

1st Workshop on Memory-Centric Computing: Research Challenges & Closing Remarks

Geraldo F. Oliveira

<https://geraldofojunior.github.io>

ASPLOS 2025

30 March 2025

Eliminating the Adoption Barriers

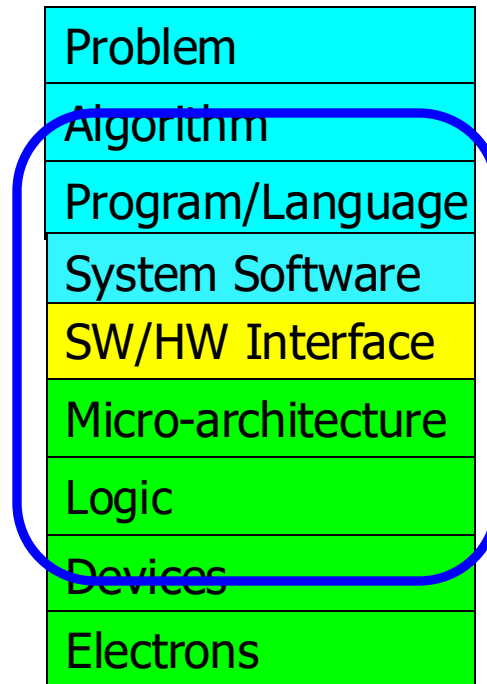
How to Enable Adoption of Processing in Memory

Potential Barriers to Adoption of PIM

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

All can be solved with change of mindset

We Need to Revisit the Entire Stack



We can get there step by step

Adoption: How to Keep It Simple?

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

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

Junwhan Ahn Sungjoo Yoo Onur Mutlu[†] Kiyoungh Choi

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

Seoul National University

[†]Carnegie Mellon University

Adoption: How to Ease Programmability? (I)

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

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

Geraldo F. Oliveira*

Alain Kohli*

David Novo[‡]

Juan Gómez-Luna*

Onur Mutlu*

**ETH Zürich*

[‡]*LIRMM, Univ. Montpellier, CNRS*

Adoption: How to Ease Programmability? (II)

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

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

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

Adoption: How to Maintain Coherence? (I)

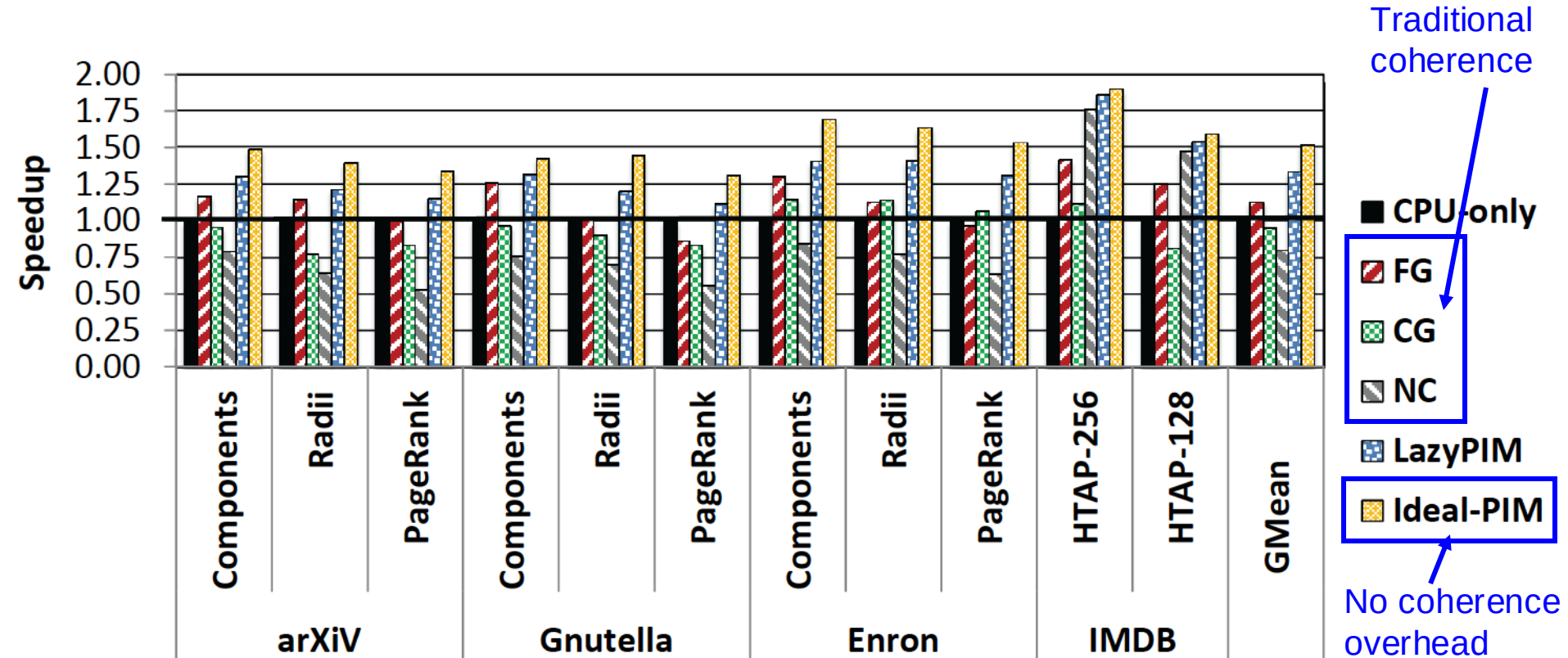
- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,
"**LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory**"
IEEE Computer Architecture Letters (**CAL**), June 2016.

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

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

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

Challenge: Coherence for Hybrid CPU-PIM Apps



Adoption: How to Maintain Coherence? (II)

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

CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators

Amirali Boroumand[†]

Saugata Ghose[†]

Minesh Patel[★]

Hasan Hassan[★]

Brandon Lucia[†]

Rachata Ausavarungnirun^{†‡}

Kevin Hsieh[†]

Nastaran Hajinazar^{◊†}

Krishna T. Malladi[§]

Hongzhong Zheng[§]

Onur Mutlu^{★†}

[†]Carnegie Mellon University

[★]ETH Zürich

[‡]KMUTNB

[◊]Simon Fraser University

[§]Samsung Semiconductor, Inc.

Adoption: How to Support Synchronization?

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

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

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

Adoption: How to Support Virtual Memory?

