

# Memory-Centric Computing

Onur Mutlu

[omutlu@gmail.com](mailto:omutlu@gmail.com)

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

24 October 2022

Sonova Internal Workshop

**SAFARI**

**ETH** zürich

**Carnegie Mellon**

Computing

is Bottlenecked by Data



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

---

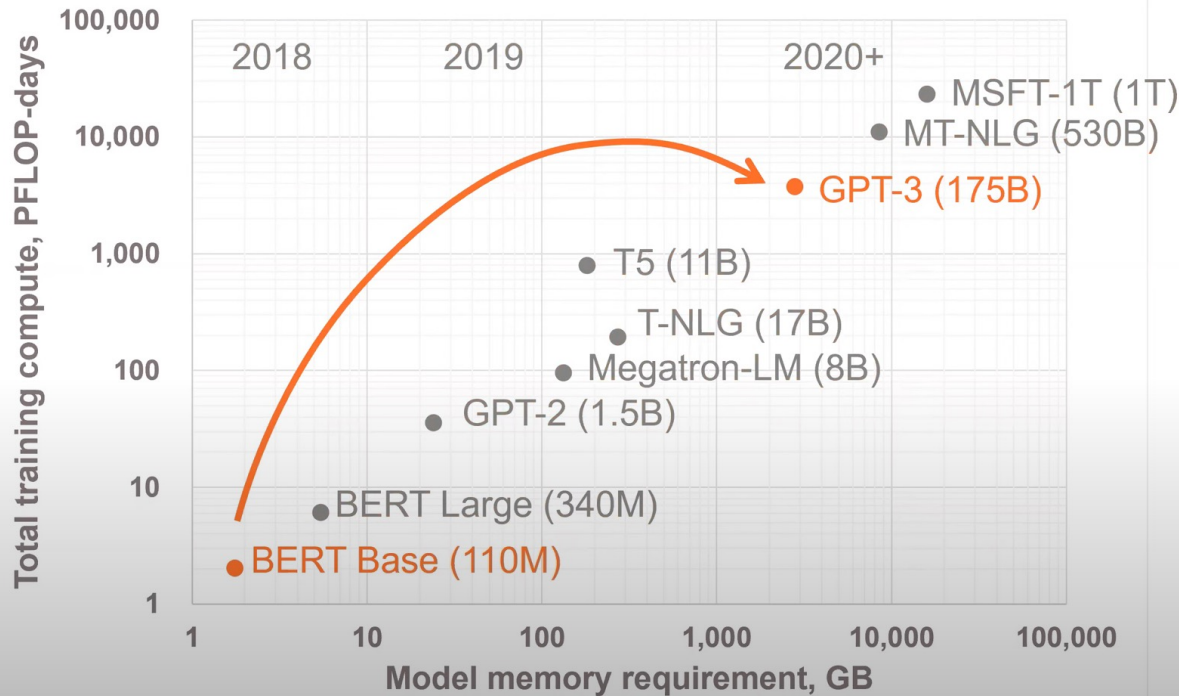
- Important workloads are all data intensive
- They require rapid and efficient processing of large amounts of data
- Data is increasing
  - We can generate more than we can process

# 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

# Data is Key for Future Workloads

---



## In-memory Databases

[Mao+, EuroSys'12;  
Clapp+ (Intel), IISWC'15]



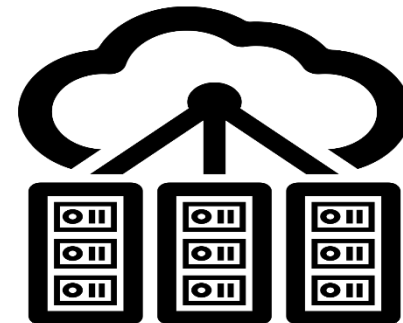
## In-Memory Data Analytics

[Clapp+ (Intel), IISWC'15;  
Awan+, BDCloud'15]



## Graph/Tree Processing

[Xu+, IISWC'12; Umuroglu+, FPL'15]



## Datacenter Workloads

[Kanev+ (Google), ISCA'15]

# Data Overwhelms Modern Machines

---



**In-memory Databases**



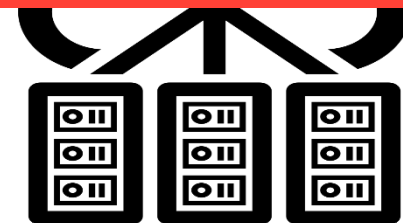
**Graph/Tree Processing**

**Data → performance & energy bottleneck**



**In-Memory Data Analytics**

[Clapp+ (Intel), IISWC'15;  
Awan+, BDCloud'15]



**Datacenter Workloads**

[Kanev+ (Google), ISCA'15]

# Data is Key for Future Workloads



**Chrome**

Google's web browser



**TensorFlow Mobile**

Google's machine learning  
framework



**Video Playback**

Google's **video codec**



**Video Capture**

Google's **video codec**

# Data Overwhelms Modern Machines



**Chrome**



**TensorFlow Mobile**

Data → performance & energy bottleneck

**VP9**



**Video Playback**

Google's **video codec**

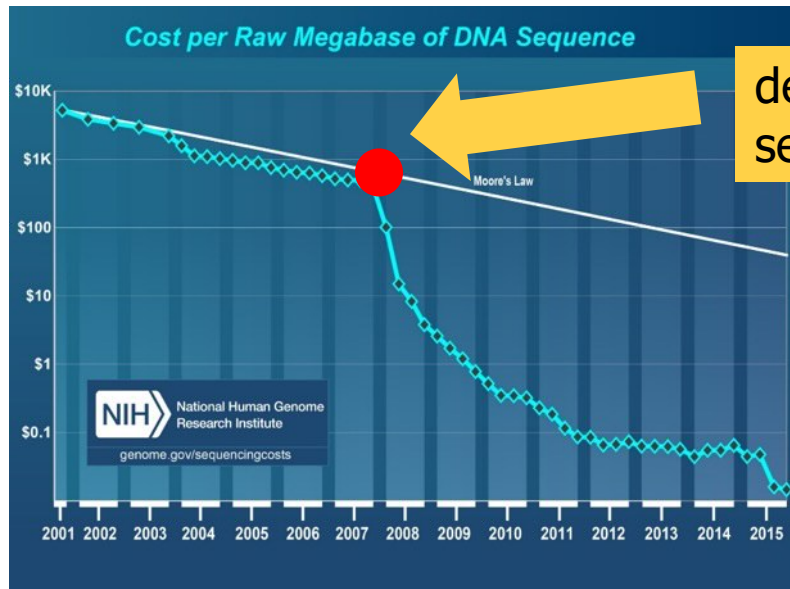
**VP9**



**Video Capture**

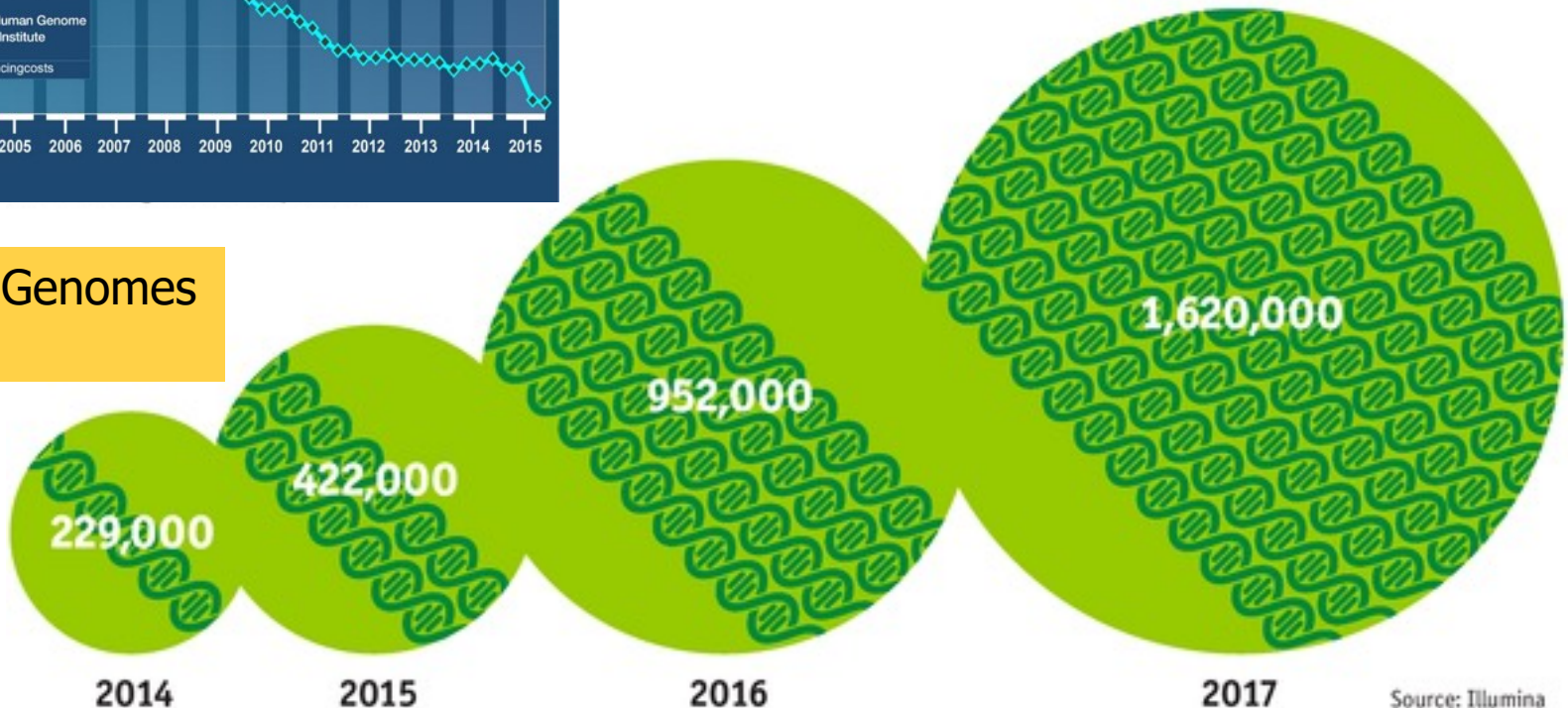
Google's **video codec**

# Data is Key for Future Workloads



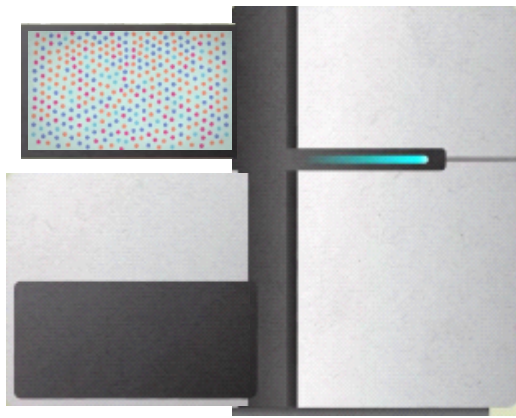
development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced



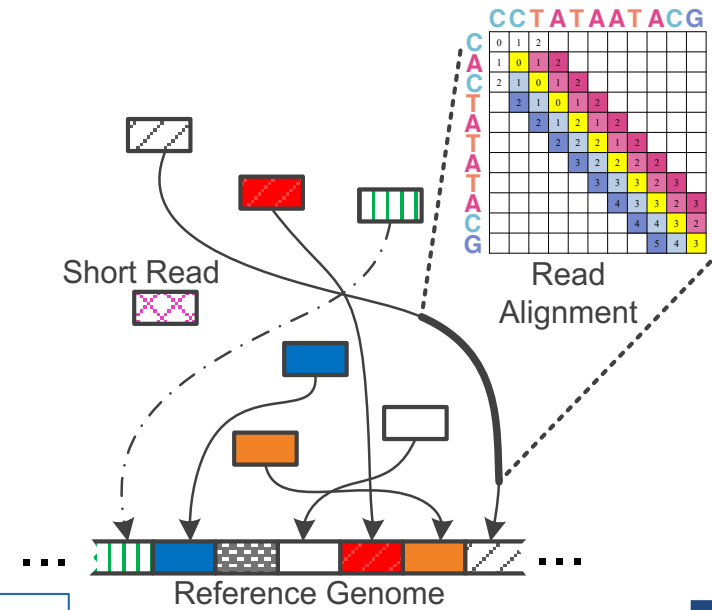
The Economist





Billions of Short Reads

ATATATACGTACTAGTACGT  
 TTTAGTACGTACGT  
 ATACGTACTAGTACGT  
 CGCCCCTACGTA  
 ACGTACTAGTACGT  
 TTAGTACGTACGT  
 TACGTACTAAAGTACGT  
 TACGTACTAGTACGT  
 TTTAAACGTA  
 CGTACTAGTACGT  
 GGGAGTACGTACGT



## 1 Sequencing

# Genome Analysis

## 2 Read Mapping

Data → performance & energy bottleneck

read4: CGCTTCCAT  
 read5: CCATGACGC  
 read6: TTCCATGAC



## 3 Variant Calling

## 4 Scientific Discovery



# We Need Faster & Scalable Genome Analysis



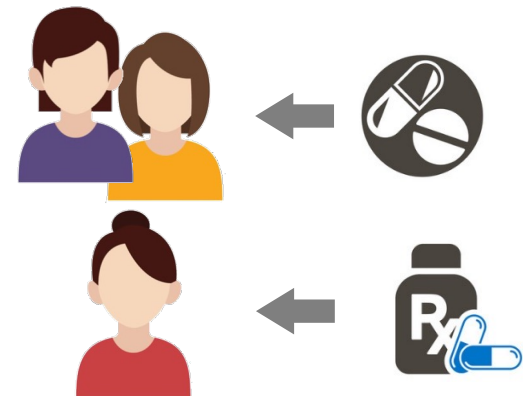
Understanding **genetic variations**,  
**species**, **evolution**, ...



Predicting the **presence** and **relative abundance** of **microbes** in a sample



Rapid surveillance of **disease outbreaks**



Developing **personalized medicine**

# 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](#)]

# 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** ▼

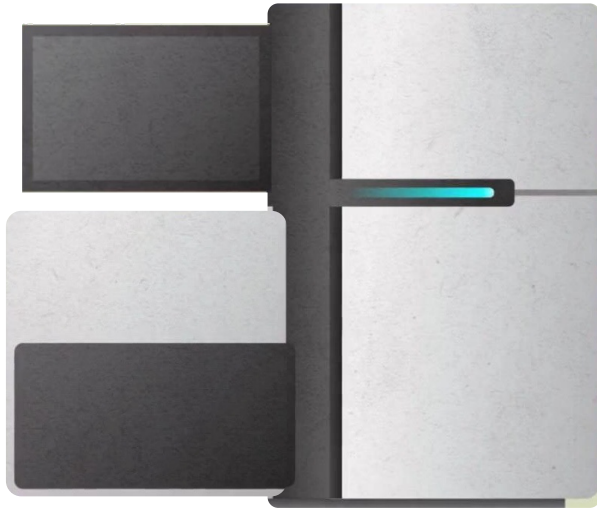


Oxford Nanopore MinION

Data → performance & energy bottleneck

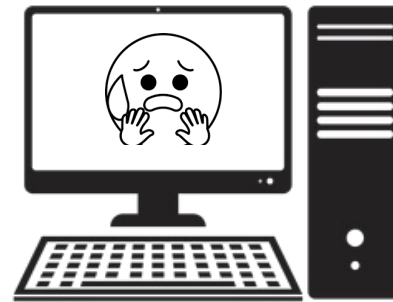
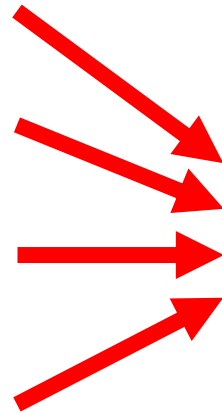
# One Problem with (Genome) Analysis Today

---



**Special-Purpose** Machine  
for **Data Generation**

**FAST**



**General-Purpose** Machine  
for **Data Analysis**

**SLOW**

**Slow and inefficient processing capability**

---

**SAFARI** This picture is similar for many “data generators & analyzers” today

# Accelerating Genome Analysis [IEEE MICRO 2020]

---

- Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,  
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#)  
[IEEE Micro \(IEEE MICRO\)](#), Vol. 40, No. 5, pages 65-75, September/October 2020.  
[\[Slides \(pptx\)\(pdf\)\]](#)  
[\[Talk Video \(1 hour 2 minutes\)\]](#)

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

**Mohammed Alser**

ETH Zürich

**Zülal Bingöl**

Bilkent University

**Damla Senol Cali**

Carnegie Mellon University

**Jeremie Kim**

ETH Zurich and Carnegie Mellon University

**Saugata Ghose**

University of Illinois at Urbana–Champaign and  
Carnegie Mellon University

**Can Alkan**

Bilkent University

**Onur Mutlu**

ETH Zurich, Carnegie Mellon University, and  
Bilkent University



# Beginner Reading on Genome Analysis

Mohammed Alser, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu

**"From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis"**

Computational and Structural Biotechnology Journal, 2022

[[Source code](#)]



ELSEVIER

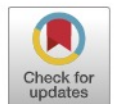


journal homepage: [www.elsevier.com/locate/csbj](http://www.elsevier.com/locate/csbj)



Review

From molecules to genomic variations: Accelerating genome analysis via intelligent algorithms and architectures



Mohammed Alser\*, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu\*

ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland

**SAFARI**

<https://arxiv.org/pdf/2205.07957.pdf>

# FPGA-based Near-Memory Analytics

---

- 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<sup>◇</sup> Mohammed Alser<sup>◇</sup> Damla Senol Cali<sup>✕</sup>

Dionysios Diamantopoulos<sup>▽</sup> Juan Gómez-Luna<sup>◇</sup>

Henk Corporaal<sup>★</sup> Onur Mutlu<sup>◇✕</sup>

<sup>◇</sup>*ETH Zürich*    <sup>✕</sup>*Carnegie Mellon University*

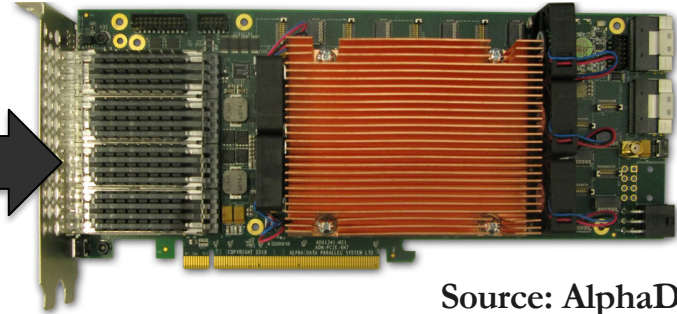
<sup>★</sup>*Eindhoven University of Technology*    <sup>▽</sup>*IBM Research Europe*

# Near-Memory Acceleration using FPGAs



Source: IBM

IBM POWER9 CPU



Source: AlphaData

HBM-based FPGA board

## Near-HBM FPGA-based accelerator

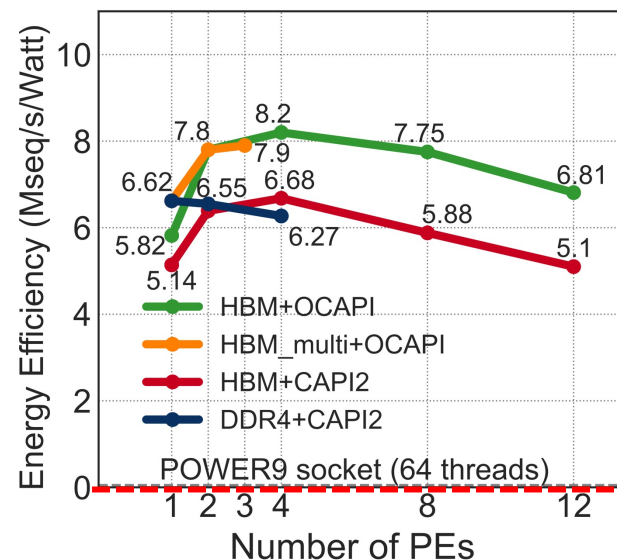
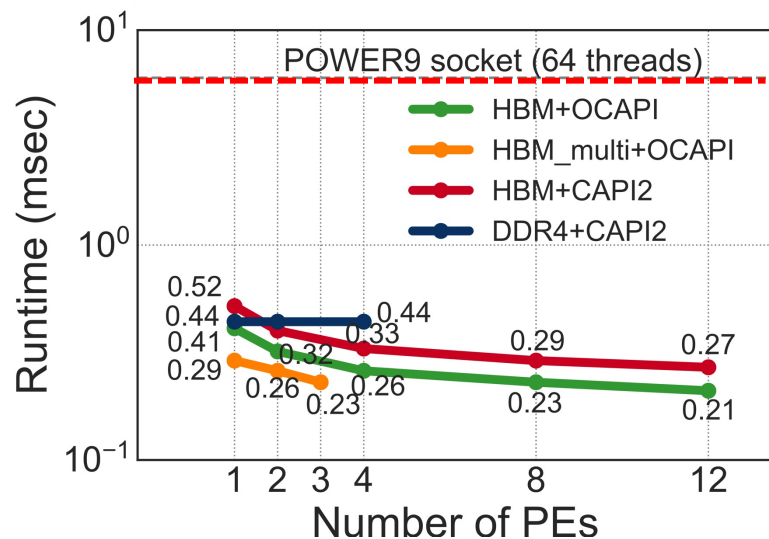
**Two communication technologies:** CAPI2 and OCAPI

**Two memory technologies:** DDR4 and HBM

**Two workloads:** Weather Modeling and Genome Analysis



# Performance & Energy Greatly Improve



**5-27× performance** vs. a 16-core (64-thread) IBM POWER9 CPU

**12-133× energy efficiency** vs. a 16-core (64-thread) IBM POWER9 CPU

**HBM alleviates memory bandwidth contention vs. DDR4**

# GenASM Framework [MICRO 2020]

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

# In-Storage Genome 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.*

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

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

[[Lightning Talk Video](#) (90 seconds)]

[[Talk Video](#) (17 minutes)]

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

Nika Mansouri Ghiasi<sup>1</sup> Jisung Park<sup>1</sup> Harun Mustafa<sup>1</sup> Jeremie Kim<sup>1</sup> Ataberk Olgun<sup>1</sup>  
Arvid Gollwitzer<sup>1</sup> Damla Senol Cali<sup>2</sup> Can Firtina<sup>1</sup> Haiyu Mao<sup>1</sup> Nour Almadhoun Alserr<sup>1</sup>  
Rachata Ausavarungnirun<sup>3</sup> Nandita Vijaykumar<sup>4</sup> Mohammed Alser<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich <sup>2</sup>Bionano Genomics <sup>3</sup>KMUTNB <sup>4</sup>University of Toronto

# 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 Cali<sup>1</sup> Konstantinos Kanellopoulos<sup>2</sup> Joël Lindegger<sup>2</sup> Zülal Bingöl<sup>3</sup>  
Gurpreet S. Kalsi<sup>4</sup> Ziyi Zuo<sup>5</sup> Can Firtina<sup>2</sup> Meryem Banu Cavlak<sup>2</sup> Jeremie Kim<sup>2</sup>  
Nika Mansouri Ghiasi<sup>2</sup> Gagandeep Singh<sup>2</sup> Juan Gómez-Luna<sup>2</sup> Nour Almadhoun Alserr<sup>2</sup>  
Mohammed Alser<sup>2</sup> Sreenivas Subramoney<sup>4</sup> Can Alkan<sup>3</sup> Saugata Ghose<sup>6</sup> Onur Mutlu<sup>2</sup>

<sup>1</sup>Bionano Genomics <sup>2</sup>ETH Zürich <sup>3</sup>Bilkent University <sup>4</sup>Intel Labs

<sup>5</sup>Carnegie Mellon University <sup>6</sup>University of Illinois Urbana-Champaign

# Future of Genome Sequencing & Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu  
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#) IEEE Micro, August 2020.



MinION from ONT

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

Sept.-Oct. 2020, pp. 65-75, vol. 40

DOI Bookmark: [10.1109/MM.2020.3013728](https://doi.org/10.1109/MM.2020.3013728)

## FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications

July-Aug. 2021, pp. 39-48, vol. 41

DOI Bookmark: [10.1109/MM.2021.3088396](https://doi.org/10.1109/MM.2021.3088396)



SmidgION from ONT



# More on Fast & Efficient Genome Analysis ...

- Onur Mutlu,  
**"Accelerating Genome Analysis: A Primer on an Ongoing Journey"**  
*Invited Lecture at [Technion](#), Virtual, 26 January 2021.*  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (1 hour 37 minutes, including Q&A)]  
[[Related Invited Paper \(at IEEE Micro, 2020\)](#)]



Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

740 views • Premiered Feb 6, 2021

35 0 SHARE SAVE ...

**SAFARI**



Onur Mutlu Lectures  
15.9K subscribers

<https://www.youtube.com/watch?v=r7sn41IH-4A>

ANALYTICS

EDIT VIDEO

# Detailed Lectures on Genome Analysis

---

- **Computer Architecture, Fall 2020, Lecture 3a**
  - **Introduction to Genome Sequence Analysis** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5>
- **Computer Architecture, Fall 2020, Lecture 8**
  - **Intelligent Genome Analysis** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14>
- **Computer Architecture, Fall 2020, Lecture 9a**
  - **GenASM: Approx. String Matching Accelerator** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15>
- **Accelerating Genomics Project Course, Fall 2020, Lecture 1**
  - **Accelerating Genomics** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId>



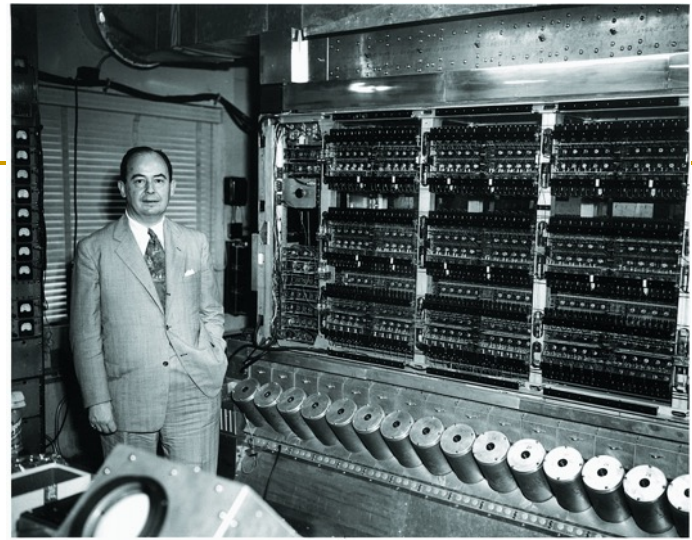
# Data Overwhelms Modern Machines ...

---

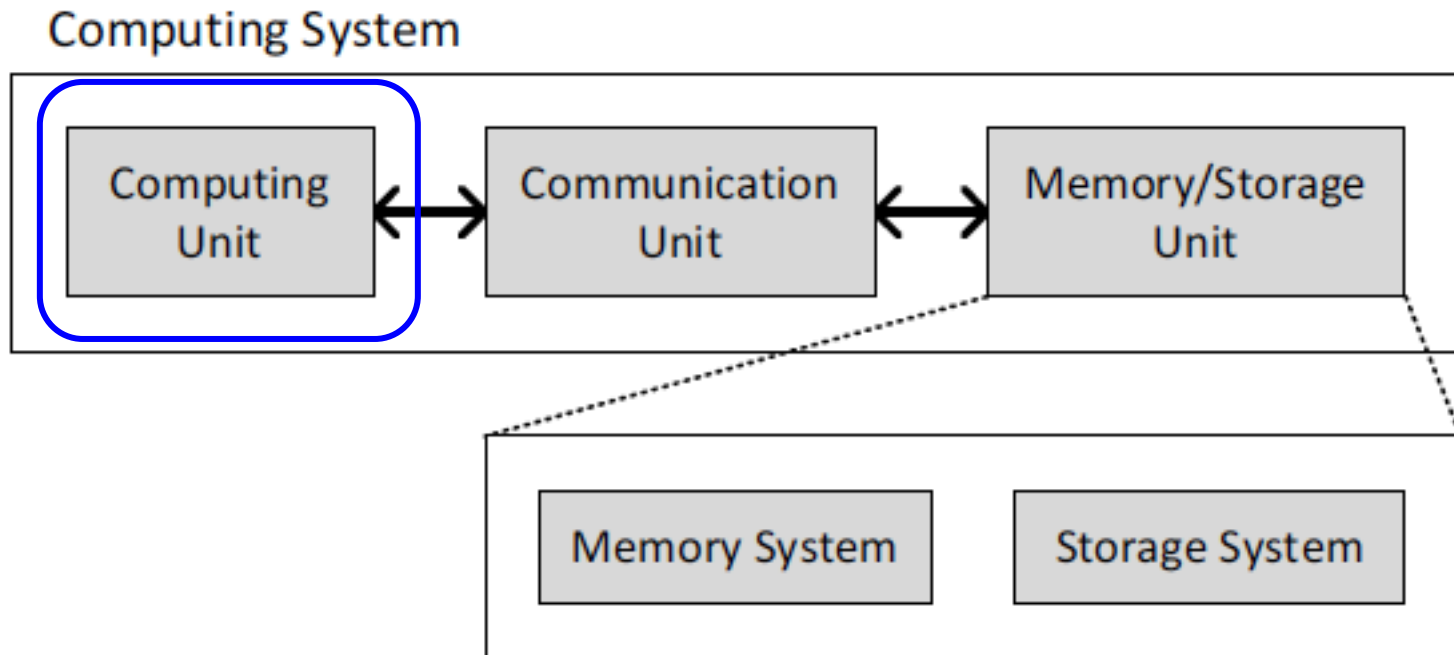
- Storage/memory capability
- Communication capability
- Computation capability
- Greatly impacts robustness, energy, performance, cost

# A Computing System

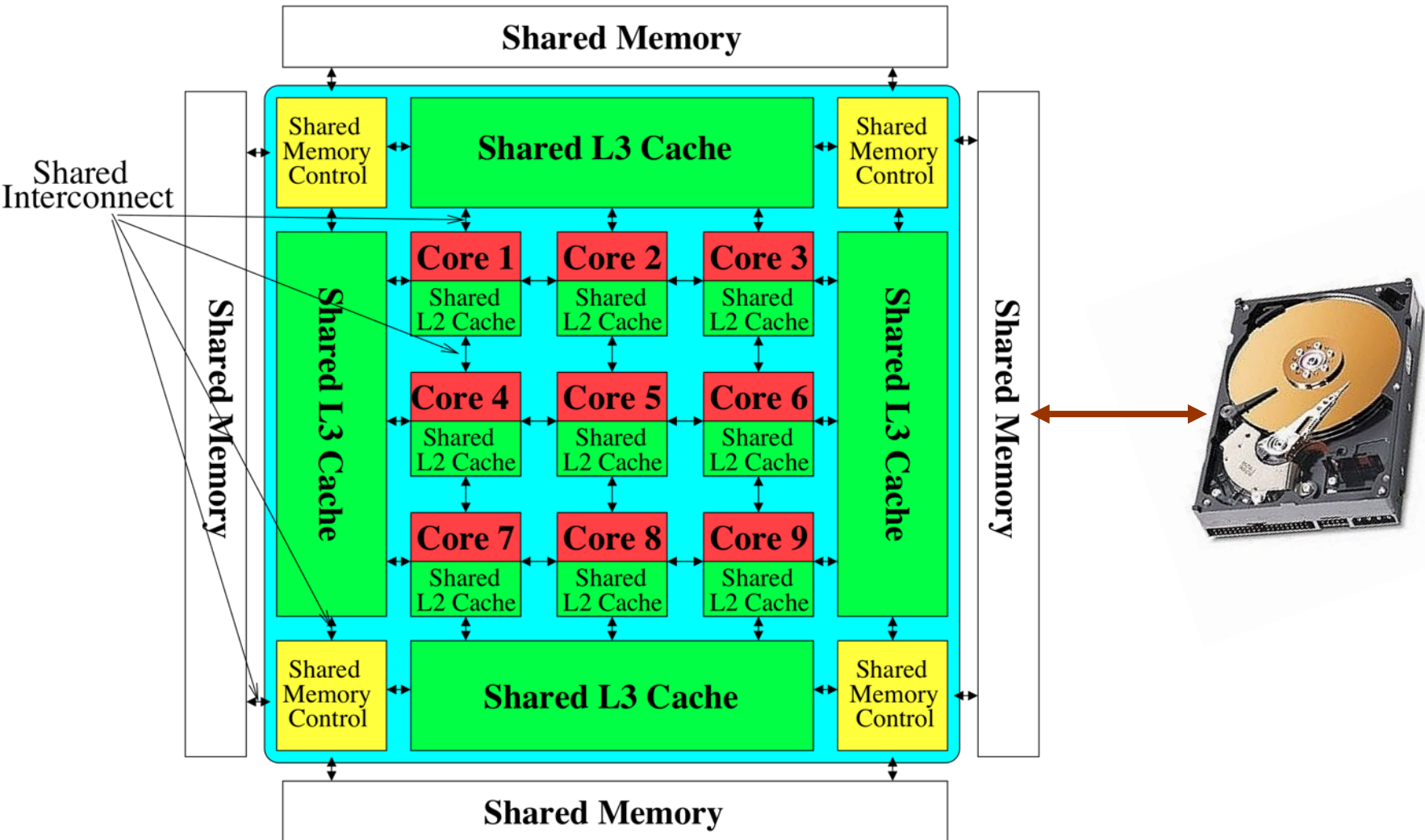
- Three key components
- Computation
- Communication
- Storage/memory



Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.



# Perils of Processor-Centric Design

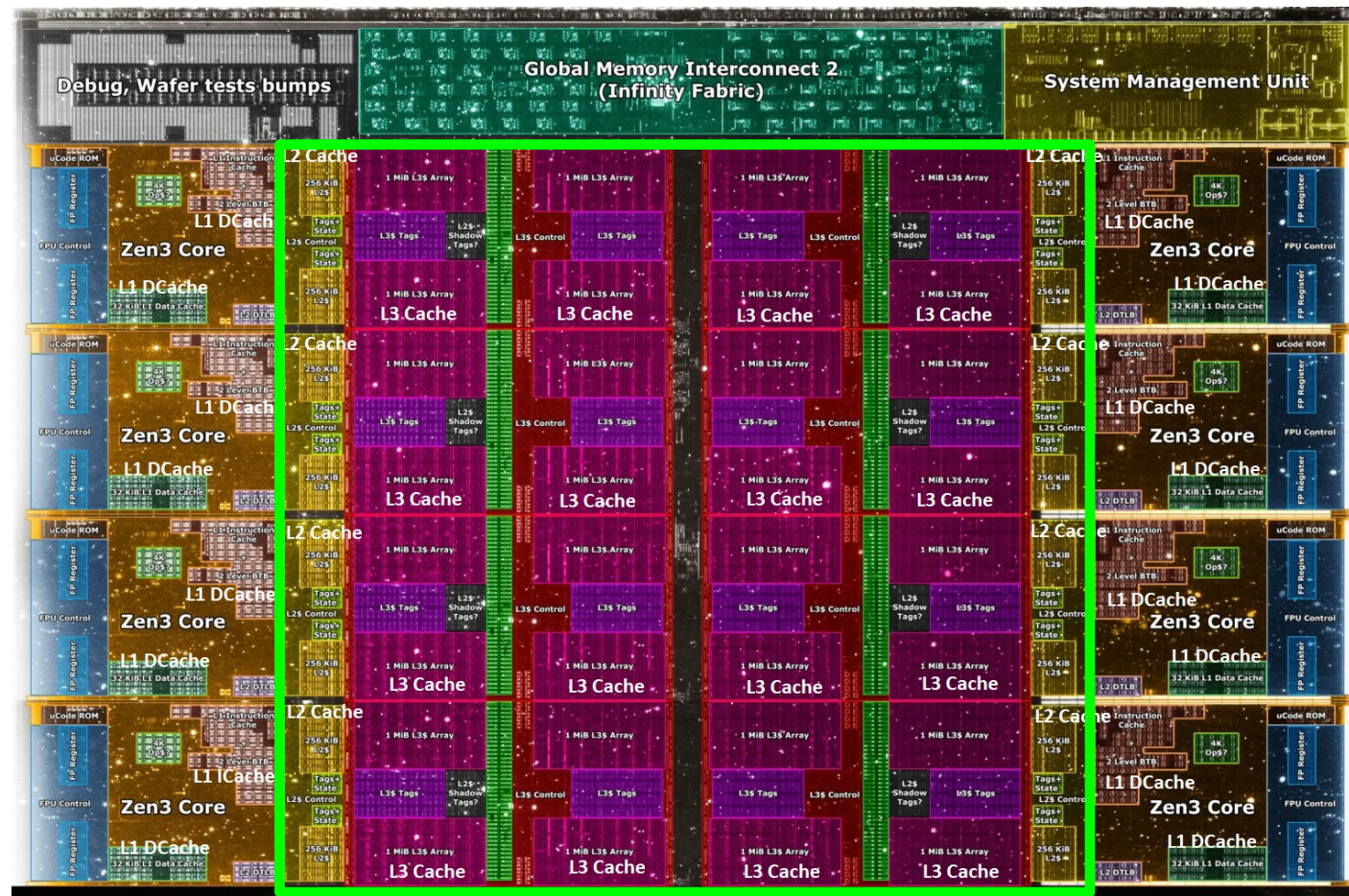


**Most of the system is dedicated to storing and moving data**

**Yet, system is still bottlenecked by memory**



# Deeper and Larger Memory Hierarchies



## Core Count:

8 cores/16 threads

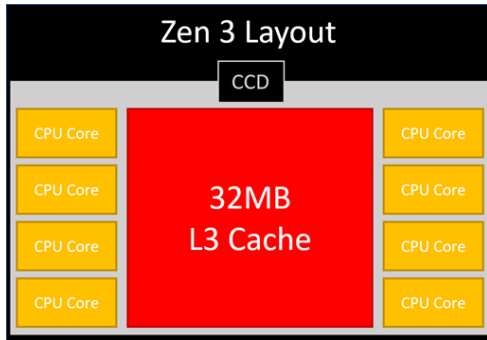
L1 Caches:  
32 KB per core

L2 Caches:  
512 KB per core

L3 Cache:  
32 MB shared

# AMD Ryzen 5000, 2020

# AMD's 3D Last Level Cache (2021)

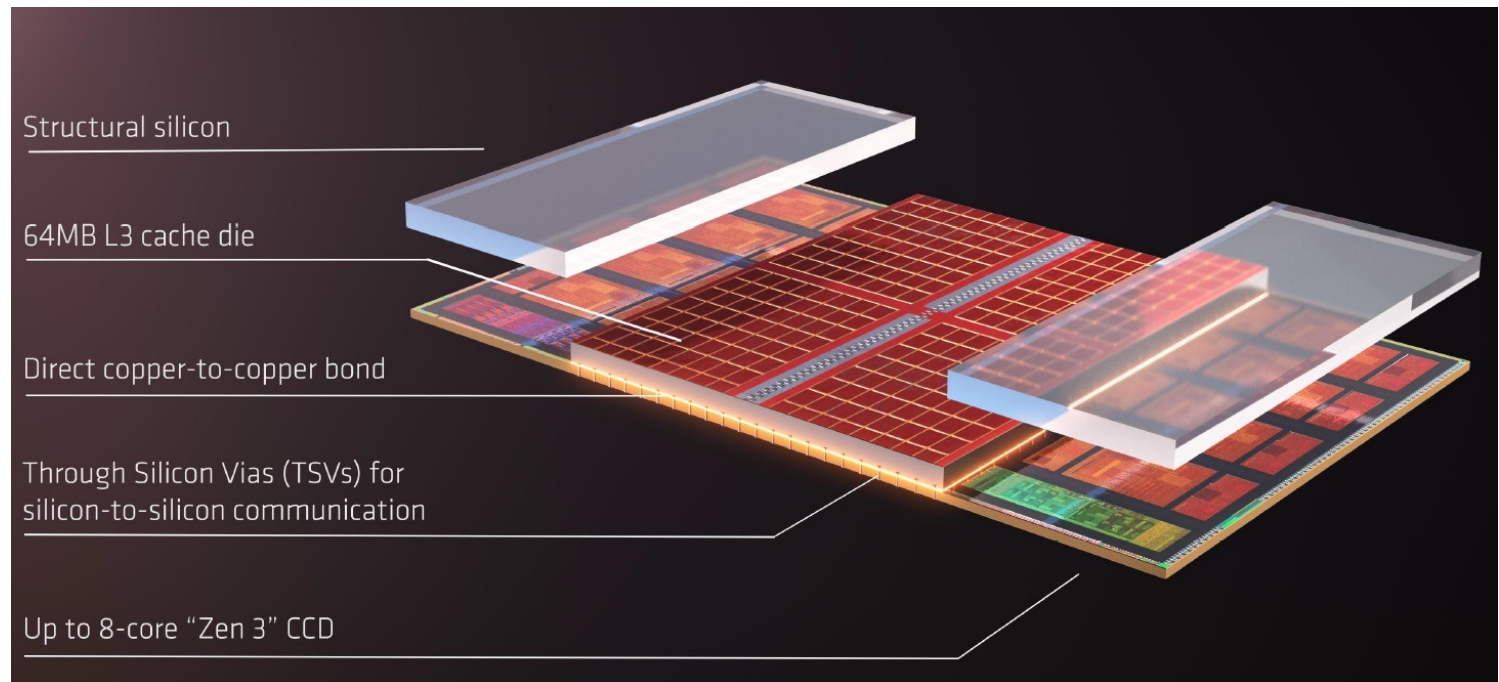


<https://community.microcenter.com/discussion/5134/comparing-zen-3-to-zen-2>

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

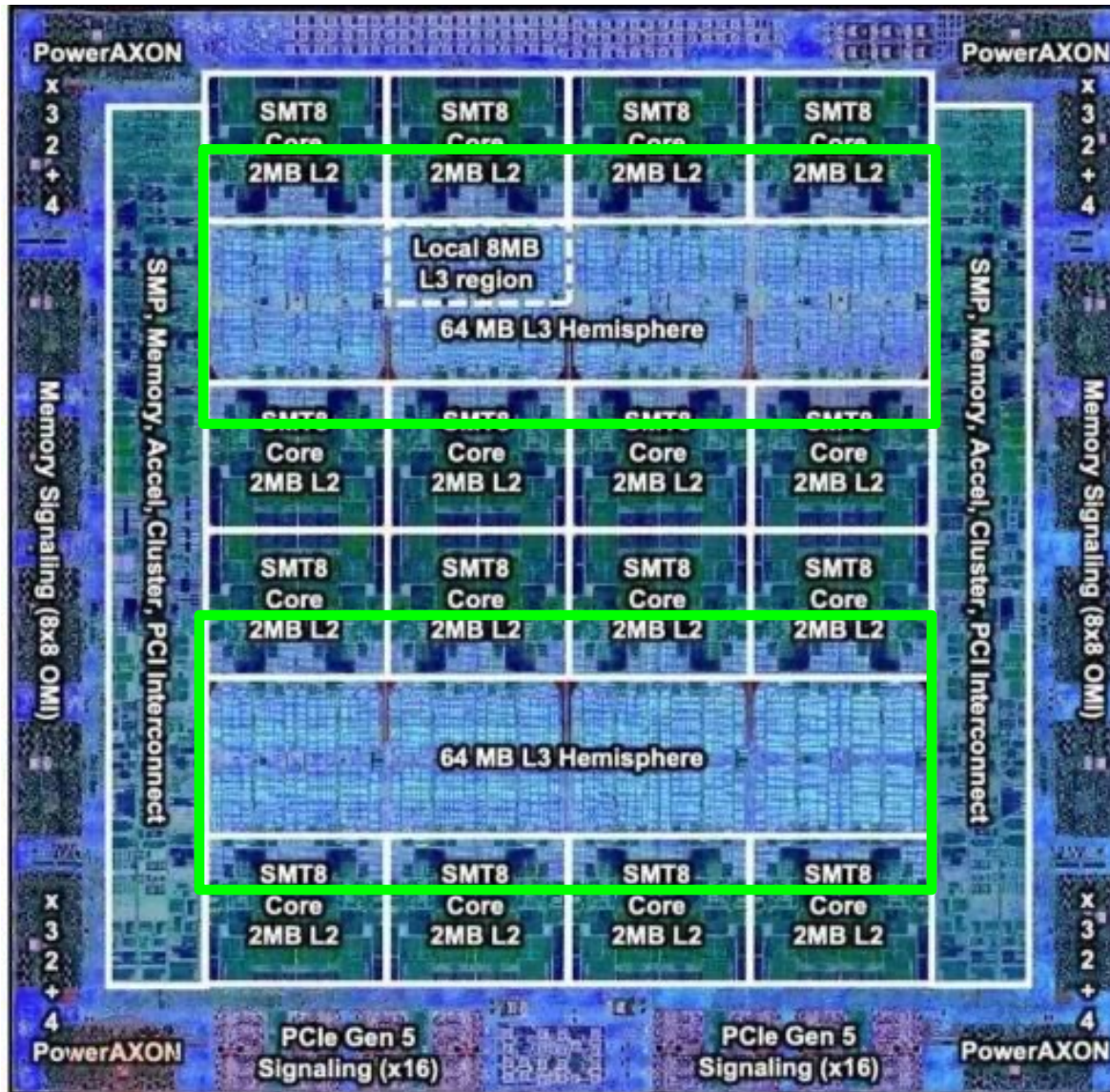
**Additional 64 MB L3 cache die**  
**stacked on top of the processor die**

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





# Deeper and Larger Memory Hierarchies



IBM POWER10,  
2020

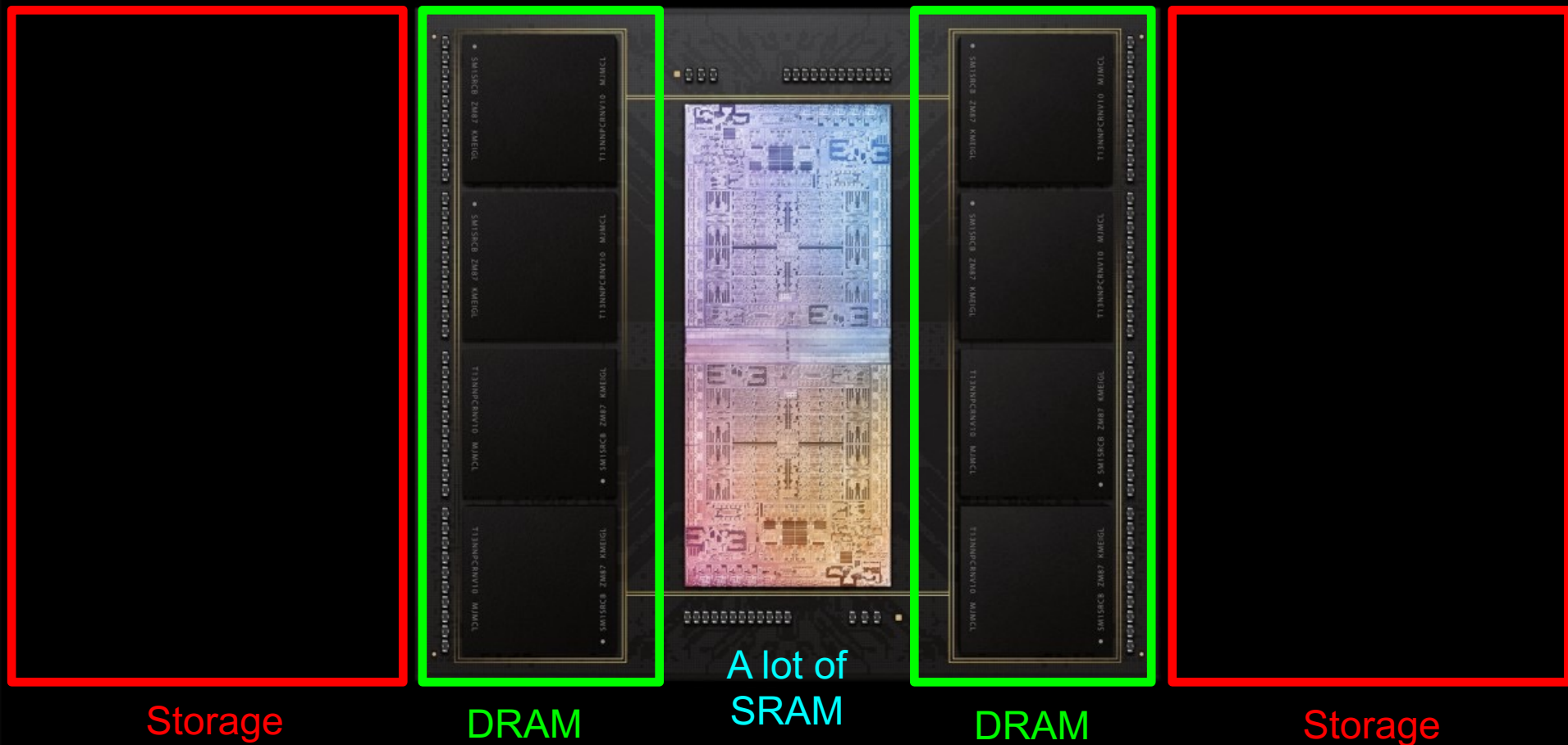
Cores:  
15-16 cores,  
8 threads/core

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



**Chrome**



**TensorFlow Mobile**

Data → performance & energy bottleneck

**VP9**



**Video Playback**

Google's **video codec**

**VP9**



**Video Capture**

Google's **video codec**

# Data Movement Overwhelms Modern Machines

- 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 Boroumand<sup>1</sup>

Saugata Ghose<sup>1</sup>

Youngsok Kim<sup>2</sup>

Rachata Ausavarungnirun<sup>1</sup>

Eric Shiu<sup>3</sup>

Rahul Thakur<sup>3</sup>

Daehyun Kim<sup>4,3</sup>

Aki Kuusela<sup>3</sup>

Allan Knies<sup>3</sup>

Parthasarathy Ranganathan<sup>3</sup>

Onur Mutlu<sup>5,1</sup>

## An Intelligent Architecture Handles Data Well

# How to Handle Data Well

---

- **Ensure data does not overwhelm** the components
  - via intelligent algorithms, architectures & system designs: algorithm-architecture-devices
- **Take advantage of** vast amounts of **data** and metadata
  - to improve architectural & system-level decisions
- **Understand and exploit** properties of (different) **data**
  - to improve algorithms & architectures in various metrics



# Corollaries: Computing Systems Today ...

---

- Are **processor-centric** vs. **data-centric**
- Make **designer-dictated** decisions vs. **data-driven**
- Make **component-based myopic** decisions vs. **data-aware**

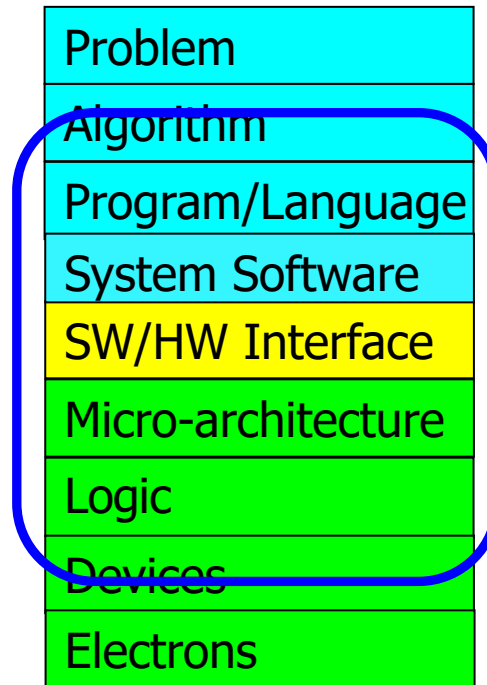
**Data-centric**

**Data-driven**

**Data-aware**

# We Need to Revisit the Entire Stack

---



**We can get there step by step**

# 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

# Data-Centric (Memory-Centric) Architectures

# Data-Centric Architectures: Properties

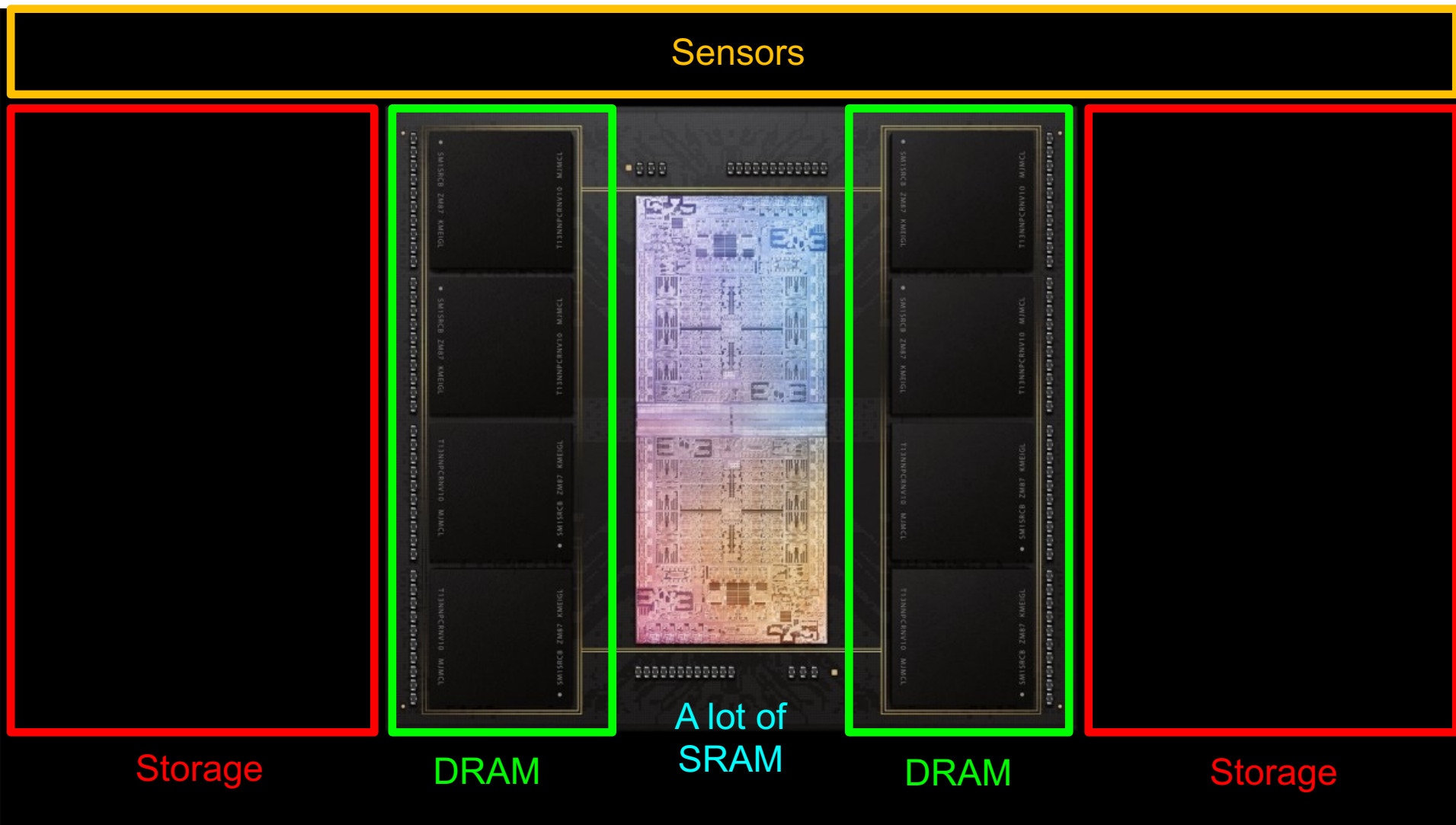
---

- **Process data where it resides** (where it makes sense)
  - Processing in and near memory & sensor structures
- **Low-latency & low-energy data access**
- **Low-cost data storage & processing**
  - High capacity memory at low cost: hybrid memory, compression
- **Intelligent data management**
  - Intelligent controllers handling robustness, security, cost, perf.



# Processing Data Where It Makes Sense

# Process Data Where It Makes Sense



Apple M1 Ultra System (2022)

# 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

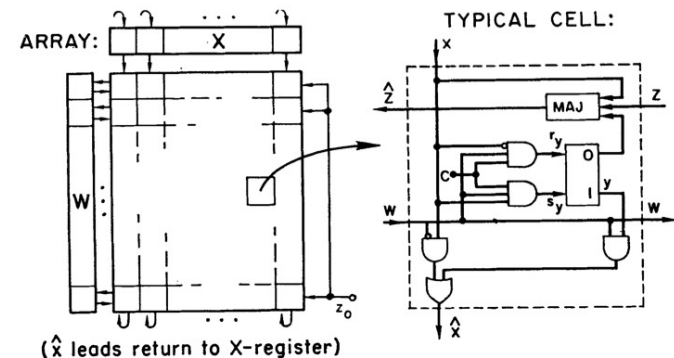
## Cellular Logic-in-Memory Arrays

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 problems with Memory Technology**

- ❑ Memory technology scaling is not going well (e.g., RowHammer)
- ❑ Many scaling issues demand intelligence in memory

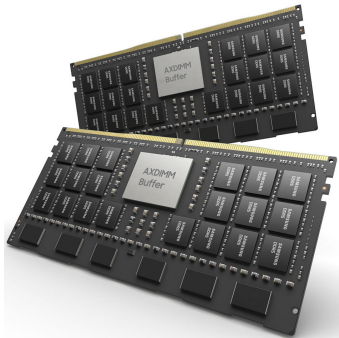
## ■ **Huge demand from Applications & Systems**

- ❑ 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

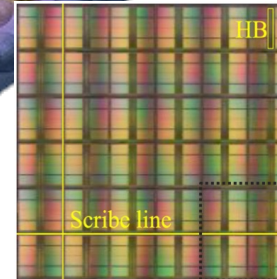
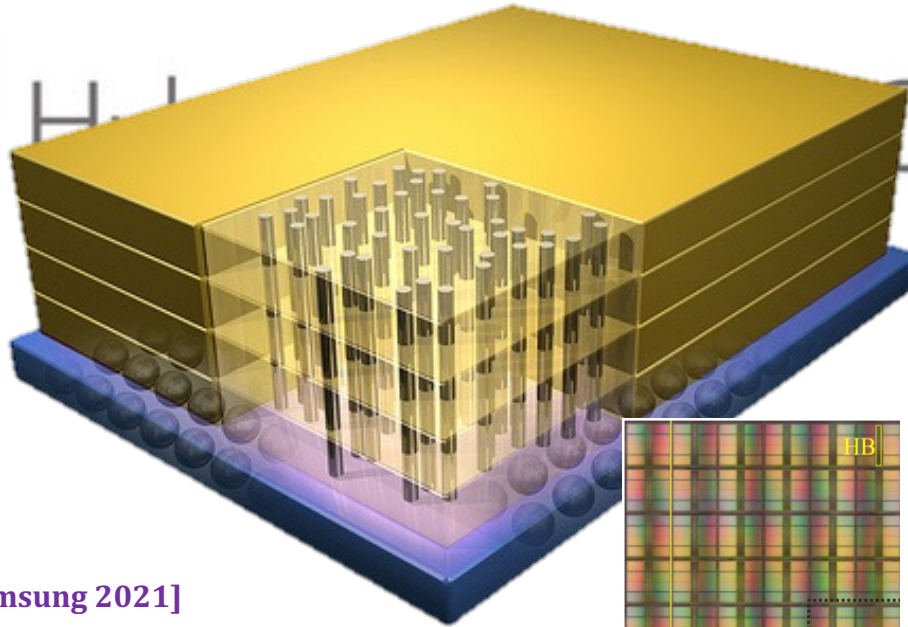
## ■ **Designs are squeezed in the middle**



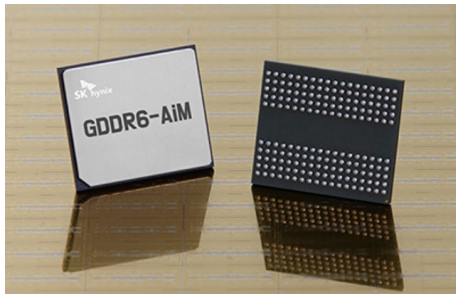
# Processing-in-Memory Landscape Today



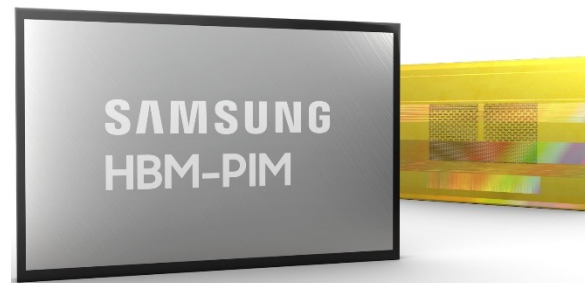
[Samsung 2021]



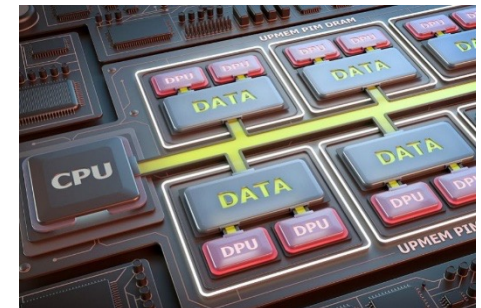
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]



[UPMEM 2019]



# Why In-Memory Computation Today?

---

- **Push from Technology**

- DRAM Scaling at jeopardy

- Controllers close to DRAM

- Industry open to new memory architectures

# Memory Scaling Issues **Are** Real

---

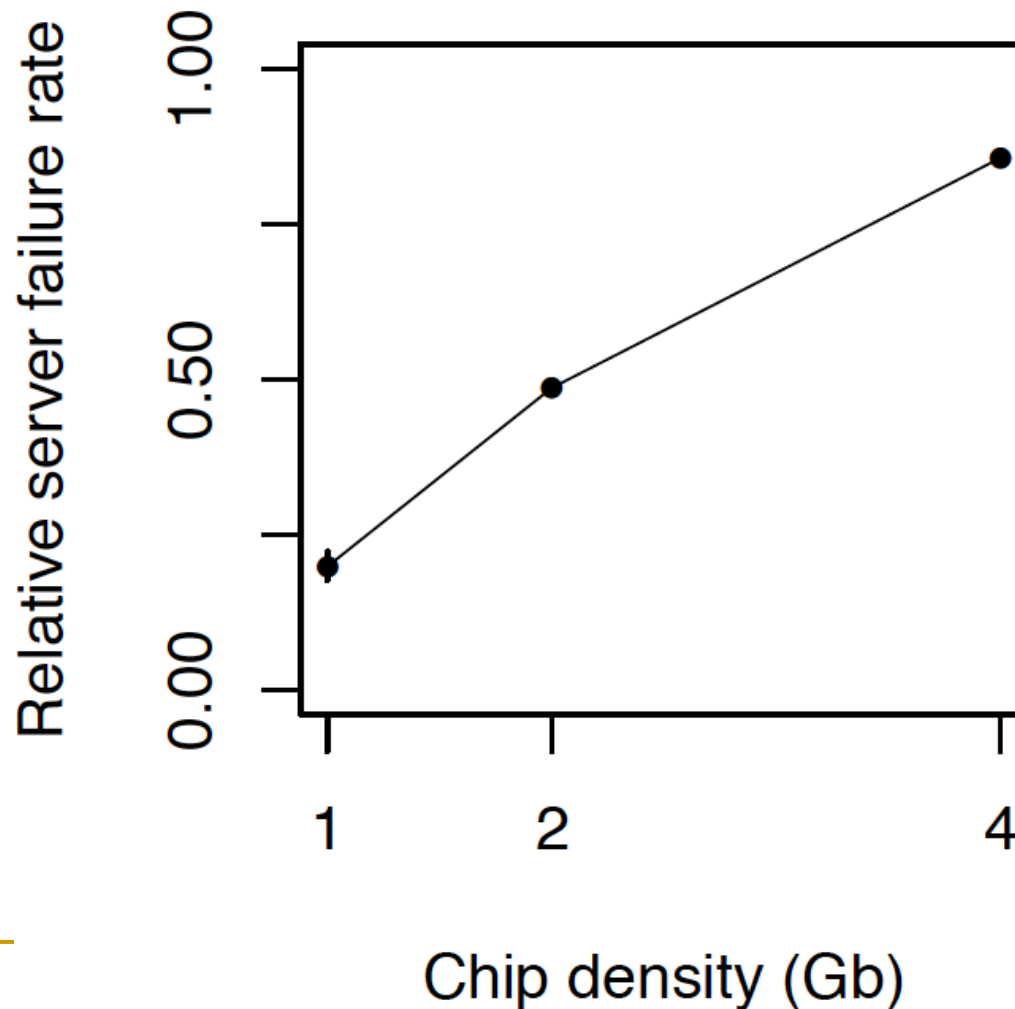
- Onur Mutlu,  
**"Memory Scaling: A Systems Architecture Perspective"**  
*Proceedings of the 5th International Memory Workshop (IMW)*, Monterey, CA, May 2013. Slides  
(pptx) (pdf)  
EETimes Reprint

## Memory Scaling: A Systems Architecture Perspective

Onur Mutlu  
Carnegie Mellon University  
onur@cmu.edu  
<http://users.ece.cmu.edu/~omutlu/>

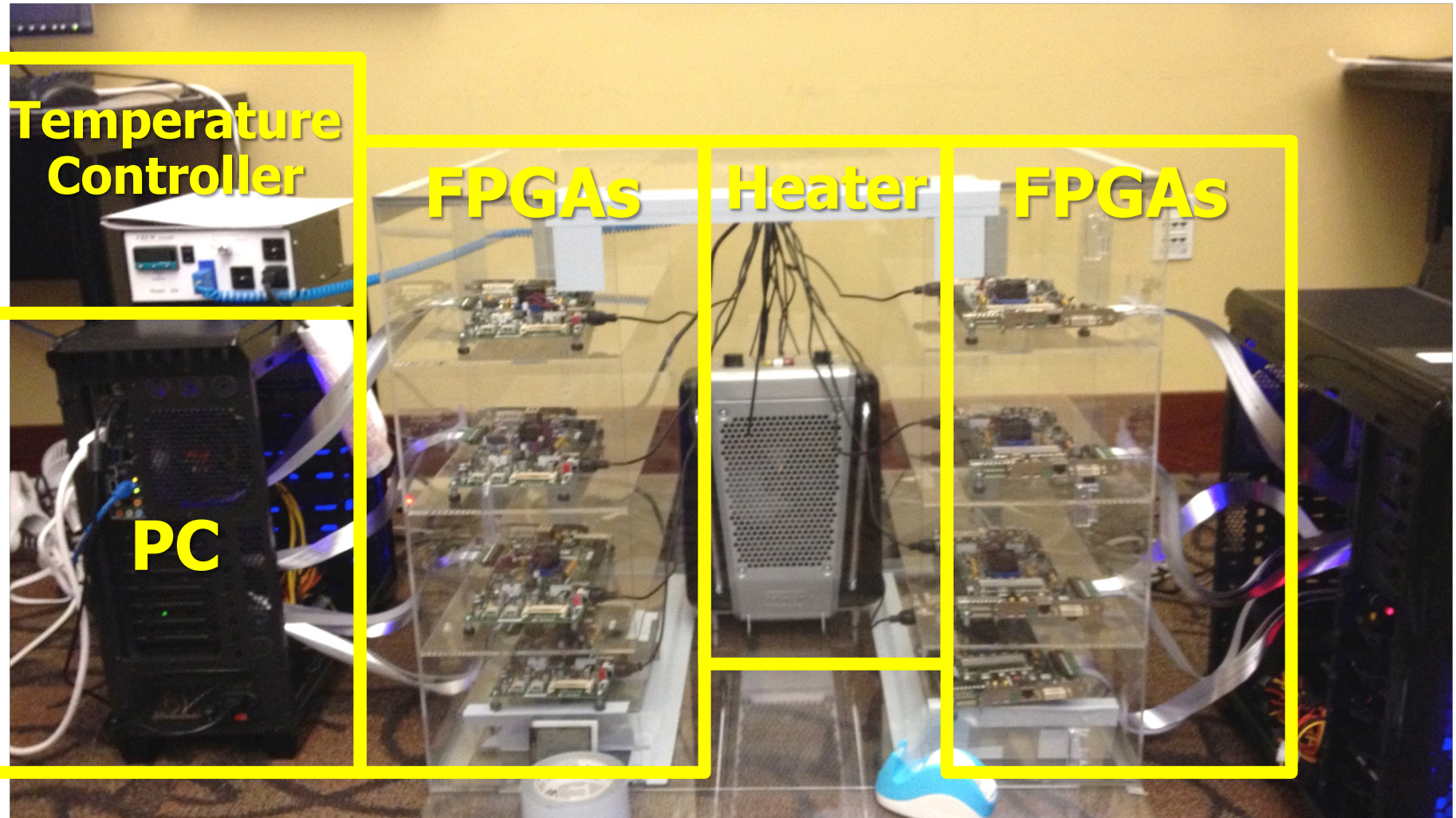
# As Memory Scales, It Becomes Unreliable

- 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*

# Infrastructures to Understand Such Issues



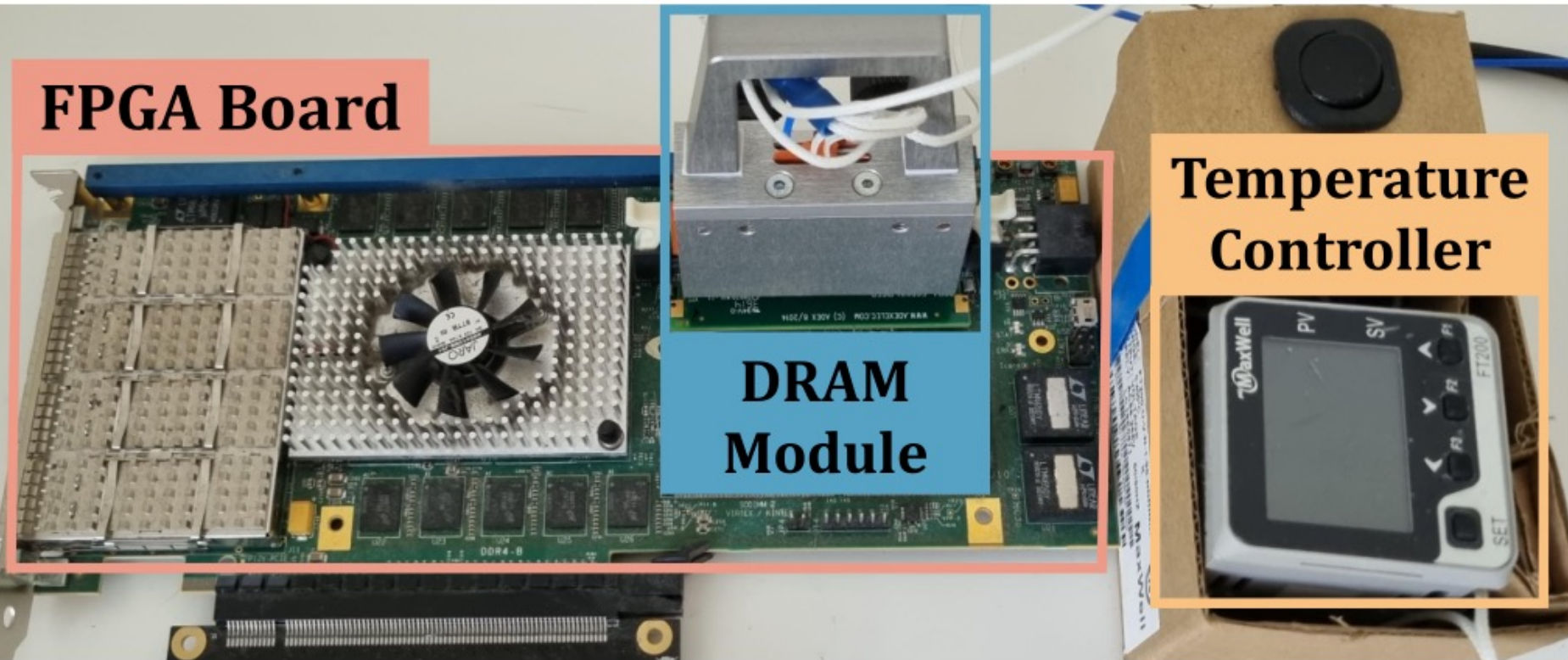


# Memory Testing Infrastructures

**FPGA Board**

**DRAM  
Module**

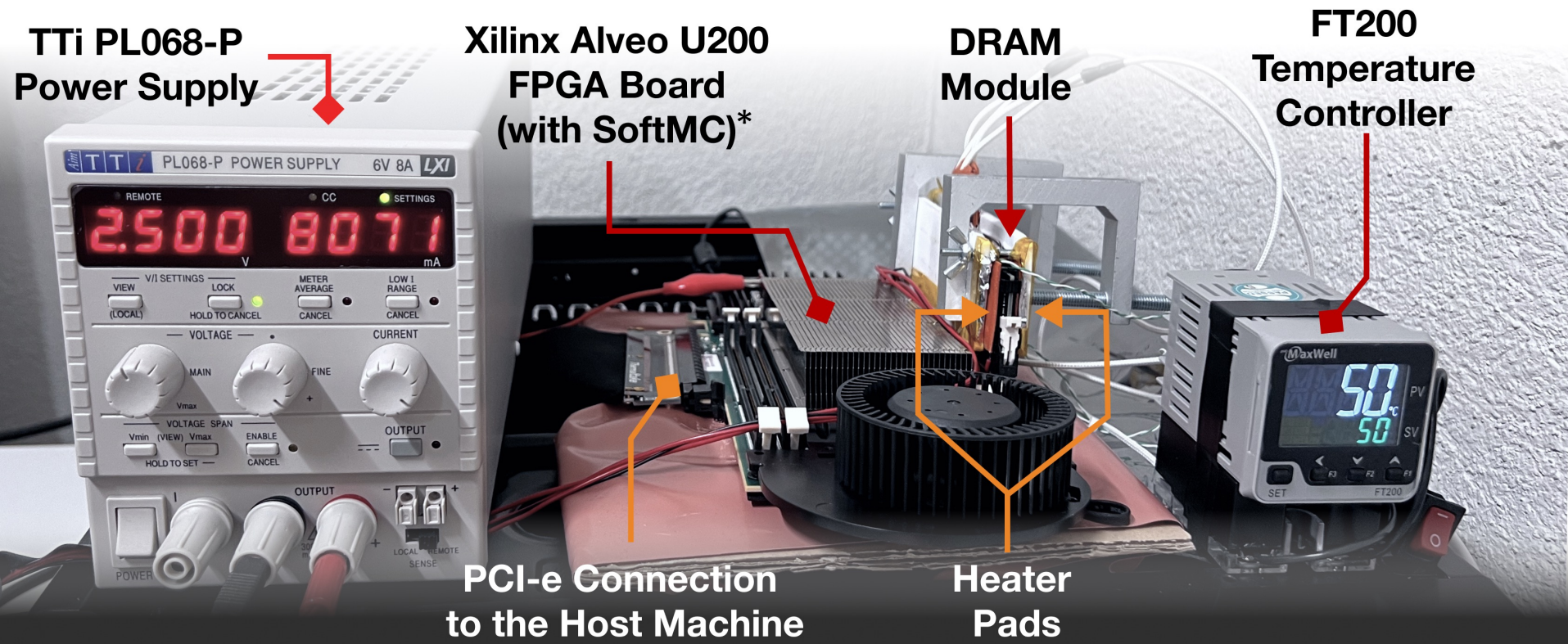
**Temperature  
Controller**



*\* 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 ( $\pm 1\text{mV}$ )**



# A Curious Phenomenon [Kim et al., ISCA 2014]

---

One can  
predictably induce errors  
in most DRAM memory chips

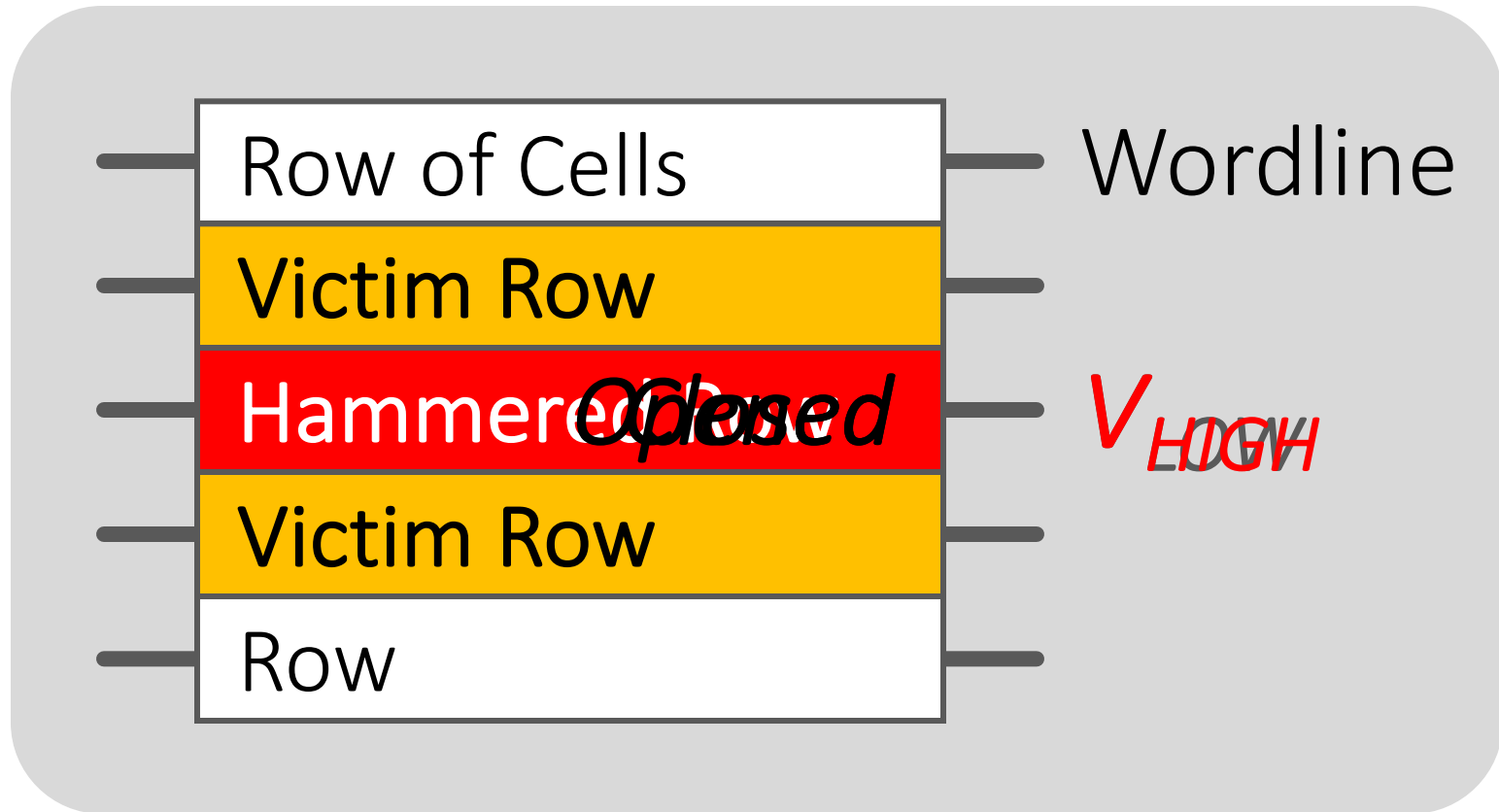
Kim+, "[Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors](#)," ISCA 2014.



