

Reinventing Computing & Storage

Onur Mutlu

omutlu@gmail.com

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

7 April 2023

OlympusMons Award Keynote Speech

SAFARI

ETH zürich

Carnegie Mellon

Computing

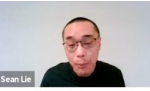
is Bottlenecked by Data

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

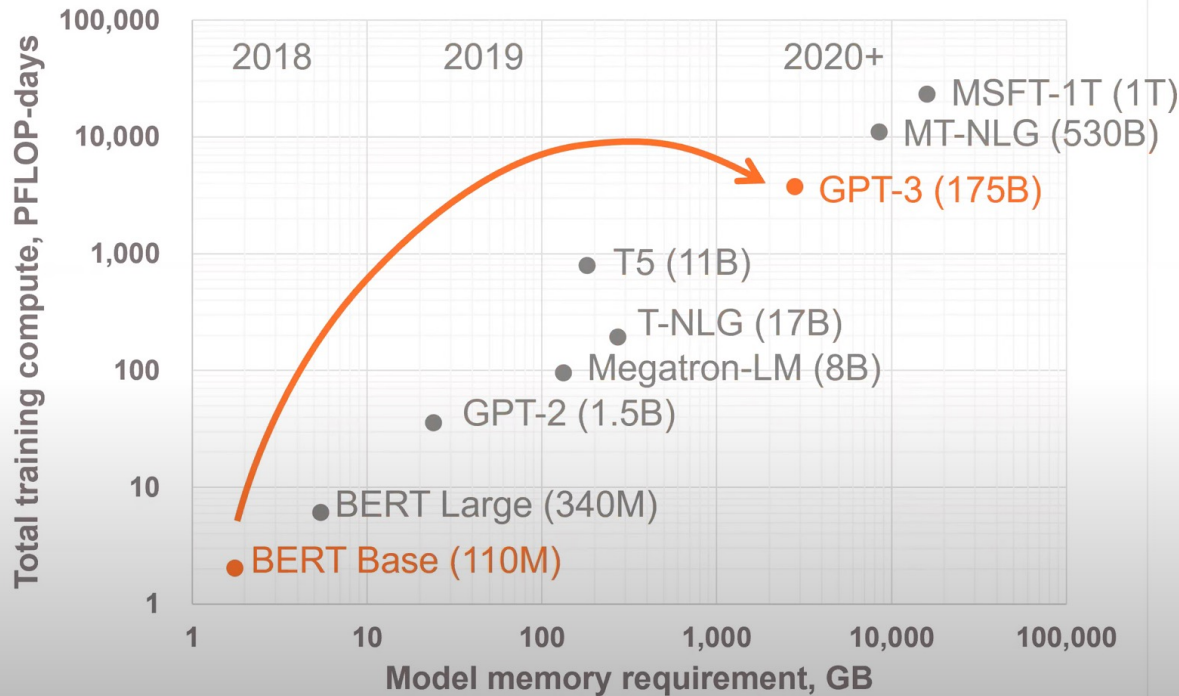
- Important workloads are all data intensive
- They require rapid and efficient processing of large amounts of data
- Data is increasing
 - We can generate more than we can process
 - We need to perform more sophisticated analyses on more data

Huge Demand for Performance & Efficiency

Exponential Growth of Neural Networks



Memory and compute requirements



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

Tomorrow, **multi-trillion**
parameter models

Data is Key for Future Workloads



In-memory Databases

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



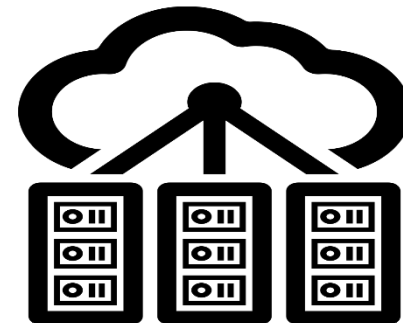
In-Memory Data Analytics

[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



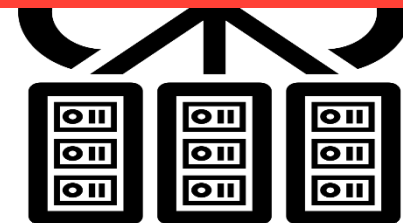
Graph/Tree Processing

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**

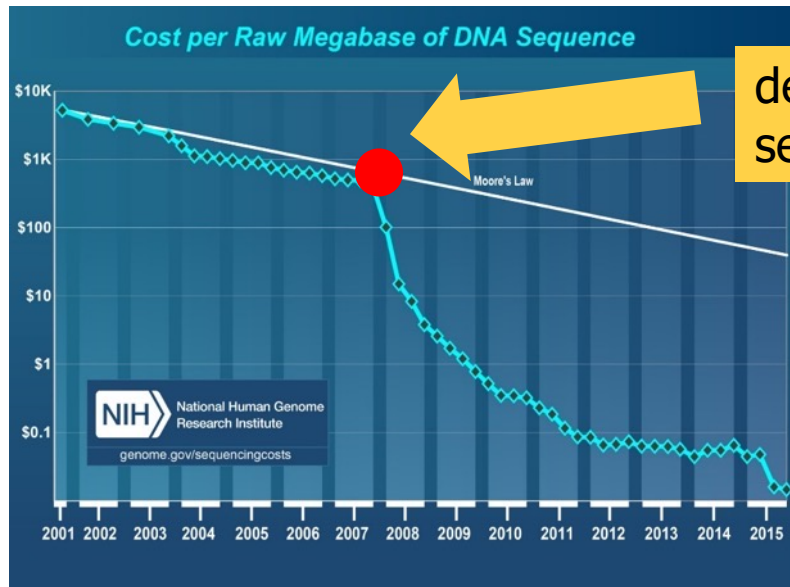
VP9



Video Capture

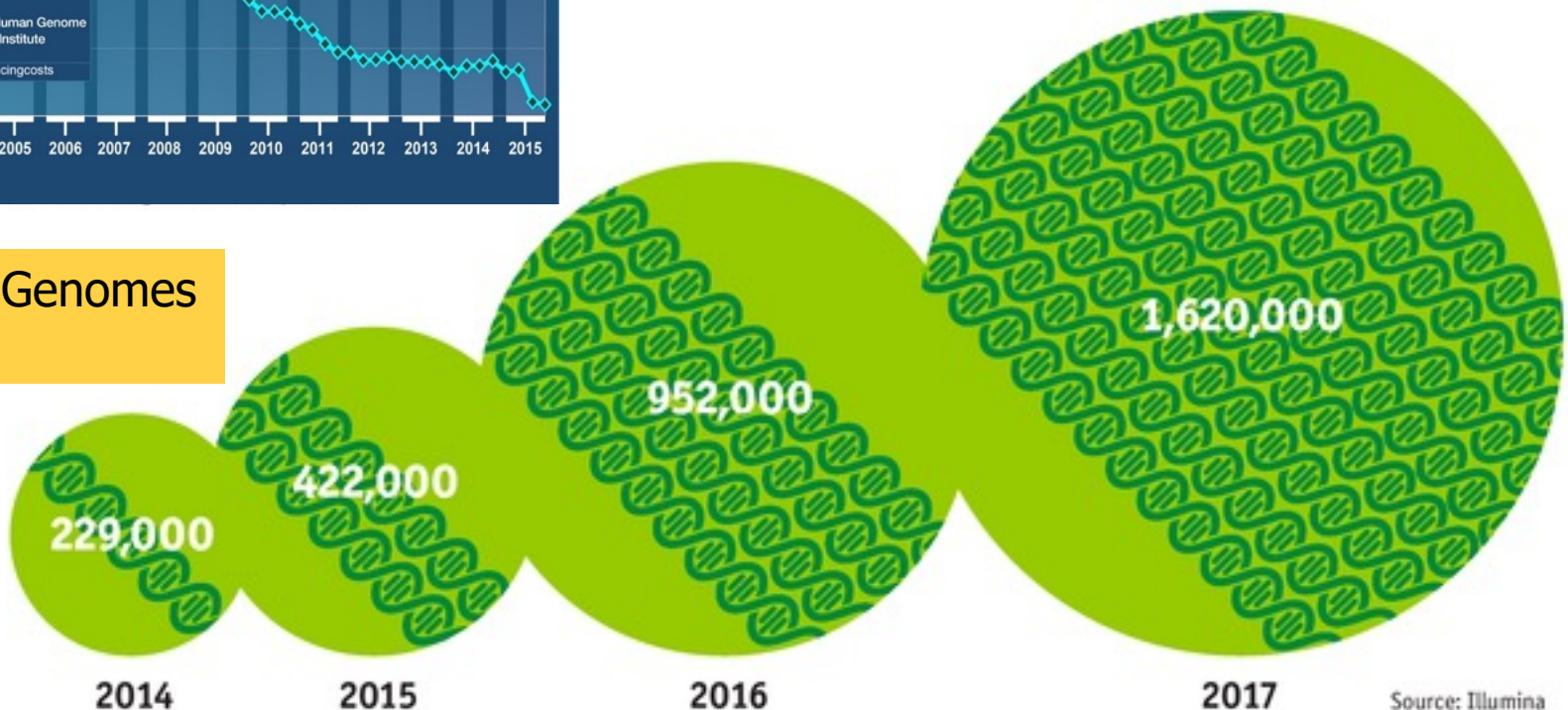
Google's **video codec**

Data is Key for Future Workloads

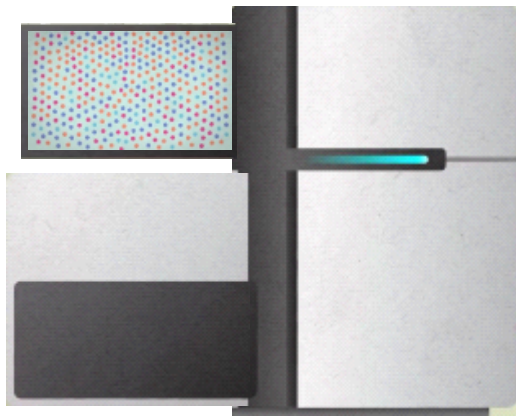


development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

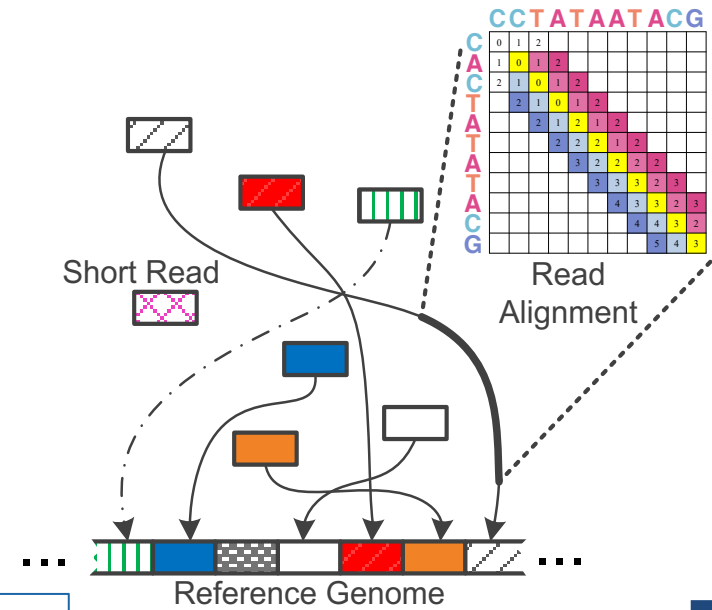


The Economist



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



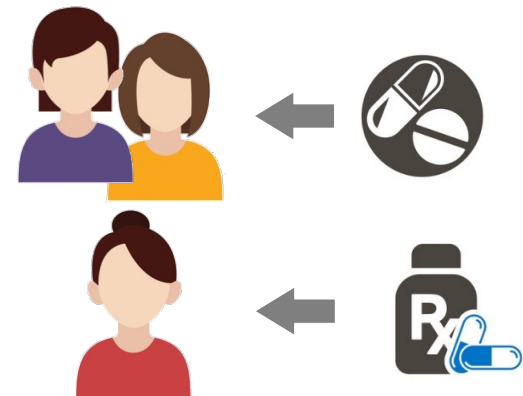
Understanding **genetic variations**,
species, **evolution**, ...



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>

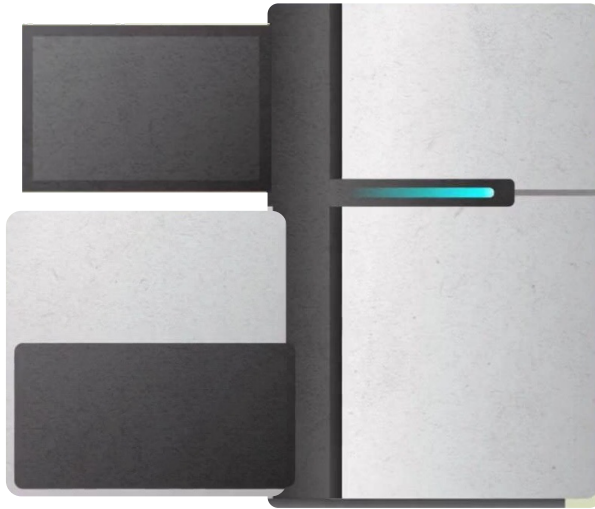
Published: 02 April 2018 **Article history** ▼



Oxford Nanopore MinION

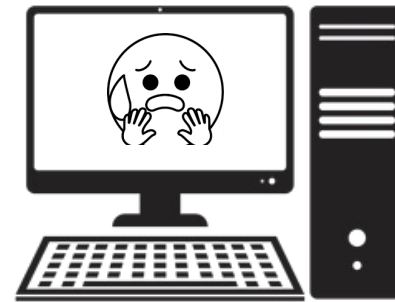
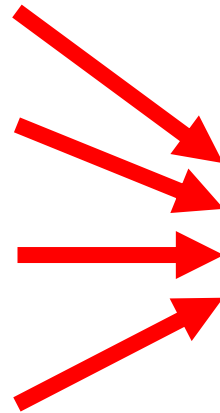
Data → performance & energy bottleneck

Problems with (Genome) Analysis Today



Special-Purpose Machine
for **Data Generation**

FAST



General-Purpose Machine
for **Data Analysis**

SLOW

Slow and inefficient processing capability
Large amounts of data movement

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



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

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

Henk Corporaal[★] Onur Mutlu^{◇✕}

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

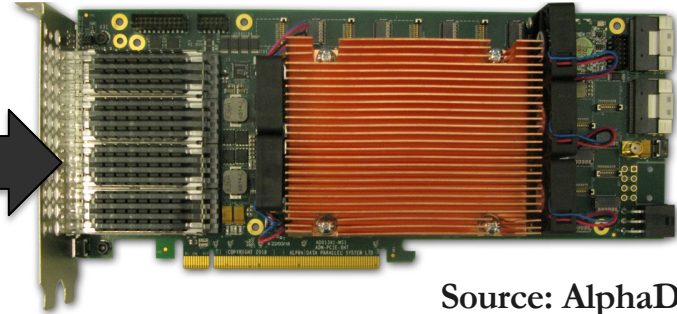
[★]*Eindhoven University of Technology* [▽]*IBM Research Europe*

Near-Memory Acceleration using FPGAs



Source: IBM

IBM POWER9 CPU



Source: AlphaData

HBM-based FPGA board

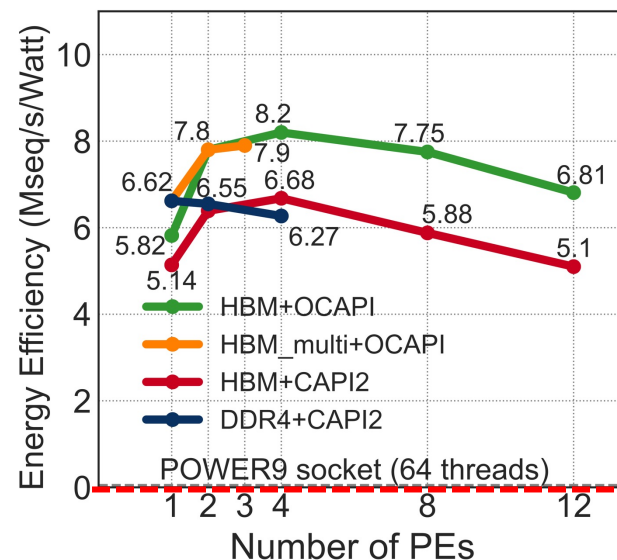
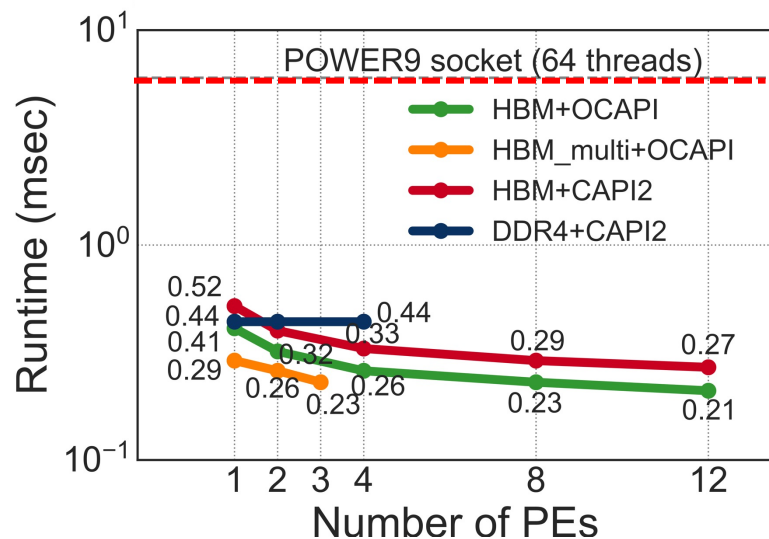
Near-HBM FPGA-based accelerator

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

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

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

Accelerating Sequence-to-Graph Mapping

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

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

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

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

⁵Carnegie Mellon University ⁶University of Illinois Urbana-Champaign

Accelerating Basecalling + Read Mapping

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

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

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

Designing & Accelerating Basecallers

A Framework for Designing Efficient Deep Learning-Based Genomic Basecallers

Gagandeep Singh^a Mohammed Alser^{*a} Alireza Khodamoradi^{*b}
Kristof Denolf^b Can Firtina^a Meryem Banu Cavlak^a
Henk Corporaal^c Onur Mutlu^a
^aETH Zürich ^bAMD ^cEindhoven University of Technology

Nanopore sequencing is a widely-used high-throughput genome sequencing technology that can sequence long fragments of a genome. Nanopore sequencing generates noisy electrical signals that need to be converted into a standard string of DNA nucleotide bases (i.e., A, C, G, T) using a computational step called *basecalling*. The accuracy and speed of basecalling have critical implications for every subsequent step in genome analysis. Currently, basecallers are developed mainly based on deep learning techniques to provide high sequencing accuracy without considering the compute demands of such tools. We observe that state-of-the-art basecallers (i.e., Guppy, Bonito, Fast-Bonito) are slow, inefficient, and memory-hungry

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



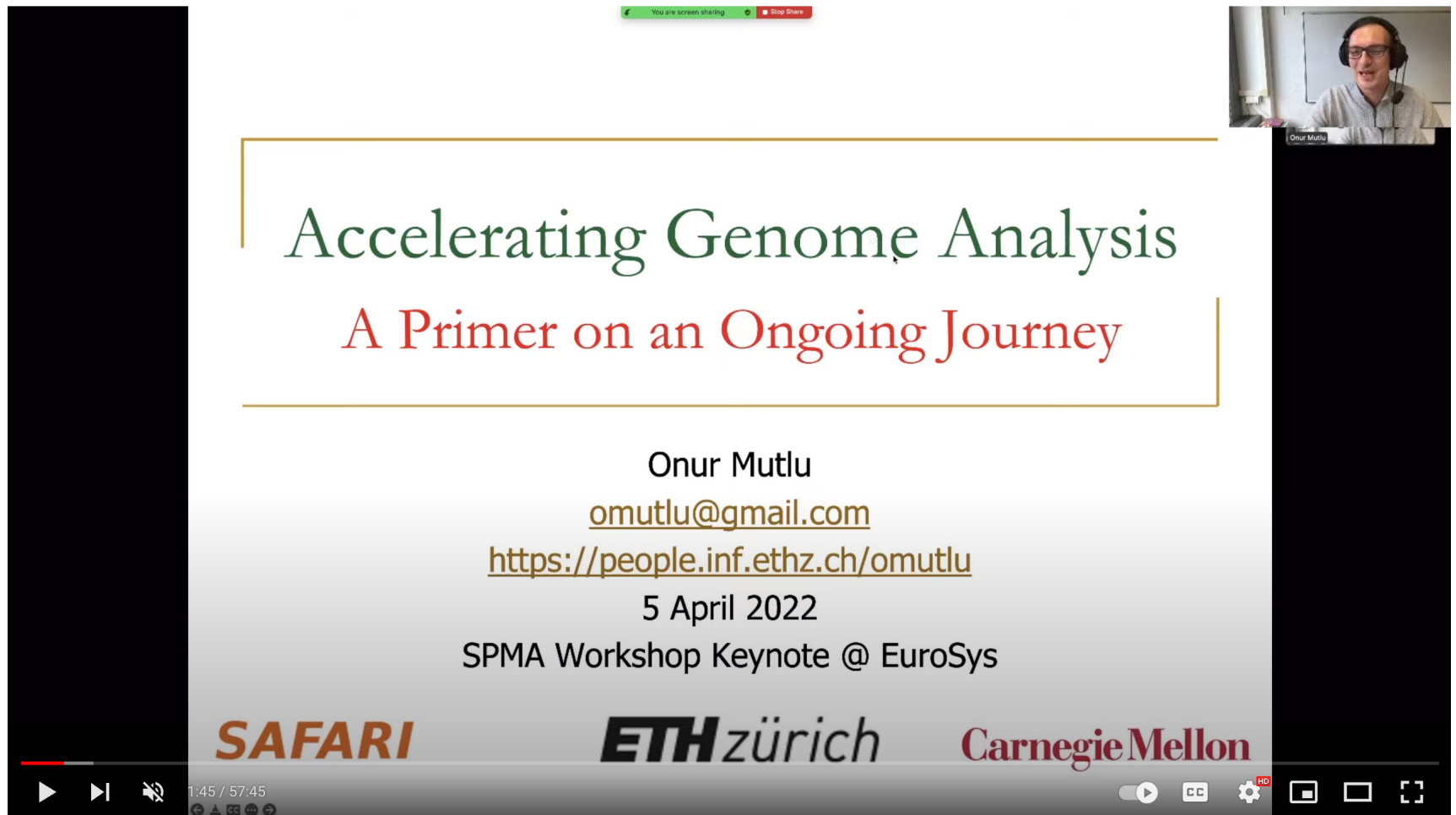
Onur Mutlu Lectures
15.9K subscribers

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

ANALYTICS

EDIT VIDEO

More on Fast & Efficient Genome Analysis ...



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

Accelerating Genome Analysis
A Primer on an Ongoing Journey

Onur Mutlu
omutlu@gmail.com
<https://people.inf.ethz.ch/omutlu>
5 April 2022
SPMA Workshop Keynote @ EuroSys

Logos at the bottom: **SAFARI**, **ETH** zürich, and **Carnegie Mellon**.

Video player controls at the bottom show a progress bar at 1:45 / 57:45, play/pause, volume, and other standard controls.

Accelerating Genome Analysis - Onur Mutlu (Keynote Talk at Systems for Post-Moore Arch. @ EuroSys)



Onur Mutlu Lectures
28.7K subscribers

[Analytics](#)

[Edit video](#)

16



Share

Download

Clip

Save



<https://www.youtube.com/watch?v=NCagwf0ivT0>

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>

Genomics Course (Fall 2022)

Fall 2022 Edition:

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

Spring 2022 Edition:

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

Youtube Livestream (Fall 2022):

- https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFIQvdxOdizD_EhVAMVQV

Youtube Livestream (Spring 2022):

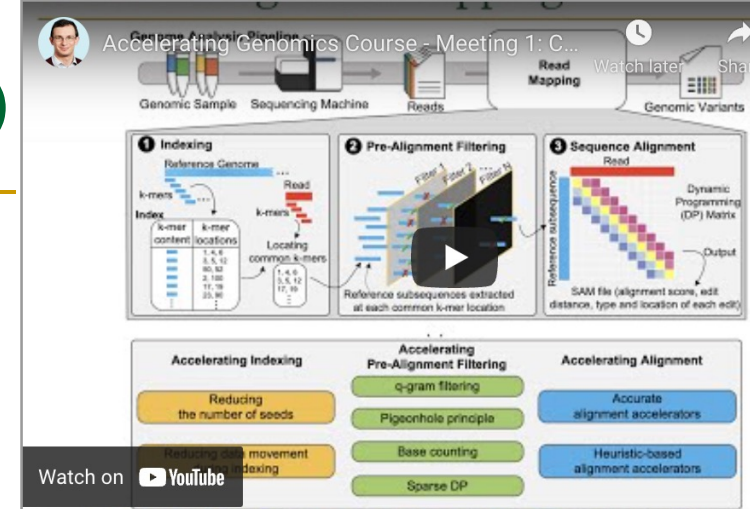
- https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18

Project course

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

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

SAFARI



Spring 2022 Meetings/Schedule

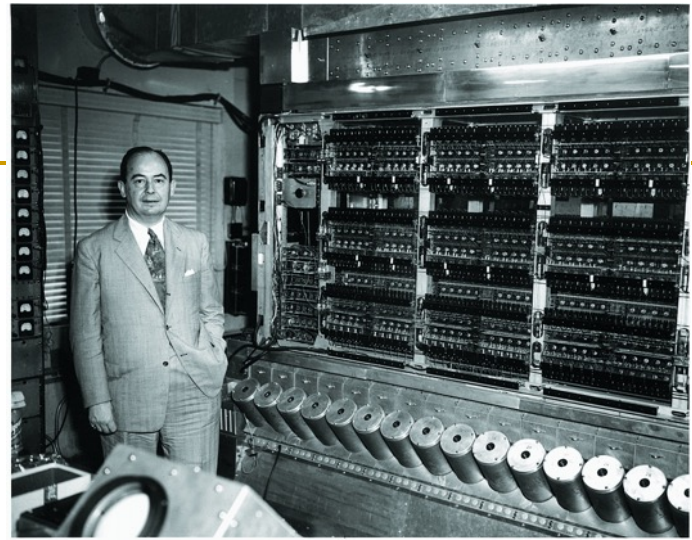
Week	Date	Livestream	Meeting	Learning Materials
W1	11.3 Fri.	YouTube Live	M1: P&S Accelerating Genomics Course Introduction & Project Proposals (PDF) (PPT)	Required Materials Recommended Materials
W2	18.3 Fri.	YouTube Live	M2: Introduction to Sequencing (PDF) (PPT)	
W3	25.3 Fri.	YouTube Premiere	M3: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

Data Overwhelms Modern Machines ...

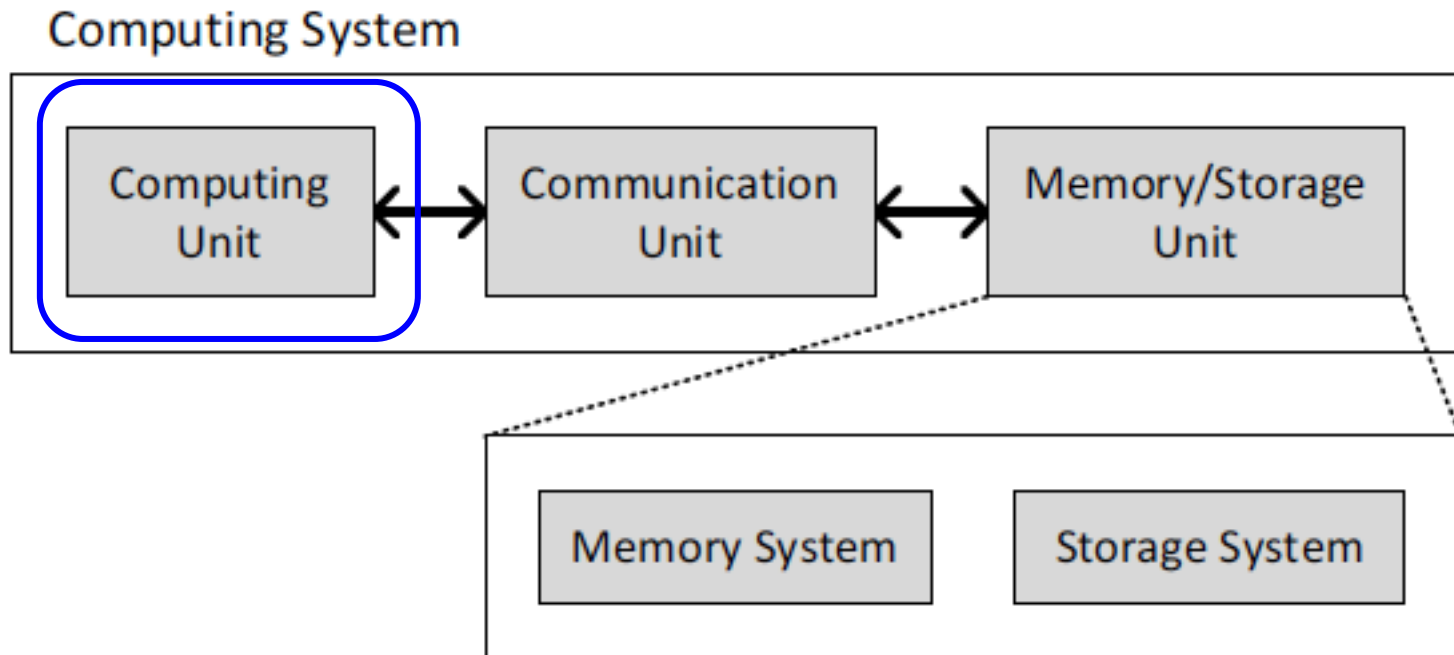
- Storage/memory capability
- Communication capability
- Computation capability
- Greatly impacts robustness, energy, performance, cost