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

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

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

Adoption: Code and Data Mapping

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

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

[Slides (pptx) (pdf)]

[Lightning Session Slides (pptx) (pdf)]

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

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

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

DAMOV Analysis Methodology & Workloads

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

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

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

NANDITA VIJAYKUMAR, University of Toronto, Canada

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

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

ONUR MUTLU, ETH Zürich, Switzerland

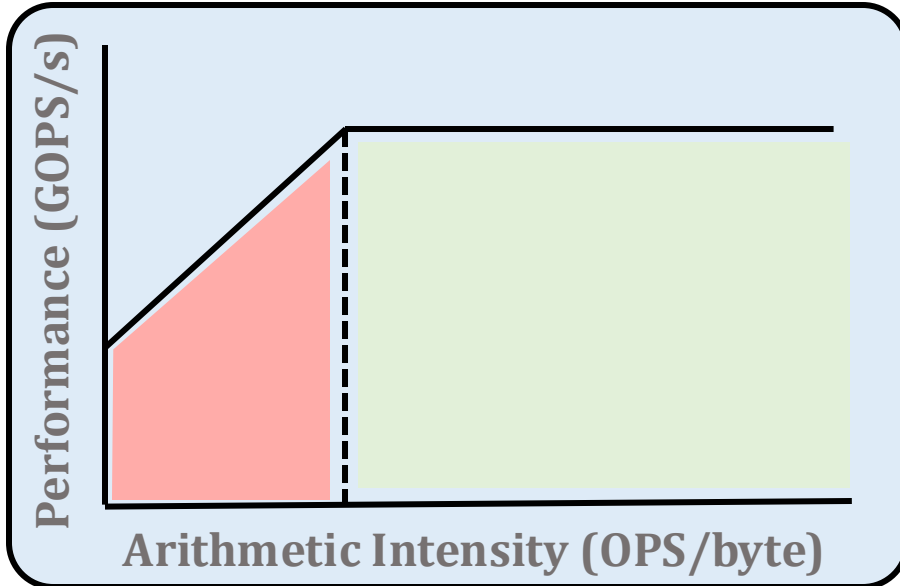
Data movement between the CPU and main memory is a first-order obstacle against improving performance, scalability, and energy efficiency in modern systems. Computer systems employ a range of techniques to reduce overheads tied to data movement, spanning from traditional mechanisms (e.g., deep multi-level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

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

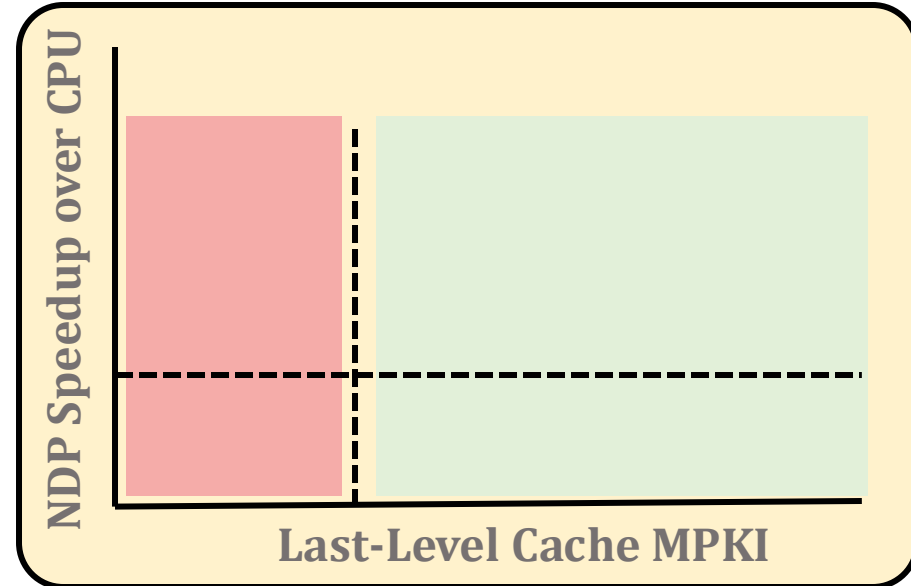
Identifying Memory Bottlenecks

- Multiple approaches to identify applications that:
 - suffer from data movement bottlenecks
 - take advantage of NDP
- Existing approaches are not comprehensive enough