Rowhammer

---

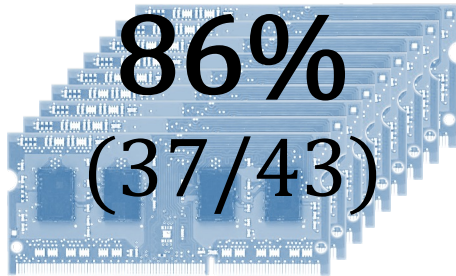
# Modern Memory is Prone to Disturbance Errors



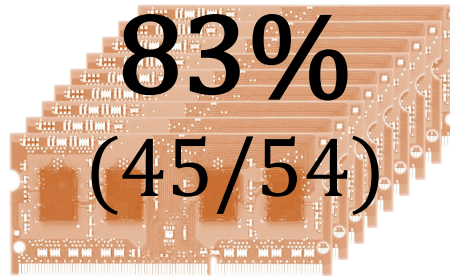
Repeatedly reading a row enough times (before memory gets refreshed) induces **disturbance errors** in **adjacent rows** in **most real DRAM chips you can buy today**

# Most DRAM Modules Are Vulnerable

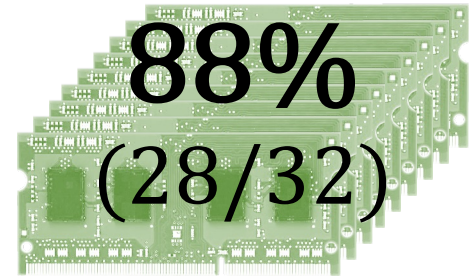
A company



B company



C company



Up to  
 $1.0 \times 10^7$   
errors

Up to  
 $2.7 \times 10^6$   
errors

Up to  
 $3.3 \times 10^5$   
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

# The Story of RowHammer Tutorial ...

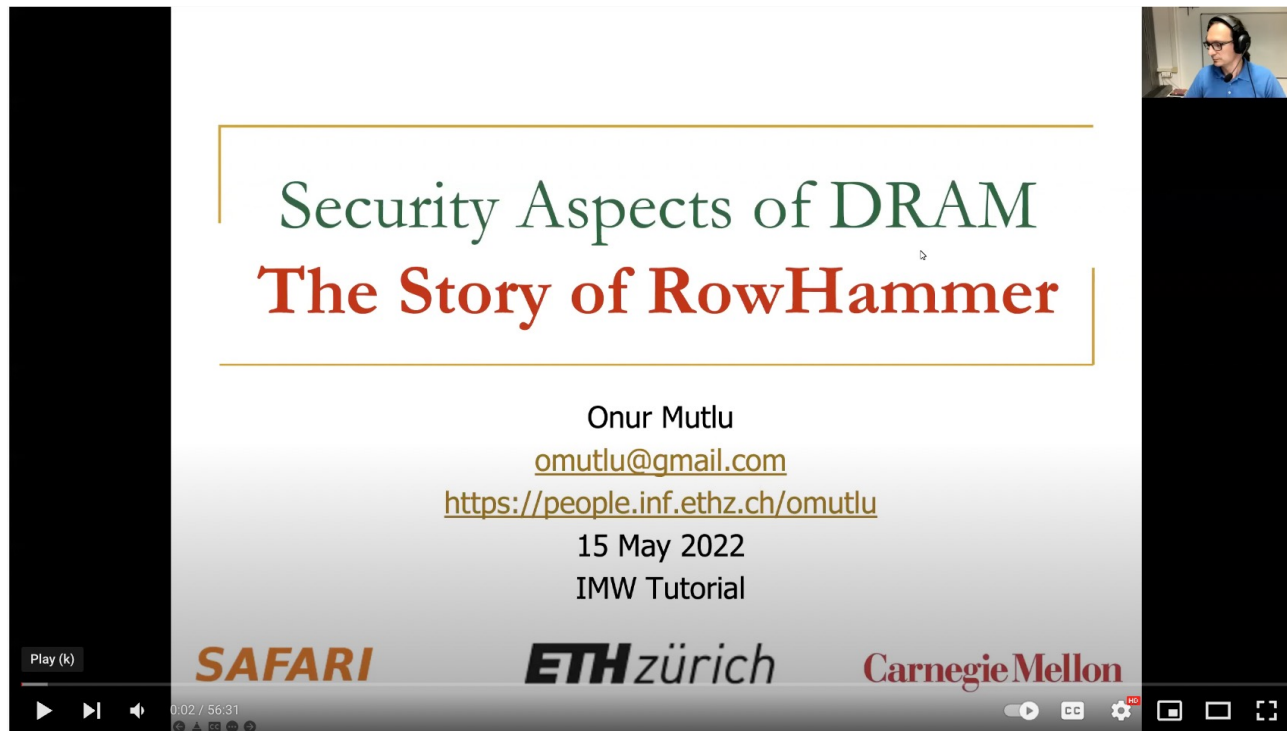
Onur Mutlu,

**"Security Aspects of DRAM: The Story of RowHammer"**

*Invited Tutorial at 14th IEEE Electron Devices Society International Memory Workshop (IMW), Dresden, Germany, May 2022.*

[Slides (pptx)(pdf)]

[Tutorial Video (57 minutes)]



Recent Premieres

The Story of RowHammer – Invited Tutorial at IMW 2022 (Intl. Memory Workshop) - Onur Mutlu

598 views • Premiered Jul 27, 2022

👍 19 🗨 DISLIKE ➦ SHARE ⬇ DOWNLOAD 🔗 CLIP 📌 SAVE ...



Onur Mutlu Lectures  
27.6K subscribers

<https://www.youtube.com/watch?v=37hWqlkQRG0>

ANALYTICS

EDIT VIDEO

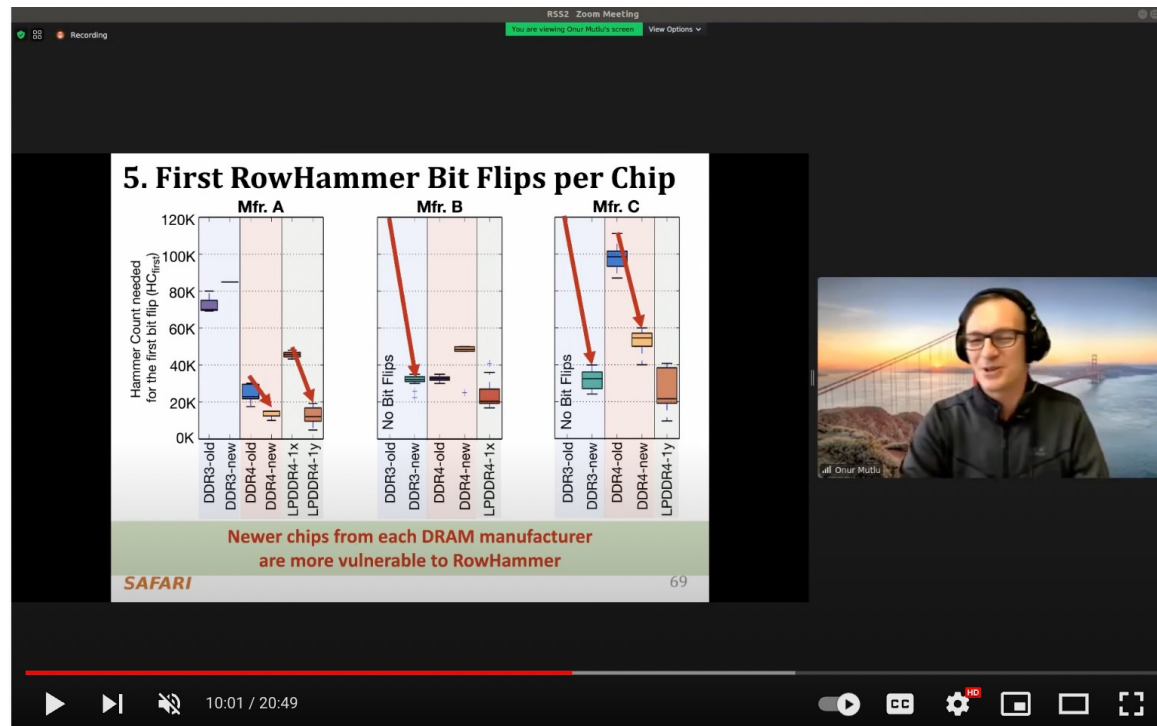


# 10 Years of RowHammer in 20 Minutes

- Onur Mutlu,  
**"The Story of RowHammer"**

*Invited Talk at the Workshop on Robust and Safe Software 2.0 (RSS2), held with the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, 28 February 2022.*

[Slides (pptx) (pdf)]



The Story of RowHammer - Invited Talk in Robust & Safe Software Workshop (ASPLOS 2022) - Onur Mutlu

402 views • Premiered Apr 27, 2022

👍 17    🗑 DISLIKE    ➦ SHARE    ⬇ DOWNLOAD    ✂ CLIP    ⚙ SAVE    ...



Onur Mutlu Lectures  
24.5K subscribers

<https://www.youtube.com/watch?v=ctKTRyi96Bk>

SUBSCRIBED



# Memory Scaling Issues **Are** Real

---

- 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](#)).***

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

Yoongu Kim<sup>1</sup>   Ross Daly\*   Jeremie Kim<sup>1</sup>   Chris Fallin\*   Ji Hye Lee<sup>1</sup>  
Donghyuk Lee<sup>1</sup>   Chris Wilkerson<sup>2</sup>   Konrad Lai   Onur Mutlu<sup>1</sup>

<sup>1</sup>Carnegie Mellon University   <sup>2</sup>Intel 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

---

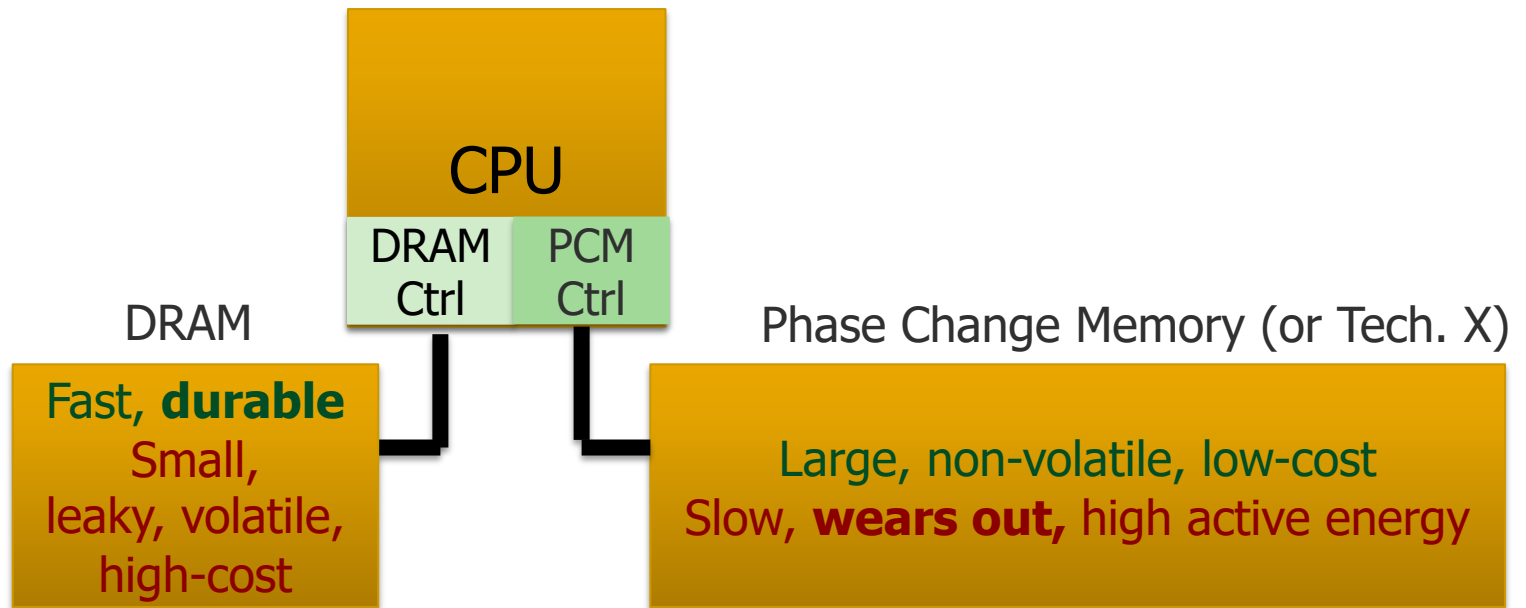
- Onur Mutlu and Jeremie Kim,  
**"RowHammer: A Retrospective"**  
*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security*, 2019.  
[[Preliminary arXiv version](#)]  
[[Slides from COSADE 2019 \(pptx\)](#)]  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

## RowHammer: A Retrospective

Onur Mutlu<sup>§‡</sup>      Jeremie S. Kim<sup>‡§</sup>  
§ETH Zürich      ‡Carnegie Mellon University

# Hybrid Memory Enables Better Scaling

---



Hardware/software manage data allocation & movement  
**to achieve the best of multiple technologies**

Meza+, "[Enabling Efficient and Scalable Hybrid Memories](#)," IEEE Comp. Arch. Letters, 2012.

Yoon+, "[Row Buffer Locality Aware Caching Policies for Hybrid Memories](#)," ICCD 2012 Best Paper Award.

## Main Memory Needs Intelligent Controllers



# An Example Intelligent Controller

- A. Giray Yaglikci, Minesh Patel, Jeremie S. Kim, Roknoddin Azizi, Ataberk Olgun, Lois Orosa, Hasan Hassan, Jisung Park, Konstantinos Kanellopoulos, Taha Shahroodi, Saugata Ghose, and Onur Mutlu,

## **"BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows"**

*Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA)*, Virtual, February-March 2021.

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

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

[[Intel Hardware Security Academic Awards Short Talk Slides \(pptx\)](#) ([pdf](#))]

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

[[Short Talk Video](#) (7 minutes)]

[[Intel Hardware Security Academic Awards Short Talk Video](#) (2 minutes)]

[[BlockHammer Source Code](#)]

***Intel Hardware Security Academic Award Finalist (one of 4 finalists out of 34 nominations)***

## **BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows**

A. Giray Yağlıkçı<sup>1</sup> Minesh Patel<sup>1</sup> Jeremie S. Kim<sup>1</sup> Roknoddin Azizi<sup>1</sup> Ataberk Olgun<sup>1</sup> Lois Orosa<sup>1</sup>

Hasan Hassan<sup>1</sup> Jisung Park<sup>1</sup> Konstantinos Kanellopoulos<sup>1</sup> Taha Shahroodi<sup>1</sup> Saugata Ghose<sup>2</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>*ETH Zürich*

<sup>2</sup>*University of Illinois at Urbana-Champaign*

# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

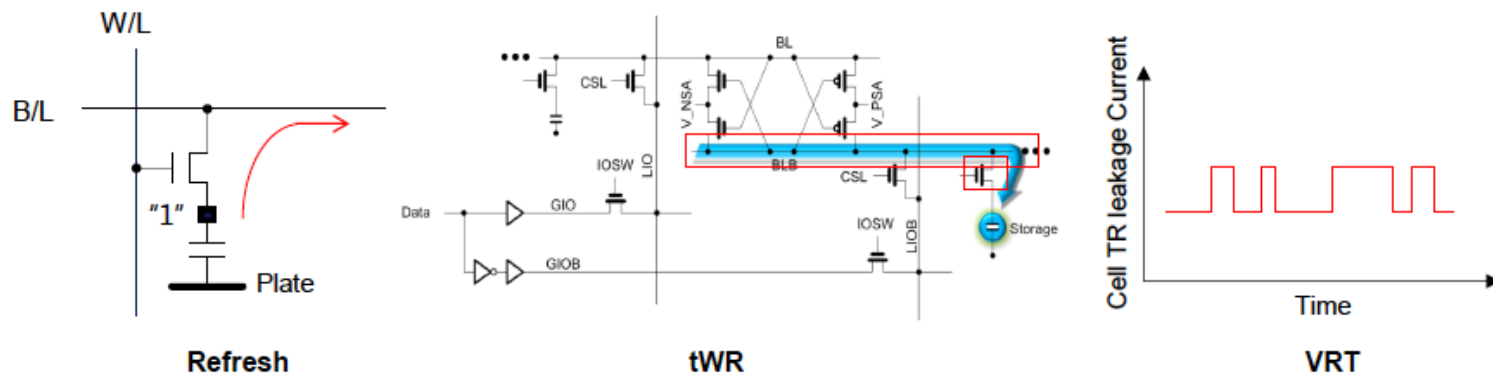
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

### ❖ tWR

- Contact resistance between the cell capacitor and access transistor increasing
- On-current of the cell access transistor decreasing
- Bit-line resistance increasing

### ❖ VRT

- Occurring more frequently with cell capacitance decreasing



# Call for Intelligent Memory Controllers

## DRAM Process Scaling Challenges

### ❖ Refresh

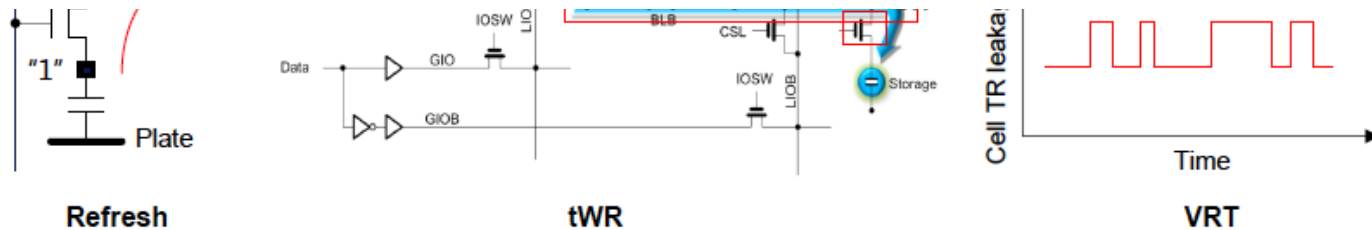
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

## Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

Uksong Kang, Hak-soo Yu, Churoo Park, \*Hongzhong Zheng,  
\*\*John Halbert, \*\*Kuljit Bains, SeongJin Jang, and Joo Sun Choi

*Samsung Electronics, Hwasung, Korea / \*Samsung Electronics, San Jose / \*\*Intel*



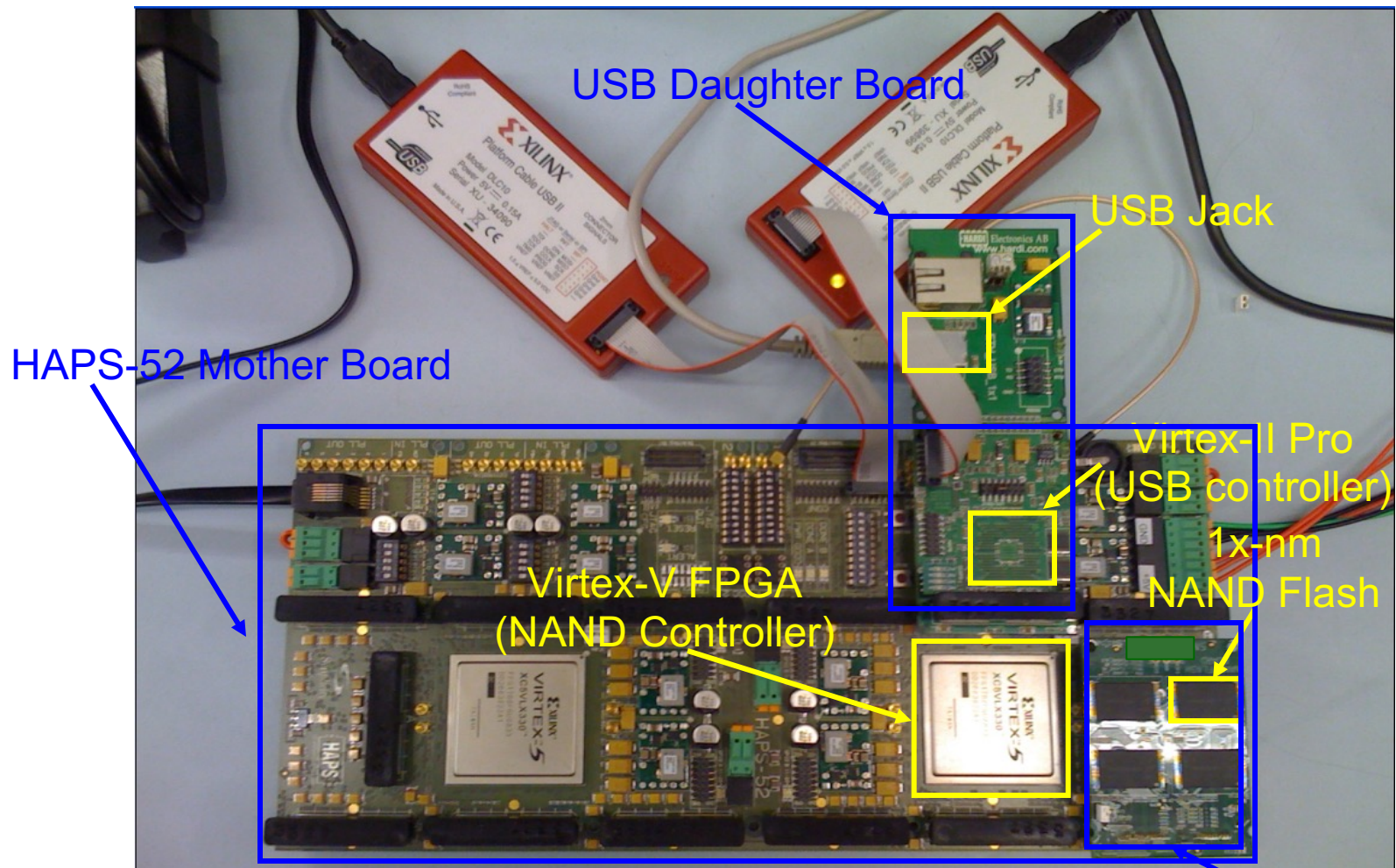
# Another Example Intelligent Controller

- Minesh Patel, Geraldo F. de Oliveira Jr., and Onur Mutlu,  
**"HARP: Practically and Effectively Identifying Uncorrectable Errors in Memory Chips That Use On-Die Error-Correcting Codes"**  
*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)]  
[[HARP Source Code \(Officially Artifact Evaluated with All Badges\)](#)]



## HARP: Practically and Effectively Identifying Uncorrectable Errors in Memory Chips That Use On-Die Error-Correcting Codes

# Aside: Intelligent Controller for NAND Flash



[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018, SIGMETRICS 2018]

Cai+, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives," Proc. IEEE 2017.





*Proceedings of the IEEE, Sept. 2017*



## Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

*This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD's reliability and lifetime.*

By YU CAI, SAUGATA GHOSE, ERICH F. HARATSCH, YIXIN LUO, AND ONUR MUTLU

<https://arxiv.org/pdf/1706.08642>



# 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<sup>†</sup> Engin Ipek<sup>†</sup> Onur Mutlu<sup>‡</sup> Doug Burger<sup>†</sup>

<sup>†</sup>Computer Architecture Group  
Microsoft Research  
Redmond, WA  
{blee, ipek, dburger}@microsoft.com

<sup>‡</sup>Computer Architecture Laboratory  
Carnegie Mellon University  
Pittsburgh, PA  
onur@cmu.edu

Intelligent  
Memory Controllers  
Can Avoid Many Failures  
& Enable Better Scaling

## Main Memory Needs Intelligent Controllers

# Why In-Memory Computation Today?

---

## ■ **Push from Technology**

- ❑ DRAM Scaling at jeopardy
  - Controllers close to DRAM
  - Industry open to new memory architectures

## ■ **Pull from Systems and Applications**

- ❑ Data access is the major system and application bottleneck
- ❑ Systems are energy & power limited
- ❑ Data movement much more energy-hungry than computation

# Three Key Systems & Application Trends

---

## 1. Data access is the 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



# 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)

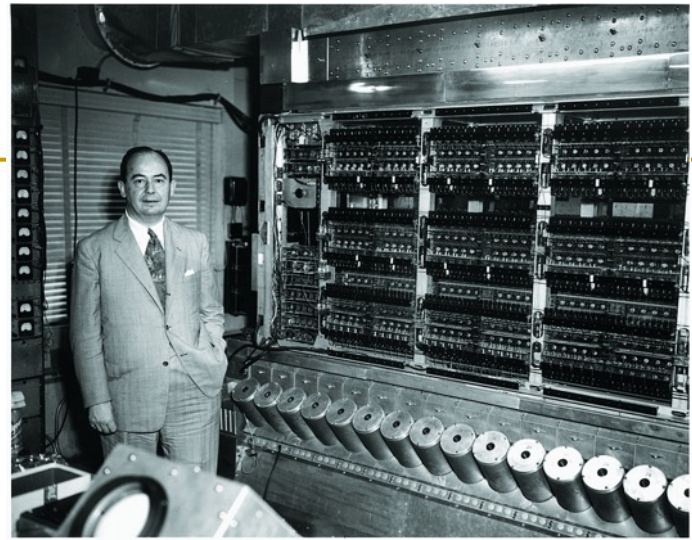
# The Problem

---

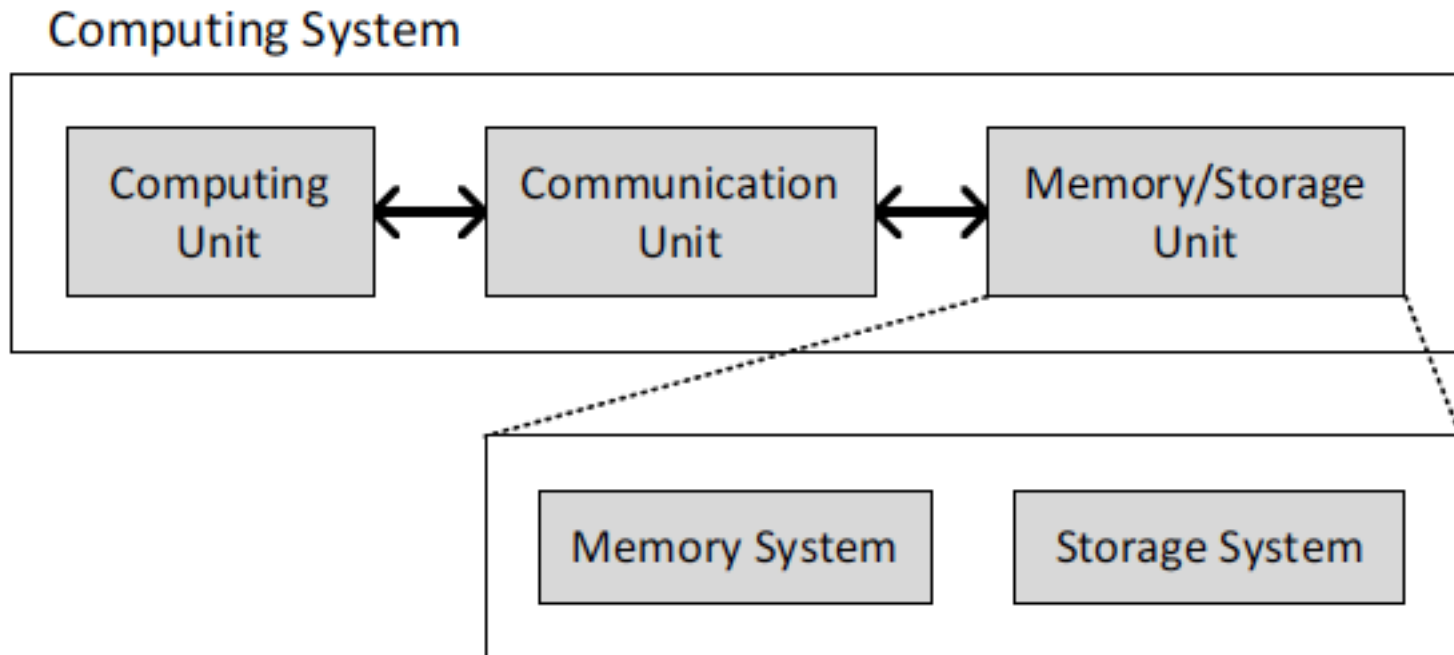
Processing of data  
is performed  
far away from the data

# A Computing System

- Three key components
- Computation
- Communication
- Storage/memory



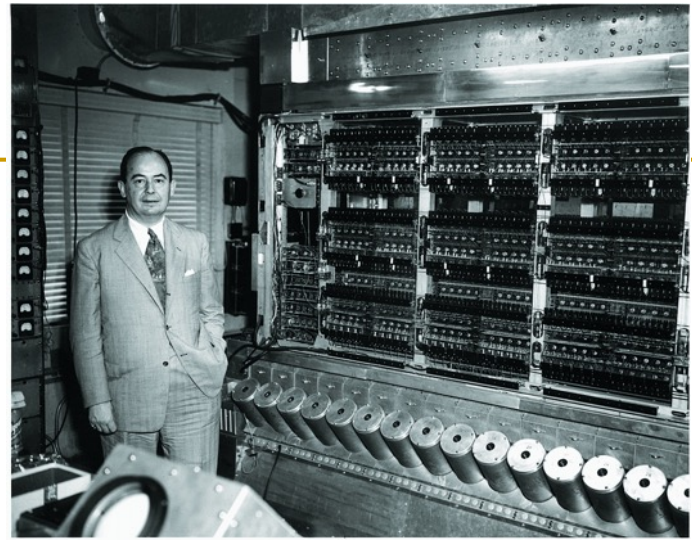
Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.





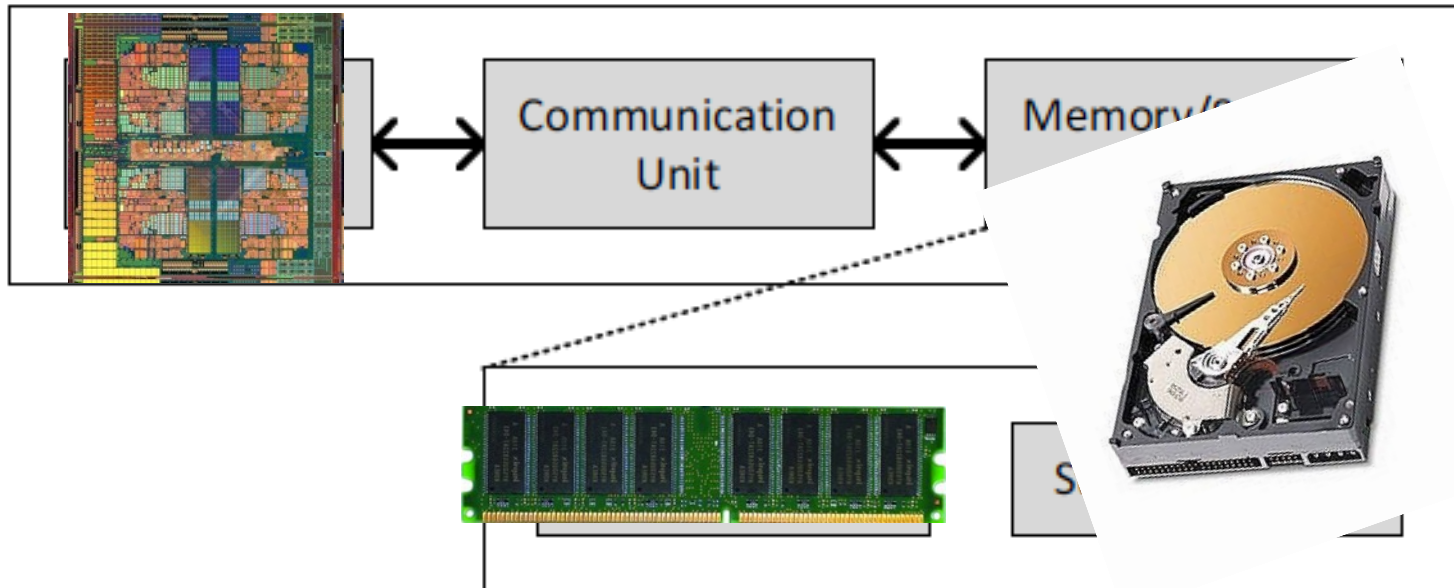
# A Computing System

- Three key components
- Computation
- Communication
- Storage/memory



Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.

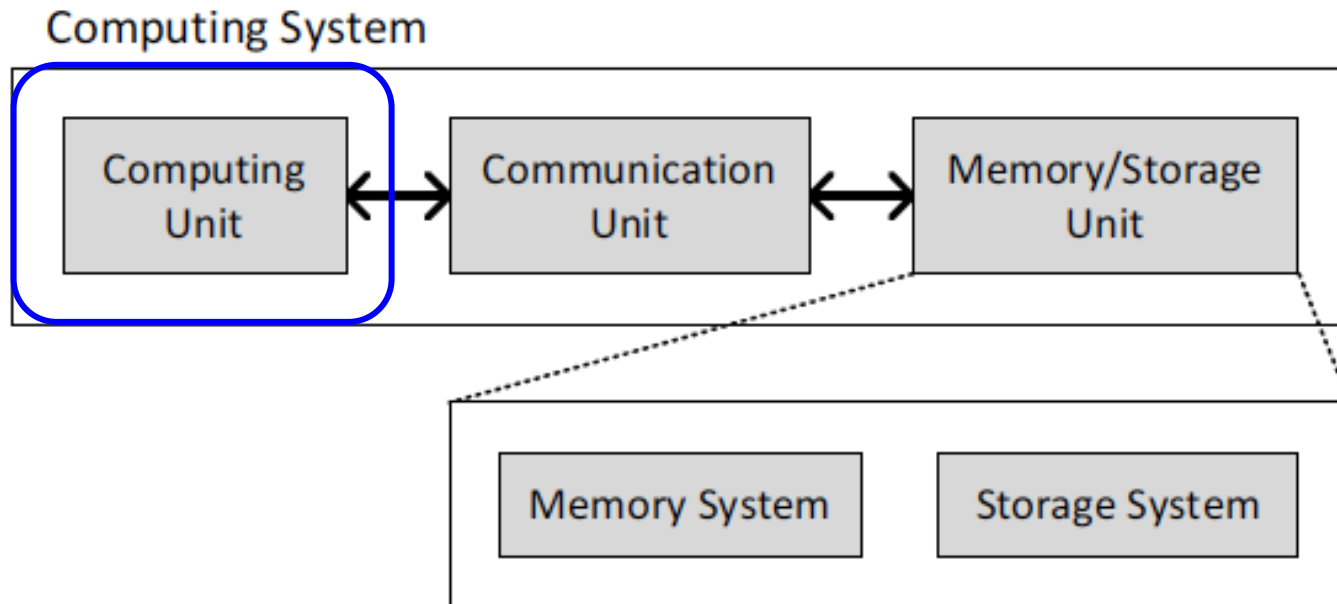
## Computing System



# Today's Computing Systems

---

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



# It's the Memory, Stupid!

---

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

**RICHARD SITES**

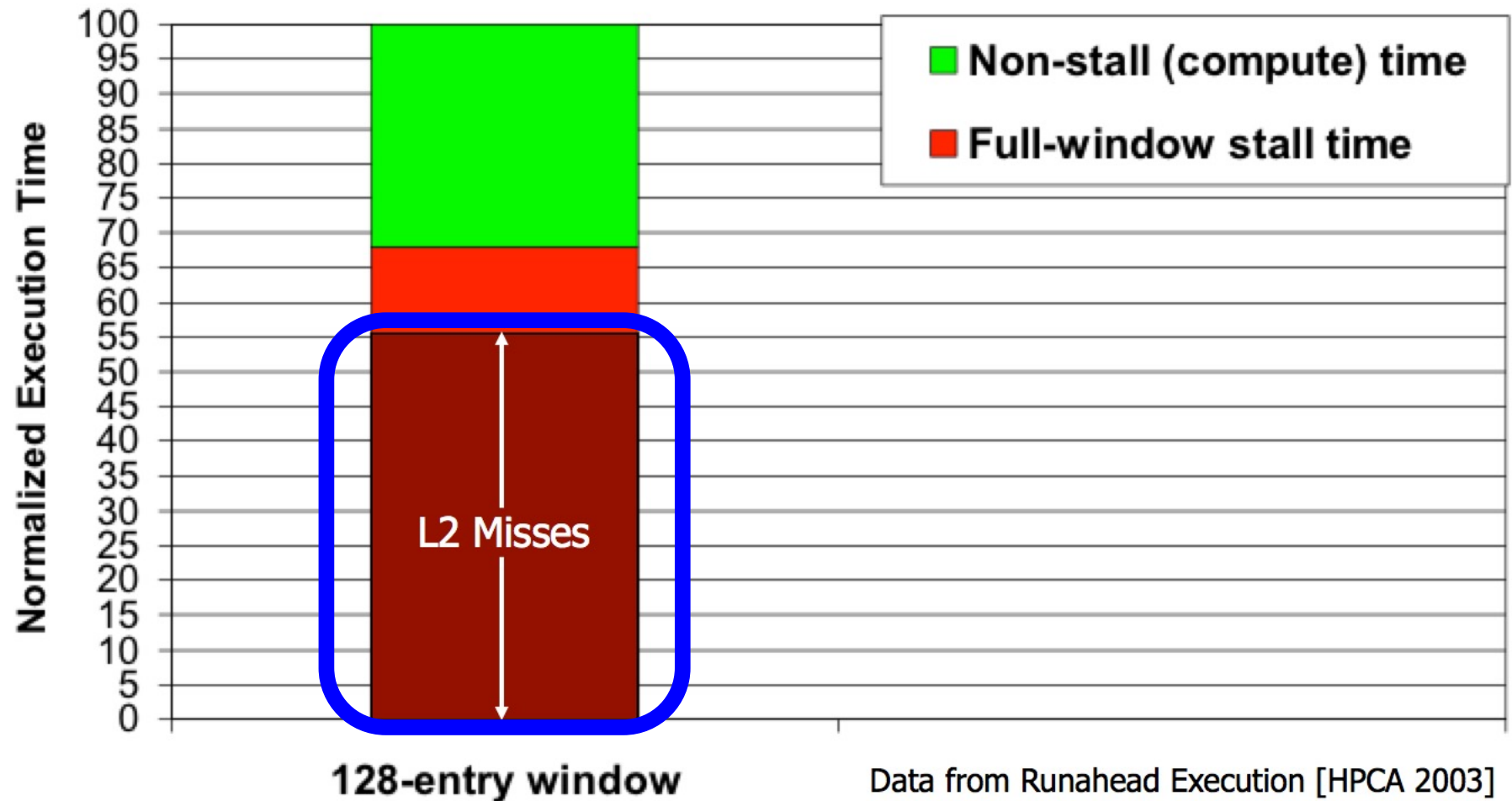
## **It's the Memory, Stupid!**

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 §

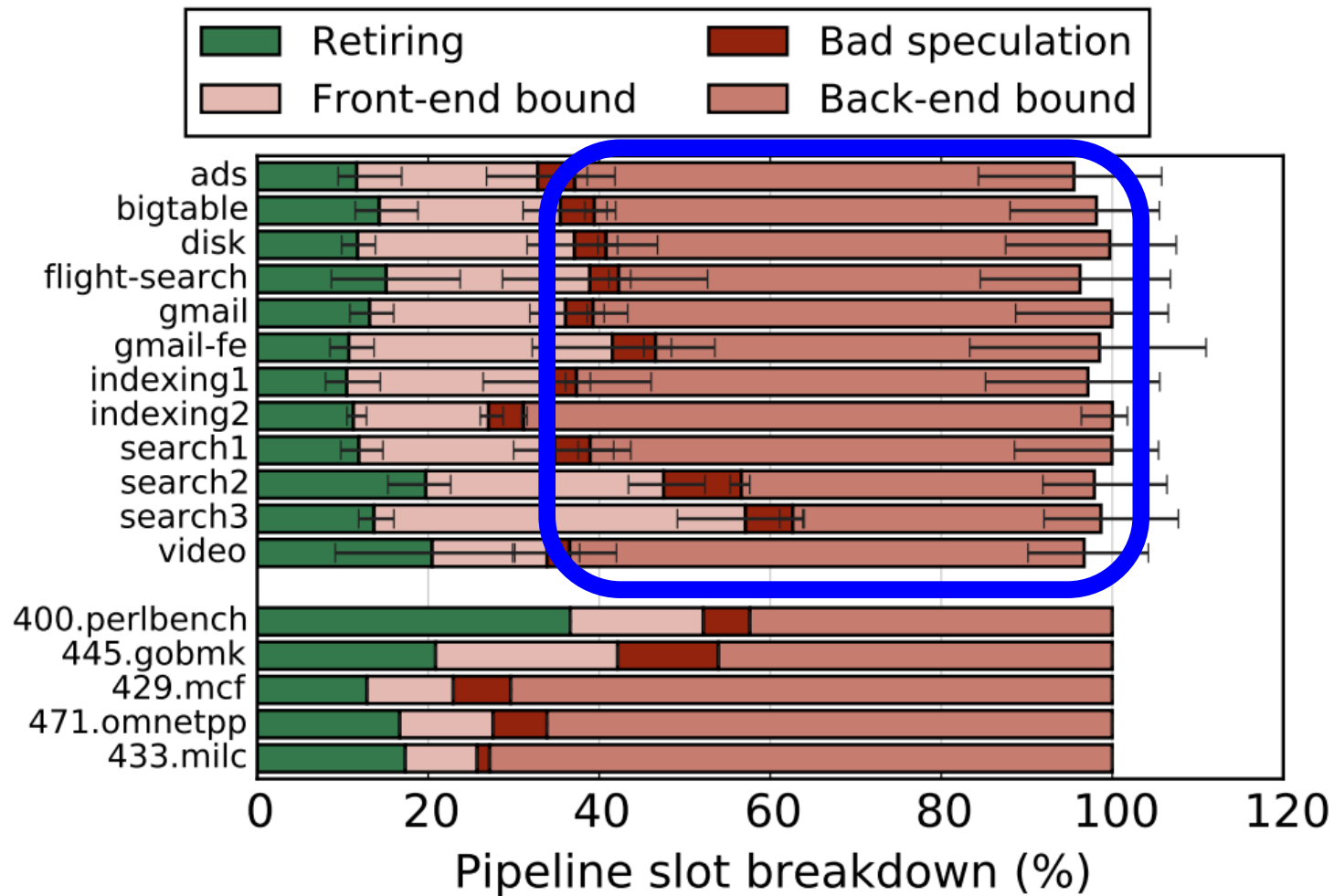
§ECE Department  
The University of Texas at Austin  
{onur,patt}@ece.utexas.edu

†Microprocessor Research  
Intel Labs  
jared.w.stark@intel.com

‡Desktop Platforms Group  
Intel Corporation  
chris.wilkerson@intel.com

# The Performance Perspective (Today)

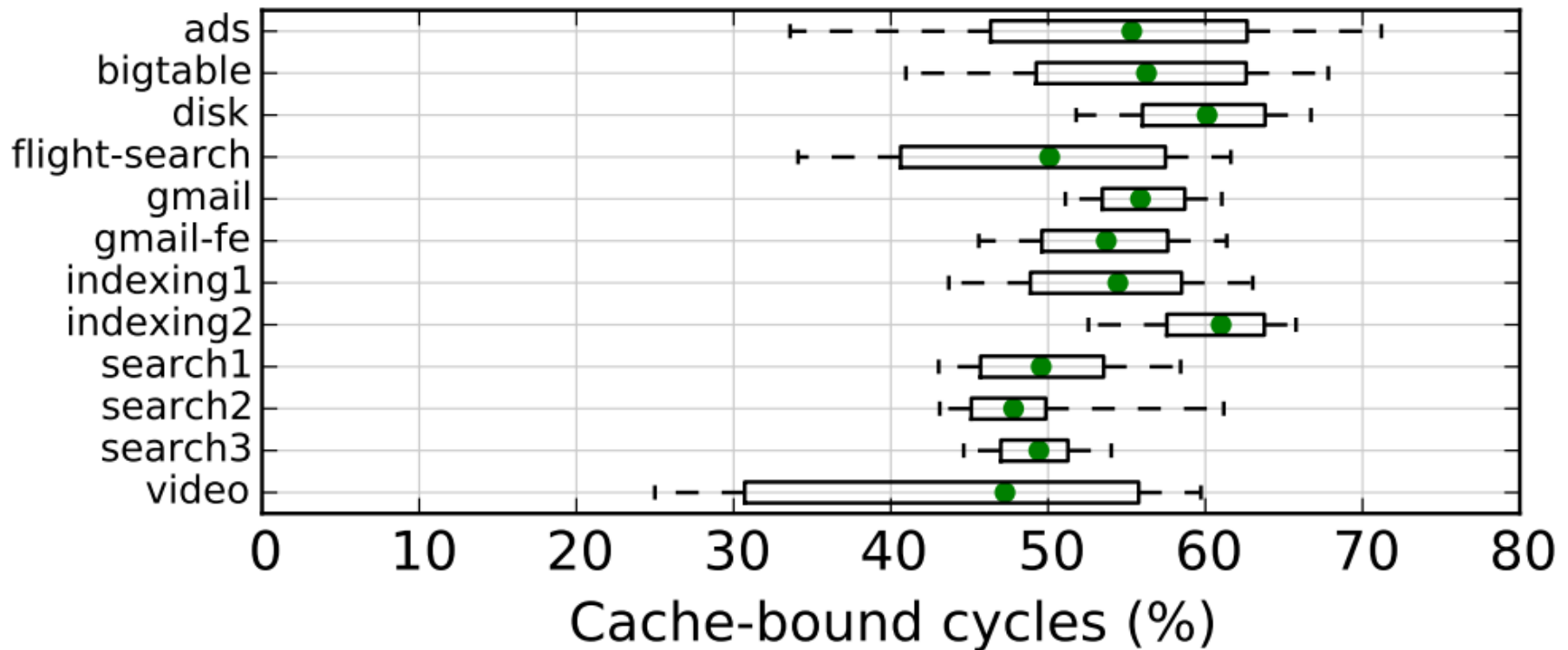
- All of Google's Data Center Workloads (2015):





# The Performance Perspective (Today)

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



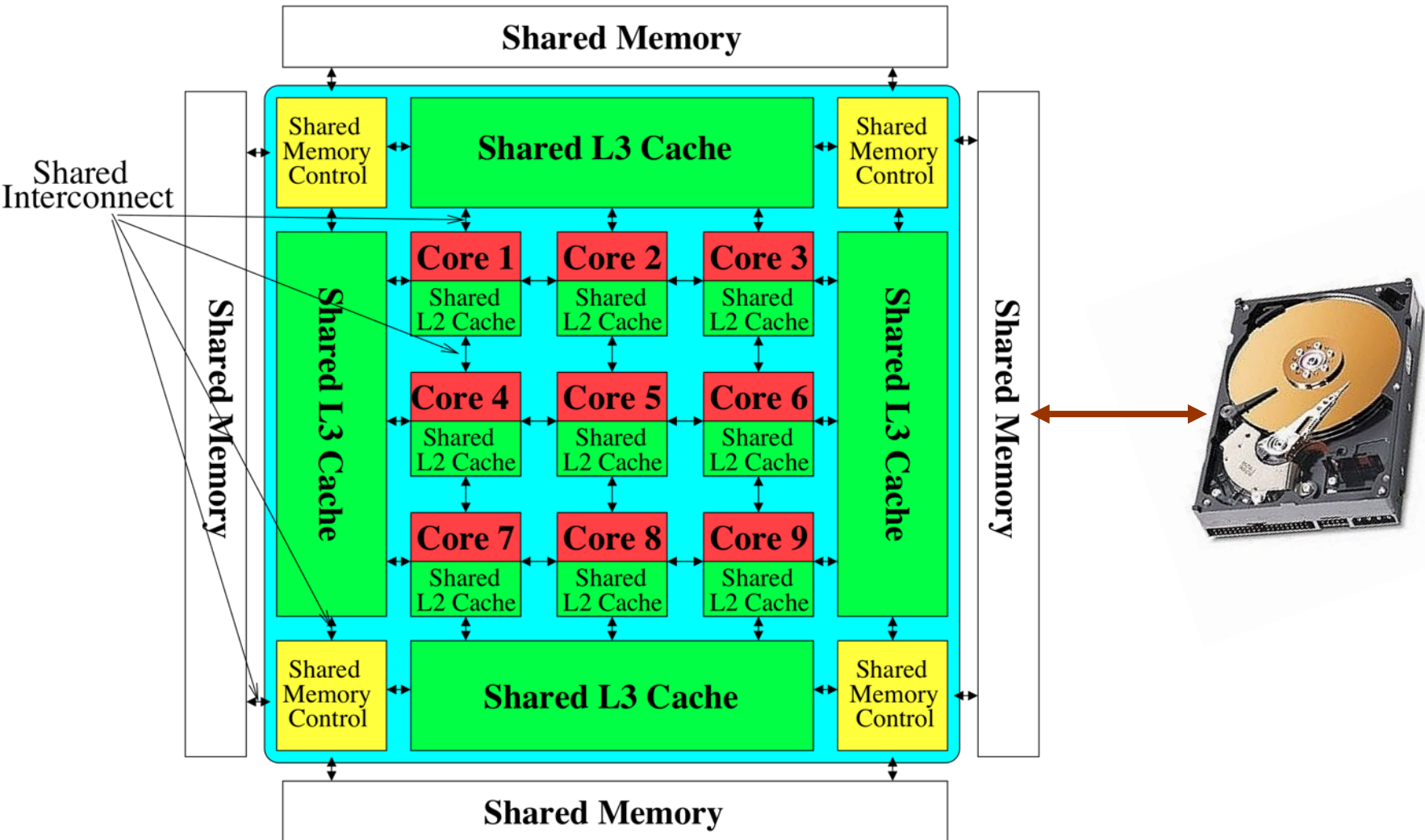
**Figure 11: Half of cycles are spent stalled on caches.**

# Perils of Processor-Centric Design

---

- **Grossly-imbalanced systems**
  - ❑ Processing done only in **one place**
  - ❑ All else just stores and moves data: **data moves a lot**
    - Energy inefficient
    - Low performance
    - Complex
- **Overly complex and bloated processor (and accelerators)**
  - ❑ To tolerate data access from memory
  - ❑ Complex hierarchies and mechanisms
    - Energy inefficient
    - Low performance
    - Complex

# Perils of Processor-Centric Design



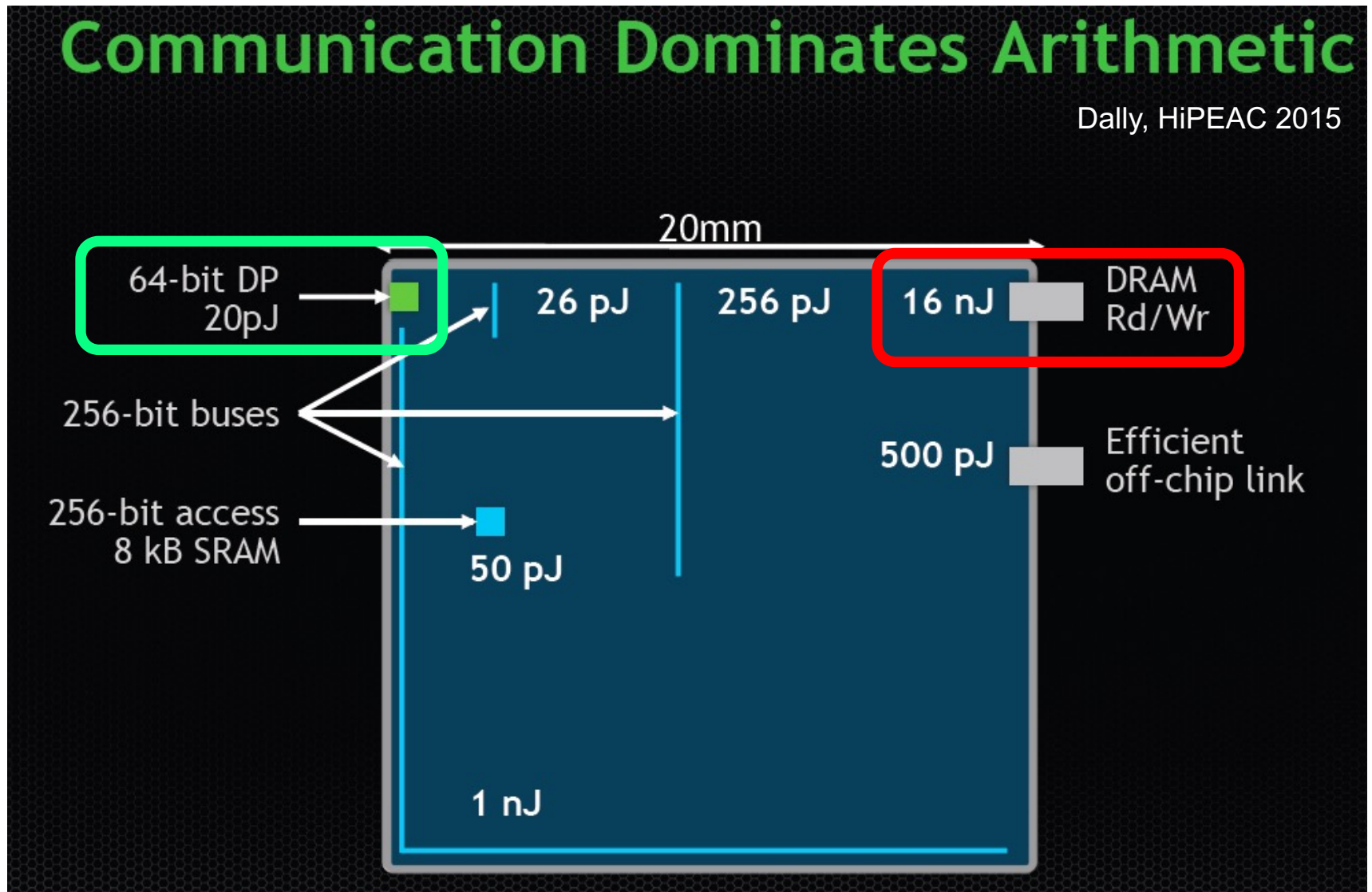
**Most of the system is dedicated to storing and moving data**

**Yet, system is still bottlenecked by memory**

# The Energy Perspective

## Communication Dominates Arithmetic

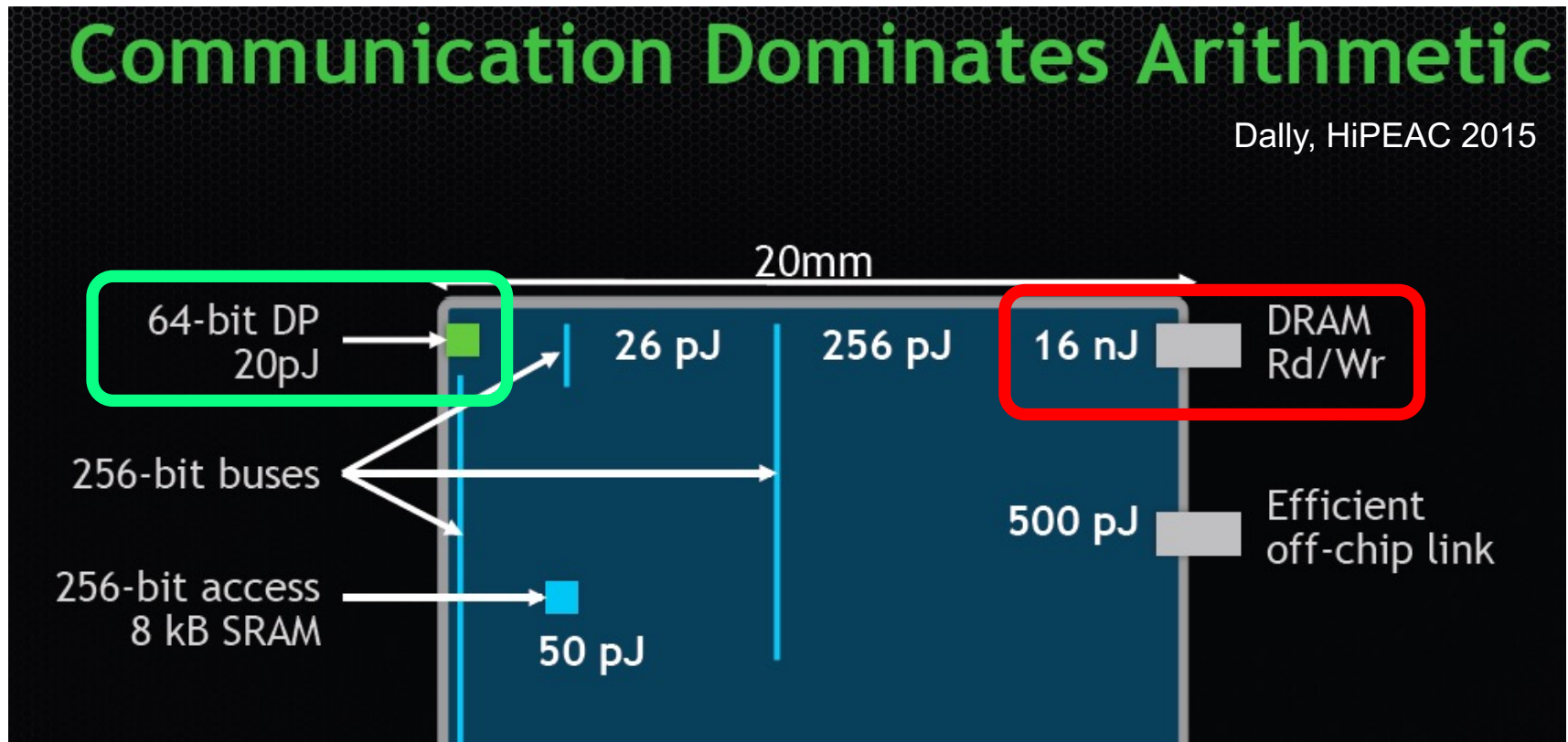
Dally, HiPEAC 2015



# Data Movement vs. Computation Energy

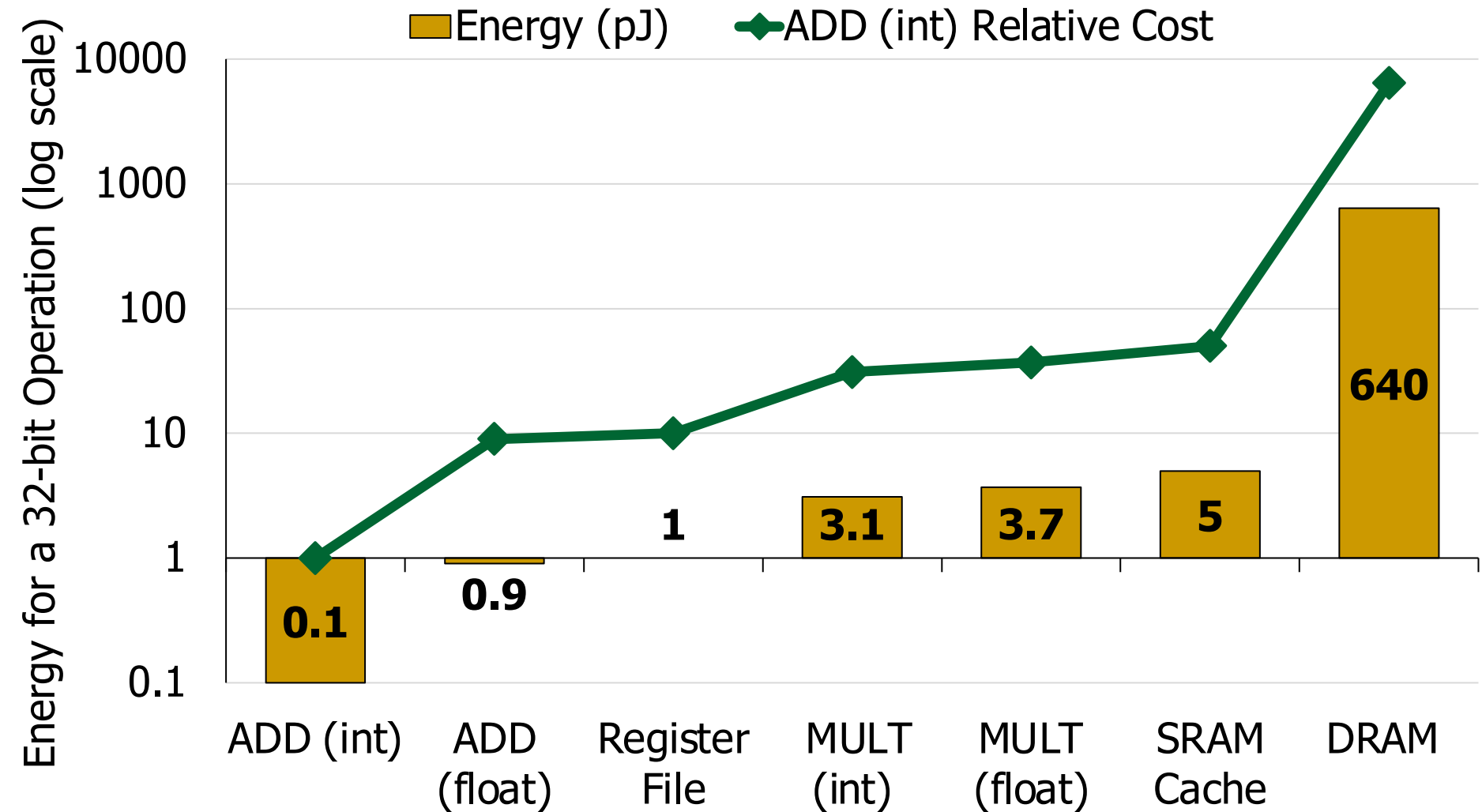
## Communication Dominates Arithmetic

Dally, HiPEAC 2015



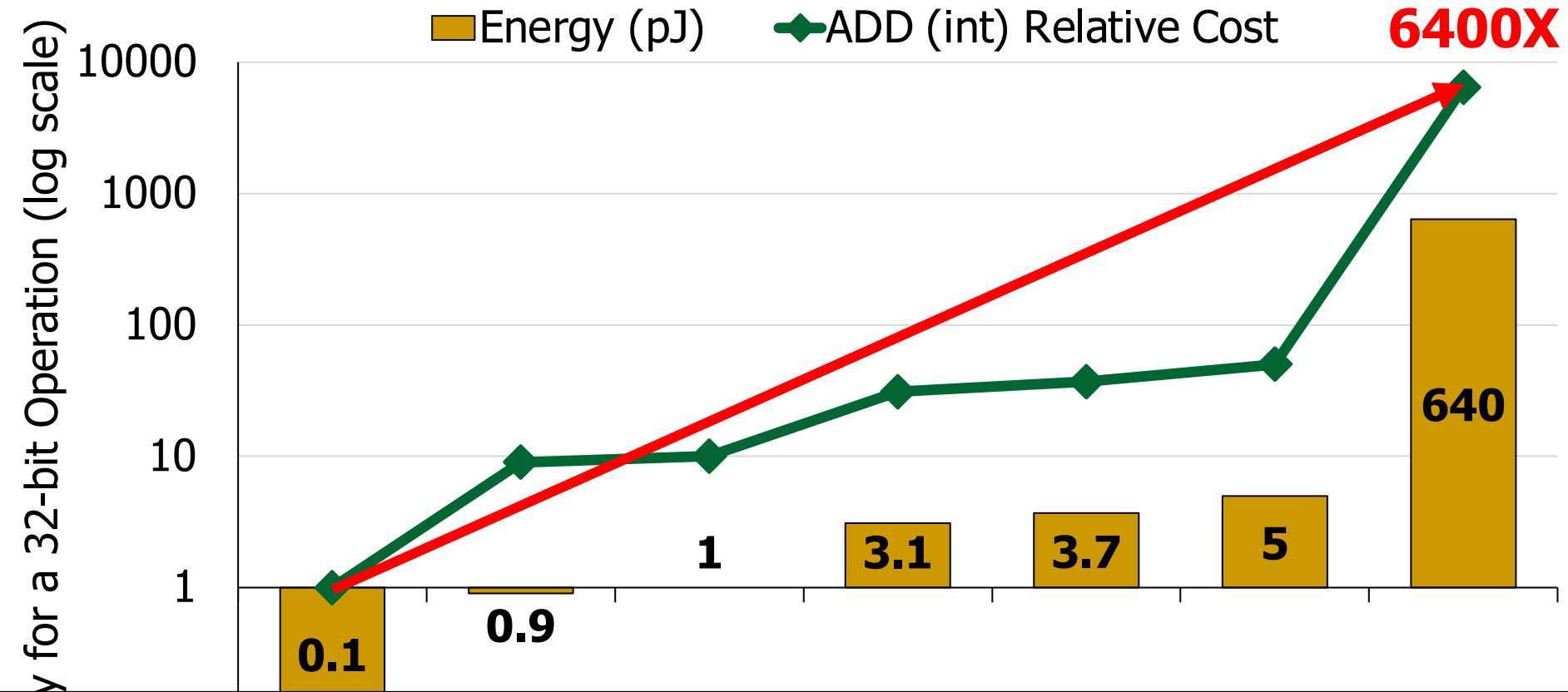
A memory access consumes  $\sim 100-1000\times$  the energy of a complex addition

# Data Movement vs. Computation Energy





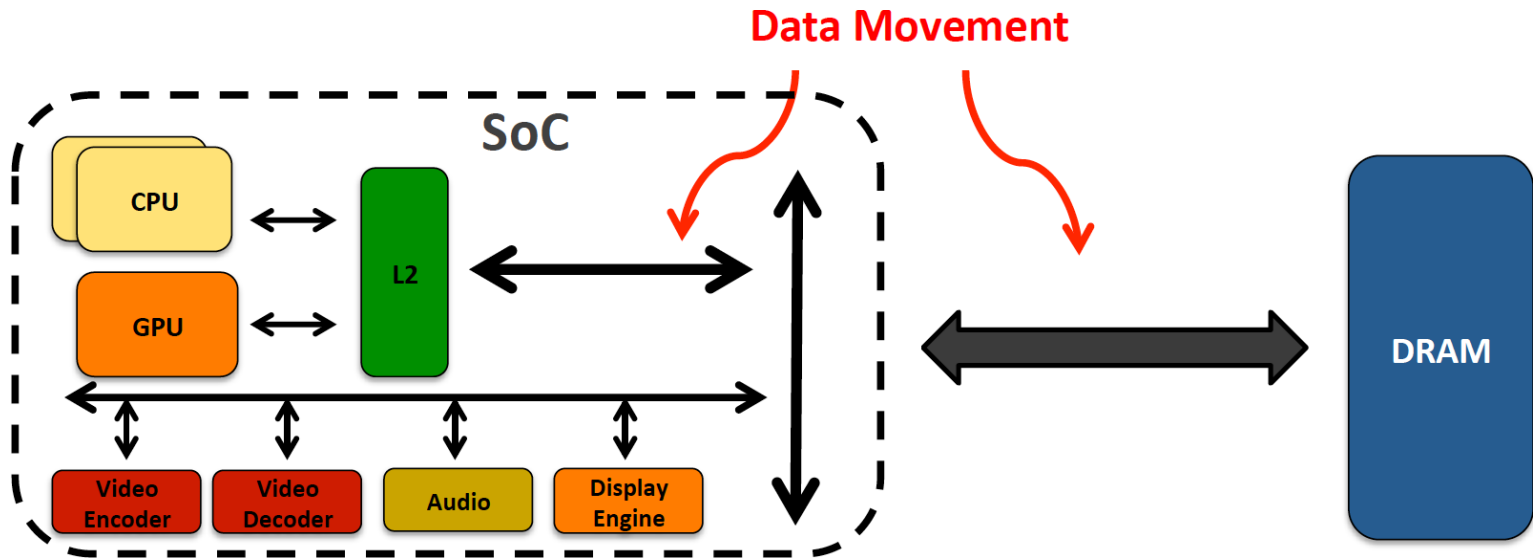
# Data Movement vs. Computation Energy



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

# Data Movement vs. Computation Energy

- **Data movement** is a major system energy bottleneck
  - ❑ Comprises 41% of mobile system energy during web browsing [2]
  - ❑ Costs  $\sim 115$  times as much energy as an ADD operation [1, 2]



[1]: Reducing data Movement Energy via Online Data Clustering and Encoding (MICRO'16)

[2]: Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms (IISWC'14)

# 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 Boroumand<sup>1</sup>

Saugata Ghose<sup>1</sup>

Youngsok Kim<sup>2</sup>

Rachata Ausavarungnirun<sup>1</sup>

Eric Shiu<sup>3</sup>

Rahul Thakur<sup>3</sup>

Daehyun Kim<sup>4,3</sup>

Aki Kuusela<sup>3</sup>

Allan Knies<sup>3</sup>

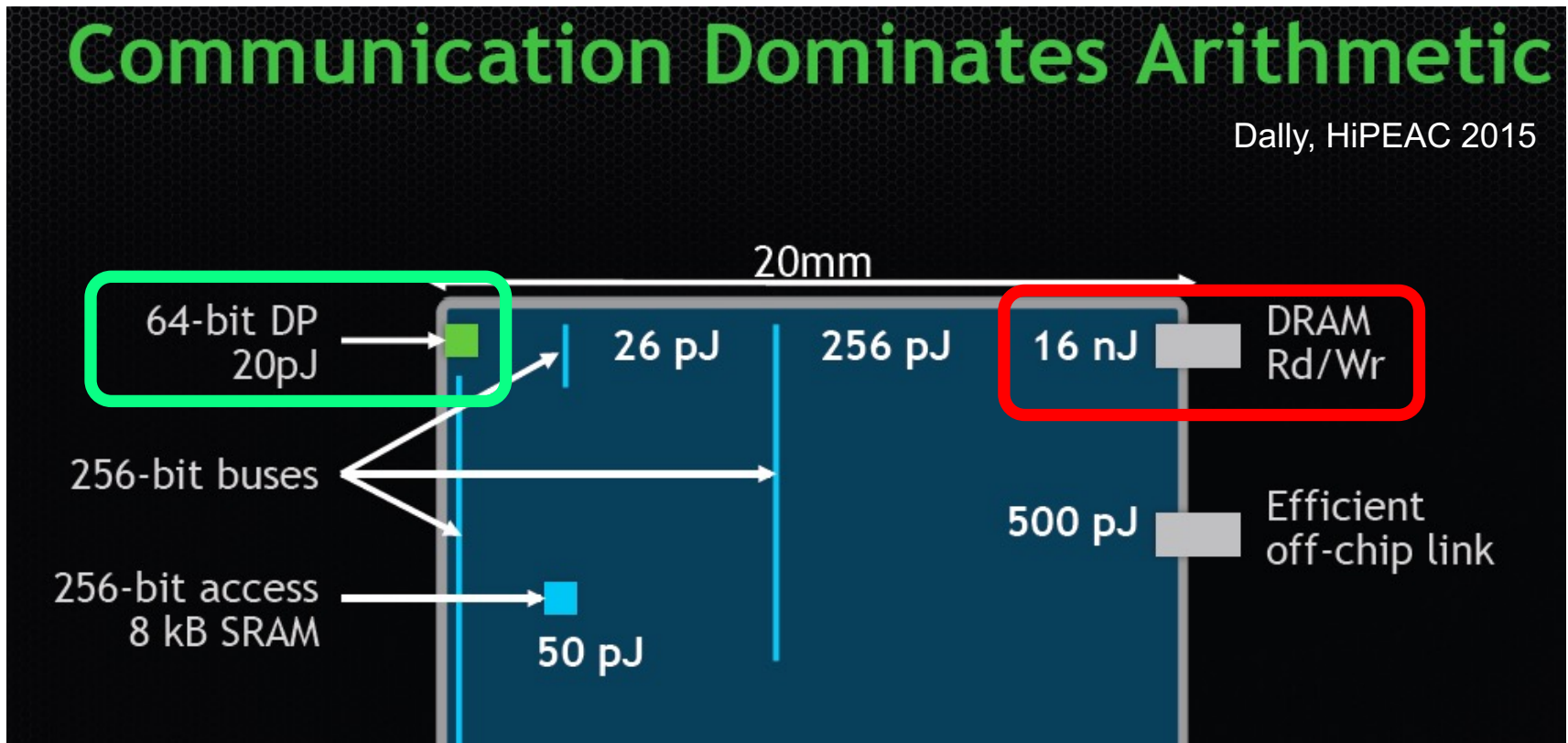
Parthasarathy Ranganathan<sup>3</sup>

Onur Mutlu<sup>5,1</sup>

# We Do Not Want to Move Data!

## Communication Dominates Arithmetic

Dally, HiPEAC 2015



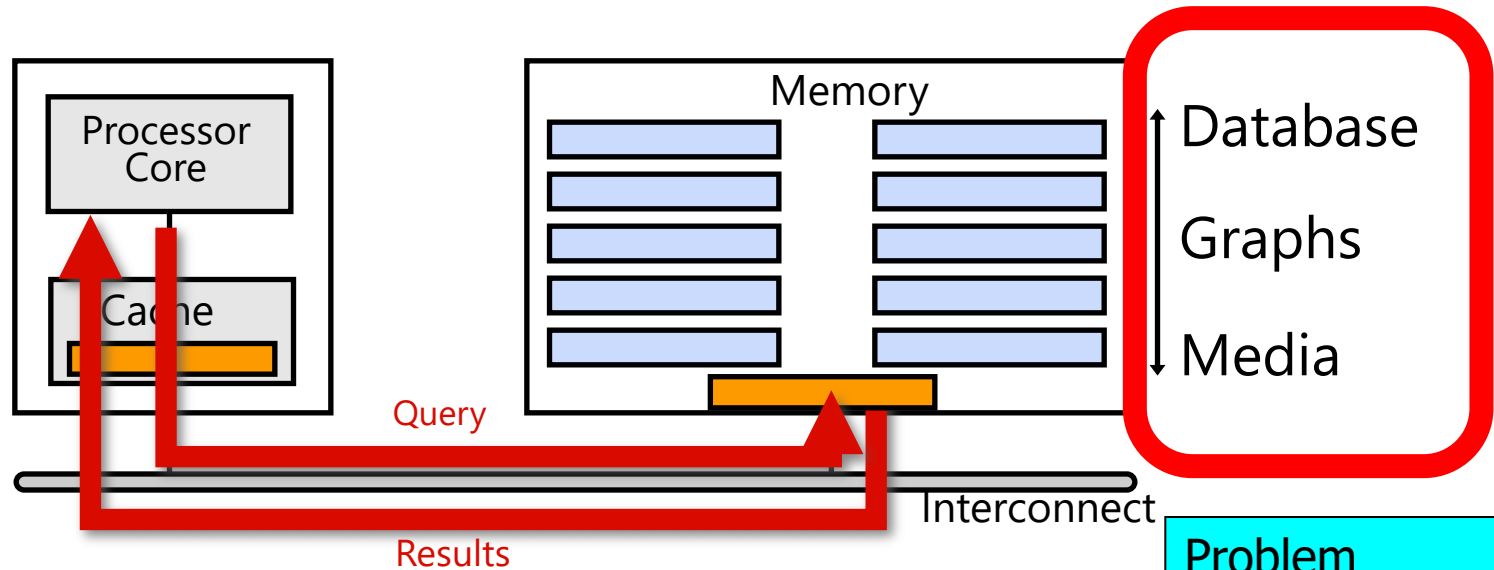
A memory access consumes  $\sim 100$ - $1000\times$  the energy of a complex addition

# 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

# Goal: Processing Inside Memory



- 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          |



# PIM Review and Open Problems

---

## A Modern Primer on Processing in Memory

Onur Mutlu<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirun<sup>d</sup>

*SAFARI Research Group*

<sup>a</sup>*ETH Zürich*

<sup>b</sup>*Carnegie Mellon University*

<sup>c</sup>*University of Illinois at Urbana-Champaign*

<sup>d</sup>*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.*

# A Modern Primer on Processing in Memory

Onur Mutlu<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirun<sup>d</sup>

SAFARI Research Group

<sup>a</sup>ETH Zürich

<sup>b</sup>Carnegie Mellon University

<sup>c</sup>University of Illinois at Urbana-Champaign

<sup>d</sup>King Mongkut's University of Technology North Bangkok

---

## Abstract

Modern computing systems are overwhelmingly designed to move data to computation. This design choice goes directly against at least three key trends in computing that cause performance, scalability and energy bottlenecks: (1) data access is a key bottleneck as many important applications are increasingly data-intensive, and memory bandwidth and energy do not scale well, (2) energy consumption is a key limiter in almost all computing platforms, especially server and mobile systems, (3) data movement, especially off-chip to on-chip, is very expensive in terms of bandwidth, energy and latency, much more so than computation. These trends are especially severely-felt in the data-intensive server and energy-constrained mobile systems of today.

At the same time, conventional memory technology is facing many technology scaling challenges in terms of reliability, energy, and performance. As a result, memory system architects are open to organizing memory in different ways and making it more intelligent, at the expense of higher cost. The emergence of 3D-stacked memory plus logic, the adoption of error correcting codes inside the latest DRAM chips, proliferation of different main memory standards and chips, specialized for different purposes (e.g., graphics, low-power, high bandwidth, low latency), and the necessity of designing new solutions to serious reliability and security issues, such as the RowHammer phenomenon, are an evidence of this trend.

This chapter discusses recent research that aims to practically enable computation close to data, an approach we call *processing-in-memory* (PIM). PIM places computation mechanisms in or near where the data is stored (i.e., inside the memory chips, in the logic layer of 3D-stacked memory, or in the memory controllers), so that data movement between the computation units and memory is reduced or eliminated. While the general idea of PIM is not new, we discuss motivating trends in applications as well as memory circuits/technology that greatly exacerbate the need for enabling it in modern computing systems. We examine at least two promising new approaches to designing PIM systems to accelerate important data-intensive applications: (1) *processing using memory* by exploiting analog operational properties of DRAM chips to perform massively-parallel operations in memory, with low-cost changes, (2) *processing near memory* by exploiting 3D-stacked memory technology design to provide high memory bandwidth and low memory latency to in-memory logic. In both approaches, we describe and tackle relevant cross-layer research, design, and adoption challenges in devices, architecture, systems, and programming models. Our focus is on the development of in-memory processing designs that can be adopted in real computing platforms at low cost. We conclude by discussing work on solving key challenges to the practical adoption of PIM.

**Keywords:** memory systems, data movement, main memory, processing-in-memory, near-data processing, computation-in-memory, processing using memory, processing near memory, 3D-stacked memory, non-volatile memory, energy efficiency, high-performance computing, computer architecture, computing paradigm, emerging technologies, memory scaling, technology scaling, dependable systems, robust systems, hardware security, system security, latency, low-latency computing

|                                                                                                |           |
|------------------------------------------------------------------------------------------------|-----------|
| <b>1 Introduction</b>                                                                          | <b>2</b>  |
| <b>2 Major Trends Affecting Main Memory</b>                                                    | <b>4</b>  |
| <b>3 The Need for Intelligent Memory Controllers to Enhance Memory Scaling</b>                 | <b>6</b>  |
| <b>4 Perils of Processor-Centric Design</b>                                                    | <b>9</b>  |
| <b>5 Processing-in-Memory (PIM): Technology Enablers and Two Approaches</b>                    | <b>12</b> |
| 5.1 New Technology Enablers: 3D-Stacked Memory and Non-Volatile Memory . . .                   | 12        |
| 5.2 Two Approaches: Processing Using Memory (PUM) vs. Processing Near Memory (PNM) . . . . .   | 13        |
| <b>6 Processing Using Memory (PUM)</b>                                                         | <b>14</b> |
| 6.1 RowClone . . . . .                                                                         | 14        |
| 6.2 Ambit . . . . .                                                                            | 15        |
| 6.3 Gather-Scatter DRAM . . . . .                                                              | 17        |
| 6.4 In-DRAM Security Primitives . . . . .                                                      | 17        |
| <b>7 Processing Near Memory (PNM)</b>                                                          | <b>18</b> |
| 7.1 Tesseract: Coarse-Grained Application-Level PNM Acceleration of Graph Processing . . . . . | 19        |
| 7.2 Function-Level PNM Acceleration of Mobile Consumer Workloads . . . . .                     | 20        |
| 7.3 Programmer-Transparent Function-Level PNM Acceleration of GPU Applications . . . . .       | 21        |
| 7.4 Instruction-Level PNM Acceleration with PIM-Enabled Instructions (PEI) . .                 | 21        |
| 7.5 Function-Level PNM Acceleration of Genome Analysis Workloads . . . . .                     | 22        |
| 7.6 Application-Level PNM Acceleration of Time Series Analysis . . . . .                       | 23        |
| <b>8 Enabling the Adoption of PIM</b>                                                          | <b>24</b> |
| 8.1 Programming Models and Code Generation for PIM . . . . .                                   | 24        |
| 8.2 PIM Runtime: Scheduling and Data Mapping . . . . .                                         | 25        |
| 8.3 Memory Coherence . . . . .                                                                 | 27        |
| 8.4 Virtual Memory Support . . . . .                                                           | 27        |
| 8.5 Data Structures for PIM . . . . .                                                          | 28        |
| 8.6 Benchmarks and Simulation Infrastructures . . . . .                                        | 29        |
| 8.7 Real PIM Hardware Systems and Prototypes . . . . .                                         | 30        |
| 8.8 Security Considerations . . . . .                                                          | 30        |
| <b>9 Conclusion and Future Outlook</b>                                                         | <b>31</b> |

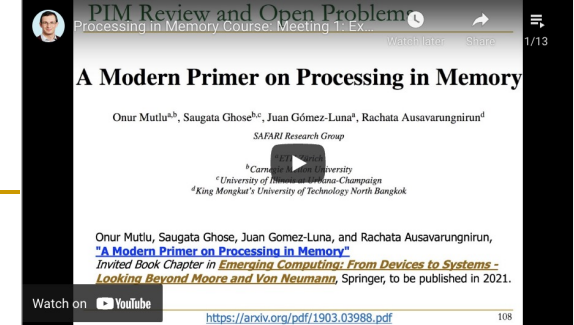
Main memory, built using the Dynamic Random Access Memory (DRAM) technology, is a major component in nearly all computing systems, including servers, cloud platforms, mobile/embedded devices, and sensor systems. Across all of these systems, the data working set sizes of modern applications are rapidly growing, while the need for fast analysis of such data is increasing. Thus, main memory is becoming an increasingly significant bottleneck across a wide variety of computing systems and applications [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Alleviating the main memory bottleneck requires the memory capacity, energy, cost, and performance to all scale in an efficient manner across technology generations. Unfortunately, it has become increasingly difficult in recent years, especially the past decade, to scale all of these dimensions [1, 2, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49], and thus the main memory bottleneck has been worsening.