A Computing System

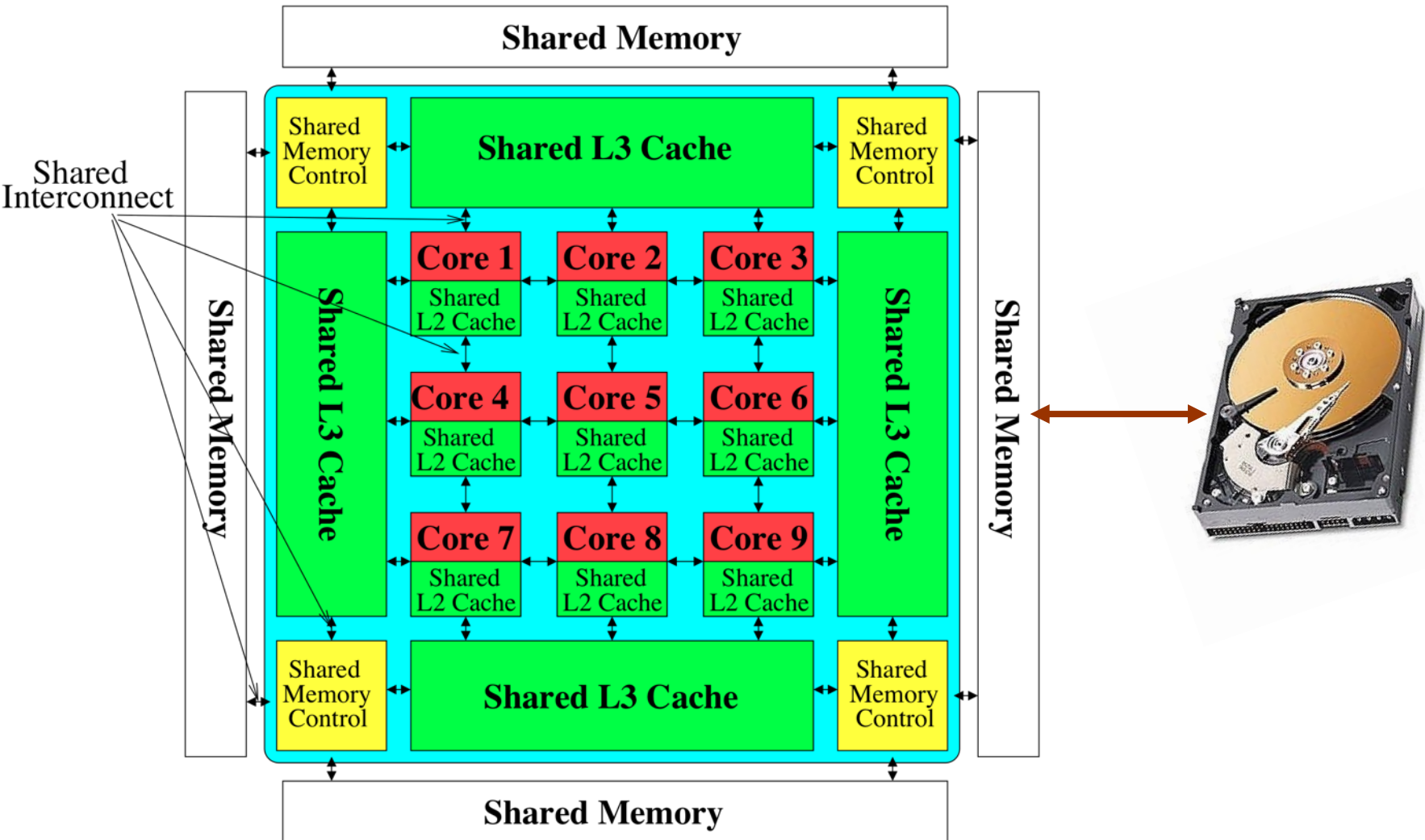
- Three key components
- Computation
- Communication
- Storage/memory



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



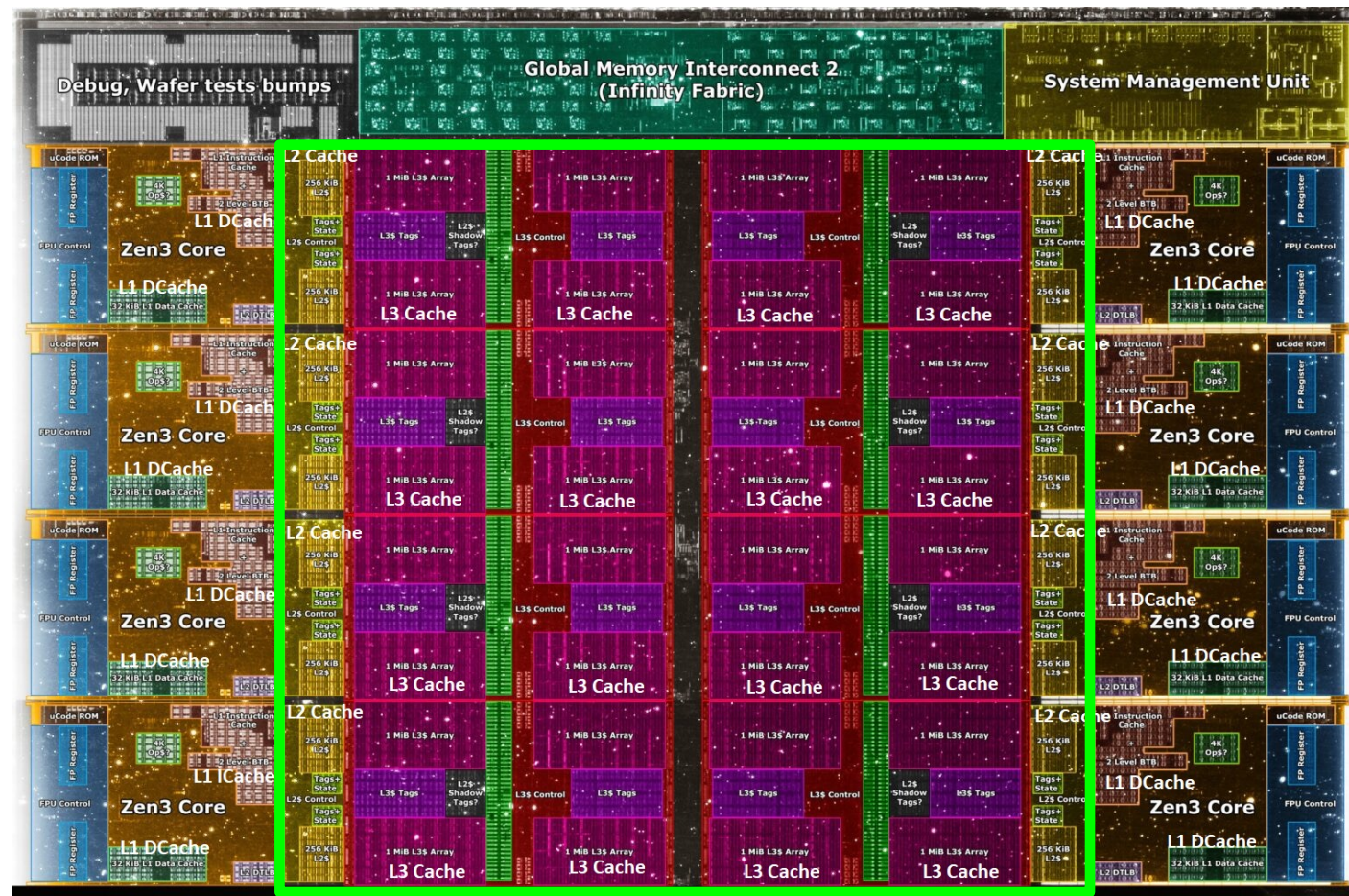
Perils of Processor-Centric Design



Most of the system is dedicated to storing and moving data

Yet, system is still bottlenecked by memory & storage

Deeper and Larger Memory Hierarchies



Core Count:

8 cores/16 threads

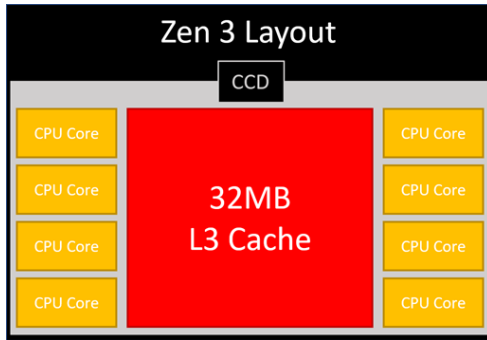
L1 Caches:
32 KB per core

L2 Caches:
512 KB per core

L3 Cache:
32 MB shared

AMD Ryzen 5000, 2020

AMD's 3D Last Level Cache (2021)

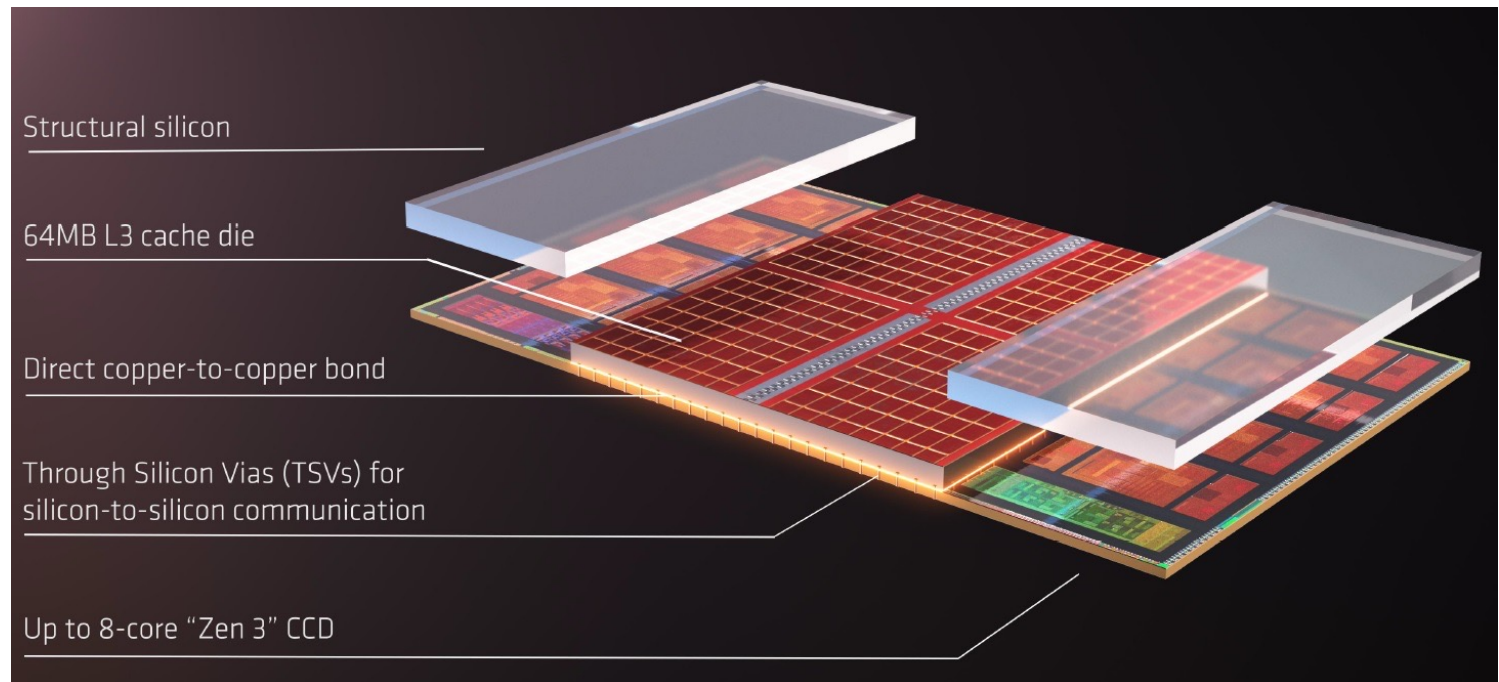


<https://community.microcenter.com/discussion/5134/comparing-zen-3-to-zen-2>

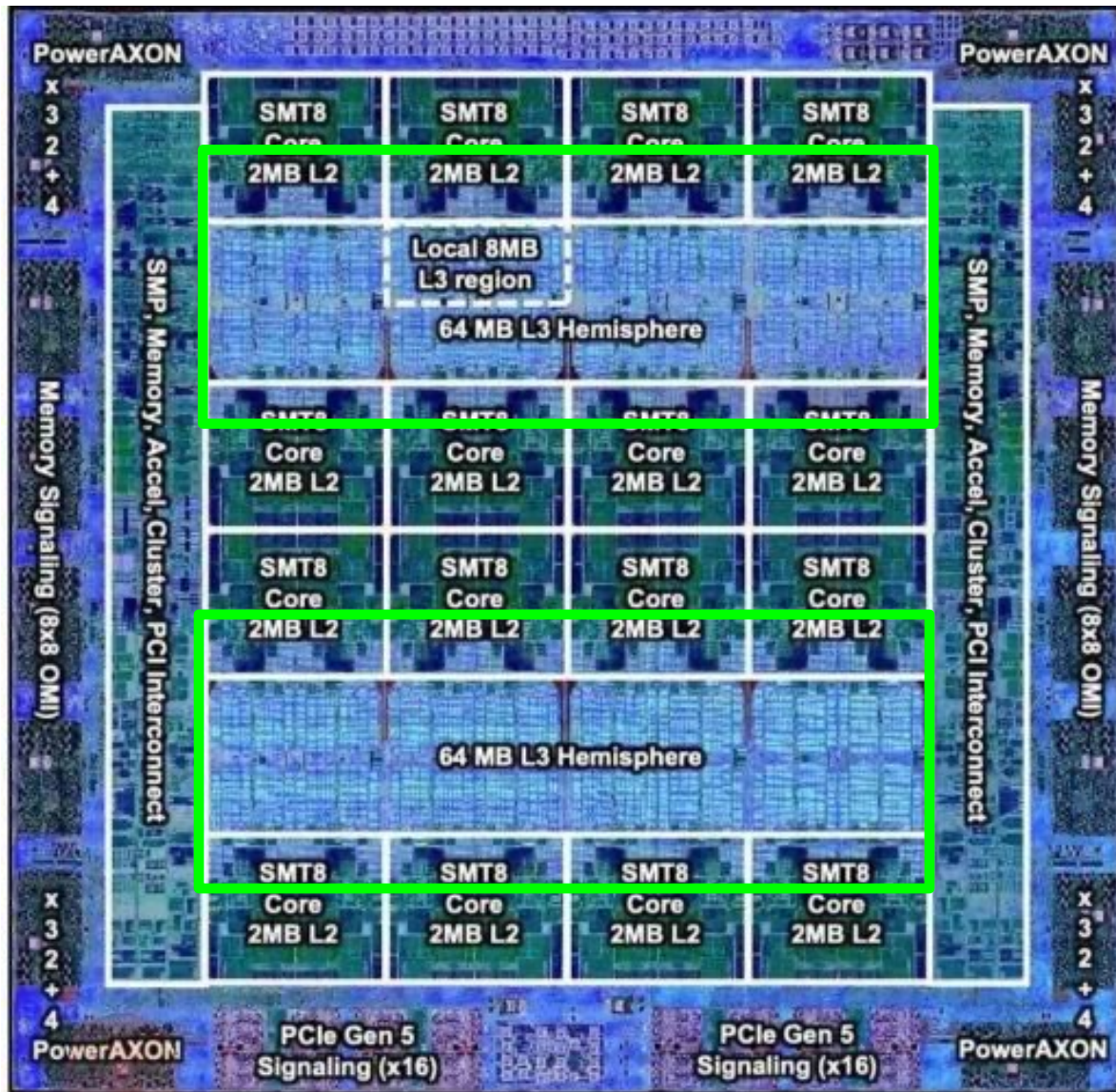
AMD increases the L3 size of their 8-core Zen 3 processors from 32 MB to 96 MB

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



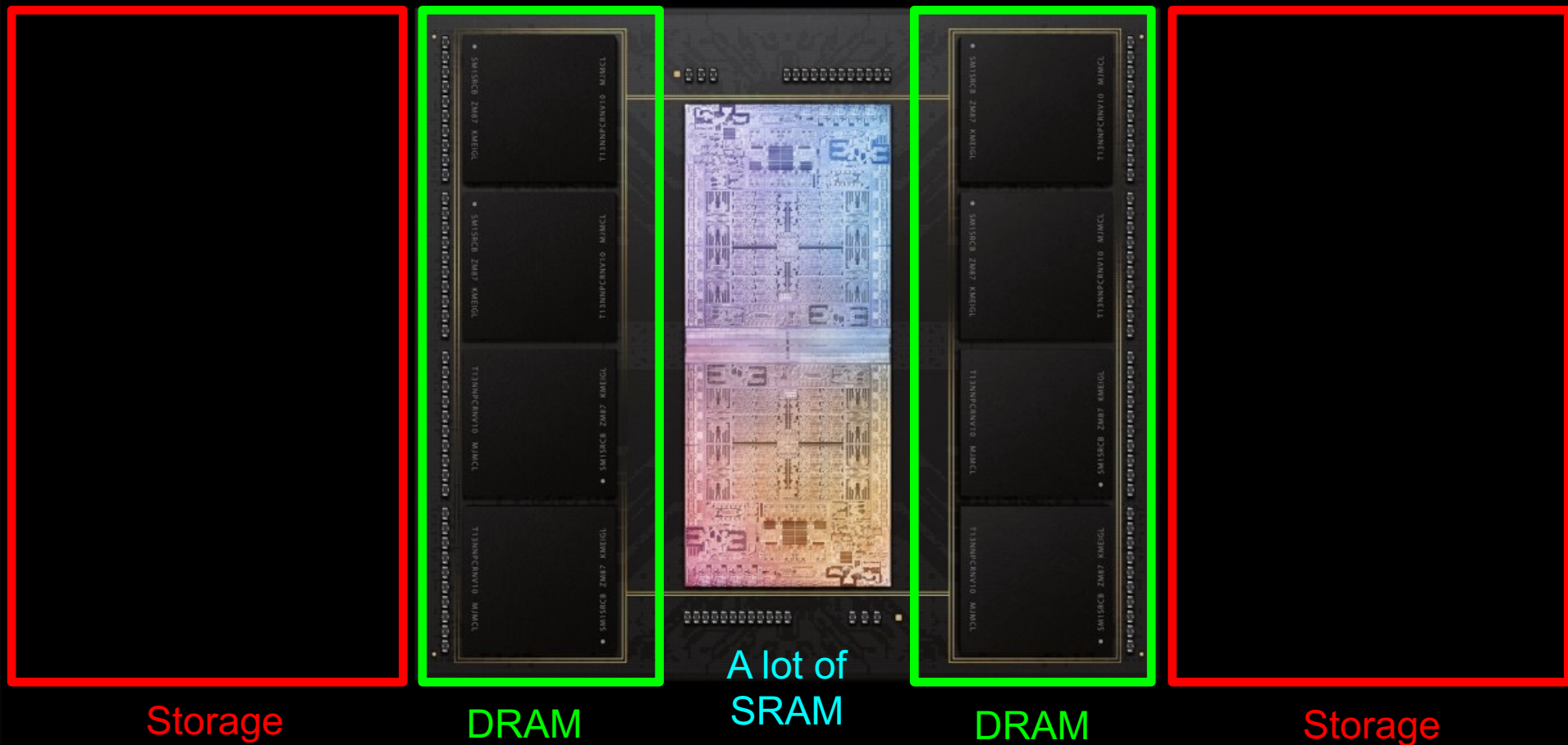
IBM POWER10,
2020

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 Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

Data Movement Overwhelms Accelerators

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

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

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

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

Saugata Ghose[‡]
Xiaoyu Ma[§]

Berkin Akin[§]
Eric Shiu[§]

Ravi Narayanaswami[§]
Onur Mutlu^{*†}

[†]Carnegie Mellon Univ. [◇]Stanford Univ. [‡]Univ. of Illinois Urbana-Champaign [§]Google ^{*}ETH Zürich

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 ...

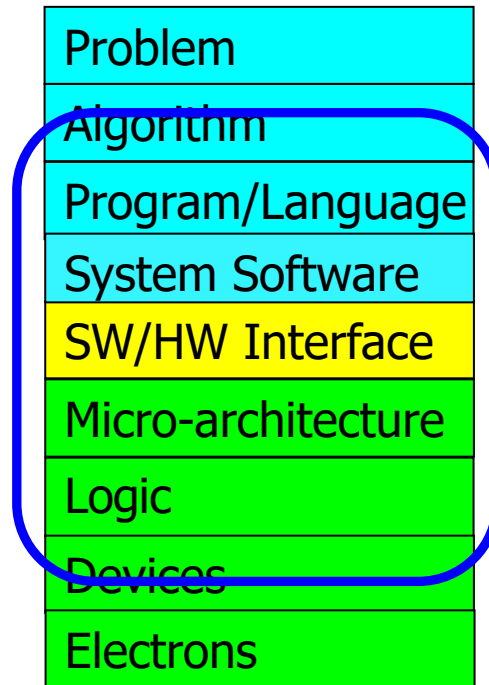
- Are **processor-centric** vs. **data-centric**
- Make **designer-dictated** decisions vs. **data-driven**
- Make **component-based myopic** decisions vs. **data-aware**

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-Driven (Self-Optimizing) Architectures

System Architecture Design Today

- Human-driven
 - Humans design the policies (how to do things)
- Many (too) simple, short-sighted policies all over the system
- No automatic data-driven policy learning
- (Almost) no learning: cannot take lessons from past actions

**Can we design
fundamentally intelligent architectures?**

An Intelligent Architecture

- Data-driven
 - Machine learns the “best” policies (how to do things)
- Sophisticated, workload-driven, changing, far-sighted policies
- Automatic data-driven policy learning
- All controllers are intelligent data-driven agents

**We need to rethink design
(of all controllers)**

Self-Optimizing Memory Controllers

- Engin Ipek, Onur Mutlu, José F. Martínez, and Rich Caruana,
"Self Optimizing Memory Controllers: A Reinforcement Learning Approach"
Proceedings of the 35th International Symposium on Computer Architecture (ISCA), pages 39-50, Beijing, China, June 2008.

Self-Optimizing Memory Controllers: A Reinforcement Learning Approach

Engin İpek^{1,2} Onur Mutlu² José F. Martínez¹ Rich Caruana¹

¹Cornell University, Ithaca, NY 14850 USA

²Microsoft Research, Redmond, WA 98052 USA

Self-Optimizing Memory Prefetchers

Rahul Bera, Konstantinos Kanellopoulos, Anant Nori, Taha Shahroodi, Sreenivas Subramoney, and Onur Mutlu,
"Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning"
Proceedings of the 54th International Symposium on Microarchitecture (MICRO), Virtual, October 2021.

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

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

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

[[Talk Video](#) (20 minutes)]

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

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

[[arXiv version](#)]

Officially artifact evaluated as available, reusable and reproducible.



Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera¹

Konstantinos Kanellopoulos¹

Anant V. Nori²

Taha Shahroodi^{3,1}

Sreenivas Subramoney²

Onur Mutlu¹

¹ETH Zürich

²Processor Architecture Research Labs, Intel Labs

³TU Delft

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

Learning-Based Off-Chip Load Predictors

- Rahul Bera, Konstantinos Kanellopoulos, Shankar Balachandran, David Novo, Ataberk Olgun, Mohammad Sadrosadati, and Onur Mutlu,
"Hermes: Accelerating Long-Latency Load Requests via Perceptron-Based Off-Chip Load Prediction"

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

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

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

[[Talk Video](#) (12 minutes)]

[[Lecture Video](#) (25 minutes)]

[[arXiv version](#)]

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

Officially artifact evaluated as available, reusable and reproducible.

Best paper award at MICRO 2022.



Hermes: Accelerating Long-Latency Load Requests via Perceptron-Based Off-Chip Load Prediction

Rahul Bera¹ Konstantinos Kanellopoulos¹ Shankar Balachandran² David Novo³
Ataberk Olgun¹ Mohammad Sadrosadati¹ Onur Mutlu¹

¹ETH Zürich ²Intel Processor Architecture Research Lab ³LIRMM, Univ. Montpellier, CNRS

<https://arxiv.org/pdf/2209.00188.pdf>

Self-Optimizing Hybrid SSD Controllers

Gagandeep Singh, Rakesh Nadig, Jisung Park, Rahul Bera, Nastaran Hajinazar, David Novo, Juan Gomez-Luna, Sander Stuijk, Henk Corporaal, and Onur Mutlu,

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

Proceedings of the 49th International Symposium on Computer Architecture (ISCA), New York, June 2022.

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

[[arXiv version](#)]

[[Sibyl Source Code](#)]

[[Talk Video](#) (16 minutes)]

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

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

¹ETH Zürich

²Eindhoven University of Technology

³LIRMM, Univ. Montpellier, CNRS

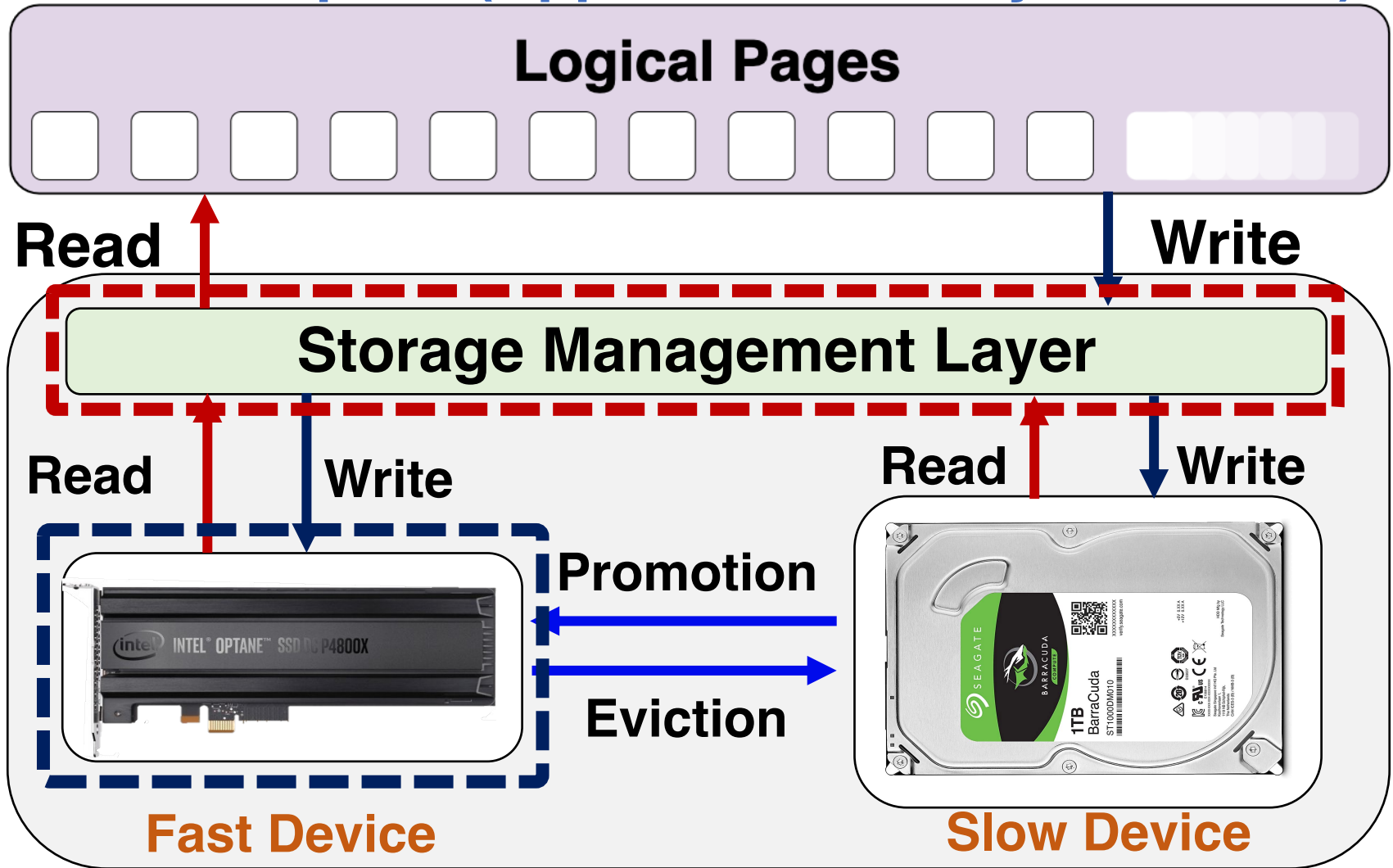
Sibyl:

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, Sander Stuijk, Henk Corporaal,
Onur Mutlu

Hybrid Storage System Basics

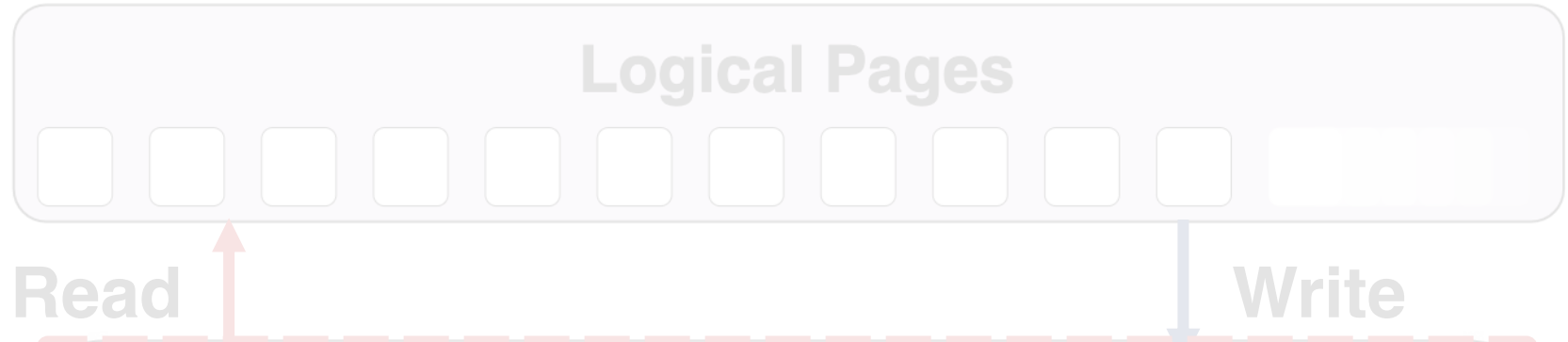
Address Space (Application/File System View)



Hybrid Storage System

Hybrid Storage System Basics

Logical Address Space (Application/File System View)



Performance of a hybrid storage system
highly depends on the ability of the
storage management layer



Key Shortcomings in Prior Techniques

We observe **two key shortcomings** that significantly limit the performance benefits of prior techniques

1. Lack of **adaptivity to**:
 - a) Workload changes
 - b) Changes in device types and configuration

2. Lack of **extensibility** to more devices

Our Goal

A **data-placement mechanism**
that can provide:

1. **Adaptivity**, by **continuously learning** and **adapting** to the application and underlying device characteristics
2. **Easy extensibility** to incorporate a wide range of hybrid storage configurations

Our Proposal

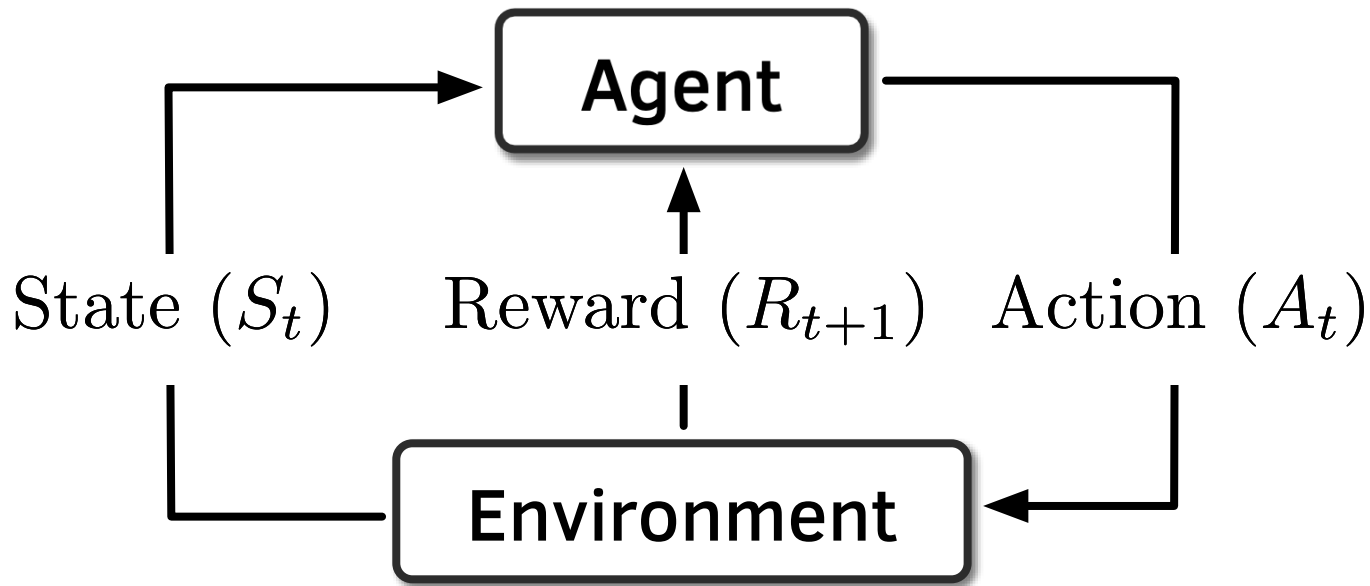


Sibyl

Formulates data placement in
hybrid storage systems as a
reinforcement learning problem