Roofline model

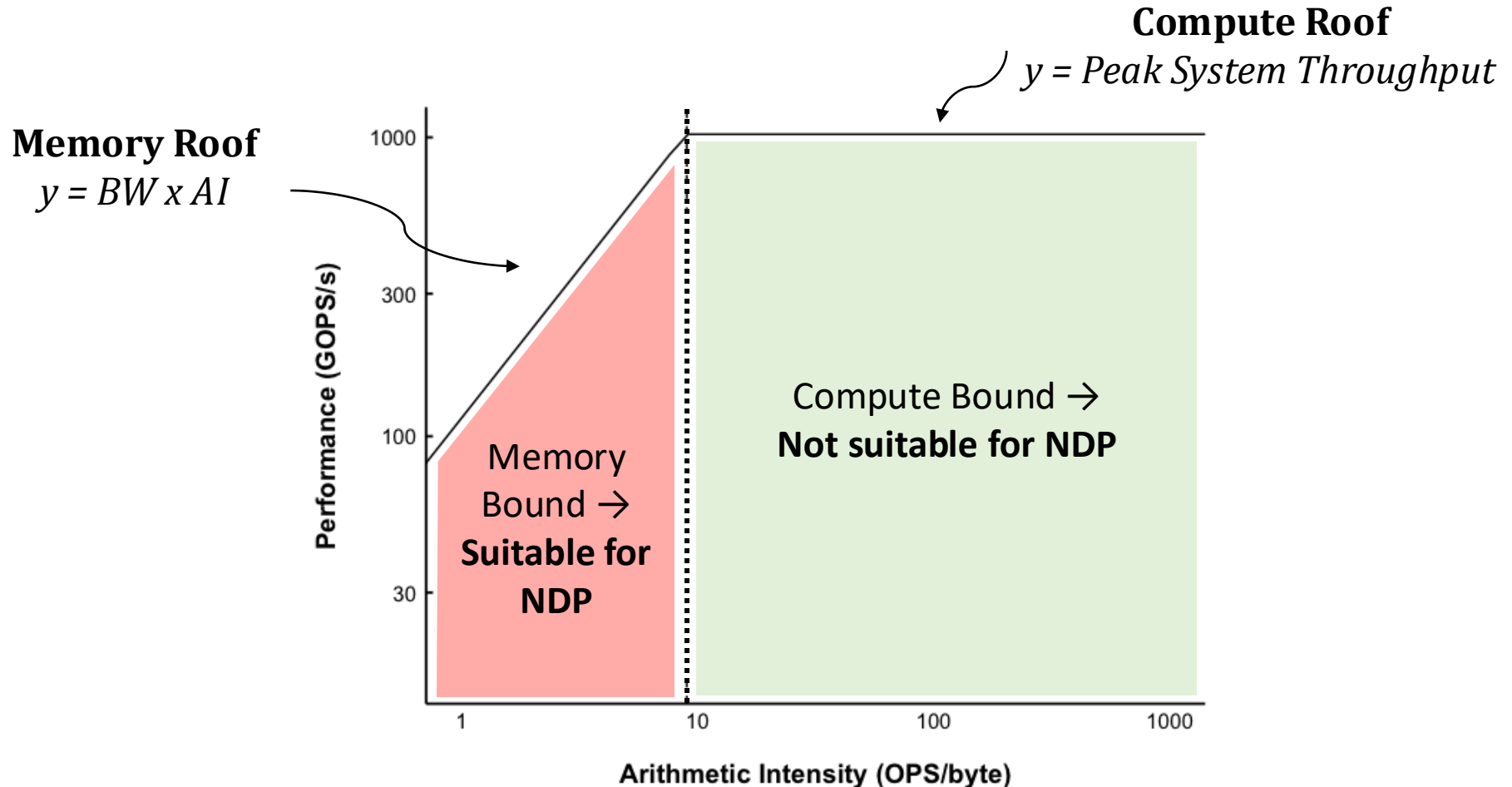


High LLC MPKI



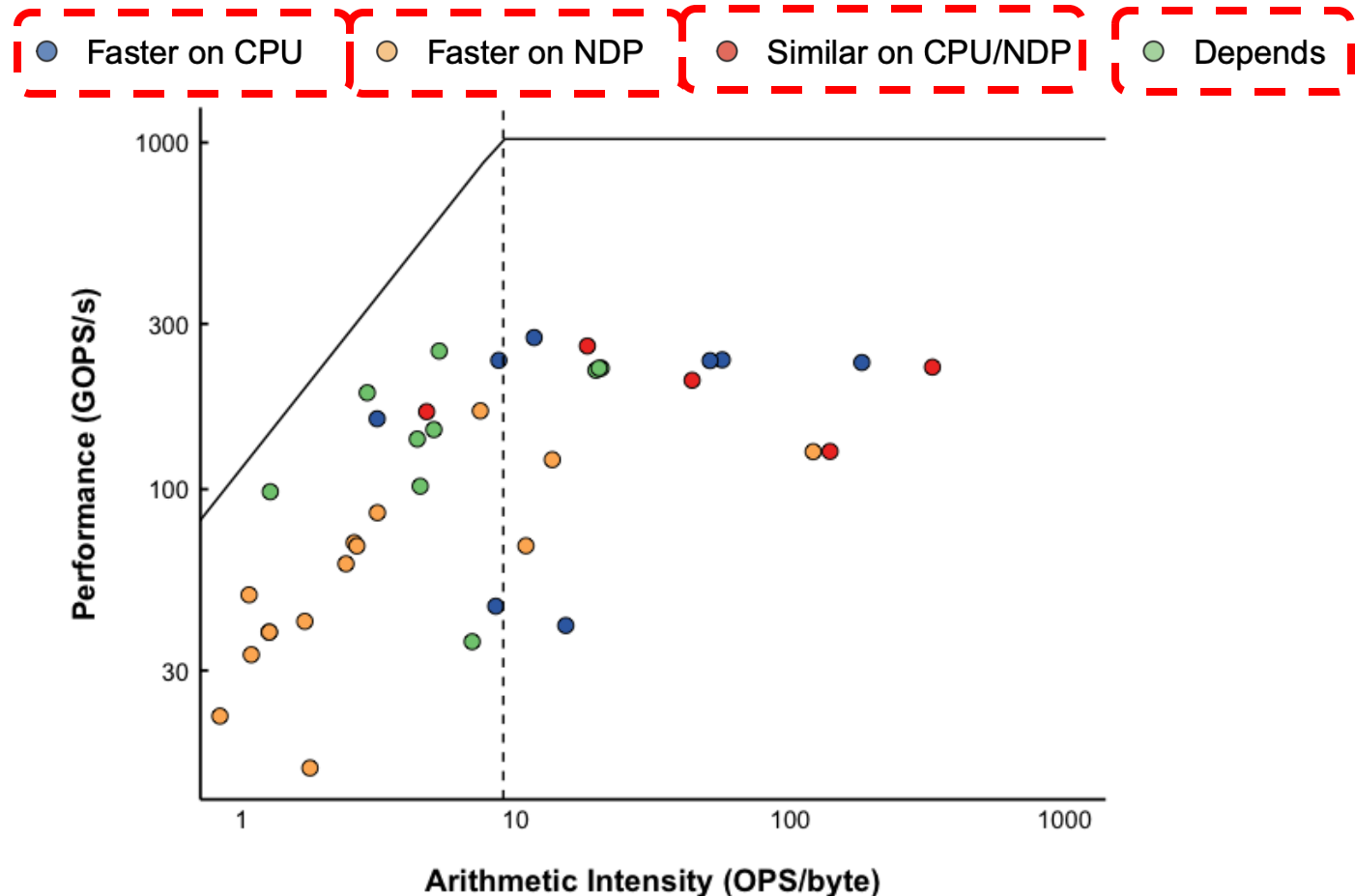
Limitations of Prior Approaches (1/2)

- **Roofline model** → identifies when an application is *bounded* by **compute** or **memory** units



