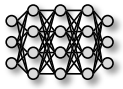
IANUS - Integrated Accelerator based on NPU-PIM

- PIM memory for main memory of both NPU and PIM compute units



IANUS

☺ Efficient memory usage



[Seo et al ASPLOS'24]

Heterogeneous system with PIM and NPU

- IANUS : Heterogeneous architecture with commercial NPU with commercial PIM.

- IANUS is an ***domain-specific syste***ms for the **end-to-end LLM inference.**
  - Match the diverse compute characteristics in LLM
  - Efficient memory usage

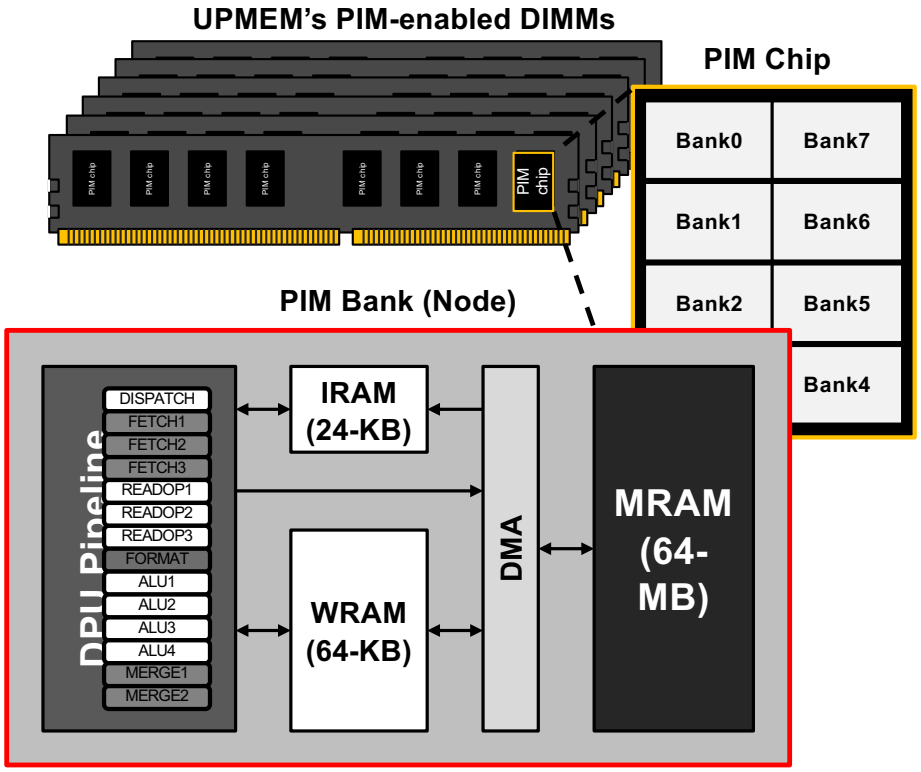
- **Unified Memory** to share memory capacity between PIM and NPU
  - Maximize (limited) memory capacity
  - Minimize (unnecessary) data movement

- **PIM Access Scheduling** effectively maximizes parallelism between NPU and PIM.
  - Workload mapping & scheduling
  - Control memory accesses to the shared (unified) memory

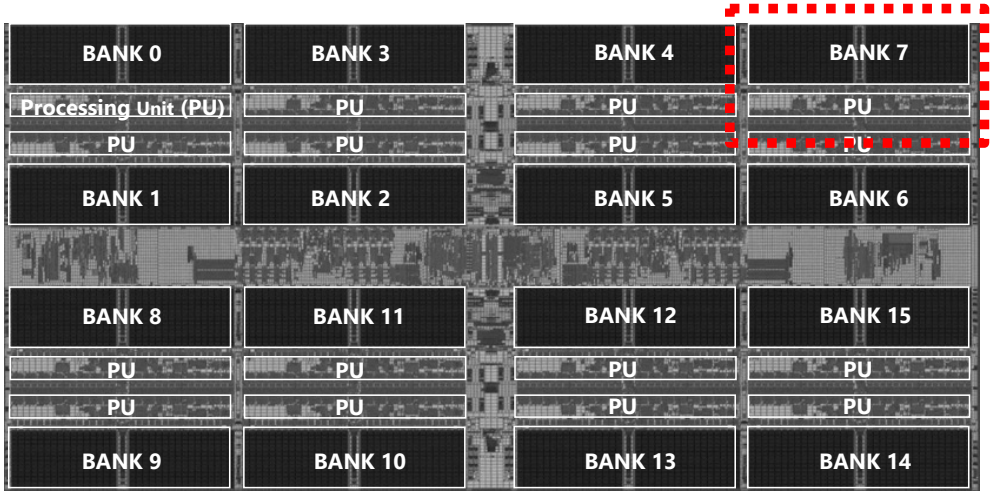
[Seo et al ASPLOS'24]