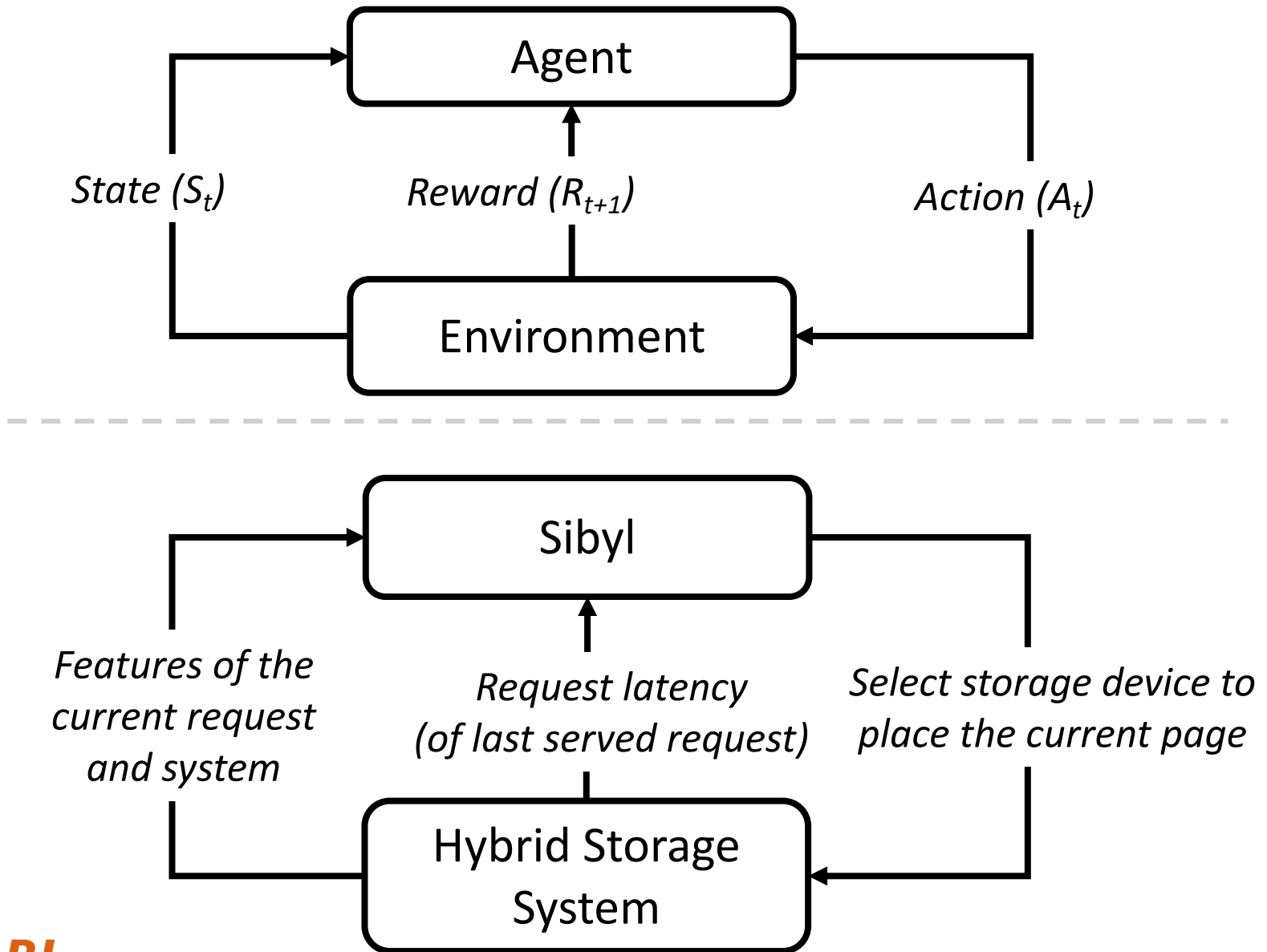
Sybil is an oracle that makes accurate prophecies
<https://en.wikipedia.org/wiki/Sibyl>

Basics of Reinforcement Learning (RL)

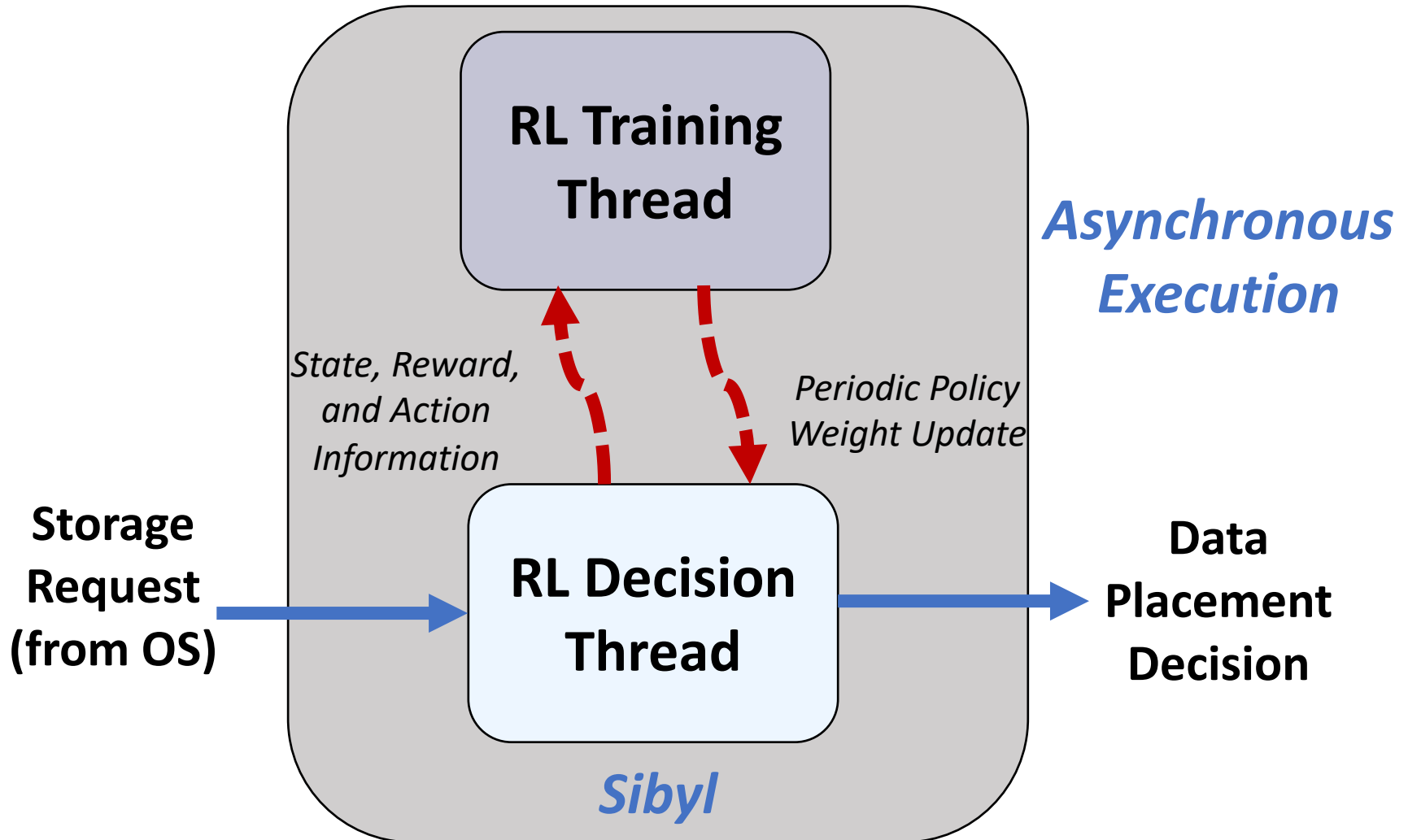


Agent learns to take an **action** in a given **state** to maximize a numerical **reward**

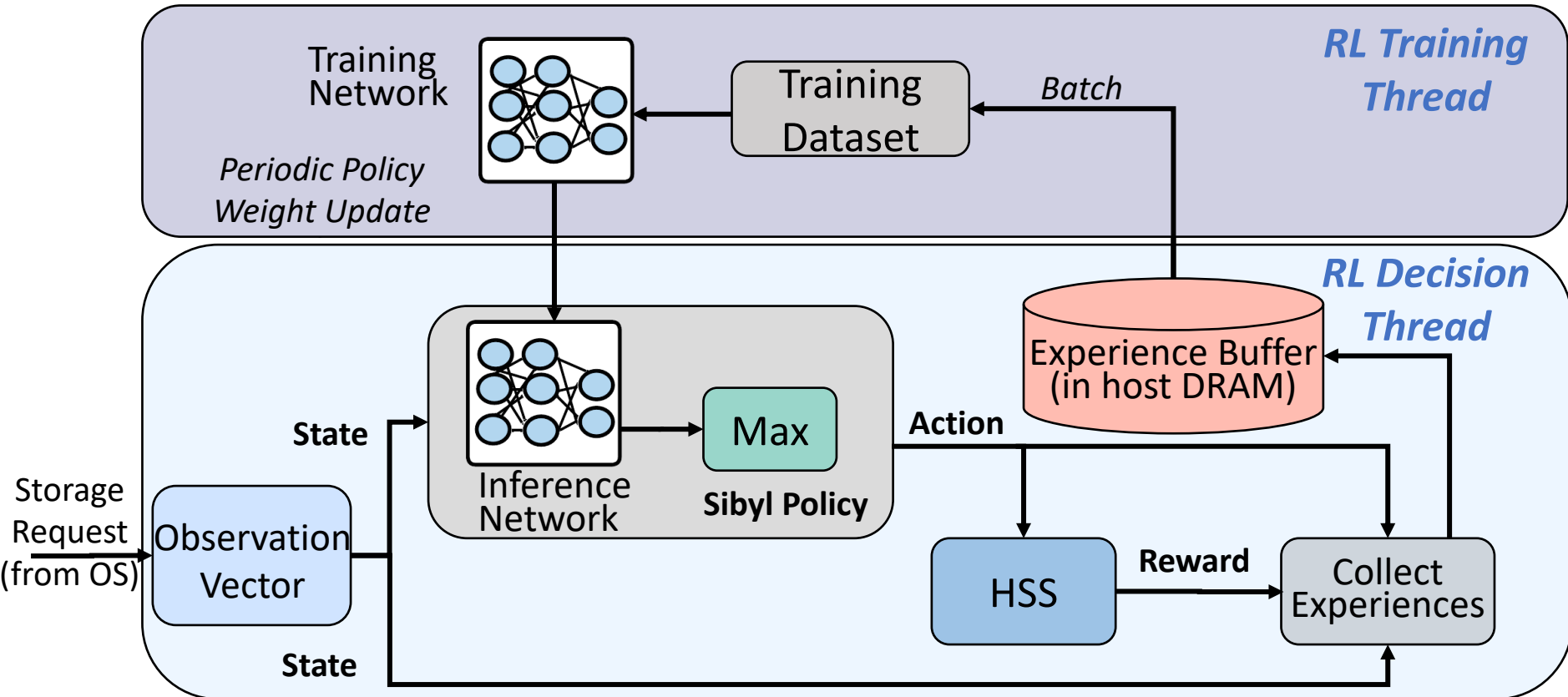
Formulating Data Placement as RL



Sibyl Execution



Sibyl Design: Overview



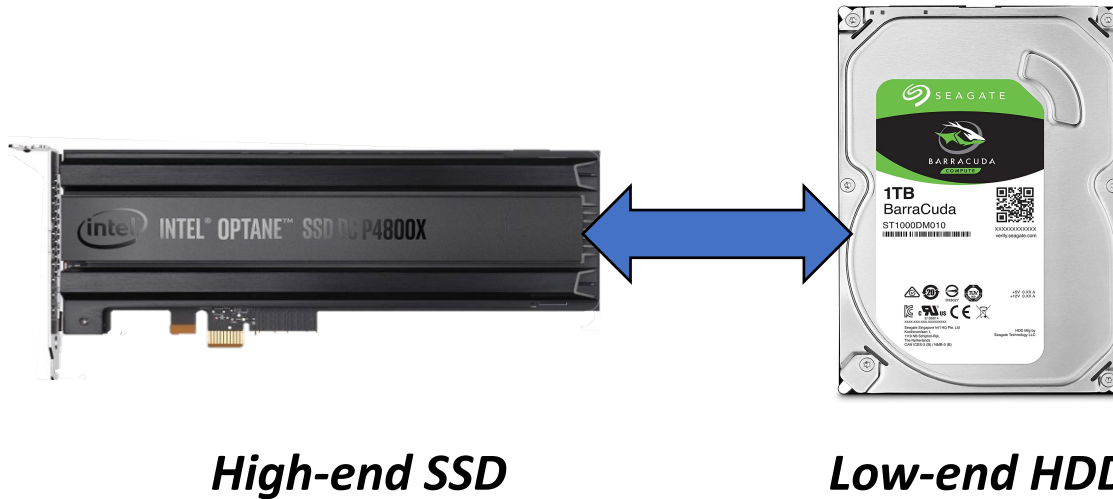
Evaluation Methodology (1/3)

- **Real system** with various HSS configurations
 - Dual-hybrid and tri-hybrid systems

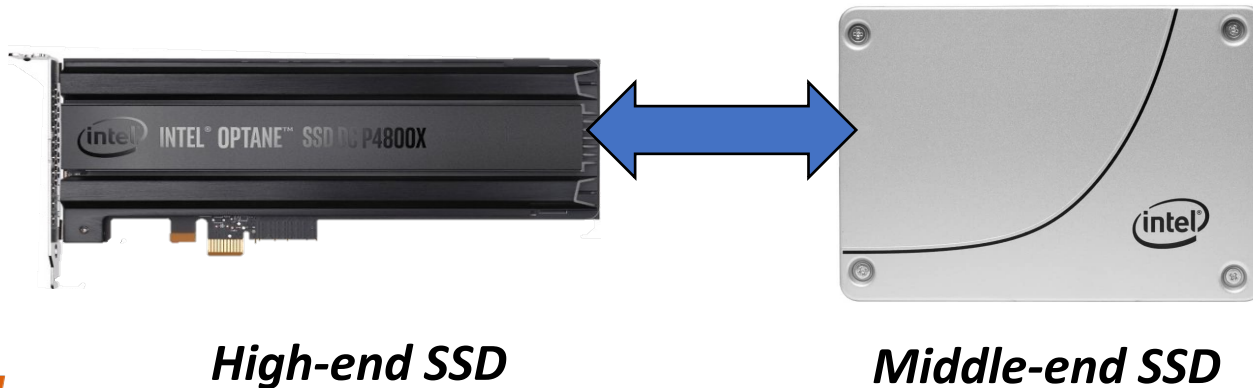


Evaluation Methodology (2/3)

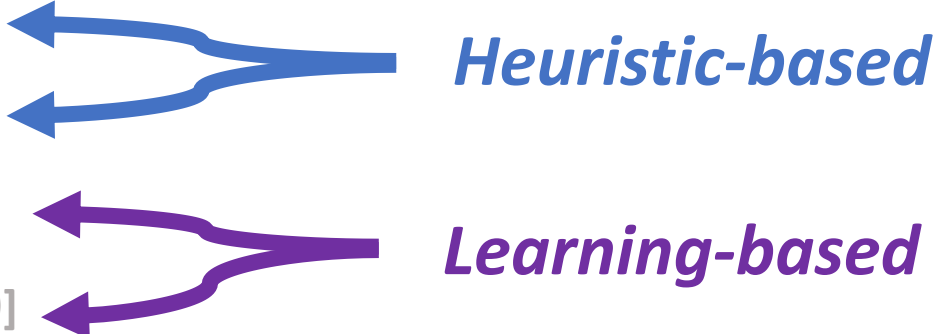
Cost-Oriented HSS Configuration



Performance-Oriented HSS Configuration



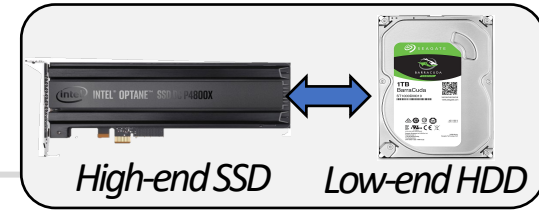
Evaluation Methodology (3/3)

- **18 different workloads** from:
 - MSR Cambridge and Filebench Suites
- **Four** state-of-the-art data placement baselines:
 - CDE [Matsui+, Proc. IEEE'17]
 - HPS [Meswani+, HPCA'15]
 - Archivist [Ren+, ICCD'19]
 - RNN-HSS [Doudali+, HPDC'19]

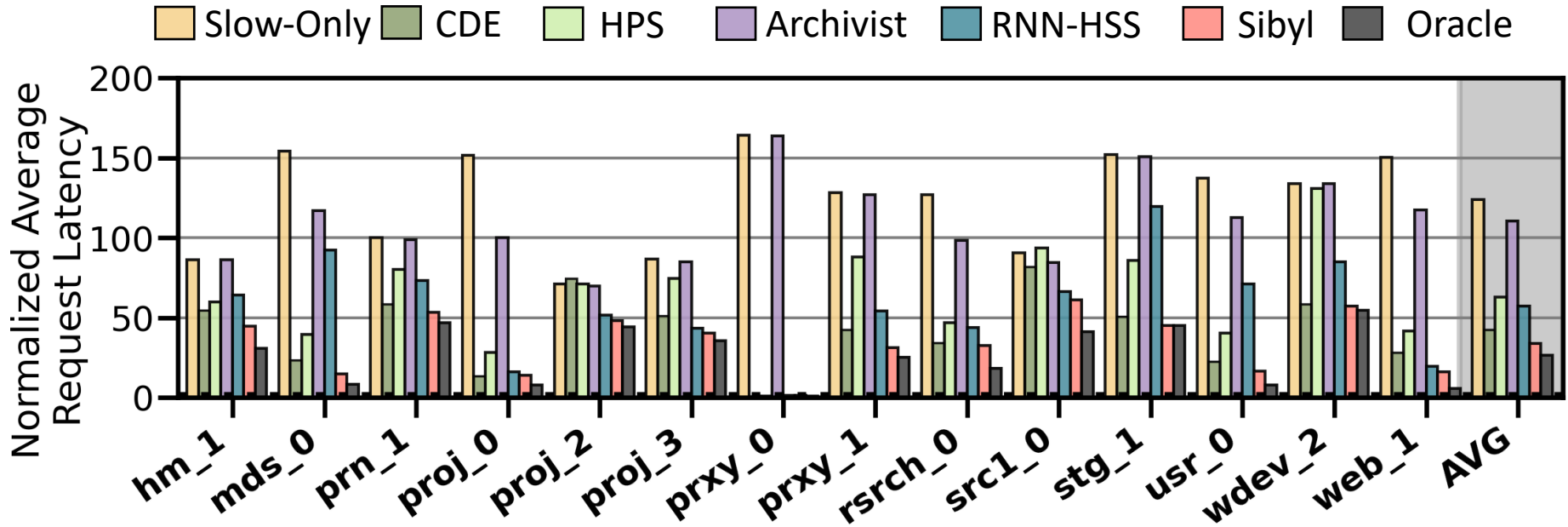
Heuristic-based

Learning-based

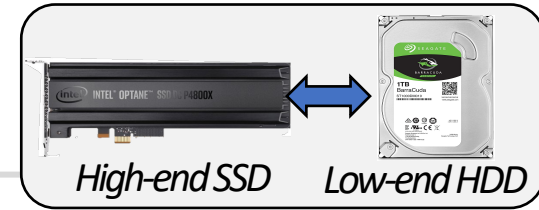
Performance Analysis



Cost-Oriented HSS Configuration

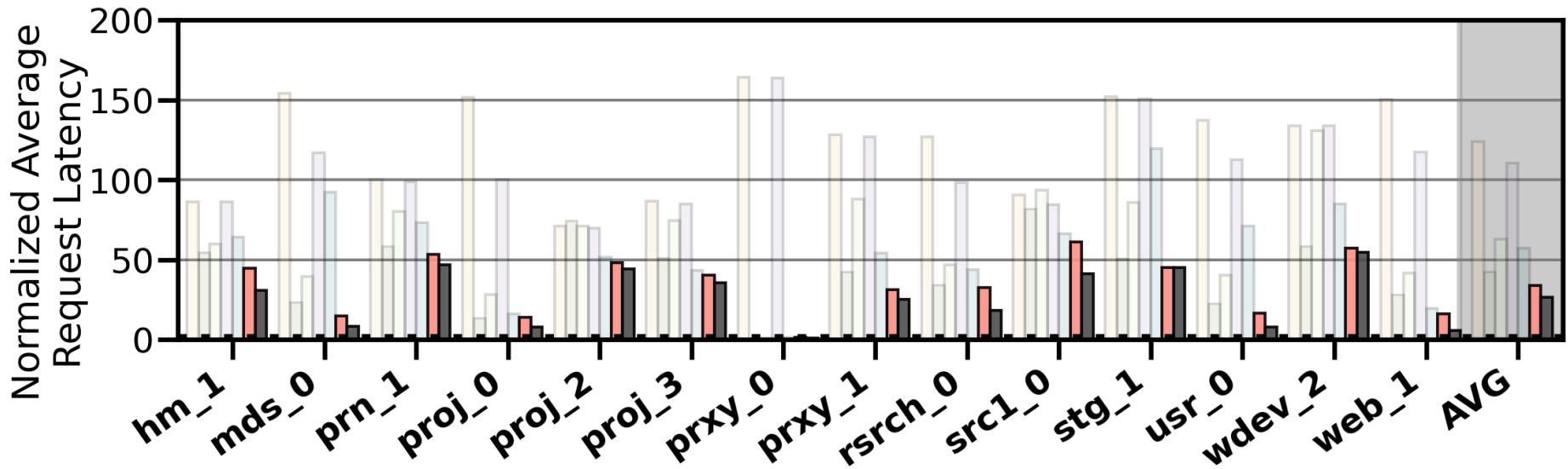


Performance Analysis



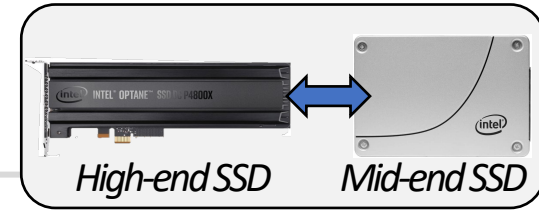
Cost-Oriented HSS Configuration

Slow-Only CDE HPS Archivist RNN-HSS Sibyl Oracle

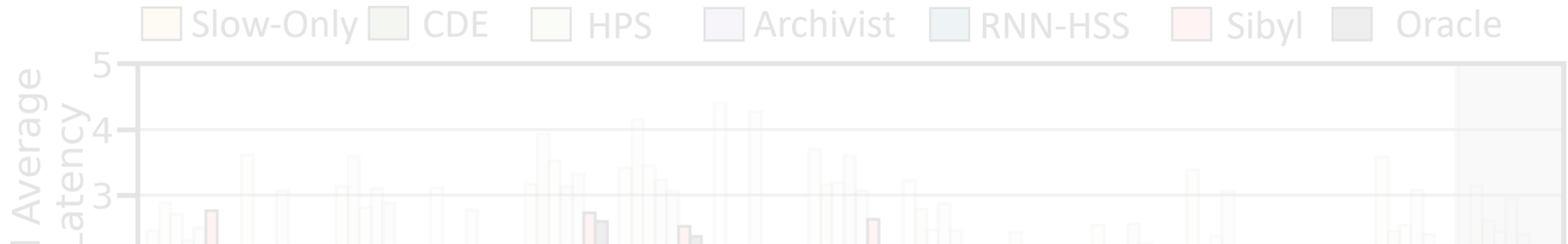


Sibyl consistently **outperforms all the baselines**
for all the workloads

Performance Analysis

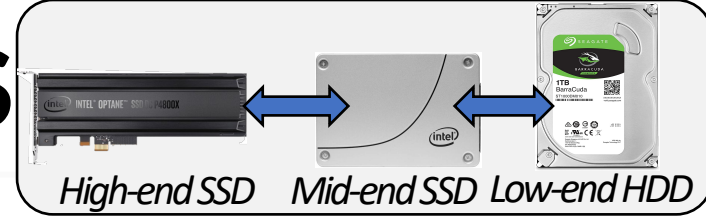


Performance-Oriented HSS Configuration



Sibyl achieves **80% of the performance of an oracle policy** that has complete knowledge of future access patterns

Performance on Tri-HSS



Extending Sibyl for **more devices**:

1. Add a new action

Sibyl **outperforms** the state-of-the-art data placement policy by **48.2% in a real tri-hybrid system**

Sibyl reduces the system architect's burden by providing **ease of extensibility**

Sibyl: Summary

- **We introduced Sibyl**, the first reinforcement learning-based data placement technique in hybrid storage systems that provides
 - **Adaptivity**
 - **Easily extensibility**
 - **Ease of design and implementation**
- **We evaluated Sibyl** on **real systems** using many different workloads
 - In a tri-HSS configuration, Sibyl **outperforms** the state-of-the-art-data placement policy by **48.2%**
 - Sibyl achieves **80% of the performance** of an oracle policy with a storage overhead of only **124.4 KiB**

Data-Driven (Self-Optimizing) Computing Architectures

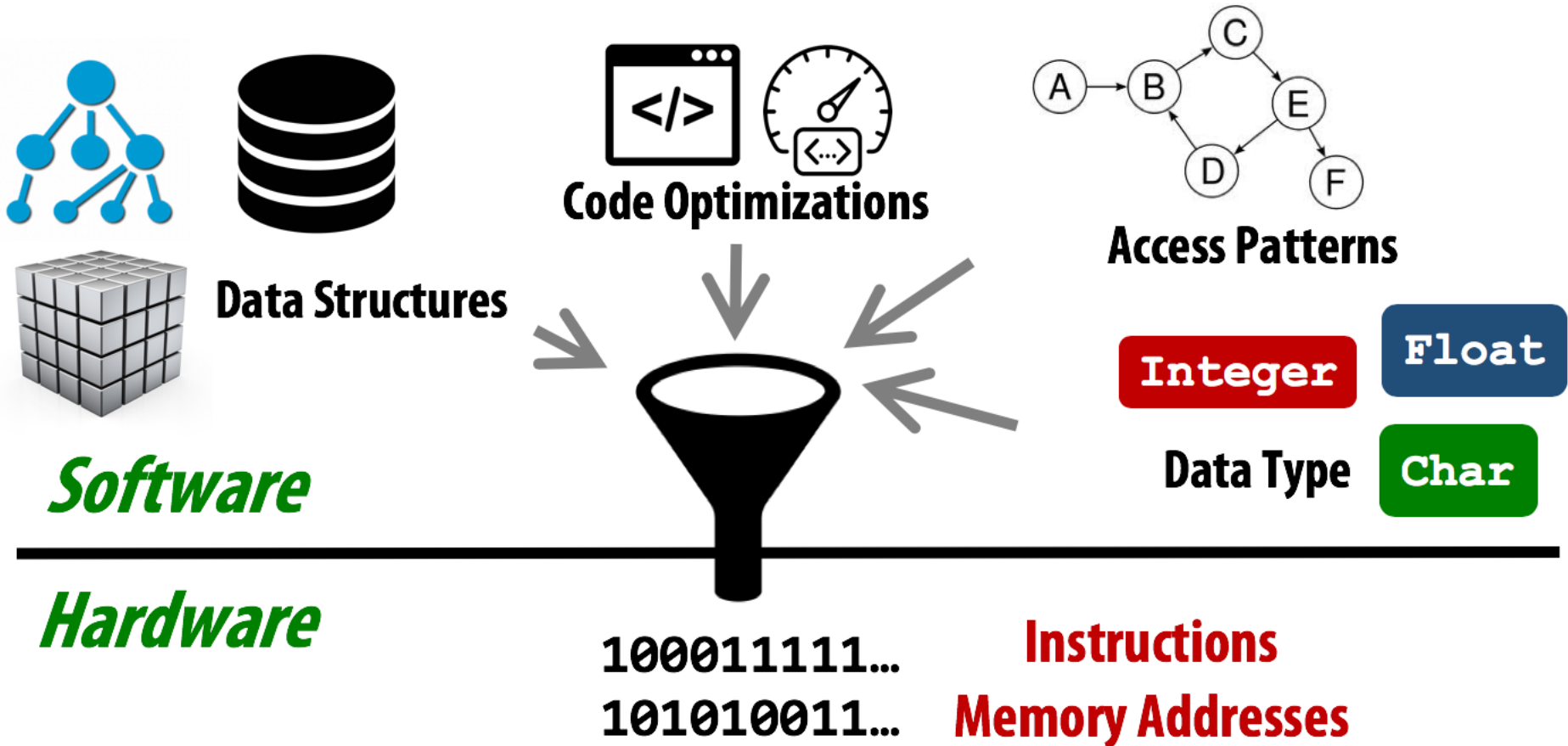
Data-Characteristic-Aware Architectures

Data-Aware Architectures

- A data-aware architecture understands what it can do with and to each piece of data
- It makes use of different properties of data to improve performance, efficiency and other metrics
 - Compressibility
 - Approximability
 - Locality
 - Sparsity
 - Criticality for Computation X
 - Access Semantics
 - ...

One Problem: Limited Expressiveness

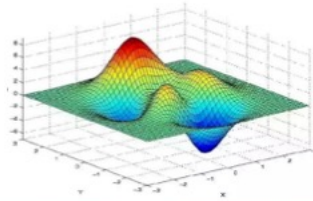
Higher-level information is not visible to HW



A Solution: More Expressive Interfaces

Performance

Software



Functionality



**ISA
Virtual Memory**

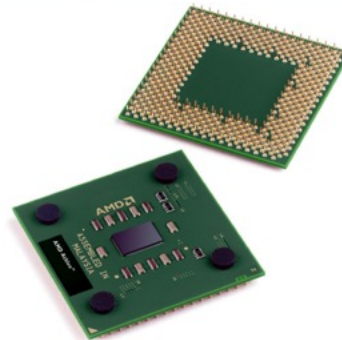
**Higher-level
Program
Semantics**

**Expressive
Memory
“XMem”**

Hardware



wiseGEEK



Expressive (Memory) Interfaces

- Nandita Vijaykumar, Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko, Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons and Onur Mutlu, **"A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory"**
Proceedings of the 45th International Symposium on Computer Architecture (ISCA), Los Angeles, CA, USA, June 2018.
[[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Video](#)]

A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory

Nandita Vijaykumar^{†§} Abhilasha Jain[†] Diptesh Majumdar[†] Kevin Hsieh[†] Gennady Pekhimenko[‡]
Eiman Ebrahimi[⌘] Nastaran Hajinazar[†] Phillip B. Gibbons[†] Onur Mutlu^{§†}

[†]Carnegie Mellon University

[‡]University of Toronto

[⌘]NVIDIA

[†]Simon Fraser University

[§]ETH Zürich

Expressive (Memory) Interfaces for GPUs

- Nandita Vijaykumar, Eiman Ebrahimi, Kevin Hsieh, Phillip B. Gibbons and Onur Mutlu, **"The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs"**
Proceedings of the 45th International Symposium on Computer Architecture (ISCA), Los Angeles, CA, USA, June 2018.
[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Talk Slides \(pptx\) \(pdf\)](#)]
[[Lightning Talk Video](#)]

The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs

Nandita Vijaykumar ^{†§}	Eiman Ebrahimi [‡]	Kevin Hsieh [†]
Phillip B. Gibbons [†]	Onur Mutlu ^{§†}	
[†] Carnegie Mellon University	[‡] NVIDIA	[§] ETH Zürich

Open-Source Frameworks for Data-Aware Systems

- Nandita Vijaykumar, Ataberk Olgun, Konstantinos Kanellopoulos, F. Nisa Bostanci, Hasan Hassan, Mehrshad Lotfi, Phillip B. Gibbons, and Onur Mutlu,
"MetaSys: A Practical Open-source Metadata Management System to Implement and Evaluate Cross-layer Optimizations"
ACM Transactions on Architecture and Code Optimization (TACO), June 2022.
[arXiv version]
Presented at the *18th HiPEAC Conference*, Toulouse, France, January 2023.
[Slides (pptx) (pdf)]
[Preliminary Talk Video (14 minutes)]
[SAFARI Live Seminar Video (1 hour 26 minutes)]
[MetaSys Source Code]
Best paper award at HiPEAC 2023.