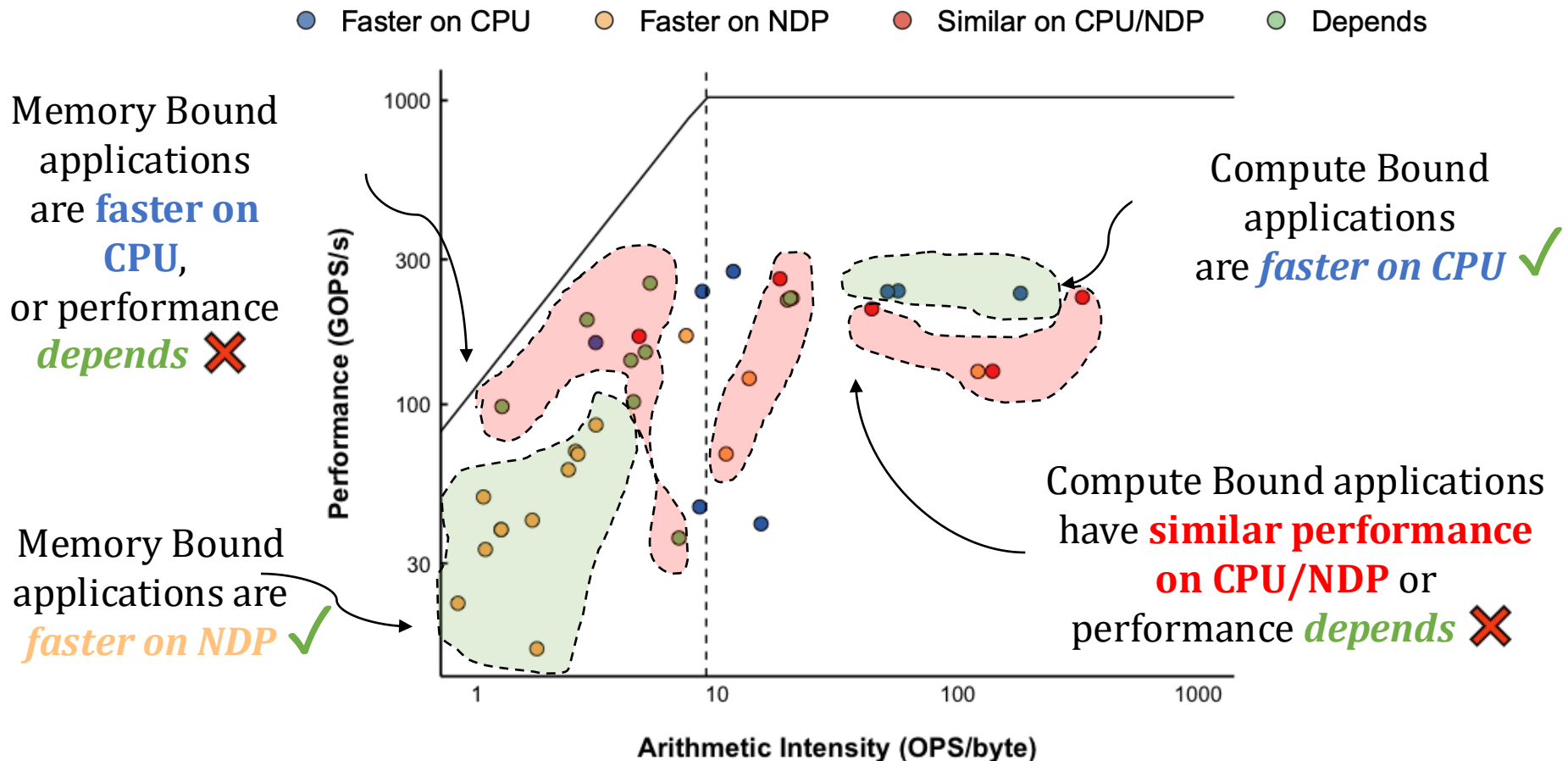
Limitations of Prior Approaches (1/2)

- **Roofline model** → identifies when an application is *bounded* by **compute** or **memory** units



Limitations of Prior Approaches (1/2)

- **Roofline model** → identifies when an application is *bounded* by **compute** or **memory** units

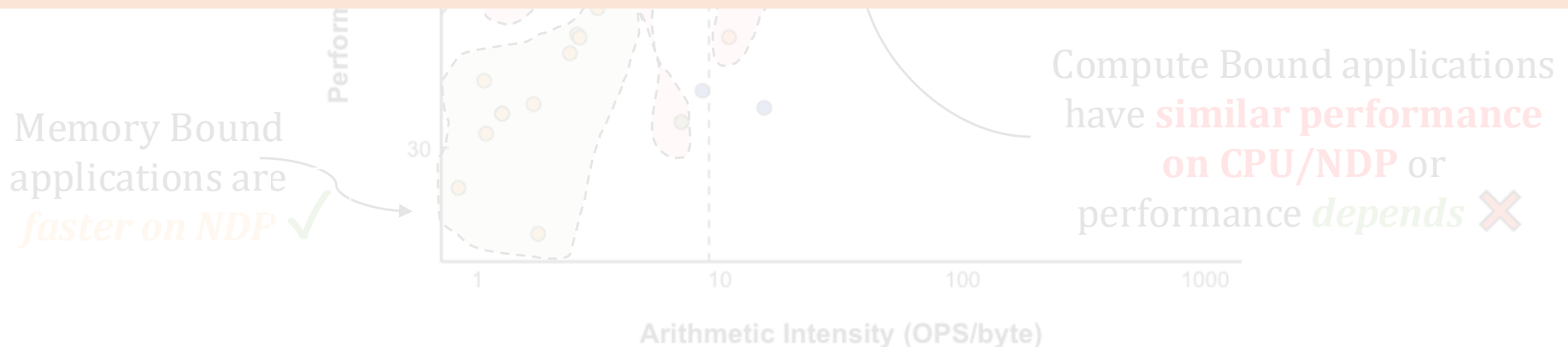


Limitations of Prior Approaches (1/2)

- **Roofline model** → identifies when an application is *bounded* by **compute** or **memory** units

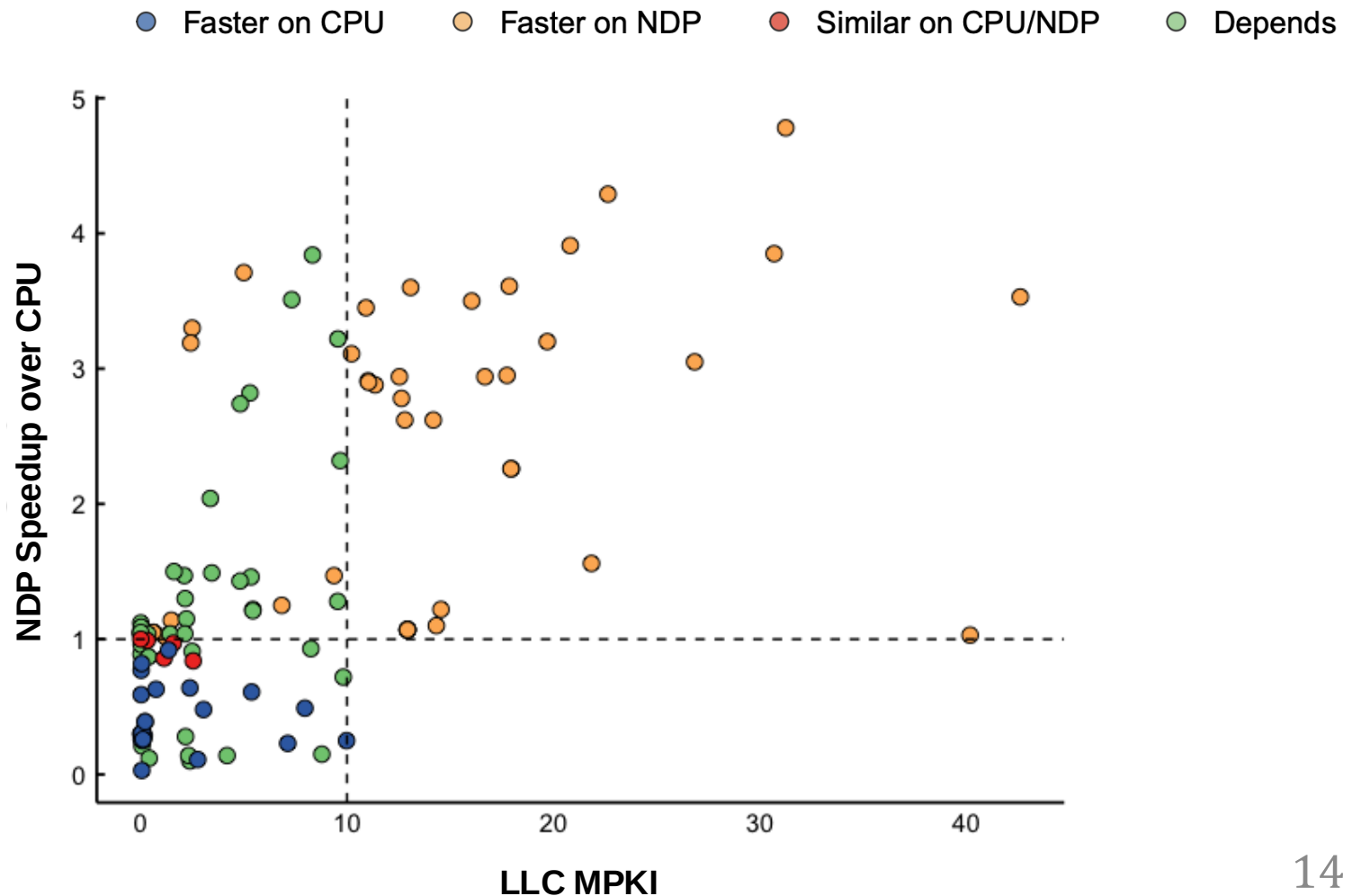
● Faster on CPU ● Faster on NDP ● Similar on CPU/NDP ● Depends