A major reason for the main memory bottleneck is the high energy and latency cost associated with *data movement*. In modern computers, to perform any operation on data that resides in main memory, the processor must retrieve the data from main memory. This requires the memory controller to issue commands to a DRAM module across a relatively slow and power-hungry off-chip bus (known as the *memory channel*). The DRAM module sends the requested data across the memory channel, after which the data is placed in the caches and registers. The CPU can perform computation on the data once the data is in its registers. Data movement from the DRAM to the CPU incurs long latency and consumes a significant amount of energy [7, 50, 51, 52, 53, 54]. These costs are often exacerbated by the fact that much of the data brought into the caches is *not reused* by the CPU [52, 53, 55, 56], providing little benefit in return for the high latency and energy cost.

The cost of data movement is a fundamental issue with the *processor-centric* nature of contemporary computer systems. The CPU is considered to be the master in the system, and computation is performed only in the processor (and accelerators). In contrast, data storage and communication units, including the main memory, are treated as unintelligent workers that are incapable of computation. As a result of this processor-centric design paradigm, data moves a lot in the system between the computation units and communication/ storage units so that computation can be done on it. With the increasingly *data-centric* nature of contemporary and emerging appli-



# PIM Course (Spring 2022)



## Spring 2022 Meetings/Schedule

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

## Spring 2022 Edition:

- https://safari.ethz.ch/projects\_and\_seminar/s/spring2022/doku.php?id=processing\_in\_memory

## Youtube Livestream:

- 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>

We Need to Think Differently  
from the Past Approaches

# Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory



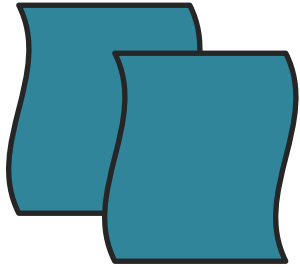
# Approach 1: Processing Using Memory

---

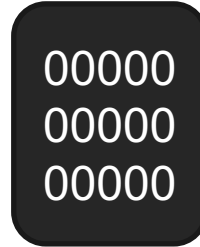
- Take advantage of operational principles of memory to perform **bulk data movement and computation in memory**
  - Can **exploit internal connectivity** to move data
  - Can **exploit analog computation capability**
  - ...
- Examples: RowClone, In-DRAM AND/OR, Gather/Scatter DRAM
  - RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data (Seshadri et al., MICRO 2013)
  - Fast Bulk Bitwise AND and OR in DRAM (Seshadri et al., IEEE CAL 2015)
  - Gather-Scatter DRAM: In-DRAM Address Translation to Improve the Spatial Locality of Non-unit Strided Accesses (Seshadri et al., MICRO 2015)
  - "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology" (Seshadri et al., MICRO 2017)

# Starting Simple: Data Copy and Initialization

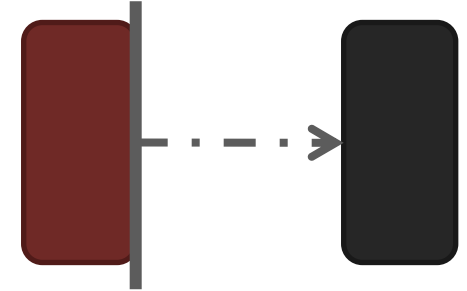
*memmove & memcpy: 5% cycles in Google's datacenter [Kanev+ ISCA'15]*



**Forking**



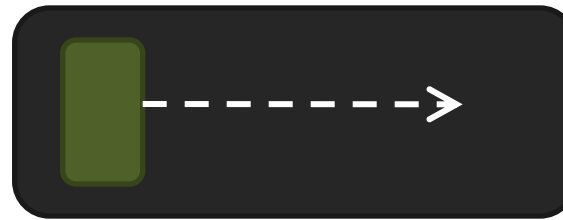
**Zero initialization  
(e.g., security)**



**Checkpointing**



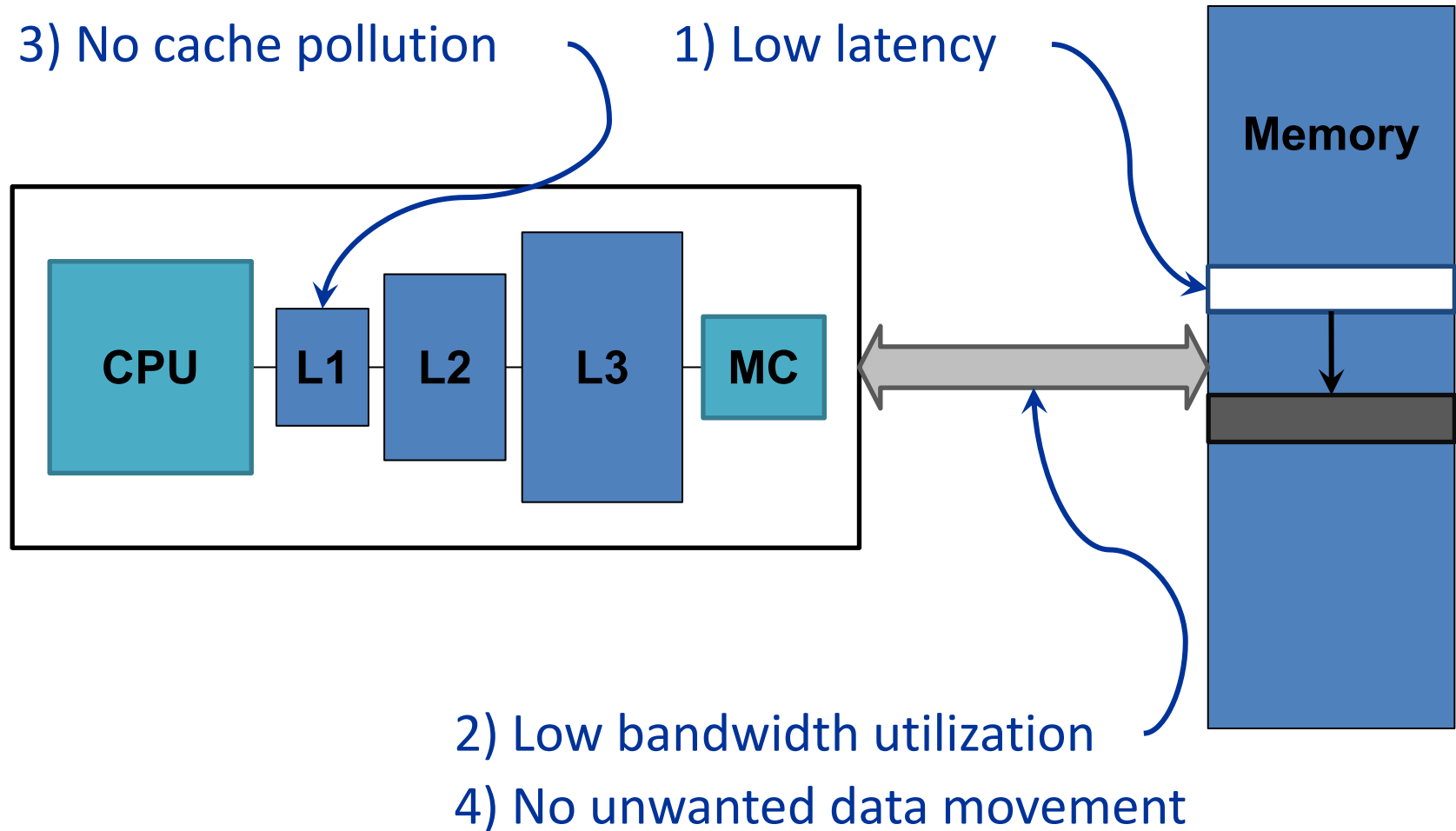
**VM Cloning  
Deduplication**



**Page Migration**

...  
**Many more**

# Future Systems: In-Memory Copy

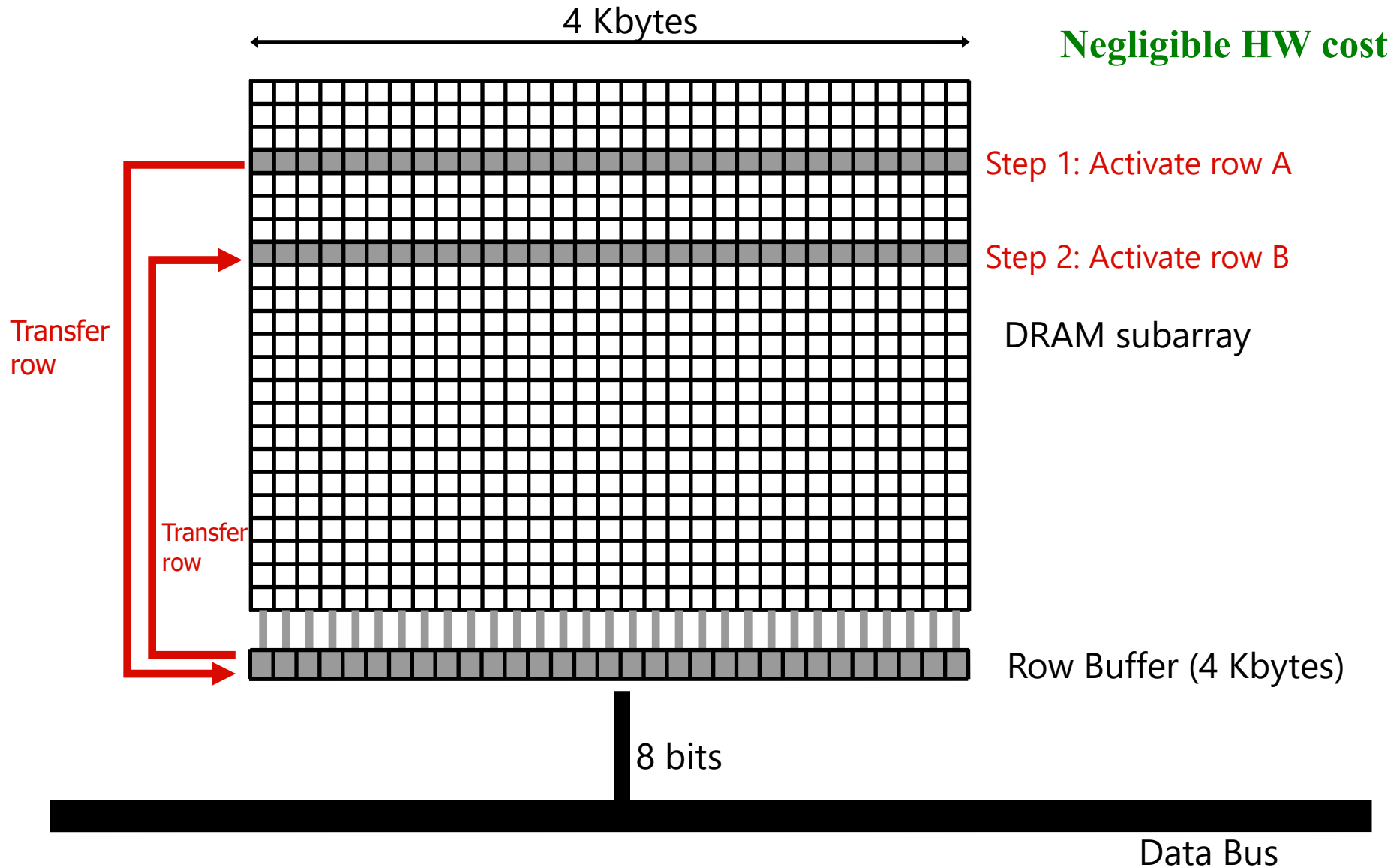


1046ns, 3.6uJ → 90ns, 0.04uJ

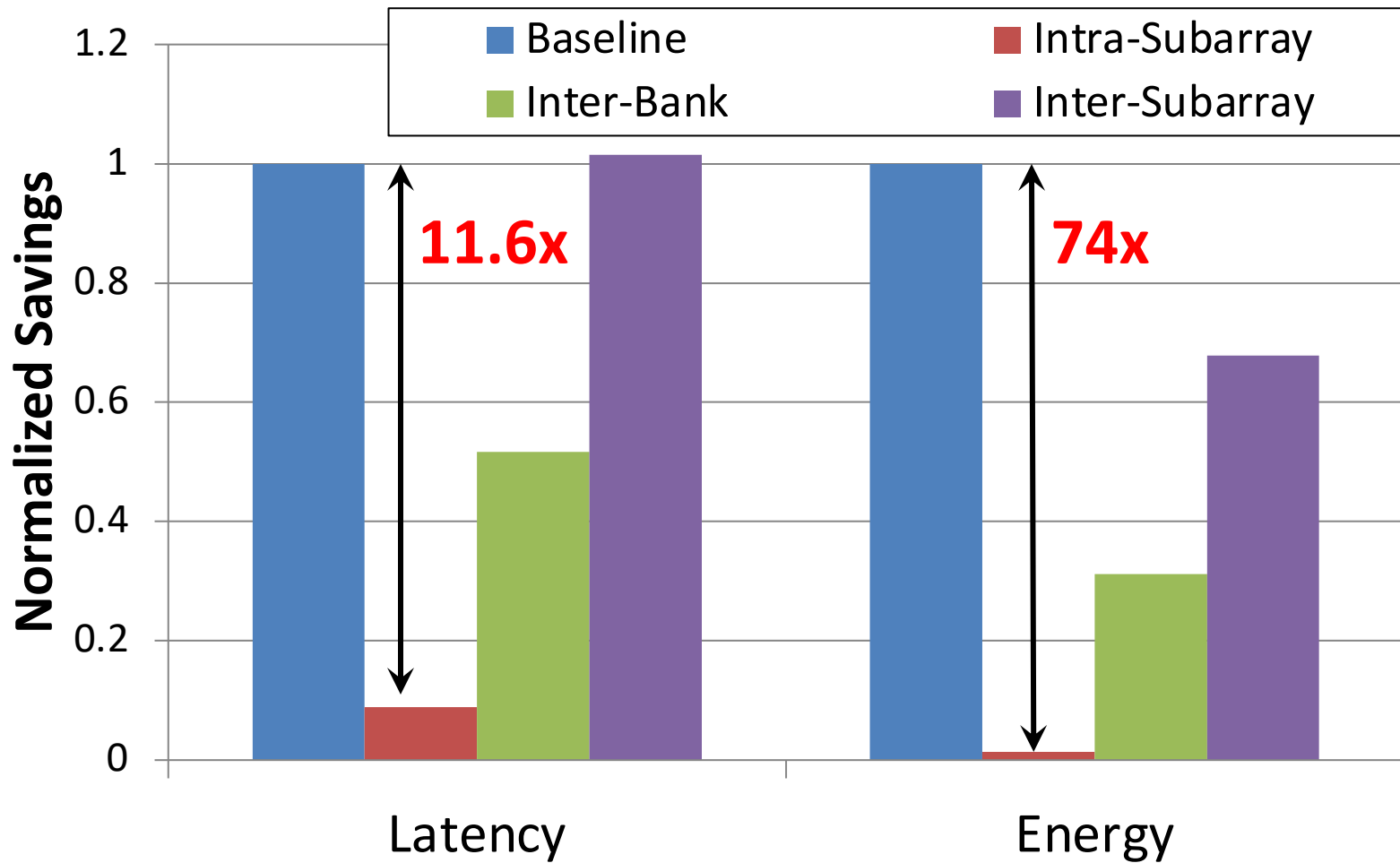
# RowClone: In-DRAM Row Copy

**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<sup>§†</sup>

Juan Gómez Luna<sup>§</sup>  
Hasan Hassan<sup>§</sup>

Konstantinos Kanellopoulos<sup>§</sup>  
Oğuz Ergin<sup>†</sup>    Onur Mutlu<sup>§</sup>

Behzad Salami<sup>§\*</sup>

<sup>§</sup>ETH Zürich

<sup>†</sup>TOBB ETÜ

<sup>\*</sup>BSC

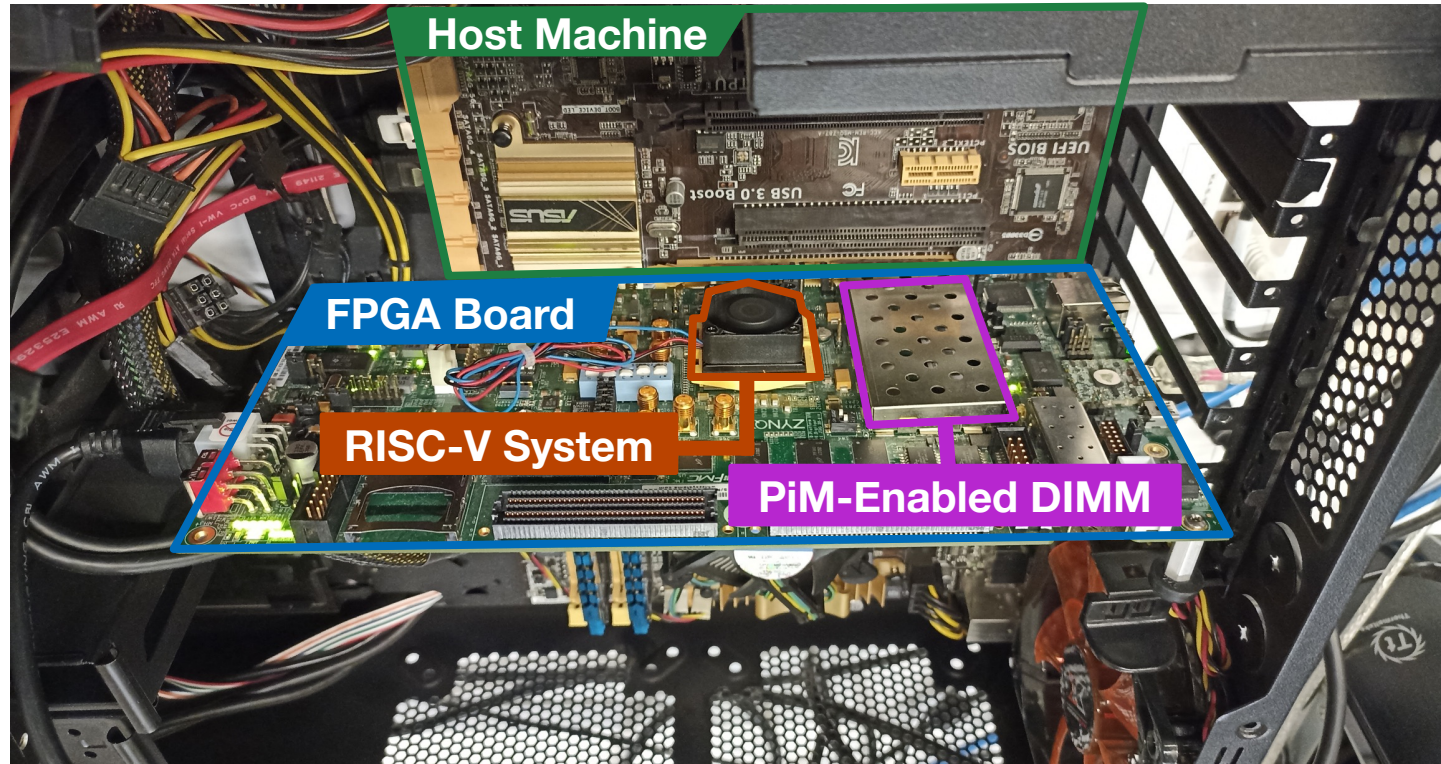
<https://arxiv.org/pdf/2111.00082.pdf>

<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:

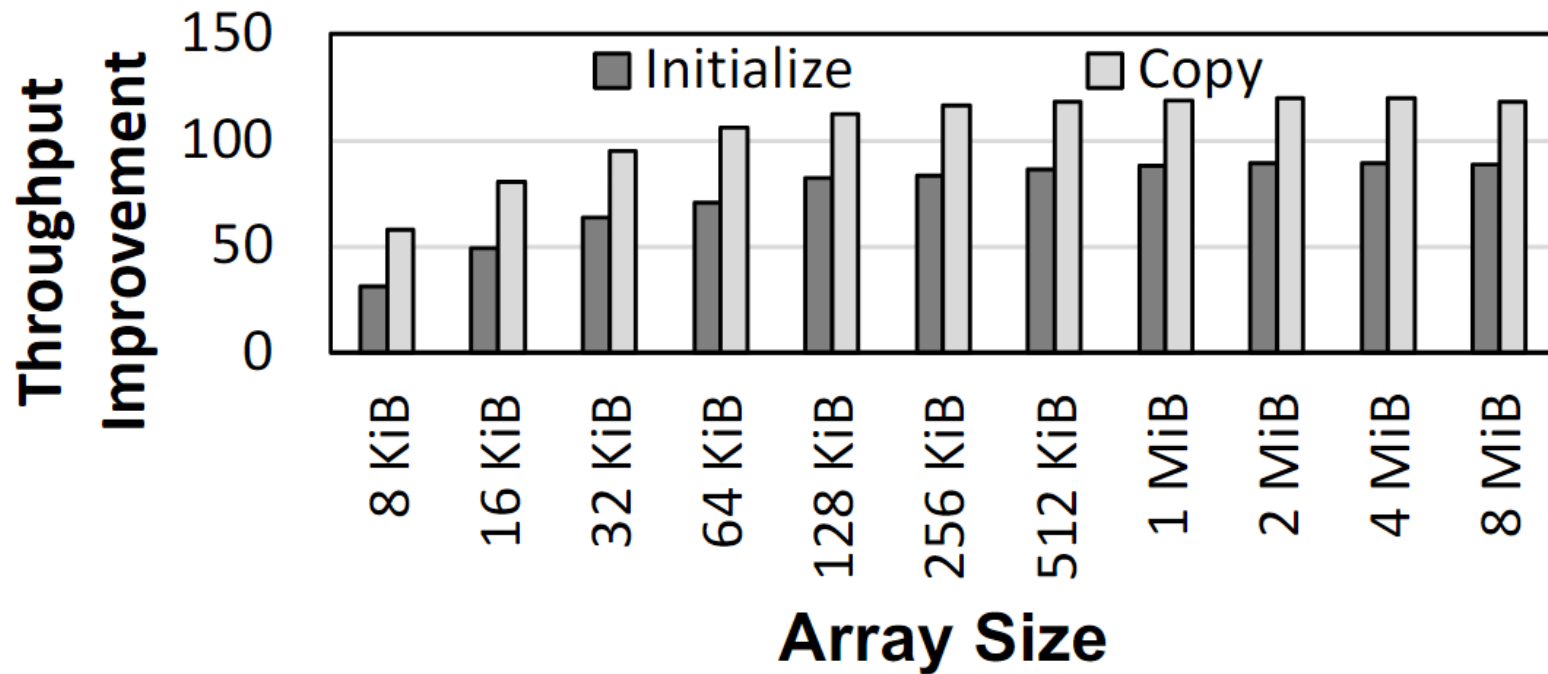
- 1- Open IP Catalog
- 2- Find "Memory Interface Generator (MIG 7 Series)" IP and double click

<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**



# 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 components: four 'CPU core' blocks, one 'mini-CPU core', one 'video core', one 'imaging core', and four 'GPU (throughput) core' blocks. Below these is a large 'LLC' (Last Level Cache) block, which is connected to a 'Memory Controller' block. The 'Memory Controller' is connected to a 'Memory Bus'. To the right of the system box is a large 'Memory' block. A red rounded rectangle highlights a 'Specialized compute-capability in memory' block within the memory, which is also connected to the 'Memory Bus'. A video player interface is overlaid on the bottom of the diagram, showing a red text overlay: 'Memory similar to a "conventional" accelerator'. The video player also shows a progress bar at 43:48 / 2:03:45 and various control icons.

Onur Mutlu

Memory

Specialized compute-capability in memory

Memory Bus

LLC

Memory Controller

CPU core

mini-CPU core

video core

imaging core

GPU (throughput) core

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

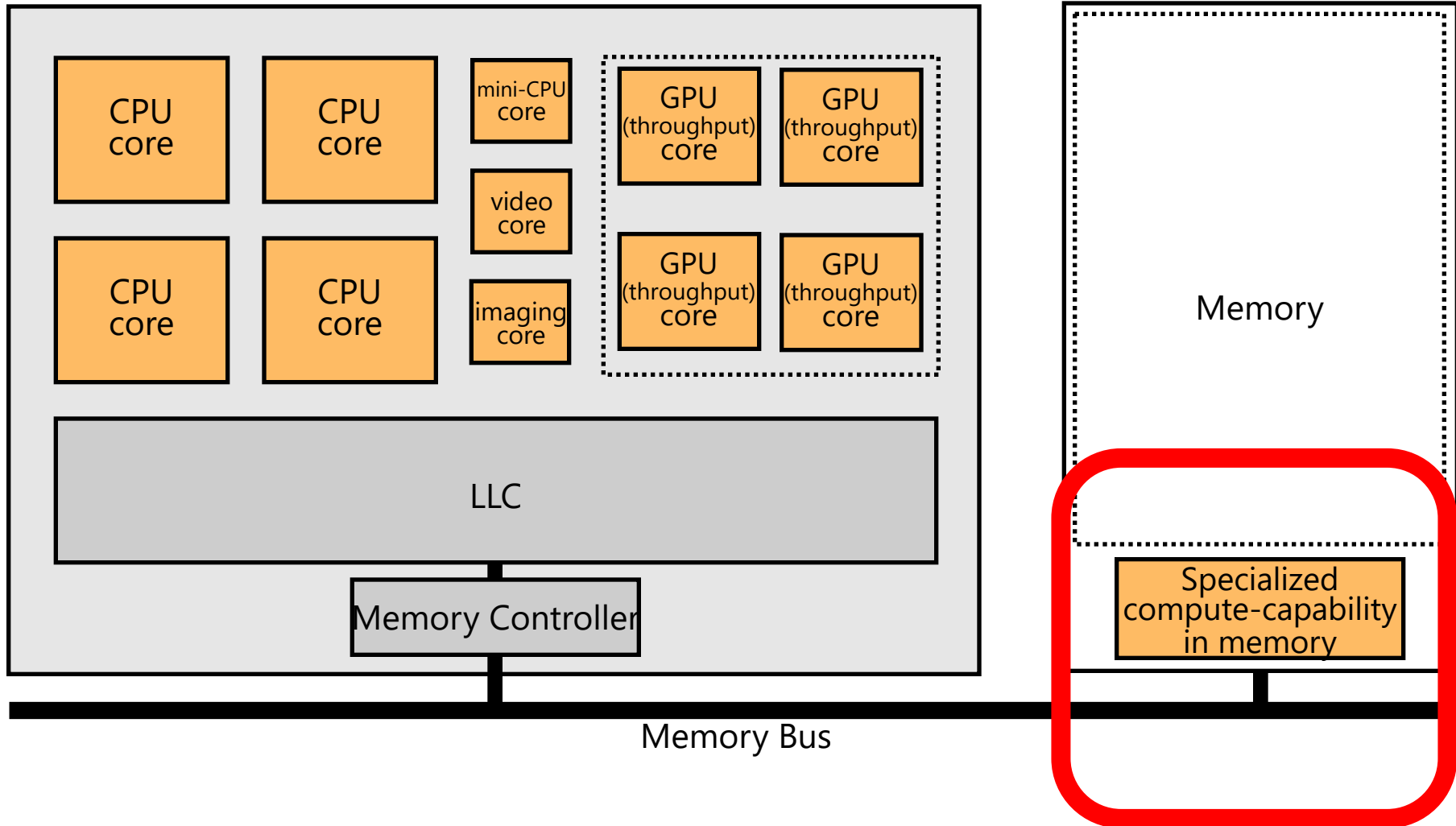
SUBSCRIBED



[https://www.youtube.com/watch?v=n6Pwg1qax\\_E&list=PL5Q2soXY2Zi\\_7UBNmC9B8Yr5JSwTG9yH4&index=4](https://www.youtube.com/watch?v=n6Pwg1qax_E&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4&index=4)



# Mindset: Memory as an Accelerator



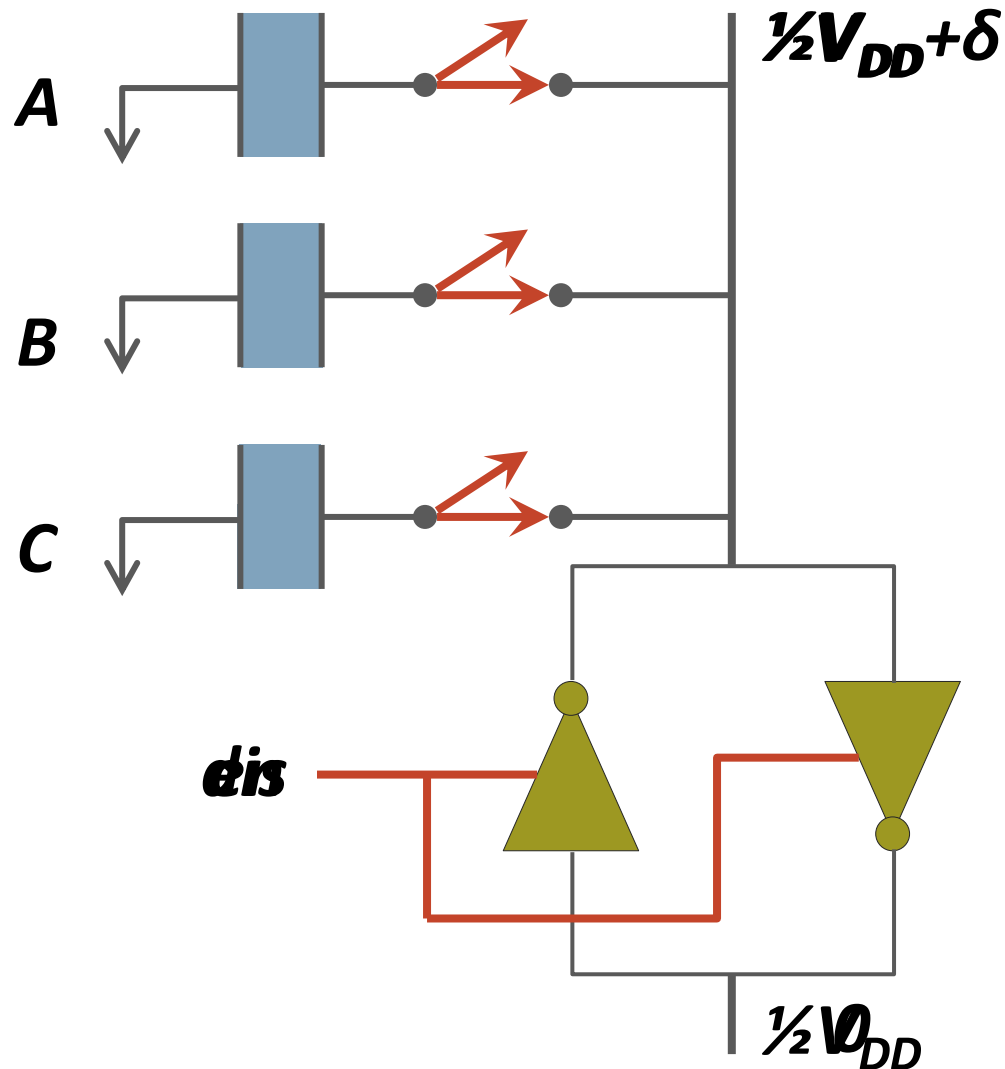
**Memory similar to a "conventional" accelerator**

# (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

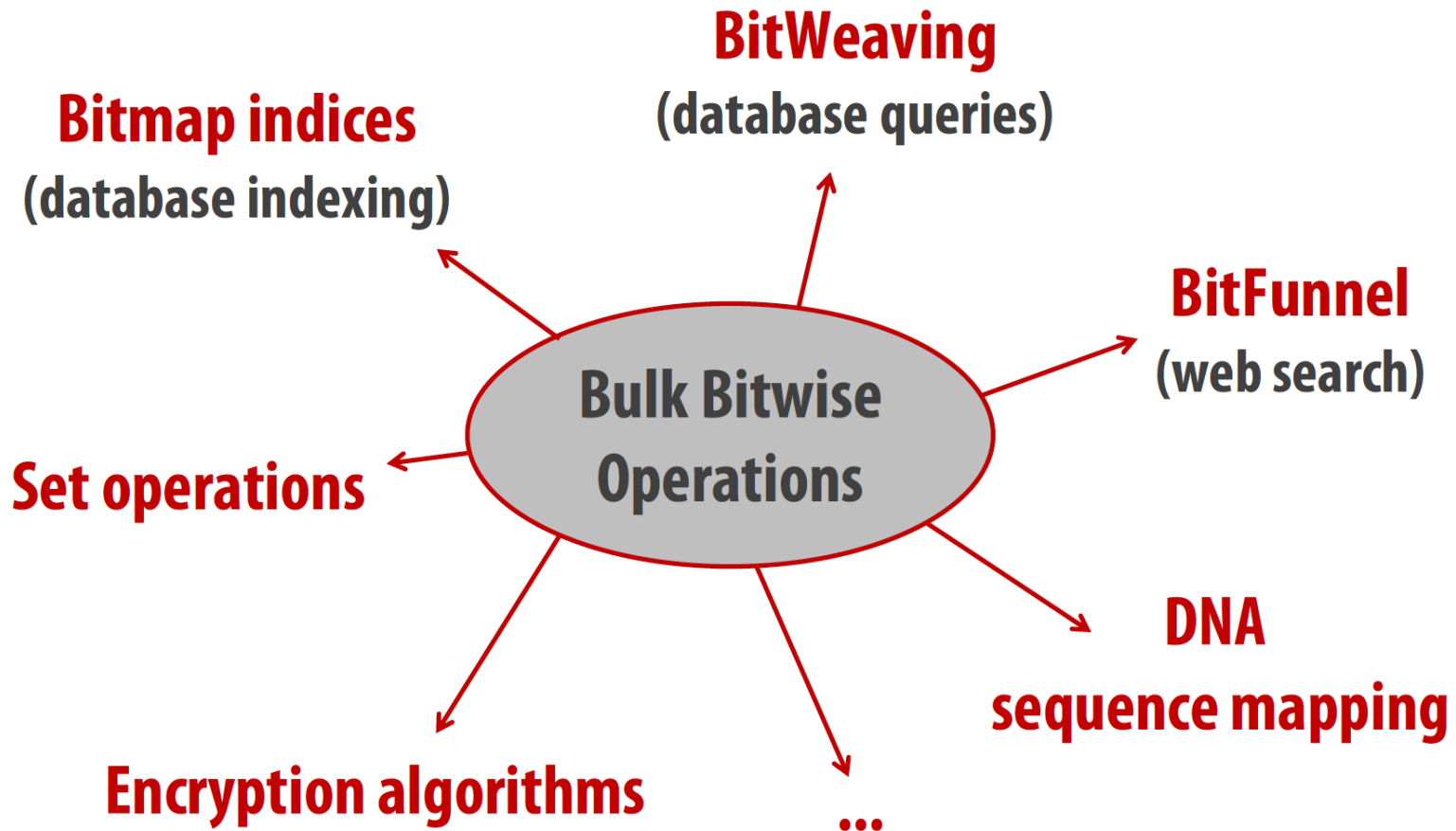


**Final State**  
 **$AB + BC + AC$**

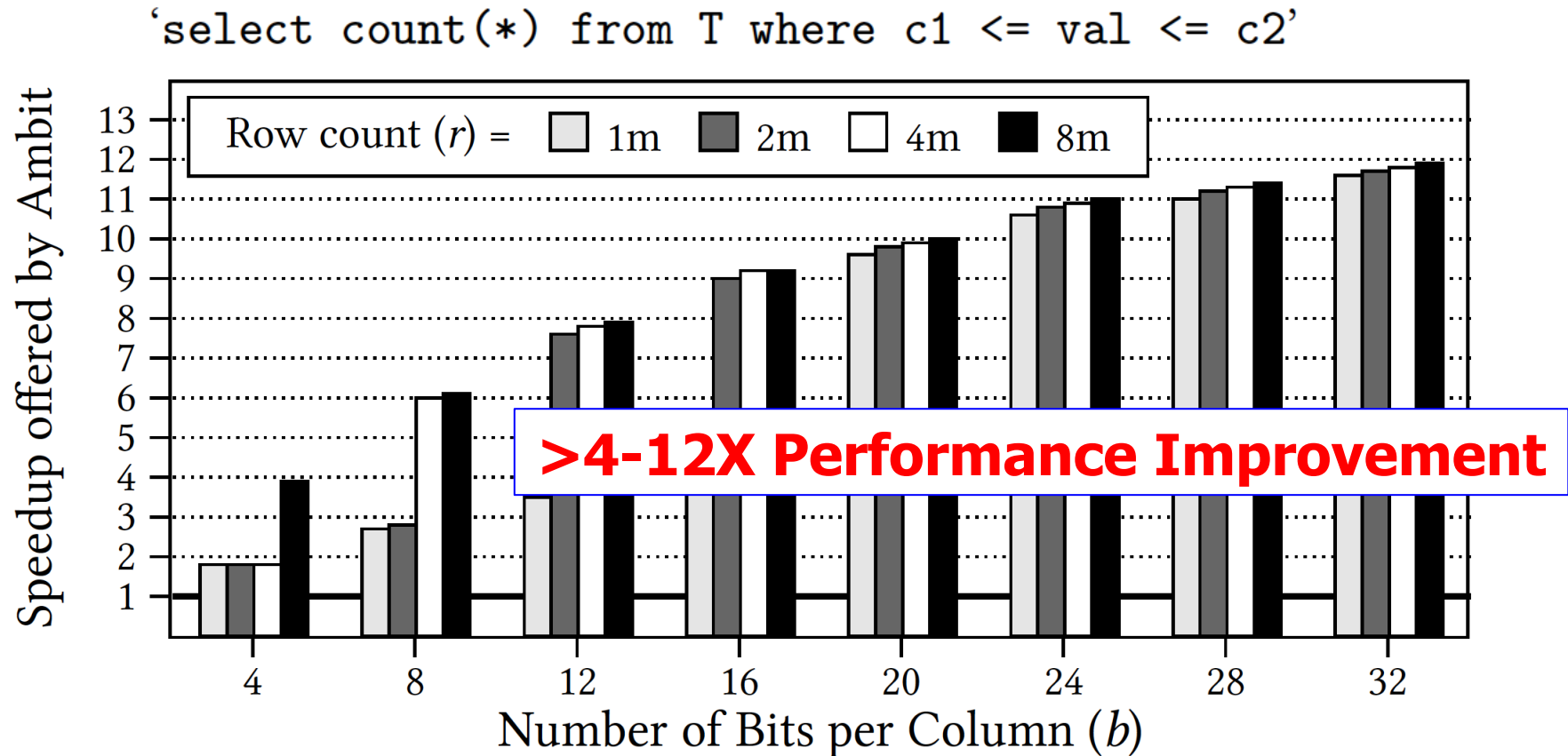
**$C(A + B) +$   
 **$\sim C(AB)$****

# 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 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<sup>1,5</sup> Donghyuk Lee<sup>2,5</sup> Thomas Mullins<sup>3,5</sup> Hasan Hassan<sup>4</sup> Amirali Boroumand<sup>5</sup>  
Jeremie Kim<sup>4,5</sup> Michael A. Kozuch<sup>3</sup> Onur Mutlu<sup>4,5</sup> Phillip B. Gibbons<sup>5</sup> Todd C. Mowry<sup>5</sup>

<sup>1</sup>Microsoft Research India   <sup>2</sup>NVIDIA Research   <sup>3</sup>Intel   <sup>4</sup>ETH Zürich   <sup>5</sup>Carnegie 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

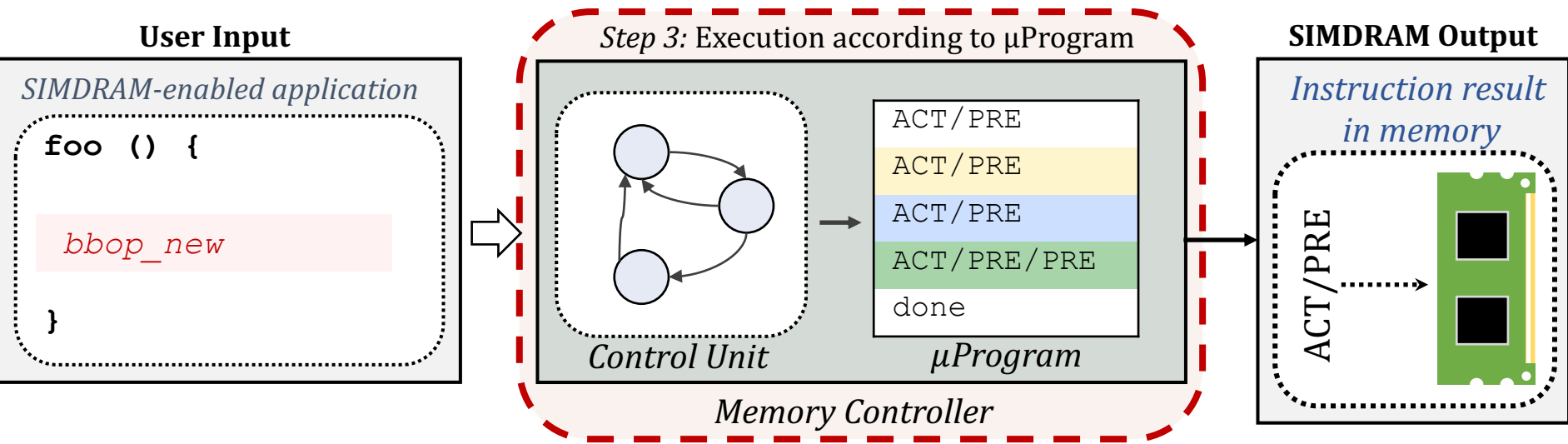
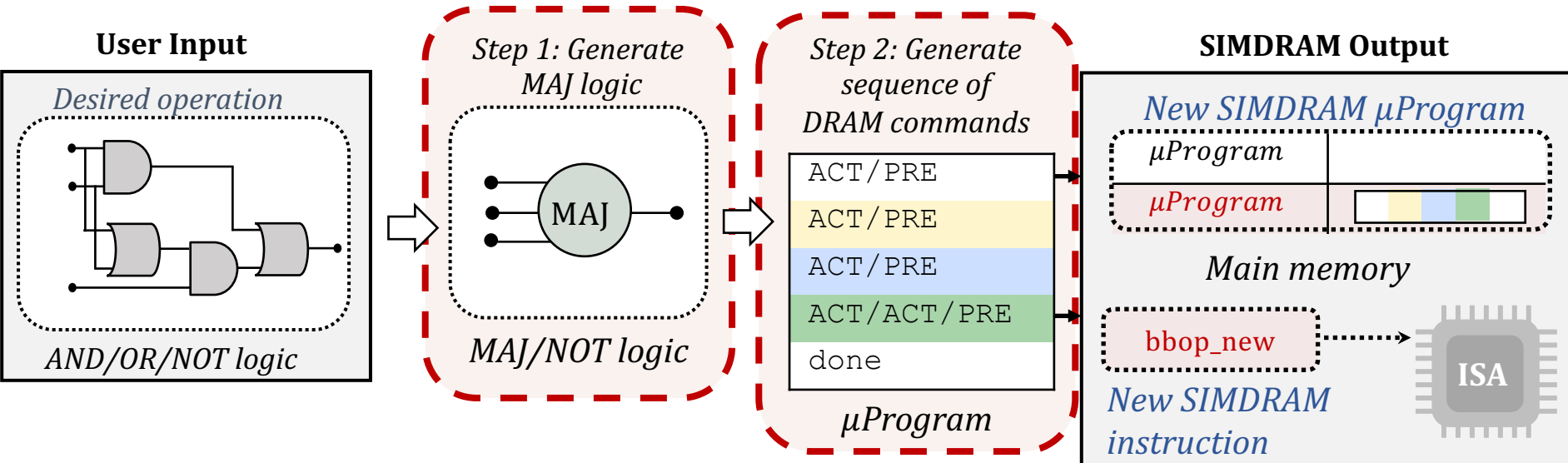
|                                    |                                   |                             |                                  |
|------------------------------------|-----------------------------------|-----------------------------|----------------------------------|
| *Nastaran Hajinazar <sup>1,2</sup> | *Geraldo F. Oliveira <sup>1</sup> | Sven Gregorio <sup>1</sup>  | João Dinis Ferreira <sup>1</sup> |
| Nika Mansouri Ghiasi <sup>1</sup>  | Minesh Patel <sup>1</sup>         | Mohammed Alser <sup>1</sup> | Saugata Ghose <sup>3</sup>       |
|                                    | Juan Gómez-Luna <sup>1</sup>      | Onur Mutlu <sup>1</sup>     |                                  |

<sup>1</sup>ETH Zürich

<sup>2</sup>Simon Fraser University

<sup>3</sup>University 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 <sup>1,2</sup> | *Geraldo F. Oliveira <sup>1</sup> | Sven Gregorio <sup>1</sup>  | João Dinis Ferreira <sup>1</sup> |
| Nika Mansouri Ghiasi <sup>1</sup>  | Minesh Patel <sup>1</sup>         | Mohammed Alser <sup>1</sup> | Saugata Ghose <sup>3</sup>       |
|                                    | Juan Gómez-Luna <sup>1</sup>      | Onur Mutlu <sup>1</sup>     |                                  |

<sup>1</sup>ETH Zürich

<sup>2</sup>Simon Fraser University

<sup>3</sup>University of Illinois at Urbana–Champaign

# In-DRAM Lookup-Table Based Execution

- Appears at MICRO 2022



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

João Dinis Ferreira<sup>§</sup>

Gabriel Falcao<sup>†</sup>

Juan Gómez-Luna<sup>§</sup>

Mohammed Alser<sup>§</sup>

Lois Orosa<sup>§∇</sup>

Mohammad Sadrosadati<sup>§</sup>

Jeremie S. Kim<sup>§</sup>

Geraldo F. Oliveira<sup>§</sup>

Taha Shahroodi<sup>‡</sup>

Anant Nori<sup>\*</sup>

Onur Mutlu<sup>§</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*IT, University of Coimbra*

<sup>∇</sup>*Galicia Supercomputing Center*

<sup>‡</sup>*TU Delft*

<sup>\*</sup>*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<sup>†§</sup>

Minesh Patel<sup>§</sup>

Hasan Hassan<sup>§</sup>

Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>§</sup>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<sup>‡§</sup>

Minesh Patel<sup>§</sup>

Hasan Hassan<sup>§</sup>

Lois Orosa<sup>§</sup>

Onur Mutlu<sup>§‡</sup>

<sup>‡</sup>Carnegie Mellon University

<sup>§</sup>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<sup>§†</sup>

Minesh Patel<sup>§</sup>

A. Giray Yağlıkçı<sup>§</sup>

Haocong Luo<sup>§</sup>

Jeremie S. Kim<sup>§</sup>

F. Nisa Bostanci<sup>§†</sup>

Nandita Vijaykumar<sup>§⊙</sup>

Oğuz Ergin<sup>†</sup>

Onur Mutlu<sup>§</sup>

<sup>§</sup>ETH Zürich

<sup>†</sup>TOBB University of Economics and Technology

<sup>⊙</sup>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<sup>†§</sup>

Ataberk Olgun<sup>†§</sup>

Lois Orosa<sup>§</sup>

A. Giray Yağlıkçı<sup>§</sup>

Jeremie S. Kim<sup>§</sup>

Hasan Hassan<sup>§</sup>

Oğuz Ergin<sup>†</sup>

Onur Mutlu<sup>§</sup>

<sup>†</sup>*TOBB University of Economics and Technology*

<sup>§</sup>*ETH Zürich*

# In-Flash Bulk Bitwise Execution

---

- Appears at MICRO 2022

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

Jisung Park<sup>§∇</sup> Roknoddin Azizi<sup>§</sup> Geraldo F. Oliveira<sup>§</sup> Mohammad Sadrosadati<sup>§</sup>  
Rakesh Nadig<sup>§</sup> David Novo<sup>†</sup> Juan Gómez-Luna<sup>§</sup> Myungsuk Kim<sup>‡</sup> Onur Mutlu<sup>§</sup>

<sup>§</sup>*ETH Zürich*    <sup>∇</sup>*POSTECH*    <sup>†</sup>*LIRMM, Univ. Montpellier, CNRS*    <sup>‡</sup>*Kyungpook National University*

# Pinatubo: RowClone and Bitwise Ops in PCM

---

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

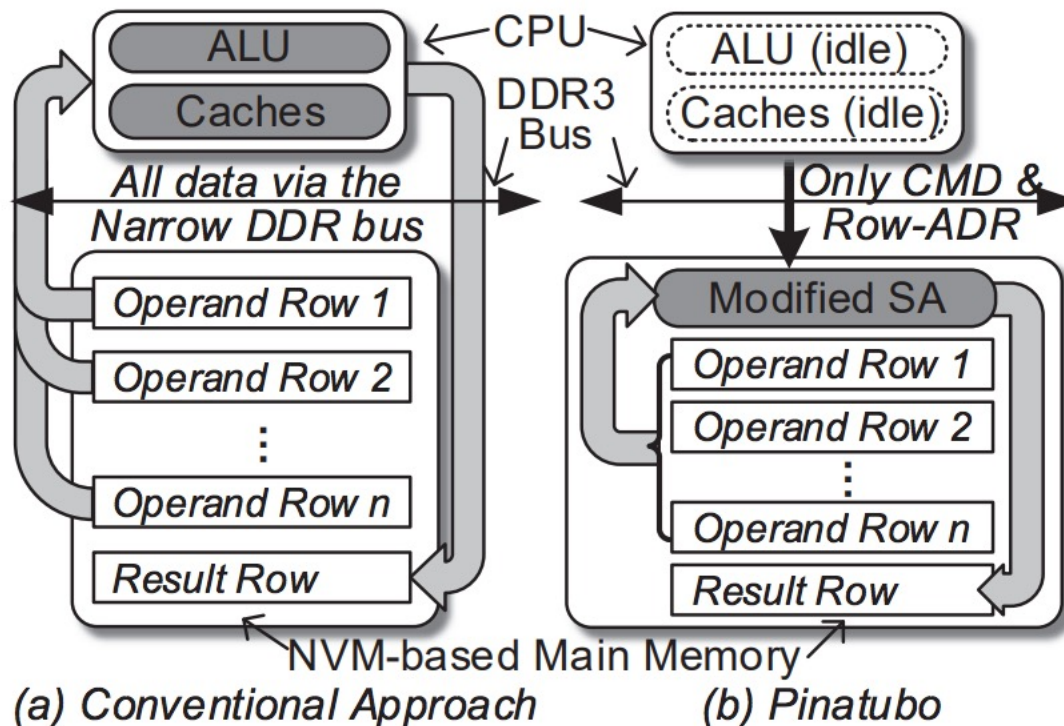
Shuangchen Li<sup>1\*</sup>, Cong Xu<sup>2</sup>, Qiaosha Zou<sup>1,5</sup>, Jishen Zhao<sup>3</sup>, Yu Lu<sup>4</sup>, and Yuan Xie<sup>1</sup>

University of California, Santa Barbara<sup>1</sup>, Hewlett Packard Labs<sup>2</sup>

University of California, Santa Cruz<sup>3</sup>, Qualcomm Inc.<sup>4</sup>, Huawei Technologies Inc.<sup>5</sup>  
{shuangchenli, yuanxie}@ece.ucsb.edu<sup>1</sup>



# Pinatubo: RowClone and Bitwise Ops in PCM



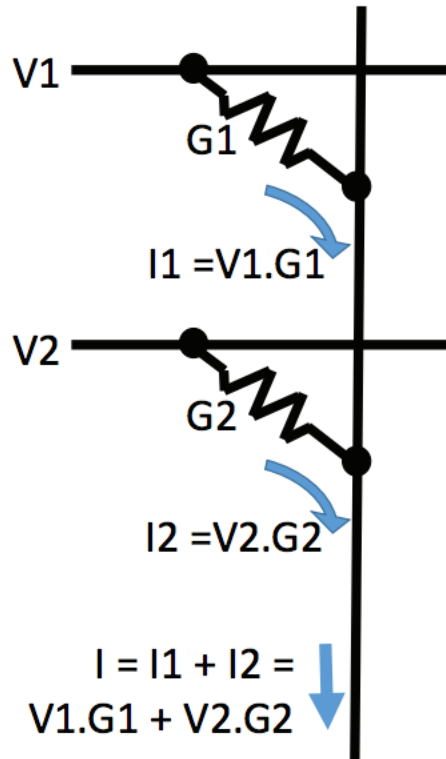
**Figure 2: Overview:** (a) Computing-centric approach, moving tons of data to CPU and write back. (b) The proposed Pinatubo architecture, performs  $n$ -row bitwise operations inside NVM in one step.

# In-Memory Crossbar Array Operations

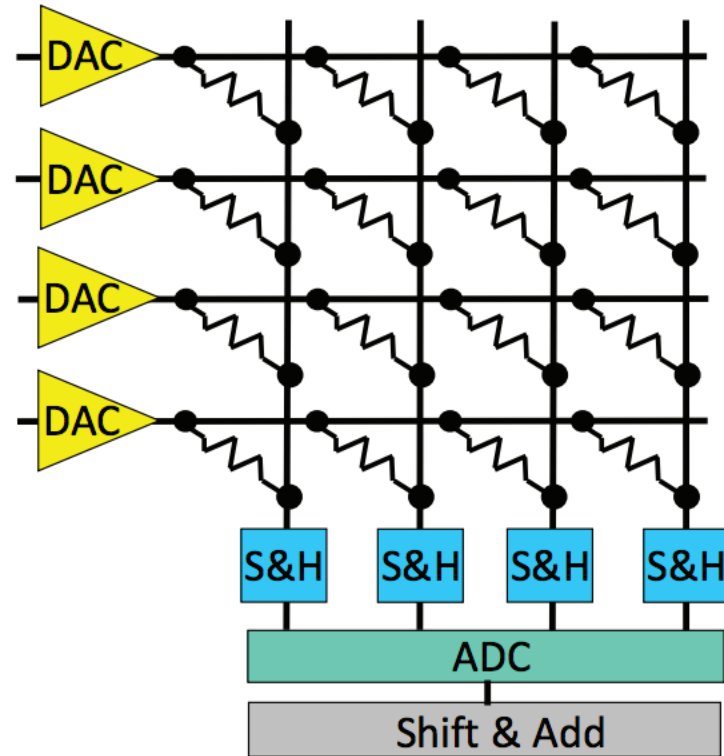
---

- Some emerging NVM technologies have crossbar array structure
  - Memristors, resistive RAM, phase change mem, STT-MRAM, ...
- Crossbar arrays can be used to perform dot product operations using “analog computation capability”
  - Can operate on multiple pieces of data using Kirchhoff's laws
    - Bitline current is a sum of products of wordline  $V \times (1 / \text{cell } R)$
  - Computation is in analog domain inside the crossbar array
- Need peripheral circuitry for  $D \rightarrow A$  and  $A \rightarrow D$  conversion of inputs and outputs

# Aside: In-Memory Crossbar Computation



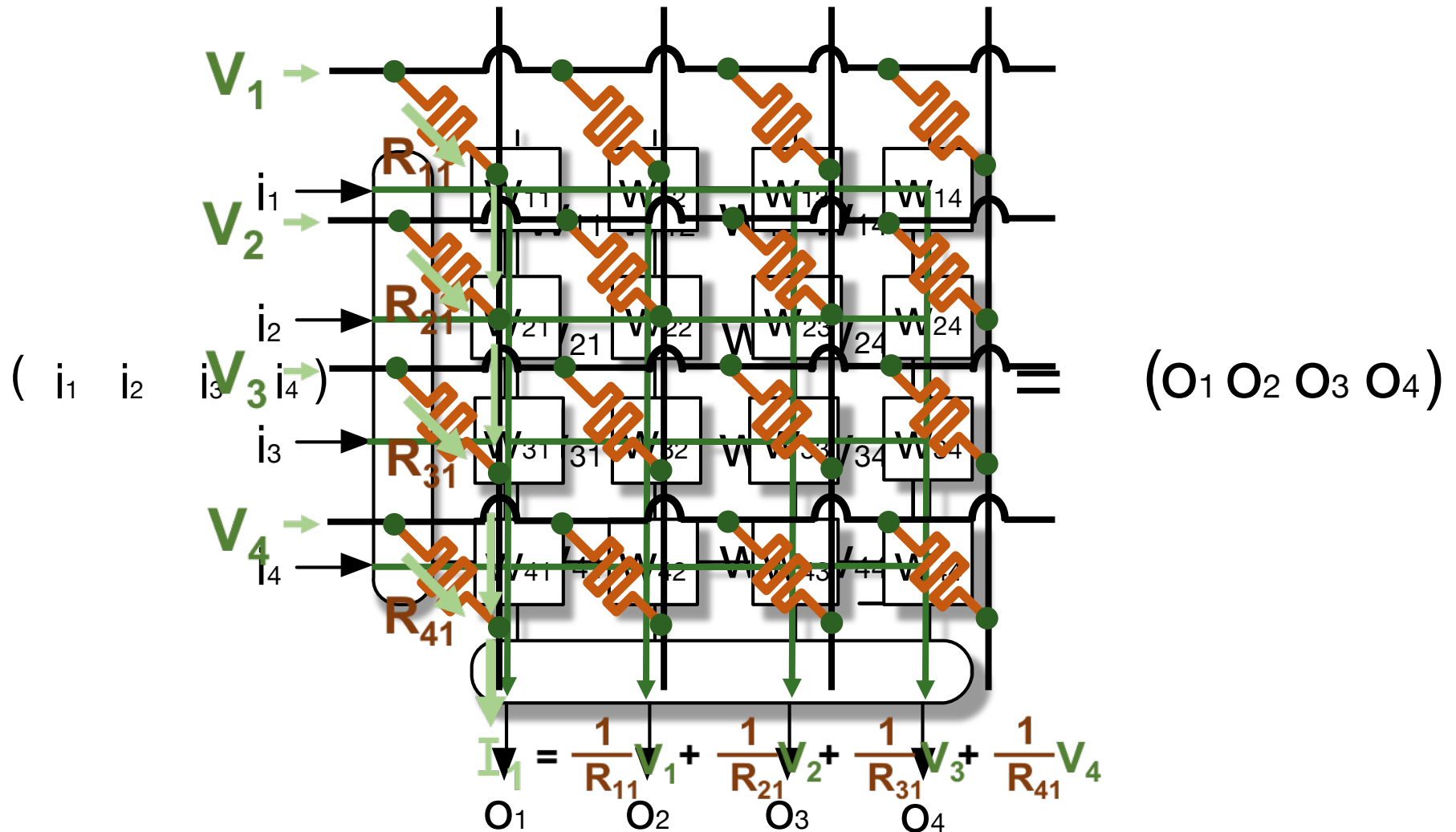
(a) Multiply-Accumulate operation



(b) Vector-Matrix Multiplier

Fig. 1. (a) Using a bitline to perform an analog sum of products operation. (b) A memristor crossbar used as a vector-matrix multiplier.

# Aside: In-Memory Crossbar Computation



# 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.

# Accelerating Genome Analysis w/ Processing using NVM

---

- Appears at MICRO 2022

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

Haiyu Mao<sup>1</sup> Mohammed Alser<sup>1</sup> Mohammad Sadrosadati<sup>1</sup> Can Firtina<sup>1</sup> Akanksha Baranwal<sup>1</sup>  
Damla Senol Cali<sup>2</sup> Aditya Manglik<sup>1</sup> Nour Almadhoun Alserr<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>*ETH Zürich*

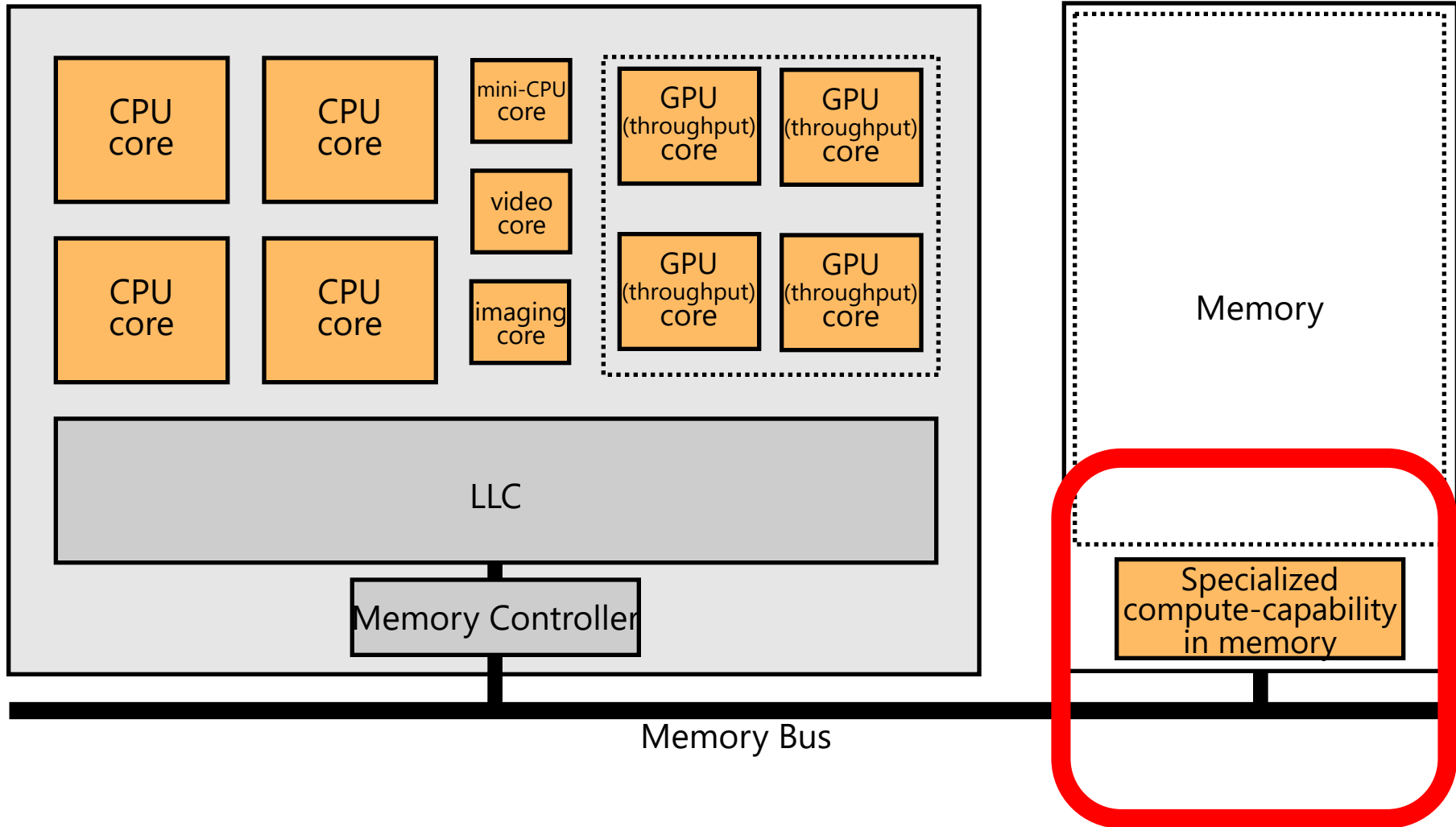
<sup>2</sup>*Bionano Genomics*



# Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory

# 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

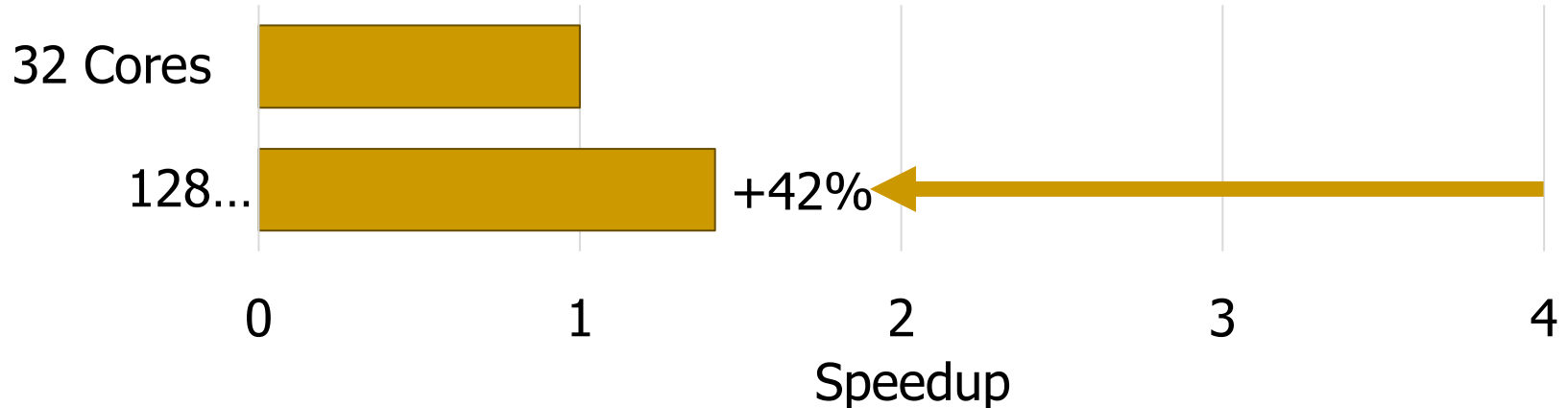


300 Million  
Twitter Users



30 Billion  
Instagram Photos

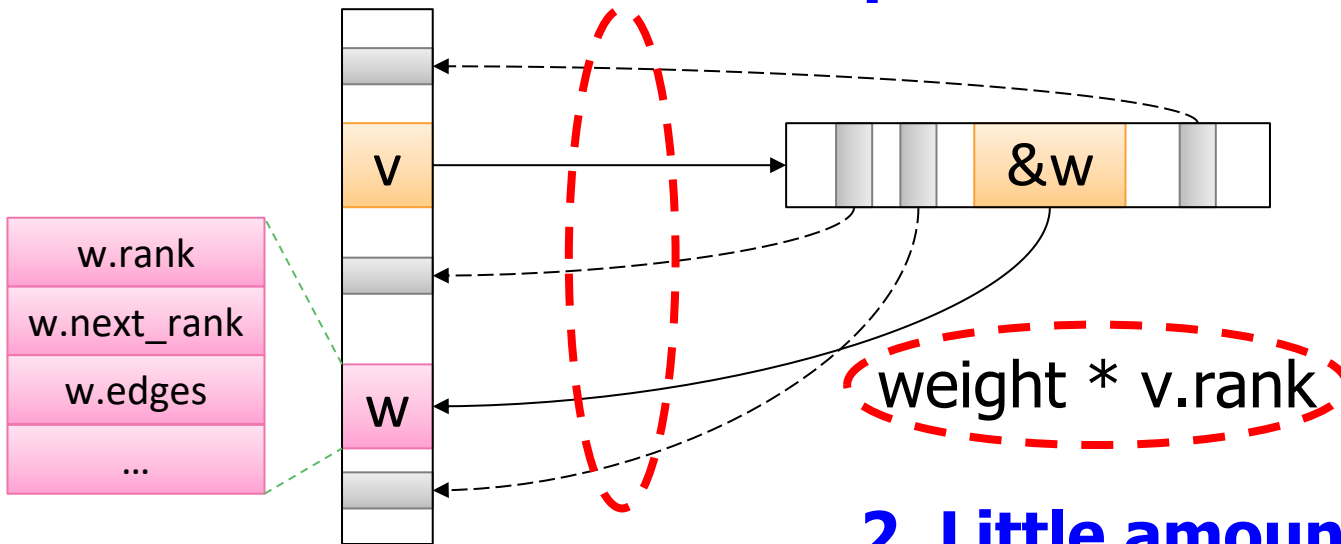
- Scalable large-scale graph processing is challenging



# Key Bottlenecks in Graph Processing

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

**1. Frequent random memory accesses**



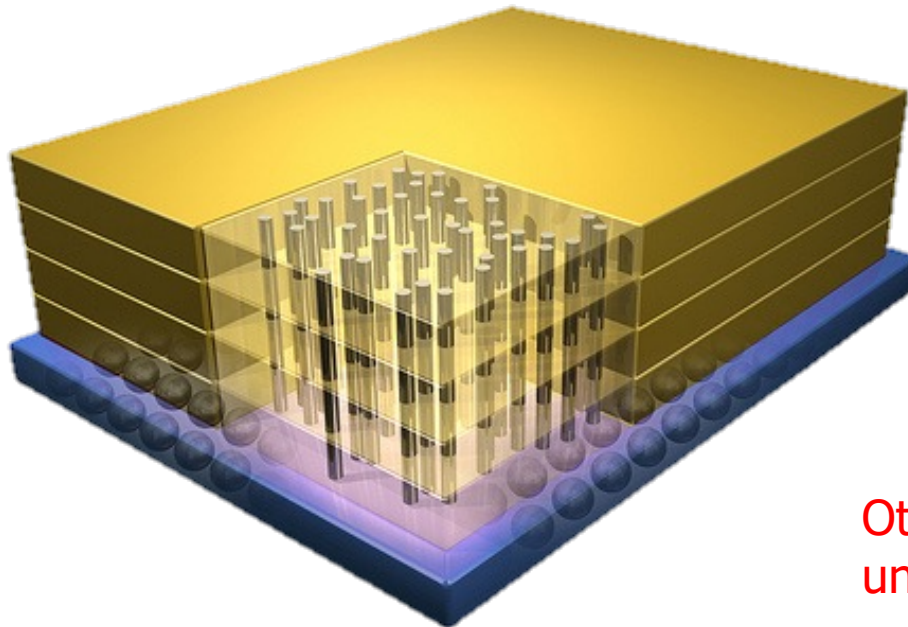
**2. Little amount of computation**

# Opportunity: 3D-Stacked Logic+Memory

---



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



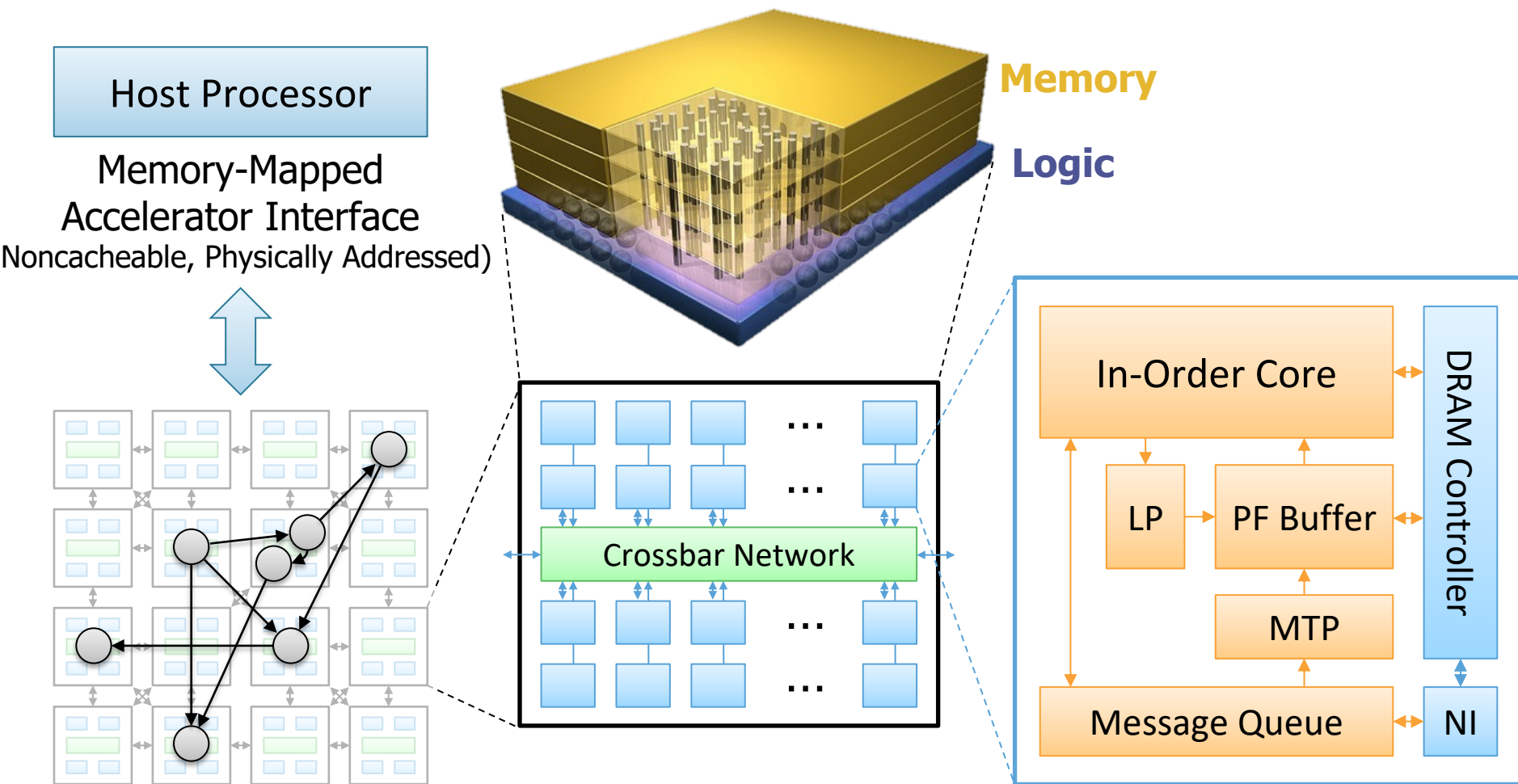
Memory

Logic

Other "True 3D" technologies  
under development

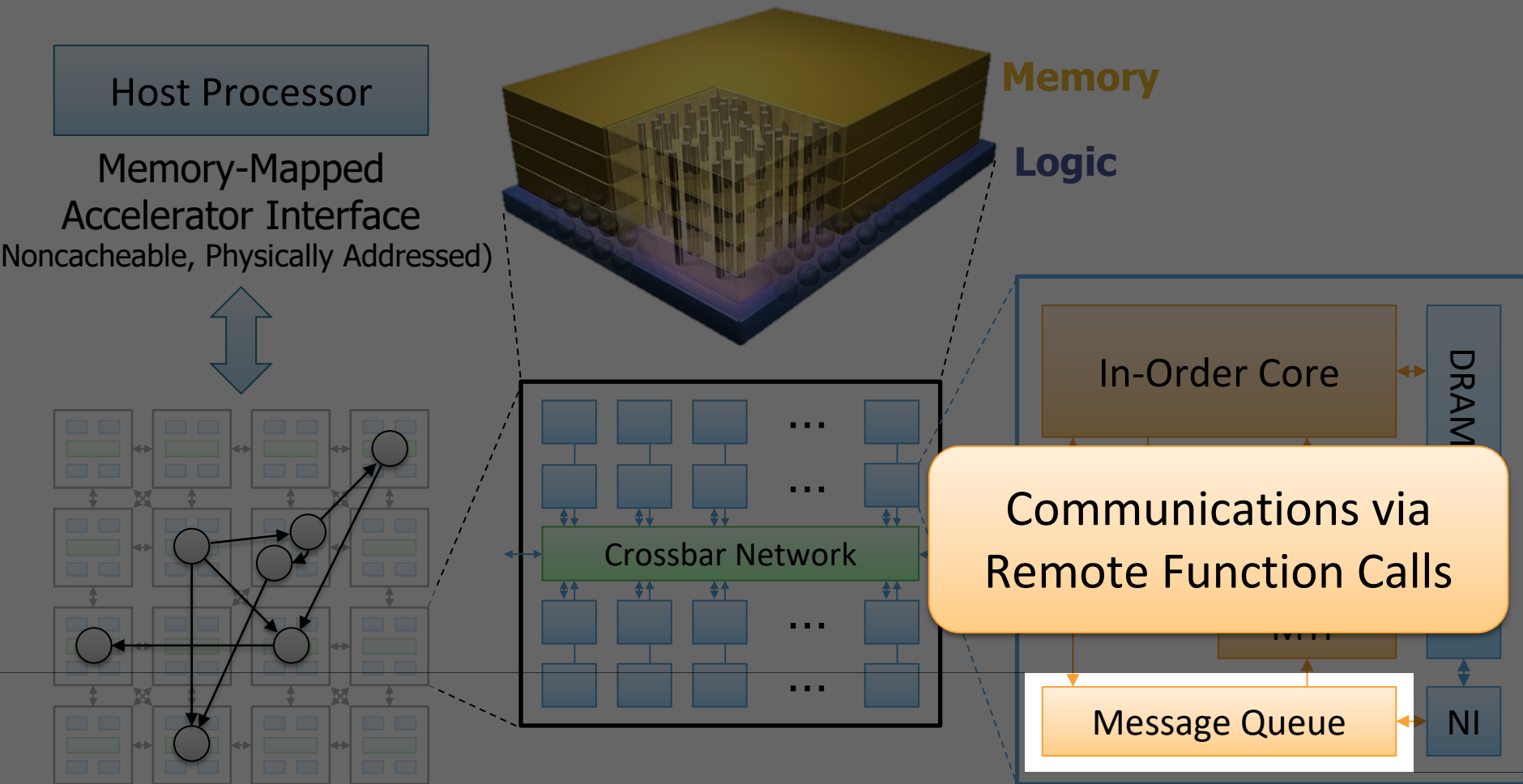
# Tesseract System for Graph Processing

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

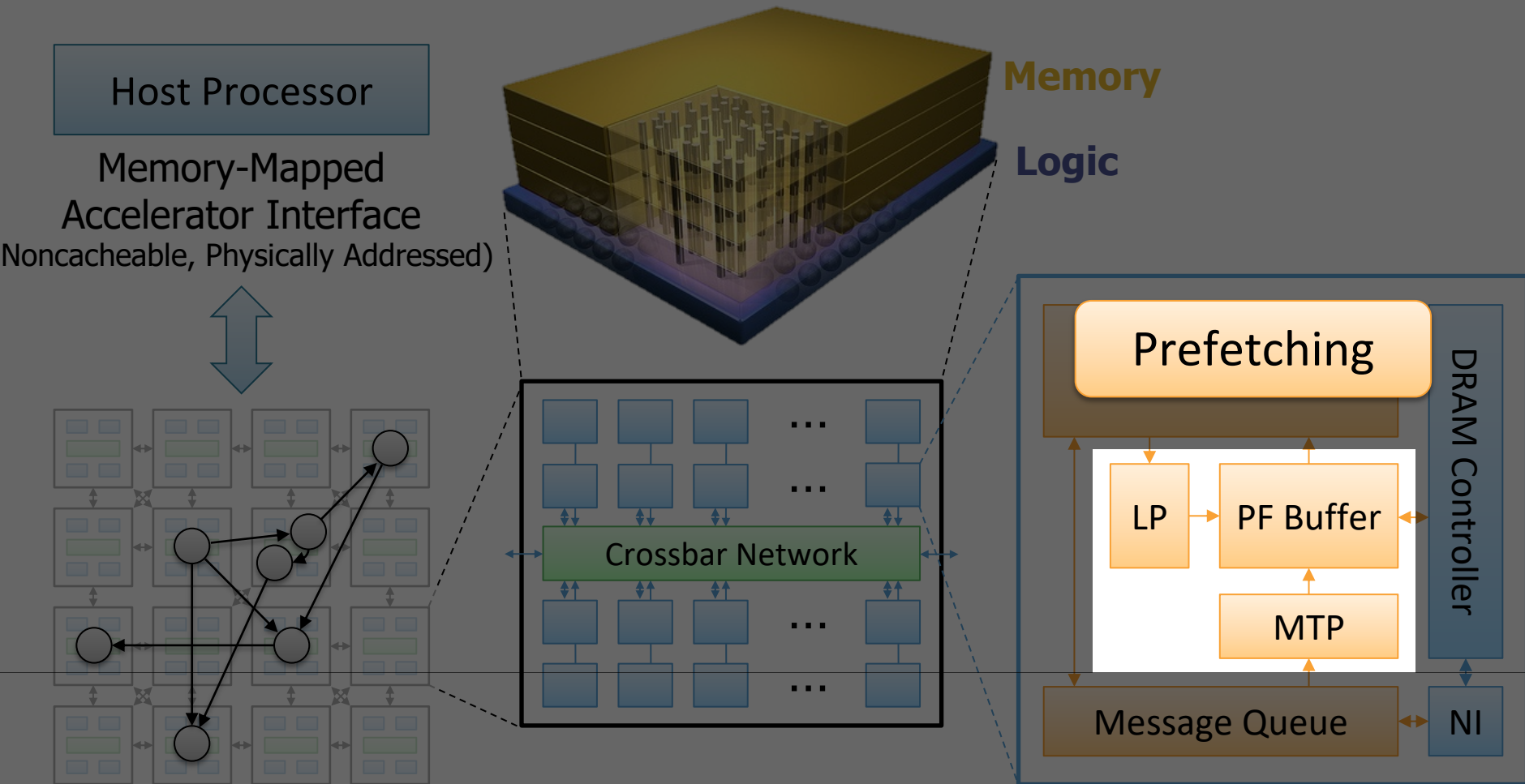




# Tesseract System for Graph Processing

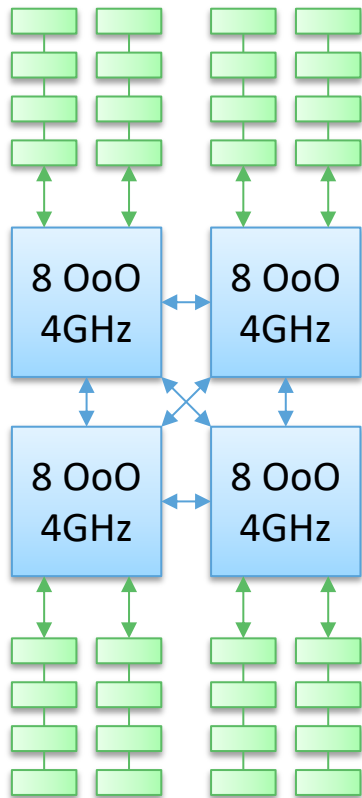


# Tesseract System for Graph Processing



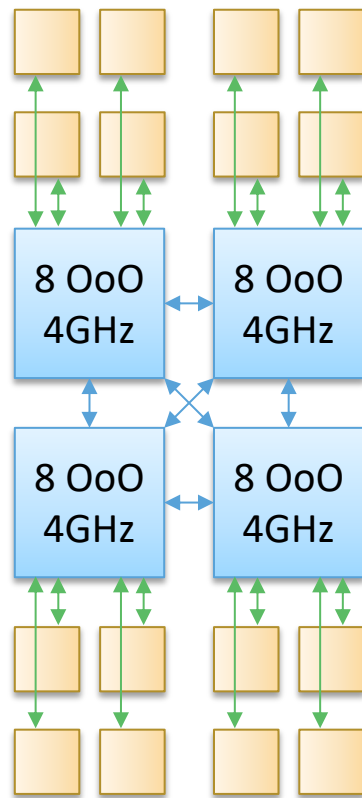
# Evaluated Systems

DDR3-OoO



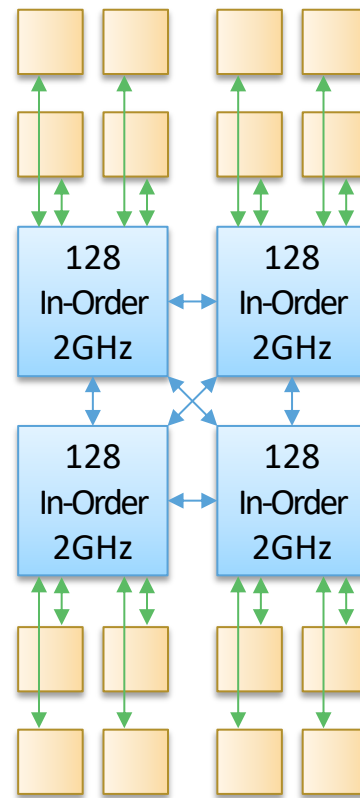
102.4GB/s

HMC-OoO



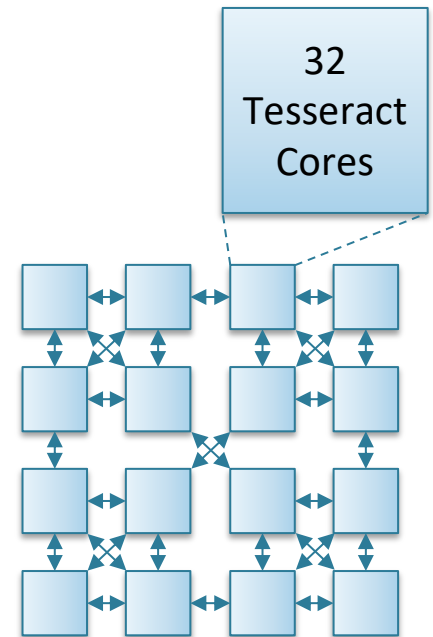
640GB/s

HMC-MC



640GB/s

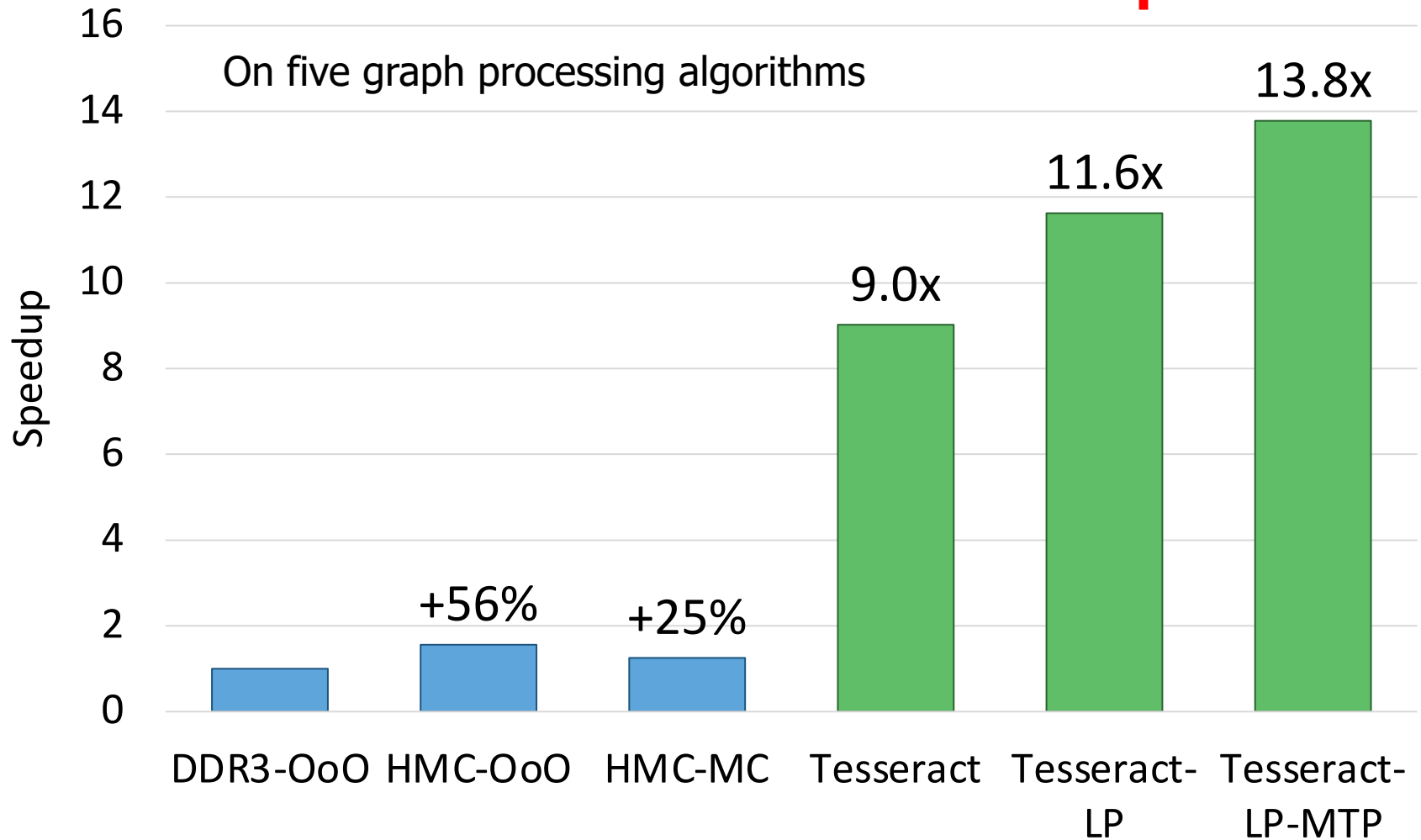
**Tesseract**



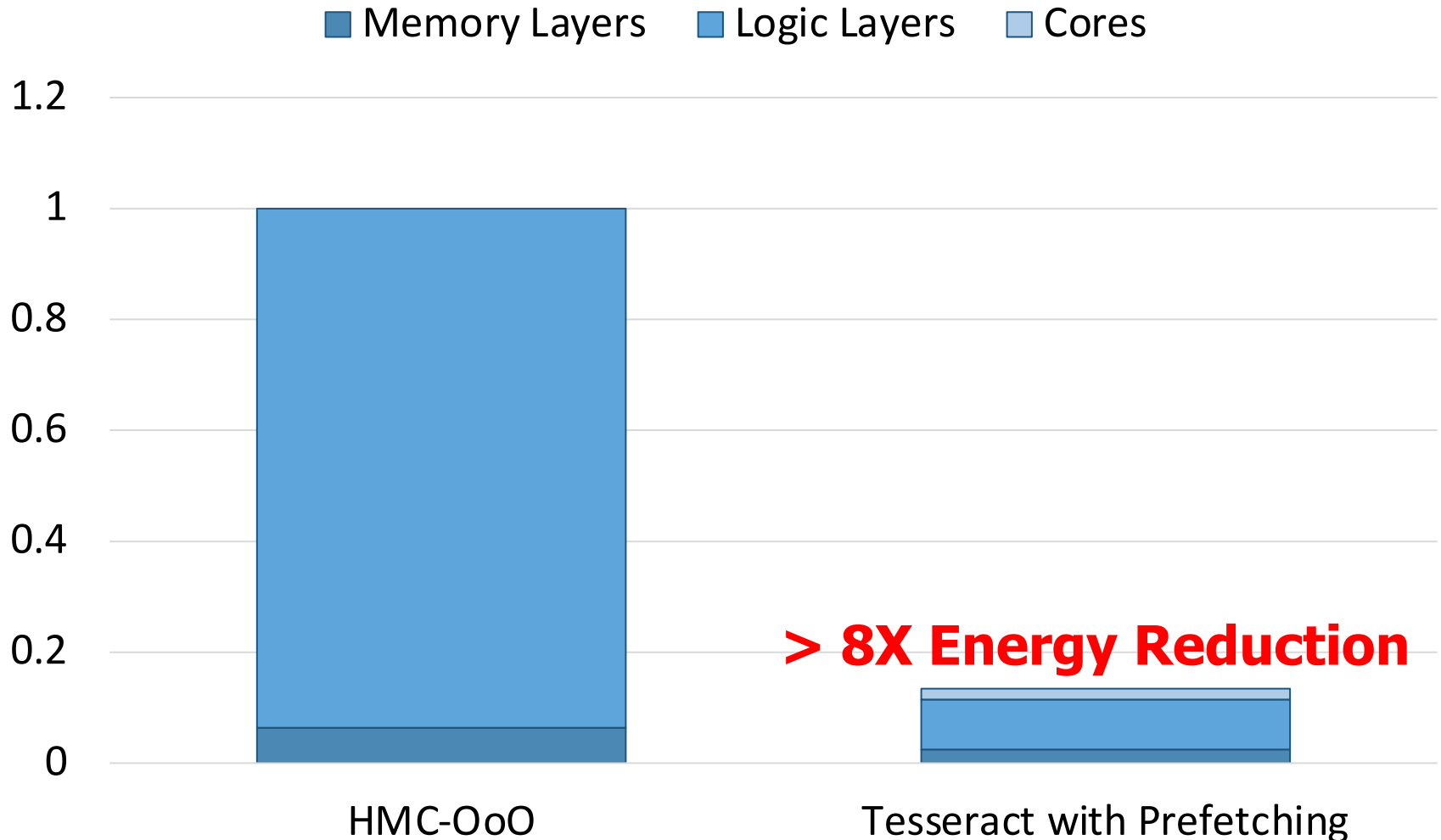
**8TB/s**

# 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.***

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

Junwhan Ahn   Sungpack Hong<sup>§</sup>   Sungjoo Yoo   Onur Mutlu<sup>†</sup>   Kiyoungh Choi