MetaSys: A Practical Open-Source Metadata Management System to Implement and Evaluate Cross-Layer Optimizations

Nandita Vijaykumar* Ataberk Olgun[§] Konstantinos Kanellopoulos[§] Hasan Hassan[§]
Mehrshad Lotfi[§] Phillip B. Gibbons[†] Onur Mutlu[§]

*University of Toronto

[§]ETH Zürich

[†]Carnegie Mellon University

Heterogeneous-Reliability Memory

- Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, and Onur Mutlu,
"Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory"
Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, GA, June 2014. [[Summary](#)]
[[Slides \(pptx\)](#)] [[pdf](#)] [[Coverage on ZDNet](#)]

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

Yixin Luo Sriram Govindan* Bikash Sharma* Mark Santaniello* Justin Meza
Aman Kansal* Jie Liu* Badriddine Khessib* Kushagra Vaid* Onur Mutlu

Carnegie Mellon University, yixinluo@cs.cmu.edu, {meza, onur}@cmu.edu

*Microsoft Corporation, {srgovin, bsharma, marksan, kansal, jie.liu, bknessib, kvaid}@microsoft.com

EDEN: Data-Aware Efficient DNN Inference

- Skanda Koppula, Lois Orosa, A. Giray Yaglikci, Roknoddin Azizi, Taha Shahroodi, Konstantinos Kanellopoulos, and Onur Mutlu,
"EDEN: Enabling Energy-Efficient, High-Performance Deep Neural Network Inference Using Approximate DRAM"
Proceedings of the 52nd International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, October 2019.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Poster \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Video](#) (90 seconds)]
[[Full Talk Lecture](#) (38 minutes)]

EDEN: Enabling Energy-Efficient, High-Performance Deep Neural Network Inference Using Approximate DRAM

Skanda Koppula Lois Orosa A. Giray Yağlıkçı
Roknoddin Azizi Taha Shahroodi Konstantinos Kanellopoulos Onur Mutlu

ETH Zürich

SMASH: SW/HW Indexing Acceleration

- Konstantinos Kanellopoulos, Nandita Vijaykumar, Christina Giannoula, Roknoddin Azizi, Skanda Koppula, Nika Mansouri Ghiasi, Taha Shahroodi, Juan Gomez-Luna, and Onur Mutlu,

"SMASH: Co-designing Software Compression and Hardware-Accelerated Indexing for Efficient Sparse Matrix Operations"

Proceedings of the 52nd International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, October 2019.

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

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

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

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

[[Full Talk Lecture](#) (30 minutes)]

SMASH: Co-designing Software Compression and Hardware-Accelerated Indexing for Efficient Sparse Matrix Operations

Konstantinos Kanellopoulos¹ Nandita Vijaykumar^{2,1} Christina Giannoula^{1,3} Roknoddin Azizi¹
Skanda Koppula¹ Nika Mansouri Ghiasi¹ Taha Shahroodi¹ Juan Gomez Luna¹ Onur Mutlu^{1,2}

¹ETH Zürich

²Carnegie Mellon University

³National Technical University of Athens

Rethinking Virtual Memory

- Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,
"The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework"
Proceedings of the 47th International Symposium on Computer Architecture (ISCA), Valencia, Spain, June 2020.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]
[[ARM Research Summit Poster \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (26 minutes)]
[[Lightning Talk Video](#) (3 minutes)]

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

Nastaran Hajinazar^{*†} Pratyush Patel[✕] Minesh Patel^{*} Konstantinos Kanellopoulos^{*} Saugata Ghose[‡]
Rachata Ausavarungnirun[⊙] Geraldo F. Oliveira^{*} Jonathan Appavoo[◇] Vivek Seshadri[▽] Onur Mutlu^{*‡}

^{*}ETH Zürich [†]Simon Fraser University [✕]University of Washington [‡]Carnegie Mellon University
[⊙]King Mongkut's University of Technology North Bangkok [◇]Boston University [▽]Microsoft Research India

Data-Characteristic-Aware Computing Architectures

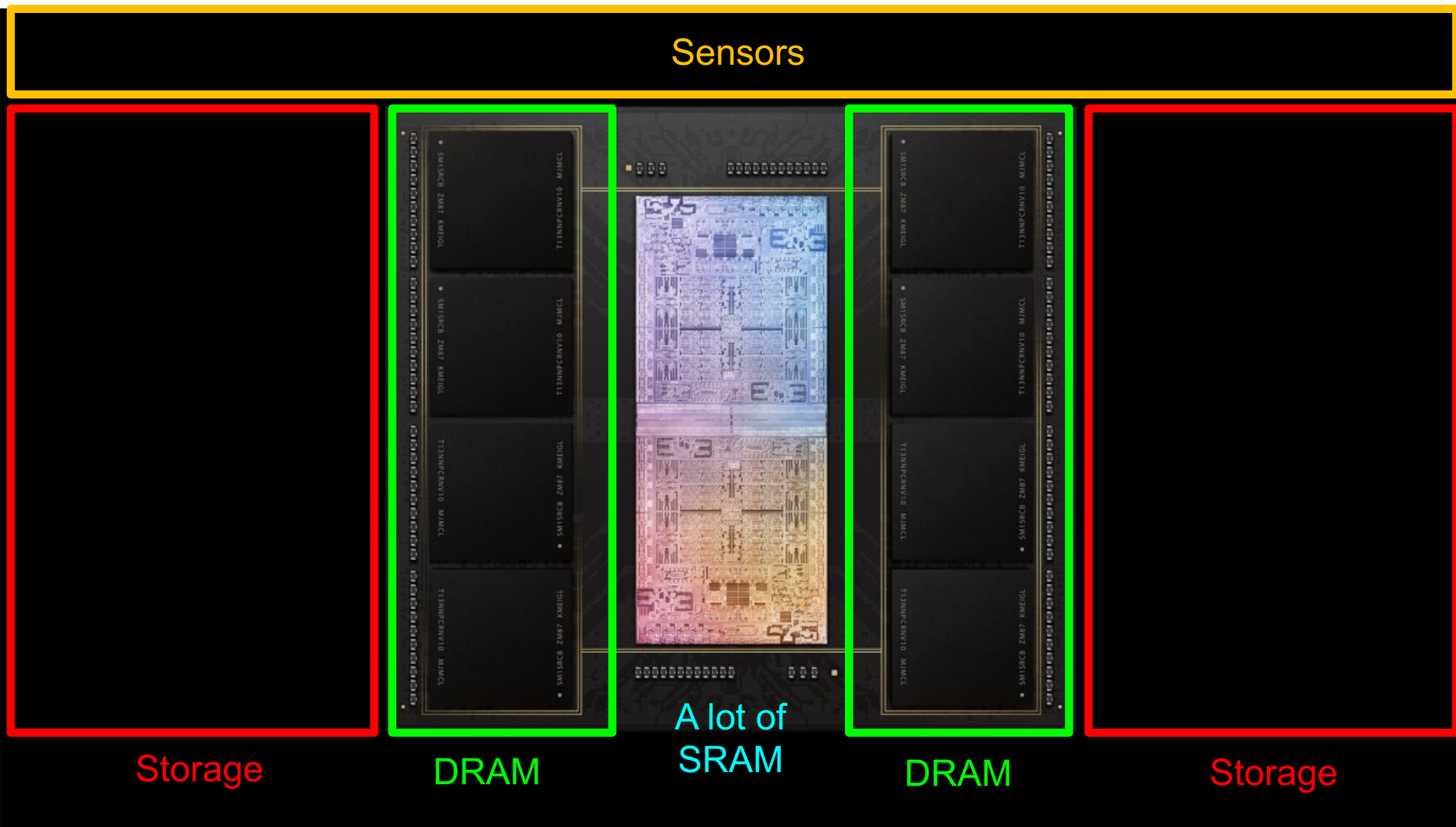
Data-Centric (Memory-Centric) Architectures

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

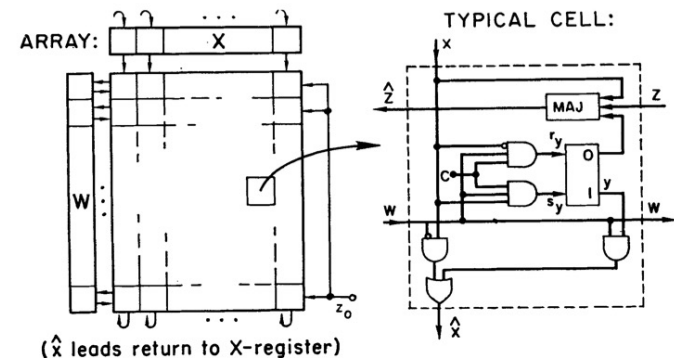
Cellular Logic-in-Memory Arrays

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**

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

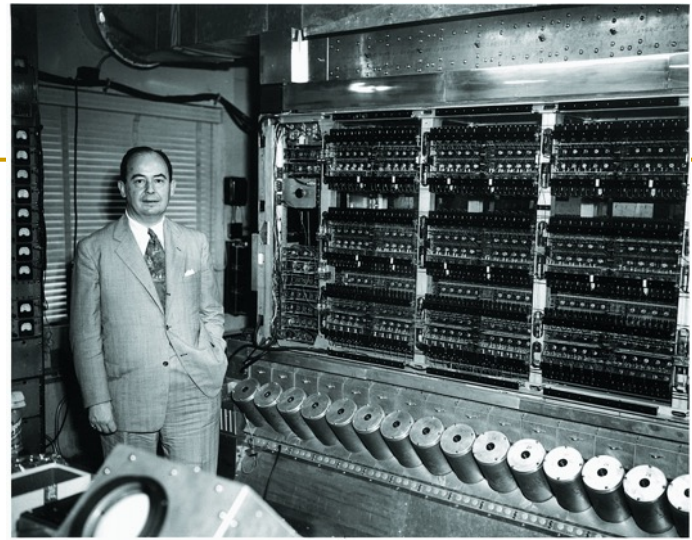
Our current
design principles
cause great energy waste
(and great performance loss)

The Problem

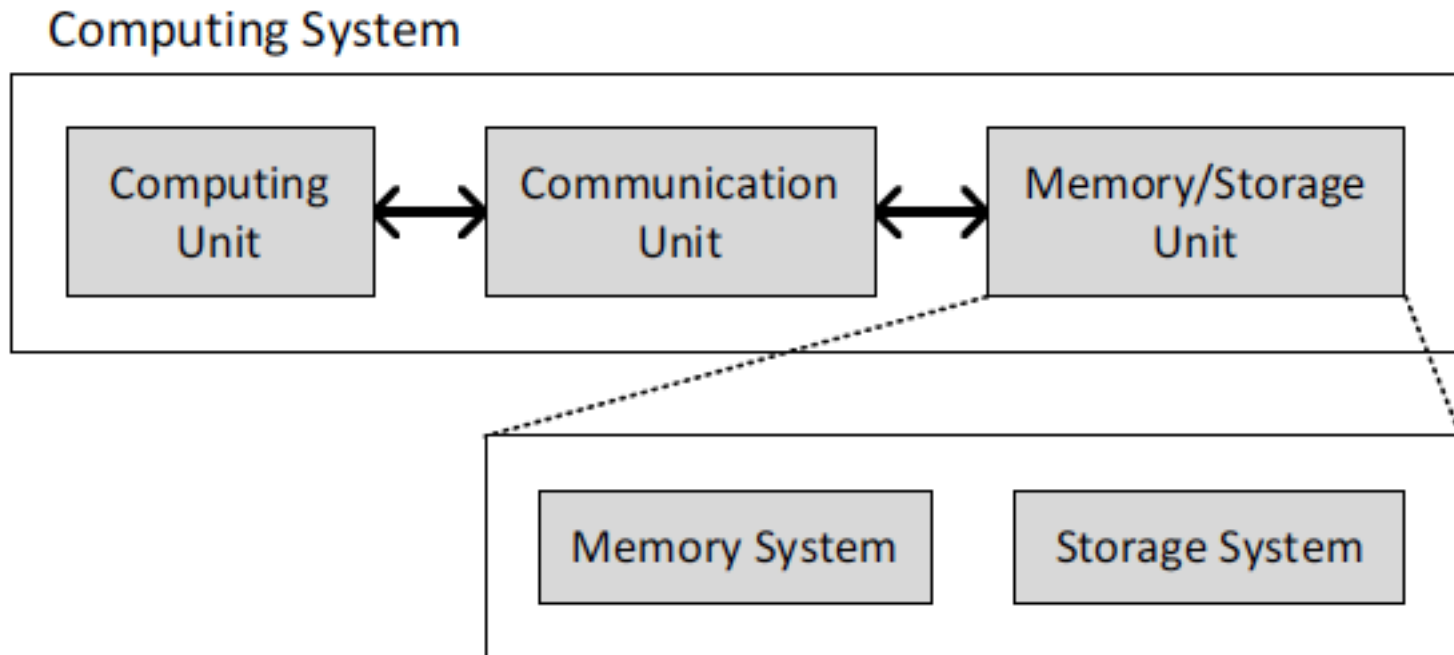
Processing of data
is performed
far away from the data

A Computing System

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

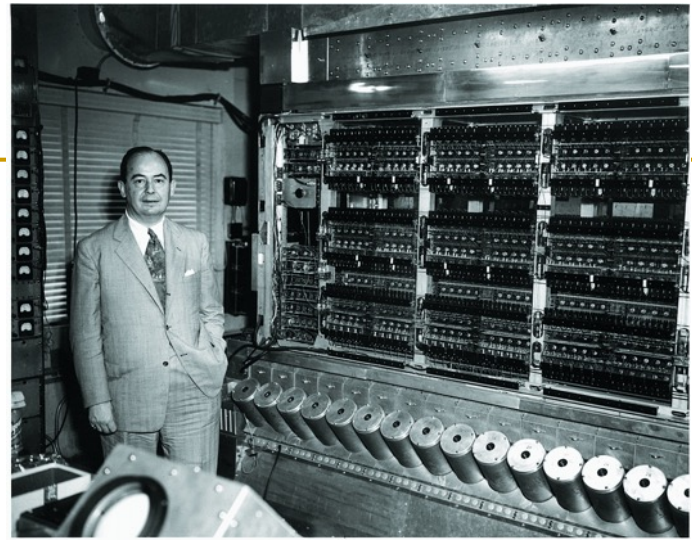


Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.



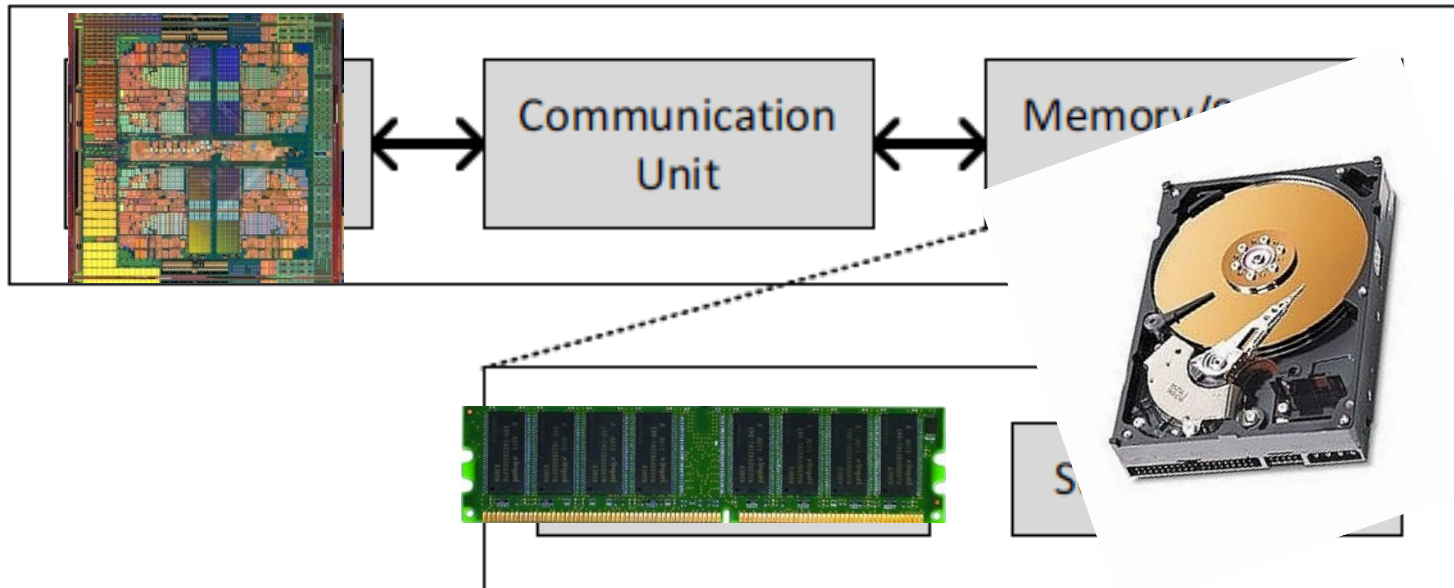
A Computing System

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



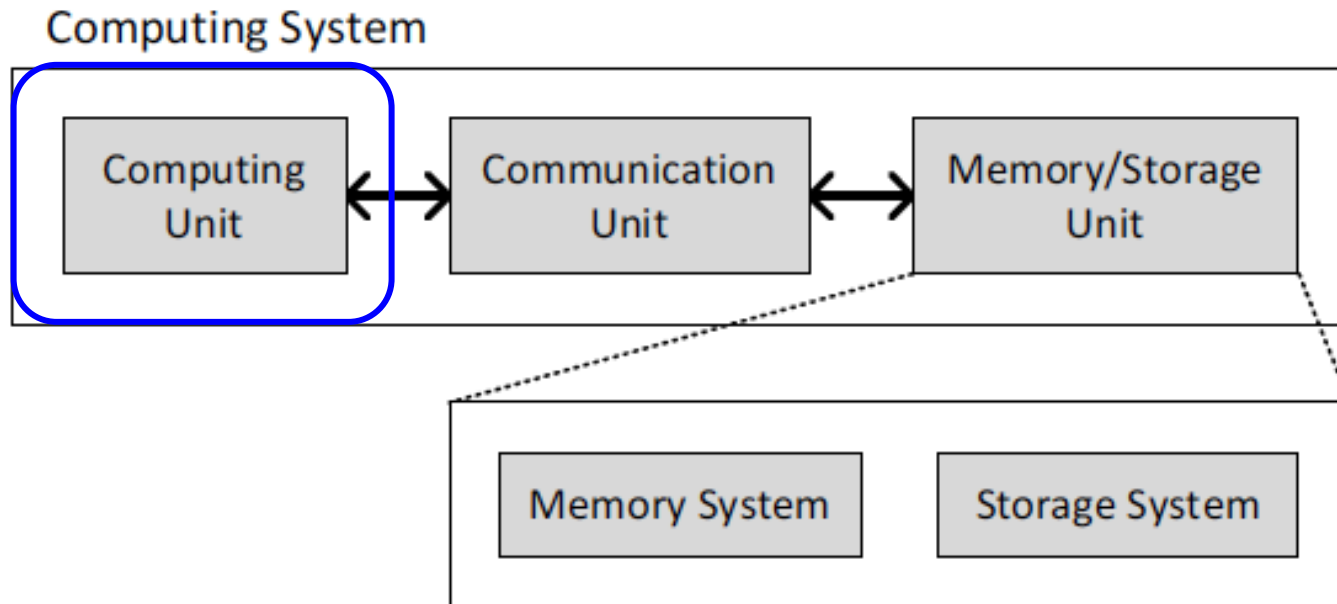
Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.

Computing System



Today's Computing Systems

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



Energy Waste in Mobile Devices

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

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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

Energy Waste in Accelerators

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

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

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

Amirali Boroumand^{†◇}

Geraldo F. Oliveira^{*}

Saugata Ghose[‡]

Xiaoyu Ma[§]

Berkin Akin[§]

Eric Shiu[§]

Ravi Narayanaswami[§]

Onur Mutlu^{*†}

[†]Carnegie Mellon Univ.

[◇]Stanford Univ.

[‡]Univ. of Illinois Urbana-Champaign

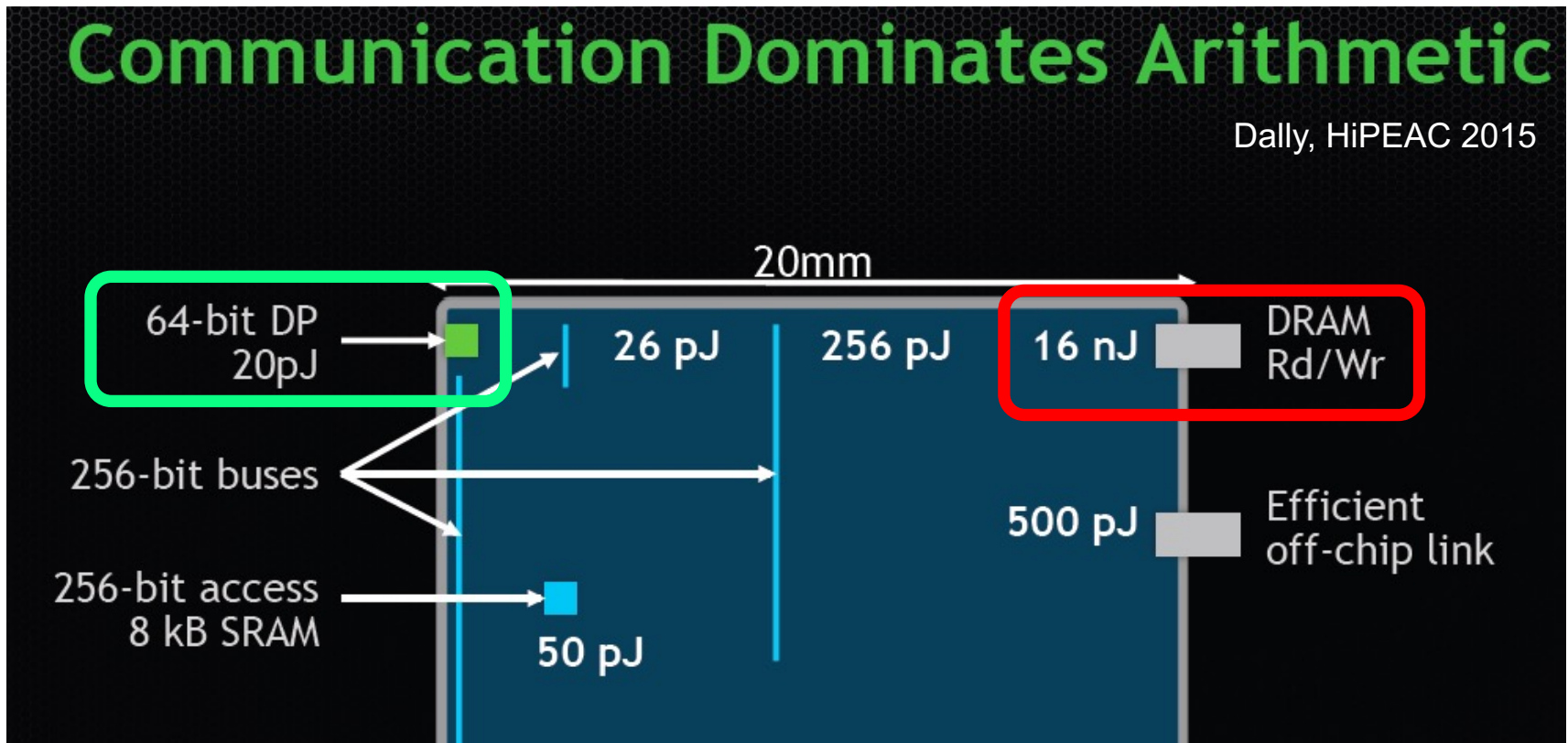
[§]Google

^{*}ETH Zürich

We Do Not Want to Move Data!

Communication Dominates Arithmetic

Dally, HiPEAC 2015

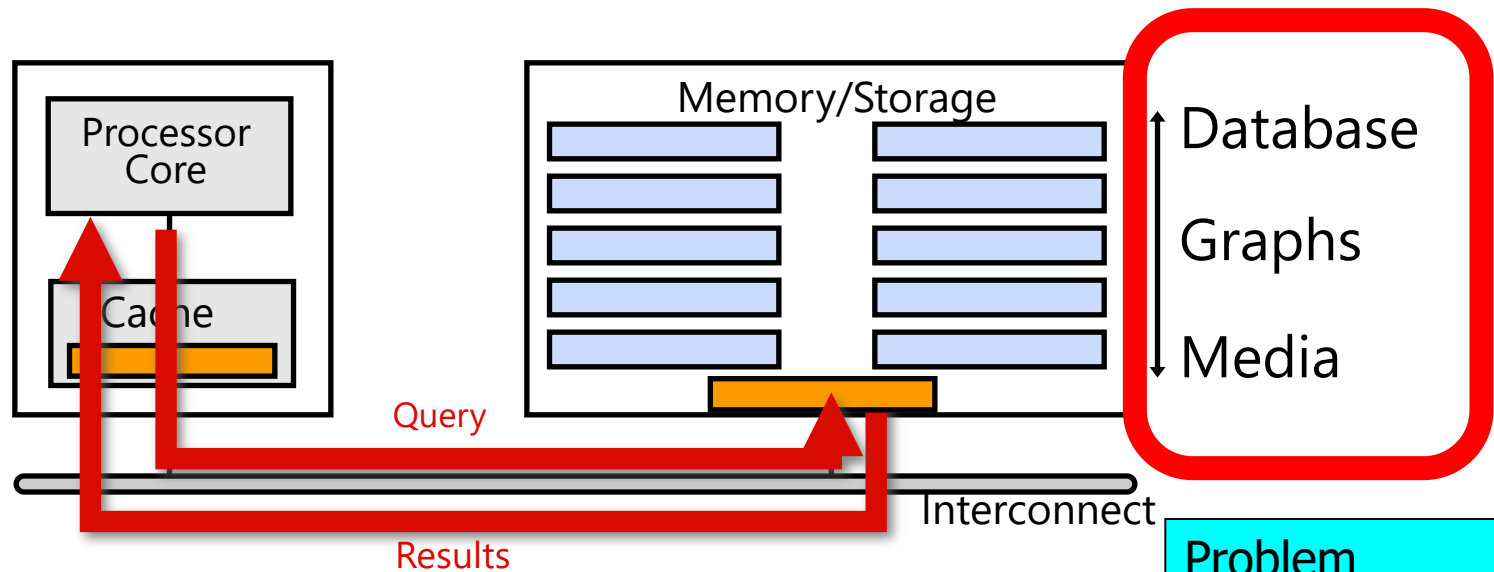


A memory access consumes $\sim 100-1000X$ the energy of a complex addition

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/Storage



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

Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Logic
Devices
Electrons

PIM Review and Open Problems

A Modern Primer on Processing in Memory

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

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

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

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

"A Modern Primer on Processing in Memory"

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

A Modern Primer on Processing in Memory

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

SAFARI Research Group

^aETH Zürich

^bCarnegie Mellon University

^cUniversity of Illinois at Urbana-Champaign

^dKing 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

1 Introduction	2
2 Major Trends Affecting Main Memory	4
3 The Need for Intelligent Memory Controllers to Enhance Memory Scaling	6
4 Perils of Processor-Centric Design	9
5 Processing-in-Memory (PIM): Technology Enablers and Two Approaches	12
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
6 Processing Using Memory (PUM)	14
6.1 RowClone	14
6.2 Ambit	15
6.3 Gather-Scatter DRAM	17
6.4 In-DRAM Security Primitives	17
7 Processing Near Memory (PNM)	18
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
8 Enabling the Adoption of PIM	24
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
9 Conclusion and Future Outlook	31

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 (Fall 2022)

■ Fall 2022 Edition:

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

■ Spring 2022 Edition:

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

■ Youtube Livestream (Fall 2022):

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

■ Youtube Livestream (Spring 2022):

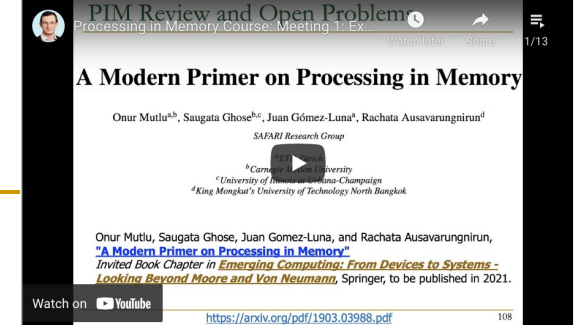
- <https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYUK9EsXKhQKRPyX>

■ Project course

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

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

SAFARI



Spring 2022 Meetings/Schedule

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

SSD Course (Spring 2023)

Spring 2023 Edition:

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

Fall 2022 Edition:

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

Youtube Livestream (Spring 2023):

- https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB

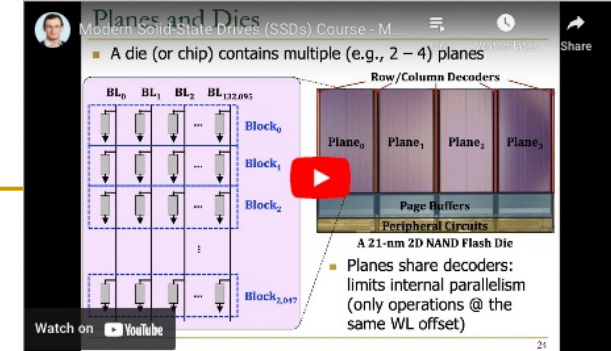
Youtube Livestream (Fall 2022):

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

Project course

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

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



Fall 2022 Meetings/Schedule

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

Genomics Course (Fall 2022)

Fall 2022 Edition:

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

Spring 2022 Edition:

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

Youtube Livestream (Fall 2022):

- https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFIQvdxOdizD_EhVAMVQV

Youtube Livestream (Spring 2022):

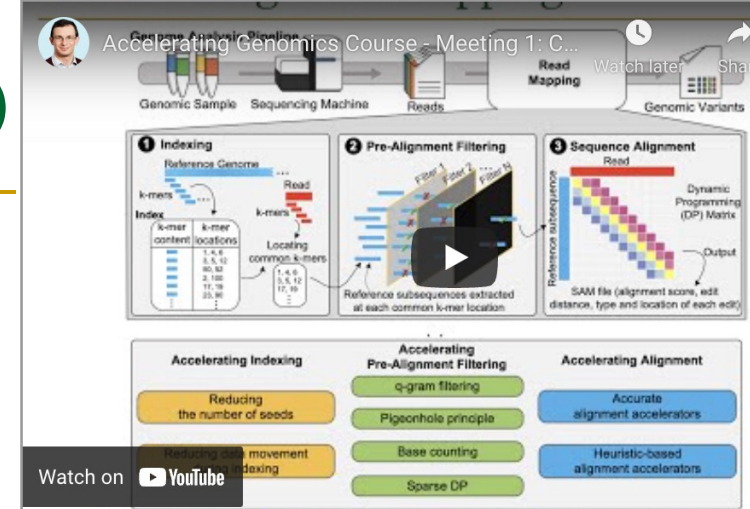
- https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18