Roofline model **does not accurately account** for the **NDP suitability** of memory-bound applications



Limitations of Prior Approaches (2/2)

- Application with a last-level cache **MPKI > 10**
→ **memory intensive** and **benefits from NDP**



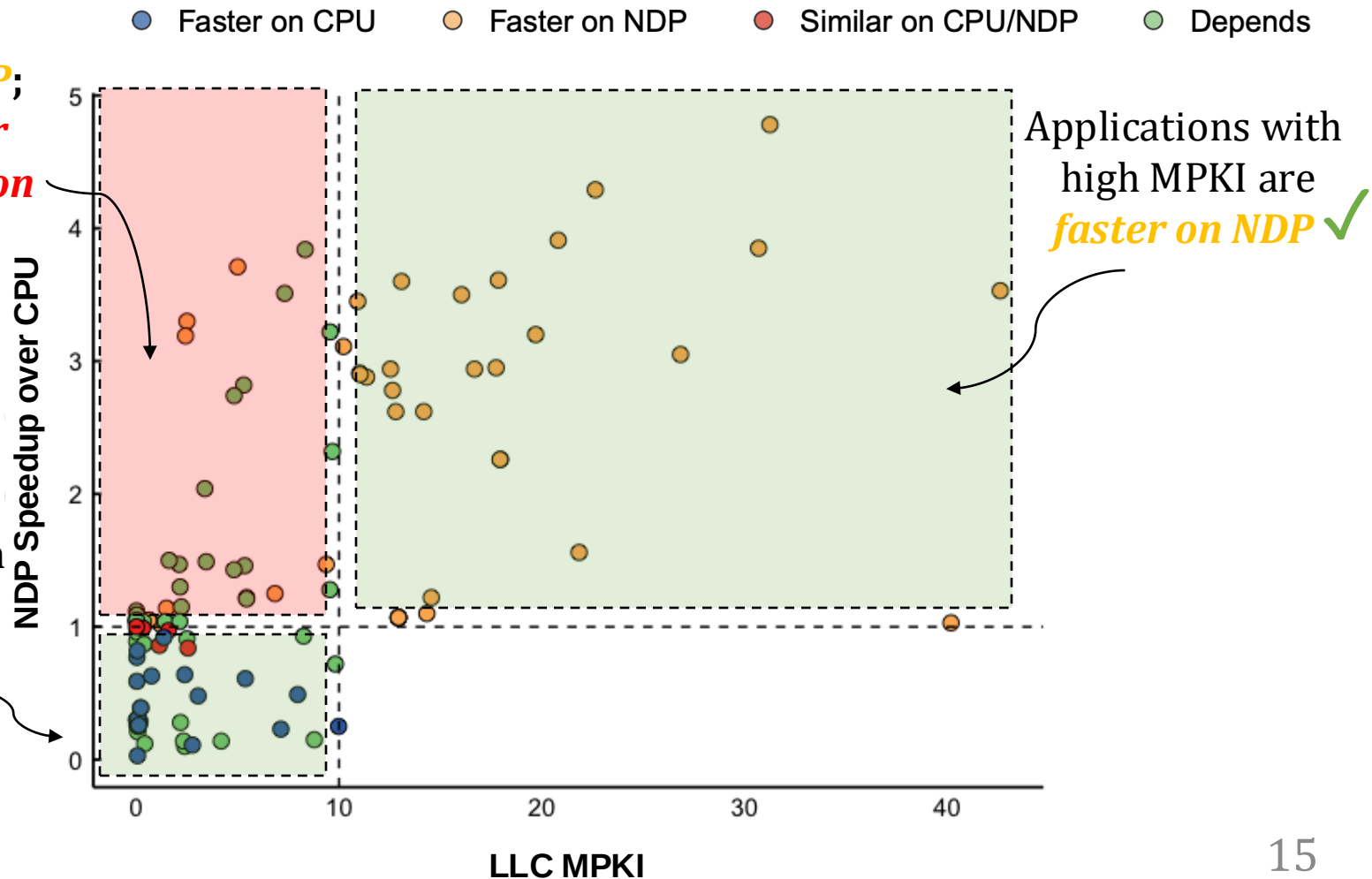
Limitations of Prior Approaches (2/2)

- Application with a last-level cache **MPKI > 10**
→ **memory intensive** and **benefits from NDP**

Applications with low
MPKI can be

faster on NDP;
have *similar*
performance on
CPU/NDP or;
performance
can *depends*
✗

Applications with
low MPKI are
faster on CPU
✓



Limitations of Prior Approaches (2/2)

- Application with a last-level cache $\text{MPKI} > 10$
→ **memory intensive** and **benefits from NDP**

Applications with low
MPKI can be
faster on NDP;

