

Memory-Centric Computing

Onur Mutlu

omutlu@gmail.com

<https://people.inf.ethz.ch/omutlu>

20 August 2024

Rambus

SAFARI

ETH zürich

Computing

is Bottlenecked by Data

Data is Key for AI, ML, Genomics, ...

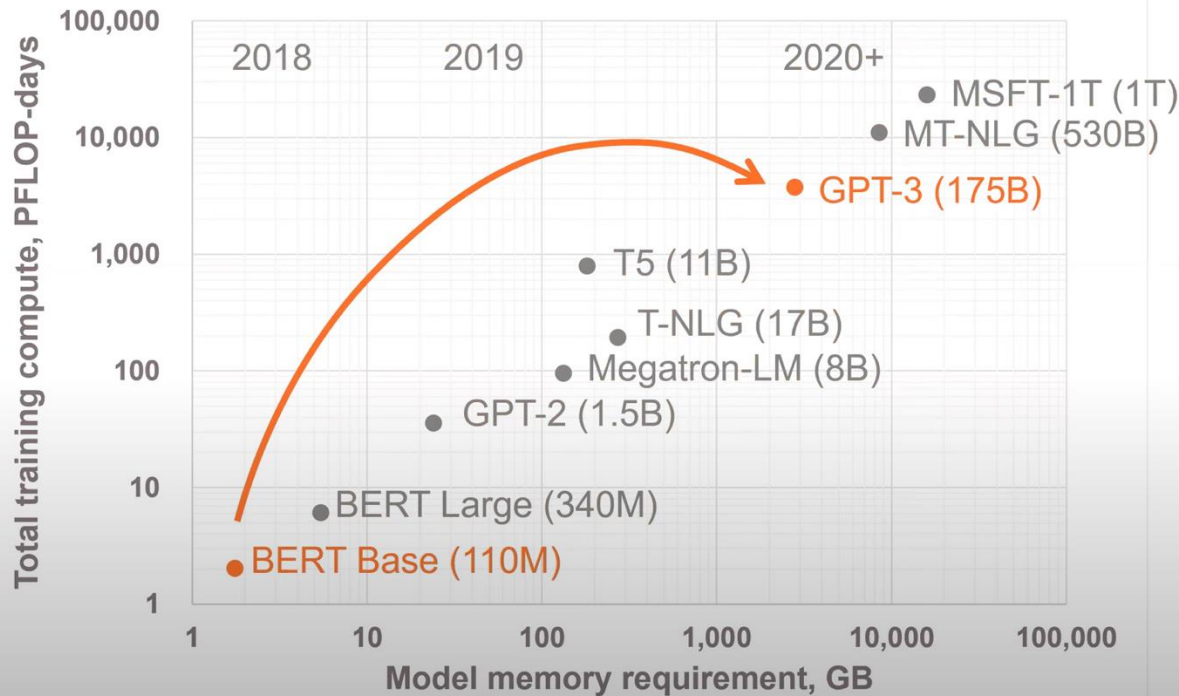
- Important workloads are all data intensive
- They require rapid and efficient processing of large amounts of data
- Data is increasing
 - We can generate more than we can process
 - We need to perform more sophisticated analyses on more data

Huge Demand for Performance & Efficiency

Exponential Growth of Neural Networks



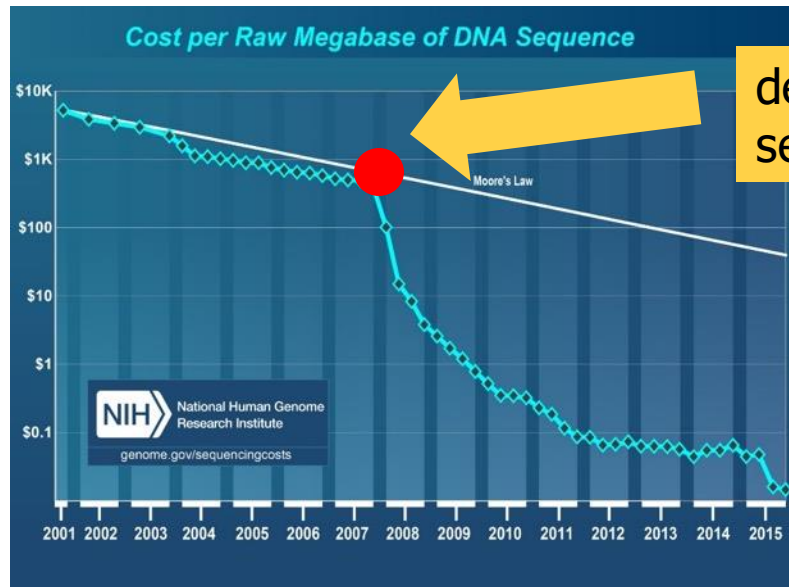
Memory and compute requirements



1800x more compute
In just **2 years**

Tomorrow, **multi-trillion**
parameter models

Huge Demand for Performance & Efficiency

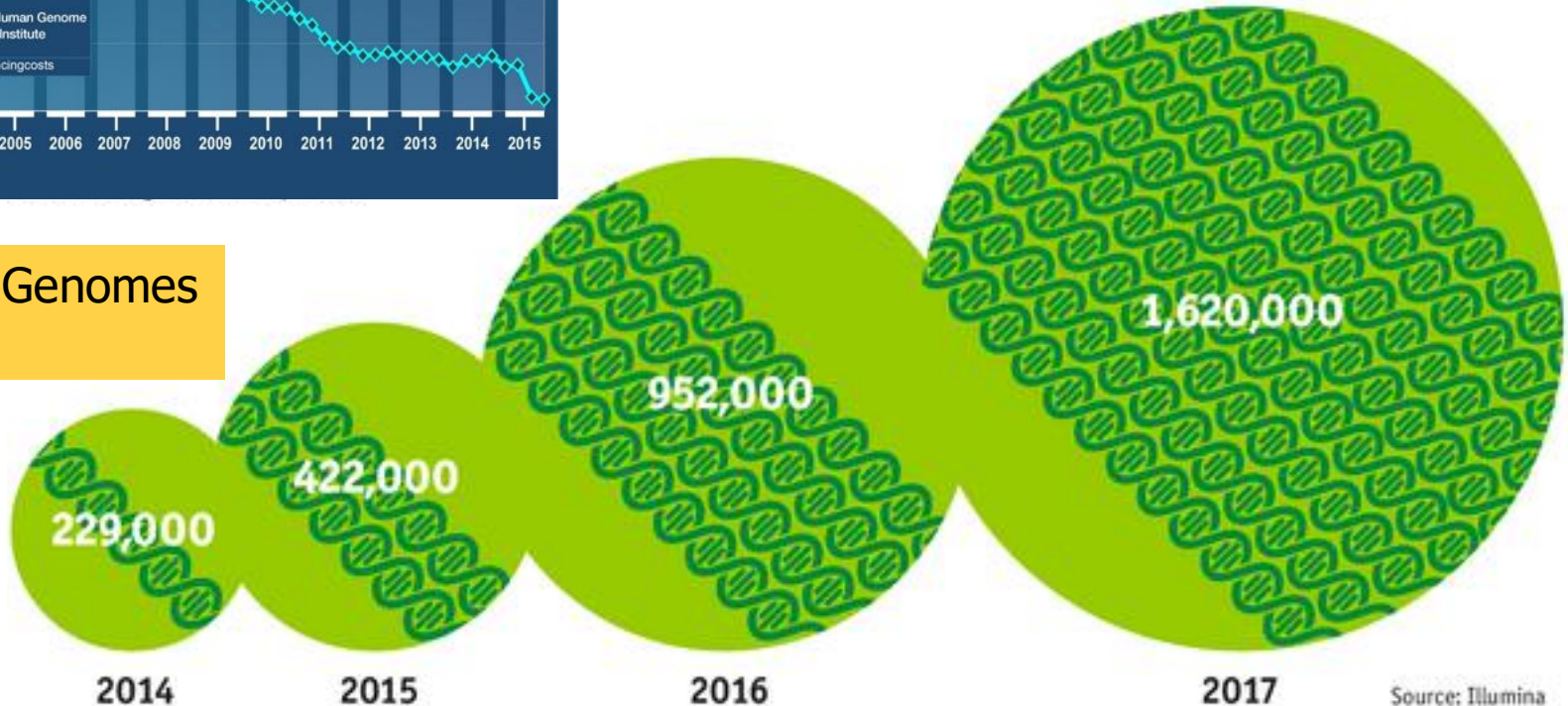


development of new sequencing technologies



Oxford Nanopore MinION

Number of Genomes Sequenced



The Economist

Do We Want This?



Or This?



High Performance,

Energy Efficient,

Sustainable

(All at the Same Time)

The Problem

Data access is the major performance and energy bottleneck

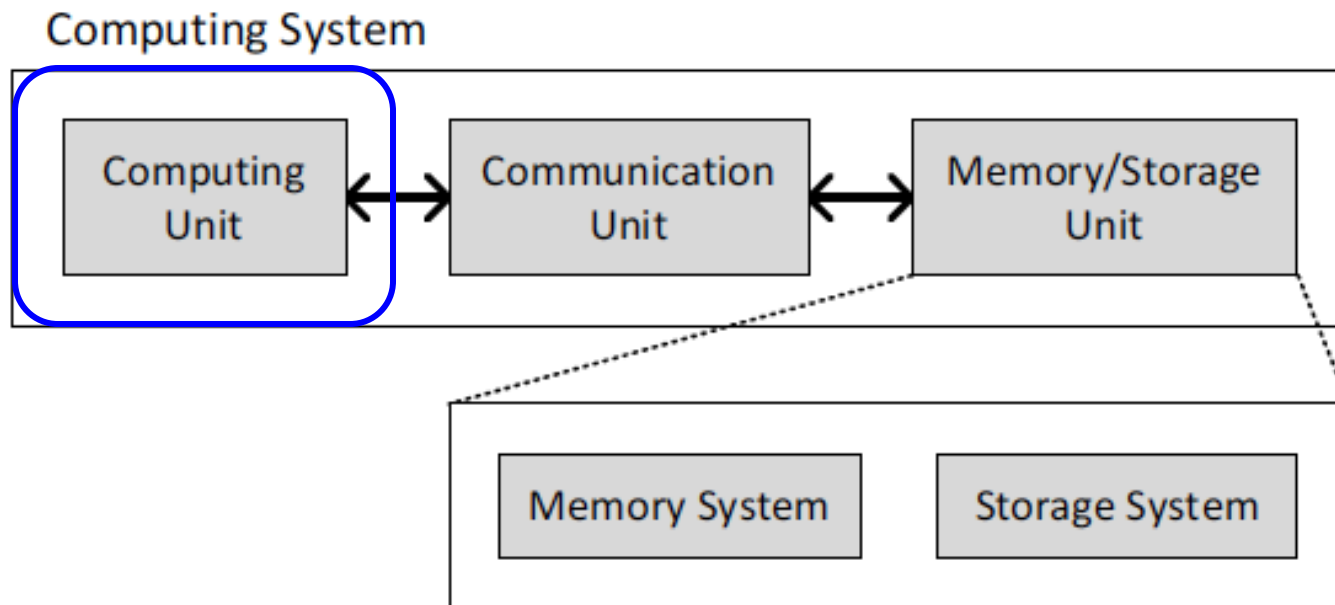
Our current
design principles
cause great energy waste
(and great performance loss)

The Problem

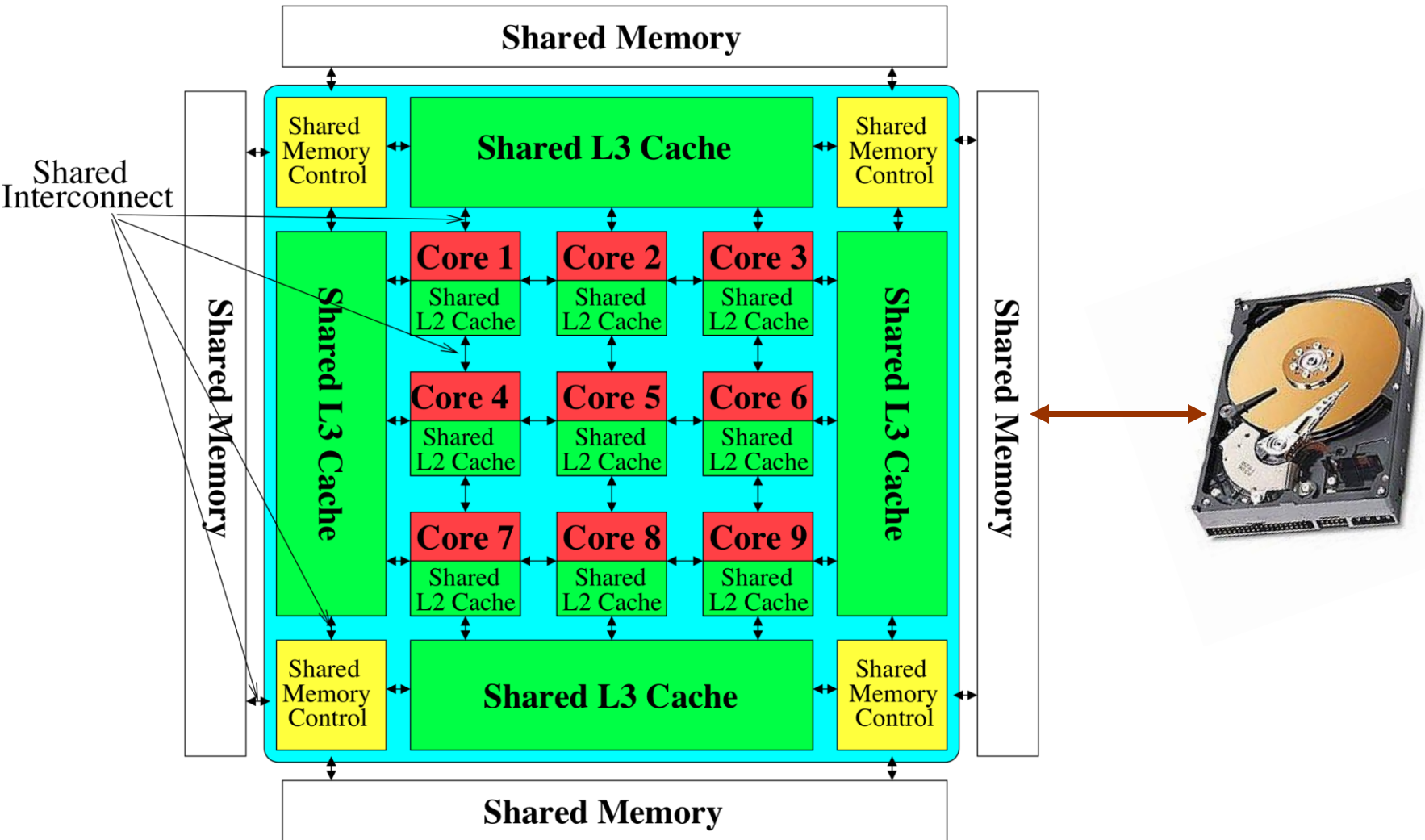
Processing of data
is performed
far away from the data

Today's Computing Systems

- Processor centric
- All data processed in the processor → at great system cost



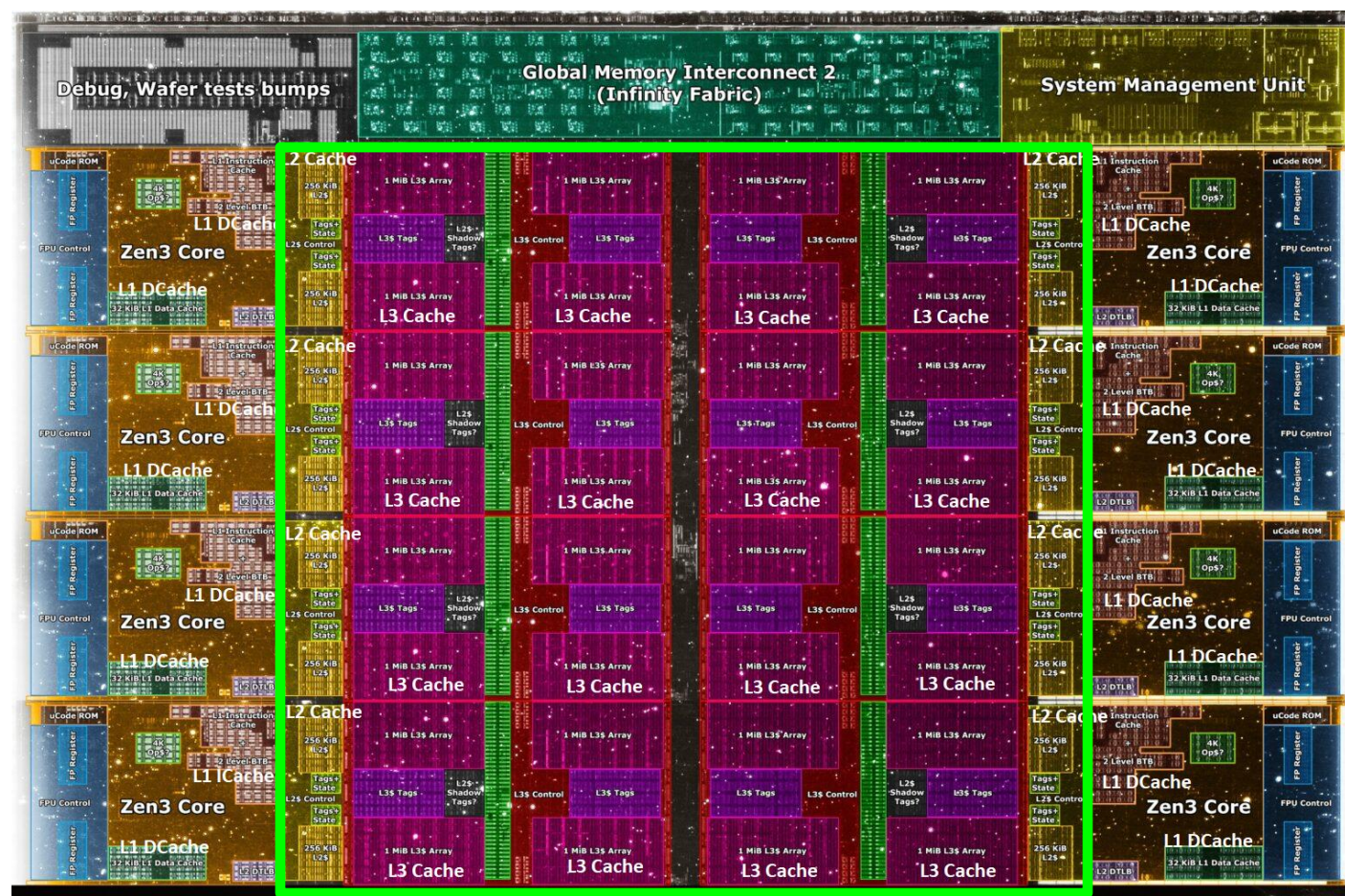
Perils of Processor-Centric Design



Most of the system is dedicated to storing and moving data

Yet, system is still bottlenecked by memory

Deeper and Larger Memory Hierarchies



Core Count:

8 cores/16 threads

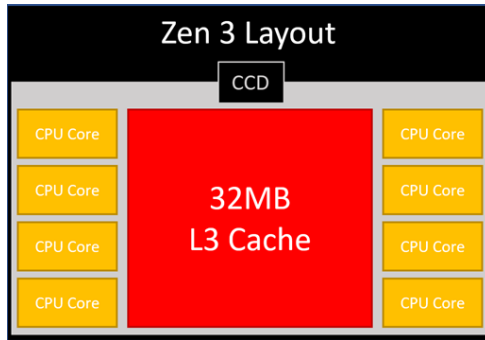
L1 Caches:
32 KB per core

L2 Caches:
512 KB per core

L3 Cache:
32 MB shared

AMD Ryzen 5000, 2020

AMD's 3D Last Level Cache (2021)

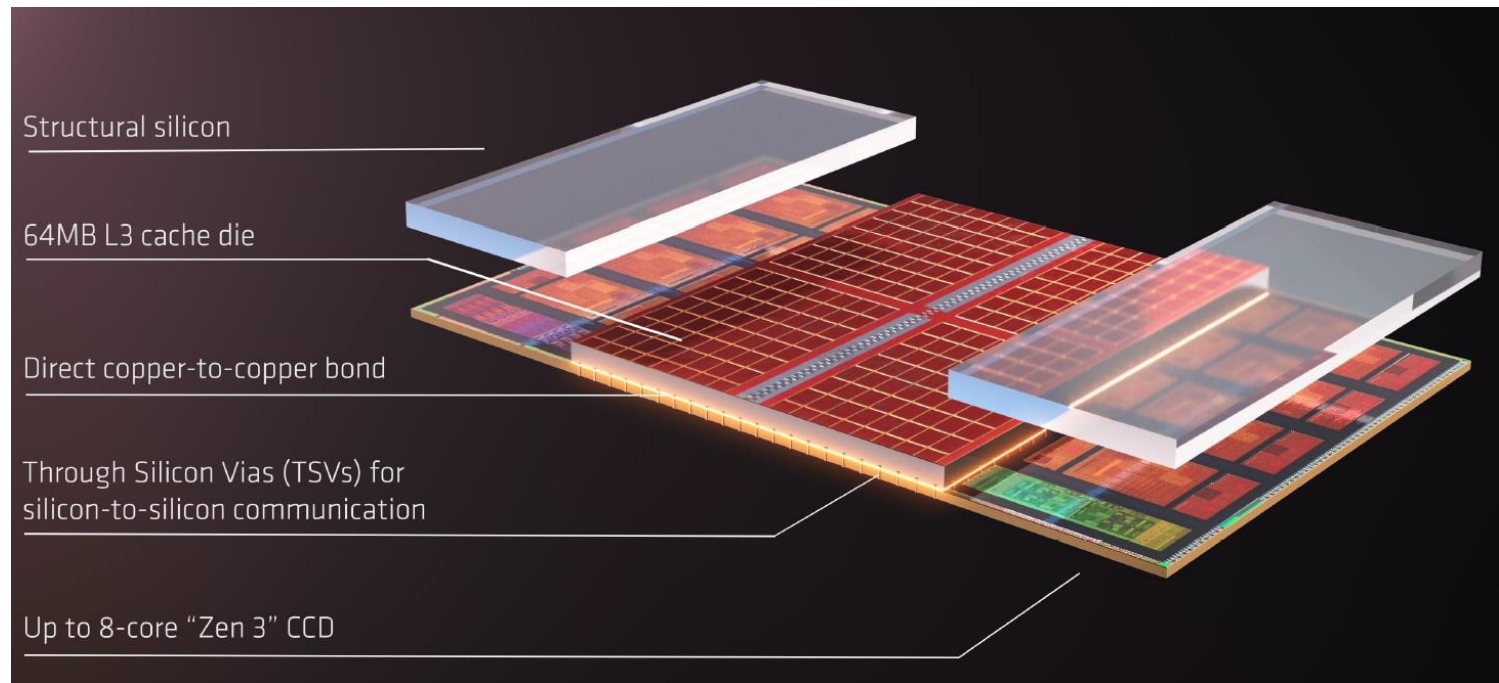


<https://community.microcenter.com/discussion/5134/comparing-zen-3-to-zen-2>

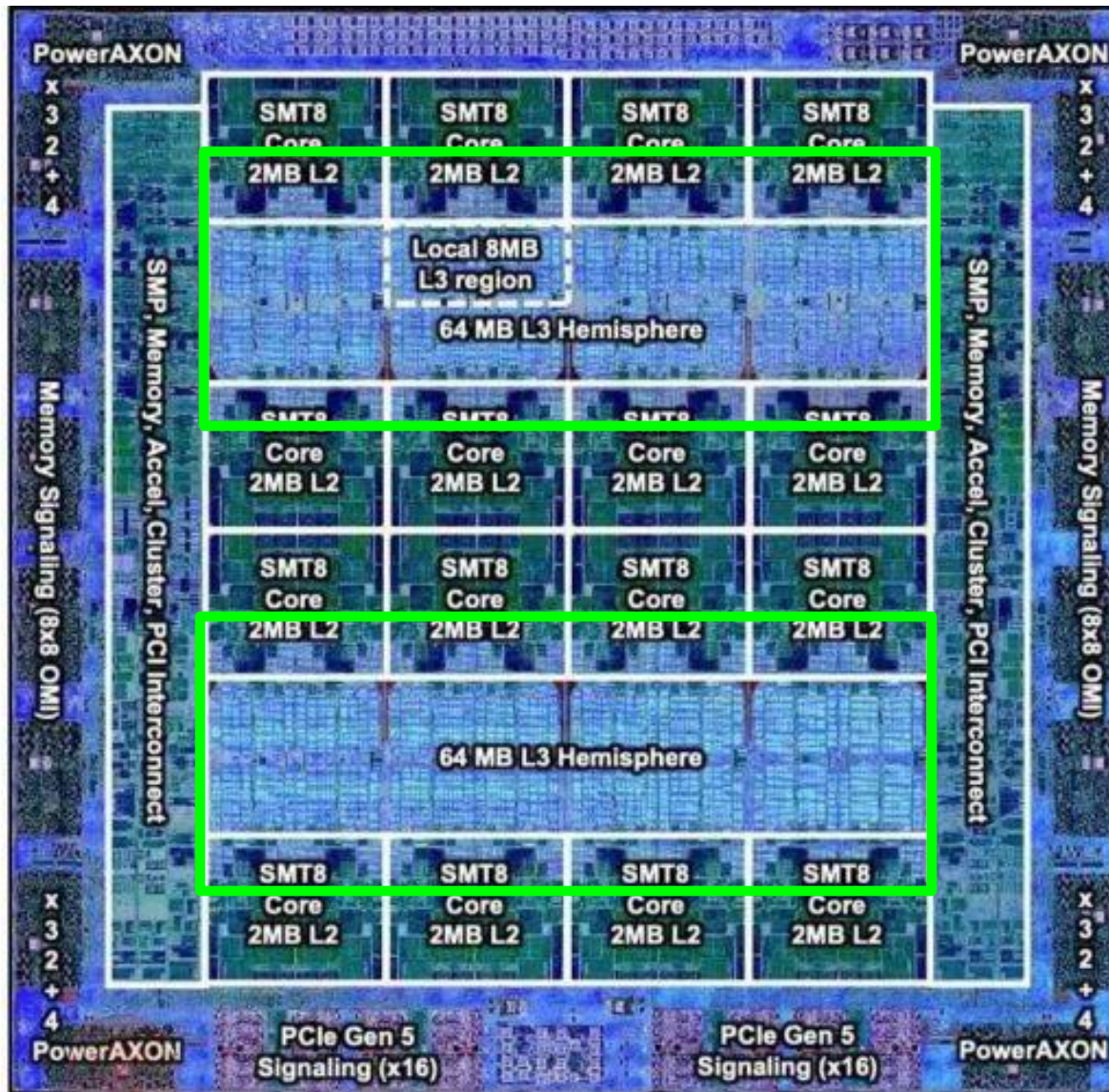
AMD increases the L3 size of their 8-core Zen 3 processors from 32 MB to 96 MB

Additional 64 MB L3 cache die
stacked on top of the processor die

- Connected using Through Silicon Vias (TSVs)
- Total of 96 MB L3 cache



Deeper and Larger Memory Hierarchies



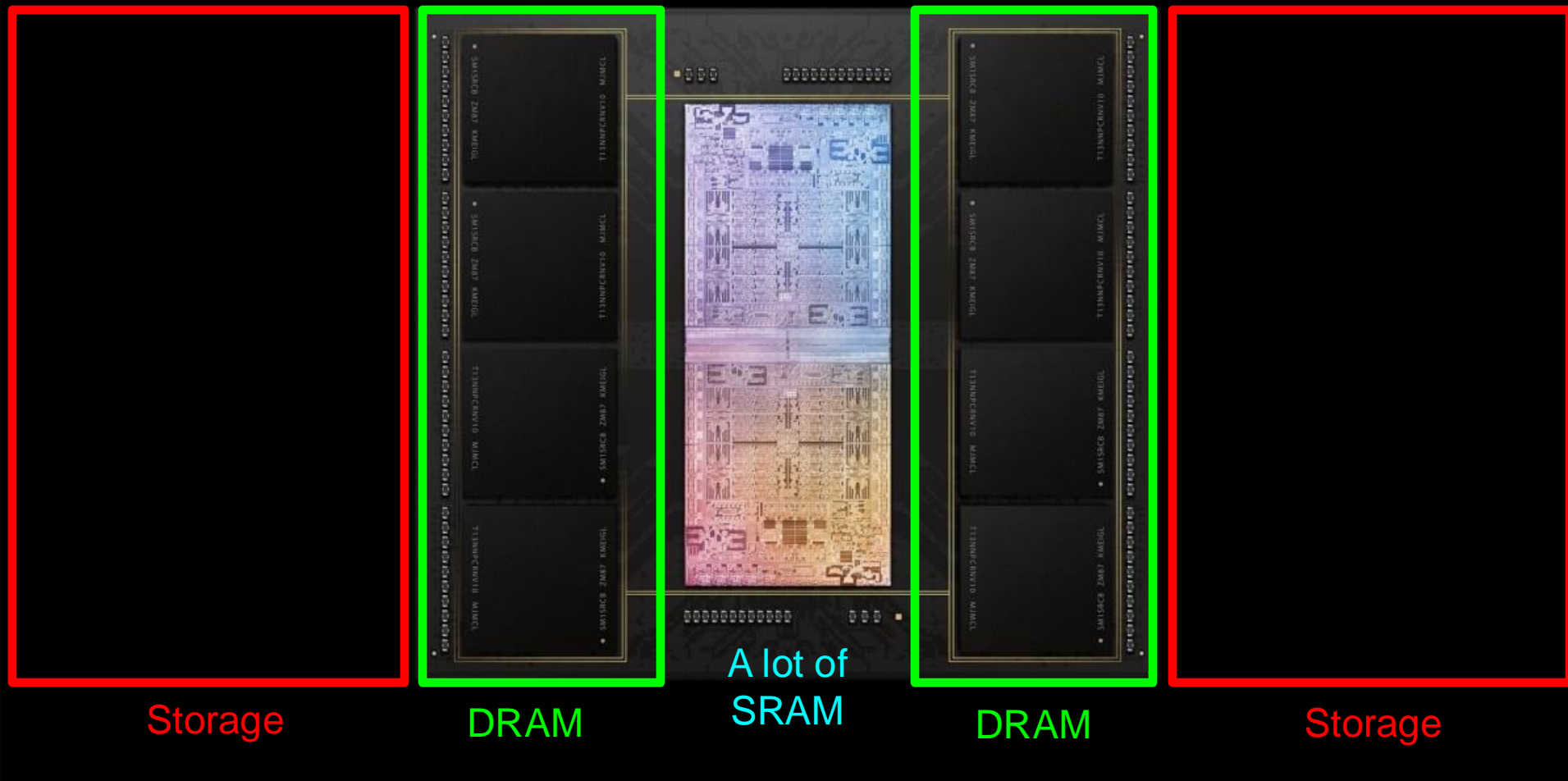
IBM POWER10,
2020

Cores:
15-16 cores,
8 threads/core

L2 Caches:
2 MB per core

L3 Cache:
120 MB shared

Deeper and Larger Memory Hierarchies



Apple M1 Ultra System (2022)

Data Overwhelms Modern Machines ...

- Storage/memory capability
- Communication capability
- Computation capability
- Greatly impacts robustness, energy, performance, cost

It's the Memory, Stupid!

- **“It's the Memory, Stupid!”** (Richard Sites, MPR, 1996)

RICHARD SITES

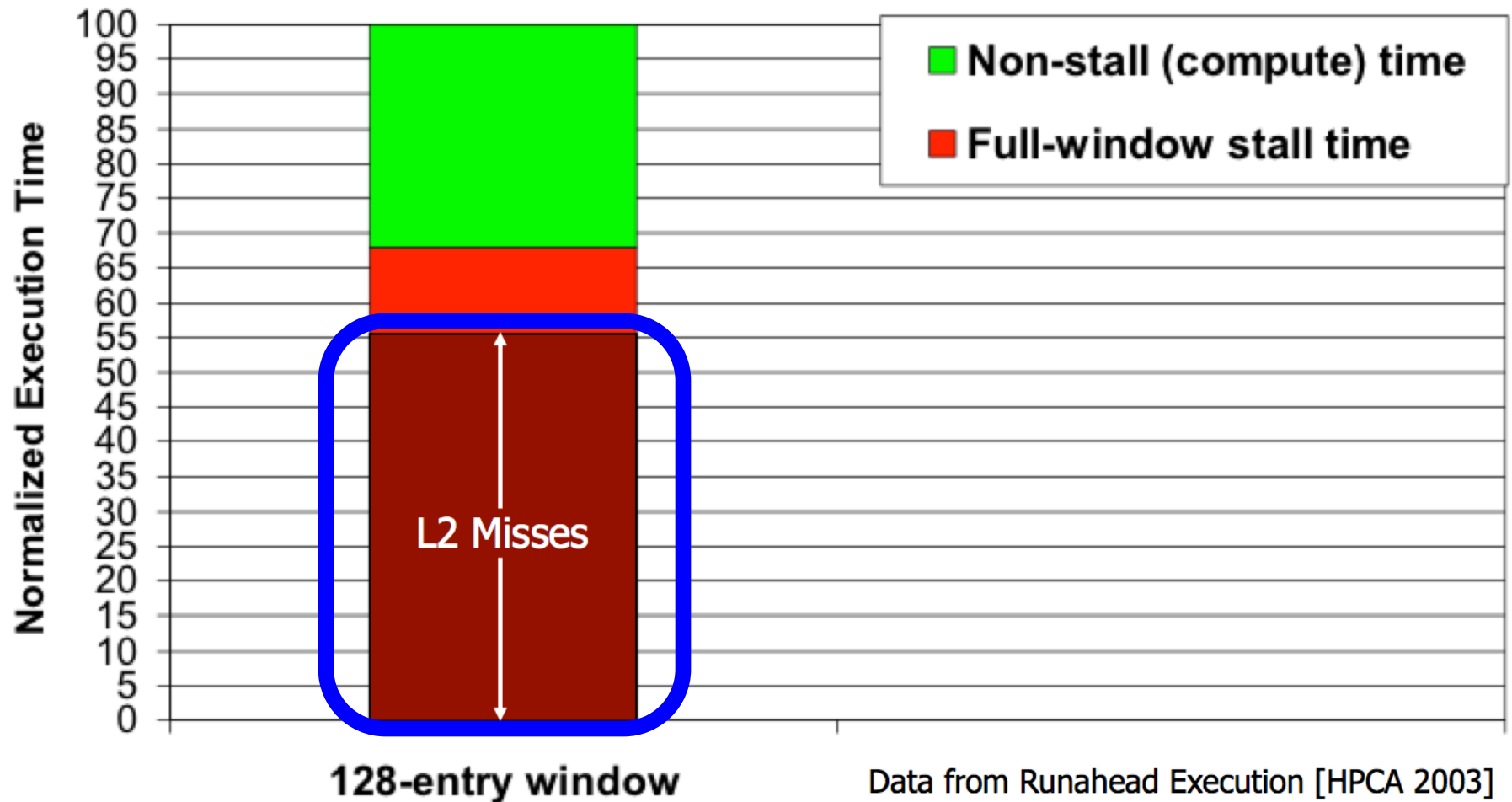
It's the Memory, Stupid!

When we started the Alpha architecture design in 1988, we estimated a 25-year lifetime and a relatively modest 32% per year compounded performance improvement of implementations over that lifetime (1,000× total). We guestimated about 10× would come from CPU clock improvement, 10× from multiple instruction issue, and 10× from multiple processors.

5, 1996  MICROPROCESSOR REPORT

I expect that over the coming decade memory subsystem design will be the *only* important design issue for microprocessors.

The Performance Perspective



The Performance Perspective

- Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt,
"Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors"

Proceedings of the 9th International Symposium on High-Performance Computer Architecture (HPCA), pages 129-140, Anaheim, CA, February 2003. [Slides \(pdf\)](#)

One of the 15 computer arch. papers of 2003 selected as Top Picks by IEEE Micro. HPCA Test of Time Award (awarded in 2021).

Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors

Onur Mutlu § Jared Stark † Chris Wilkerson ‡ Yale N. Patt §

§ECE Department

The University of Texas at Austin

{onur,patt}@ece.utexas.edu

†Microprocessor Research

Intel Labs

jared.w.stark@intel.com

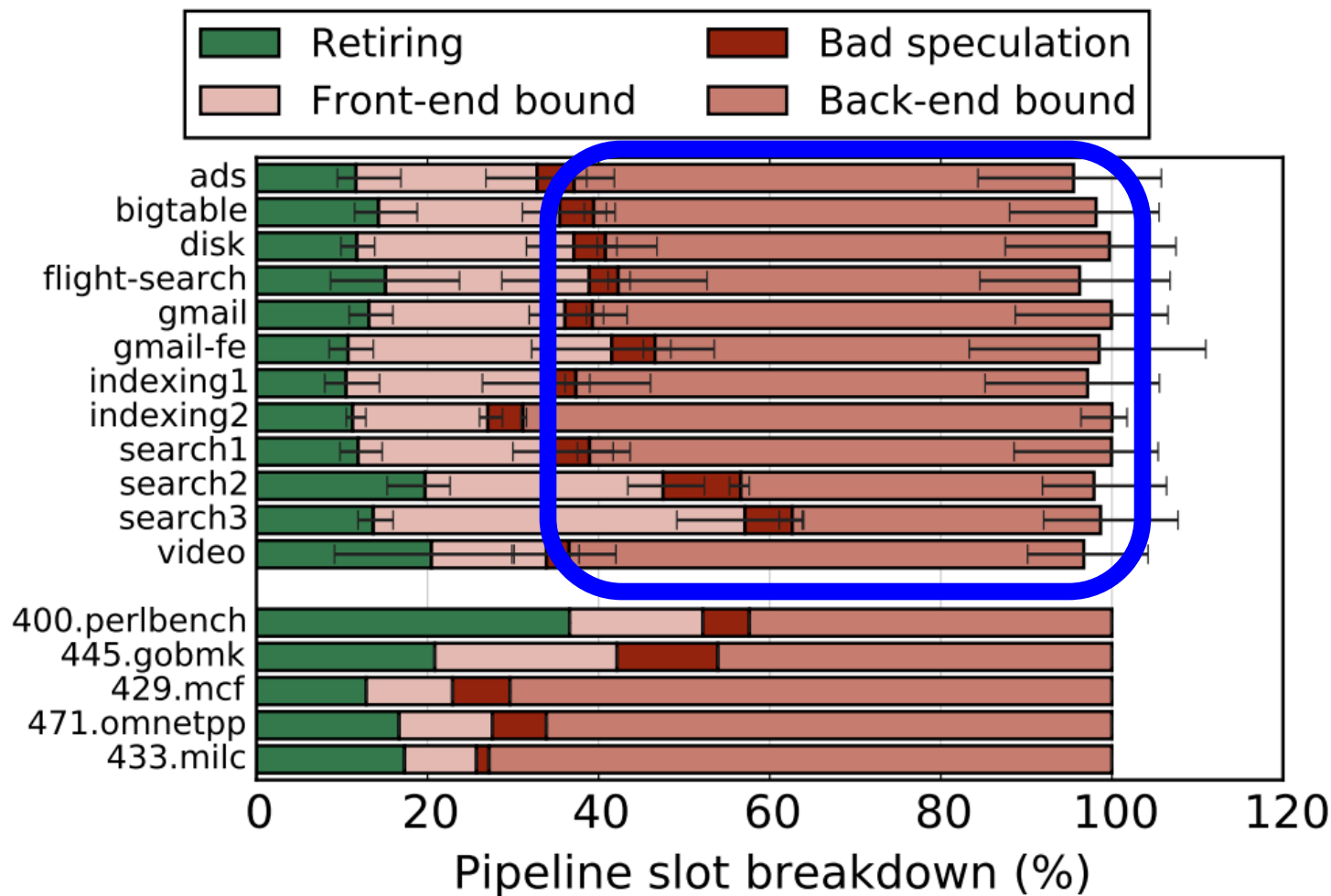
‡Desktop Platforms Group

Intel Corporation

chris.wilkerson@intel.com

The Performance Perspective (Today)

- All of Google's Data Center Workloads (2015):



Three Key Systems Trends

1. Data access is a major bottleneck

- Applications are increasingly data hungry

2. Energy consumption is a key limiter

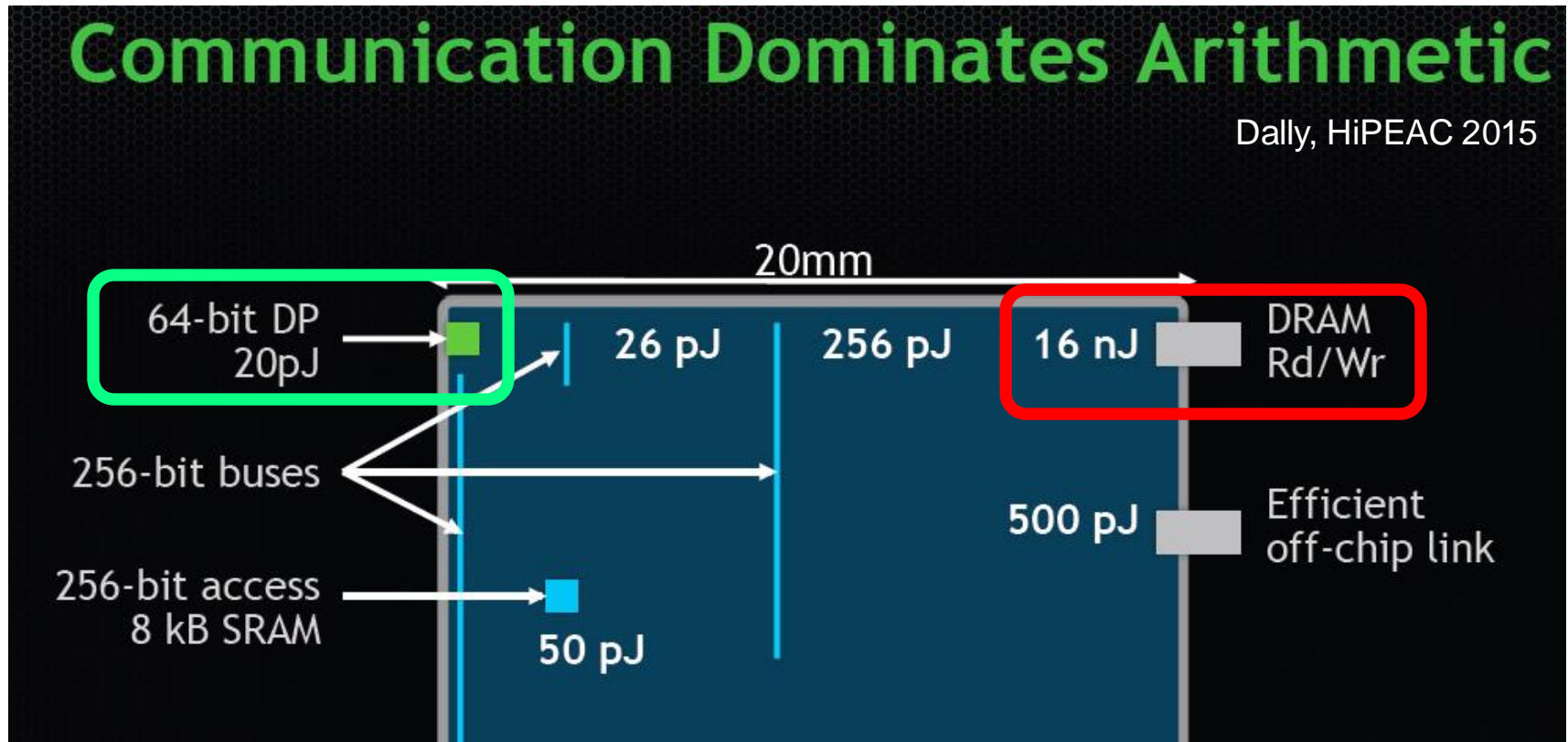
3. Data movement energy dominates compute

- Especially true for off-chip to on-chip movement

Data Movement vs. Computation Energy

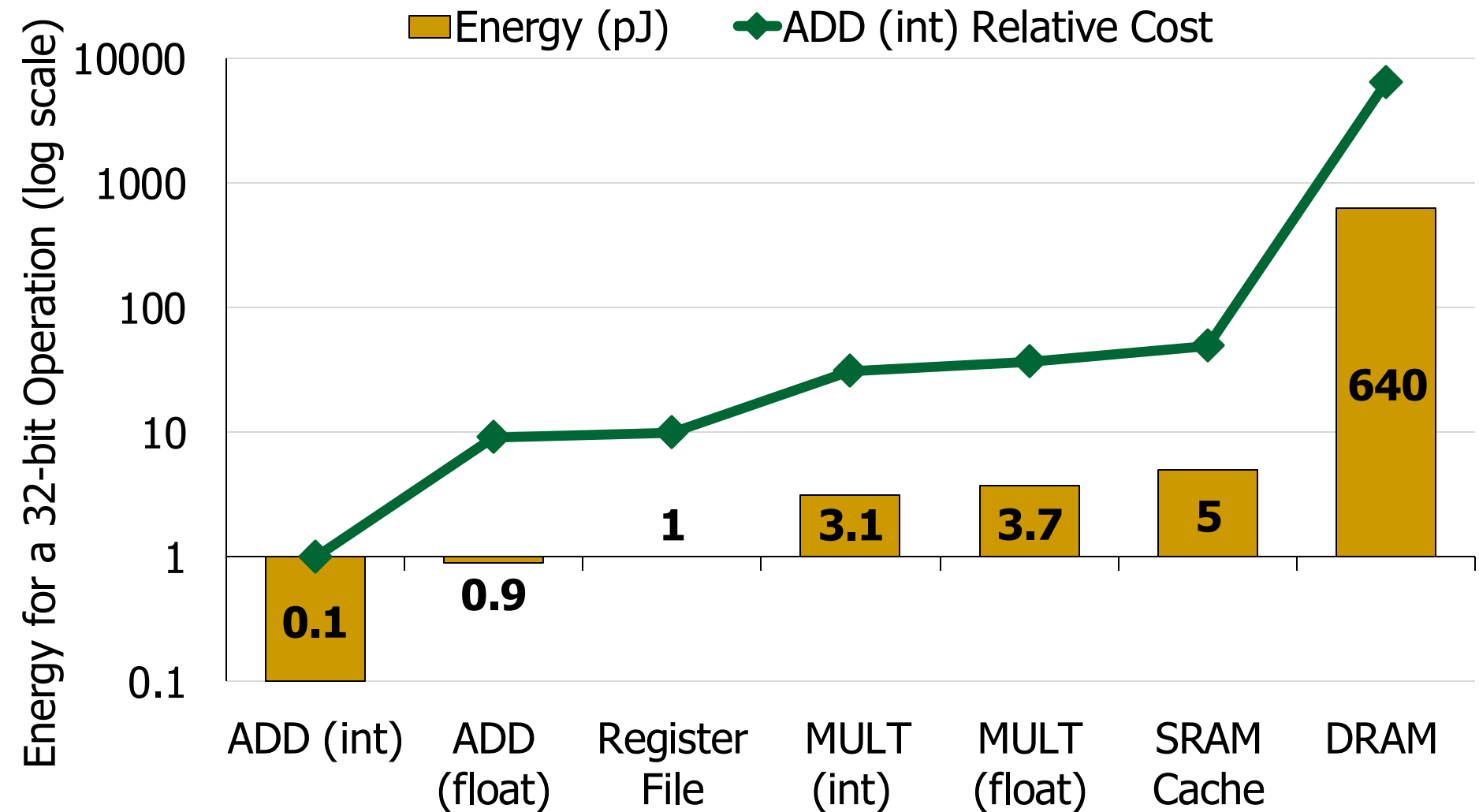
Communication Dominates Arithmetic

Dally, HiPEAC 2015

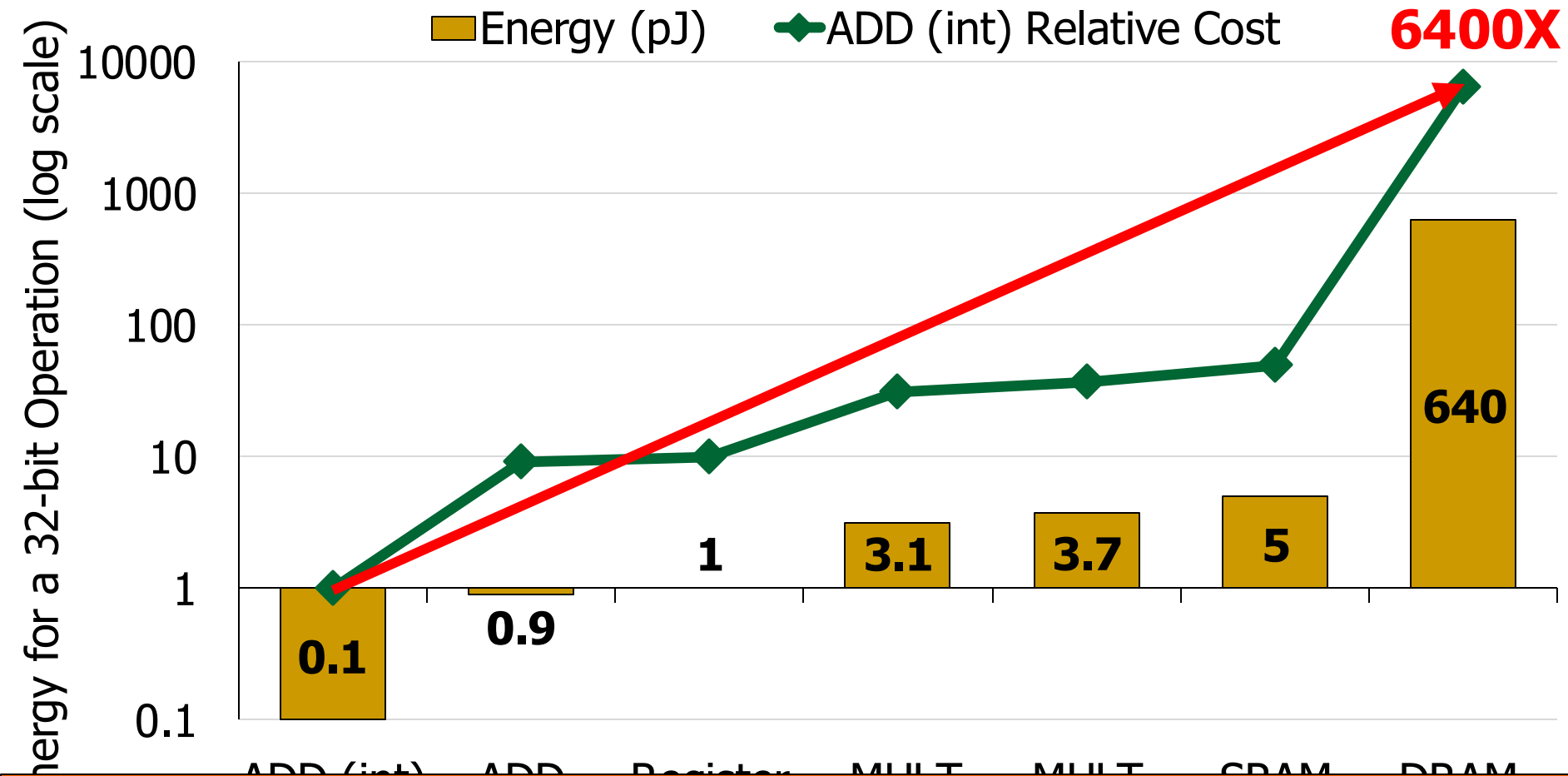


A memory access consumes $\sim 100\text{-}1000\times$ the energy of a complex addition

Data Movement vs. Computation Energy



Data Movement vs. Computation Energy



A memory access consumes 6400X the energy of a simple integer addition

Energy Waste in Mobile Devices

- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu, ["Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks"](#) *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Williamsburg, VA, USA, March 2018.

62.7% of the total system energy
is spent on **data movement**

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

Energy Waste in Accelerators

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,
["Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"](#)
Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (14 minutes)]

**> 90% of the total system energy
is spent on **memory** in large ML models**

Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand^{†◇}
Geraldo F. Oliveira^{*}

Saugata Ghose[‡]
Xiaoyu Ma[§]

Berkin Akin[§]
Eric Shiu[§]

Ravi Narayanaswami[§]
Onur Mutlu^{*†}

[†]Carnegie Mellon Univ.

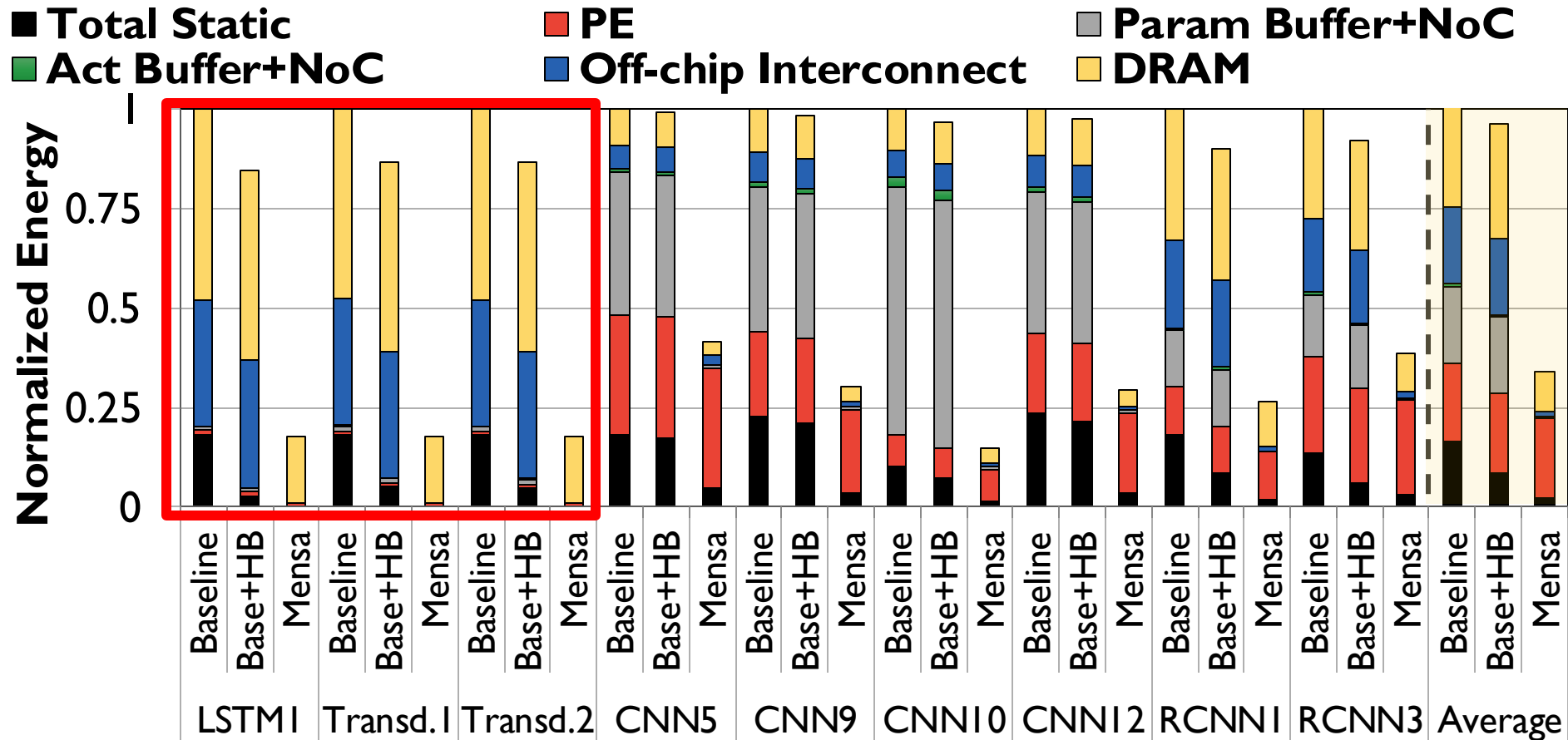
[◇]Stanford Univ.

[‡]Univ. of Illinois Urbana-Champaign

[§]Google

^{*}ETH Zürich

Example Energy Breakdowns

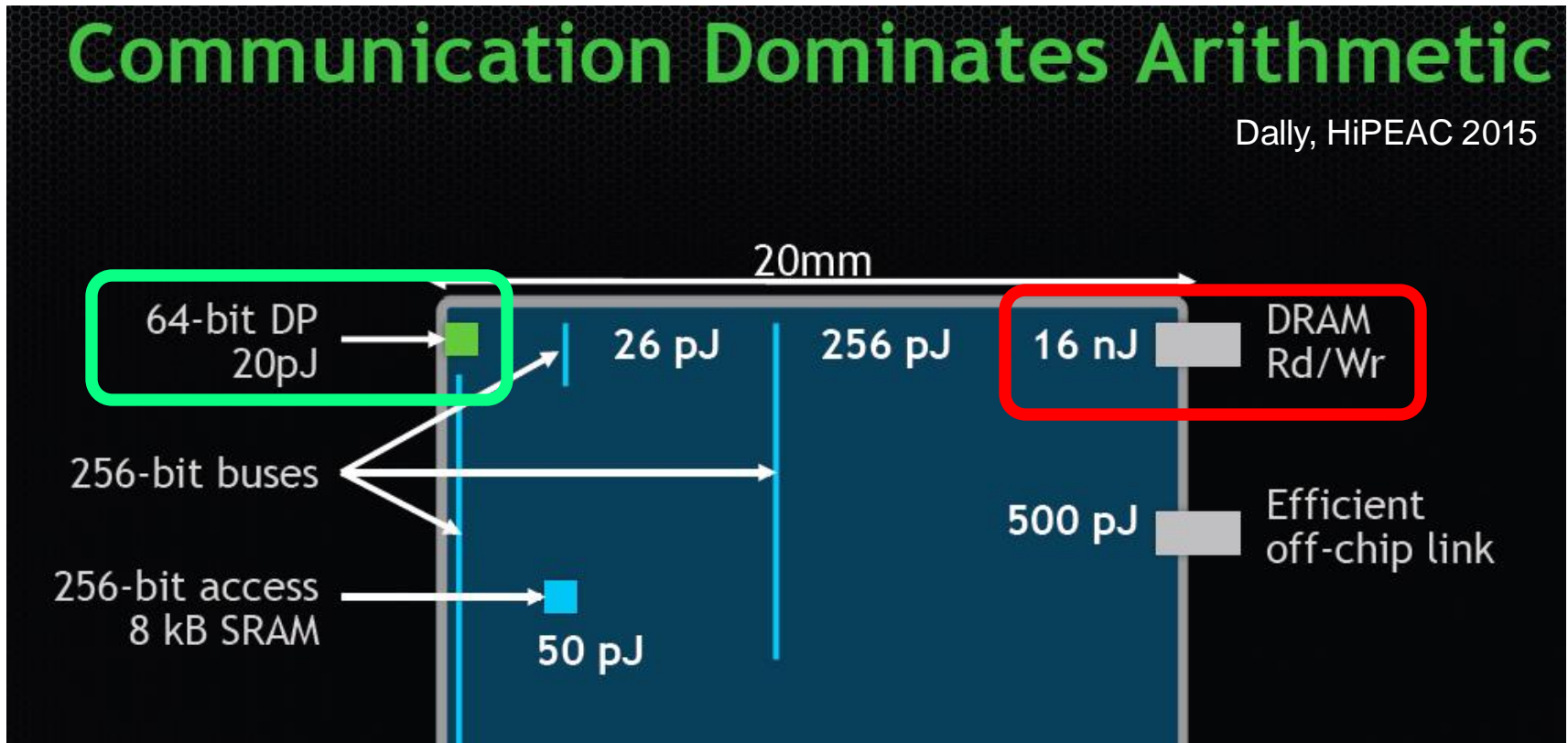


**In LSTMs and Transducers used by Google,
>90% energy spent on off-chip interconnect and DRAM**

We Do Not Want to Move Data!

Communication Dominates Arithmetic

Dally, HiPEAC 2015



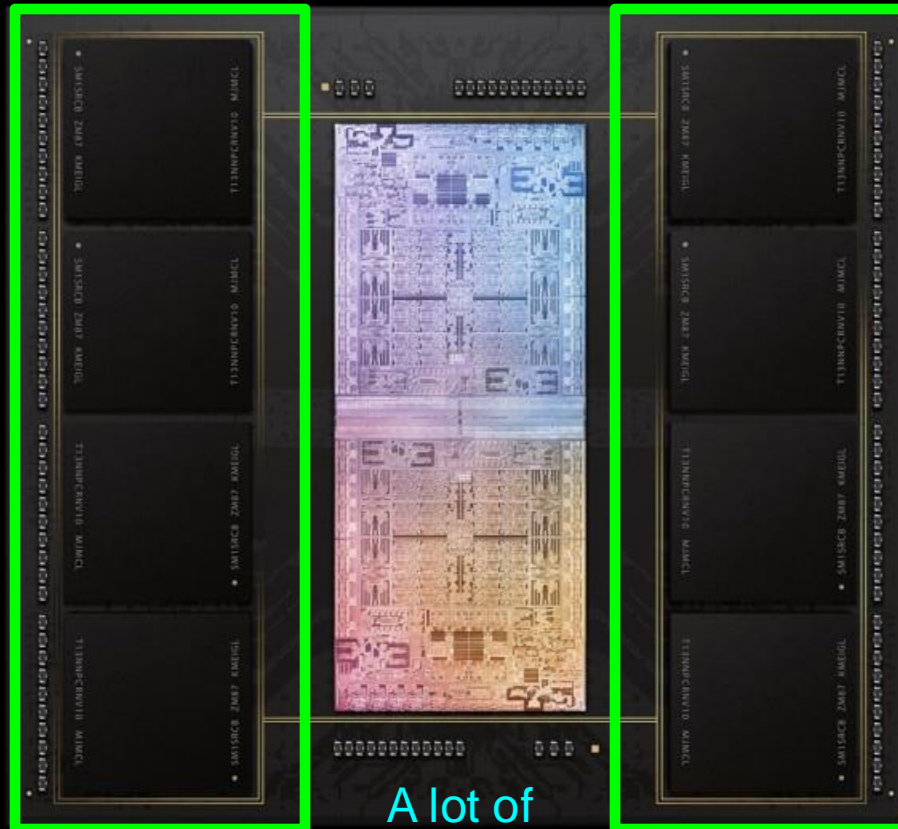
A memory access consumes $\sim 100\text{-}1000\times$ the energy of a complex addition

We Need A Paradigm Shift To ...

- Enable computation with minimal data movement
- Compute where it makes sense (where data resides)
- Make computing architectures more data-centric

Process Data Where It Makes Sense

Sensors



A lot of
SRAM

Storage

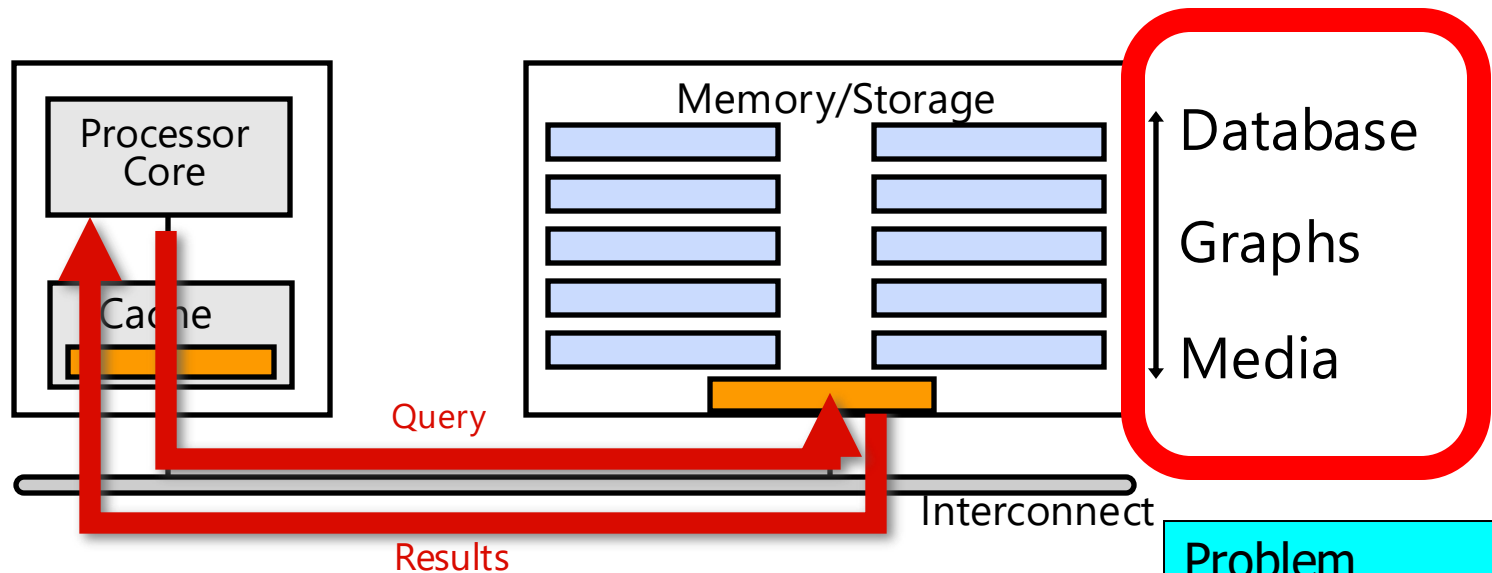
DRAM

DRAM

Storage

Apple M1 Ultra System (2022)

Goal: Processing Inside Memory/Storage



- Many questions ... How do we design the:
 - ❑ compute-capable memory & controllers?
 - ❑ processors & communication units?
 - ❑ software & hardware interfaces?
 - ❑ system software, compilers, languages?
 - ❑ algorithms & theoretical foundations?

Problem

Algorithm

Program/Language

System Software

SW/HW Interface

Micro-architecture

Logic

Devices

Electrons

Processing in/near Memory: An Old Idea

- Kautz, "Cellular Logic-in-Memory Arrays", IEEE TC 1969.

IEEE TRANSACTIONS ON COMPUTERS, VOL. C-18, NO. 8, AUGUST 1969

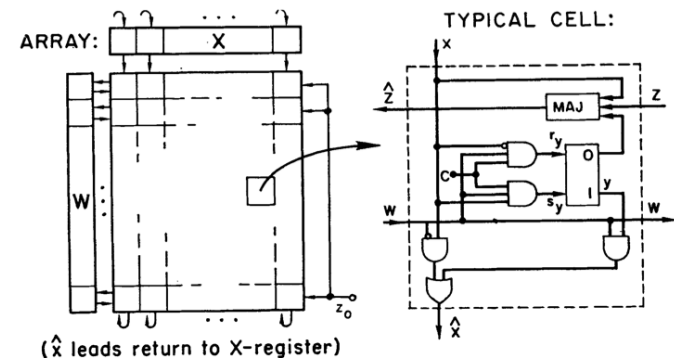
Cellular Logic-in-Memory Arrays

WILLIAM H. KAUTZ, MEMBER, IEEE

Abstract—As a direct consequence of large-scale integration, many advantages in the design, fabrication, testing, and use of digital circuitry can be achieved if the circuits can be arranged in a two-dimensional iterative, or cellular, array of identical elementary networks, or cells. When a small amount of storage is included in each cell, the same array may be regarded either as a logically enhanced memory array, or as a logic array whose elementary gates and connections can be "programmed" to realize a desired logical behavior.

In this paper the specific engineering features of such cellular logic-in-memory (CLIM) arrays are discussed, and one such special-purpose array, a cellular sorting array, is described in detail to illustrate how these features may be achieved in a particular design. It is shown how the cellular sorting array can be employed as a single-address, multiword memory that keeps in order all words stored within it. It can also be used as a content-addressed memory, a pushdown memory, a buffer memory, and (with a lower logical efficiency) a programmable array for the realization of arbitrary switching functions. A second version of a sorting array, operating on a different sorting principle, is also described.

Index Terms—Cellular logic, large-scale integration, logic arrays logic in memory, push-down memory, sorting, switching functions.



$$\begin{aligned}\hat{x} &= \bar{w}x + wy \\ s_y &= wcx, r_y = wc\bar{x} \\ \hat{z} &= M(x, \bar{y}, z) = x\bar{y} + z(x + \bar{y})\end{aligned}$$

Fig. 1. Cellular sorting array I.

Processing in/near Memory: An Old Idea

- Stone, “A Logic-in-Memory Computer,” IEEE TC 1970.

A Logic-in-Memory Computer

HAROLD S. STONE

Abstract—If, as presently projected, the cost of microelectronic arrays in the future will tend to reflect the number of pins on the array rather than the number of gates, the logic-in-memory array is an extremely attractive computer component. Such an array is essentially a microelectronic memory with some combinational logic associated with each storage element.

Why In-Memory Computation Today?

- **Huge demand from Applications & Systems**
 - ❑ Data access bottleneck
 - ❑ Energy & power bottlenecks
 - ❑ Data movement energy dominates computation energy
 - ❑ Need all at the same time: performance, energy, sustainability
 - ❑ We can improve all metrics by minimizing data movement
- **Huge problems with Memory Technology**
 - ❑ Memory technology scaling is not going well (e.g., RowHammer)
 - ❑ Many scaling issues demand intelligence in memory
- **Designs are squeezed in the middle**

One can
predictably induce errors
in most DRAM memory chips

Kim+, "[Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors](#)," ISCA 2014.



Rowhammer

RowHammer [ISCA 2014]

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"

Proceedings of the 41st International Symposium on Computer Architecture (ISCA), Minneapolis, MN, June 2014.

[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Session Slides \(pptx\) \(pdf\)](#)] [[Source Code and Data](#)] [[Lecture Video](#) (1 hr 49 mins), 25 September 2020]

One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD ([link](#)).

Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 ([Retrospective \(pdf\)](#) [Full Issue](#)).

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim¹ Ross Daly* Jeremie Kim¹ Chris Fallin* Ji Hye Lee¹
Donghyuk Lee¹ Chris Wilkerson² Konrad Lai Onur Mutlu¹

¹Carnegie Mellon University ²Intel Labs

Main Memory Needs Intelligent Controllers

Industry's Intelligent DRAM Controllers (I)

ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

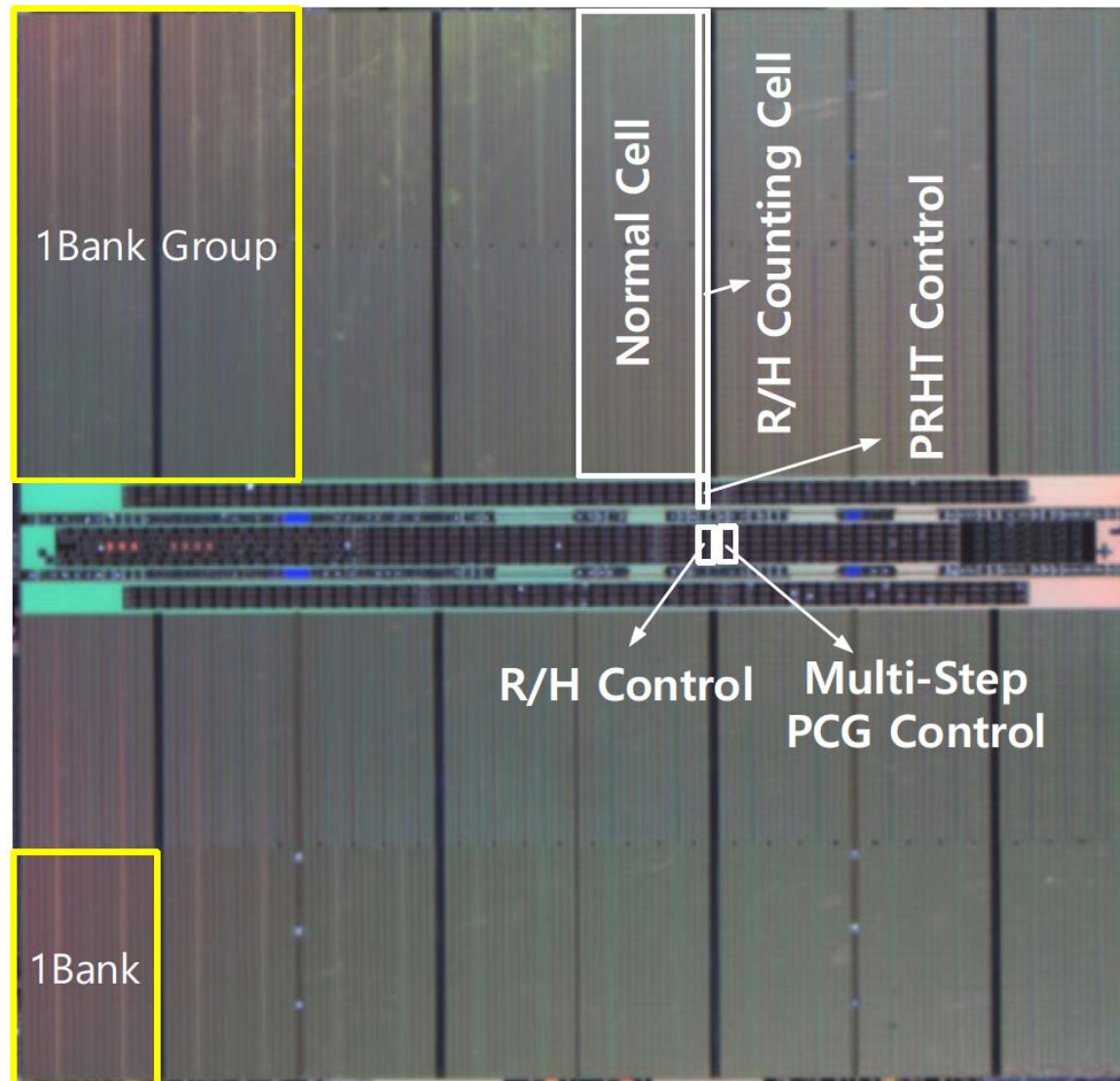


Industry's Intelligent DRAM Controllers (II)

SK hynix Semiconductor, Icheon, Korea

DRAM products have been recently adopted in a wide range of high-performance computing applications: such as in cloud computing, in big data systems, and IoT devices. This demand creates larger memory capacity requirements, thereby requiring aggressive DRAM technology node scaling to reduce the cost per bit [1,2]. However, DRAM manufacturers are facing technology scaling challenges due to row hammer and refresh retention time beyond 1a-nm [2]. Row hammer is a failure mechanism, where repeatedly activating a DRAM row disturbs data in adjacent rows. Scaling down severely threatens reliability since a reduction of DRAM cell size leads to a reduction in the intrinsic row hammer tolerance [2,3]. To improve row hammer tolerance, there is a need to probabilistically activate adjacent rows with carefully sampled active addresses and to improve intrinsic row hammer tolerance [2]. In this paper, row-hammer-protection and refresh-management schemes are presented to guarantee DRAM security and reliability despite the aggressive scaling from 1a-nm to sub 10-nm nodes. The probabilistic-aggressor-tracking scheme with a refresh-management function (RFM) and per-row hammer tracking (PRHT) improve DRAM resilience. A multi-step precharge reinforces intrinsic row-hammer tolerance and a core-bias modulation improves retention time: even in the face of cell-transistor degradation due to technology scaling. This comprehensive scheme leads to a reduced probability of failure, due to row hammer attacks, by 93.1% and an improvement in retention time by 17%.

Industry's Intelligent DRAM Controllers (III)




ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyoung Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

RowHammer Solutions in JEDEC (2024)



Global Standards for the Microelectronics Industry

STANDARDS & DOCUMENTS

COMMITTEES

NEWS

EVENTS & MEETINGS

JOIN

DDR5 SDRAM

JESD79-5C

Apr 2024

Release Number: Version 1.30

Version 1.30

This standard defines the DDR5 SDRAM specification, including features, functionalities, AC and DC characteristics, packages, and ball/signal assignments. The purpose of this Standard is to define the minimum set of requirements for JEDEC compliant 8 Gb through 32 Gb for x4, x8, and x16 DDR5 SDRAM devices. This standard was created based on the DDR4 standards (JESD79-4) and some aspects of the DDR, DDR2, DDR3, and LPDDR4 standards (JESD79, JESD79-2, JESD79-3, and JESD209-4).

Committee(s): [JC-42](#), [JC-42.3](#)

A RowHammer Survey Across the Stack

- Onur Mutlu and Jeremie Kim,
["RowHammer: A Retrospective"](#)
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.
[[Preliminary arXiv version](#)]
[[Slides from COSADE 2019 \(pptx\)](#)]
[[Slides from VLSI-SOC 2020 \(pptx\) \(pdf\)](#)]
[[Talk Video](#) (1 hr 15 minutes, with Q&A)]

RowHammer: A Retrospective

Onur Mutlu^{§‡} Jeremie S. Kim^{‡§}
§ETH Zürich ‡Carnegie Mellon University

A RowHammer Survey: Recent Update

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci,
"Fundamentally Understanding and Solving RowHammer"
Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, January 2023.
[[arXiv version](#)]
[[Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (26 minutes)]

Fundamentally Understanding and Solving RowHammer

Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

Ataberk Olgun
ataberk.olgund@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

A. Giray Yağlıkçı
giray.yaglikci@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

<https://arxiv.org/pdf/2211.07613.pdf>



- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tugrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu, **"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"**

Proceedings of the 50th International Symposium on Computer Architecture (ISCA), Orlando, FL, USA, June 2023.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Video](#) (3 minutes)]

[[RowPress Source Code and Datasets \(Officially Artifact Evaluated with All Badges\)](#)]

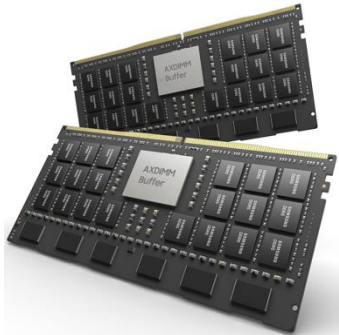
***Officially artifact evaluated as available, reusable and reproducible.
Best artifact award at ISCA 2023. IEEE Micro Top Pick in 2024.***

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

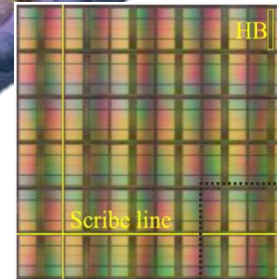
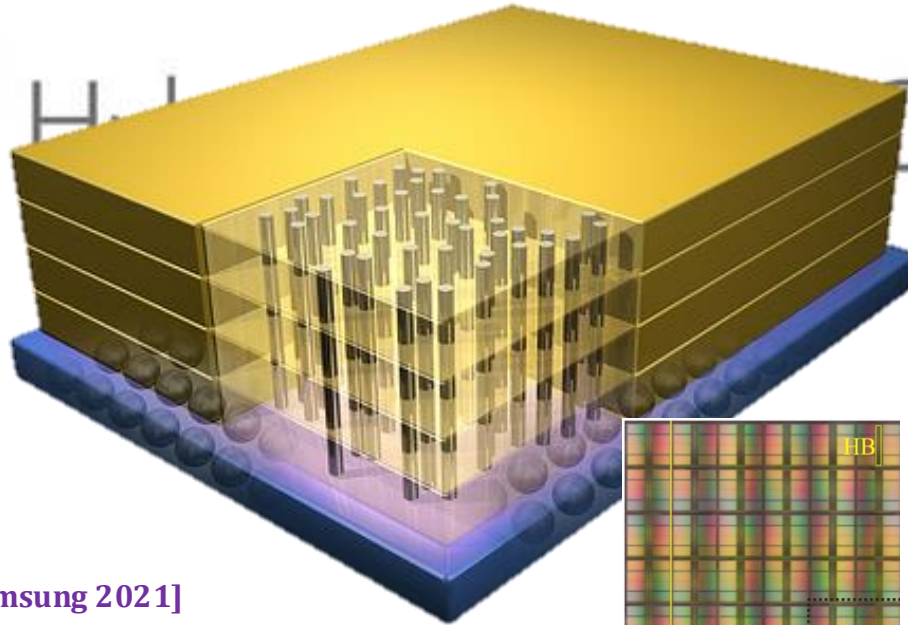
Haocong Luo Ataberk Olgun A. Giray Yağlıkçı Yahya Can Tuğrul Steve Rhyner
Meryem Banu Cavlak Joël Lindegger Mohammad Sadrosadati Onur Mutlu

ETH Zürich

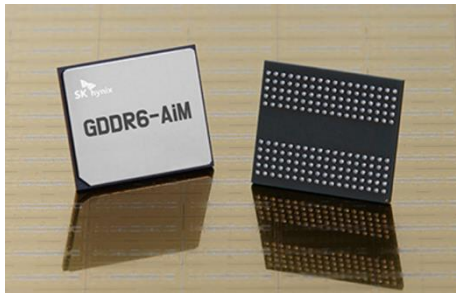
Processing-in-Memory Landscape Today



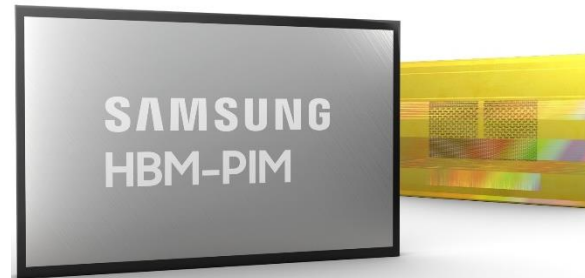
[Samsung 2021]



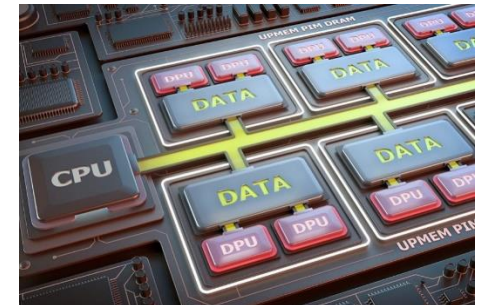
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]



[UPMEM 2019]

Processing-in-Memory Landscape Today

IEEE COMPUTER ARCHITECTURE LETTERS, VOL. 22, NO. 1, JANUARY-JUNE

Computational CXL-Memory Solution for Accelerating Memory-Intensive Applications

Joonseop Sim^{ID}, Soohong Ahn^{ID}, Taeyoung Ahn^{ID},
Seungyong Lee^{ID}, Myunghyun Rhee, Jooyoung Kim^{ID},
Kwangsik Shin, Donguk Moon^{ID},
Euseok Kim, and Kyoung Park^{ID}

Abstract—CXL interface is the up-to-date technology that enables effective memory expansion by providing a memory-sharing protocol in configuring heterogeneous devices. However, its limited physical bandwidth can be a significant bottleneck for emerging data-intensive applications. In this work, we propose a novel CXL-based memory disaggregation architecture with a real-world prototype demonstration, which overcomes the bandwidth limitation of the CXL interface using near-data processing. The experimental results demonstrate that our design achieves up to $1.9\times$ better performance/power efficiency than the existing CPU system.