Project course

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

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

SAFARI



Spring 2022 Meetings/Schedule

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

We Need to Think Differently
from the Past Approaches

Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory

In-Storage Genomic Data Filtering [ASPLOS 2022]

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

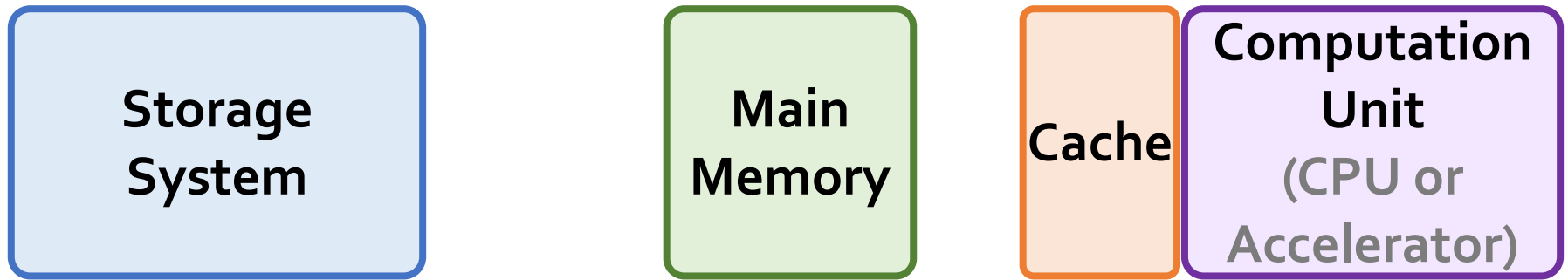
GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

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

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

Genome Sequence Analysis

Data Movement from Storage

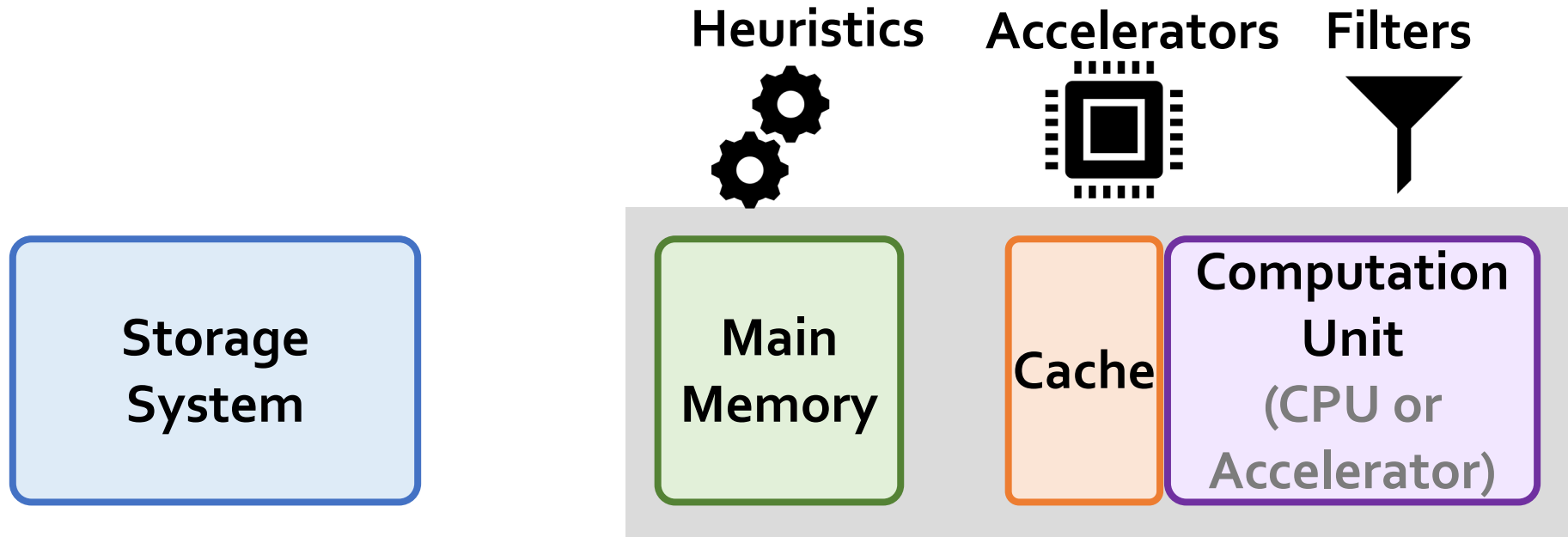


Computation overhead



Data movement overhead

Compute-Centric Accelerators



Computation overhead

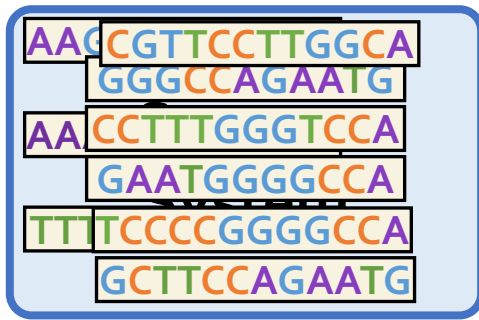


Data movement overhead

Key Idea: In-Storage Filtering



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



Filtered Reads

**Main
Memory**

Cache

**Computation
Unit**
(CPU or
Accelerator)

Exactly-matching reads

Do not need expensive approximate string matching during alignment

Non-matching reads

Do not have potential matching locations and can skip alignment

GenStore



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

GenStore-Enabled
Storage
System

Main
Memory

Cache

Computation
Unit
(CPU or
Accelerator)



Computation overhead



Data movement overhead

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

In-Storage Genomic Data Filtering [ASPLOS 2022]

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

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

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

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

Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory

In-Flash Bulk Bitwise Execution

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

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

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

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

Summary: Flash-Cosmos



The first work that enables
in-flash multi-operand bulk bitwise operations
with a single sensing operation and high reliability



Improves performance
by 32x/25x/3.5x over OSP/ISP/ParaBit



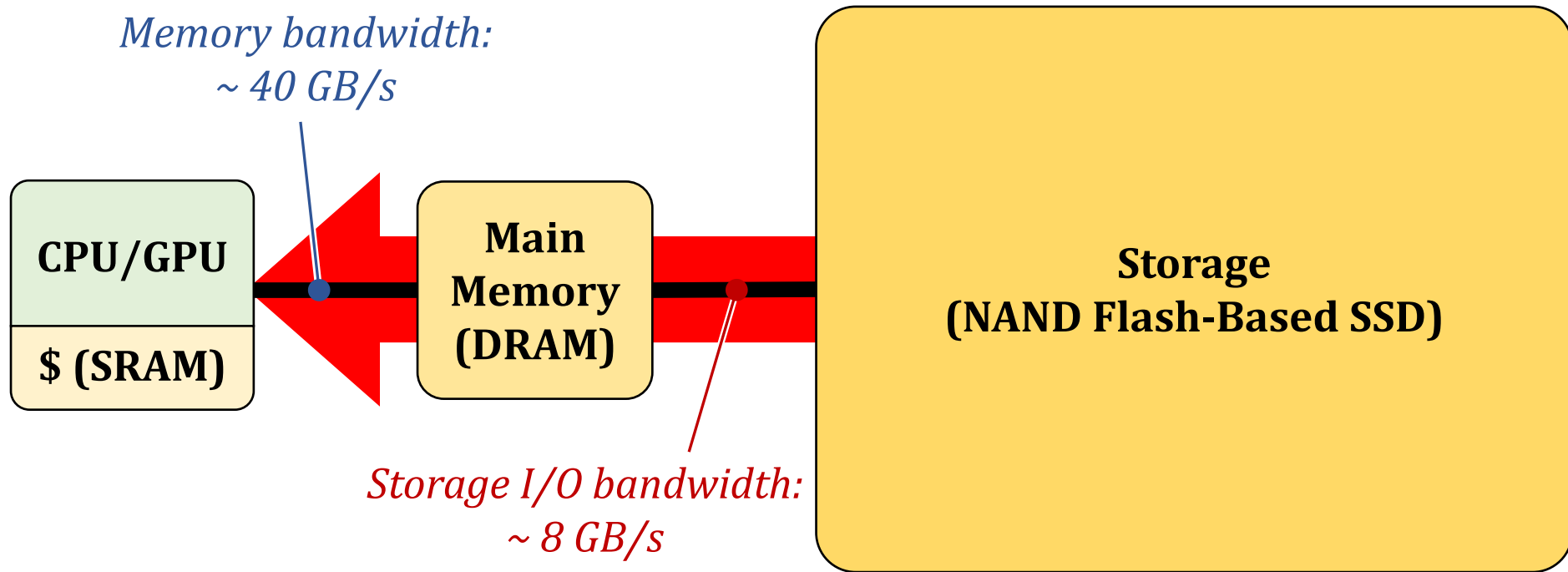
Improves energy efficiency
by 95x/13.4x/3.3x over OSP/ISP/ParaBit



Low-cost & requires no changes to flash cell arrays

Data-Movement Bottleneck

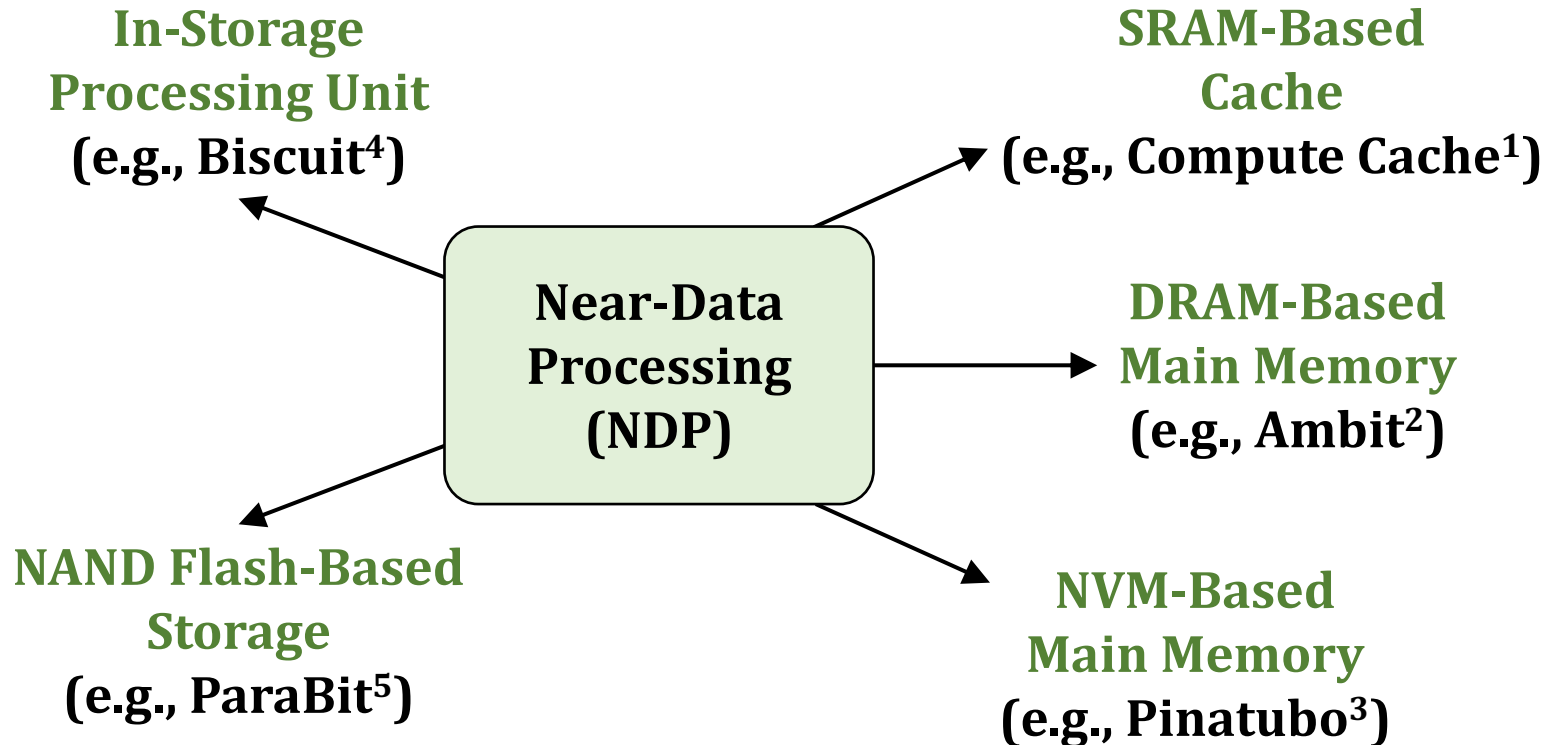
- Conventional systems: Outside-storage processing (OSP) that must move the entire data to CPUs/GPUs through the memory hierarchy



External I/O bandwidth of storage systems is the **main bottleneck** in conventional systems (OSP)

Near-Data Processing for Bitwise Operations

- Can **effectively mitigate** data movement by performing simple bitwise operations **where the data resides**



¹Aga+, "Compute Caches," HPCA, 2017

²Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO, 2017

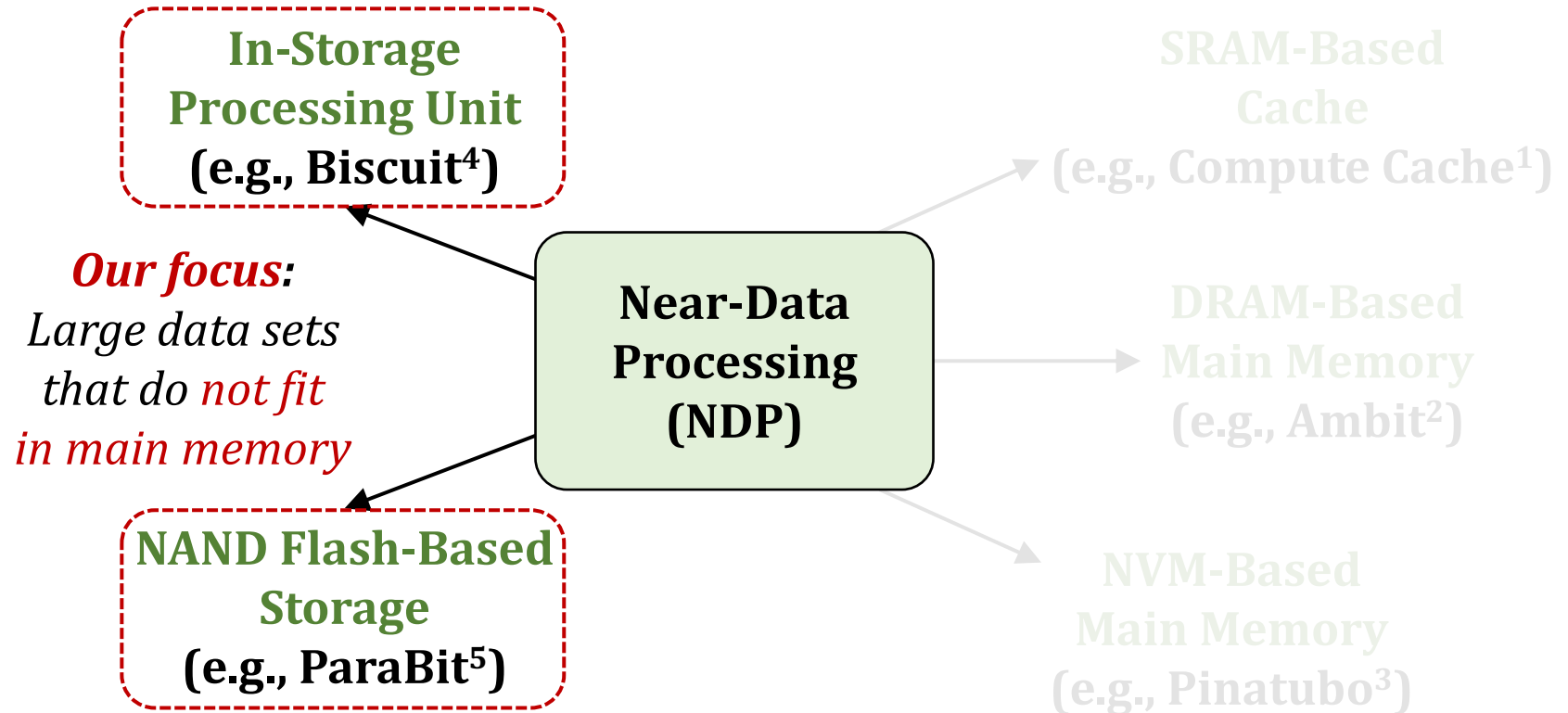
³Li+, "Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-Volatile Memories," DAC, 2016

⁴Gu+, "Biscuit: A Framework for Near-Data Processing of Big Data Workloads," ISCA, 2016

⁵Gao+, "ParaBit: Processing Parallel Bitwise Operations in NAND Flash Memory Based SSDs," MICRO, 2021

Near-Data Processing for Bitwise Operations

- Can **effectively mitigate** data movement by performing simple bitwise operations **where the data resides**



¹Aga+, "Compute Caches," HPCA, 2017

²Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO, 2017

³Li+, "Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-Volatile Memories," DAC, 2016

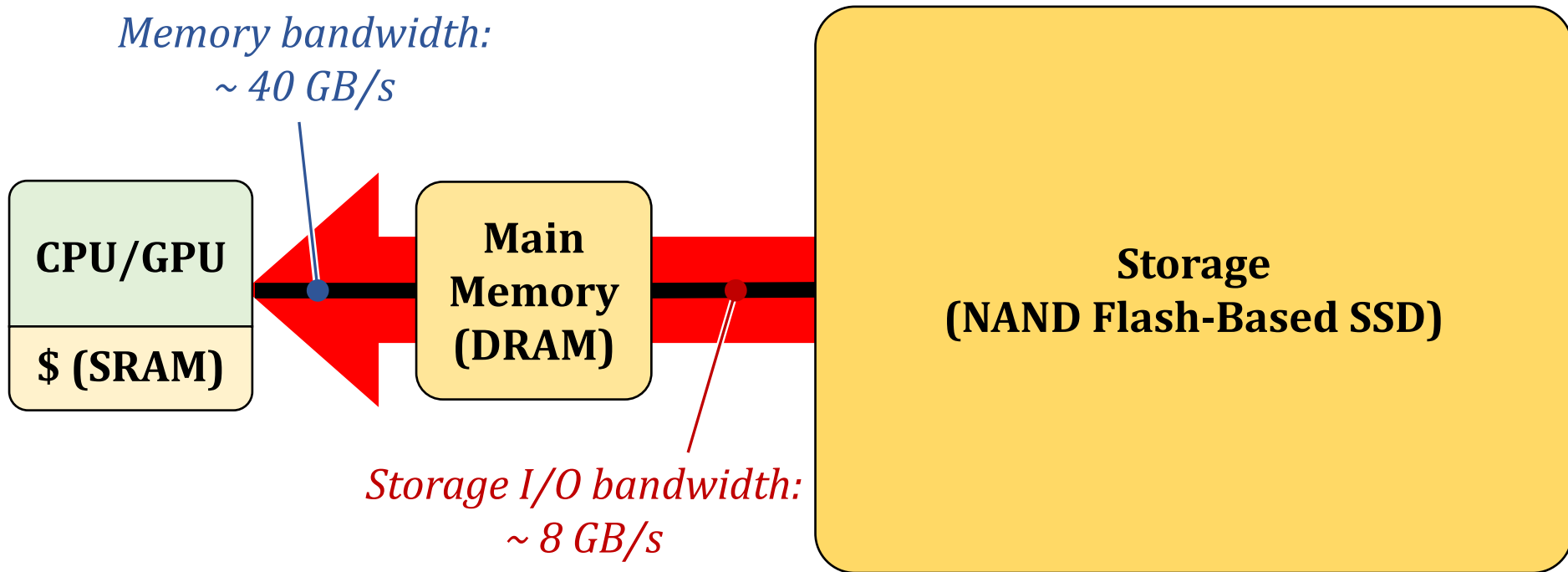
⁴Gu+, "Biscuit: A Framework for Near-Data Processing of Big Data Workloads," ISCA, 2016

⁵Gao+, "ParaBit: Processing Parallel Bitwise Operations in NAND Flash Memory Based SSDs," MICRO, 2021

In-Storage Processing (ISP)

- Uses **in-storage compute units** (embedded cores or FPGA) to send **only the computation results**

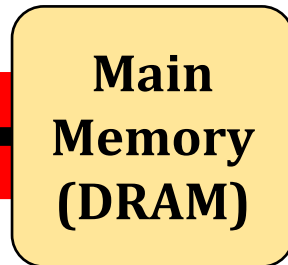
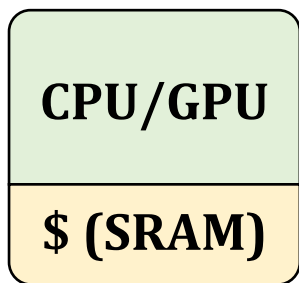
*Memory bandwidth:
~ 40 GB/s*



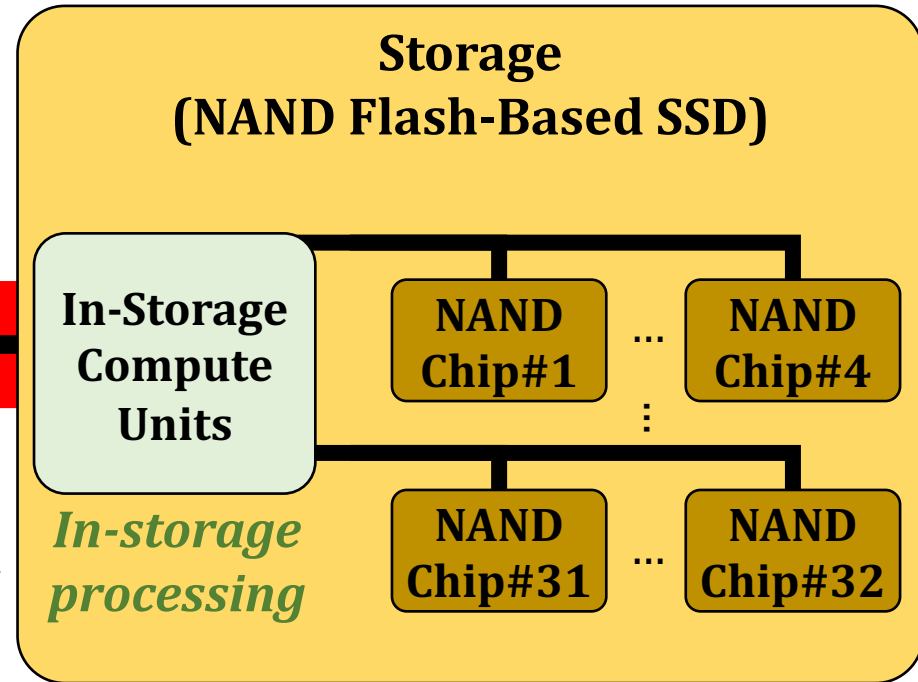
In-Storage Processing (ISP)

- Uses **in-storage compute units** (embedded cores or FPGA) to send **only the computation results**

*Memory bandwidth:
~ 40 GB/s*

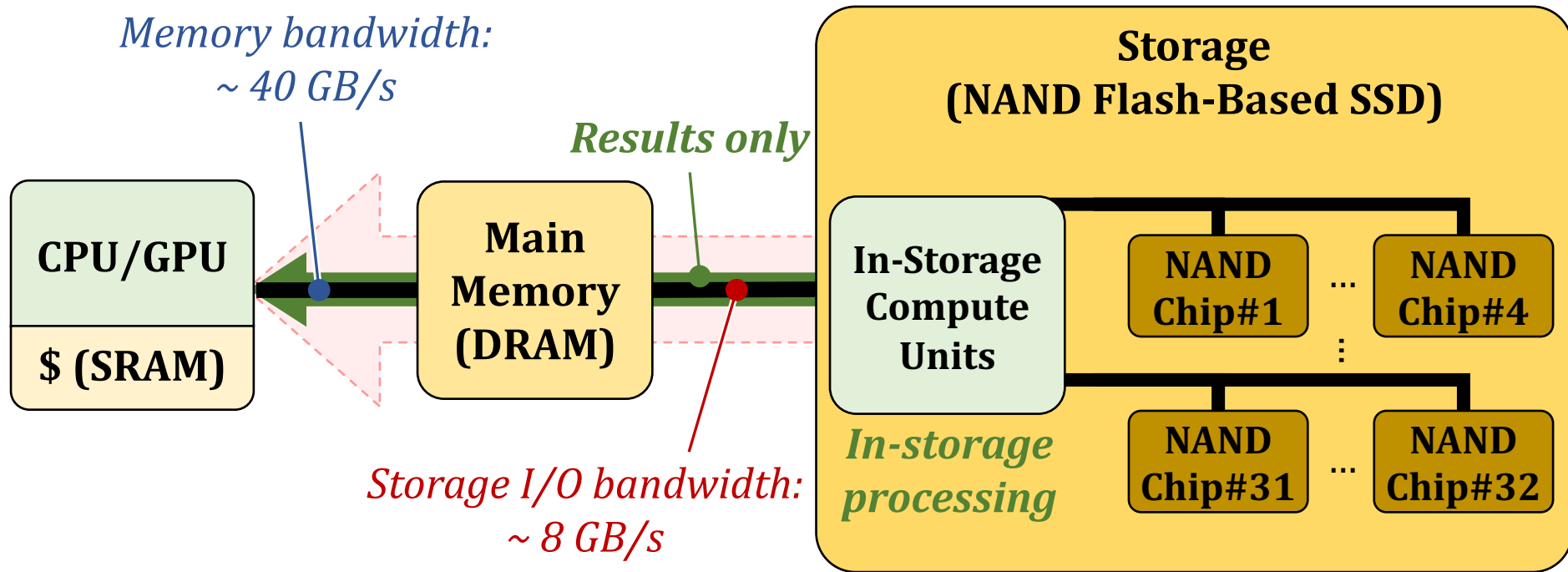


*Storage I/O bandwidth:
~ 8 GB/s*



In-Storage Processing (ISP)

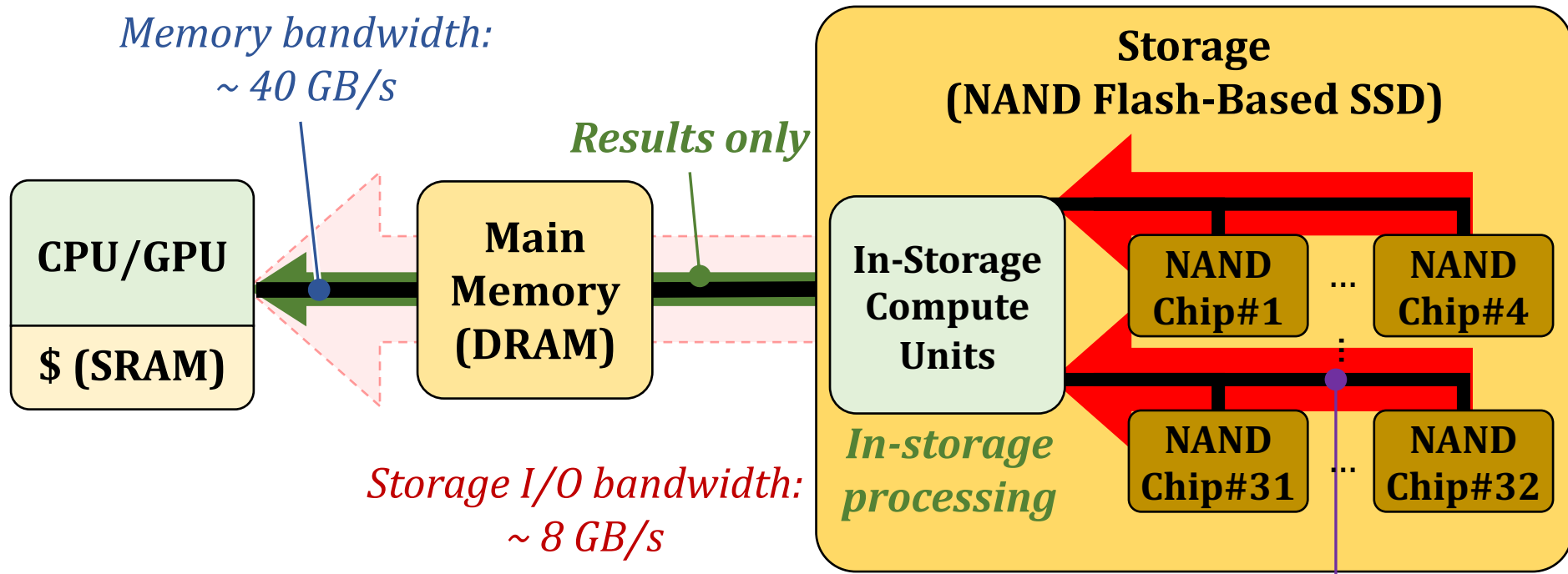
- Uses **in-storage compute units** (embedded cores or FPGA) to send **only the computation results**



ISP can mitigate data movement overhead by **reducing SSD-external data movement**

In-Storage Processing (ISP)

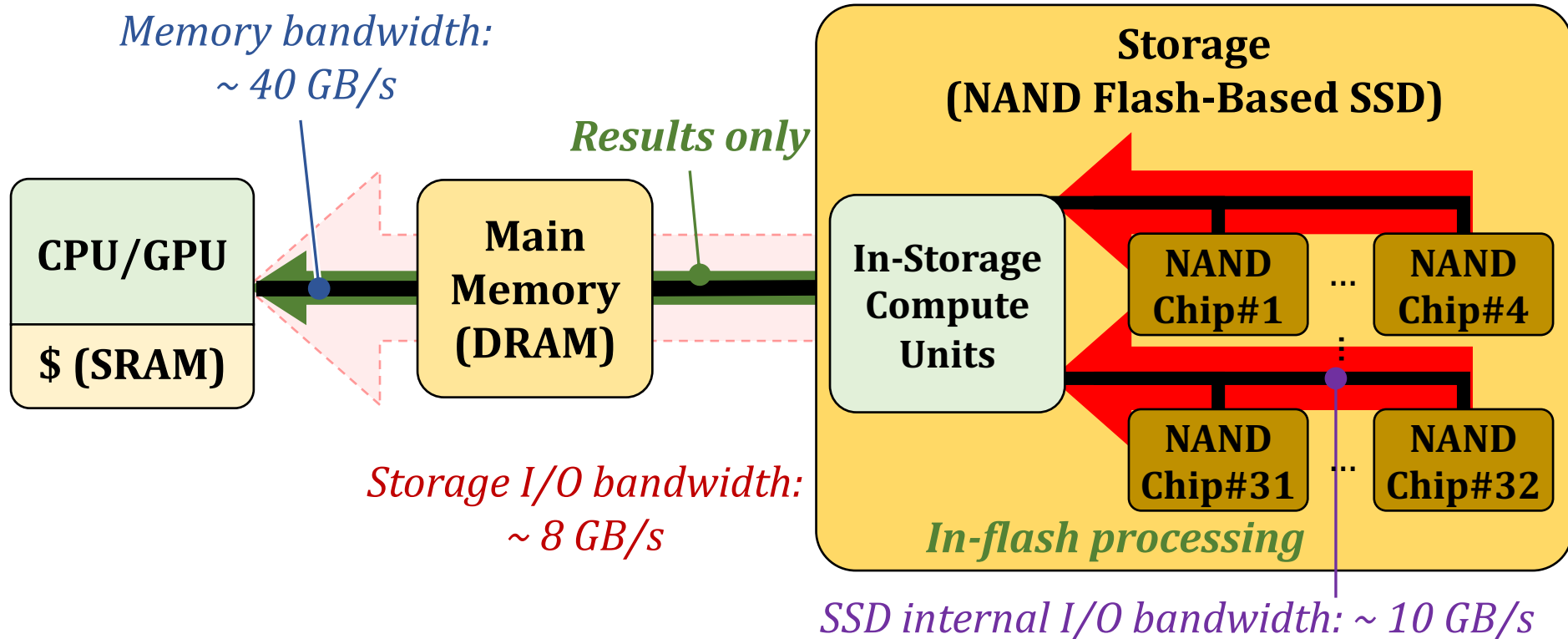
- Uses **in-storage compute units** (embedded cores or FPGA) to send **only the computation results**