PIM locality & PIM interconnect

Processing-in-Memory (PIM) Locality

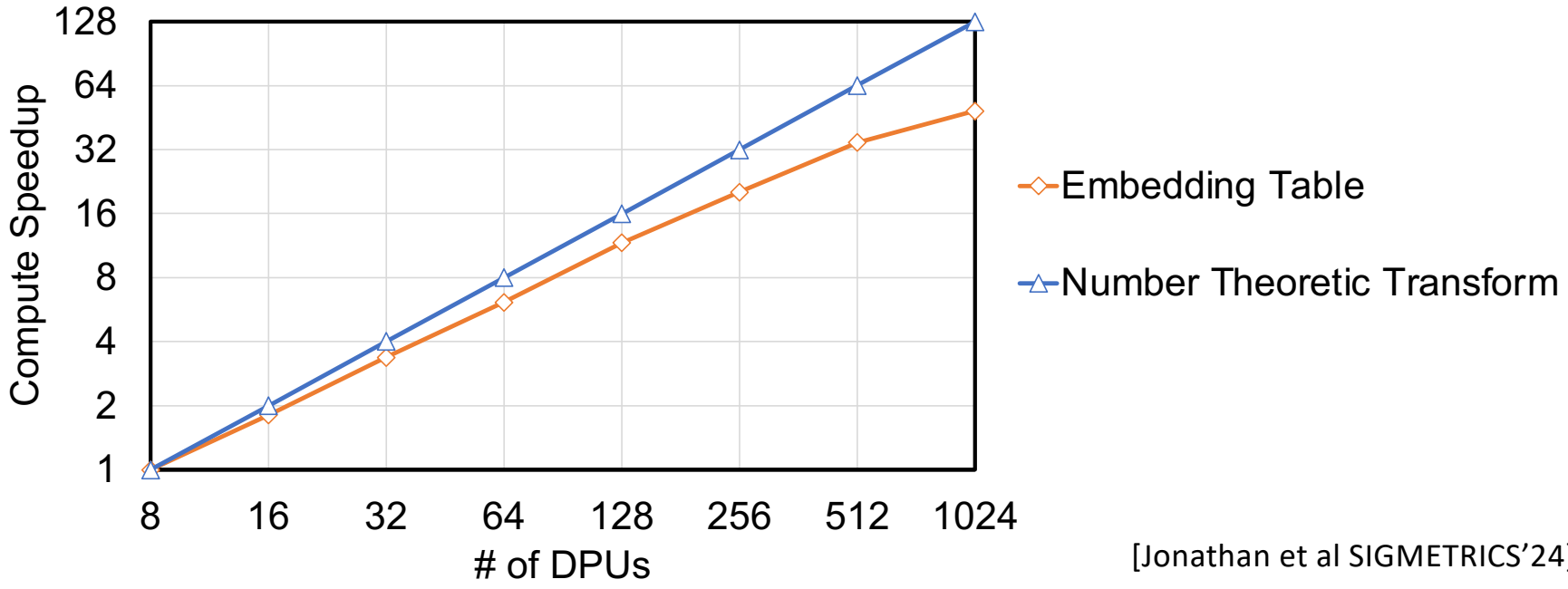


- PIM == Compute near memory
  == move compute to data
- But need PIM locality (i.e., data near compute)



[SK Hynix AiM ISSCC'23]

PIM Scalability



[Jonathan et al SIGMETRICS'24]

Processing-in-Memory provides ***compute* (kernel)** scalability performance but what about ***overall*** performance?

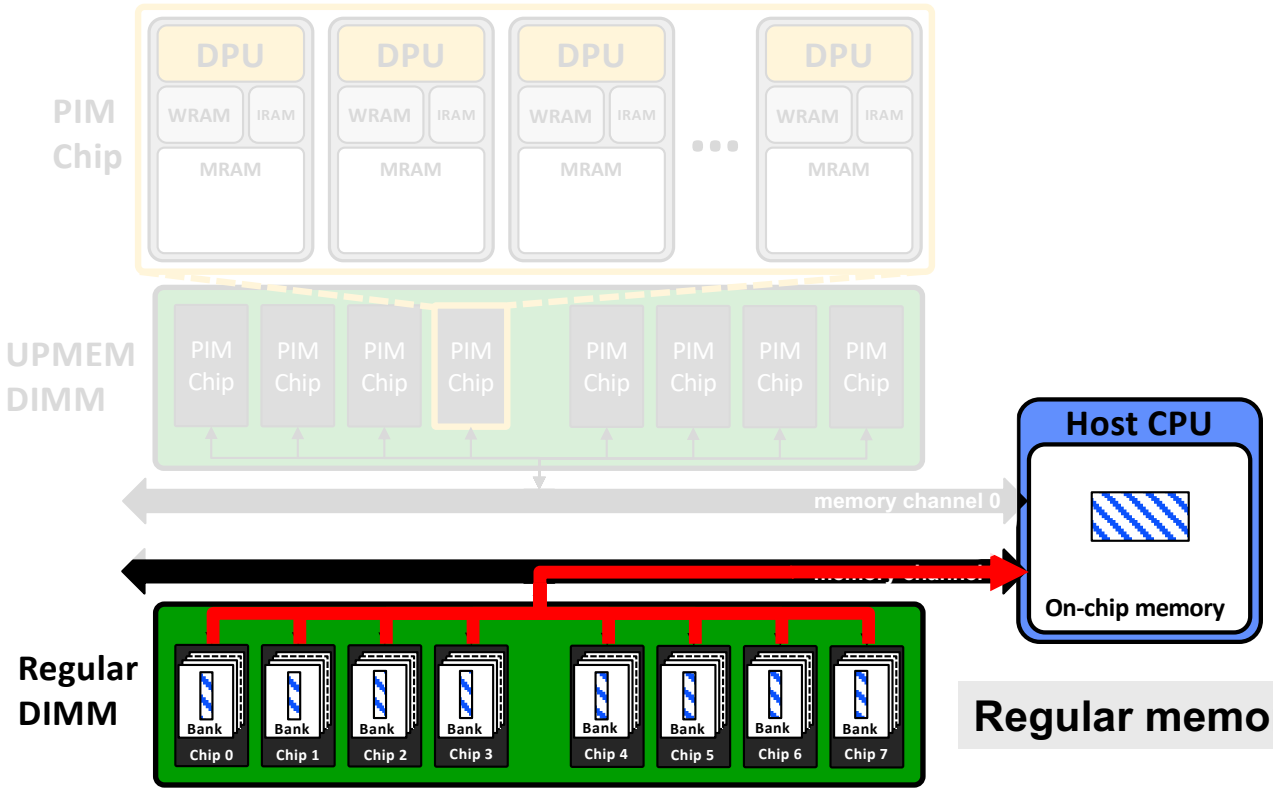
Limitations of PIM Scalability

- Data Movement
  - *Partitioned* memory system as there is a separate main memory and PIM memory
  - Data movement between regular (main) memory and PIM-memory is needed.

- PIM-to-PIM Data Movement
  - No (direct) PIM-to-PIM communication is supported in modern PIM.
  - PIM-to-PIM communication is achieved through the host CPU.