Index Terms—Compute express link (CXL), near-data-processing (NDP)

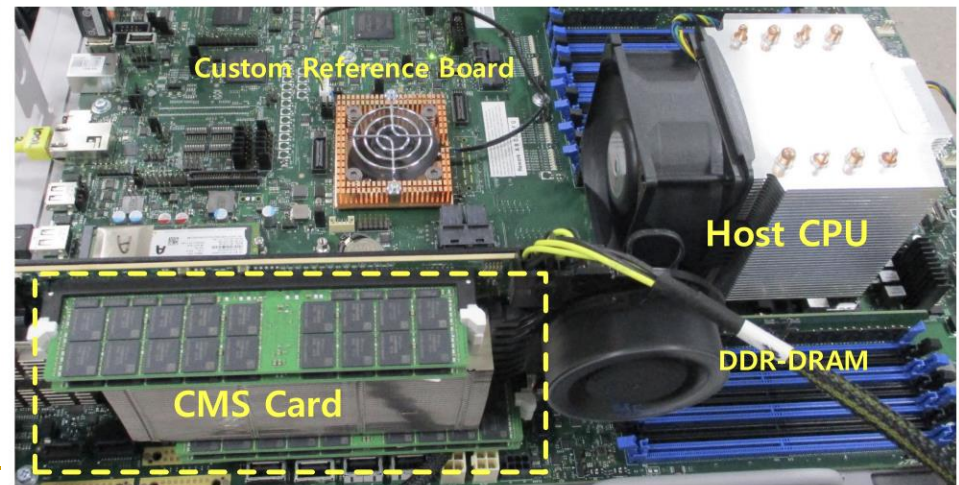


Fig. 6. FPGA prototype of proposed CMS card.

Processing-in-Memory Landscape Today

Samsung Processing in Memory Technology at Hot Chips 2023

By Patrick Kennedy - August 28, 2023



Samsung PIM PNM For Transformer Based AI HC35_Page_24

PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu^{a,b}, Saugata Ghose^{b,c}, Juan Gómez-Luna^a, Rachata Ausavarungnirun^d

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

^d*King Mongkut's University of Technology North Bangkok*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,

"A Modern Primer on Processing in Memory"

*Invited Book Chapter in **Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann**, Springer, to be published in 2021.*

PIM Course (Fall 2022)

■ Fall 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory

■ Spring 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory

■ Youtube Livestream (Fall 2022):

- <https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0GOBrnKy>

■ Youtube Livestream (Spring 2022):

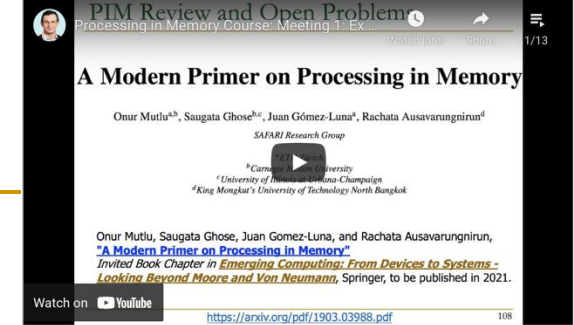
- <https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX>

■ Project course

- Taken by Bachelor's/Master's students
- Processing-in-Memory lectures
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>

SAFARI

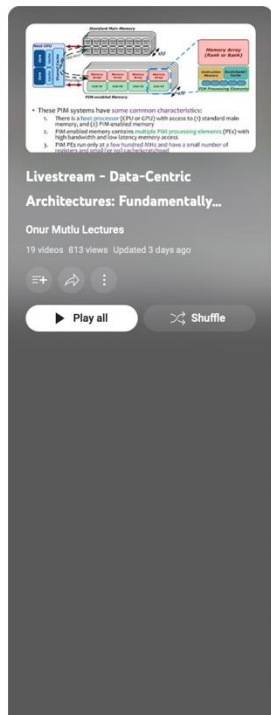


Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	10.03 Thu.	YouTube Live	M1: P&S PIM Course Presentation (PDF) (PPT)	Required Materials Recommended Materials	HW 0 Out
W2	15.03 Tue.		Hands-on Project Proposals		
	17.03 Thu.	YouTube Premiere	M2: Real-world PIM: UPMEM PIM (PDF) (PPT)		
W3	24.03 Thu.	YouTube Live	M3: Real-world PIM: Microbenchmarking of UPMEM PIM (PDF) (PPT)		
W4	31.03 Thu.	YouTube Live	M4: Real-world PIM: Samsung HBM-PIM (PDF) (PPT)		
W5	07.04 Thu.	YouTube Live	M5: How to Evaluate Data Movement Bottlenecks (PDF) (PPT)		
W6	14.04 Thu.	YouTube Live	M6: Real-world PIM: SK Hynix AIM (PDF) (PPT)		
W7	21.04 Thu.	YouTube Premiere	M7: Programming PIM Architectures (PDF) (PPT)		
W8	28.04 Thu.	YouTube Premiere	M8: Benchmarking and Workload Suitability on PIM (PDF) (PPT)		
W9	05.05 Thu.	YouTube Premiere	M9: Real-world PIM: Samsung AxDIMM (PDF) (PPT)		
W10	12.05 Thu.	YouTube Premiere	M10: Real-world PIM: Alibaba HB-PNM (PDF) (PPT)		
W11	19.05 Thu.	YouTube Live	M11: SpMV on a Real PIM Architecture (PDF) (PPT)		
W12	26.05 Thu.	YouTube Live	M12: End-to-End Framework for Processing-using-Memory (PDF) (PPT)		
W13	02.06 Thu.	YouTube Live	M13: Bit-Serial SIMD Processing using DRAM (PDF) (PPT)		
W14	09.06 Thu.	YouTube Live	M14: Analyzing and Mitigating ML Inference Bottlenecks (PDF) (PPT)		
W15	15.06 Thu.	YouTube Live	M15: In-Memory HTAP Databases with HW/SW Co-design (PDF) (PPT)		
W16	23.06 Thu.	YouTube Live	M16: In-Storage Processing for Genome Analysis (PDF) (PPT)		
W17	18.07 Mon.	YouTube Premiere	M17: How to Enable the Adoption of PIM? (PDF) (PPT)		
W18	09.08 Tue.	YouTube Premiere	SS1: ISVLSI 2022 Special Session on PIM (PDF & PPT)		


Processing-in-Memory Course (Spring 2023)

- Short weekly lectures
- Hands-on projects



- PIM Course: Lecture 1: Data-Centric Architectures: Improving Performance & Energy (Spring 2023)**
Onur Mutlu Lectures • 1.1K views • Streamed 3 months ago
1:14:16
- PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks (Spring 2023)**
Onur Mutlu Lectures • 332 views • 2 months ago
16:37
- ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads**
Onur Mutlu Lectures • 1.5K views • Streamed 2 months ago
6:27:39
- PIM Course: Lecture 3: Real-world PIM: UPMEM PIM (Spring 2023)**
Onur Mutlu Lectures • 411 views • 2 months ago
15:43
- PIM Course: Lecture 4: Real-world PIM: Microbenchmarking of UPMEM PIM (Spring 2023)**
Onur Mutlu Lectures • 188 views • 2 months ago
24:10
- Análisis Experimental de una Arquitectura PIM - Juan Gómez Luna - Lecture in Spanish @ U. de Córdoba**
Onur Mutlu Lectures • 169 views • 2 months ago
2:27:12
- PIM Course: Lecture 5: Real-world PIM: Samsung HBM-PIM (Spring 2023)**
Onur Mutlu Lectures • 483 views • 2 months ago
24:08
- PIM Course: Lecture 6: Real-world PIM: SK Hynix AIM (Spring 2023)**
Onur Mutlu Lectures • 573 views • 1 month ago
35:50
- PIM Course: Lecture 7: Real-world PIM: Samsung AxDIMM (Spring 2023)**
Onur Mutlu Lectures • 325 views • 1 month ago
21:32

https://www.youtube.com/playlist?list=PL5Q2soXY2zi_F0buoAZVSq_o6UySWQHvZ

**SAFARI Project & Seminars Courses**
(Spring 2023)

Search

Recent Changes Media Manager Sitemap

Trace: • heterogeneous_systems • processing_in_memory

Home

Courses

- SoftMC
- Ramulator
- Accelerating Genomics
- Mobile Genomics
- Processing-in-Memory**
- Heterogeneous Systems
- Modern SSDs
- Hardware/Software Co-design

processing_in_memory

Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)

Course Description

Data movement between the memory units and the compute units of current computing systems is a major performance and energy bottleneck. From large-scale servers to mobile devices, data movement costs dominate computation costs in terms of both performance and energy consumption. For example, data movement between the main memory and the processing cores accounts for 62% of the total system energy in consumer applications. As a result, the data movement bottleneck is a huge burden that greatly limits the energy efficiency and performance of modern computing systems. This phenomenon is an undesired effect of the dichotomy between memory and the processor, which leads to the data movement bottleneck.

Many modern and important workloads such as machine learning, computational biology, graph processing, databases, video analytics, and real-time data analytics suffer greatly from the data movement bottleneck. These workloads are exemplified by irregular memory accesses, relatively low data reuse, low cache line utilization, low arithmetic intensity (i.e., ratio of operations per accessed byte), and large datasets that greatly exceed the main memory size. The computation in these workloads cannot usually compensate for the data movement costs. In order to alleviate this data movement bottleneck, we need a paradigm shift from the traditional processor-centric design, where all computation takes place in the compute units, to a more data-centric design where processing elements are placed closer to or inside where the data resides. This paradigm of computing is known as Processing-in-Memory (PIM).

This is your perfect P&S if you want to become familiar with the main PIM technologies, which represent "the next big thing" in Computer Architecture. You will work hands-on with the first real-world PIM architecture, will explore different PIM architecture designs for important workloads, and will develop tools to enable research of future PIM systems. Projects in this course span software and hardware as well as the software/hardware interface. You can potentially work on developing and optimizing new workloads for the first real-world PIM hardware or explore new PIM designs in simulators, or do something else that can forward our understanding of the PIM paradigm.

Prerequisites of the course:

- Digital Design and Computer Architecture (or equivalent course).
- Familiarity with C/C++ programming.
- Interest in future computer architectures and computing paradigms.
- Interest in discovering why things do or do not work and solving problems
- Interest in making systems efficient and usable

Table of Contents

- Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)
- Course Description
- Mentors
- Lecture Video Playlist on YouTube
- Spring 2023 Meetings/Schedule
- Past Lecture Video Playlists on YouTube
- Learning Materials
- Assignments

https://safari.ethz.ch/projects_and_seminars/spring2023/doku.php?id=processing_in_memory

SSD Course (Spring 2023)

Spring 2023 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2023/doku.php?id=modern_ssd

Fall 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=modern_ssd

Youtube Livestream (Spring 2023):

- https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2Shtm4z57&pp=iAQB

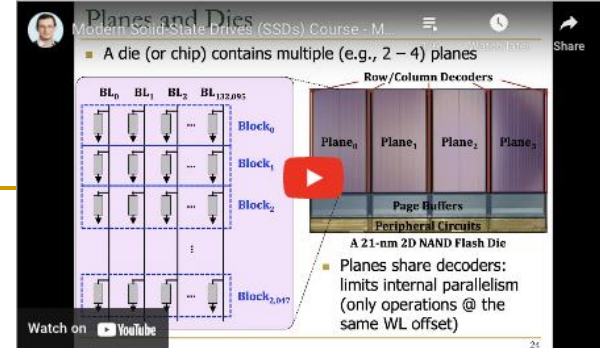
Youtube Livestream (Fall 2022):

- <https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&pp=iAQB>

Project course

- Taken by Bachelor's/Master's students
- SSD Basics and Advanced Topics
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>




Fall 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	06.10		M1: P&S Course Presentation PDF PPT	Required Recommended	
W2	12.10	YouTube Live	M2: Basics of NAND Flash-Based SSDs PDF PPT	Required Recommended	
W3	19.10	YouTube Live	M3: NAND Flash Read/Write Operations PDF PPT	Required Recommended	
W4	26.10	YouTube Live	M4: Processing inside NAND Flash PDF PPT	Required Recommended	
W5	02.11	YouTube Live	M5: Advanced NAND Flash Commands & Mapping PDF PPT	Required Recommended	
W6	09.11	YouTube Live	M6: Processing inside Storage PDF PPT	Required Recommended	
W7	23.11	YouTube Live	M7: Address Mapping & Garbage Collection PDF PPT	Required Recommended	
W8	30.11	YouTube Live	M8: Introduction to MQSim PDF PPT	Required Recommended	
W9	14.12	YouTube Live	M9: Fine-Grained Mapping and Multi-Plane Operation-Aware Block Management PDF PPT	Required Recommended	
W10	04.01.2023	YouTube Premiere	M10a: NAND Flash Basics PDF PPT	Required Recommended	
			M10b: Reducing Solid-State Drive Read Latency by Optimizing Read-Retry PDF PPT Paper	Required Recommended	
			M10c: Evanescence: Architectural Support for Efficient Data Sanitization in Modern Flash-Based Storage Systems PDF PPT Paper	Required Recommended	
			M10d: DeepSketch: A New Machine Learning-Based Reference Search Technique for Post-Deduplication Delta Compression PDF PPT Paper	Required Recommended	
W11	11.01	YouTube Live	M11: FLIN: Enabling Fairness and Enhancing Performance in Modern NVMe Solid State Drives PDF PPT	Required	
W12	25.01	YouTube Premiere	M12: Flash Memory and Solid-State Drives PDF PPT	Recommended	

PIM Tutorials [MICRO'23, ISCA'23, ASPLOS'23, HPCA'23, ISCA'24]

■ Lectures + Hands-on labs + Invited talks



ISCA 2023 Real-World PIM Tutorial

Search

Recent Changes Media Manager Sitemap

Trace: • start

Real-world Processing-in-Memory Systems for Modern Workloads

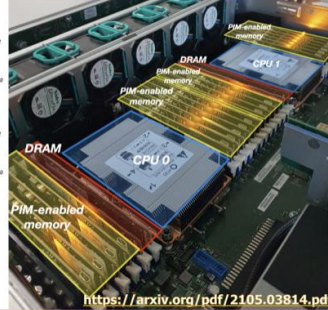
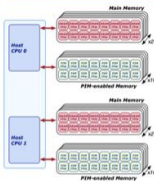
Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System



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

Table of Contents

- Real-world Processing-in-Memory Systems for Modern Workloads
- Tutorial Description
- Organizers
- Agenda (June 18, 2023)
 - Lectures (tentative)
 - Hands-on Labs (tentative)
 - Learning Materials

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

<https://www.youtube.com/live/GIb5EgSrWk0>

<https://events.safari.ethz.ch/isca-pim-tutorial/>

PIM Tutorial at ISCA 2024

ISCA 2024 Memory-Centric Computing Systems Tutorial

Saturday, June 29, Buenos Aires, Argentina

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

Program: <https://events.safari.ethz.ch/isca24-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



<https://www.youtube.com/watch?v=KV2MXvcBgb0>

<https://events.safari.ethz.ch/isca24-memorycentric-tutorial>

Upcoming PIM Tutorial at MICRO 2024

MICRO 2024 - Tutorial on Memory-Centric Computing Systems

Saturday, November 2nd, Austin, Texas, USA

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

Program: <https://events.safari.ethz.ch/micro24-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



<https://www.youtube.com/watch?v=KV2MXvcBgb0>

<https://events.safari.ethz.ch/micro24-memorycentric-tutorial/>

We Need to Think Differently
from the Past Approaches

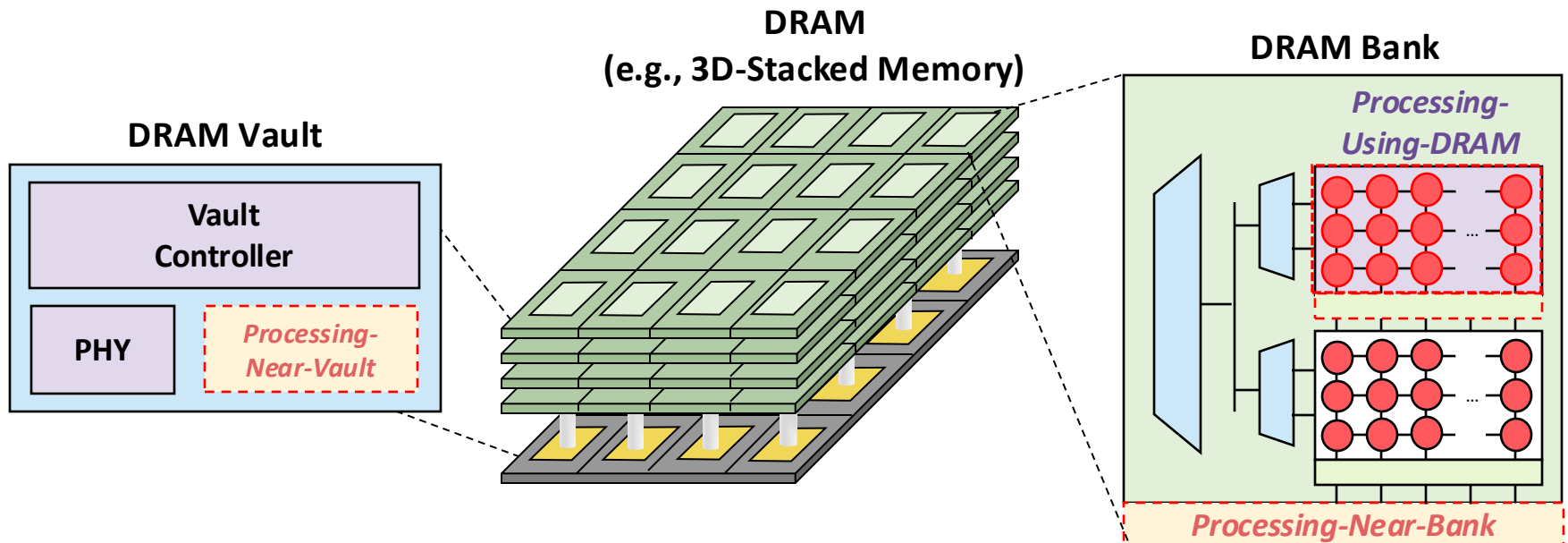
Processing in Memory: Two Approaches

1. Processing **using** Memory
2. Processing **near** Memory

Processing-in-Memory: Nature of Computation

Two main approaches for Processing-in-Memory:

- 1 Processing-Near-Memory:** Design compute logic and memory separately (today) and integrate logic closer to memory
- 2 Processing-Using-Memory:** Use analog operational principles of memory circuitry to perform computation (no compute logic)



A PIM Taxonomy

- **Nature** (of computation)

- **Using**: Use operational properties of memory structures
- **Near**: Add logic close to memory structures

- **Technology**

- Flash, DRAM, SRAM, RRAM, MRAM, FeRAM, PCM, 3D, ...

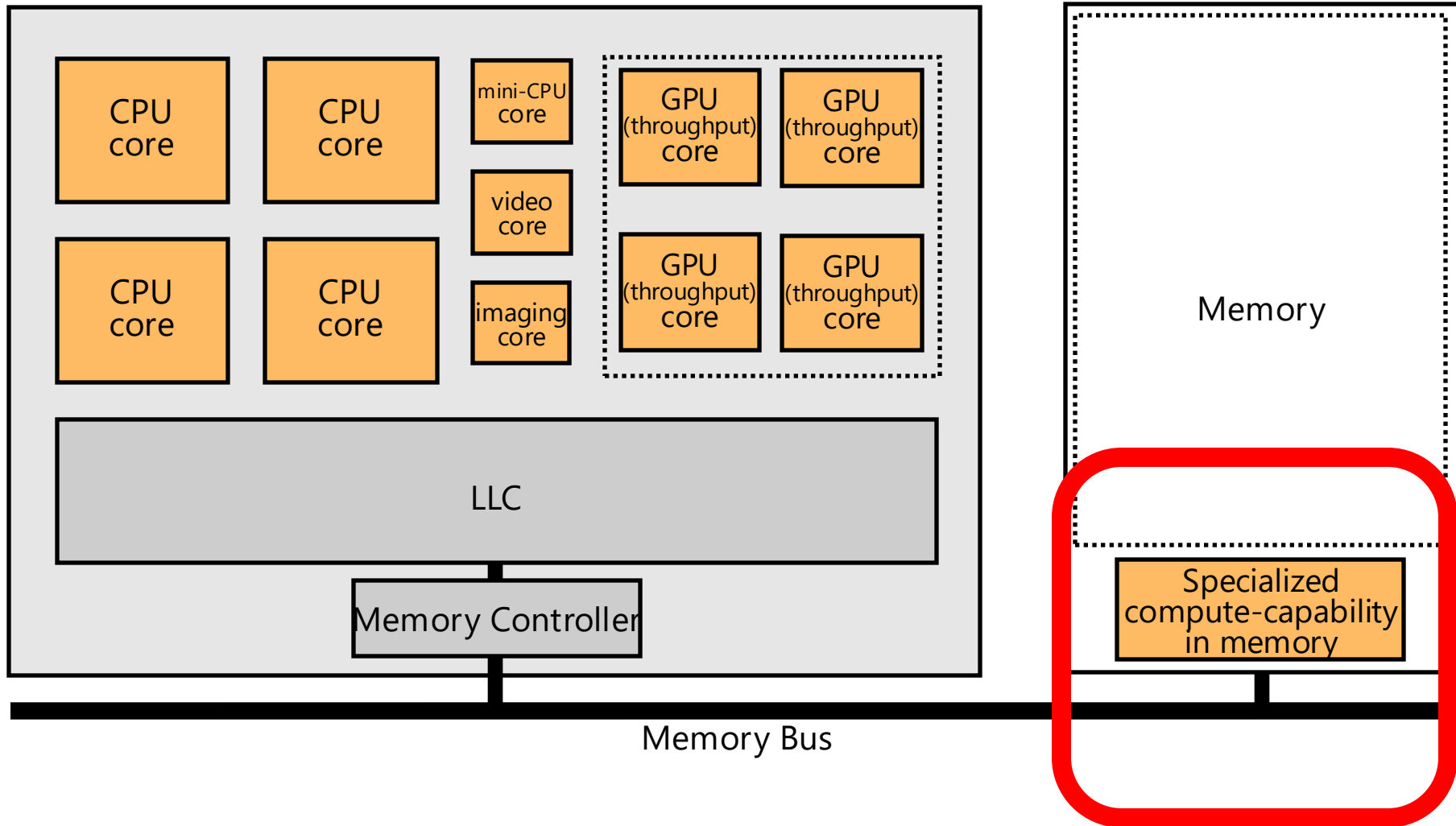
- **Location**

- Sensor, Cold Storage, Hard Disk, SSD, Main Memory, Cache, Register File, Memory Controller, Interconnect, ...

- A tuple of the three determines “PIM type”

- One can combine multiple “PIM types” in a system

Mindset: Memory as an Accelerator



Memory similar to a "conventional" accelerator

Example PIM Type: Processing using DRAM

- Nature: Using
- Technology: DRAM
- Location: Main Memory

- Processing using DRAM in Main Memory

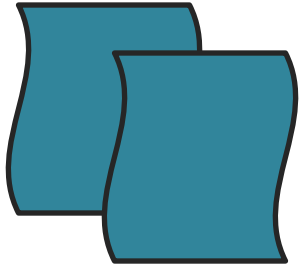
- Seshadri+, "Fast Bulk Bitwise AND and OR in DRAM", IEEE CAL 2015.
- Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO 2017.
- Hajinazar+, "SIMDRAM: A Framework for Bit-Serial SIMD Processing using DRAM," ASPLOS 2021.
- Oliveira+, "MIMDRAM: An End-to-End Processing-Using-DRAM System for High-Throughput, Energy-Efficient and Programmer-Transparent Multiple-Instruction Multiple-Data Processing," HPCA 2024.

Processing using DRAM

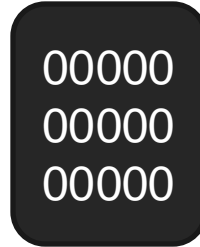
- We can support
 - Bulk bitwise AND, OR, NOT, MAJ
 - Bulk bitwise COPY and INIT/ZERO
 - True Random Number Generation; Physical Unclonable Functions
 - Lookup Table based more complex computation
- At low cost
- Using analog computation capability of DRAM
 - Idea: activating (multiple) rows performs computation
 - Even in commodity off-the-shelf DRAM chips!
- 30-77X performance and energy improvement
 - Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO 2017.
 - Seshadri+"RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.

Starting Simple: Data Copy and Initialization

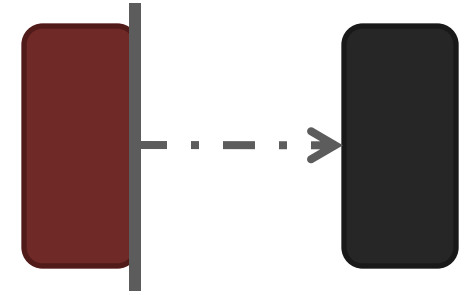
memmove & memcpy: 5% cycles in Google's datacenter [Kanev+ ISCA'15]



Forking



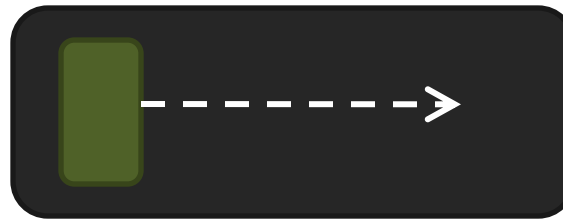
**Zero initialization
(e.g., security)**



Checkpointing



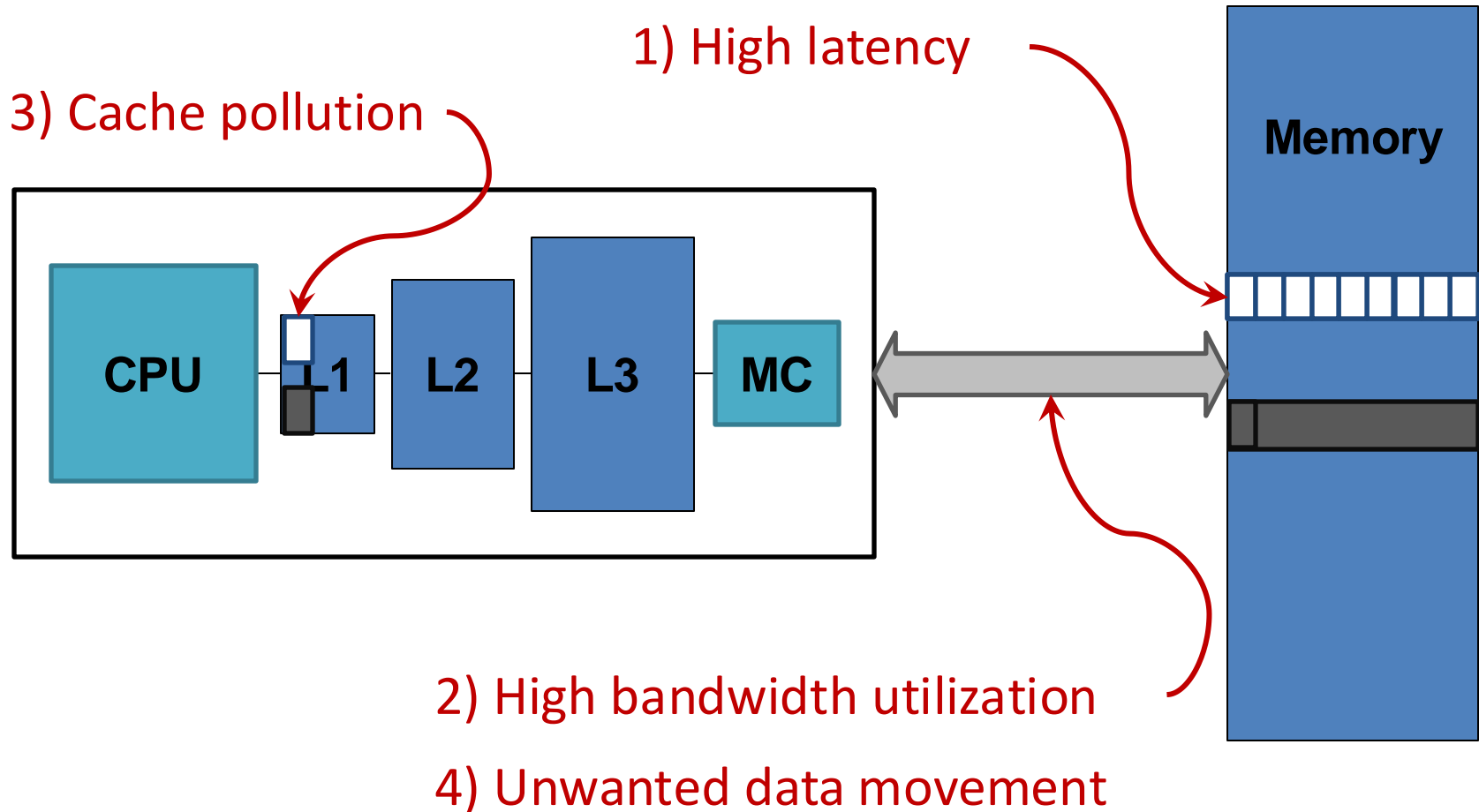
**VM Cloning
Deduplication**



Page Migration

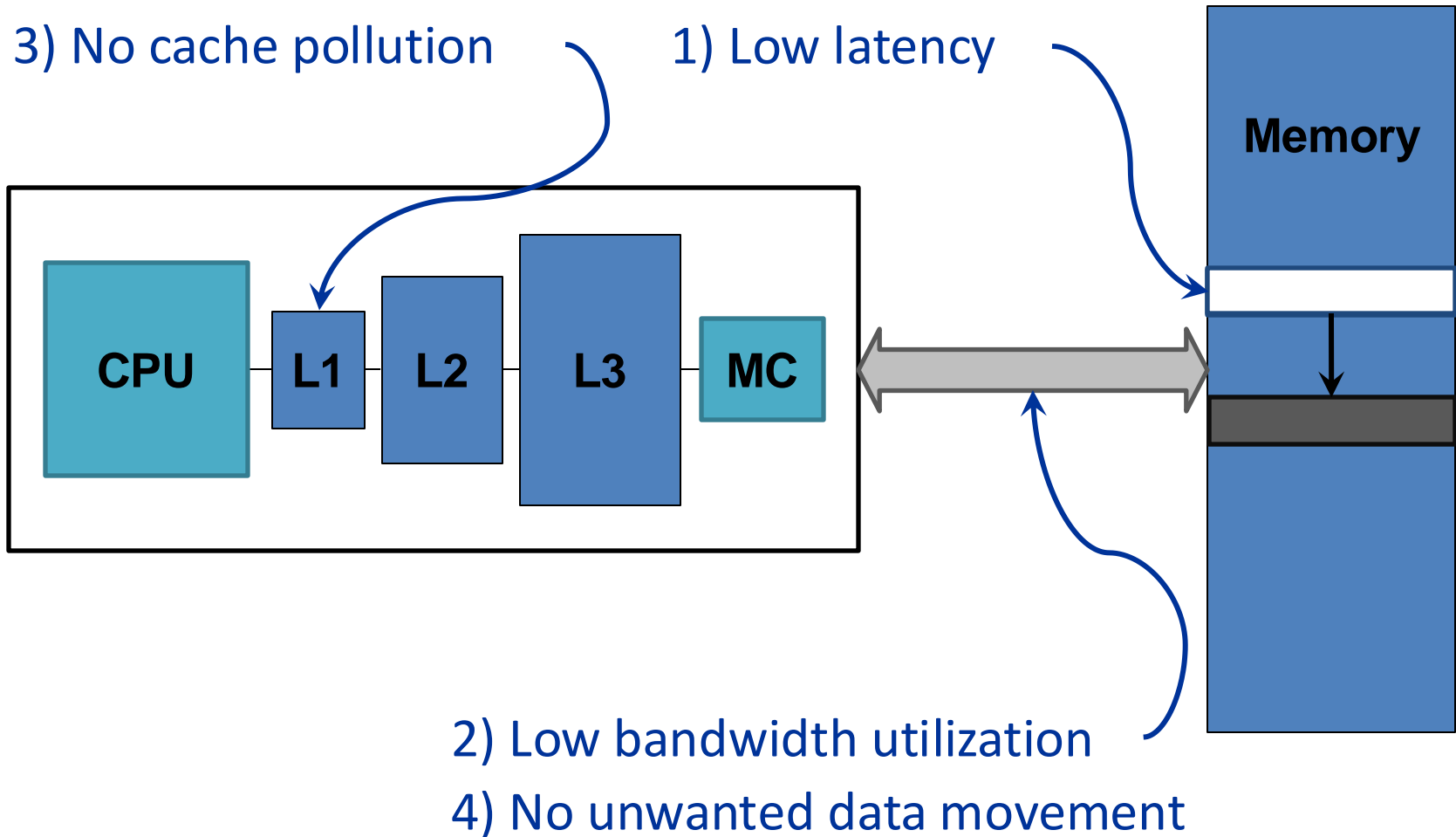
...
Many more

Today's Systems: Bulk Data Copy



1046ns, 3.6uJ (for 4KB page copy via DMA)

Future Systems: In-Memory Copy

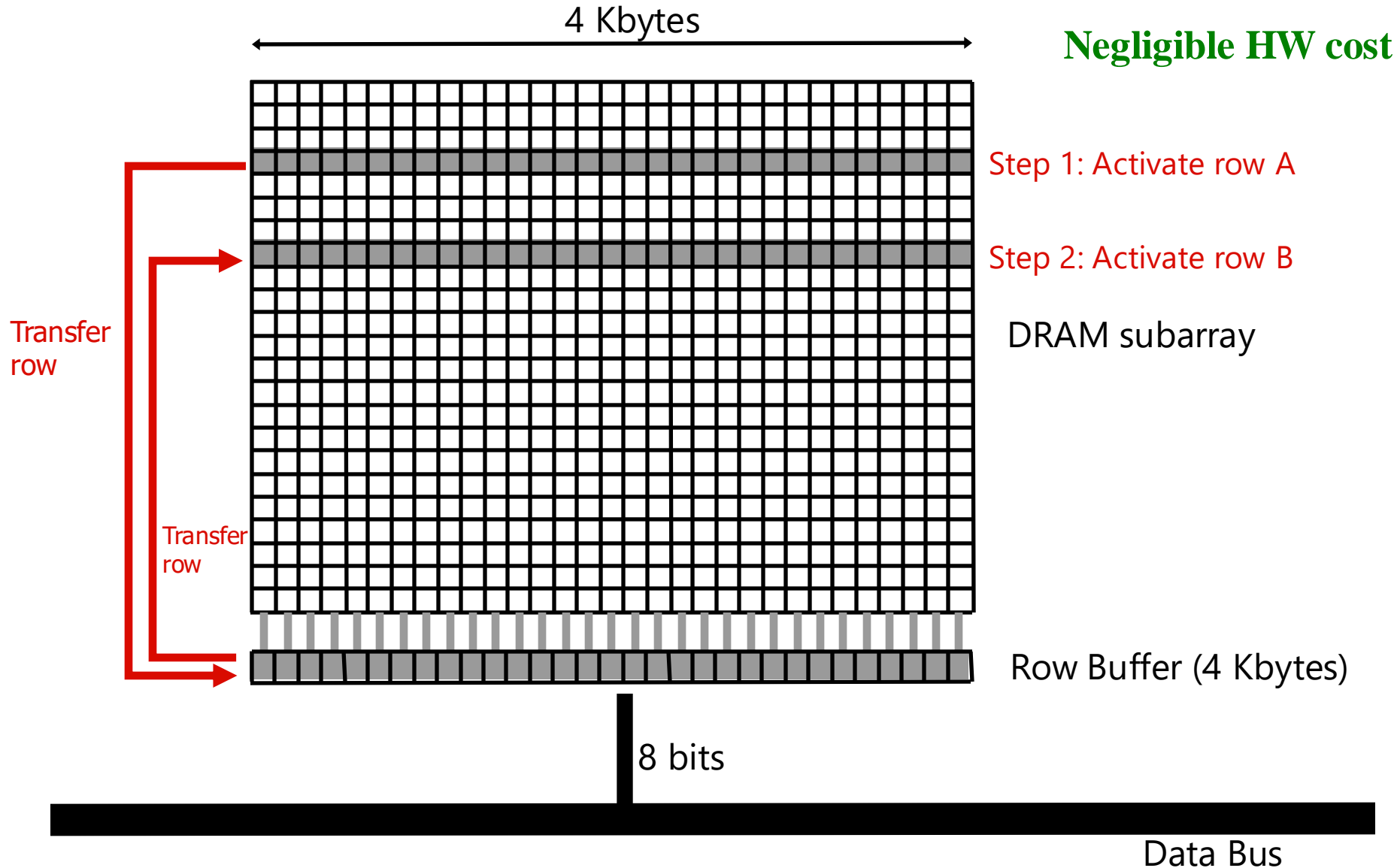


1046ns, 3.6uJ → 90ns, 0.04uJ

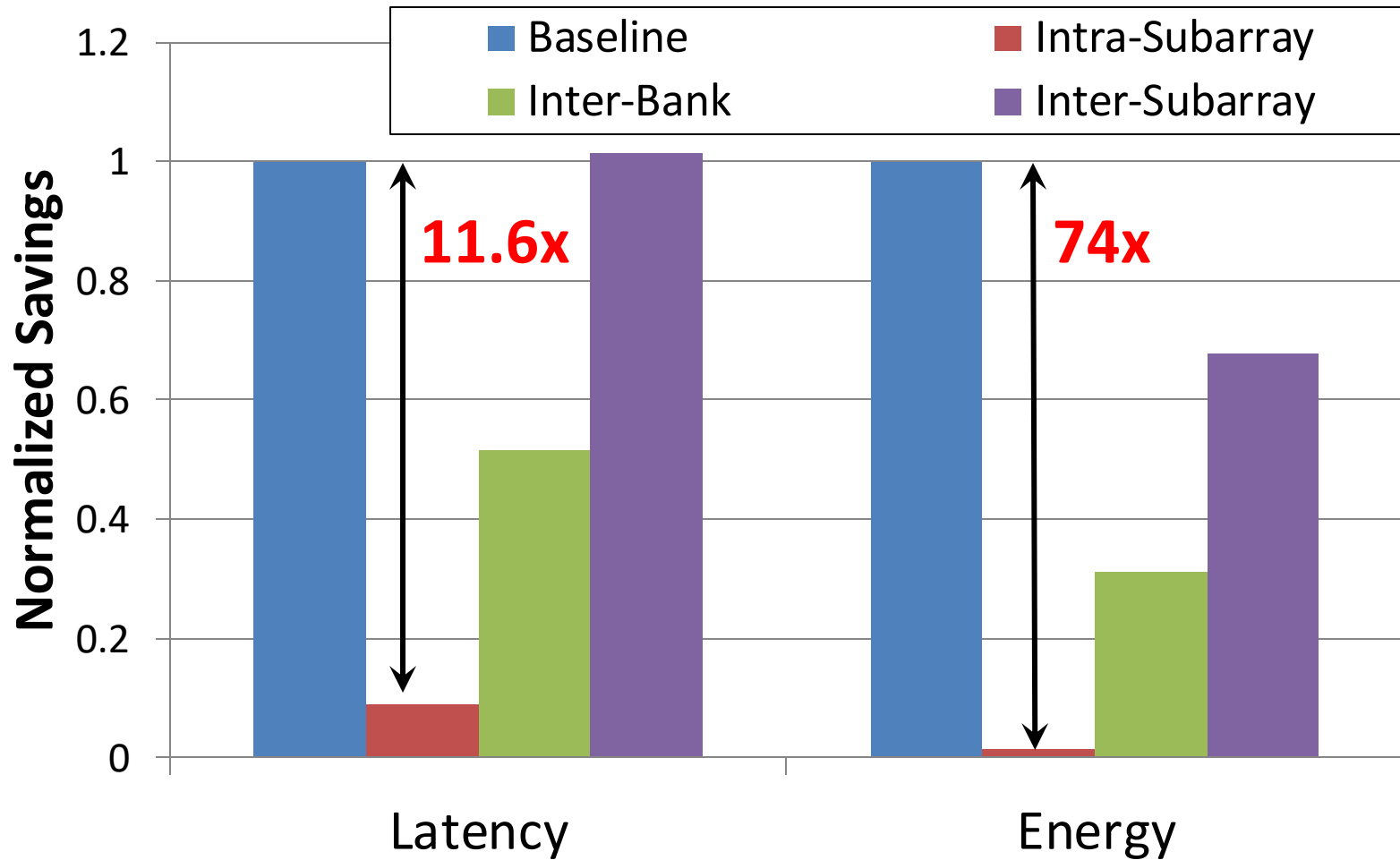
RowClone: In-DRAM Row Copy

Idea: Two consecutive ACTivates

Negligible HW cost



RowClone: Latency and Energy Savings



Seshadri et al., "RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.

More on RowClone

- Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry,
"RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization"
Proceedings of the 46th International Symposium on Microarchitecture (MICRO), Davis, CA, December 2013. [[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Session Slides \(pptx\)](#)] [[pdf](#)] [[Poster \(pptx\)](#)] [[pdf](#)]

RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization

Vivek Seshadri Yoongu Kim Chris Fallin* Donghyuk Lee
vseshadr@cs.cmu.edu yoongukim@cmu.edu cfallin@c1f.net donghyuk1@cmu.edu

Rachata Ausavarungnirun Gennady Pekhimenko Yixin Luo
rachata@cmu.edu gpekhime@cs.cmu.edu yixinluo@andrew.cmu.edu

Onur Mutlu Phillip B. Gibbons† Michael A. Kozuch† Todd C. Mowry
onur@cmu.edu phillip.b.gibbons@intel.com michael.a.kozuch@intel.com tcm@cs.cmu.edu

Carnegie Mellon University †Intel Pittsburgh

RowClone in Off-the-Shelf DRAM Chips

- Idea: Violate DRAM timing parameters to mimic RowClone

ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAMs

Fei Gao

feig@princeton.edu

Department of Electrical Engineering
Princeton University

Georgios Tziantzioulis

georgios.tziantzioulis@princeton.edu

Department of Electrical Engineering
Princeton University

David Wentzlaff

wentzlaf@princeton.edu

Department of Electrical Engineering
Princeton University

Real Processing Using Memory Prototype

- End-to-end RowClone & TRNG using off-the-shelf DRAM chips
- Idea: Violate DRAM timing parameters to mimic RowClone

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun^{§†}

Juan Gómez Luna[§]
Hasan Hassan[§]

Konstantinos Kanellopoulos[§]
Oğuz Ergin[†] Onur Mutlu[§]

Behzad Salami^{§*}

[§]ETH Zürich

[†]TOBB ETÜ

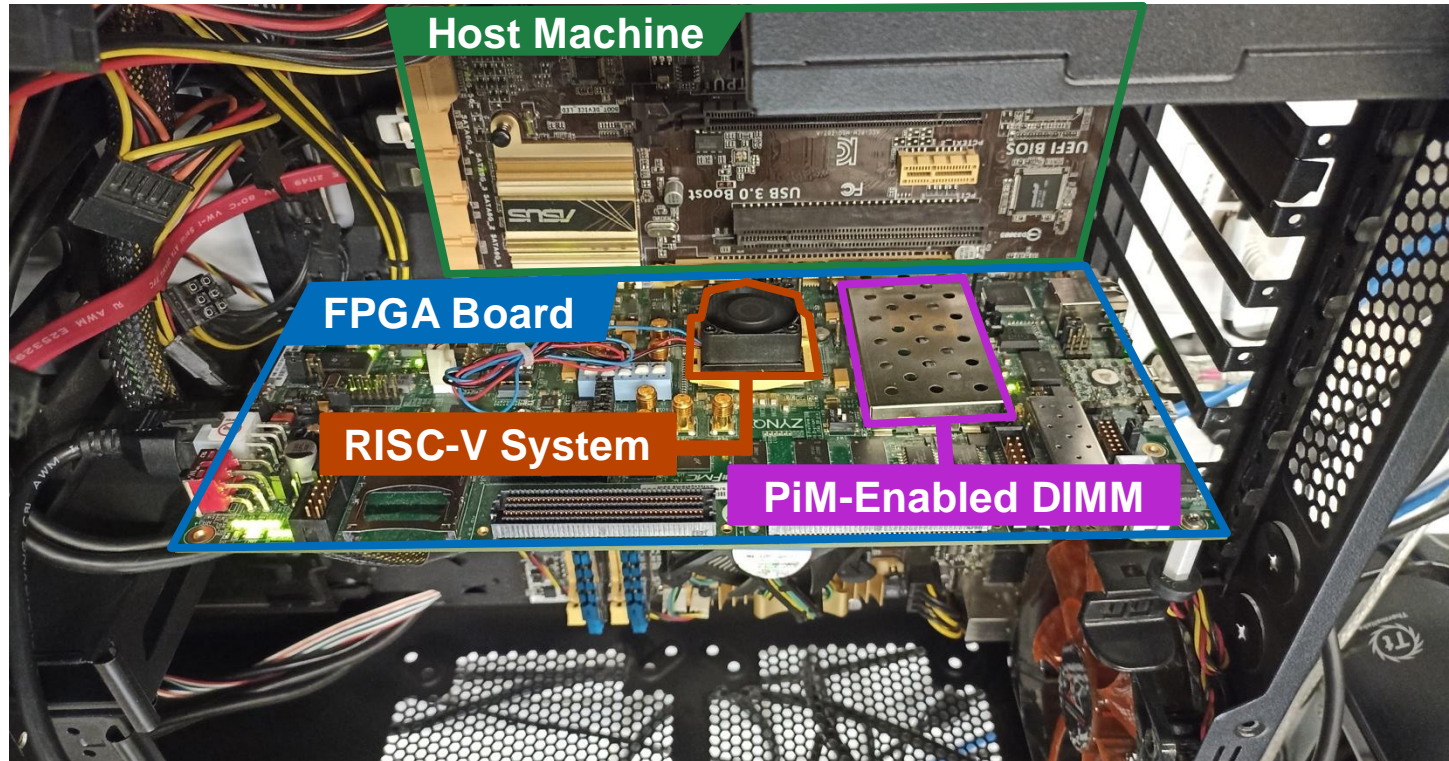
^{*}BSC

<https://arxiv.org/pdf/2111.00082.pdf>

<https://github.com/cmu-safari/pidram>

<https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s>

Real Processing-using-Memory Prototype



<https://arxiv.org/pdf/2111.00082.pdf>

<https://github.com/cmu-safari/pidram>

<https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s>

Real Processing-using-Memory Prototype

☰ README.md

Building a PiDRAM Prototype

To build PiDRAM's prototype on Xilinx ZC706 boards, developers need to use the two sub-projects in this directory. `fpga-zynq` is a repository branched off of [UCB-BAR's fpga-zynq](#) repository. We use `fpga-zynq` to generate rocket chip designs that support end-to-end DRAM PuM execution. `controller-hardware` is where we keep the main Vivado project and Verilog sources for PiDRAM's memory controller and the top level system design.

Rebuilding Steps

1. Navigate into `fpga-zynq` and read the README file to understand the overall workflow of the repository
 - Follow the readme in `fpga-zynq/rocket-chip/riscv-tools` to install dependencies
2. Create the Verilog source of the rocket chip design using the `ZynqCopyFPGAConfig`
 - Navigate into `zc706`, then run `make rocket CONFIG=ZynqCopyFPGAConfig -j<number of cores>`
3. Copy the generated Verilog file (should be under `zc706/src`) and overwrite the same file in `controller-hardware/source/hdl/impl/rocket-chip`
4. Open the Vivado project in `controller-hardware/Vivado_Project` using Vivado 2016.2
5. Generate a bitstream
6. Copy the bitstream (`system_top.bit`) to `fpga-zynq/zc706`
7. Use the `./build_script.sh` to generate the new `boot.bin` under `fpga-images-zc706`, you can use this file to program the FPGA using the SD-Card
 - For details, follow the relevant instructions in `fpga-zynq/README.md`

You can run programs compiled with the RISC-V Toolchain supplied within the `fpga-zynq` repository. To install the toolchain, follow the instructions under `fpga-zynq/rocket-chip/riscv-tools`.

Generating DDR3 Controller IP sources

We cannot provide the sources for the Xilinx PHY IP we use in PiDRAM's memory controller due to licensing issues. We describe here how to regenerate them using Vivado 2016.2. First, you need to generate the IP RTL files:

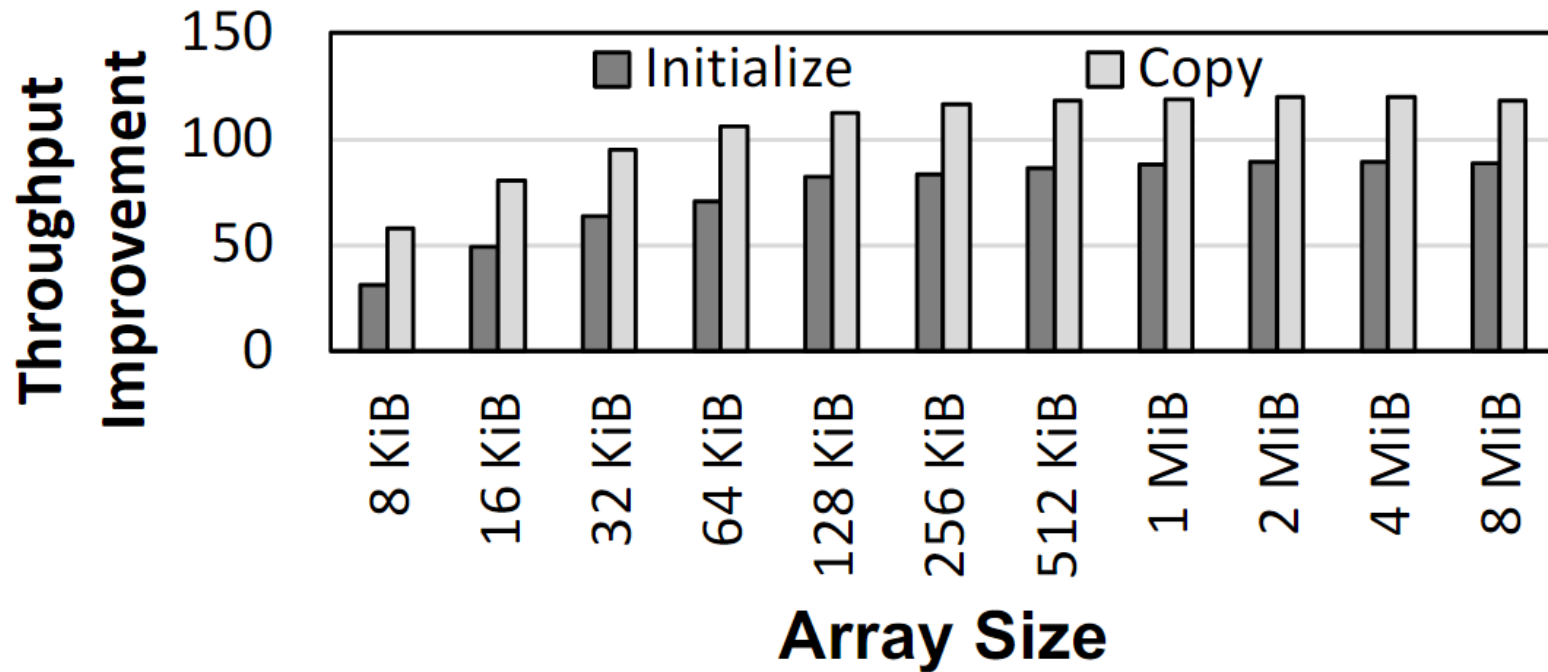
- 1- Open IP Catalog
- 2- Find "Memory Interface Generator (MIG 7 Series)" IP and double click

<https://arxiv.org/pdf/2111.00082.pdf>

<https://github.com/cmu-safari/pidram>

<https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s>

Microbenchmark Copy/Initialization Throughput



**In-DRAM Copy and Initialization
improve throughput by 119x and 89x**

More on PiDRAM

- Ataberk Olgun, Juan Gomez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oguz Ergin, and Onur Mutlu,
["PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM"](#)
[ACM Transactions on Architecture and Code Optimization \(TACO\)](#), March 2023.
[\[arXiv version\]](#)
Presented at the [18th HiPEAC Conference](#), Toulouse, France, January 2023.
[\[Slides \(pptx\) \(pdf\)\]](#)
[\[Longer Lecture Slides \(pptx\) \(pdf\)\]](#)
[\[Lecture Video \(40 minutes\)\]](#)
[\[PiDRAM Source Code\]](#)

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]ETH Zürich

[†]TOBB University of Economics and Technology

RowClone Extensions and Follow-Up Work

- Can this be improved to do faster inter-subarray copy?
 - Yes, see LISA [Chang et al., HPCA 2016]
- Can we enable data movement at smaller granularities within a bank?
 - Yes, see FIGARO [Wang et al., MICRO 2020]
- Can this be improved to do better inter-bank copy?
 - Yes, see Network-on-Memory [CAL 2020]
- Can similar ideas and DRAM properties be used to perform computation on data?
 - Yes, see Ambit [Seshadri et al., CAL 2015, MICRO 2017]

LISA: Increasing Connectivity in DRAM

- Kevin K. Chang, Prashant J. Nair, Saugata Ghose, Donghyuk Lee, Moinuddin K. Qureshi, and Onur Mutlu,
"Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM"
Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA), Barcelona, Spain, March 2016.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Source Code](#)]

Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM

Kevin K. Chang[†], Prashant J. Nair^{*}, Donghyuk Lee[†], Saugata Ghose[†], Moinuddin K. Qureshi^{*}, and Onur Mutlu[†]

[†]Carnegie Mellon University ^{*}Georgia Institute of Technology

FIGARO: Fine-Grained In-DRAM Copy

- Yaohua Wang, Lois Orosa, Xiangjun Peng, Yang Guo, Saugata Ghose, Minesh Patel, Jeremie S. Kim, Juan Gómez Luna, Mohammad Sadrosadati, Nika Mansouri Ghiasi, and Onur Mutlu,
"FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching"
Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.

FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching

Yaohua Wang^{*} Lois Orosa[†] Xiangjun Peng^{⊙*} Yang Guo^{*} Saugata Ghose^{◇‡} Minesh Patel[†]
Jeremie S. Kim[†] Juan Gómez Luna[†] Mohammad Sadrosadati[§] Nika Mansouri Ghiasi[†] Onur Mutlu^{†‡}

^{*}National University of Defense Technology [†]ETH Zürich [⊙]Chinese University of Hong Kong

[◇]University of Illinois at Urbana–Champaign [‡]Carnegie Mellon University [§]Institute of Research in Fundamental Sciences

Network-On-Memory: Fast Inter-Bank Copy

- Seyyed Hossein SeyyedAghaei Rezaei, Mehdi Modarressi, Rachata Ausavarungnirun, Mohammad Sadrosadati, Onur Mutlu, and Masoud Daneshtalab,
"NoM: Network-on-Memory for Inter-Bank Data Transfer in Highly-Banked Memories"
IEEE Computer Architecture Letters (**CAL**), to appear in 2020.

NOm: NETWORK-ON-MEMORY FOR INTER-BANK DATA TRANSFER IN HIGHLY-BANKED MEMORIES

Seyyed Hossein SeyyedAghaei Rezaei¹
Mohammad Sadrosadati³

Mehdi Modarressi^{1,3}
Onur Mutlu⁴

Rachata Ausavarungnirun²
Masoud Daneshtalab⁵

¹University of Tehran

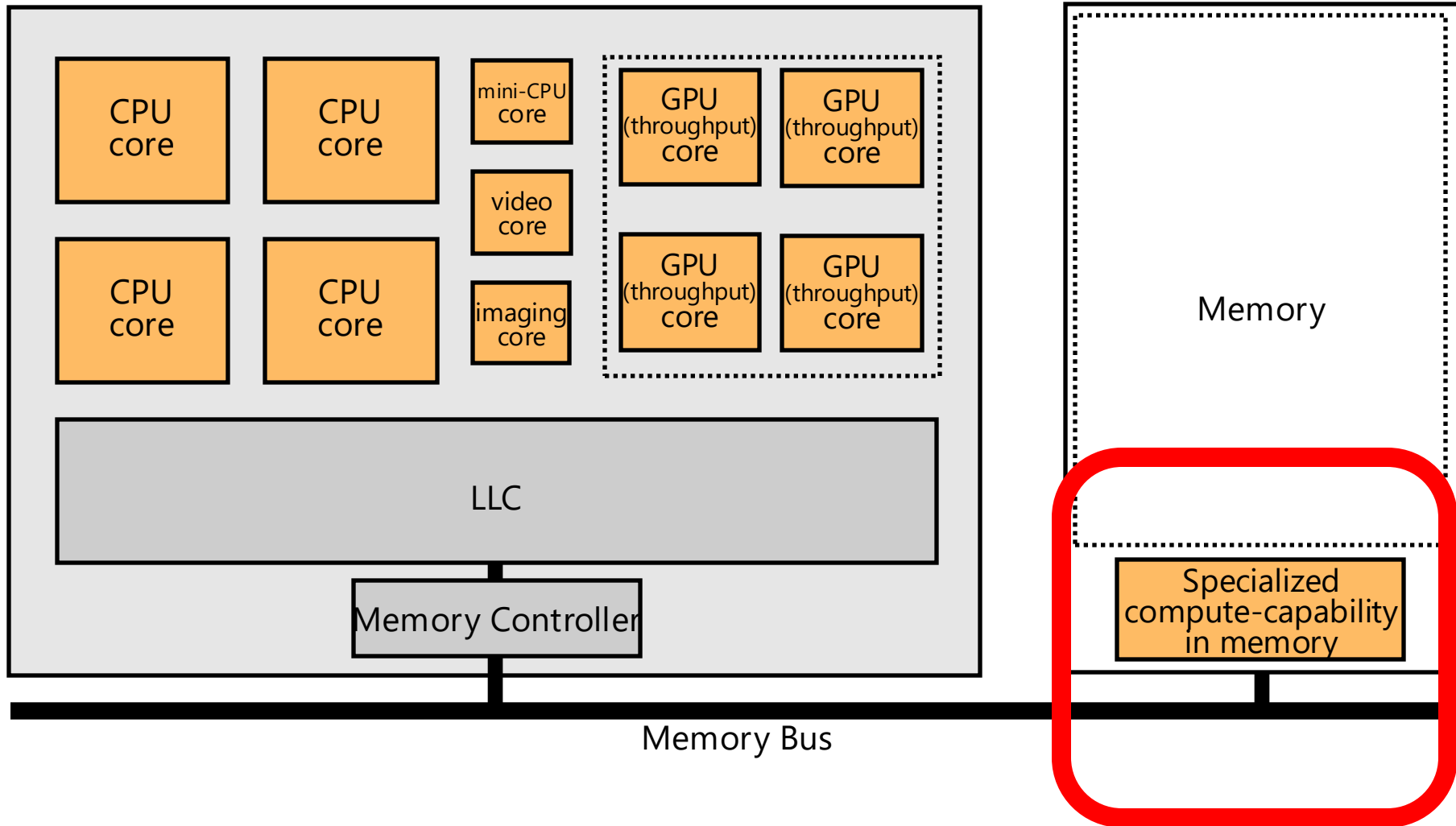
²King Mongkut's University of Technology North Bangkok

³Institute for Research in Fundamental Sciences

⁴ETH Zürich

⁵Mälardalens University

Mindset: Memory as an Accelerator



Memory similar to a "conventional" accelerator

Lecture on RowClone & Processing using DRAM

Mindset: Memory as an Accelerator

The diagram illustrates a system architecture where memory is treated as an accelerator. On the left, a large gray box represents the system, containing several orange boxes for different processing units: four 'CPU core' boxes, one 'mini-CPU core', one 'video core', one 'imaging core', and four 'GPU (throughput) core' boxes. Below these is a large gray box labeled 'LLC' (Last Level Cache), and at the bottom is a 'Memory Controller'. A thick black line labeled 'Memory Bus' runs horizontally across the bottom. To the right of the system box is a large white box labeled 'Memory'. Inside the 'Memory' box, at the bottom, is a smaller orange box labeled 'Specialized compute-capability in memory', which is highlighted by a red rounded rectangle. A video player interface is overlaid on the bottom of the slide, showing a red progress bar and the text 'Memory similar to a "conventional" accelerator'.

Onur Mutlu

Memory

Specialized compute-capability in memory

Memory Bus

Memory similar to a "conventional" accelerator

DEPARTMENT OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING (D-ITET)

Seminar in Computer Arch. - Meeting 3: RowClone: In-Memory Data Copy and Initialization (Fall 2021)

292 views • Streamed live on Oct 7, 2021

21 0 SHARE SAVE ...



Onur Mutlu Lectures
19.1K subscribers

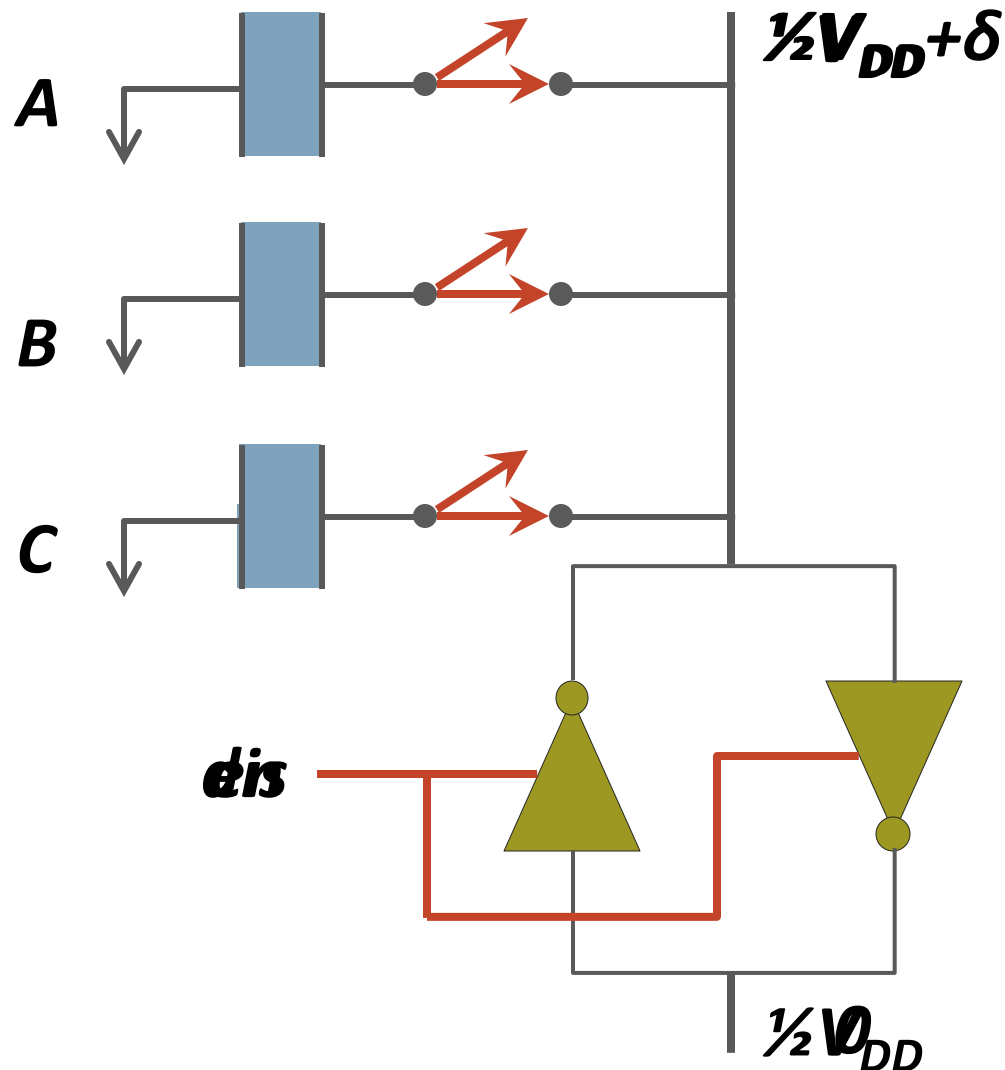
SUBSCRIBED



(Truly) In-Memory Computation

- We can support in-DRAM AND, OR, NOT, MAJ
- At low cost
- Using analog computation capability of DRAM
 - Idea: activating multiple rows performs computation
- 30-60X performance and energy improvement
 - Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO 2017.
- New memory technologies enable even more opportunities
 - Memristors, resistive RAM, phase change mem, STT-MRAM, ...
 - Can operate on data with minimal movement

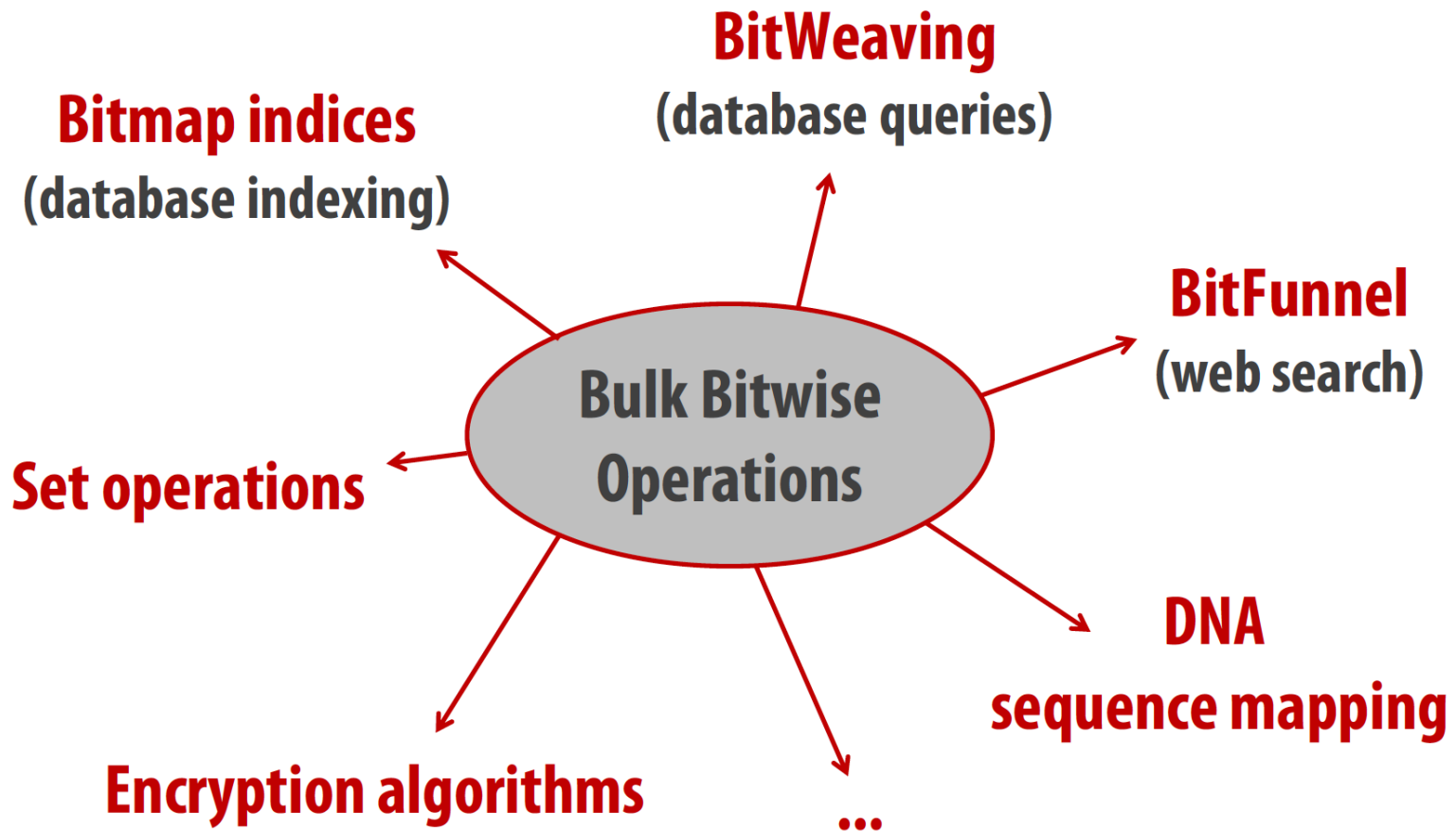
In-DRAM AND/OR: Triple Row Activation



Final State
 $AB + BC + AC$

$C(A + B) + \sim C(AB)$

Bulk Bitwise Operations in Workloads



In-DRAM Acceleration of Database Queries

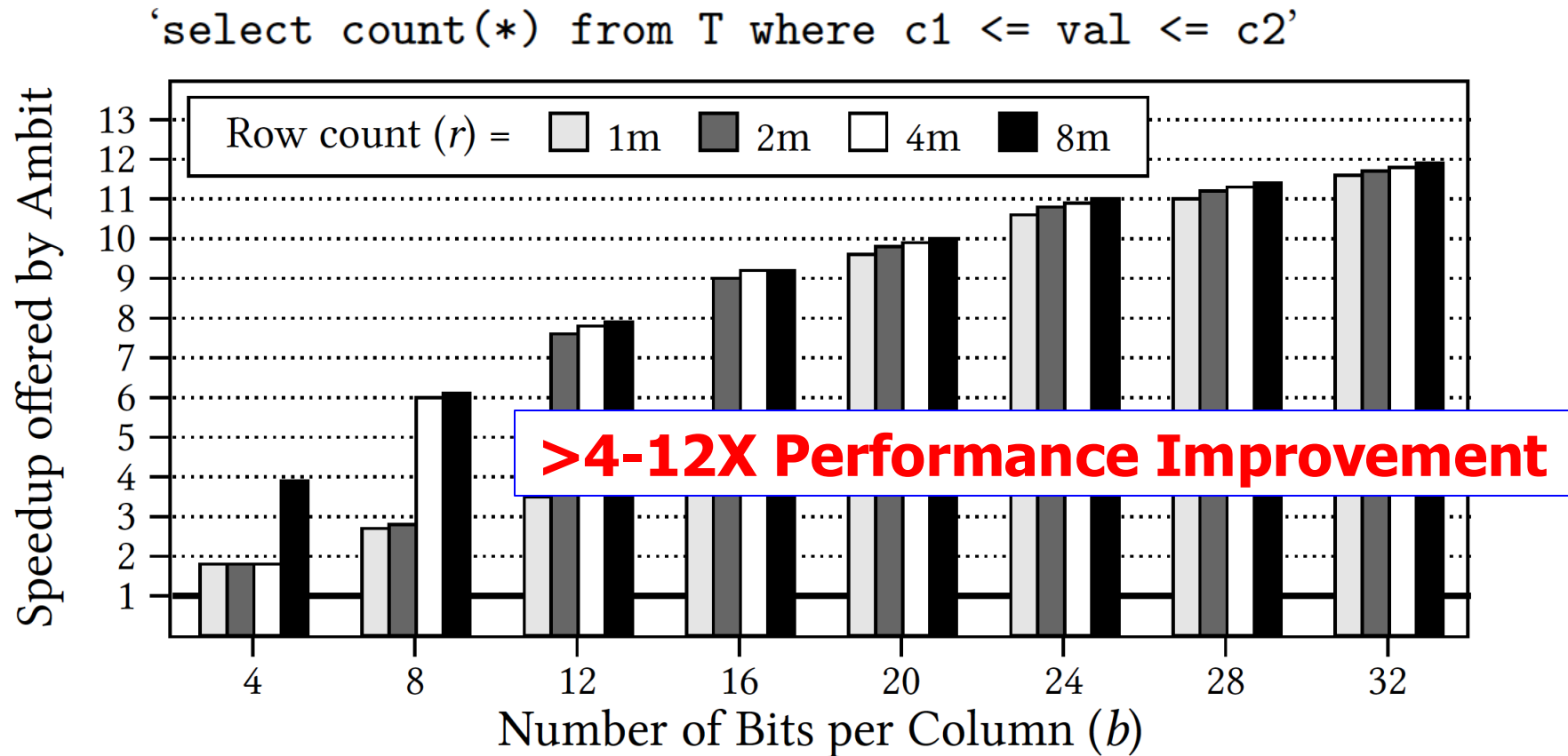


Figure 11: Speedup offered by Ambit over baseline CPU with SIMD for BitWeaving

Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations using Commodity DRAM Technology," MICRO 2017.

More on In-DRAM Bulk AND/OR

- Vivek Seshadri, Kevin Hsieh, Amirali Boroumand, Donghyuk Lee, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,
["Fast Bulk Bitwise AND and OR in DRAM"](#)
[IEEE Computer Architecture Letters](#) (**CAL**), April 2015.

Fast Bulk Bitwise AND and OR in DRAM

Vivek Seshadri*, Kevin Hsieh*, Amirali Boroumand*, Donghyuk Lee*,
Michael A. Kozuch†, Onur Mutlu*, Phillip B. Gibbons†, Todd C. Mowry*

*Carnegie Mellon University

†Intel Pittsburgh

More on Ambit

- Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,
["Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology"](#)
Proceedings of the 50th International Symposium on Microarchitecture (MICRO), Boston, MA, USA, October 2017.
[\[Slides \(pptx\) \(pdf\)\]](#) [\[Lightning Session Slides \(pptx\) \(pdf\)\]](#) [\[Poster \(pptx\) \(pdf\)\]](#)

Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology

Vivek Seshadri^{1,5} Donghyuk Lee^{2,5} Thomas Mullins^{3,5} Hasan Hassan⁴ Amirali Boroumand⁵
Jeremie Kim^{4,5} Michael A. Kozuch³ Onur Mutlu^{4,5} Phillip B. Gibbons⁵ Todd C. Mowry⁵

¹Microsoft Research India ²NVIDIA Research ³Intel ⁴ETH Zürich ⁵Carnegie Mellon University

In-DRAM Bulk Bitwise Execution

- Vivek Seshadri and Onur Mutlu,
"In-DRAM Bulk Bitwise Execution Engine"
Invited Book Chapter in Advances in Computers, to appear
in 2020.
[[Preliminary arXiv version](#)]

In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri
Microsoft Research India
visesha@microsoft.com

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch

SIMDRAM Framework

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu, **"SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM"** *Proceedings of the 26th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Virtual, March-April 2021.
[[2-page Extended Abstract](#)]
[[Short Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Short Talk Video](#) (5 mins)]
[[Full Talk Video](#) (27 mins)]

SIMDRAM: A Framework for Bit-Serial SIMD Processing using DRAM

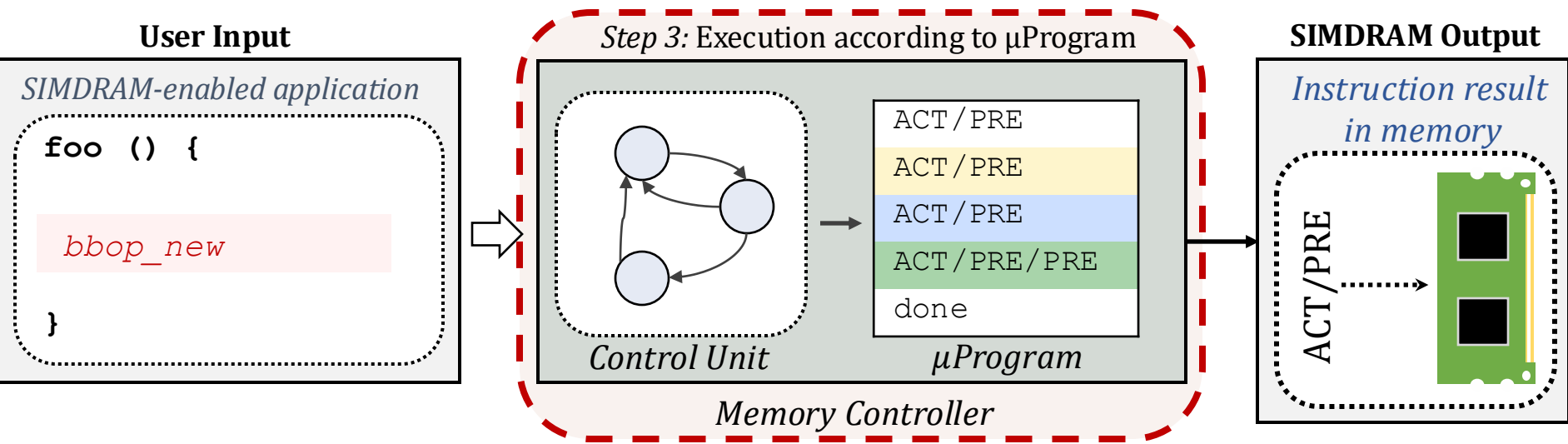
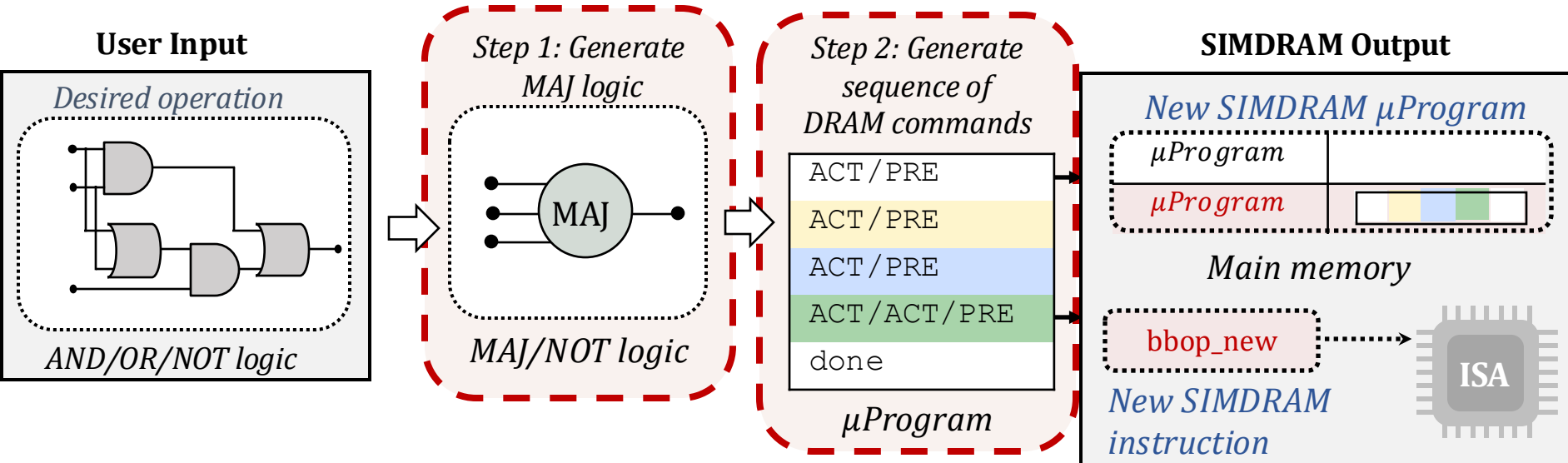
*Nastaran Hajinazar ^{1,2}	*Geraldo F. Oliveira ¹	Sven Gregorio ¹	João Dinis Ferreira ¹
Nika Mansouri Ghiasi ¹	Minesh Patel ¹	Mohammed Alser ¹	Saugata Ghose ³
	Juan Gómez-Luna ¹	Onur Mutlu ¹	

¹ETH Zürich

²Simon Fraser University

³University of Illinois at Urbana–Champaign

SIMDRAM Framework: Overview



SIMDRAM Key Results

Evaluated on:

- 16 complex in-DRAM operations
- 7 commonly-used real-world applications

SIMDRAM provides:

- **88×** and **5.8×** the **throughput** of a **CPU** and a **high-end GPU**, respectively, over **16 operations**
- **257×** and **31×** the **energy efficiency** of a **CPU** and a **high-end GPU**, respectively, over **16 operations**
- **21×** and **2.1×** the **performance** of a **CPU** and a **high-end GPU**, over **seven real-world applications**

SAFARI

More on SIMD RAM

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu, [**"SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM"**](#) *Proceedings of the 26th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Virtual, March-April 2021.
[[2-page Extended Abstract](#)]
[[Short Talk Slides \(pptx\)](#)] ([pdf](#))
[[Talk Slides \(pptx\)](#)] ([pdf](#))
[[Short Talk Video](#) (5 mins)]
[[Full Talk Video](#) (27 mins)]

SIMDRAM: A Framework for Bit-Serial SIMD Processing using DRAM

*Nastaran Hajinazar ^{1,2}	*Geraldo F. Oliveira ¹	Sven Gregorio ¹	João Dinis Ferreira ¹
Nika Mansouri Ghiasi ¹	Minesh Patel ¹	Mohammed Alser ¹	Saugata Ghose ³
	Juan Gómez-Luna ¹	Onur Mutlu ¹	

¹ETH Zürich

²Simon Fraser University

³University of Illinois at Urbana–Champaign

MIMDRAM: More Flexible Processing using DRAM

- **Appears at HPCA 2024** <https://arxiv.org/pdf/2402.19080.pdf>

MIMDRAM: An End-to-End Processing-Using-DRAM System for High-Throughput, Energy-Efficient and Programmer-Transparent Multiple-Instruction Multiple-Data Computing

Geraldo F. Oliveira[†] Ataberk Olgun[†] Abdullah Giray Yağlıkçı[†] F. Nisa Bostancı[†]
Juan Gómez-Luna[†] Saugata Ghose[‡] Onur Mutlu[†]

[†] *ETH Zürich*

[‡] *Univ. of Illinois Urbana-Champaign*

*Our **goal** is to design a flexible PUD system that overcomes the limitations caused by the large and rigid granularity of PUD. To this end, we propose MIMDRAM, a hardware/software co-designed PUD system that introduces new mechanisms to allocate and control only the necessary resources for a given PUD operation. The key idea of MIMDRAM is to leverage fine-grained DRAM (i.e., the ability to independently access smaller segments of a large DRAM row) for PUD computation. MIMDRAM exploits this key idea to enable a multiple-instruction multiple-data (MIMD) execution model in each DRAM subarray (and SIMD execution within each DRAM row segment).*

MIMDRAM: Executive Summary

Problem: Processing-Using-DRAM (PUD) suffers from three issues caused by DRAM's large and rigid access granularity

- Underutilization due to data parallelism variation in (and across) applications
- Limited computation support due to a lack of interconnects
- Challenging programming model due to a lack of compilers

Goal: Design a flexible PUD system that overcomes the three limitations caused by DRAM's large and rigid access granularity

Key Mechanism: MIMDRAM, a hardware/software co-design PUD system

- **Key idea:** leverage fine-grained DRAM for PUD operation
- **HW:** - simple changes to the DRAM array, enabling concurrent PUD operations
- low-cost interconnects at the DRAM peripherals for data reduction
- **SW:** - compiler and OS support to generate and map PUD instructions

Key Results: MIMDRAM achieves

- **14.3x, 30.6x, and 6.8x** the energy efficiency of state-of-the-art PUD systems, a high-end CPU and GPU, respectively
- Small area cost to a DRAM chip (**1.11%**) and CPU die (**0.6%**)

In-DRAM Lookup-Table Based Execution

João Dinis Ferreira, Gabriel Falcao, Juan Gómez-Luna, Mohammed Alser, Lois Orosa, Mohammad Sadrosadati, Jeremie S. Kim, Geraldo F. Oliveira, Taha Shahroodi, Anant Nori, and Onur Mutlu,

"pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables"

Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Longer Lecture Slides \(pptx\)](#) ([pdf](#))]

[[Lecture Video](#) (26 minutes)]

[[arXiv version](#)]

[[Source Code](#) (Officially Artifact Evaluated with All Badges)]

Officially artifact evaluated as available, reusable and reproducible.



pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables

João Dinis Ferreira[§]

Gabriel Falcao[†]

Juan Gómez-Luna[§]

Mohammed Alser[§]

Lois Orosa^{§∇}

Mohammad Sadrosadati[§]

Jeremie S. Kim[§]

Geraldo F. Oliveira[§]

Taha Shahroodi[‡]

Anant Nori^{*}

Onur Mutlu[§]

[§]ETH Zürich

[†]IT, University of Coimbra

[∇]Galicia Supercomputing Center

[‡]TU Delft

^{*}Intel

In-DRAM Physical Unclonable Functions

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu,
"The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices"
Proceedings of the 24th International Symposium on High-Performance Computer Architecture (HPCA), Vienna, Austria, February 2018.
[[Lightning Talk Video](#)]
[[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Session Slides \(pptx\)](#)] [[pdf](#)]
[[Full Talk Lecture Video](#) (28 minutes)]

The DRAM Latency PUF:

Quickly Evaluating Physical Unclonable Functions

by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim^{†§}

Minesh Patel[§]

Hasan Hassan[§]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[§]ETH Zürich

In-DRAM True Random Number Generation

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, Lois Orosa, and Onur Mutlu,
"D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput"
Proceedings of the 25th International Symposium on High-Performance Computer Architecture (HPCA), Washington, DC, USA, February 2019.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Full Talk Video](#) (21 minutes)]
[[Full Talk Lecture Video](#) (27 minutes)]
Top Picks Honorable Mention by IEEE Micro.