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

Seoul National University

<sup>§</sup>Oracle Labs

<sup>†</sup>Carnegie Mellon University



# 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)]

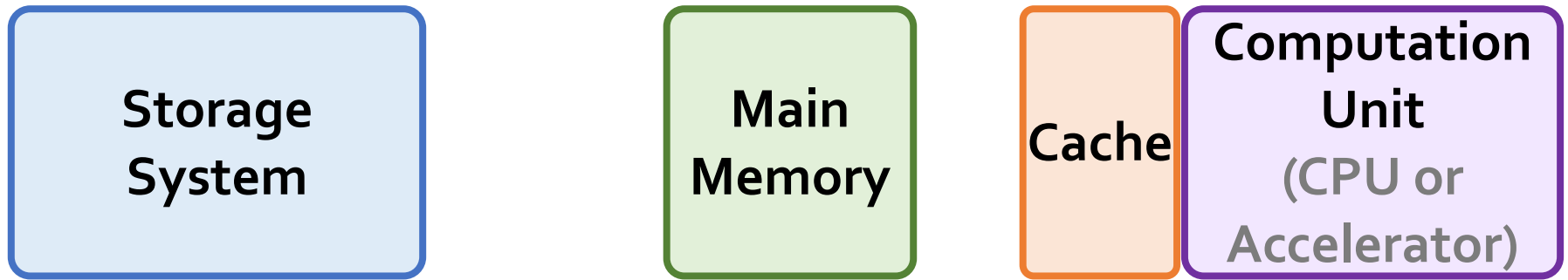
## GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi<sup>1</sup> Jisung Park<sup>1</sup> Harun Mustafa<sup>1</sup> Jeremie Kim<sup>1</sup> Ataberk Olgun<sup>1</sup>  
Arvid Gollwitzer<sup>1</sup> Damla Senol Cali<sup>2</sup> Can Firtina<sup>1</sup> Haiyu Mao<sup>1</sup> Nour Almadhoun Alserr<sup>1</sup>  
Rachata Ausavarungnirun<sup>3</sup> Nandita Vijaykumar<sup>4</sup> Mohammed Alser<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich <sup>2</sup>Bionano Genomics <sup>3</sup>KMUTNB <sup>4</sup>University of Toronto

# Genome Sequence Analysis

**Data Movement from Storage**

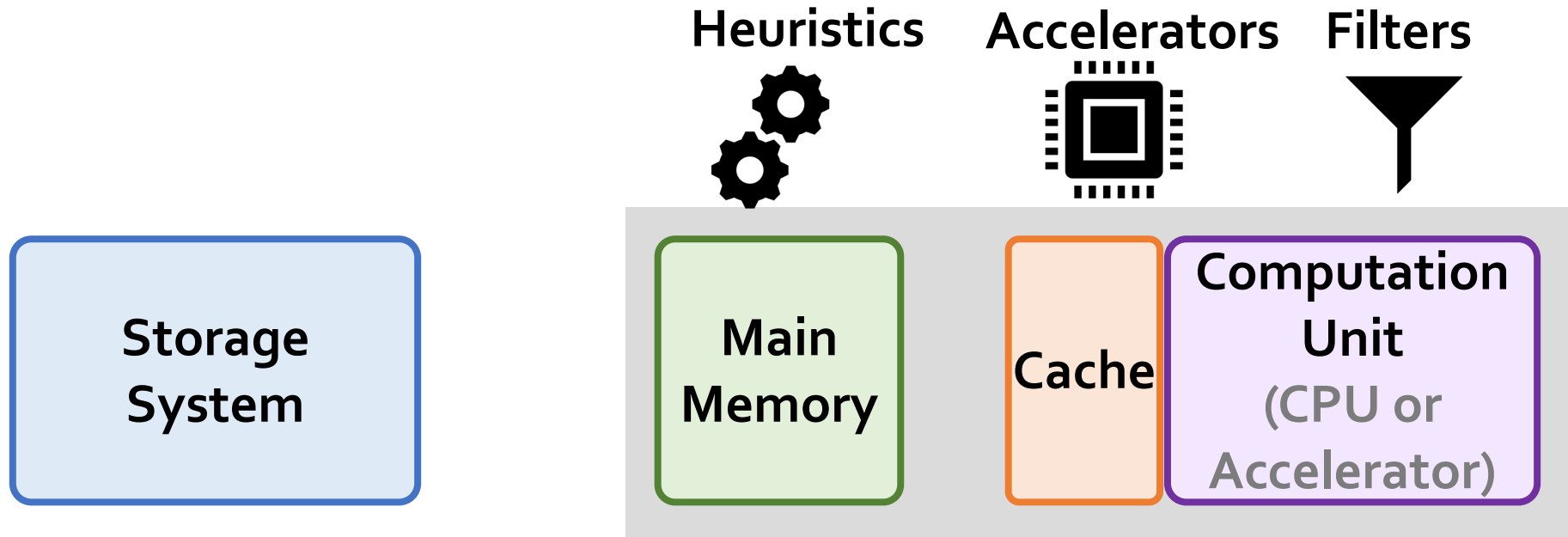


**Computation overhead**



**Data movement overhead**

# Compute-Centric Accelerators



Computation overhead

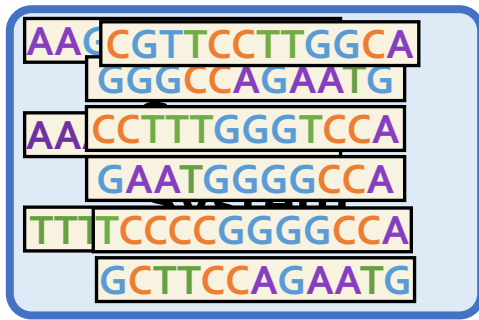


Data movement overhead

# 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

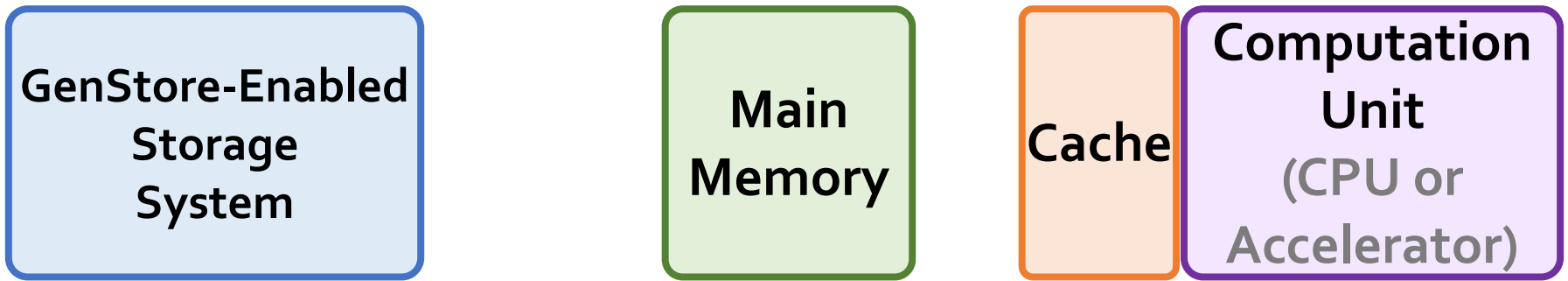
## **Non-matching reads**

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 Ghiasi<sup>1</sup> Jisung Park<sup>1</sup> Harun Mustafa<sup>1</sup> Jeremie Kim<sup>1</sup> Ataberk Olgun<sup>1</sup>  
Arvid Gollwitzer<sup>1</sup> Damla Senol Cali<sup>2</sup> Can Firtina<sup>1</sup> Haiyu Mao<sup>1</sup> Nour Almadhoun Alserr<sup>1</sup>  
Rachata Ausavarungnirun<sup>3</sup> Nandita Vijaykumar<sup>4</sup> Mohammed Alser<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich <sup>2</sup>Bionano Genomics <sup>3</sup>KMUTNB <sup>4</sup>University of Toronto

# 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**

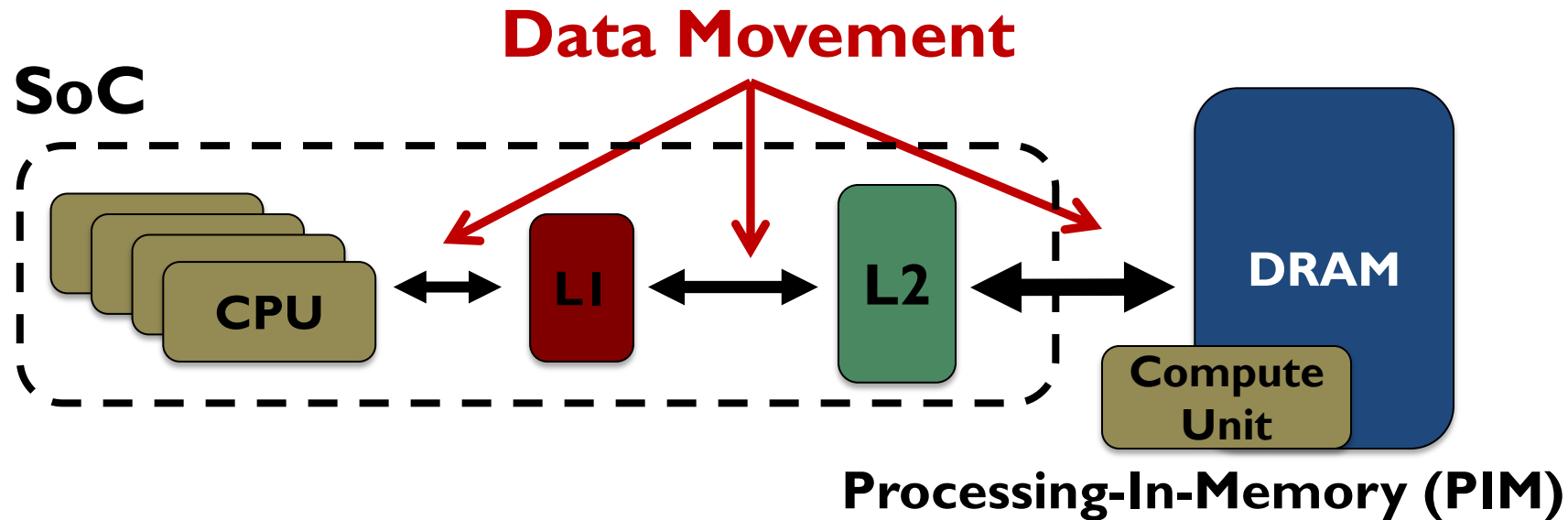


**Video Capture**

Google's **video codec**

# Energy Cost of Data Movement

**1<sup>st</sup> 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

**2<sup>nd</sup> 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**

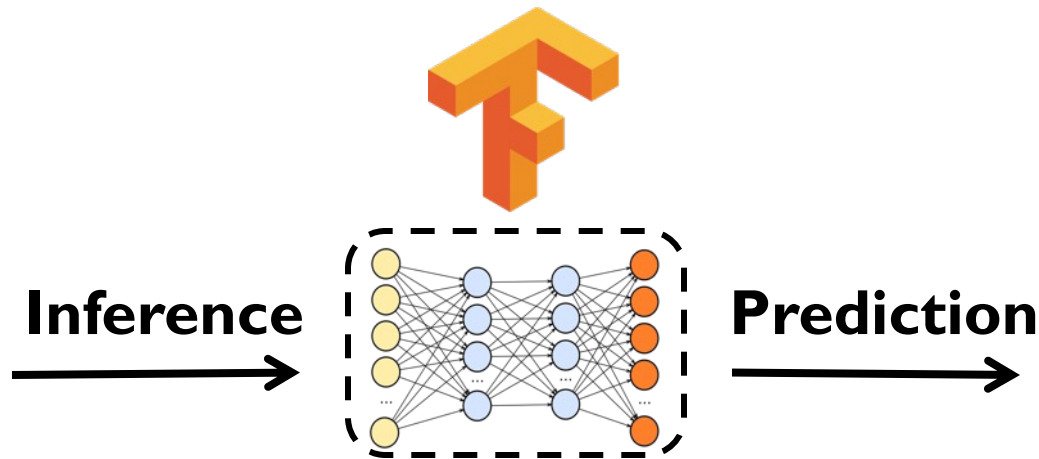
Google's **video codec**



**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

# 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 Boroumand<sup>1</sup>

Saugata Ghose<sup>1</sup>

Youngsok Kim<sup>2</sup>

Rachata Ausavarungnirun<sup>1</sup>

Eric Shiu<sup>3</sup>

Rahul Thakur<sup>3</sup>

Daehyun Kim<sup>4,3</sup>

Aki Kuusela<sup>3</sup>

Allan Knies<sup>3</sup>

Parthasarathy Ranganathan<sup>3</sup>

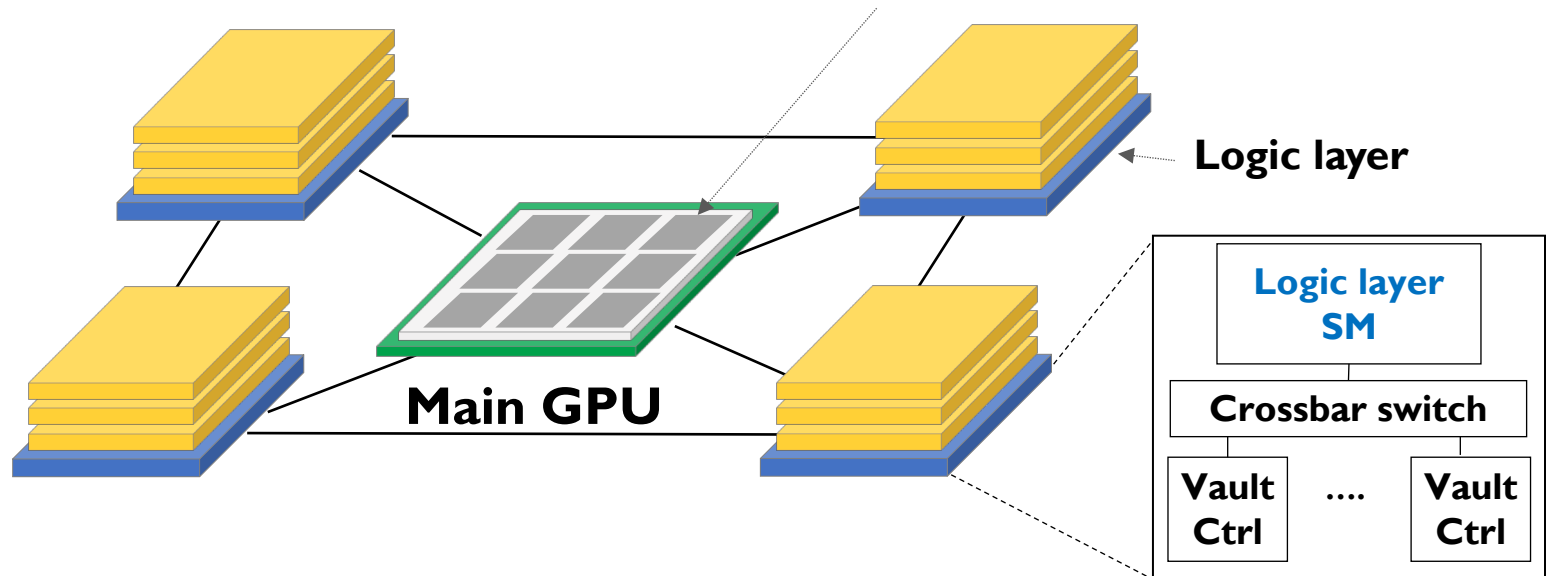
Onur Mutlu<sup>5,1</sup>



# 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<sup>‡</sup> Eiman Ebrahimi<sup>†</sup> Gwangsun Kim\* Niladrish Chatterjee<sup>†</sup> Mike O'Connor<sup>†</sup>  
Nandita Vijaykumar<sup>‡</sup> Onur Mutlu<sup>§‡</sup> Stephen W. Keckler<sup>†</sup>

<sup>‡</sup>Carnegie Mellon University <sup>†</sup>NVIDIA <sup>\*</sup>KAIST <sup>§</sup>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 Pattnaik<sup>1</sup>    Xulong Tang<sup>1</sup>    Adwait Jog<sup>2</sup>    Onur Kayiran<sup>3</sup>  
Asit K. Mishra<sup>4</sup>    Mahmut T. Kandemir<sup>1</sup>    Onur Mutlu<sup>5,6</sup>    Chita R. Das<sup>1</sup>

<sup>1</sup>Pennsylvania State University    <sup>2</sup>College of William and Mary  
<sup>3</sup>Advanced Micro Devices, Inc.    <sup>4</sup>Intel Labs    <sup>5</sup>ETH Zürich    <sup>6</sup>Carnegie 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<sup>†</sup> Samira Khan<sup>‡</sup> Nandita Vijaykumar<sup>†</sup>  
Kevin K. Chang<sup>†</sup> Amirali Boroumand<sup>†</sup> Saugata Ghose<sup>†</sup> Onur Mutlu<sup>§†</sup>  
<sup>†</sup>Carnegie Mellon University    <sup>‡</sup>University of Virginia    <sup>§</sup>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<sup>\*</sup>, Khubaib<sup>†</sup>, Eiman Ebrahimi<sup>‡</sup>, Onur Mutlu<sup>§</sup>, Yale N. Patt<sup>\*</sup>

*<sup>\*</sup>The University of Texas at Austin   <sup>†</sup>Apple   <sup>‡</sup>NVIDIA   <sup>§</sup>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](#)]

## Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads

Milad Hashemi\*, Onur Mutlu<sup>§</sup>, Yale N. Patt\*

\*The University of Texas at Austin    <sup>§</sup>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<sup>a,b,c</sup>    Dionysios Diamantopoulos<sup>c</sup>    Christoph Hagleitner<sup>c</sup>    Juan Gómez-Luna<sup>b</sup>  
Sander Stuijk<sup>a</sup>    Onur Mutlu<sup>b</sup>    Henk Corporaal<sup>a</sup>  
<sup>a</sup>Eindhoven University of Technology    <sup>b</sup>ETH Zürich    <sup>c</sup>IBM Research Europe, Zurich



# Accelerating Approximate String Matching

- 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, 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.*  
[[Lightning 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<sup>†⋈</sup> Gurpreet S. Kalsi<sup>⋈</sup> Zülal Bingöl<sup>▽</sup> Can Firtina<sup>◇</sup> Lavanya Subramanian<sup>‡</sup> Jeremie S. Kim<sup>◇†</sup>  
Rachata Ausavarungnirun<sup>⊙</sup> Mohammed Alser<sup>◇</sup> Juan Gomez-Luna<sup>◇</sup> Amirali Boroumand<sup>†</sup> Anant Nori<sup>⋈</sup>  
Allison Scibisz<sup>†</sup> Sreenivas Subramoney<sup>⋈</sup> Can Alkan<sup>▽</sup> Saugata Ghose<sup>\*†</sup> Onur Mutlu<sup>◇†▽</sup>  
<sup>†</sup>Carnegie Mellon University   <sup>⋈</sup>Processor Architecture Research Lab, Intel Labs   <sup>▽</sup>Bilkent University   <sup>◇</sup>ETH Zürich  
<sup>‡</sup>Facebook   <sup>⊙</sup>King Mongkut's University of Technology North Bangkok   <sup>\*</sup>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 Cali<sup>1</sup> Konstantinos Kanellopoulos<sup>2</sup> Joël Lindegger<sup>2</sup> Zülal Bingöl<sup>3</sup>  
Gurpreet S. Kalsi<sup>4</sup> Ziyi Zuo<sup>5</sup> Can Firtina<sup>2</sup> Meryem Banu Cavlak<sup>2</sup> Jeremie Kim<sup>2</sup>  
Nika Mansouri Ghiasi<sup>2</sup> Gagandeep Singh<sup>2</sup> Juan Gómez-Luna<sup>2</sup> Nour Almadhoun Alserr<sup>2</sup>  
Mohammed Alser<sup>2</sup> Sreenivas Subramoney<sup>4</sup> Can Alkan<sup>3</sup> Saugata Ghose<sup>6</sup> Onur Mutlu<sup>2</sup>

<sup>1</sup>Bionano Genomics <sup>2</sup>ETH Zürich <sup>3</sup>Bilkent University <sup>4</sup>Intel Labs

<sup>5</sup>Carnegie Mellon University <sup>6</sup>University of Illinois Urbana-Champaign

# Accelerating Basecalling + Read Mapping

---

- Appears at MICRO 2022

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

Haiyu Mao<sup>1</sup>   Mohammed Alser<sup>1</sup>   Mohammad Sadrosadati<sup>1</sup>   Can Firtina<sup>1</sup>   Akanksha Baranwal<sup>1</sup>  
Damla Senol Cali<sup>2</sup>   Aditya Manglik<sup>1</sup>   Nour Almadhoun Alserr<sup>1</sup>   Onur Mutlu<sup>1</sup>  
<sup>1</sup>*ETH Zürich*   <sup>2</sup>*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<sup>§</sup>

Ricardo Quisiant<sup>§</sup>

Christina Giannoula<sup>†</sup>

Mohammed Alser<sup>‡</sup>

Juan Gómez-Luna<sup>‡</sup>

Eladio Gutiérrez<sup>§</sup>

Oscar Plata<sup>§</sup>

Onur Mutlu<sup>‡</sup>

<sup>§</sup>*University of Malaga*

<sup>†</sup>*National Technical University of Athens*

<sup>‡</sup>*ETH Zürich*

# 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 Besta<sup>1</sup>, Raghavendra Kanakagiri<sup>2</sup>, Grzegorz Kwasniewski<sup>1</sup>, Rachata Ausavarungnirun<sup>3</sup>, Jakub Beránek<sup>4</sup>, Konstantinos Kanellopoulos<sup>1</sup>, Kacper Janda<sup>5</sup>, Zur Vonarburg-Shmaria<sup>1</sup>, Lukas Gianinazzi<sup>1</sup>, Ioana Stefan<sup>1</sup>, Juan Gómez-Luna<sup>1</sup>, Marcin Copik<sup>1</sup>, Lukas Kapp-Schwoerer<sup>1</sup>, Salvatore Di Girolamo<sup>1</sup>, Nils Blach<sup>1</sup>, Marek Konieczny<sup>5</sup>, Onur Mutlu<sup>1</sup>, Torsten Hoefler<sup>1</sup>

<sup>1</sup>ETH Zurich, Switzerland  
Thailand

<sup>2</sup>IIT Tirupati, India

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

<sup>5</sup>AGH-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<sup>†</sup>      Saugata Ghose<sup>◇</sup>      Geraldo F. Oliveira<sup>‡</sup>      Onur Mutlu<sup>‡</sup>  
<sup>†</sup>*Google*      <sup>◇</sup>*Univ. of Illinois Urbana-Champaign*      <sup>‡</sup>*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<sup>†◇</sup>

Geraldo F. Oliveira<sup>\*</sup>

Saugata Ghose<sup>‡</sup>

Xiaoyu Ma<sup>§</sup>

Berkin Akin<sup>§</sup>

Eric Shiu<sup>§</sup>

Ravi Narayanaswami<sup>§</sup>

Onur Mutlu<sup>\*†</sup>

<sup>†</sup>*Carnegie Mellon Univ.*

<sup>◇</sup>*Stanford Univ.*

<sup>‡</sup>*Univ. of Illinois Urbana-Champaign*

<sup>§</sup>*Google*

<sup>\*</sup>*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**

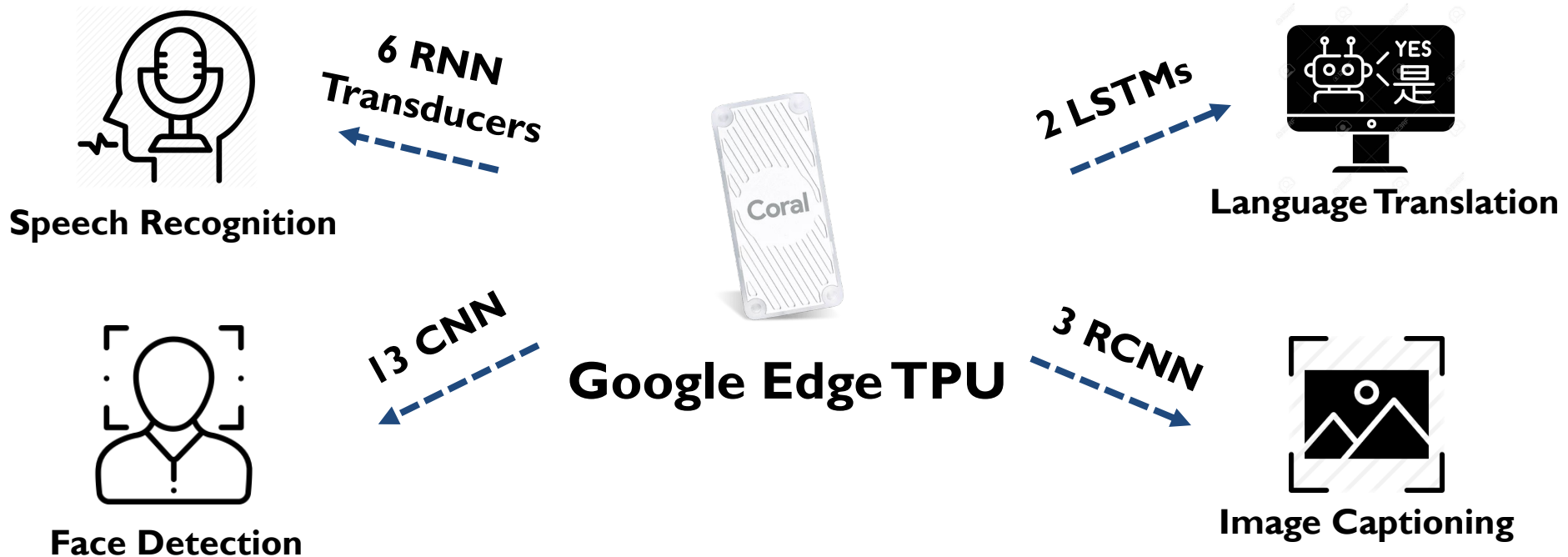
- Mensa consists of heterogeneous accelerators whose dataflow and hardware are specialized for specific families of layers

**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

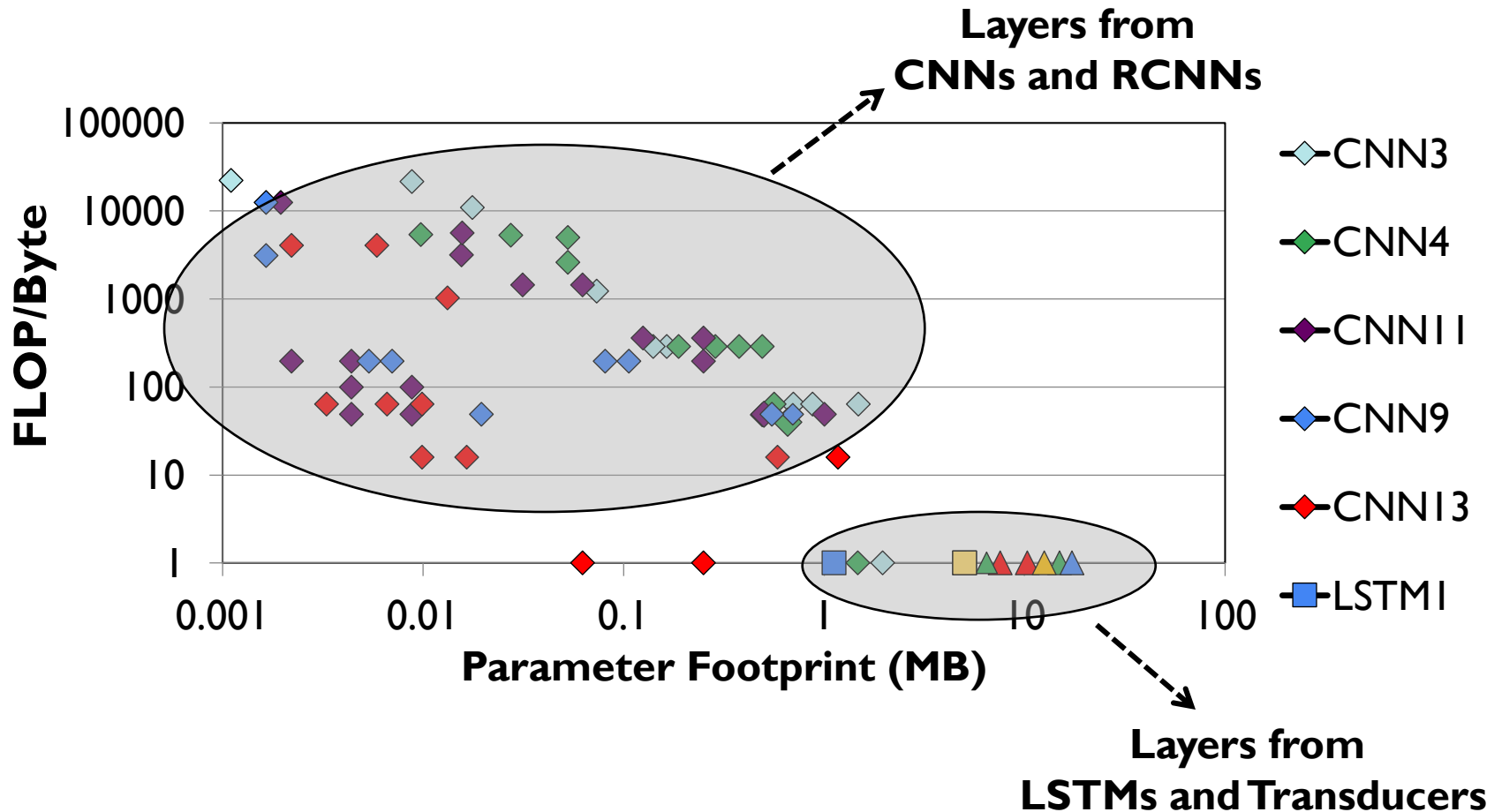
# Google Edge Neural Network Models

We analyze inference execution using 24 edge NN models



# Diversity Across the Models

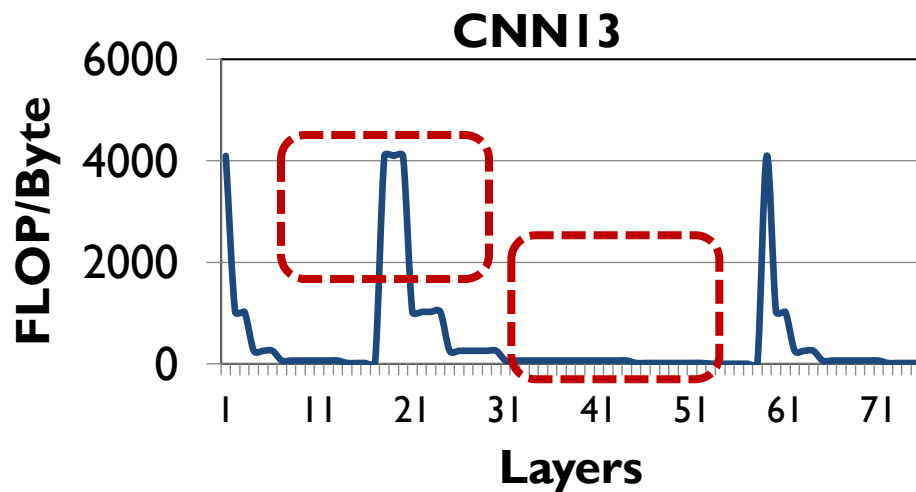
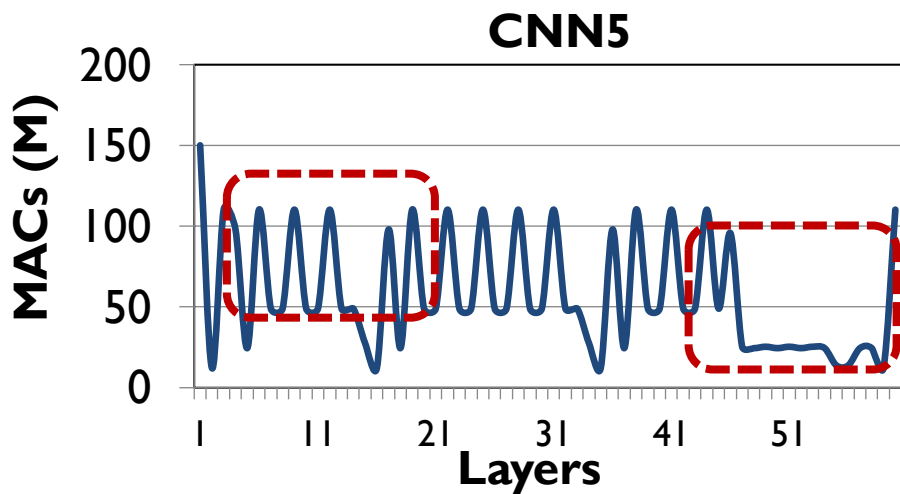
**Insight 1:** there is **significant variation** in terms of layer characteristics **across the models**



# 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:



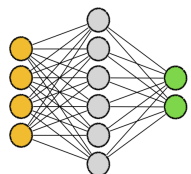
Variation in **MAC intensity**: up to **200x** across layers

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

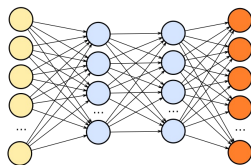
# Mensa High-Level Overview

## Edge TPU Accelerator

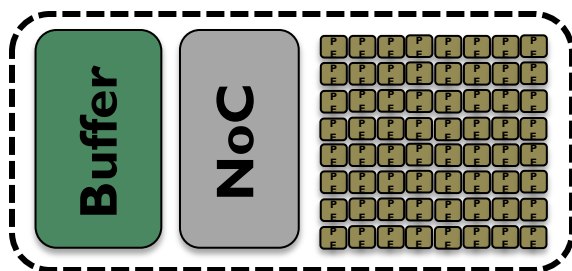
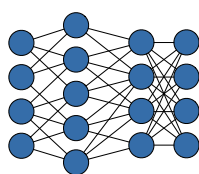
Model A



Model B



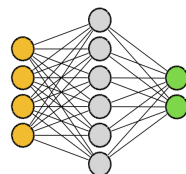
Model C



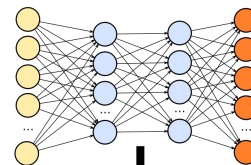
Monolithic Accelerator

## Mensa

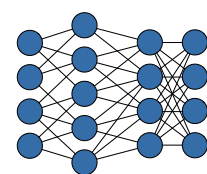
Model A



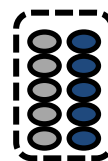
Model B



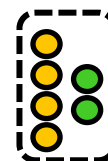
Model C



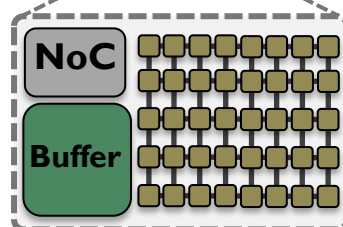
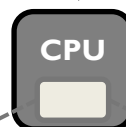
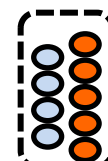
Family 1



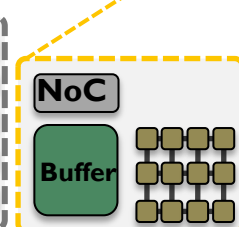
Family 2



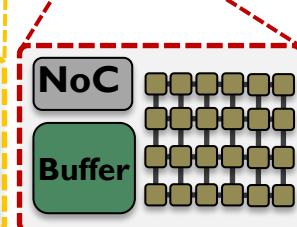
Family 3



Acc. 1



Acc. 2

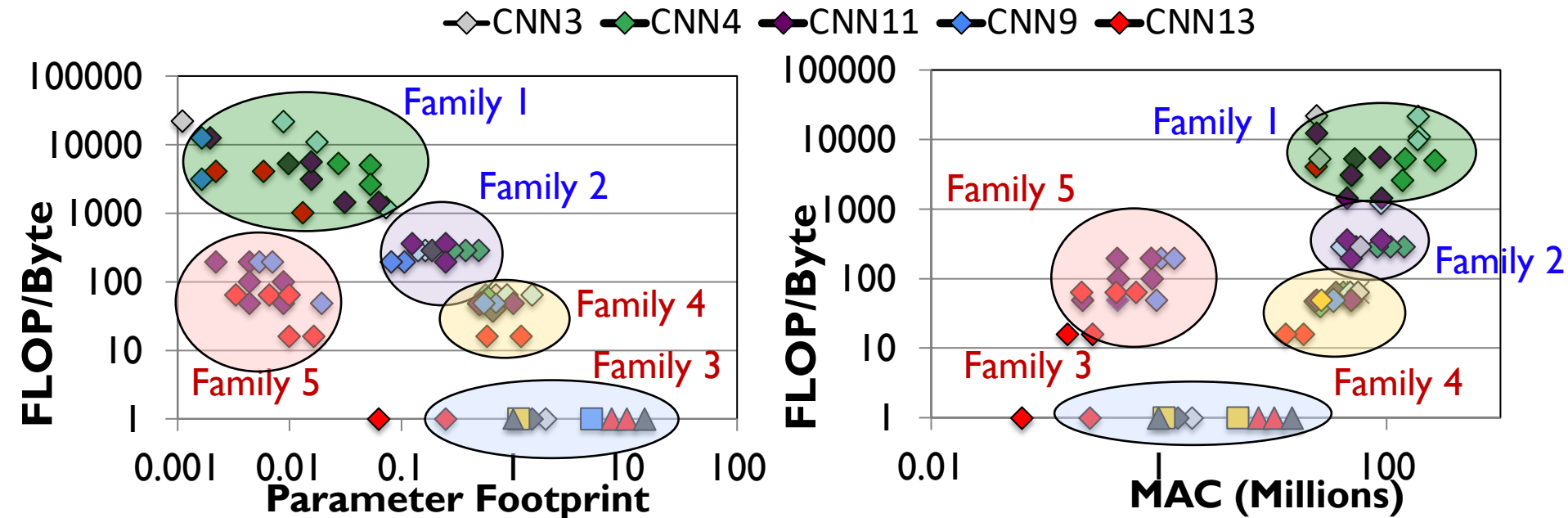


Acc. 3

Heterogeneous Accelerators

# Identifying Layer Families

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

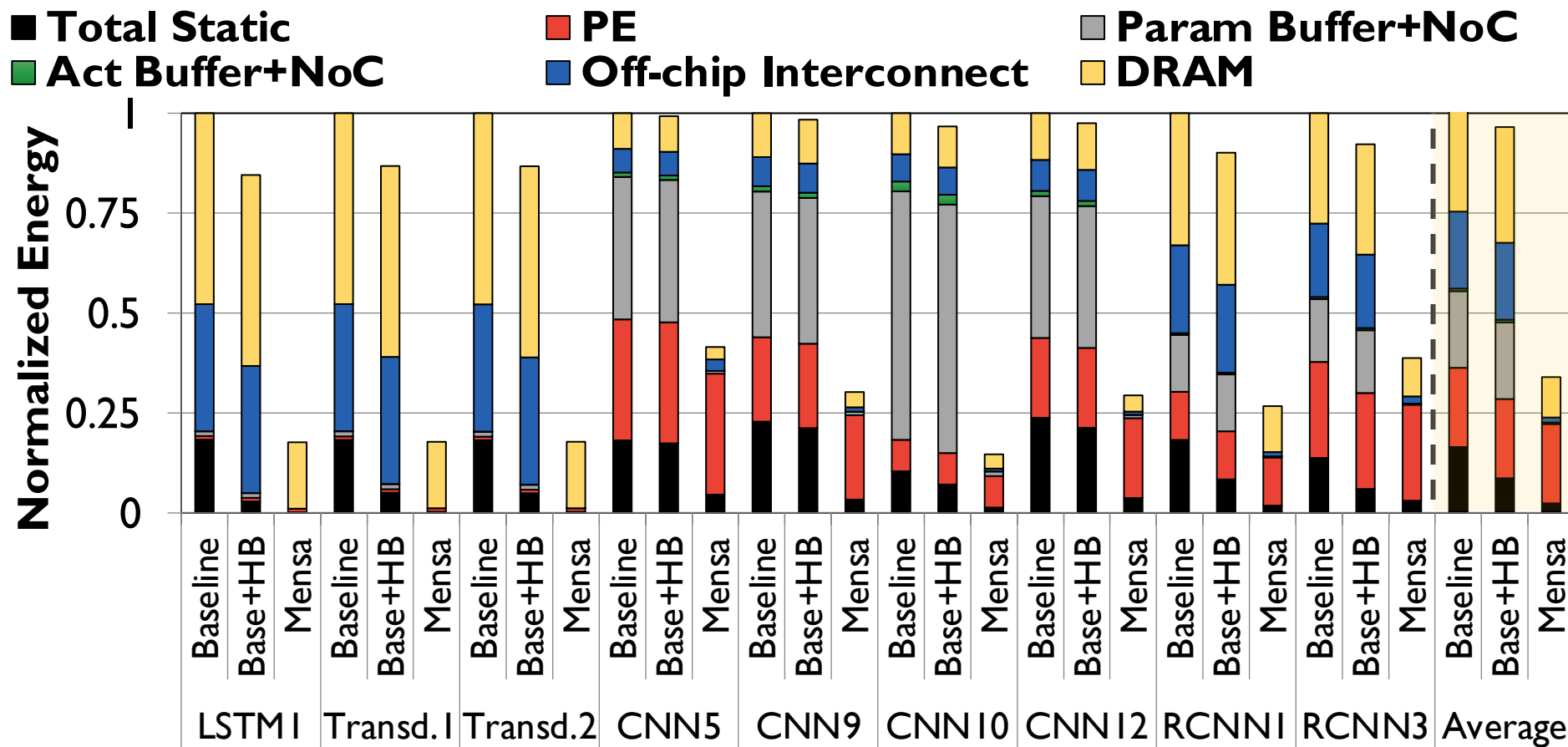


**Families 1 & 2: low parameter footprint, high data reuse and **MAC** intensity**  
→ compute-centric layers

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

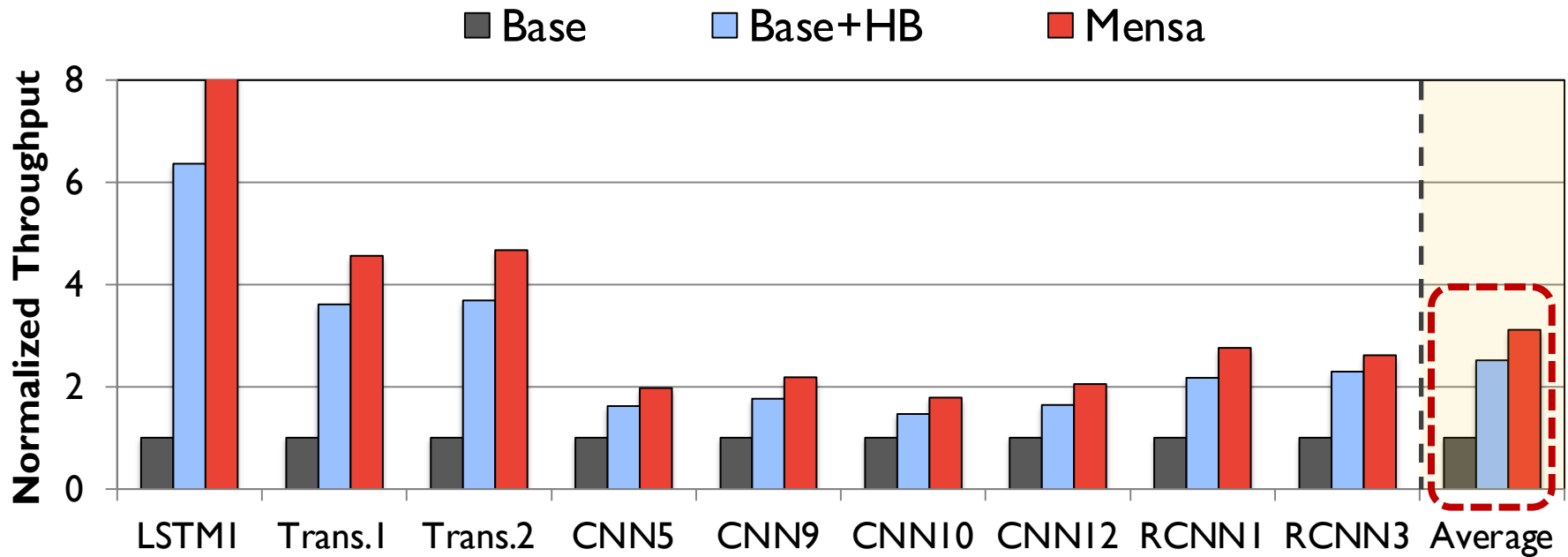


# Mensa: Energy Reduction



**Mensa-G reduces energy consumption by 3.0X**  
compared to the baseline EdgeTPU

# 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<sup>†◇</sup>

Geraldo F. Oliveira<sup>\*</sup>

Saugata Ghose<sup>‡</sup>

Xiaoyu Ma<sup>§</sup>

Berkin Akin<sup>§</sup>

Eric Shiu<sup>§</sup>

Ravi Narayanaswami<sup>§</sup>

Onur Mutlu<sup>\*†</sup>

<sup>†</sup>*Carnegie Mellon Univ.*

<sup>◇</sup>*Stanford Univ.*

<sup>‡</sup>*Univ. of Illinois Urbana-Champaign*

<sup>§</sup>*Google*

<sup>\*</sup>*ETH Zürich*

# 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<sup>◇</sup> Mohammed Alser<sup>◇</sup> Damla Senol Cali<sup>✕</sup>

Dionysios Diamantopoulos<sup>▽</sup> Juan Gómez-Luna<sup>◇</sup>

Henk Corporaal<sup>★</sup> Onur Mutlu<sup>◇✕</sup>

<sup>◇</sup>*ETH Zürich*    <sup>✕</sup>*Carnegie Mellon University*

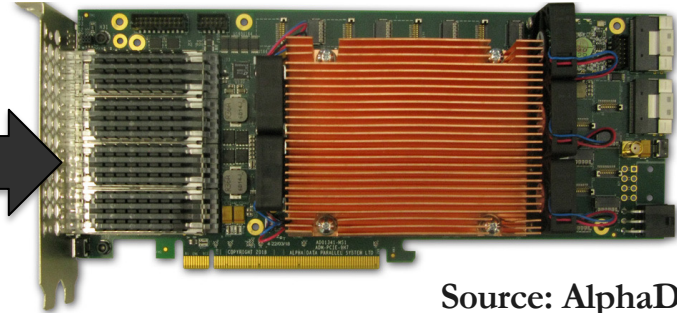
<sup>★</sup>*Eindhoven University of Technology*    <sup>▽</sup>*IBM Research Europe*

# Near-Memory Acceleration Using FPGAs



Source: IBM

IBM POWER9 CPU



Source: AlphaData

HBM-based FPGA board

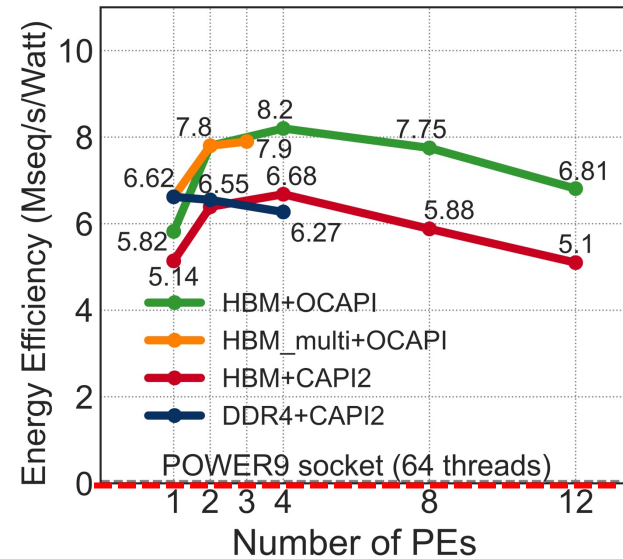
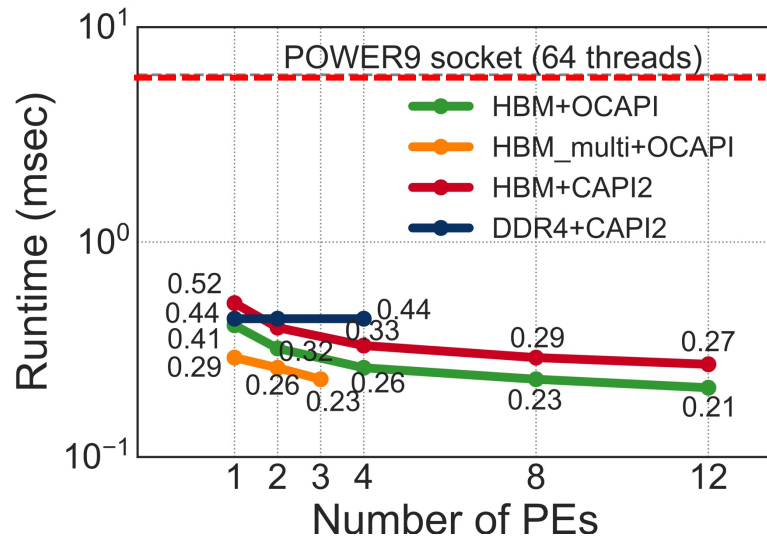
## Near-HBM FPGA-based accelerator

**Two communication technologies:** CAPI2 and OCAPI

**Two memory technologies:** DDR4 and HBM

**Two workloads:** Weather Modeling and Genome Analysis

# Performance & Energy Greatly Improve



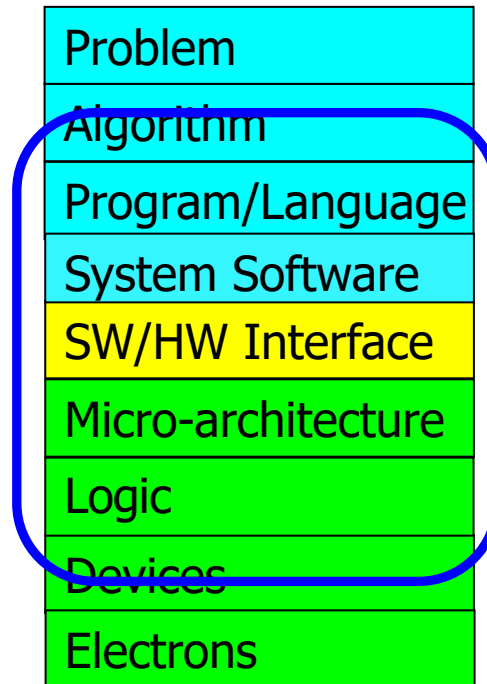
**5-27× performance** vs. a 16-core (64-thread) IBM POWER9 CPU

**12-133× energy efficiency** vs. a 16-core (64-thread) IBM POWER9 CPU

**HBM alleviates memory bandwidth contention vs. DDR4**

# 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<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirun<sup>d</sup>

*SAFARI Research Group*

<sup>a</sup>*ETH Zürich*

<sup>b</sup>*Carnegie Mellon University*

<sup>c</sup>*University of Illinois at Urbana-Champaign*

<sup>d</sup>*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<sup>†</sup>   Amirali Boroumand<sup>†</sup>   Jeremie S. Kim<sup>†§</sup>   Juan Gómez-Luna<sup>§</sup>   Onur Mutlu<sup>§†</sup>

<sup>†</sup>*Carnegie Mellon University*

<sup>§</sup>*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

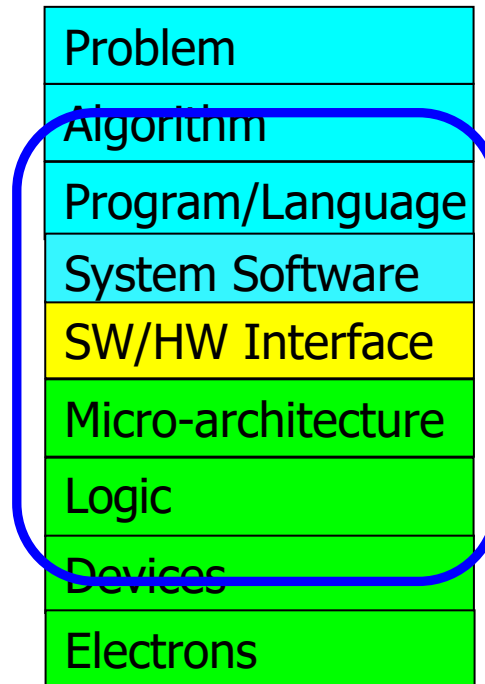
---

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: Accelerating Key Applications (I)

---

- Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,  
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#)  
[IEEE Micro \(IEEE MICRO\)](#), Vol. 40, No. 5, pages 65-75, September/October 2020.  
[[Slides \(pptx\)\(pdf\)](#)]  
[[Talk Video \(1 hour 2 minutes\)](#)]

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

**Mohammed Alser**

ETH Zürich

**Zülal Bingöl**

Bilkent University

**Damla Senol Cali**

Carnegie Mellon University

**Jeremie Kim**

ETH Zurich and Carnegie Mellon University

**Saugata Ghose**

University of Illinois at Urbana–Champaign and  
Carnegie Mellon University

**Can Alkan**

Bilkent University

**Onur Mutlu**

ETH Zurich, Carnegie Mellon University, and  
Bilkent University



# Adoption: Accelerating Key Applications (II)

---

- 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 Ghiasi<sup>1</sup> Jisung Park<sup>1</sup> Harun Mustafa<sup>1</sup> Jeremie Kim<sup>1</sup> Ataberk Olgun<sup>1</sup>  
Arvid Gollwitzer<sup>1</sup> Damla Senol Cali<sup>2</sup> Can Firtina<sup>1</sup> Haiyu Mao<sup>1</sup> Nour Almadhoun Alserr<sup>1</sup>  
Rachata Ausavarungnirun<sup>3</sup> Nandita Vijaykumar<sup>4</sup> Mohammed Alser<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich <sup>2</sup>Bionano Genomics <sup>3</sup>KMUTNB <sup>4</sup>University of Toronto

# Adoption: Accelerating Key Applications (III)

---

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoun 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 \(pdf\)\]](#) [\[Lightning Session Slides \(pdf\)\]](#)

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

Junwhan Ahn   Sungpack Hong<sup>§</sup>   Sungjoo Yoo   Onur Mutlu<sup>†</sup>   Kiyoun Choi  
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University   <sup>§</sup>Oracle Labs   <sup>†</sup>Carnegie Mellon University

# Adoption: Accelerating Key Applications (IV)

- 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 Besta<sup>1</sup>, Raghavendra Kanakagiri<sup>2</sup>, Grzegorz Kwasniewski<sup>1</sup>, Rachata Ausavarungnirun<sup>3</sup>, Jakub Beránek<sup>4</sup>, Konstantinos Kanellopoulos<sup>1</sup>, Kacper Janda<sup>5</sup>, Zur Vonarburg-Shmaria<sup>1</sup>, Lukas Gianinazzi<sup>1</sup>, Ioana Stefan<sup>1</sup>, Juan Gómez-Luna<sup>1</sup>, Marcin Copik<sup>1</sup>, Lukas Kapp-Schwoerer<sup>1</sup>, Salvatore Di Girolamo<sup>1</sup>, Nils Blach<sup>1</sup>, Marek Konieczny<sup>5</sup>, Onur Mutlu<sup>1</sup>, Torsten Hoefler<sup>1</sup>

<sup>1</sup>ETH Zurich, Switzerland  
Thailand

<sup>2</sup>IIT Tirupati, India

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

<sup>5</sup>AGH-UST, Poland

# Adoption: Accelerating Key Applications (V)

---

- 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<sup>†◇</sup>

Geraldo F. Oliveira<sup>\*</sup>

Saugata Ghose<sup>‡</sup>

Xiaoyu Ma<sup>§</sup>

Berkin Akin<sup>§</sup>

Eric Shiu<sup>§</sup>

Ravi Narayanaswami<sup>§</sup>

Onur Mutlu<sup>\*†</sup>

<sup>†</sup>*Carnegie Mellon Univ.*

<sup>◇</sup>*Stanford Univ.*

<sup>‡</sup>*Univ. of Illinois Urbana-Champaign*

<sup>§</sup>*Google*

<sup>\*</sup>*ETH Zürich*

# Adoption: Accelerating Key Applications (VI)

---

- 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<sup>a,b,c</sup>    Dionysios Diamantopoulos<sup>c</sup>    Christoph Hagleitner<sup>c</sup>    Juan Gómez-Luna<sup>b</sup>  
Sander Stuijk<sup>a</sup>    Onur Mutlu<sup>b</sup>    Henk Corporaal<sup>a</sup>  
<sup>a</sup>Eindhoven University of Technology    <sup>b</sup>ETH Zürich    <sup>c</sup>IBM Research Europe, Zurich

# Adoption: Accelerating Key Applications (VII)

---

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

# Adoption: Accelerating Key Applications (VIII)

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



# Adoption: Accelerating Key Applications (IX)

---

- 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 Cali<sup>1</sup> Konstantinos Kanellopoulos<sup>2</sup> Joël Lindegger<sup>2</sup> Zülal Bingöl<sup>3</sup>  
Gurpreet S. Kalsi<sup>4</sup> Ziyi Zuo<sup>5</sup> Can Firtina<sup>2</sup> Meryem Banu Cavlak<sup>2</sup> Jeremie Kim<sup>2</sup>  
Nika Mansouri Ghiasi<sup>2</sup> Gagandeep Singh<sup>2</sup> Juan Gómez-Luna<sup>2</sup> Nour Almadhoun Alserr<sup>2</sup>  
Mohammed Alser<sup>2</sup> Sreenivas Subramoney<sup>4</sup> Can Alkan<sup>3</sup> Saugata Ghose<sup>6</sup> Onur Mutlu<sup>2</sup>

<sup>1</sup>Bionano Genomics   <sup>2</sup>ETH Zürich   <sup>3</sup>Bilkent University   <sup>4</sup>Intel Labs  
<sup>5</sup>Carnegie Mellon University   <sup>6</sup>University of Illinois Urbana-Champaign

# Adoption: Accelerating Key Applications (X)

---

- 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<sup>†</sup>      Saugata Ghose<sup>◇</sup>      Geraldo F. Oliveira<sup>‡</sup>      Onur Mutlu<sup>‡</sup>  
<sup>†</sup>*Google*      <sup>◇</sup>*Univ. of Illinois Urbana-Champaign*      <sup>‡</sup>*ETH Zürich*

# Adoption: Accelerating Key Applications (XI)

---

- Appears at MICRO 2022

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

Haiyu Mao<sup>1</sup>   Mohammed Alser<sup>1</sup>   Mohammad Sadrosadati<sup>1</sup>   Can Firtina<sup>1</sup>   Akanksha Baranwal<sup>1</sup>  
Damla Senol Cali<sup>2</sup>   Aditya Manglik<sup>1</sup>   Nour Almadhoun Alserr<sup>1</sup>   Onur Mutlu<sup>1</sup>

<sup>1</sup>*ETH Zürich*

<sup>2</sup>*Bionano Genomics*

# 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<sup>†</sup>   Kiyoun Choi

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

Seoul National University

<sup>†</sup>Carnegie Mellon University

# Adoption: How to Maintain Coherence? (I)

---

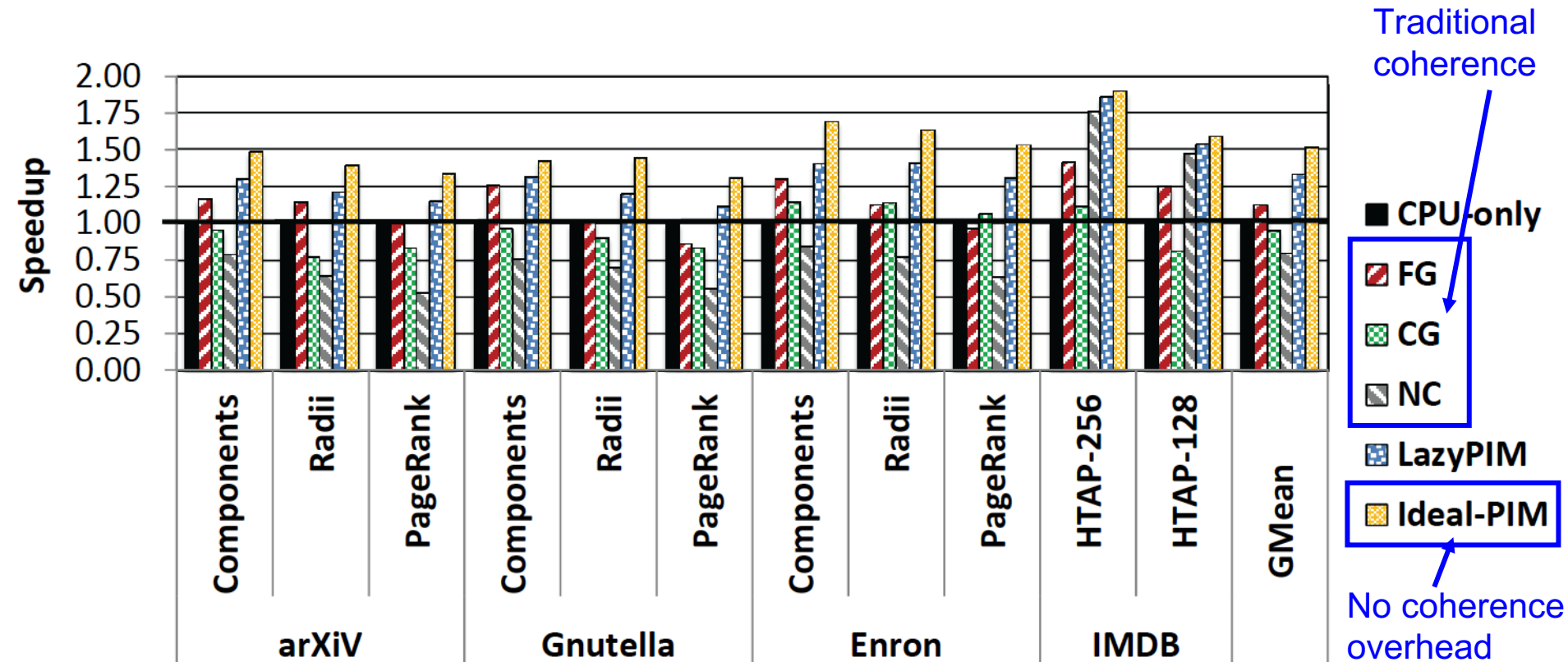
- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,  
**"LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory"**  
*IEEE Computer Architecture Letters* (**CAL**), June 2016.