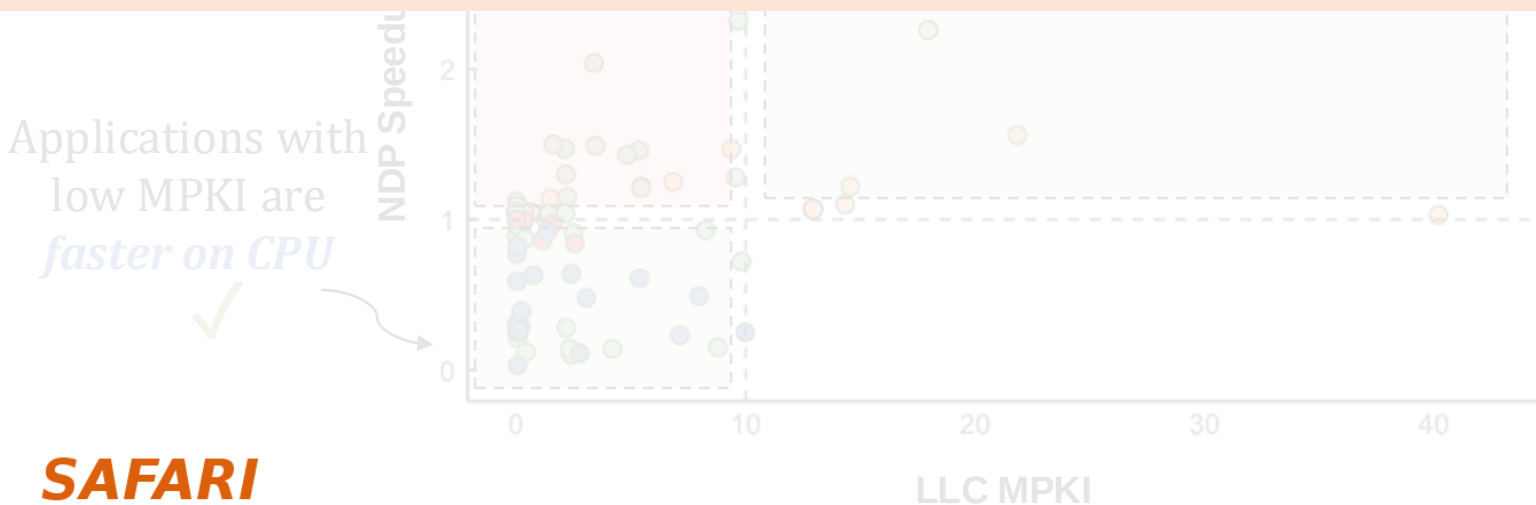
● Faster on CPU

● Faster on NDP

● Similar on CPU/NDP

● Depends

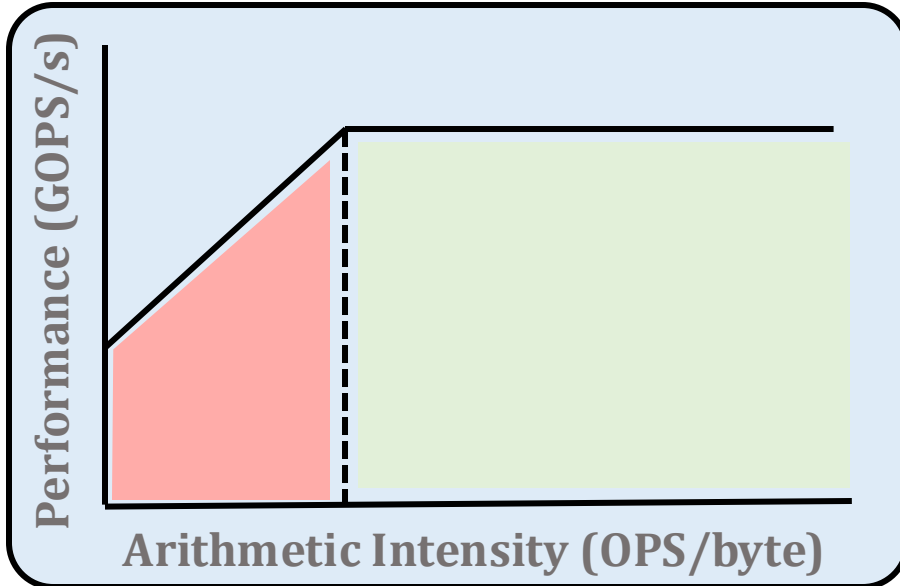
**LLC MPKI does not accurately account
for the NDP suitability of memory-bound applications**



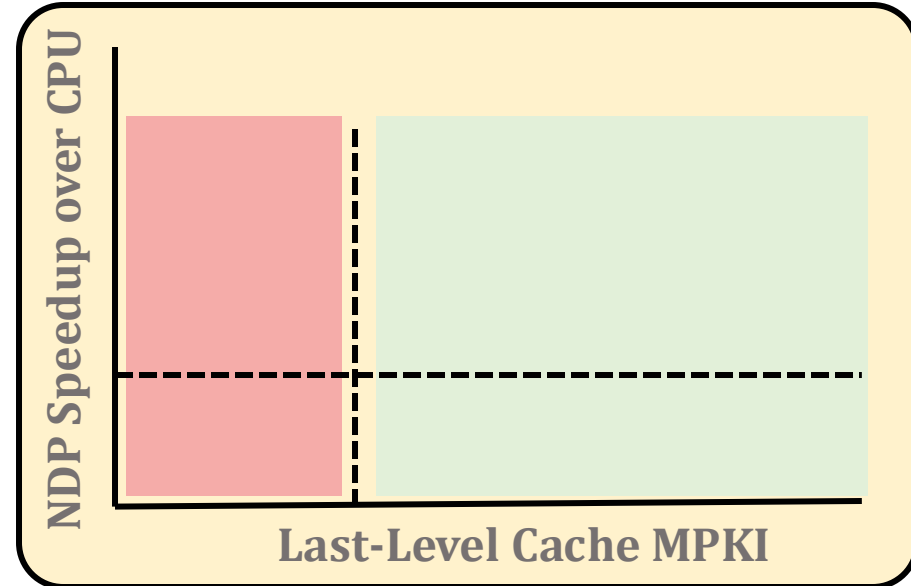
Identifying Memory Bottlenecks

- Multiple approaches to identify applications that:
 - suffer from data movement bottlenecks
 - take advantage of NDP
- Existing approaches are not comprehensive enough

Roofline model



High LLC MPKI

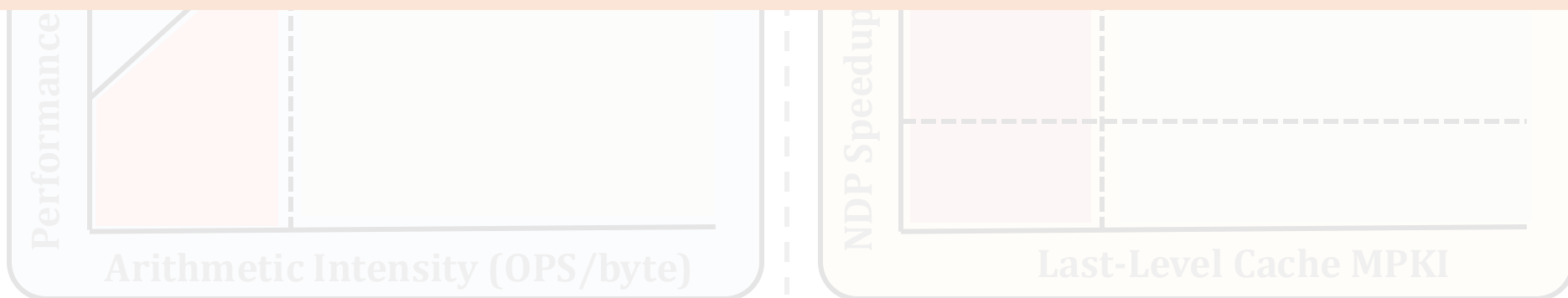


The Problem

- Multiple approaches to identify applications that:
 - suffer from data movement bottlenecks
 - take advantage of NDP