D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim^{‡§} Minesh Patel[§] Hasan Hassan[§] Lois Orosa[§] Onur Mutlu^{§‡}
[‡]Carnegie Mellon University [§]ETH Zürich

In-DRAM True Random Number Generation

- Ataberk Olgun, Minesh Patel, A. Giray Yaglikci, Haocong Luo, Jeremie S. Kim, F. Nisa Bostanci, Nandita Vijaykumar, Oguz Ergin, and Onur Mutlu,
"QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips"
Proceedings of the 48th International Symposium on Computer Architecture (ISCA), Virtual, June 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Short Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (25 minutes)]
[[SAFARI Live Seminar Video](#) (1 hr 26 mins)]

QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips

Ataberk Olgun^{§†}

Minesh Patel[§]

A. Giray Yağlıkçı[§]

Haocong Luo[§]

Jeremie S. Kim[§]

F. Nisa Bostancı^{§†}

Nandita Vijaykumar^{§⊙}

Oğuz Ergin[†]

Onur Mutlu[§]

[§]ETH Zürich

[†]TOBB University of Economics and Technology

[⊙]University of Toronto

In-DRAM True Random Number Generation

- F. Nisa Bostanci, Ataberk Olgun, Lois Orosa, A. Giray Yaglikci, Jeremie S. Kim, Hasan Hassan, Oguz Ergin, and Onur Mutlu,

"DR-STRaNGe: End-to-End System Design for DRAM-based True Random Number Generators"

Proceedings of the 28th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, April 2022.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Short Talk Slides \(pptx\)](#) ([pdf](#))]

DR-STRaNGe: End-to-End System Design for DRAM-based True Random Number Generators

F. Nisa Bostanci^{†§}

Ataberk Olgun^{†§}

Lois Orosa[§]

A. Giray Yağlıkçı[§]

Jeremie S. Kim[§]

Hasan Hassan[§]

Oğuz Ergin[†]

Onur Mutlu[§]

[†]*TOBB University of Economics and Technology*

[§]*ETH Zürich*

DRAM Chips Are Already (Quite) Capable!

- **Appears at HPCA 2024** <https://arxiv.org/pdf/2402.18736.pdf>

Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel Yahya Can Tuğrul Ataberk Olgun F. Nisa Bostancı A. Giray Yağlıkçı
Geraldo F. Oliveira Haocong Luo Juan Gómez-Luna Mohammad Sadrosadati Onur Mutlu

ETH Zürich

We experimentally demonstrate that COTS DRAM chips are capable of performing 1) functionally-complete Boolean operations: NOT, NAND, and NOR and 2) many-input (i.e., more than two-input) AND and OR operations. We present an extensive characterization of new bulk bitwise operations in 256 off-the-shelf modern DDR4 DRAM chips. We evaluate the reliability of these operations using a metric called success rate: the fraction of correctly performed bitwise operations. Among our 19 new observations, we highlight four major results. First, we can perform the NOT operation on COTS DRAM chips with 98.37% success rate on average. Second, we can perform up to 16-input NAND, NOR, AND, and OR operations on COTS DRAM chips with high reliability (e.g., 16-input NAND, NOR, AND, and OR with average success rate of 94.94%, 95.87%, 94.94%, and 95.85%, respectively). Third, data pattern only slightly

DRAM Chips Are Already (Quite) Capable!

- <https://arxiv.org/pdf/2312.02880.pdf>

PULSAR: Simultaneous Many-Row Activation for Reliable and High-Performance Computing in Off-the-Shelf DRAM Chips

Ismail Emir Yuksel Yahya Can Tugrul F. Nisa Bostanci Abdullah Giray Yaglikci Ataberk Olgun
Geraldo F. Oliveira Melina Soysal Haocong Luo Juan Gomez Luna Mohammad Sadrosadati
Onur Mutlu

ETH Zurich

We propose PULSAR, a new technique to enable high-success-rate and high-performance PuM operations in off-the-shelf DRAM chips. PULSAR leverages our new observation that a carefully-crafted sequence of DRAM commands simultaneously activates up to 32 DRAM rows. PULSAR overcomes the limitations of existing techniques by 1) replicating the input data to improve the success rate and 2) enabling new bulk bitwise operations (e.g., many-input majority, *Multi-RowInit*, and *Bulk-Write*) to improve the performance.

DRAM Chips Are Already (Quite) Capable!

- **Appears at DSN 2024**



Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel¹ Yahya Can Tuğrul^{1,2} F. Nisa Bostancı¹ Geraldo F. Oliveira¹
A. Giray Yağlıkçı¹ Ataberk Olgun¹ Melina Soysal¹ Haocong Luo¹
Juan Gómez-Luna¹ Mohammad Sadrosadati¹ Onur Mutlu¹

¹*ETH Zürich* ²*TOBB University of Economics and Technology*

The Capability of COTS DRAM Chips

We demonstrate that COTS DRAM chips:

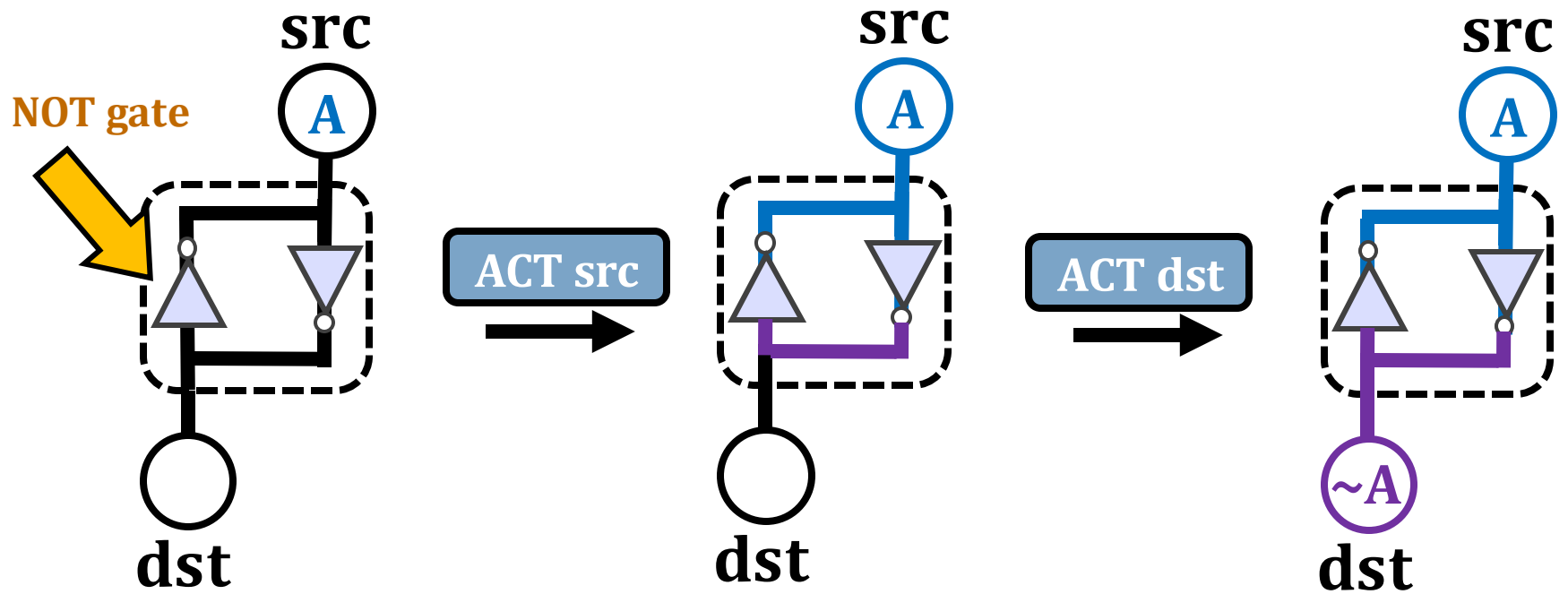
1 Can simultaneously activate up to 48 rows in two neighboring subarrays

2 Can perform **NOT operation** with up to **32 output operands**

3 Can perform up to **16-input AND, NAND, OR, and NOR** operations

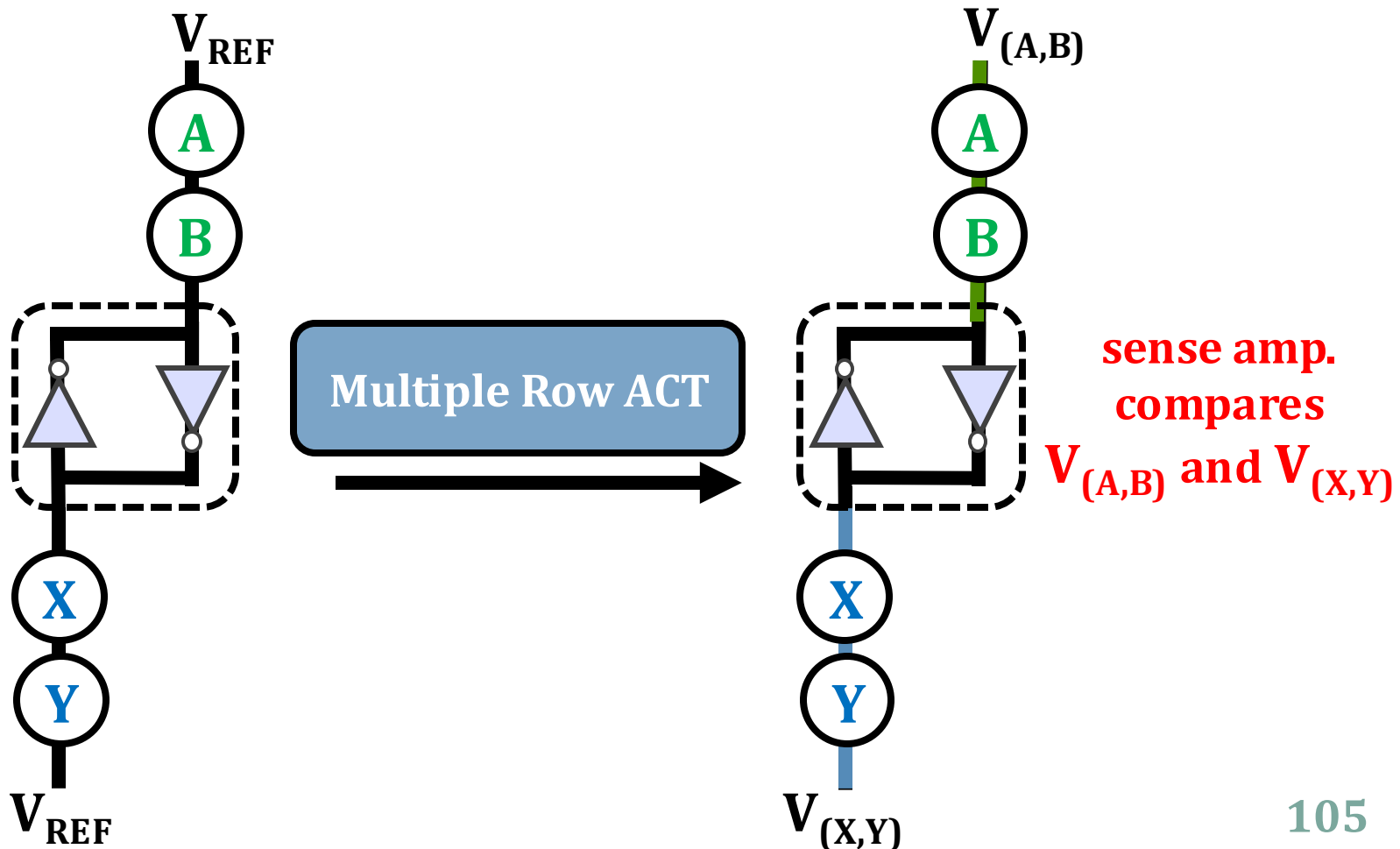
Key Idea: NOT Operation

Connect rows in neighboring subarrays through a **NOT gate** by simultaneously activating rows



Key Idea: NAND, NOR, AND, OR

Manipulate the bitline voltage to express
a wide variety of functions using
multiple-row activation in neighboring subarrays



FC-DRAM: Executive Summary

- **Motivation: Processing-using-DRAM** can alleviate the performance and energy bottlenecks caused by **data movement**
 - **Prior works** show that **existing DRAM chips** can perform **three-input majority** and **two-input AND and OR** operations
- **Problem: Proof-of-concept** demonstrations on **commercial off-the-shelf (COTS)** DRAM chips do not provide
 - **functionally-complete** operations (e.g., NAND or NOR)
 - **NOT operation**
 - **AND** and **OR** operations with **more than two inputs**
- **Experimental Study: 256 DDR4 chips** from **two major manufacturers**
- **Key Results:**
 - COTS DRAM chips can perform **NOT** and **{2, 4, 8, 16}-input AND, NAND, OR, and NOR** operations with **very high reliability (>94% success rate)**
 - **Data pattern** and **temperature** only slightly affect the reliability of these operations (**<1.98% decrease in success rate**)

In-Flash Bulk Bitwise Execution

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungsuk Kim, and Onur Mutlu, **"Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory"**
Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Longer Lecture Slides \(pptx\)](#)] [[pdf](#)]
[[Lecture Video](#) (44 minutes)]
[[arXiv version](#)]

Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

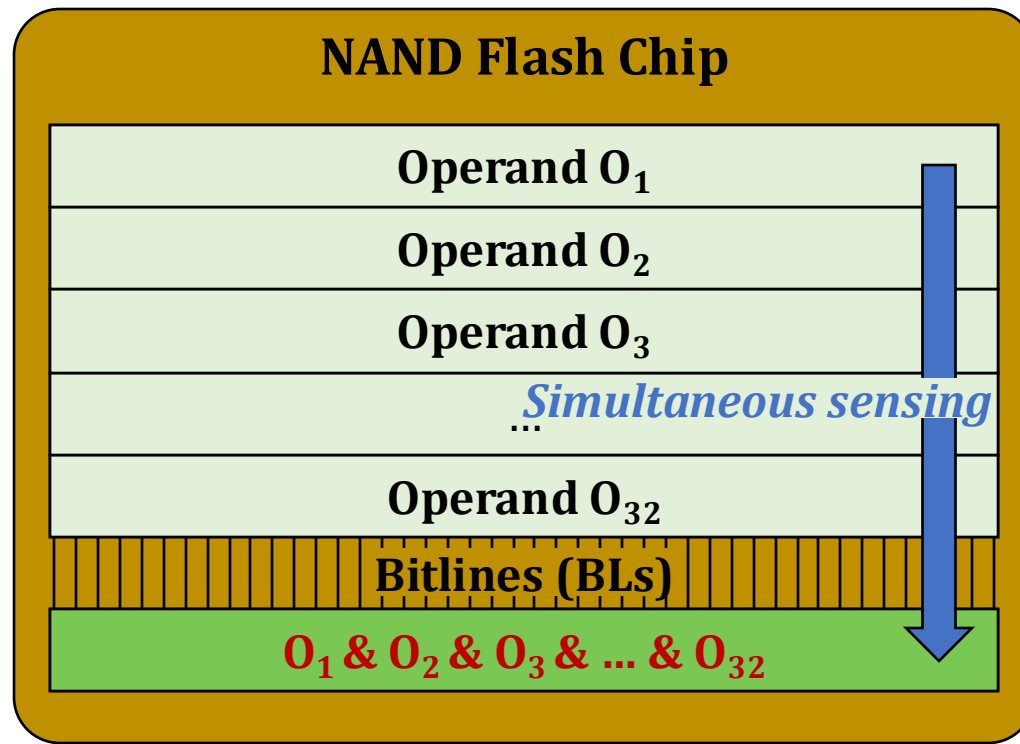
Jisung Park^{§∇} Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]ETH Zürich [∇]POSTECH [†]LIRMM, Univ. Montpellier, CNRS [‡]Kyungpook National University

Flash-Cosmos: Basic Ideas

■ Flash-Cosmos enables

- Computation on multiple operands with a single sensing operation
- Accurate computation results by eliminating raw bit errors in stored data



Multi-Wordline Sensing (MWS): Bitwise AND

- Intra-Block MWS:

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs

A bitline reads as '1' only when all the target cells store '1'

→ Equivalent to the bitwise AND of all the target cells

Operate
as a *resistance* (1)
or an *open switch* (0)

WL₂

WL₃

WL₄

BL₁

BL₂

BL₃

BL₄

Result: 0

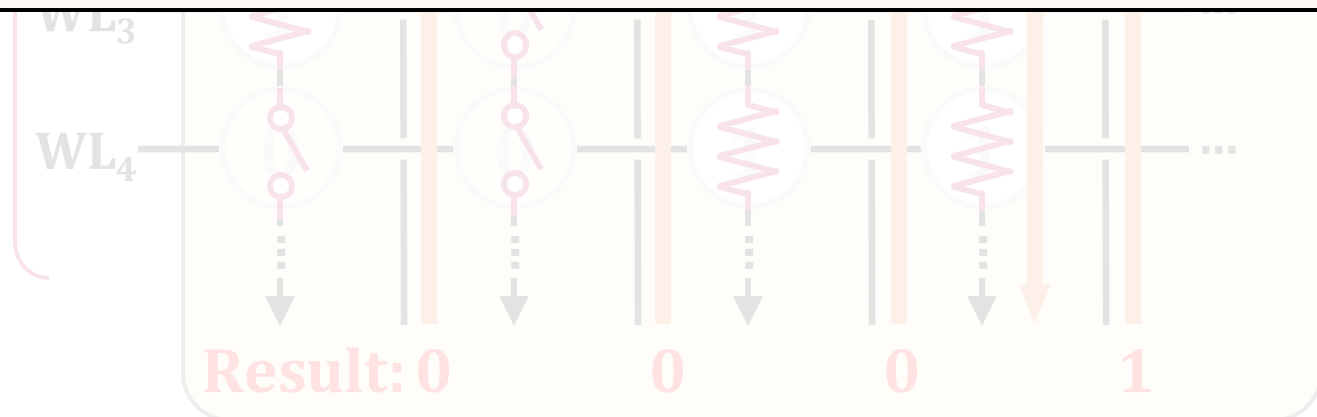
0

0

0

Multi-Wordline Sensing (MWS): Bitwise AND

Flash-Cosmos (Intra-Block MWS) enables
bitwise AND of multiple pages in the same block
via a single sensing operation



Other Types of Bitwise Operations

Flash-Cosmos also enables
other types of bitwise operations
(NOT/NAND/NOR/XOR/XNOR)
leveraging **existing features** of NAND flash memory

Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park^{§∇} Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*LIRMM, Univ. Montpellier, CNRS* [‡]*Kyungpook National University*



<https://arxiv.org/abs/2209.05566.pdf>

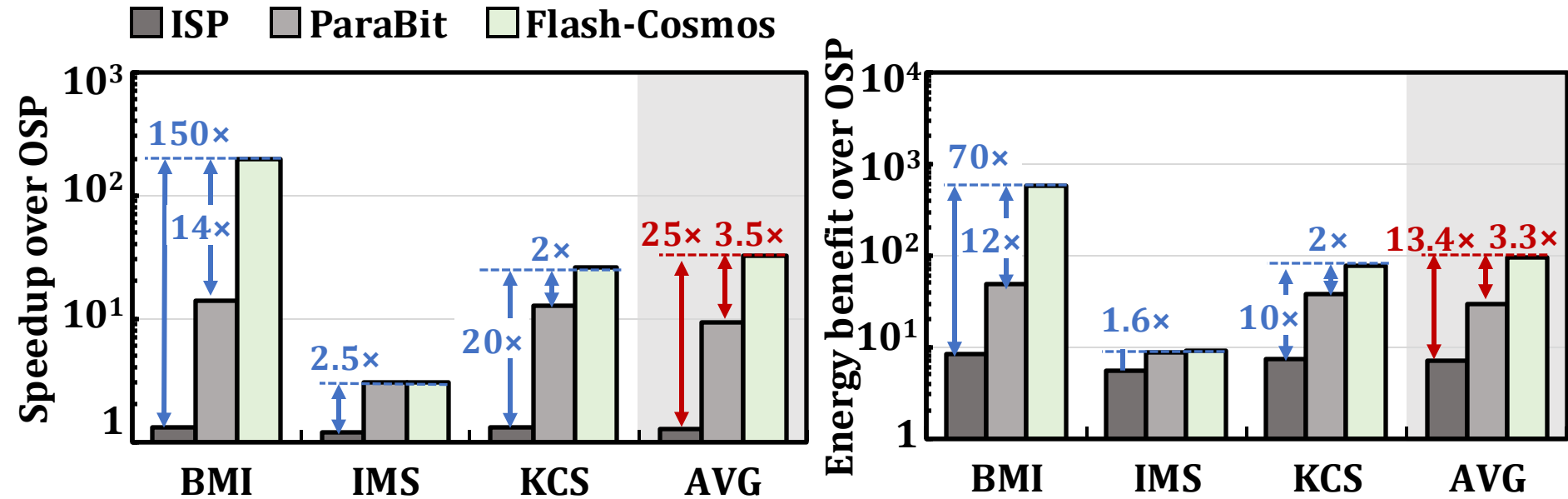
Results: Real-Device Characterization

No changes to the cell array
of commodity NAND flash chips

Can have many operands
(AND: up to 48, OR: up to 4)
with small increase in sensing latency (< 10%)

ESP significantly improves
the reliability of computation results
(no observed bit error in the tested flash cells)

Results: Performance & Energy



Flash-Cosmos provides **significant performance & energy benefits** over all the baselines

The larger the number of operands,
the higher the performance & energy benefits

Pinatubo: RowClone and Bitwise Ops in PCM

Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-volatile Memories

Shuangchen Li^{1*}, Cong Xu², Qiaosha Zou^{1,5}, Jishen Zhao³, Yu Lu⁴, and Yuan Xie¹

University of California, Santa Barbara¹, Hewlett Packard Labs²
University of California, Santa Cruz³, Qualcomm Inc.⁴, Huawei Technologies Inc.⁵
{shuangchenli, yuanxie}@ece.ucsb.edu¹

Other Readings on Processing using NVM

- Shafiee+, “ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars”, ISCA 2016.
- Chi+, “PRIME: A Novel Processing-in-memory Architecture for Neural Network Computation in ReRAM-based Main Memory”, ISCA 2016.
- Prezioso+, “Training and Operation of an Integrated Neuromorphic Network based on Metal-Oxide Memristors”, Nature 2015
- Ambrogio+, “Equivalent-accuracy accelerated neural-network training using analogue memory”, Nature 2018.

Processing in Memory: Two Approaches

1. Processing using Memory
2. **Processing near Memory**

PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu^{a,b}, Saugata Ghose^{b,c}, Juan Gómez-Luna^a, Rachata Ausavarungnirun^d

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

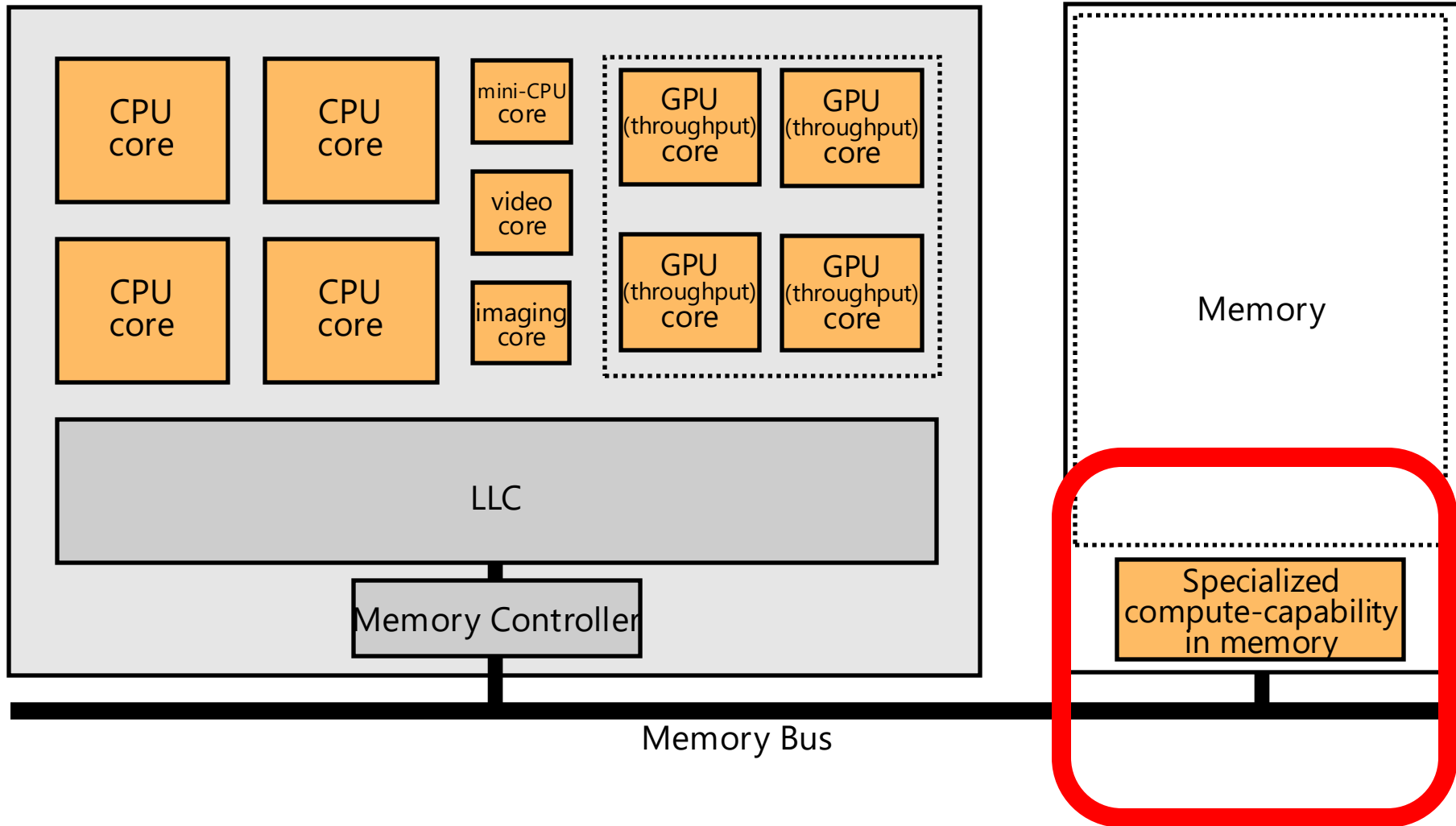
^d*King Mongkut's University of Technology North Bangkok*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,

"A Modern Primer on Processing in Memory"

*Invited Book Chapter in **Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann**, Springer, to be published in 2021.*

Mindset: Memory as an Accelerator



Memory similar to a "conventional" accelerator

Accelerating In-Memory Graph Analytics

- Large graphs are everywhere (circa 2015)



36 Million
Wikipedia Pages



1.4 Billion
Facebook Users

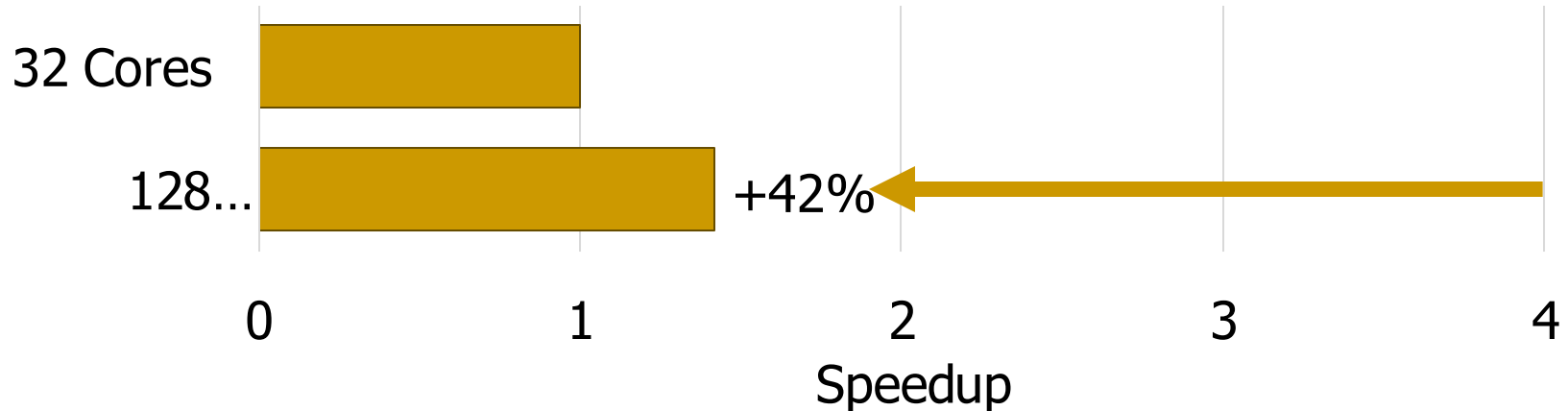


300 Million
Twitter Users



30 Billion
Instagram Photos

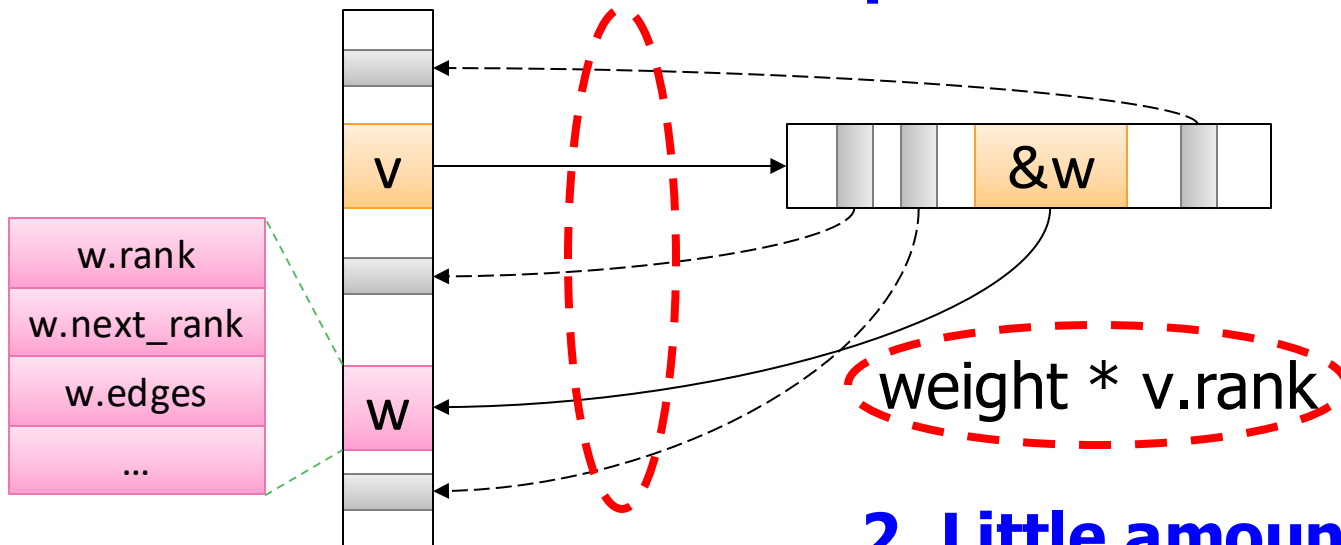
- Scalable large-scale graph processing is challenging



Key Bottlenecks in Graph Processing

```
for (v: graph.vertices) {  
  for (w: v.successors) {  
    w.next_rank += weight * v.rank;  
  }  
}
```

1. Frequent random memory accesses

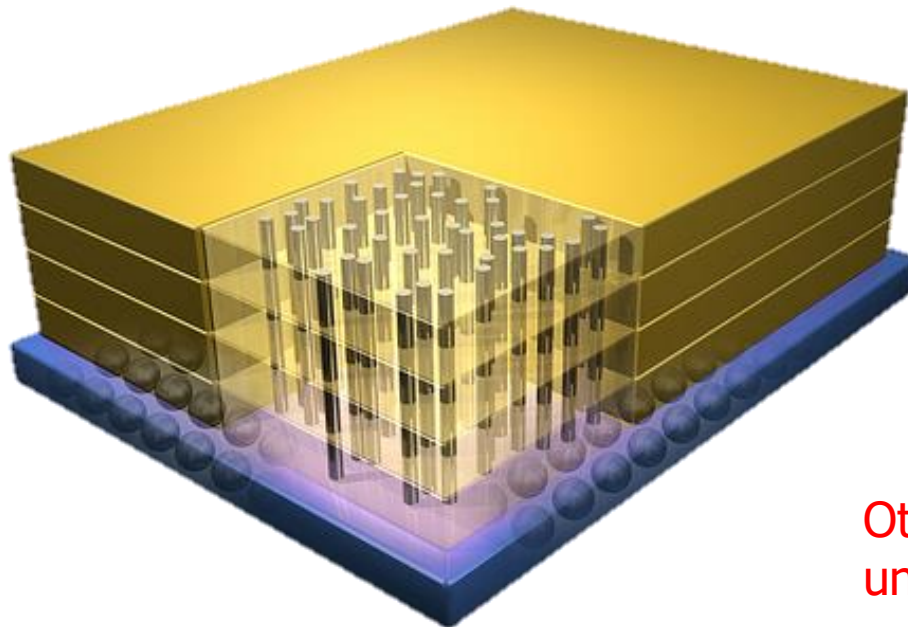


2. Little amount of computation

Opportunity: 3D-Stacked Logic+Memory



Hybrid Memory Cube
C O N S O R T I U M



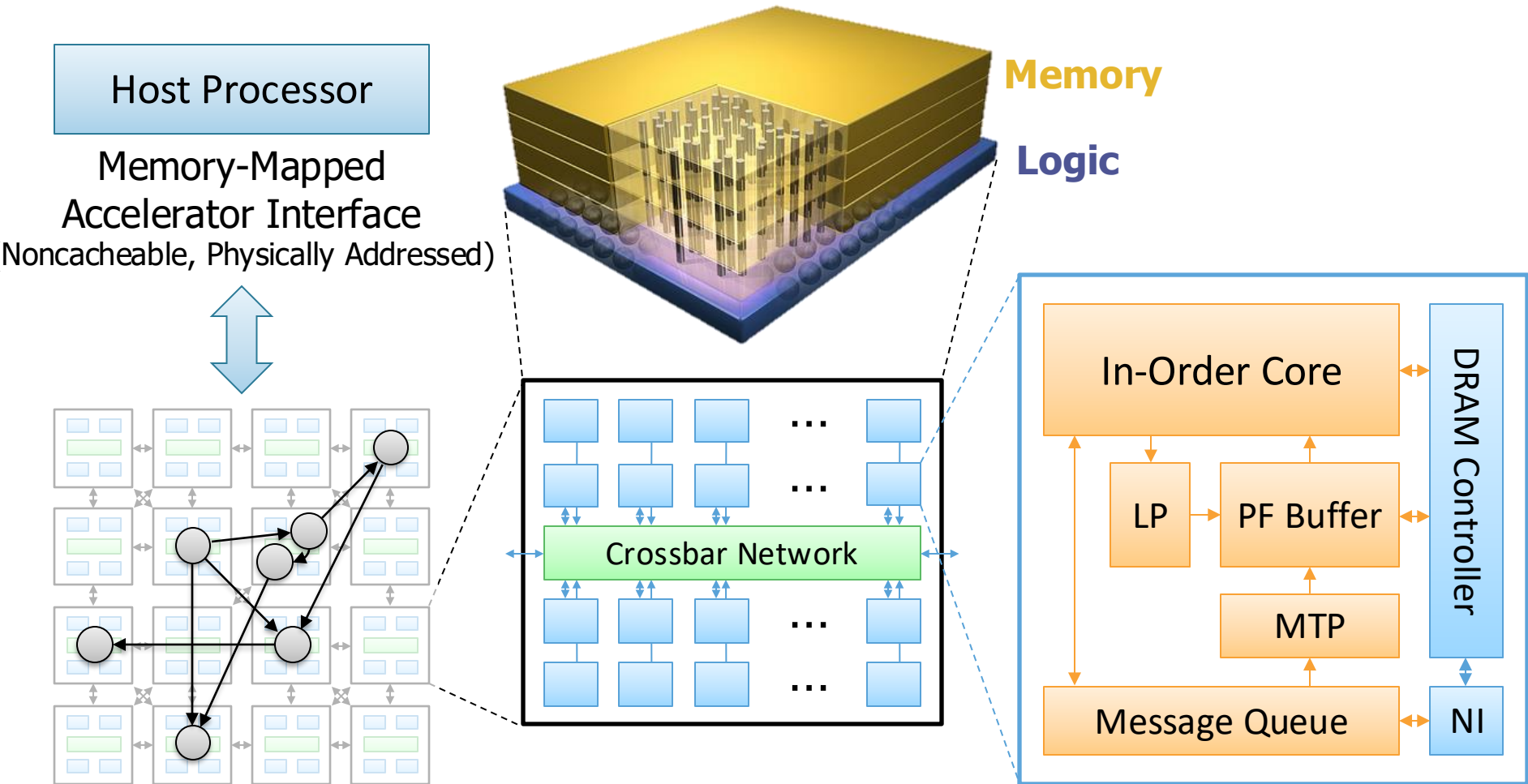
Memory

Logic

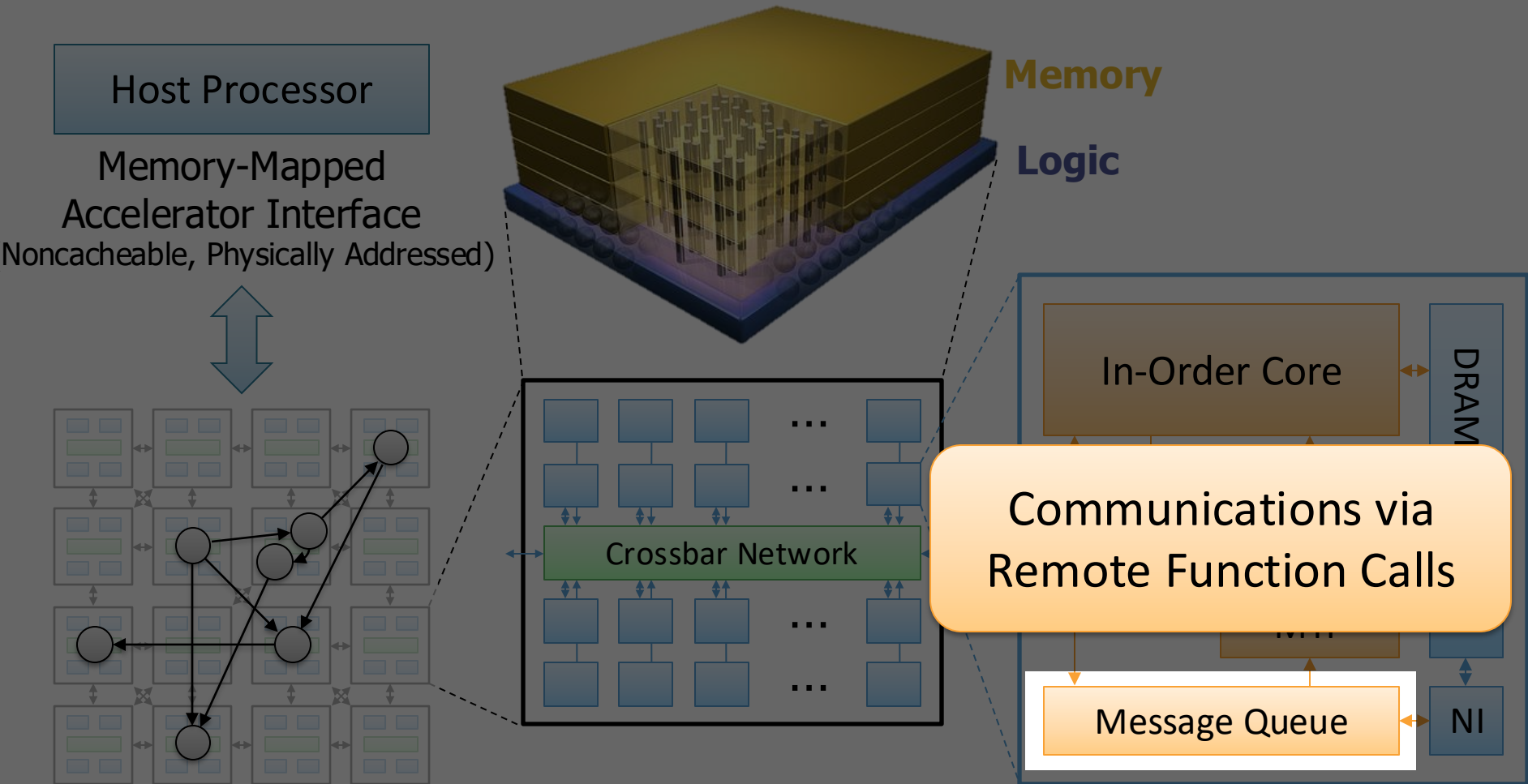
Other "True 3D" technologies
under development

Tesseract System for Graph Processing

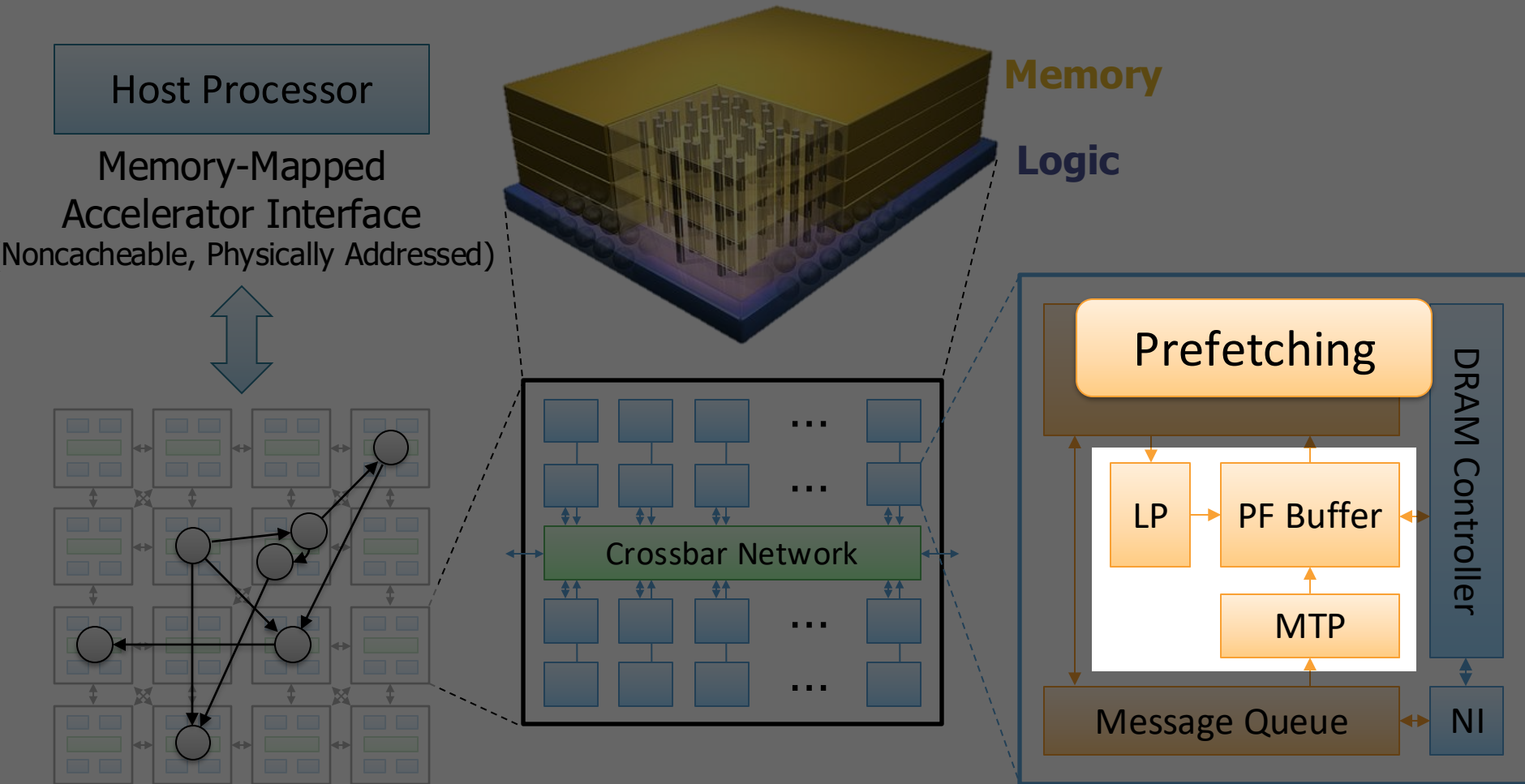
Interconnected set of 3D-stacked memory+logic chips with simple cores



Tesseract System for Graph Processing

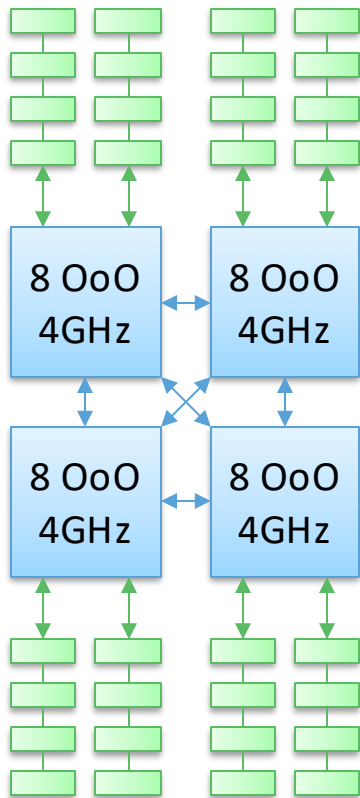


Tesseract System for Graph Processing



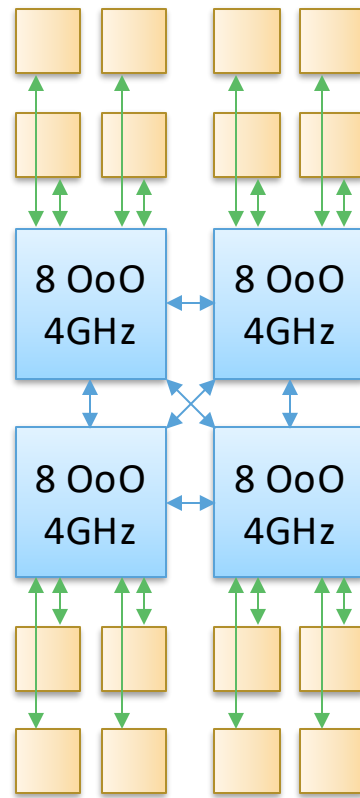
Evaluated Systems

DDR3-OoO



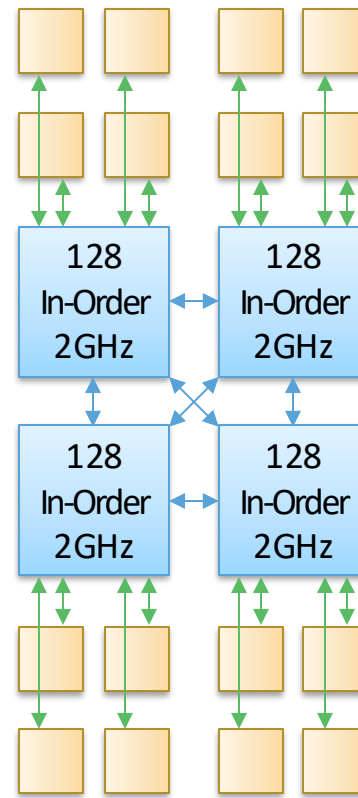
102.4GB/s

HMC-OoO



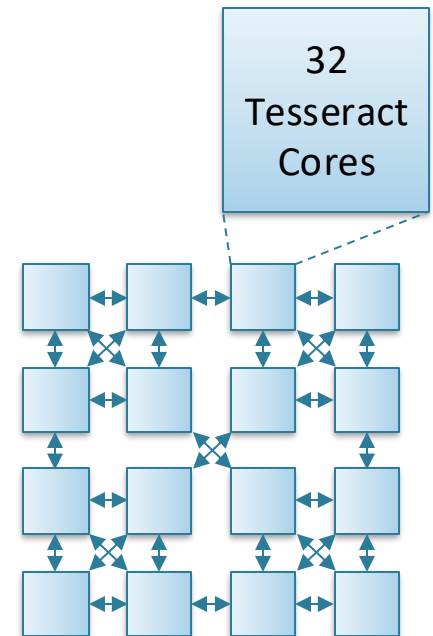
640GB/s

HMC-MC



640GB/s

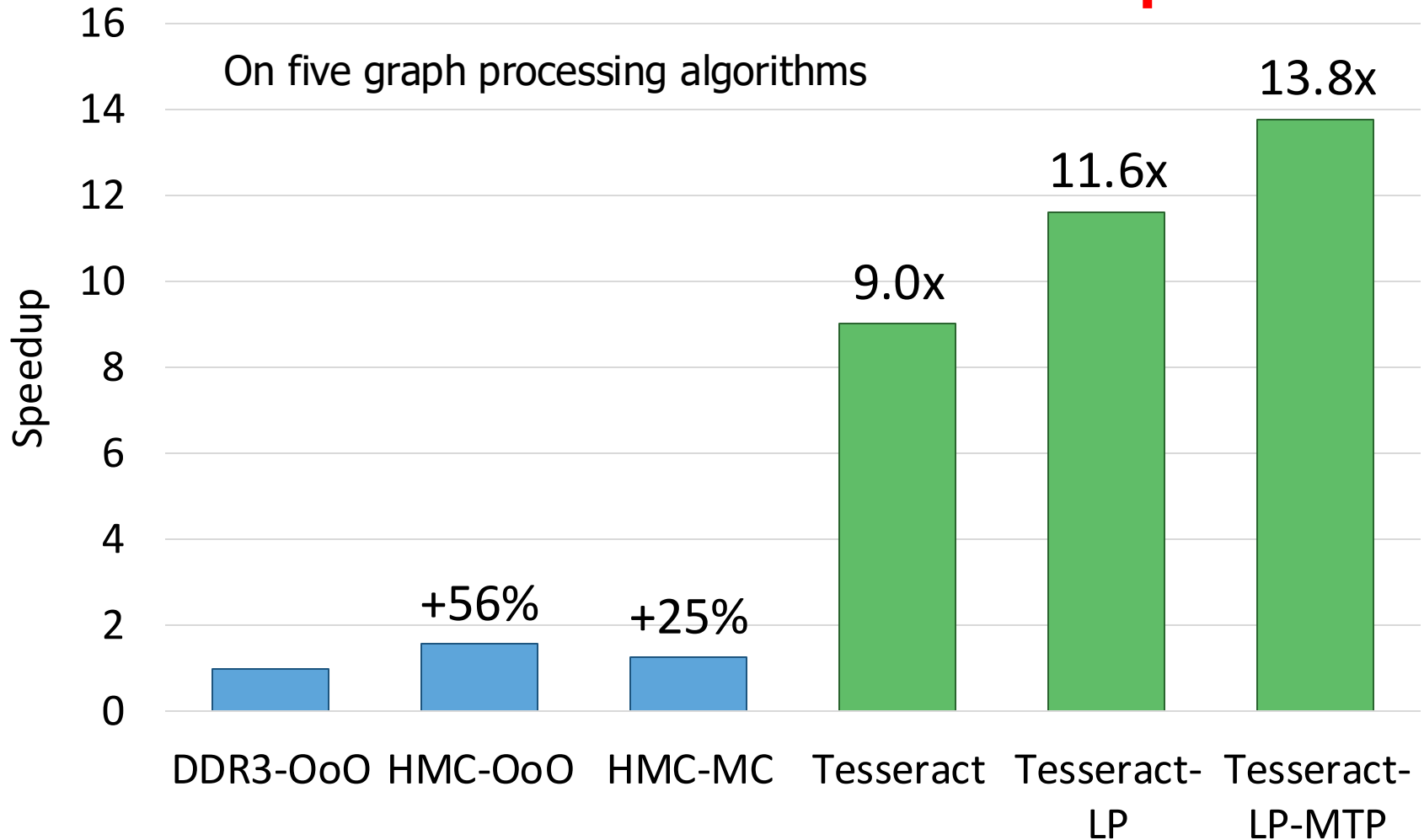
Tesseract



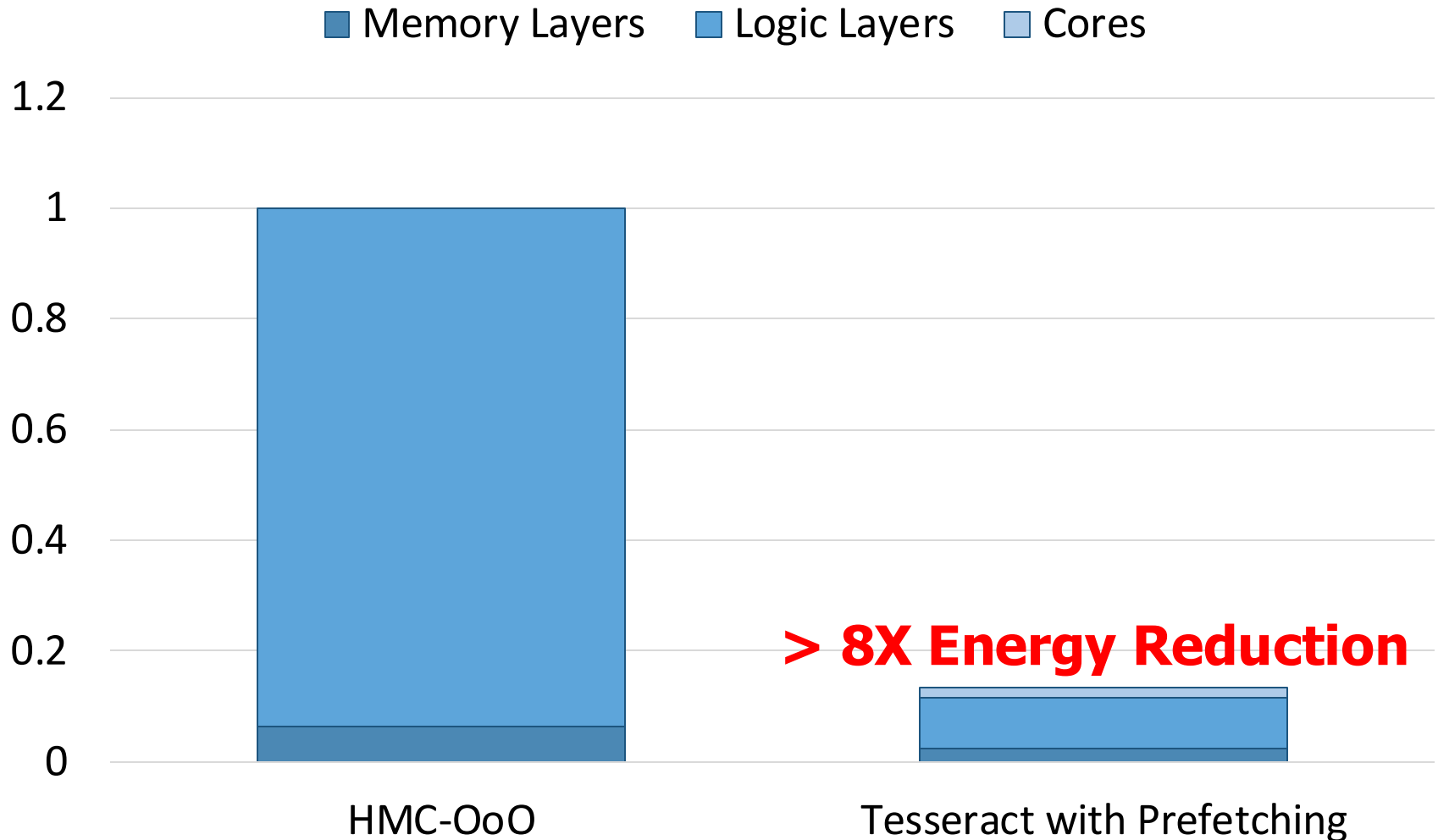
8TB/s

Tesseract Graph Processing Performance

>13X Performance Improvement



Tesseract Graph Processing System Energy



More on Tesseract

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoungh Choi,
"A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"
Proceedings of the 42nd International Symposium on Computer Architecture (ISCA), Portland, OR, June 2015.
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
Top Picks Honorable Mention by IEEE Micro.
Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing

Junwhan Ahn Sungpack Hong[§] Sungjoo Yoo Onur Mutlu[†] Kiyoungh Choi
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

[§]Oracle Labs

[†]Carnegie Mellon University

A Short Retrospective @ 50 Years of ISCA

Retrospective: A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing

Junwhan Ahn[†] Sungpack Hong[‡] Sungjoo Yoo[▽] Onur Mutlu[§] Kiyoung Choi[▽]
[†]Google DeepMind [‡]Oracle Labs [§]ETH Zürich [▽]Seoul National University

Abstract—Our ISCA 2015 paper [1] provides a new programmable processing-in-memory (PIM) architecture and system design that can accelerate key data-intensive applications, with a focus on graph processing workloads. Our major idea was to completely rethink the system, including the programming model, data partitioning mechanisms, system support, instruction set architecture, along with near-memory execution units and their communication architecture, such that an important workload can be accelerated at a maximum level using a distributed system of well-connected near-memory accelerators. We built our accelerator system, Tesseract, using 3D-stacked memories with logic layers, where each logic layer contains general-purpose processing cores and cores communicate with each other using a message-passing programming model. Cores could be specialized for graph processing (or any other application to be accelerated).

To our knowledge, our paper was the first to completely design a near-memory accelerator system from scratch such that it is both generally programmable and specifically customizable to accelerate important applications, with a case study on major graph processing workloads. Ensuing work in academia and industry showed that similar approaches to system design can greatly benefit both graph processing workloads and other applications, such as machine learning, for which ideas from Tesseract seem to have been influential.

This short retrospective provides a brief analysis of our ISCA 2015 paper and its impact. We briefly describe the major ideas and contributions of the work, discuss later works that built on it or were influenced by it, and make some educated guesses on what the future may bring on PIM and accelerator systems.

I. BACKGROUND, APPROACH & MINDSET

We started our research when 3D-stacked memories (e.g., [2–4]) were viable and seemed to have promise for building effective and practical processing-near-memory systems. Such near-memory systems could lead to improvements, but there was little to no research that examined how an accelerator could be completely (re-)designed using such near-memory technology, from its hardware architecture to its programming model and software system, and what the performance and energy benefits could be of such a re-design. We set out to answer these questions in our ISCA 2015 paper [1].

We followed several major principles to design our accelerator from the ground up. We believe these principles are still important: a major contribution and influence of our work was in putting all of these together in a cohesive full-system design and demonstrating the large performance and energy benefits that can be obtained from such a design. We see a similar approach in many modern large-scale accelerator systems in machine learning today (e.g., [5–9]). Our principles are:

1. *Near-memory execution* to enable/exploit the high data access bandwidth modern workloads (e.g., graph processing) need and to reduce data movement and access latency.

2. *General programmability* so that the system can be easily adopted, extended, and customized for many workloads.

3. *Maximal acceleration capability* to maximize the performance and energy benefits. We set ourselves free from backward compatibility and cost constraints. We aimed to completely re-design the system stack. Our goal was to explore the maximal performance and energy efficiency benefits we can gain from a near-memory accelerator if we had complete freedom to change things as much as we needed. We contrast this approach to the *minimal intrusion* approach we also explored in a separate ISCA 2015 paper [10].

4. *Customizable to specific workloads*, such that we can maximize acceleration benefits. Our focus workload was graph

analytics/processing, a key workload at the time and today. However, our design principles are not limited to graph processing and the system we built is customizable to other workloads as well, e.g., machine learning, genome analysis.

5. *Memory-capacity-proportional performance*, i.e., processing capability should proportionally grow (i.e., scale) as memory capacity increases and vice versa. This enables scaling of data-intensive workloads that need both memory and compute.

6. *Exploit new technology (3D stacking)* that enables tight integration of memory and logic and helps multiple above principles (e.g., enables customizable near-memory acceleration capability in the logic layer of a 3D-stacked memory chip).

7. *Good communication and scaling capability* to support scalability to large dataset sizes and to enable memory-capacity-proportional performance. To this end, we provided scalable communication mechanisms between execution cores and carefully interconnected small accelerator chips to form a large distributed system of accelerator chips.

8. *Maximal and efficient use of memory bandwidth* to supply the high-bandwidth data access that modern workloads need. To this end, we introduced new, specialized mechanisms for prefetching and a programming model that helps leverage application semantics for hardware optimization.

II. CONTRIBUTIONS AND INFLUENCE

We believe the major contributions of our work were 1) complete rethinking of how an accelerator system should be designed to enable maximal acceleration capability, and 2) the design and analysis of such an accelerator with this mindset and using the aforementioned principles to demonstrate its effectiveness in an important class of workloads.

One can find examples of our approach in modern large-scale machine learning (ML) accelerators, which are perhaps the most successful incarnation of scalable near-memory execution architectures. ML infrastructure today (e.g., [5–9]) consists of accelerator chips, each containing compute units and high-bandwidth memory, tightly packaged together, and features scale-up capability enabled by connecting thousands of such chips with high-bandwidth interconnection links. The system-wide rethinking that was done to enable such accelerators and many of the principles used in such accelerators resemble our ISCA 2015 paper’s approach.

The “memory-capacity-proportional performance” principle we explored in the paper shares similarities with how ML workloads are scaled up today. Similar to how we carefully sharded graphs across our accelerator chips to greatly improve effective memory bandwidth in our paper, today’s ML workloads are sharded across a large number of accelerators by leveraging data/model parallelism and optimizing the placement to balance communication overheads and compute scalability [11, 12]. With the advent of large generative models requiring high memory bandwidth for fast training and inference, the scaling behavior where capacity and bandwidth are scaled together has become an essential architectural property to support modern data-intensive workloads.

The “maximal acceleration capability” principle we used in Tesseract provides much larger performance and energy improvements and better customization than the “minimalist” approach that our other ISCA 2015 paper on *PIM-Enabled Instructions* [10] explored: “minimally change” an existing

system to incorporate (near-memory) acceleration capability to ease programming and keep costs low. So far, the industry has more widely adopted the maximal approach to overcome the pressing scaling bottlenecks of major workloads. The key enabler that bridges the programmability gap between the maximal approach favoring large performance & energy benefits and the minimal approach favoring ease of programming is compilation techniques. These techniques lower well-defined high-level constructs into lower-level primitives [12, 13]; our ISCA 2015 papers [1, 10] and a follow-up work [14] explore them lightly. We believe that a good programming model that enables large benefits coupled with support for it across the entire system stack (including compilers & hardware) will continue to be important for effective near-memory system and accelerator designs [14]. We also believe that the maximal versus minimal approaches that are initially explored in our two ISCA 2015 papers is a useful way of exploring emerging technologies (e.g., near-memory accelerators) to better understand the tradeoffs of system designs that exploit such technologies.

III. INFLUENCE ON LATER WORKS

Our paper was at the beginning of a proliferation of scalable near-memory processing systems designed to accelerate key applications (see [15] for many works on the topic). Tesseract has inspired many near-memory system ideas (e.g., [16–28]) and served as the de facto comparison point for such systems, including near-memory graph processing accelerators that built on Tesseract and improved various aspects of Tesseract. Since machine learning accelerators that use high-bandwidth memory (e.g., [5, 29]) and industrial PIM prototypes (e.g., [30–41]) are now in the market, near-memory processing is no longer an “eccentric” architecture it used to be when Tesseract was originally published.

Graph processing & analytics workloads remain as an important and growing class of applications in various forms, ranging from large-scale industrial graph analysis engines (e.g., [42]) to graph neural networks [43]. Our focus on large-scale graph processing in our ISCA 2015 paper increased attention to this domain in the computer architecture community, resulting in subsequent research on efficient hardware architectures for graph processing (e.g., [44–46]).

IV. SUMMARY AND FUTURE OUTLOOK

We believe that our ISCA 2015 paper’s principled rethinking of system design to accelerate an important class of data-intensive workloads provided significant value and enabled/influenced a large body of follow-on works and ideas. We expect that such rethinking of system design for key workloads, especially with a focus on “maximal acceleration capability,” will continue to be critical as pressing technology and application scaling challenges increasingly require us to think differently to substantially improve performance and energy (as well as other metrics). We believe the principles exploited in Tesseract are fundamental and they will remain useful and likely become even more important as systems become more constrained due to the continuously-increasing memory access and computation demands of future workloads. We also project that as hardware substrates for near-memory acceleration (e.g., 3D stacking, in-DRAM computation, NVM-based PIM, processing using memory [15]) evolve and mature, systems will take advantage of them even more, likely using principles similar to those used in the design of Tesseract.

REFERENCES

- [1] J. Ahn et al., “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing,” in ISCA, 2015.
- [2] Hybrid Memory Cube Consortium, “HMC Specification 1.1,” 2013.
- [3] J. Jeddah and B. Keeth, “Hybrid Memory Cube: New DRAM Architecture Increases Density and Performance,” in VLSIT, 2012.
- [4] JEDEC, “High Bandwidth Memory (HBM) DRAM,” Standard No. JESD235, 2013.

- [5] N. Jouppi et al., “TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embedding,” in ISCA, 2023.
- [6] J. Fowers et al., “A Configurable Cloud-Scale DNN Processor for Real-Time AI,” in ISCA, 2018.
- [7] S. Lie, “Cerebras Architecture Deep Dive: First Look Inside the Hardware/Software Co-Design for Deep Learning,” in IEEE Micro, 2023.
- [8] E. Talpes et al., “The Microarchitectural Switch of D0, Tesla’s Exa-Scale Computer,” in IEEE Micro, 2023.
- [9] A. Ishii and R. Wells, “NVLink-Network Switch - NVIDIA’s Switch Chip for High Communication-Bandwidth SuperPODs,” in Hot Chips, 2022.
- [10] J. Ahn et al., “PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture,” in ISCA, 2015.
- [11] R. Pope et al., “Efficiently Scaling Transformer Inference,” in MLSys, 2023.
- [12] D. Lepikhin et al., “GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding,” in ICLR, 2021.
- [13] S. Wang et al., “Overlap Communication with Dependent Computation via Decomposition in Large Deep Learning Models,” in ASPLOS, 2023.
- [14] J. Ahn et al., “AIM: Energy-Efficient Aggregation Inside the Memory Hierarchy,” ACM TACO, vol. 13, no. 4, 2016.
- [15] O. Mutlu et al., “A Modern Primer on Processing in Memory,” Emerging Computing: From Devices to Systems, 2021, https://arxiv.org/abs/2012.03112.
- [16] M. Zhang et al., “GraphP: Reducing Communication for PIM-Based Graph Processing with Efficient Data Partition,” in HPCA, 2018.
- [17] L. Song, “GraphR: Accelerating Graph Processing Using ReRAM,” in HPCA, 2019.
- [18] Z. Zhu et al., “GraphQ: Scalable PIM-Based Graph Processing,” in MICRO, 2019.
- [19] G. Dai et al., “GraphH: A Processing-in-Memory Architecture for Large-Scale Graph Processing,” IEEE TCAD, 2018.
- [20] G. Li et al., “GraphIA: An In-situ Accelerator for Large-Scale Graph Processing,” in MEMSYS, 2018.
- [21] S. Rheindt et al., “NEMESIS: Near-Memory Graph Copy Enhanced System-Software,” in MEMSYS, 2019.
- [22] L. Belayneh and V. Bertacco, “GraphVine: Exploiting Multicast for Scalable Graph Analytics,” in DATE, 2020.
- [23] N. Challapala et al., “Gaus-X: Graph Analytics Accelerator Supporting Sparse Data Representation using Crossbar Architectures,” in ISCA, 2020.
- [24] M. Zhou et al., “Ultra Efficient Accelerator for De Novo Genome Assembly by Near-Memory Computing,” in ISCA, 2021.
- [25] X. Xie et al., “Space-A: Sparse Matrix Vector Multiplication on Processing-in-Memory Accelerator,” in HPCA, 2021.
- [26] M. Zhou et al., “HyGraph: Accelerating Graph Processing with Hybrid Memory-Centric Computing,” in ISCA, 2021.
- [27] M. Lenjani et al., “Gearbox: A Case for Supporting Accumulation Dispatching and Hybrid Partitioning in PIM-based Accelerators,” in ISCA, 2022.
- [28] M. Orenes-Vera et al., “Dalorex: A Data-Local Program Execution and Architecture for Memory-Bound Applications,” in HPCA, 2023.
- [29] J. Choquette, “Nvidia Hopper GPU: Scaling Performance,” in Hot Chips, 2022.
- [30] F. Devaux, “The True Processing in Memory Accelerator,” in Hot Chips, 2019.
- [31] J. Gómez-Luna et al., “Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System,” IEEE Access, 2022.
- [32] J. Gómez-Luna et al., “Evaluating Machine Learning Workloads on Memory-Centric Computing Systems,” in ISPAAS, 2023.
- [33] S. Lee et al., “Hardware Architectures and Software Stack for PIM Based on Commercial DRAM Technology: Industrial Product,” in ISCA, 2021.
- [34] Y.-C. Kwon et al., “25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2 TTOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications,” in ISSCC, 2021.
- [35] L. Ke et al., “Near-Memory Processing in Action: Accelerating Personalized Recommendation with Accelerated Ax/DIMM,” IEEE Micro, 2021.
- [36] D. Lee et al., “Improving In-Memory Database Operations with Accelerated DIMM (Ax/DIMM),” in DaMoN, 2022.
- [37] S. Lee et al., “A 1nm 125V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting ITTOPS MAC Operation and Various Activation Functions for Deep-Learning Applications,” in ISSCC, 2022.
- [38] D. Niu et al., “184QPS/W 64Mb/mm² 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System,” in ISSCC, 2022.
- [39] Y. Kwon, “System Architecture and Software Stack for GDDR6-AIM,” in HCS, 2022.
- [40] G. Singh et al., “FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications,” IEEE Micro, 2021.
- [41] G. Singh et al., “Accelerating Weather Prediction using Near-Memory Reconfigurable Fabric,” ACM TACO, 2019.
- [42] S. Hong et al., “PGX-D: A Fast Distributed Graph Processing Engine,” in SC, 2015.
- [43] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in ICLR, 2017.
- [44] L. Nai et al., “GraphPIM: Enabling Instruction-Level PIM Offloading in Graph Computing Frameworks,” in HPCA, 2017.
- [45] M. Besta et al., “MUSA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems,” in MICRO, 2021.
- [46] T. J. Ham et al., “Graphicionado: A High-Performance and Energy-Efficient Accelerator for Graph Analytics,” in MICRO, 2016.

Accelerating Graph Pattern Mining

- Maciej Besta, Raghavendra Kanakagiri, Grzegorz Kwasniewski, Rachata Ausavarungnirun, Jakub Beránek, Konstantinos Kanellopoulos, Kacper Janda, Zur Vonarburg-Shmaria, Lukas Gianinazzi, Ioana Stefan, Juan Gómez-Luna, Marcin Copik, Lukas Kapp-Schwoerer, Salvatore Di Girolamo, Nils Blach, Marek Konieczny, Onur Mutlu, and Torsten Hoefler,

"SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems"

Proceedings of the 54th International Symposium on Microarchitecture (MICRO), Virtual, October 2021.