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

Amirali Boroumand<sup>†</sup>, Saugata Ghose<sup>†</sup>, Minesh Patel<sup>†</sup>, Hasan Hassan<sup>†§</sup>, Brandon Lucia<sup>†</sup>,  
Kevin Hsieh<sup>†</sup>, Krishna T. Malladi<sup>\*</sup>, Hongzhong Zheng<sup>\*</sup>, and Onur Mutlu<sup>††</sup>

<sup>†</sup>Carnegie Mellon University   <sup>\*</sup>Samsung Semiconductor, Inc.   <sup>§</sup>TOBB ETÜ   <sup>‡</sup>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<sup>†</sup>

Saugata Ghose<sup>†</sup>

Minesh Patel<sup>★</sup>

Hasan Hassan<sup>★</sup>

Brandon Lucia<sup>†</sup>

Rachata Ausavarungnirun<sup>†‡</sup>

Kevin Hsieh<sup>†</sup>

Nastaran Hajinazar<sup>◇†</sup>

Krishna T. Malladi<sup>§</sup>

Hongzhong Zheng<sup>§</sup>

Onur Mutlu<sup>★†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>★</sup>ETH Zürich

<sup>‡</sup>KMUTNB

<sup>◇</sup>Simon Fraser University

<sup>§</sup>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<sup>†‡</sup> Nandita Vijaykumar<sup>\*‡</sup> Nikela Papadopoulou<sup>†</sup> Vasileios Karakostas<sup>†</sup> Ivan Fernandez<sup>§‡</sup>  
Juan Gómez-Luna<sup>‡</sup> Lois Orosa<sup>‡</sup> Nectarios Koziris<sup>†</sup> Georgios Goumas<sup>†</sup> Onur Mutlu<sup>‡</sup>  
<sup>†</sup>*National Technical University of Athens*    <sup>‡</sup>*ETH Zürich*    <sup>\*</sup>*University of Toronto*    <sup>§</sup>*University of Malaga*

# Adoption: How to Support Virtual Memory?

---

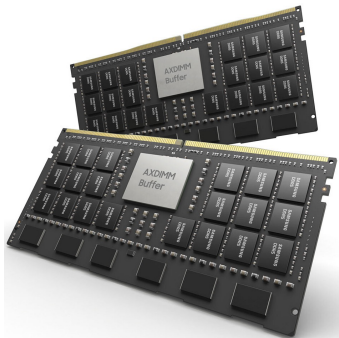
- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,  
["Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation"](#)  
*Proceedings of the 34th IEEE International Conference on Computer Design (ICCD)*, Phoenix,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

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

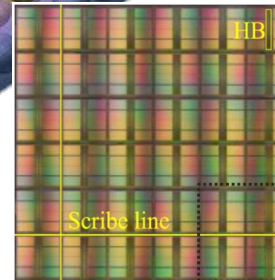
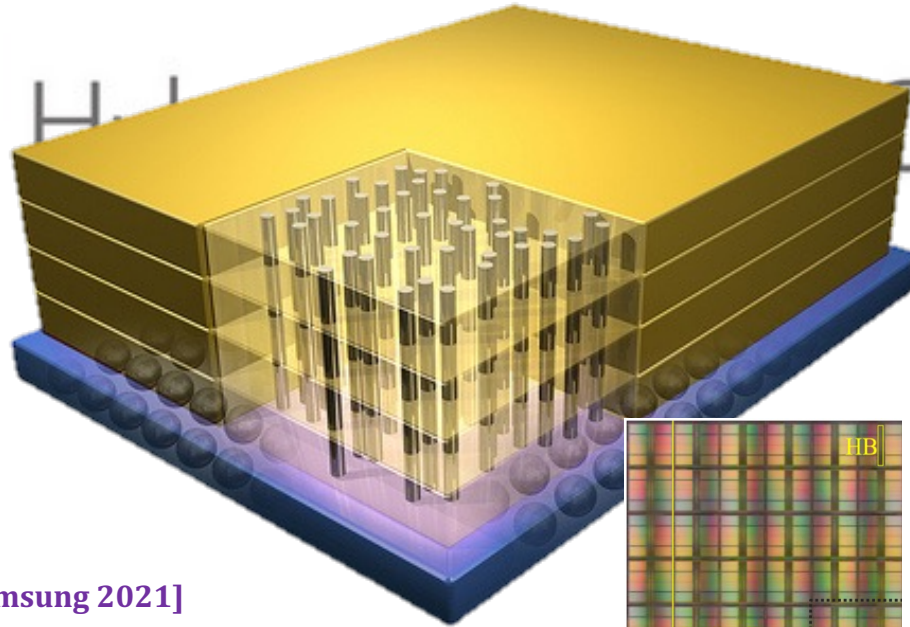
Kevin Hsieh<sup>†</sup>   Samira Khan<sup>‡</sup>   Nandita Vijaykumar<sup>†</sup>  
Kevin K. Chang<sup>†</sup>   Amirali Boroumand<sup>†</sup>   Saugata Ghose<sup>†</sup>   Onur Mutlu<sup>§†</sup>  
<sup>†</sup>*Carnegie Mellon University*   <sup>‡</sup>*University of Virginia*   <sup>§</sup>*ETH Zürich*

## Processing-in-Memory in the Real World

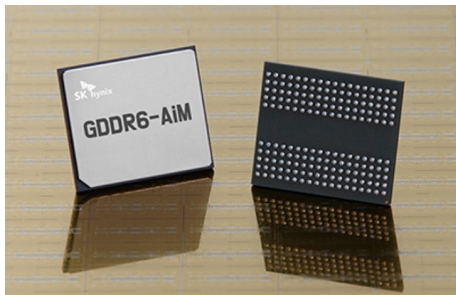
# Processing-in-Memory Landscape Today



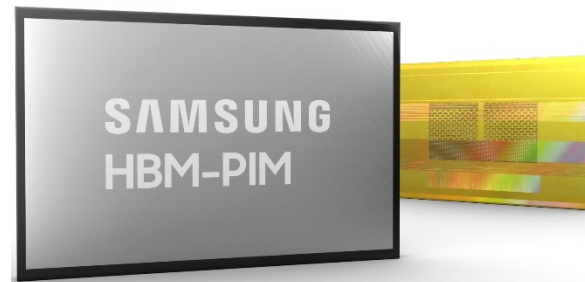
[Samsung 2021]



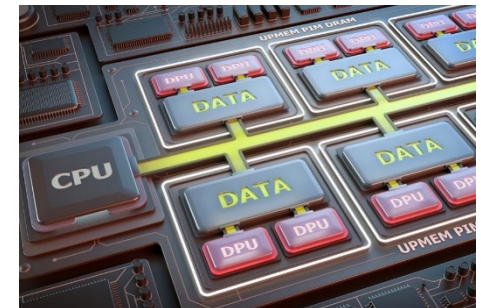
[Alibaba 2022]



[SK Hynix 2022]



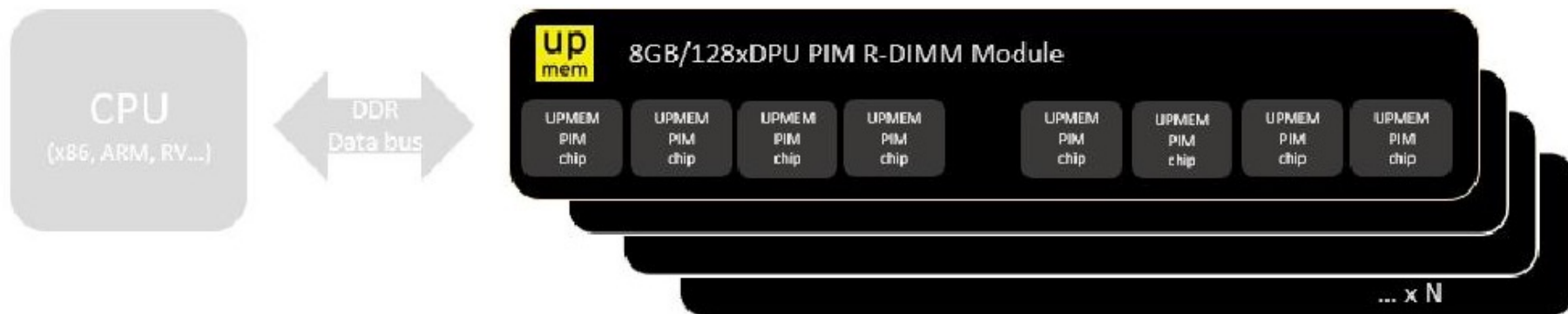
[Samsung 2021]



[UPMEM 2019]

# UPMEM Processing-in-DRAM Engine (2019)

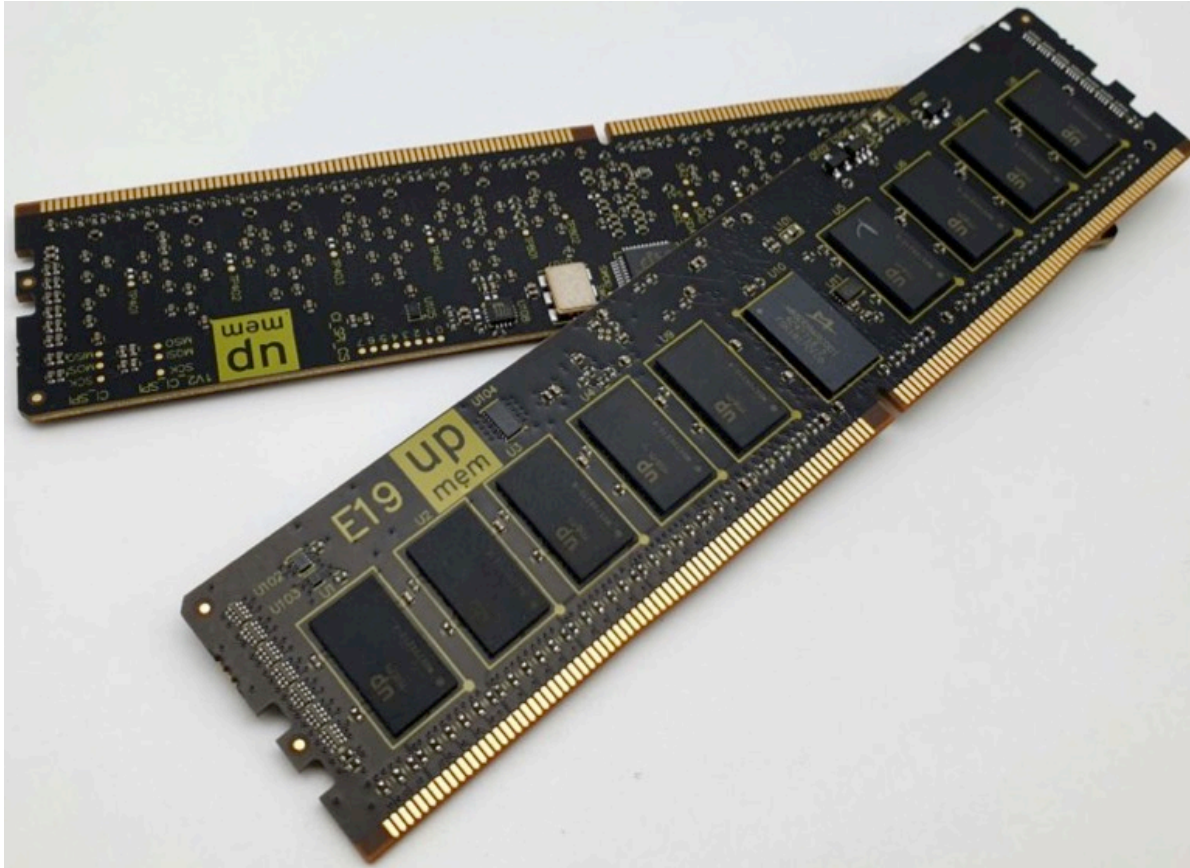
- **Processing in DRAM Engine**
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.
- Replaces **standard DIMMs**
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory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





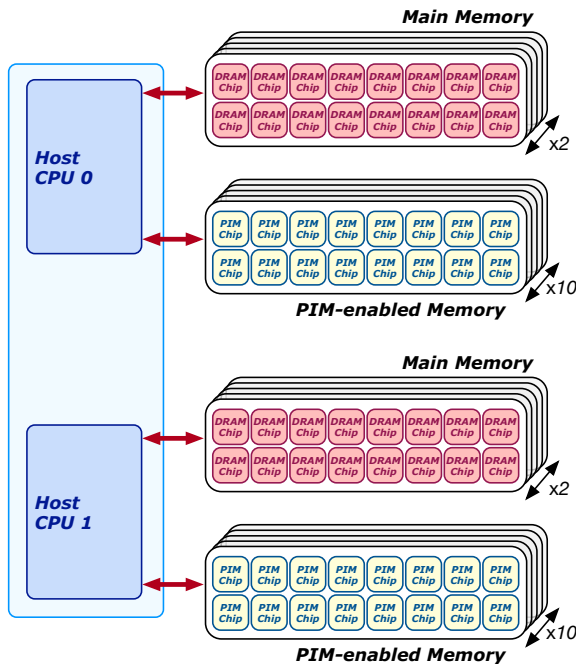
# UPMEM Memory Modules

- 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



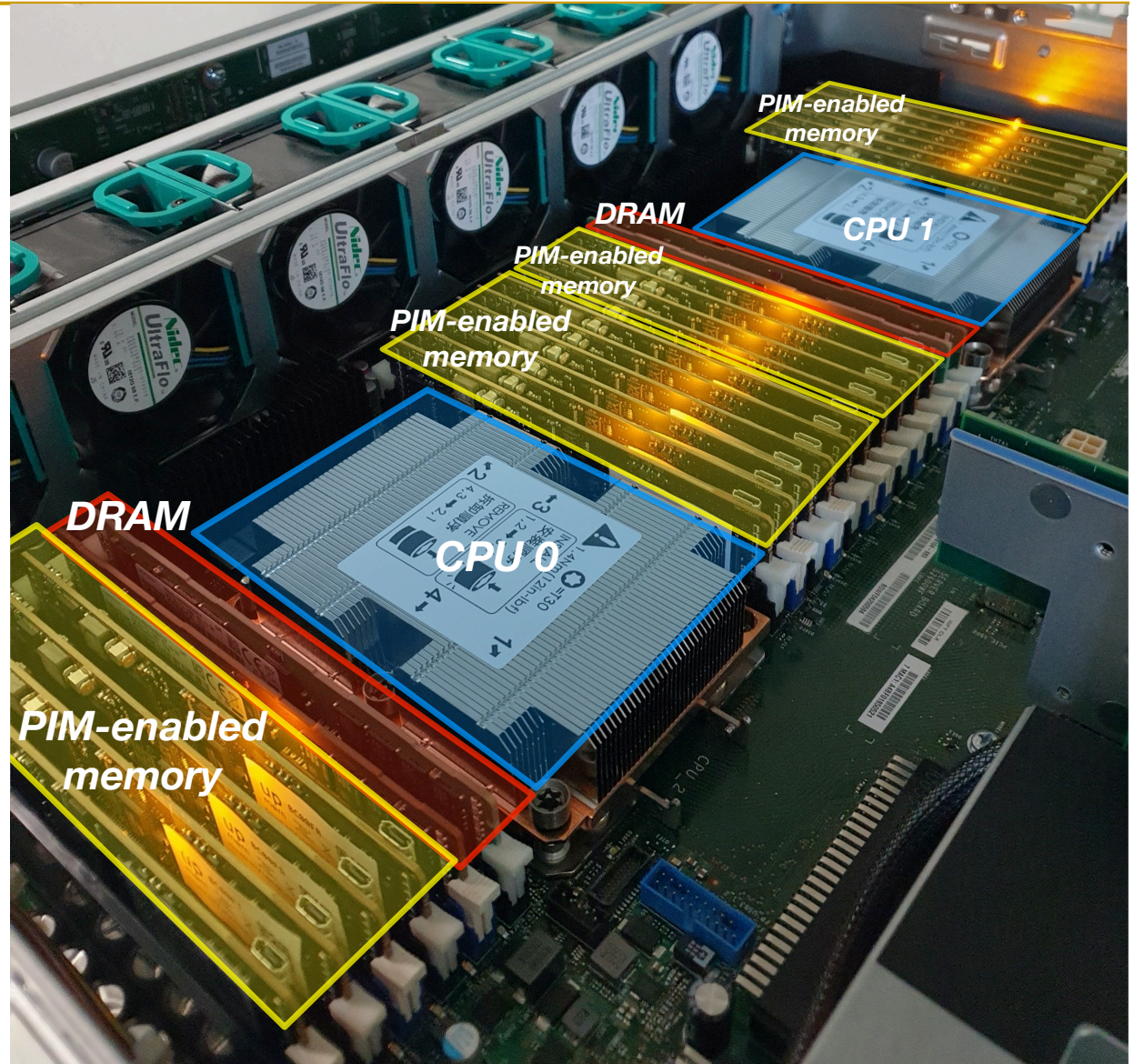
## Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

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.

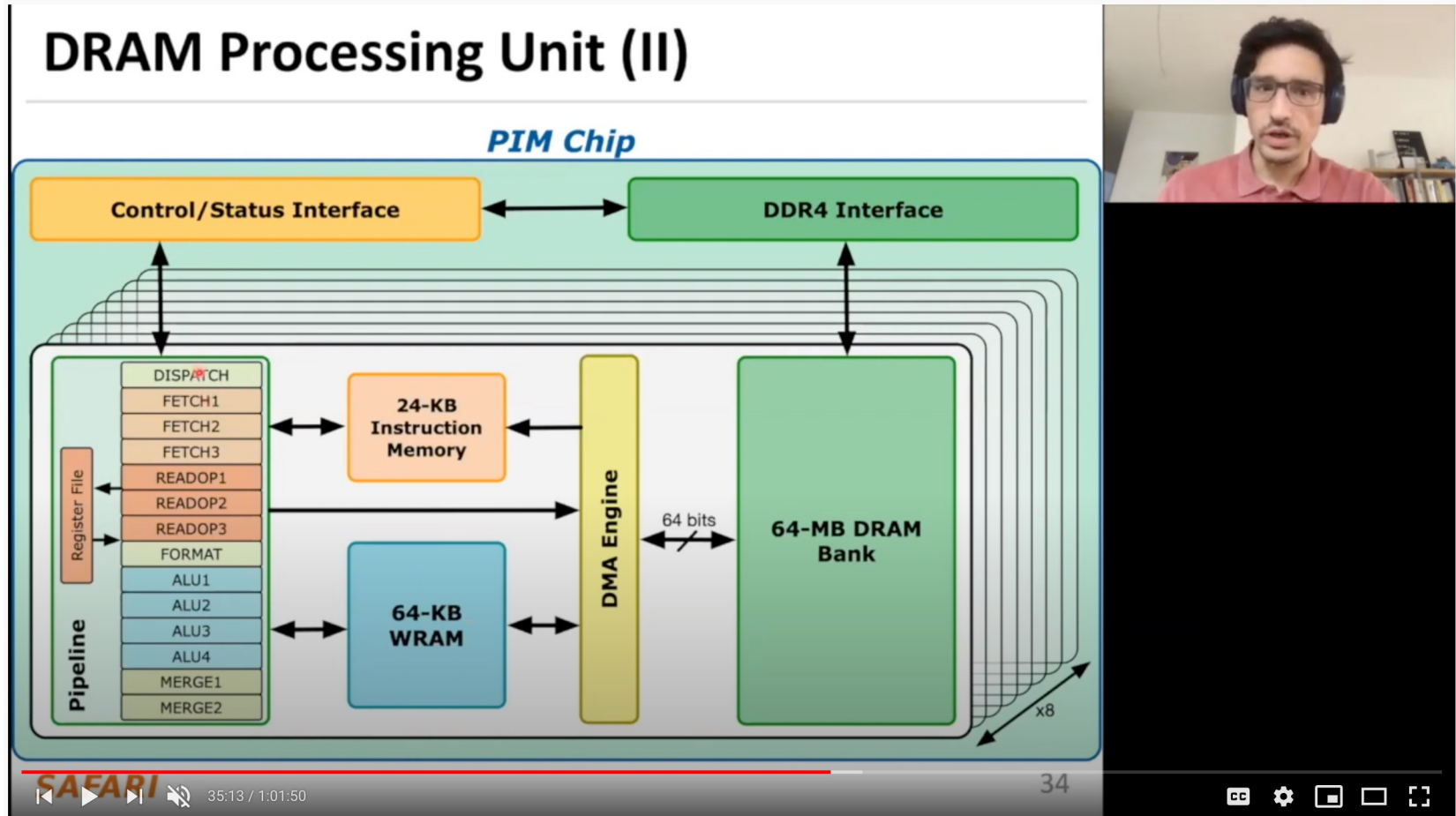
This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute 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.



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



# 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

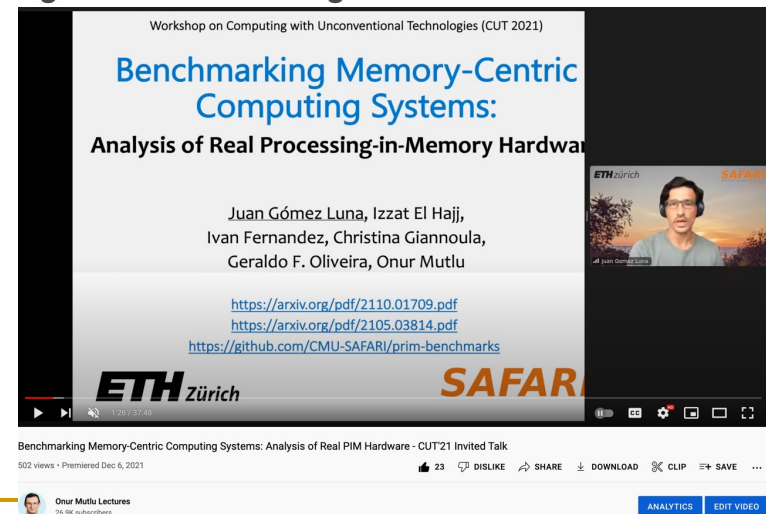
Izzat El Hajj, American University of Beirut

Ivan Fernandez, University of Malaga

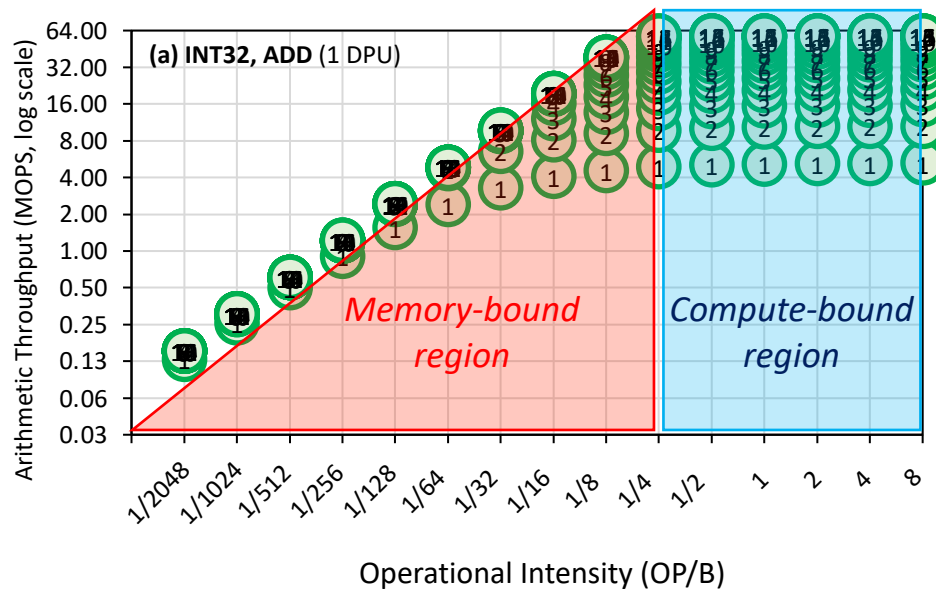
Christina Giannoula, National Technical University of Athens

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

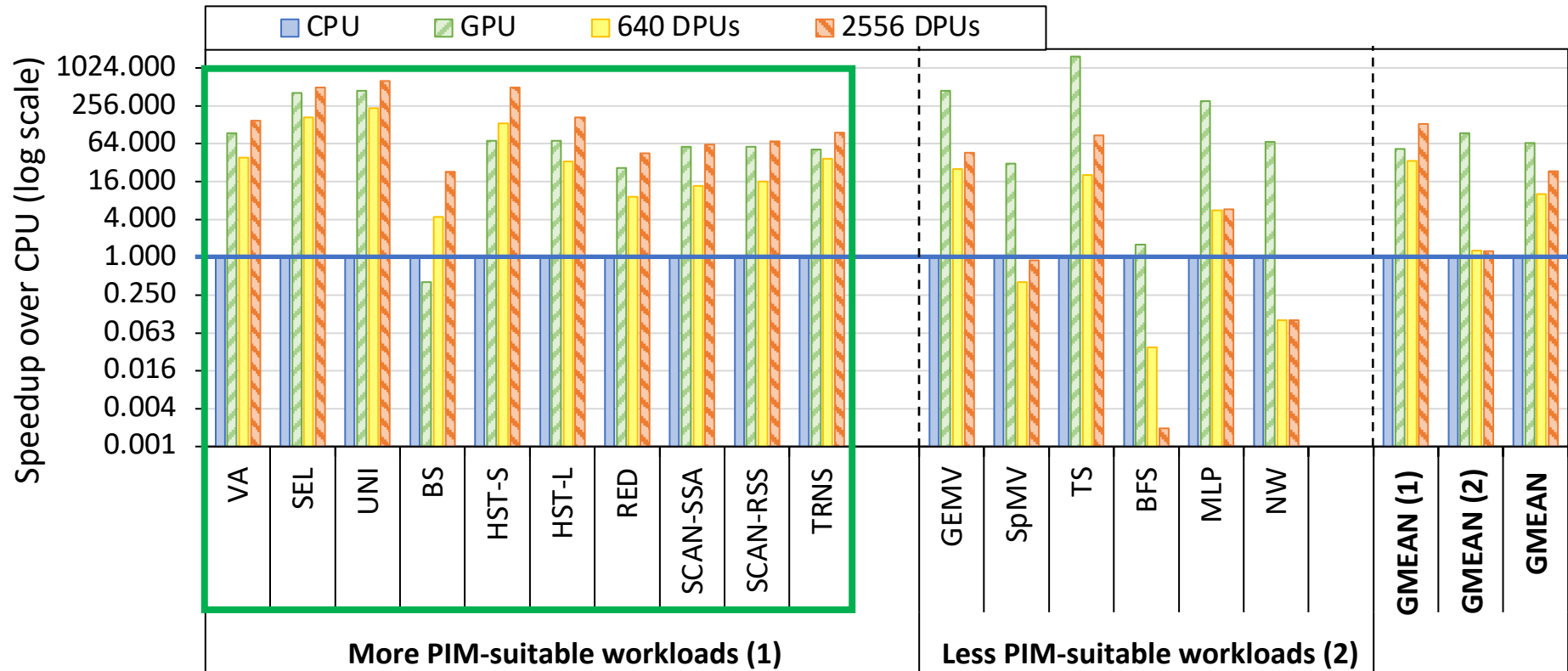
The UPMEM PIM architecture is fundamentally compute bound. As a result, the most suitable workloads are memory-bound.

# 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 <sup>†</sup>    | 350 MHz   | 894.6 GOPS       | 159.75 GB | 1.7 TB/s        | 383 W <sup>†</sup> |
| 640-DPU PIM System              | 2x nm        | 640                   | 267 MHz   | 170.9 GOPS       | 40 GB     | 333.75 GB/s     | 96 W <sup>†</sup>  |

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



## KEY TAKEAWAY 2

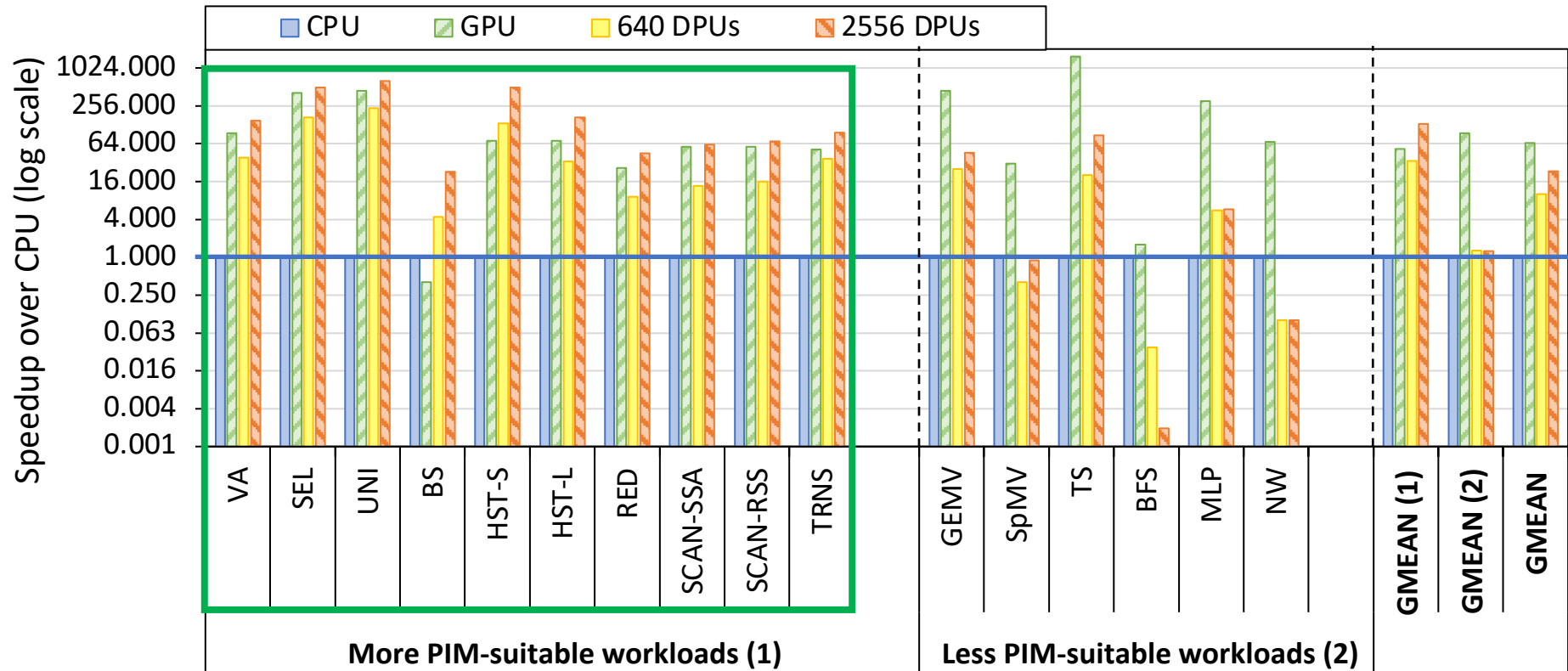
The most well-suited workloads for the UPMEM PIM architecture use no arithmetic operations or use only simple operations (e.g., bitwise operations and integer addition/subtraction).

# 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 <sup>9</sup>    | 350 MHz   | 894.6 GOPS       | 159.75 GB | 1.7 TB/s        | 383 W <sup>†</sup> |
| 640-DPU PIM System              | 2x nm        | 640                   | 267 MHz   | 170.9 GOPS       | 40 GB     | 333.75 GB/s     | 96 W <sup>†</sup>  |

\*Estimated GFLOPS = 3.3 GHz × 4 cores × 2 instructions per cycle.  
<sup>9</sup>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](mailto: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

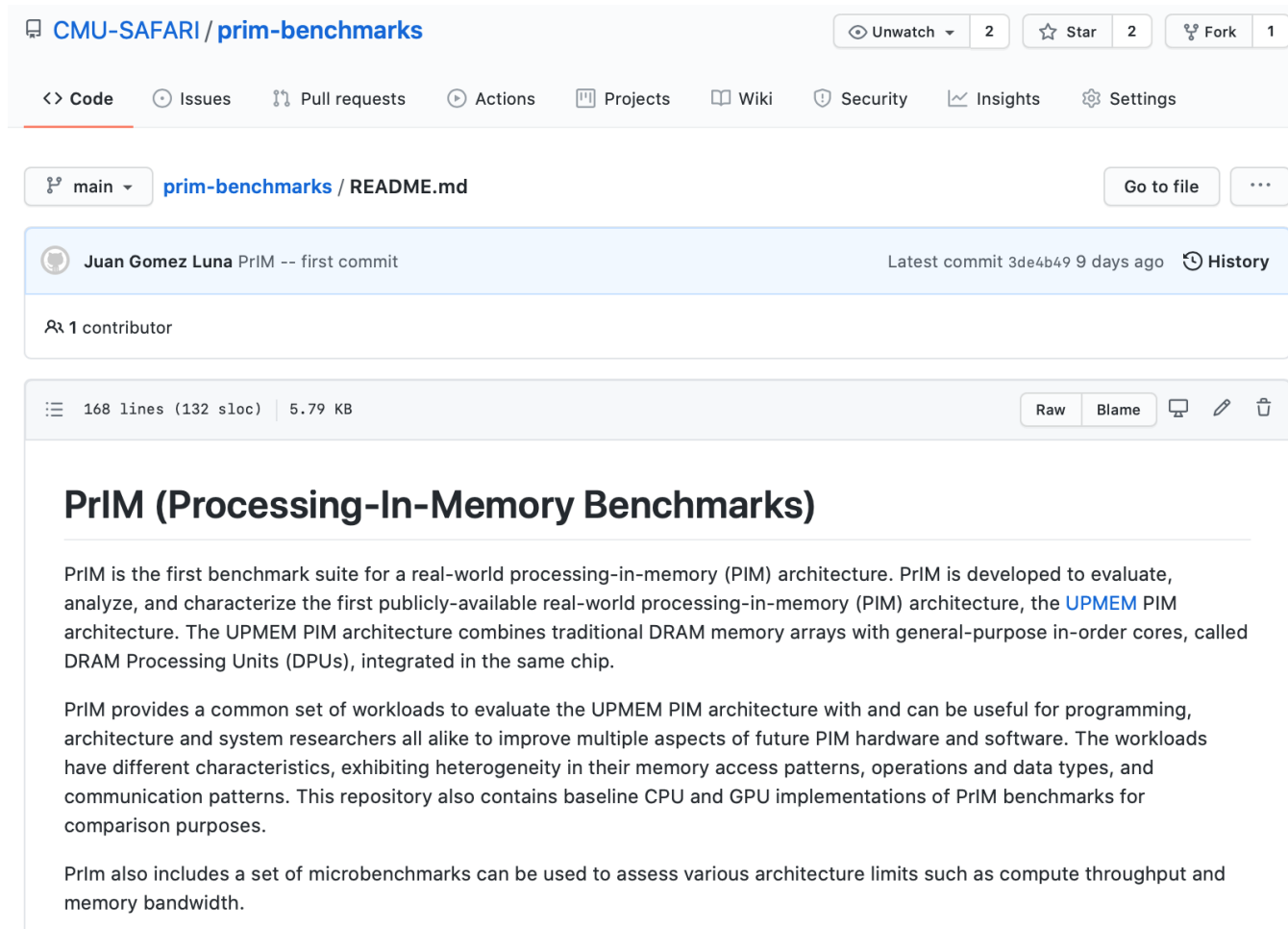
|                   |                                          |                                 |                                                    |                     |                   |
|-------------------|------------------------------------------|---------------------------------|----------------------------------------------------|-------------------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



The screenshot shows the GitHub repository page for `CMU-SAFARI/prim-benchmarks`. At the top, there are buttons for 'Unwatch', 'Star' (2), and 'Fork' (1). Below these are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'main' branch is selected, and the file `prim-benchmarks / README.md` is open. The commit history shows a single commit by Juan Gomez Luna. The file statistics indicate 168 lines (132 sloc) and 5.79 KB. The README content describes the PrIM benchmark suite, its purpose, and its components.

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-LUNA<sup>1</sup>, IZZAT EL HAJJ<sup>2</sup>, IVAN FERNANDEZ<sup>1,3</sup>, CHRISTINA GIANNOULA<sup>1,4</sup>,  
GERALDO F. OLIVEIRA<sup>1</sup>, AND ONUR MUTLU<sup>1</sup>**

<sup>1</sup>ETH Zürich

<sup>2</sup>American University of Beirut

<sup>3</sup>University of Malaga

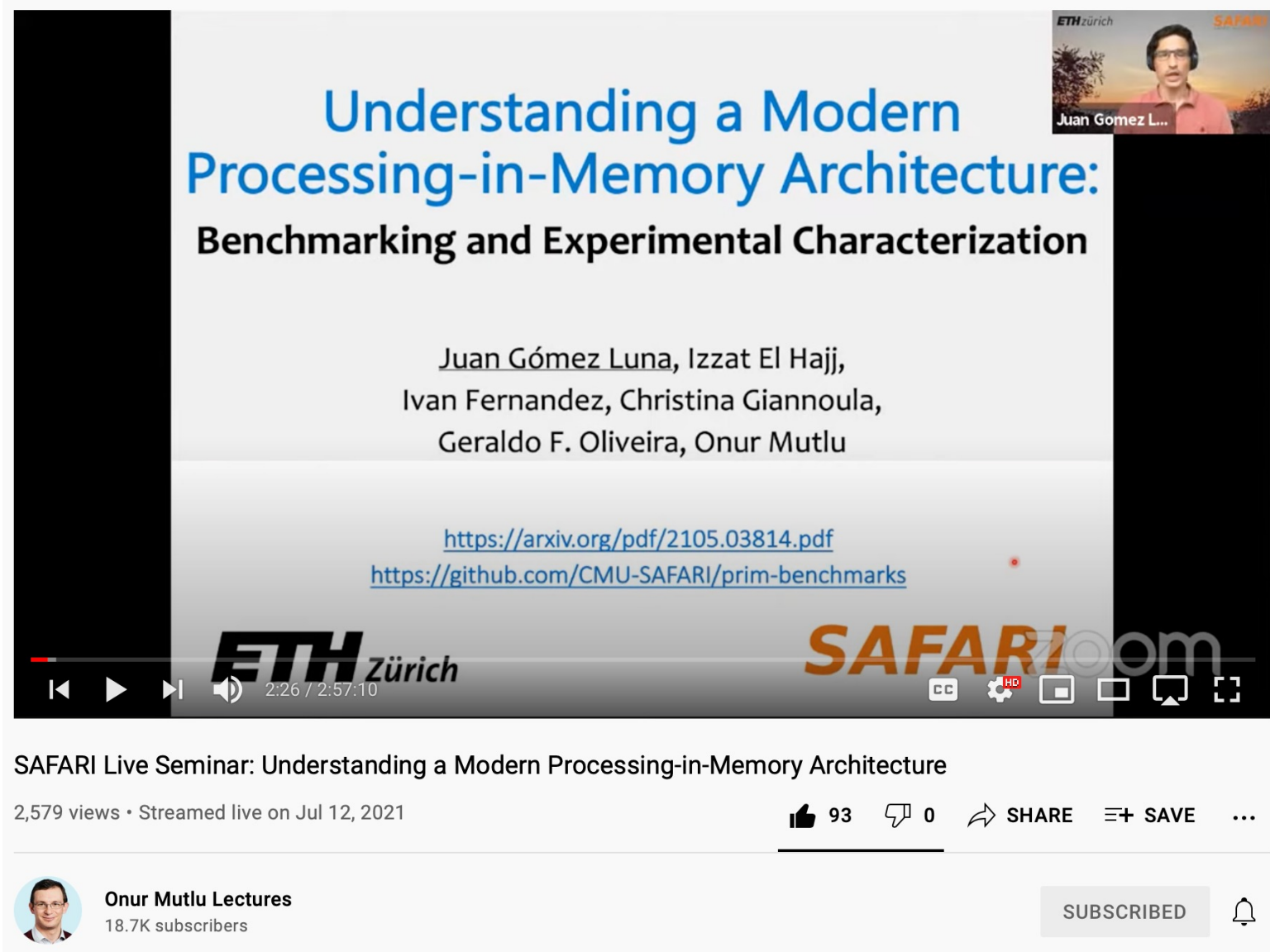
<sup>4</sup>National Technical University of Athens

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

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

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

# Understanding a Modern PIM Architecture



The image shows a YouTube video player interface. The video title is "Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization". The speaker is Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu. The video is from the SAFARI Live Seminar series. The video player shows a progress bar at 2:26 / 2:57:10. The video is from the channel "Onur Mutlu Lectures" with 18.7K subscribers. The video has 2,579 views and was streamed live on Jul 12, 2021. The video player includes a video player with a title, description, and a video player interface with a progress bar and controls. The video player is embedded in a YouTube page. The video player shows a progress bar at 2:26 / 2:57:10. The video is from the channel "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 ...

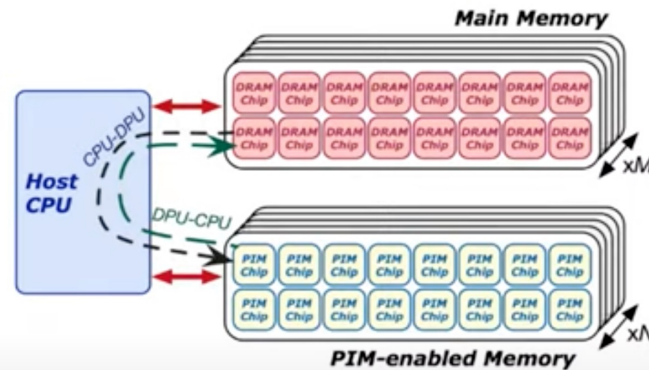
 **Onur Mutlu Lectures**  
18.7K subscribers

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

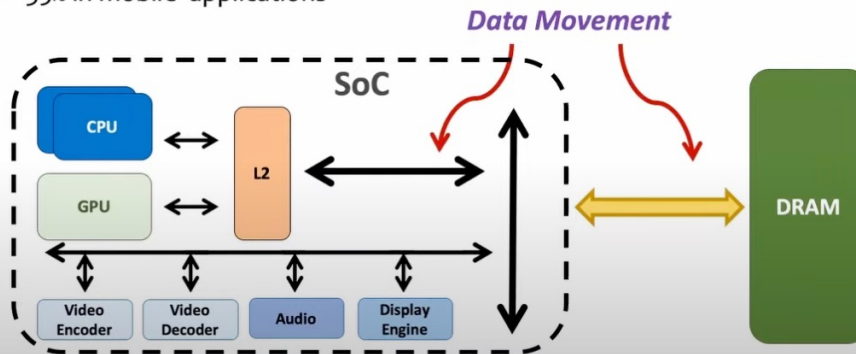
[https://www.youtube.com/watch?v=D8Hjy2IU9l4&list=PL5Q2soXY2Zi\\_tOTAYm--dYByNPL7JhwR9](https://www.youtube.com/watch?v=D8Hjy2IU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9)



# 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](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-Luna<sup>1</sup> Yuxin Guo<sup>1</sup> Sylvan Brocard<sup>2</sup> Julien Legriel<sup>2</sup>  
Remy Cimadomo<sup>2</sup> Geraldo F. Oliveira<sup>1</sup> Gagandeep Singh<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich <sup>2</sup>UPMEM

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

Juan Gómez-Luna<sup>1</sup> Yuxin Guo<sup>1</sup> Sylvan Brocard<sup>2</sup> Julien Legriel<sup>2</sup>  
Remy Cimadomo<sup>2</sup> Geraldo F. Oliveira<sup>1</sup> Gagandeep Singh<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich <sup>2</sup>UPMEM

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



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

Juan Gómez Luna, Yuxin Guo, Sylvan Brocard,  
Julien Legriel, Remy Cimadomo, Geraldo F. Oliveira,  
Gagandeep Singh, Onur Mutlu

<https://arxiv.org/pdf/2206.06022.pdf>  
[juang@ethz.ch](mailto:juang@ethz.ch)

**ETH** Zürich **SAFARI** **up mem**

3:07:11 / 3:36:35 • Dr. Juan Gómez-Luna, "Machine Learning Training on a Real Processing-In-Memory System" >

1,345 views • Premiered Aug 9, 2022

61 DISLIKE SHARE DOWNLOAD CLIP SAVE ...

Onur Mutlu Lectures  
26.9K subscribers

ANALYTICS EDIT VIDEO

# SpMV Multiplication on Real PIM Systems

---

- Appears in 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

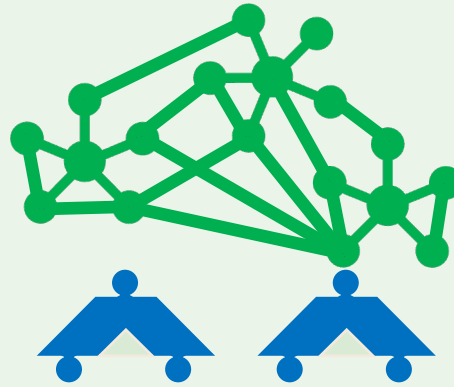
NECTARIOS KOZIRIS, National Technical University of Athens, Greece

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

 **CSLab**  
National Technical University of Athens



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



The screenshot shows a YouTube video player with a presentation slide. The slide features the SparseP logo, which consists of a green network graph above two blue stylized human figures. Below the logo, the title "SparseP" is written in large blue letters, followed by the subtitle "Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures" in smaller blue letters. The presenter's name, "Christina Giannoula", is listed in red, followed by the names of other contributors: "Ivan Fernandez, Juan Gomez-Luna, Nectarios Koziris, Georgios Goumas, Onur Mutlu". At the bottom of the slide, there are logos for "SAFARI", "ETH zürich", "CSLab", and "UNIVERSITY OF MALAGA". A small video inset in the top right corner shows the presenter, Christina Giannoula, with her name "Christina Gian..." visible below it. The video player interface includes a progress bar at 0:02 / 55:25, a volume icon, and various control buttons like play, pause, and full screen.

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

# 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



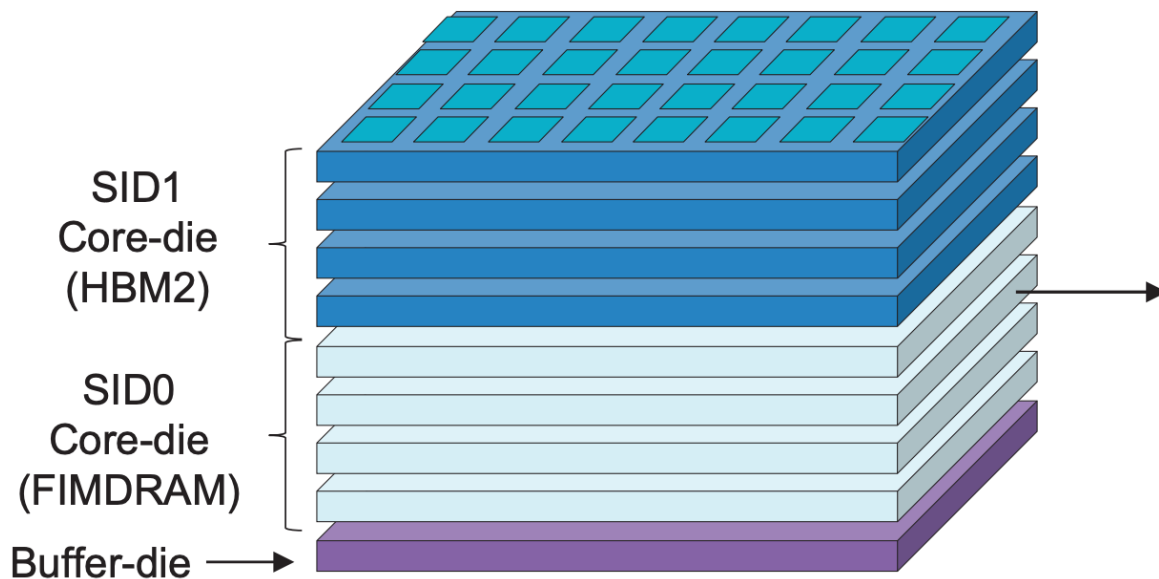
*The new architecture will deliver over twice the system performance and reduce energy consumption by more than 70%*

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 Kwon<sup>1</sup>, Suk Han Lee<sup>1</sup>, Jaehoon Lee<sup>1</sup>, Sang-Hyuk Kwon<sup>1</sup>, Je Min Ryu<sup>1</sup>, Jong-Pil Son<sup>1</sup>, Seongil O<sup>1</sup>, Hak-Soo Yu<sup>1</sup>, Haesuk Lee<sup>1</sup>, Soo Young Kim<sup>1</sup>, Youngmin Cho<sup>1</sup>, Jin Guk Kim<sup>1</sup>, Jongyoon Choi<sup>1</sup>, Hyun-Sung Shin<sup>1</sup>, Jin Kim<sup>1</sup>, BengSeng Phuah<sup>1</sup>, HyoungMin Kim<sup>1</sup>, Myeong Jun Song<sup>1</sup>, Ahn Choi<sup>1</sup>, Daeho Kim<sup>1</sup>, SooYoung Kim<sup>1</sup>, Eun-Bong Kim<sup>1</sup>, David Wang<sup>2</sup>, Shinhaeng Kang<sup>1</sup>, Yuhwan Ro<sup>3</sup>, Seungwoo Seo<sup>3</sup>, JoonHo Song<sup>3</sup>, Jaeyoun Youn<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

<sup>1</sup>Samsung Electronics, Hwaseong, Korea

<sup>2</sup>Samsung Electronics, San Jose, CA

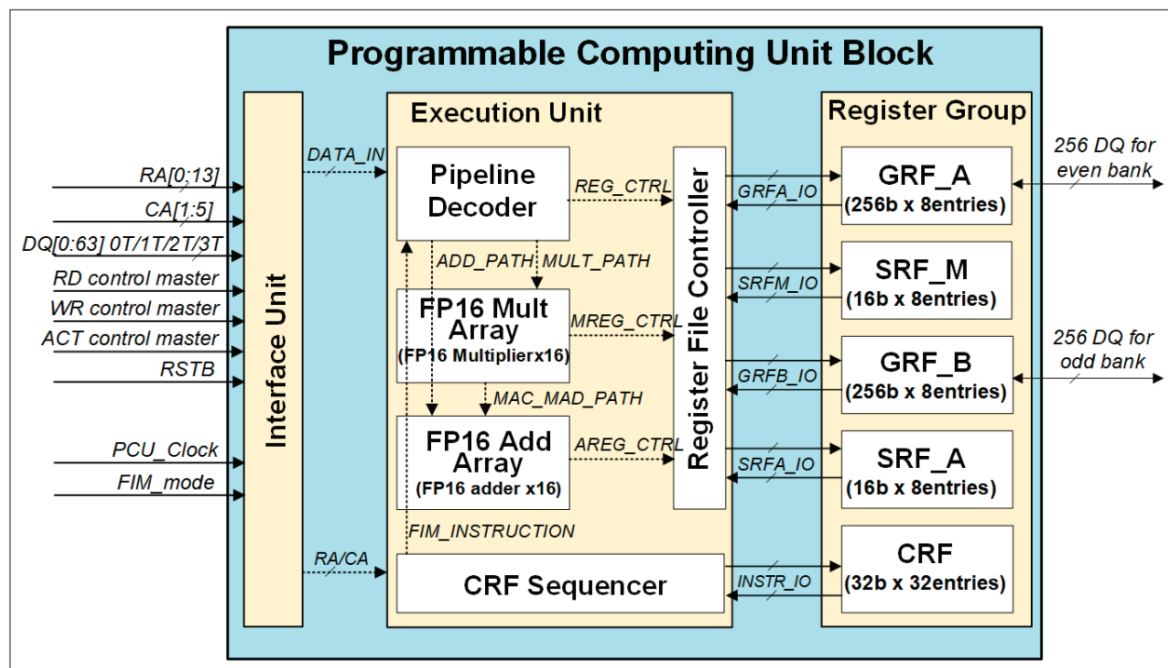
<sup>3</sup>Samsung Electronics, Suwon, Korea

# 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 Kwon<sup>1</sup>, Suk Han Lee<sup>1</sup>, Jaehoon Lee<sup>1</sup>, Sang-Hyuk Kwon<sup>1</sup>, Je Min Ryu<sup>1</sup>, Jong-Pil Son<sup>1</sup>, Seongil O<sup>1</sup>, Hak-Soo Yu<sup>1</sup>, Haesuk Lee<sup>1</sup>, Soo Young Kim<sup>1</sup>, Youngmin Cho<sup>1</sup>, Jin Guk Kim<sup>1</sup>, Jongyoon Choi<sup>1</sup>, Hyun-Sung Shin<sup>1</sup>, Jin Kim<sup>1</sup>, BengSeng Phuah<sup>1</sup>, HyungMin Kim<sup>1</sup>, Myeong Jun Song<sup>1</sup>, Ahn Choi<sup>1</sup>, Daeho Kim<sup>1</sup>, SooYoung Kim<sup>1</sup>, Eun-Bong Kim<sup>1</sup>, David Wang<sup>1</sup>, Shinhaeng Kang<sup>1</sup>, Yuhwan Ro<sup>1</sup>, Seungwoo Seo<sup>1</sup>, JoonHo Song<sup>1</sup>, Jaeyoun Youn<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

<sup>1</sup>Samsung Electronics, Hwaseong, Korea  
<sup>2</sup>Samsung Electronics, San Jose, CA  
<sup>3</sup>Samsung 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 Kwon<sup>1</sup>, Suk Han Lee<sup>1</sup>, Jaehoon Lee<sup>1</sup>, Sang-Hyuk Kwon<sup>1</sup>, Je Min Ryu<sup>1</sup>, Jong-Pil Son<sup>1</sup>, Seongil O<sup>1</sup>, Hak-Soo Yu<sup>1</sup>, Haesuk Lee<sup>1</sup>, Soo Young Kim<sup>1</sup>, Youngmin Cho<sup>1</sup>, Jin Guk Kim<sup>1</sup>, Jongyoon Choi<sup>1</sup>, Hyun-Sung Shin<sup>1</sup>, Jin Kim<sup>1</sup>, BengSeng Phuah<sup>1</sup>, HyungMin Kim<sup>1</sup>, Myeong Jun Song<sup>1</sup>, Ahn Choi<sup>1</sup>, Daeho Kim<sup>1</sup>, SooYoung Kim<sup>1</sup>, Eun-Bong Kim<sup>1</sup>, David Wang<sup>2</sup>, Shinhaeng Kang<sup>1</sup>, Yuhwan Ro<sup>1</sup>, Seungwoo Seo<sup>1</sup>, JoonHo Song<sup>1</sup>, Jaeyoun Youn<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

<sup>1</sup>Samsung Electronics, Hwaseong, Korea

<sup>2</sup>Samsung Electronics, San Jose, CA

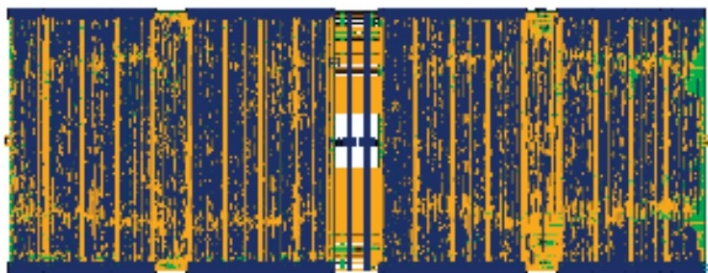
<sup>3</sup>Samsung Electronics, Suwon, Korea



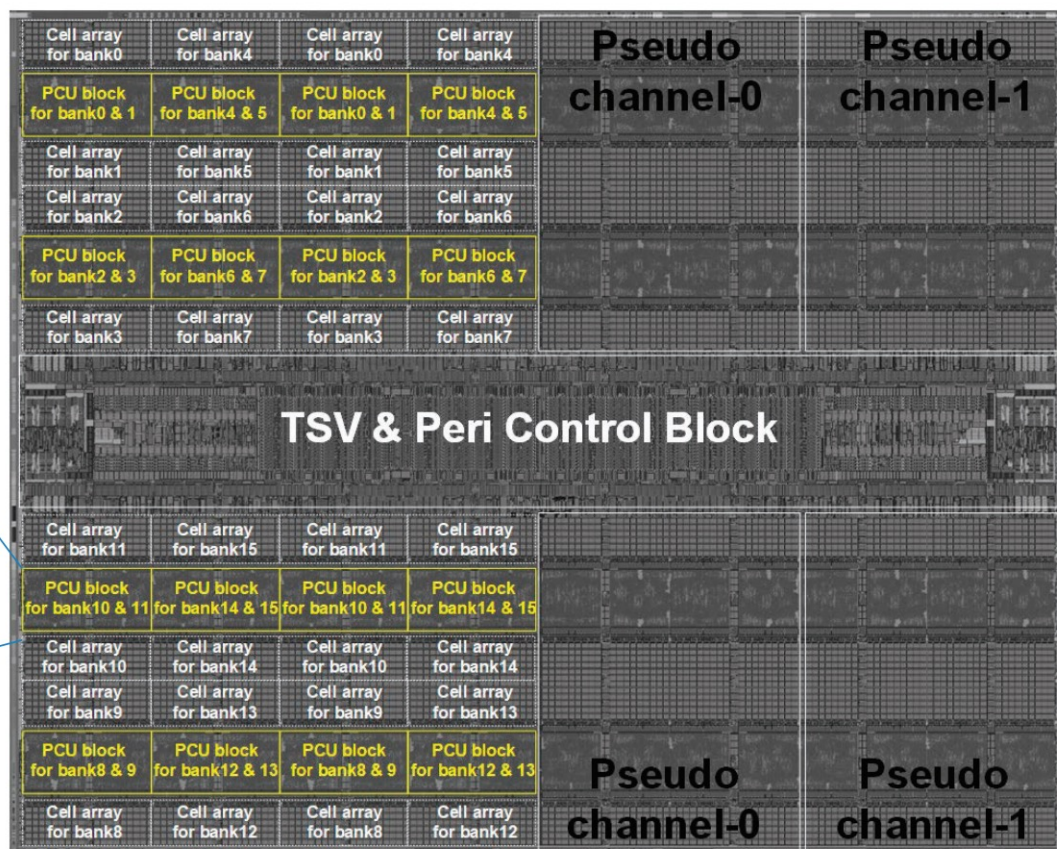
# Samsung Function-in-Memory DRAM (2021)

## 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

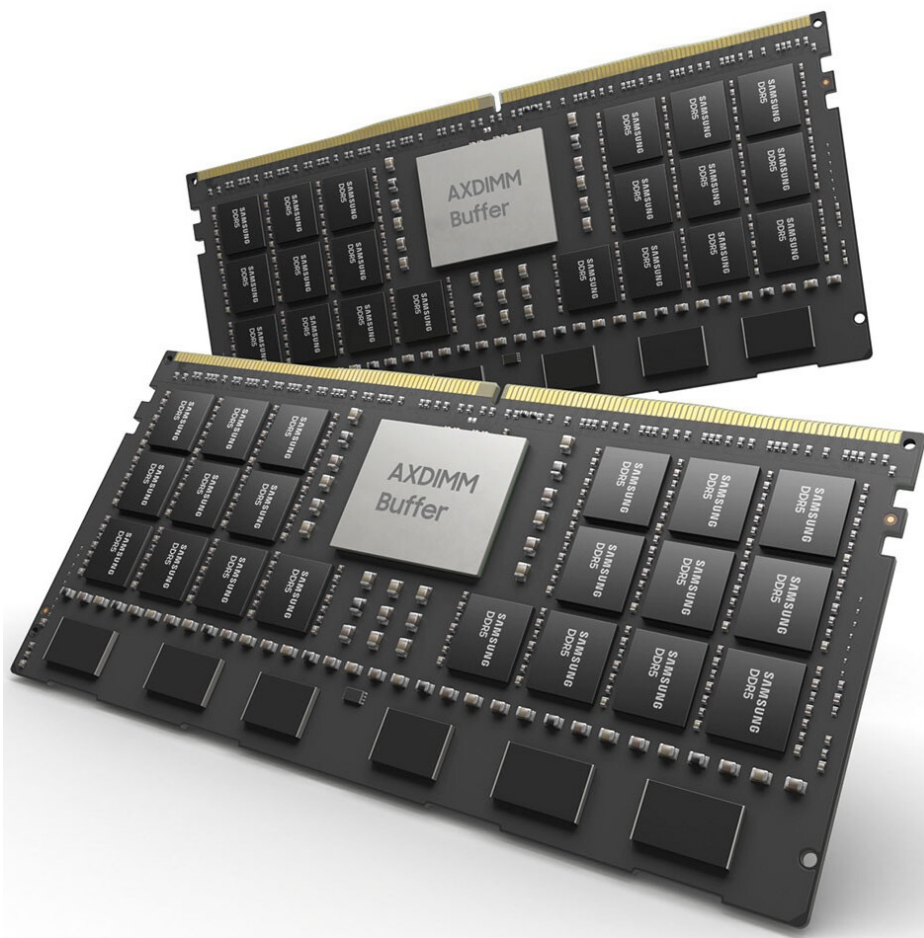
25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon<sup>1</sup>, Suk Han Lee<sup>1</sup>, Jaehoon Lee<sup>1</sup>, Sang-Hyuk Kwon<sup>1</sup>, Je Min Ryu<sup>1</sup>, Jong-Pil Son<sup>1</sup>, Seongil O<sup>1</sup>, Hak-Soo Yu<sup>1</sup>, Haesuk Lee<sup>1</sup>, Soo Young Kim<sup>1</sup>, Youngmin Cho<sup>1</sup>, Jin Guk Kim<sup>1</sup>, Jongyeon Choi<sup>1</sup>, Hyun-Sung Shim<sup>1</sup>, Jin Kim<sup>1</sup>, BengSeng Phuah<sup>1</sup>, HyounMin Kim<sup>1</sup>, Myeong Jun Song<sup>1</sup>, Ahn Chai<sup>1</sup>, Daeho Kim<sup>1</sup>, SooYoung Kim<sup>1</sup>, Eun-Bong Kim<sup>1</sup>, David Wang<sup>2</sup>, Shinfaeng Kang<sup>3</sup>, Yulwan Ro<sup>3</sup>, Seungwoo Seo<sup>3</sup>, JoonHo Song<sup>3</sup>, Jaeyoun Yoon<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

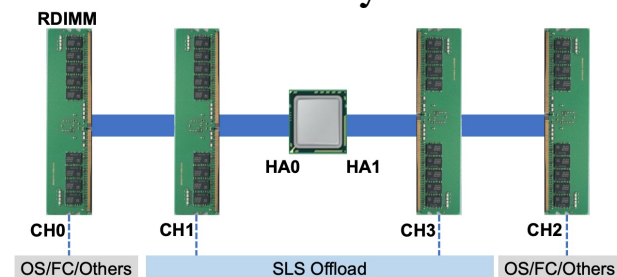
<sup>1</sup>Samsung Electronics, Hwaseong, Korea  
<sup>2</sup>Samsung Electronics, San Jose, CA  
<sup>3</sup>Samsung Electronics, Suwon, Korea

# Samsung AxDIMM (2021)

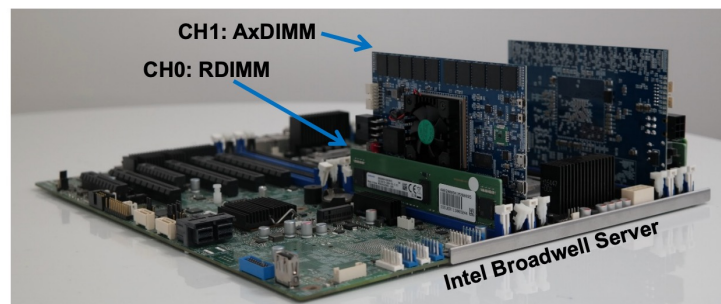
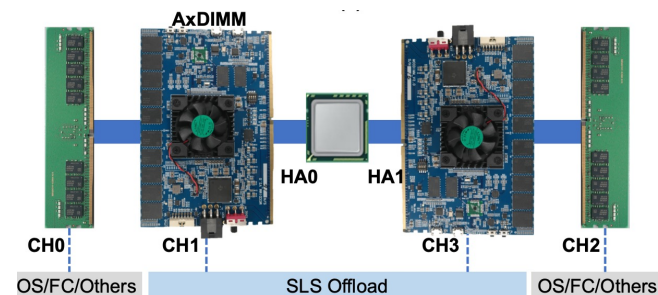
- DDRx-PIM
  - DLRM recommendation system



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](http://www.skhynix.com)) announced on February 16 that it has developed PIM\*, a next-generation memory chip with computing capabilities.

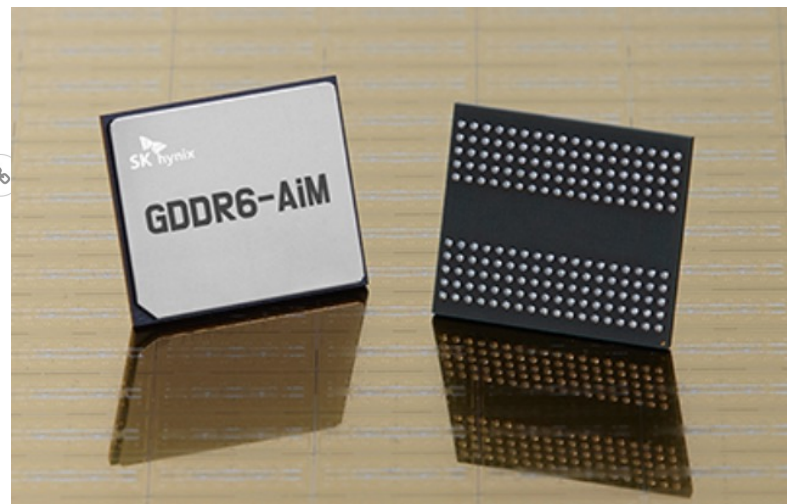
*\*PIM(Processing In Memory): A next-generation technology that provides a solution for data congestion issues for AI and big data by adding computational functions to semiconductor memory*

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.

# AliBaba PIM Recommendation System (2022)

ISSCC 2022 / February 24, 2022 / 8:30 AM

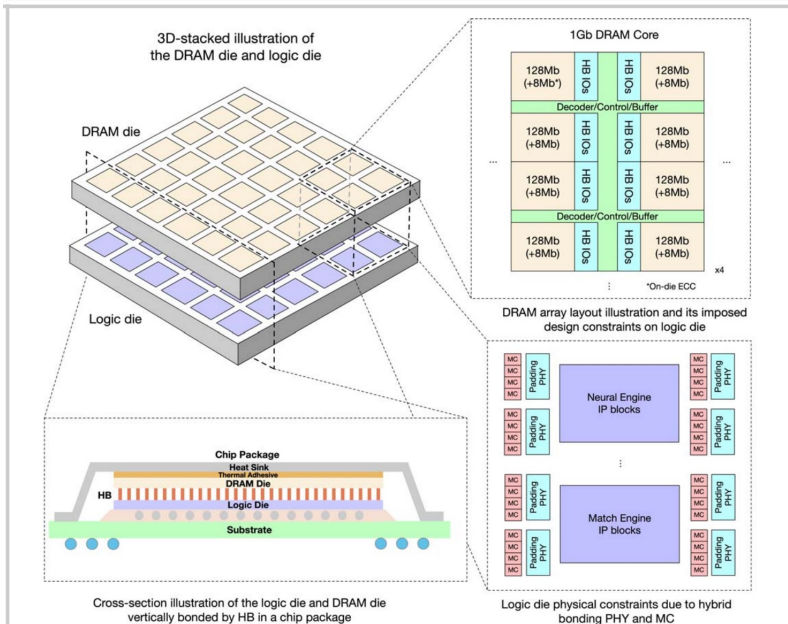


Figure 29.1.2: Illustration of 3D-stacked chip, cross-illustration of package, DRAM array layout and design blocks on logic die.

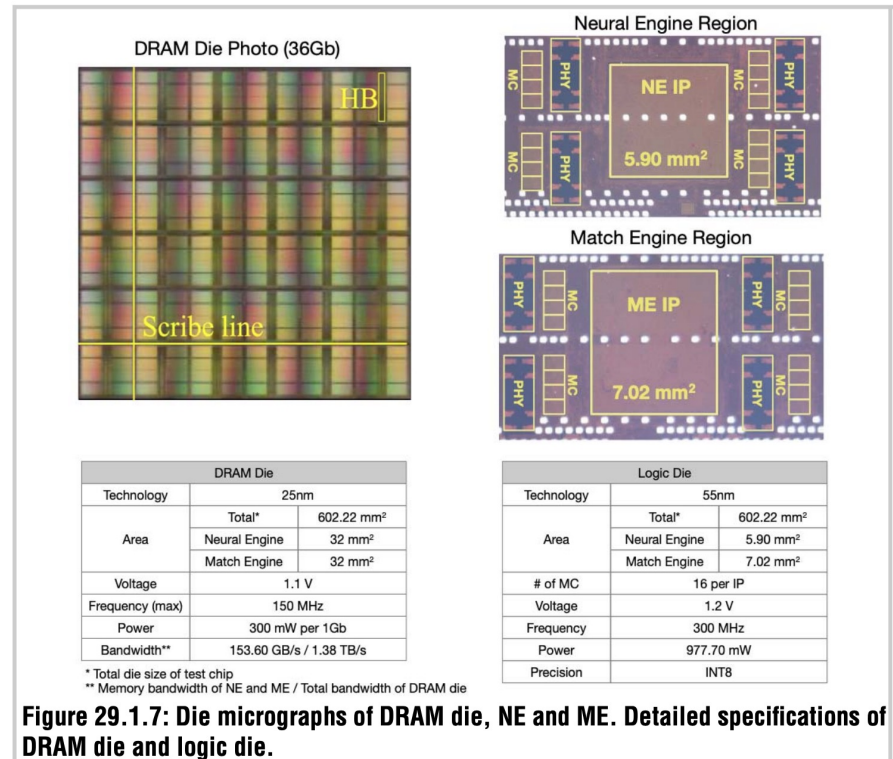


Figure 29.1.7: Die micrographs of DRAM die, NE and ME. Detailed specifications of DRAM die and logic die.

## 29.1 184QPS/W 64Mb/mm<sup>2</sup> 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System

Dimin Niu<sup>1</sup>, Shuangchen Li<sup>1</sup>, Yuhao Wang<sup>1</sup>, Wei Han<sup>1</sup>, Zhe Zhang<sup>2</sup>, Yijin Guan<sup>2</sup>, Tianchan Guan<sup>3</sup>, Fei Sun<sup>1</sup>, Fei Xue<sup>1</sup>, Lide Duan<sup>1</sup>, Yuanwei Fang<sup>1</sup>, Hongzhong Zheng<sup>1</sup>, Xiping Jiang<sup>4</sup>, Song Wang<sup>4</sup>, Fengguo Zuo<sup>4</sup>, Yubing Wang<sup>4</sup>, Bing Yu<sup>4</sup>, Qiwei Ren<sup>4</sup>, Yuan Xie<sup>1</sup>

# More Developments Ongoing (2022)

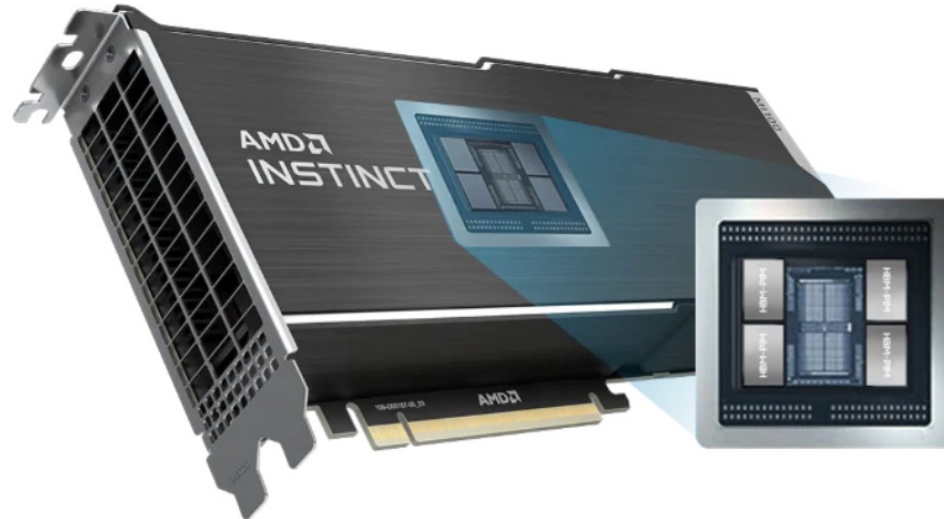
Tech Blog

## Samsung Electronics Semiconductor Unveils Cutting-edge Memory Technology to Accelerate Next-generation AI

Samsung Leading the Industry with HBM-PIM that supports Hyperscale AI and PNM technology based on CXL™

Implementation of AI-accelerator Equipped with HBM-PIM and GPU

In cooperation with AMD, Samsung installed HBM-PIM memory in AMD instinct™ Mi100 accelerator. Samsung then materialized an HBM-PIM Cluster based on this, and then applied it to various large-scale AI and High-Performance Computing (HPC) applications.



## 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

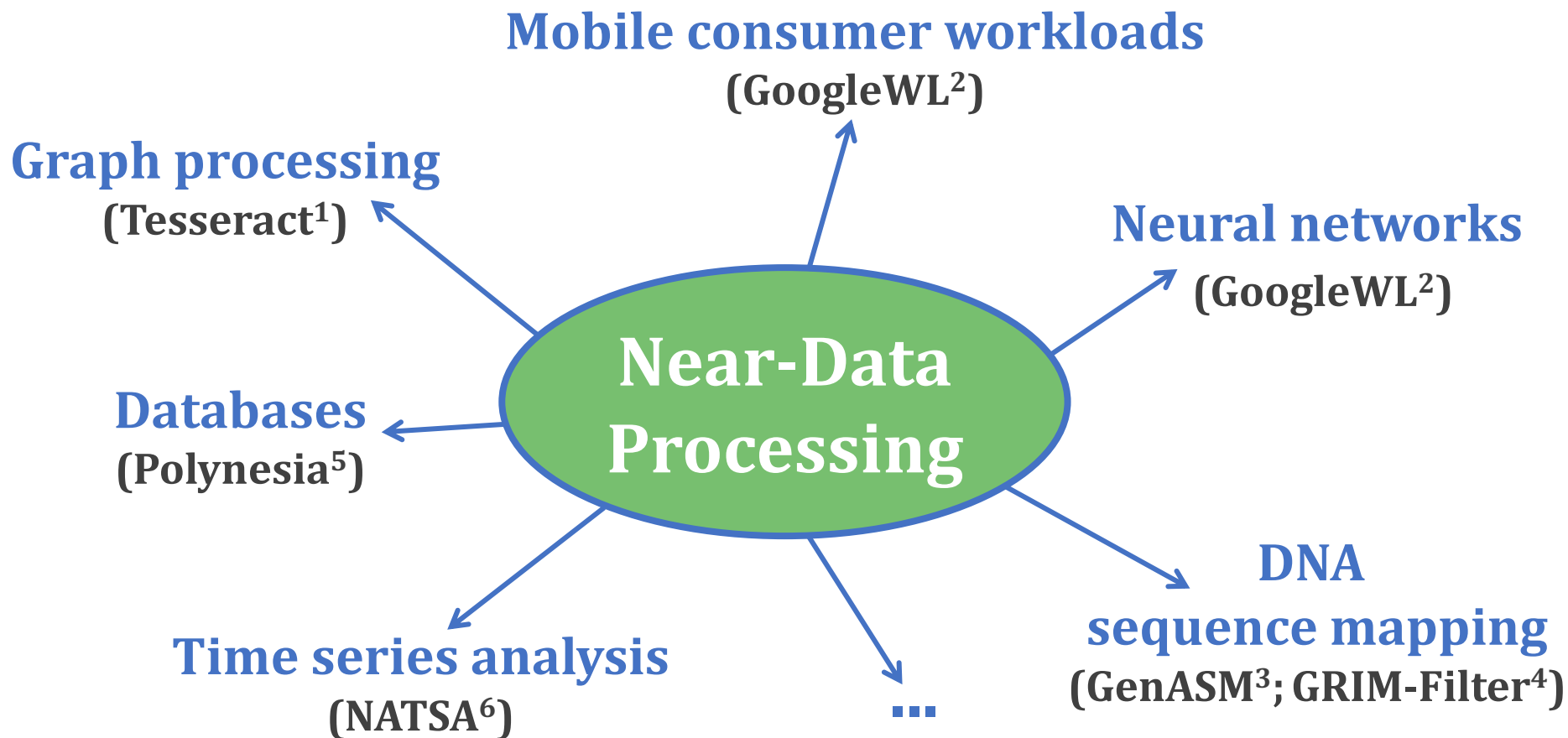
ONUR MUTLU, ETH Zürich, Switzerland

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

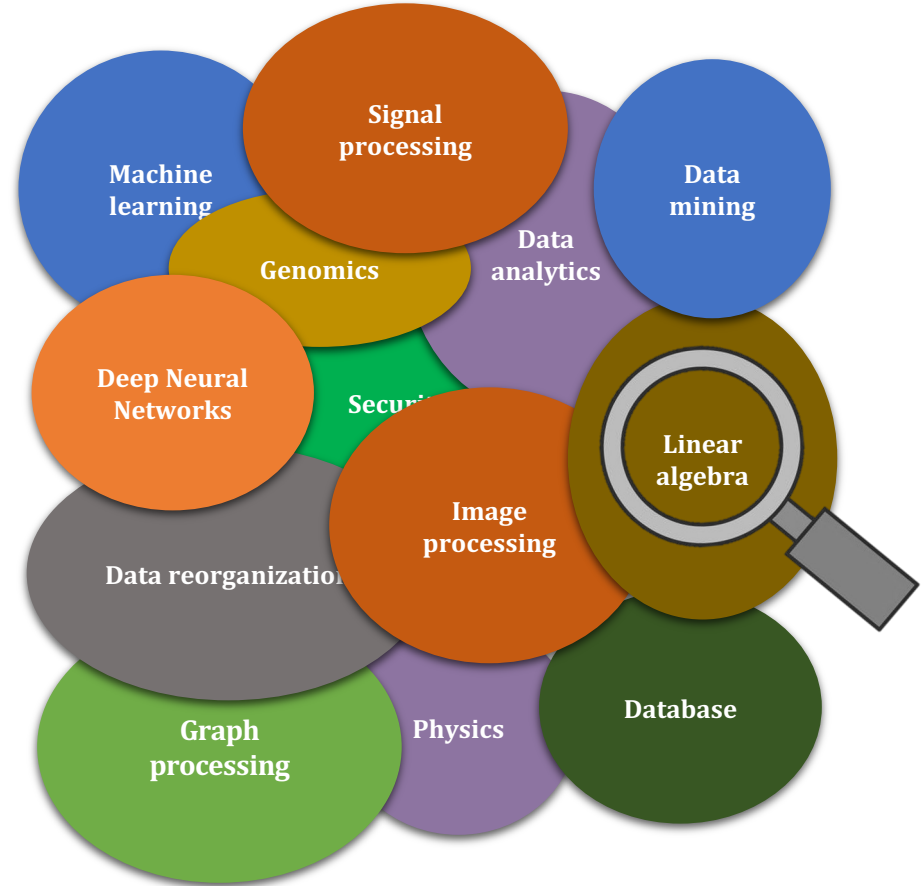
# Step 1: Application Profiling

- 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

...

**SAFARI**





# 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

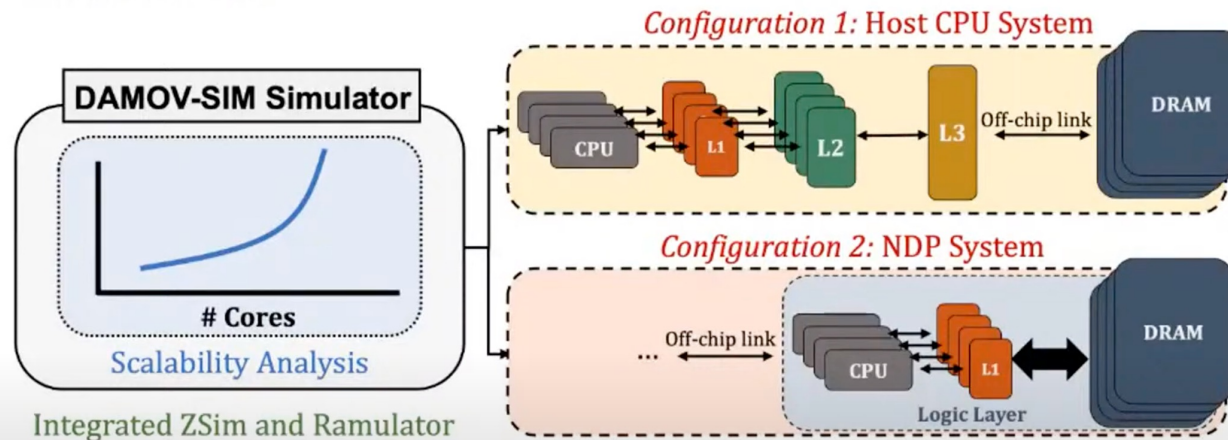
No packages published  
[Publish your first package](#)

### Languages

# More on DAMOV Analysis Methodology & Workloads

## Step 3: Memory Bottleneck Classification (2/2)

- **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](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

## Fundamentally Energy-Efficient (Data-Centric) Computing Architectures

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



# Computing Architectures with Minimal Data Movement

# More Info in This Longer Tutorial...

---

- Onur Mutlu,  
**"Memory-Centric Computing"**  
*Education Class at Embedded Systems Week (ESWEEK),*  
Virtual, 9 October 2021.  
[Slides (pptx) (pdf)]  
[Abstract (pdf)]  
[Talk Video (2 hours, including Q&A)]  
[Invited Paper at DATE 2021]  
["A Modern Primer on Processing in Memory" paper]

**<https://www.youtube.com/watch?v=N1Ac1ov1JOM>**

# Memory-Centric Computing

Onur Mutlu

[omutlu@gmail.com](mailto:omutlu@gmail.com)

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

9 October 2021

ESWEEK Education Class

**SAFARI**

**ETH** zürich

Carnegie Mellon



1:08 / 2:00:10



Embedded Systems Week (ESWEEK) 2021 Lecture - Memory-Centric Computing - Onur Mutlu - 9 October 2021

509 views • Premiered Dec 6, 2021

28 DISLIKE SHARE SAVE ...



Onur Mutlu Lectures  
20.7K subscribers

<https://www.youtube.com/watch?v=N1Ac1ov1JOM>

ANALYTICS

EDIT VIDEO

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

271

# Concluding Remarks

# Concluding Remarks

---

- We must design systems to be **balanced, high-performance, energy-efficient** (all at the same time) → intelligent systems
  - **Data-centric, data-driven, data-aware**
- 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**
  - ...
- Future of **truly memory-centric computing** is bright
  - We need to do research & design across the computing stack

**Data-centric**

**Data-driven**

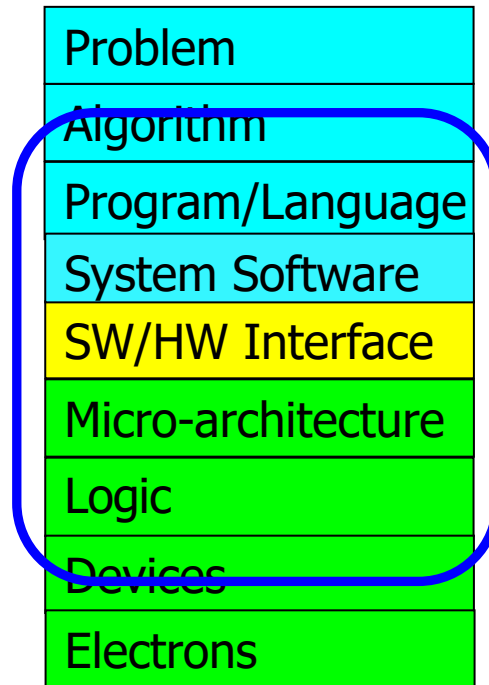
**Data-aware**





# We Need to Revisit the Entire Stack

---



**We can get there step by step**

# 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

# 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](#)

# Acknowledgments

---

**SAFARI**

*SAFARI Research Group*

*safari.ethz.ch*

Think BIG, Aim HIGH!

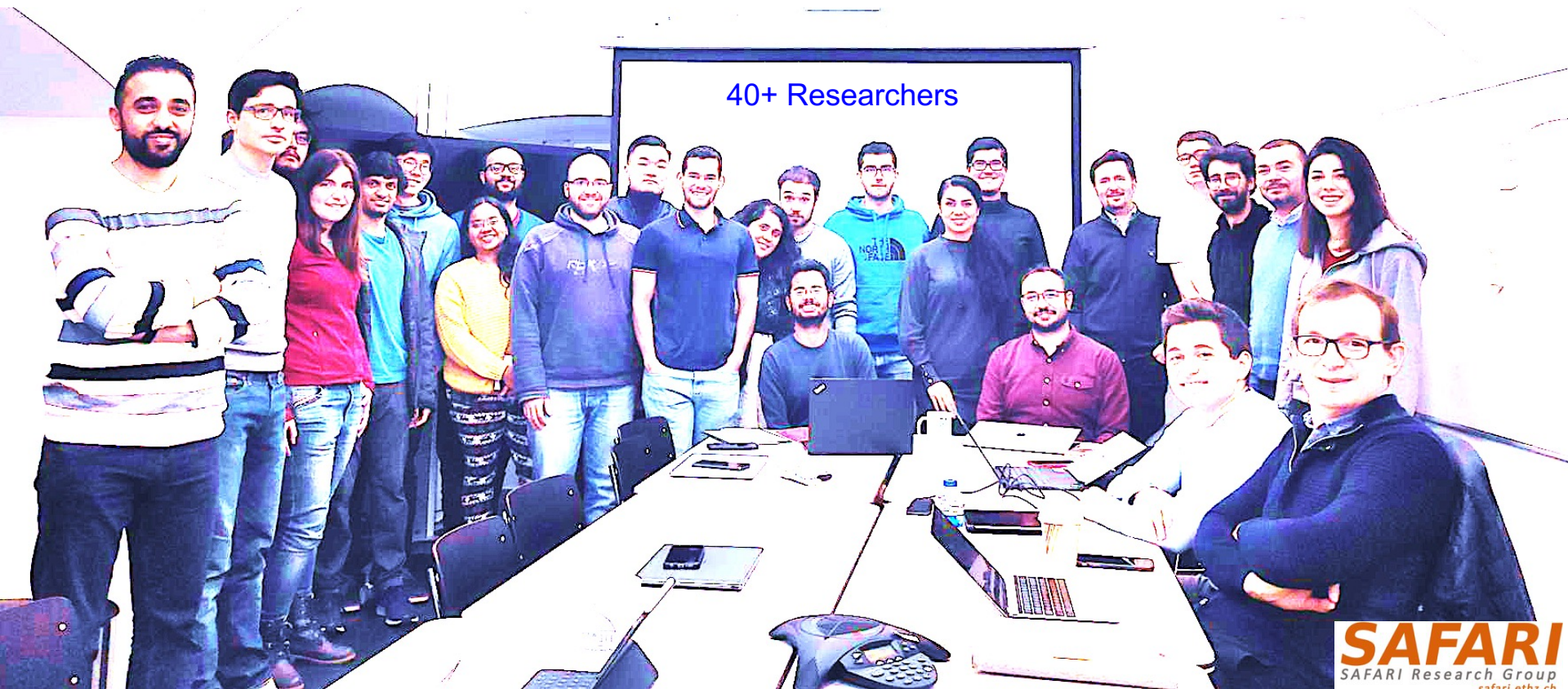
<https://safari.ethz.ch>

---

# Onur Mutlu's SAFARI Research Group

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

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



Think BIG, Aim HIGH!

**SAFARI**

<https://safari.ethz.ch>



# SAFARI Newsletter December 2021 Edition

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

**SAFARI**  
SAFARI Research Group

*Think Big, Aim High*

**ETH** zürich



View in your browser  
December 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/>

# 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.

📍 ETH Zurich and Carnegie Mellon U... 🔗 <https://safari.ethz.ch/> ✉ [omutlu@gmail.com](mailto:omutlu@gmail.com)

[Overview](#) [Repositories 71](#) [Projects](#) [Packages](#) [Teams 1](#) [People 44](#) [Settings](#)

### Pinned

Customize pins

 **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++ ☆ 311 🍷 161

 **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 ☆ 53 🍷 21

 **DAMOV** Public

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 processin...