SSD-internal bandwidth
becomes the **new bottleneck** in ISP

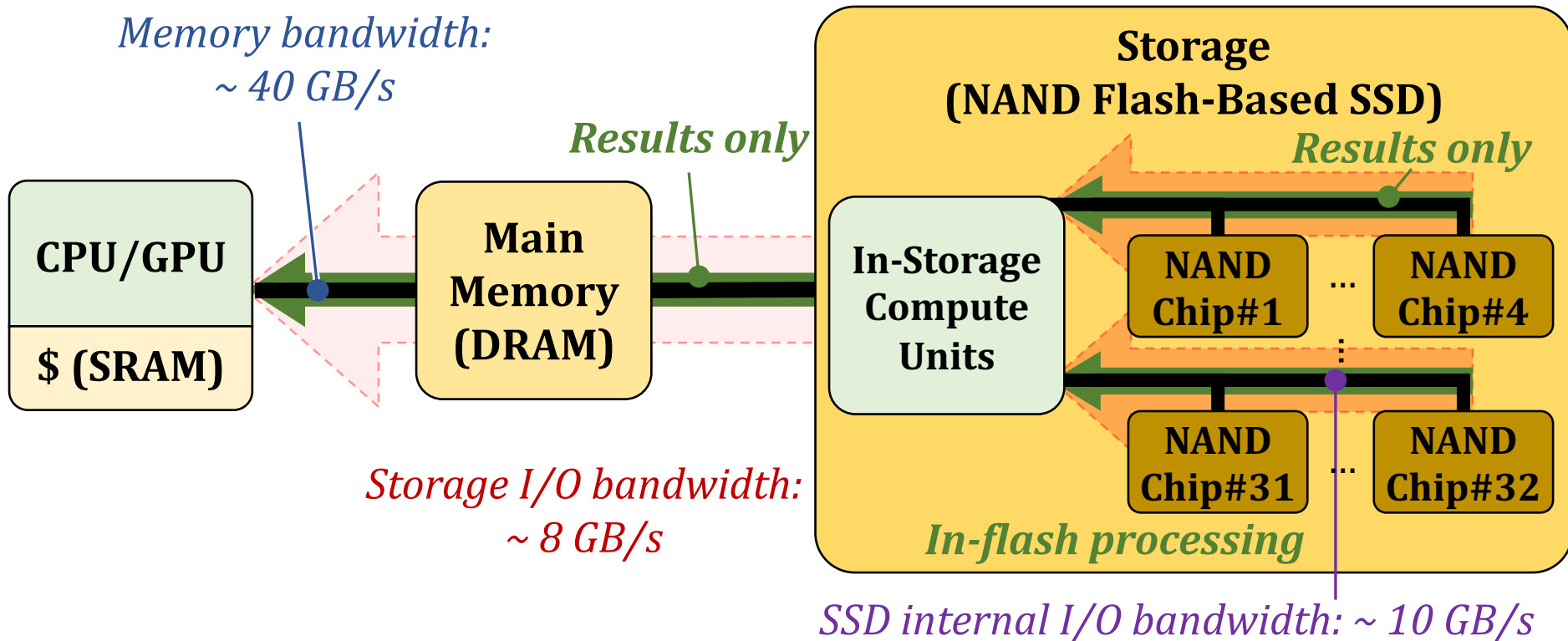
In-Flash Processing (IFP)

- Performs computation *inside* NAND flash chips



In-Flash Processing (IFP)

- Performs computation *inside* NAND flash chips

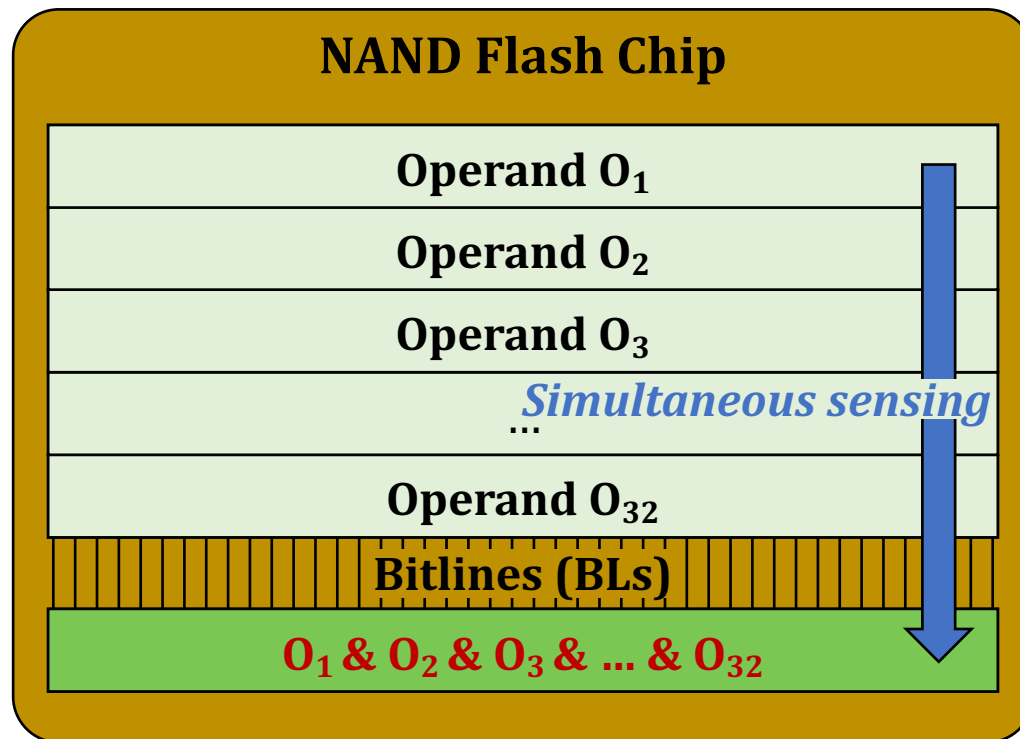


IFP fundamentally mitigates data movement

Our Proposal: Flash-Cosmos

▪ Flash-Cosmos enables

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



Key Ideas of Flash-Cosmos



Multi-Wordline Sensing (MWS)

to enable in-flash bulk bitwise operations via a single sensing operation



Enhanced SLC-Mode Programming (ESP)

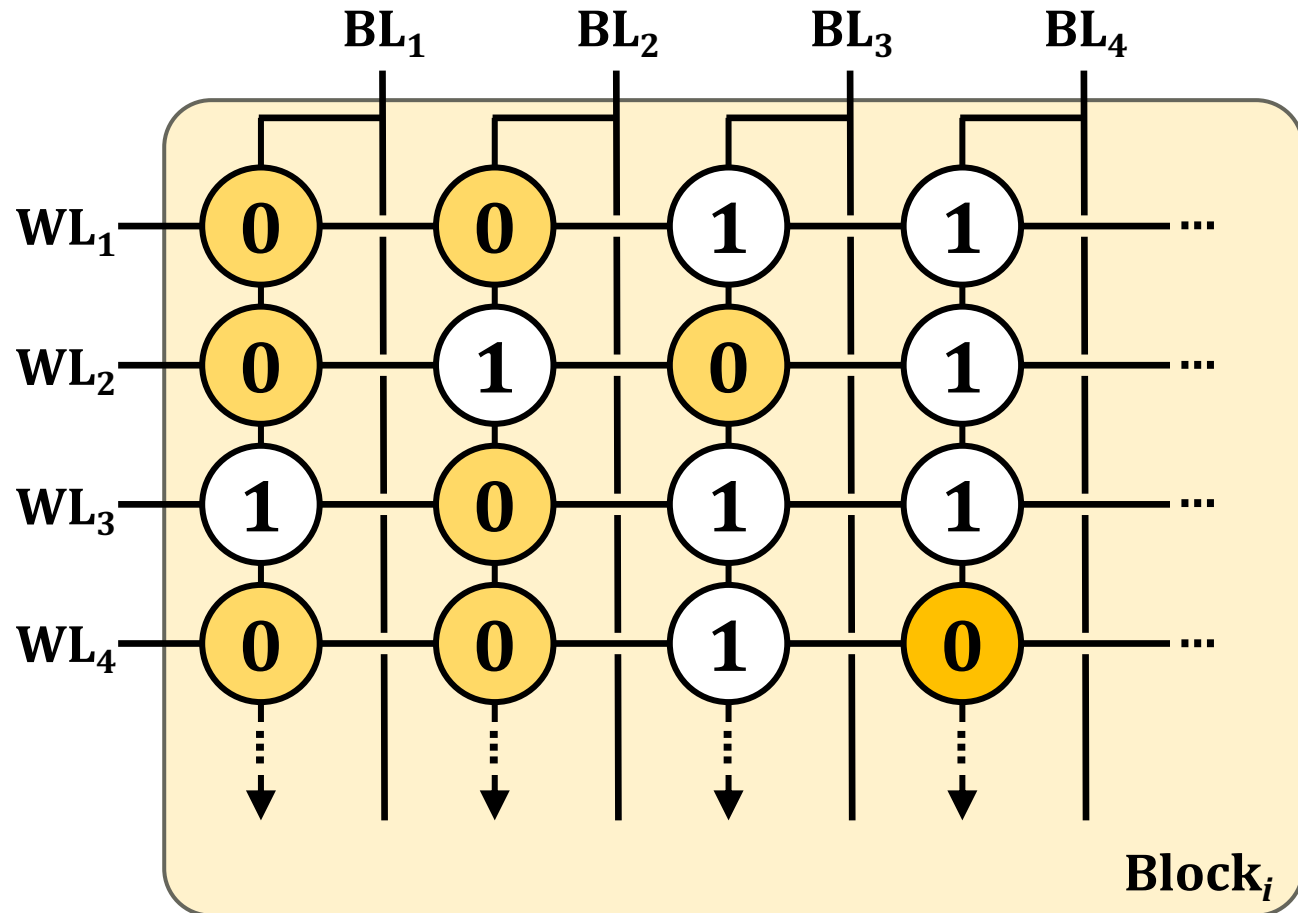
to eliminate raw bit errors in stored data (and thus in computation results)

Multi-Wordline Sensing (MWS): Bitwise AND

- **Intra-Block MWS:**

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs

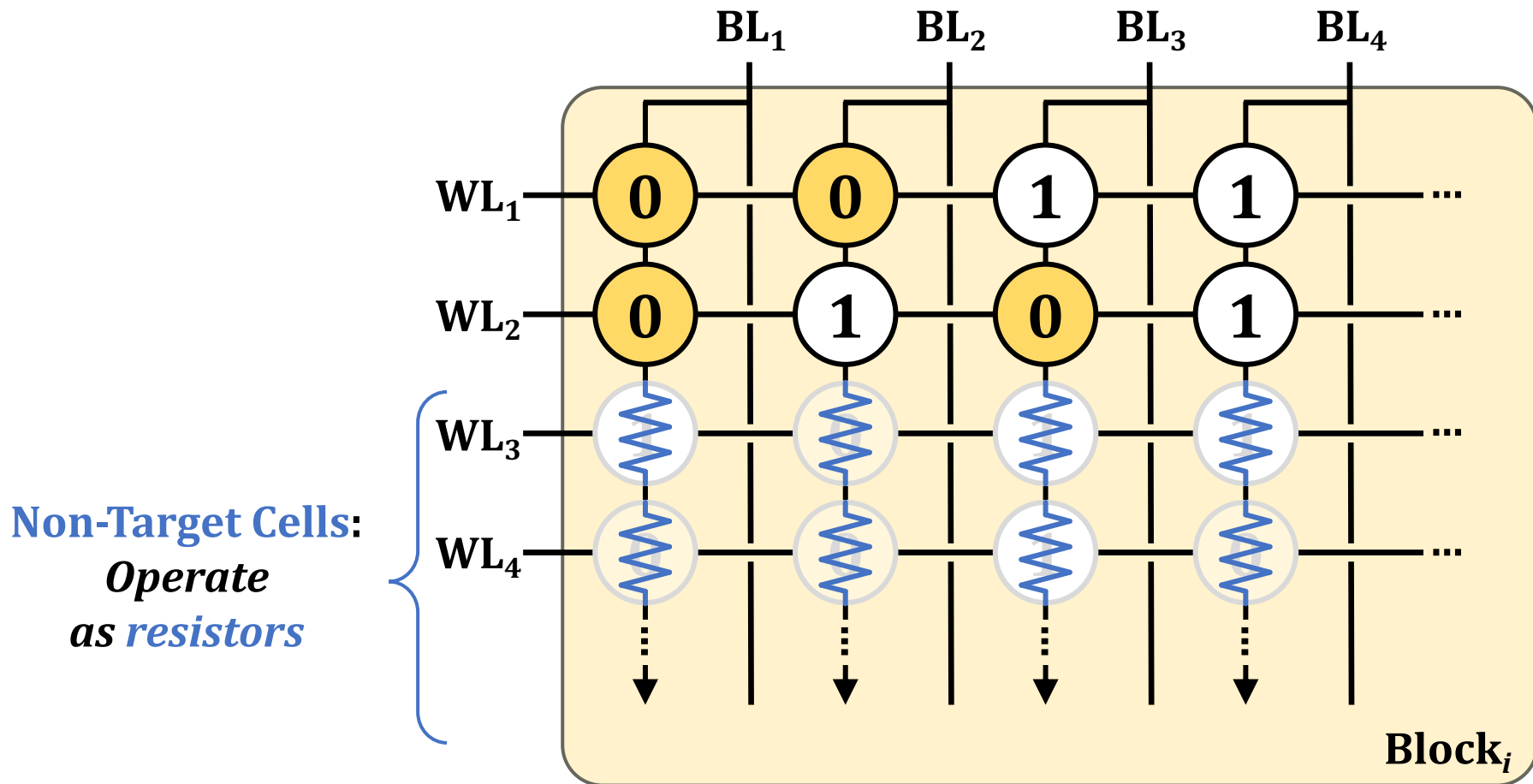


Multi-Wordline Sensing (MWS): Bitwise AND

■ Intra-Block MWS:

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs

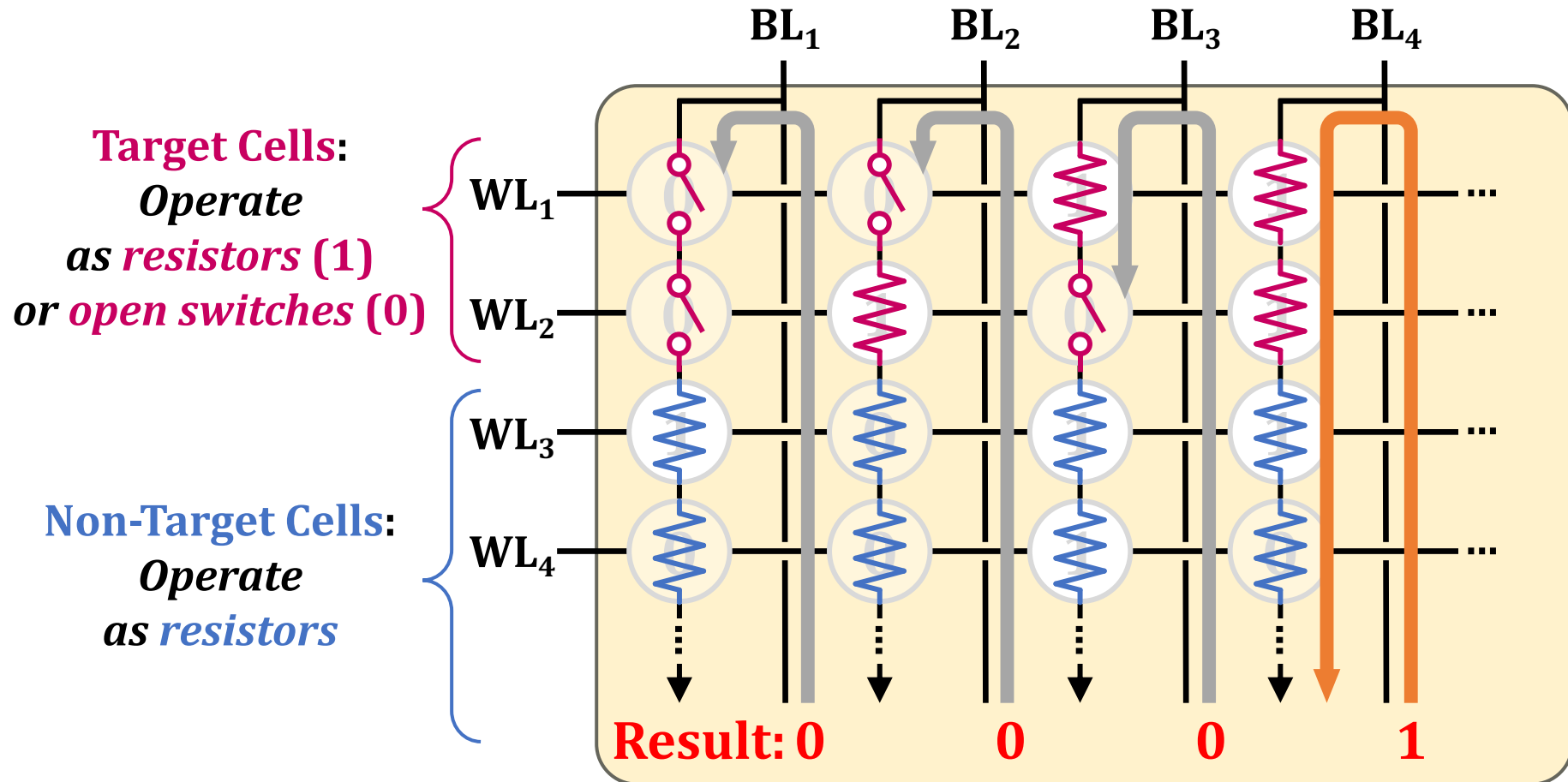


Multi-Wordline Sensing (MWS): Bitwise AND

■ Intra-Block MWS:

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs



Multi-Wordline Sensing (MWS): Bitwise AND

■ Intra-Block MWS:

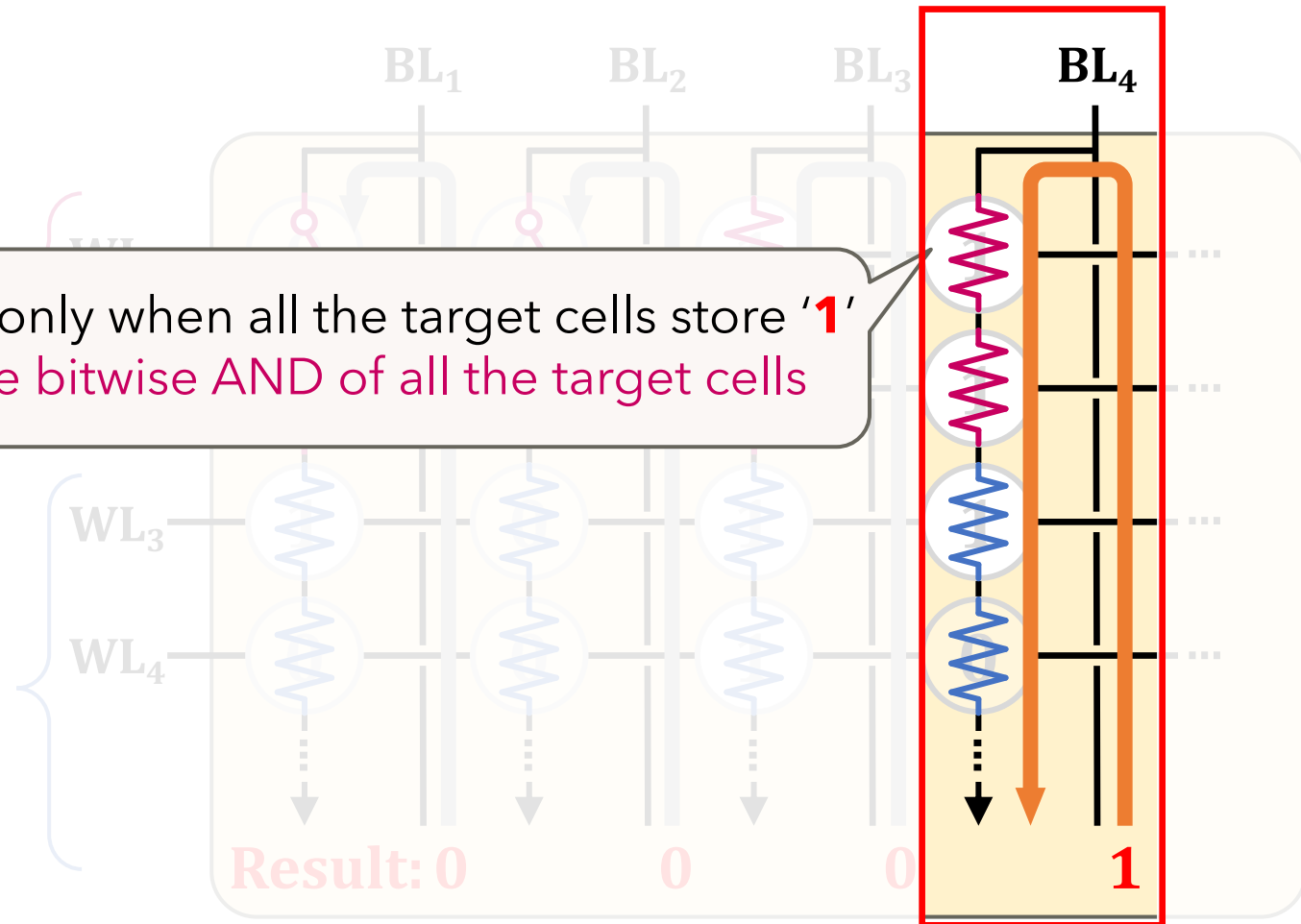
Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs

Target Cell:

A bitline reads as '1' only when all the target cells store '1'
→ Equivalent to the bitwise AND of all the target cells

Non-Target Cell:
*Operate
as a resistance*

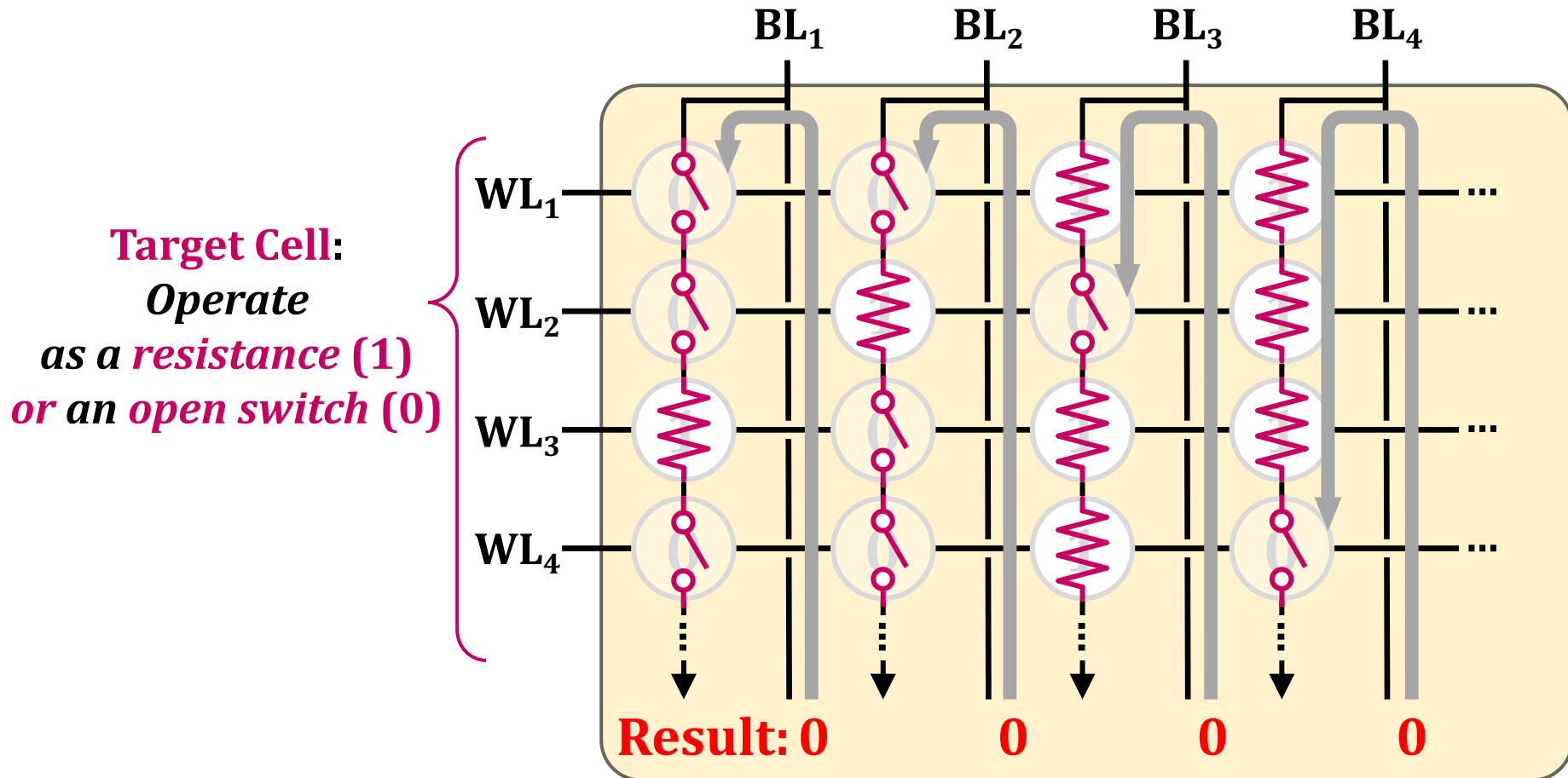


Multi-Wordline Sensing (MWS): Bitwise AND

- **Intra-Block MWS:**

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs



Multi-Wordline Sensing (MWS): Bitwise AND

■ Intra-Block MWS:

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs

A bitline reads as '**1**' only when all the target cells store '**1**'
→ Equivalent to the bitwise AND of all the target cells

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

WL₂

WL₃

WL₄

BL₁

BL₂

BL₃

BL₄

Result: 0

0

0

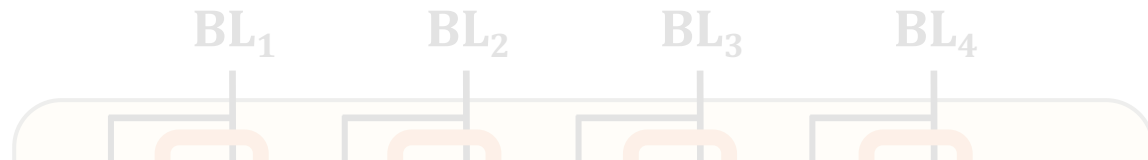
0

Multi-Wordline Sensing (MWS): Bitwise AND

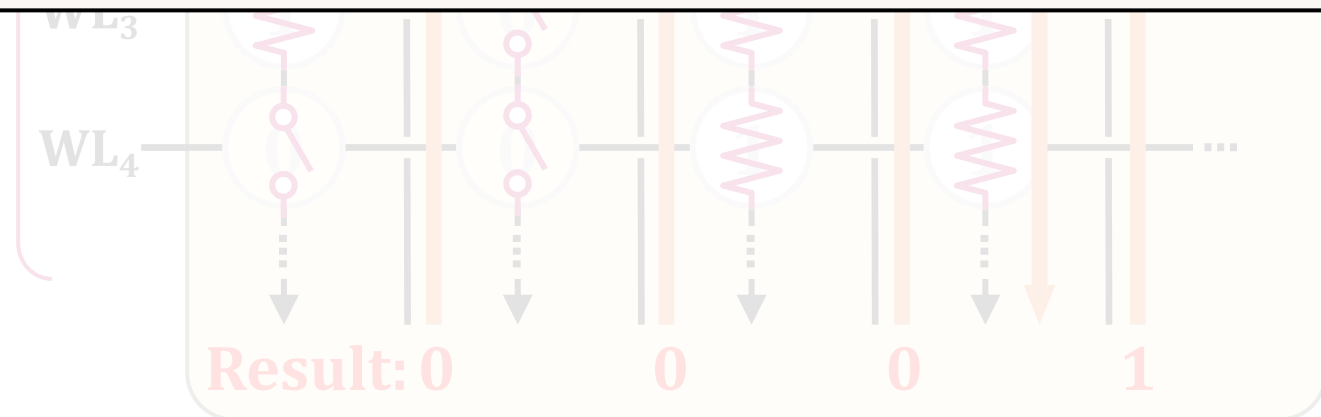
- Intra-Block MWS:

Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs



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

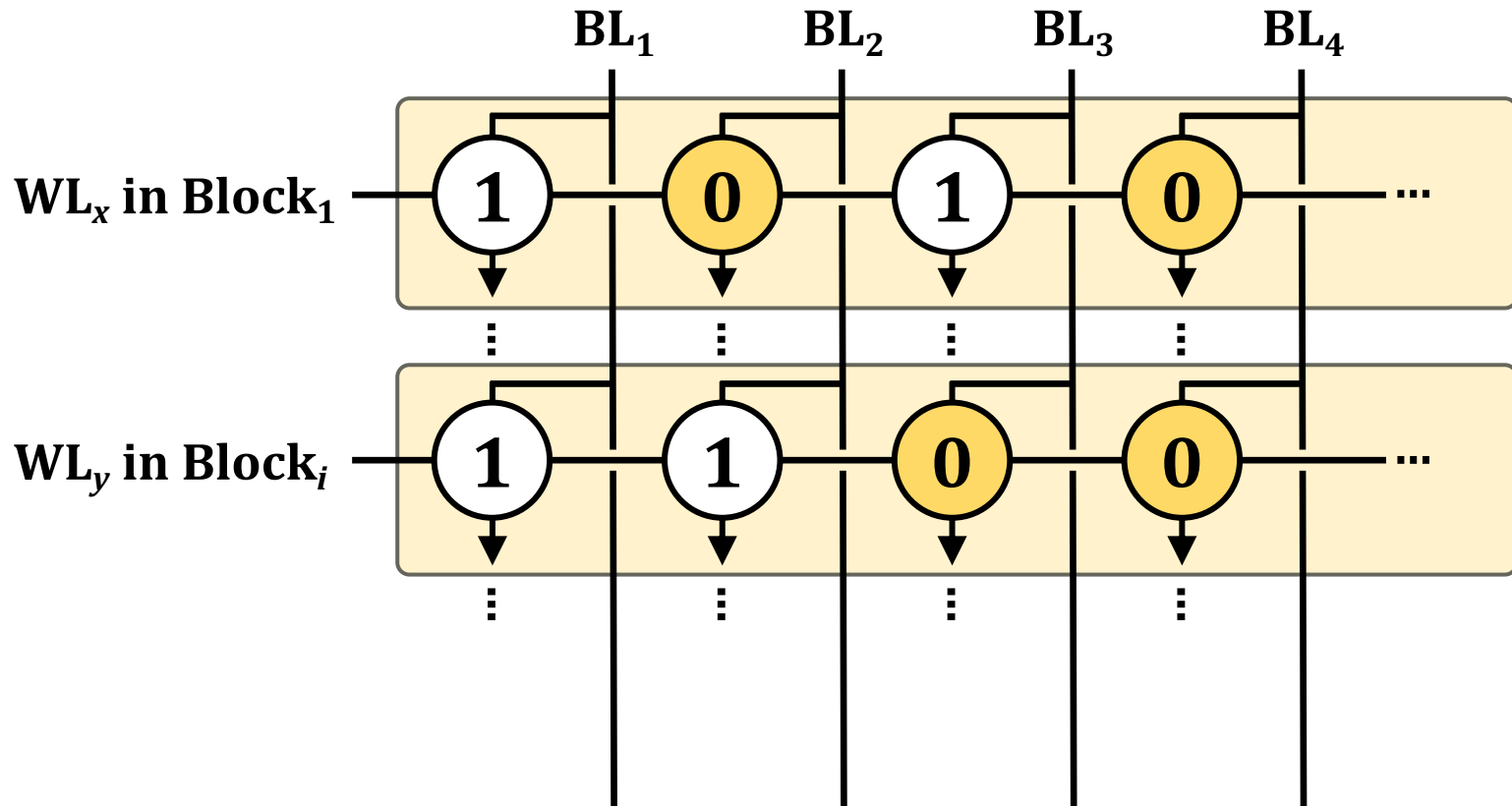


Multi-Wordline Sensing (MWS): Bitwise OR

■ Inter-Block MWS:

Simultaneously activates multiple WLs in different blocks

→ Bitwise OR of the stored data in the WLs

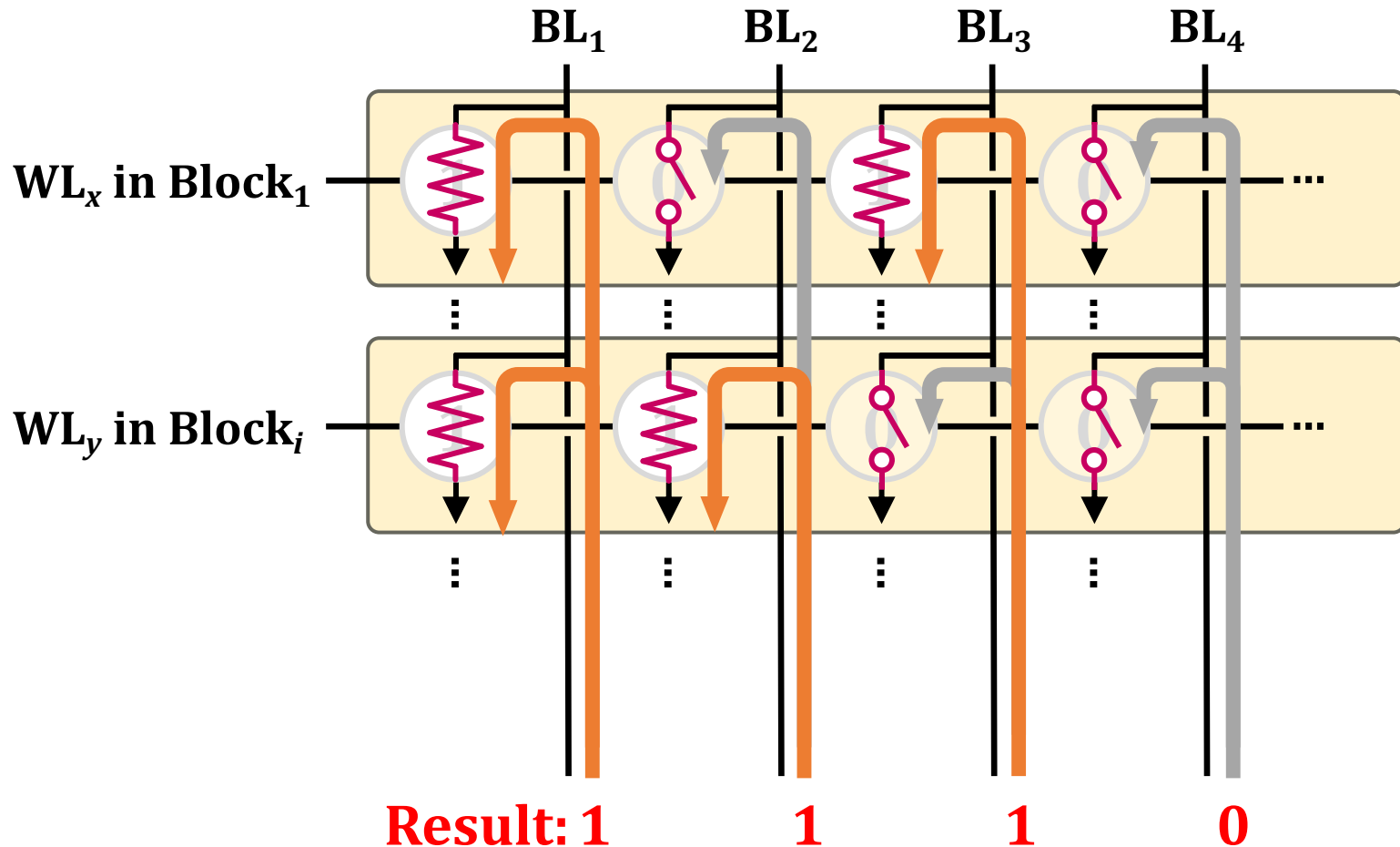


Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:**

Simultaneously activates multiple WLs in different blocks

→ Bitwise OR of the stored data in the WLs

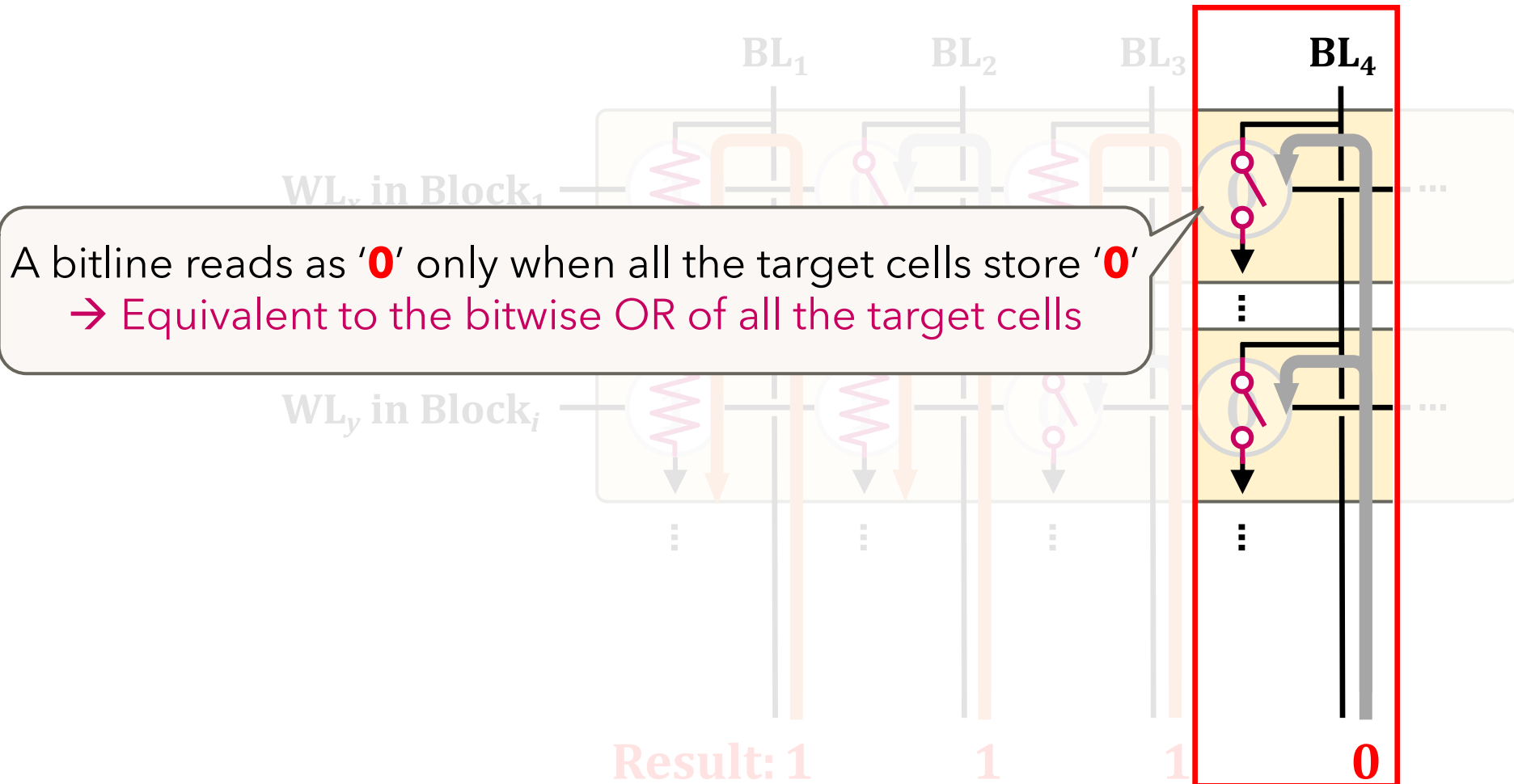


Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:**

Simultaneously activates multiple WLs in different blocks

→ Bitwise OR of the stored data in the WLs

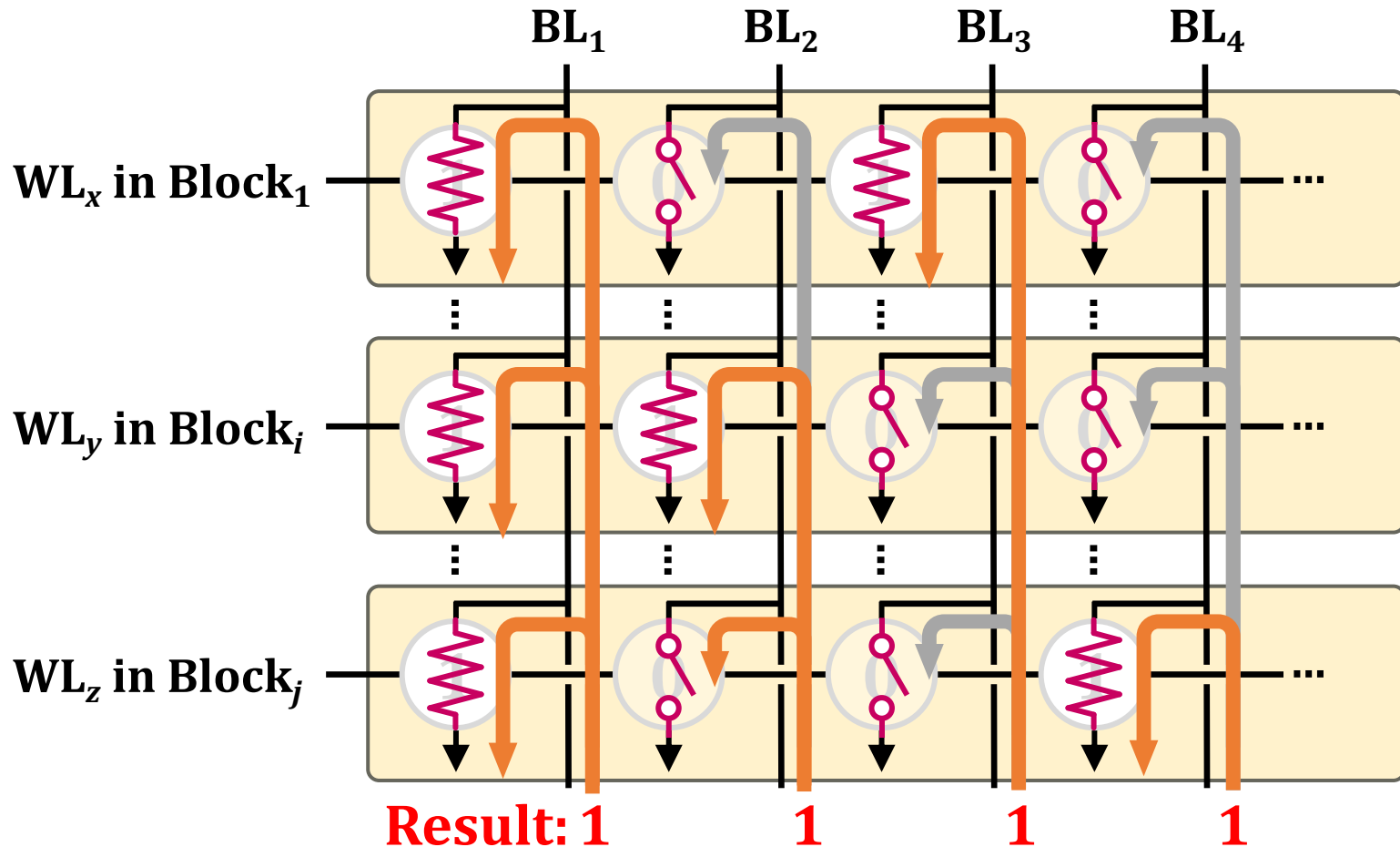


Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:**

Simultaneously activates multiple WLs in different blocks

→ Bitwise OR of the stored data in the WLs



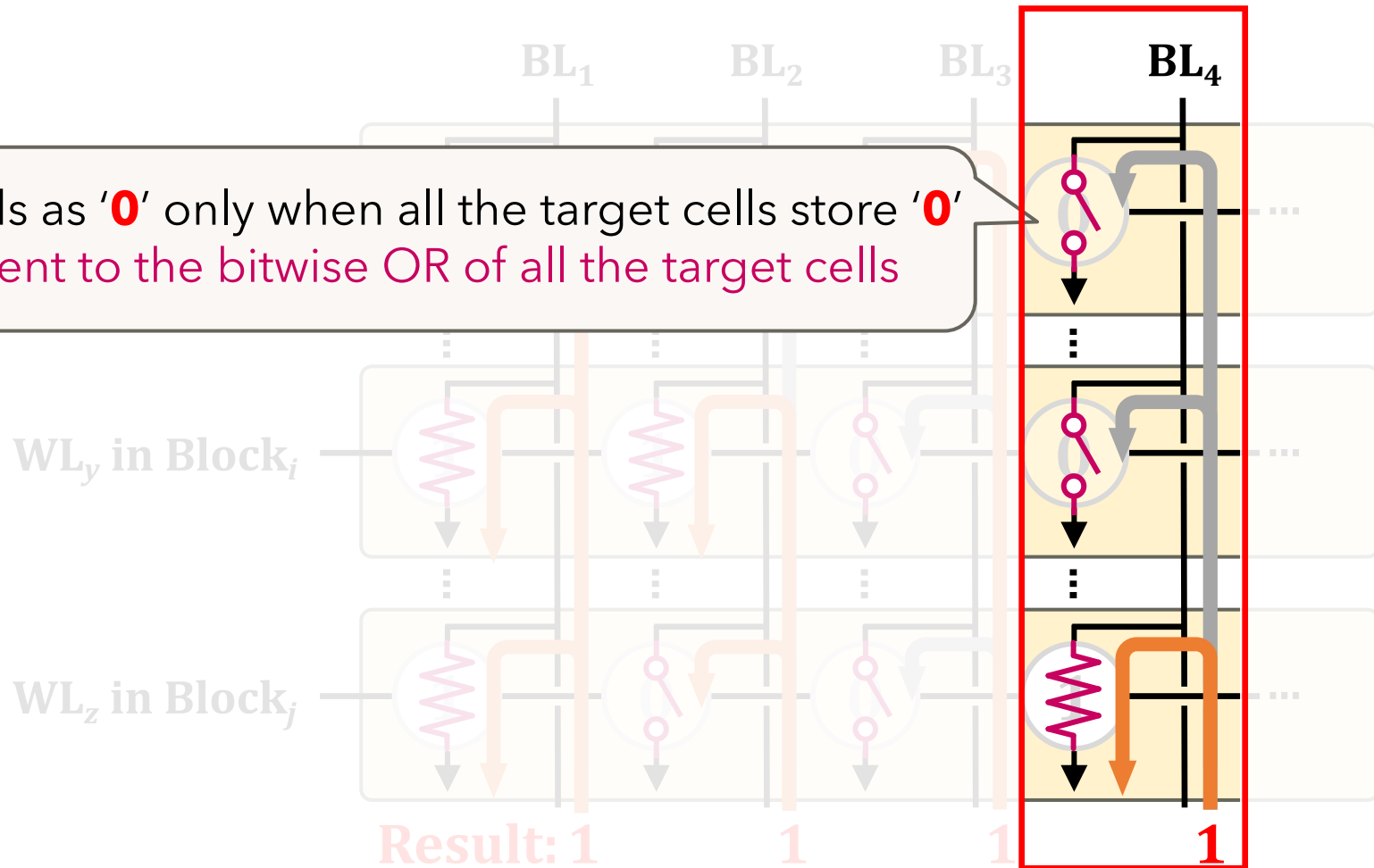
Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:**

Simultaneously activates multiple WLs in different blocks

→ Bitwise OR of the stored data in the WLs

A bitline reads as '**0**' only when all the target cells store '**0**'
→ Equivalent to the bitwise OR of all the target cells



Multi-Wordline Sensing (MWS): Bitwise OR

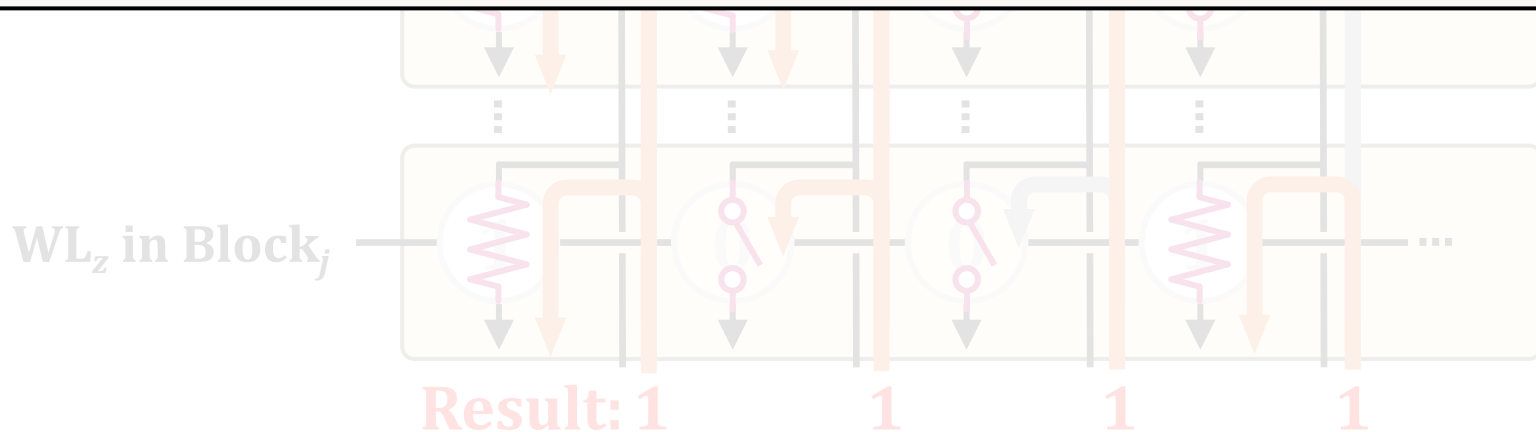
■ Inter-Block MWS:

Simultaneously activates multiple WLs in different blocks

→ Bitwise OR of the stored data in the WLs



Flash-Cosmos (Inter-Block MWS) enables
bitwise OR of multiple pages in different blocks
via a single sensing operation



Other Types of Bitwise Operations

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

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

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

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



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

Key Ideas



Multi-Wordline Sensing (MWS)

to enable in-flash bulk bitwise operations via a single sensing operation



Enhanced SLC-Mode Programming (ESP)

to eliminate raw bit errors in stored data (and thus in computation results)

Enhanced SLC-Mode Programming (ESP)

- **Goal:** eliminate raw bit errors in stored data (and computation results)
- **Key ideas**
 - Programs only a single bit per cell (SLC-mode programming)
 - Trades storage density for reliable computation
 - Performs more precise programming of the cells
 - Trades programming latency for reliable computation

Maximizes the reliability margin
between the different states of flash cells

Enhanced SLC-Mode Programming (ESP)

- To eliminate raw bit errors in stored data (and computation results)

Flash-Cosmos (ESP) enables
reliable in-flash computation
by trading storage density & programming latency

Storage & latency overheads affect
only data used in in-flash computation

Evaluation Methodology

■ Real-device characterization

- To validate the feasibility and reliability of Flash-Cosmos
- Using 160 48-WL-layer 3D Triple-Level Cell NAND flash chips
 - 3,686,400 tested wordlines
- Under worst-case operating conditions
 - Under a 1-year retention time at 10K P/E cycles
 - Worst-case data patterns

■ System-level evaluation

- Using the state-of-the-art SSD simulator (MQSim [Tavakkol+, FAST'18])
- Three real-world applications
 - Bitmap Indices (BMI): Bitwise AND of up to ~1,000 operands
 - Image Segmentation (IMS): Bitwise AND of 3 operands
 - K-clique Star Listing (KCS): Bitwise OR of up to 32 operands
- Baselines
 - Outside-Storage Processing (OSP): A multi-core CPU (Intel i7-11700K)
 - In-Storage Processing (ISP): An in-storage hardware accelerator
 - ParaBit [Gao+, MICRO'21]: State-of-the-art in-flash processing mechanism

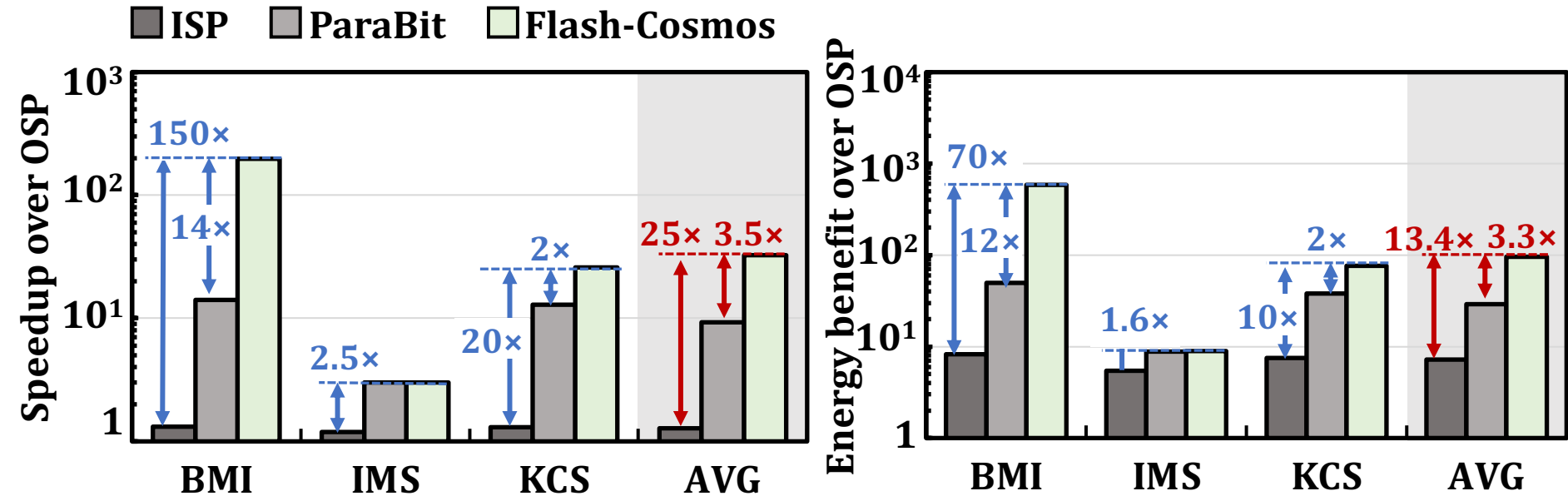
Results: Real-Device Characterization

Both intra- and inter-block MWS operations require no changes to the cell array of commodity NAND flash chips

Both MWS operations can activate multiple WLs (intra: up to 48, inter: up to 4) at the same time with small increase in sensing latency ($< 10\%$)

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

Results: Performance & Energy



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

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

Summary: Flash-Cosmos



The first work that enables
in-flash multi-operand bulk bitwise operations
with a single sensing operation and high reliability



Improves performance
by 32x/25x/3.5x over OSP/ISP/ParaBit



Improves energy efficiency
by 95x/13.4x/3.3x over OSP/ISP/ParaBit



Low-cost & requires no changes to flash cell arrays

In-Flash Bulk Bitwise Execution

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

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

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

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

Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory

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

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

Computing Architectures with Minimal Data Movement

Storage-Centric Computing (I)

Storage system is a heterogeneous computing device with hybrid memory

Storage system enables data-centric design of systems & workloads

Application-driven customization enables a powerful data-centric engine

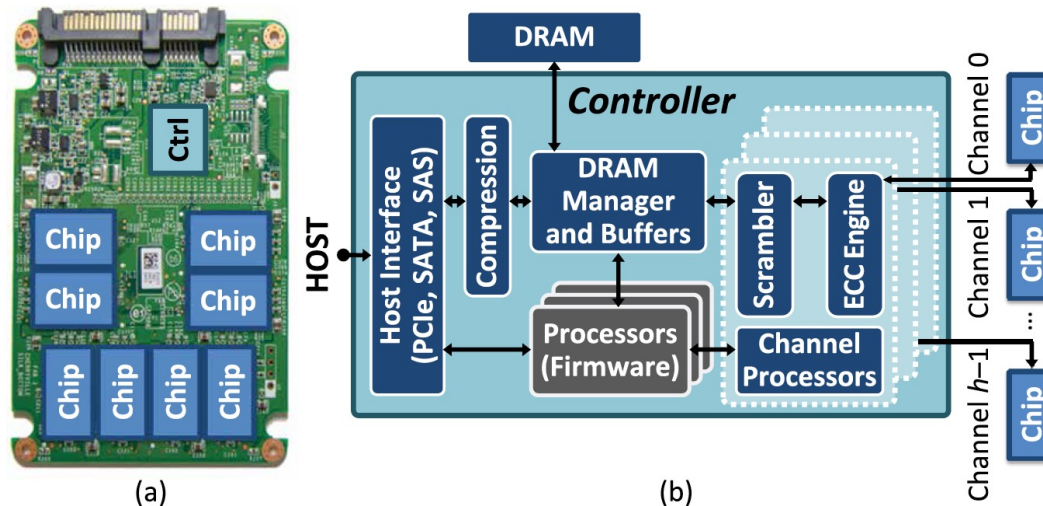
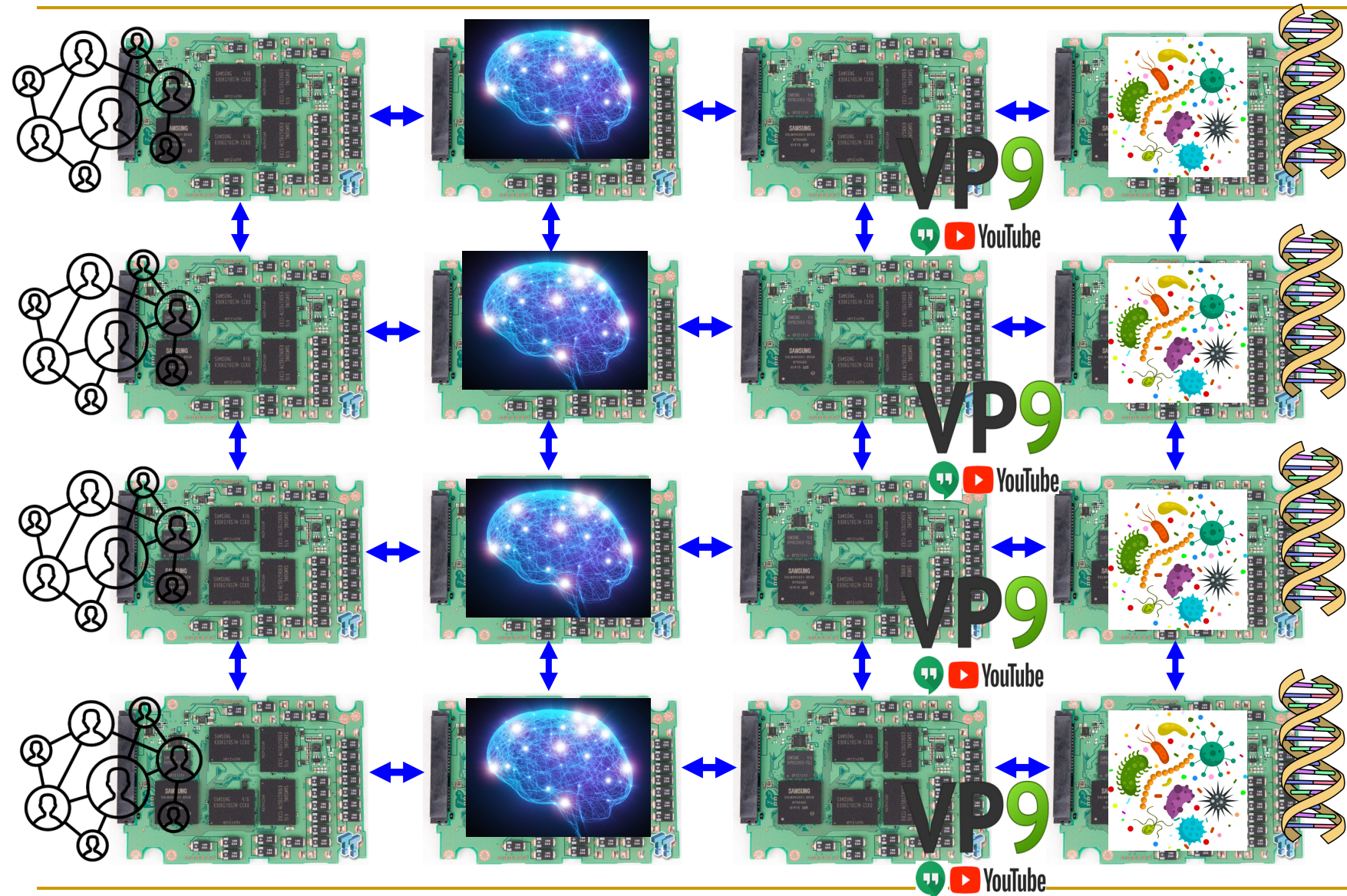


Fig. 1. (a) SSD system architecture, showing controller (Ctrl) and chips. (b) Detailed view of connections between controller components and chips.