[[Slides \(pdf\)](#)]

[[Talk Video](#) (22 minutes)]

[[Lightning Talk Video](#) (1.5 minutes)]

[[Full arXiv version](#)]

SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems

Maciej Besta¹, Raghavendra Kanakagiri², Grzegorz Kwasniewski¹, Rachata Ausavarungnirun³, Jakub Beránek⁴, Konstantinos Kanellopoulos¹, Kacper Janda⁵, Zur Vonarburg-Shmaria¹, Lukas Gianinazzi¹, Ioana Stefan¹, Juan Gómez-Luna¹, Marcin Copik¹, Lukas Kapp-Schwoerer¹, Salvatore Di Girolamo¹, Nils Blach¹, Marek Konieczny⁵, Onur Mutlu¹, Torsten Hoefler¹

¹ETH Zurich, Switzerland
Thailand

²IIT Tirupati, India

³King Mongkut's University of Technology North Bangkok,
⁴Technical University of Ostrava, Czech Republic

⁵AGH-UST, Poland

Accelerating Neural Network Inference

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,
"Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"
Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (14 minutes)]

Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand^{†◇}

Geraldo F. Oliveira^{*}

Saugata Ghose[‡]

Xiaoyu Ma[§]

Berkin Akin[§]

Eric Shiu[§]

Ravi Narayanaswami[§]

Onur Mutlu^{*†}

[†]*Carnegie Mellon Univ.*

[◇]*Stanford Univ.*

[‡]*Univ. of Illinois Urbana-Champaign*

[§]*Google*

^{*}*ETH Zürich*

Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand

Saugata Ghose

Berkin Akin

Ravi Narayanaswami

Geraldo F. Oliveira

Xiaoyu Ma

Eric Shiu

Onur Mutlu

PACT 2021

SAFARI

Carnegie Mellon



UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN



ETH zürich

Executive Summary

Context: We extensively analyze a state-of-the-art edge ML accelerator (Google Edge TPU) using 24 Google edge models

- Wide range of models (CNNs, LSTMs, Transducers, RCNNs)

Problem: The Edge TPU accelerator suffers from **three challenges:**

- It operates **significantly below** its peak throughput
- It operates **significantly below** its theoretical energy efficiency
- It **inefficiently** handles memory accesses

Key Insight: These shortcomings arise from **the monolithic design** of the Edge TPU accelerator

- The Edge TPU accelerator design does not account for **layer heterogeneity**

Key Mechanism: A new framework called **Mensa**

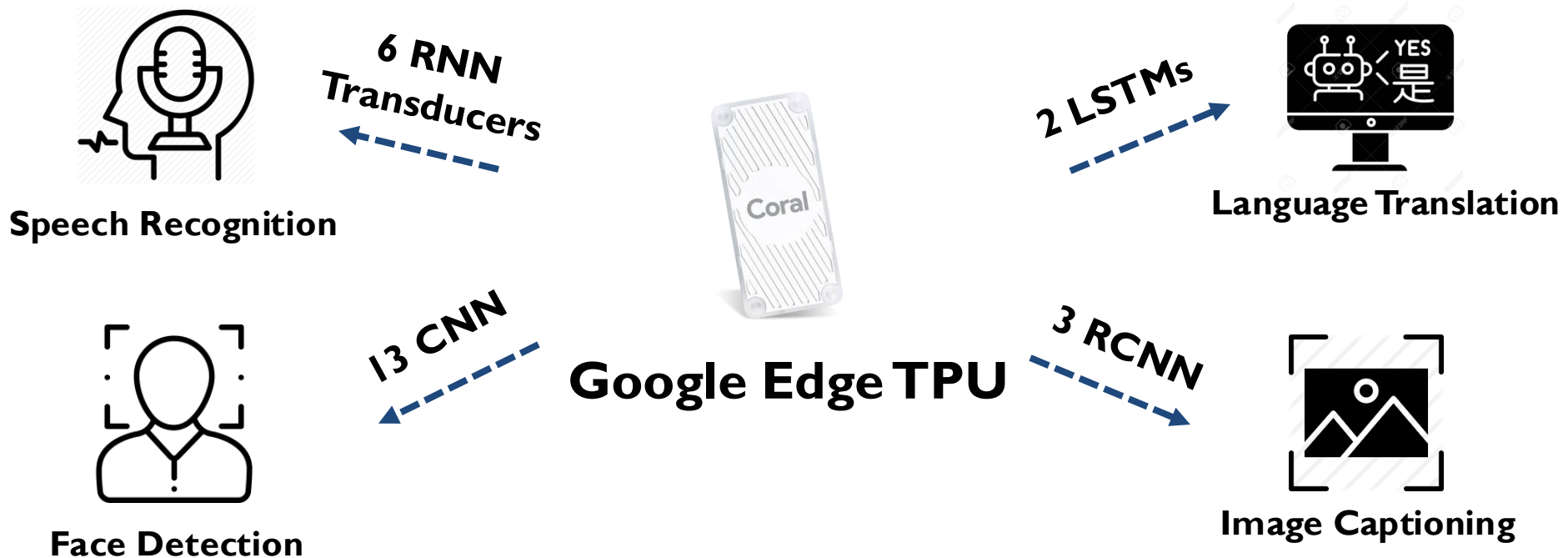
- Mensa consists of heterogeneous accelerators whose dataflow and hardware are specialized for specific families of layers

Key Results: We design a version of Mensa for Google edge ML models

- Mensa improves performance and energy by **3.0X** and **3.1X**
- Mensa reduces cost and improves area efficiency

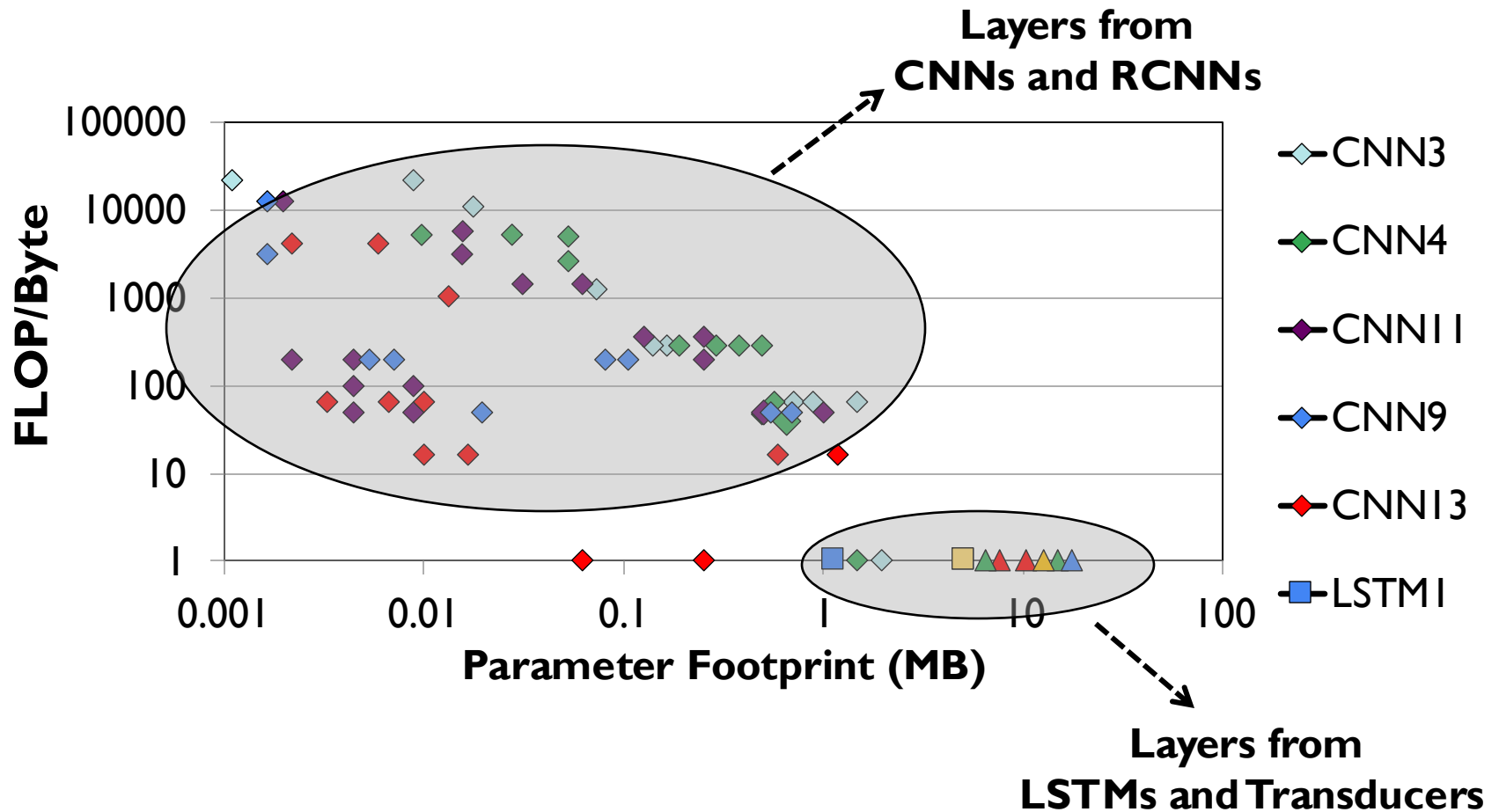
Google Edge Neural Network Models

We analyze inference execution using 24 edge NN models



Diversity Across the Models

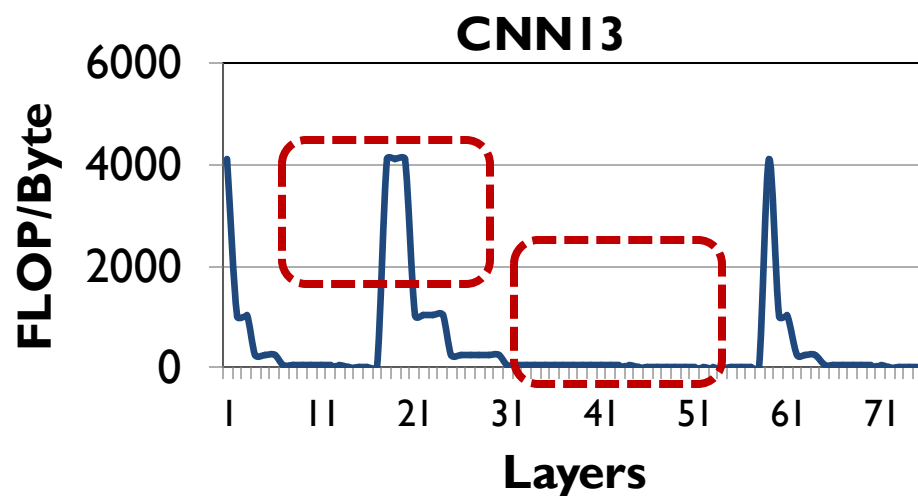
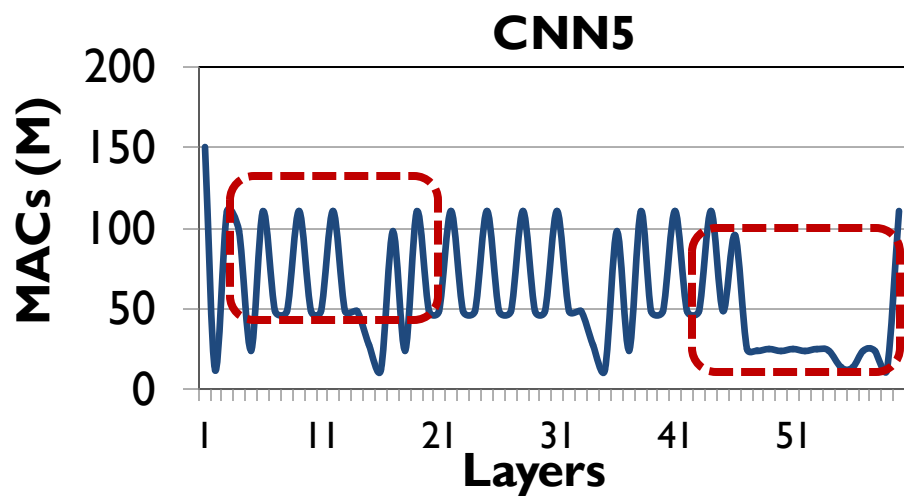
Insight 1: there is **significant variation** in terms of layer characteristics **across the models**



Diversity Within the Models

Insight 2: even **within** each model, layers exhibit **significant variation** in terms of layer characteristics

For example, our analysis of edge **CNN** models shows:



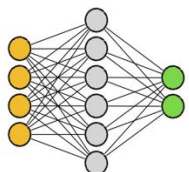
Variation in **MAC intensity**: up to **200x** across layers

Variation in **FLOP/Byte**: up to **244x** across layers

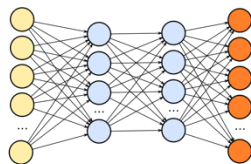
Mensa High-Level Overview

Edge TPU Accelerator

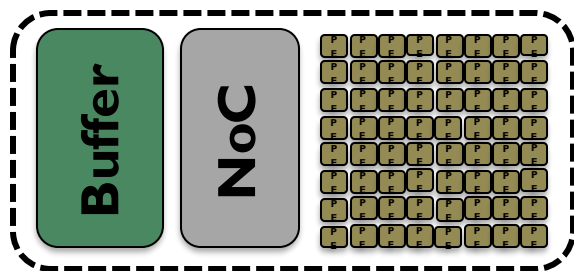
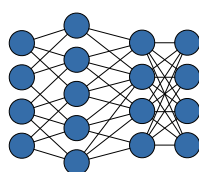
Model A



Model B



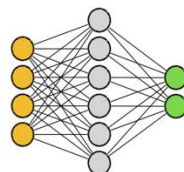
Model C



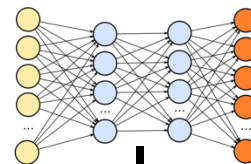
Monolithic Accelerator

Mensa

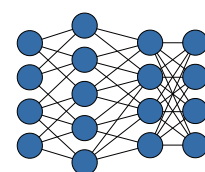
Model A



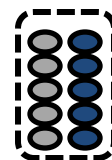
Model B



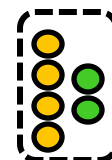
Model C



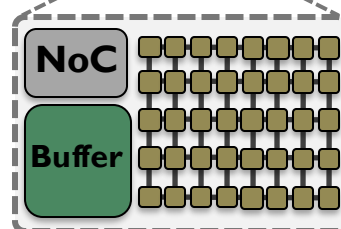
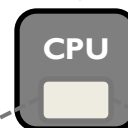
Family 1



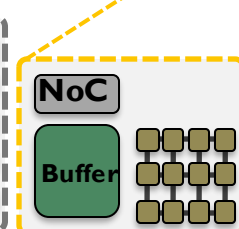
Family 2



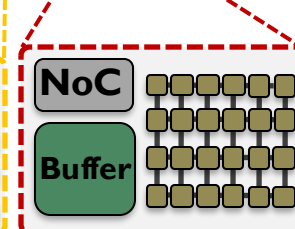
Family 3



Acc. 1



Acc. 2

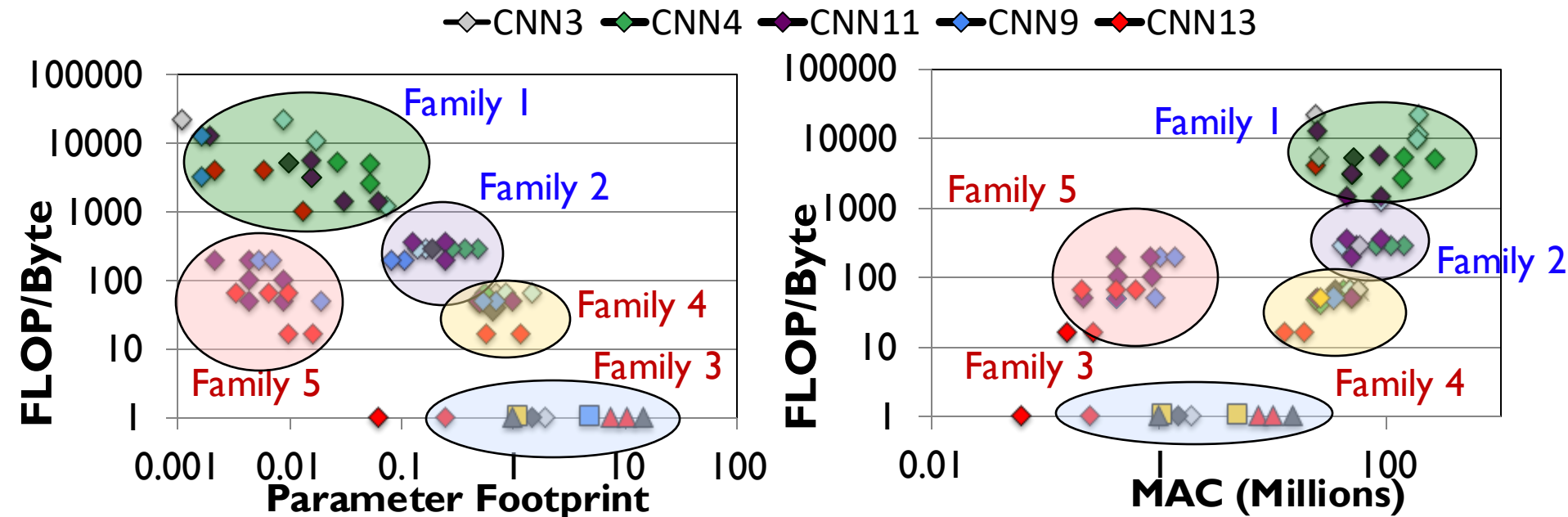


Acc. 3

Heterogeneous Accelerators

Identifying Layer Families

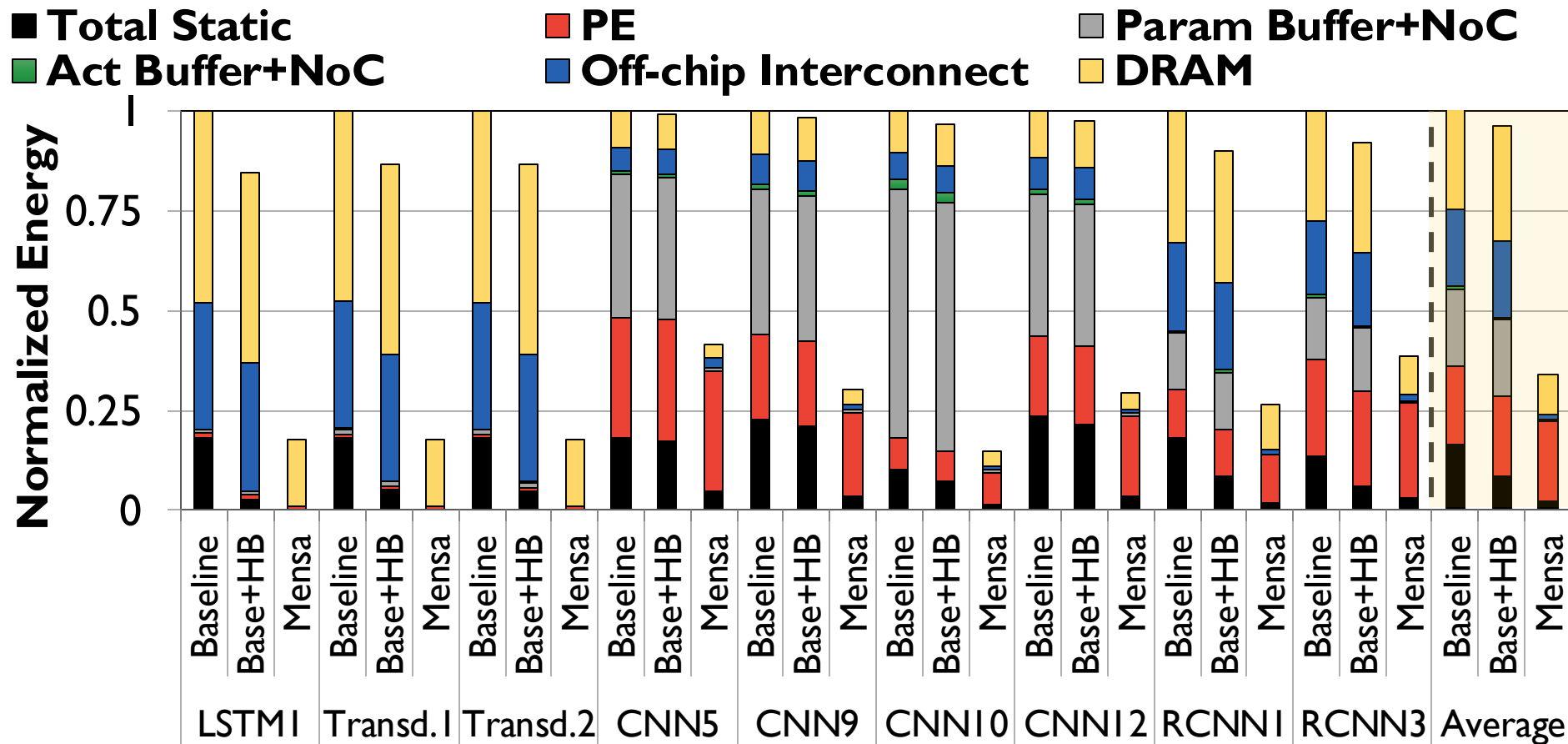
Key observation: the majority of layers group into a small number of layer families



Families 1 & 2: low parameter footprint, high data reuse and **MAC intensity**
→ compute-centric layers

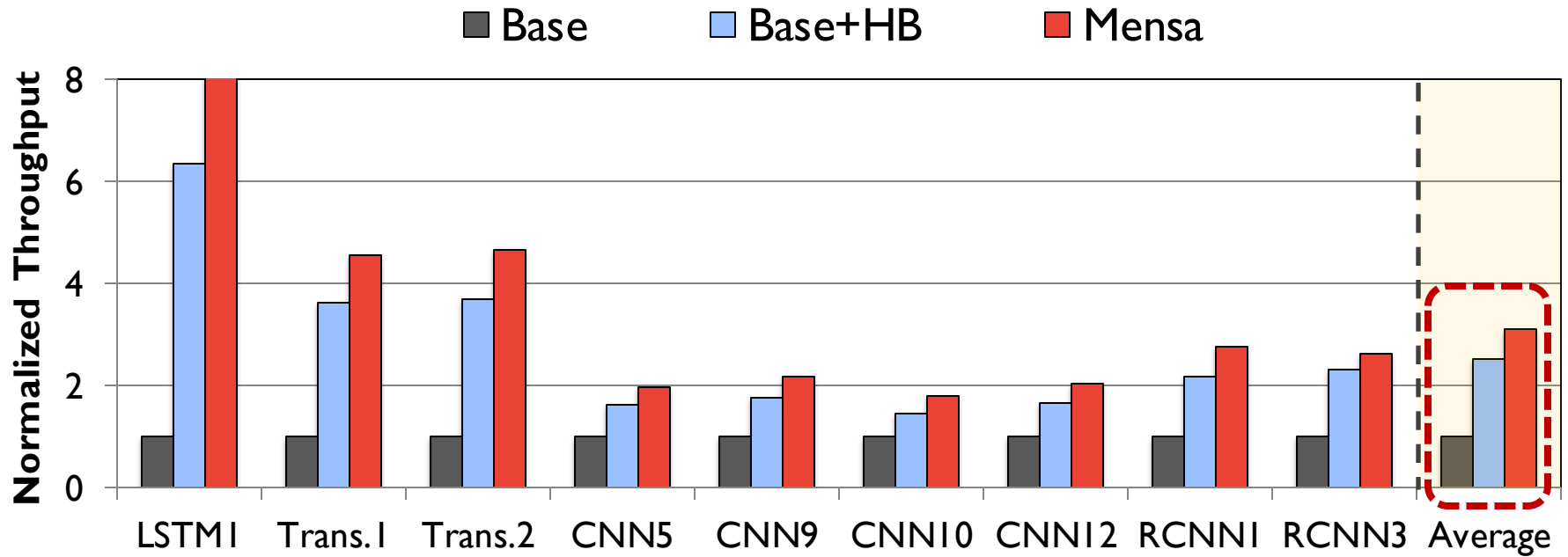
Families 3, 4 & 5: high parameter footprint, low data reuse and **MAC intensity**
→ data-centric layers

Mensa: Energy Reduction



Mensa-G reduces energy consumption by 3.0X
compared to the baseline Edge TPU

Mensa: Throughput Improvement



Mensa-G improves inference throughput by 3.1X
compared to the baseline Edge TPU

Mensa: Highly-Efficient ML Inference

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,
"Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"
Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (14 minutes)]

Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand^{†◇}

Geraldo F. Oliveira^{*}

Saugata Ghose[‡]

Xiaoyu Ma[§]

Berkin Akin[§]

Eric Shiu[§]

Ravi Narayanaswami[§]

Onur Mutlu^{*†}

[†]*Carnegie Mellon Univ.*

[◇]*Stanford Univ.*

[‡]*Univ. of Illinois Urbana-Champaign*

[§]*Google*

^{*}*ETH Zürich*

In-Storage Genomic Data Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,
["GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"](#)
Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, February-March 2022.
[[Lightning Talk Slides \(pptx\)](#)] ([pdf](#))
[[Lightning Talk Video](#) (90 seconds)]

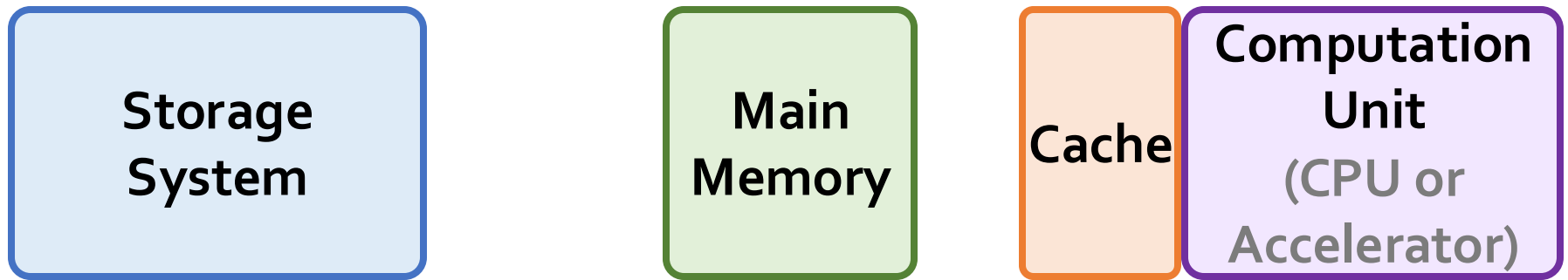
GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi¹ Jisung Park¹ Harun Mustafa¹ Jeremie Kim¹ Ataberk Olgun¹
Arvid Gollwitzer¹ Damla Senol Cali² Can Firtina¹ Haiyu Mao¹ Nour Almadhoun Alserr¹
Rachata Ausavarungnirun³ Nandita Vijaykumar⁴ Mohammed Alser¹ Onur Mutlu¹

¹ETH Zürich ²Bionano Genomics ³KMUTNB ⁴University of Toronto

Genome Sequence Analysis

Data Movement from Storage

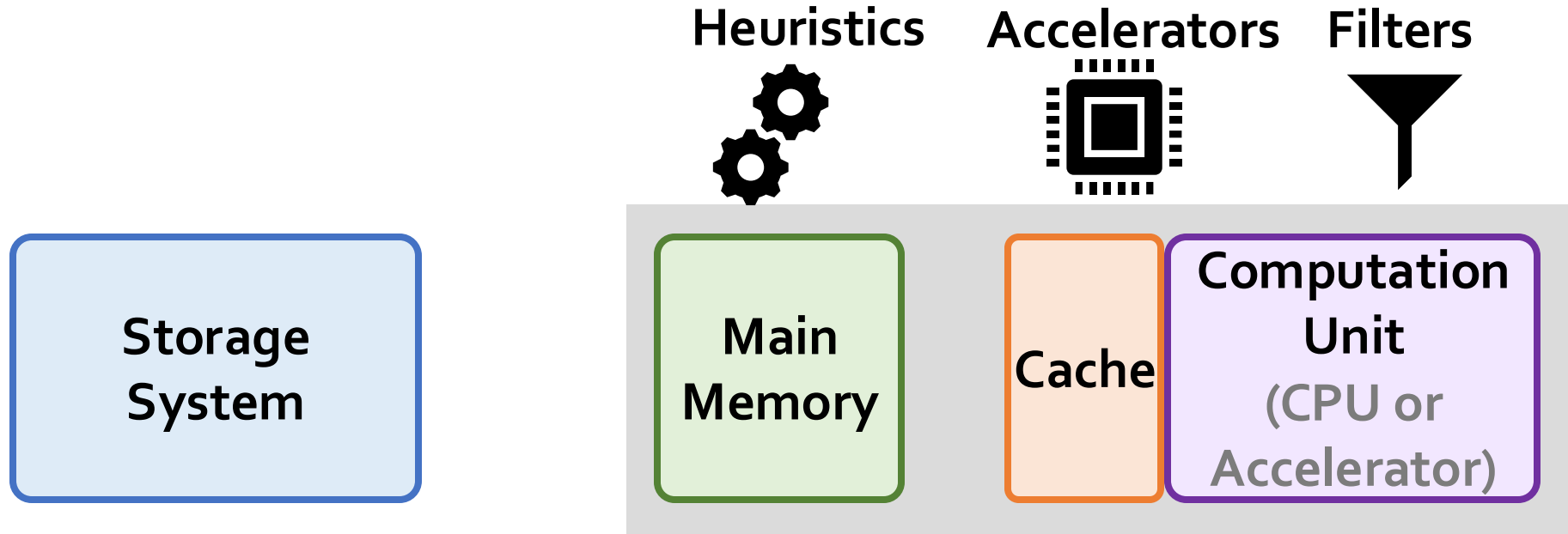


Computation overhead



Data movement overhead

Compute-Centric Accelerators



Computation overhead

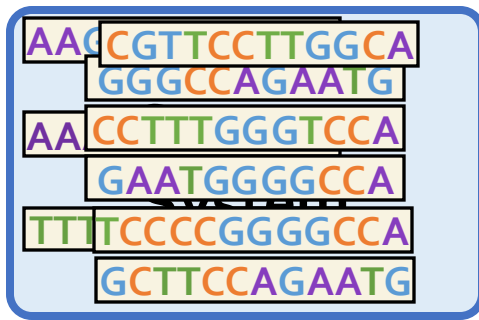


Data movement overhead

Key Idea: In-Storage Filtering



Filter reads that do not require alignment inside the storage system



Filtered Reads

Main
Memory

Cache

Computation
Unit
(CPU or
Accelerator)

Exactly-matching reads

Do not need expensive approximate string matching during alignment

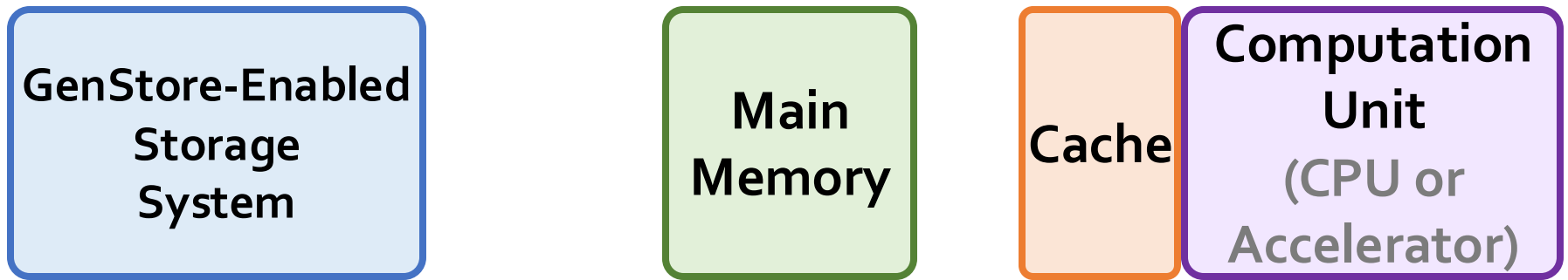
Non-matching reads

Do not have potential matching locations and can skip alignment

GenStore



*Filter reads that do **not** require alignment
inside the storage system*



Computation overhead



Data movement overhead

**GenStore provides significant speedup (1.4x - 33.6x) and
energy reduction (3.9x - 29.2x) at low cost**

In-Storage Genomic Data Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,
"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"
Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, February-March 2022.
[[Lightning Talk Slides \(pptx\) \(pdf\)](#)]
[[Lightning Talk Video](#) (90 seconds)]

GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi¹ Jisung Park¹ Harun Mustafa¹ Jeremie Kim¹ Ataberk Olgun¹
Arvid Gollwitzer¹ Damla Senol Cali² Can Firtina¹ Haiyu Mao¹ Nour Almadhoun Alserr¹
Rachata Ausavarungnirun³ Nandita Vijaykumar⁴ Mohammed Alser¹ Onur Mutlu¹

¹ETH Zürich ²Bionano Genomics ³KMUTNB ⁴University of Toronto

In-Storage Metagenomics [ISCA 2024]

- Nika Mansouri Ghiasi, Mohammad Sadrosadati, Harun Mustafa, Arvid Gollwitzer, Can Firtina, Julien Eudine, Haiyu Mao, Joel Lindegger, Meryem Banu Cavlak, Mohammed Alser, Jisung Park, and Onur Mutlu,

"MegIS: High-Performance and Low-Cost Metagenomic Analysis with In-Storage Processing"

Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA), Buenos Aires, Argentina, July 2024.

[\[Slides \(pptx\) \(pdf\)\]](#)

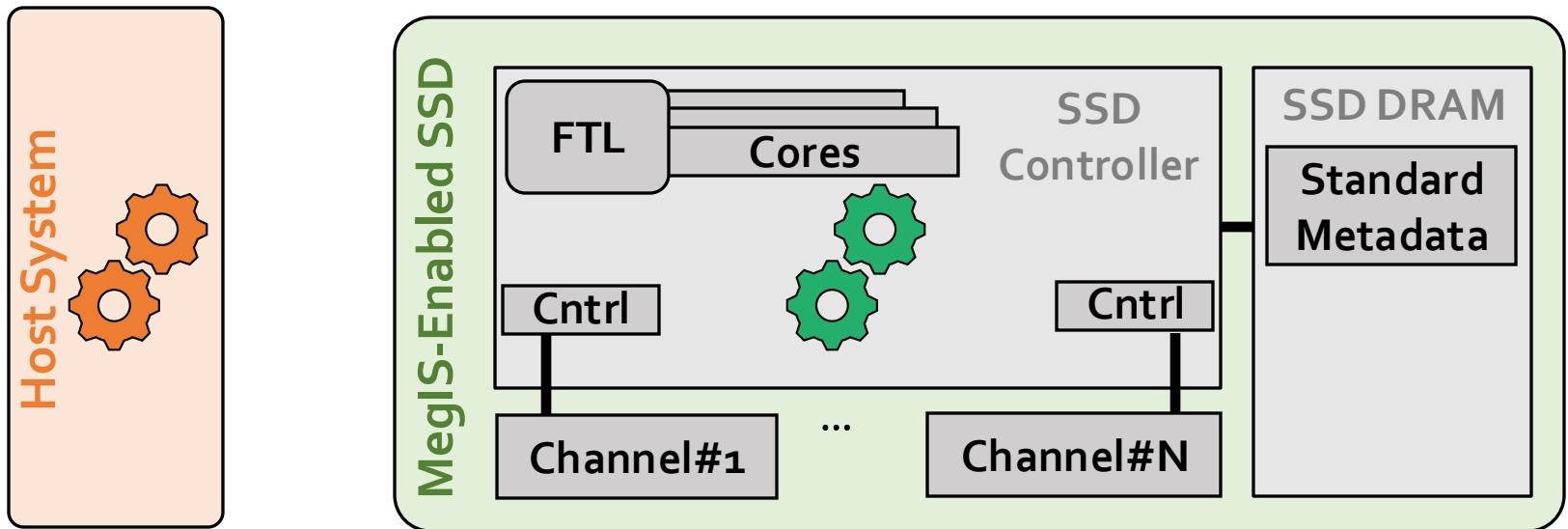
[\[arXiv version\]](#)

MegIS: High-Performance, Energy-Efficient, and Low-Cost Metagenomic Analysis with In-Storage Processing

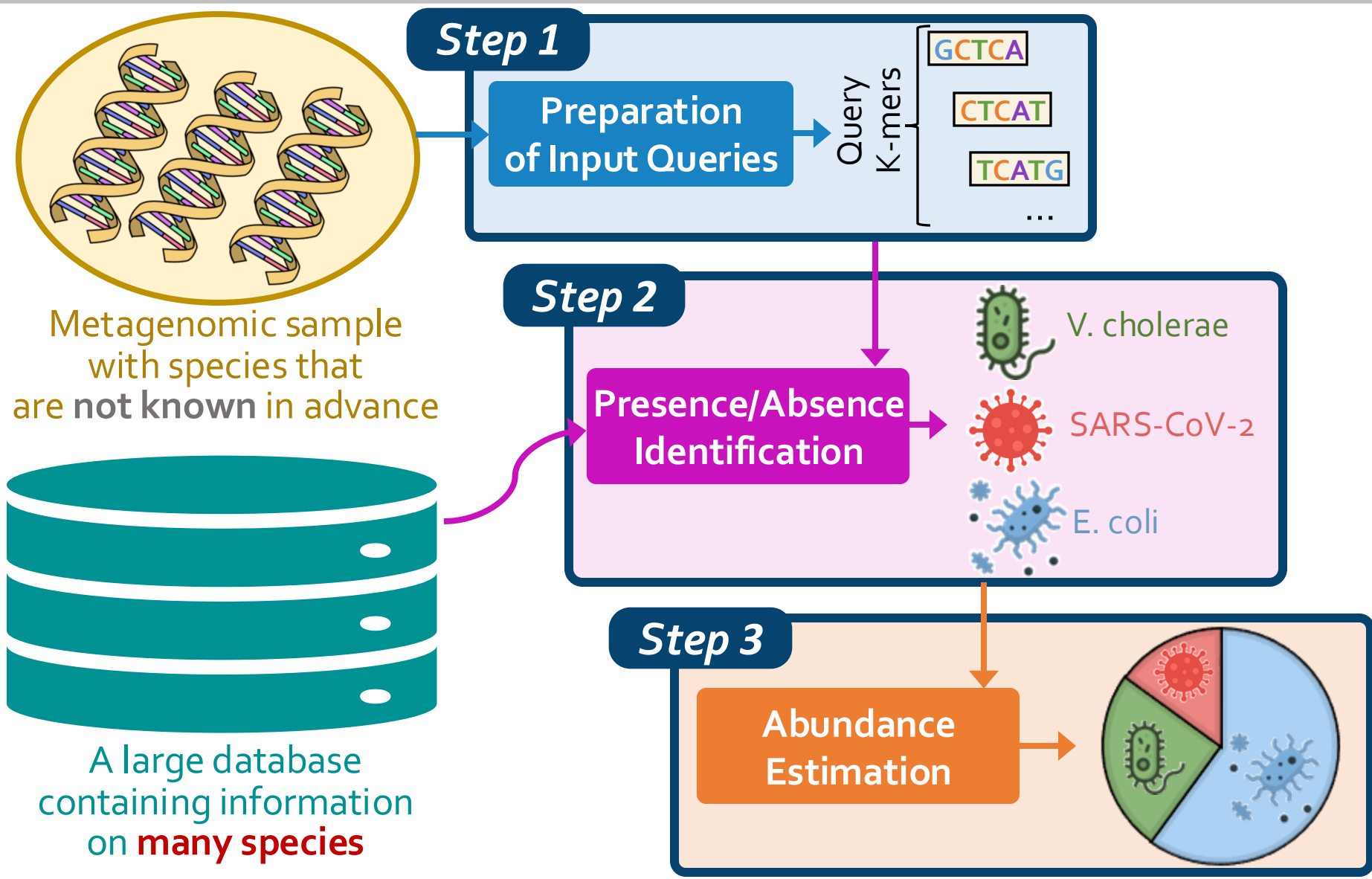
Nika Mansouri Ghiasi¹ Mohammad Sadrosadati¹ Harun Mustafa¹ Arvid Gollwitzer¹
Can Firtina¹ Julien Eudine¹ Haiyu Mao¹ Joël Lindegger¹ Meryem Banu Cavlak¹
Mohammed Alser¹ Jisung Park² Onur Mutlu¹
¹ETH Zürich ²POSTECH

MegIS: Metagenomics In-Storage

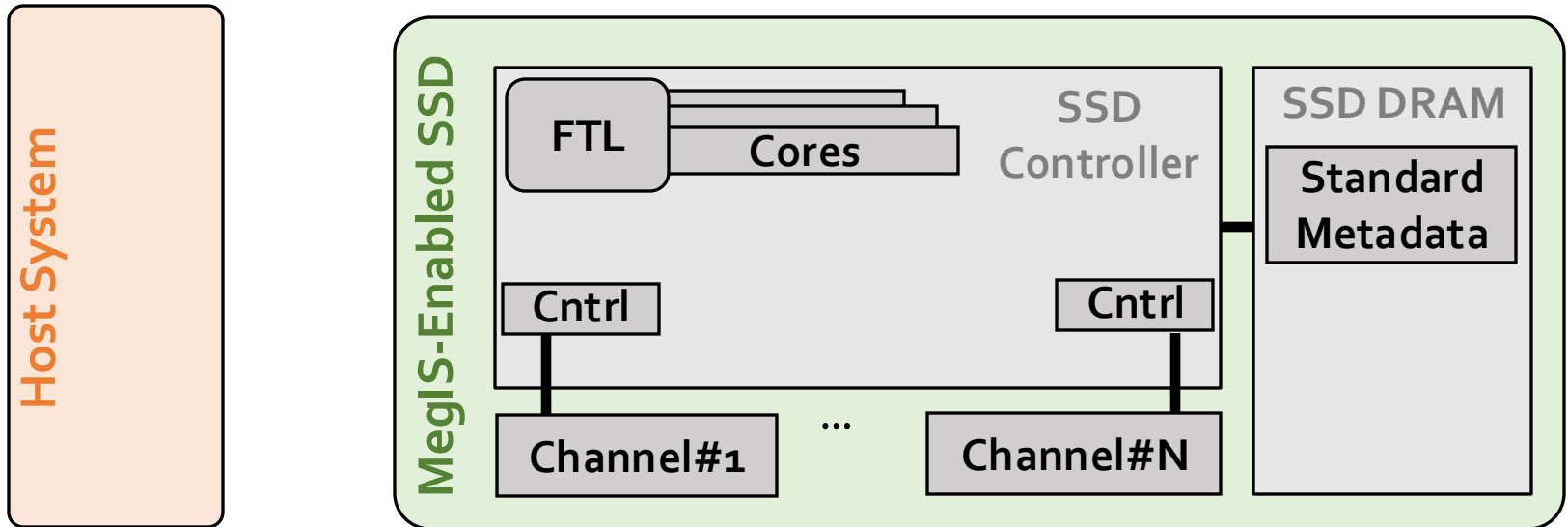
- First in-storage system for *end-to-end* metagenomic analysis
- **Idea:** Cooperative in-storage processing for metagenomic analysis
 - Hardware/software co-design between



MegIS's Steps



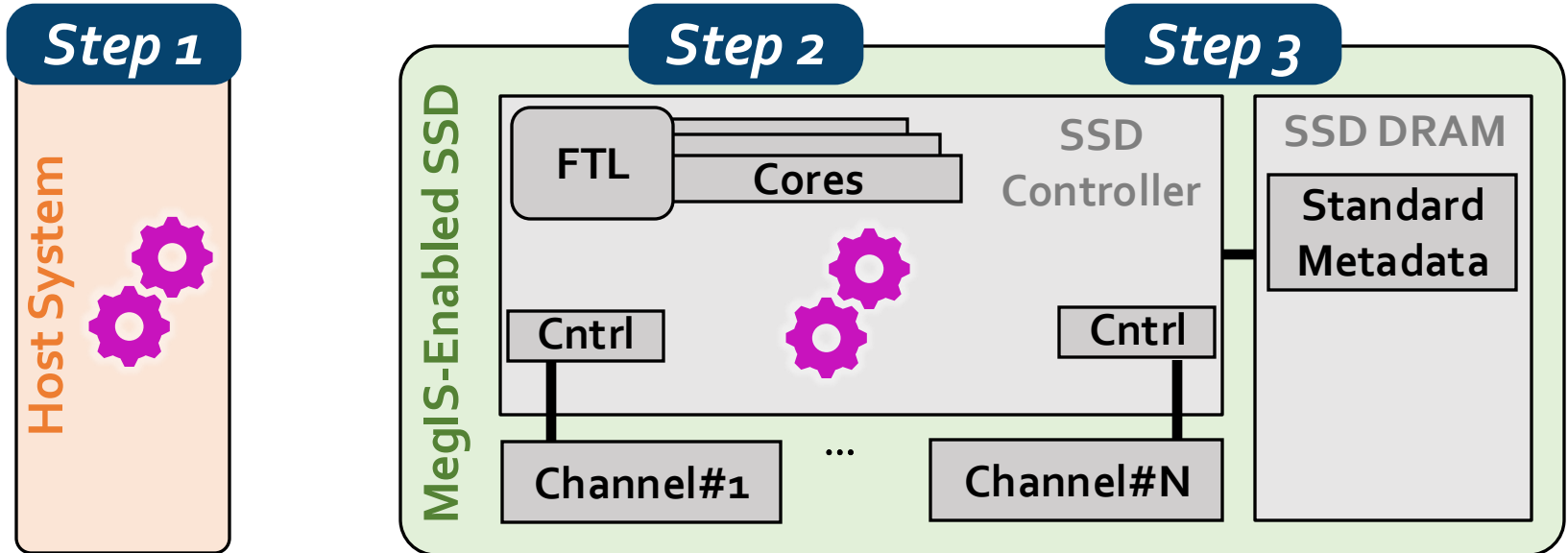
MegIS Hardware-Software Co-Design



MegIS Hardware-Software Co-Design

Task partitioning and mapping

- *Each step executes in its most suitable system*



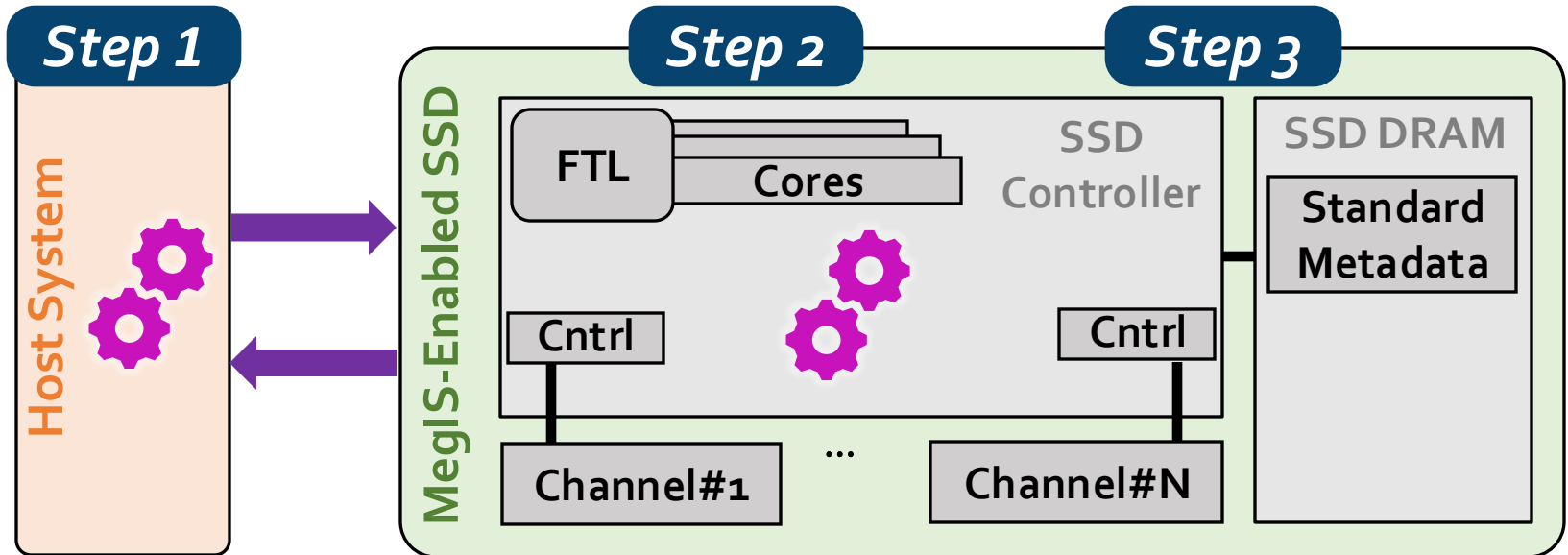
MegIS Hardware-Software Co-Design

Task partitioning and mapping

- Each step executes in its most suitable system

Data/computation flow coordination

- Reduce communication overhead
- Reduce #writes to flash chips



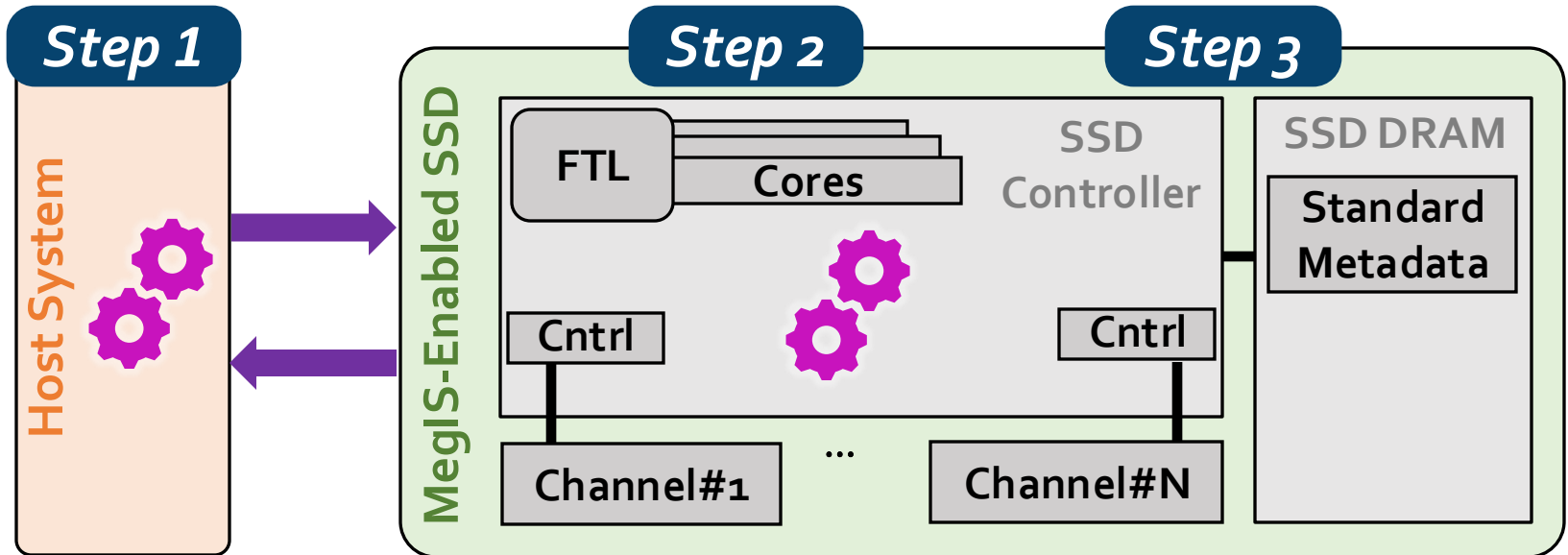
MegIS Hardware-Software Co-Design

Task partitioning and mapping

- *Each step executes in its most suitable system*

Data/computation flow coordination

- *Reduce communication overhead*
- *Reduce #writes to flash chips*



Storage-aware algorithms

- *Enable efficient access patterns to the SSD*

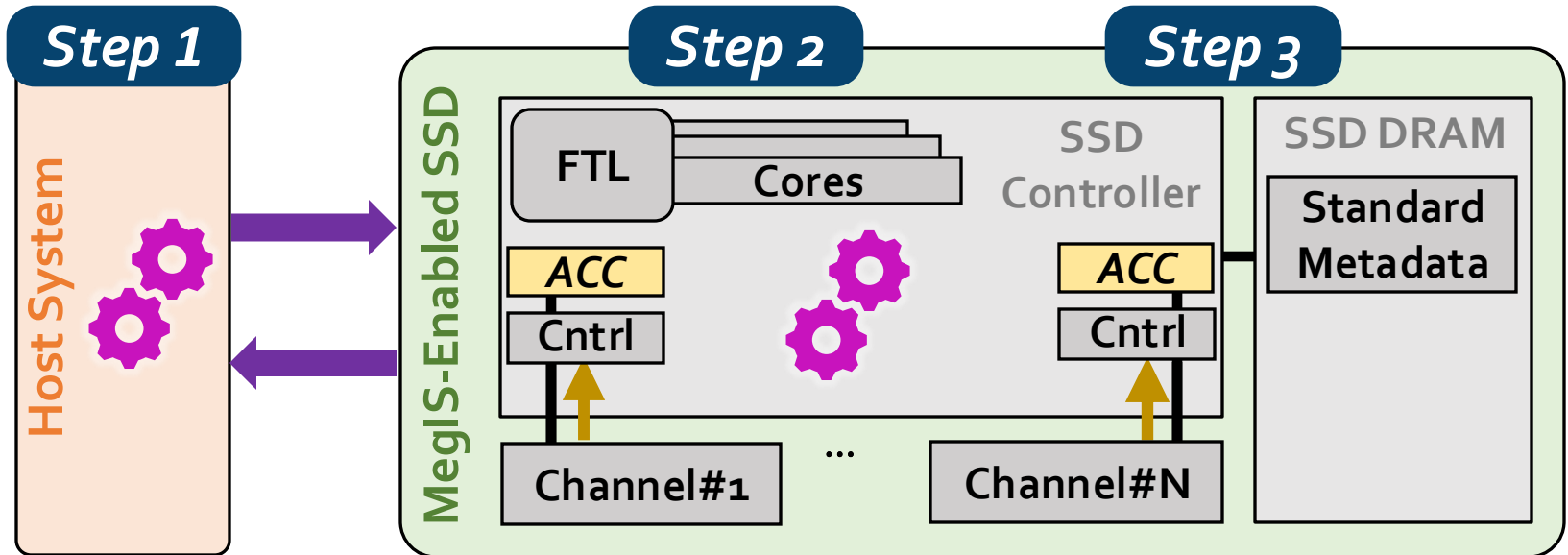
MegIS Hardware-Software Co-Design

Task partitioning and mapping

- Each step executes in its most suitable system

Data/computation flow coordination

- Reduce communication overhead
- Reduce #writes to flash chips



Storage-aware algorithms

- Enable efficient access patterns to the SSD

Lightweight in-storage accelerators

- Minimize SRAM/DRAM buffer spaces needed inside the SSD

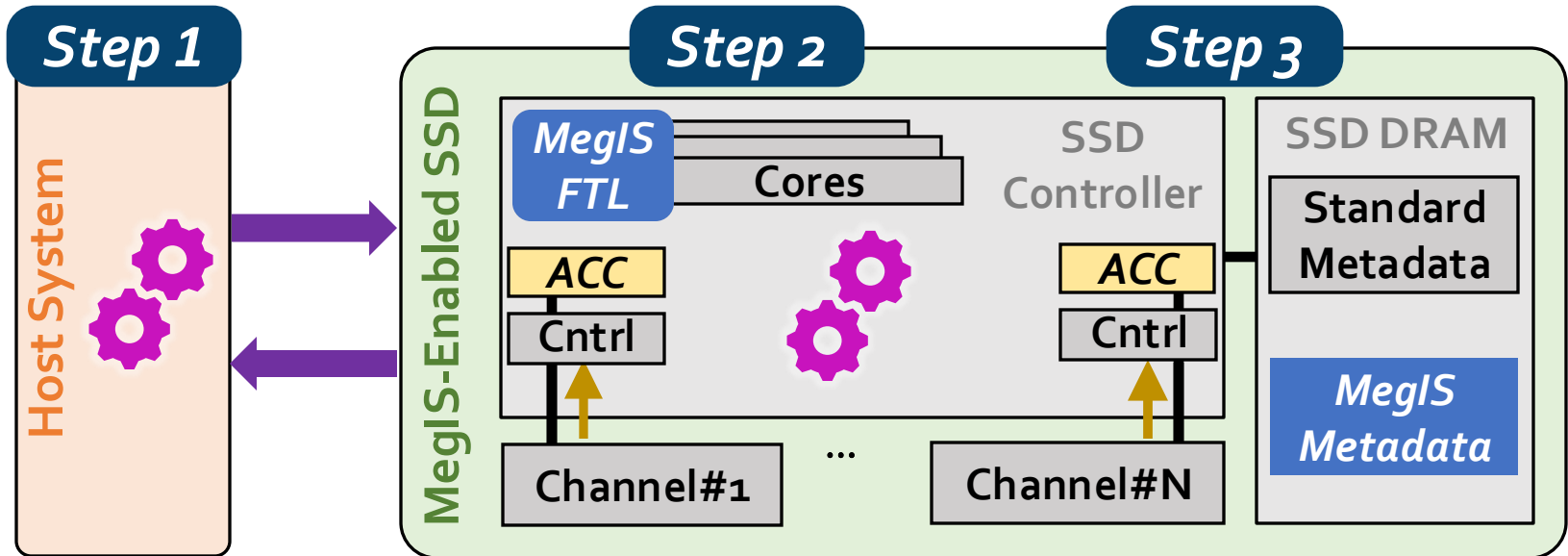
MegIS Hardware-Software Co-Design

Task partitioning and mapping

- Each step executes in its most suitable system

Data/computation flow coordination

- Reduce communication overhead
- Reduce #writes to flash chips



Storage-aware algorithms

- Enable efficient access patterns to the SSD

Lightweight in-storage accelerators

- Minimize SRAM/DRAM buffer spaces needed inside the SSD

Data mapping scheme and Flash Translation Layer (FTL)

- Specialize to the characteristics of metagenomic analysis
- Leverage the SSD's full internal bandwidth

Evaluation: Methodology Overview

Performance, Energy, and Power Analysis

Hardware Components

- Synthesized Verilog model for the in-storage accelerators
- MQSim [Tavakkol+, FAST'18] for SSD's internal operations
- Ramulator [Kim+, CAL'15] for SSD's internal DRAM

Software Components

Measure on a real system:

- AMD® EPYC® CPU with 128 physical cores
- 1-TB DRAM

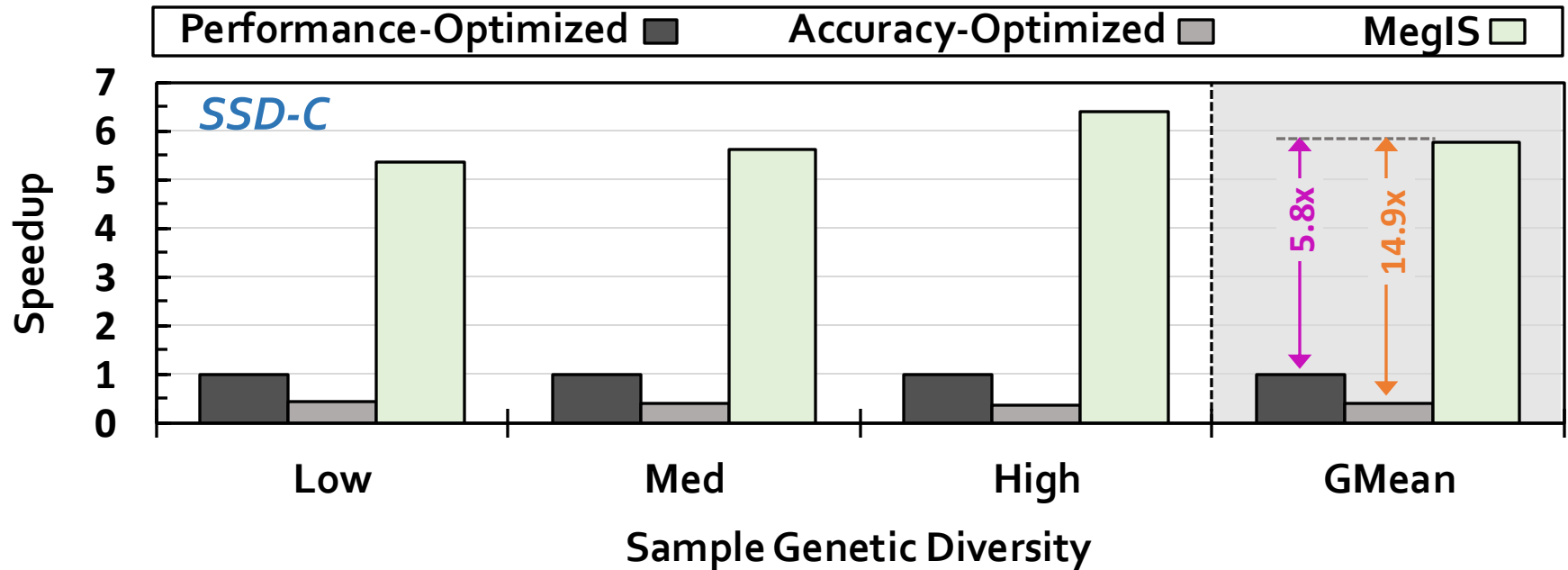
Baseline Comparison Points

- **Performance-optimized software**, Kraken2 [Genome Biology'19]
- **Accuracy-optimized software**, Metalign [Genome Biology'20]
- **PIM hardware-accelerated tool** (using processing-in-memory), Sieve [ISCA'21]

SSD Configurations

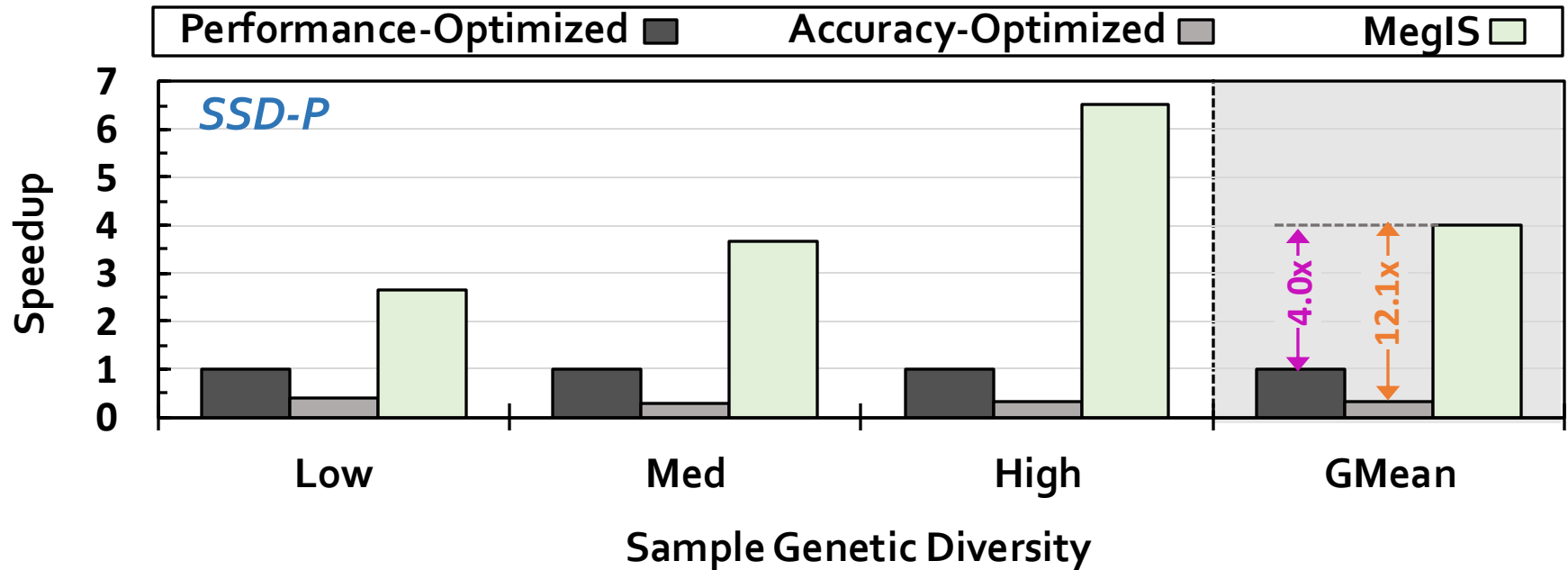
- **SSD-C**: with SATA3 interface (0.5 GB/s sequential read bandwidth)
- **SSD-P**: with PCIe Gen4 interface (7 GB/s sequential read bandwidth)

Evaluation: Speedup over the Software Baselines



MegIS provides significant speedup over both
Performance-Optimized and **Accuracy-Optimized** baselines

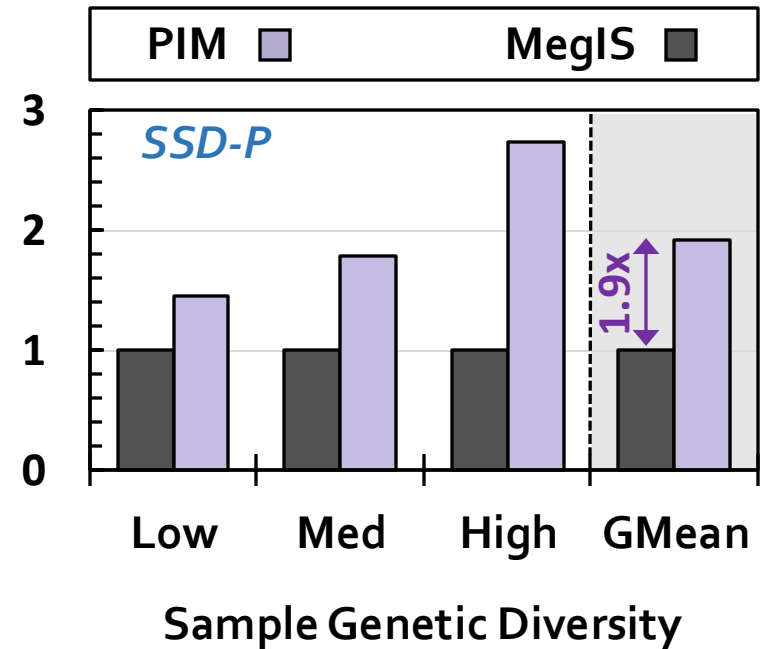
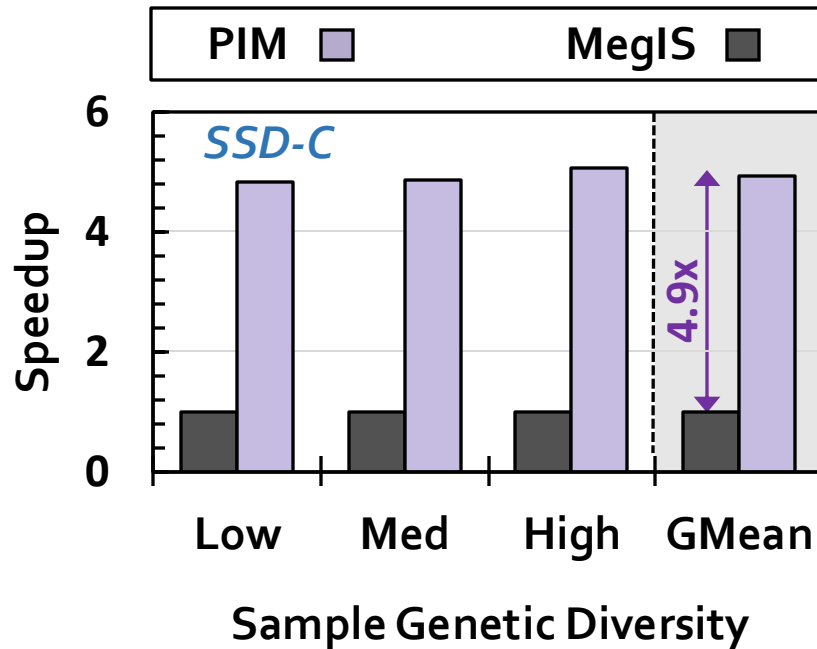
Evaluation: Speedup over the Software Baselines



MegIS provides significant speedup over both
Performance-Optimized and **Accuracy-Optimized** baselines

MegIS improves performance on both
cost-optimized and performance-optimized SSDs

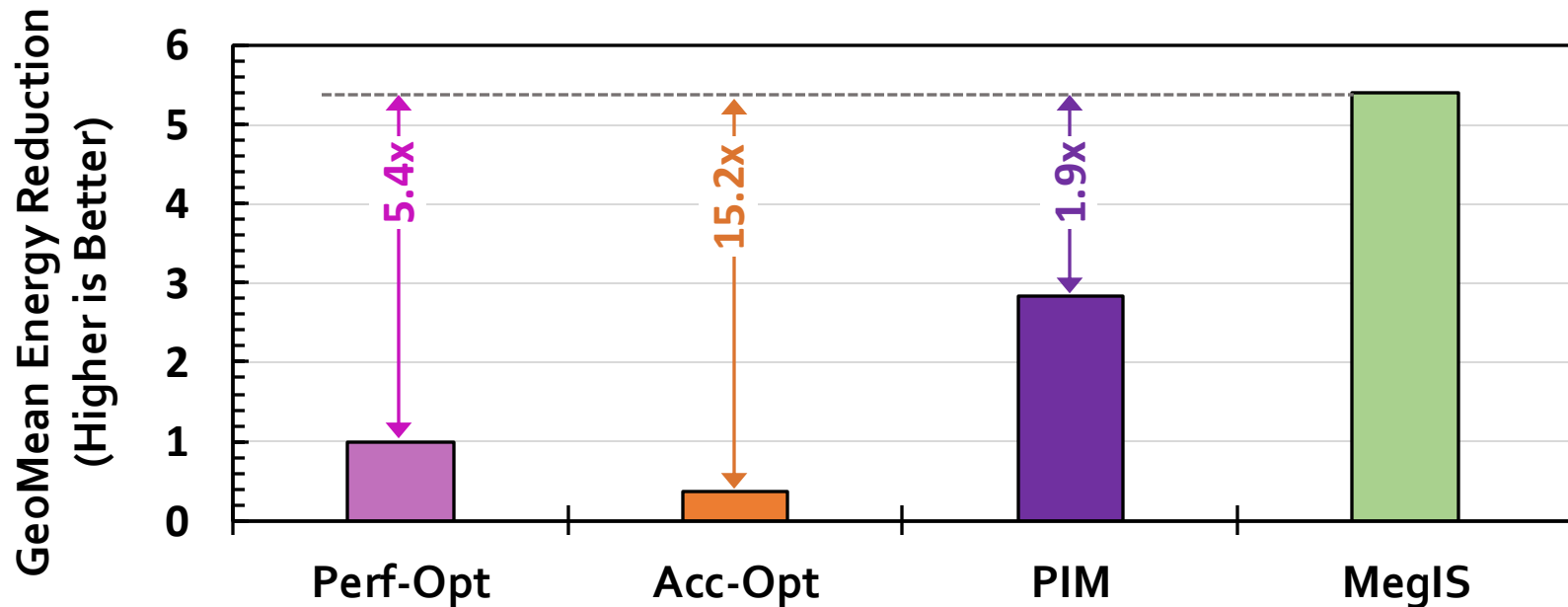
Evaluation: Speedup over the PIM Baseline



MegIS provides significant speedup over the **PIM** baseline

Evaluation: Reduction in Energy Consumption

- On average across different input sets and SSDs



MegIS provides significant energy reduction over the **Performance-Optimized**, **Accuracy-Optimized**, and **PIM** baselines

Evaluation: Accuracy, Area, and Power

Accuracy

- **Same accuracy** as the **accuracy-optimized** baseline
- **Significantly higher accuracy** than the **performance-optimized** and **PIM** baselines
 - 4.6 – 5.2× higher F1 scores
 - 3 – 24% lower L1 norm error

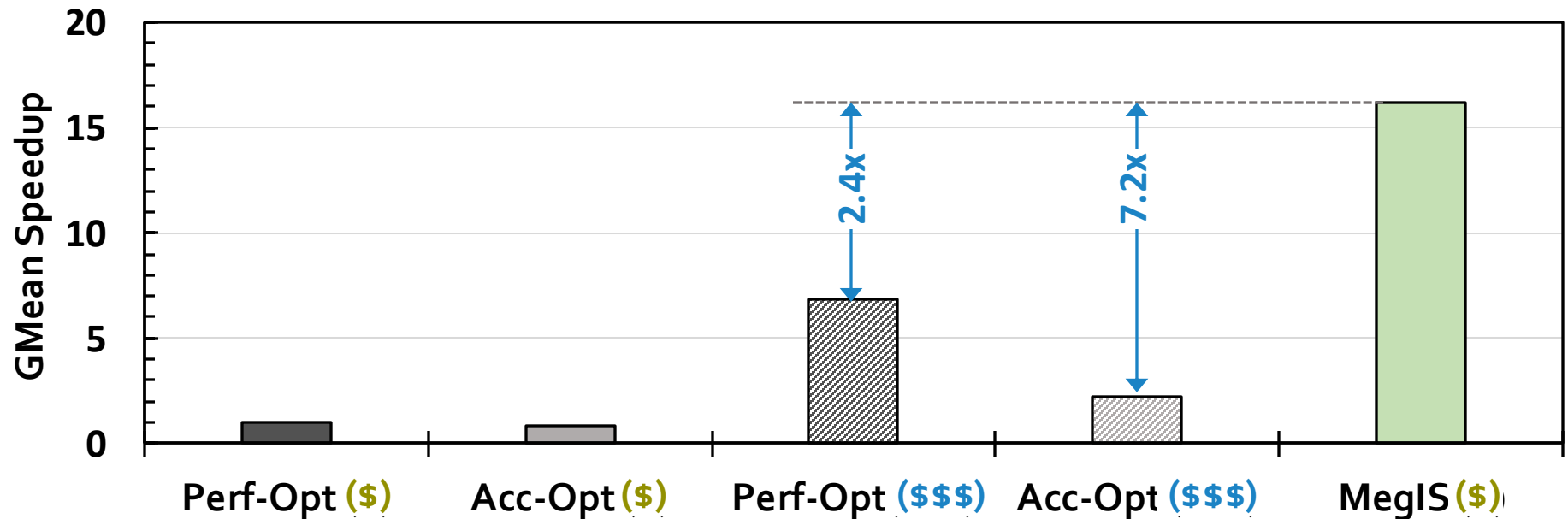
Area and Power

Total for an 8-channel SSD:

- **Area:** 0.04 mm² (*Only **1.7%** of the area of three ARM Cortex R4 cores in an SSD controller*)
- **Power:** 7.658 mW

Evaluation: System Cost-Efficiency

- **Cost-optimized system (\$):** With SSD-C and 64-GB DRAM
- **Performance-optimized system (\$\$\$):** With SSD-P and 1-TB DRAM



**MegIS outperforms the baselines
even when running on a much less costly system**

Evaluation: System Cost-Efficiency

- **Cost-optimized system (\$):** With SSD-C and 64-GB DRAM
- **Performance-optimized system (\$\$\$):** With SSD-P and 1-TB DRAM

20

**MegIS improves system cost-efficiency
and makes metagenomics more accessible
for wider adoption**

Perf-Opt (\$) Acc-Opt (\$) Perf-Opt (\$\$\$) Acc-Opt (\$\$\$) MegIS (\$)

MegIS outperforms the baselines
even when running on a much less costly system

In-Storage Metagenomics [ISCA 2024]

- Nika Mansouri Ghiasi, Mohammad Sadrosadati, Harun Mustafa, Arvid Gollwitzer, Can Firtina, Julien Eudine, Haiyu Mao, Joel Lindegger, Meryem Banu Cavlak, Mohammed Alser, Jisung Park, and Onur Mutlu,

"MegIS: High-Performance and Low-Cost Metagenomic Analysis with In-Storage Processing"

Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA), Buenos Aires, Argentina, July 2024.

[\[Slides \(pptx\) \(pdf\)\]](#)

[\[arXiv version\]](#)

MegIS: High-Performance, Energy-Efficient, and Low-Cost Metagenomic Analysis with In-Storage Processing

Nika Mansouri Ghiasi¹ Mohammad Sadrosadati¹ Harun Mustafa¹ Arvid Gollwitzer¹
Can Firtina¹ Julien Eudine¹ Haiyu Mao¹ Joël Lindegger¹ Meryem Banu Cavlak¹
Mohammed Alser¹ Jisung Park² Onur Mutlu¹
¹ETH Zürich ²POSTECH

Tools to Decide What to Execute Where

DAMOV Analysis Methodology & Workloads

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana-Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

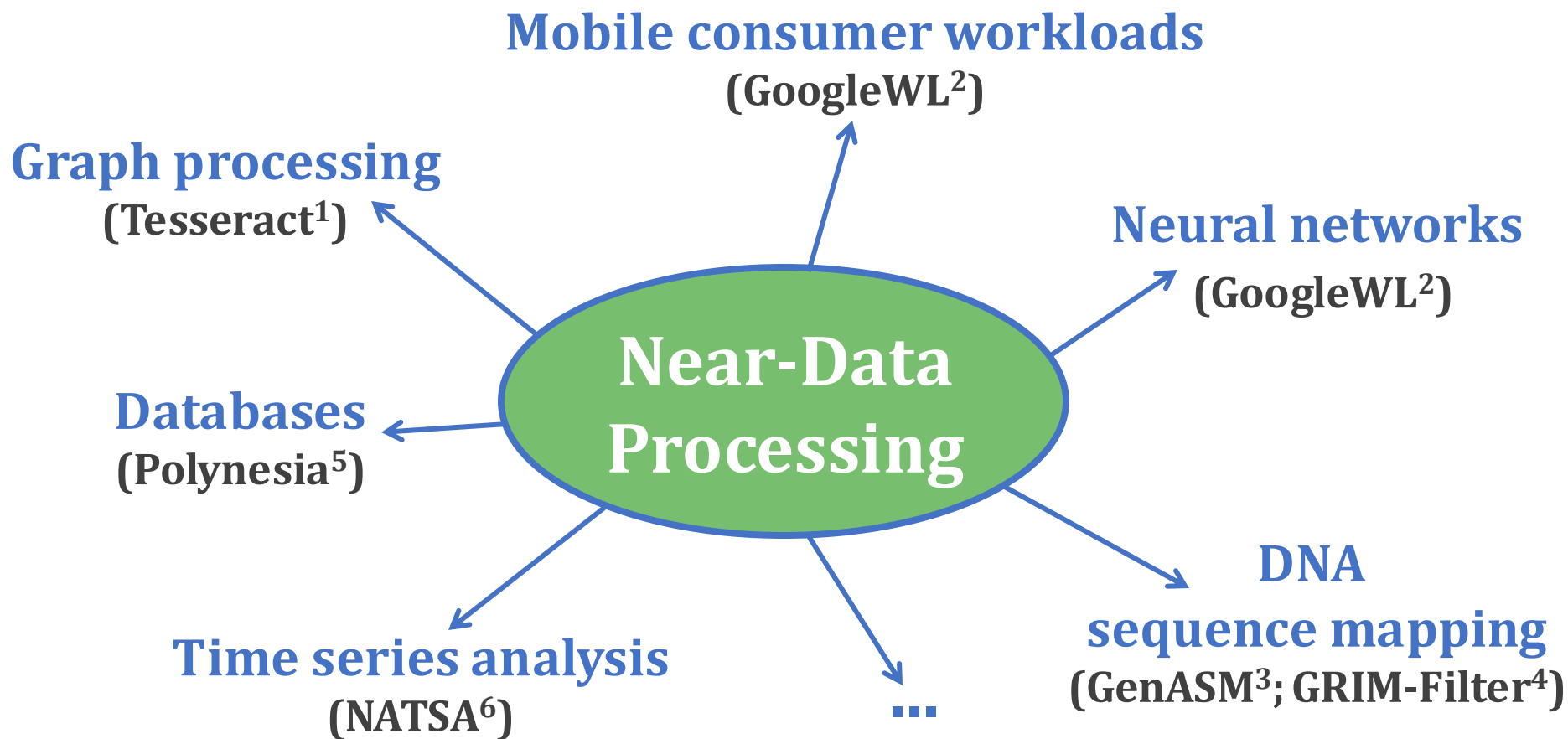
MOHAMMAD SADROSADATI, Institute for Research in Fundamental Sciences (IPM), Iran & ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Data movement between the CPU and main memory is a first-order obstacle against improving performance, scalability, and energy efficiency in modern systems. Computer systems employ a range of techniques to reduce overheads tied to data movement, spanning from traditional mechanisms (e.g., deep multi-level cache hierarchies, aggressive hardware prefetchers) to emerging techniques such as Near-Data Processing (NDP), where some computation is moved close to memory. Prior NDP works investigate the root causes of data movement bottlenecks using different profiling methodologies and tools. However, there is still a lack of understanding about the key metrics that can identify different data movement bottlenecks and their relation to traditional and emerging data movement mitigation mechanisms. Our goal is to methodically identify potential sources of data movement over a broad set of applications and to comprehensively compare traditional compute-centric data movement mitigation techniques (e.g., caching and prefetching) to more memory-centric techniques (e.g., NDP), thereby developing a rigorous understanding of the best techniques to mitigate each source of data movement.

With this goal in mind, we perform the first large-scale characterization of a wide variety of applications, across a wide range of application domains, to identify fundamental program properties that lead to data movement to/from main memory. We develop the first systematic methodology to classify applications based on the sources contributing to data movement bottlenecks. From our large-scale characterization of 77K functions across 345 applications, we select 144 functions to form the first open-source benchmark suite (DAMOV) for main memory data movement studies. We select a diverse range of functions that (1) represent different types of data movement bottlenecks, and (2) come from a wide range of application domains. Using NDP as a case study, we identify new insights about the different data movement bottlenecks and use these insights to determine the most suitable data movement mitigation mechanism for a particular application. We open-source DAMOV and the complete source code for our new characterization methodology at <https://github.com/CMU-SAFARI/DAMOV>.

When to Employ Near-Data Processing?



[1] Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing," ISCA, 2015

[2] Boroumand+, "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS, 2018

[3] Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis," MICRO, 2020

[4] Kim+, "GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies," BMC Genomics, 2018

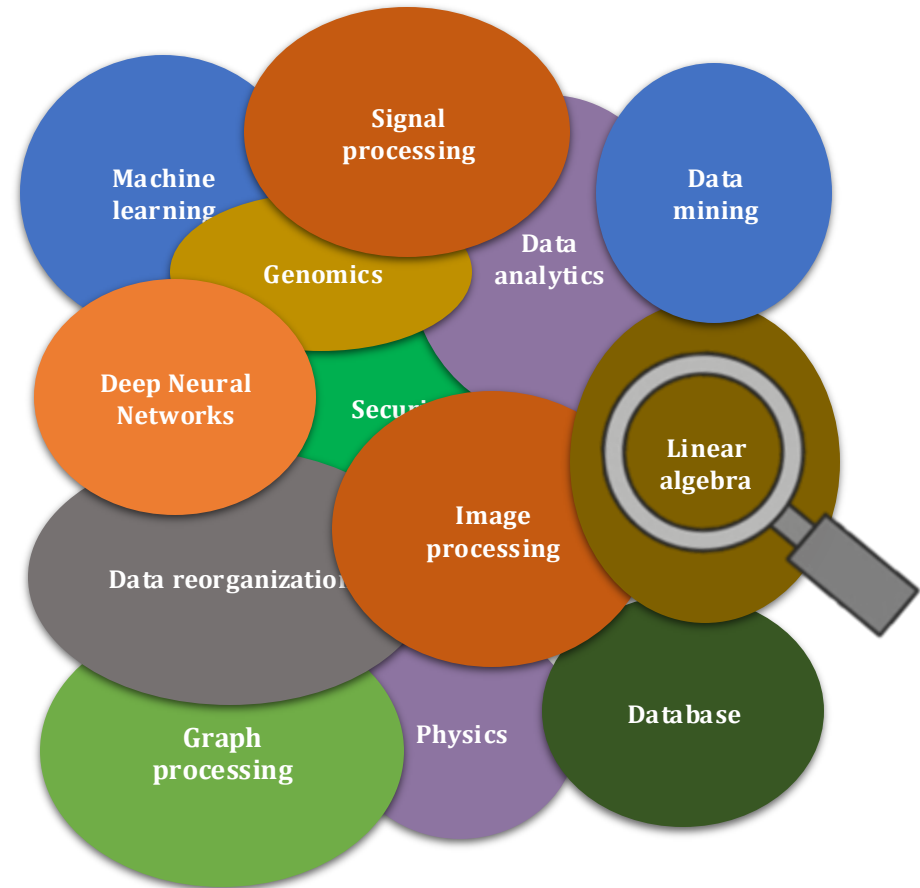
[5] Boroumand+, "Polynesia: Enabling Effective Hybrid Transactional/Analytical Databases with Specialized Hardware/Software Co-Design," arXiv:2103.00798 [cs.AR], 2021

[6] Fernandez+, "NATSA: A Near-Data Processing Accelerator for Time Series Analysis," ICCD, 2020

Step 1: Application Profiling

- We analyze 345 applications from distinct domains:

- Graph Processing
- Deep Neural Networks
- Physics
- High-Performance Computing
- Genomics
- Machine Learning
- Databases
- Data Reorganization
- Image Processing
- Map-Reduce
- Benchmarking
- Linear Algebra



Step 3: Memory Bottleneck Analysis

**Six classes of
data movement bottlenecks:**

each class \leftrightarrow data movement
mitigation mechanism

Memory Bottleneck Class

1a: *DRAM
Bandwidth*

1b: *DRAM Latency*

1c: *L1/L2
Cache Capacity*

2a: *L3 Cache
Contention*

2b: *L1 Cache
Capacity*

2c: *Compute-Bound*

DAMOV is Open Source

- We open-source our **benchmark suite** and our **toolchain**

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing. Described by Oliveira et al. (preliminary version at <https://arxiv.org/pdf/2105.03725.pdf>)

Readme

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages



omutlu Update README.md

ce1b4ea 17 days ago 5 commits

simulator	Cleaning	19 days ago
README.md	Update README.md	17 days ago
get_workloads.sh	DAMOV -- first commit	19 days ago

README.md

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including [BWA](#), [Chai](#), [Darknet](#), [GASE](#), [Hardware Effects](#), [Hashjoin](#), [HPCC](#), [HPCG](#), [Ligra](#), [PARSEC](#), [Parboil](#), [PolyBench](#), [Phoenix](#), [Rodinia](#), [SPLASH-2](#), [STREAM](#).

DAMOV-SIM

DAMOV
Benchmarks

SAFARI

DAMOV is Open Source

- Benchmark suite and toolchain are open sourced

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About

DAMOV is a benchmark suite and a

Get DAMOV at:

<https://github.com/CMU-SAFARI/DAMOV>

README.md

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including [BWA](#), [Chai](#), [Darknet](#), [GASE](#), [Hardware Effects](#), [Hashjoin](#), [HPCC](#), [HPCG](#), [Ligra](#), [PARSEC](#), [Parboil](#), [PolyBench](#), [Phoenix](#), [Rodinia](#), [SPLASH-2](#), [STREAM](#).

Readme

Releases

No releases published
[Create a new release](#)

Packages

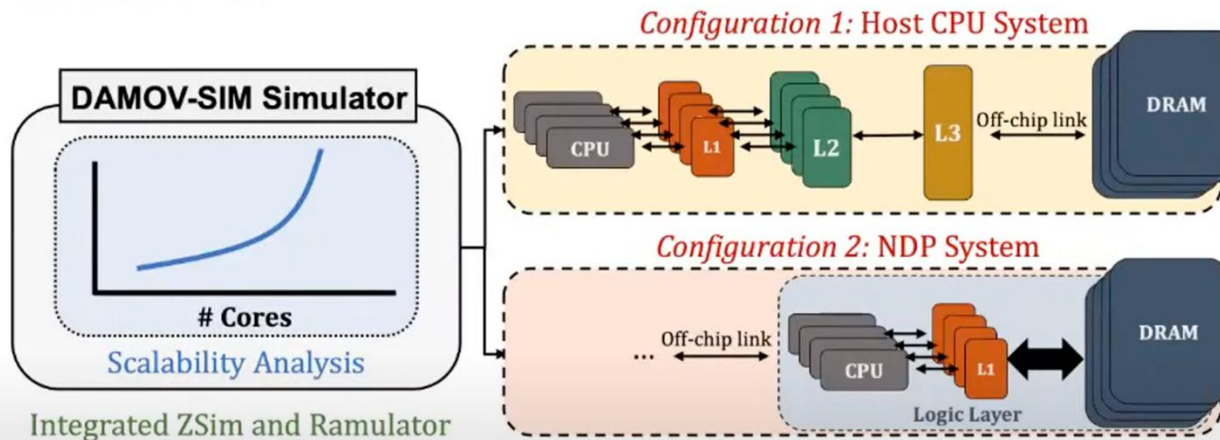
No packages published
[Publish your first package](#)

Languages

More on DAMOV Analysis Methodology & Workloads

Step 3: Memory Bottleneck Classification (2/)

- **Goal:** identify the specific sources of data movement bottlenecks



- **Scalability Analysis:**
 - 1, 4, 16, 64, and 256 out-of-order/in-order host and NDP CPU cores
 - 3D-stacked memory as main memory

SAFARI DAMOV-SIM: <https://github.com/CMU-SAFARI/DAMOV> 30

SAFARI Live Seminar: DAMOV: A New Methodology & Benchmark Suite for Data Movement Bottlenecks

352 views • Streamed live on Jul 22, 2021

18 0 SHARE SAVE ...