🔴 C++ ☆ 26 🍷 4

 **SneakySnake** Public

SneakySnake🐍 is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude...

⬛ VHDL ☆ 41 🍷 8

 **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++ ☆ 146 🍷 93

 **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](http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf).

⬛ C ☆ 189 🍷 41

# 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 "Onur Mutlu Lectures" channel, which has 26.9K subscribers. The video has 1,286 views and premiered on Aug 9, 2022. The video player shows a thumbnail of a presentation slide with the text "In-Memory Processing ISVLSI 2022 Special Session" and "IEEE Computer Society Annual Symposium on VLSI". The video is currently at 0:04 / 3:36:35. The video player also shows a small inset video of a speaker in the top right corner.

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

# Comp Arch (Fall 2021)

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

- Computer Architecture FS20: Course Webpage
- Computer Architecture FS20: Lecture Videos
- Digitaltechnik SS21: Course Webpage
- Digitaltechnik SS21: Lecture Videos
- Moodle
- HotCRP
- Verilog Practice Website (HDLBits)

## 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](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>

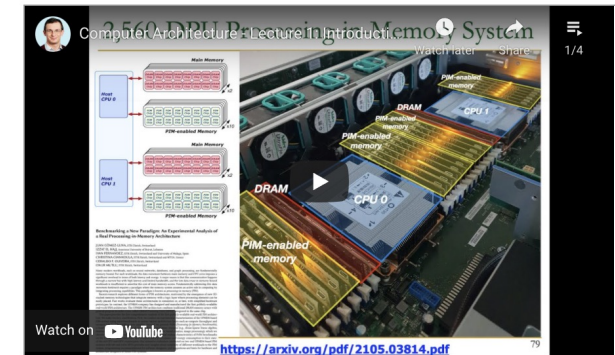
## Master's level course

- 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

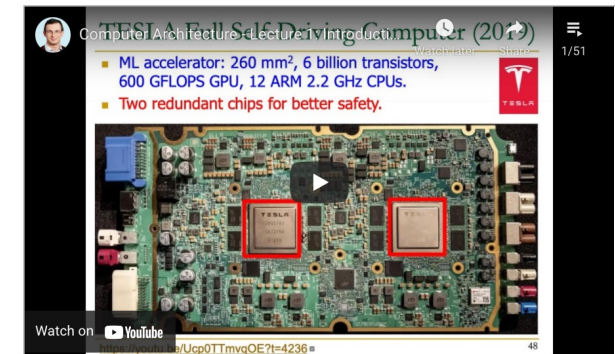
<https://www.youtube.com/onurmutlulectures>

## Lecture Video Playlist on YouTube

### Livestream Lecture Playlist



### Recorded Lecture Playlist



## Fall 2021 Lectures & Schedule

| Week | Date       | Livestream   | Lecture                                                                                      | Readings            | Lab       | HW       |
|------|------------|--------------|----------------------------------------------------------------------------------------------|---------------------|-----------|----------|
| W1   | 30.09 Thu. | YouTube Live | L1: Introduction and Basics<br><a href="#">PDF</a> <a href="#">PPT</a>                       | Required Mentioned  | Lab 1 Out | HW 0 Out |
|      | 01.10 Fri. | YouTube Live | L2: Trends, Tradeoffs and Design Fundamentals<br><a href="#">PDF</a> <a href="#">PPT</a>     | Required Mentioned  |           |          |
| W2   | 07.10 Thu. | YouTube Live | L3a: Memory Systems: Challenges and Opportunities<br><a href="#">PDF</a> <a href="#">PPT</a> | Described Suggested |           | HW 1 Out |
|      |            |              | L3b: Course Info & Logistics<br><a href="#">PDF</a> <a href="#">PPT</a>                      |                     |           |          |
|      |            |              | L3c: Memory Performance Attacks<br><a href="#">PDF</a> <a href="#">PPT</a>                   | Described Suggested |           |          |
|      | 08.10 Fri. | YouTube Live | L4a: Memory Performance Attacks<br><a href="#">PDF</a> <a href="#">PPT</a>                   | Described Suggested | Lab 2 Out |          |
|      |            |              | L4b: Data Retention and Memory Refresh<br><a href="#">PDF</a> <a href="#">PPT</a>            | Described Suggested |           |          |
|      |            |              | L4c: RowHammer<br><a href="#">PDF</a> <a href="#">PPT</a>                                    | 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](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN)

## Bachelor's course

- 2<sup>nd</sup> semester at ETH Zurich
- Rigorous introduction into "How Computers Work"
- Digital Design/Logic
- Computer Architecture
- 10 FPGA Lab Assignments

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



Trace: - schedule

Home

Announcements

Materials

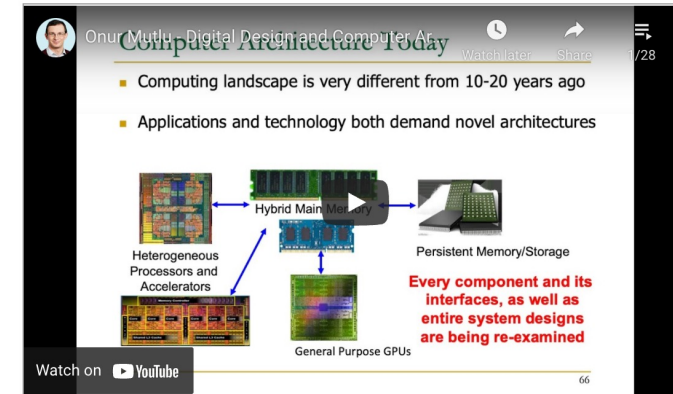
- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Optional HWs
- Labs
- Extra Assignments
- Exams
- Technical Docs

Resources

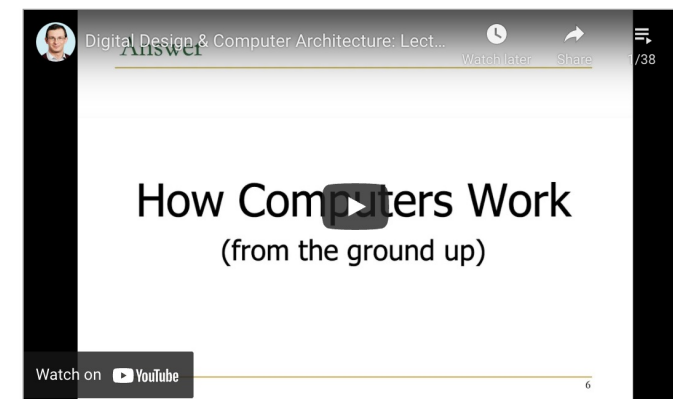
- Computer Architecture (CMU) SS15: Lecture Videos
- Computer Architecture (CMU) SS15: Course Website
- Digitaltechnik SS18: Lecture Videos
- Digitaltechnik SS18: Course Website
- Digitaltechnik SS19: Lecture Videos
- Digitaltechnik SS19: Course Website
- Digitaltechnik SS20: Lecture Videos
- Digitaltechnik SS20: Course Website
- Moodle

## Lecture Video Playlist on YouTube

Livestream Lecture Playlist



Recorded Lecture Playlist

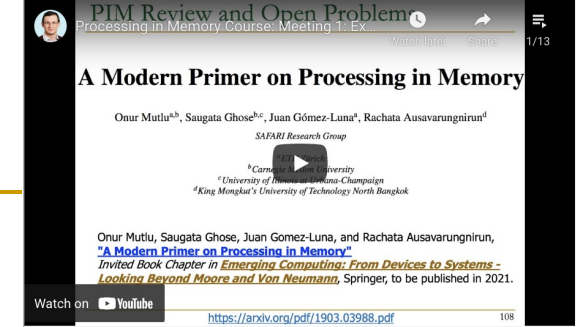


## Spring 2021 Lectures/Schedule

| Week | Date       | Livestream   | Lecture                                                           | Readings                           | Lab | HW |
|------|------------|--------------|-------------------------------------------------------------------|------------------------------------|-----|----|
| W1   | 25.02 Thu. | YouTube Live | L1: Introduction and Basics<br>Q2A (PDF) Q2B (PPT)                | Required<br>Suggested<br>Mentioned |     |    |
|      |            |              | L2a: Tradeoffs, Metrics, Mindset<br>Q2A (PDF) Q2B (PPT)           | Required                           |     |    |
|      | 26.02 Fri. | YouTube Live | L2b: Mysteries in Computer Architecture<br>Q2A (PDF) Q2B (PPT)    | Required<br>Mentioned              |     |    |
| W2   | 04.03 Thu. | YouTube Live | L3a: Mysteries in Computer Architecture II<br>Q2A (PDF) Q2B (PPT) | Required<br>Suggested<br>Mentioned |     |    |



# PIM Course (Spring 2022)



## Spring 2022 Meetings/Schedule

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

## Spring 2022 Edition:

- [https://safari.ethz.ch/projects\\_and\\_seminars/spring2022/doku.php?id=processing\\_in\\_memory](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory)

## Youtube Livestream:

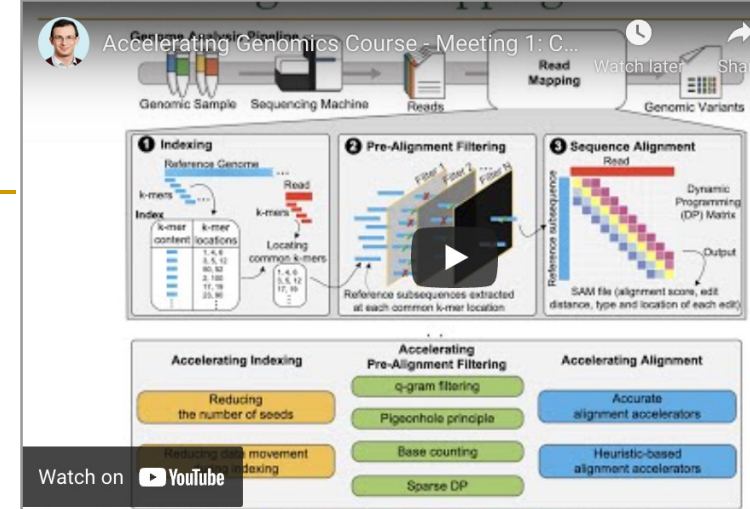
- <https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX>

## 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>

# Genomics (Spring 2022)



## Spring 2022 Meetings/Schedule

| Week | Date       | Livestream                       | Meeting                                                                                                                         | Learning Materials                          |
|------|------------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| W1   | 11.3 Fri.  | <a href="#">YouTube Live</a>     | <b>M1: P&amp;S Accelerating Genomics Course Introduction &amp; Project Proposals</b><br><a href="#">PDF</a> <a href="#">PPT</a> | Required Materials<br>Recommended Materials |
| W2   | 18.3 Fri.  | <a href="#">YouTube Live</a>     | <b>M2: Introduction to Sequencing</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                |                                             |
| W3   | 25.3 Fri.  | <a href="#">YouTube Premiere</a> | <b>M3: Read Mapping</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                              |                                             |
| W4   | 01.04 Fri. | <a href="#">YouTube Premiere</a> | <b>M4: GateKeeper</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                                |                                             |
| W5   | 08.04 Fri. | <a href="#">YouTube Premiere</a> | <b>M5: MAGNET &amp; Shouji</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                       |                                             |
| W6   | 15.4 Fri.  | <a href="#">YouTube Premiere</a> | <b>M6: SneakySnake</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                               |                                             |
| W7   | 29.4 Fri.  | <a href="#">YouTube Premiere</a> | <b>M7: GenStore</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                                  |                                             |
| W8   | 06.05 Fri. | <a href="#">YouTube Premiere</a> | <b>M8: GRIM-Filter</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                               |                                             |
| W9   | 13.05 Fri. | <a href="#">YouTube Premiere</a> | <b>M9: Genome Assembly</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                           |                                             |
| W10  | 20.05 Fri. | <a href="#">YouTube Live</a>     | <b>M10: Genomic Data Sharing Under Differential Privacy</b><br><a href="#">PDF</a> <a href="#">PPT</a>                          |                                             |
| W11  | 10.06 Fri. | <a href="#">YouTube Premiere</a> | <b>M11: Accelerating Genome Sequence Analysis</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                    |                                             |

## Spring 2022 Edition:

- [https://safari.ethz.ch/projects\\_and\\_seminars/spring2022/doku.php?id=bioinformatics](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics)

## Youtube Livestream:

- [https://www.youtube.com/watch?v=DEL5A\\_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU\\_Cxxjw-u18](https://www.youtube.com/watch?v=DEL5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18)

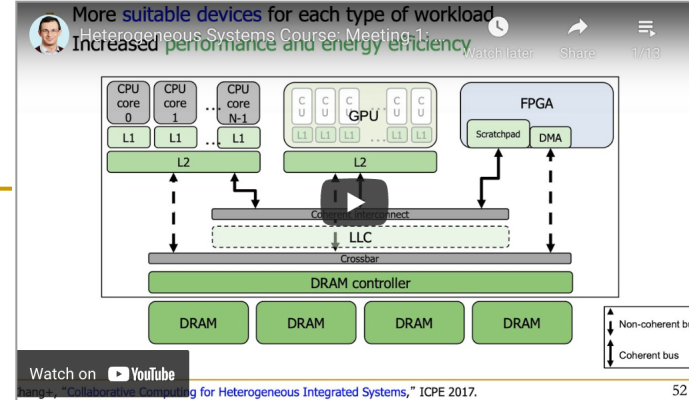
## Project course

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

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



# 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. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M1: P&amp;S Course Presentation</b><br><a href="#">PDF</a> <a href="#">PPT</a>                    | Required Materials<br>Recommended Materials | HW 0 Out    |
| W2   | 22.03 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M2: SIMD Processing and GPUs</b><br><a href="#">PDF</a> <a href="#">PPT</a>                       |                                             |             |
| W3   | 29.03 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M3: GPU Software Hierarchy</b><br><a href="#">PDF</a> <a href="#">PPT</a>                         |                                             |             |
| W4   | 05.04 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M4: GPU Memory Hierarchy</b><br><a href="#">PDF</a> <a href="#">PPT</a>                           |                                             |             |
| W5   | 12.04 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M5: GPU Performance Considerations</b><br><a href="#">PDF</a> <a href="#">PPT</a>                 |                                             |             |
| W6   | 19.04 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M6: Parallel Patterns: Reduction</b><br><a href="#">PDF</a> <a href="#">PPT</a>                   |                                             |             |
| W7   | 26.04 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M7: Parallel Patterns: Histogram</b><br><a href="#">PDF</a> <a href="#">PPT</a>                   |                                             |             |
| W8   | 03.05 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M8: Parallel Patterns: Convolution</b><br><a href="#">PDF</a> <a href="#">PPT</a>                 |                                             |             |
| W9   | 10.05 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M9: Parallel Patterns: Prefix Sum (Scan)</b><br><a href="#">PDF</a> <a href="#">PPT</a>           |                                             |             |
| W10  | 17.05 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M10: Parallel Patterns: Sparse Matrices</b><br><a href="#">PDF</a> <a href="#">PPT</a>            |                                             |             |
| W11  | 24.05 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M11: Parallel Patterns: Graph Search</b><br><a href="#">PDF</a> <a href="#">PPT</a>               |                                             |             |
| W12  | 01.06 Wed. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M12: Parallel Patterns: Merge Sort</b><br><a href="#">PDF</a> <a href="#">PPT</a>                 |                                             |             |
| W13  | 07.06 Tue. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M13: Dynamic Parallelism</b><br><a href="#">PDF</a> <a href="#">PPT</a>                           |                                             |             |
| W14  | 15.06 Wed. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M14: Collaborative Computing</b><br><a href="#">PDF</a> <a href="#">PPT</a>                       |                                             |             |
| W15  | 24.06 Fri. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M15: GPU Acceleration of Genome Sequence Alignment</b><br><a href="#">PDF</a> <a href="#">PPT</a> |                                             |             |
| W16  | 14.07 Thu. | <a href="#">YouTube</a> <a href="#">Premiere</a> | <b>M16: Accelerating Agent-based Simulations</b><br><a href="#">PDF</a> <a href="#">ODP</a>          |                                             |             |

# HW/SW Co-Design (Spring 2022)

## Spring 2022 Edition:

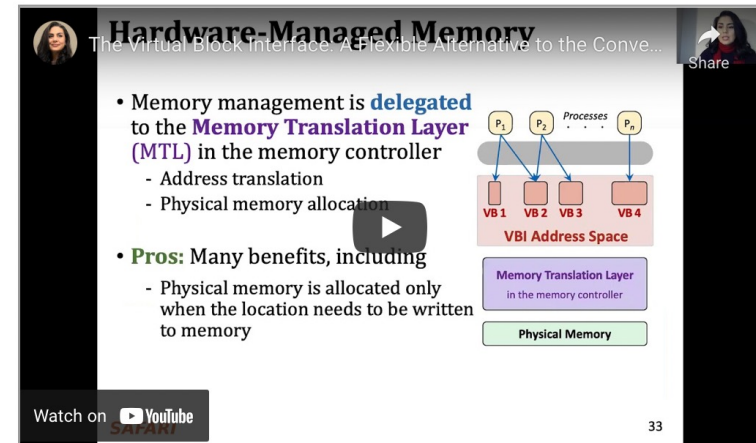
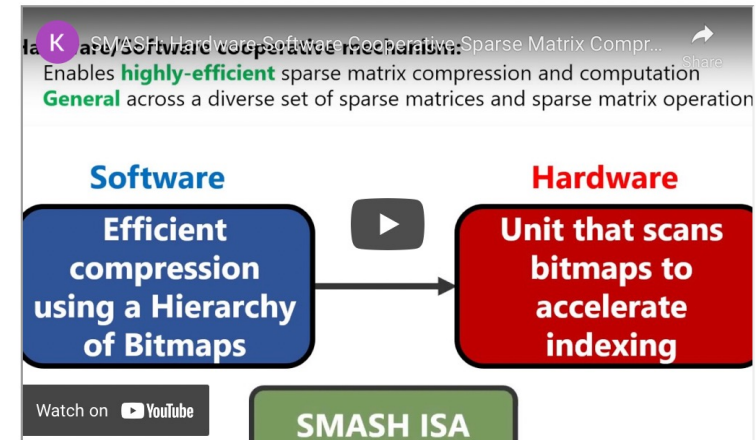
- https://safari.ethz.ch/projects\_and\_seminars/spring2022/doku.php?id=hw\_sw\_co\_design

## Youtube Livestream:

- https://youtube.com/playlist?list=PL5Q2soXY2Zi8nH7un3ghD2nutKWWDk-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 | YouTube Live | Intro to HW/SW Co-Design<br>(PPTX) (PDF) | Required  | HW 0 Out    |
| W1   | 23.03 |              | Project selection                        | Required  |             |
| W2   | 30.03 | YouTube Live | Virtual Memory (I)<br>(PPTX) (PDF)       |           |             |
| W3   | 13.04 | YouTube Live | Virtual Memory (II)<br>(PPTX) (PDF)      |           |             |

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

# Solid-State Drives (Spring 2022)

## ■ Spring 2022 Edition:

- [https://safari.ethz.ch/projects\\_and\\_seminars/spring2022/doku.php?id=modern\\_sds](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=modern_sds)

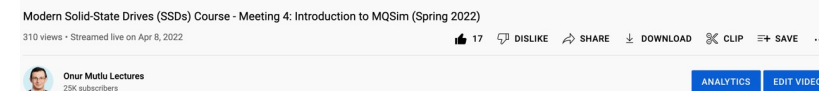
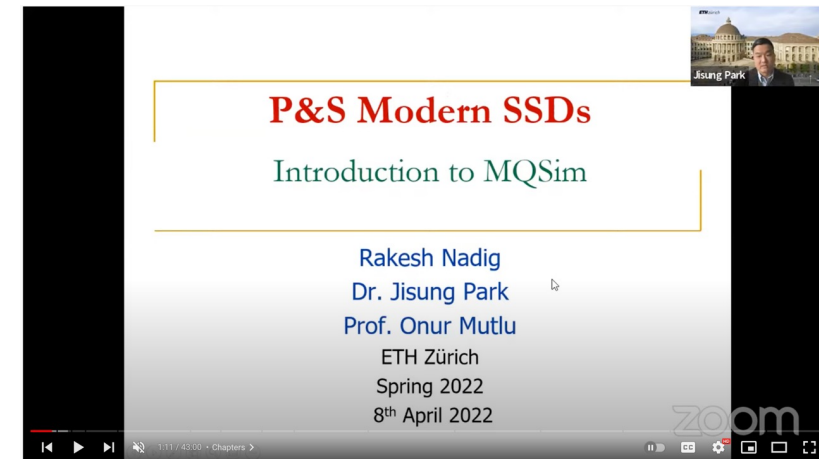
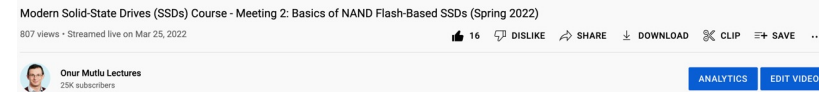
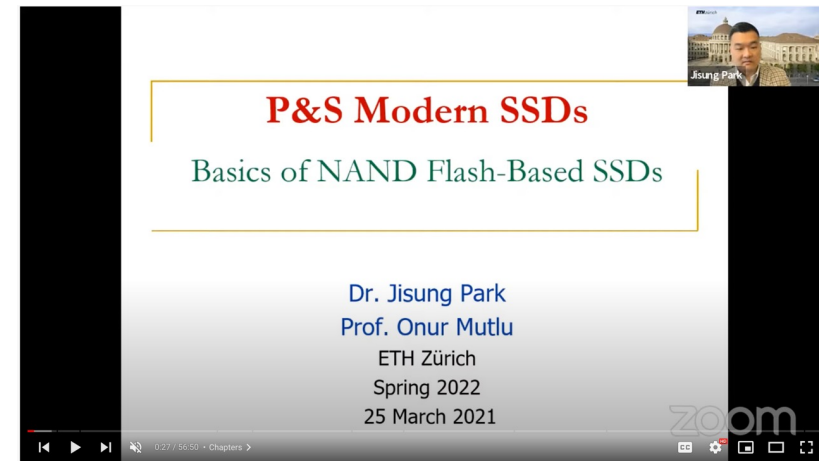
## ■ Youtube Livestream:

- <https://www.youtube.com/watch?v=q4rm71DsY4&list=PL5Q2soXY2Zi8vabcse1kL22DEcgMI2RAq>

## ■ 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>



# RowHammer & DRAM Exploration (Fall 2022)

## Fall 2022 Edition:

- [https://safari.ethz.ch/projects\\_and\\_seminars/fall2022/doku.php?id=softmc](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](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](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

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

### Lecture Video Playlist on YouTube

Lecture Playlist



### 2022 Meetings/Schedule (Tentative)

| Week | Date       | Livestream | Meeting                                                               | Learning Materials                          | Assignments |
|------|------------|------------|-----------------------------------------------------------------------|---------------------------------------------|-------------|
| W0   | 23.02 Wed. |            | <b>P&amp;S SoftMC Tutorial</b>                                        | SoftMC Tutorial Slides<br>(PDF)  (PPT)      |             |
| W1   | 08.03 Tue. |            | <b>M1: Logistics &amp; Intro to DRAM and SoftMC</b><br>(PDF)  (PPT)   | Required Materials<br>Recommended Materials | HW0         |
| W2   | 15.03 Tue. |            | <b>M2: Revisiting RowHammer</b><br>(PDF)  (PPT)                       | (Paper PDF)                                 |             |
| W3   | 22.03 Tue. |            | <b>M3: Uncovering in-DRAM TRR &amp; TRRespass</b><br>(PDF)  (PPT)     |                                             |             |
| W4   | 29.03 Tue. |            | <b>M4: Deeper Look Into RowHammer's Sensitivities</b><br>(PDF)  (PPT) |                                             |             |
| W5   | 05.04 Tue. |            | <b>M5: QUAC-TRNG</b><br>(PDF)  (PPT)                                  |                                             |             |
| W6   | 12.04 Tue. |            | <b>M6: PiDRAM</b><br>(PDF)  (PPT)                                     |                                             |             |

# Exploration of Emerging Memory Systems (Fall 2022)

## Fall 2022 Edition:

- [https://safari.ethz.ch/projects\\_and\\_seminars/fall2022/doku.php?id=ramulator](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](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](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. | <a href="#">YouTube</a> <a href="#">Video</a> | <b>M1: Logistics &amp; Intro to Simulating Memory Systems Using Ramulator</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT) |                    | HW0         |
| W2   | 16.03 Fri. | <a href="#">YouTube</a> <a href="#">Video</a> | <b>M2: Tutorial on Using Ramulator</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                                        |                    |             |
| W3   | 25.02 Fri. | <a href="#">YouTube</a> <a href="#">Video</a> | <b>M3: BlockHammer</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                                                        |                    |             |
| W4   | 01.04 Fri. | <a href="#">YouTube</a> <a href="#">Video</a> | <b>M4: CLR-DRAM</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                                                           |                    |             |
| W5   | 08.04 Fri. | <a href="#">YouTube</a> <a href="#">Video</a> | <b>M5: SIMDRAM</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                                                            |                    |             |
| W6   | 29.04 Fri. | <a href="#">YouTube</a> <a href="#">Video</a> | <b>M6: DAMOV</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                                                              |                    |             |
| W7   | 06.05 Fri. | <a href="#">YouTube</a> <a href="#">Video</a> | <b>M7: Synchron</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                                                           |                    |             |

# Memory-Centric Computing

Onur Mutlu

[omutlu@gmail.com](mailto:omutlu@gmail.com)

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

24 October 2022

Sonova Internal Workshop

**SAFARI**

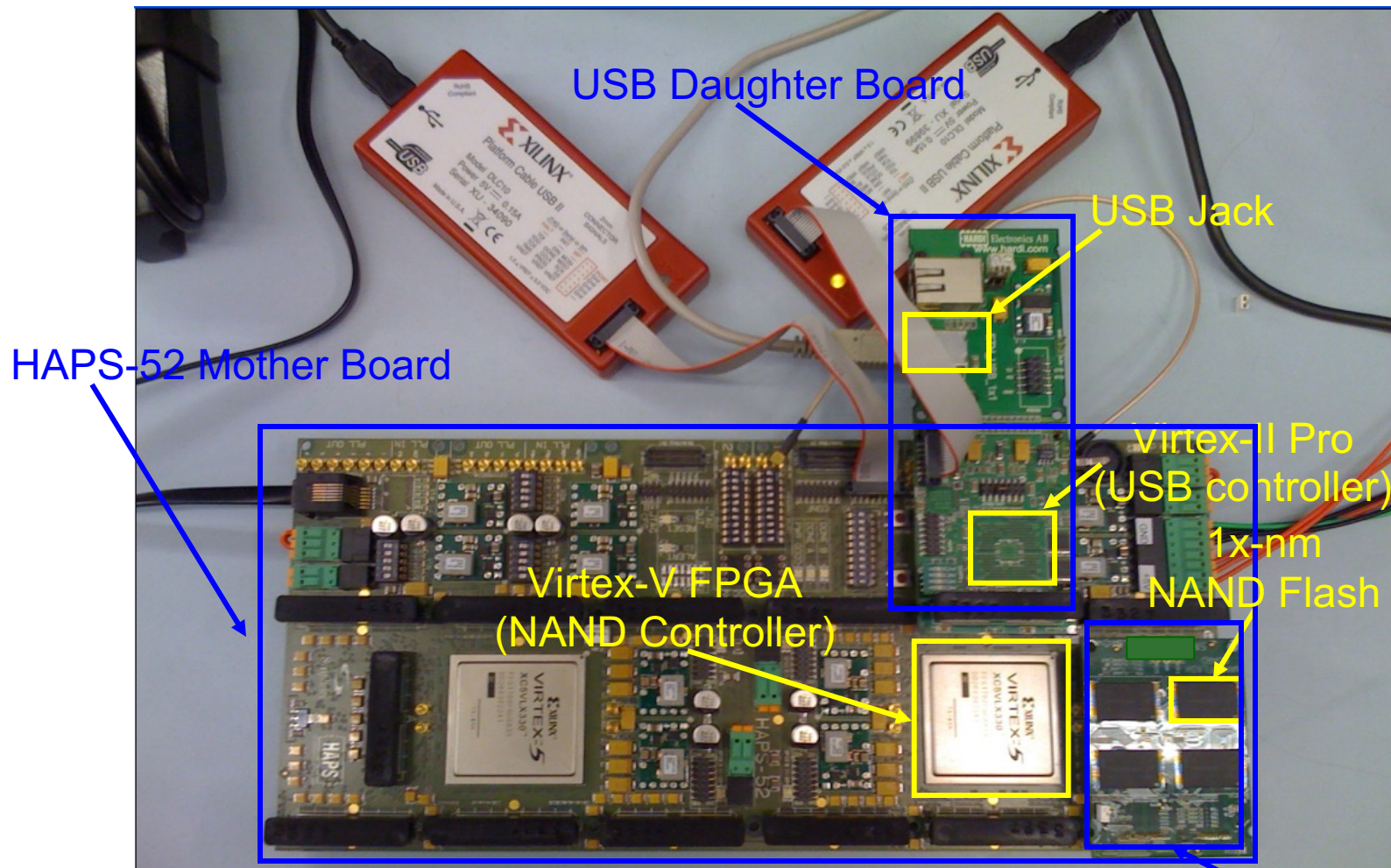
**ETH** zürich

**Carnegie Mellon**

# Backup Slides



# Aside: Intelligent Controller for NAND Flash



[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018, SIGMETRICS 2018]

NAND Daughter Board

Cai+, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives," Proc. IEEE 2017.



*Proceedings of the IEEE, Sept. 2017*



## Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

*This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD's reliability and lifetime.*

By YU CAI, SAUGATA GHOSE, ERICH F. HARATSCH, YIXIN LUO, AND ONUR MUTLU

<https://arxiv.org/pdf/1706.08642>

# System Desirables

---

- Self-managing, independent components
- All components intelligent & equal partners
- Easy collaboration & partitioning across all components
- Fine-grained communication of data & tasks
- Seamless caching & translation & protection anywhere
- Execution anywhere without rewriting code
- Flexibility, adaptability, self-optimization

**Open minds**

# SAFARI Research Group

# SAFARI Newsletter April 2020 Edition

---

- <https://safari.ethz.ch/safari-newsletter-april-2020/>



[View in your browser](#)

*Think Big, Aim High*



Dear SAFARI friends,

2019 and the first three months of 2020 have been very positive eventful times for SAFARI.



# SAFARI Newsletter January 2021 Edition

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



Newsletter  
January 2021

*Think Big, Aim High, and  
Have a Wonderful 2021!*



Dear SAFARI friends,

Happy New Year! We are excited to share our group highlights with you in this second edition of the SAFARI newsletter (You can find the first edition from April 2020 [here](#)). 2020 has

# SAFARI Newsletter December 2021 Edition

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

**SAFARI**  
SAFARI Research Group

*Think Big, Aim High*

**ETH** zürich



View in your browser  
December 2021





# A Talk on Impactful Research & Teaching



The video player shows a presentation slide with the following content:

Applying to Grad School  
& Doing Impactful Research

Onur Mutlu  
[omutlu@gmail.com](mailto:omutlu@gmail.com)  
<https://people.inf.ethz.ch/omutlu>  
13 June 2020  
Undergraduate Architecture Mentoring Workshop @ ISCA 2021

Logos for SAFARI, ETH zürich, and Carnegie Mellon are displayed at the bottom of the slide.

Below the video player, the video title is "Arch. Mentoring Workshop @ISCA'21 - Applying to Grad School & Doing Impactful Research - Onur Mutlu". It has 1,563 views and premiered on Jun 16, 2021. The channel is "Onur Mutlu Lectures" with 17.2K subscribers. The video description mentions a panel talk at the Undergraduate Architecture Mentoring Workshop at ISCA 2021, with a link to <https://sites.google.com/wisc.edu/uar...>

# An Interview on Computing Futures



Interview with Onur Mutlu @ ISCA 2019 on computing research & education (after Maurice Wilkes Award)

6,749 views • Oct 19, 2019

👍 195 🗨️ 0 ➦ SHARE ➦ SAVE ...



**Onur Mutlu Lectures**  
19.1K subscribers

ANALYTICS

EDIT VIDEO

# Latest Longer & Detailed Tutorial on PIM

---

- Onur Mutlu,  
**"Memory-Centric Computing"**  
*Education Class at Embedded Systems Week (ESWEEK),*  
Virtual, 9 October 2021.  
[Slides (pptx) (pdf)]  
[Abstract (pdf)]  
[Talk Video (2 hours, including Q&A)]  
[Invited Paper at DATE 2021]  
["A Modern Primer on Processing in Memory" paper]

**<https://www.youtube.com/watch?v=N1Ac1ov1JOM>**

# Memory-Centric Computing

Onur Mutlu

[omutlu@gmail.com](mailto:omutlu@gmail.com)

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

9 October 2021

ESWEEK Education Class

**SAFARI**

**ETH** zürich

Carnegie Mellon



1:08 / 2:00:10



Embedded Systems Week (ESWEEK) 2021 Lecture - Memory-Centric Computing - Onur Mutlu - 9 October 2021

509 views • Premiered Dec 6, 2021

28 DISLIKE SHARE SAVE ...



Onur Mutlu Lectures  
20.7K subscribers

<https://www.youtube.com/watch?v=N1Ac1ov1JOM>

ANALYTICS

EDIT VIDEO

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

307

# Detailed Lectures on PIM (I)

---

- **Computer Architecture, Fall 2020, Lecture 6**
  - **Computation in Memory** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=oGcZAGwfEUE&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=12>
- **Computer Architecture, Fall 2020, Lecture 7**
  - **Near-Data Processing** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=j2GIigqn1Qw&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=13>
- **Computer Architecture, Fall 2020, Lecture 11a**
  - **Memory Controllers** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=TeG773OgiMQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=20>
- **Computer Architecture, Fall 2020, Lecture 12d**
  - **Real Processing-in-DRAM with UPMEM** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=25>

# Detailed Lectures on PIM (II)

---

- **Computer Architecture, Fall 2020, Lecture 15**
  - **Emerging Memory Technologies** (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=AIE1rD9G\\_YU&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=28](https://www.youtube.com/watch?v=AIE1rD9G_YU&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=28)
- **Computer Architecture, Fall 2020, Lecture 16a**
  - **Opportunities & Challenges of Emerging Memory Technologies** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=29>
- **Computer Architecture, Fall 2020, Guest Lecture**
  - **In-Memory Computing: Memory Devices & Applications** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=wNmQqHiEZnk&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=41>



# Comp Arch (Current)

## Fall 2021 Edition:

- <https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule>

## Youtube Livestream:

- [https://www.youtube.com/watch?v=4yfkM\\_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILKTOF](https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILKTOF)

## 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


Computer Architecture - Fall 2021

Recent Changes
Media Manager
Schedule

Trace: readings · start · schedule

Home

Announcements

Materials

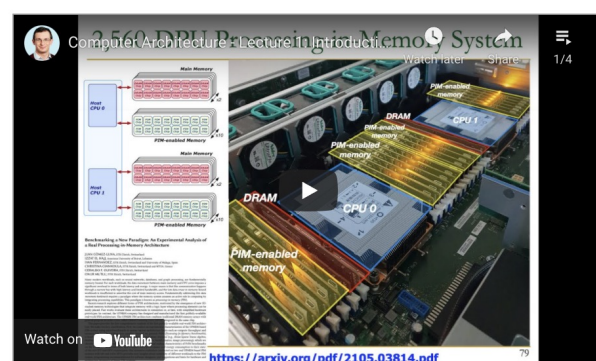
Resources

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

- 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

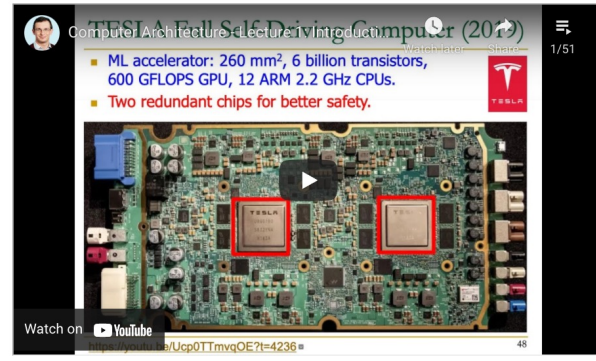


2.560 DRU Processing in Memory System

Watch on YouTube

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

Recorded Lecture Playlist



TESLA Full Self-Driving Computer (2020)

- ML accelerator: 260 mm<sup>2</sup>, 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.

Watch on YouTube

Fall 2021 Lectures & Schedule

| Week | Date       | Livestream   | Lecture                                                                                      | Readings            | Lab       | HW       |
|------|------------|--------------|----------------------------------------------------------------------------------------------|---------------------|-----------|----------|
| W1   | 30.09 Thu. | YouTube Live | L1: Introduction and Basics<br><a href="#">PDF</a> <a href="#">PPT</a>                       | Required Mentioned  | Lab 1 Out | HW 0 Out |
|      | 01.10 Fri. | YouTube Live | L2: Trends, Tradeoffs and Design Fundamentals<br><a href="#">PDF</a> <a href="#">PPT</a>     | Required Mentioned  |           |          |
| W2   | 07.10 Thu. | YouTube Live | L3a: Memory Systems: Challenges and Opportunities<br><a href="#">PDF</a> <a href="#">PPT</a> | Described Suggested |           | HW 1 Out |
|      |            |              | L3b: Course Info & Logistics<br><a href="#">PDF</a> <a href="#">PPT</a>                      |                     |           |          |
|      |            |              | L3c: Memory Performance Attacks<br><a href="#">PDF</a> <a href="#">PPT</a>                   | Described Suggested |           |          |
|      | 08.10 Fri. | YouTube Live | L4a: Memory Performance Attacks<br><a href="#">PDF</a> <a href="#">PPT</a>                   | Described Suggested | Lab 2 Out |          |
|      |            |              | L4b: Data Retention and Memory Refresh<br><a href="#">PDF</a> <a href="#">PPT</a>            | Described Suggested |           |          |
|      |            |              | L4c: RowHammer<br><a href="#">PDF</a> <a href="#">PPT</a>                                    | Described Suggested |           |          |



# PIM Course (Current)

## Fall 2021 Edition:

- https://safari.ethz.ch/projects\_and\_seminars/fall2021/doku.php?id=processing\_in\_memory

## Youtube Livestream:

- 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

PIM Review and Open Problems  
Processing in Memory Course: Meeting 1: Ex...

Watch later Share 1/10

### A Modern Primer on Processing in Memory

Onur Mutlu<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirun<sup>d</sup>

SAFARI Research Group

<sup>a</sup>ETH Zürich  
<sup>b</sup>Carnegie Mellon University  
<sup>c</sup>University of Illinois at Urbana-Champaign  
<sup>d</sup>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.

Watch on YouTube

<https://arxiv.org/pdf/1903.03988.pdf> 108

## Fall 2021 Meetings/Schedule

| Week | Date       | Livestream                                   | Meeting                                                                                                            | Learning Materials                          | Assignments |
|------|------------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------|---------------------------------------------|-------------|
| W1   | 05.10 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M1: P&amp;S PIM Course Presentation</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                  | Required Materials<br>Recommended Materials | HW 0 Out    |
| W2   | 12.10 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M2: Real-World PIM Architectures</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                     |                                             |             |
| W3   | 19.10 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M3: Real-World PIM Architectures II</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                  |                                             |             |
| W4   | 26.10 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M4: Real-World PIM Architectures III</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                 |                                             |             |
| W5   | 02.11 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M5: Real-World PIM Architectures IV</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                  |                                             |             |
| W6   | 09.11 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M6: End-to-End Framework for Processing-using-Memory</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT) |                                             |             |
| W7   | 16.11 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M7: How to Evaluate Data Movement Bottlenecks</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)        |                                             |             |
| W8   | 23.11 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M8: Programming PIM Architectures</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                    |                                             |             |
| W9   | 30.11 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M9: Benchmarking and Workload Suitability on PIM</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)     |                                             |             |
| W10  | 07.12 Tue. | <a href="#">YouTube</a> <a href="#">Live</a> | <b>M10: Bit-Serial SIMD Processing using DRAM</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)           |                                             |             |



Onur Mutlu Lectures

16.9K subscribers

CUSTOMIZE CHANNEL

MANAGE VIDEOS

HOME

VIDEOS

PLAYLISTS

COMMUNITY

CHANNELS

ABOUT



Popular uploads ▶ PLAY ALL

**How Computers Work (from the ground up)**

1:33:25

**Computer Architecture - Lecture 1: Introduction and...**

2:37:30

**Processing in DRAM Engine**

2:24:11

**ML accelerator: 260 mm², 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs**

2:39:20

**Design of Digital Circuits - Lecture 1: Introduction and Basics**

1:22:29

**Fewest Misses ≠ Best Performance**

2:33:20

### First Course in Computer Architecture & Digital Design 2021-2013

**Computing landscape is very different from 10-20 years ago**

28

**How Computers Work (from the ground up)**

38

**Design of Digital Circuits - Lecture 1: Introduction and Basics**

35

**ML accelerator: 260 mm², 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs**

28

**Bahnhof Stadelhofen**

23

**Spring 2015 -- Computer Architecture Lectures --...**

39

### Advanced Computer Architecture Courses 2020-2012

**ML accelerator: 260 mm², 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs**

51

**Processing in DRAM Engine**

39

**Look backward (to the past)**

38

**ML accelerator: 260 mm², 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs**

28

**Fall 2015 - 740 Computer Architecture**

14

**Unsuspected Slowness**

60

### Special Courses on Memory Systems

**Computer Architecture - Lecture 14b: Flash, Memory and Solid-State Drives**

22

**Champéry Winter School 2020 - Memory Systems and Memory...**

4

**Perugia NIPS Summer School 2019**

6

**SAMOS Tutorial 2019 - Memory Systems**

5

**It's the Memory, Stupid**

12

**ACACES 2018 Lectures -- Memory Systems and Memory...**

6

# Data-Driven Architectures

# Corollaries: Architectures Today ...

---

- Architectures are **terrible at dealing with data**
  - ❑ Designed to mainly store and move data vs. to compute
  - ❑ They are **processor-centric** as opposed to **data-centric**
- Architectures are **terrible at taking advantage of vast amounts of data** (and metadata) available to them
  - ❑ Designed to make simple decisions, ignoring lots of data
  - ❑ They make **human-driven decisions** vs. **data-driven** decisions
- Architectures are **terrible at knowing and exploiting different properties of application data**
  - ❑ Designed to treat all data as the same
  - ❑ They make **component-aware decisions** vs. **data-aware**

# Exploiting Data to Design Intelligent 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

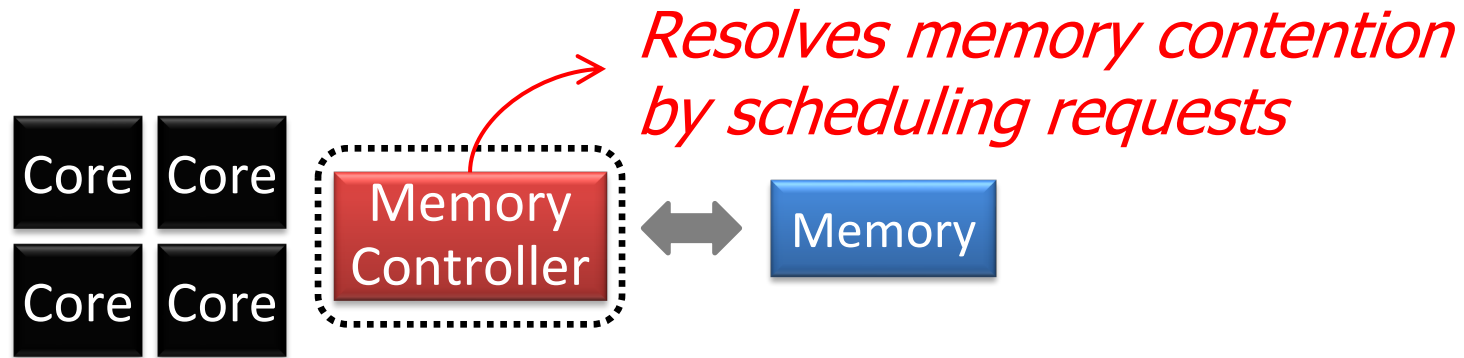
**How do we start?**



# Self-Optimizing Memory Controllers

# Memory Controller

---



How to schedule requests to maximize system performance?

# Why are Memory Controllers Difficult to Design?

---

- Need to obey **DRAM timing constraints** for correctness
  - There are many (50+) timing constraints in DRAM
  - tWTR: Minimum number of cycles to wait before issuing a read command after a write command is issued
  - tRC: Minimum number of cycles between the issuing of two consecutive activate commands to the same bank
  - ...
- Need to **keep track of many resources** to prevent conflicts
  - Channels, banks, ranks, data bus, address bus, row buffers, ...
- Need to handle **DRAM refresh**
- Need to **manage power** consumption
- Need to **optimize performance & QoS** (in the presence of constraints)
  - Reordering is not simple
  - Fairness and QoS needs complicates the scheduling problem
- ...

# Many Memory Timing Constraints

---

| Latency                               | Symbol    | DRAM cycles | Latency                                        | Symbol    | DRAM cycles |
|---------------------------------------|-----------|-------------|------------------------------------------------|-----------|-------------|
| Precharge                             | $t_{RP}$  | 11          | Activate to read/write                         | $t_{RCD}$ | 11          |
| Read column address strobe            | $CL$      | 11          | Write column address strobe                    | $CWL$     | 8           |
| Additive                              | $AL$      | 0           | Activate to activate                           | $t_{RC}$  | 39          |
| Activate to precharge                 | $t_{RAS}$ | 28          | Read to precharge                              | $t_{RTP}$ | 6           |
| Burst length                          | $t_{BL}$  | 4           | Column address strobe to column address strobe | $t_{CCD}$ | 4           |
| Activate to activate (different bank) | $t_{RRD}$ | 6           | Four activate windows                          | $t_{FAW}$ | 24          |
| Write to read                         | $t_{WTR}$ | 6           | Write recovery                                 | $t_{WR}$  | 12          |

Table 4. DDR3 1600 DRAM timing specifications

- From Lee et al., “[DRAM-Aware Last-Level Cache Writeback: Reducing Write-Caused Interference in Memory Systems](#),” HPS Technical Report, April 2010.

# Many Memory Timing Constraints

- Kim et al., "A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM," ISCA 2012.
- Lee et al., "Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture," HPCA 2013.

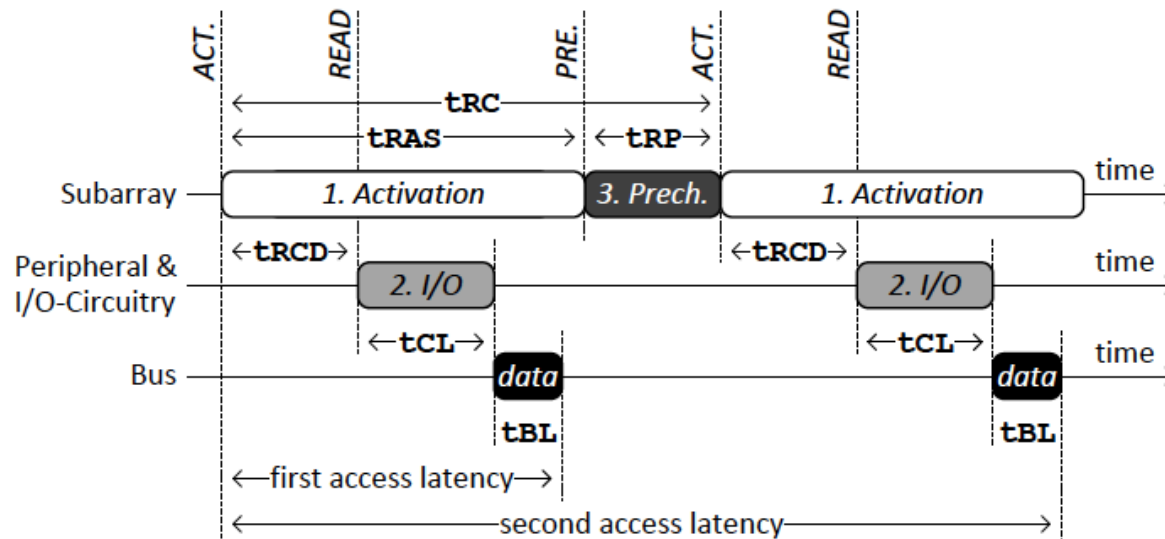


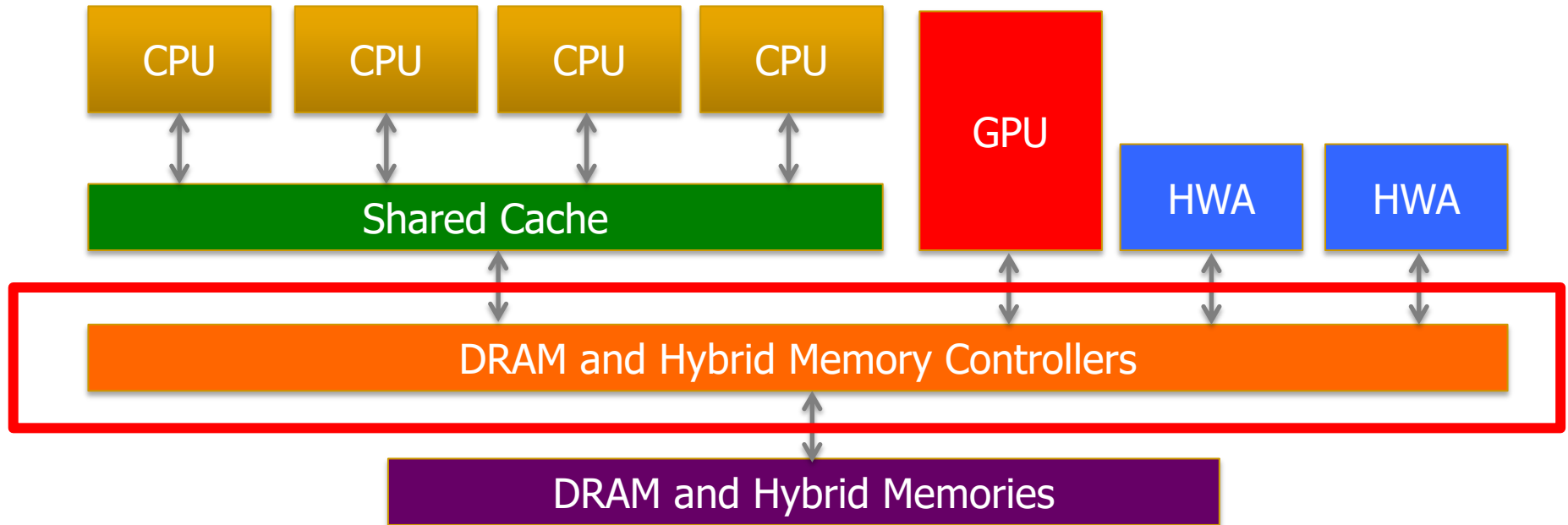
Figure 5. Three Phases of DRAM Access

Table 2. Timing Constraints (DDR3-1066) [43]

| Phase | Commands     | Name              | Value   |
|-------|--------------|-------------------|---------|
| 1     | ACT → READ   | tRCD              | 15ns    |
|       | ACT → WRITE  |                   |         |
|       | ACT → PRE    | tRAS              | 37.5ns  |
| 2     | READ → data  | tCL               | 15ns    |
|       | WRITE → data | tCWL              | 11.25ns |
|       | data burst   | tBL               | 7.5ns   |
| 3     | PRE → ACT    | tRP               | 15ns    |
| 1 & 3 | ACT → ACT    | tRC<br>(tRAS+tRP) | 52.5ns  |

# Memory Controller Design Is Becoming More Difficult

---



- Heterogeneous agents: CPUs, GPUs, and HWAs
- Main memory interference between CPUs, GPUs, HWAs
- Many timing constraints for various memory types
- Many goals at the same time: performance, fairness, QoS, energy efficiency, ...

# Reality and Dream

---

- **Reality**: It difficult to design a policy that maximizes performance, QoS, energy-efficiency, ...
  - ❑ Too many things to think about
  - ❑ Continuously changing workload and system behavior
- **Dream**: Wouldn't it be nice if the DRAM controller automatically found a good scheduling policy on its own?



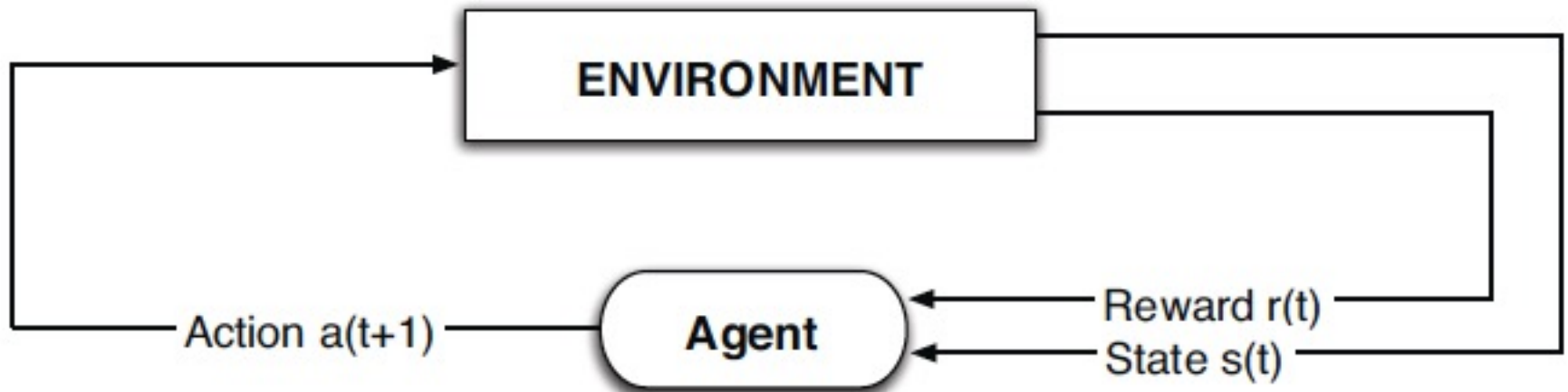
# Self-Optimizing DRAM Controllers

---

- Problem: DRAM controllers are difficult to design
  - It is difficult for human designers to design a policy that can adapt itself very well to different workloads and different system conditions
- Idea: A memory controller that adapts its scheduling policy to workload behavior and system conditions using machine learning.
- Observation: Reinforcement learning maps nicely to memory control.
- Design: Memory controller is a reinforcement learning agent
  - It dynamically and continuously learns and employs the best scheduling policy to maximize long-term performance.

# Self-Optimizing DRAM Controllers

---

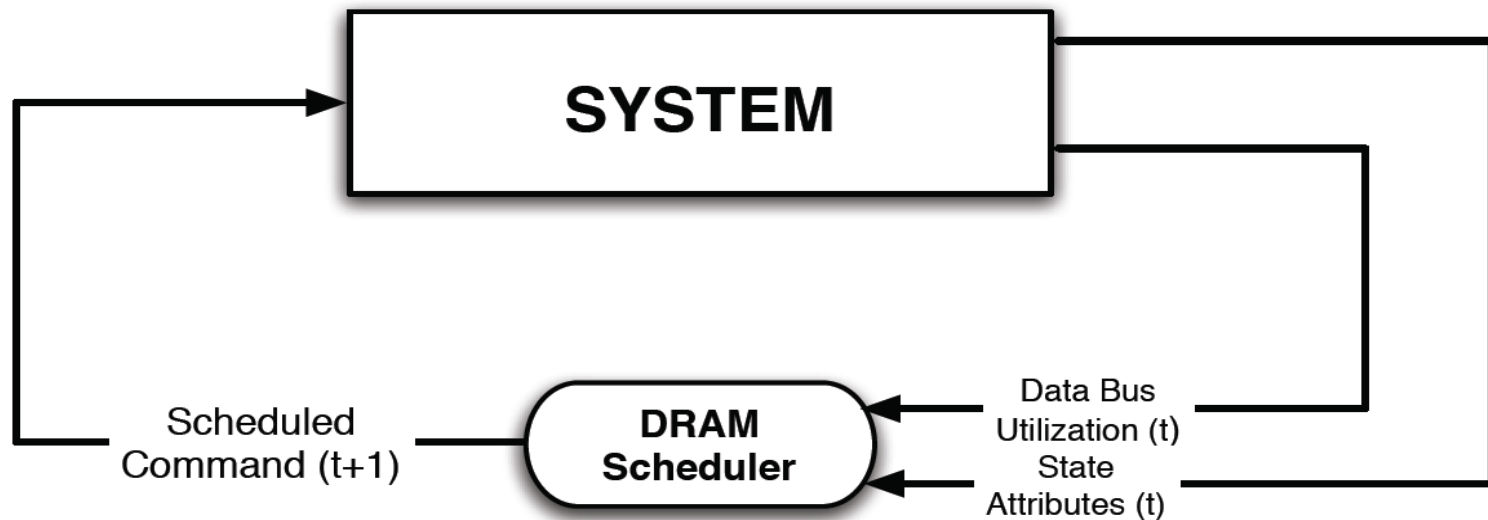


Goal: Learn to choose actions to maximize  $r_0 + \gamma r_1 + \gamma^2 r_2 + \dots$  ( $0 \leq \gamma < 1$ )

**Figure 2:** (a) Intelligent agent based on reinforcement learning principles;

# Self-Optimizing DRAM Controllers

- Dynamically adapt the memory scheduling policy via interaction with the system at runtime
  - Associate system states and actions (commands) with long term reward values: **each action at a given state leads to a learned reward**
  - **Schedule command with highest estimated long-term reward value in each state**
  - **Continuously update reward values for  $\langle \text{state}, \text{action} \rangle$  pairs based on feedback from system**



# Self-Optimizing DRAM 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.*

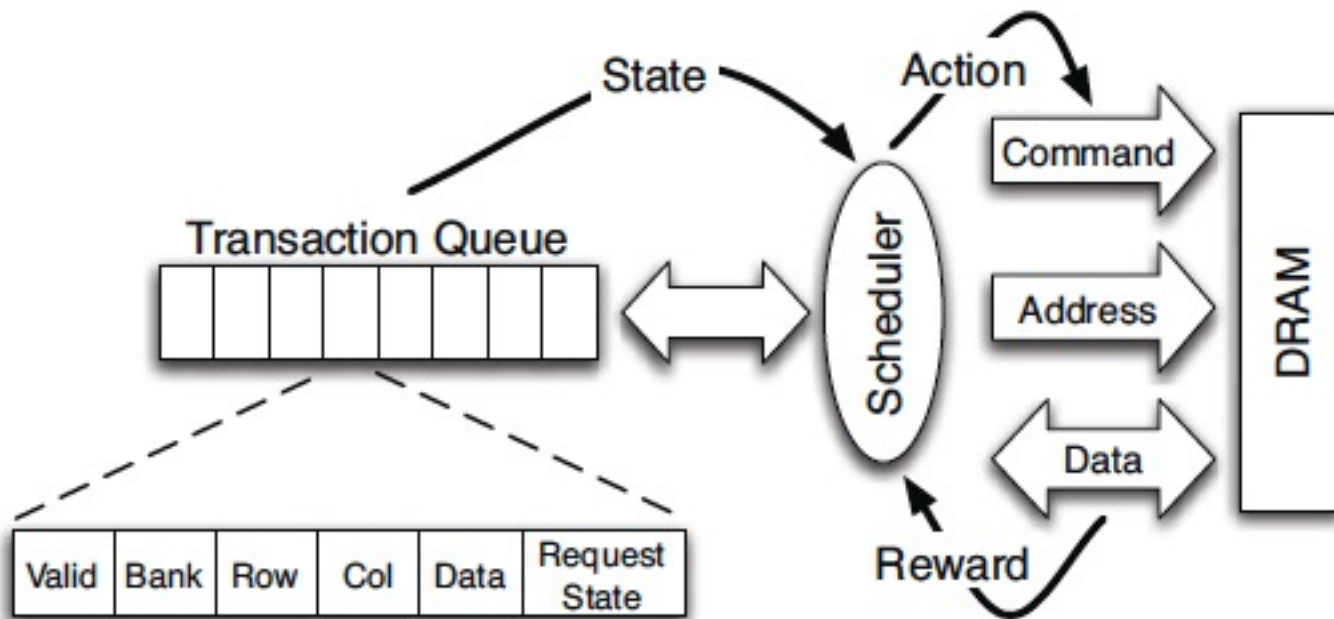


Figure 4: High-level overview of an RL-based scheduler.

# States, Actions, Rewards

---

## ❖ Reward function

- +1 for scheduling Read and Write commands
- 0 at all other times

Goal is to maximize long-term data bus utilization

## ❖ State attributes

- Number of reads, writes, and load misses in transaction queue
- Number of pending writes and ROB heads waiting for referenced row
- Request's relative ROB order

## ❖ Actions

- Activate
- Write
- Read - load miss
- Read - store miss
- Precharge - pending
- Precharge - preemptive
- NOP

# Performance Results

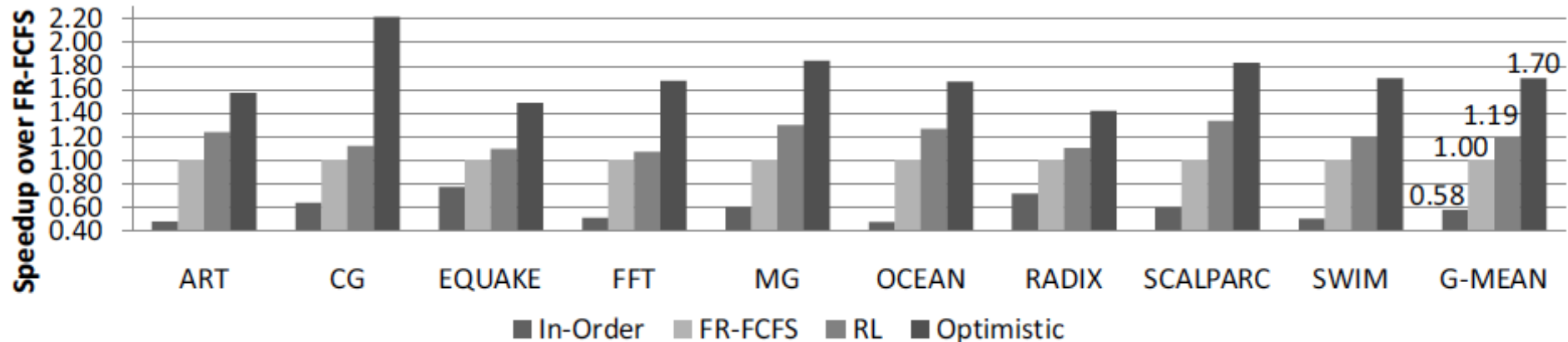


Figure 7: Performance comparison of in-order, FR-FCFS, RL-based, and optimistic memory controllers

**Large, robust performance improvements over many human-designed policies**

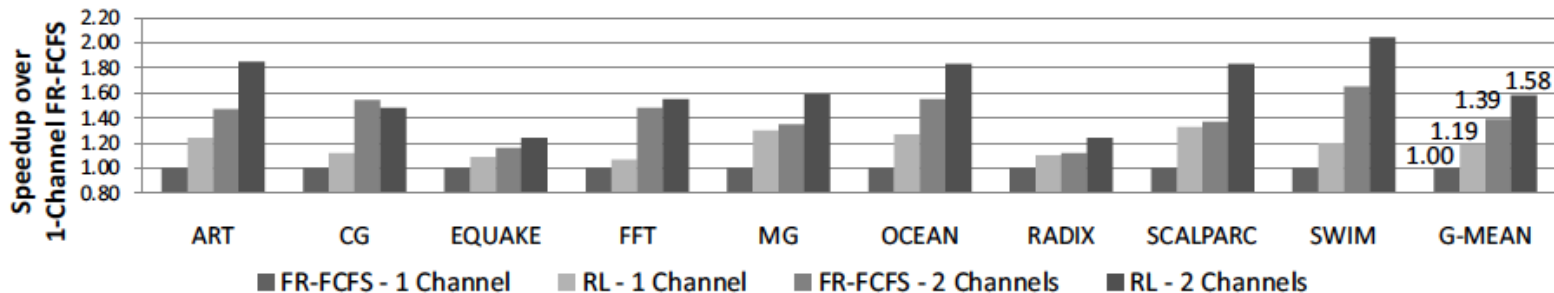


Figure 15: Performance comparison of FR-FCFS and RL-based memory controllers on systems with 6.4GB/s and 12.8GB/s peak DRAM bandwidth

# Self Optimizing DRAM Controllers

---

- + Continuous learning in the presence of changing environment

- + Reduced designer burden in finding a good scheduling policy.

Designer specifies:

- 1) What system variables might be useful

- 2) What target to optimize, but not how to optimize it

- How to specify different objectives? (e.g., fairness, QoS, ...)

- Hardware complexity?

- Design mindset and flow



# More on Self-Optimizing DRAM 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.

## Self-Optimizing Memory Controllers: A Reinforcement Learning Approach

Engin İpek<sup>1,2</sup>   Onur Mutlu<sup>2</sup>   José F. Martínez<sup>1</sup>   Rich Caruana<sup>1</sup>

<sup>1</sup>Cornell University, Ithaca, NY 14850 USA

<sup>2</sup>Microsoft 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](#)]

## Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera<sup>1</sup>    Konstantinos Kanellopoulos<sup>1</sup>    Anant V. Nori<sup>2</sup>    Taha Shahroodi<sup>3,1</sup>  
Sreenivas Subramoney<sup>2</sup>    Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Processor Architecture Research Labs, Intel Labs

<sup>3</sup>TU Delft



# Pythia

## A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera, Konstantinos Kanellopoulos, Anant V. Nori,  
Taha Shahroodi, Sreenivas Subramoney, Onur Mutlu

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



# Executive Summary

- **Background:** Prefetchers predict addresses of future memory requests by associating memory access patterns with program context (called **feature**)
- **Problem:** Three key shortcomings of prior prefetchers:
  - Predict mainly using a **single program feature**
  - Lack **inherent system awareness** (e.g., memory bandwidth usage)
  - Lack **in-silicon customizability**
- **Goal:** Design a prefetching framework that:
  - Learns from **multiple features** and **inherent system-level feedback**
  - Can be **customized in silicon** to use different features and/or prefetching objectives
- **Contribution:** Pythia, which formulates prefetching as reinforcement learning problem
  - Takes **adaptive** prefetch decisions using multiple features and system-level feedback
  - Can be **customized in silicon** for target workloads via simple configuration registers
  - Proposes a **realistic and practical** implementation of RL algorithm in hardware
- **Key Results:**
  - Evaluated using a wide range of workloads from SPEC CPU, PARSEC, Ligra, Cloudsuite
  - Outperforms best prefetcher (in 1-core config.) by **3.4%, 7.7% and 17%** in 1/4/bw-constrained cores
  - Up to **7.8% more performance** over basic Pythia across Ligra workloads via simple customization

# Key Shortcomings in Prior Prefetchers

---

- We observe **three key shortcomings** that significantly limit performance benefits of prior prefetchers

**1** Predict mainly using a **single program feature**

**2** Lack inherent **system awareness**

**3** Lack **in-silicon customizability**

# Our Goal

---

A **prefetching framework** that can:

1. Learn to prefetch using **multiple features** and **inherent system-level feedback** information
2. Be **easily customized in silicon** to use different features and/or change prefetcher's objectives

# Our Proposal

---



## Pythia

Formulates prefetching as a  
**reinforcement learning problem**



# Basics of Reinforcement Learning (RL)

---

- Algorithmic approach to learn to take an **action** in a given **situation** to maximize a numerical **reward**

Agent

Environment

- Agent stores **Q-values** for *every* state-action pair
  - **Expected return** for taking an action in a state
  - Given a state, selects action that provides **highest** Q-value

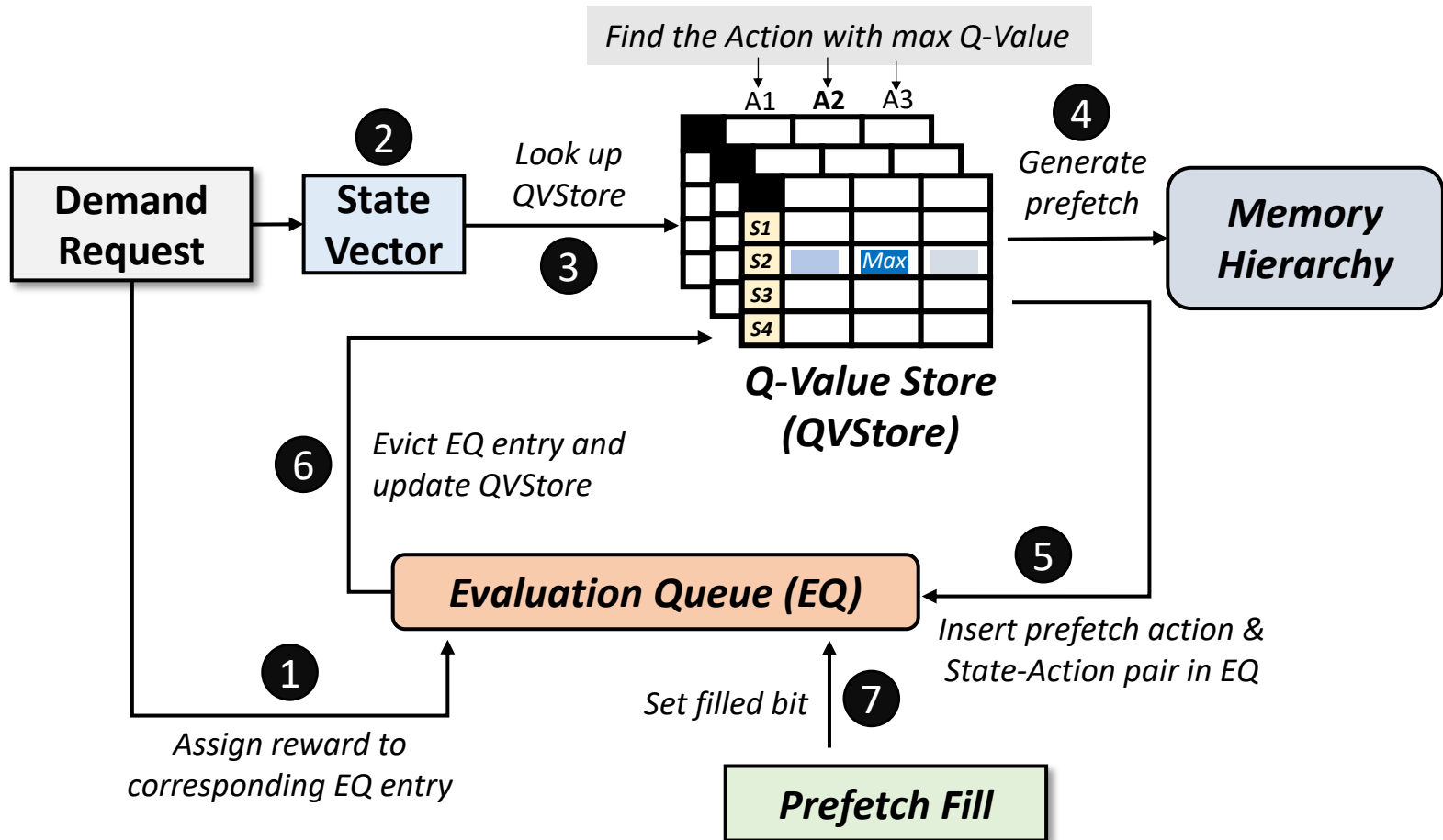
# Formulating Prefetching as RL

---

-----

# Pythia Overview

- **Q-Value Store**: Records Q-values for *all* state-action pairs
- **Evaluation Queue**: A FIFO queue of recently-taken actions



# Simulation Methodology

---

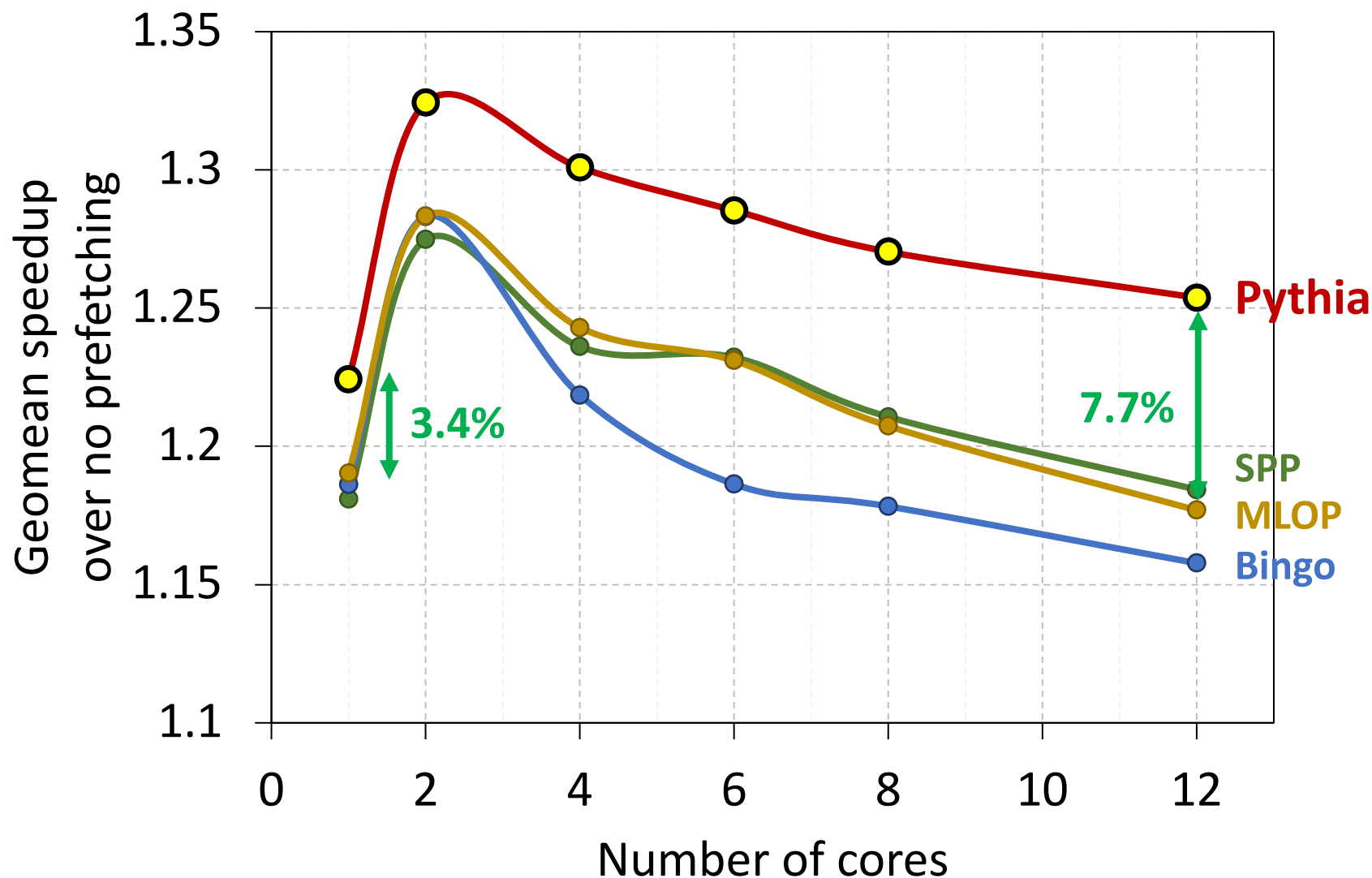
- **Champsim** [3] trace-driven simulator
- **150** single-core memory-intensive workload traces
  - SPEC CPU2006 and CPU2017
  - PARSEC 2.1
  - Ligra
  - Cloudsuite
- Homogeneous and heterogeneous multi-core mixes
- **Five** state-of-the-art prefetchers
  - SPP [Kim+, MICRO'16]
  - Bingo [Bakhshalipour+, HPCA'19]
  - MLOP [Shakerinava+, 3<sup>rd</sup> Prefetching Championship, 2019 ]
  - SPP+DSPatch [Bera+, MICRO'19]
  - SPP+PPF [Bhatia+, ISCA'20]

# Basic Pythia Configuration

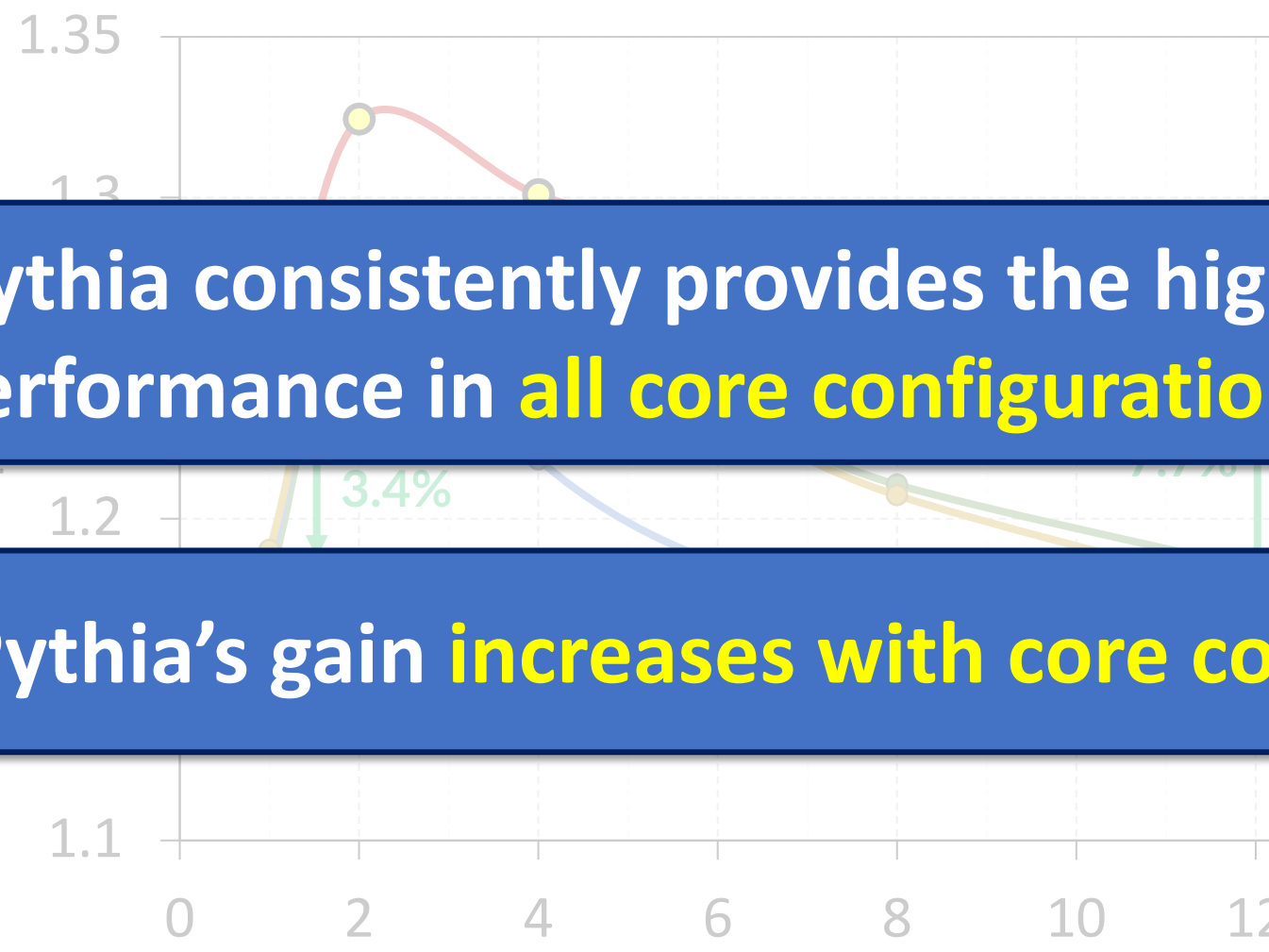
---

- Derived from **automatic design-space exploration**
- **State:** 2 features
  - PC+Delta
  - Sequence of last-4 deltas
- **Actions:** 16 prefetch offsets
  - Ranging between -6 to +32. Including 0.
- **Rewards:**
  - $R_{AT} = +20$ ;  $R_{AL} = +12$ ;  $R_{NP-H} = -2$ ;  $R_{NP-L} = -4$ ;
  - $R_{IN-H} = -14$ ;  $R_{IN-L} = -8$ ;  $R_{CL} = -12$

# Performance with Varying Core Count



# Performance with Varying Core Count



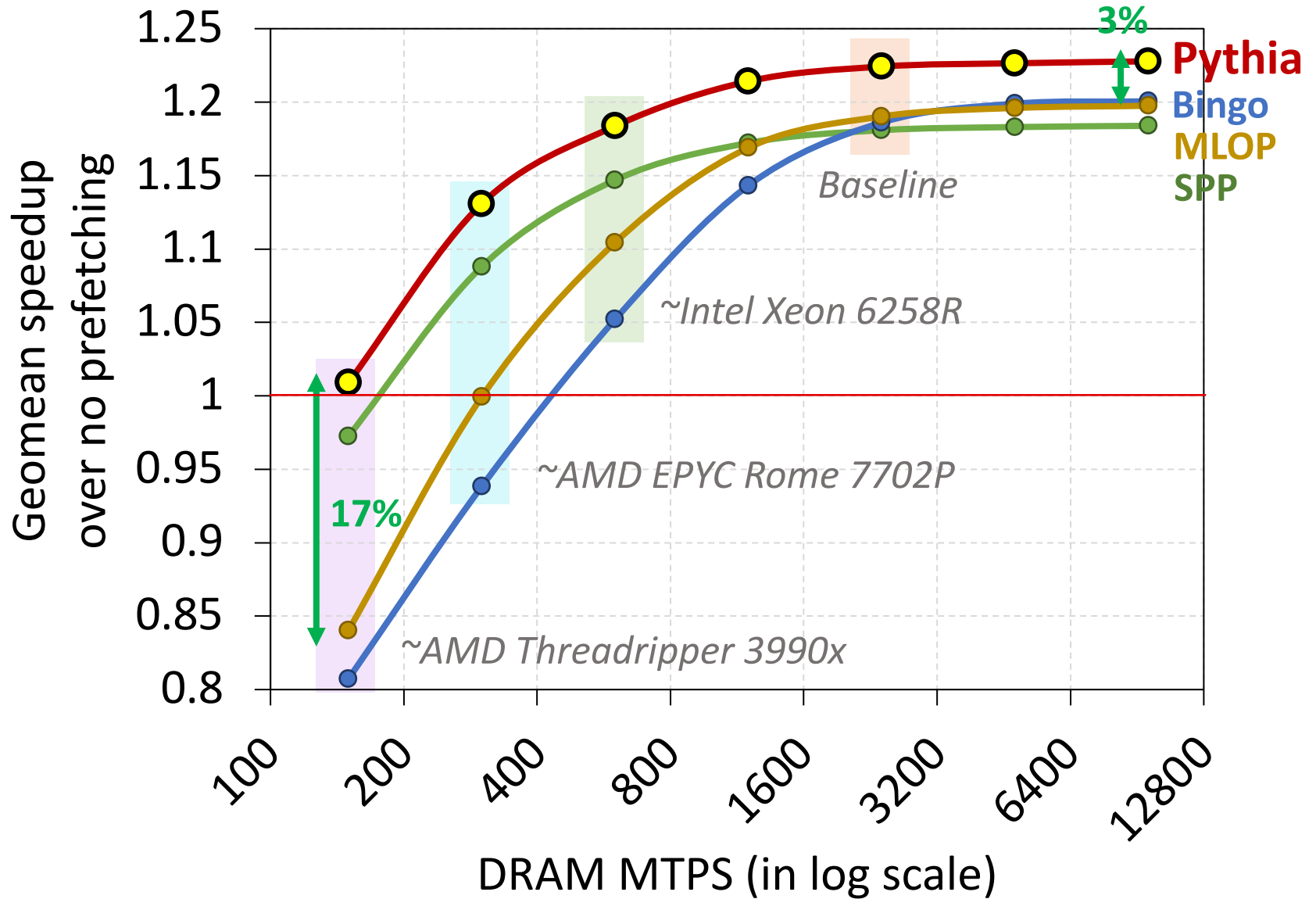
The graph shows performance on the y-axis (ranging from 1.1 to 1.35) against the number of cores on the x-axis (ranging from 0 to 12). Pythia (red line) starts at ~1.28 at 2 cores, peaks at ~1.32 at 4 cores, and then declines. Other models (blue, green, orange lines) show lower performance, with a green line showing a 3.4% gain at 2 cores and a 1.7% gain at 8 cores. A vertical green line at 12 cores is labeled 'SDD'.