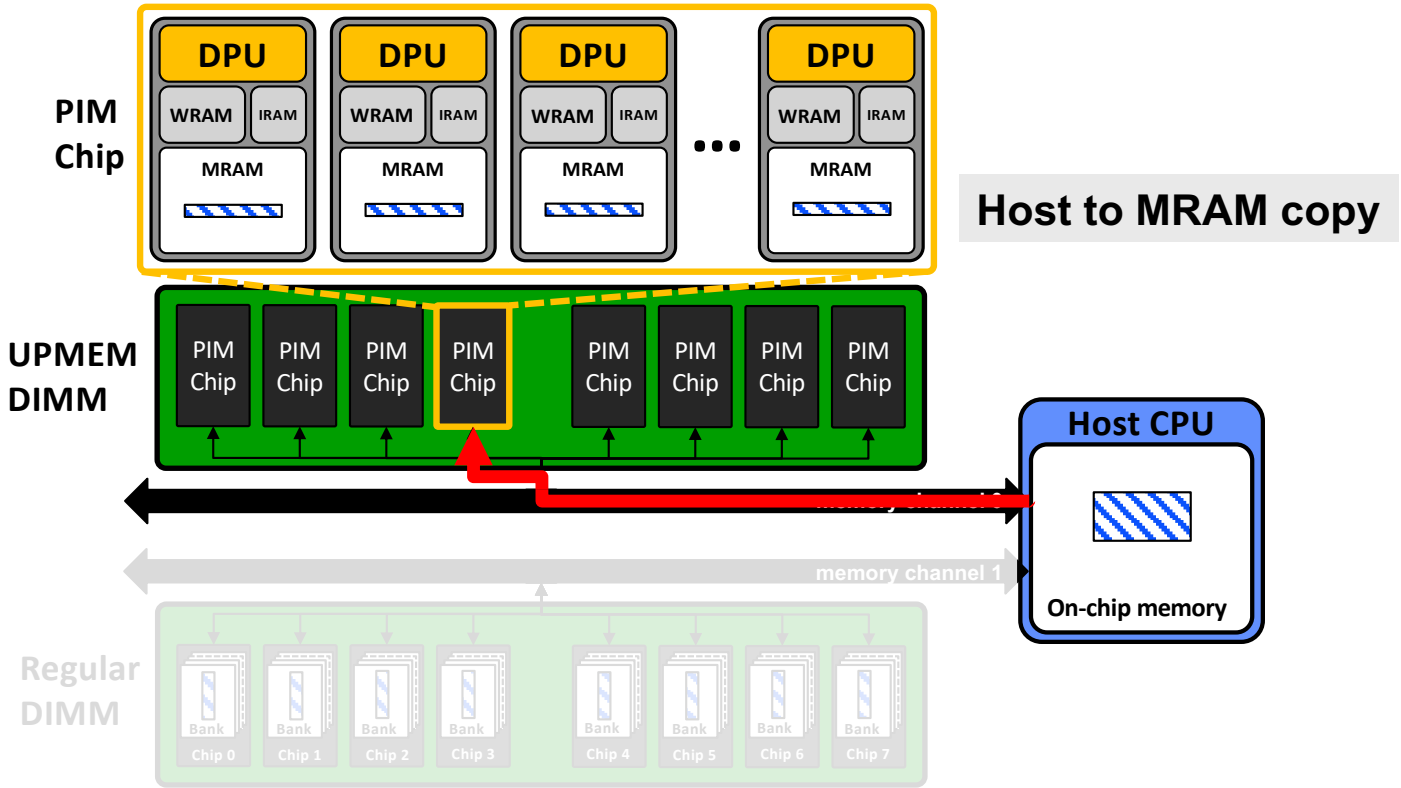
- Many prior work have identified similar problems (etc., PRiM benchmarks)

Host-to-PIM Communication



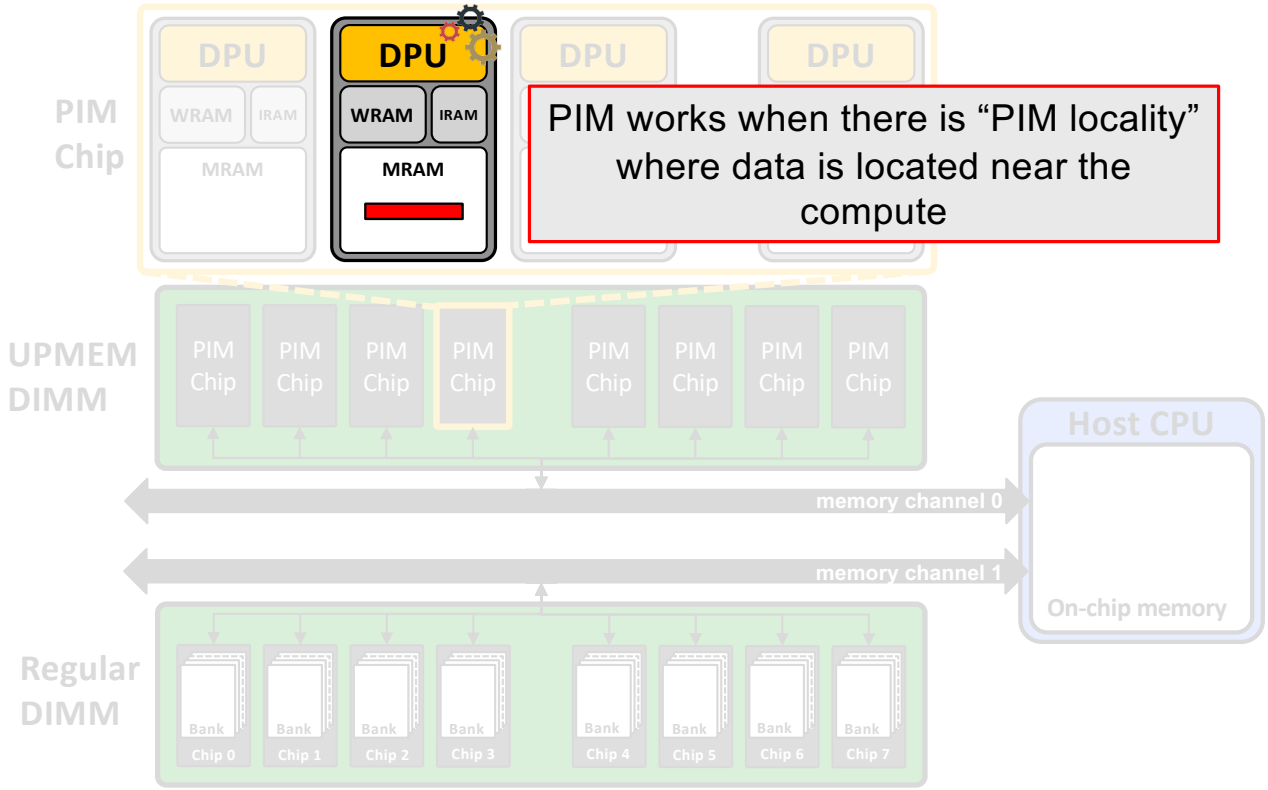
[Jonathan et al SIGMETRICS'24]