No available methodology can comprehensively:

- **identify** data movement bottlenecks
- **correlate** them with the **most suitable** data movement mitigation mechanism

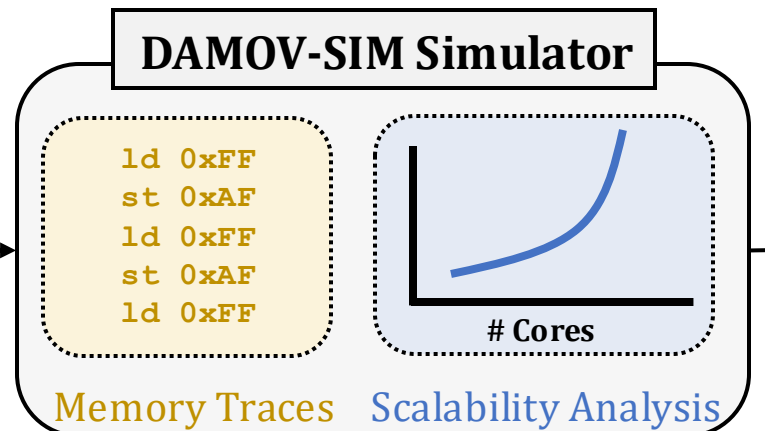
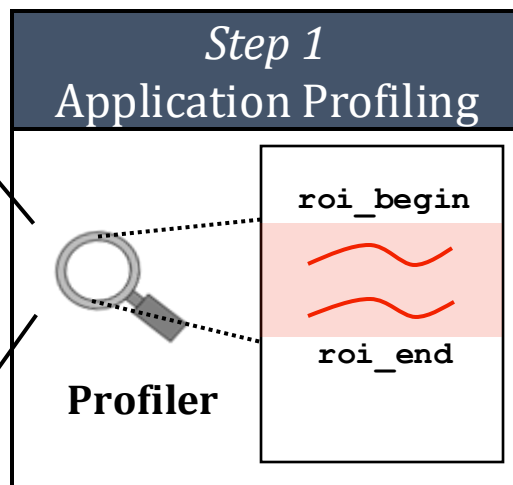
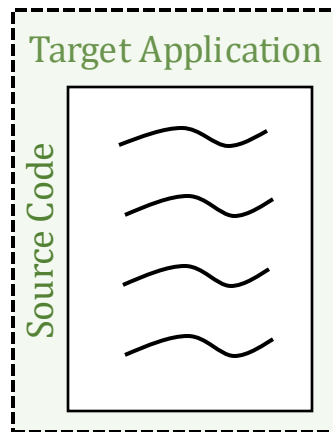


Our Goal

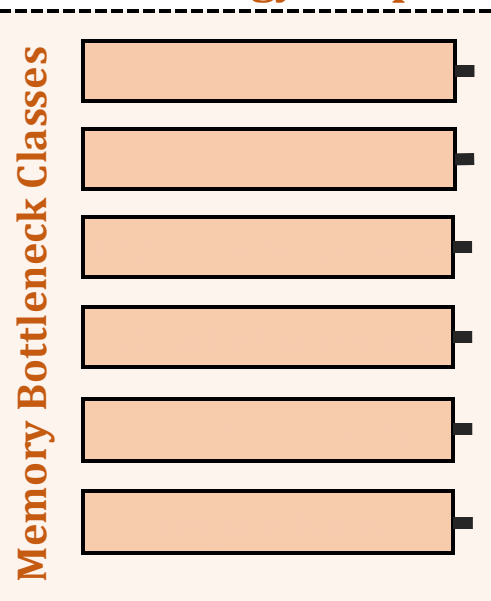
- **Our Goal:** develop a methodology to:
 - **methodically identify** sources of data movement bottlenecks
 - **comprehensively compare** compute- and memory-centric data movement mitigation techniques

Methodology Overview

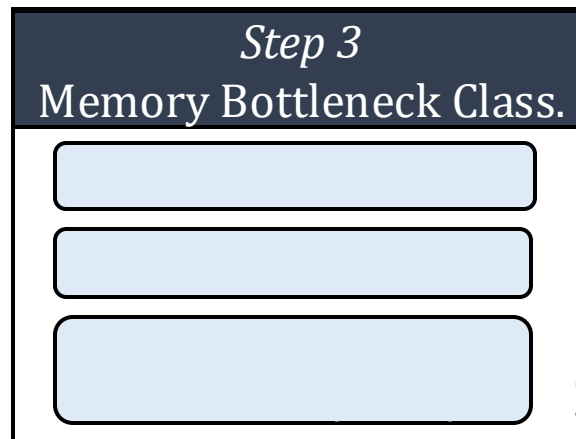
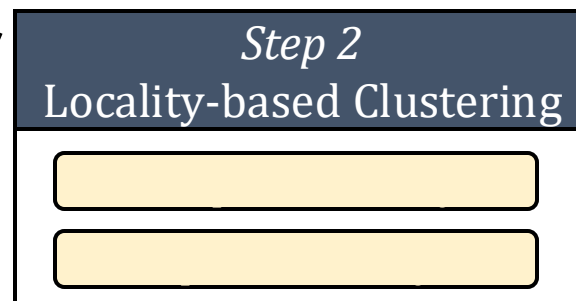
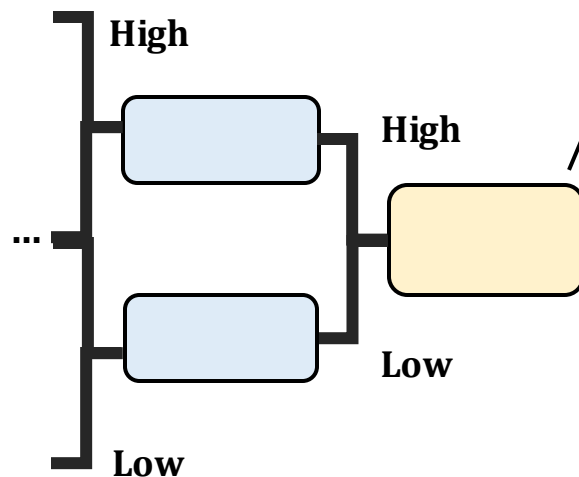
User Input