1. Pythia consistently provides the highest performance in **all core configurations**

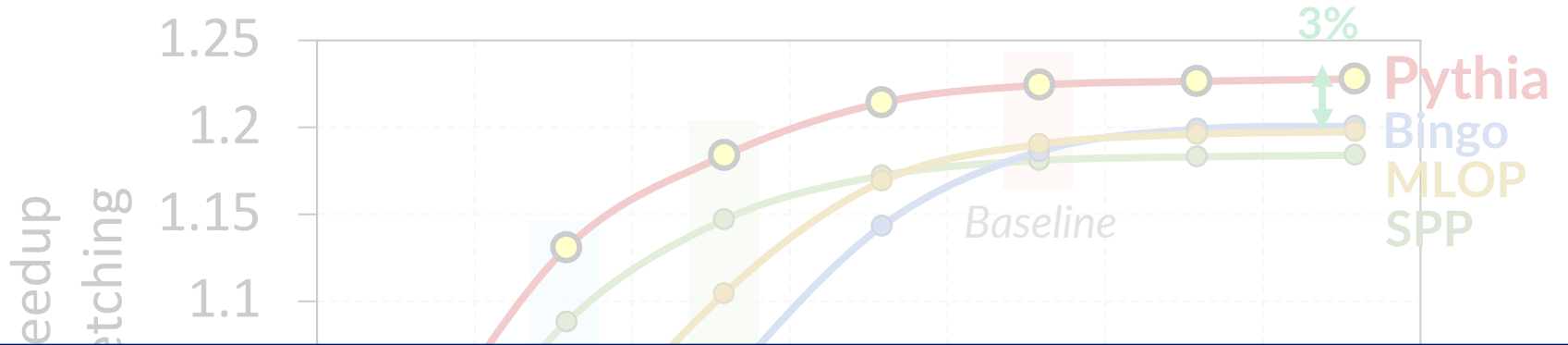
2. Pythia's gain **increases with core count**



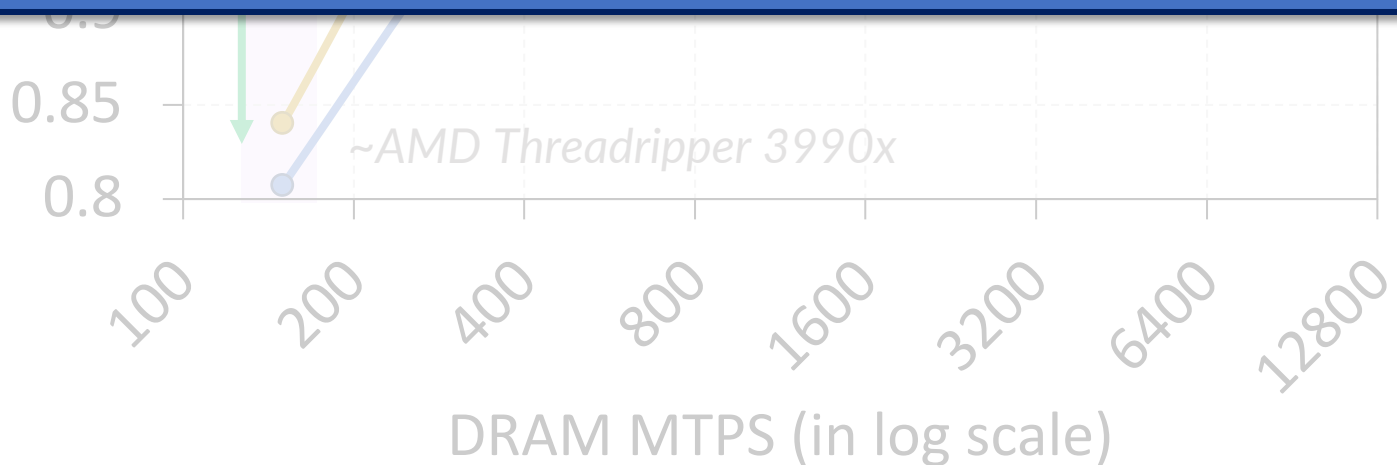
# Performance with Varying DRAM Bandwidth



# Performance with Varying DRAM Bandwidth



**Pythia outperforms prior best prefetchers for a wide range of DRAM bandwidth configurations**



# Pythia's Overhead

- **25.5 KB** of total metadata storage **per core**
  - Only simple tables
- We also model functionally-accurate Pythia with full complexity in **Chisel** [4] HDL



**1.03% area** overhead



**0.4% power** overhead



**Satisfies prediction latency**

*of a desktop-class 4-core Skylake processor (Xeon D2132IT, 60W)*

# More in the Paper

- Performance comparison with **unseen traces**
  - Pythia provides equally high performance benefits

• Comparison against **multi-level prefetchers**

## **Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning**

Rahul Bera<sup>1</sup>    Konstantinos Kanellopoulos<sup>1</sup>    Anant V. Nori<sup>2</sup>    Taha Shahroodi<sup>3,1</sup>  
Sreenivas Subramoney<sup>2</sup>    Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Processor Architecture Research Labs, Intel Labs

<sup>3</sup>TU Delft

<https://arxiv.org/pdf/2109.12021.pdf>

- **Performance sensitivity** towards different features and hyperparameter values

- Detailed single-core and four-core performance

# Pythia is Open Source



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

- MICRO'21 **artifact evaluated**
- **Champsim source** code + **Chisel** modeling code
- **All traces** used for evaluation

The screenshot shows the GitHub repository for CMU-SAFARI/Pythia. The repository is public and has 3 unwatchers, 7 stars, and 2 forks. The main navigation bar includes links to Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository is currently on the master branch, with 1 branch and 5 tags. The commit history shows a recent update to the README by rahulbera 2 days ago, with 38 commits in total. The file list includes directories for branch, config, experiments, inc, prefetcher, replacement, scripts, src, and tracer, as well as files for .gitignore, CITATION.cff, LICENSE, and LICENSE.champsim. The right sidebar contains an 'About' section describing Pythia as a customizable hardware prefetching framework, a link to the arXiv paper, and a list of related topics including machine-learning, reinforcement-learning, computer-architecture, prefetcher, microarchitecture, cache-replacement, branch-predictor, champsim-simulator, and champsim-tracer. Below the 'About' section are links to the README, View license, and Cite this repository. The 'Releases' section shows 5 releases.

| File             | Commit Message                                                | Commit Date  |
|------------------|---------------------------------------------------------------|--------------|
| branch           | Initial commit for MICRO'21 artifact evaluation               | 2 months ago |
| config           | Initial commit for MICRO'21 artifact evaluation               | 2 months ago |
| experiments      | Added chart visualization in Excel template                   | 2 months ago |
| inc              | Updated README                                                | 6 days ago   |
| prefetcher       | Initial commit for MICRO'21 artifact evaluation               | 2 months ago |
| replacement      | Initial commit for MICRO'21 artifact evaluation               | 2 months ago |
| scripts          | Added md5 checksum for all artifact traces to verify download | 2 months ago |
| src              | Initial commit for MICRO'21 artifact evaluation               | 2 months ago |
| tracer           | Initial commit for MICRO'21 artifact evaluation               | 2 months ago |
| .gitignore       | Initial commit for MICRO'21 artifact evaluation               | 2 months ago |
| CITATION.cff     | Added citation file                                           | 6 days ago   |
| LICENSE          | Updated LICENSE                                               | 2 months ago |
| LICENSE.champsim | Initial commit for MICRO'21 artifact evaluation               | 2 months ago |



# Pythia

## A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera, Konstantinos Kanellopoulos, Anant V. Nori,  
Taha Shahroodi, Sreenivas Subramoney, Onur Mutlu

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



# 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](#)]

## Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera<sup>1</sup>    Konstantinos Kanellopoulos<sup>1</sup>    Anant V. Nori<sup>2</sup>    Taha Shahroodi<sup>3,1</sup>  
Sreenivas Subramoney<sup>2</sup>    Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Processor Architecture Research Labs, Intel Labs

<sup>3</sup>TU Delft



# Self-Optimizing Hybrid Storage Systems

---

- To appear in ISCA 2022

## **Sibyl: Adaptive and Extensible Data Placement in Hybrid Storage Systems Using Online Reinforcement Learning**

Gagandeep Singh<sup>1</sup>   Rakesh Nadig<sup>1</sup>   Jisung Park<sup>1</sup>   Rahul Bera<sup>1</sup>   Nastaran Hajinazar<sup>1</sup>  
David Novo<sup>3</sup>   Juan Gómez-Luna<sup>1</sup>   Sander Stuijk<sup>2</sup>   Henk Corporaal<sup>2</sup>   Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Eindhoven University of Technology

<sup>3</sup>LIRMM, Univ. Montpellier, CNRS

<https://arxiv.org/pdf/2205.07394.pdf>

# Sibyl:

## Adaptive and Extensible Data Placement in Hybrid Storage Systems Using Online Reinforcement Learning

Gagandeep Singh, Rakesh Nadig, Jisung Park, Rahul Bera, Nastaran Hajinazar,  
David Novo, Juan Gómez-Luna, Onur Mutlu

# Executive Summary

**Background:** Hybrid storage systems (HSSs) complement different storage technologies to extend the overall capacity and reduce the system cost with minimal effect on the application performance

**Problem:** Accurately identify the performance-critical data of an application and placing it in the “best-fit” storage device. Three key shortcomings of prior data placement policies (heuristic-based and supervised learning-based) of hybrid storage systems:

- Lack of **adaptability**
- Lack of **device awareness (e.g., read/write latencies of each device)**
- Lack of **extensibility**

**Goal:** Develop a new, efficient, and high performance data-placement mechanism for hybrid storage systems that can:

- Dynamically derive an adaptive data-placement strategy by **continuously learning and adapting** to the **application and underlying device characteristics**
- **Easily extensible** to incorporate a wide range of hybrid storage configurations.

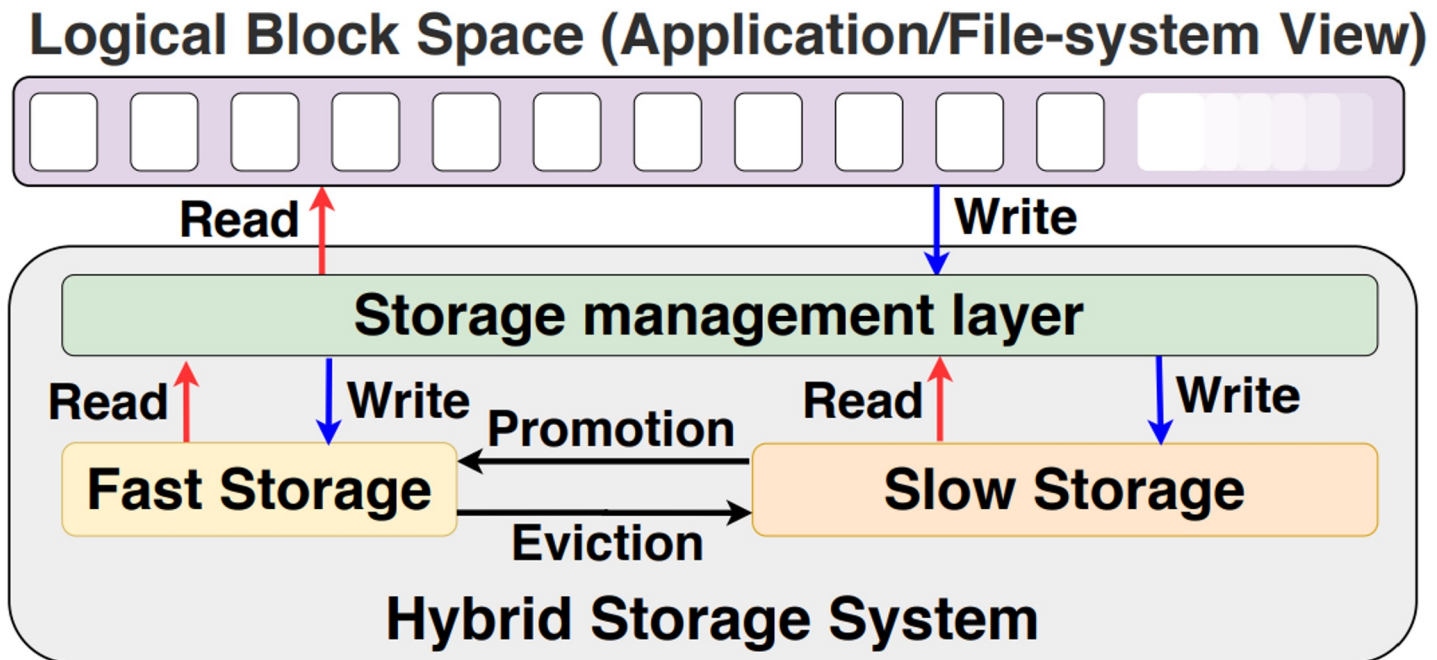
**Key Idea:** Sibyl, an online reinforcement learning-based self-optimizing mechanism for data placement that:

- **Dynamically learns** from past experiences and **continuously adapts** its policy to improve long-term performance by interacting with the hybrid storage system
- **Learns the asymmetry in the read/write latencies** present in modern hybrid storage devices while **taking into account** the **inherent characteristics of an application**

**Key Results:** Sibyl is evaluated on a real system with multiple device configurations

- Evaluated using a **wide range of workloads** from MSR Cambridge and Filebench
- In a performance (cost) optimized hybrid storage configuration, Sibyl provides up to **21.6% (19.9%)** performance improvement compared to prior data placement policies
- On a tri-hybrid storage system, Sibyl outperforms a heuristics-based policy by **23.9% -48.2%**
- Sibyl achieves **80%** performance of an oracle policy with storage overhead of **124.4 KiB**

# Hybrid Storage Systems



---

# Key Shortcomings of Prior Data Placement Techniques

We observe **three key shortcomings** that significantly limit performance benefits of data-placement techniques

Lack of **adaptability**

Lack of **device awareness**

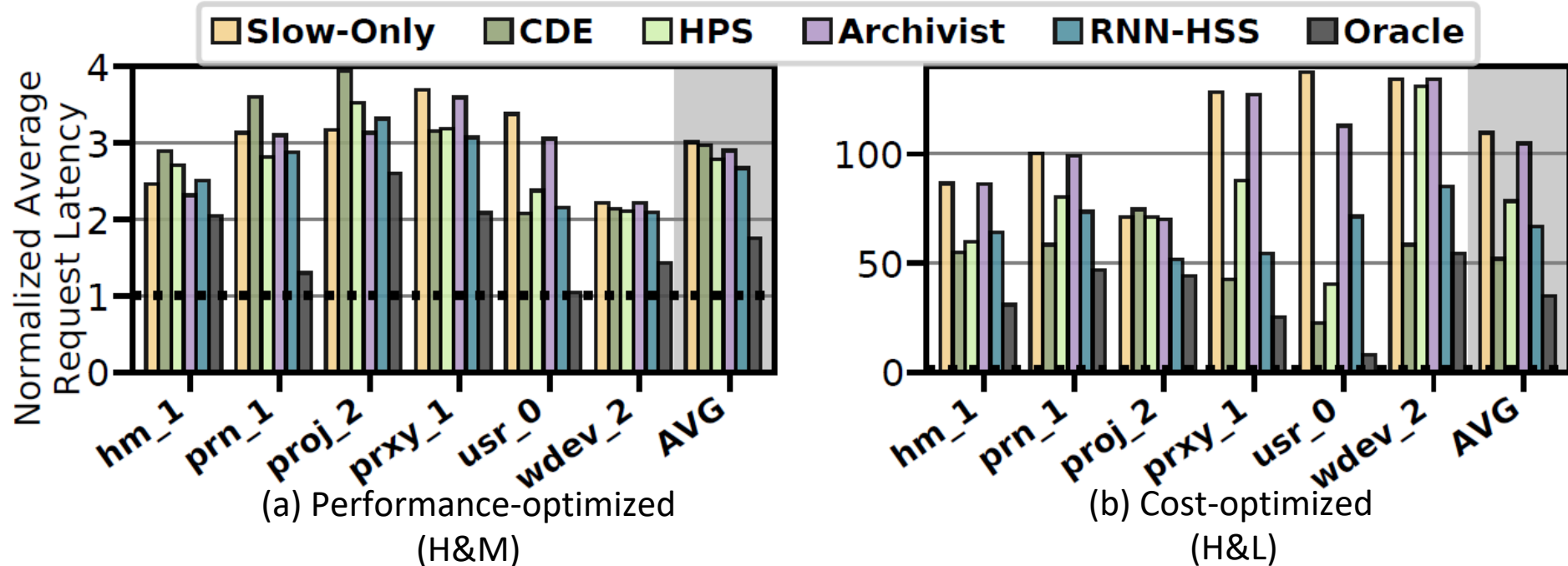
Lack of **extensibility**

# Lack of Adaptability (1/2)

- Prior heuristic-based techniques consider only a **few characteristics** (e.g., access frequency) to perform data placement
- **Statically tuned characteristics** (based on **fixed thresholds**) are ineffective when used on a wide range of applications and system configurations
- Supervised learning techniques need **labeled data** and **frequent retraining** to adapt to varying workloads and system conditions

Prior techniques offer **41.1% lower performance** compared to an Oracle policy

# Lack of Adaptability (2/2)



CDE shows an average performance gap of 41.1% (32.6%) to Oracle for H&M (H&L)

HPS shows an average performance gap of 37.2% (55.5%) to Oracle for H&M (H&L)



---

# Lack of Device Awareness

Prior data placement techniques:

- **do not adapt** well to changes in underlying device characteristics (e.g., storage read latency)
- **do not consider the data migration cost** between storage devices while making a data placement decision
- **are highly inefficient** in hybrid storage systems that have devices with significantly different read/write latencies

---

# Lack of Extensibility

- Prior data placement techniques are typically **designed** for a hybrid storage system with **only two storage devices**
- **Significant effort** is required to extend the data placement policies for more than two devices

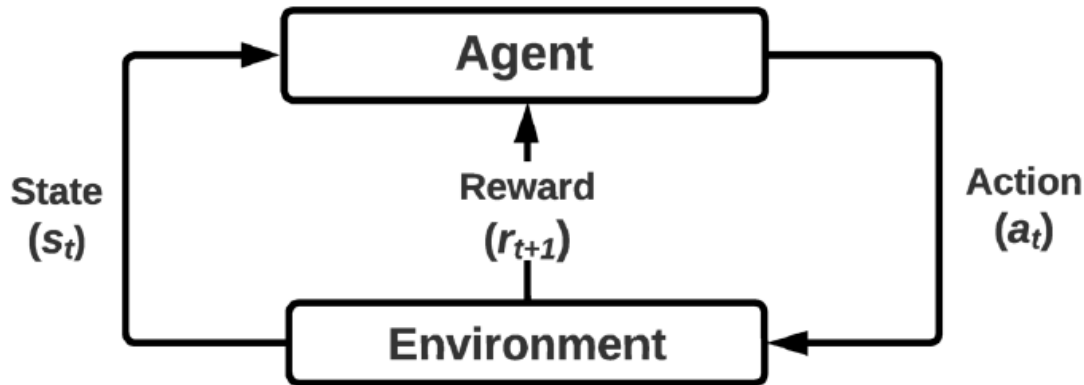
Compared to a RL-based solution, a heuristic-based policy provides **48.2% lower performance** when extended from two to three devices

# Our Goal

A **data-placement mechanism** that can

- dynamically derive an adaptive data-placement strategy by **continuously learning** and **adapting** to the application and underlying device characteristics
- be **easily extended** to incorporate a wide range of hybrid storage configurations

# Basics of Reinforcement Learning



- RL is a framework for decision making
  - An **autonomous agent observes** the current **state** of the environment
  - It **interacts** with the environment by taking **actions**
  - Agent is **rewarded** or **penalized** based on the consequences of its actions
  - Agent tries to maximize the cumulative reward

---

# Applying RL to Data Placement

Key factors in applying RL for data placement in a hybrid storage system

- RL agent needs to be **aware of:**
  - **asymmetry in read/write latencies** of a storage device
  - **differences in latencies** across hybrid storage devices
  - **application access patterns**
- Data placement module should decide which actions to reward and penalize (credit assignment)
- Low implementation overhead

# RL State

- Feature selection is performed to select only the most correlated features that affect data placement
- Divide the states into a small number of bins to reduce the state space

| Feature  | Description                                         | # of bins | Encoding (bits) |
|----------|-----------------------------------------------------|-----------|-----------------|
| $size_t$ | Size of the requested page (in pages)               | 8         | 8               |
| $type_t$ | Type of the current request (read/write)            | 2         | 4               |
| $intr_t$ | Access interval of the requested page               | 64        | 8               |
| $cnt_t$  | Access count of the requested page                  | 64        | 8               |
| $cap_t$  | Remaining capacity in the fast storage device       | 8         | 8               |
| $curr_t$ | Current placement of the requested page (fast/slow) | 2         | 4               |

# Reward

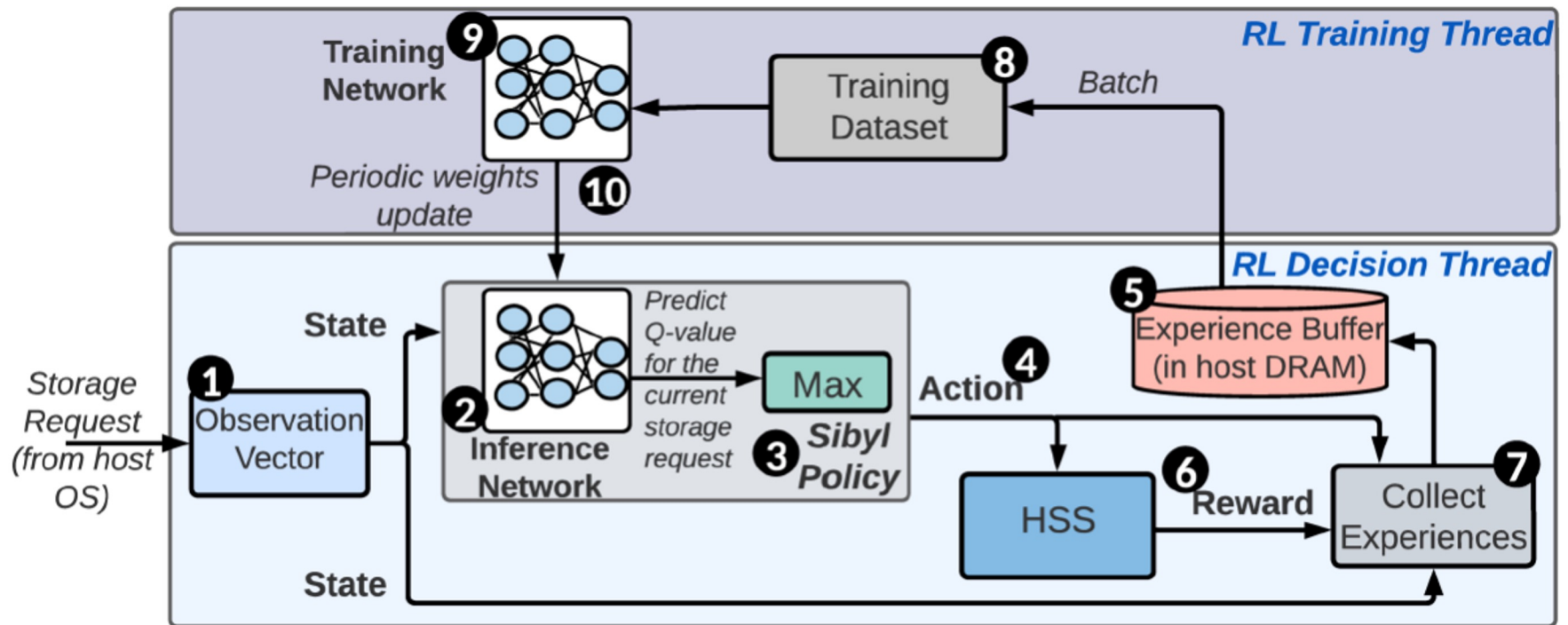
- For every action at time-step  $t$ , Sibyl gets a reward from the environment at time-step  $t + 1$
- **Reward** acts as a **feedback** to the agent's past action
- **Request latency** faithfully captures the status of the hybrid storage system
- **Penalty** value is chosen to prevent the agent from aggressively servicing all the requests from the faster device

$$R = \begin{cases} \frac{1}{L_t} & \text{if no eviction} \\ \max(0, \frac{1}{L_t} - R_p) & \text{if an eviction happens} \end{cases}$$

$L_t$  = latency of the request  
 $R_p$  = eviction penalty



# Overview of Sibyl



The two threads run asynchronously to prevent training delay from affecting the inference time

# Hyper-parameter Tuning

- Different hyper-parameter configurations were chosen using the design of experiments (DoE) technique

| Hyper-parameter                     | Design Space     | Chosen Value |
|-------------------------------------|------------------|--------------|
| Discount factor ( $\gamma$ )        | 0-1              | 0.9          |
| Learning rate ( $\alpha$ )          | $1e^{-5} - 1e^0$ | $1e^{-4}$    |
| Exploration rate ( $\epsilon$ )     | 0-1              | 0.001        |
| Batch size                          | 64-256           | 128          |
| Experience buffer size ( $e_{EB}$ ) | 10-10000         | 1000         |

---

# Evaluation Methodology

- Evaluated on a **real system** with different hybrid storage configurations
- Hybrid storage system constitutes one contiguous logical block address space
- A custom block driver was implemented to manage the I/O requests to the storage devices
- We evaluate **three** different hybrid storage configurations
  - Performance-optimized (H&M)
  - Cost-optimized (H&L)
  - Tri-hybrid storage system

# Evaluation Methodology

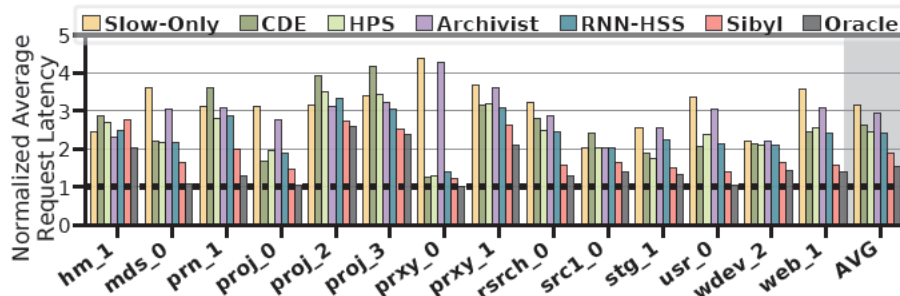
|                                         |                                                                                                              |                    |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------|--------------------|
| <b>Host System</b>                      | AMD Ryzen 7 2700G [146], 8-cores@3.5 GHz, 8×64/32 KiB L1-I/D, 4 MiB L2, 8 MiB L3, 16 GiB RDIMM DDR4 2666 MHz |                    |
| <b>Storage Devices</b>                  | <b>Characteristics</b>                                                                                       |                    |
| H: Intel Optane SSD P4800X [94]         | 375 GB, PCIe 3.0 NVMe, SLC, R/W: 2.4/2 GB/s, random R/W: 550000/500000 IOPS                                  |                    |
| M: Intel SSD D3-S4510 [96]              | 1.92 TB, SATA TLC (3D), R/W: 550/510 MB/s, random R/W: 895000/21000 IOPS                                     |                    |
| L: Seagate HDD ST1000DM010 [98]         | 1 TB, SATA 6Gb/s 7200 RPM<br>Max. Sustained Transfer Rate: 210 MB/s                                          |                    |
| L <sub>SSD</sub> : ADATA SU630 SSD [99] | 960 GB, SATA 6 Gb/s, TLC,<br>Max R/W: 520/450 MB/s                                                           |                    |
| <b>HSS Configurations</b>               | <b>Fast Device</b>                                                                                           | <b>Slow Device</b> |
| H&M (Performance-oriented)              | high-end (H)                                                                                                 | middle-end (M)     |
| H&L (Cost-oriented)                     | high-end (H)                                                                                                 | low-end (L)        |

---

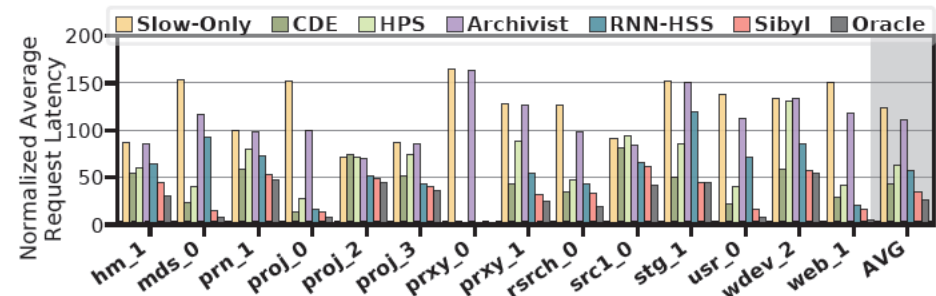
# Evaluation Methodology

- 18 different workloads from MSR Cambridge and FileBench suites
- Sibyl is compared against **four baselines**
  - Heuristic-based policies
    - Cold data eviction (CDE) [Matsui et. al., "Design of Hybrid SSDs With Storage Class Memory and NAND Flash Memory," IEEE 2017]
    - History Page Scheduler (HPS) [Meswani et.al., "Heterogeneous Memory Architectures: A HW/SW Approach for Mixing Die-stacked and Off-package Memories," HPCA, 2015]
  - Supervised learning-based policies
    - Recurrent neural network (RNN)-based technique adapted from Kleio [Doudali et.al., "Kleio: A Hybrid Memory Page Scheduler with Machine Intelligence," HPDC, 2019]
    - Neural network-based classifier based on Archivist [Ren et.al., "Archivist: A Machine Learning Assisted Data Placement Mechanism for Hybrid Storage Systems," ICCD, 2019 ]

# Latency Improvement



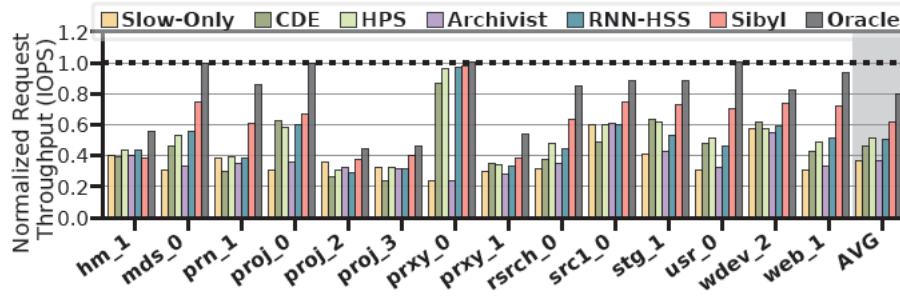
(a) Performance-optimized  
(H&M)



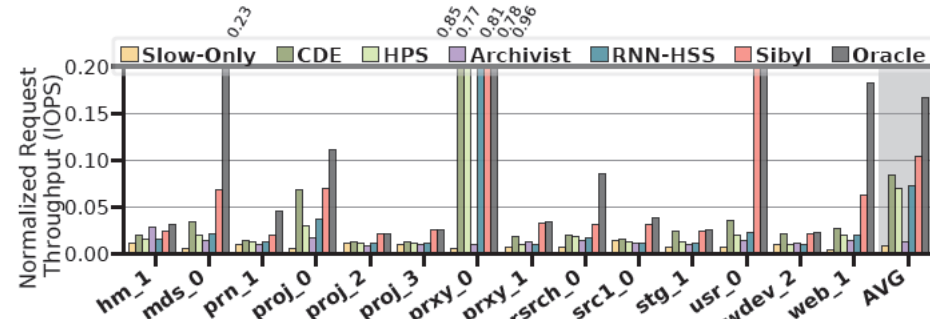
(b) Cost-optimized  
(H&L)

| Configuration | CDE   | HPS   | Archivist | RNN-HSS |
|---------------|-------|-------|-----------|---------|
| H&M           | 28.1% | 23.2% | 36.1%     | 21.6%   |
| H&L           | 19.9% | 45.9% | 68.8%     | 34.1%   |

# Throughput Improvement



(a) Performance-optimized  
(H&M)

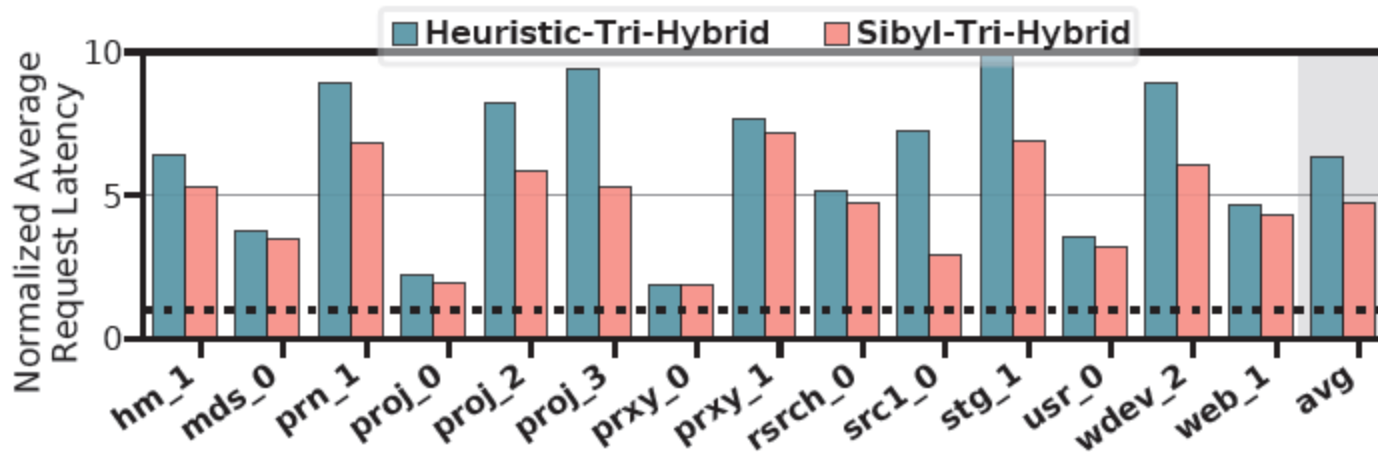


(b) Cost-optimized  
(H&L)

| Configuration | CDE   | HPS   | Archivist | RNN-HSS |
|---------------|-------|-------|-----------|---------|
| H&M           | 32.6% | 21.9% | 54.2%     | 22.7%   |
| H&L           | 22.8% | 49.1% | 86.9%     | 41.9%   |

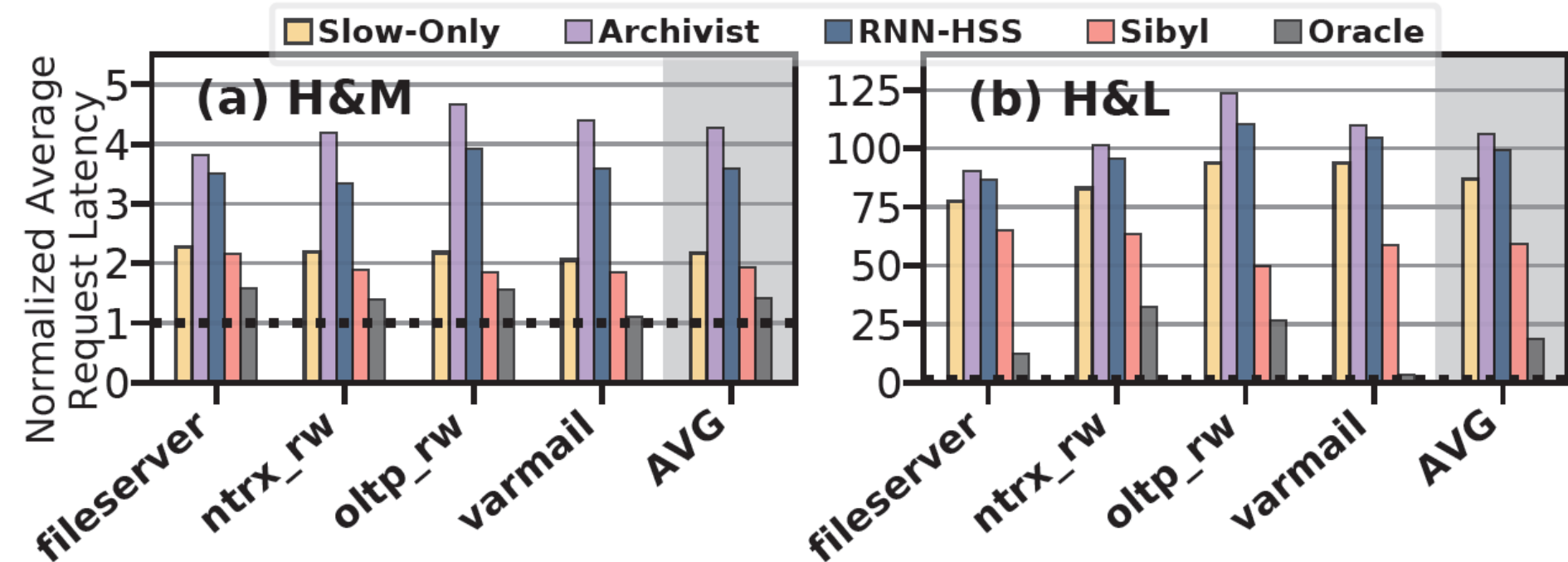


# Latency in Tri-Hybrid System



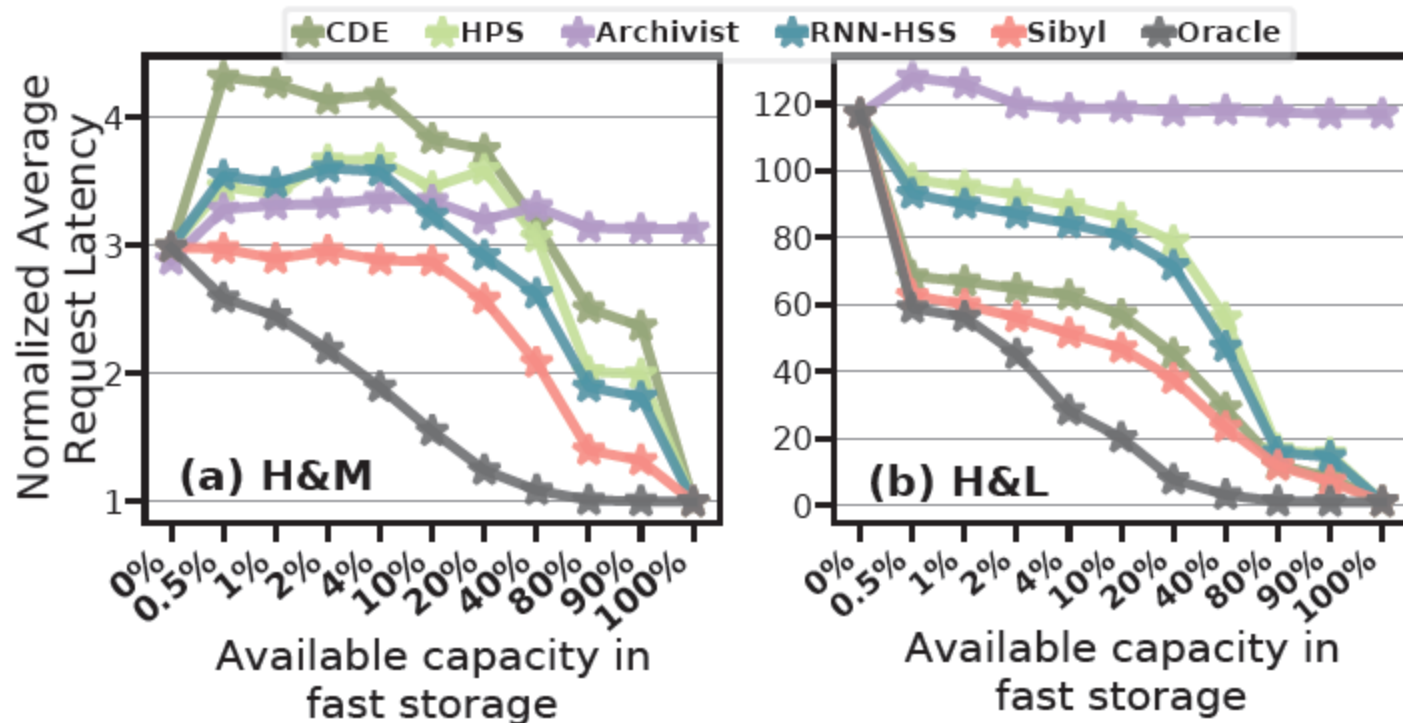
Sibyl outperforms the heuristic-based data placement policy for tri-hybrid system by 48.2% on average across all workloads

# Latency for Unseen Workloads

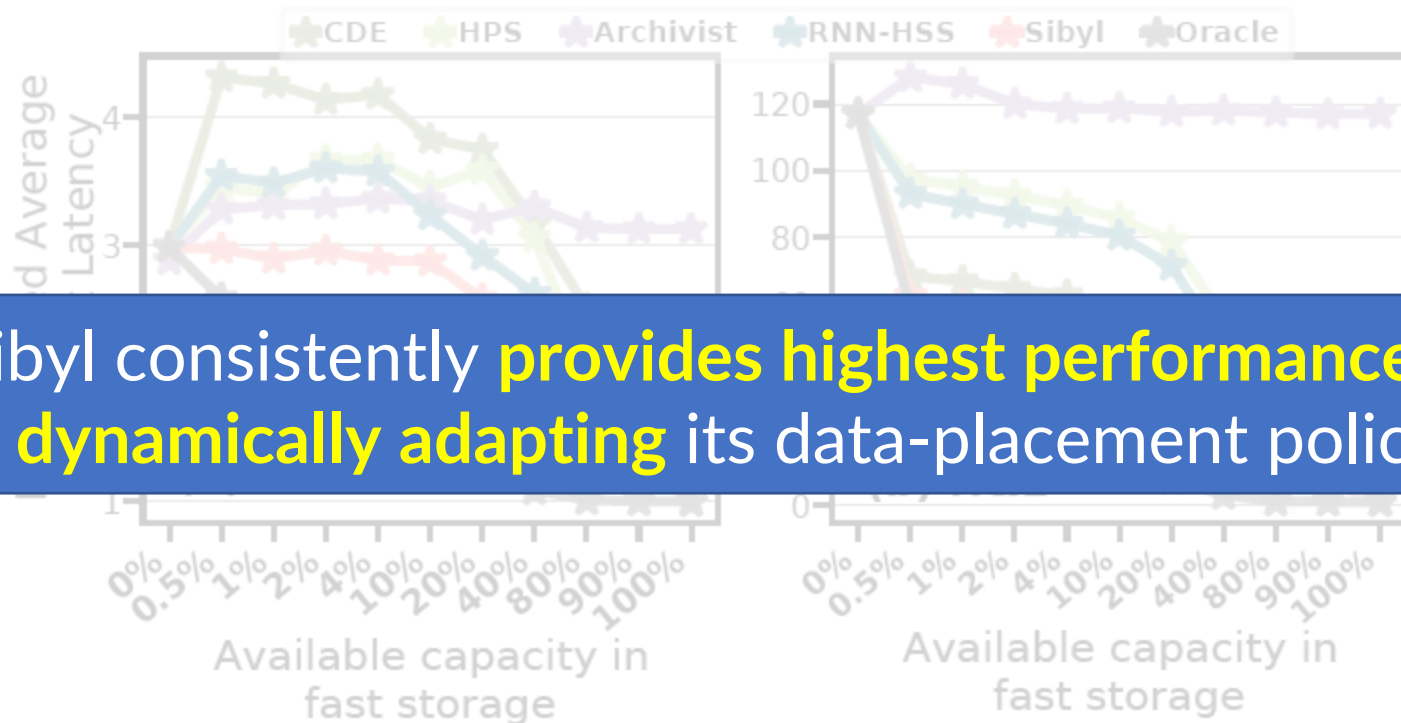


In H&M (H&L) configurations, Sibyl outperforms RNN-HSS and Archivist by 46.1% (54.6%) and 8.5% (44.1%) respectively

# Sensitivity to Fast Storage Capacity



# Sensitivity to Fast Storage Capacity



Sibyl consistently **provides highest performance** by **dynamically adapting** its data-placement policy

---

# Overhead Analysis

- **Performance Overhead**
  - **~10ns** for every inference on the evaluated system; this is several orders of magnitude less than I/O latency of high-end SSD
- **Implementation Overhead**
  - **124.4 KiB** of implementation overhead
- **Metadata overhead**
  - 0.1% of the total storage capacity when using a 4 KiB data placement granularity
  - **40-bit** metadata overhead per data placement unit

# For More on Sybil

---

- To appear in ISCA 2022

## **Sibyl: Adaptive and Extensible Data Placement in Hybrid Storage Systems Using Online Reinforcement Learning**

Gagandeep Singh<sup>1</sup>   Rakesh Nadig<sup>1</sup>   Jisung Park<sup>1</sup>   Rahul Bera<sup>1</sup>   Nastaran Hajinazar<sup>1</sup>  
David Novo<sup>3</sup>   Juan Gómez-Luna<sup>1</sup>   Sander Stuijk<sup>2</sup>   Henk Corporaal<sup>2</sup>   Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Eindhoven University of Technology

<sup>3</sup>LIRMM, Univ. Montpellier, CNRS

<https://arxiv.org/pdf/2205.07394.pdf>

# 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)**



## **Data-Driven** **(Self-Optimizing)** **Computing Architectures**

# Data-Aware Architectures

# Corollaries: Architectures Today ...

---

- Architectures are **terrible at dealing with data**
  - ❑ Designed to mainly store and move data vs. to compute
  - ❑ They are **processor-centric** as opposed to **data-centric**
- Architectures are **terrible at taking advantage of vast amounts of data** (and metadata) available to them
  - ❑ Designed to make simple decisions, ignoring lots of data
  - ❑ They make **human-driven decisions** vs. **data-driven** decisions
- Architectures are **terrible at knowing and exploiting different properties of application data**
  - ❑ Designed to treat all data as the same
  - ❑ They make **component-aware decisions** vs. **data-aware**

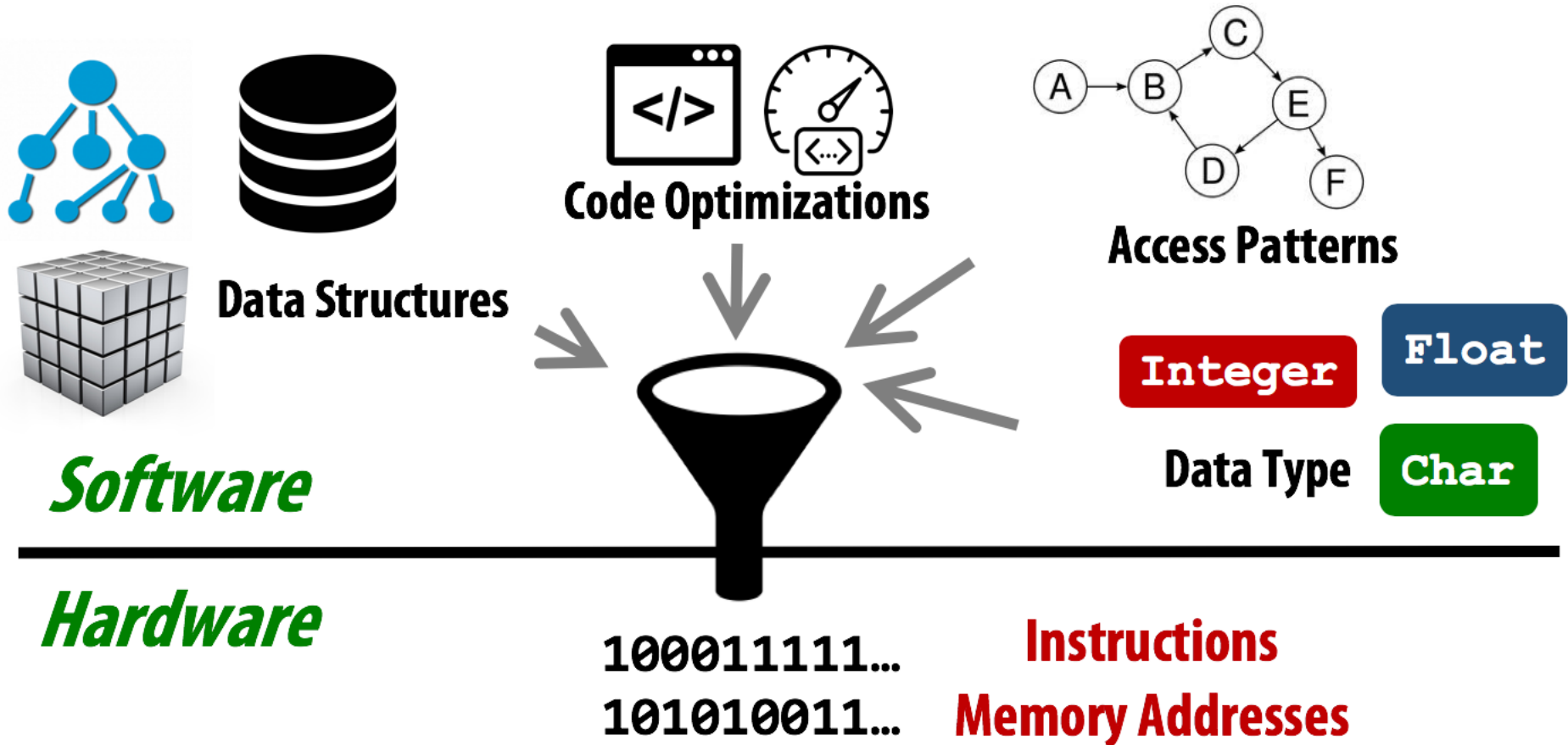
# Data-Aware Architectures

---

- 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

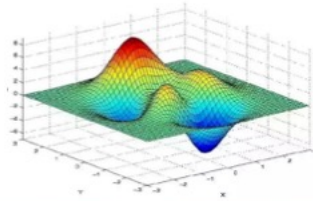
## Higher-level information is not visible to HW



# A Solution: More Expressive Interfaces

**Performance**

**Software**



**Functionality**



**ISA  
Virtual Memory**

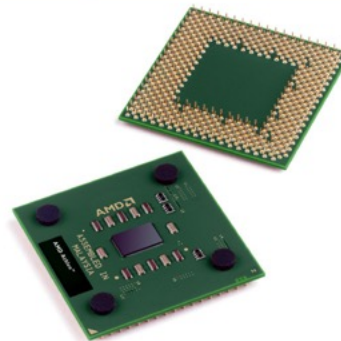
**Higher-level  
Program  
Semantics**

**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<sup>†§</sup> Abhilasha Jain<sup>†</sup> Diptesh Majumdar<sup>†</sup> Kevin Hsieh<sup>†</sup> Gennady Pekhimenko<sup>‡</sup>  
Eiman Ebrahimi<sup>⌘</sup> Nastaran Hajinazar<sup>†</sup> Phillip B. Gibbons<sup>†</sup> Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>University of Toronto

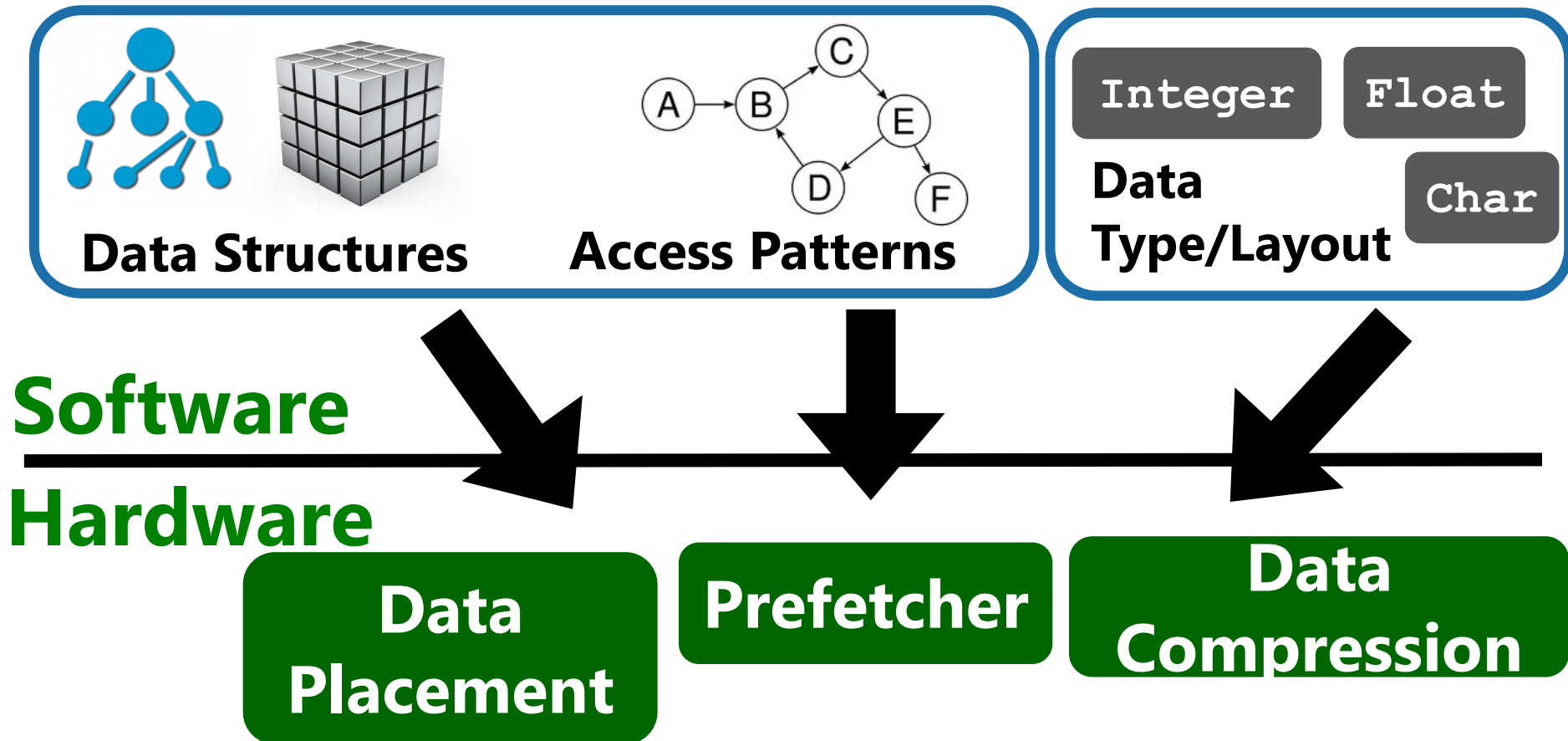
<sup>⌘</sup>NVIDIA

<sup>†</sup>Simon Fraser University

<sup>§</sup>ETH Zürich



# SW provides key program information to HW



# Broader goal: Enable many cross-layer optimizations

## Express:

**Data structures**

**Access semantics**

**Data types**

**Working set**

**Reuse**

**Access frequency**

...

## Optimizations:

**Cache Management**

**Data Placement in DRAM**

**Data Compression**

**Approximation**

**DRAM Cache Management**

**NVM Management**

**NUCA/NUMA**

**Optimizations**

...

## Benefits:

**More efficient HW:**

✓ **Performance**

**Reduced SW  
burden:**

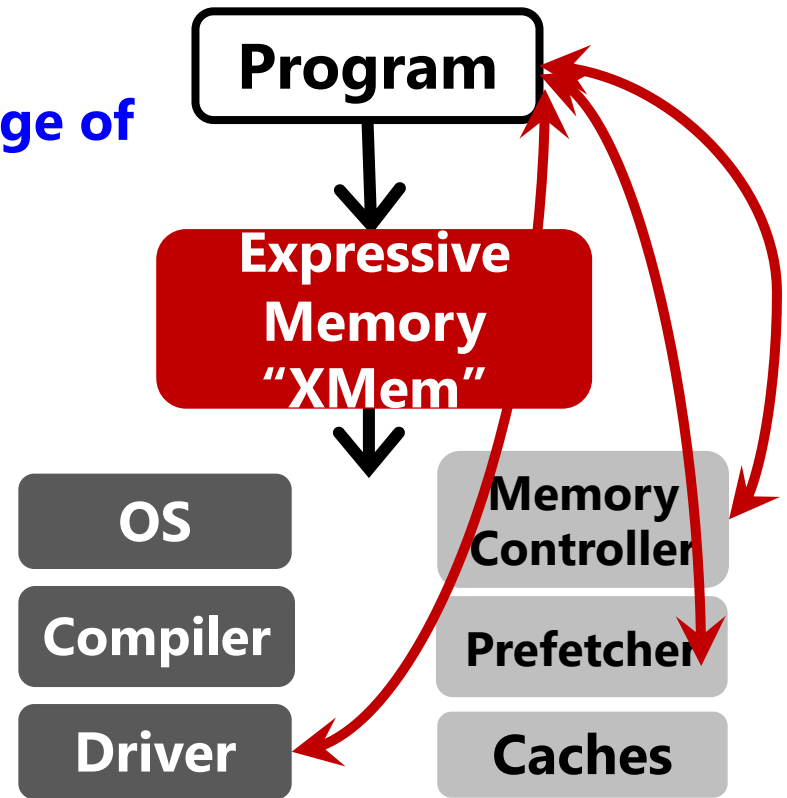
✓ **Programmability**

✓ **Portability**

# Our approach: Rich cross-layer abstractions

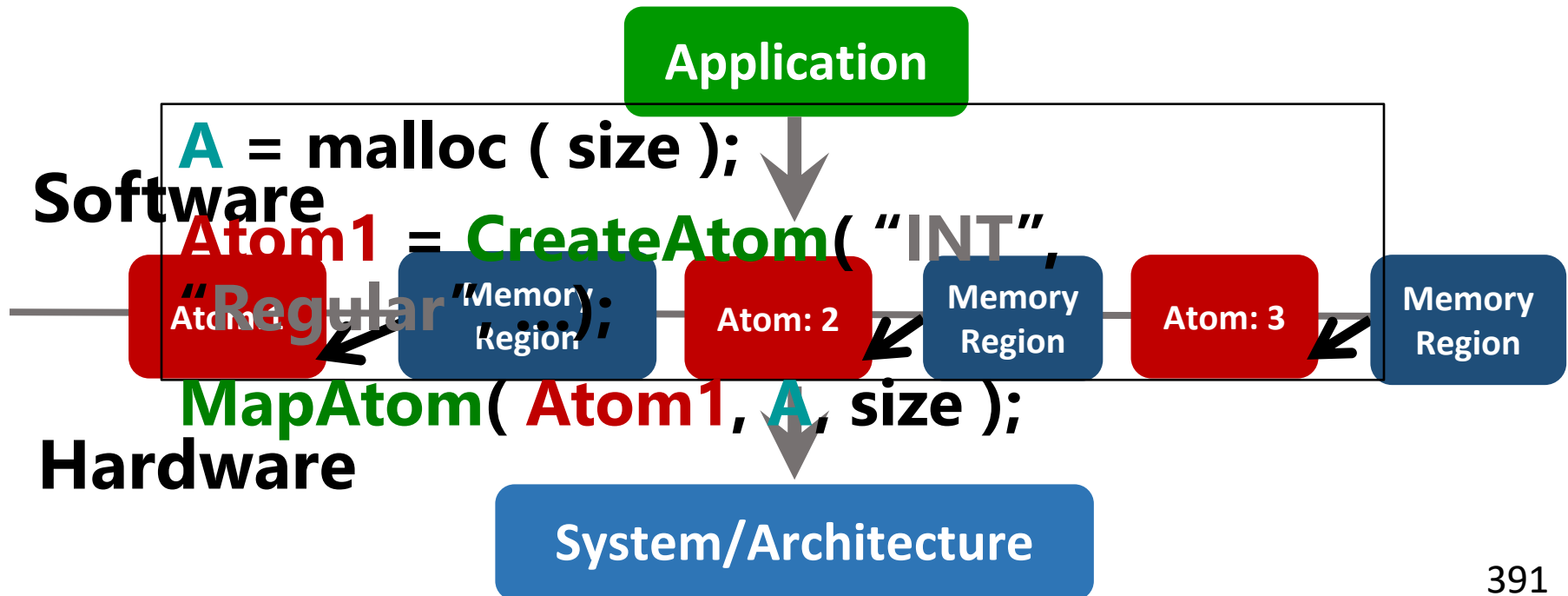
1. **Generality:** Enable a wide range of cross-layer approaches
2. **Minimize programmer effort**
3. **Overhead**

Approach: Flexibly associate specific semantic information with any **data & code**



# Example: XMem

- **Goal:** convey data semantics to the hardware enables more intelligent management of resources.
- **XMem:** introduces a new HW/SW abstraction, called *Atom*, for conveying data semantics



# XMem Aids/Enables Many Optimizations

**Table 1: Summary of the example memory optimizations that XMem aids.**

| Memory optimization                                   | Example semantics provided by XMem (described in §3.3)                                                                       | Example Benefits of XMem                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cache management                                      | (i) Distinguishing between data structures or pools of similar data; (ii) Working set size; (iii) Data reuse                 | Enables: (i) applying different caching policies to different data structures or pools of data; (ii) avoiding cache thrashing by <i>knowing</i> the active working set size; (iii) bypassing/prioritizing data that has no/high reuse. (§5)                                                                                                                                               |
| Page placement in DRAM<br>e.g., [23, 24]              | (i) Distinguishing between data structures; (ii) Access pattern; (iii) Access intensity                                      | Enables page placement at the <i>data structure</i> granularity to (i) isolate data structures that have high row buffer locality and (ii) spread out concurrently-accessed irregular data structures across banks and channels to improve parallelism. (§6)                                                                                                                              |
| Cache/memory compression<br>e.g., [25–32]             | (i) Data type: integer, float, char; (ii) Data properties: sparse, pointer, data index                                       | Enables using a <i>different compression algorithm</i> for each data structure based on data type and data properties, e.g., sparse data encodings, FP-specific compression, delta-based compression for pointers [27].                                                                                                                                                                   |
| Data prefetching<br>e.g., [33–36]                     | (i) Access pattern: strided, irregular, irregular but repeated (e.g., graphs), access stride; (ii) Data type: index, pointer | Enables (i) <i>highly accurate</i> software-driven prefetching while leveraging the benefits of hardware prefetching (e.g., by being memory bandwidth-aware, avoiding cache thrashing); (ii) using different prefetcher <i>types</i> for different data structures: e.g., stride [33], tile-based [20], pattern-based [34–37], data-based for indices/pointers [38, 39], etc.             |
| DRAM cache management<br>e.g., [40–46]                | (i) Access intensity; (ii) Data reuse; (iii) Working set size                                                                | (i) Helps avoid cache thrashing by knowing working set size [44]; (ii) Better DRAM cache management via reuse behavior and access intensity information.                                                                                                                                                                                                                                  |
| Approximation in memory<br>e.g., [47–53]              | (i) Distinguishing between pools of similar data; (ii) Data properties: tolerance towards approximation                      | Enables (i) each memory component to track how approximable data is (at a fine granularity) to inform approximation techniques; (ii) data placement in heterogeneous reliability memories [54].                                                                                                                                                                                           |
| Data placement: NUMA systems<br>e.g., [55, 56]        | (i) Data partitioning across threads (i.e., relating data to threads that access it); (ii) Read-Write properties             | Reduces the need for profiling or data migration (i) to co-locate data with threads that access it and (ii) to identify Read-Only data, thereby enabling techniques such as replication.                                                                                                                                                                                                  |
| Data placement: hybrid memories<br>e.g., [16, 57, 58] | (i) Read-Write properties (Read-Only/Read-Write); (ii) Access intensity; (iii) Data structure size; (iv) Access pattern      | Avoids the need for profiling/migration of data in hybrid memories to (i) effectively manage the asymmetric read-write properties in NVM (e.g., placing Read-Only data in the NVM) [16, 57]; (ii) make tradeoffs between data structure "hotness" and size to allocate fast/high bandwidth memory [14]; and (iii) leverage row-buffer locality in placement based on access pattern [45]. |
| Managing NUCA systems<br>e.g., [15, 59]               | (i) Distinguishing pools of similar data; (ii) Access intensity; (iii) Read-Write or Private-Shared properties               | (i) Enables using different cache policies for different data pools (similar to [15]); (ii) Reduces the need for reactive mechanisms that detect sharing and read-write characteristics to inform cache policies.                                                                                                                                                                         |

# 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<sup>†§</sup> Abhilasha Jain<sup>†</sup> Diptesh Majumdar<sup>†</sup> Kevin Hsieh<sup>†</sup> Gennady Pekhimenko<sup>‡</sup>  
Eiman Ebrahimi<sup>⌘</sup> Nastaran Hajinazar<sup>†</sup> Phillip B. Gibbons<sup>†</sup> Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>University of Toronto

<sup>⌘</sup>NVIDIA

<sup>†</sup>Simon Fraser University

<sup>§</sup>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<sup>†§</sup>      Eiman Ebrahimi<sup>‡</sup>      Kevin Hsieh<sup>†</sup>  
Phillip B. Gibbons<sup>†</sup>      Onur Mutlu<sup>§†</sup>  
<sup>†</sup>Carnegie Mellon University      <sup>‡</sup>NVIDIA      <sup>§</sup>ETH Zürich



# Locality Descriptor: Executive Summary

Exploiting data locality in GPUs is a challenging task

Flexible,  
architecture-  
agnostic interface

Application

Access to  
program  
semantics

Software

Locality  
Descriptor

Hardware

Data  
Placement

Cache  
Management

CTA  
Scheduling

Data  
Prefetching

...

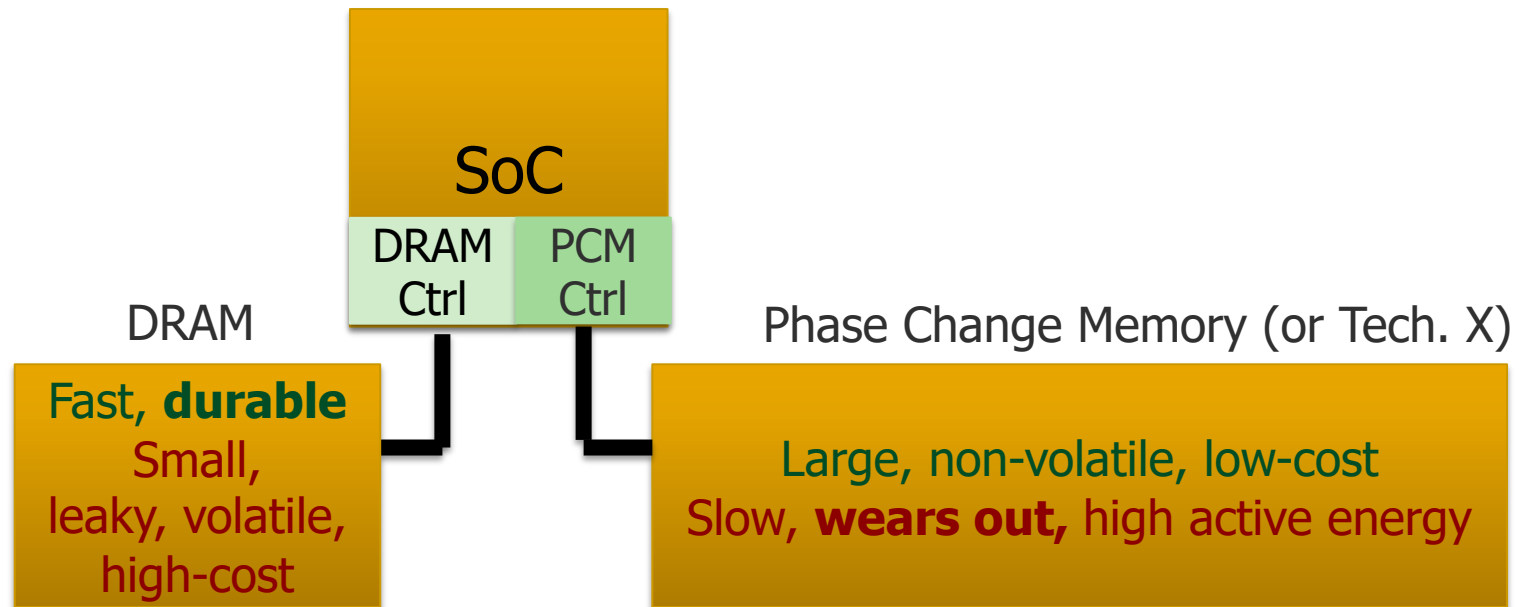
Performance Benefits:

26.6% (up to 46.6%) from cache locality

53.7% (up to 2.8x) from NUMA locality

# An Example: Hybrid Memory Management

---



Hardware/software manage data allocation and movement  
to achieve the best of multiple technologies

Meza+, "[Enabling Efficient and Scalable Hybrid Memories](#)," IEEE Comp. Arch. Letters, 2012.

Yoon+, "[Row Buffer Locality Aware Caching Policies for Hybrid Memories](#)," ICCD 2012 Best Paper Award.

# An Example: 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](#)]

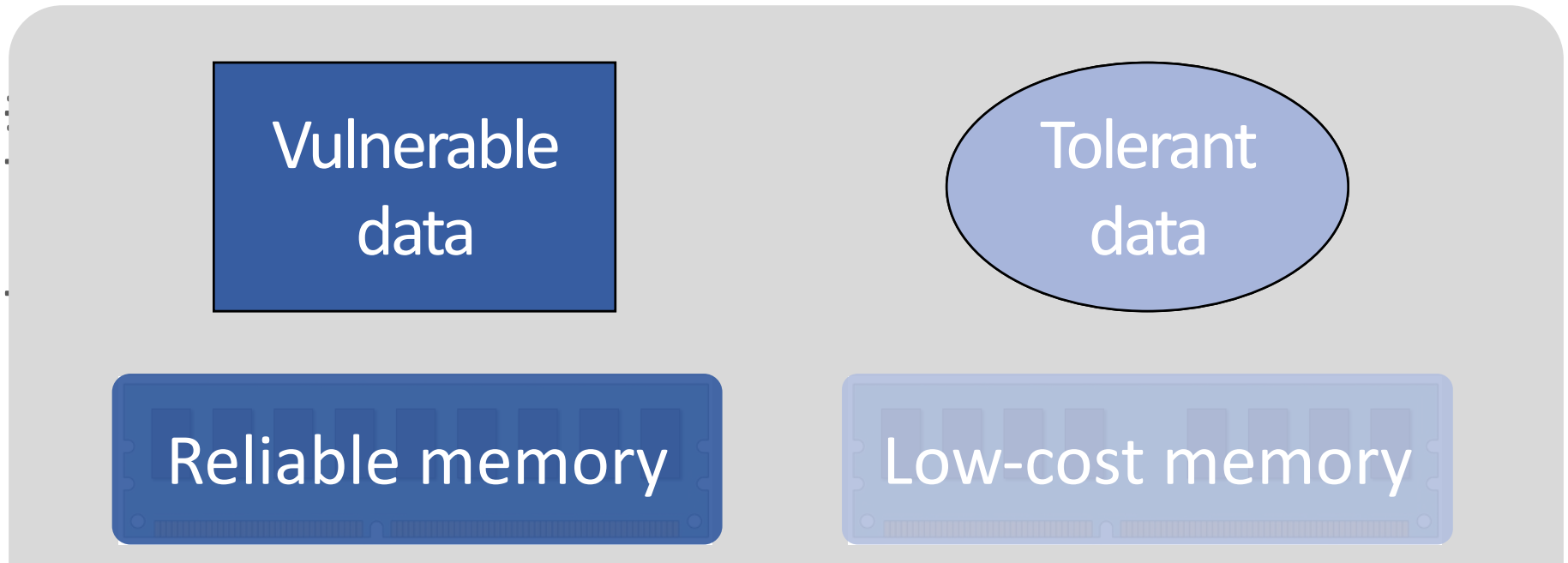
## Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory

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, bknessib, kvaid}@microsoft.com

# Exploiting Memory Error Tolerance with Hybrid Memory Systems



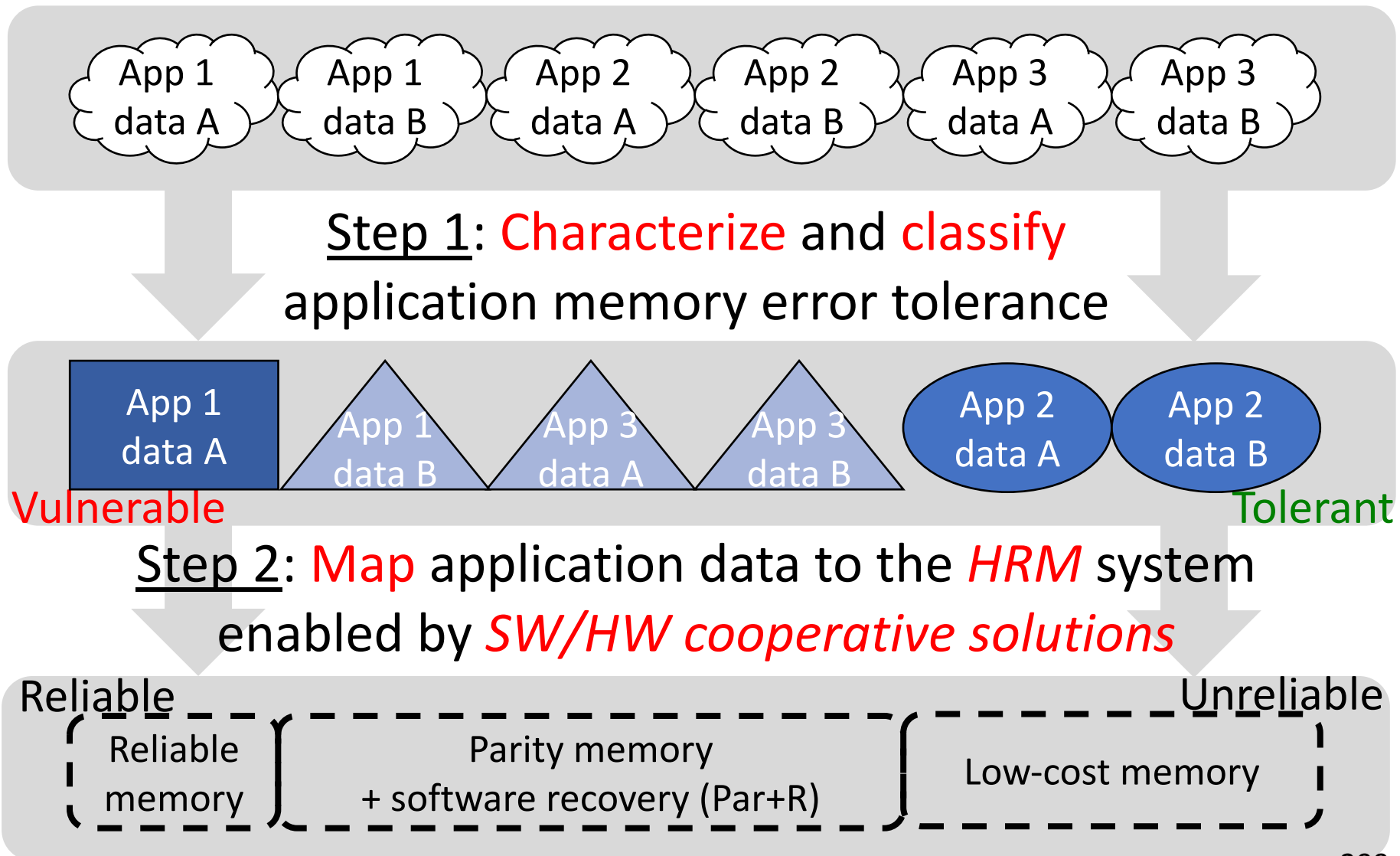
On Microsoft's Web Search workload

Reduces server hardware **cost** by **4.7 %**

Achieves single server **availability** target of **99.90 %**

**Heterogeneous-Reliability Memory** [DSN 2014]

# Heterogeneous-Reliability Memory



# More on 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](#)]

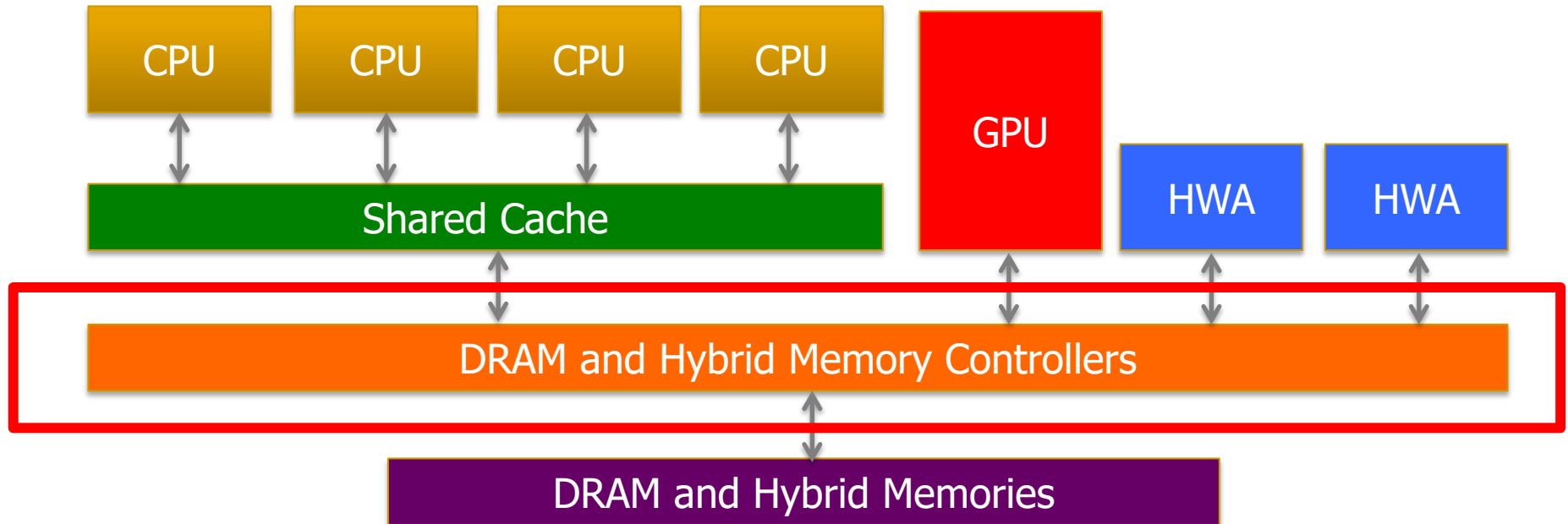
## Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory

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, bk Hessib, kvaid}@microsoft.com

# Data-Aware Cross-Layer Hybrid System Management



- Heterogeneous agents: CPUs, GPUs, and HWAs
- Main memory interference between CPUs, GPUs, HWAs
- Many timing constraints for various memory types
- Many goals at the same time: performance, fairness, QoS, energy efficiency, ...