Host-to-PIM Communication



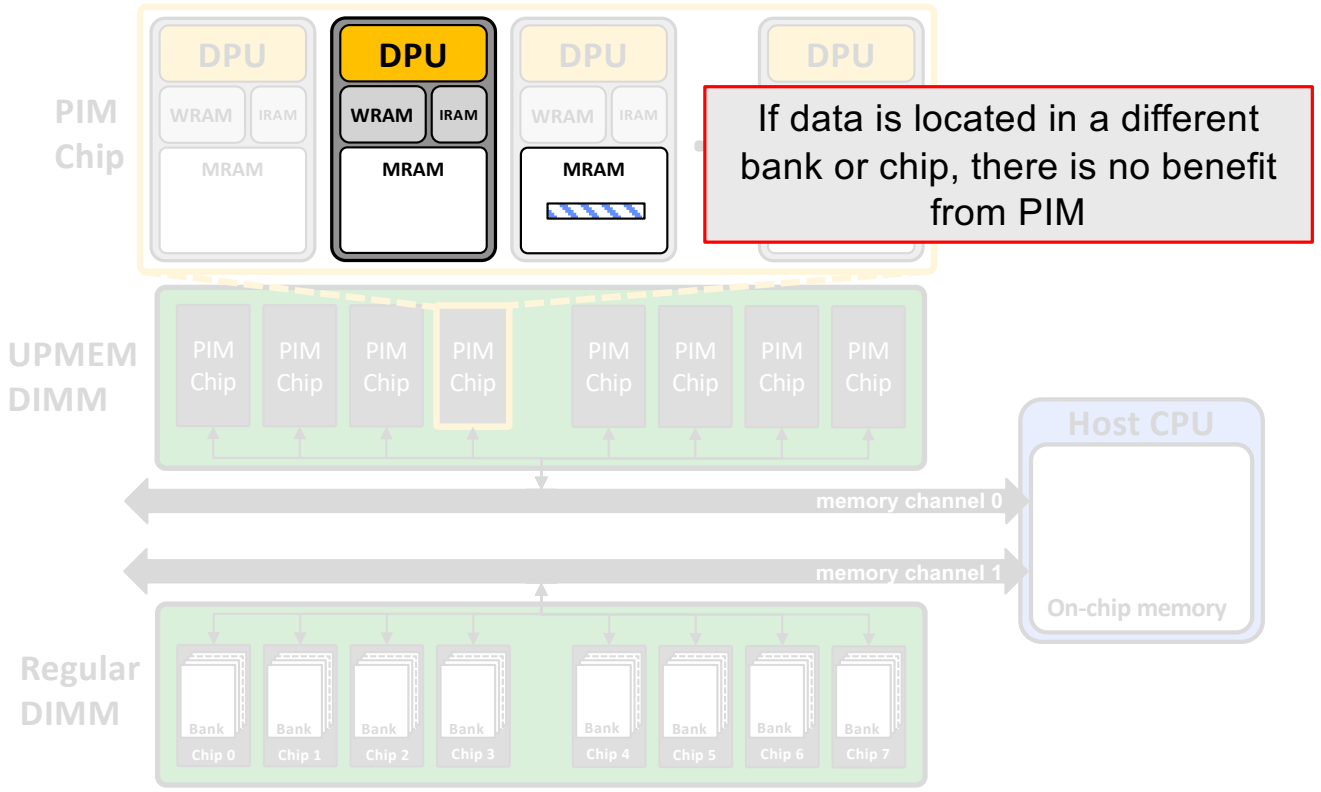
[Jonathan et al SIGMETRICS'24]

PIM-to-PIM Communication



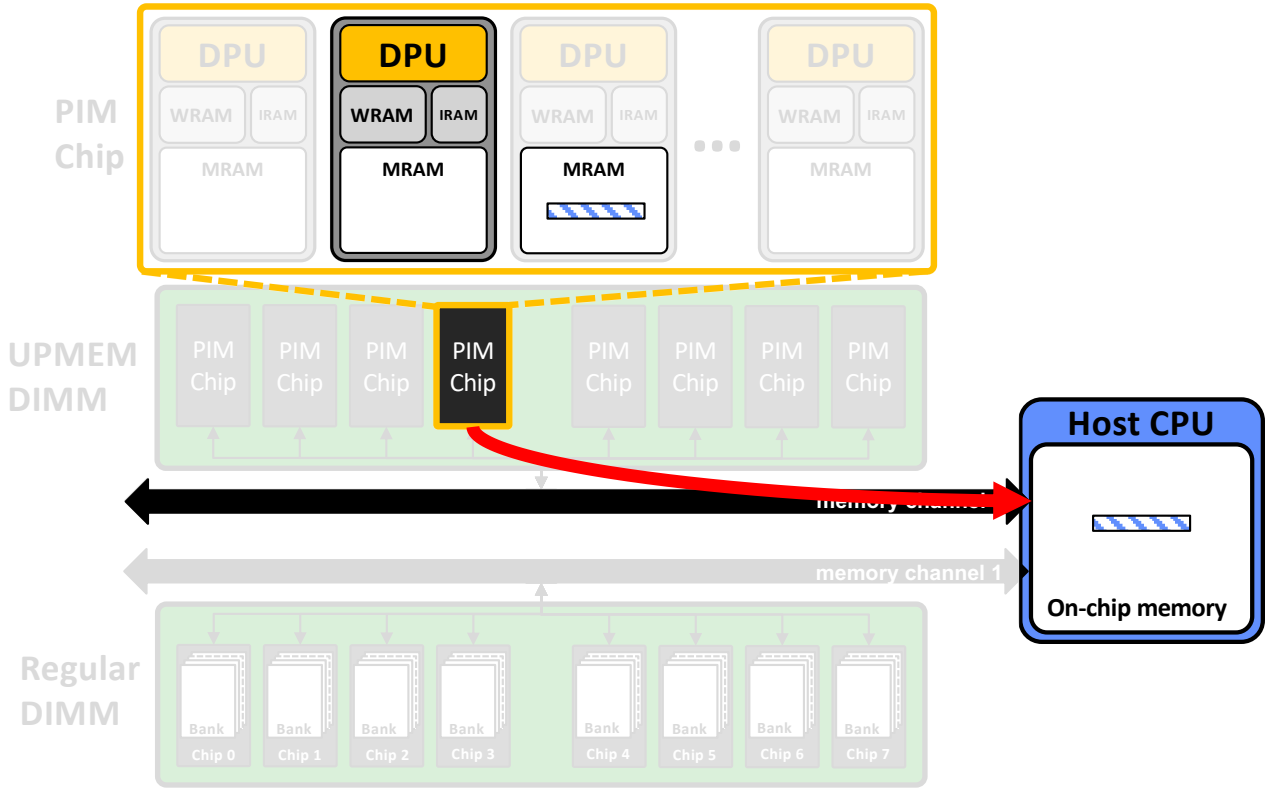
[Jonathan et al SIGMETRICS’24]