Onur Mutlu Lectures
17.7K subscribers

ANALYTICS

EDIT VIDEO

https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=3

More on DAMOV Methods & Benchmarks

- Geraldo F. Oliveira, Juan Gomez-Luna, Lois Orosa, Saugata Ghose, Nandita Vijaykumar, Ivan fernandez, Mohammad Sadrosadati, and Onur Mutlu,
"DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks"
IEEE Access, 8 September 2021.
Preprint in arXiv, 8 May 2021.
[[arXiv preprint](#)]
[[IEEE Access version](#)]
[[DAMOV Suite and Simulator Source Code](#)]
[[SAFARI Live Seminar Video](#) (2 hrs 40 mins)]
[[Short Talk Video](#) (21 minutes)]

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

MOHAMMAD SADROSADATI, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Ramulator 2.0 for PIM Systems

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
"Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
*Preprint on **arxiv**, August 2023.*
[\[arXiv version\]](#)
[\[Ramulator 2.0 Source Code\]](#)

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

<https://arxiv.org/pdf/2308.11030.pdf>

MQSim: Simulating Storage [FAST 2018]

- Arash Tavakkol, Juan Gomez-Luna, Mohammad Sadrosadati, Saugata Ghose, and Onur Mutlu,
"MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices"
Proceedings of the 16th USENIX Conference on File and Storage Technologies (FAST), Oakland, CA, USA, February 2018.
[Slides (pptx)] [pdf]
[Source Code]

MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices

Arash Tavakkol[†], Juan Gómez-Luna[†], Mohammad Sadrosadati[†], Saugata Ghose[‡], Onur Mutlu^{†‡}

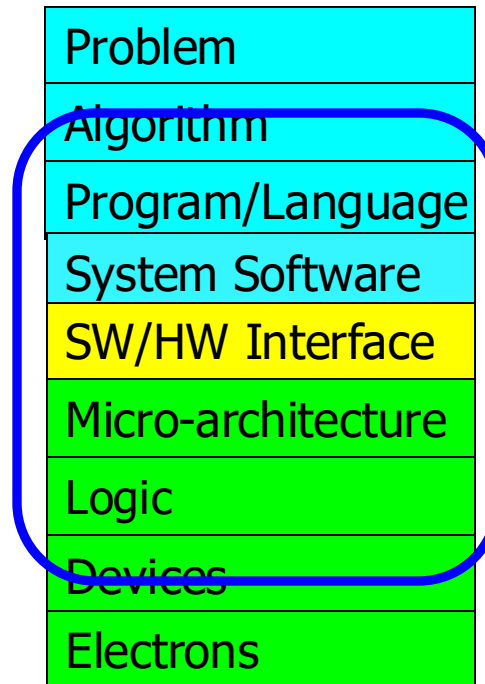
[†]*ETH Zürich*

[‡]*Carnegie Mellon University*

<https://github.com/CMU-SAFARI/MQSim>

We Need to Revisit the Entire Stack

- With a **memory-centric mindset**



We can get there step by step

PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu^{a,b}, Saugata Ghose^{b,c}, Juan Gómez-Luna^a, Rachata Ausavarungnirun^d

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

^d*King Mongkut's University of Technology North Bangkok*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,

"A Modern Primer on Processing in Memory"

*Invited Book Chapter in **Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann**, Springer, to be published in 2021.*

PIM Review and Open Problems (II)

A Workload and Programming Ease Driven Perspective of Processing-in-Memory

Saugata Ghose[†] Amirali Boroumand[†] Jeremie S. Kim^{†§} Juan Gómez-Luna[§] Onur Mutlu^{§†}

[†]*Carnegie Mellon University*

[§]*ETH Zürich*

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu,

"Processing-in-Memory: A Workload-Driven Perspective"

Invited Article in IBM Journal of Research & Development, Special Issue on Hardware for Artificial Intelligence, to appear in November 2019.

[Preliminary arXiv version]

Processing in Memory: Adoption Challenges

1. Processing **using** Memory
2. Processing **near** Memory

How to Enable Adoption of Processing in Memory

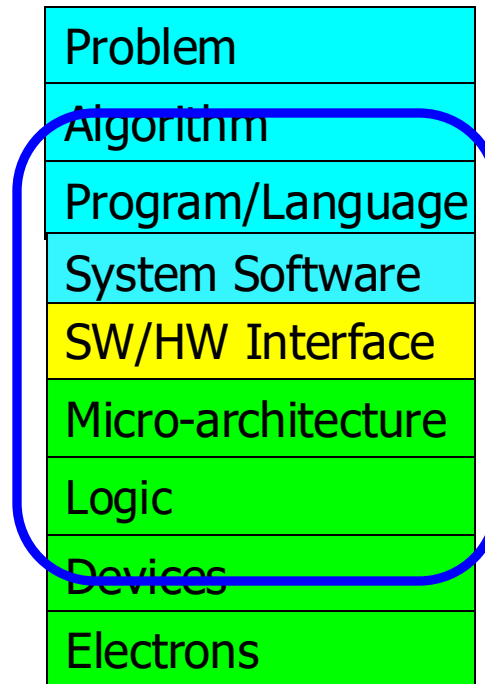
Potential Barriers to Adoption of PIM

1. **Applications & software** for PIM
2. Ease of **programming** (interfaces and compiler/HW support)
3. **System** and **security** support: coherence, synchronization, virtual memory, isolation, communication interfaces, ...
4. **Runtime** and **compilation** systems for adaptive scheduling, data mapping, access/sharing control, ...
5. **Infrastructures** to assess benefits and feasibility

All can be solved with change of mindset

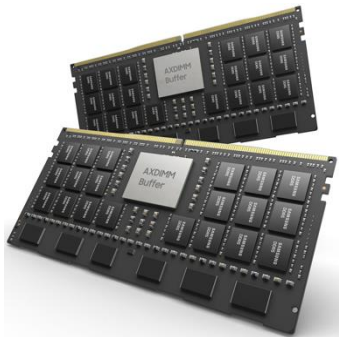
We Need to Revisit the Entire Stack

- With a **memory-centric mindset**

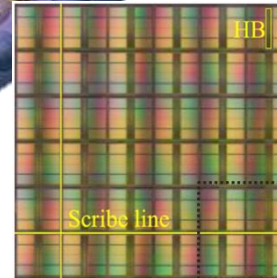
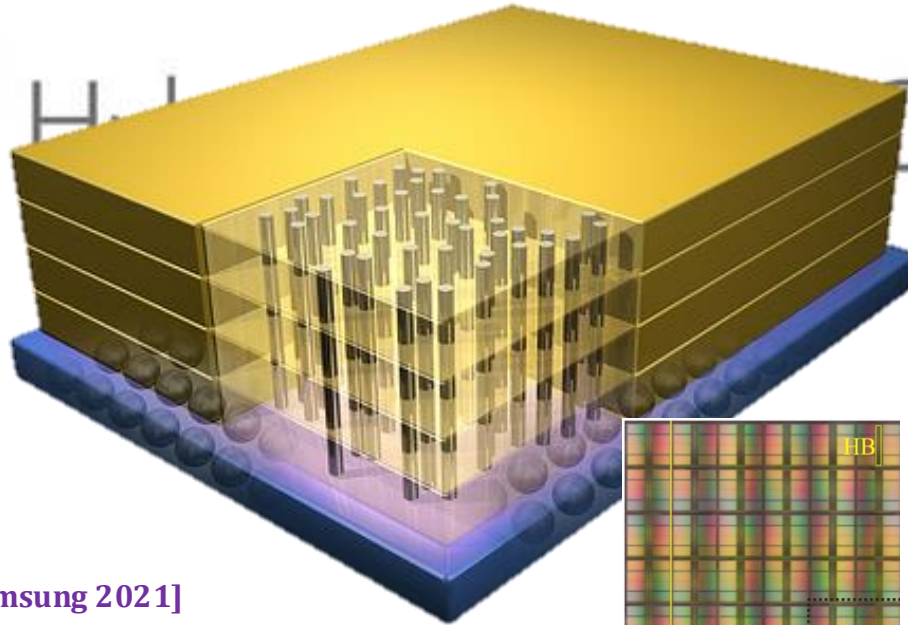


We can get there step by step

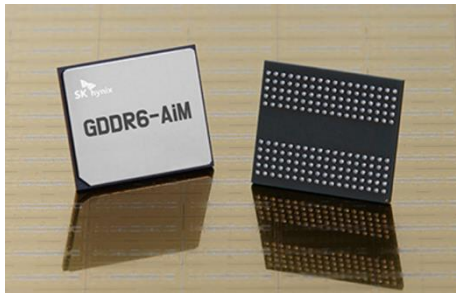
Processing-in-Memory Landscape Today



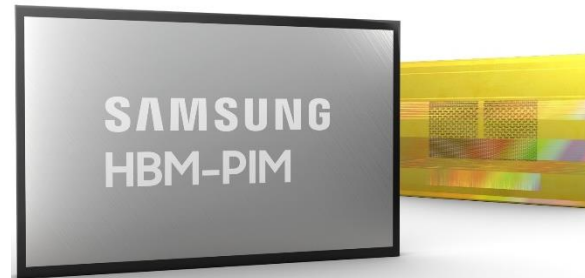
[Samsung 2021]



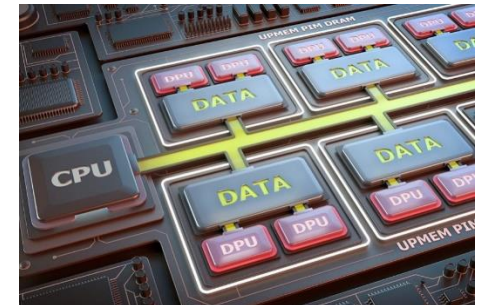
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]



[UPMEM 2019]

Adoption: How to Keep It Simple?

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoun Choi, **"PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"** *Proceedings of the 42nd International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2015.
[[Slides \(pdf\)](#)] [[Lightning Session Slides \(pdf\)](#)]

PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture

Junwhan Ahn Sungjoo Yoo Onur Mutlu[†] Kiyoun Choi

junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

[†]Carnegie Mellon University

Adoption: How to Keep It Simple?

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoun Choi, **"PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"** *Proceedings of the 42nd International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2015.
[[Slides \(pdf\)](#)] [[Lightning Session Slides \(pdf\)](#)]

PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture

Junwhan Ahn Sungjoo Yoo Onur Mutlu[†] Kiyoun Choi

junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

[†]Carnegie Mellon University

Adoption: How to Ease Programmability? (I)

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler, **"Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems"**

Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Session Slides \(pptx\)](#) ([pdf](#))]

Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems

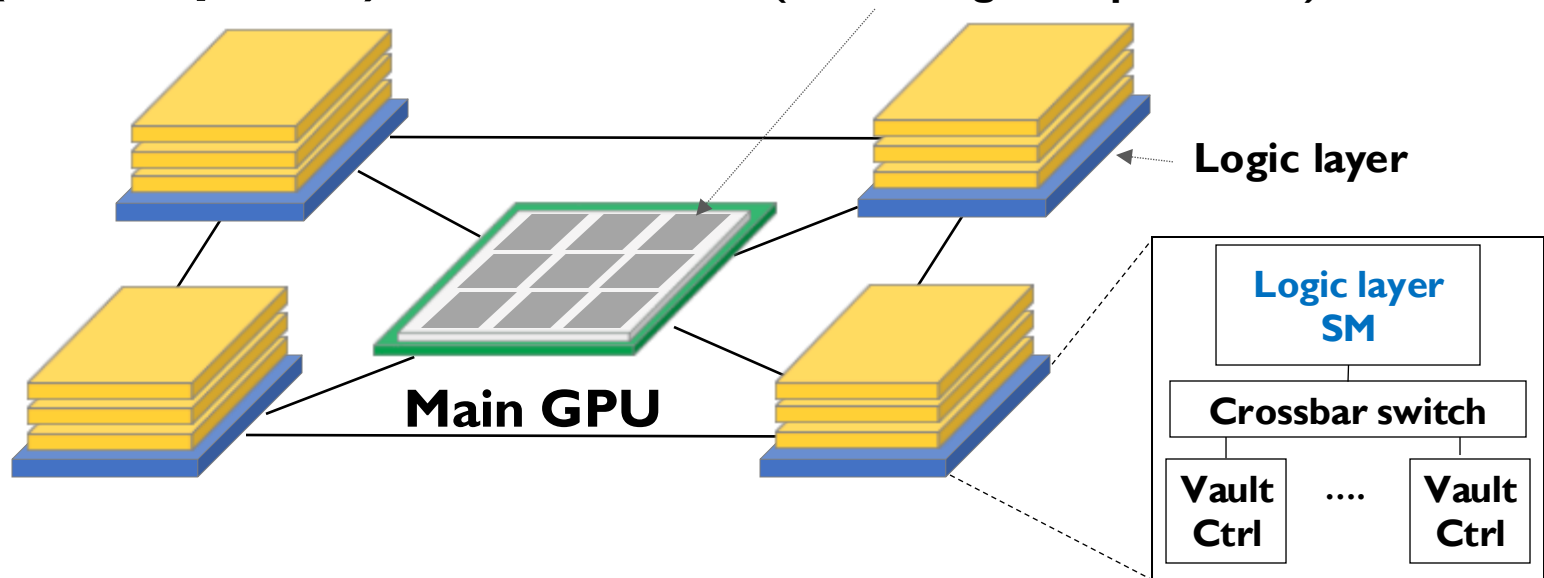
Kevin Hsieh[‡] Eiman Ebrahimi[†] Gwangsun Kim* Niladrish Chatterjee[†] Mike O'Connor[†]
Nandita Vijaykumar[‡] Onur Mutlu^{§‡} Stephen W. Keckler[†]

[‡]Carnegie Mellon University [†]NVIDIA ^{*}KAIST [§]ETH Zürich

Truly Distributed GPU Processing with PIM

**3D-stacked memory
(memory stack)**

SM (Streaming Multiprocessor)



```
__global__  
void applyScaleFactorsKernel( uint8_T * const out,  
                             uint8_T const * const in, const double *factor,  
                             size_t const numRows, size_t const numCols )  
{  
    // Work out which pixel we are working on.  
    const int rowIdx = blockIdx.x * blockDim.x + threadIdx.x;  
    const int colIdx = blockIdx.y;  
    const int sliceIdx = threadIdx.z;  
  
    // Check this thread isn't off the image  
    if( rowIdx >= numRows ) return;  
  
    // Compute the index of my element  
    size_t linearIdx = rowIdx + colIdx*numRows +  
                      sliceIdx*numRows*numCols;
```


Adoption: How to Ease Programmability? (II)

- Geraldo F. Oliveira, Alain Kohli, David Novo, Juan Gómez-Luna, Onur Mutlu,
“DaPPA: A Data-Parallel Framework for Processing-in-Memory Architectures,”
in *PACT SRC Student Competition*, Vienna, Austria, October 2023.

DaPPA: A Data-Parallel Framework for Processing-in-Memory Architectures

Geraldo F. Oliveira*

Alain Kohli*

David Novo[‡]

Juan Gómez-Luna*

Onur Mutlu*

**ETH Zürich*

[‡]*LIRMM, Univ. Montpellier, CNRS*

Adoption: How to Ease Programmability? (III)

- Jinfan Chen, Juan Gómez-Luna, Izzat El Hajj, YuXin Guo, and Onur Mutlu,
"SimplePIM: A Software Framework for Productive and Efficient Processing in Memory"
Proceedings of the 32nd International Conference on Parallel Architectures and Compilation Techniques (PACT), Vienna, Austria, October 2023.

SimplePIM: A Software Framework for Productive and Efficient Processing-in-Memory

Jinfan Chen¹ Juan Gómez-Luna¹ Izzat El Hajj² Yuxin Guo¹ Onur Mutlu¹
¹ETH Zürich ²American University of Beirut

Adoption: How to Ease Programmability? (IV)

- Geraldo F. Oliveira, Juan Gomez-Luna, Lois Orosa, Saugata Ghose, Nandita Vijaykumar, Ivan fernandez, Mohammad Sadrosadati, and Onur Mutlu,
"DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks"
IEEE Access, 8 September 2021.
Preprint in arXiv, 8 May 2021.
[[arXiv preprint](#)]
[[IEEE Access version](#)]
[[DAMOV Suite and Simulator Source Code](#)]
[[SAFARI Live Seminar Video](#) (2 hrs 40 mins)]
[[Short Talk Video](#) (21 minutes)]

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

MOHAMMAD SADROSADATI, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Adoption: How to Ease Programmability? (V)

■ Appears in IEEE TETC 2023

ALP: Alleviating CPU-Memory Data Movement Overheads in Memory-Centric Systems

Nika Mansouri Ghiasi, Nandita Vijaykumar, Geraldo F. Oliveira, Lois Orosa, Ivan Fernandez, Mohammad Sadrosadati, Konstantinos Kanellopoulos, Nastaran Hajinazar, Juan Gómez Luna, Onur Mutlu

Abstract—Recent advances in memory technology have enabled near-data processing (NDP) to tackle main memory bottlenecks in modern systems. Prior works partition applications into segments (e.g., instructions, loops, functions) and execute memory-bound segments of the applications on NDP computation units, while mapping the cache-friendly application segments to host CPU cores that access a deeper cache hierarchy. Partitioning applications between NDP and host cores causes inter-segment data movement overhead, which is the overhead from moving data generated from one segment and used in the consecutive segments. This overhead can be large if the segments map to cores in different parts of the system (i.e., host and NDP). Prior works take two approaches to the inter-segment data movement overhead when partitioning applications between NDP and host cores. The first class of works maps segments to NDP or host cores based on the properties of each segment, neglecting the performance impact of the inter-segment data movement. Such partitioning techniques suffer from inter-segment data movement overhead. The second class of works maps segments to host or NDP cores based on the overall memory bandwidth savings of each segment (which depends on the memory bandwidth savings within each segment and the inter-segment data movement overhead between other segments). These works do not offload each segment to the best-fitting core if they incur high inter-segment data movement overhead. Therefore these works miss some of the potential NDP performance benefits. We show that mapping each segment (here basic block) to its best-fitting core based on the properties of each segment, assuming no inter-segment data movement, can provide substantial performance benefits. However, we show that the inter-segment data movement reduces this benefit significantly.

To this end, we introduce ALP, a new programmer-transparent technique to leverage the performance benefits of NDP by *alleviating* the performance impact of inter-segment data movement between host and memory and enabling efficient partitioning of applications between host and NDP cores. ALP alleviates the inter-segment data movement overhead by *proactively and accurately* transferring the required data between the segments mapped on host and NDP cores. This is based on the key observation that the instructions that generate the inter-segment data stay the same across different executions of a program on different input sets. ALP uses a compiler pass to identify these instructions and uses specialized hardware support to transfer data between the host and NDP cores at runtime. Using both the compiler and runtime information, ALP efficiently maps application segments to either host or NDP cores considering 1) the properties of each segment, 2) the inter-segment data movement overhead between different segments, and 3) whether this inter-segment data movement overhead can be alleviated proactively and in a timely manner. We evaluate ALP across a wide range of workloads and show on average 54.3% and 45.4% speedup compared to executing the application only on the host CPU or only the NDP cores, respectively.

Adoption: How to Maintain Coherence? (I)

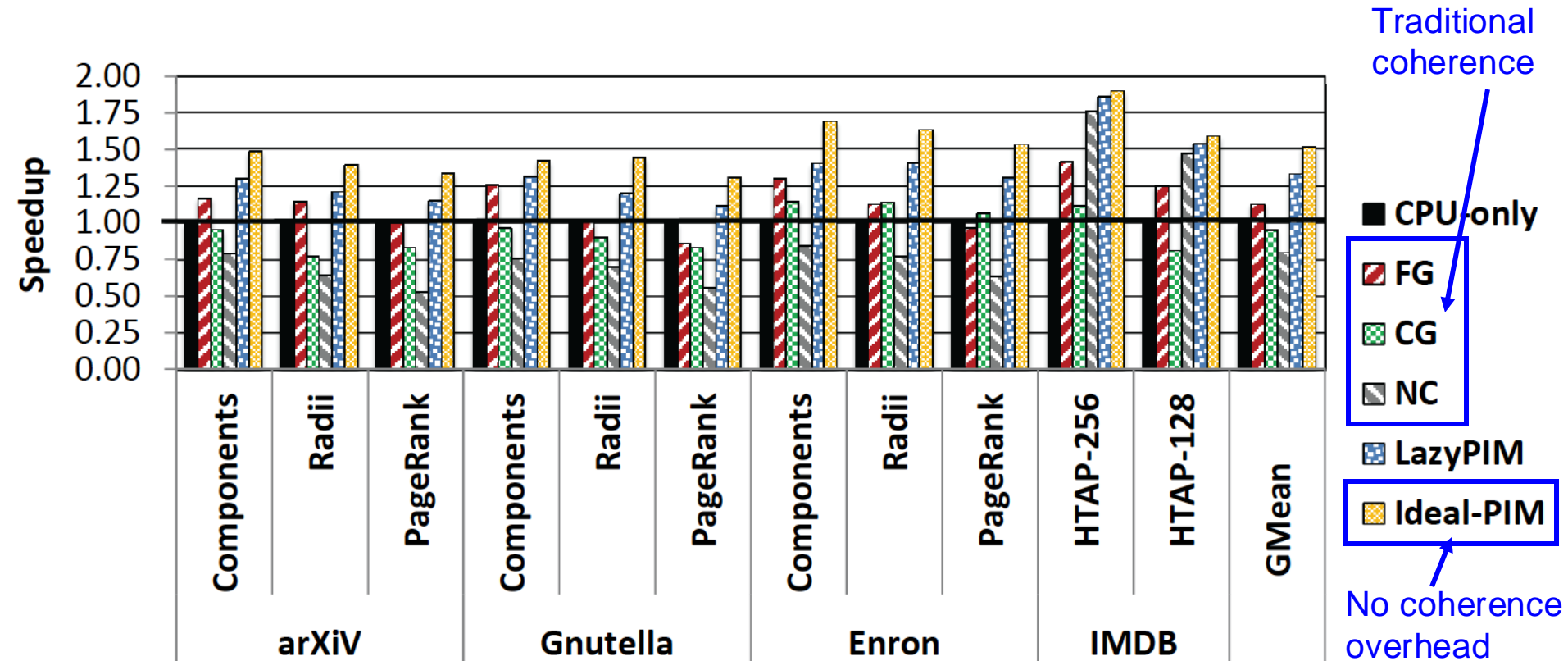
- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,
["LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory"](#)
[IEEE Computer Architecture Letters \(CAL\)](#), June 2016.

LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory

Amirali Boroumand[†], Saugata Ghose[†], Minesh Patel[†], Hasan Hassan^{†§}, Brandon Lucia[†],
Kevin Hsieh[†], Krishna T. Malladi^{*}, Hongzhong Zheng^{*}, and Onur Mutlu^{††}

[†] *Carnegie Mellon University* ^{*} *Samsung Semiconductor, Inc.* [§] *TOBB ETÜ* [‡] *ETH Zürich*

Challenge: Coherence for Hybrid CPU-PIM Apps



Adoption: How to Maintain Coherence? (II)

- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,
"CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators"
Proceedings of the 46th International Symposium on Computer Architecture (ISCA), Phoenix, AZ, USA, June 2019.

CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators

Amirali Boroumand[†]

Saugata Ghose[†]

Minesh Patel[★]

Hasan Hassan[★]

Brandon Lucia[†]

Rachata Ausavarungnirun^{†‡}

Kevin Hsieh[†]

Nastaran Hajinazar^{◇†}

Krishna T. Malladi[§]

Hongzhong Zheng[§]

Onur Mutlu^{★†}

[†]Carnegie Mellon University

[★]ETH Zürich

[‡]KMUTNB

[◇]Simon Fraser University

[§]Samsung Semiconductor, Inc.

Adoption: How to Support Synchronization?

- Christina Giannoula, Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas, Ivan Fernandez, Juan Gómez-Luna, Lois Orosa, Nectarios Koziris, Georgios Goumas, Onur Mutlu, **["SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures"](#)**
Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Short Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (21 minutes)]
[[Short Talk Video](#) (7 minutes)]

SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures

Christina Giannoula^{†‡} Nandita Vijaykumar^{*‡} Nikela Papadopoulou[†] Vasileios Karakostas[†] Ivan Fernandez^{§‡}
Juan Gómez-Luna[‡] Lois Orosa[‡] Nectarios Koziris[†] Georgios Goumas[†] Onur Mutlu[‡]
[†]*National Technical University of Athens* [‡]*ETH Zürich* ^{*}*University of Toronto* [§]*University of Malaga*

Adoption: How to Support Virtual Memory?

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,
["Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation"](#)
Proceedings of the 34th IEEE International Conference on Computer Design (ICCD), Phoenix, AZ, USA, October 2016.

Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation

Kevin Hsieh[†] Samira Khan[‡] Nandita Vijaykumar[†]
Kevin K. Chang[†] Amirali Boroumand[†] Saugata Ghose[†] Onur Mutlu^{§†}
[†]Carnegie Mellon University [‡]University of Virginia [§]ETH Zürich

Adoption: Evaluation Infrastructures

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
"Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
*Preprint on **arxiv**, August 2023.*
[\[arXiv version\]](#)
[\[Ramulator 2.0 Source Code\]](#)

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

<https://arxiv.org/pdf/2308.11030.pdf>

Processing-in-Memory: Challenges

To fully support PIM systems, we need to develop:

- 1 Workload characterization methodologies and benchmark suites targeting PIM architectures
- 2 Frameworks that can facilitate the implementation of complex operations and algorithms using PIM primitives
- 3 Compiler support and compiler optimizations targeting PIM architectures
- 4 Operating system support for PIM-aware virtual memory, memory management, data allocation and mapping
- 5 End-to-End System-on-Chip Design Beyond DRAM

The lack of tools and system support for PIM architectures limit the adoption of PIM systems

An Example: SimplePIM Framework

- Jinfan Chen, Juan Gómez-Luna, Izzat El Hajj, YuXin Guo, and Onur Mutlu,
"SimplePIM: A Software Framework for Productive and Efficient Processing in Memory"
Proceedings of the 32nd International Conference on Parallel Architectures and Compilation Techniques (PACT), Vienna, Austria, October 2023.

SimplePIM: A Software Framework for Productive and Efficient Processing-in-Memory

Jinfan Chen¹ Juan Gómez-Luna¹ Izzat El Hajj² Yuxin Guo¹ Onur Mutlu¹
¹ETH Zürich ²American University of Beirut

Executive Summary

- Real PIM hardware is now available, e.g., UPMEM PIM
- However, **programming real PIM hardware is challenging**, e.g., need to:
 - Distribute data across PIM memory banks,
 - Manage data transfers between host cores and PIM cores, between PIM cores, and between DRAM bank and PIM scratchpad
 - Launch PIM kernels on the PIM cores, etc.
 - Synchronize properly between threads
- **SimplePIM** is a **high-level programming framework for real PIM hardware**
 - Iterators such as `map`, `reduce`, and `zip`
 - Collective communication with `broadcast`, `scatter`, and `gather`
- Implementation on UPMEM and evaluation with six different workloads
 - Reduction, vector add, histogram, linear/logistic regression, K-means
 - **4.4x fewer lines of code compared to hand-optimized code**
 - Between **15% and 43% faster than hand-optimized code** for three workloads
- Source code: <https://github.com/CMU-SAFARI/SimplePIM>

Concluding Remarks

Fundamentally Energy-Efficient (Data-Centric) Computing Architectures

Fundamentally High-Performance **(Data-Centric)** Computing Architectures

Computing Architectures with Minimal Data Movement

Concluding Remarks

- It is time to design **principled system architectures** to solve the **memory problem**
- We must design systems to be **balanced, high-performance, and energy-efficient** → **memory-centric**
 - Enable computation capabilities in memory
- This can
 - Lead to **orders-of-magnitude** improvements
 - **Enable new applications & computing platforms**
 - **Enable better understanding of nature**
 - ...
- Future of **truly memory-centric computing** is bright
 - We need to do research & design across the computing stack

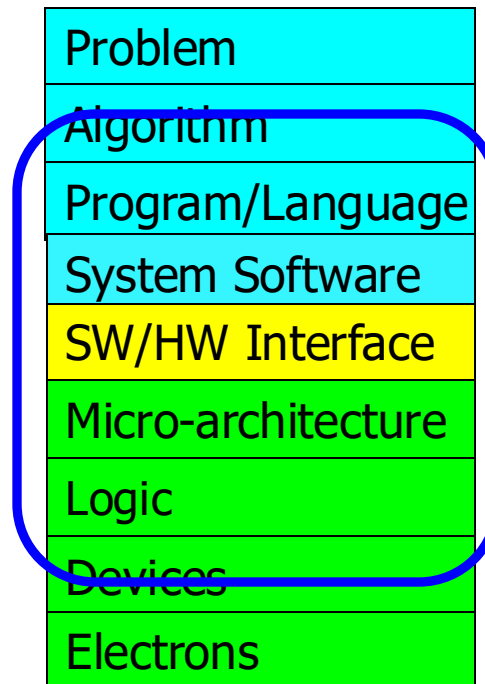
Data-centric

Data-driven

Data-aware

We Need to Revisit the Entire Stack

- With a **memory-centric mindset**



We can get there step by step

PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu^{a,b}, Saugata Ghose^{b,c}, Juan Gómez-Luna^a, Rachata Ausavarungnirun^d

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

^d*King Mongkut's University of Technology North Bangkok*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,

"A Modern Primer on Processing in Memory"

*Invited Book Chapter in **Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann**, Springer, to be published in 2021.*

Referenced Papers, Talks, Artifacts

- All are available at

<https://people.inf.ethz.ch/omutlu/projects.htm>

<https://www.youtube.com/onurmutlulectures>

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

Funding Acknowledgments

- Alibaba, AMD, ASML, Google, Facebook, Hi-Silicon, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware, Xilinx
- NSF
- NIH
- GSRC
- SRC
- CyLab
- EFCL
- SNSF

Thank you!

Acknowledgments



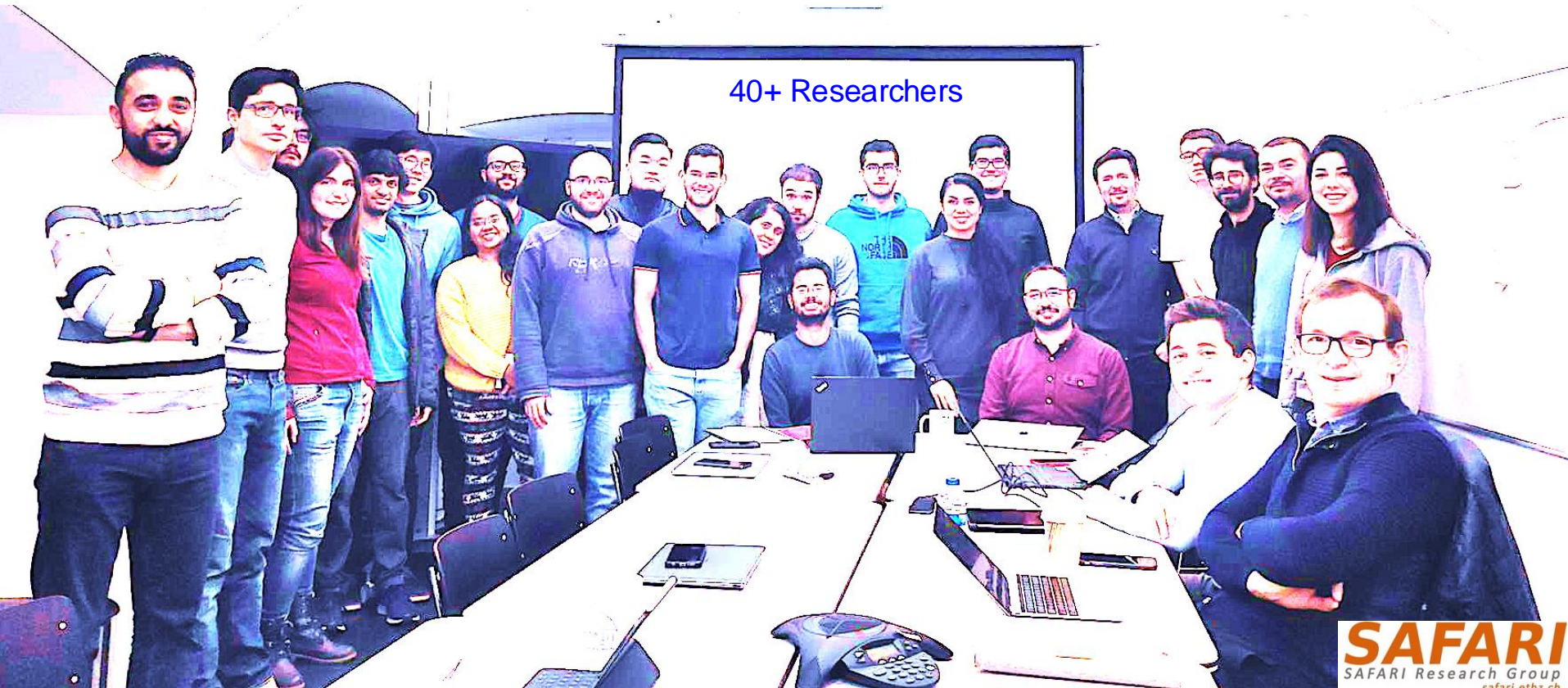
Think BIG, Aim HIGH!

<https://safari.ethz.ch>

SAFARI Research Group

Computer architecture, HW/SW, systems, bioinformatics, security, memory

<https://safari.ethz.ch/safari-newsletter-january-2021/>



Think BIG, Aim HIGH!

<https://safari.ethz.ch>

SAFARI Research Group: December 2021

- <https://safari.ethz.ch/safari-newsletter-december-2021/>

SAFARI
SAFARI Research Group

Think Big, Aim High

ETH zürich



View in your browser
December 2021



SAFARI Newsletter June 2023 Edition

- <https://safari.ethz.ch/safari-newsletter-june-2023/>

SAFARI
SAFARI Research Group

Think Big, Aim High

ETH zürich



[View in your browser](#)

June 2023



SAFARI Newsletter July 2024 Edition

- <https://safari.ethz.ch/safari-newsletter-july-2024/>



SAFARI Introduction & Research

Computer architecture, HW/SW, systems, bioinformatics, security, memory



Seminar in Computer Architecture - Lecture 5: Potpourri of Research Topics (Spring 2023)



Onur Mutlu Lectures
32.6K subscribers



719 views Streamed 1 month ago Livestream - Seminar in Computer Architecture - ETH Zürich (Spring 2023)

SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

<https://www.youtube.com/watch?v=mV2OuB2djEs>

Open Source Tools: SAFARI GitHub



SAFARI Research Group at ETH Zurich and Carnegie Mellon University

Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.

👤 440 followers 📍 ETH Zurich and Carnegie Mellon U... 🔗 <https://safari.ethz.ch/> ✉ omutlu@gmail.com

🏠 Overview 📁 Repositories 98 📁 Projects 📁 Packages 👤 People 13

📁 ramulator Public

A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...

● C++ ☆ 519 🍴 206

📁 prim-benchmarks Public

PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publ...

● C ☆ 125 🍴 45

📁 MQSim Public

MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

● C++ ☆ 265 🍴 144

📁 rowhammer Public

Source code for testing the Row Hammer error mechanism in DRAM devices. Described in the ISCA 2014 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf.

● C ☆ 210 🍴 43

📁 SoftMC Public

SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

● Verilog ☆ 120 🍴 27

📁 Pythia Public

A customizable hardware prefetching framework using online reinforcement learning as described in the MICRO 2021 paper by Bera et al. (<https://arxiv.org/pdf/2109.12021.pdf>).

● C++ ☆ 107 🍴 34

Memory-Centric Computing

Onur Mutlu

omutlu@gmail.com

<https://people.inf.ethz.ch/omutlu>

20 August 2024

Rambus

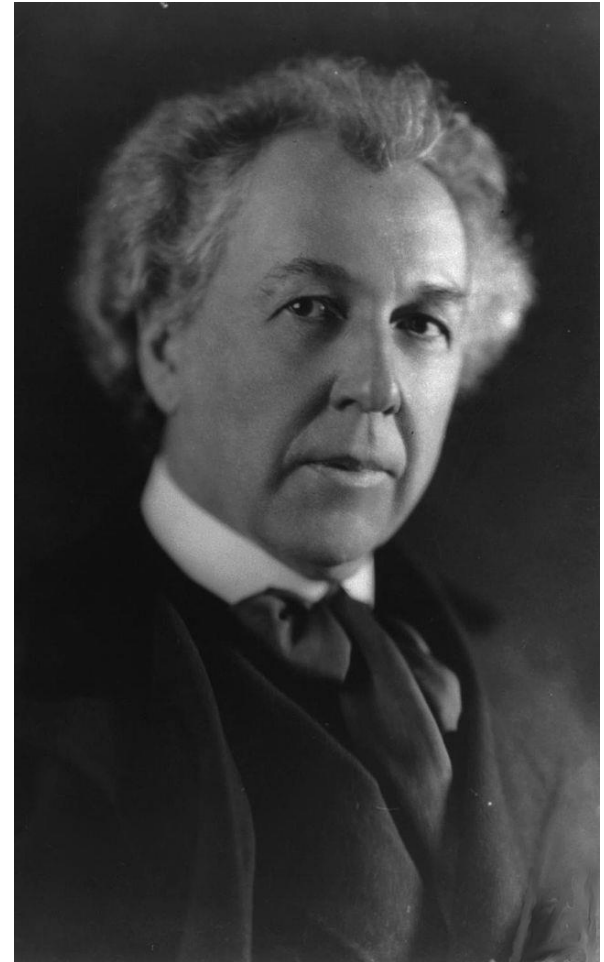
SAFARI

ETH zürich

Principled Design

A Quote from A Famous Architect

- “architecture [...] based upon **principle**, and not upon **precedent**”



Precedent-Based Design?

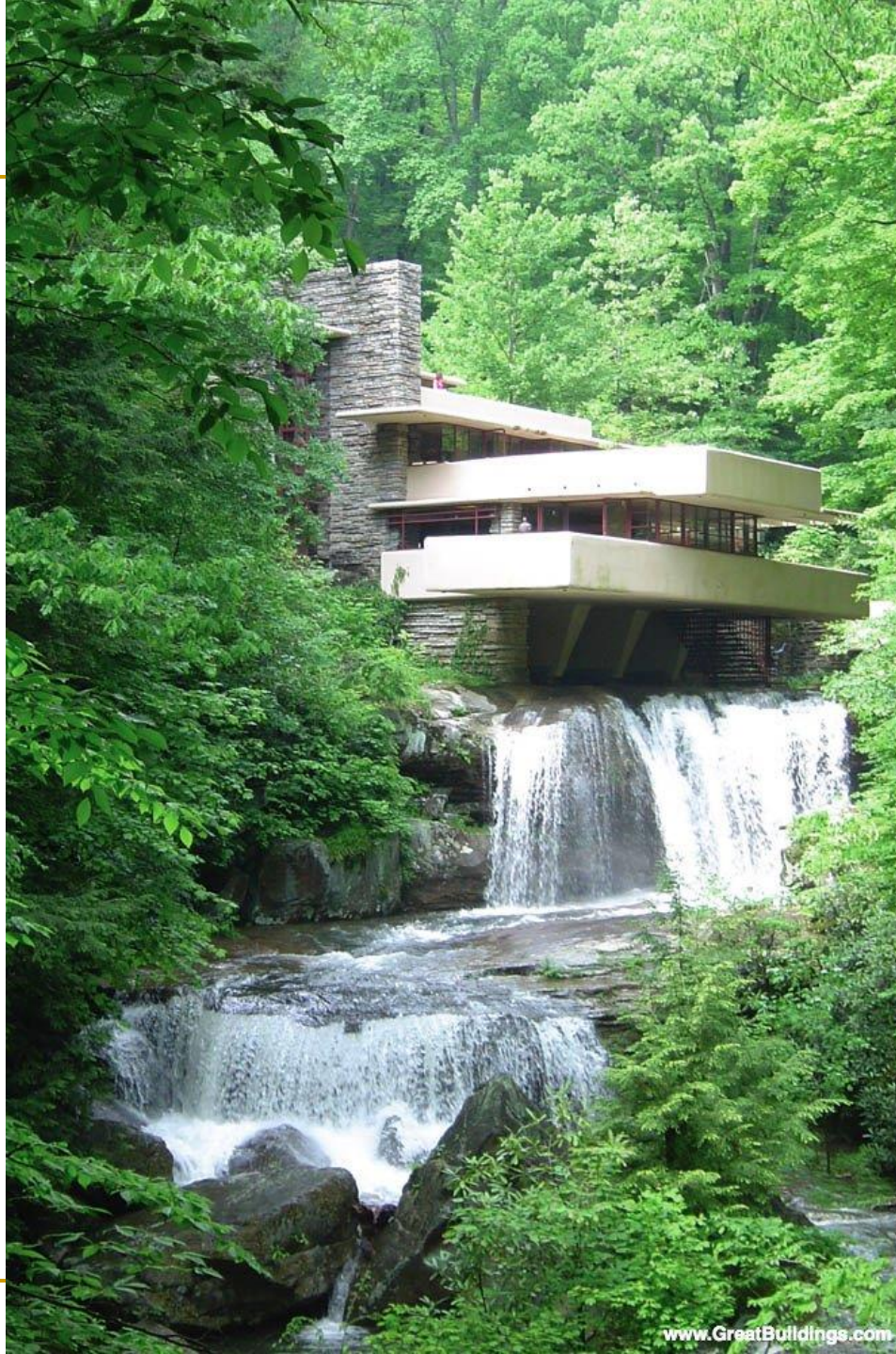
- “architecture [...] based upon **principle**, and not upon **precedent**”



Principled Design

- “architecture [...] based upon **principle**, and not upon **precedent**”





The Overarching Principle

Organic architecture

From Wikipedia, the free encyclopedia

Organic architecture is a [philosophy](#) of [architecture](#) which promotes harmony between human habitation and the natural world through design approaches so sympathetic and well integrated with its site, that buildings, furnishings, and surroundings become part of a unified, interrelated composition.

A well-known example of organic architecture is [Fallingwater](#), the residence Frank Lloyd Wright designed for the Kaufmann family in rural Pennsylvania. Wright had many choices to locate a home on this large site, but chose to place the home directly over the waterfall and creek creating a close, yet noisy dialog with the rushing water and the steep site. The horizontal striations of stone masonry with daring [cantilevers](#) of colored beige concrete blend with native rock outcroppings and the wooded environment.

Another Example: Precedent-Based Design



Principled Design



Another Principled Design



Source: By Martín Gómez Tagle - Lisbon, Portugal, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=13764903>

Source: <http://www.arcspace.com/exhibitions/unsorted/santiago-calatrava/>

Another Principled Design



Principle Applied to Another Structure



The Overarching Principle

Zoomorphic architecture

From Wikipedia, the free encyclopedia

Zoomorphic architecture is the practice of using animal forms as the inspirational basis and blueprint for architectural design. "While animal forms have always played a role adding some of the deepest layers of meaning in architecture, it is now becoming evident that a new strand of **biomorphism** is emerging where the meaning derives not from any specific representation but from a more general allusion to biological processes."^[1]

Some well-known examples of Zoomorphic architecture can be found in the **TWA Flight Center** building in **New York City**, by **Eero Saarinen**, or the **Milwaukee Art Museum** by **Santiago Calatrava**, both inspired by the form of a bird's wings.^[3]

Overarching Principles for Computing?



Concluding Remarks

- It is time to design **principled system architectures** to solve the **memory problem**
- Design complete systems to be balanced, high-performance, and energy-efficient, i.e., **data-centric (or memory-centric)**
- Enable computation capability inside and close to memory
- **This** can
 - ❑ Lead to **orders-of-magnitude** improvements
 - ❑ **Enable new applications & computing platforms**
 - ❑ **Enable better understanding of nature**
 - ❑ ...

Backup Slides

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand

Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun,
Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela,
Allan Knies, Parthasarathy Ranganathan, Onur Mutlu

SAFARI

Carnegie Mellon

Google



SEOUL
NATIONAL
UNIVERSITY

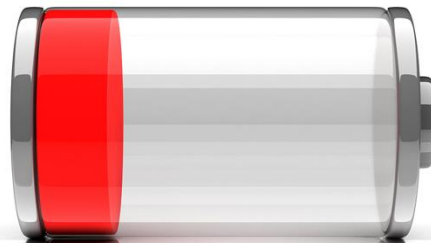
ETH zürich

Consumer Devices



Consumer devices are everywhere!

**Energy consumption is
a first-class concern in consumer devices**



Popular Consumer Workloads



Chrome

Google's web browser



TensorFlow Mobile

Google's machine learning
framework

VP9



Video Playback

Google's **video codec**

VP9

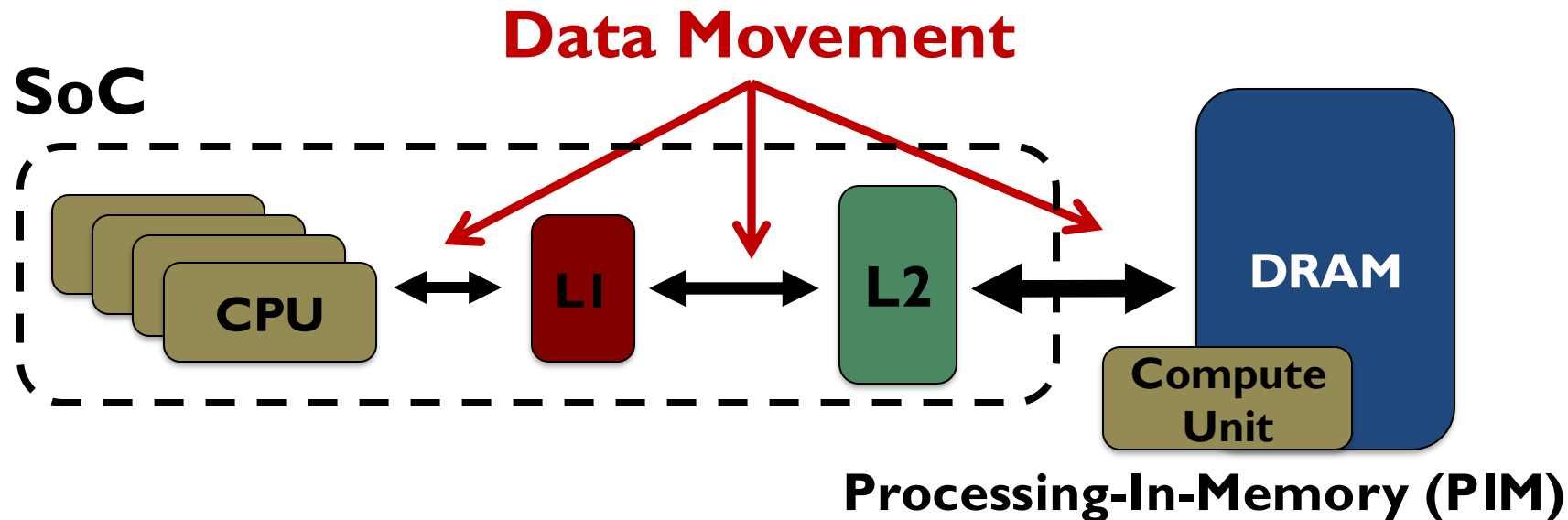


Video Capture

Google's **video codec**

Energy Cost of Data Movement

1st key observation: **62.7%** of the total system energy is spent on **data movement**



Potential solution: move computation **close to data**

Challenge: limited area and energy budget

Using PIM to Reduce Data Movement

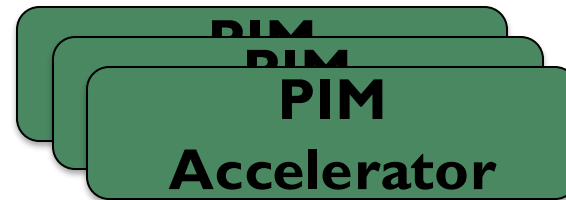
2nd key observation: a significant fraction of the **data movement** often comes from **simple functions**

We can design lightweight logic to implement these simple functions in **memory**

Small embedded
low-power core



Small fixed-function
accelerators



Offloading to PIM logic reduces energy and improves performance, on average, by 2.3X and 2.2X

Workload Analysis



Chrome

Google's web browser



TensorFlow Mobile

Google's machine learning
framework



Video Playback

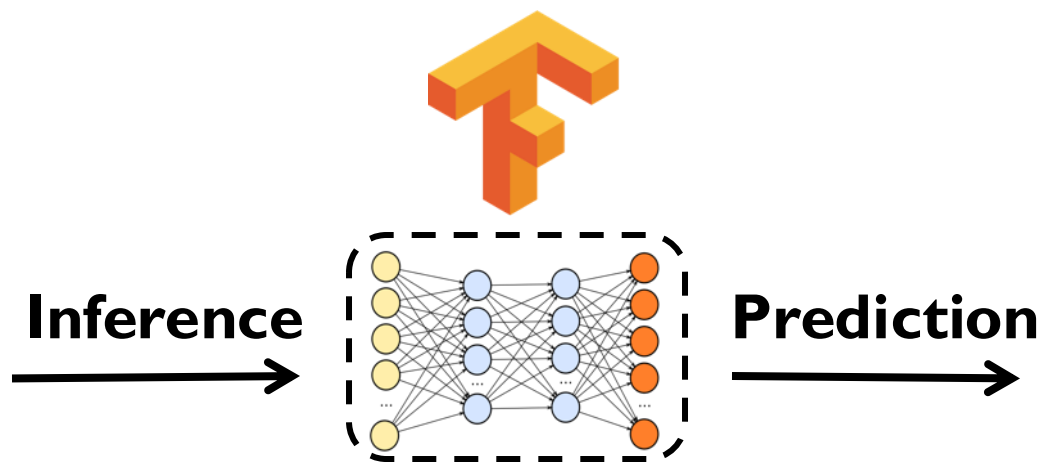
Google's **video codec**



Video Capture

Google's **video codec**

TensorFlow Mobile

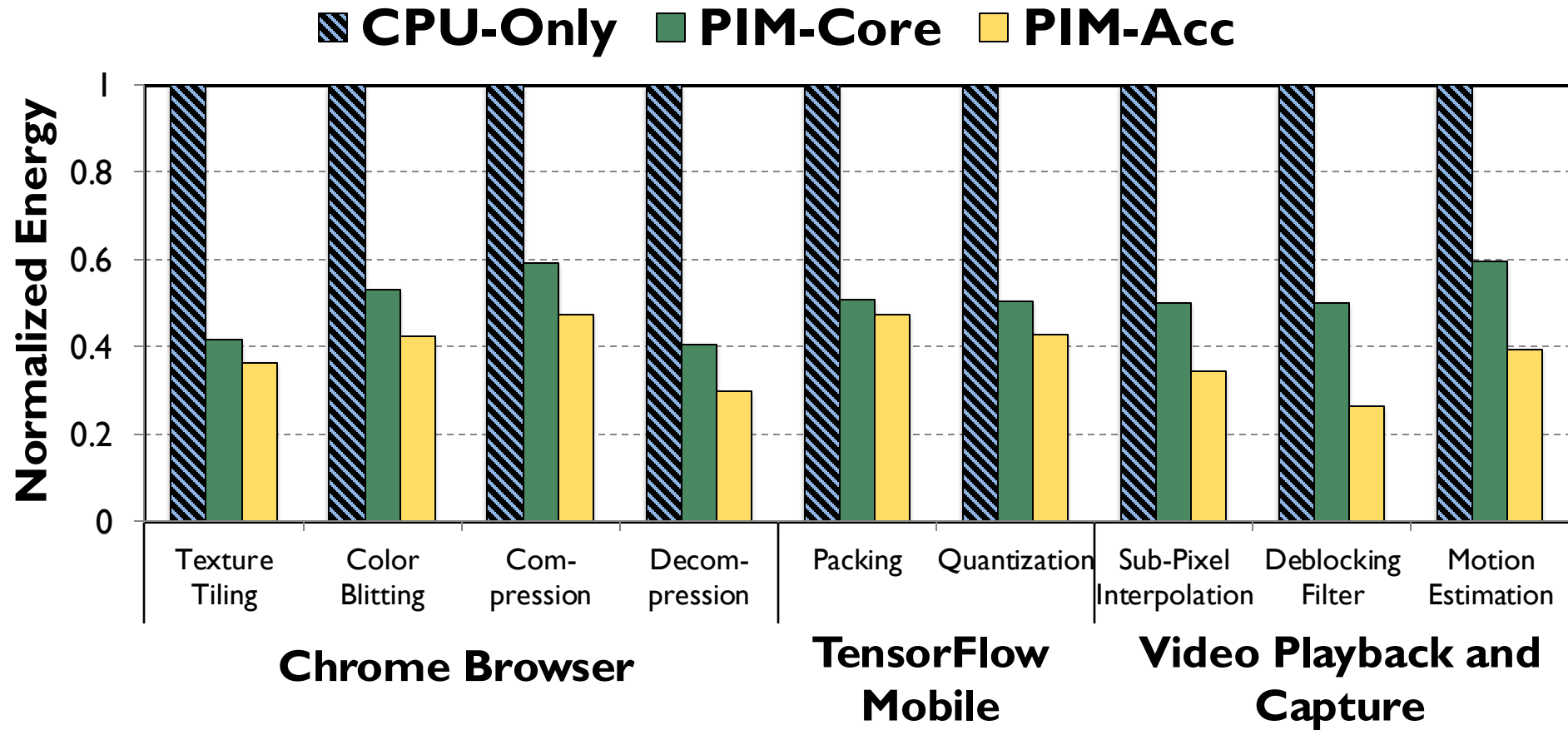


57.3% of the inference energy is spent on data movement



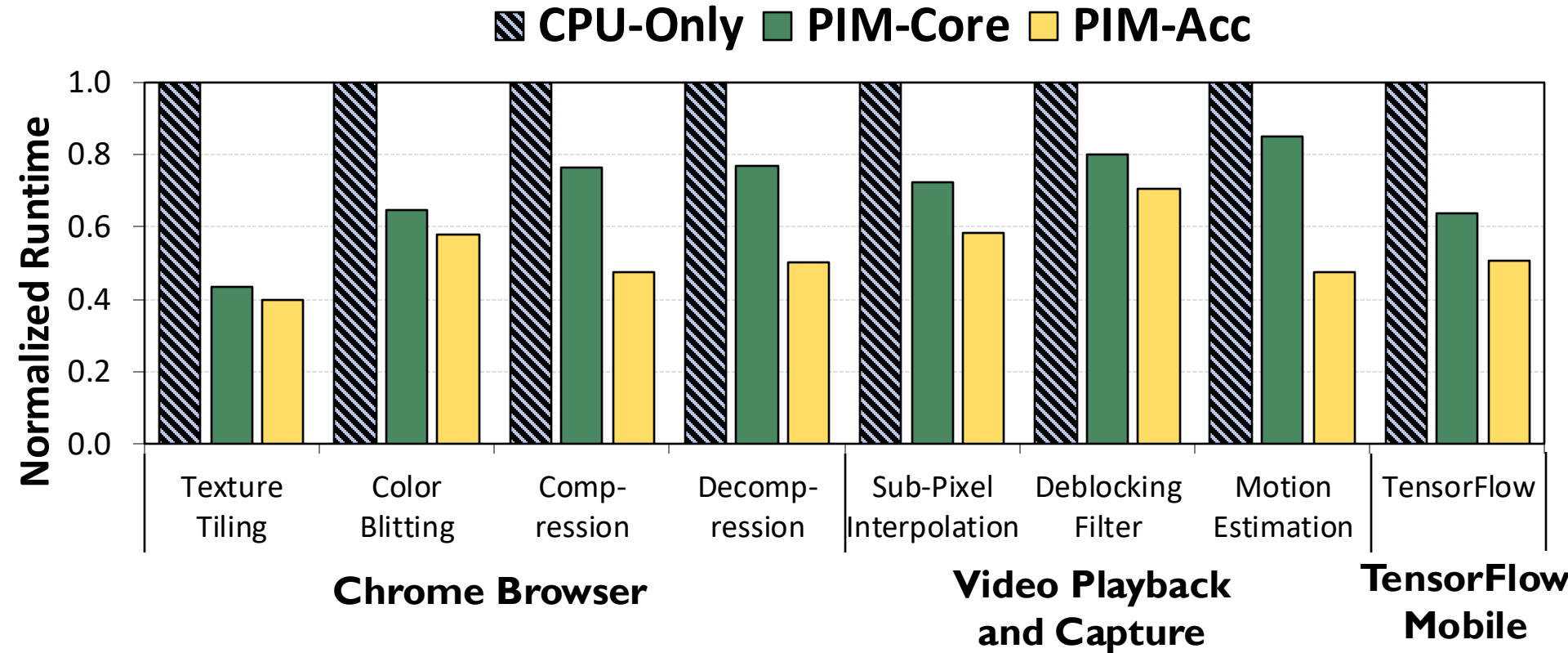
54.4% of the **data movement** energy comes from packing/unpacking and quantization

Normalized Energy



PIM core and PIM accelerator reduce
energy consumption on average by 49.1% and 55.4%

Normalized Runtime



Offloading these kernels to **PIM core** and **PIM accelerator** reduces **program runtime** on average by **44.6%** and **54.2%**

More on PIM for Mobile Devices

- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu,

"Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks"

Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Williamsburg, VA, USA, March 2018.

[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Session Slides \(pptx\) \(pdf\)](#)] [[Poster \(pptx\) \(pdf\)](#)]
[[Lightning Talk Video](#) (2 minutes)]
[[Full Talk Video](#) (21 minutes)]

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

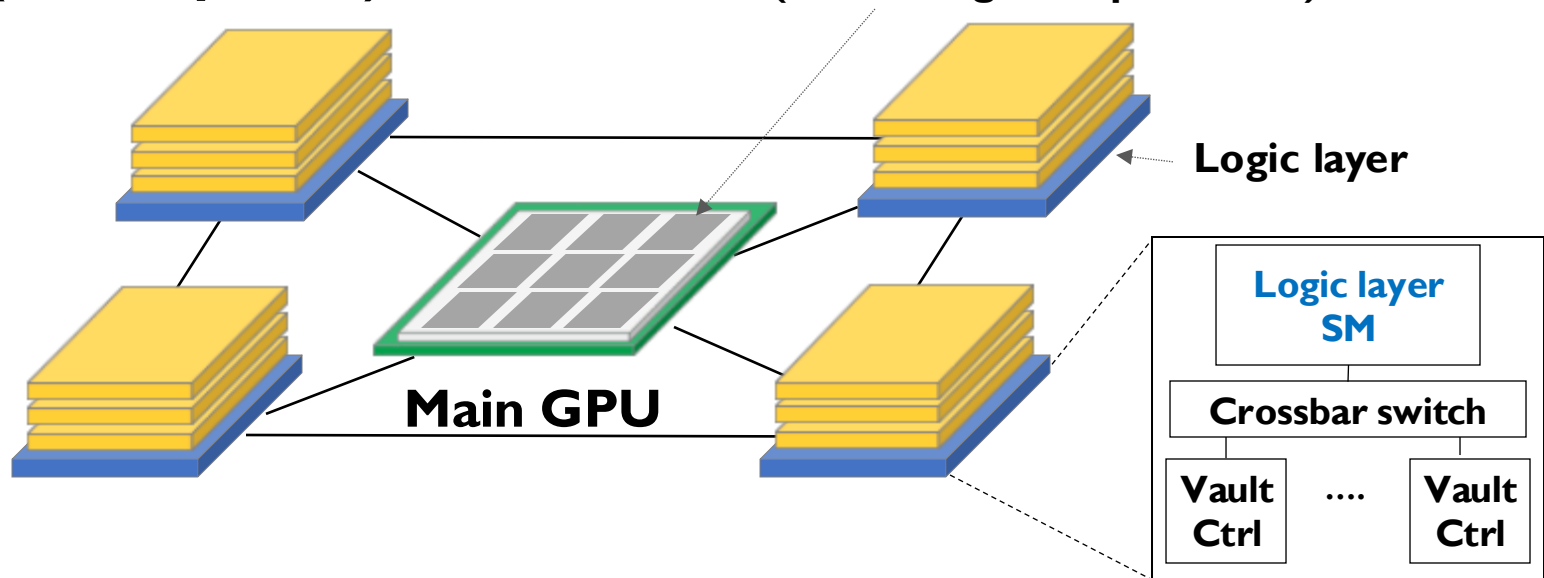
Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

Truly Distributed GPU Processing with PIM

**3D-stacked memory
(memory stack)**

SM (Streaming Multiprocessor)



```
__global__  
void applyScaleFactorsKernel( uint8_T * const out,  
                             uint8_T const * const in, const double *factor,  
                             size_t const numRows, size_t const numCols )  
{  
    // Work out which pixel we are working on.  
    const int rowIdx = blockIdx.x * blockDim.x + threadIdx.x;  
    const int colIdx = blockIdx.y;  
    const int sliceIdx = threadIdx.z;  
  
    // Check this thread isn't off the image  
    if( rowIdx >= numRows ) return;  
  
    // Compute the index of my element  
    size_t linearIdx = rowIdx + colIdx*numRows +  
                      sliceIdx*numRows*numCols;
```


Accelerating GPU Execution with PIM (I)

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler, **"Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems"**

Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Session Slides \(pptx\)](#) ([pdf](#))]

Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems

Kevin Hsieh[‡] Eiman Ebrahimi[†] Gwangsun Kim^{*} Niladrish Chatterjee[†] Mike O'Connor[†]
Nandita Vijaykumar[‡] Onur Mutlu^{§‡} Stephen W. Keckler[†]

[‡]Carnegie Mellon University [†]NVIDIA ^{*}KAIST [§]ETH Zürich

Accelerating GPU Execution with PIM (II)

- Ashutosh Pattnaik, Xulong Tang, Adwait Jog, Onur Kayiran, Asit K. Mishra, Mahmut T. Kandemir, Onur Mutlu, and Chita R. Das,
"Scheduling Techniques for GPU Architectures with Processing-In-Memory Capabilities"
Proceedings of the 25th International Conference on Parallel Architectures and Compilation Techniques (PACT), Haifa, Israel, September 2016.

Scheduling Techniques for GPU Architectures with Processing-In-Memory Capabilities

Ashutosh Pattnaik¹ Xulong Tang¹ Adwait Jog² Onur Kayiran³
Asit K. Mishra⁴ Mahmut T. Kandemir¹ Onur Mutlu^{5,6} Chita R. Das¹

¹Pennsylvania State University ²College of William and Mary
³Advanced Micro Devices, Inc. ⁴Intel Labs ⁵ETH Zürich ⁶Carnegie Mellon University

Accelerating Linked Data Structures

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,
"Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation"
Proceedings of the 34th IEEE International Conference on Computer Design (ICCD), Phoenix, AZ, USA, October 2016.

Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation

Kevin Hsieh[†] Samira Khan[‡] Nandita Vijaykumar[†]
Kevin K. Chang[†] Amirali Boroumand[†] Saugata Ghose[†] Onur Mutlu^{§†}
[†]*Carnegie Mellon University* [‡]*University of Virginia* [§]*ETH Zürich*

Accelerating Dependent Cache Misses

- Milad Hashemi, Khubaib, Eiman Ebrahimi, Onur Mutlu, and Yale N. Patt, **"Accelerating Dependent Cache Misses with an Enhanced Memory Controller"**

Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.

[Slides (pptx) (pdf)]

[Lightning Session Slides (pptx) (pdf)]

Accelerating Dependent Cache Misses with an Enhanced Memory Controller

Milad Hashemi*, Khubaib[†], Eiman Ebrahimi[‡], Onur Mutlu[§], Yale N. Patt*

**The University of Texas at Austin [†]Apple [‡]NVIDIA [§]ETH Zürich & Carnegie Mellon University*

Accelerating Runahead Execution

- Milad Hashemi, Onur Mutlu, and Yale N. Patt,
"Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads"
Proceedings of the 49th International Symposium on Microarchitecture (MICRO), Taipei, Taiwan, October 2016.
[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Session Slides \(pdf\)](#)] [[Poster \(pptx\) \(pdf\)](#)]
Best paper session.

Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads

Milad Hashemi*, Onur Mutlu[§], Yale N. Patt*

**The University of Texas at Austin* [§]*ETH Zürich*

Accelerating Climate Modeling

- Gagandeep Singh, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gómez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal, **"NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling"**

Proceedings of the 30th International Conference on Field-Programmable Logic and Applications (FPL), Gothenburg, Sweden, September 2020.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (23 minutes)]

Nominated for the Stamatis Vassiliadis Memorial Award.

NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling

Gagandeep Singh^{a,b,c} Dionysios Diamantopoulos^c Christoph Hagleitner^c Juan Gómez-Luna^b
Sander Stuijk^a Onur Mutlu^b Henk Corporaal^a

^aEindhoven University of Technology

^bETH Zürich

^cIBM Research Europe, Zurich

Accelerating Approximate String Matching

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"**
Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.
[[Lighting Talk Video](#) (1.5 minutes)]
[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (18 minutes)]
[[Slides \(pptx\)](#) ([pdf](#))]

GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis

Damla Senol Cali^{†⋈} Gurpreet S. Kalsi[⋈] Zülal Bingöl[▽] Can Firtina[◇] Lavanya Subramanian[‡] Jeremie S. Kim^{◇†}
Rachata Ausavarungnirun[○] Mohammed Alser[◇] Juan Gomez-Luna[◇] Amirali Boroumand[†] Anant Nori[⋈]
Allison Scibisz[†] Sreenivas Subramoney[⋈] Can Alkan[▽] Saugata Ghose^{*†} Onur Mutlu^{◇†▽}
[†]Carnegie Mellon University [⋈]Processor Architecture Research Lab, Intel Labs [▽]Bilkent University [◇]ETH Zürich
[‡]Facebook [○]King Mongkut's University of Technology North Bangkok ^{*}University of Illinois at Urbana-Champaign

Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zülal Bingöl, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika Mansouri Ghiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
"SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
Proceedings of the 49th International Symposium on Computer Architecture (ISCA), New York, June 2022.
[[arXiv version](#)]

SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Zülal Bingöl³
Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim²
Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr²
Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

¹Bionano Genomics ²ETH Zürich ³Bilkent University ⁴Intel Labs

⁵Carnegie Mellon University ⁶University of Illinois Urbana-Champaign

Accelerating Basecalling + Read Mapping

- Haiyu Mao, Mohammed Alser, Mohammad Sadrosadati, Can Firtina, Akanksha Baranwal, Damla Senol Cali, Aditya Manglik, Nour Almadhoun Alserr, and Onur Mutlu, **"GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping"**
Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Longer Lecture Slides \(pptx\)](#)] [[pdf](#)]
[[Lecture Video](#) (25 minutes)]
[[arXiv version](#)]

GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping

Haiyu Mao¹ Mohammed Alser¹ Mohammad Sadrosadati¹ Can Firtina¹ Akanksha Baranwal¹
Damla Senol Cali² Aditya Manglik¹ Nour Almadhoun Alserr¹ Onur Mutlu¹
¹*ETH Zürich* ²*Bionano Genomics*

Accelerating Time Series Analysis

- Ivan Fernandez, Ricardo Quisiant, Christina Giannoula, Mohammed Alser, Juan Gómez-Luna, Eladio Gutiérrez, Oscar Plata, and Onur Mutlu,
"NATSA: A Near-Data Processing Accelerator for Time Series Analysis"
Proceedings of the 38th IEEE International Conference on Computer Design (ICCD), Virtual, October 2020.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (10 minutes)]
[[Source Code](#)]

NATSA: A Near-Data Processing Accelerator for Time Series Analysis

Ivan Fernandez [§]	Ricardo Quisiant [§]	Christina Giannoula [†]	Mohammed Alser [‡]
Juan Gómez-Luna [‡]	Eladio Gutiérrez [§]	Oscar Plata [§]	Onur Mutlu [‡]
[§] <i>University of Malaga</i>	[†] <i>National Technical University of Athens</i>	[‡] <i>ETH Zürich</i>	

Accelerating Graph Pattern Mining

- Maciej Besta, Raghavendra Kanakagiri, Grzegorz Kwasniewski, Rachata Ausavarungnirun, Jakub Beránek, Konstantinos Kanellopoulos, Kacper Janda, Zur Vonarburg-Shmaria, Lukas Gianinazzi, Ioana Stefan, Juan Gómez-Luna, Marcin Copik, Lukas Kapp-Schwoerer, Salvatore Di Girolamo, Nils Blach, Marek Konieczny, Onur Mutlu, and Torsten Hoefler,

"SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems"

Proceedings of the 54th International Symposium on Microarchitecture (MICRO), Virtual, October 2021.

[[Slides \(pdf\)](#)]

[[Talk Video](#) (22 minutes)]

[[Lightning Talk Video](#) (1.5 minutes)]

[[Full arXiv version](#)]

SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems

Maciej Besta¹, Raghavendra Kanakagiri², Grzegorz Kwasniewski¹, Rachata Ausavarungnirun³, Jakub Beránek⁴, Konstantinos Kanellopoulos¹, Kacper Janda⁵, Zur Vonarburg-Shmaria¹, Lukas Gianinazzi¹, Ioana Stefan¹, Juan Gómez-Luna¹, Marcin Copik¹, Lukas Kapp-Schwoerer¹, Salvatore Di Girolamo¹, Nils Blach¹, Marek Konieczny⁵, Onur Mutlu¹, Torsten Hoefler¹

¹ETH Zurich, Switzerland
Thailand

²IIT Tirupati, India

³King Mongkut's University of Technology North Bangkok,
⁴Technical University of Ostrava, Czech Republic

⁵AGH-UST, Poland

Accelerating HTAP Database Systems

- Amirali Boroumand, Saugata Ghose, Geraldo F. Oliveira, and Onur Mutlu,
"Polynesia: Enabling High-Performance and Energy-Efficient Hybrid Transactional/Analytical Databases with Hardware/Software Co-Design"
Proceedings of the 38th International Conference on Data Engineering (ICDE),
Virtual, May 2022.
[[arXiv version](#)]
[[Slides \(pptx\)](#) ([pdf](#))]
[[Short Talk Slides \(pptx\)](#) ([pdf](#))]

Polynesia: Enabling High-Performance and Energy-Efficient Hybrid Transactional/Analytical Databases with Hardware/Software Co-Design

Amirali Boroumand[†] Saugata Ghose[◇] Geraldo F. Oliveira[‡] Onur Mutlu[‡]
[†]*Google* [◇]*Univ. of Illinois Urbana-Champaign* [‡]*ETH Zürich*

Accelerating Neural Network Inference

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,
"Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"
Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (14 minutes)]

Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand^{†◇}

Geraldo F. Oliveira^{*}

Saugata Ghose[‡]

Xiaoyu Ma[§]

Berkin Akin[§]

Eric Shiu[§]

Ravi Narayanaswami[§]

Onur Mutlu^{*†}

[†]*Carnegie Mellon Univ.*

[◇]*Stanford Univ.*

[‡]*Univ. of Illinois Urbana-Champaign*

[§]*Google*

^{*}*ETH Zürich*

Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand

Saugata Ghose

Berkin Akin

Ravi Narayanaswami

Geraldo F. Oliveira

Xiaoyu Ma

Eric Shiu

Onur Mutlu

PACT 2021

SAFARI

Carnegie Mellon



UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN



ETH zürich

Executive Summary

Context: We extensively analyze a state-of-the-art edge ML accelerator (Google Edge TPU) using 24 Google edge models

- Wide range of models (CNNs, LSTMs, Transducers, RCNNs)

Problem: The Edge TPU accelerator suffers from **three challenges:**

- It operates **significantly below** its peak throughput
- It operates **significantly below** its theoretical energy efficiency
- It **inefficiently** handles memory accesses

Key Insight: These shortcomings arise from **the monolithic design** of the Edge TPU accelerator

- The Edge TPU accelerator design does not account for **layer heterogeneity**

Key Mechanism: A new framework called **Mensa**

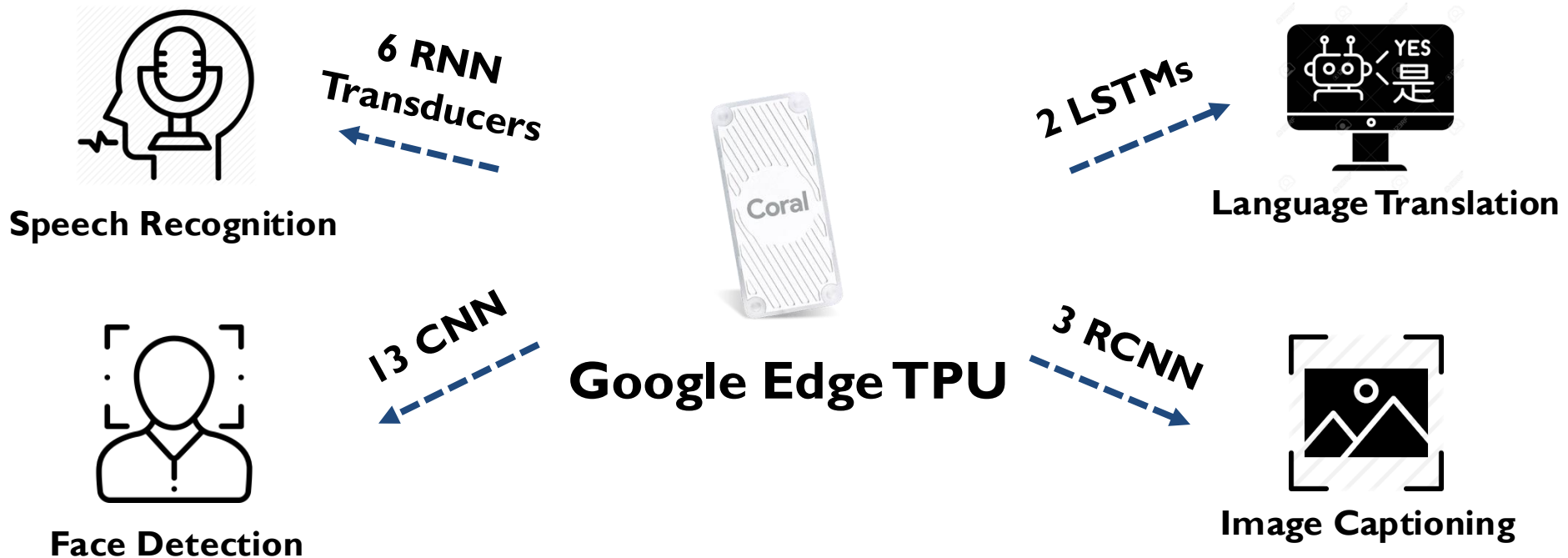
- Mensa consists of heterogeneous accelerators whose dataflow and hardware are specialized for specific families of layers

Key Results: We design a version of Mensa for Google edge ML models

- Mensa improves performance and energy by **3.0X** and **3.1X**
- Mensa reduces cost and improves area efficiency

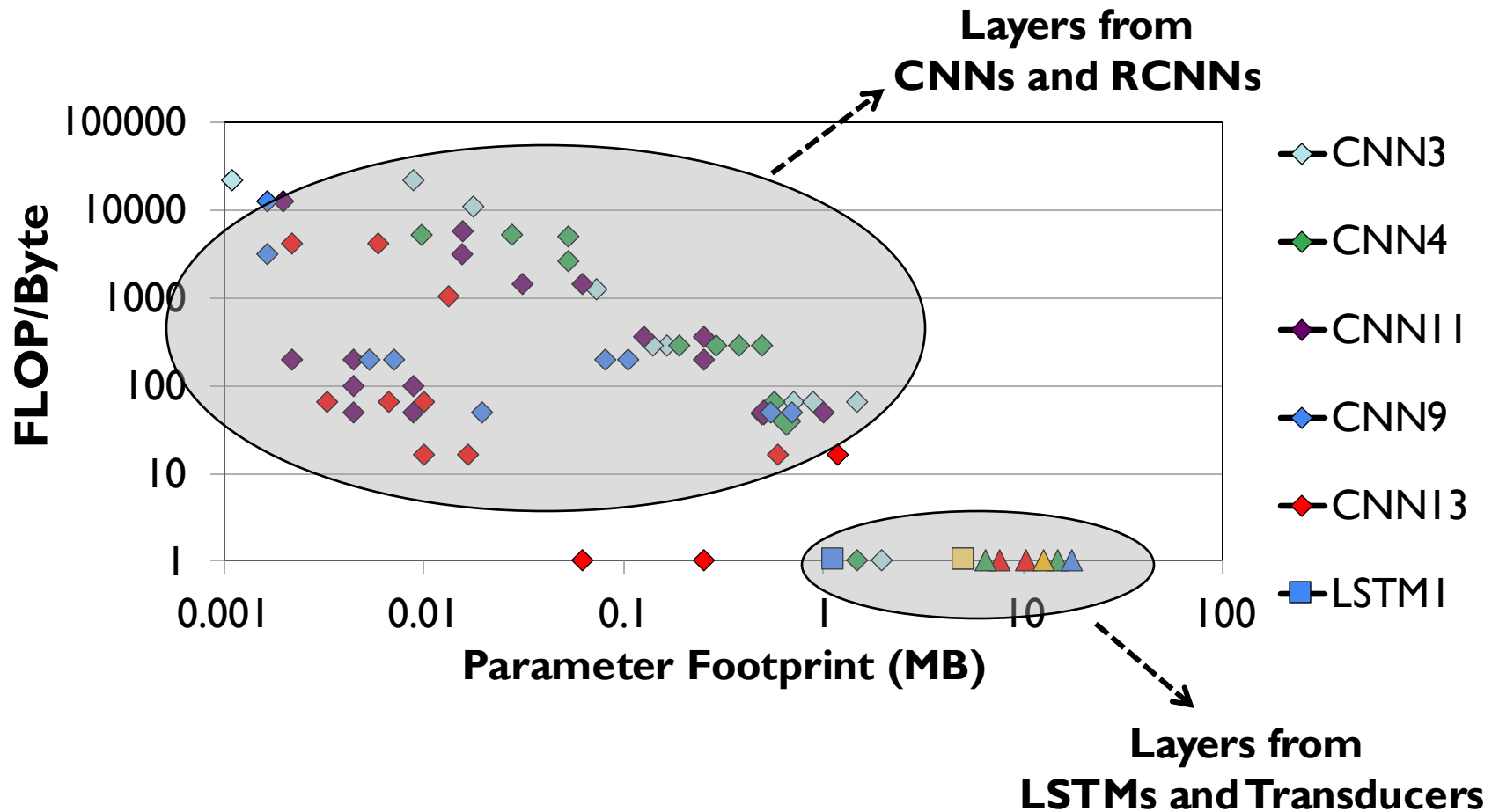
Google Edge Neural Network Models

We analyze inference execution using 24 edge NN models



Diversity Across the Models

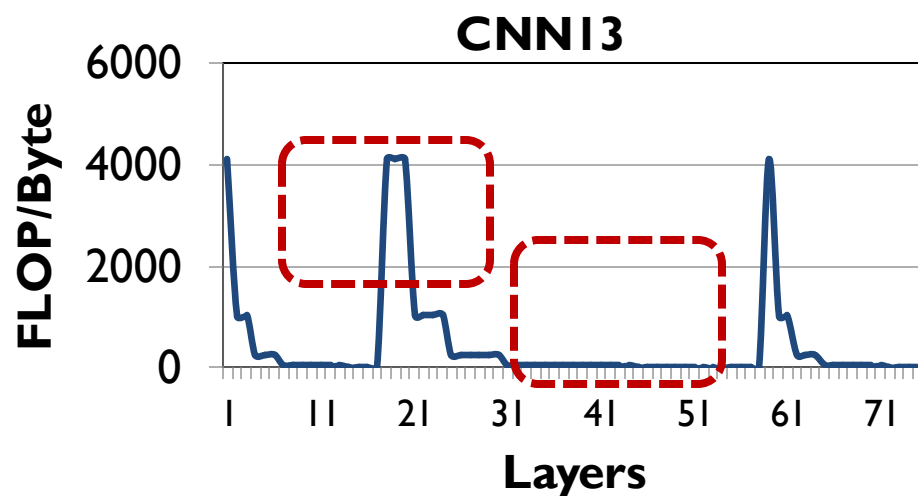
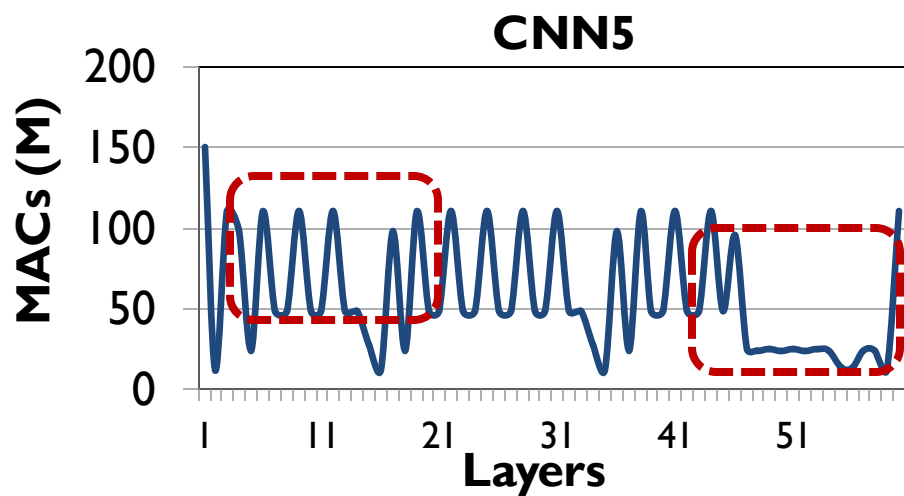
Insight 1: there is **significant variation** in terms of layer characteristics **across the models**



Diversity Within the Models

Insight 2: even **within** each model, layers exhibit **significant variation** in terms of layer characteristics

For example, our analysis of edge **CNN** models shows:



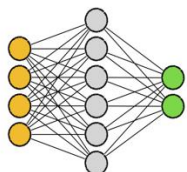
Variation in **MAC intensity**: up to **200x** across layers

Variation in **FLOP/Byte**: up to **244x** across layers

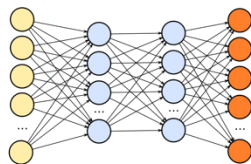
Mensa High-Level Overview

Edge TPU Accelerator

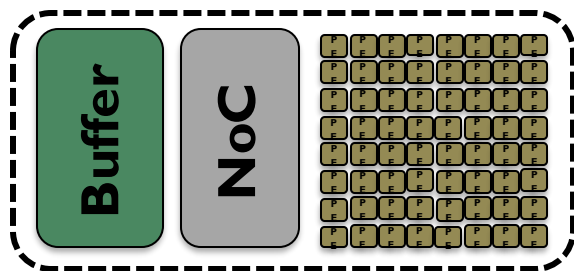
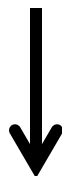
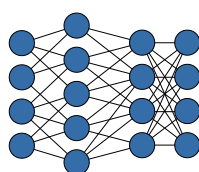
Model A



Model B



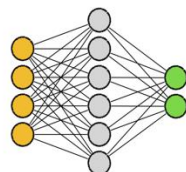
Model C



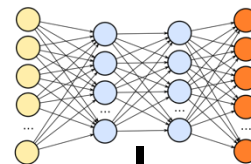
Monolithic Accelerator

Mensa

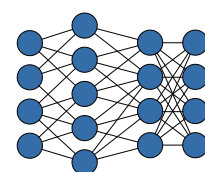
Model A



Model B

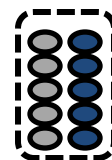


Model C

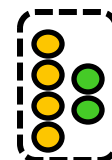


Runtime

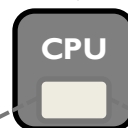
Family 1



Family 2



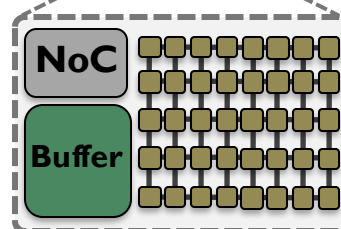
Family 3



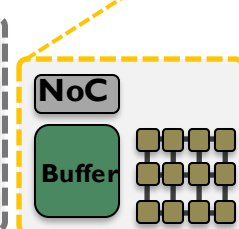
CPU



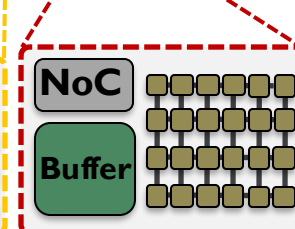
3D-Stacked DRAM



Acc. 1



Acc. 2

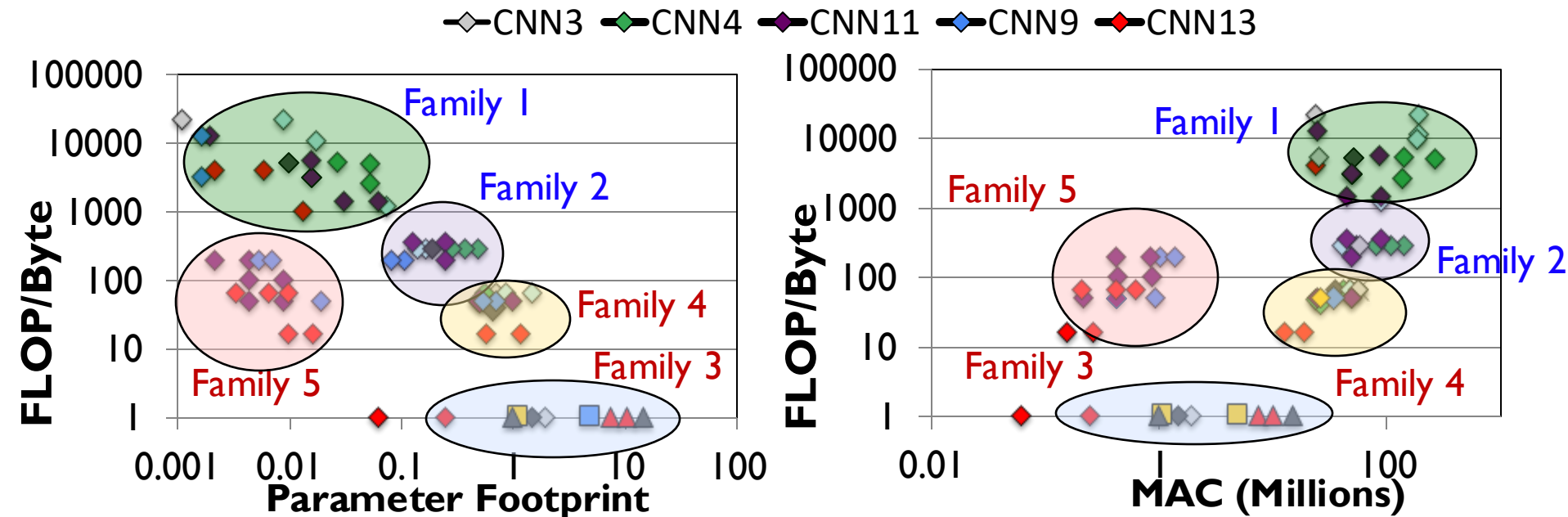


Acc. 3

Heterogeneous Accelerators

Identifying Layer Families

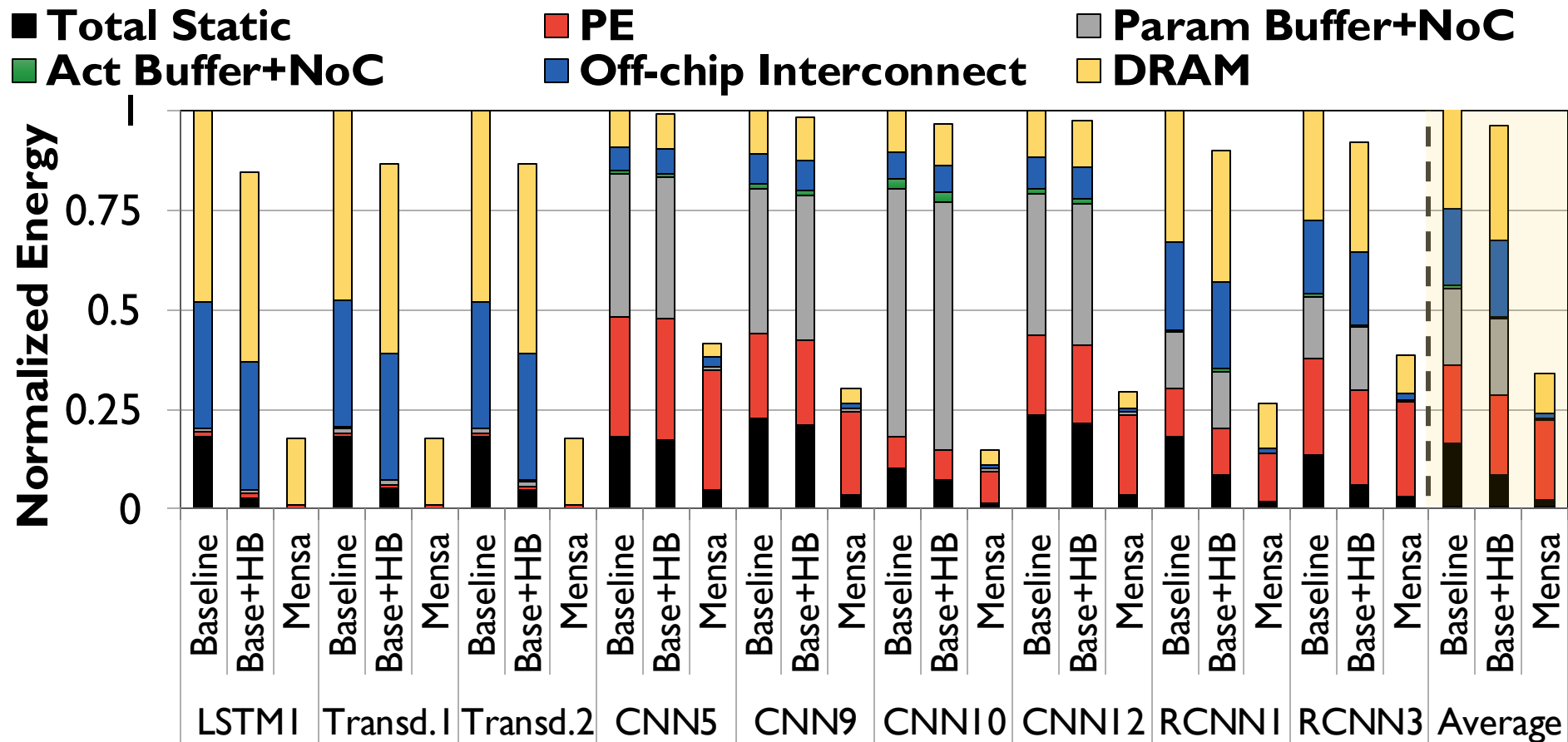
Key observation: the majority of layers group into a small number of layer families



Families 1 & 2: low parameter footprint, high data reuse and **MAC intensity**
→ compute-centric layers

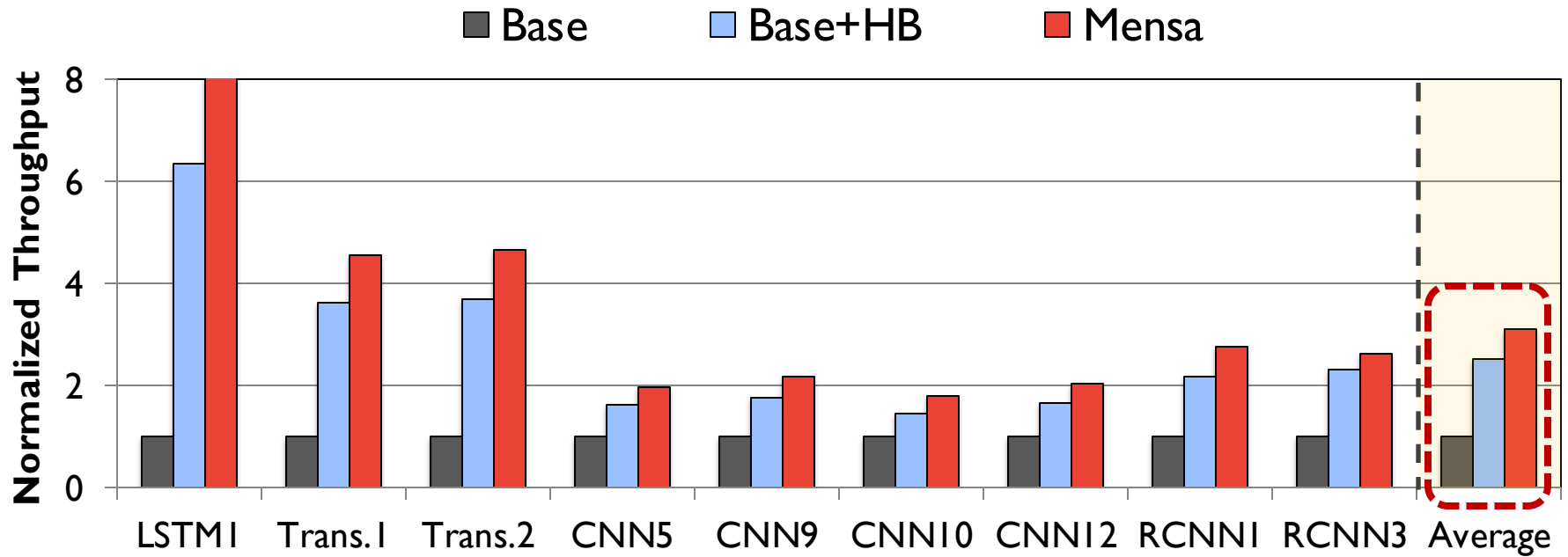
Families 3, 4 & 5: high parameter footprint, low data reuse and **MAC intensity**
→ data-centric layers

Mensa: Energy Reduction



Mensa-G reduces energy consumption by 3.0X
compared to the baseline Edge TPU

Mensa: Throughput Improvement



Mensa-G improves inference throughput by 3.1X
compared to the baseline Edge TPU

Mensa: Highly-Efficient ML Inference

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,
"Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"
Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (14 minutes)]

Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand^{†◇}

Geraldo F. Oliveira^{*}

Saugata Ghose[‡]

Xiaoyu Ma[§]

Berkin Akin[§]

Eric Shiu[§]

Ravi Narayanaswami[§]

Onur Mutlu^{*†}

[†]*Carnegie Mellon Univ.*

[◇]*Stanford Univ.*

[‡]*Univ. of Illinois Urbana-Champaign*

[§]*Google*

^{*}*ETH Zürich*

Accelerating Data-Intensive Workloads

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoun Choi, **"PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"** *Proceedings of the 42nd International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2015. [[Slides \(pdf\)](#)] [[Lightning Session Slides \(pdf\)](#)]

PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture

Junwhan Ahn Sungjoo Yoo Onur Mutlu[†] Kiyoun Choi

junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

[†]Carnegie Mellon University

FPGA-based Processing Near Memory

- Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gómez-Luna, Henk Corporaal, and Onur Mutlu, ["FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications"](#) *IEEE Micro (IEEE MICRO)*, 2021.

FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications

Gagandeep Singh[◇] Mohammed Alser[◇] Damla Senol Cali[✕]

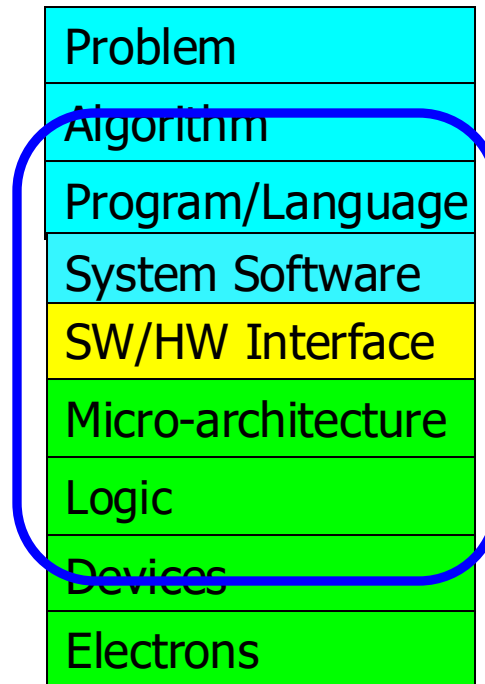
Dionysios Diamantopoulos[▽] Juan Gómez-Luna[◇]

Henk Corporaal^{*} Onur Mutlu^{◇✕}

[◇]*ETH Zürich* [✕]*Carnegie Mellon University*

^{*}*Eindhoven University of Technology* [▽]*IBM Research Europe*

We Need to Revisit the Entire Stack



We can get there step by step

PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu^{a,b}, Saugata Ghose^{b,c}, Juan Gómez-Luna^a, Rachata Ausavarungnirun^d

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

^d*King Mongkut's University of Technology North Bangkok*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,

"A Modern Primer on Processing in Memory"

*Invited Book Chapter in **Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann**, Springer, to be published in 2021.*

PIM Review and Open Problems (II)

A Workload and Programming Ease Driven Perspective of Processing-in-Memory

Saugata Ghose[†] Amirali Boroumand[†] Jeremie S. Kim^{†§} Juan Gómez-Luna[§] Onur Mutlu^{§†}

[†]*Carnegie Mellon University*

[§]*ETH Zürich*

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu,

"Processing-in-Memory: A Workload-Driven Perspective"

Invited Article in IBM Journal of Research & Development, Special Issue on Hardware for Artificial Intelligence, to appear in November 2019.

[Preliminary arXiv version]

Processing in Memory: Adoption Challenges

1. Processing **using** Memory
2. Processing **near** Memory

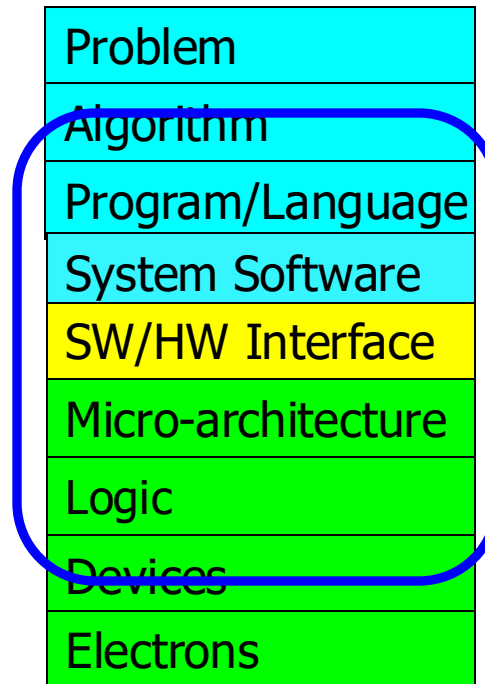
How to Enable Adoption of Processing in Memory

Potential Barriers to Adoption of PIM

1. **Applications & software** for PIM
2. Ease of **programming** (interfaces and compiler/HW support)
3. **System** and **security** support: coherence, synchronization, virtual memory, isolation, communication interfaces, ...
4. **Runtime** and **compilation** systems for adaptive scheduling, data mapping, access/sharing control, ...
5. **Infrastructures** to assess benefits and feasibility

All can be solved with change of mindset

We Need to Revisit the Entire Stack



We can get there step by step

Adoption: How to Keep It Simple?

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoun Choi, **"PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"** *Proceedings of the 42nd International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2015.
[[Slides \(pdf\)](#)] [[Lightning Session Slides \(pdf\)](#)]

PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture

Junwhan Ahn Sungjoo Yoo Onur Mutlu[†] Kiyoun Choi

junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

[†]Carnegie Mellon University

Adoption: How to Ease Programmability? (I)

- Geraldo F. Oliveira, Alain Kohli, David Novo, Juan Gómez-Luna, Onur Mutlu,
“DaPPA: A Data-Parallel Framework for Processing-in-Memory Architectures,”
in *PACT SRC Student Competition*, Vienna, Austria, October 2023.

DaPPA: A Data-Parallel Framework for Processing-in-Memory Architectures

Geraldo F. Oliveira*

Alain Kohli*

David Novo[‡]

Juan Gómez-Luna*

Onur Mutlu*

**ETH Zürich*

[‡]*LIRMM, Univ. Montpellier, CNRS*

Adoption: How to Ease Programmability? (II)

- Jinfan Chen, Juan Gómez-Luna, Izzat El Hajj, YuXin Guo, and Onur Mutlu,
"SimplePIM: A Software Framework for Productive and Efficient Processing in Memory"
Proceedings of the 32nd International Conference on Parallel Architectures and Compilation Techniques (PACT), Vienna, Austria, October 2023.

SimplePIM: A Software Framework for Productive and Efficient Processing-in-Memory

Jinfan Chen¹ Juan Gómez-Luna¹ Izzat El Hajj² Yuxin Guo¹ Onur Mutlu¹
¹ETH Zürich ²American University of Beirut

Adoption: How to Ease Programmability? (III)

- Geraldo F. Oliveira, Juan Gomez-Luna, Lois Orosa, Saugata Ghose, Nandita Vijaykumar, Ivan fernandez, Mohammad Sadrosadati, and Onur Mutlu,
"DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks"
IEEE Access, 8 September 2021.
Preprint in arXiv, 8 May 2021.
[[arXiv preprint](#)]
[[IEEE Access version](#)]
[[DAMOV Suite and Simulator Source Code](#)]
[[SAFARI Live Seminar Video](#) (2 hrs 40 mins)]
[[Short Talk Video](#) (21 minutes)]

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

MOHAMMAD SADROSADATI, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Adoption: How to Maintain Coherence? (I)

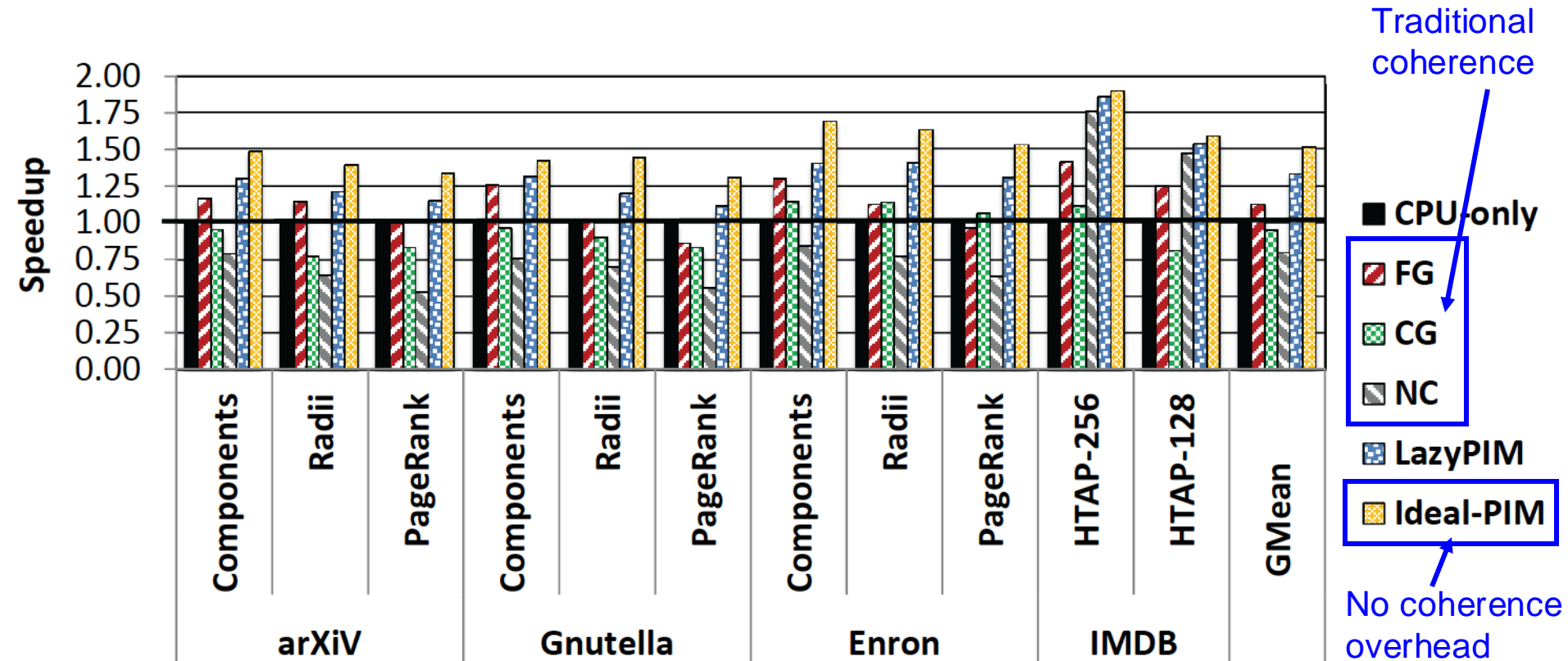
- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,
"**LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory**"
IEEE Computer Architecture Letters (**CAL**), June 2016.

LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory

Amirali Boroumand[†], Saugata Ghose[†], Minesh Patel[†], Hasan Hassan^{†§}, Brandon Lucia[†],
Kevin Hsieh[†], Krishna T. Malladi^{*}, Hongzhong Zheng^{*}, and Onur Mutlu^{††}

[†] *Carnegie Mellon University* ^{*} *Samsung Semiconductor, Inc.* [§] *TOBB ETÜ* [‡] *ETH Zürich*

Challenge: Coherence for Hybrid CPU-PIM Apps



Adoption: How to Maintain Coherence? (II)

- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,
"CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators"
Proceedings of the 46th International Symposium on Computer Architecture (ISCA), Phoenix, AZ, USA, June 2019.

CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators

Amirali Boroumand[†]

Saugata Ghose[†]

Minesh Patel[★]

Hasan Hassan[★]

Brandon Lucia[†]

Rachata Ausavarungnirun^{†‡}

Kevin Hsieh[†]

Nastaran Hajinazar^{◊†}

Krishna T. Malladi[§]

Hongzhong Zheng[§]

Onur Mutlu^{★†}

[†]Carnegie Mellon University

[★]ETH Zürich

[‡]KMUTNB

[◊]Simon Fraser University

[§]Samsung Semiconductor, Inc.

Adoption: How to Support Synchronization?

- Christina Giannoula, Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas, Ivan Fernandez, Juan Gómez-Luna, Lois Orosa, Nectarios Koziris, Georgios Goumas, Onur Mutlu, **["SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures"](#)**
Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Short Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (21 minutes)]
[[Short Talk Video](#) (7 minutes)]

SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures

Christina Giannoula^{†‡} Nandita Vijaykumar^{*‡} Nikela Papadopoulou[†] Vasileios Karakostas[†] Ivan Fernandez^{§‡}
Juan Gómez-Luna[‡] Lois Orosa[‡] Nectarios Koziris[†] Georgios Goumas[†] Onur Mutlu[‡]
[†]*National Technical University of Athens* [‡]*ETH Zürich* ^{*}*University of Toronto* [§]*University of Malaga*

Adoption: How to Support Virtual Memory?

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,
["Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation"](#)
Proceedings of the 34th IEEE International Conference on Computer Design (ICCD), Phoenix, AZ, USA, October 2016.

Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation

Kevin Hsieh[†] Samira Khan[‡] Nandita Vijaykumar[†]
Kevin K. Chang[†] Amirali Boroumand[†] Saugata Ghose[†] Onur Mutlu^{§†}
[†]Carnegie Mellon University [‡]University of Virginia [§]ETH Zürich

Adoption: Evaluation Infrastructures

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
"Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
*Preprint on **arxiv**, August 2023.*
[\[arXiv version\]](#)
[\[Ramulator 2.0 Source Code\]](#)

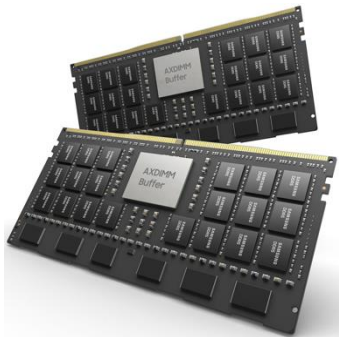
Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

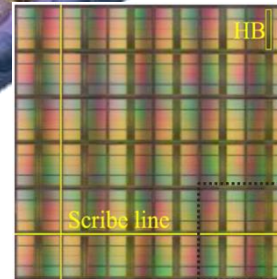
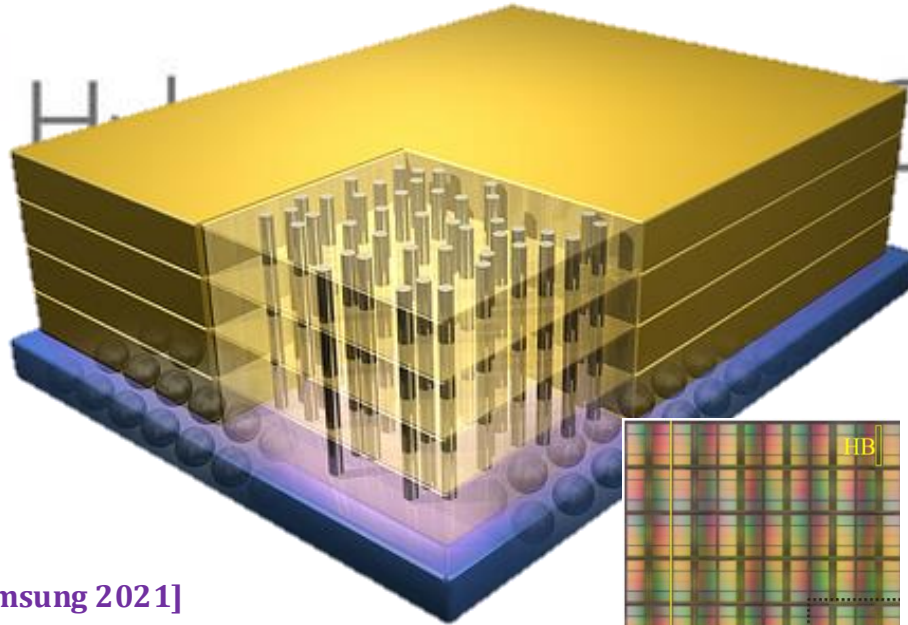
<https://arxiv.org/pdf/2308.11030.pdf>

Processing-in-Memory in the Real World

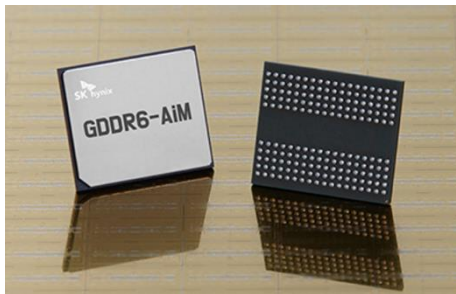
Processing-in-Memory Landscape Today



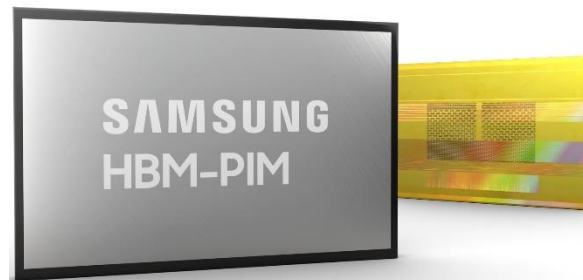
[Samsung 2021]



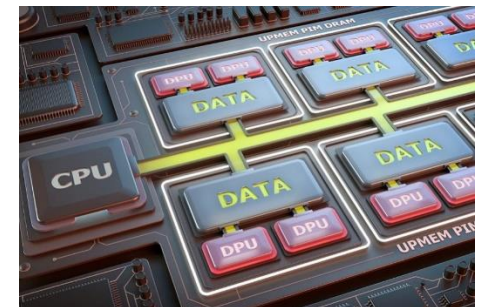
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]



[UPMEM 2019]

Processing-in-Memory Landscape Today

IEEE COMPUTER ARCHITECTURE LETTERS, VOL. 22, NO. 1, JANUARY-JUNE

Computational CXL-Memory Solution for Accelerating Memory-Intensive Applications

Joonseop Sim^{ID}, Soohong Ahn^{ID}, Taeyoung Ahn^{ID},
Seungyong Lee^{ID}, Myunghyun Rhee, Jooyoung Kim^{ID},
Kwangsik Shin, Donguk Moon^{ID},
Euseok Kim, and Kyoung Park^{ID}

Abstract—CXL interface is the up-to-date technology that enables effective memory expansion by providing a memory-sharing protocol in configuring heterogeneous devices. However, its limited physical bandwidth can be a significant bottleneck for emerging data-intensive applications. In this work, we propose a novel CXL-based memory disaggregation architecture with a real-world prototype demonstration, which overcomes the bandwidth limitation of the CXL interface using near-data processing. The experimental results demonstrate that our design achieves up to $1.9\times$ better performance/power efficiency than the existing CPU system.

Index Terms—Compute express link (CXL), near-data-processing (NDP)

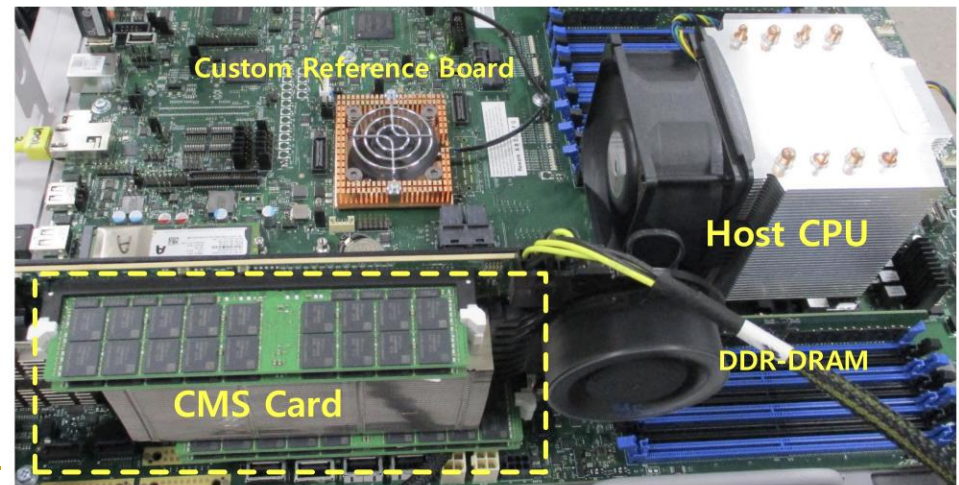


Fig. 6. FPGA prototype of proposed CMS card.

Processing-in-Memory Landscape Today

Samsung Processing in Memory Technology at Hot Chips 2023


By Patrick Kennedy - August 28, 2023



Samsung PIM PNM For Transformer Based AI HC35_Page_24

Real PIM Tutorials [MICRO'23, ISCA'23, ASPLOS'23, HPCA'23]

- June, March, Feb : Lectures + Hands-on labs + Invited talks



ISCA 2023 Real-World PIM Tutorial

Search

[Recent Changes](#) [Media Manager](#) [Sitemap](#)

Trace: • [start](#)

Real-world Processing-in-Memory Systems for Modern Workloads

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

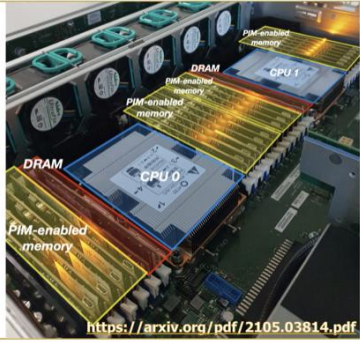
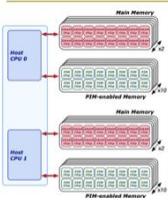
Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

Table of Contents

- [Real-world Processing-in-Memory Systems for Modern Workloads](#)
- [Tutorial Description](#)
- [Organizers](#)
- [Agenda \(June 18, 2023\)](#)
- [Lectures \(tentative\)](#)
- [Hands-on Labs \(tentative\)](#)
- [Learning Materials](#)

2,560-DPU Processing-in-Memory System



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

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

<https://events.safari.ethz.ch/isca-pim-tutorial/>

Real PIM Tutorial [ISCA 2023]

■ June 18: Lectures + Hands-on labs + Invited talks

ISCA 2023 Real-World PIM Tutorial Sunday, June 18, Orlando, Florida

Organizers: Juan Gómez Luna, Onur Mutlu, Ataberk Olgun
Program: <https://events.safari.ethz.ch/isca-pim-tutorial/>



Overview PIM | PNM | UPMEM PIM |
PNM for neural networks |
PNM for recommender systems |
PNM for ML workloads |
How to enable PIM? | PUM prototypes
Hands-on Labs: Benchmarking |
Accelerating real-world workloads



International Symposium on Computer Architecture (ISCA)

Real-world Processing-in-Memory Systems for Modern Workloads

<https://www.youtube.com/live/GIb5EgSrWk0?feature=share>

Room: Magnolia 16
Marriott World Center Orlando
Orlando, FL, USA
July 18th, 2023

SAFARI zoom

ISCA 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

Onur Mutlu Lectures
33.9K subscribers

Subscribed

57

Share

Download

Clip

...

1,687 views Streamed live on Jun 18, 2023 Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)

[https://www.youtube.com/
live/GIb5EgSrWk0](https://www.youtube.com/live/GIb5EgSrWk0)

[https://events.safari.ethz.ch/
isca-pim-tutorial/](https://events.safari.ethz.ch/isca-pim-tutorial/)

Tutorial Materials

Time	Speaker	Title	Materials
8:55am-9:00am	Dr. Juan Gómez Luna	Welcome & Agenda	(PDF) (PPT)
9:00am-10:20am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
10:20am-11:00am	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures / Programming General-purpose PIM	(PDF) (PPT)
11:20am-11:50am	Prof. Izzat El Hajj	High-throughput Sequence Alignment using Real Processing-in-Memory Systems	(PDF) (PPT)
11:50am-12:30pm	Dr. Christina Giannoula	SparseP: Towards Efficient Sparse Matrix Vector Multiplication for Real Processing-In-Memory Systems	(PDF) (PPT)
2:00pm-2:45pm	Dr. Sukhan Lee	Introducing Real-world HBM-PIM Powered System for Memory-bound Applications	(PDF) (PPT)
2:45pm-3:30pm	Dr. Juan Gómez Luna / Ataberk Olgun	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components / PUM Prototypes: PiDRAM	(PDF) (PPT) (PDF) (PPT)
4:00pm-4:40pm	Dr. Juan Gómez Luna	Accelerating Modern Workloads on a General-purpose PIM System	(PDF) (PPT)
4:40pm-5:20pm	Dr. Juan Gómez Luna	Adoption Issues: How to Enable PIM?	(PDF) (PPT)
5:20pm-5:30pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)

Real PIM Tutorial [ASPLOS 2023]

■ March 26: Lectures + Hands-on labs + Invited talks

Tutorial Materials

Time	Speaker	Title	Materials
9:00am-10:20am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
10:40am-12:00pm	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM	(PDF) (PPT)
1:40pm-2:20pm	Prof. Alexandra (Sasha) Fedorova (UBC)	Processing in Memory in the Wild	(PDF) (PPT)
2:20pm-3:20pm	Dr. Juan Gómez Luna & Ataberk Olgun	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components	(PDF) (PPT) (PDF) (PPT)
3:40pm-4:10pm	Dr. Juan Gómez Luna	Adoption issues: How to enable PIM? Accelerating Modern Workloads on a General-purpose PIM System	(PDF) (PPT) (PDF) (PPT)
4:10pm-4:50pm	Dr. Yongkee Kwon & Eddy (Chanwook) Park (SK Hynix)	System Architecture and Software Stack for GDDR6-AiM	(PDF) (PPT)
4:50pm-5:00pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)

ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

Onur Mutlu Lectures
32.1K subscribers

Subscribed

33

Share

Clip

Save

Views Streamed 7 days ago Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)

LOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

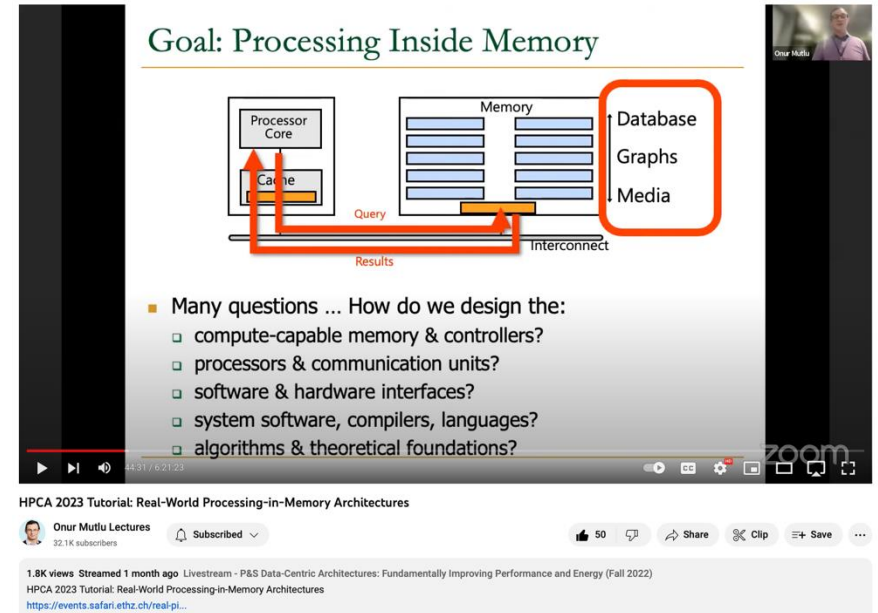
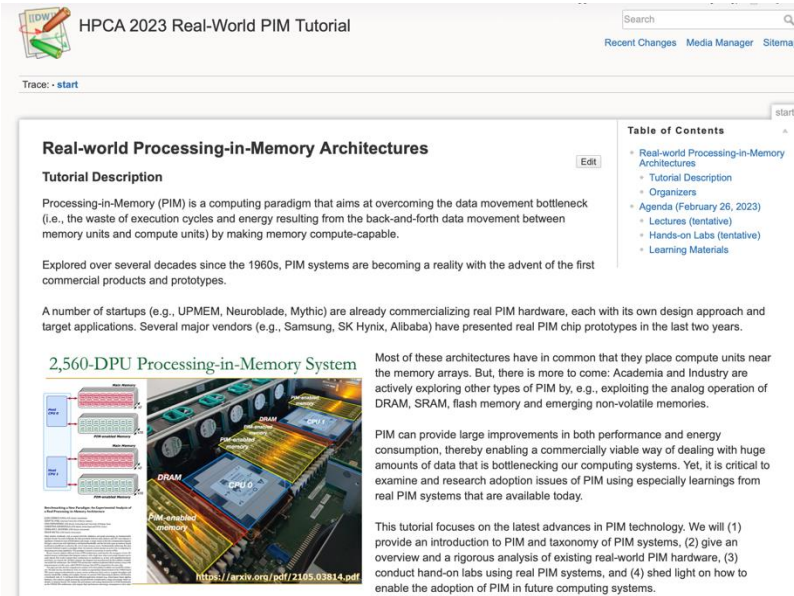
<https://events.safari.ethz.ch/asplos-2023/>

<https://www.youtube.com/watch?v=oYCaLcT0Kmo>

<https://events.safari.ethz.ch/asplos-pim-tutorial/>

Real PIM Tutorial [HPCA 2023]

■ February 26: Lectures + Hands-on labs + Invited Talks



Time	Speaker	Title	Materials
8:00am-8:40am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
8:40am-10:00am	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM	(PDF) (PPT)
10:20am-11:00am	Dr. Dimin Niu	A 3D Logic-to-DRAM Hybrid Bonding Process-Near-Memory Chip for Recommendation System	
11:00am-11:40am	Dr. Christina Giannoula	SparseP: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures	(PDF) (PPT)
1:30pm-2:10pm	Dr. Juan Gómez Luna	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components	(PDF) (PPT)
2:10pm-2:50pm	Dr. Manuel Le Gallo	Deep Learning Inference Using Computational Phase-Change Memory	
2:50pm-3:30pm	Dr. Juan Gómez Luna	PIM Adoption Issues: How to Enable PIM Adoption?	(PDF) (PPT)
3:40pm-5:40pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)

<https://www.youtube.com/watch?v=f5-nT1tbz5w>

<https://events.safari.ethz.ch/real-pim-tutorial/>

Latest Real PIM Tutorial [MICRO 2023]

■ October 29: Lectures + Hands-on labs + Invited talks

MICRO 2023 Real-World PIM Tutorial

Search

Recent Changes Media Manager Sitemap

Trace: - start

Real-world Processing-in-Memory Systems for Modern Workloads

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

Table of Contents

- Real-world Processing-in-Memory Systems for Modern Workloads
- Tutorial Description
- Livestream
- Organizers
- Agenda (Tentative, October 29, 2023)
- Lectures
- Learning Materials

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

2,560-DPU Processing-in-Memory System

Live in 74 days
October 29 at 6:00 PM

Notify me

Subscribed

MICRO 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

Onur Mutlu Lectures
34.6K subscribers

5 5

Share Save

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

<https://www.youtube.com/watch?v=ohUooNSIxOI>

<https://events.safari.ethz.ch/micro-pim-tutorial>

Agenda (Tentative, October 29, 2023)

Lectures

1. Introduction: PIM as a paradigm to overcome the data movement bottleneck.
2. PIM taxonomy: PNM (processing near memory) and PUM (processing using memory).
3. General-purpose PNM: UPMEM PIM.
4. PNM for neural networks: Samsung HBM-PIM, SK Hynix AiM.
5. PNM for recommender systems: Samsung AxDIMM, Alibaba PNM.
6. PUM prototypes: PiDRAM, SRAM-based PUM, Flash-based PUM.
7. Other approaches: Neuroblade, Mythic.
8. Adoption issues: How to enable PIM?
9. Hands-on labs: Programming a real PIM system.

Upcoming PIM Tutorial at ISCA 2024

ISCA 2024 Memory-Centric Computing Systems Tutorial

Saturday, June 29, Buenos Aires, Argentina

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

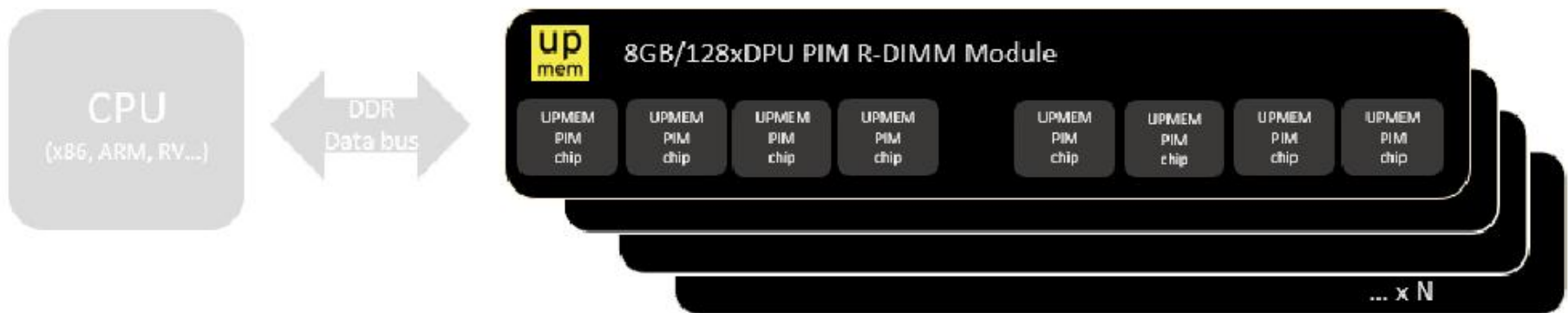
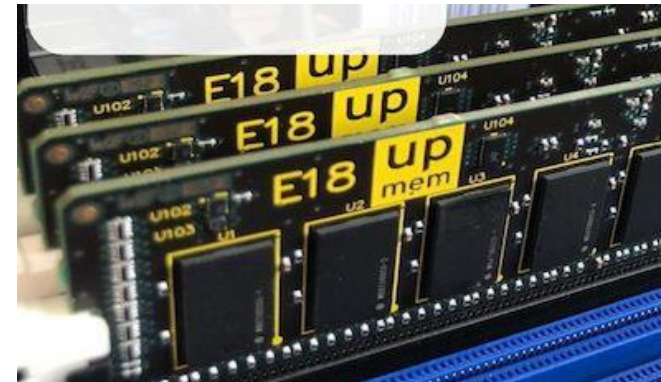
Program: <https://events.safari.ethz.ch/isca24-memorycentric-tutorial/>

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



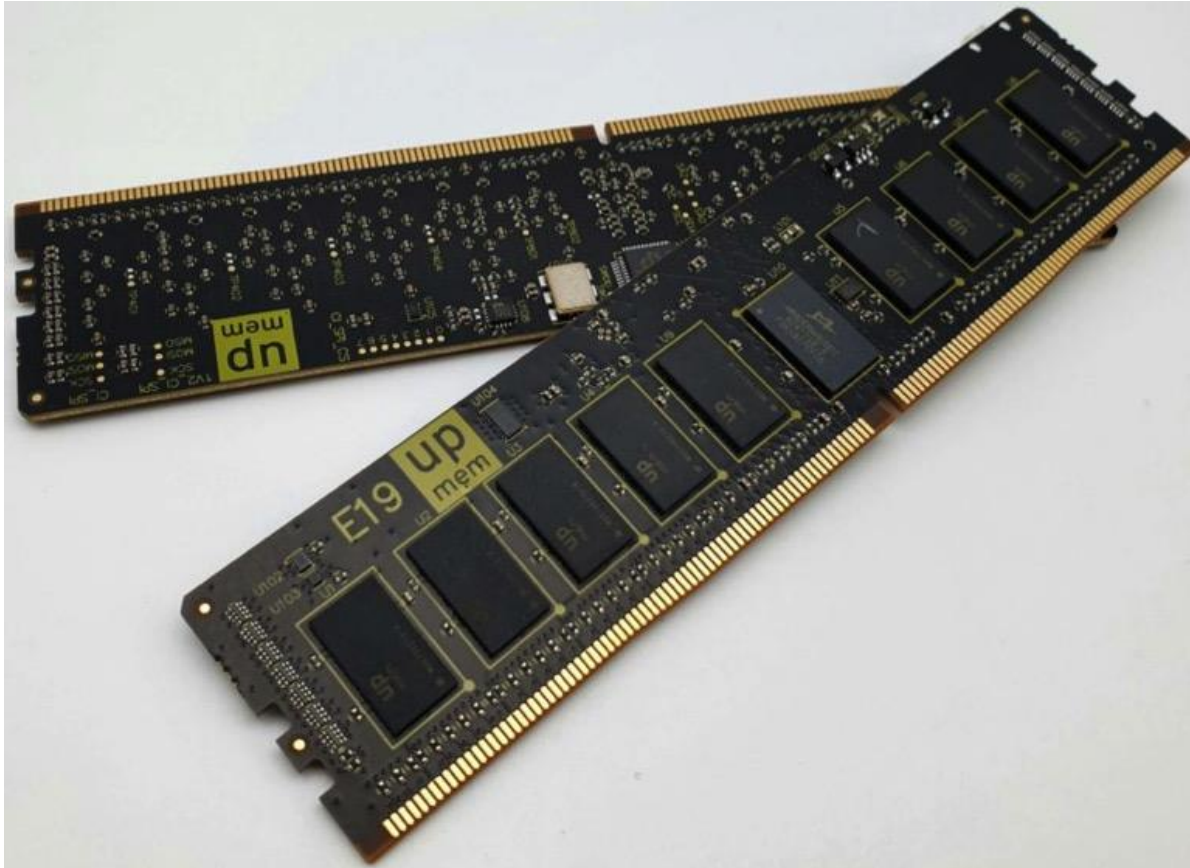
UPMEM Processing-in-DRAM Engine (2019)

- **Processing in DRAM Engine**
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.
- Replaces **standard DIMMs**
 - DDR4 R-DIMM modules
 - 8GB+128 DPUs (16 PIM chips)
 - Standard 2x-nm DRAM process
 - **Large amounts of** compute & memory bandwidth

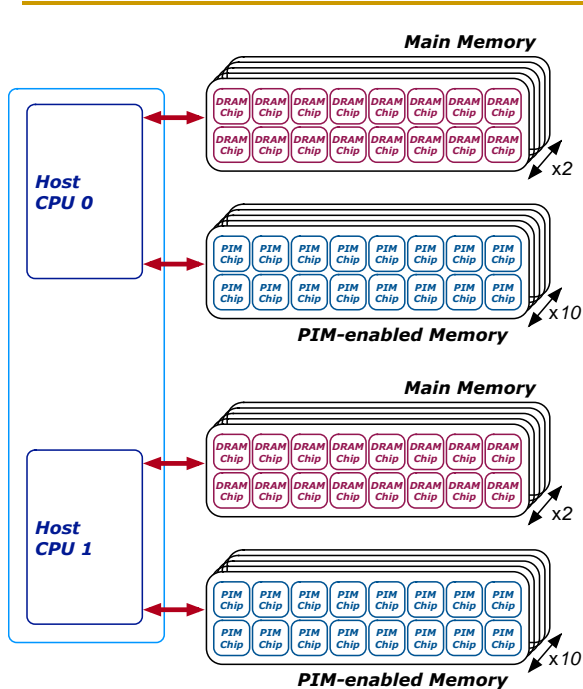


UPMEM Memory Modules

- E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
- P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz



2,560-DPU Processing-in-Memory System



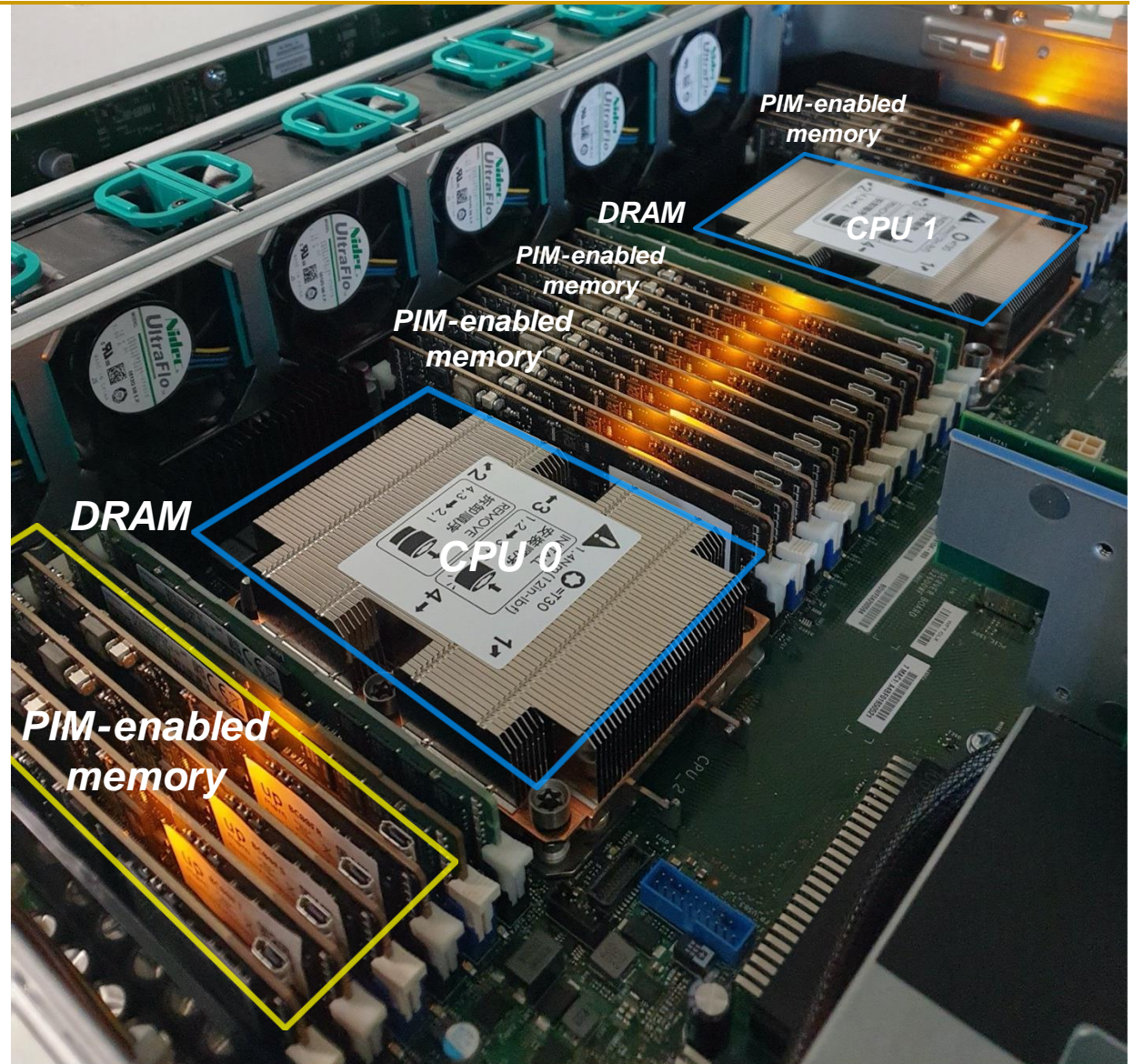
Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
 IZZAT EL HAJJ, American University of Beirut, Lebanon
 IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
 CHRISTINA GIANNOULA, ETH Zürich, Switzerland and NTUA, Greece
 GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
 ONUR MUTLU, ETH Zürich, Switzerland

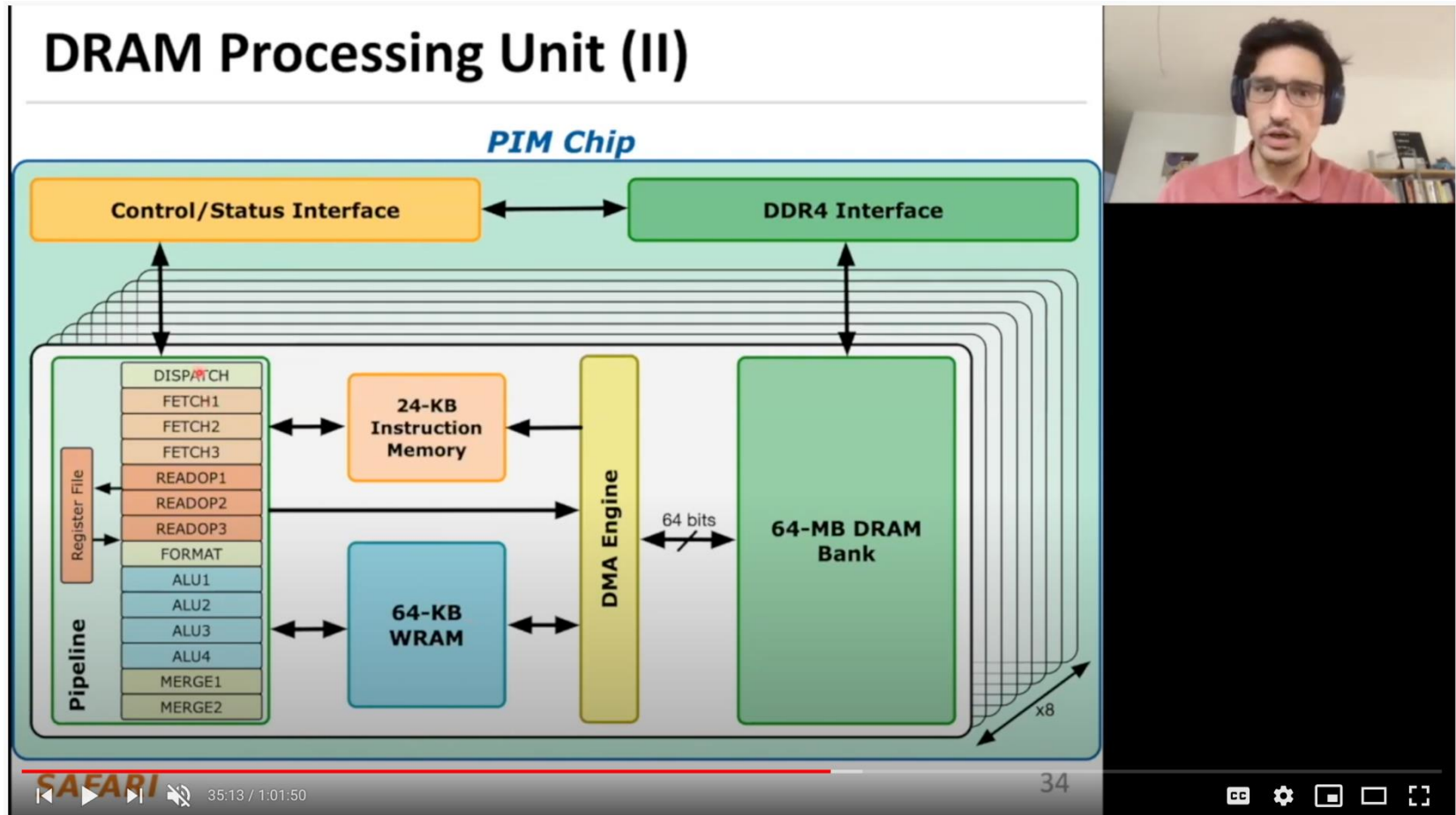
Many modern workloads, such as neural networks, databases, and graph processing, are fundamentally memory-bound. For such workloads, the data movement between main memory and CPU cores imposes a significant overhead in terms of both latency and energy. A major reason is that this communication happens through a narrow bus with high latency and limited bandwidth, and the low data reuse in memory-bound workloads is insufficient to amortize the cost of main memory access. Fundamentally addressing this *data movement bottleneck* requires a paradigm where the memory system assumes an active role in computing by integrating processing capabilities. This paradigm is known as *processing-in-memory (PIM)*.

Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware prototypes. In contrast, the UPMEM company has designed and manufactured the first publicly-available real-world 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.

This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present *PrIM (Processing-In-Memory benchmarks)*, a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 640 and 2,560 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.



More on the UPMEM PIM System



ETH ZÜRICH HAUPTGEBÄUDE

Computer Architecture - Lecture 12d: Real Processing-in-DRAM with UPMEM (ETH Zürich, Fall 2020)

1,120 views • Oct 31, 2020

30 0 SHARE SAVE ...



Onur Mutlu Lectures
16.7K subscribers

ANALYTICS

EDIT VIDEO

<https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=26>

Experimental Analysis of the UPMEM PIM Engine

Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

IZZAT EL HAJJ, American University of Beirut, Lebanon

IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain

CHRISTINA GIANNOULA, ETH Zürich, Switzerland and NTUA, Greece

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Many modern workloads, such as neural networks, databases, and graph processing, are fundamentally memory-bound. For such workloads, the data movement between main memory and CPU cores imposes a significant overhead in terms of both latency and energy. A major reason is that this communication happens through a narrow bus with high latency and limited bandwidth, and the low data reuse in memory-bound workloads is insufficient to amortize the cost of main memory access. Fundamentally addressing this *data movement bottleneck* requires a paradigm where the memory system assumes an active role in computing by integrating processing capabilities. This paradigm is known as *processing-in-memory* (PIM).

Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware prototypes. In contrast, the UPMEM company has designed and manufactured the first publicly-available real-world 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.

This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present *PrIM* (*Processing-In-Memory benchmarks*), a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 640 and 2,556 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.

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>

Recent SRC TECHCON Presentation

■ Dr. Juan Gomez-Luna

- Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware
- Based on two major works
 - <https://arxiv.org/pdf/2105.03814.pdf>
 - <https://arxiv.org/pdf/2207.07886.pdf>



Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-In-Memory Hardware

Year: 2021, Pages: 1-7

DOI Bookmark: [10.1109/IGSC54211.2021.9651614](https://doi.org/10.1109/IGSC54211.2021.9651614)

Authors

Juan Gómez-Luna, ETH Zürich

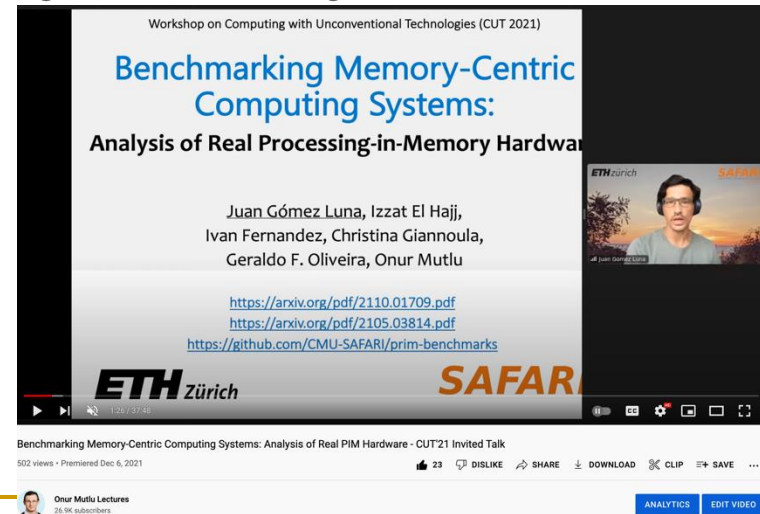
Izzat El Hajj, American University of Beirut

Ivan Fernandez, University of Malaga

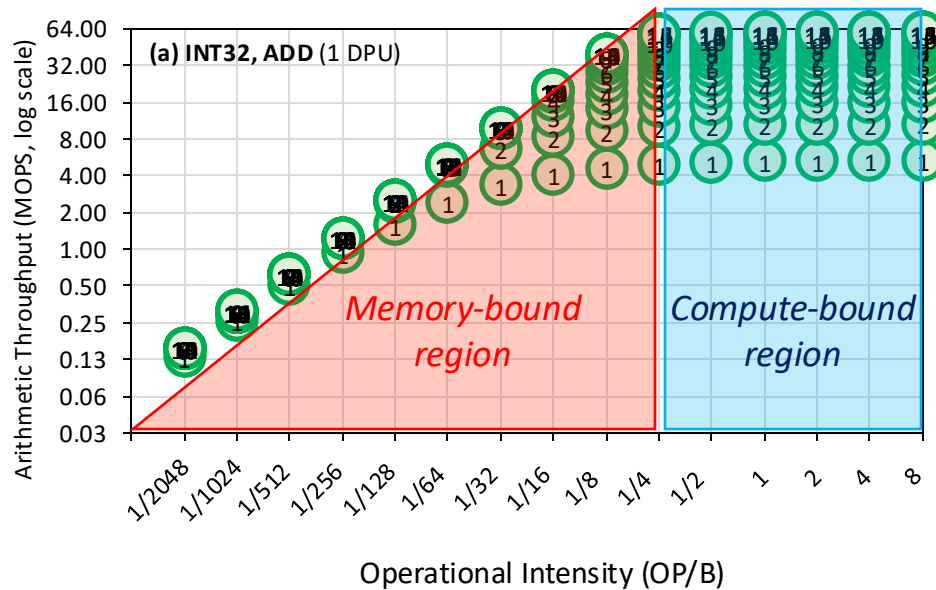
Christina Giannoula, National Technical University of Athens

Geraldo F. Oliveira, ETH Zürich

Onur Mutlu, ETH Zürich



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

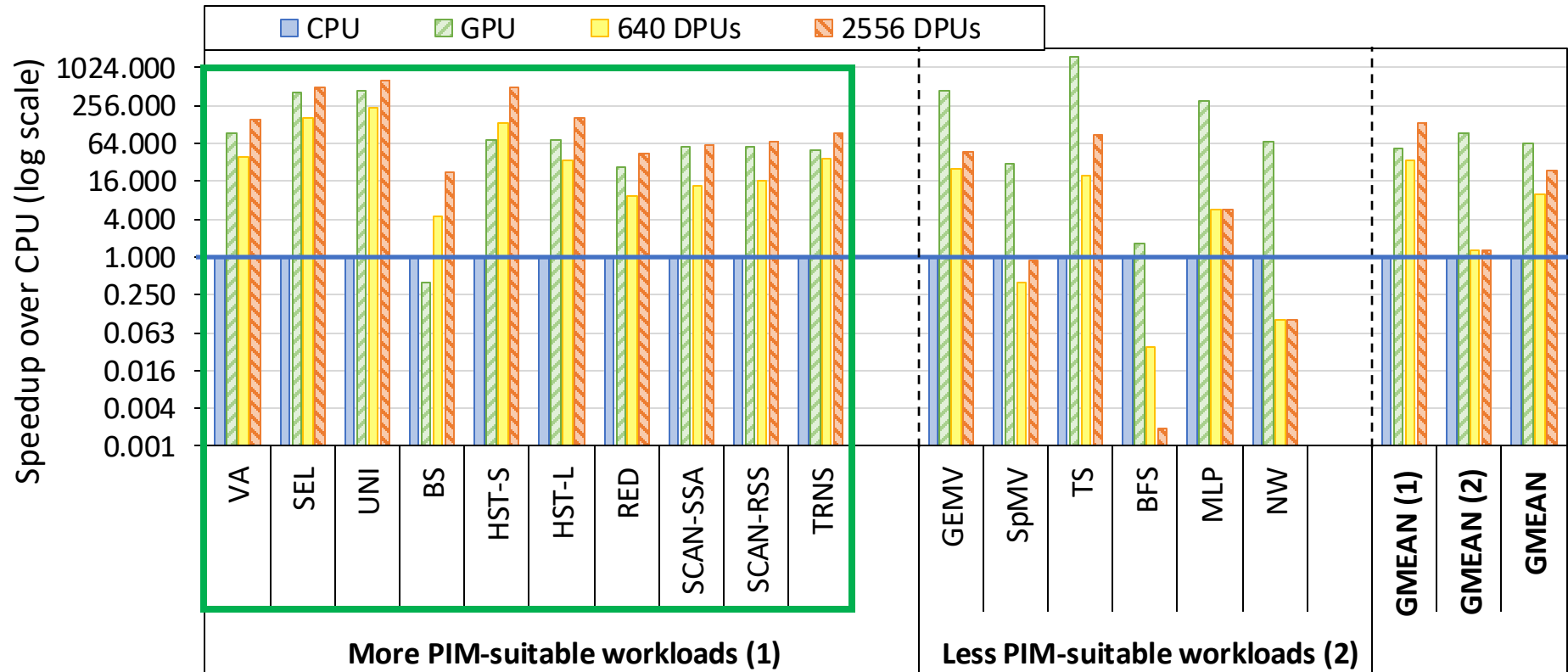
The UPMEM PIM architecture is fundamentally compute bound. As a result, the most suitable workloads are memory-bound.

Key Takeaway 2

Table 4: Evaluated CPU, GPU, and UPMEM-based PIM Systems.

System	Process Node	Processor Cores			Memory		TDP
		Total Cores	Frequency	Peak Performance	Capacity	Total Bandwidth	
Intel Xeon E3-1225 v6 CPU [241]	14 nm	4 (8 threads)	3.3 GHz	26.4 GFLOPS*	32 GB	37.5 GB/s	73 W
NVIDIA Titan V GPU [277]	14 nm	80 (5,120 SIMD lanes)	1.2 GHz	12,288.0 GFLOPS	12 GB	652.8 GB/s	250 W
2,556-DPU PIM System	2x nm	2,556 ⁹	350 MHz	894.6 GOPS	159.75 GB	1.7 TB/s	383 W [†]
640-DPU PIM System	2x nm	640	267 MHz	170.9 GOPS	40 GB	333.75 GB/s	96 W [†]

* Estimated GFLOPS = 3.3 GHz × 4 cores × 2 instructions per cycle.
⁹ Estimated TDP = $\frac{\text{Total DPU}}{\text{DPU}_1/\text{chip}} \times 1.2 \text{ W/chip}$ [199].



KEY TAKEAWAY 2

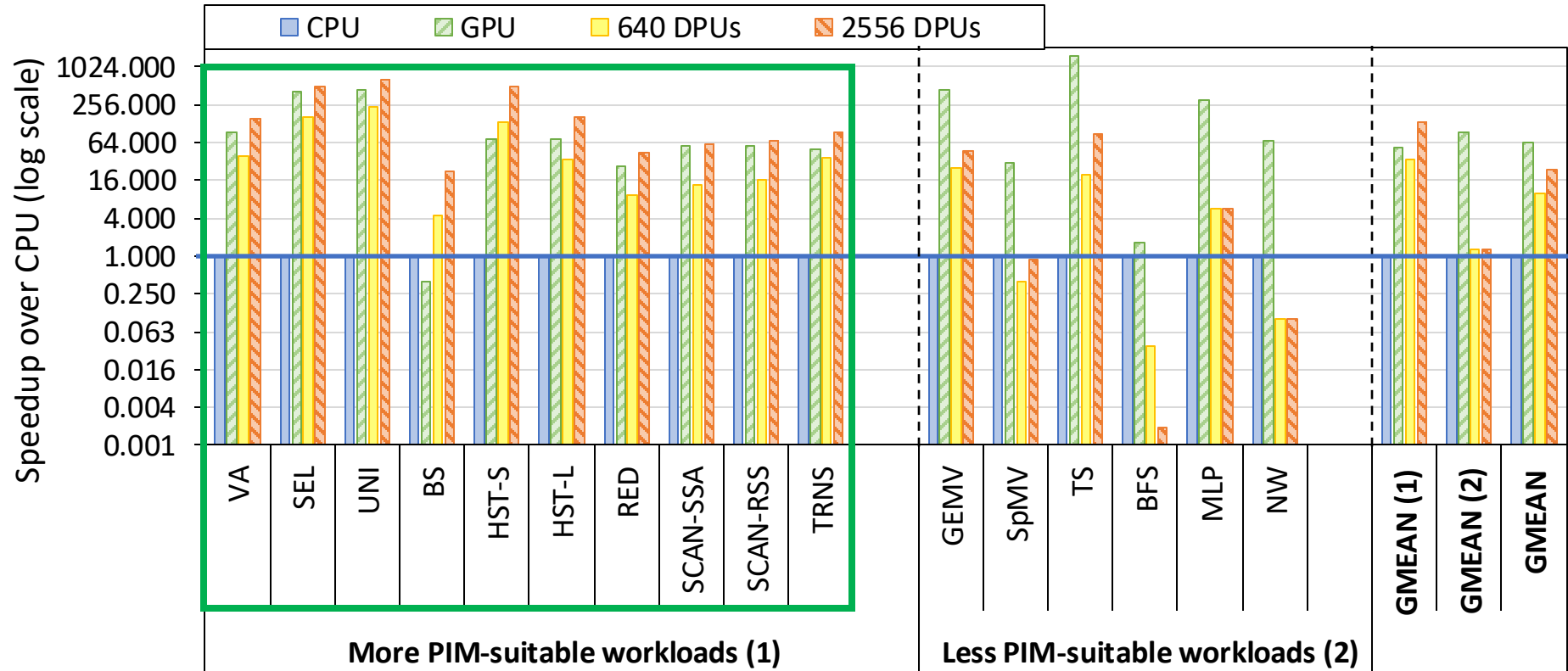
The most well-suited workloads for the UPMEM PIM architecture use no arithmetic operations or use only simple operations (e.g., bitwise operations and integer addition/subtraction).

Key Takeaway 3

Table 4: Evaluated CPU, GPU, and UPMEM-based PIM Systems.

System	Process Node	Processor Cores			Memory		TDP
		Total Cores	Frequency	Peak Performance	Capacity	Total Bandwidth	
Intel Xeon E3-1225 v6 CPU [241]	14 nm	4 (8 threads)	3.3 GHz	26.4 GFLOPS*	32 GB	37.5 GB/s	73 W
NVIDIA Titan V GPU [277]	14 nm	80 (5,120 SIMD lanes)	1.2 GHz	12,288.0 GFLOPS	12 GB	652.8 GB/s	250 W
2,556-DPU PIM System	2x nm	2,556 ⁹	350 MHz	894.6 GOPS	159.75 GB	1.7 TB/s	383 W [†]
640-DPU PIM System	2x nm	640	267 MHz	170.9 GOPS	40 GB	333.75 GB/s	96 W [†]

* Estimated GFLOPS = 3.3 GHz × 4 cores × 2 instructions per cycle.
⁹ Estimated TDP = $\frac{\text{Total DPU}}{\text{DPU}_1 \text{ chip}} \times 1.2 \text{ W/chip}$ [199].



KEY TAKEAWAY 3

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

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>

UPMEM PIM System Summary & Analysis

- Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu,
"Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware"
*Invited Paper at Workshop on Computing with Unconventional Technologies (**CUT**), Virtual, October 2021.*
[[arXiv version](#)]
[[PrIM Benchmarks Source Code](#)]
[[Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (37 minutes)]
[[Lightning Talk Video](#) (3 minutes)]

Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware

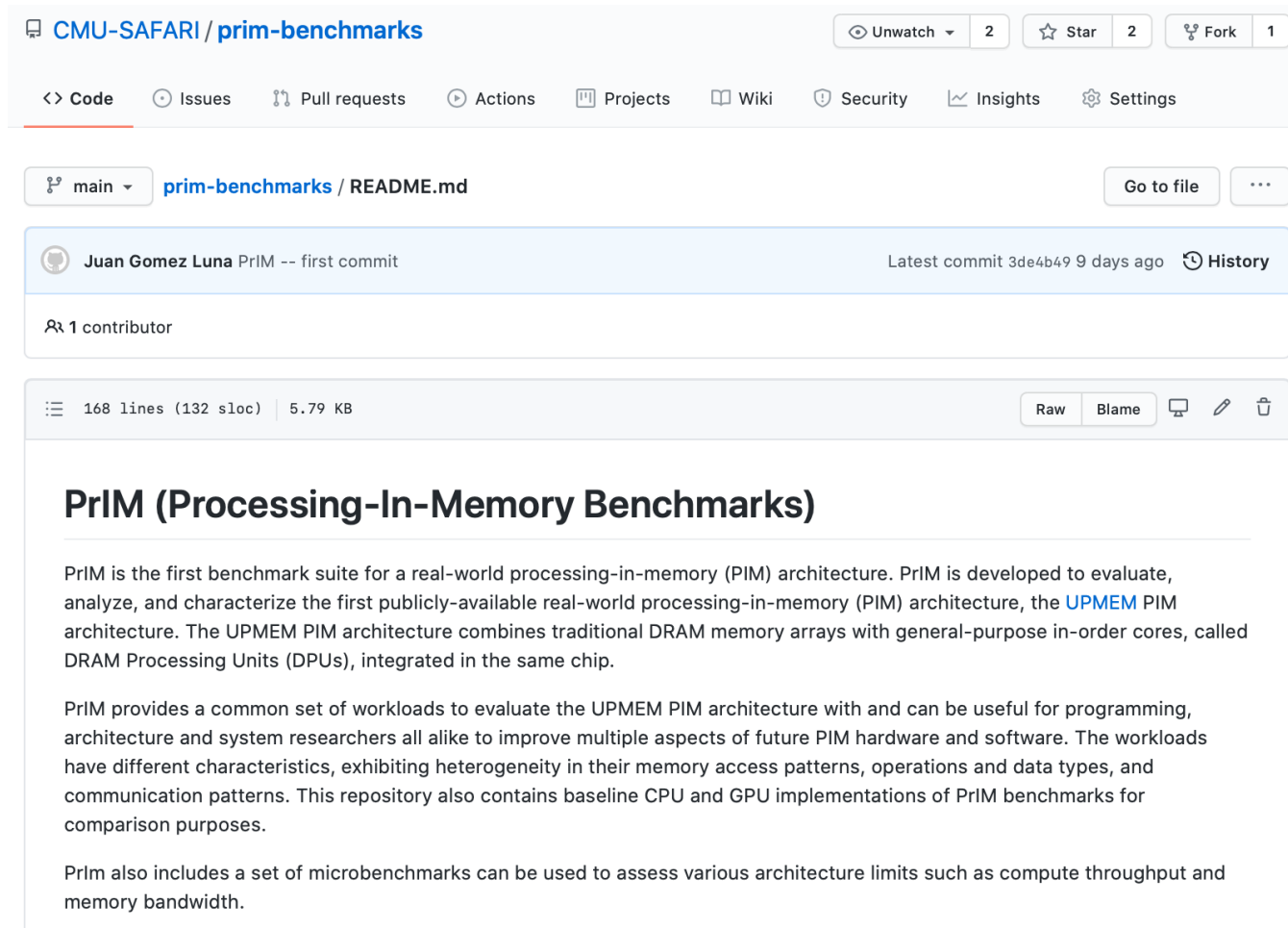
Juan Gómez-Luna	Izzat El Hajj	Ivan Fernandez	Christina Giannoula	Geraldo F. Oliveira	Onur Mutlu
<i>ETH Zürich</i>	<i>American University of Beirut</i>	<i>University of Malaga</i>	<i>National Technical University of Athens</i>	<i>ETH Zürich</i>	<i>ETH Zürich</i>

PrIM Benchmarks: Application Domains

Domain	Benchmark	Short name
Dense linear algebra	Vector Addition	VA
	Matrix-Vector Multiply	GEMV
Sparse linear algebra	Sparse Matrix-Vector Multiply	SpMV
Databases	Select	SEL
	Unique	UNI
Data analytics	Binary Search	BS
	Time Series Analysis	TS
Graph processing	Breadth-First Search	BFS
Neural networks	Multilayer Perceptron	MLP
Bioinformatics	Needleman-Wunsch	NW
Image processing	Image histogram (short)	HST-S
	Image histogram (large)	HST-L
Parallel primitives	Reduction	RED
	Prefix sum (scan-scan-add)	SCAN-SSA
	Prefix sum (reduce-scan-scan)	SCAN-RSS
	Matrix transposition	TRNS

PrIM Benchmarks are Open Source

- 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 PIM Architecture

Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System

**JUAN GÓMEZ-LUNA¹, IZZAT EL HAJJ², IVAN FERNANDEZ^{1,3}, CHRISTINA GIANNOULA^{1,4},
GERALDO F. OLIVEIRA¹, AND ONUR MUTLU¹**

¹ETH Zürich

²American University of Beirut

³University of Malaga

⁴National Technical University of Athens

Corresponding author: Juan Gómez-Luna (e-mail: juang@ethz.ch).

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

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

Understanding a Modern PIM Architecture



The video player shows a lecture titled "Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization". The speaker is Juan Gómez Luna, with co-authors Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu. The video includes links to the arXiv paper and GitHub benchmarks. The player interface shows a progress bar at 2:26 / 2:57:10, and the channel name "Onur Mutlu Lectures" with 18.7K subscribers. The video has 2,579 views and was streamed live on Jul 12, 2021.

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>

ETH Zürich SAFARI

2:26 / 2:57:10

SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture

2,579 views • Streamed live on Jul 12, 2021

93 0 SHARE SAVE ...

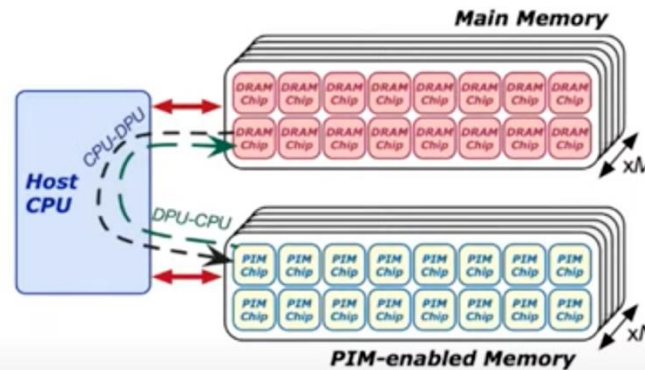
Onur Mutlu Lectures
18.7K subscribers

SUBSCRIBED

More on Analysis of the UPMEM PIM Engine

Inter-DPU Communication

- There is **no direct communication channel between DPUs**



- Inter-DPU communication takes place via the host CPU using CPU-DPU and DPU-CPU transfers
- Example communication patterns:
 - Merging of partial results to obtain the final result
 - Only DPU-CPU transfers
 - Redistribution of intermediate results for further computation
 - DPU-CPU transfers and CPU-DPU transfers



SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture

1,868 views • Streamed live on Jul 12, 2021

81 0 SHARE SAVE ...



Onur Mutlu Lectures
17.6K subscribers

Talk Title: Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization
Dr. Juan Gómez-Luna, SAFARI Research Group, D-ITET, ETH Zurich

ANALYTICS

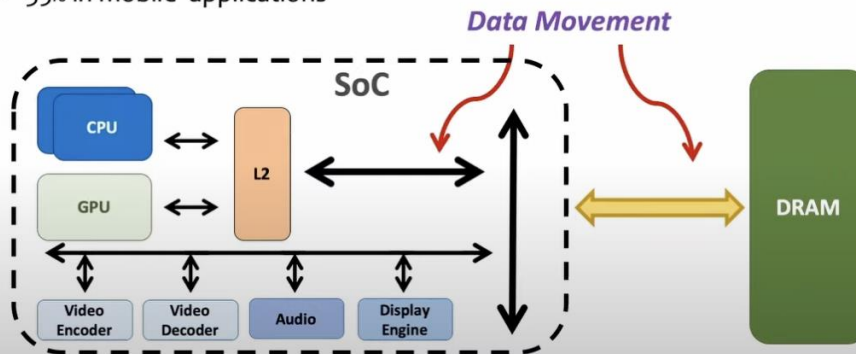
EDIT VIDEO

https://www.youtube.com/watch?v=D8Hjy2iU9I4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9

More on Analysis of the UPMEM PIM Engine

Data Movement in Computing Systems

- Data movement dominates performance and is a major system energy bottleneck
- Total system energy: data movement accounts for
 - 62% in consumer applications*,
 - 40% in scientific applications*,
 - 35% in mobile applications*



* Boroumand et al., "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS 2018

* Kestor et al., "Quantifying the Energy Cost of Data Movement in Scientific Applications," IISWC 2013

* Pandiyan and Wu, "Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms," IISWC 2014

SAFARI

3

Understanding a Modern Processing-in-Memory Arch: Benchmarking & Experimental Characterization; 21m

3,482 views • Premiered Jul 25, 2021

38 0 SHARE SAVE ...