# Another Example: EDEN for DNNs

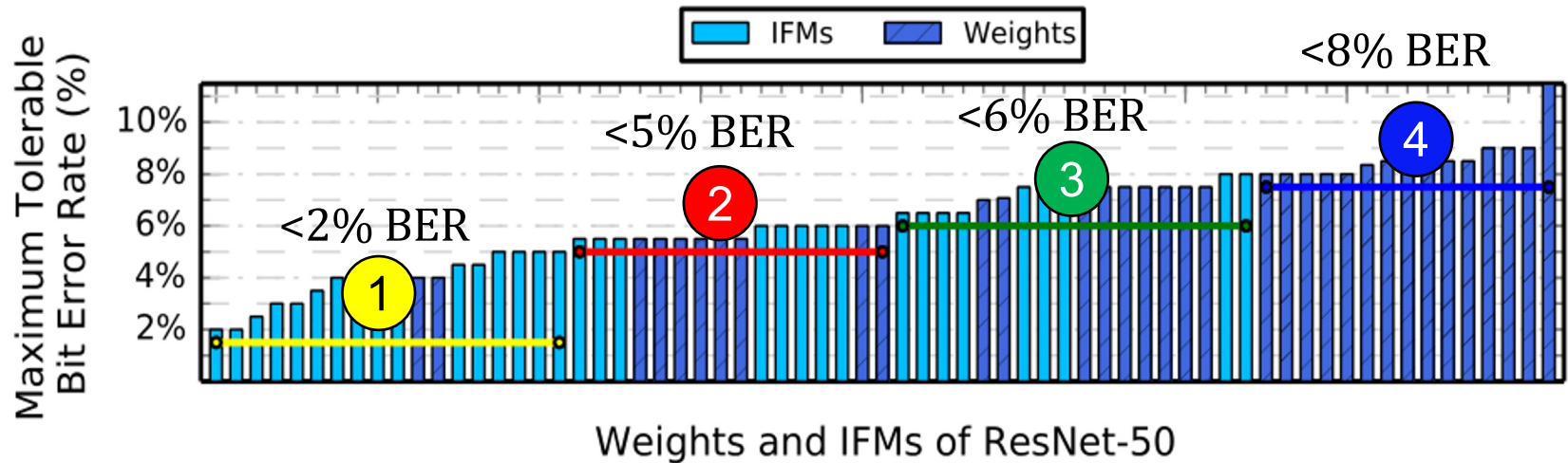
---

- Deep Neural Network evaluation is very DRAM-intensive (especially for large networks)
1. Some data and layers in DNNs are very tolerant to errors
  2. Reduce DRAM latency and voltage on such data and layers
  3. While still achieving a user-specified DNN accuracy target by making training DRAM-error-aware

**Data-aware management of DRAM latency and voltage for Deep Neural Network Inference**

# Example DNN Data Type to DRAM Mapping

Mapping example of ResNet-50:



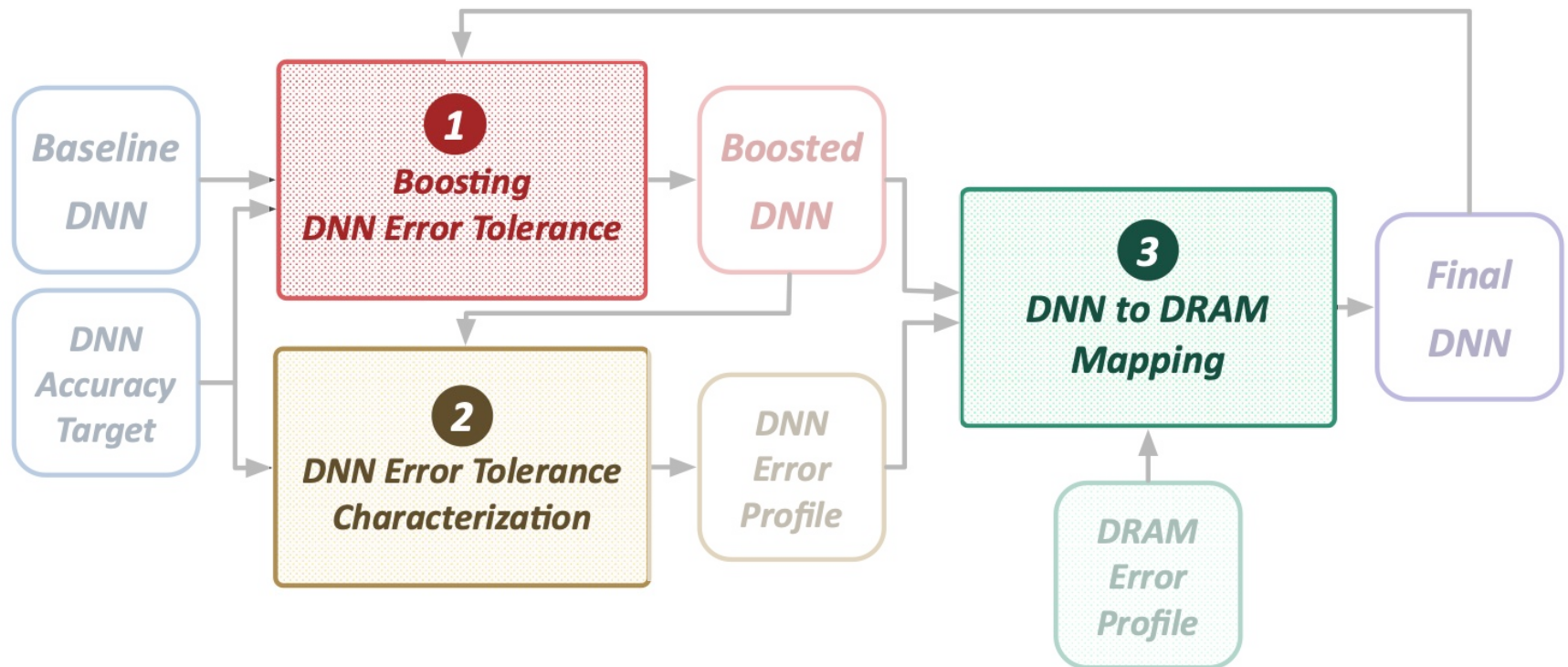
**Map more error-tolerant DNN layers**  
**to DRAM partitions with lower voltage/latency**

**4 DRAM partitions** with different error rates

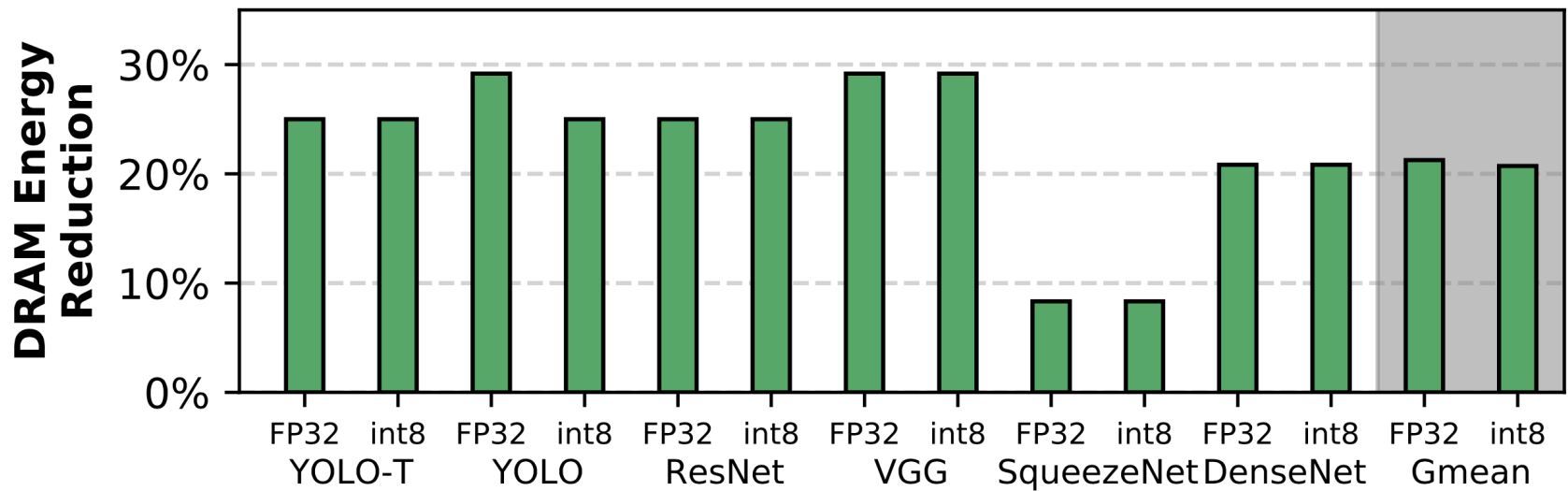
# EDEN: Overview

Key idea: Enable **accurate, efficient** DNN inference using **approximate DRAM**

EDEN is an **iterative** process that has **3 key steps**

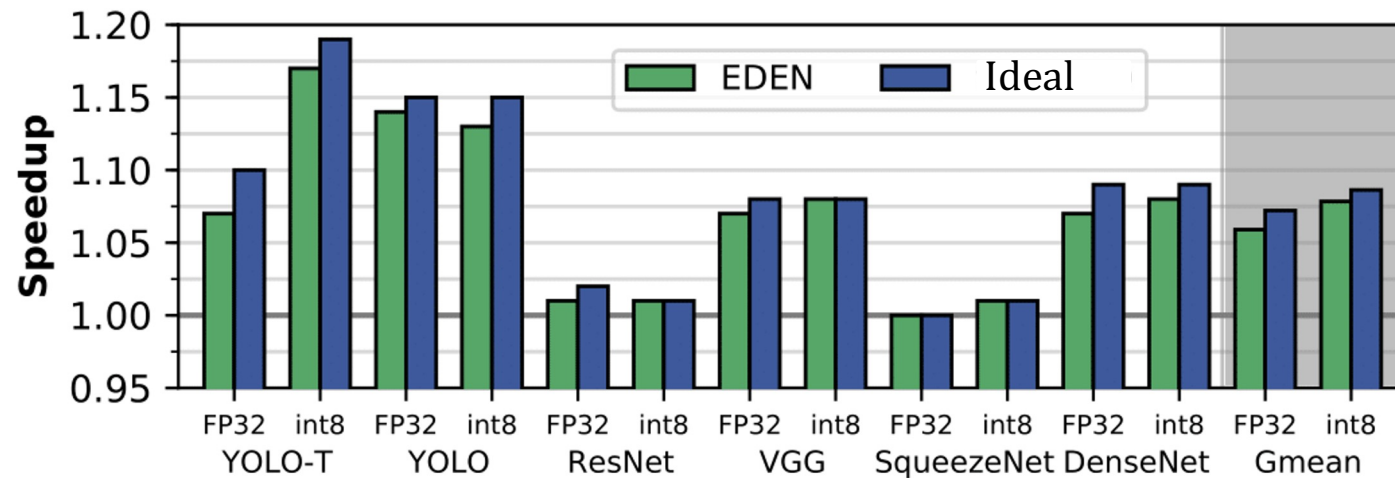


# CPU: DRAM Energy Evaluation



**Average 21% DRAM energy reduction**  
maintaining accuracy within 1% of original

# CPU: Performance Evaluation



**Average 8% system speedup**  
Some workloads achieve **17% speedup**

EDEN achieves **close to the ideal** speedup  
possible via tRCD scaling

# GPU, Eyeriss, and TPU: Energy Evaluation

- GPU: average **37% energy reduction**
- Eyeriss: average **31% energy reduction**
- TPU: average **32% energy reduction**

# 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.  
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]  
[[Lightning Talk Video](#) (90 seconds)]

## 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 Kanellopoulos<sup>1</sup> Nandita Vijaykumar<sup>2,1</sup> Christina Giannoula<sup>1,3</sup> Roknoddin Azizi<sup>1</sup>  
Skanda Koppula<sup>1</sup> Nika Mansouri Ghiasi<sup>1</sup> Taha Shahroodi<sup>1</sup> Juan Gomez Luna<sup>1</sup> Onur Mutlu<sup>1,2</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Carnegie Mellon University

<sup>3</sup>National Technical University of Athens

# Data-Aware Virtual Memory Framework

---

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)*, Virtual, 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)]

[[Lecture Video](#) (43 minutes)]

## The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework

Nastaran Hajinazar<sup>\*†</sup> Pratyush Patel<sup>⌘</sup> Minesh Patel<sup>\*</sup> Konstantinos Kanellopoulos<sup>\*</sup> Saugata Ghose<sup>‡</sup>  
Rachata Ausavarungnirun<sup>⊙</sup> Geraldo F. Oliveira<sup>\*</sup> Jonathan Appavoo<sup>◇</sup> Vivek Seshadri<sup>▽</sup> Onur Mutlu<sup>\*‡</sup>

<sup>\*</sup>ETH Zürich <sup>†</sup>Simon Fraser University <sup>⌘</sup>University of Washington <sup>‡</sup>Carnegie Mellon University

<sup>⊙</sup>King Mongkut's University of Technology North Bangkok <sup>◇</sup>Boston University <sup>▽</sup>Microsoft Research India

# SW/HW Climate Modeling Accelerator

---

- 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<sup>a,b,c</sup>    Dionysios Diamantopoulos<sup>c</sup>    Christoph Hagleitner<sup>c</sup>    Juan Gómez-Luna<sup>b</sup>  
Sander Stuijk<sup>a</sup>    Onur Mutlu<sup>b</sup>    Henk Corporaal<sup>a</sup>  
<sup>a</sup>Eindhoven University of Technology    <sup>b</sup>ETH Zürich    <sup>c</sup>IBM Research Europe, Zurich

# HW/SW Time Series Analysis Accelerator

---

- 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<sup>§</sup>

Ricardo Quisiant<sup>§</sup>

Christina Giannoula<sup>†</sup>

Mohammed Alser<sup>‡</sup>

Juan Gómez-Luna<sup>‡</sup>

Eladio Gutiérrez<sup>§</sup>

Oscar Plata<sup>§</sup>

Onur Mutlu<sup>‡</sup>

<sup>§</sup>*University of Malaga*

<sup>†</sup>*National Technical University of Athens*

<sup>‡</sup>*ETH Zürich*

# 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<sup>◇</sup> Mohammed Alser<sup>◇</sup> Damla Senol Cali<sup>✕</sup>

Dionysios Diamantopoulos<sup>▽</sup> Juan Gómez-Luna<sup>◇</sup>

Henk Corporaal<sup>★</sup> Onur Mutlu<sup>◇✕</sup>

<sup>◇</sup>*ETH Zürich*    <sup>✕</sup>*Carnegie Mellon University*

<sup>★</sup>*Eindhoven University of Technology*    <sup>▽</sup>*IBM Research Europe*

# 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<sup>†</sup> Samira Khan<sup>‡</sup> Nandita Vijaykumar<sup>†</sup>  
Kevin K. Chang<sup>†</sup> Amirali Boroumand<sup>†</sup> Saugata Ghose<sup>†</sup> Onur Mutlu<sup>§†</sup>  
<sup>†</sup>*Carnegie Mellon University*   <sup>‡</sup>*University of Virginia*   <sup>§</sup>*ETH Zürich*

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



# Accelerating Genome Analysis [IEEE MICRO 2020]

---

- Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,  
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#)  
[IEEE Micro \(IEEE MICRO\)](#), Vol. 40, No. 5, pages 65-75, September/October 2020.  
[\[Slides \(pptx\)\(pdf\)\]](#)  
[\[Talk Video \(1 hour 2 minutes\)\]](#)

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

**Mohammed Alser**

ETH Zürich

**Zülal Bingöl**

Bilkent University

**Damla Senol Cali**

Carnegie Mellon University

**Jeremie Kim**

ETH Zurich and Carnegie Mellon University

**Saugata Ghose**

University of Illinois at Urbana–Champaign and  
Carnegie Mellon University

**Can Alkan**

Bilkent University

**Onur Mutlu**

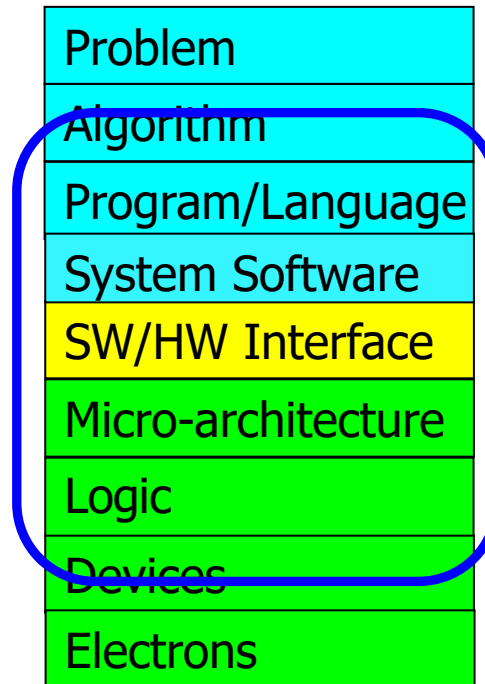
ETH Zurich, Carnegie Mellon University, and  
Bilkent University

## **Data-Aware (Expressive)**

## **Computing Architectures**

# We Need to **Rethink** the Entire Stack

---



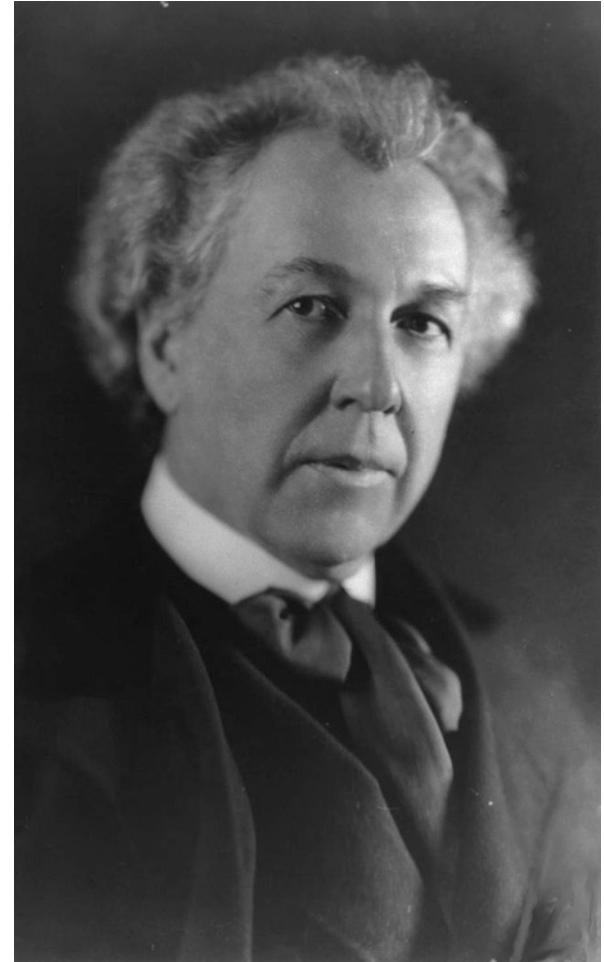
**We can get there case by case**

# Principled Architectures & What They Can Enable

# 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



Source: By 準建築人手札網站 Forgemind ArchiMedia - Flickr: IMG\_2489.JPG, CC BY 2.0

Source: <https://www.dezeen.com/2016/08/29/santiago-calatrava-oculus-world-trade-center-transportation-hub-new-york-photographs-hufton-crow/>  
<https://commons.wikimedia.org/wiki/index.php?curid=91498396>, [https://en.wikipedia.org/wiki/Santiago\\_Calatrava](https://en.wikipedia.org/wiki/Santiago_Calatrava)



# 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."<sup>[1]</sup>

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.<sup>[3]</sup>

# Overarching Principles for Computing?

---



# Readings, Videos, Reference Materials

# More on My Research & Teaching

# Brief Self Introduction

---



## ■ Onur Mutlu

- ❑ Full Professor @ ETH Zurich ITET (INFK), since September 2015
- ❑ Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-...
- ❑ PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
- ❑ <https://people.inf.ethz.ch/omutlu/>
- ❑ [omutlu@gmail.com](mailto:omutlu@gmail.com) (Best way to reach me)
- ❑ <https://people.inf.ethz.ch/omutlu/projects.htm>

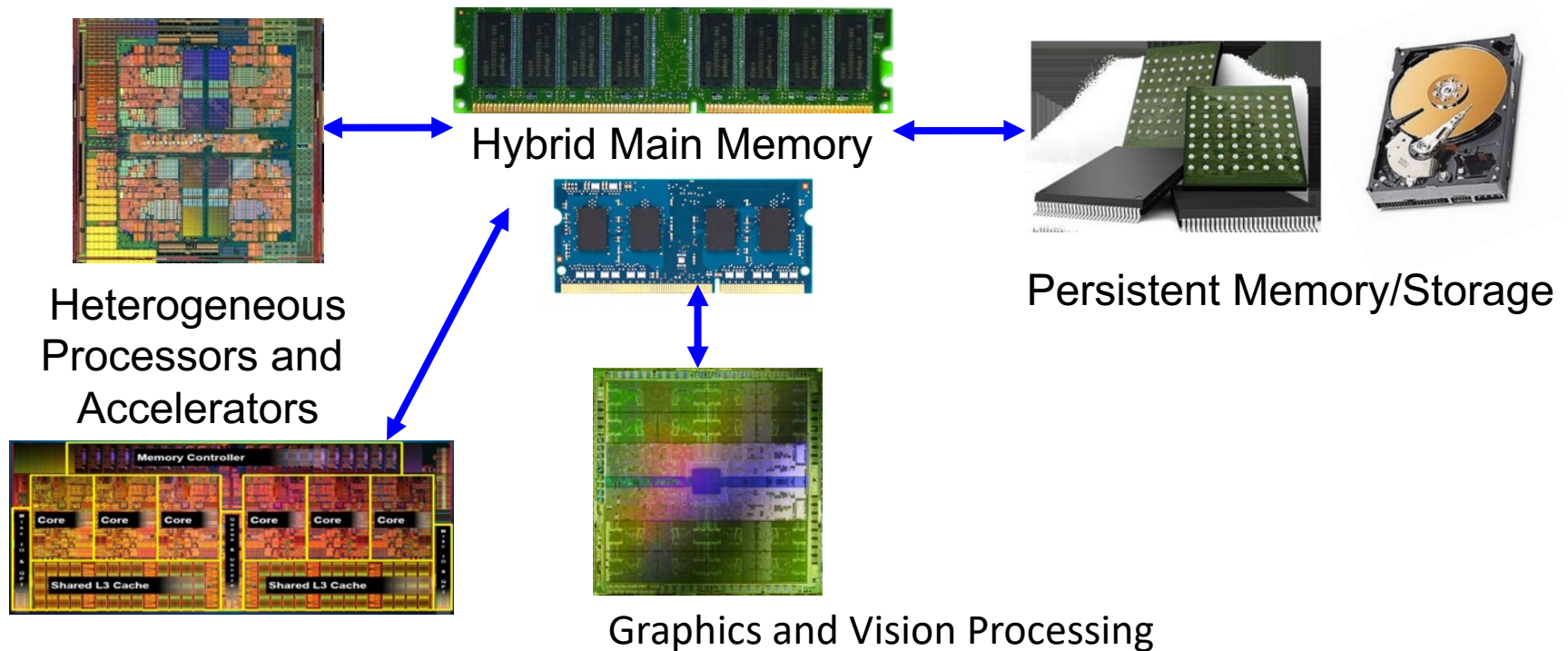
## ■ Research and Teaching in:

- ❑ Computer architecture, computer systems, hardware security, bioinformatics
- ❑ Memory and storage systems
- ❑ Hardware security, safety, predictability
- ❑ Fault tolerance
- ❑ Hardware/software cooperation
- ❑ Architectures for bioinformatics, health, medicine
- ❑ ...

# Current Research Mission

---

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



**Build fundamentally better architectures**

# Four Key Current Directions

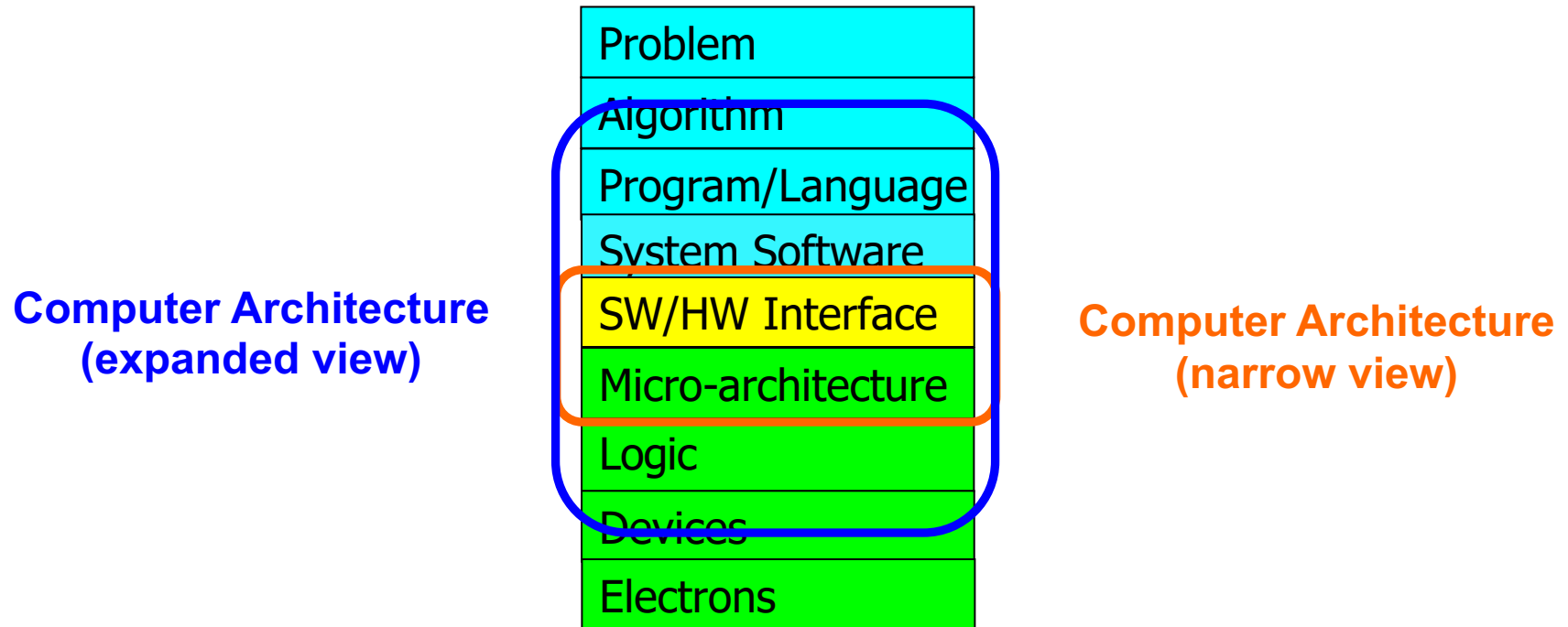
---

- Fundamentally Secure/Reliable/Safe Architectures
- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures
- Fundamentally Low-Latency and Predictable Architectures
- Architectures for AI/ML, Genomics, Medicine, Health



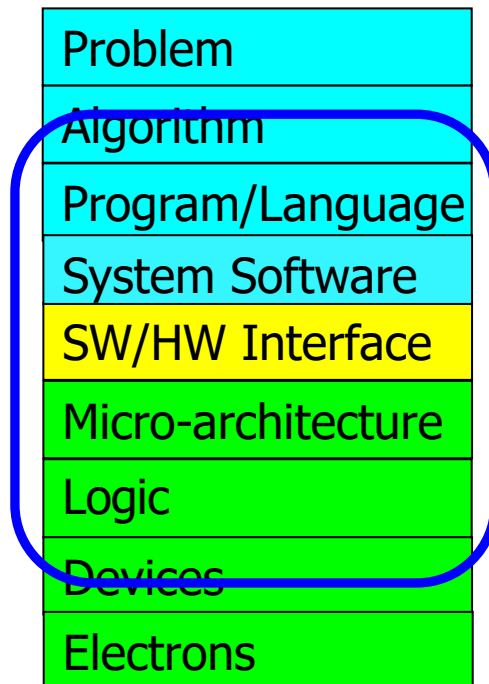
# The Transformation Hierarchy

---



To achieve the highest **energy efficiency** and **performance**:

**we must take the expanded view**  
of computer architecture

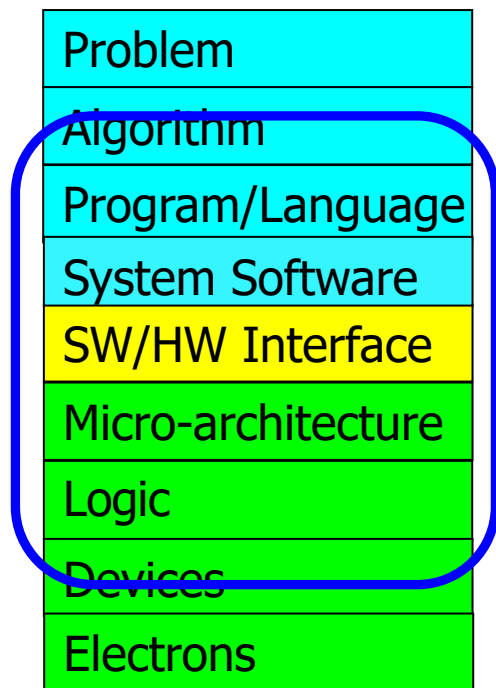


**Co-design across the hierarchy:**  
**Algorithms to devices**

**Specialize as much as possible**  
**within the design goals**

# Current Research Mission & Major Topics

## Build fundamentally better architectures



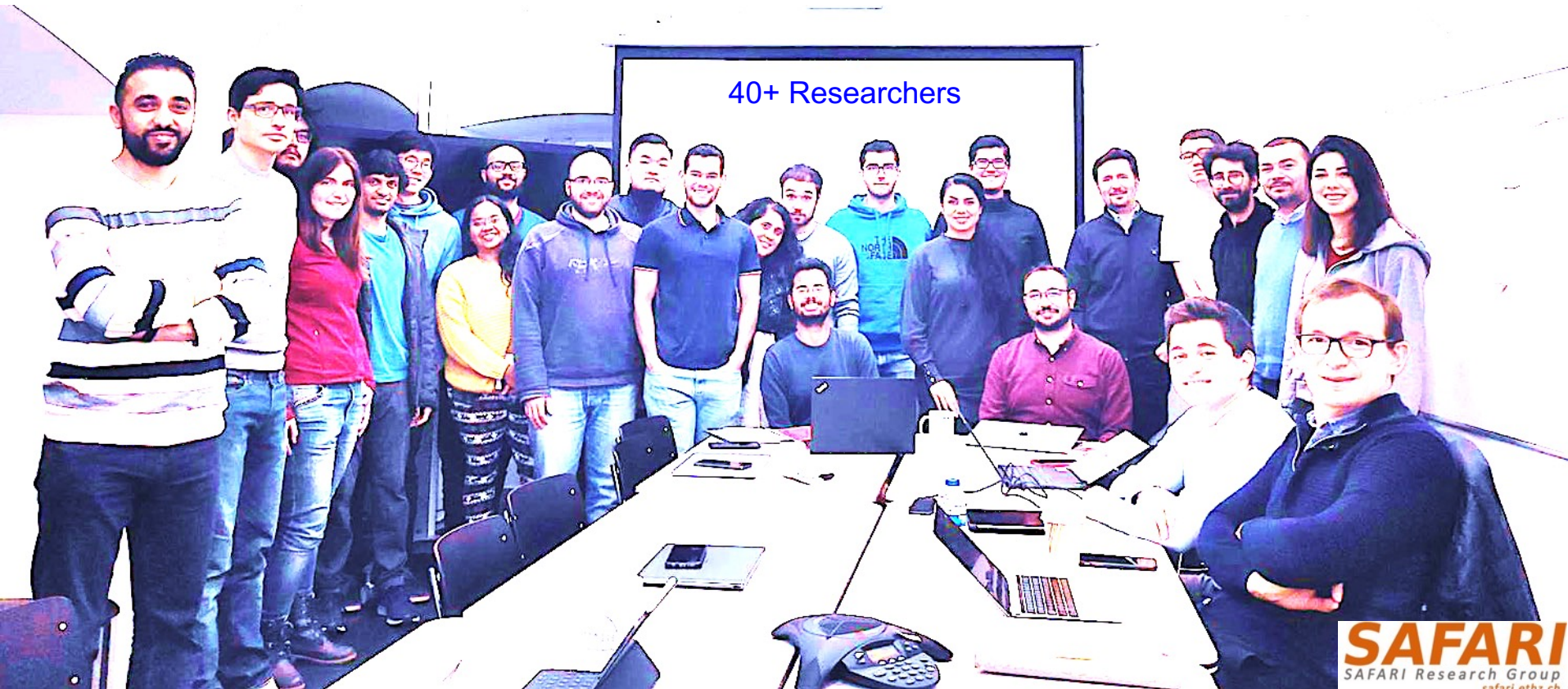
**Broad research  
spanning apps, systems, logic  
with architecture at the center**

- Data-centric arch. for low energy & high perf.
  - Proc. in Mem/DRAM, NVM, unified mem/storage
- Low-latency & predictable architectures
  - Low-latency, low-energy yet low-cost memory
  - QoS-aware and predictable memory systems
- Fundamentally secure/reliable/safe arch.
  - Tolerating all bit flips; patchable HW; secure mem
- Architectures for ML/AI/Genomics/Health/Med
  - Algorithm/arch./logic co-design; full heterogeneity
- Data-driven and data-aware architectures
  - ML/AI-driven architectural controllers and design
  - Expressive memory and expressive systems

# Onur Mutlu's SAFARI Research Group

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

<https://safari.ethz.ch/safari-newsletter-april-2020/>



**SAFARI**  
SAFARI Research Group  
[safari.ethz.ch](https://safari.ethz.ch)

Think BIG, Aim HIGH!

**SAFARI**

<https://safari.ethz.ch>

# SAFARI Newsletter April 2020 Edition

---

- <https://safari.ethz.ch/safari-newsletter-april-2020/>



[View in your browser](#)

*Think Big, Aim High*



Dear SAFARI friends,

2019 and the first three months of 2020 have been very positive eventful times for SAFARI.



# SAFARI Newsletter January 2021 Edition

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



**SAFARI**  
SAFARI Research Group

Newsletter  
January 2021

*Think Big, Aim High, and  
Have a Wonderful 2021!*



Dear SAFARI friends,

Happy New Year! We are excited to share our group highlights with you in this second edition of the SAFARI newsletter (You can find the first edition from April 2020 [here](#)). 2020 has

# SAFARI Newsletter December 2021 Edition

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

**SAFARI**  
SAFARI Research Group

*Think Big, Aim High*

**ETH** zürich



View in your browser  
December 2021





# 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/>

# 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.

📍 ETH Zurich and Carnegie Mellon ... 🔗 <https://safari.ethz.ch/> ✉ [omutlu@gmail.com](mailto:omutlu@gmail.com)

[🏠 Overview](#) [💻 Repositories 55](#) [📦 Packages](#) [👤 People 40](#) [👥 Teams 1](#) [📁 Projects](#) [⚙ Settings](#)

### Pinned

Customize your pins

**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++ ☆ 250 🍴 130

**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 ☆ 18 🍴 8

**DAMOV** Public ⋮

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 processin...

● C++ ☆ 12 🍴 1

### 📁 Repositories

Type ▾ Language ▾ Sort ▾ New

**Pythia**

A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning.

● C++ ☆ 0 🍴 1 🔄 0 📄 0 Updated yesterday

**BurstLink**

☆ 0 🍴 0 🔄 0 📄 0 Updated 21 days ago

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

445

# SAFARI PhD and Post-Doc Alumni

---

- <https://safari.ethz.ch/safari-alumni/>
- Minesh Patel (ETH Zurich), MICRO 2020 and DSN 2020 Best Paper Awards; ISCA Hall of Fame 2021
- Damla Senol Cali (Bionano Genomics), SRC TECHCON 2019 Best Student Presentation Award
- Nastaran Hajinazar (ETH Zurich)
- Gagandeep Singh (ETH Zurich), FPL 2020 Best Paper Award Finalist
- Amirali Boroumand (Stanford Univ → Google), SRC TECHCON 2018 Best Student Presentation Award
- Jeremie Kim (ETH Zurich), 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 Ausavarungrun (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)

# Principle: Teaching and Research

---

...

Teaching drives Research

Research drives Teaching

...

# Principle: Learning and Scholarship

---

Focus on  
learning and scholarship

# Principle: Insight and Ideas

---

Focus on Insight

Encourage New Ideas

# Principle: Learning and Scholarship

---

The quality of your work  
defines your impact



# Principle: Good Mindset, Goals & Focus

---

You can make a  
good impact  
on the world

# Research & Teaching: Some Overview Talks

---

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

## ■ Future Computing Architectures

- [https://www.youtube.com/watch?v=kgiZISOcGFM&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=1](https://www.youtube.com/watch?v=kgiZISOcGFM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=1)

## ■ Enabling In-Memory Computation

- [https://www.youtube.com/watch?v=njX\\_14584Jw&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=16](https://www.youtube.com/watch?v=njX_14584Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=16)

## ■ Accelerating Genome Analysis

- [https://www.youtube.com/watch?v=r7sn41IH-4A&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=41](https://www.youtube.com/watch?v=r7sn41IH-4A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=41)

## ■ Rethinking Memory System Design

- [https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=3](https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=3)

## ■ Intelligent Architectures for Intelligent Machines

- [https://www.youtube.com/watch?v=c6\\_LgzuNdkw&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=25](https://www.youtube.com/watch?v=c6_LgzuNdkw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=25)

## ■ The Story of RowHammer

- [https://www.youtube.com/watch?v=sgd7PHQQ1AI&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=39](https://www.youtube.com/watch?v=sgd7PHQQ1AI&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=39)

# Online Courses & Lectures

---

## ■ **First Computer Architecture & Digital Design Course**

- ❑ Digital Design and Computer Architecture
- ❑ **Spring 2021 Livestream** Edition:  
[https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi\\_uej3aY39YB5pfW4SJ7LIN](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN)

## ■ **Advanced Computer Architecture Course**

- ❑ Computer Architecture
- ❑ **Fall 2021 Livestream** Edition:  
<https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN>
- ❑ **Fall 2020** Edition:  
[https://www.youtube.com/watch?v=4yfkM\\_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF](https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF)



Onur Mutlu Lectures

16.9K subscribers

CUSTOMIZE CHANNEL

MANAGE VIDEOS

HOME

VIDEOS

PLAYLISTS

COMMUNITY

CHANNELS

ABOUT



Popular uploads ▶ PLAY ALL

How Computers Work (from the ground up)

1:33:25

Digital Design & Computer Architecture: Lecture 1: Introduction and Basics

49K views • 1 year ago

Computer Architecture - Lecture 1: Introduction and...

36K views • 3 years ago

Computer Architecture - Lecture 1: Introduction and...

31K views • 1 year ago

Computer Architecture - Lecture 1: Introduction and...

30K views • 8 months ago

Design of Digital Circuits - Lecture 1: Introduction and...

22K views • 2 years ago

Computer Architecture - Lecture 2: Fundamentals,...

17K views • 3 years ago

### First Course in Computer Architecture & Digital Design 2021-2013

Livestream - Digital Design and Computer Architecture - ETH...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Digital Design & Computer Architecture - ETH Zürich...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Design of Digital Circuits - ETH Zürich - Spring 2019

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Design of Digital Circuits - ETH Zürich - Spring 2018

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Digital Circuits and Computer Architecture - ETH Zurich - ...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Spring 2015 -- Computer Architecture Lectures -- ...

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

### Advanced Computer Architecture Courses 2020-2012

Computer Architecture - ETH Zürich - Fall 2020

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Computer Architecture - ETH Zürich - Fall 2019

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Computer Architecture - ETH Zürich - Fall 2018

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Computer Architecture - ETH Zürich - Fall 2017

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Fall 2015 - 740 Computer Architecture

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

Fall 2013 - 740 Computer Architecture - Carnegie Mellon

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

### Special Courses on Memory Systems

Memory Technology Lectures

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Champéry Winter School 2020 - Memory Systems and Memory...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Perugia NIPS Summer School 2019

Onur Mutlu Lectures

VIEW FULL PLAYLIST

SAMOS Tutorial 2019 - Memory Systems

Onur Mutlu Lectures

VIEW FULL PLAYLIST

TU Wien 2019 - Memory Systems and Memory-Centric...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

ACACES 2018 Lectures -- Memory Systems and Memory...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Research Talks

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

SAFARI

# DDCA (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

<https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=schedule>

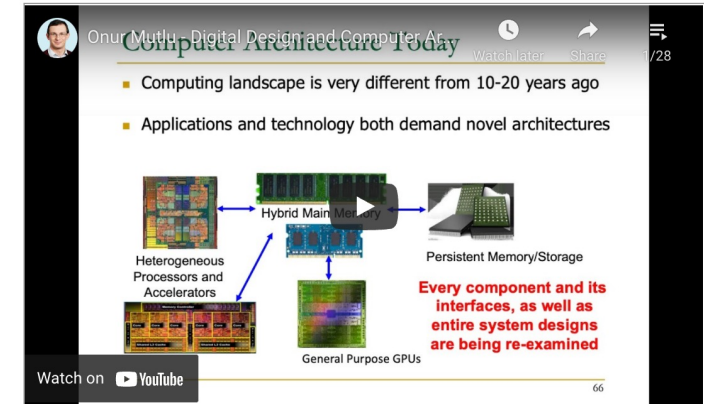
[https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi\\_uej3aY39YB5pfW4SJ7LIN](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN)

## Bachelor's course

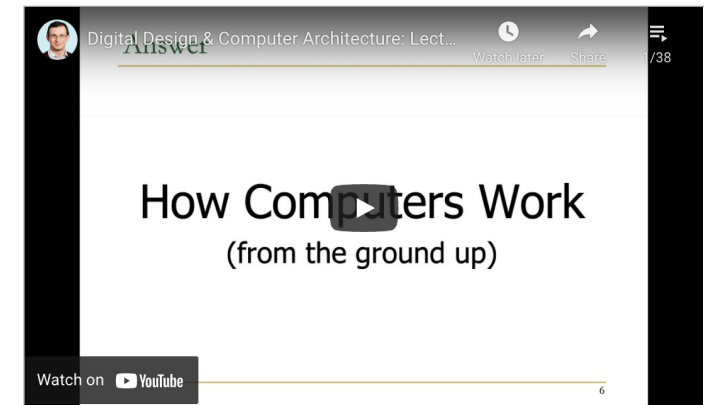
- ❑ 2<sup>nd</sup> semester at ETH Zurich
- ❑ Rigorous introduction into "How Computers Work"
- ❑ Digital Design/Logic
- ❑ Computer Architecture
- ❑ 10 FPGA Lab Assignments

## Lecture Video Playlist on YouTube

Livestream Lecture Playlist



Recorded Lecture Playlist



## Spring 2021 Lectures/Schedule

| Week | Date       | Livestream   | Lecture                                                               | Readings                           | Lab | HW |
|------|------------|--------------|-----------------------------------------------------------------------|------------------------------------|-----|----|
| W1   | 25.02 Thu. | YouTube Live | L1: Introduction and Basics<br>02:00 (PDF) 02:00 (PPT)                | Required<br>Suggested<br>Mentioned |     |    |
|      | 26.02 Fri. | YouTube Live | L2a: Tradeoffs, Metrics, Mindset<br>02:00 (PDF) 02:00 (PPT)           | Required                           |     |    |
|      |            |              | L2b: Mysteries in Computer Architecture<br>02:00 (PDF) 02:00 (PPT)    | Required<br>Mentioned              |     |    |
| W2   | 04.03 Thu. | YouTube Live | L3a: Mysteries in Computer Architecture II<br>02:00 (PDF) 02:00 (PPT) | Required<br>Suggested<br>Mentioned |     |    |

# Comp Arch (Fall 2020)

- <https://safari.ethz.ch/architecture/fall2020/doku.php?id=schedule>
- <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



Computer Architecture - Fall 2020

Search

Recent Changes Media Manager Sitemap

Trace: start schedule

Home

Announcements

Materials

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

Resources

- Computer Architecture FS19: Course Webpage
- Computer Architecture FS19: Lecture Videos
- Digitaltechnik SS20: Course Webpage
- Digitaltechnik SS20: Lecture Videos
- Moodle
- Piazza (Q&A)
- HotCRP
- Verilog Practice Website (HDLBits)

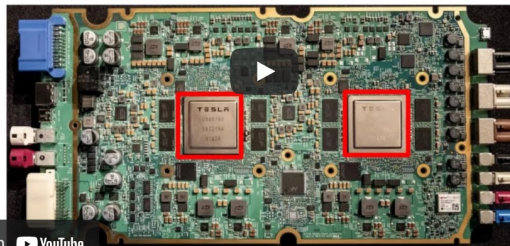
Lecture Video Playlist on YouTube

Lecture Playlist



Computer Architecture - Lecture: Introduction

- ML accelerator: 260 mm<sup>2</sup>, 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.



Watch on YouTube

<https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN>

48

## Fall 2020 Lectures & Schedule

| Week | Date          | Lecture                                                                                                                                                     | Readings               | Lab          | HW          |
|------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|--------------|-------------|
| W1   | 17.09<br>Thu. | <b>L1: Introduction and Basics</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>                       | Described<br>Suggested |              | HW 0<br>Out |
|      |               | <b>L2a: Memory Performance Attacks</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>                   | Described<br>Suggested | Lab 1<br>Out |             |
|      | 18.09<br>Fri. | <b>L2b: Data Retention and Memory Refresh</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>            | Described<br>Suggested |              |             |
|      |               | <b>L2c: Course Logistics</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>                             |                        |              |             |
| W2   | 24.09<br>Thu. | <b>L3a: Introduction to Genome Sequence Analysis</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>     | Described<br>Suggested |              | HW 1<br>Out |
|      |               | <b>L3b: Memory Systems: Challenges and Opportunities</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a> | Described<br>Suggested |              |             |
|      | 25.09<br>Fri. | <b>L4a: Memory Systems: Solution Directions</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>          | Described<br>Suggested |              |             |
|      |               | <b>L4b: RowHammer</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>                                    | Described<br>Suggested |              |             |
| W3   | 01.10<br>Thu. | <b>L5a: RowHammer in 2020: TRRespass</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>                 | Described<br>Suggested |              |             |
|      |               | <b>L5b: RowHammer in 2020: Revisiting RowHammer</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>      | Described<br>Suggested |              |             |
|      |               | <b>L5c: Secure and Reliable Memory</b><br><a href="#">CORA (PDF)</a> <a href="#">PPT</a><br><a href="#">YouTube</a> <a href="#">Video</a>                   | Described              |              |             |



Search

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

---

[Trace](#) • [readings](#) • [start](#) • [schedule](#)

---

Home
schedule

## Lecture Video Playlist on YouTube

[Livestream Lecture Playlist](#)

**Resources**

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

**Materials**

- Computer Architecture FS20: Course Webpage
- Digitaltechnik SS21: Course Webpage
- Digitaltechnik SS21: Lecture Videos
- Moodle
- HotCRP
- Verilog Practice Website (HDLBits)

Watch on

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

Recorded Lecture Playlist
1/51

Watch on

<https://www.youtube.com/watch?v=Ucp0TTmvgQE>

## Fall 2021 Lectures & Schedule

| Week | Date       | Livestream                   | Lecture                                                                                                       | Readings            | Lab       | HW       |
|------|------------|------------------------------|---------------------------------------------------------------------------------------------------------------|---------------------|-----------|----------|
| W1   | 30.09 Thu. | <a href="#">YouTube Live</a> | <b>L1: Introduction and Basics</b><br><a href="#">aaa(PDF)</a> <a href="#">aaa(PPT)</a>                       | Required Mentioned  | Lab 1 Out | HW 0 Out |
|      | 01.10 Fri. | <a href="#">YouTube Live</a> | <b>L2: Trends, Tradeoffs and Design Fundamentals</b><br><a href="#">aaa(PDF)</a> <a href="#">aaa(PPT)</a>     | Required Mentioned  |           |          |
| W2   | 07.10 Thu. | <a href="#">YouTube Live</a> | <b>L3a: Memory Systems: Challenges and Opportunities</b><br><a href="#">aaa(PDF)</a> <a href="#">aaa(PPT)</a> | Described Suggested |           | HW 1 Out |
|      |            |                              | <b>L3b: Course Info &amp; Logistics</b><br><a href="#">aaa(PDF)</a> <a href="#">aaa(PPT)</a>                  |                     |           |          |
|      |            |                              | <b>L3c: Memory Performance Attacks</b><br><a href="#">aaa(PDF)</a> <a href="#">aaa(PPT)</a>                   |                     |           |          |
|      | 08.10 Fri. | <a href="#">YouTube Live</a> | <b>L4a: Memory Performance Attacks</b><br><a href="#">aaa(PDF)</a> <a href="#">aaa(PPT)</a>                   | Described Suggested | Lab 2 Out |          |
|      |            |                              | <b>L4b: Data Retention and Memory Refresh</b><br><a href="#">aaa(PDF)</a> <a href="#">aaa(PPT)</a>            | Described Suggested |           |          |
|      |            |                              | <b>L4c: RowHammer</b><br><a href="#">aaa(PDF)</a> <a href="#">aaa(PPT)</a>                                    | Described Suggested |           |          |

- <https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule>
- **Youtube Livestream:**
  - [https://www.youtube.com/watch?v=4yfkM\\_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF](https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF)
- **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



# Seminar (Spring'21)

■ [https://safari.ethz.ch/architecture\\_seminar/spring2021/doku.php?id=schedule](https://safari.ethz.ch/architecture_seminar/spring2021/doku.php?id=schedule)

■ [https://www.youtube.com/watch?v=t3m93ZpLOyw&list=PL5Q2soXY2Zi\\_awYdjmWVIUegsbY7TPGW4](https://www.youtube.com/watch?v=t3m93ZpLOyw&list=PL5Q2soXY2Zi_awYdjmWVIUegsbY7TPGW4)

■ Critical analysis course

- Taken by Bachelor's/Masters/PhD students
- Cutting-edge research topics + fundamentals in Computer Architecture
- 20+ research papers, presentations, analyses



Seminar in Computer Architecture - Spring 2021

Recent Changes

Media Manager

Sitemap

Trace: [start](#) - [schedule](#)

Home

Materials

- Announcements
- Lectures/Schedule**
- Lecture Buzzwords
- Readings
- Sessions
- Papers
- Synthesis Report
- Homework

Past Course Materials

- Fall 2020
- Spring 2020
- Fall 2019
- Spring 2019

Resources

Computer Architecture

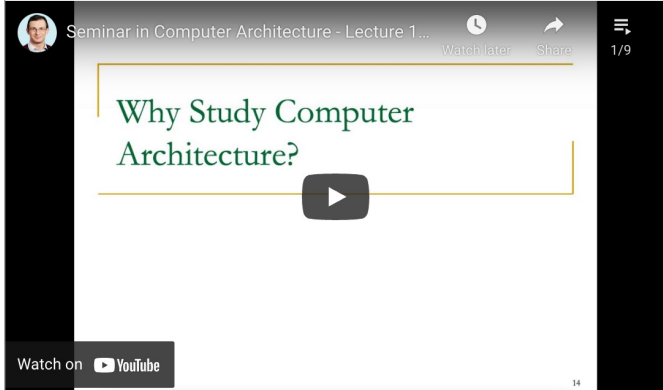
- Fall 2020
- Fall 2020: Lecture Videos
- Fall 2019
- Fall 2019: Lecture Videos
- Fall 2018
- Fall 2018: Lecture Videos

Digital Design and Computer Architecture

- Spring 2020
- Spring 2020: Lecture Videos
- Spring 2019
- Spring 2019: Lecture Videos

Lecture Video Playlist on YouTube

Lecture Playlist



Watch on 

Spring 2021 Lectures/Schedule

| Week | Date       | Livestream                                                                            | Lecture                                                                                                                                                                       | Readings  | Assignments |
|------|------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-------------|
| W1   | 25.02 Thu. |    | <b>L1a: Introduction and Basics</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                                                                | Suggested |             |
|      |            |                                                                                       | Optional Lecture: Design Fundamentals<br><a href="#">PDF</a> <a href="#">PPT</a>                                                                                              |           |             |
|      |            |                                                                                       | <b>L1b: Course Logistics</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                                                                       | Suggested |             |
| W2   | 04.03 Thu. |    | <b>L2: Example Review: RowClone</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                                                                | Suggested |             |
| W3   | 11.03 Thu. |    | <b>L3: Example Review: Memory Channel Partitioning</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                                             | Suggested |             |
| W4   | 18.03 Thu. |    | <b>L4: Example Review: GateKeeper</b><br><a href="#">PDF</a> <a href="#">PPT</a>                                                                                              | Suggested |             |
| W5   | 25.03 Thu. |                                                                                       | <b>S1.1: Spectre Attacks: Exploiting Speculative Execution, S&amp;P 2019</b><br><a href="#">PPT</a> <a href="#">PDF</a>                                                       | Mentioned |             |
|      |            |  | <b>S1.2: BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows, HPCA 2021</b><br><a href="#">PPT</a> <a href="#">PDF</a>                   |           |             |
| W6   | 01.04 Thu. |                                                                                       | <b>S2.1: D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput, HPCA 2019</b><br><a href="#">PPT</a> <a href="#">PDF</a> |           |             |
|      |            |  | <b>S2.2: ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAMs, MICRO 2019</b><br><a href="#">PPT</a> <a href="#">PDF</a>                                                  | Mentioned |             |
| W7   | 15.04 Thu. |                                                                                       | <b>S3.1: PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture,</b>                                                                      | Mentioned |             |

# Seminar (Current)

- [https://safari.ethz.ch/architecture\\_seminar/fall2021/doku.php?id=schedule](https://safari.ethz.ch/architecture_seminar/fall2021/doku.php?id=schedule)
- **Youtube Livestream:**
  - [https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi\\_7UBNmC9B8Yr5JSwTG9yH4](https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4)
- Critical analysis course
  - Taken by Bachelor's/Masters/PhD students
  - Cutting-edge research topics + fundamentals in Computer Architecture
  - 20+ research papers, presentations, analyses



Seminar in Computer Architecture - Fall 2021

Recent Changes Media Manager Sitemap

Trace: start schedule

Home

Materials

- Announcements
- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Sessions
- Papers
- Synthesis Report
- Homework

Past Course Materials

- Spring 2021
- Fall 2020
- Spring 2020
- Fall 2019
- Spring 2019

Resources

Computer Architecture

- Fall 2021
- Fall 2021: Lecture Videos
- Fall 2020
- Fall 2020: Lecture Videos
- Fall 2019
- Fall 2019: Lecture Videos
- Fall 2018
- Fall 2018: Lecture Videos

Digital Design and Computer Architecture

- Spring 2021
- Spring 2021: Lecture Videos
- Spring 2020
- Spring 2020: Lecture Videos
- Spring 2019
- Spring 2019: Lecture Videos

Lecture Video Playlist on YouTube

Lecture Playlist



Watch on YouTube

Fall 2021 Lectures/Schedule

| Week | Date       | Livestream   | Lecture                                | Readings  | Assignments |
|------|------------|--------------|----------------------------------------|-----------|-------------|
| W1   | 23.09 Thu. | YouTube Live | L1a: Course Logistics                  | Suggested |             |
|      |            |              | L1b: Introduction and Basics           | Suggested |             |
|      |            |              | L1c: Architectural Design Fundamentals | Suggested |             |
|      |            |              |                                        |           |             |
| W2   | 30.09 Thu. | YouTube Live | L2: GateKeeper                         | Suggested |             |
| W3   | 07.10 Thu. | YouTube Live | L3: RowClone (Processing using DRAM)   | Suggested |             |

# Hands-On Projects & Seminars Courses

- [https://safari.ethz.ch/projects\\_and\\_seminars/doku.php](https://safari.ethz.ch/projects_and_seminars/doku.php)



## SAFARI Project & Seminars Courses (Spring 2021)

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

Trace: • [start](#)

### Home

### Projects

- [SoftMC](#)
- [Ramulator](#)
- [Accelerating Genomics](#)
- [Mobile Genomics](#)
- [Processing-in-Memory](#)
- [Heterogeneous Systems](#)
- [SSD Simulator](#)

[start](#)

## SAFARI Projects & Seminars Courses (Spring 2021)

Welcome to the wiki for Project and Seminar courses SAFARI offers.

### Courses we offer:

- Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments
- Designing and Evaluating Memory Systems and Modern Software Workloads with Ramulator
- Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms
- Genome Sequencing on Mobile Devices
- Exploring the Processing-in-Memory Paradigm for Future Computing Systems
- Hands-on Acceleration on Heterogeneous Computing Systems
- Understanding and Designing Modern NAND Flash-Based Solid-State Drives (SSDs) by Building a Practical SSD Simulator

# PIM Course (Current)

## Fall 2021 Edition:

- [https://safari.ethz.ch/projects\\_and\\_seminars/fall2021/doku.php?id=processing\\_in\\_memory](https://safari.ethz.ch/projects_and_seminars/fall2021/doku.php?id=processing_in_memory)

## Youtube Livestream:

- <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

The screenshot shows a YouTube video player interface. The video title is "PIM Review and Open Problems" with a subtitle "Processing in Memory Course: Meeting 1: Ex...". Below the title, the authors are listed: Onur Mutlu<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirun<sup>d</sup>. The affiliations are: SAFARI Research Group, <sup>a</sup>ETH Zürich, <sup>b</sup>Carnegie Mellon University, <sup>c</sup>University of Illinois at Urbana-Champaign, and <sup>d</sup>King Mongkut's University of Technology North Bangkok. The video description mentions that the authors have contributed a book chapter titled "A Modern Primer on Processing in Memory" to the book "Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann", published by Springer in 2021. The video player includes a "Watch on YouTube" button and a link to the PDF: <https://arxiv.org/pdf/1903.03988.pdf>. The video progress bar shows 1/10.

## Fall 2021 Meetings/Schedule

| Week | Date       | Livestream                                                    | Meeting                                                                                                            | Learning Materials                          | Assignments |
|------|------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|---------------------------------------------|-------------|
| W1   | 05.10 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M1: P&amp;S PIM Course Presentation</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                  | Required Materials<br>Recommended Materials | HW 0 Out    |
| W2   | 12.10 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M2: Real-World PIM Architectures</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                     |                                             |             |
| W3   | 19.10 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M3: Real-World PIM Architectures II</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                  |                                             |             |
| W4   | 26.10 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M4: Real-World PIM Architectures III</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                 |                                             |             |
| W5   | 02.11 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M5: Real-World PIM Architectures IV</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                  |                                             |             |
| W6   | 09.11 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M6: End-to-End Framework for Processing-using-Memory</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT) |                                             |             |
| W7   | 16.11 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M7: How to Evaluate Data Movement Bottlenecks</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)        |                                             |             |
| W8   | 23.11 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M8: Programming PIM Architectures</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)                    |                                             |             |
| W9   | 30.11 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M9: Benchmarking and Workload Suitability on PIM</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)     |                                             |             |
| W10  | 07.12 Tue. | <a href="#">You</a> <a href="#">Tube</a> <a href="#">Live</a> | <b>M10: Bit-Serial SIMD Processing using DRAM</b><br><a href="#">PDF</a> (PDF) <a href="#">PPT</a> (PPT)           |                                             |             |



# SAFARI Live Seminars (I)

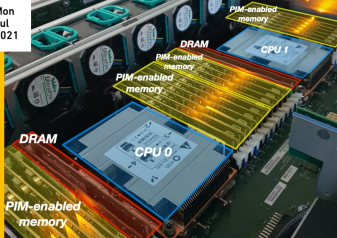
## SAFARI Live Seminars in Computer Architecture

Dr. Juan Gómez Luna, ETH Zurich

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

**SAFARI**  
SAFARI Research Group

**12** Mon Jul 2021

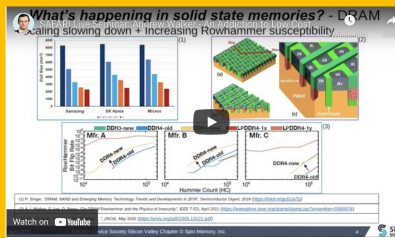
## SAFARI Live Seminars in Computer Architecture

Dr. Andrew Walker, Schiltron Corporation & Nexgen Power Systems

An Addition to Low Cost Per Memory Bit – How to Recognize It and What to Do About It

**SAFARI**  
SAFARI Research Group

**19** Mo Jul 2021

## SAFARI Live Seminars in Computer Architecture

Geraldo F. Oliveira, ETH Zurich

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

**SAFARI**  
SAFARI Research Group

**22** Do Jul 2021



**Near-Data Processing (2/2)**

**UPMEM (2019)** Samsung HBM-PIM

Near-DRAM-banks processing for general-purpose computing

0.9 TOPS compute throughput<sup>1</sup>

**Samsung FIMDRAM (2021)**

Near-DRAM-banks processing for neural networks

1.2 TFLOPS compute throughput<sup>2</sup>

The goal of Near-Data Processing (NDP) is to mitigate data movement

**SAFARI** 7

## SAFARI Live Seminars in Computer Architecture

Gennady Pekhimenko, University of Toronto

Efficient DNN Training at Scale: from Algorithms to Hardware

**SAFARI**  
SAFARI Research Group

**5** Do Aug 2021



**DNN Training vs. Inference**

Step 1 - Forward Pass (makes a prediction)

Step 2 - Backward Pass (calculates error gradients)

Generated in the forward pass

Used in the backward pass

DNN training requires stashing feature maps for the backward pass (not required in inference)

## SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Power Management Mechanisms in Modern Microprocessors and Their Security Implications

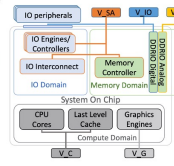
**SAFARI**  
SAFARI Research Group

**16** Mo Aug 2021



**Overview of a Modern SoC Architecture**

- 3 domains in modern thermally-constrained mobile SoC: Compute, Memory, IO
- Several voltage sources exist, and some of them are shared between domains
- IO controllers and engines, IO interconnect, memory controller, and DDRIO typically each has an independent clock



## SAFARI Live Seminars in Computer Architecture

Ataberk Olgun, TOBB & ETH Zurich

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

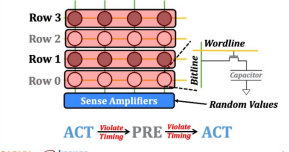
**SAFARI**  
SAFARI Research Group

**15** Mi Sep 2021



**Using QUAC to Generate Random Values**

Use QUAC to activate DRAM rows that are initialized with conflicting data (e.g., two '1's and two '0's) to generate random values



## SAFARI Live Seminars in Computer Architecture

Minesh Patel, ETH Zurich

Enabling Effective Error Mitigation in Memory Chips That Use On-Die ECCs

**SAFARI**  
SAFARI Research Group

**21** Tue Sep 2021



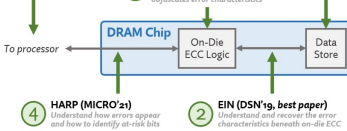
**Position Paper (Ongoing)** Arguing for increased transparency of DRAM reliability characteristics

**REAPER (ISCA'17)** Understand the basic properties of DRAM data-retention errors

**BEER (MICRO'20, best paper)** Determine exactly how on-die ECCs adjust error characteristics

**HARP (MICRO'21)** Understand how errors appear and how to identify at-risk bits

**EIN (DSN'19, best paper)** Understand and recover the error characteristics beneath on-die ECC



## SAFARI Live Seminars in Computer Architecture

Christina Giannoula, National Technical University of Athens

Efficient Synchronization Support for Near-Data-Processing Architectures

**SAFARI**  
SAFARI Research Group

**27** Mo Sep 2021



**NDP Synchronization Solution Space**

Shared Memory

Message-passing

Hardware Cache Coherence

Remote Atomics

Specialized Hardware Support

Software-based Schemes

Specialized Hardware Support

NDP Systems: SynCron [HPCA'21]

## SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Security Implications of Power Management Mechanisms in Modern Processors, Current Studies and Future Trends

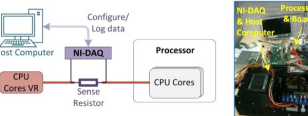
**SAFARI**  
SAFARI Research Group

**4** Mo Okt 2021



**Experimental Methodology**

- We experimentally study three modern Intel processors
- Haswell, Coffee Lake, and Cannon Lake
- We measure voltage and current using a Data Acquisition card (NI-DAQ)



<https://safari.ethz.ch/safari-seminar-series/>

# SAFARI Live Seminars (II)

SAFARI Live Seminars in Computer Architecture

Nastaran Hajinazar, ETH Zurich

Data-Centric and Data-Aware Frameworks for Fundamentally Efficient Data Handling in Modern Computing Systems

**SAFARI**  
SAFARI Research Group

**27** Wed Oct 2021

**Overview of Our Approach**

Data and the **efficient computation of data** should be the ultimate priority of the system

- **Data-Centric Architectures**
  - Enable computation with minimal data movement
  - Compute where data resides
- **Data-Aware Architectures**
  - Understand what they can do with and to each piece of data
  - Make use of different properties of data to improve performance, efficiency, etc.

SAFARI 15



## SAFARI Live Seminar: Nastaran Hajinazar 27 Oct 2021

Posted on October 1, 2021 by ewent

Join us for our [SAFARI Live Seminar](#) with [Nastaran Hajinazar](#).

Wednesday, October 27 at 7:00 pm Zurich time (CEST)

SAFARI Live Seminars in Computer Architecture

Damla Senol Cali, Bionano Genomics

Accelerating Genome Sequence Analysis via Efficient Hardware/Algorithm Co-Design

**SAFARI**  
SAFARI Research Group

**7** Sun Nov 2021

**Our Goal & Approach**

- **Our Goal:**  
Accelerating genome sequence analysis by **efficient hardware/algorithm co-design**
- **Our Approach:**
  - (1) Analyze the **multiple steps** and the **associated tools** in the genome sequence analysis pipeline,
  - (2) Expose the **tradeoffs** between accuracy, performance, memory usage and scalability, and
  - (3) Co-design **fast and efficient algorithms** along with **scalable and energy-efficient customized hardware accelerators** for the key bottleneck steps of the pipeline

Damla Senol Cali SAFARI 10



## SAFARI Live Seminar: Damla Senol Cali 07 Nov 2021

Posted on October 18, 2021 by ewent

Join us for our [SAFARI Live Seminar](#) with [Damla Senol Cali](#).

Sunday, November 07 at 6:00 pm Zurich time (CEST)

SAFARI Live Seminars in Computer Architecture

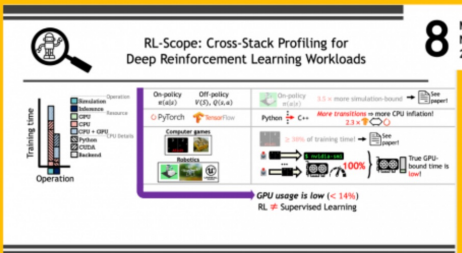
Gennady Pekhimenko, University of Toronto

Machine Learning Tools in Action

**ETH zürich**  
**SAFARI**  
SAFARI Research Group

**8** Mo Nov 2021

**RL-Scope: Cross-Stack Profiling for Deep Reinforcement Learning Workloads**



## SAFARI Live Seminar: Gennady Pekhimenko 08 Nov 2021

Posted on November 1, 2021 by ewent

Join us for our [SAFARI Live Seminar](#) with [Gennady Pekhimenko](#).

Monday, November 08 at 4:00 pm Zurich time (CET)

SAFARI Live Seminars in Computer Architecture

Serghei Mangul, Mangul Lab, USC

Opportunities and challenges of computational data-driven immunology

**SAFARI**  
SAFARI Research Group  
**ETH zürich**

**11** Thu Nov 2021

Opportunities and challenges of computational data-driven immunology



Serghei Mangul, Ph.D  
Assistant Professor,  
University of Southern California  
<https://mangul-lab.usc.edu/lab/>

## SAFARI Live Seminar: Serghei Mangul 11 Nov 2021

Posted on November 5, 2021 by ewent

Join us for our [SAFARI Live Seminar](#) with [Serghei Mangul](#).

Thursday, November 11 at 11:00 am Zurich time (CET), ETH Zentrum ETZ K91

[https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi\\_tOTAYm--dYByNPL7JhwR9&index=1](https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=1)

# Open-Source Artifacts

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



# 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.

📍 ETH Zurich and Carnegie Mellon ... 🔗 <https://safari.ethz.ch/> ✉ [omutlu@gmail.com](mailto:omutlu@gmail.com)

[🏠 Overview](#) [💻 Repositories 55](#) [📦 Packages](#) [👤 People 40](#) [👥 Teams 1](#) [📁 Projects](#) [⚙ Settings](#)

### Pinned

Customize your pins

**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++ ☆ 250 🍴 130

**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 ☆ 18 🍴 8

**DAMOV** Public ⋮

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 processin...

● C++ ☆ 12 🍴 1

### 📁 Repositories

Type ▾ Language ▾ Sort ▾ New

**Pythia**

A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning.

● C++ ☆ 0 🍴 1 🔄 0 📄 0 Updated yesterday

**BurstLink**

☆ 0 🍴 0 🔄 0 📄 0 Updated 21 days ago

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

465

Repositories 45 Packages People 12 Projects

Find a repository... Type Language Sort

## COVIDHunter

COVIDHunter 🦠: An accurate and flexible COVID-19 outbreak simulation model that forecasts the strength of future mitigation measures and the numbers of cases, hospitalizations, and deaths for a given day, while considering the potential effect of environmental conditions. Described by Alser et al. (preliminary version at <https://arxiv.org/abs/2003.09475>...

simulation epidemiology covid-19 covid-19-data covid-19-tracker  
 reproduction-number covidhunter

Swift MIT 1 5 0 0 Updated 9 hours ago

## SNP-Selective-Hiding

An optimization-based mechanism 🧠 to selectively hide the minimum number of overlapping SNPs among the family members 👤 who participated in the genomic studies (i.e. GWAS). Our goal is to distort the dependencies among the family members in the original database for achieving better privacy without significantly degrading the data utility.

gwas genomics data-privacy differential-privacy  
 genomic-data-analysis laplace-distribution genomic-privacy

MATLAB 0 0 0 0 Updated 10 hours ago

## SneakySnake

SneakySnake 🐍 is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude) expedites sequence alignment calculation for both short and long reads. Described in the Bioinformatics (2020) by Alser et al. <https://arxiv.org/abs/2003.09475>...

fpga gpu smith-waterman needleman-wunsch  
 sequence-alignment long-reads minimap2

VHDL GPL-3.0 6 31 0 1 Updated on May 12

## ramulator

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 IEEE CAL 2015 paper by Kim et al. at [http://users.ece.cmu.edu/~omutlu/pub/ramulator\\_dram\\_simulator-ieee-cal15.pdf](http://users.ece.cmu.edu/~omutlu/pub/ramulator_dram_simulator-ieee-cal15.pdf)

C++ MIT 121 237 47 4 Updated on May 11

### Top languages

C++ C C# AGS Script  
 VHDL

### Most used topics

dram reliability  
 error-correcting-codes  
 experimental-data  
 pre-alignment-filtering

### People

12 >



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

# An Interview on Research and Education

---

- **Computing Research and Education (@ ISCA 2019)**
  - [https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi\\_4oP9LdL3cc8G6NIjD2Ydz](https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi_4oP9LdL3cc8G6NIjD2Ydz)
  
- **Maurice Wilkes Award Speech (10 minutes)**
  - [https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJl&index=15](https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=15)

# More Thoughts and Suggestions

---

- Onur Mutlu,  
["Some Reflections \(on DRAM\)"](#)  
*Award Speech for [ACM SIGARCH Maurice Wilkes Award](#), at the **ISCA** Awards Ceremony, Phoenix, AZ, USA, 25 June 2019.*  
[\[Slides \(pptx\) \(pdf\)\]](#)  
[\[Video of Award Acceptance Speech \(Youtube; 10 minutes\) \(Youku; 13 minutes\)\]](#)  
[\[Video of Interview after Award Acceptance \(Youtube; 1 hour 6 minutes\) \(Youku; 1 hour 6 minutes\)\]](#)  
[\[News Article on "ACM SIGARCH Maurice Wilkes Award goes to Prof. Onur Mutlu"\]](#)
  
- Onur Mutlu,  
["How to Build an Impactful Research Group"](#)  
*[57th Design Automation Conference Early Career Workshop \(\*\*DAC\*\*\)](#), Virtual, 19 July 2020.*  
[\[Slides \(pptx\) \(pdf\)\]](#)

# More Thoughts and Suggestions (II)

---

- Onur Mutlu,  
**"Computer Architecture: Why Is It So Important and Exciting Today?"**  
Invited Lecture at *Izmir Institute of Technology (IYTE)*, Virtual, 16 October 2020.  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (2 hours 12 minutes)]
  
- Onur Mutlu,  
**"Applying to Graduate School & Doing Impactful Research"**  
Invited Panel Talk at *the 3rd Undergraduate Mentoring Workshop, held with the 48th International Symposium on Computer Architecture (ISCA)*, Virtual, 18 June 2021.  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (50 minutes)]

# A Talk on Impactful Research & Teaching



The video player shows a presentation slide with the title "Applying to Grad School & Doing Impactful Research" in a green serif font, enclosed in a thin gold border. Below the title, the speaker's name "Onur Mutlu" is listed, followed by his email "omutlu@gmail.com" and a URL "https://people.inf.ethz.ch/omutlu". The date "13 June 2020" and the event "Undergraduate Architecture Mentoring Workshop @ ISCA 2021" are also displayed. At the bottom of the slide, the logos for "SAFARI", "ETH zürich", and "Carnegie Mellon" are shown. The video player interface includes a progress bar at 0:27 / 50:31, a small video feed of the speaker in the top right corner, and a bottom bar with engagement metrics (74 likes, 1 comment), share options, and buttons for "ANALYTICS" and "EDIT VIDEO".

Applying to Grad School  
& Doing Impactful Research

Onur Mutlu  
[omutlu@gmail.com](mailto:omutlu@gmail.com)  
<https://people.inf.ethz.ch/omutlu>  
13 June 2020  
Undergraduate Architecture Mentoring Workshop @ ISCA 2021

SAFARI ETH zürich Carnegie Mellon

Arch. Mentoring Workshop @ISCA'21 - Applying to Grad School & Doing Impactful Research - Onur Mutlu  
1,563 views • Premiered Jun 16, 2021

Onur Mutlu Lectures  
17.2K subscribers

Panel talk at Undergraduate Architecture Mentoring Workshop at ISCA 2021  
(<https://sites.google.com/wisc.edu/uar...>)

# An Interview on Computing Futures



Interview with Onur Mutlu @ ISCA 2019 on computing research & education (after Maurice Wilkes Award)

6,749 views • Oct 19, 2019

👍 195 🗨️ 0 ➦ SHARE ➦ SAVE ...



**Onur Mutlu Lectures**  
19.1K subscribers

ANALYTICS

EDIT VIDEO



# Papers, Talks, Videos, Artifacts

---

- All are available at

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

<http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en>

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

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

# Fundamental Thinking

# Historical: Opportunities at the Bottom

---

## There's Plenty of Room at the Bottom

---

From Wikipedia, the free encyclopedia

**"There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics"** was a lecture given by [physicist Richard Feynman](#) at the annual [American Physical Society](#) meeting at [Caltech](#) on December 29, 1959.<sup>[1]</sup> Feynman considered the possibility of direct manipulation of individual atoms as a more powerful form of synthetic chemistry than those used at the time. Although versions of the talk were reprinted in a few popular magazines, it went largely unnoticed and did not inspire the conceptual beginnings of the field. Beginning in the 1980s, nanotechnology advocates cited it to establish the scientific credibility of their work.

# Historical: Opportunities at the Bottom (II)

---

## There's Plenty of Room at the Bottom

---

From Wikipedia, the free encyclopedia

Feynman considered some ramifications of a general ability to manipulate matter on an atomic scale. He was particularly interested in the possibilities of denser computer circuitry, and microscopes that could see things much smaller than is possible with scanning electron microscopes. These ideas were later realized by the use of the scanning tunneling microscope, the atomic force microscope and other examples of scanning probe microscopy and storage systems such as Millipede, created by researchers at IBM.

Feynman also suggested that it should be possible, in principle, to make nanoscale machines that "arrange the atoms the way we want", and do chemical synthesis by mechanical manipulation.

He also presented the possibility of "swallowing the doctor", an idea that he credited in the essay to his friend and graduate student Albert Hibbs. This concept involved building a tiny, swallowable surgical robot.

# Historical: Opportunities at the Top

## REVIEW

### There's plenty of room at the Top: What will drive computer performance after Moore's law?

 Charles E. Leiserson<sup>1</sup>,  Neil C. Thompson<sup>1,2,\*</sup>,  Joel S. Emer<sup>1,3</sup>,  Bradley C. Kuszmaul<sup>1,†</sup>, Butler W. Lampson<sup>1,4</sup>,  ...

+ See all authors and affiliations

*Science* 05 Jun 2020:  
Vol. 368, Issue 6495, eaam9744  
DOI: 10.1126/science.aam9744

Much of the improvement in computer performance comes from decades of miniaturization of computer components, a trend that was foreseen by the Nobel Prize-winning physicist Richard Feynman in his 1959 address, “There’s Plenty of Room at the Bottom,” to the American Physical Society. In 1975, Intel founder Gordon Moore predicted the regularity of this miniaturization trend, now called Moore’s law, which, until recently, doubled the number of transistors on computer chips every 2 years.

Unfortunately, semiconductor miniaturization is running out of steam as a viable way to grow computer performance—there isn’t much more room at the “Bottom.” If growth in computing power stalls, practically all industries will face challenges to their productivity. Nevertheless, opportunities for growth in computing performance will still be available, especially at the “Top” of the computing-technology stack: software, algorithms, and hardware architecture.

# Axiom, Revisited

---

There **is** plenty of room both at the top and at the bottom

but **much more so**

when you

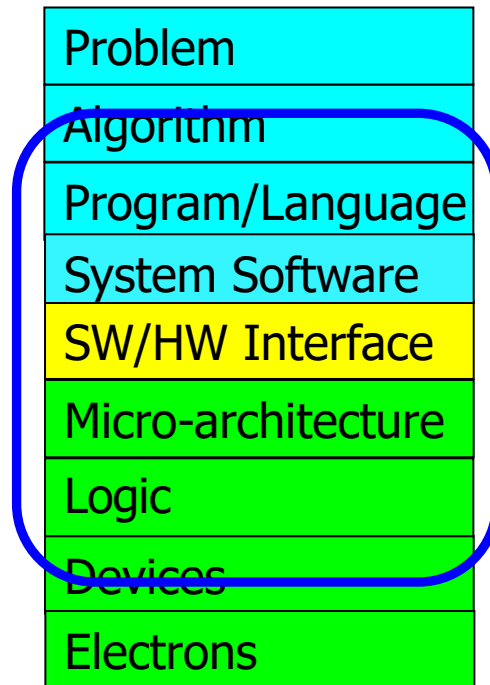
**communicate well between and optimize across**

**the top and the bottom**

# Hence the Expanded View

---

**Computer Architecture  
(expanded view)**





**Data-centric**

**Data-driven**

**Data-aware**

# End of Backup Slides