PIM-to-PIM Communication



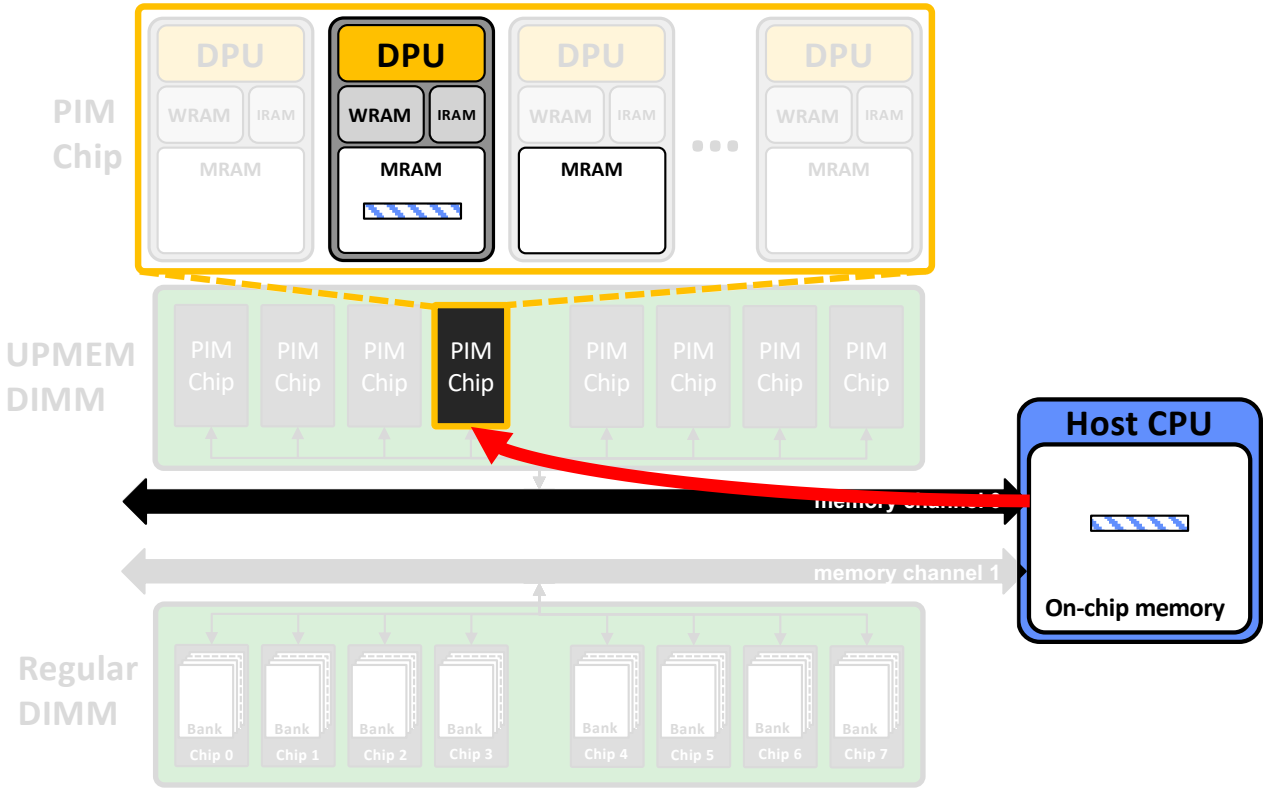
[Jonathan et al SIGMETRICS'24]

PIM-to-PIM Communication



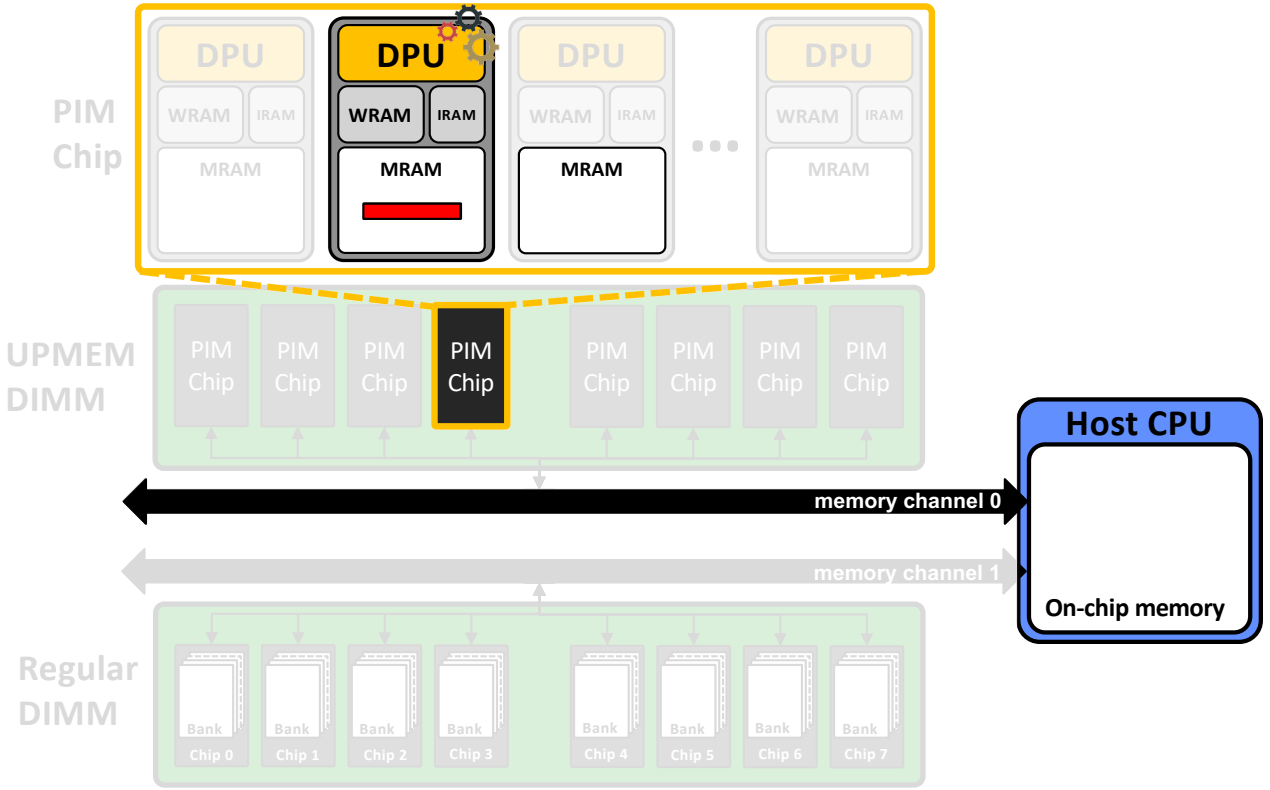
[Jonathan et al SIGMETRICS'24]