Onur Mutlu Lectures

17.9K subscribers

ANALYTICS

EDIT VIDEO

https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXY2Zi8_VVChACnON4sfh2bJ5IrD&index=159

ML Training on a Real PIM System

Machine Learning Training on a Real Processing-in-Memory System

Juan Gómez-Luna¹ Yuxin Guo¹ Sylvan Brocard² Julien Legriel²
Remy Cimadomo² Geraldo F. Oliveira¹ Gagandeep Singh¹ Onur Mutlu¹

¹ETH Zürich ²UPMEM

An Experimental Evaluation of Machine Learning Training on a Real Processing-in-Memory System

Juan Gómez-Luna¹ Yuxin Guo¹ Sylvan Brocard² Julien Legriel²
Remy Cimadomo² Geraldo F. Oliveira¹ Gagandeep Singh¹ Onur Mutlu¹

¹ETH Zürich ²UPMEM

Short version: <https://arxiv.org/pdf/2206.06022.pdf>

Long version: <https://arxiv.org/pdf/2207.07886.pdf>

<https://www.youtube.com/watch?v=qeukNs5XI3g&t=11226s>

ML Training on a Real PIM System

- Need to optimize data representation
 - (1) fixed-point
 - (2) quantization
 - (3) hybrid precision
- Use **lookup tables (LUTs)** to implement complex functions (e.g., sigmoid)
- Optimize data placement & layout for **streaming**
- Large speedups: **2.8X/27X vs. CPU, 1.3x/3.2x vs. GPU**

ML Training on Real PIM Talk Video

Comparison to CPU and GPU (III)

- Decision tree and K-means with Criteo 1TB dataset

(a) Decision Tree

Method	DTR (ms)	CPU (ms)	GPU (ms)
PIM Kernel	~10,000	~100,000	~10,000
CPU-PIM	~10,000	~100,000	~10,000
Inter PIM	~10,000	~100,000	~10,000
PIM-CPU	~10,000	~100,000	~10,000

(b) K-means

Method	KME (ms)	CPU (ms)	GPU (ms)
PIM Kernel	~10,000	~100,000	~10,000
CPU-PIM	~10,000	~100,000	~10,000
Inter PIM	~10,000	~100,000	~10,000
PIM-CPU	~10,000	~100,000	~10,000

PIM version of DTR is **62x** faster than the CPU version and **4.5x** faster than the GPU version

PIM version of KME is **2.7x** faster than the CPU version and **3.2x** faster than the GPU version

13:39 / 15:20 • Comparison to CPU and GPU (II) >

Machine Learning Training on Memory-centric Computing Systems, Juan Gómez-Luna for ISPASS 2023



Onur Mutlu Lectures
32.9K subscribers

Analytics

Edit video

9

Share

Download

Clip

Save

...

242 views 11 days ago Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)
Evaluating Machine Learning Workloads on Memory-centric Computing Systems

ML Training on Real PIM Systems

- Juan Gómez Luna, Yuxin Guo, Sylvan Brocard, Julien Legriel, Remy Cimadomo, Geraldo F. Oliveira, Gagandeep Singh, and Onur Mutlu,
"Evaluating Machine Learning Workloads on Memory-Centric Computing Systems"
Proceedings of the 2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Raleigh, North Carolina, USA, April 2023.
[[arXiv version](#), 16 July 2022.]
[[PIM-ML Source Code](#)]
Best paper session.

An Experimental Evaluation of Machine Learning Training on a Real Processing-in-Memory System

Juan Gómez-Luna¹ Yuxin Guo¹ Sylvan Brocard² Julien Legriel²
Remy Cimadomo² Geraldo F. Oliveira¹ Gagandeep Singh¹ Onur Mutlu¹
¹ETH Zürich ²UPMEM

<https://github.com/CMU-SAFARI/pim-ml>

SpMV Multiplication on Real PIM Systems

- Appears at SIGMETRICS 2022

***SparseP*: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Systems**

CHRISTINA GIANNOULA, ETH Zürich, Switzerland and National Technical University of Athens, Greece

IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

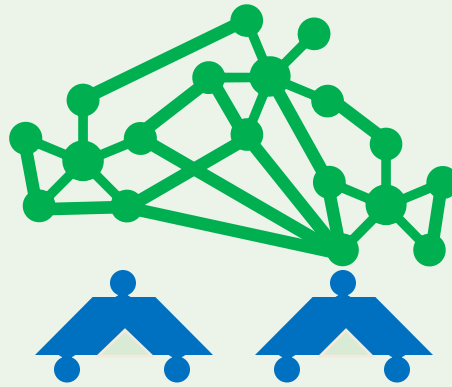
NECTARIOS KOZIRIS, National Technical University of Athens, Greece

GEORGIOS GOUMAS, National Technical University of Athens, Greece

ONUR MUTLU, ETH Zürich, Switzerland

<https://arxiv.org/pdf/2201.05072.pdf>

<https://github.com/CMU-SAFARI/SparseP>



SparseP

Towards Efficient Sparse Matrix Vector Multiplication
on Real Processing-In-Memory Architectures

Christina Giannoula

Ivan Fernandez, Juan Gomez-Luna,
Nectarios Koziris, Georgios Goumas, Onur Mutlu

SAFARI **ETH** zürich



UNIVERSIDAD
DE MÁLAGA

SparseP: Key Contributions

1. Efficient SpMV kernels for current & future PIM systems

- SparseP library = 25 SpMV kernels
 - Compression, data types, data partitioning, synchronization, load balancing

SparseP is Open-Source

SparseP: <https://github.com/CMU-SAFARI/SparseP>

2. Comprehensive analysis of SpMV on the first commercially-available real PIM system



- 26 sparse matrices
- Comparisons to state-of-the-art CPU and GPU systems
- Recommendations for software, system and hardware designers

Recommendations for Architects and Programmers

Full Paper: <https://arxiv.org/pdf/2201.05072.pdf>

SparseP Talk Video



SparseP

Towards Efficient Sparse Matrix Vector Multiplication
on Real Processing-In-Memory Architectures

Christina Giannoula
Ivan Fernandez, Juan Gomez-Luna,
Nectarios Koziris, Georgios Goumas, Onur Mutlu

SAFARI ETH zürich CSLab

0:02 / 55:25

Christina Gian...

Processing-in-Memory Course: Lecture 11: SpMV on a Real PIM Architecture - Spring 2022

149 views • Streamed live on May 19, 2022

👍 12 🗑 DISLIKE ➦ SHARE ⬇ DOWNLOAD ✂ CLIP ≡+ SAVE ...



Onur Mutlu Lectures
25K subscribers

ANALYTICS

EDIT VIDEO

More on SparseP

Christina Giannoula, Ivan Fernandez, Juan Gomez-Luna, Nectarios Koziris, Georgios Goumas, and Onur Mutlu,

"SparseP: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures"

Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS), Mumbai, India, June 2022.

[\[Extended arXiv Version\]](#)

[\[Abstract\]](#)

[\[Slides \(pptx\) \(pdf\)\]](#)

[\[Long Talk Slides \(pptx\) \(pdf\)\]](#)

[\[SparseP Source Code\]](#)

[\[Talk Video \(16 minutes\)\]](#)

[\[Long Talk Video \(55 minutes\)\]](#)

SparseP: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Systems

CHRISTINA GIANNOULA, ETH Zürich, Switzerland and National Technical University of Athens, Greece

IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

NECTARIOS KOZIRIS, National Technical University of Athens, Greece

GEORGIOS GOUMAS, National Technical University of Athens, Greece

ONUR MUTLU, ETH Zürich, Switzerland

[**https://github.com/CMU-SAFARI/SparseP**](https://github.com/CMU-SAFARI/SparseP)

Transcendental Functions on Real PIM Systems

- Maurus Item, Juan Gómez Luna, Yuxin Guo, Geraldo F. Oliveira, Mohammad Sadrosadati, and Onur Mutlu,

"TransPimLib: Efficient Transcendental Functions for Processing-in-Memory Systems"

*Proceedings of the 2023 IEEE International Symposium on Performance Analysis of Systems and Software (**ISPASS**), Raleigh, North Carolina, USA, April 2023.*

[[arXiv version](#)]

[[Slides \(pptx\)](#) ([pdf](#))]

[[TransPimLib Source Code](#)]

[[Talk Video](#) (17 minutes)]

TransPimLib: Efficient Transcendental Functions for Processing-in-Memory Systems

Maurus Item
Geraldo F. Oliveira

Juan Gómez-Luna
Mohammad Sadrosadati

Yuxin Guo
Onur Mutlu

ETH Zürich

[**https://github.com/CMU-SAFARI/transpimlib**](https://github.com/CMU-SAFARI/transpimlib)

Sequence Alignment on Real PIM Systems

- Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez Luna, Onur Mutlu, and Izzat El Hajj,
"A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems"
Bioinformatics, [published online on] 27 March 2023.
[Online link at Bioinformatics Journal]
[arXiv preprint]
[AiM Source Code]

A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab¹ Amir Nassereldine¹ Mohammed Alser² Juan Gómez Luna²
Onur Mutlu² Izzat El Hajj¹

¹American University of Beirut ²ETH Zürich

<https://github.com/CMU-SAFARI/alignment-in-memory>



Summary

- Sequence alignment on traditional systems is limited by the **memory bandwidth bottleneck**
- **Processing-in-memory (PIM)** overcomes this bottleneck by placing cores near the memory
- Our framework, **Alignment-in-Memory (AIM)**, is a PIM framework that supports multiple alignment algorithms (NW, SWG, GenASM, WFA)
 - Implemented on UPMEM, the first real PIM system
- Results show **substantial speedups over both CPUs (1.8X-28X) and GPUs (1.2X-2.7X)**
- AIM is available at:
 - <https://github.com/CMU-SAFARI/alignment-in-memory>

Homomorphic Operations on Real PIM Systems

- Harshita Gupta, Mayank Kabra, Juan Gómez-Luna, Konstantinos Kanellopoulos, and Onur Mutlu,
"Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System"
Proceedings of the 2023 IEEE International Symposium on Workload Characterization Poster Session (IISWC), Ghent, Belgium, October 2023.
[\[arXiv version\]](#)
[\[Lightning Talk Slides \(pptx\) \(pdf\)\]](#)
[\[Poster \(pptx\) \(pdf\)\]](#)

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

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

Samsung Function-in-Memory DRAM (2021)



Samsung Develops Industry's First High Bandwidth Memory with AI Processing Power

Korea on February 17, 2021

Audio



Share



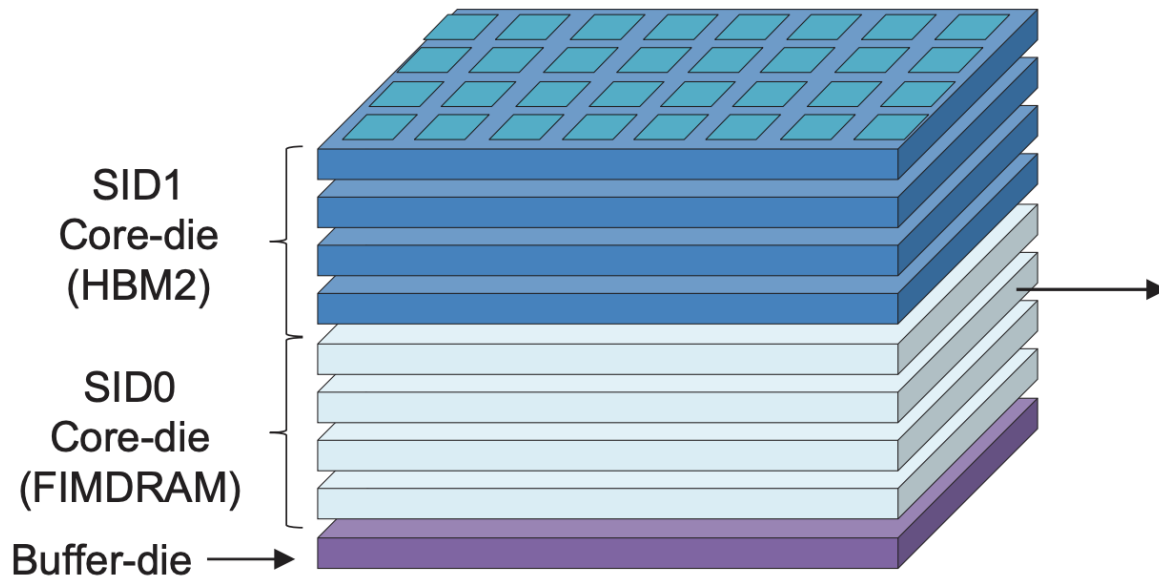
*The new architecture will deliver over twice the system performance
and reduce energy consumption by more than 70%*

Samsung Electronics, the world leader in advanced memory technology, today announced that it has developed the industry's first High Bandwidth Memory (HBM) integrated with artificial intelligence (AI) processing power – the HBM-PIM. **The new processing-in-memory (PIM) architecture brings powerful AI computing capabilities inside high-performance memory, to accelerate large-scale processing in data centers, high performance computing (HPC) systems and AI-enabled mobile applications.**

Kwangil Park, senior vice president of Memory Product Planning at Samsung Electronics stated, "Our groundbreaking HBM-PIM is the industry's first programmable PIM solution tailored for diverse AI-driven workloads such as HPC, training and inference. We plan to build upon this breakthrough by further collaborating with AI solution providers for even more advanced PIM-powered applications."

Samsung Function-in-Memory DRAM (2021)

■ FIMDRAM based on HBM2



[3D Chip Structure of HBM with FIMDRAM]

Chip Specification

128DQ / 8CH / 16 banks / BL4

32 PCU blocks (1 FIM block/2 banks)

1.2 TFLOPS (4H)

**FP16 ADD /
Multiply (MUL) /
Multiply-Accumulate (MAC) /
Multiply-and- Add (MAD)**

ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyoon Choi¹, Hyun-Sung Shin¹, Jin Kim¹, BengSeng Phuah¹, HyoungMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang², Shinhaeng Kang¹, Yuhwan Ro³, Seungwoo Seo³, JoonHo Song³, Jaeyoun Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

¹Samsung Electronics, Hwaseong, Korea

²Samsung Electronics, San Jose, CA

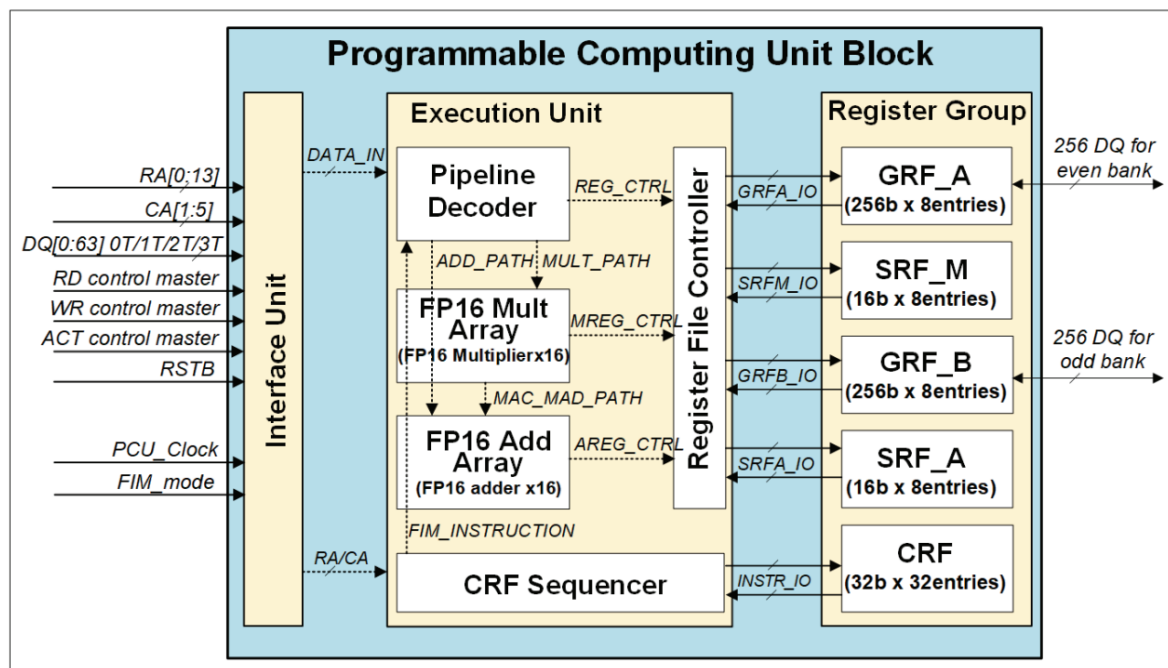
³Samsung Electronics, Suwon, Korea

Samsung Function-in-Memory DRAM (2021)

Programmable Computing Unit

■ Configuration of PCU block

- Interface unit to control data flow
- Execution unit to perform operations
- Register group
 - 32 entries of CRF for instruction memory
 - 16 GRF for weight and accumulation
 - 16 SRF to store constants for MAC operations



[Block diagram of PCU in FIMDRAM]

ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6Gb Function-in-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyoon Choi¹, Hyun-Sung Shin¹, Jin Kim¹, BengSeng Phuah¹, HyoungMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, Sooyoung Kim¹, Eun-Bong Kim¹, David Wang¹, Shinhaeng Kang¹, Yuhwan Ro¹, Seungwoo Seo¹, JoonHo Song¹, Jaeyoun Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

¹Samsung Electronics, Hwaseong, Korea

²Samsung Electronics, San Jose, CA

³Samsung Electronics, Suwon, Korea

Samsung Function-in-Memory DRAM (2021)

[Available instruction list for FIM operation]

Type	CMD	Description
Floating Point	ADD	FP16 addition
	MUL	FP16 multiplication
	MAC	FP16 multiply-accumulate
	MAD	FP16 multiply and add
Data Path	MOVE	Load or store data
	FILL	Copy data from bank to GRFs
Control Path	NOP	Do nothing
	JUMP	Jump instruction
	EXIT	Exit instruction

ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-in-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

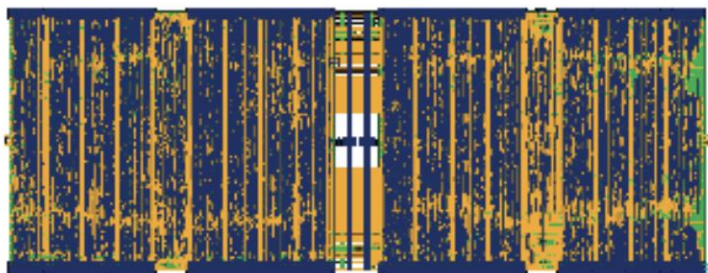
Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo-Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyoon Choi¹, Hyun-Sung Shin¹, Jin Kim¹, BengSeng Phuah¹, HyounMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang¹, Shinhaeng Kang¹, Yuhwan Ro¹, Seungwoo Seo¹, JoonHo Song¹, Jaeyoun Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

¹Samsung Electronics, Hwaseong, Korea
²Samsung Electronics, San Jose, CA
³Samsung Electronics, Suwon, Korea

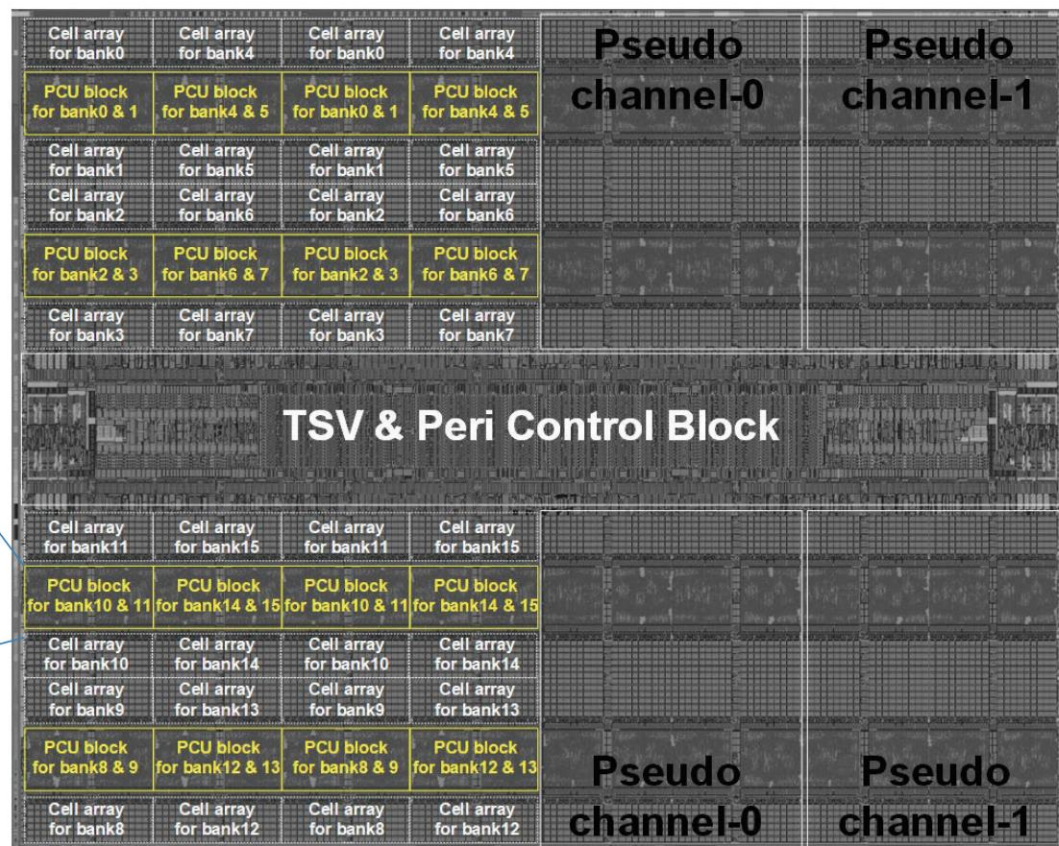
Samsung Function-in-Memory DRAM (2021)

Chip Implementation

- Mixed design methodology to implement FIMDRAM
 - Full-custom + Digital RTL



[Digital RTL design for PCU block]



ISSCC 2021 / SESSION 25 / DRAM / 25.4

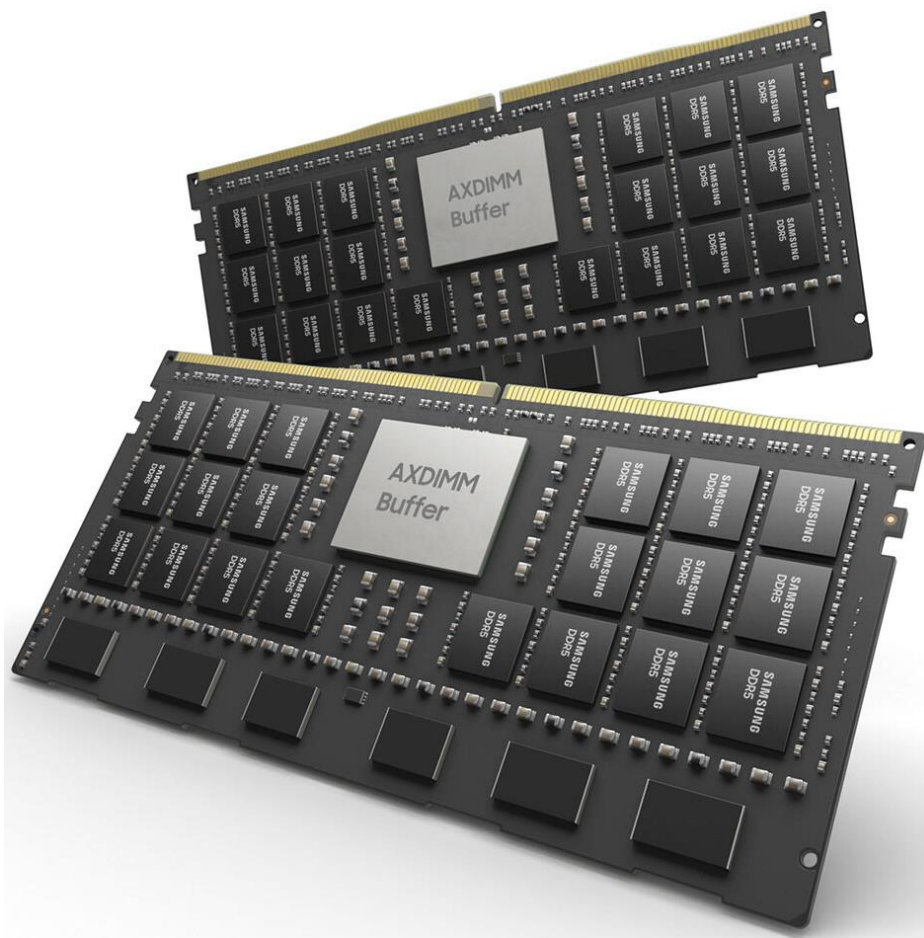
25.4 A 20nm 6Gb Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyeon Choi¹, Hyun-Sung Shim¹, Jin Kim¹, BengSeng Phuah¹, HyounMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang¹, Shinhaeng Kang¹, Yuhwan Ro¹, Seungwoo Seo¹, JoonHo Song¹, Jaeyoun Yoon¹, Kyomin Sohn¹, Nam Sung Kim¹

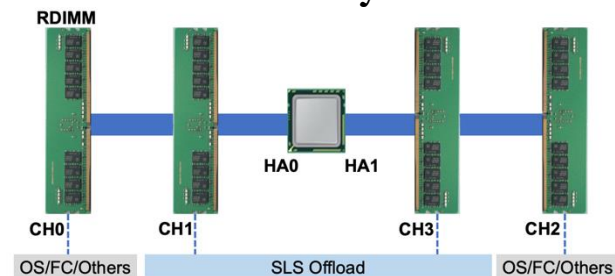
¹Samsung Electronics, Hwaseong, Korea
²Samsung Electronics, San Jose, CA
³Samsung Electronics, Suwon, Korea

Samsung AxDIMM (2021)

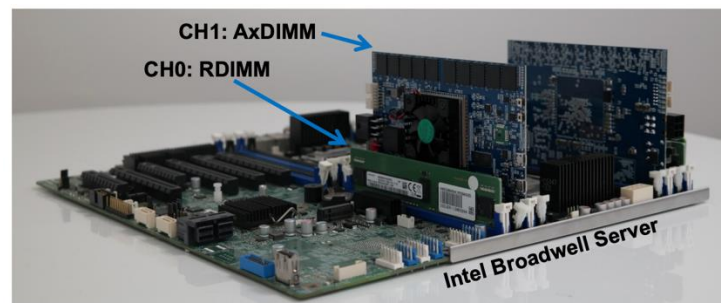
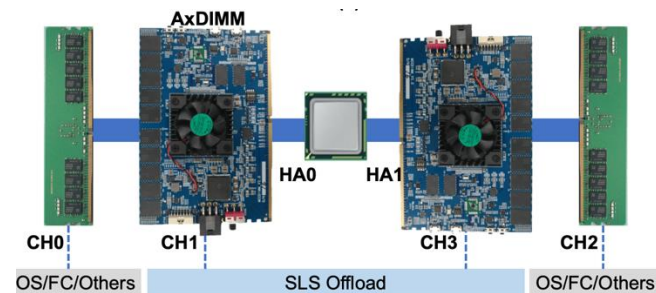
- DDRx-PIM
 - DLRM recommendation system



Baseline System



AxDIMM System



SK Hynix Accelerator-in-Memory (2022)

SK hynix Develops PIM, Next-Generation AI Accelerator

February 16, 2022



Seoul, February 16, 2022

SK hynix (or "the Company", www.skhynix.com) announced on February 16 that it has developed PIM*, a next-generation memory chip with computing capabilities.

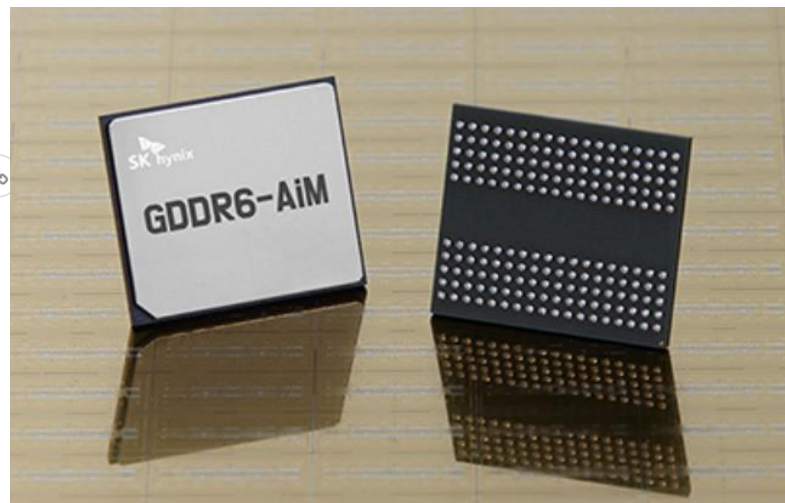
**PIM(Processing In Memory): A next-generation technology that provides a solution for data congestion issues for AI and big data by adding computational functions to semiconductor memory*

It has been generally accepted that memory chips store data and CPU or GPU, like human brain, process data. SK hynix, following its challenge to such notion and efforts to pursue innovation in the next-generation smart memory, has found a breakthrough solution with the development of the latest technology.

SK hynix plans to showcase its PIM development at the world's most prestigious semiconductor conference, 2022 ISSCC*, in San Francisco at the end of this month. The company expects continued efforts for innovation of this technology to bring the memory-centric computing, in which semiconductor memory plays a central role, a step closer to the reality in devices such as smartphones.

**ISSCC: The International Solid-State Circuits Conference will be held virtually from Feb. 20 to Feb. 24 this year with a theme of "Intelligent Silicon for a Sustainable World"*

For the first product that adopts the PIM technology, SK hynix has developed a sample of GDDR6-AiM (Accelerator* in memory). The GDDR6-AiM adds computational functions to GDDR6* memory chips, which process data at 16Gbps. A combination of GDDR6-AiM with CPU or GPU instead of a typical DRAM makes certain computation speed 16 times faster. GDDR6-AiM is widely expected to be adopted for machine learning, high-performance computing, and big data computation and storage.



11.1 A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications

Seongju Lee, SK hynix, Icheon, Korea

In Paper 11.1, SK Hynix describes an 1ynm, GDDR6-based accelerator-in-memory with a command set for deep-learning operation. The 8Gb design achieves a peak throughput of 1TFLOPS with 1GHz MAC operations and supports major activation functions to improve accuracy.

SK Hynix Accelerator-in-Memory (2022)

System Architecture and Software Stack for GDDR6-AiM

Yongkee Kwon and Chanwook Park
SK hynix inc.

5:42 / 5:42

ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads



Onur Mutlu Lectures
32.1K subscribers

Analytics

Edit video

33



Share

Download

Clip

Save



1,146 views Streamed live on Mar 26, 2023 Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)

ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

<https://events.safari.ethz.ch/asplos-...>

<https://www.youtube.com/watch?v=oYCaLcT0Kmo>

AliBaba PIM Recommendation System (2022)

ISSCC 2022 / February 24, 2022 / 8:30 AM

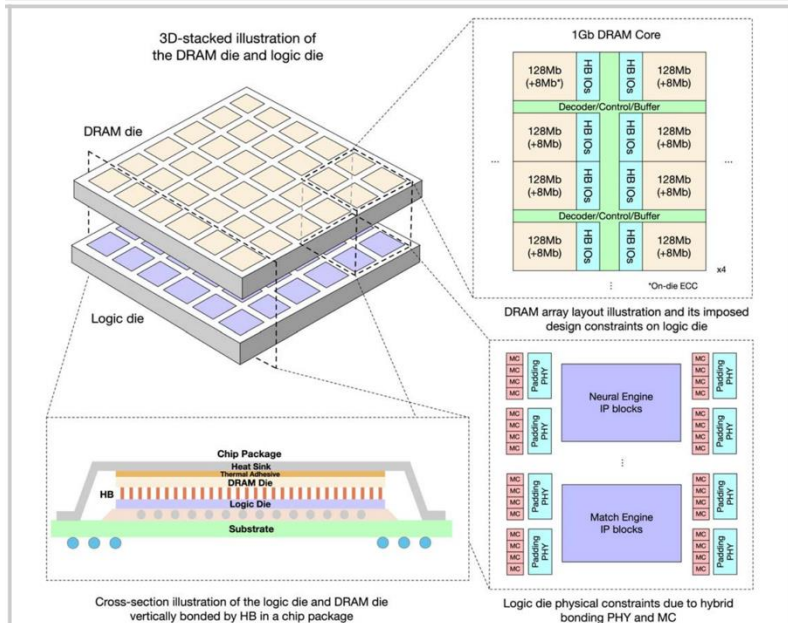
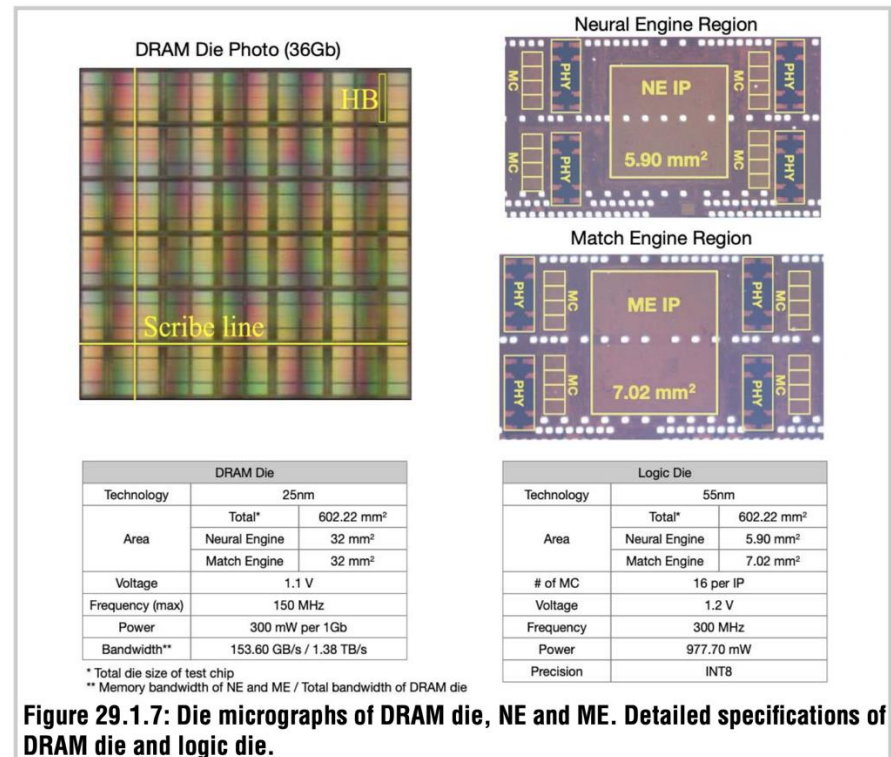


Figure 29.1.2: Illustration of 3D-stacked chip, cross-illustration of package, DRAM array layout and design blocks on logic die.



29.1 184QPS/W 64Mb/mm² 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System

Dimin Niu¹, Shuangchen Li¹, Yuhao Wang¹, Wei Han¹, Zhe Zhang², Yijin Guan², Tianchan Guan³, Fei Sun¹, Fei Xue¹, Lide Duan¹, Yuanwei Fang¹, Hongzhong Zheng¹, Xiping Jiang⁴, Song Wang⁴, Fengguo Zuo⁴, Yubing Wang⁴, Bing Yu⁴, Qiwei Ren⁴, Yuan Xie¹

SK Hynix CXL Processing Near Memory (2023)

IEEE COMPUTER ARCHITECTURE LETTERS, VOL. 22, NO. 1, JANUARY-JUNE

Computational CXL-Memory Solution for Accelerating Memory-Intensive Applications

Joonseop Sim^{ID}, Soohong Ahn^{ID}, Taeyoung Ahn^{ID},
Seungyong Lee^{ID}, Myunghyun Rhee, Jooyoung Kim^{ID},
Kwangsik Shin, Donguk Moon^{ID},
Euseok Kim, and Kyoung Park^{ID}

Abstract—CXL interface is the up-to-date technology that enables effective memory expansion by providing a memory-sharing protocol in configuring heterogeneous devices. However, its limited physical bandwidth can be a significant bottleneck for emerging data-intensive applications. In this work, we propose a novel CXL-based memory disaggregation architecture with a real-world prototype demonstration, which overcomes the bandwidth limitation of the CXL interface using near-data processing. The experimental results demonstrate that our design achieves up to $1.9\times$ better performance/power efficiency than the existing CPU system.

Index Terms—Compute express link (CXL), near-data-processing (NDP)

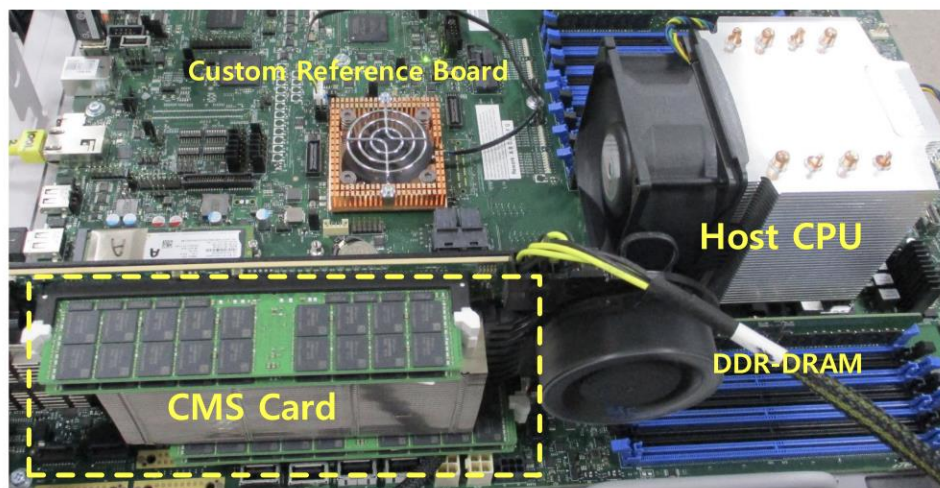


Fig. 6. FPGA prototype of proposed CMS card.

Samsung CXL Processing Near Memory (2023)

Samsung Processing in Memory Technology at Hot Chips 2023

By Patrick Kennedy - August 28, 2023



Samsung PIM PNM For Transformer Based AI HC35_Page_24

Processing-in-Memory in the Real World

DAMOV Analysis Methodology & Workloads

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana-Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

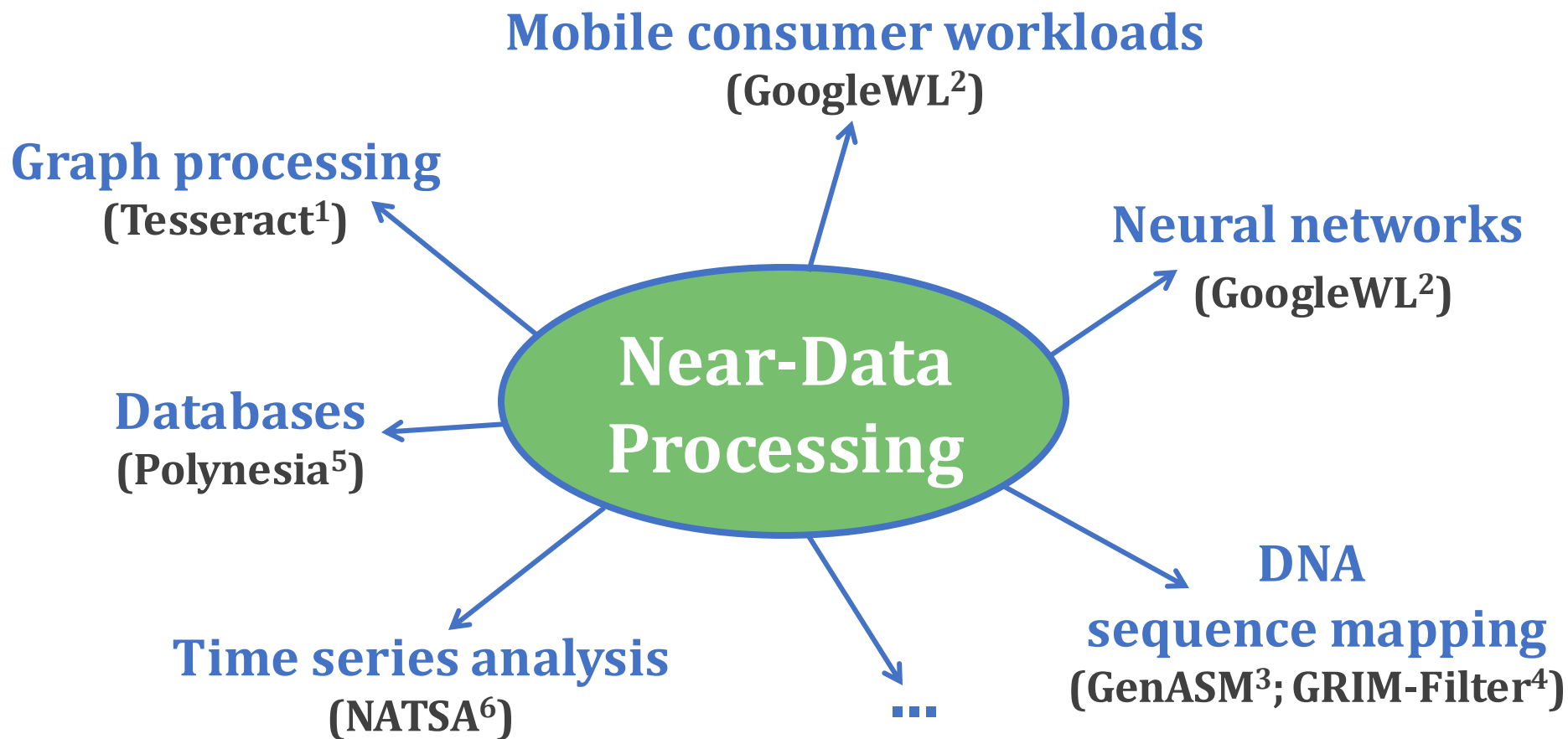
MOHAMMAD SADROSADATI, Institute for Research in Fundamental Sciences (IPM), Iran & ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Data movement between the CPU and main memory is a first-order obstacle against improving performance, scalability, and energy efficiency in modern systems. Computer systems employ a range of techniques to reduce overheads tied to data movement, spanning from traditional mechanisms (e.g., deep multi-level cache hierarchies, aggressive hardware prefetchers) to emerging techniques such as Near-Data Processing (NDP), where some computation is moved close to memory. Prior NDP works investigate the root causes of data movement bottlenecks using different profiling methodologies and tools. However, there is still a lack of understanding about the key metrics that can identify different data movement bottlenecks and their relation to traditional and emerging data movement mitigation mechanisms. Our goal is to methodically identify potential sources of data movement over a broad set of applications and to comprehensively compare traditional compute-centric data movement mitigation techniques (e.g., caching and prefetching) to more memory-centric techniques (e.g., NDP), thereby developing a rigorous understanding of the best techniques to mitigate each source of data movement.

With this goal in mind, we perform the first large-scale characterization of a wide variety of applications, across a wide range of application domains, to identify fundamental program properties that lead to data movement to/from main memory. We develop the first systematic methodology to classify applications based on the sources contributing to data movement bottlenecks. From our large-scale characterization of 77K functions across 345 applications, we select 144 functions to form the first open-source benchmark suite (DAMOV) for main memory data movement studies. We select a diverse range of functions that (1) represent different types of data movement bottlenecks, and (2) come from a wide range of application domains. Using NDP as a case study, we identify new insights about the different data movement bottlenecks and use these insights to determine the most suitable data movement mitigation mechanism for a particular application. We open-source DAMOV and the complete source code for our new characterization methodology at <https://github.com/CMU-SAFARI/DAMOV>.

When to Employ Near-Data Processing?



[1] Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing," ISCA, 2015

[2] Boroumand+, "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS, 2018

[3] Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis," MICRO, 2020

[4] Kim+, "GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies," BMC Genomics, 2018

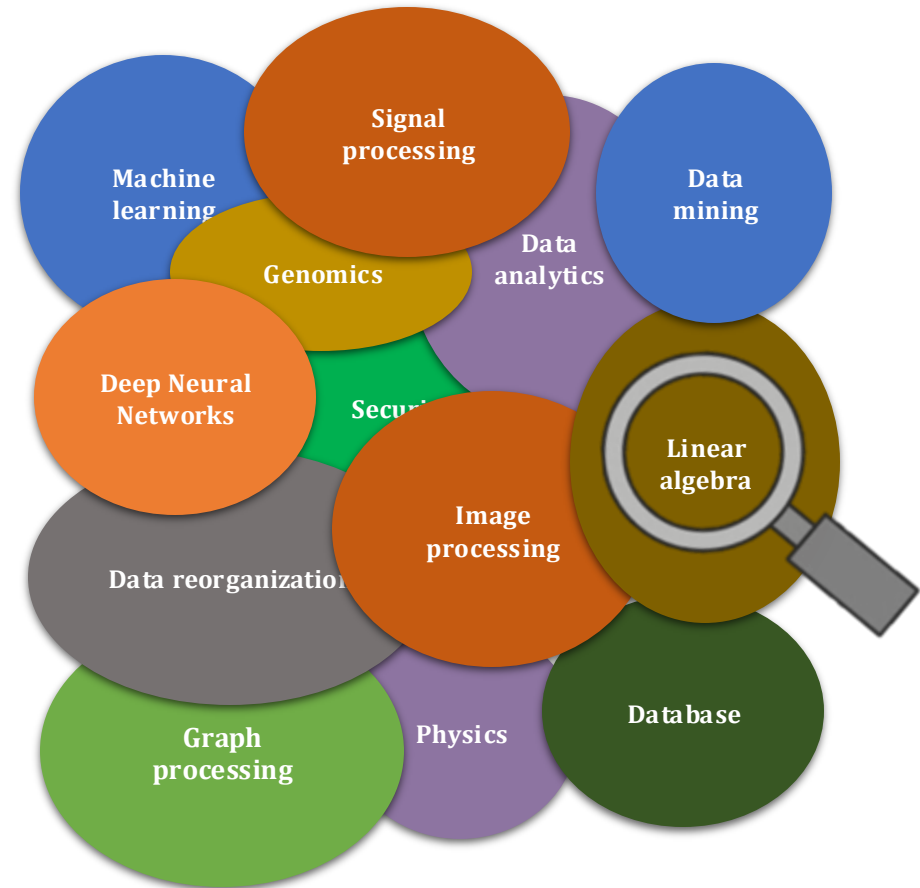
[5] Boroumand+, "Polynesia: Enabling Effective Hybrid Transactional/Analytical Databases with Specialized Hardware/Software Co-Design," arXiv:2103.00798 [cs.AR], 2021

[6] Fernandez+, "NATSA: A Near-Data Processing Accelerator for Time Series Analysis," ICCD, 2020

Step 1: Application Profiling

- We analyze 345 applications from distinct domains:

- Graph Processing
- Deep Neural Networks
- Physics
- High-Performance Computing
- Genomics
- Machine Learning
- Databases
- Data Reorganization
- Image Processing
- Map-Reduce
- Benchmarking
- Linear Algebra



Step 3: Memory Bottleneck Analysis

**Six classes of
data movement bottlenecks:**

each class \leftrightarrow data movement
mitigation mechanism

Memory Bottleneck Class

1a: *DRAM
Bandwidth*

1b: *DRAM Latency*

1c: *L1/L2
Cache Capacity*

2a: *L3 Cache
Contention*

2b: *L1 Cache
Capacity*

2c: *Compute-Bound*

DAMOV is Open Source

- We open-source our **benchmark suite** and our **toolchain**

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing. Described by Oliveira et al. (preliminary version at <https://arxiv.org/pdf/2105.03725.pdf>)

Readme

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages



omutlu Update README.md

ce1b4ea 17 days ago 5 commits

simulator	Cleaning	19 days ago
README.md	Update README.md	17 days ago
get_workloads.sh	DAMOV -- first commit	19 days ago

README.md

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including [BWA](#), [Chai](#), [Darknet](#), [GASE](#), [Hardware Effects](#), [Hashjoin](#), [HPCC](#), [HPCG](#), [Ligra](#), [PARSEC](#), [Parboil](#), [PolyBench](#), [Phoenix](#), [Rodinia](#), [SPLASH-2](#), [STREAM](#).

DAMOV-SIM

DAMOV
Benchmarks

SAFARI

DAMOV is Open Source

- We open-source our [benchmark suite](#) and our [toolchain](#)

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About

DAMOV is a benchmark suite and a

Get DAMOV at:

<https://github.com/CMU-SAFARI/DAMOV>

README.md

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including [BWA](#), [Chai](#), [Darknet](#), [GASE](#), [Hardware Effects](#), [Hashjoin](#), [HPCC](#), [HPCG](#), [Ligra](#), [PARSEC](#), [Parboil](#), [PolyBench](#), [Phoenix](#), [Rodinia](#), [SPLASH-2](#), [STREAM](#).

Readme

Releases

No releases published
[Create a new release](#)

Packages

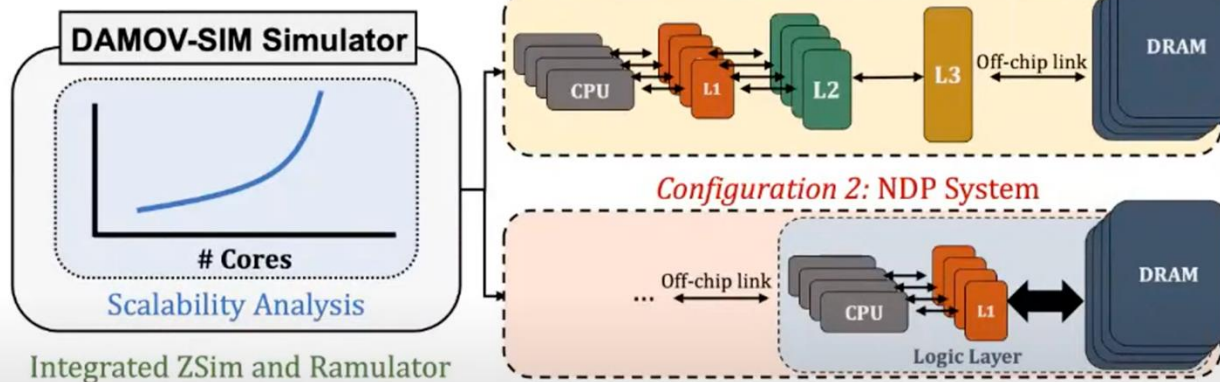
No packages published
[Publish your first package](#)

Languages

More on DAMOV Analysis Methodology & Workloads

Step 3: Memory Bottleneck Classification (2/)

- **Goal:** identify the specific sources of data movement bottlenecks



- **Scalability Analysis:**
 - 1, 4, 16, 64, and 256 out-of-order/in-order host and NDP CPU cores
 - 3D-stacked memory as main memory

DAMOV-SIM: <https://github.com/CMU-SAFARI/DAMOV>

SAFARI Live Seminar: DAMOV: A New Methodology & Benchmark Suite for Data Movement Bottlenecks

352 views • Streamed live on Jul 22, 2021

18 0 SHARE SAVE ...



Onur Mutlu Lectures
17.7K subscribers

ANALYTICS

EDIT VIDEO

https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=3

More on DAMOV Methods & Benchmarks

- Geraldo F. Oliveira, Juan Gomez-Luna, Lois Orosa, Saugata Ghose, Nandita Vijaykumar, Ivan fernandez, Mohammad Sadrosadati, and Onur Mutlu,
"DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks"
IEEE Access, 8 September 2021.
Preprint in arXiv, 8 May 2021.
[[arXiv preprint](#)]
[[IEEE Access version](#)]
[[DAMOV Suite and Simulator Source Code](#)]
[[SAFARI Live Seminar Video](#) (2 hrs 40 mins)]
[[Short Talk Video](#) (21 minutes)]

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

MOHAMMAD SADROSADATI, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Ramulator 2.0 for PIM Systems

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
"Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
*Preprint on **arxiv**, August 2023.*
[\[arXiv version\]](#)
[\[Ramulator 2.0 Source Code\]](#)

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

<https://arxiv.org/pdf/2308.11030.pdf>

We Need to Exploit Good Principles

- Data-centric system design
- All components intelligent
- Better (cross-layer) communication, better interfaces
- Better-than-worst-case design
- Heterogeneity
- Flexibility, adaptability

Open minds

A Blueprint for Fundamentally Better Architectures

- Onur Mutlu,
"Intelligent Architectures for Intelligent Computing Systems"
*Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (**DATE**), Virtual, February 2021.*
[[Slides \(pptx\)](#)] [[pdf](#)]
[[IEDM Tutorial Slides \(pptx\)](#)] [[pdf](#)]
[[Short DATE Talk Video](#) (11 minutes)]
[[Longer IEDM Tutorial Video](#) (1 hr 51 minutes)]

Intelligent Architectures for Intelligent Computing Systems

Onur Mutlu
ETH Zurich
omutlu@gmail.com

More Resources

Five Key Issues in Future Platforms

- Fundamentally Robust (Secure/Reliable/Safe) Architectures
- Fundamentally Energy-Efficient & High-Performance Systems
 - Memory-centric (Data-centric) Architectures
- Fundamentally Low-Latency and Predictable Architectures
- Fundamentally Intelligent and Evolving Architectures
 - ML/AI-Assisted (Data-driven) and Data-aware Architectures
- Architectures for ML/AI, Genomics, Medicine, Health, ...

Special Research Sessions & Courses

- Special Session at ISVLSI 2022: 9 cutting-edge talks



The image shows a YouTube video player interface. The video title is "In-Memory Processing ISVLSI 2022 Special Session". Below the title, it says "IEEE Computer Society Annual Symposium on VLSI". The video is from the "ISVLSI 2022" series, held at "Adonis room, Ailathon resort, Paphos, Cyprus" on "July 4th, 2022". The video player shows a progress bar at 0:04 / 3:36:35. The video is by "Onur Mutlu Lectures" and has 1,286 views. The video is titled "Introduction to the ISVLSI 2022 Special Session on Processing-in-Memory". The video player includes standard YouTube controls like play, pause, volume, and full screen. The video is part of a series by "Onur Mutlu Lectures" who has 26.9K subscribers. The video player also shows engagement metrics like 61 likes and options to dislike, share, download, clip, save, and analytics.

In-Memory Processing
ISVLSI 2022 Special Session

IEEE Computer Society Annual Symposium on VLSI

ISVLSI 2022

Adonis room
Ailathon resort, Paphos, Cyprus
July 4th, 2022

0:04 / 3:36:35 • Dr. Juan Gómez-Luna, "Introduction to the ISVLSI 2022 Special Session on Processing-in-Memory" >

ISVLSI 2022 Special Session on Processing-in-Memory

1,286 views • Premiered Aug 9, 2022

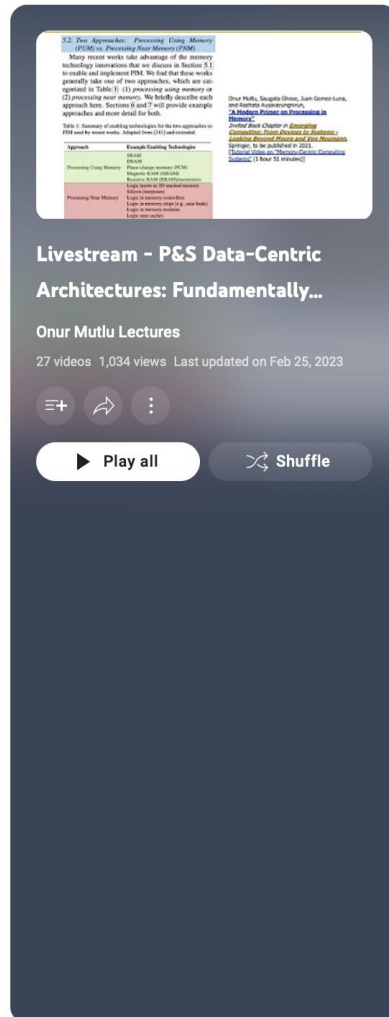
61 DISLIKE SHARE DOWNLOAD CLIP SAVE ...




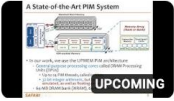
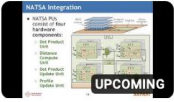




Onur Mutlu Lectures
26.9K subscribers

ANALYTICS EDIT VIDEO

Special Research Sessions & Courses (II)

■ Special Session at ISVLSI 2022: 9 cutting-edge talks



- 19  **GenStore: In-Storage Filtering for High-Performance and Energy-Efficient Genome Analysis**
Onur Mutlu Lectures • Premieres 3/12/23, 7:00 PM
- 20  **Introduction to the ISVLSI 2022 Special Session on Processing-in-Memory**
Onur Mutlu Lectures • 286 views • 2 days ago
- 21  **Heterogeneous Data-Centric Architectures for Data-Intensive Applications: Case Studies in ML and DB**
Onur Mutlu Lectures • 2 waiting • Premieres 3/10/23, 7:00 PM
- 22  **Machine Learning Training on a Real Processing-In-Memory System**
Onur Mutlu Lectures • Premieres 3/14/23, 7:00 PM
- 23  **Exploiting Near-Data Processing to Accelerate Time Series Analysis**
Onur Mutlu Lectures • Premieres 3/11/23, 7:00 PM
- 24  **PiDRAM: An FPGA-Based Framework for End-To-End Evaluation of Processing-In-DRAM Techniques**
Onur Mutlu Lectures • Premieres 3/9/23, 7:00 PM
- 25  **The Road to Widely Deploying Processing-In-Memory: Challenges and Opportunities**
Onur Mutlu Lectures • 399 views • 1 day ago
- 26  **SparseP: Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures**
Onur Mutlu Lectures • 1 waiting • Premieres 3/13/23, 7:00 PM
- 27  **HPCA 2023 Tutorial: Real-World Processing-in-Memory Architectures**
Onur Mutlu Lectures • 1.6K views • Streamed 10 days ago

Comp Arch (Fall 2021)

Fall 2021 Edition:

- https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule

Fall 2020 Edition:

- https://safari.ethz.ch/architecture/fall2020/doku.php?id=schedule

Youtube Livestream (2021):

- https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF


Youtube Livestream (2020):

- https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN

Master's level course

- Taken by Bachelor's/Masters/PhD students
- Cutting-edge research topics + fundamentals in Computer Architecture
- 5 Simulator-based Lab Assignments
- Potential research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>



Computer Architecture - Fall 2021

Recent Changes Media Manager Sitemap

Trace: readings start schedule

Home

Announcements

Materials

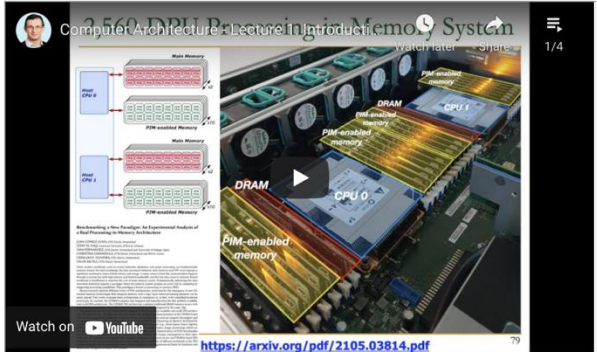
- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

Resources

- Computer Architecture FS20: Course Webpage
- Computer Architecture FS20: Lecture Videos
- Digitaltechnik SS21: Course Webpage
- Digitaltechnik SS21: Lecture Videos
- Moodle
- HotCRP
- Verilog Practice Website (HDLBits)

Lecture Video Playlist on YouTube

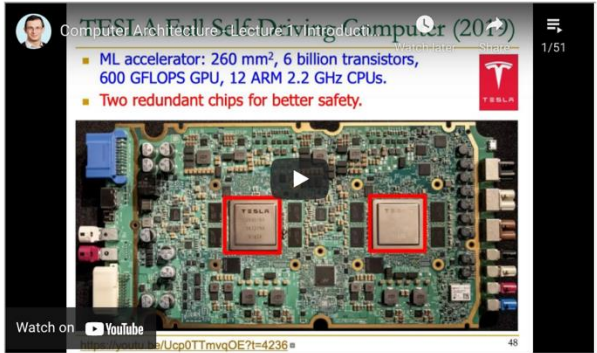
Livestream Lecture Playlist



Watch on YouTube

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

Recorded Lecture Playlist



Watch on YouTube

Fall 2021 Lectures & Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	30.09 Thu.	Yes	L1: Introduction and Basics ααα (PDF) ααα (PPT)	Required Mentioned	Lab 1 Out	HW 0 Out
	01.10 Fri.	Yes	L2: Trends, Tradeoffs and Design Fundamentals ααα (PDF) ααα (PPT)	Required Mentioned		
W2	07.10 Thu.	Yes	L3a: Memory Systems: Challenges and Opportunities ααα (PDF) ααα (PPT)	Described Suggested		HW 1 Out
			L3b: Course Info & Logistics ααα (PDF) ααα (PPT)			
			L3c: Memory Performance Attacks ααα (PDF) ααα (PPT)	Described Suggested		
	08.10 Fri.	Yes	L4a: Memory Performance Attacks ααα (PDF) ααα (PPT)	Described Suggested	Lab 2 Out	
			L4b: Data Retention and Memory Refresh ααα (PDF) ααα (PPT)	Described Suggested		
			L4c: RowHammer ααα (PDF) ααα (PPT)	Described Suggested		

DDCA (Spring 2022)

Spring 2022 Edition:

- <https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule>

Spring 2021 Edition:

- <https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=schedule>

Youtube Livestream (Spring 2022):

- <https://www.youtube.com/watch?v=cpXdE3HwvK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6>


Youtube Livestream (Spring 2021):

- https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN

Bachelor's course

- 2nd semester at ETH Zurich
- Rigorous introduction into "How Computers Work"
- Digital Design/Logic
- Computer Architecture
- 10 FPGA Lab Assignments

<https://www.youtube.com/onurmutlulectures>



Digital Design and Computer Architecture - Spring 2021

Trace: [schedule](#)

Home

Announcements

Materials


- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Optional HWs
- Labs
- Extra Assignments
- Exams
- Technical Docs

Resources

- Computer Architecture (CMU) SS15: Lecture Videos
- Computer Architecture (CMU) SS15: Course Website
- Digitaltechnik SS18: Lecture Videos
- Digitaltechnik SS18: Course Website
- Digitaltechnik SS19: Lecture Videos
- Digitaltechnik SS19: Course Website
- Digitaltechnik SS20: Lecture Videos
- Digitaltechnik SS20: Course Website
- Moodle

Lecture Video Playlist on YouTube

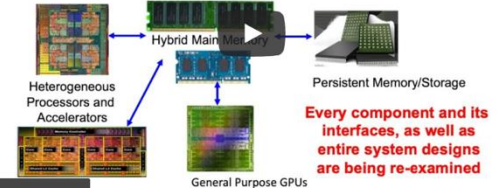
Livestream Lecture Playlist




Onur Mutlu: Digital Design and Computer Architecture Today


Computing landscape is very different from 10-20 years ago

Applications and technology both demand novel architectures




Watch on  YouTube

Recorded Lecture Playlist






Digital Design & Computer Architecture: Lect...

How Computers Work (from the ground up)

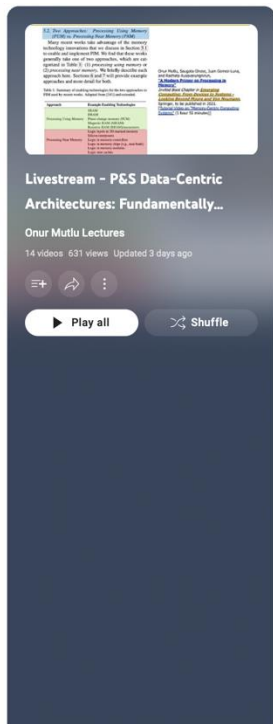
Watch on  YouTube


Spring 2021 Lectures/Schedule


Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	25.02 Thu.	 Live	L1: Introduction and Basics Q2A (PDF) Q2B (PPT)	Required Suggested Mentioned		
			L2a: Tradeoffs, Metrics, Mindset Q2A (PDF) Q2B (PPT)	Required		
	26.02 Fri.	 Live	L2b: Mysteries in Computer Architecture Q2A (PDF) Q2B (PPT)	Required Mentioned		
W2	04.03 Thu.	 Live	L3a: Mysteries in Computer Architecture II Q2A (PDF) Q2B (PPT)	Required Suggested Mentioned		


Processing-in-Memory Course (Fall 2022)

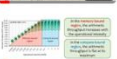
- Short weekly lectures
- Hands-on projects





1  **PIM Course: Lecture 1: Data-Centric Architectures: Improving Performance & Energy - Fall 2022**
Onur Mutlu Lectures • 1K views • 3 months ago

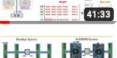
2  **PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks - Fall 2022**
Onur Mutlu Lectures • 678 views • 2 months ago


3  **PIM Course: Lecture 3: Real-world PIM: UPMEM PIM - Fall 2022**
Onur Mutlu Lectures • 455 views • 2 months ago


4  **PIM Course: Lecture 4: Real-world PIM: Microbenchmarking of UPMEM PIM - Fall 2022**
Onur Mutlu Lectures • 275 views • 2 months ago


5  **PIM Course: Lecture 5: Real-world PIM: Samsung HBM-PIM - Fall 2022**
Onur Mutlu Lectures • 725 views • 2 months ago

6  **PIM Course: Lecture 6: Real-world PIM: SK Hynix AiM - Fall 2022**
Onur Mutlu Lectures • 1K views • 2 months ago

7  **PIM Course: Lecture 7: Real-world PIM: Samsung AxDIMM - Fall 2022**
Onur Mutlu Lectures • 767 views • 1 month ago

8  **PIM Course: Lecture 8: Real-world PIM: Alibaba HB-PNM - Fall 2022**
Onur Mutlu Lectures • 383 views • 1 month ago

9  **PIM Course: Lecture 9: Programming PIM Architectures - Fall 2022**
Onur Mutlu Lectures • 367 views • 1 month ago

 **SAFARI Project & Seminars Courses (Fall 2022)**

Trace: • heterogeneous_systems • processing_in_memory

Home

Courses

- SoftMC
- Ramulator
- Accelerating Genomics
- Mobile Genomics
- **Processing-in-Memory**
- Heterogeneous Systems
- Modern SSDs

processing_in_memory

Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)

Course Description

Data movement between the memory units and the compute units of current computing systems is a major performance and energy bottleneck. From large-scale servers to mobile devices, data movement costs dominate computation costs in terms of both performance and energy consumption. For example, data movement between the main memory and the processing cores accounts for 62% of the total system energy in consumer applications. As a result, the data movement bottleneck is a huge burden that greatly limits the energy efficiency and performance of modern computing systems. This phenomenon is an undesired effect of the dichotomy between memory and the processor, which leads to the data movement bottleneck.

Many modern and important workloads such as machine learning, computational biology, graph processing, databases, video analytics, and real-time data analytics suffer greatly from the data movement bottleneck. These workloads are exemplified by irregular memory accesses, relatively low data reuse, low cache line utilization, low arithmetic intensity (i.e., ratio of operations per accessed byte), and large datasets that greatly exceed the main memory size. The computation in these workloads cannot usually compensate for the data movement costs. In order to alleviate this data movement bottleneck, we need a paradigm shift from the traditional processor-centric design, where all computation takes place in the compute units, to a more data-centric design where processing elements are placed closer to or inside where the data resides. This paradigm of computing is known as Processing-in-Memory (PIM).

This is your perfect P&S if you want to become familiar with the main PIM technologies, which represent "the next big thing" in Computer Architecture. You will work hands-on with the first real-world PIM architecture, will explore different PIM architecture designs for important workloads, and will develop tools to enable research of future PIM systems. Projects in this course span software and hardware as well as the software/hardware interface. You can potentially work on developing and optimizing new workloads for the first real-world PIM hardware or explore new PIM designs in simulators, or do something else that can forward our understanding of the PIM paradigm.

Table of Contents

- Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)
- Course Description
- Mentors
- Lecture Video Playlist on YouTube
- Spring 2022 Meetings/Schedule
- Past Lecture Video Playlists on YouTube
- Learning Materials
- Assignments

https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory

<https://youtube.com/playlist?list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0GOBrnKy>

PIM Course (Fall 2022)

■ Fall 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory

■ Spring 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory

■ Youtube Livestream (Fall 2022):

- <https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0GOBrnKy>

■ Youtube Livestream (Spring 2022):

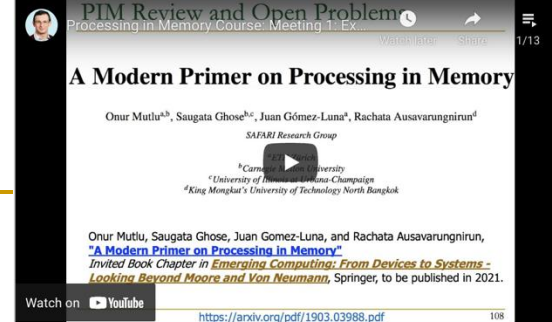
- <https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX>

■ Project course

- Taken by Bachelor's/Master's students
- Processing-in-Memory lectures
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>

SAFARI

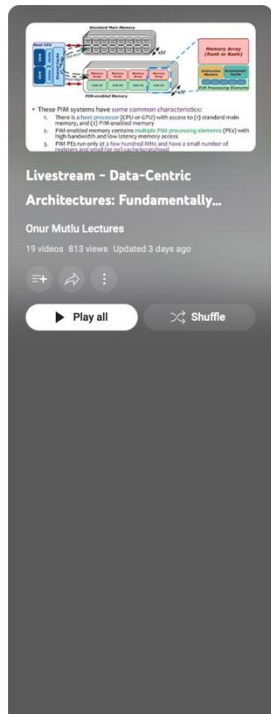


Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	10.03 Thu.	YouTube Live	M1: P&S PIM Course Presentation (PDF) (PPT)	Required Materials Recommended Materials	HW 0 Out
W2	15.03 Tue.		Hands-on Project Proposals		
	17.03 Thu.	YouTube Premiere	M2: Real-world PIM: UPMEM PIM (PDF) (PPT)		
W3	24.03 Thu.	YouTube Live	M3: Real-world PIM: Microbenchmarking of UPMEM PIM (PDF) (PPT)		
W4	31.03 Thu.	YouTube Live	M4: Real-world PIM: Samsung HBM-PIM (PDF) (PPT)		
W5	07.04 Thu.	YouTube Live	M5: How to Evaluate Data Movement Bottlenecks (PDF) (PPT)		
W6	14.04 Thu.	YouTube Live	M6: Real-world PIM: SK Hynix AIM (PDF) (PPT)		
W7	21.04 Thu.	YouTube Premiere	M7: Programming PIM Architectures (PDF) (PPT)		
W8	28.04 Thu.	YouTube Premiere	M8: Benchmarking and Workload Suitability on PIM (PDF) (PPT)		
W9	05.05 Thu.	YouTube Premiere	M9: Real-world PIM: Samsung AxDIMM (PDF) (PPT)		
W10	12.05 Thu.	YouTube Premiere	M10: Real-world PIM: Alibaba HB-PNM (PDF) (PPT)		
W11	19.05 Thu.	YouTube Live	M11: SpMV on a Real PIM Architecture (PDF) (PPT)		
W12	26.05 Thu.	YouTube Live	M12: End-to-End Framework for Processing-using-Memory (PDF) (PPT)		
W13	02.06 Thu.	YouTube Live	M13: Bit-Serial SIMD Processing using DRAM (PDF) (PPT)		
W14	09.06 Thu.	YouTube Live	M14: Analyzing and Mitigating ML Inference Bottlenecks (PDF) (PPT)		
W15	15.06 Thu.	YouTube Live	M15: In-Memory HTAP Databases with HW/SW Co-design (PDF) (PPT)		
W16	23.06 Thu.	YouTube Live	M16: In-Storage Processing for Genome Analysis (PDF) (PPT)		
W17	18.07 Mon.	YouTube Premiere	M17: How to Enable the Adoption of PIM? (PDF) (PPT)		
W18	09.08 Tue.	YouTube Premiere	SS1: ISVLSI 2022 Special Session on PIM (PDF & PPT)		


Processing-in-Memory Course (Spring 2023)

- Short weekly lectures
- Hands-on projects



- PIM Course: Lecture 1: Data-Centric Architectures: Improving Performance & Energy (Spring 2023)**
Onur Mutlu Lectures • 1.1K views • Streamed 3 months ago
1:14:16
- PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks (Spring 2023)**
Onur Mutlu Lectures • 332 views • 2 months ago
16:37
- ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads**
Onur Mutlu Lectures • 1.5K views • Streamed 2 months ago
6:27:39
- PIM Course: Lecture 3: Real-world PIM: UPMEM PIM (Spring 2023)**
Onur Mutlu Lectures • 411 views • 2 months ago
15:43
- PIM Course: Lecture 4: Real-world PIM: Microbenchmarking of UPMEM PIM (Spring 2023)**
Onur Mutlu Lectures • 188 views • 2 months ago
24:10
- Análisis Experimental de una Arquitectura PIM - Juan Gómez Luna - Lecture in Spanish @ U. de Córdoba**
Onur Mutlu Lectures • 169 views • 2 months ago
2:27:12
- PIM Course: Lecture 5: Real-world PIM: Samsung HBM-PIM (Spring 2023)**
Onur Mutlu Lectures • 483 views • 2 months ago
24:08
- PIM Course: Lecture 6: Real-world PIM: SK Hynix AIM (Spring 2023)**
Onur Mutlu Lectures • 573 views • 1 month ago
35:50
- PIM Course: Lecture 7: Real-world PIM: Samsung AxDIMM (Spring 2023)**
Onur Mutlu Lectures • 325 views • 1 month ago
21:32

https://www.youtube.com/playlist?list=PL5Q2soXY2zi_F0buoAZVSq_o6UySWQHvZ

**SAFARI Project & Seminars Courses**
(Spring 2023)

Search

Recent Changes Media Manager Sitemap

Trace: • heterogeneous_systems • processing_in_memory

Home

Courses

- SoftMC
- Ramulator
- Accelerating Genomics
- Mobile Genomics
- Processing-in-Memory
- Heterogeneous Systems
- Modern SSDs
- Hardware/Software Co-design

processing_in_memory

Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)

Course Description

Data movement between the memory units and the compute units of current computing systems is a major performance and energy bottleneck. From large-scale servers to mobile devices, data movement costs dominate computation costs in terms of both performance and energy consumption. For example, data movement between the main memory and the processing cores accounts for 62% of the total system energy in consumer applications. As a result, the data movement bottleneck limits the energy efficiency and performance of modern computing systems. This phenomenon is an undesired effect of the dichotomy between memory and the processor, which leads to the data movement bottleneck.

Many modern and important workloads such as machine learning, computational biology, graph processing, databases, video analytics, and real-time data analytics suffer greatly from the data movement bottleneck. These workloads are exemplified by irregular memory accesses, relatively low data reuse, low cache line utilization, low arithmetic intensity (i.e., ratio of operations per accessed byte), and large datasets that greatly exceed the main memory size. The computation in these workloads cannot usually compensate for the data movement costs. In order to alleviate this data movement bottleneck, we need a paradigm shift from the traditional processor-centric design, where all computation takes place in the compute units, to a more data-centric design where processing elements are placed closer to or inside where the data resides. This paradigm of computing is known as Processing-in-Memory (PIM).

This is your perfect P&S if you want to become familiar with the main PIM technologies, which represent "the next big thing" in Computer Architecture. You will work hands-on with the first real-world PIM architecture, will explore different PIM architecture designs for important workloads, and will develop tools to enable research of future PIM systems. Projects in this course span software and hardware as well as the software/hardware interface. You can potentially work on developing and optimizing new workloads for the first real-world PIM hardware or explore new PIM designs in simulators, or do something else that can forward our understanding of the PIM paradigm.

Table of Contents

- Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)
- Course Description
- Mentors
- Lecture Video Playlist on YouTube
- Spring 2023 Meetings/Schedule
- Past Lecture Video Playlists on YouTube
- Learning Materials
- Assignments


Prerequisites of the course:

- Digital Design and Computer Architecture (or equivalent course).
- Familiarity with C/C++ programming.
- Interest in future computer architectures and computing paradigms.
- Interest in discovering why things do or do not work and solving problems
- Interest in making systems efficient and usable

https://safari.ethz.ch/projects_and_seminars/spring2023/doku.php?id=processing_in_memory

Real PIM Tutorials [MICRO'23, ISCA'23, ASPLOS'23, HPCA'23]

- June, March, Feb : Lectures + Hands-on labs + Invited talks



ISCA 2023 Real-World PIM Tutorial

Search

[Recent Changes](#) [Media Manager](#) [Sitemap](#)

Trace: • [start](#)

Real-world Processing-in-Memory Systems for Modern Workloads

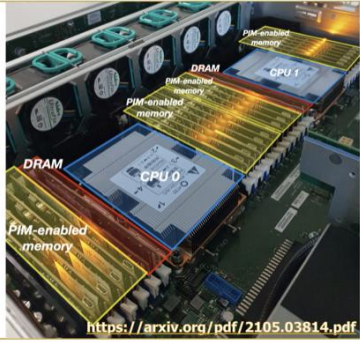
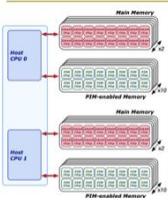
Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System



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

Table of Contents

- [Real-world Processing-in-Memory Systems for Modern Workloads](#)
- [Tutorial Description](#)
- [Organizers](#)
- [Agenda \(June 18, 2023\)](#)
- [Lectures \(tentative\)](#)
- [Hands-on Labs \(tentative\)](#)
- [Learning Materials](#)

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

<https://events.safari.ethz.ch/isca-pim-tutorial/>

Real PIM Tutorial [ISCA 2023]

■ June 18: Lectures + Hands-on labs + Invited talks

ISCA 2023 Real-World PIM Tutorial Sunday, June 18, Orlando, Florida

Organizers: Juan Gómez Luna, Onur Mutlu, Ataberk Olgun
Program: <https://events.safari.ethz.ch/isca-pim-tutorial/>



Overview PIM | PNM | UPMEM PIM |
PNM for neural networks |
PNM for recommender systems |
PNM for ML workloads |
How to enable PIM? | PUM prototypes
Hands-on Labs: Benchmarking |
Accelerating real-world workloads



International Symposium on Computer Architecture (ISCA)

Real-world Processing-in-Memory Systems for Modern Workloads

<https://www.youtube.com/live/GIb5EgSrWk0?feature=share>

Room: Magnolia 16
Marriott World Center Orlando
Orlando, FL, USA
July 18th, 2023

SAFARI zoom

ISCA 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

Onur Mutlu Lectures
33.9K subscribers

Subscribed

57

Share

Download

Clip

...