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

Storage-Centric Computing (II)



Workload-Customized Storage-Centric Computing

- Software and hardware customized for major workloads
 - ❑ Genomics
 - ❑ Video analytics
 - ❑ Data & graph analytics
 - ❑ Machine learning
 - ❑ ...
- Data-centric (processing capability in all memories)
- Data-driven (design & decision making)
- Data-aware (optimization & design)
- Unified interfaces for efficient & fast communication

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

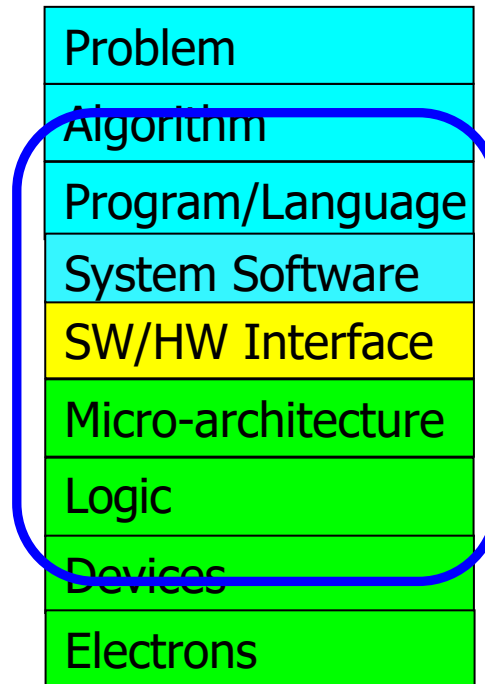
Data-centric

Data-driven

Data-aware



We Need to Revisit the Entire Stack



We can get there step by step

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

Acknowledgments

SAFARI

SAFARI Research Group

safari.ethz.ch

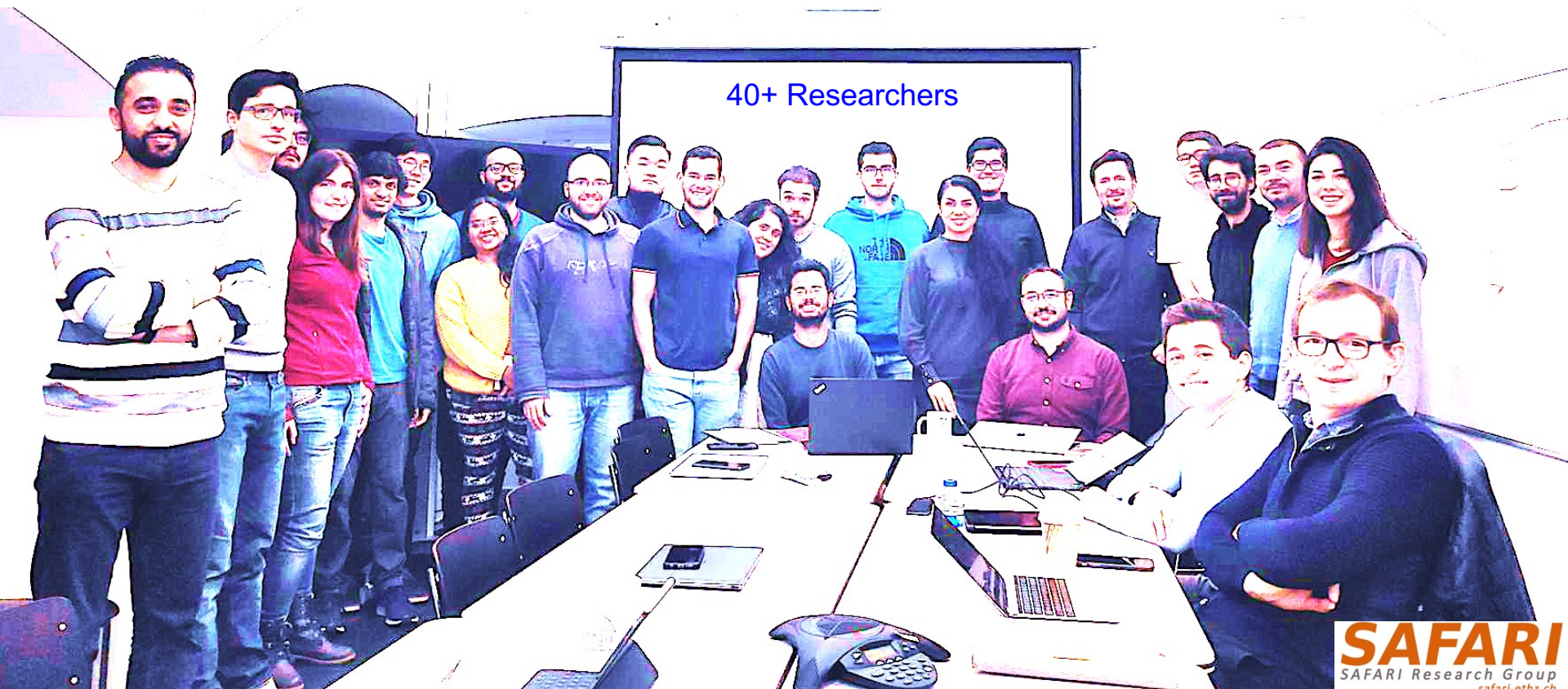
Think BIG, Aim HIGH!

<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/>



SAFARI
SAFARI Research Group
safari.ethz.ch

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**
SAFARI Research Group

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

[🏠 Overview](#) [📁 Repositories 77](#) [📁 Projects](#) [📁 Packages](#) [👤 People 13](#)

Pinned

📁 **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++ ☆ 405 🍴 184

📁 **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 ☆ 79 🍴 33

📁 **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++ ☆ 181 🍴 108

📁 **rowhammer** Public

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

● C ☆ 199 🍴 40

📁 **SparseP** Public

SparseP is the first open-source Sparse Matrix Vector Multiplication (SpMV) software package for real-world Processing-In-Memory (PIM) architectures. SparseP is developed to evaluate and characteri...

● C ☆ 50 🍴 11

📁 **SoftMC** Public

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

● Verilog ☆ 94 🍴 26

Reinventing Computing & Storage

Onur Mutlu

omutlu@gmail.com

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

7 April 2023

OlympusMons Award Keynote Speech

SAFARI

ETH zürich

Carnegie Mellon

Backup Slides

Executive Summary

- **Background:** A hybrid storage system (HSS) uses multiple different storage devices to provide high and scalable storage capacity at high performance
- **Problem:** Two key shortcomings of prior data placement policies:
 - Lack of **adaptivity to:**
 - **Workload changes**
 - **Changes in device types and configurations**
 - Lack of **extensibility** to more devices
- **Goal:** Design a data placement technique that provides:
 - **Adaptivity**, by **continuously learning and adapting** to the **application and underlying device characteristics**
 - **Easy extensibility** to incorporate a wide range of hybrid storage configurations
- **Contribution:** Sibyl, the first reinforcement learning-based data placement technique in hybrid storage systems that:
 - Provides **adaptivity** to changing workload demands and underlying device characteristics
 - Can **easily extend** to any number of storage devices
 - Provides **ease of design and implementation** that requires only a small computation overhead
- **Key Results:** Evaluate on **real systems** using a wide range of workloads
 - Sibyl **improves performance by 21.6%** compared to the best previous data placement technique in dual-HSS configuration
 - In a tri-HSS configuration, Sibyl outperforms the state-of-the-art-policy policy by **48.2%**
 - Sibyl achieves **80% of the performance** of an oracle policy with storage overhead of only **124.4 KiB**

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 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

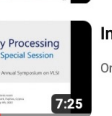
61 DISLIKE SHARE DOWNLOAD CLIP SAVE

Onur Mutlu Lectures
26.9K subscribers

ANALYTICS EDIT VIDEO

- Special Session at ISVLSI 2022: 9 cutting-edge talks

19



GenStore: In-Storage Filtering for High-Performance and Energy-Efficient Genome Analysis
Onur Mutlu Lectures • • Premieres 3/12/23, 7:00 PM

20



Introduction to the ISVLSI 2022 Special Session on Processing-in-Memory
Onur Mutlu Lectures • 286 views • 2 days ago

21



Heterogeneous Data-Centric Architectures for Data-Intensive Applications: Case Studies in ML and DB
Onur Mutlu Lectures • 2 waiting • Premieres 3/10/23, 7:00 PM

22



Machine Learning Training on a Real Processing-In-Memory System
Onur Mutlu Lectures • • Premieres 3/14/23, 7:00 PM

23



Exploiting Near-Data Processing to Accelerate Time Series Analysis
Onur Mutlu Lectures • • Premieres 3/11/23, 7:00 PM

24



PiDRAM: An FPGA-Based Framework for End-To-End Evaluation of Processing-In-DRAM Techniques
Onur Mutlu Lectures • • Premieres 3/9/23, 7:00 PM

25



The Road to Widely Deploying Processing-In-Memory: Challenges and Opportunities
Onur Mutlu Lectures • 399 views • 1 day ago

26



SparseP: Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures
Onur Mutlu Lectures • 1 waiting • Premieres 3/13/23, 7:00 PM

27



HPCA 2023 Tutorial: Real-World Processing-in-Memory Architectures
Onur Mutlu Lectures • 1.6K views • Streamed 10 days ago

Comp Arch (Fall 2021)

- 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

Youtube Livestream (2020):

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

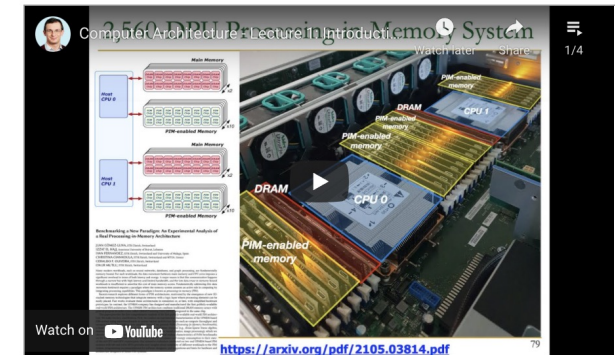
Master's level course

- 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

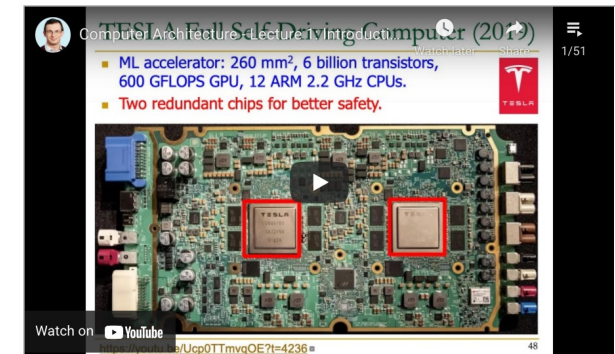
<https://www.youtube.com/onurmutlulectures>

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 PDF PPT	Required Mentioned	Lab 1 Out	HW 0 Out
	01.10 Fri.	YouTube Live	L2: Trends, Tradeoffs and Design Fundamentals PDF PPT	Required Mentioned		
W2	07.10 Thu.	YouTube Live	L3a: Memory Systems: Challenges and Opportunities PDF PPT	Described Suggested		HW 1 Out
			L3b: Course Info & Logistics PDF PPT			
			L3c: Memory Performance Attacks PDF PPT	Described Suggested		
	08.10 Fri.	YouTube Live	L4a: Memory Performance Attacks PDF PPT	Described Suggested	Lab 2 Out	
			L4b: Data Retention and Memory Refresh PDF PPT	Described Suggested		
			L4c: RowHammer PDF PPT	Described Suggested		

DDCA (Spring 2022)

Spring 2022 Edition:

- <https://safari.ethz.ch/digitaltechnik/spring2022/duku.php?id=schedule>

Spring 2021 Edition:

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

Youtube Livestream (Spring 2022):

- <https://www.youtube.com/watch?v=cpXdE3HwvK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6>

Youtube Livestream (Spring 2021):

- https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN

Bachelor's course

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

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



Trace: - schedule

Home

Announcements

Materials

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

Resources

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

Lecture Video Playlist on YouTube

Livestream Lecture Playlist

Onur Mutlu: Digital Design and Computer Architecture Today

- Computing landscape is very different from 10-20 years ago
- Applications and technology both demand novel architectures

Watch on YouTube

Recorded Lecture Playlist

Digital Design & Computer Architecture: Lect...

How Computers Work

(from the ground up)

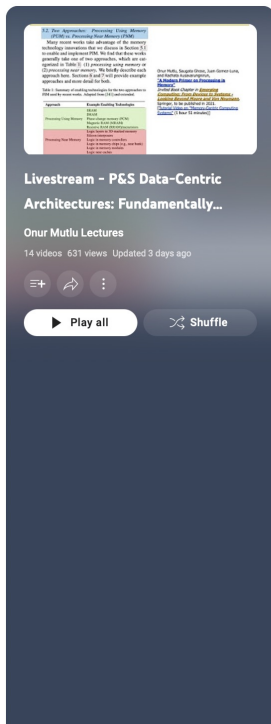
Watch on YouTube

Spring 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	25.02 Thu.	YouTube Live	L1: Introduction and Basics G23a (PDF) G23b (PPT)	Required Suggested Mentioned		
	26.02 Fri.	YouTube Live	L2a: Tradeoffs, Metrics, Mindset G23a (PDF) G23b (PPT)	Required		
			L2b: Mysteries in Computer Architecture G23a (PDF) G23b (PPT)	Required Mentioned		
W2	04.03 Thu.	YouTube Live	L3a: Mysteries in Computer Architecture II G23a (PDF) G23b (PPT)	Required Suggested Mentioned		

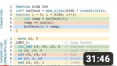
Processing-in-Memory Course (Fall 2022)

- Short weekly lectures
- Hands-on projects



1  **PIM Course: Lecture 1: Data-Centric Architectures: Improving Performance & Energy - Fall 2022**
Onur Mutlu Lectures • 1K views • 3 months ago

2  **PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks - Fall 2022**
Onur Mutlu Lectures • 678 views • 2 months ago

3  **PIM Course: Lecture 3: Real-world PIM: UPMEM PIM - Fall 2022**
Onur Mutlu Lectures • 455 views • 2 months ago

4  **PIM Course: Lecture 4: Real-world PIM: Microbenchmarking of UPMEM PIM - Fall 2022**
Onur Mutlu Lectures • 275 views • 2 months ago

5  **PIM Course: Lecture 5: Real-world PIM: Samsung HBM-PIM - Fall 2022**
Onur Mutlu Lectures • 725 views • 2 months ago

6  **PIM Course: Lecture 6: Real-world PIM: SK Hynix AiM - Fall 2022**
Onur Mutlu Lectures • 1K views • 2 months ago

7  **PIM Course: Lecture 7: Real-world PIM: Samsung AxDIMM - Fall 2022**
Onur Mutlu Lectures • 767 views • 1 month ago

8  **PIM Course: Lecture 8: Real-world PIM: Alibaba HB-PNM - Fall 2022**
Onur Mutlu Lectures • 383 views • 1 month ago

9  **PIM Course: Lecture 9: Programming PIM Architectures - Fall 2022**
Onur Mutlu Lectures • 367 views • 1 month ago

 **SAFARI Project & Seminars Courses (Fall 2022)**

Trace: • heterogeneous_systems • processing_in_memory

Home

Courses

- SoftMC
- Ramulator
- Accelerating Genomics
- Mobile Genomics
- **Processing-in-Memory**
- Heterogeneous Systems
- Modern SSDs

processing_in_memory

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

Course Description

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

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

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

Table of Contents

- Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)
- Course Description
- Mentors
- Lecture Video Playlist on YouTube
- Spring 2022 Meetings/Schedule
- Past Lecture Video Playlists on YouTube
- Learning Materials
- Assignments

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

<https://youtube.com/playlist?list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0GOBrnKy>

PIM Course (Fall 2022)

■ Fall 2022 Edition:

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

■ Spring 2022 Edition:

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

■ Youtube Livestream (Fall 2022):

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

■ Youtube Livestream (Spring 2022):

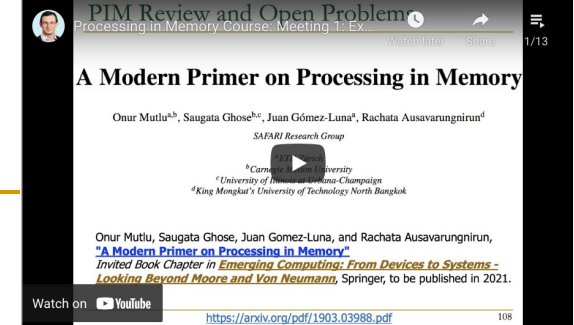
- <https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX>

■ Project course

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

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

SAFARI

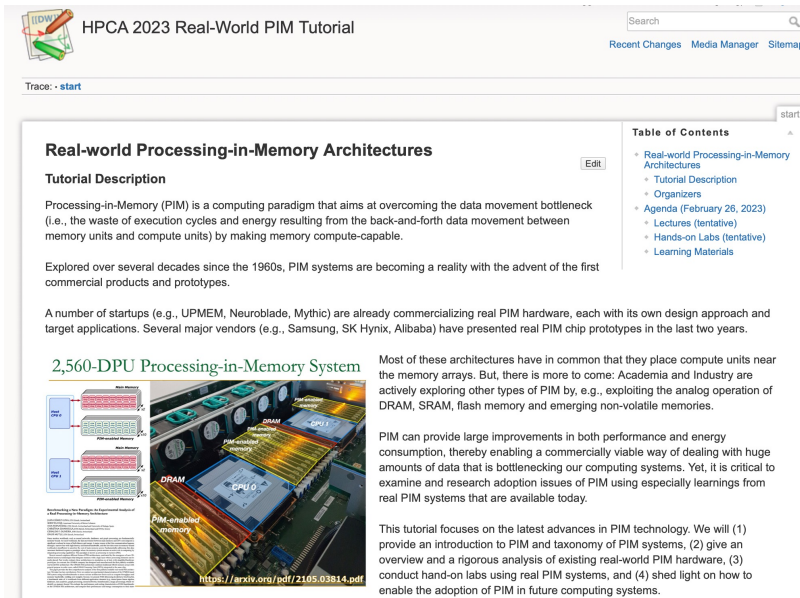


Spring 2022 Meetings/Schedule

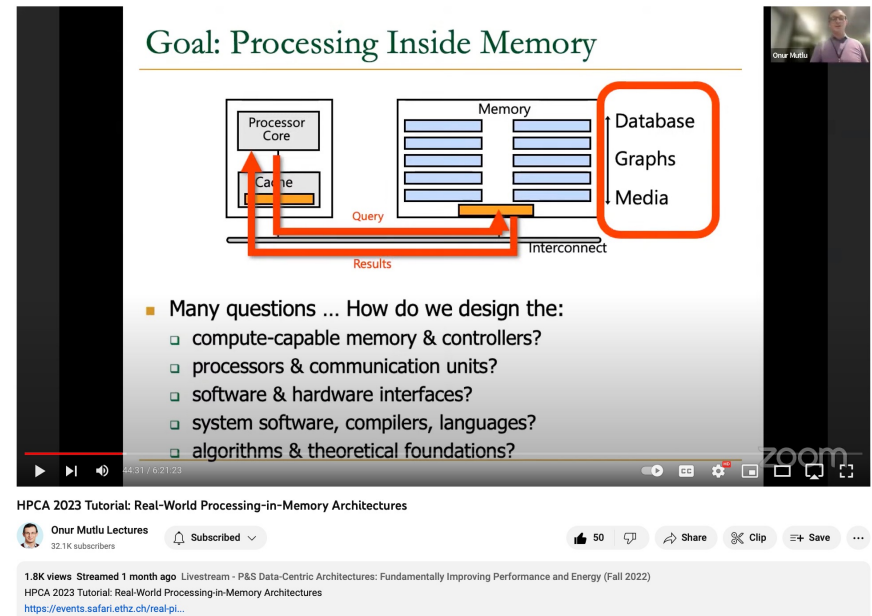
Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	10.03 Thu.	YouTube Live	M1: P&S PIM Course Presentation (PDF) (PPT)	Required Materials Recommended Materials	HW 0 Out
W2	15.03 Tue.		Hands-on Project Proposals		
	17.03 Thu.	YouTube Premiere	M2: Real-world PIM: UPMEM PIM (PDF) (PPT)		
W3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

Real PIM Tutorial (HPCA 2023)

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



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



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

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

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

Tutorial Materials

The image shows a presentation slide with a white background and a black border on the left and right sides. At the top, in a small black box, it says "ASPLOS 2023 Tutorial". Below this, the text "Real-world Processing-in-Memory Systems for Modern Workloads" is written in a black serif font. The main title of the slide is "Accelerating Modern Workloads on a General-purpose PIM System", with "Accelerating Modern Workloads" in a large blue sans-serif font and "on a General-purpose PIM System" in a smaller black sans-serif font. Below the title, the names "Dr. Juan Gómez Luna" and "Professor Onur Mutlu" are listed in a black sans-serif font. At the bottom of the slide, there are two logos: "ETH Zürich" on the left and "SAFARI!" in large red letters on the right. A small "zoom" logo is visible in the bottom right corner. The slide is displayed within a video player interface, with a red progress bar at the bottom showing the video is at 6:25:34 / 6:27:38. There are also standard video player controls like play, pause, and volume icons.

Onur Mutlu Lectures
32.1K subscribers

views Streamed 7 days ago Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)
LOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads
<https://events.safari.ethz.ch/asplon-...>

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

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

Upcoming Real PIM Tutorial (ISCA 2023)

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



ISCA 2023 Real-World PIM Tutorial

Search

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

Trace: • [start](#)

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

Tutorial Description

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

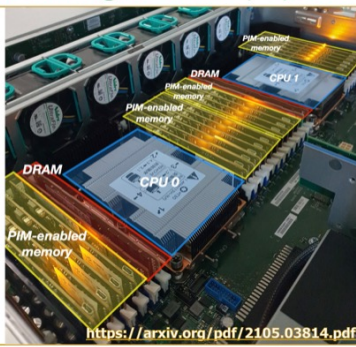
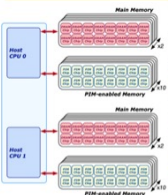
Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

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

Table of Contents

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

2,560-DPU Processing-in-Memory System



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

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

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

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

SSD Course (Spring 2023)

Spring 2023 Edition:

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

Fall 2022 Edition:

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

Youtube Livestream (Spring 2023):

- https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB

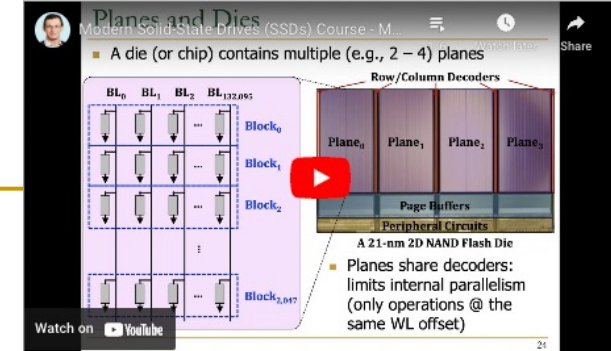
Youtube Livestream (Fall 2022):

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

Project course

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

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



Fall 2022 Meetings/Schedule

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

Genomics Course (Fall 2022)

Fall 2022 Edition:

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

Spring 2022 Edition:

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

Youtube Livestream (Fall 2022):

- https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFIQvdxOdizD_EhVAMVQV

Youtube Livestream (Spring 2022):

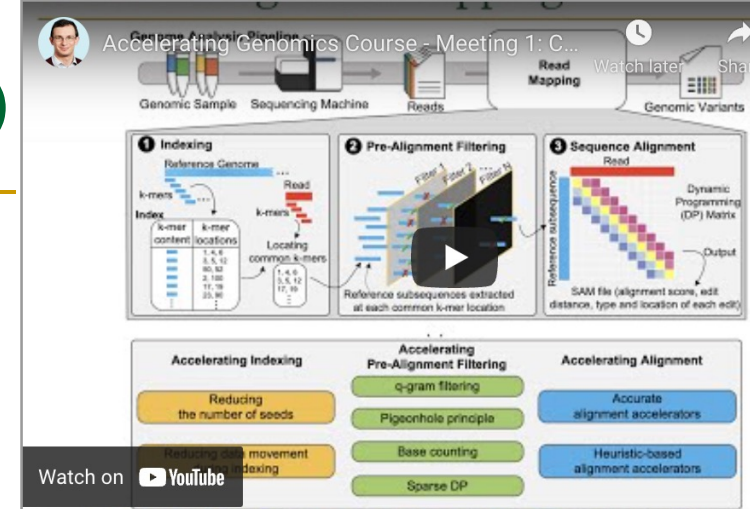
- https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18

Project course

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

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

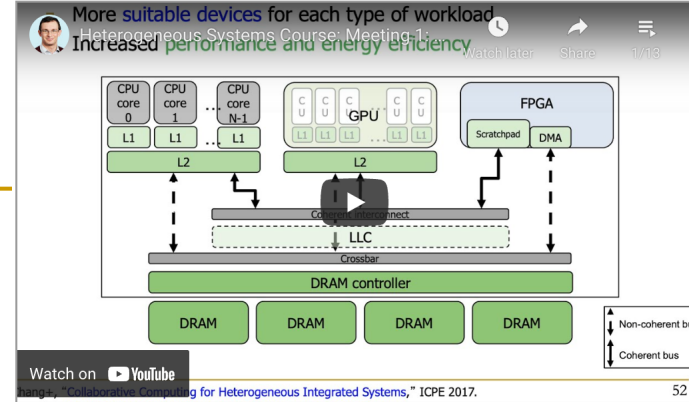
SAFARI



Spring 2022 Meetings/Schedule

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

Hetero. Systems (Spring'22)



Spring 2022 Edition:

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

Youtube Livestream:

- https://www.youtube.com/watch?v=oFO5fTrgFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm

Project course

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

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

Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	15.03 Tue.	YouTube Premiere	M1: P&S Course Presentation PDF PPT	Required Materials Recommended Materials	HW 0 Out
W2	22.03 Tue.	YouTube Premiere	M2: SIMD Processing and GPUs PDF PPT		
W3	29.03 Tue.	YouTube Premiere	M3: GPU Software Hierarchy PDF PPT		
W4	05.04 Tue.	YouTube Premiere	M4: GPU Memory Hierarchy PDF PPT		
W5	12.04 Tue.	YouTube Premiere	M5: GPU Performance Considerations PDF PPT		
W6	19.04 Tue.	YouTube Premiere	M6: Parallel Patterns: Reduction PDF PPT		
W7	26.04 Tue.	YouTube Premiere	M7: Parallel Patterns: Histogram PDF PPT		
W8	03.05 Tue.	YouTube Premiere	M8: Parallel Patterns: Convolution PDF PPT		
W9	10.05 Tue.	YouTube Premiere	M9: Parallel Patterns: Prefix Sum (Scan) PDF PPT		
W10	17.05 Tue.	YouTube Premiere	M10: Parallel Patterns: Sparse Matrices PDF PPT		
W11	24.05 Tue.	YouTube Premiere	M11: Parallel Patterns: Graph Search PDF PPT		
W12	01.06 Wed.	YouTube Premiere	M12: Parallel Patterns: Merge Sort PDF PPT		
W13	07.06 Tue.	YouTube Premiere	M13: Dynamic Parallelism PDF PPT		
W14	15.06 Wed.	YouTube Premiere	M14: Collaborative Computing PDF PPT		
W15	24.06 Fri.	YouTube Premiere	M15: GPU Acceleration of Genome Sequence Alignment PDF PPT		
W16	14.07 Thu.	YouTube Premiere	M16: Accelerating Agent-based Simulations PDF ODP		

HW/SW Co-Design (Spring 2022)

Spring 2022 Edition:

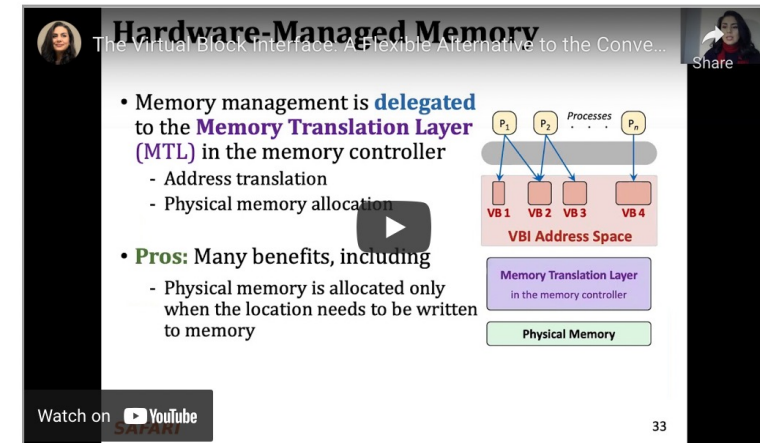
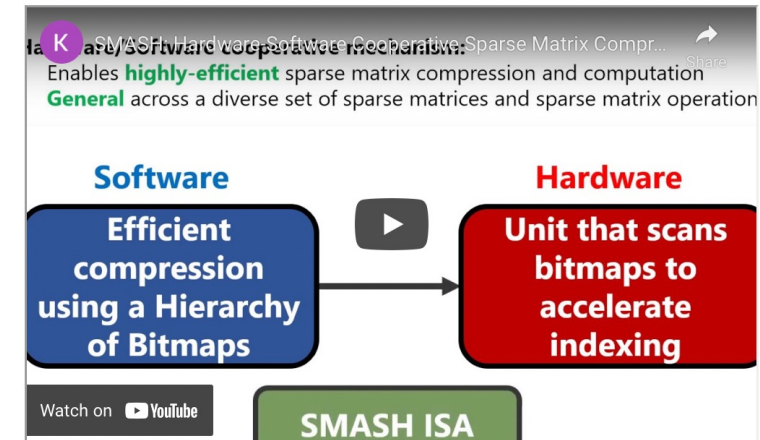
- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_co_design

Youtube Livestream:

- https://youtube.com/playlist?list=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 (PPTX) (PDF)	Required	HW 0 Out
W1	23.03		Project selection	Required	
W2	30.03	YouTube Live	Virtual Memory (I) (PPTX) (PDF)		
W3	13.04	YouTube Live	Virtual Memory (II) (PPTX) (PDF)		

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

RowHammer & DRAM Exploration (Fall 2022)

Fall 2022 Edition:

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

Spring 2022 Edition:

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

Youtube Livestream (Spring 2022):

- ❑ https://www.youtube.com/watch?v=r5QxuoJWttg&list=PL5Q2soXY2Zi_1trfCckr6PTN8WR72icUO

Bachelor's course

- ❑ Elective at ETH Zurich
- ❑ Introduction to DRAM organization & operation
- ❑ Tutorial on using FPGA-based infrastructure
- ❑ Verilog & C++
- ❑ Potential research exploration

<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.		P&S SoftMC Tutorial	SoftMC Tutorial Slides (PDF) (PPT)	
W1	08.03 Tue.		M1: Logistics & Intro to DRAM and SoftMC (PDF) (PPT)	Required Materials Recommended Materials	HW0
W2	15.03 Tue.		M2: Revisiting RowHammer (PDF) (PPT)	(Paper PDF)	
W3	22.03 Tue.		M3: Uncovering in-DRAM TRR & TRRespass (PDF) (PPT)		
W4	29.03 Tue.		M4: Deeper Look Into RowHammer's Sensitivities (PDF) (PPT)		
W5	05.04 Tue.		M5: QUAC-TRNG (PDF) (PPT)		
W6	12.04 Tue.		M6: PiDRAM (PDF) (PPT)		

Exploration of Emerging Memory Systems (Fall 2022)

Fall 2022 Edition:

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

Spring 2022 Edition:

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

Youtube Livestream (Spring 2022):

- ❑ https://www.youtube.com/watch?v=aM-lIXRQd3s&list=PL5Q2soXY2Zi_TlmlGw_Z8hBo2925ZAqV

Bachelor's course

- ❑ Elective at ETH Zurich
- ❑ Introduction to memory system simulation
- ❑ Tutorial on using Ramulator
- ❑ C++
- ❑ Potential research exploration

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

Lecture Video Playlist on YouTube

Lecture Playlist



2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	09.03 Wed.	YouTube Video	M1: Logistics & Intro to Simulating Memory Systems Using Ramulator PDF (PDF) PPT (PPT)		HW0
W2	16.03 Fri.	YouTube Video	M2: Tutorial on Using Ramulator PDF (PDF) PPT (PPT)		
W3	25.02 Fri.	YouTube Video	M3: BlockHammer PDF (PDF) PPT (PPT)		
W4	01.04 Fri.	YouTube Video	M4: CLR-DRAM PDF (PDF) PPT (PPT)		
W5	08.04 Fri.	YouTube Video	M5: SIMDRAM PDF (PDF) PPT (PPT)		
W6	29.04 Fri.	YouTube Video	M6: DAMOV PDF (PDF) PPT (PPT)		
W7	06.05 Fri.	YouTube Video	M7: Synchron PDF (PDF) PPT (PPT)		