PIM-to-PIM Communication



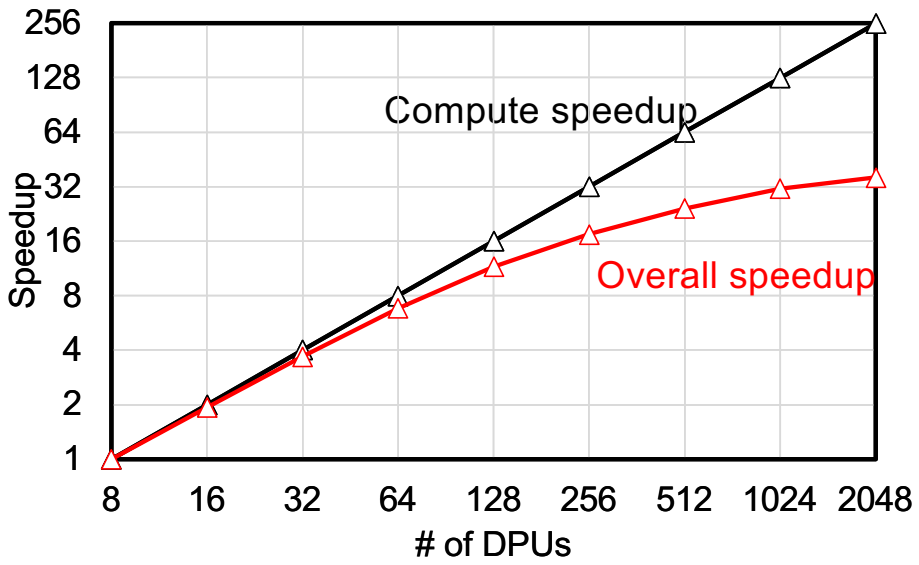
[Jonathan et al SIGMETRICS'24]

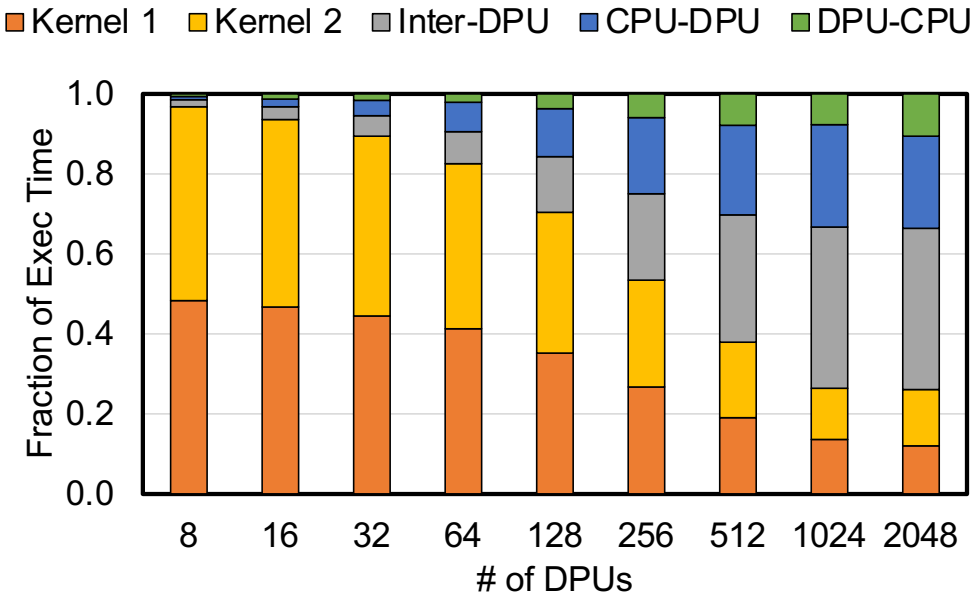
PIM-to-PIM Communication



[Jonathan et al SIGMETRICS'24]