1,687 views Streamed live on Jun 18, 2023 Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)

[https://www.youtube.com/
live/GIb5EgSrWk0](https://www.youtube.com/live/GIb5EgSrWk0)

[https://events.safari.ethz.ch/
isca-pim-tutorial/](https://events.safari.ethz.ch/isca-pim-tutorial/)

Tutorial Materials

Time	Speaker	Title	Materials
8:55am-9:00am	Dr. Juan Gómez Luna	Welcome & Agenda	(PDF) (PPT)
9:00am-10:20am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
10:20am-11:00am	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures / Programming General-purpose PIM	(PDF) (PPT)
11:20am-11:50am	Prof. Izzat El Hajj	High-throughput Sequence Alignment using Real Processing-in-Memory Systems	(PDF) (PPT)
11:50am-12:30pm	Dr. Christina Giannoula	SparseP: Towards Efficient Sparse Matrix Vector Multiplication for Real Processing-In-Memory Systems	(PDF) (PPT)
2:00pm-2:45pm	Dr. Sukhan Lee	Introducing Real-world HBM-PIM Powered System for Memory-bound Applications	(PDF) (PPT)
2:45pm-3:30pm	Dr. Juan Gómez Luna / Ataberk Olgun	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components / PUM Prototypes: PiDRAM	(PDF) (PPT) (PDF) (PPT)
4:00pm-4:40pm	Dr. Juan Gómez Luna	Accelerating Modern Workloads on a General-purpose PIM System	(PDF) (PPT)
4:40pm-5:20pm	Dr. Juan Gómez Luna	Adoption Issues: How to Enable PIM?	(PDF) (PPT)
5:20pm-5:30pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)

Real PIM Tutorial [ASPLOS 2023]

■ March 26: Lectures + Hands-on labs + Invited talks

Tutorial Materials

Time	Speaker	Title	Materials
9:00am-10:20am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
10:40am-12:00pm	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM	(PDF) (PPT)
1:40pm-2:20pm	Prof. Alexandra (Sasha) Fedorova (UBC)	Processing in Memory in the Wild	(PDF) (PPT)
2:20pm-3:20pm	Dr. Juan Gómez Luna & Ataberk Olgun	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components	(PDF) (PPT) (PDF) (PPT)
3:40pm-4:10pm	Dr. Juan Gómez Luna	Adoption issues: How to enable PIM? Accelerating Modern Workloads on a General-purpose PIM System	(PDF) (PPT) (PDF) (PPT)
4:10pm-4:50pm	Dr. Yongkee Kwon & Eddy (Chanwook) Park (SK Hynix)	System Architecture and Software Stack for GDDR6-AiM	(PDF) (PPT)
4:50pm-5:00pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)

ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

Onur Mutlu Lectures
32.1K subscribers

Subscribed

33

Share

Clip

Save

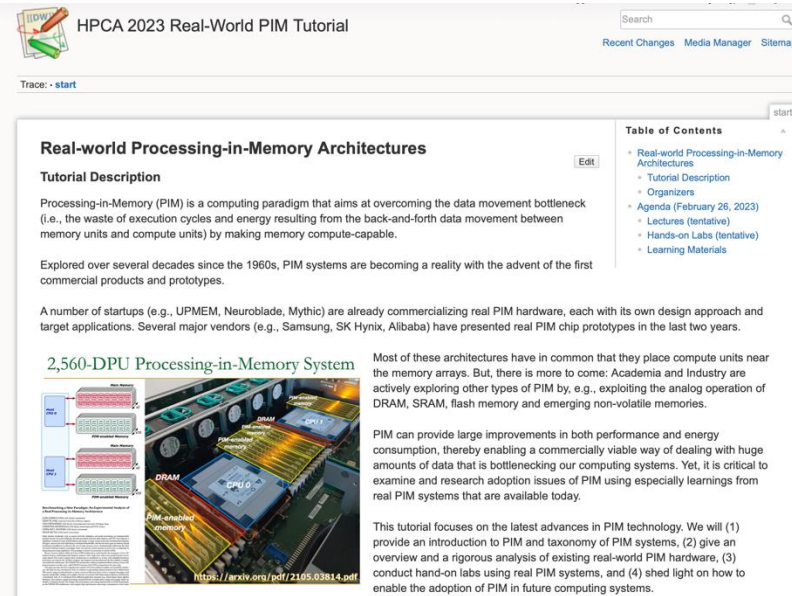
Views Streamed 7 days ago Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)
LOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads
[s://events.safari.ethz.ch/asplos-...](#)

<https://www.youtube.com/watch?v=oYCaLcT0Kmo>

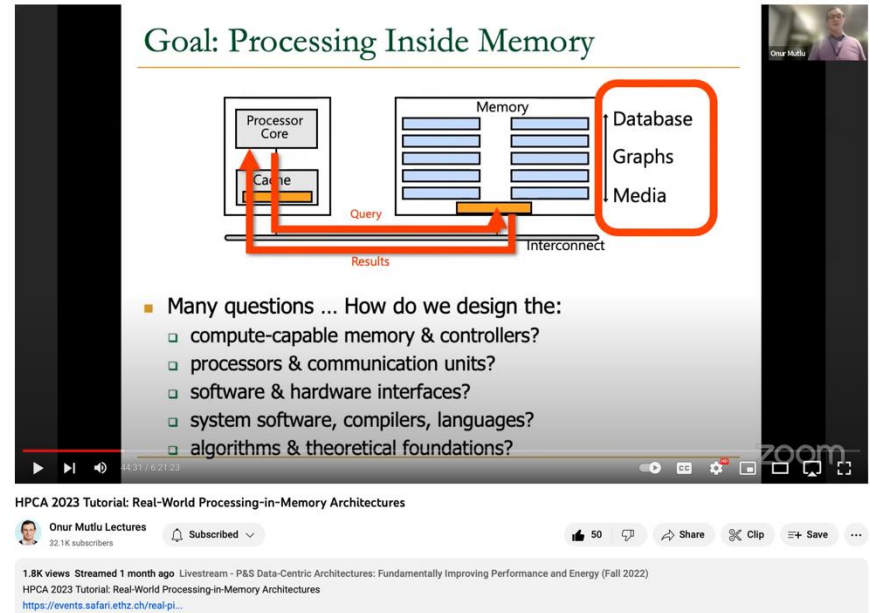
<https://events.safari.ethz.ch/asplos-pim-tutorial/>

Real PIM Tutorial [HPCA 2023]

February 26: Lectures + Hands-on labs + Invited Talks



Time	Speaker	Title	Materials
8:00am-8:40am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
8:40am-10:00am	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM	(PDF) (PPT)
10:20am-11:00am	Dr. Dimin Niu	A 3D Logic-to-DRAM Hybrid Bonding Process-Near-Memory Chip for Recommendation System	
11:00am-11:40am	Dr. Christina Giannoula	SparseP: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures	(PDF) (PPT)
1:30pm-2:10pm	Dr. Juan Gómez Luna	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components	(PDF) (PPT)
2:10pm-2:50pm	Dr. Manuel Le Gallo	Deep Learning Inference Using Computational Phase-Change Memory	
2:50pm-3:30pm	Dr. Juan Gómez Luna	PIM Adoption Issues: How to Enable PIM Adoption?	(PDF) (PPT)
3:40pm-5:40pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)



<https://www.youtube.com/watch?v=f5-nT1tbz5w>

<https://events.safari.ethz.ch/real-pim-tutorial/>

Latest Real PIM Tutorial [MICRO 2023]

■ October 29: Lectures + Hands-on labs + Invited talks

MICRO 2023 Real-World PIM Tutorial

Search

Recent Changes Media Manager Sitemap

Trace: - start

Real-world Processing-in-Memory Systems for Modern Workloads

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

Table of Contents

- Real-world Processing-in-Memory Systems for Modern Workloads
- Tutorial Description
- Livestream
- Organizers
- Agenda (Tentative, October 29, 2023)
- Lectures
- Learning Materials

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

2,560-DPU Processing-in-Memory System

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

MICRO 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

Onur Mutlu Lectures

34.6K subscribers

Live in 74 days
October 29 at 6:00 PM

5 5

Share Save

<https://www.youtube.com/watch?v=ohUooNSIxOI>

<https://events.safari.ethz.ch/micro-pim-tutorial>

Agenda (Tentative, October 29, 2023)

Lectures

1. Introduction: PIM as a paradigm to overcome the data movement bottleneck.
2. PIM taxonomy: PNM (processing near memory) and PUM (processing using memory).
3. General-purpose PNM: UPMEM PIM.
4. PNM for neural networks: Samsung HBM-PIM, SK Hynix AiM.
5. PNM for recommender systems: Samsung AxDIMM, Alibaba PNM.
6. PUM prototypes: PiDRAM, SRAM-based PUM, Flash-based PUM.
7. Other approaches: Neuroblade, Mythic.
8. Adoption issues: How to enable PIM?
9. Hands-on labs: Programming a real PIM system.

SSD Course (Spring 2023)

Spring 2023 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2023/doku.php?id=modern_ssd

Fall 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=modern_ssd

Youtube Livestream (Spring 2023):

- https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2Shtm4z57&pp=iAQB

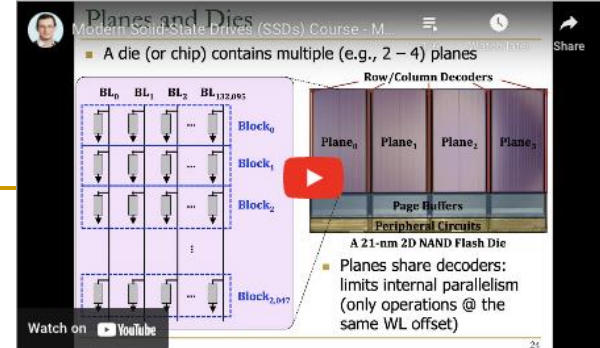
Youtube Livestream (Fall 2022):

- <https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&pp=iAQB>

Project course

- Taken by Bachelor's/Master's students
- SSD Basics and Advanced Topics
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>



Fall 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	06.10		M1: P&S Course Presentation PDF PPT	Required Recommended	
W2	12.10	YouTube Live	M2: Basics of NAND Flash-Based SSDs PDF PPT	Required Recommended	
W3	19.10	YouTube Live	M3: NAND Flash Read/Write Operations PDF PPT	Required Recommended	
W4	26.10	YouTube Live	M4: Processing inside NAND Flash PDF PPT	Required Recommended	
W5	02.11	YouTube Live	M5: Advanced NAND Flash Commands & Mapping PDF PPT	Required Recommended	
W6	09.11	YouTube Live	M6: Processing inside Storage PDF PPT	Required Recommended	
W7	23.11	YouTube Live	M7: Address Mapping & Garbage Collection PDF PPT	Required Recommended	
W8	30.11	YouTube Live	M8: Introduction to MQSim PDF PPT	Required Recommended	
W9	14.12	YouTube Live	M9: Fine-Grained Mapping and Multi-Plane Operation-Aware Block Management PDF PPT	Required Recommended	
W10	04.01.2023	YouTube Premiere	M10a: NAND Flash Basics PDF PPT	Required Recommended	
			M10b: Reducing Solid-State Drive Read Latency by Optimizing Read-Retry PDF PPT Paper	Required Recommended	
			M10c: Evanescence: Architectural Support for Efficient Data Sanitization in Modern Flash-Based Storage Systems PDF PPT Paper	Required Recommended	
			M10d: DeepSketch: A New Machine Learning-Based Reference Search Technique for Post-Deduplication Delta Compression PDF PPT Paper	Required Recommended	
W11	11.01	YouTube Live	M11: FLIN: Enabling Fairness and Enhancing Performance in Modern NVMe Solid State Drives PDF PPT	Required	
W12	25.01	YouTube Premiere	M12: Flash Memory and Solid-State Drives PDF PPT	Recommended	

Genomics Course (Fall 2022)

■ Fall 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics

■ Spring 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics

■ Youtube Livestream (Fall 2022):

- https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFIQvdxOdizD_EhVAMVQV

■ Youtube Livestream (Spring 2022):

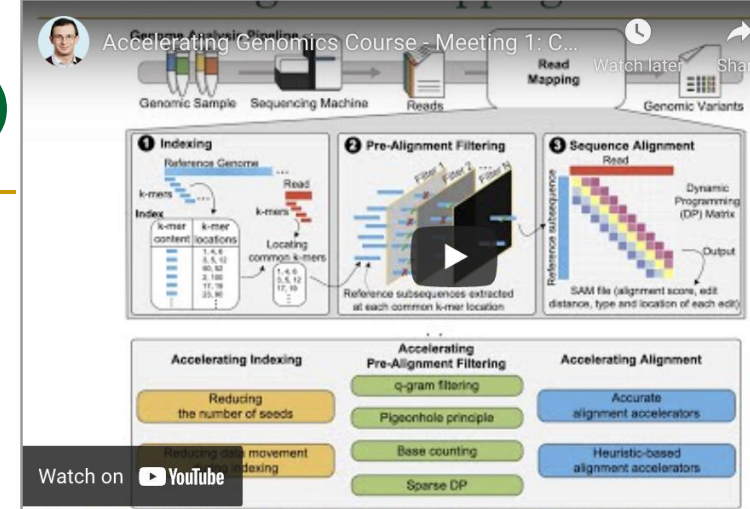
- https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18

■ Project course

- Taken by Bachelor's/Master's students
- Genomics lectures
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>

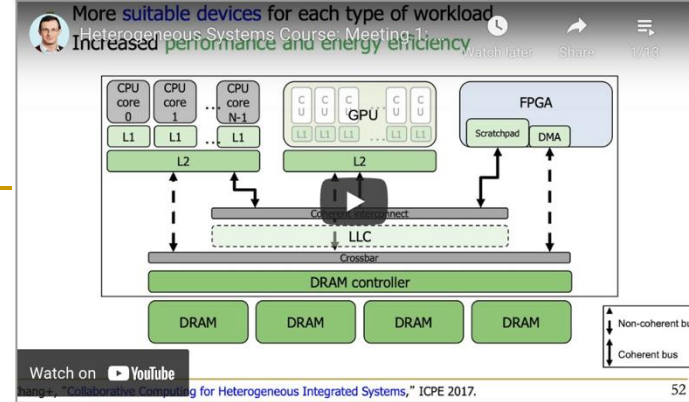
SAFARI



Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials
W1	11.3 Fri.	YouTube Live	M1: P&S Accelerating Genomics Course Introduction & Project Proposals PDF PPT	Required Materials Recommended Materials
W2	18.3 Fri.	YouTube Live	M2: Introduction to Sequencing PDF PPT	
W3	25.3 Fri.	YouTube Premiere	M3: Read Mapping PDF PPT	
W4	01.04 Fri.	YouTube Premiere	M4: GateKeeper PDF PPT	
W5	08.04 Fri.	YouTube Premiere	M5: MAGNET & Shouji PDF PPT	
W6	15.4 Fri.	YouTube Premiere	M6: SneakySnake PDF PPT	
W7	29.4 Fri.	YouTube Premiere	M7: GenStore PDF PPT	
W8	06.05 Fri.	YouTube Premiere	M8: GRIM-Filter PDF PPT	
W9	13.05 Fri.	YouTube Premiere	M9: Genome Assembly PDF PPT	
W10	20.05 Fri.	YouTube Live	M10: Genomic Data Sharing Under Differential Privacy PDF PPT	
W11	10.06 Fri.	YouTube Premiere	M11: Accelerating Genome Sequence Analysis PDF PPT	

Hetero. Systems (Spring'22)



Spring 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=heterogeneous_systems

Youtube Livestream:

- https://www.youtube.com/watch?v=oFO5fTrgFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm

Project course

- Taken by Bachelor's/Master's students
- GPU and Parallelism lectures
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>

Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	15.03 Tue.	YouTube Premiere	M1: P&S Course Presentation PDF PPT	Required Materials Recommended Materials	HW 0 Out
W2	22.03 Tue.	YouTube Premiere	M2: SIMD Processing and GPUs PDF PPT		
W3	29.03 Tue.	YouTube Premiere	M3: GPU Software Hierarchy PDF PPT		
W4	05.04 Tue.	YouTube Premiere	M4: GPU Memory Hierarchy PDF PPT		
W5	12.04 Tue.	YouTube Premiere	M5: GPU Performance Considerations PDF PPT		
W6	19.04 Tue.	YouTube Premiere	M6: Parallel Patterns: Reduction PDF PPT		
W7	26.04 Tue.	YouTube Premiere	M7: Parallel Patterns: Histogram PDF PPT		
W8	03.05 Tue.	YouTube Premiere	M8: Parallel Patterns: Convolution PDF PPT		
W9	10.05 Tue.	YouTube Premiere	M9: Parallel Patterns: Prefix Sum (Scan) PDF PPT		
W10	17.05 Tue.	YouTube Premiere	M10: Parallel Patterns: Sparse Matrices PDF PPT		
W11	24.05 Tue.	YouTube Premiere	M11: Parallel Patterns: Graph Search PDF PPT		
W12	01.06 Wed.	YouTube Premiere	M12: Parallel Patterns: Merge Sort PDF PPT		
W13	07.06 Tue.	YouTube Premiere	M13: Dynamic Parallelism PDF PPT		
W14	15.06 Wed.	YouTube Premiere	M14: Collaborative Computing PDF PPT		
W15	24.06 Fri.	YouTube Premiere	M15: GPU Acceleration of Genome Sequence Alignment PDF PPT		
W16	14.07 Thu.	YouTube Premiere	M16: Accelerating Agent-based Simulations PDF ODP		

HW/SW Co-Design (Spring 2022)

Spring 2022 Edition:

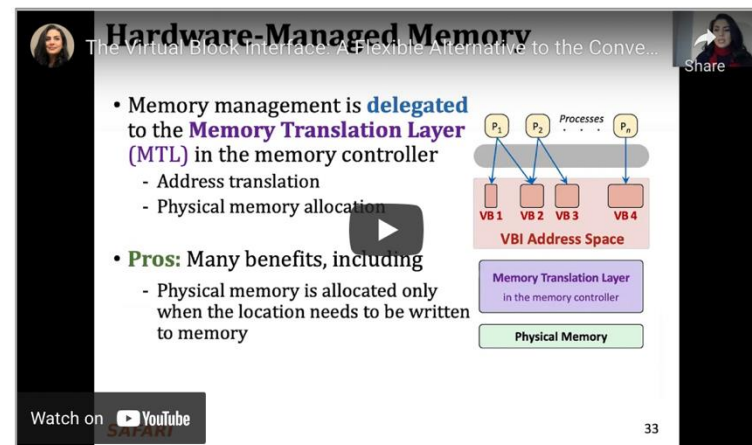
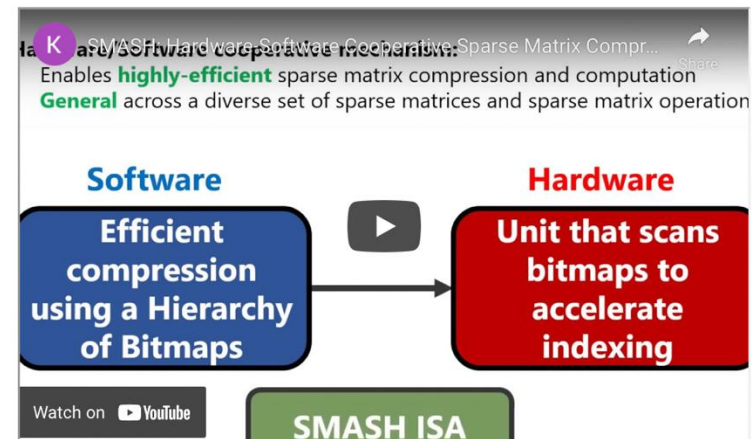
- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_co_design

Youtube Livestream:

- https://youtube.com/playlist?list=PL5Q2s_oXY2Zi8nH7un3ghD2nutKWWDk-NK

Project course

- Taken by Bachelor's/Master's students
- HW/SW co-design lectures
- Hands-on research exploration
- Many research readings



2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Materials	Assignments
W0	16.03	Live	Intro to HW/SW Co-Design PPTX PDF	Required	HW 0 Out
W1	23.03		Project selection	Required	
W2	30.03	Live	Virtual Memory (I) PPTX PDF		
W3	13.04	Live	Virtual Memory (II) PPTX PDF		

<https://www.youtube.com/onurmutlulectures>

RowHammer & DRAM Exploration (Fall 2022)

Fall 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc

Spring 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=softmc

Youtube Livestream (Spring 2022):

- https://www.youtube.com/watch?v=r5QxuoJWttg&list=PL5Q2soXY2Zi_1trfCckr6PTN8WR72icUO

Bachelor's course

- Elective at ETH Zurich
- Introduction to DRAM organization & operation
- Tutorial on using FPGA-based infrastructure
- Verilog & C++
- Potential research exploration

Lecture Video Playlist on YouTube

Lecture Playlist



2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W0	23.02 Wed.		P&S SoftMC Tutorial	SoftMC Tutorial Slides (PDF) (PPT)	
W1	08.03 Tue.		M1: Logistics & Intro to DRAM and SoftMC (PDF) (PPT)	Required Materials Recommended Materials	HW0
W2	15.03 Tue.		M2: Revisiting RowHammer (PDF) (PPT)	(Paper PDF)	
W3	22.03 Tue.		M3: Uncovering in-DRAM TRR & TRRespass (PDF) (PPT)		
W4	29.03 Tue.		M4: Deeper Look Into RowHammer's Sensitivities (PDF) (PPT)		
W5	05.04 Tue.		M5: QUAC-TRNG (PDF) (PPT)		
W6	12.04 Tue.		M6: PiDRAM (PDF) (PPT)		

<https://www.youtube.com/onurmutlulectures>

Exploration of Emerging Memory Systems (Fall 2022)

Fall 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator

Spring 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator

Youtube Livestream (Spring 2022):

- ❑ https://www.youtube.com/watch?v=aM-lIXRQd3s&list=PL5Q2soXY2Zi_TlmlGw_Z8hBo2925ZAqV

Bachelor's course

- ❑ Elective at ETH Zurich
- ❑ Introduction to memory system simulation
- ❑ Tutorial on using Ramulator
- ❑ C++
- ❑ Potential research exploration

<https://www.youtube.com/onurmutlulectures>

Lecture Video Playlist on YouTube

Lecture Playlist



2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	09.03 Wed.	Video	M1: Logistics & Intro to Simulating Memory Systems Using Ramulator (PDF) (PPT)		HW0
W2	16.03 Fri.	Video	M2: Tutorial on Using Ramulator (PDF) (PPT)		
W3	25.02 Fri.	Video	M3: BlockHammer (PDF) (PPT)		
W4	01.04 Fri.	Video	M4: CLR-DRAM (PDF) (PPT)		
W5	08.04 Fri.	Video	M5: SIMDRAM (PDF) (PPT)		
W6	29.04 Fri.	Video	M6: DAMOV (PDF) (PPT)		
W7	06.05 Fri.	Video	M7: Synchron (PDF) (PPT)		

More Backup Slides

SAFARI PhD and Post-Doc Alumni

- <https://safari.ethz.ch/safari-alumni/>
- Hasan Hassan (Rivos), **EDAA Outstanding Dissertation Award 2023**; S&P 2020 Best Paper Award, 2020 Pwnie Award, IEEE Micro TP HM 2020
- Christina Giannoula (Univ. of Toronto), **NTUA Best Dissertation Award 2023**
- Minesh Patel (Rutgers, Asst. Prof.), **DSN Carter Award Best Thesis 2022**; ETH Medal 2023; MICRO'20 & DSN'20 Best Paper Awards; ISCA HoF 2021
- Damla Senol Cali (Bionano Genomics), **SRC TECHCON 2019 Best Student Presentation Award**; RECOMB-Seq 2018 Best Poster Award
- Nastaran Hajinazar (Intel)
- Gagandeep Singh (AMD/Xilinx), **FPL 2020 Best Paper Award Finalist**
- Amirali Boroumand (Stanford Univ → Google), **SRC TECHCON 2018 Best Presentation Award**
- Jeremie Kim (Apple), **EDAA Outstanding Dissertation Award 2020**; IEEE Micro Top Picks 2019; ISCA/MICRO HoF 2021
- Nandita Vijaykumar (Univ. of Toronto, Assistant Professor), **ISCA Hall of Fame 2021**
- Kevin Hsieh (Microsoft Research, Senior Researcher)
- Justin Meza (Facebook), **HiPEAC 2015 Best Student Presentation Award**; ICCD 2012 Best Paper Award
- Mohammed Alser (ETH Zurich), **IEEE Turkey Best PhD Thesis Award 2018**
- Yixin Luo (Google), **HPCA 2015 Best Paper Session**
- Kevin Chang (Facebook), **SRC TECHCON 2016 Best Student Presentation Award**
- Rachata Ausavarungnirun (KMUNTB, Assistant Professor), **NOCS 2015 and NOCS 2012 Best Paper Award Finalist**
- Gennady Pekhimenko (Univ. of Toronto, Assistant Professor), **ISCA Hall of Fame 2021**; ASPLOS 2015 SRC Winner
- Vivek Seshadri (Microsoft Research)
- Donghyuk Lee (NVIDIA Research, Senior Researcher), **HPCA Hall of Fame 2018**
- Yoongu Kim (Software Robotics → Google), **TCAD'19 Top Pick Award**; IEEE Micro Top Picks'10; HPCA'10 Best Paper Session
- Lavanya Subramanian (Intel Labs → Facebook)

- Samira Khan (Univ. of Virginia, Assistant Professor), **HPCA 2014 Best Paper Session**
- Saugata Ghose (Univ. of Illinois, Assistant Professor), **DFRWS-EU 2017 Best Paper Award**
- Jawad Haj-Yahya (Huawei Research Zurich, Principal Researcher)
- Lois Orosa (Galicia Supercomputing Center, Director)
- Jisung Park (POSTECH, Assistant Professor)
- Gagandeep Singh (AMD/Xilinx, Researcher)
- Juan Gomez-Luna (NVIDIA, Researcher), **ISPASS 2023 Best Paper Session**

You Can Join Us!

■ <https://safari.ethz.ch/apply/>

SAFARI Researcher Applications

Sign in

This is the application submission site to be considered for being a researcher in the [SAFARI Research Group](#), directed by [Professor Onur Mutlu \(Publications and Teaching\)](#).

If you are interested in doing research in the [SAFARI Research Group](#), please make sure you apply through this submissions site and supply as many of the requested documents and information as possible. Please read and follow the provided instructions and submit as complete an application as possible (given the position you are applying for).

We suggest studying the following materials before submission:

[SAFARI Publications and Courses](#)

[Onur Mutlu's Online Lectures and Course Materials](#)

We strongly recommend that you read and analyze critically as many recent papers from our group as possible. This is the best way to prepare for the application process. Our recommendation is that you use professor Mutlu's methodology for critically analyzing papers.

[Guide On Reviewing Papers](#)

Good luck!

Welcome to the SAFARI at ETH Zurich -- PhD, Postdoc, Internship, Visiting Researcher Applications (SAFARI Researcher Applications) submissions site.

Data-Driven (Self-Optimizing) Architectures

System Architecture Design Today

- Human-driven
 - Humans design the policies (how to do things)
- Many (too) simple, short-sighted policies all over the system
- No automatic data-driven policy learning
- (Almost) no learning: cannot take lessons from past actions

**Can we design
fundamentally intelligent architectures?**

An Intelligent Architecture

- Data-driven
 - Machine learns the “best” policies (how to do things)
- Sophisticated, workload-driven, changing, far-sighted policies
- Automatic data-driven policy learning
- All controllers are intelligent data-driven agents

**We need to rethink design
(of all controllers)**

Self-Optimizing Memory Controllers

- Engin Ipek, Onur Mutlu, José F. Martínez, and Rich Caruana,
"Self Optimizing Memory Controllers: A Reinforcement Learning Approach"
Proceedings of the 35th International Symposium on Computer Architecture (ISCA), pages 39-50, Beijing, China, June 2008.
Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

Self-Optimizing Memory Controllers: A Reinforcement Learning Approach

Engin İpek^{1,2} Onur Mutlu² José F. Martínez¹ Rich Caruana¹

¹Cornell University, Ithaca, NY 14850 USA

²Microsoft Research, Redmond, WA 98052 USA

Self-Optimizing Memory Prefetchers

Rahul Bera, Konstantinos Kanellopoulos, Anant Nori, Taha Shahroodi, Sreenivas Subramoney, and Onur Mutlu,
"Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning"
Proceedings of the 54th International Symposium on Microarchitecture (MICRO), Virtual, October 2021.

[[Slides \(pptx\)](#)] ([pdf](#))

[[Short Talk Slides \(pptx\)](#)] ([pdf](#))

[[Lightning Talk Slides \(pptx\)](#)] ([pdf](#))

[[Talk Video](#)] (20 minutes)

[[Lightning Talk Video](#)] (1.5 minutes)

[[Pythia Source Code](#) (Officially Artifact Evaluated with All Badges)]

[[arXiv version](#)]

Officially artifact evaluated as available, reusable and reproducible.



Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera¹

Konstantinos Kanellopoulos¹

Anant V. Nori²

Taha Shahroodi^{3,1}

Sreenivas Subramoney²

Onur Mutlu¹

¹ETH Zürich

²Processor Architecture Research Labs, Intel Labs

³TU Delft

<https://arxiv.org/pdf/2109.12021.pdf>

Learning-Based Off-Chip Load Predictors

- Rahul Bera, Konstantinos Kanellopoulos, Shankar Balachandran, David Novo, Ataberk Olgun, Mohammad Sadrosadati, and Onur Mutlu,
"Hermes: Accelerating Long-Latency Load Requests via Perceptron-Based Off-Chip Load Prediction"
Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.
[[Slides \(pptx\)](#)] ([pdf](#))
[[Longer Lecture Slides \(pptx\)](#)] ([pdf](#))
[[Talk Video](#) (12 minutes)]
[[Lecture Video](#) (25 minutes)]
[[arXiv version](#)]
[[Source Code \(Officially Artifact Evaluated with All Badges\)](#)]
Officially artifact evaluated as available, reusable and reproducible.
Best paper award at MICRO 2022.



Hermes: Accelerating Long-Latency Load Requests via Perceptron-Based Off-Chip Load Prediction

Rahul Bera¹ Konstantinos Kanellopoulos¹ Shankar Balachandran² David Novo³
Ataberk Olgun¹ Mohammad Sadrosadati¹ Onur Mutlu¹

¹ETH Zürich ²Intel Processor Architecture Research Lab ³LIRMM, Univ. Montpellier, CNRS

<https://arxiv.org/pdf/2209.00188.pdf>

Self-Optimizing Hybrid SSD Controllers

Gagandeep Singh, Rakesh Nadig, Jisung Park, Rahul Bera, Nastaran Hajinazar, David Novo, Juan Gomez-Luna, Sander Stuijk, Henk Corporaal, and Onur Mutlu,

"Sibyl: Adaptive and Extensible Data Placement in Hybrid Storage Systems Using Online Reinforcement Learning"

Proceedings of the 49th International Symposium on Computer Architecture (ISCA), New York, June 2022.

[[Slides \(pptx\)](#) ([pdf](#))]

[[arXiv version](#)]

[[Sibyl Source Code](#)]

[[Talk Video](#) (16 minutes)]

Sibyl: Adaptive and Extensible Data Placement in Hybrid Storage Systems Using Online Reinforcement Learning

Gagandeep Singh ¹	Rakesh Nadig ¹	Jisung Park ¹	Rahul Bera ¹	Nastaran Hajinazar ¹
David Novo ³	Juan Gómez-Luna ¹	Sander Stuijk ²	Henk Corporaal ²	Onur Mutlu ¹

¹ETH Zürich

²Eindhoven University of Technology

³LIRMM, Univ. Montpellier, CNRS

Data-Driven (Self-Optimizing) Computing Architectures

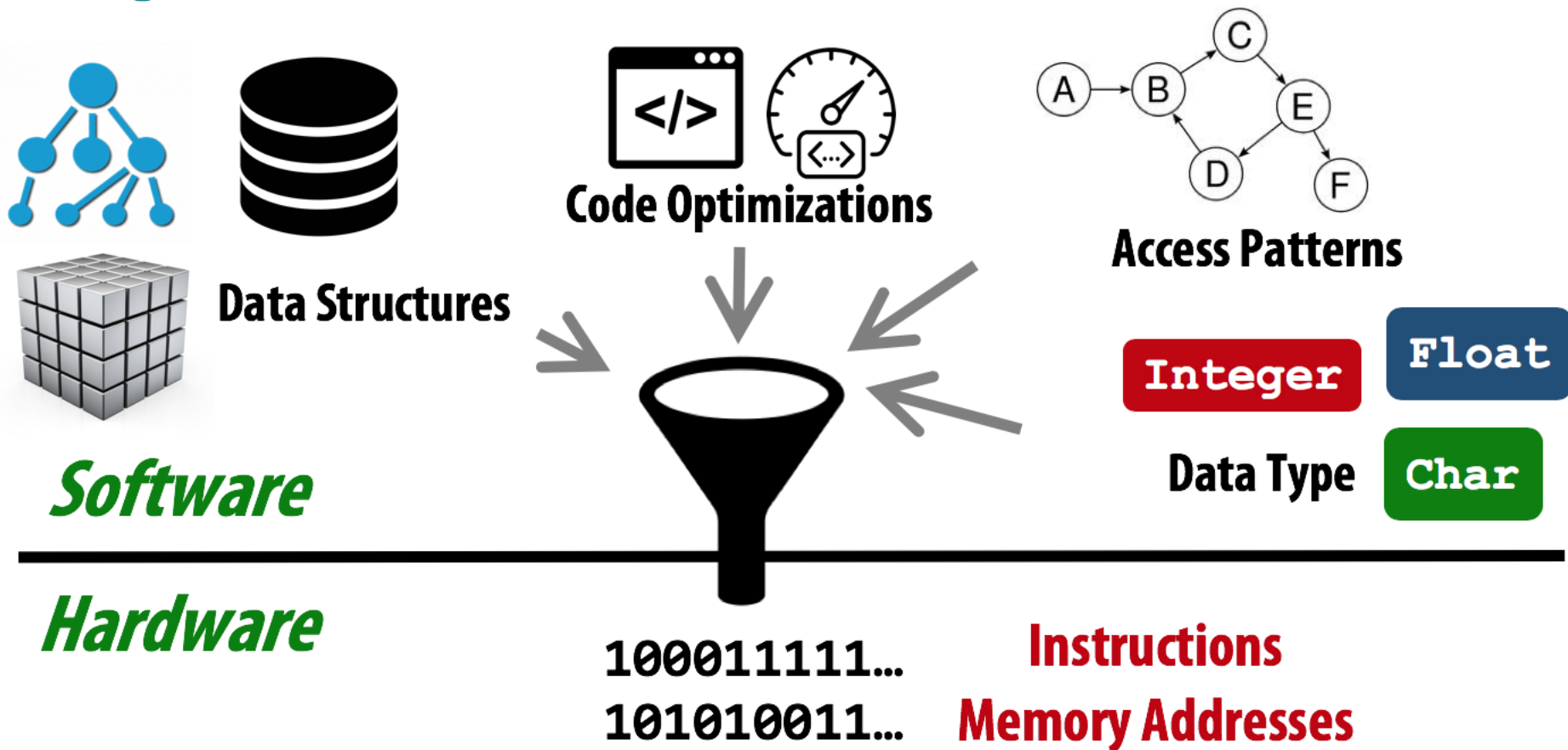
Data-Characteristic-Aware Architectures

Data-Aware Architectures

- A data-aware architecture understands what it can do with and to each piece of data
- It makes use of different properties of data to improve performance, efficiency and other metrics
 - Compressibility
 - Approximability
 - Locality
 - Sparsity
 - Criticality for Computation X
 - Access Semantics
 - ...

One Problem: Limited Expressiveness

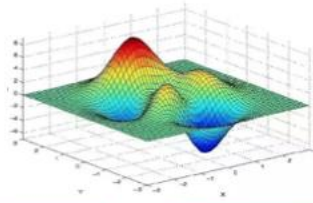
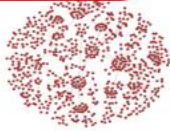
Higher-level information is not visible to HW



A Solution: More Expressive Interfaces

Performance

Software



Functionality



**ISA
Virtual Memory**

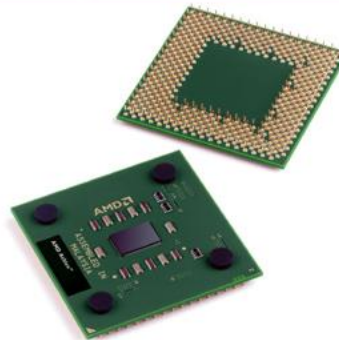
**Higher-level
Program
Semantics**

**Expressive
Memory
“XMem”**

Hardware



wiseGEEK



Expressive (Memory) Interfaces

- Nandita Vijaykumar, Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko, Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons and Onur Mutlu, **"A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory"**
Proceedings of the 45th International Symposium on Computer Architecture (ISCA), Los Angeles, CA, USA, June 2018.
[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Talk Slides \(pptx\) \(pdf\)](#)]
[[Lightning Talk Video](#)]

A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory

Nandita Vijaykumar^{†§} Abhilasha Jain[†] Diptesh Majumdar[†] Kevin Hsieh[†] Gennady Pekhimenko[‡]
Eiman Ebrahimi[⌘] Nastaran Hajinazar[†] Phillip B. Gibbons[†] Onur Mutlu^{§†}

[†]Carnegie Mellon University

[‡]University of Toronto

[⌘]NVIDIA

[†]Simon Fraser University

[§]ETH Zürich

Expressive (Memory) Interfaces for GPUs

- Nandita Vijaykumar, Eiman Ebrahimi, Kevin Hsieh, Phillip B. Gibbons and Onur Mutlu, **"The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs"**

Proceedings of the 45th International Symposium on Computer Architecture (ISCA), Los Angeles, CA, USA, June 2018.

[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Talk Slides \(pptx\) \(pdf\)](#)]

[[Lightning Talk Video](#)]

The Locality Descriptor:

A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs

Nandita Vijaykumar^{†§}

Eiman Ebrahimi[‡]

Kevin Hsieh[†]

Phillip B. Gibbons[†]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[‡]NVIDIA

[§]ETH Zürich

Open-Source Frameworks for Data-Aware Systems

- Nandita Vijaykumar, Ataberk Olgun, Konstantinos Kanellopoulos, F. Nisa Bostanci, Hasan Hassan, Mehrshad Lotfi, Phillip B. Gibbons, and Onur Mutlu,
"MetaSys: A Practical Open-source Metadata Management System to Implement and Evaluate Cross-layer Optimizations"
ACM Transactions on Architecture and Code Optimization (TACO), June 2022.
[arXiv version]
Presented at the *18th HiPEAC Conference*, Toulouse, France, January 2023.
[Slides (pptx) (pdf)]
[Preliminary Talk Video (14 minutes)]
[SAFARI Live Seminar Video (1 hour 26 minutes)]
[MetaSys Source Code]
Best paper award at HiPEAC 2023.

MetaSys: A Practical Open-Source Metadata Management System to Implement and Evaluate Cross-Layer Optimizations

Nandita Vijaykumar* Ataberk Olgun[§] Konstantinos Kanellopoulos[§] Hasan Hassan[§]
Mehrshad Lotfi[§] Phillip B. Gibbons[†] Onur Mutlu[§]

*University of Toronto

[§]ETH Zürich

[†]Carnegie Mellon University

Heterogeneous-Reliability Memory

- Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, and Onur Mutlu,
"Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory"
Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, GA, June 2014. [[Summary](#)] [[Slides \(pptx\)](#)] [[pdf](#)] [[Coverage on ZDNet](#)]

Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory

Yixin Luo Sriram Govindan* Bikash Sharma* Mark Santaniello* Justin Meza
Aman Kansal* Jie Liu* Badriddine Khessib* Kushagra Vaid* Onur Mutlu

Carnegie Mellon University, yixinluo@cs.cmu.edu, {meza, onur}@cmu.edu

*Microsoft Corporation, {srgovin, bsharma, marksan, kansal, jie.liu, bkhessib, kvaid}@microsoft.com

EDEN: Data-Aware Efficient DNN Inference

- Skanda Koppula, Lois Orosa, A. Giray Yaglikci, Roknoddin Azizi, Taha Shahroodi, Konstantinos Kanellopoulos, and Onur Mutlu,
"EDEN: Enabling Energy-Efficient, High-Performance Deep Neural Network Inference Using Approximate DRAM"
Proceedings of the 52nd International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, October 2019.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Poster \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Video](#) (90 seconds)]
[[Full Talk Lecture](#) (38 minutes)]

EDEN: Enabling Energy-Efficient, High-Performance Deep Neural Network Inference Using Approximate DRAM

Skanda Koppula Lois Orosa A. Giray Yağlıkçı
Roknoddin Azizi Taha Shahroodi Konstantinos Kanellopoulos Onur Mutlu

ETH Zürich

SMASH: SW/HW Indexing Acceleration

- Konstantinos Kanellopoulos, Nandita Vijaykumar, Christina Giannoula, Roknoddin Azizi, Skanda Koppula, Nika Mansouri Ghiasi, Taha Shahroodi, Juan Gomez-Luna, and Onur Mutlu,

"SMASH: Co-designing Software Compression and Hardware-Accelerated Indexing for Efficient Sparse Matrix Operations"

Proceedings of the 52nd International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, October 2019.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Poster \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Video](#) (90 seconds)]

[[Full Talk Lecture](#) (30 minutes)]

SMASH: Co-designing Software Compression and Hardware-Accelerated Indexing for Efficient Sparse Matrix Operations

Konstantinos Kanellopoulos¹ Nandita Vijaykumar^{2,1} Christina Giannoula^{1,3} Roknoddin Azizi¹
Skanda Koppula¹ Nika Mansouri Ghiasi¹ Taha Shahroodi¹ Juan Gomez Luna¹ Onur Mutlu^{1,2}

¹ETH Zürich

²Carnegie Mellon University

³National Technical University of Athens

Rethinking Virtual Memory

- Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,
"The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework"
Proceedings of the 47th International Symposium on Computer Architecture (ISCA), Valencia, Spain, June 2020.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]
[[ARM Research Summit Poster \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (26 minutes)]
[[Lightning Talk Video](#) (3 minutes)]

The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework

Nastaran Hajinazar^{*†} Pratyush Patel[✕] Minesh Patel^{*} Konstantinos Kanellopoulos^{*} Saugata Ghose[‡]
Rachata Ausavarungnirun[⊙] Geraldo F. Oliveira^{*} Jonathan Appavoo[◇] Vivek Seshadri[▽] Onur Mutlu^{*‡}

^{*}ETH Zürich [†]Simon Fraser University [✕]University of Washington [‡]Carnegie Mellon University
[⊙]King Mongkut's University of Technology North Bangkok [◇]Boston University [▽]Microsoft Research India

Data-Characteristic-Aware Computing Architectures

More Background Slides

Memory Scaling Issues **Are** Real

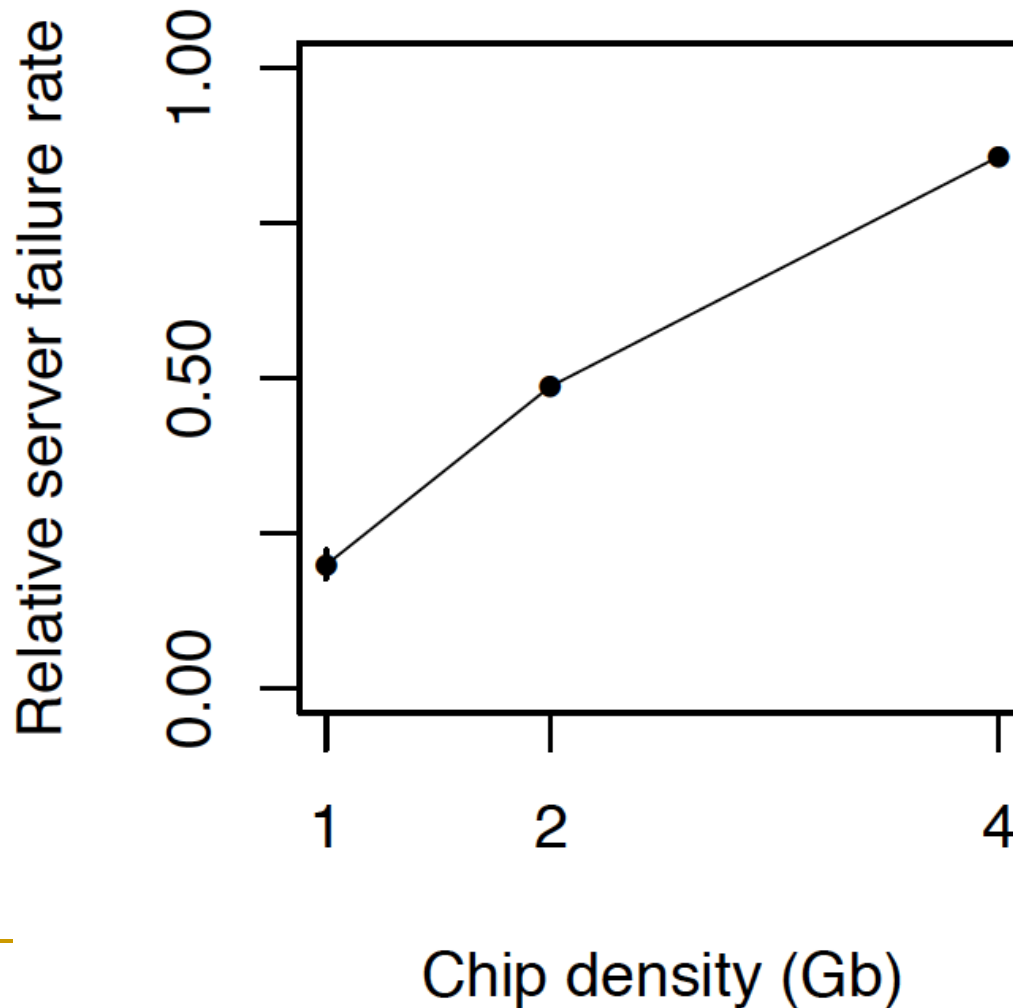
- Onur Mutlu,
"Memory Scaling: A Systems Architecture Perspective"
Proceedings of the 5th International Memory Workshop (IMW), Monterey, CA, May 2013. Slides
(pptx) (pdf)
EETimes Reprint

Memory Scaling: A Systems Architecture Perspective

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu
<http://users.ece.cmu.edu/~omutlu/>

As Memory Scales, It Becomes Unreliable

- Data from all of Facebook's servers worldwide
- Meza+, "Revisiting Memory Errors in Large-Scale Production Data Centers," DSN'15.



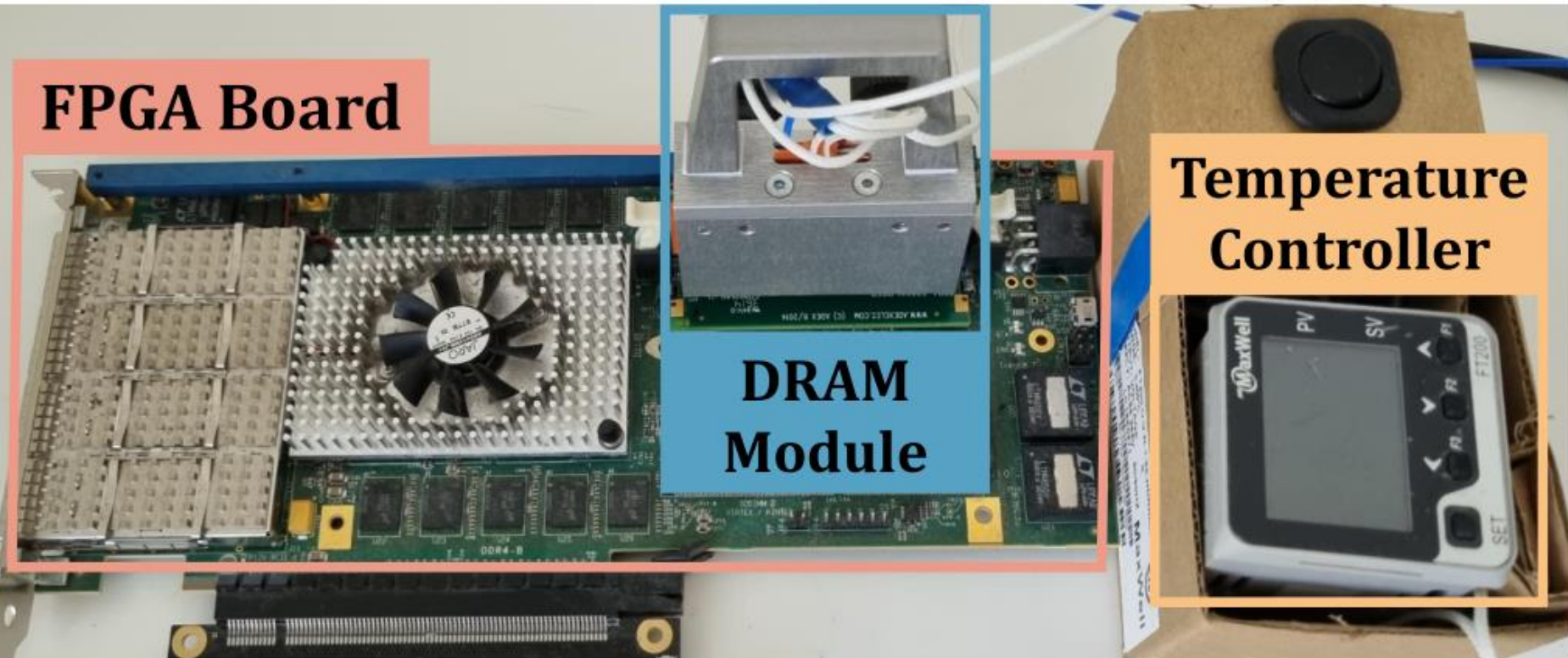
*Intuition:
quadratic
increase
in
capacity*

Memory Testing Infrastructures

FPGA Board

**DRAM
Module**

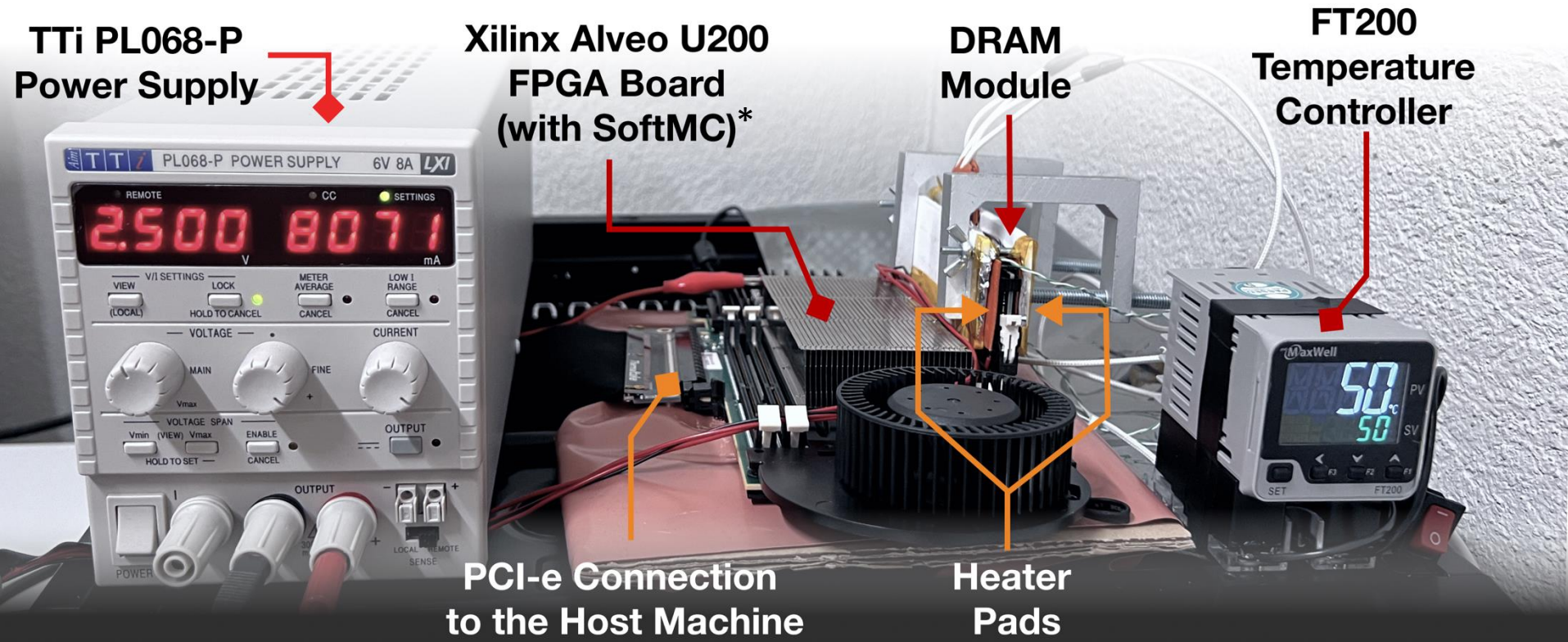
**Temperature
Controller**



** SoftMC [Hassan+, HPCA'17] enhanced for DDR4*

Updated Memory Testing Infrastructure

FPGA-based SoftMC (Xilinx Virtex UltraScale+ XCU200)



Fine-grained control over DRAM commands,
timing ($\pm 1.5\text{ns}$), **temperature ($\pm 0.1^\circ\text{C}$)**,
and **voltage ($\pm 1\text{mV}$)**

SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan, Nandita Vijaykumar, Samira Khan, Saugata Ghose, Kevin Chang, Gennady Pekhimenko, Donghyuk Lee, Oguz Ergin, and Onur Mutlu,

"SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies"

Proceedings of the 23rd International Symposium on High-Performance Computer Architecture (HPCA), Austin, TX, USA, February 2017.

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]

[Full Talk Lecture (39 minutes)]

[Source Code]

SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan^{1,2,3} Nandita Vijaykumar³ Samira Khan^{4,3} Saugata Ghose³ Kevin Chang³
Gennady Pekhimenko^{5,3} Donghyuk Lee^{6,3} Oguz Ergin² Onur Mutlu^{1,3}

¹ETH Zürich ²TOBB University of Economics & Technology ³Carnegie Mellon University
⁴University of Virginia ⁵Microsoft Research ⁶NVIDIA Research

DRAM Bender

- Ataberk Olgun, Hasan Hassan, A Giray Yağlıkçı, Yahya Can Tuğrul, Lois Orosa, Haocong Luo, Minesh Patel, Oğuz Ergin, and Onur Mutlu,
"DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips"
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 2023.
[[Extended arXiv version](#)]
[[DRAM Bender Source Code](#)]
[[DRAM Bender Tutorial Video](#) (43 minutes)]

DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips

Ataberk Olgun[§] Hasan Hassan[§] A. Giray Yağlıkçı[§] Yahya Can Tuğrul^{§†}
Lois Orosa^{§⊙} Haocong Luo[§] Minesh Patel[§] Oğuz Ergin[†] Onur Mutlu[§]
 [§]*ETH Zürich* [†]*TOBB ETÜ* [⊙]*Galician Supercomputing Center*

A Curious Phenomenon [Kim et al., ISCA 2014]

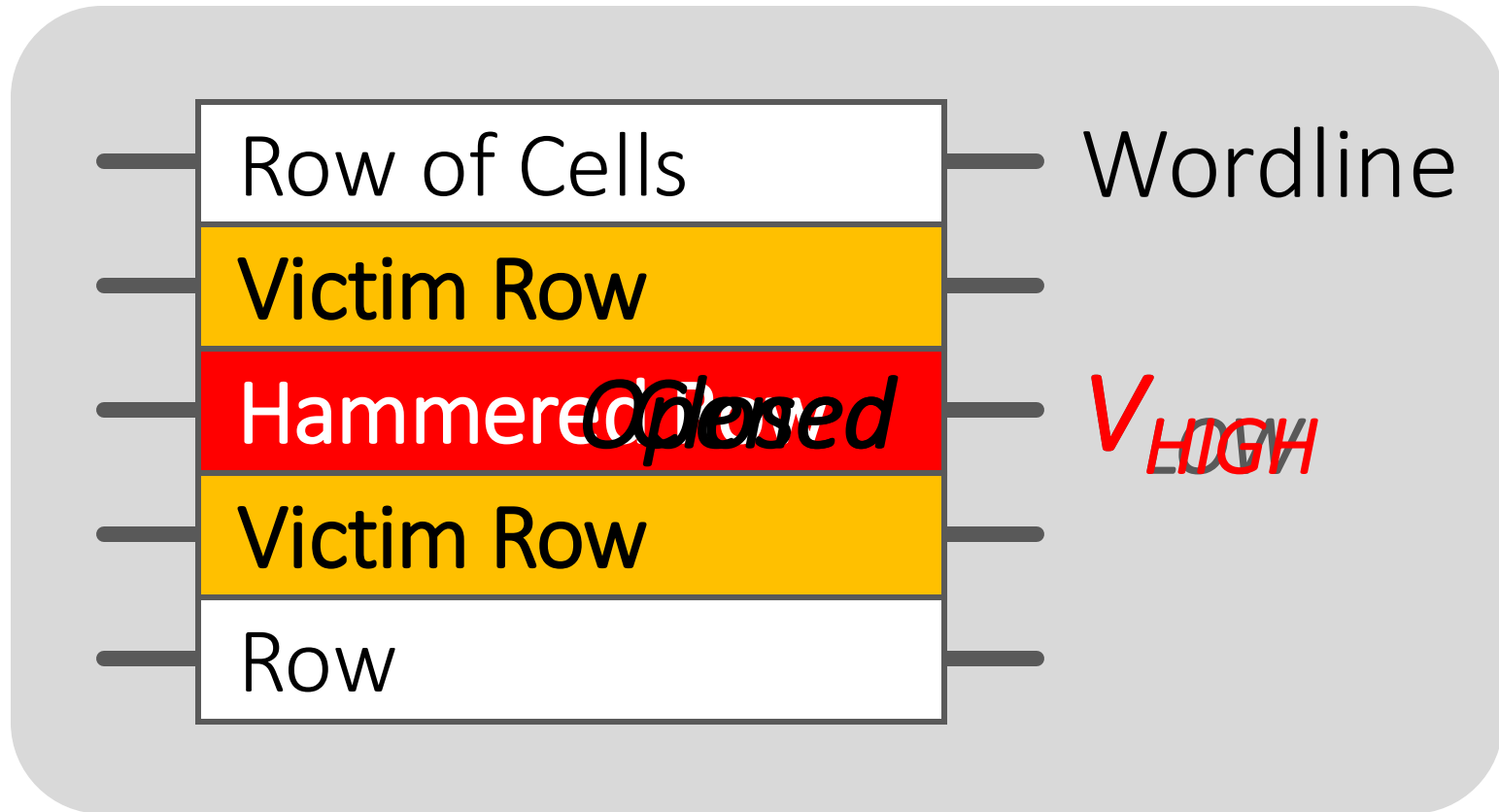
One can
predictably induce errors
in most DRAM memory chips

Kim+, "[Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors](#)," ISCA 2014.



Rowhammer

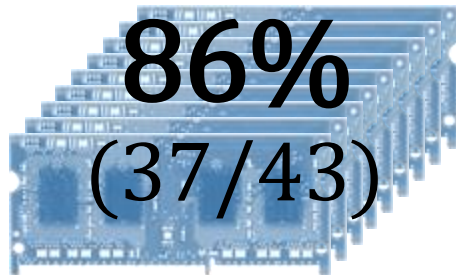
Modern Memory is Prone to Disturbance Errors



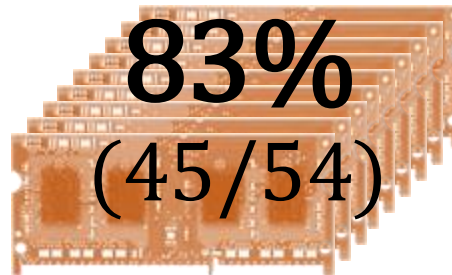
Repeatedly reading a row enough times (before memory gets refreshed) induces **disturbance errors** in **adjacent rows** in **most real DRAM chips you can buy today**

Most DRAM Modules Are Vulnerable

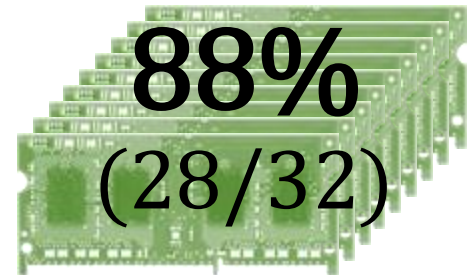
A company



B company



C company



Up to
 1.0×10^7
errors

Up to
 2.7×10^6
errors

Up to
 3.3×10^5
errors

The RowHammer Vulnerability

A simple hardware failure mechanism
can create a widespread
system security vulnerability

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS	CULTURE	DESIGN	GEAR	SCIENCE
----------	---------	--------	------	---------

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE



SHARE
18276



TWEET

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

RowHammer [ISCA 2014]

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"

Proceedings of the 41st International Symposium on Computer Architecture (ISCA), Minneapolis, MN, June 2014.

[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Session Slides \(pptx\) \(pdf\)](#)] [[Source Code and Data](#)] [[Lecture Video](#) (1 hr 49 mins), 25 September 2020]

One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD ([link](#)).

Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 ([Retrospective \(pdf\)](#) [Full Issue](#)).

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim¹ Ross Daly* Jeremie Kim¹ Chris Fallin* Ji Hye Lee¹
Donghyuk Lee¹ Chris Wilkerson² Konrad Lai Onur Mutlu¹

¹Carnegie Mellon University ²Intel Labs

Memory Scaling Issues **Are** Real

- Onur Mutlu,
"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"

*Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (**DATE**), Lausanne, Switzerland, March 2017.*

[Slides (pptx) (pdf)]

The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch
<https://people.inf.ethz.ch/omutlu>

Memory Scaling Issues **Are** Real

- Onur Mutlu and Jeremie Kim,
["RowHammer: A Retrospective"](#)
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.
[[Preliminary arXiv version](#)]
[[Slides from COSADE 2019 \(pptx\)](#)]
[[Slides from VLSI-SOC 2020 \(pptx\) \(pdf\)](#)]
[[Talk Video](#) (1 hr 15 minutes, with Q&A)]

RowHammer: A Retrospective

Onur Mutlu^{§‡} Jeremie S. Kim^{‡§}
§ETH Zürich ‡Carnegie Mellon University

Memory Scaling Issues **Are** Real

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci,
"Fundamentally Understanding and Solving RowHammer"
Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, January 2023.
[arXiv version]
[Slides (pptx) (pdf)]
[Talk Video (26 minutes)]

Fundamentally Understanding and Solving RowHammer

Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

Ataberk Olgun
ataberk.olgund@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

A. Giray Yağlıkçı
giray.yaglikci@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

Main Memory Needs Intelligent Controllers

Industry's Intelligent DRAM Controllers (I)

ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Oh Yong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

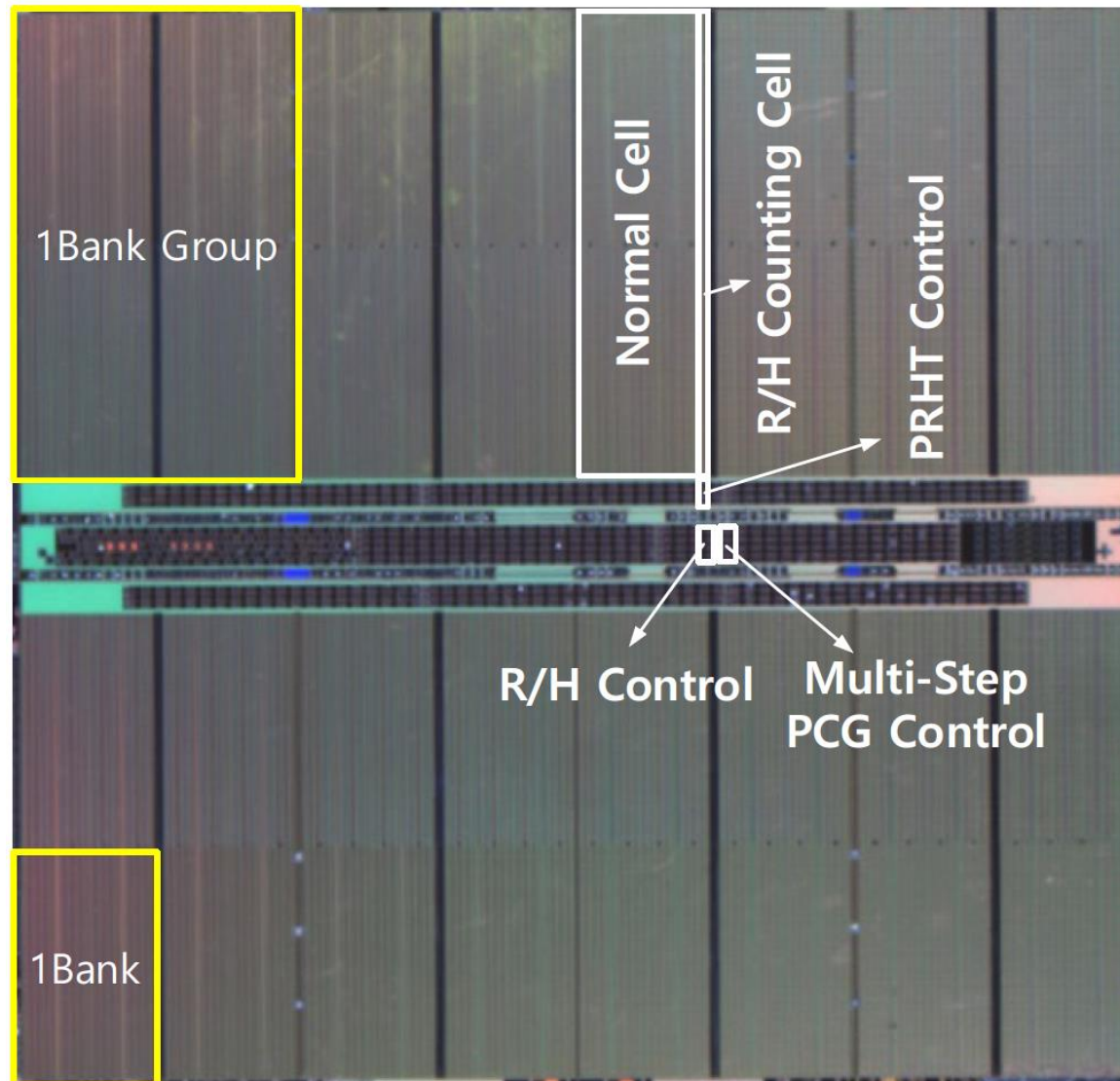


Industry's Intelligent DRAM Controllers (II)

SK hynix Semiconductor, Icheon, Korea

DRAM products have been recently adopted in a wide range of high-performance computing applications: such as in cloud computing, in big data systems, and IoT devices. This demand creates larger memory capacity requirements, thereby requiring aggressive DRAM technology node scaling to reduce the cost per bit [1,2]. However, DRAM manufacturers are facing technology scaling challenges due to row hammer and refresh retention time beyond 1a-nm [2]. Row hammer is a failure mechanism, where repeatedly activating a DRAM row disturbs data in adjacent rows. Scaling down severely threatens reliability since a reduction of DRAM cell size leads to a reduction in the intrinsic row hammer tolerance [2,3]. To improve row hammer tolerance, there is a need to probabilistically activate adjacent rows with carefully sampled active addresses and to improve intrinsic row hammer tolerance [2]. In this paper, row-hammer-protection and refresh-management schemes are presented to guarantee DRAM security and reliability despite the aggressive scaling from 1a-nm to sub 10-nm nodes. The probabilistic-aggressor-tracking scheme with a refresh-management function (RFM) and per-row hammer tracking (PRHT) improve DRAM resilience. A multi-step precharge reinforces intrinsic row-hammer tolerance and a core-bias modulation improves retention time: even in the face of cell-transistor degradation due to technology scaling. This comprehensive scheme leads to a reduced probability of failure, due to row hammer attacks, by 93.1% and an improvement in retention time by 17%.

Industry's Intelligent DRAM Controllers (III)



ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm

Seungki Hong Dongha Kim Jaehyung Lee Reum Oh
Changsik Yoo Sangjoon Hwang Jooyoung Lee

DRAM Design Team, Memory Division, Samsung Electronics

<https://arxiv.org/pdf/2302.03591v1.pdf>

Are We Now BitFlip Free?

- **Appears at ISCA 2023**

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

Haocong Luo Ataberk Olgun A. Giray Yağlıkçı Yahya Can Tuğrul Steve Rhyner
Meryem Banu Cavlak Joël Lindegger Mohammad Sadrosadati Onur Mutlu
ETH Zürich



- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tugrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu, **"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"**

Proceedings of the 50th International Symposium on Computer Architecture (ISCA), Orlando, FL, USA, June 2023.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Video](#) (3 minutes)]

[[RowPress Source Code and Datasets \(Officially Artifact Evaluated with All Badges\)](#)]

***Officially artifact evaluated as available, reusable and reproducible.
Best artifact award at ISCA 2023. IEEE Micro Top Pick.***

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

Haocong Luo Ataberk Olgun A. Giray Yağlıkçı Yahya Can Tuğrul Steve Rhyner
Meryem Banu Cavlak Joël Lindegger Mohammad Sadrosadati Onur Mutlu

ETH Zürich

A Recent RowHammer Lecture

What Is RowHammer?

- One can **predictably induce bit flips** in commodity DRAM chips
 - >80% of the tested DRAM chips are vulnerable
- First example of how a **simple hardware failure mechanism** can create a **widespread system security vulnerability**

WIRED Forget Software—Now Hackers Are Exploiting Physics

BUSINESS CULTURE DESIGN GEAR SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

Stanford

Stanford Seminar - RowHammer, RowPress and Beyond: Can We Be Free of Bitflips (Soon)?

Stanford Online 529K subscribers

Subscribed

35

Share

Download

Save

1.2K views 1 month ago

Intel Optane Persistent Memory (2019)

- Non-volatile main memory
- Based on 3D-XPoint Technology



Emerging Memories Also Need Intelligent Controllers

- Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger,
"Architecting Phase Change Memory as a Scalable DRAM Alternative"
Proceedings of the 36th International Symposium on Computer Architecture (ISCA), pages 2-13, Austin, TX, June 2009. [Slides \(pdf\)](#)
One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight. 2022 Persistent Impact Prize.

Architecting Phase Change Memory as a Scalable DRAM Alternative

Benjamin C. Lee[†] Engin Ipek[†] Onur Mutlu[‡] Doug Burger[†]

[†]Computer Architecture Group
Microsoft Research
Redmond, WA
{blee, ipek, dburger}@microsoft.com

[‡]Computer Architecture Laboratory
Carnegie Mellon University
Pittsburgh, PA
onur@cmu.edu

Intelligent
Memory Controllers
Can Avoid Many Failures
& Enable Better Scaling

Flash Cosmos

Key Ideas of Flash-Cosmos



Multi-Wordline Sensing (MWS)

to enable in-flash bulk bitwise operations
via a single sensing operation



Enhanced SLC-Mode Programming (ESP)

to eliminate raw bit errors in stored data
(and thus in computation results)

Multi-Wordline Sensing (MWS): Bitwise AND

- Intra-Block MWS:

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs

A bitline reads as '1' only when all the target cells store '1'

→ Equivalent to the bitwise AND of all the target cells

Operate
as a *resistance* (1)
or an *open switch* (0)

WL₂

WL₃

WL₄

BL₁

BL₂

BL₃

BL₄

Result: 0

0

0

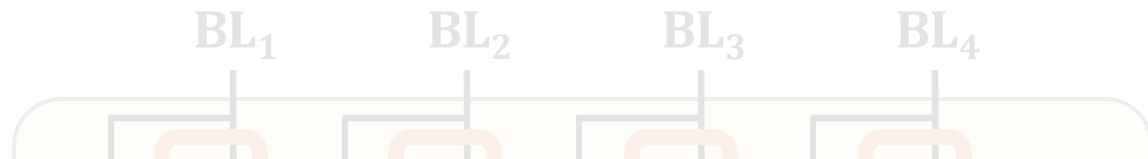
0

Multi-Wordline Sensing (MWS): Bitwise AND

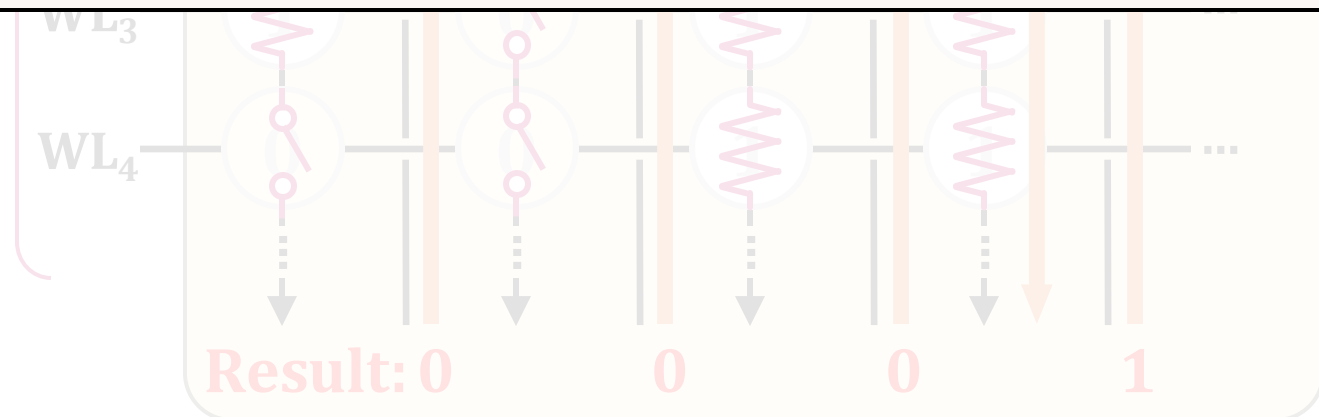
- Intra-Block MWS:

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs



Flash-Cosmos (Intra-Block MWS) enables bitwise AND of multiple pages in the same block via a single sensing operation



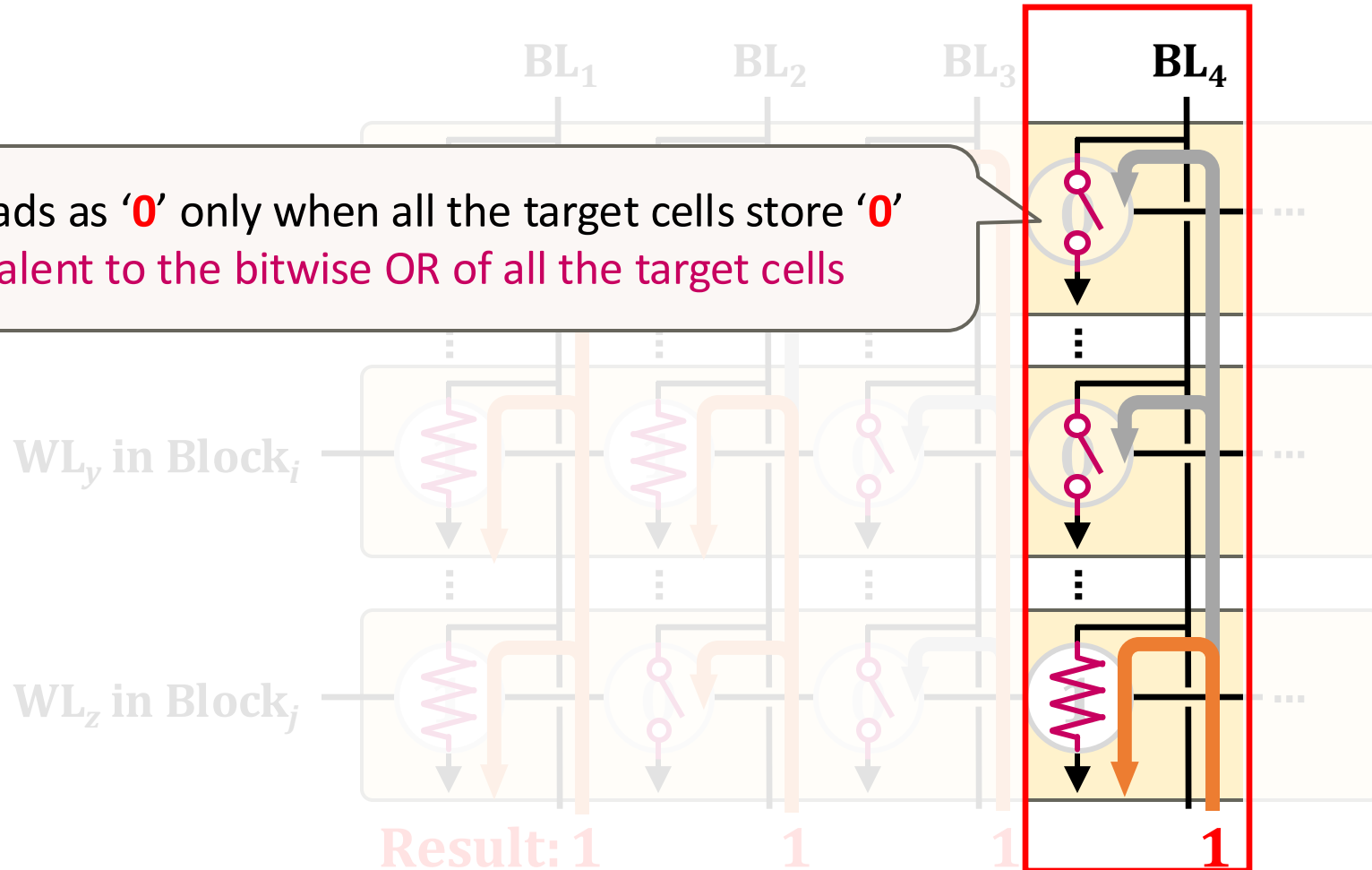
Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:**

Simultaneously activates multiple WLs in different blocks

→ **Bitwise OR** of the stored data in the WLs

A bitline reads as '**0**' only when all the target cells store '**0**'
→ Equivalent to the bitwise OR of all the target cells



Multi-Wordline Sensing (MWS): Bitwise OR

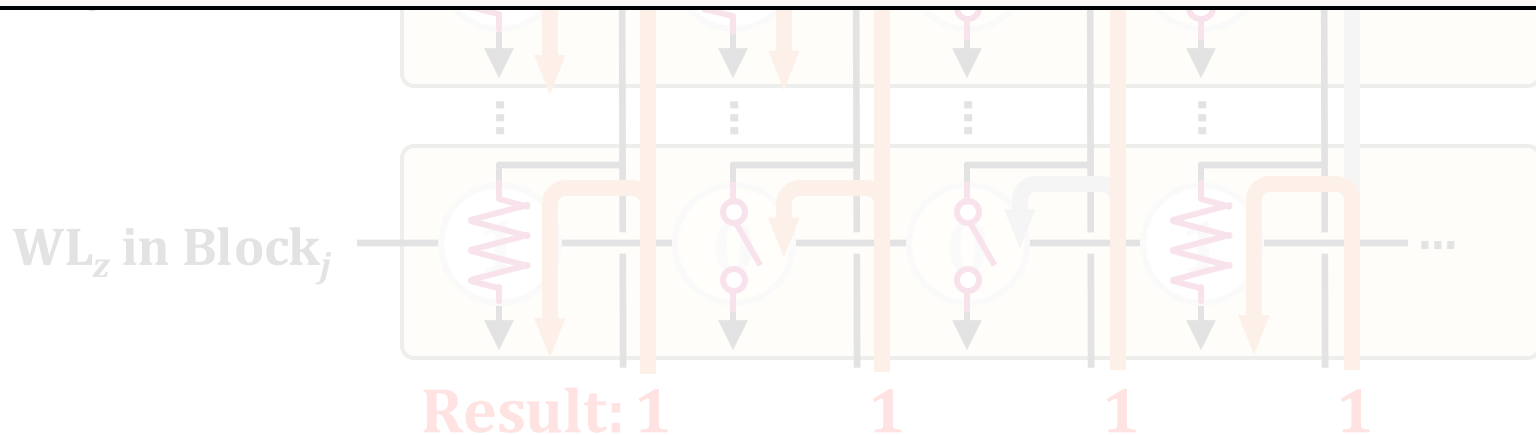
- Inter-Block MWS:

Simultaneously activates multiple WLs in different blocks

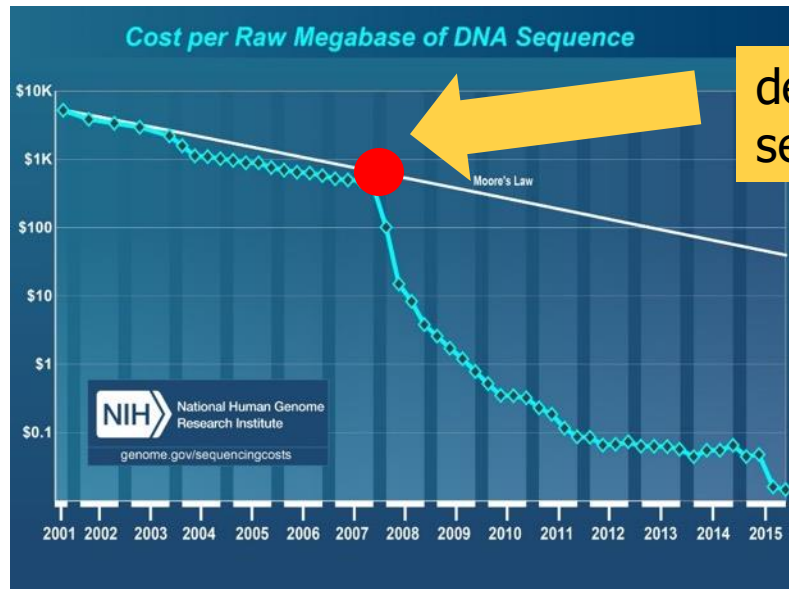
→ Bitwise OR of the stored data in the WLs



Flash-Cosmos (Inter-Block MWS) enables
bitwise OR of multiple pages in different blocks
via a single sensing operation



Data is Key for Future Workloads

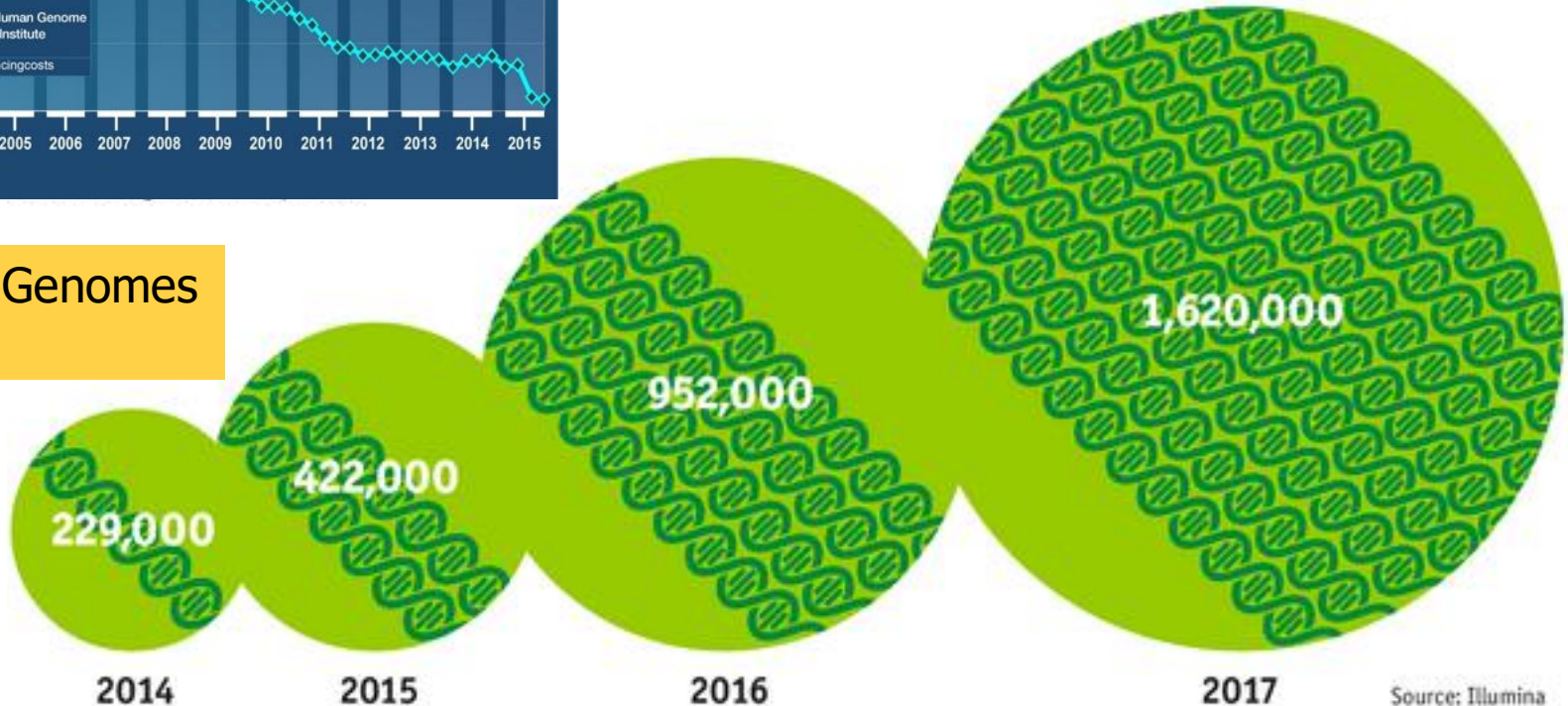


development of new sequencing technology



Oxford Nanopore MinION

Number of Genomes Sequenced



The Economist

New Genome Sequencing Technologies

Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

Damla Senol Cali ✉, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

Briefings in Bioinformatics, bby017, <https://doi.org/10.1093/bib/bby017>

Published: 02 April 2018 **Article history** ▼



Oxford Nanopore MinION

Senol Cali+, “**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**,” *Briefings in Bioinformatics*, 2018.

[\[Open arxiv.org version\]](#)

End of Backup Slides