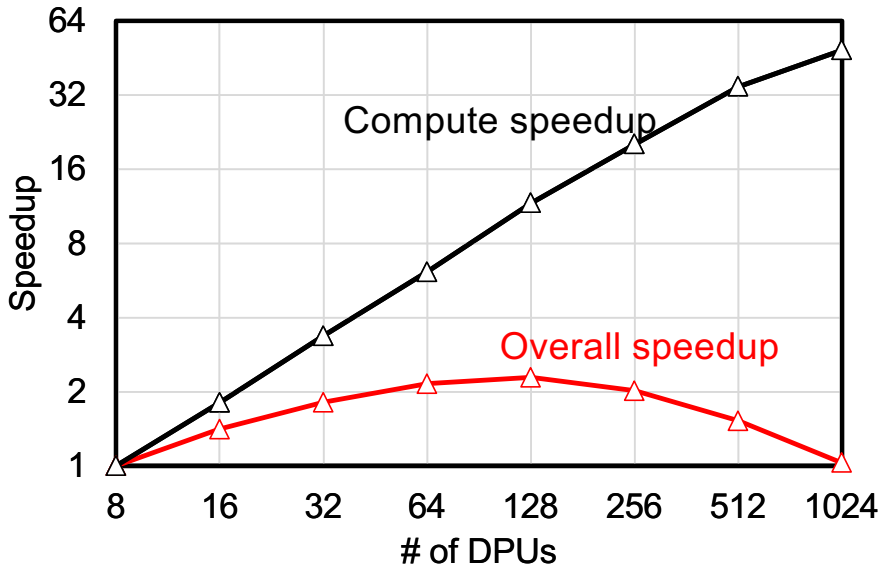
NTT Evaluations

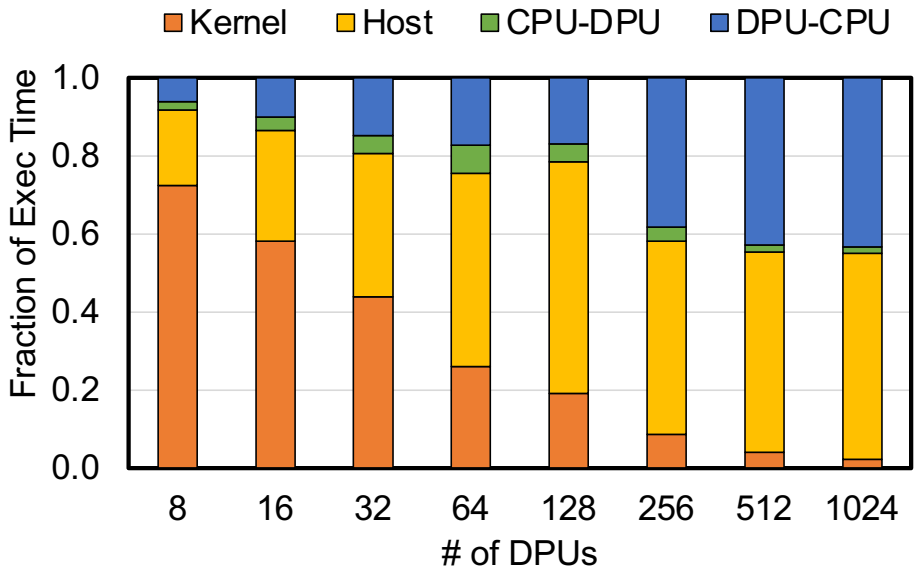




[Jonathan et al SIGMETRICS'24]

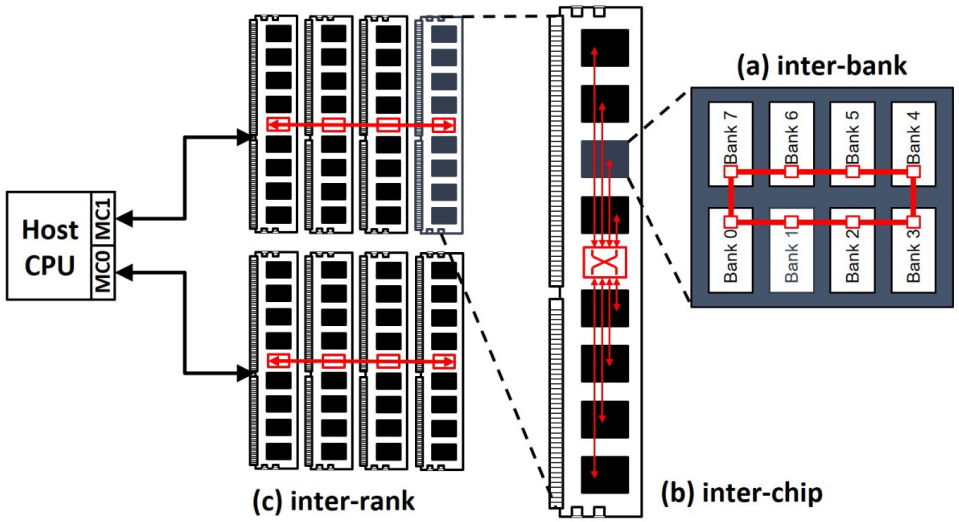
Embedding Table Evaluations



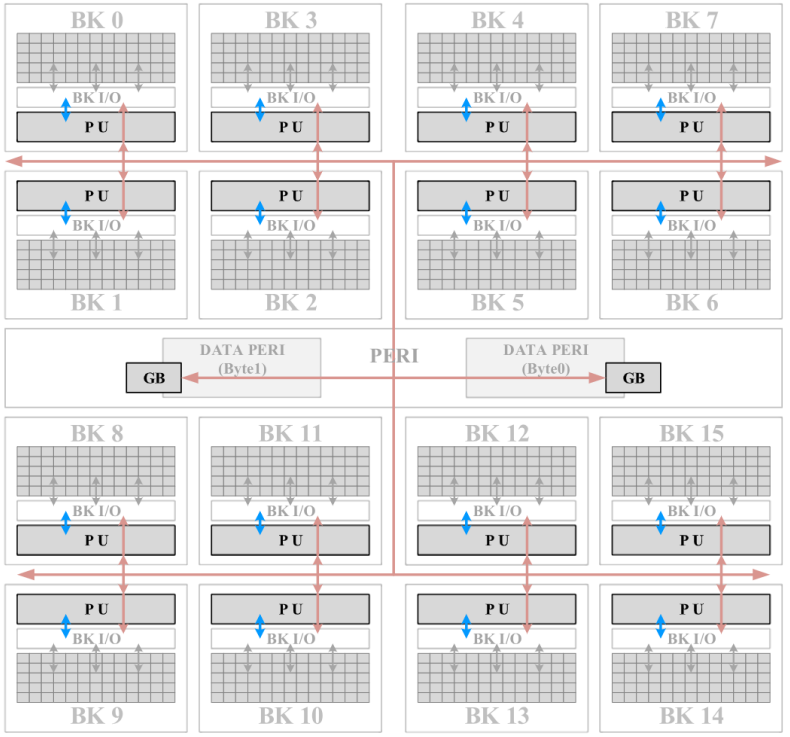


[Jonathan et al SIGMETRICS'24]

Domain-Specific PIM Interconnect



[Son et al. HPCA'24]



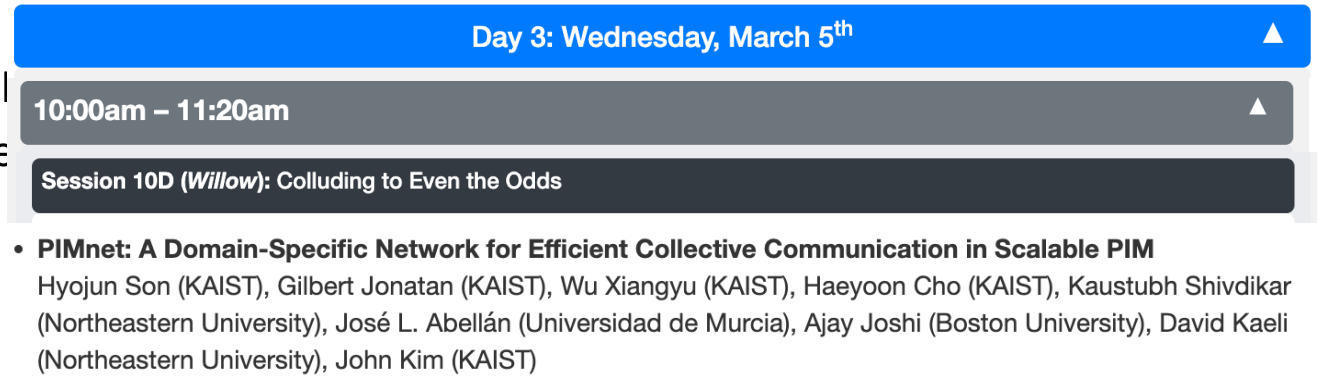
[SK hynix AiM ISSCC'22]

PIM Interconnect Challenges (Adoption of PIMnet?)

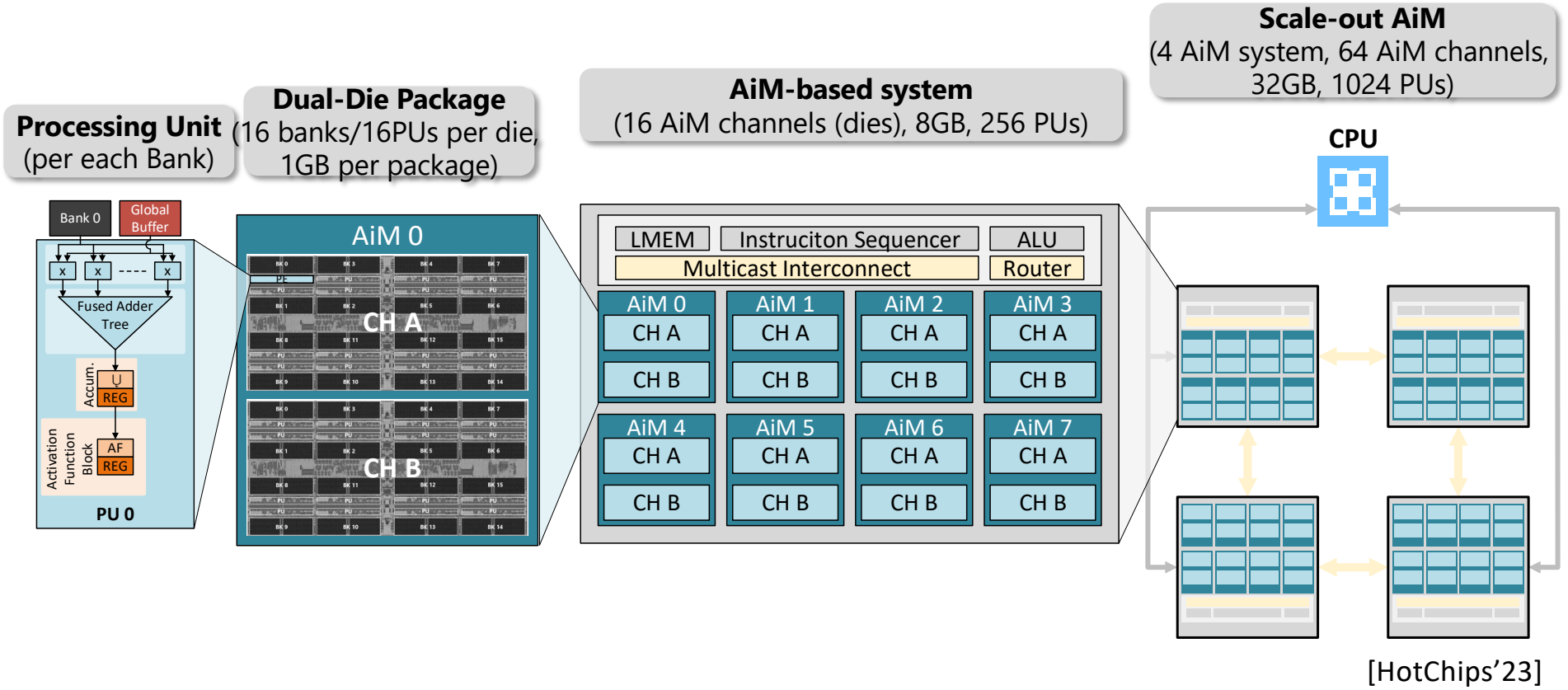
- Need both hardware and software support for PIM-to-PIM communication to enable scalability for some workloads.

- Hardware constraint
  - Packaging constraints
  - Limited amount of logic and bandwidth available

- Software challenges
  - Minimal impact on DDI
  - Programming interface



Scale-out PIM Interconnect



[HotChips'23]

Summary

- Memory-centric vs data-centric vs communication-centric
  ➔ It is all about data movement or data movement that is exposed

- Domain-specific architecture are (obviously) important in memory-centric computing but it is not only processor but also other components of the system (memory, network)

- Domain-specific *system* architectures presents new opportunities; minimize data movement through *unified* memory organization.

- Dedicated interconnect or domain-specific networks are necessary to enable in memory-centric computing for
  - Scale-up PIM interconnect
  - Scale-out PIM interconnect