Methodology Output

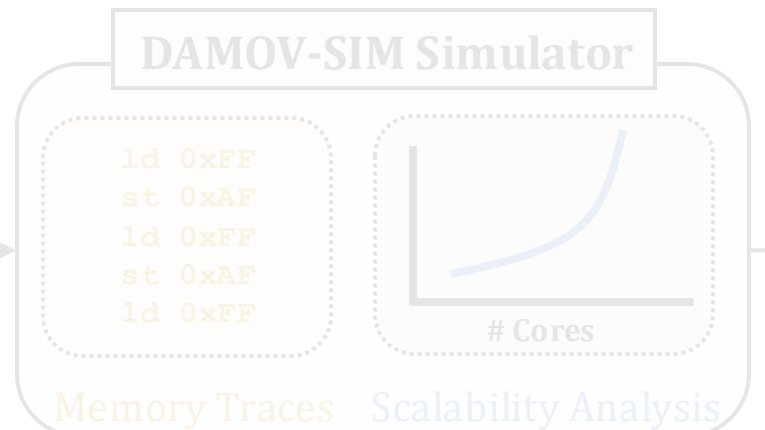
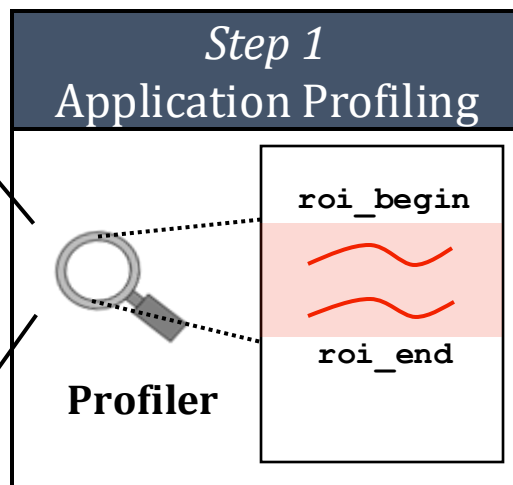
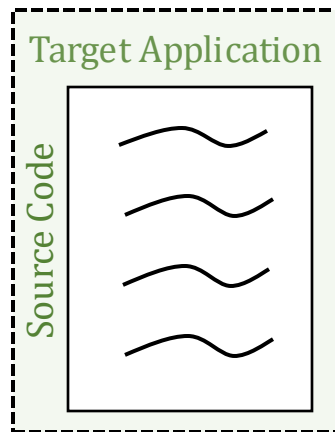


SAFARI

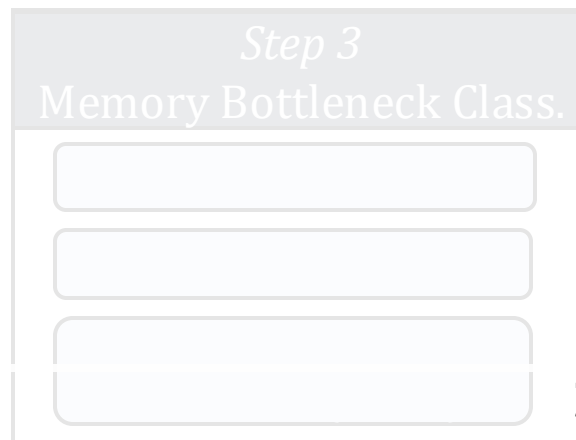
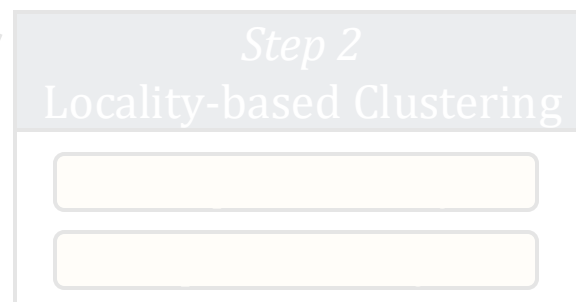
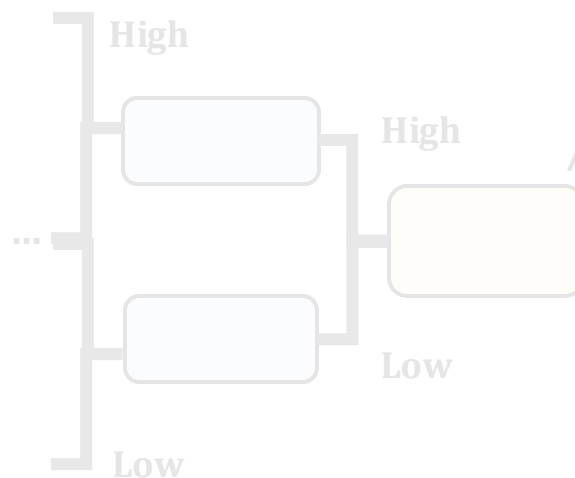


Methodology Overview

User Input



Methodology Output

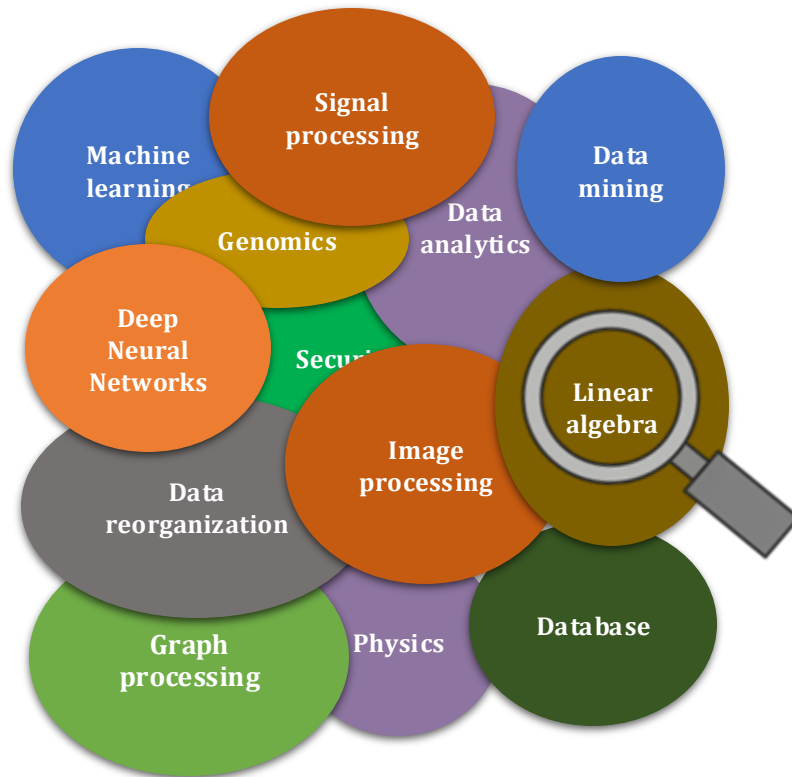


SAFARI

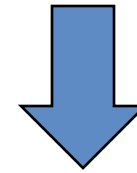
265

Step 1: Application Profiling

Goal: Identify **application functions** that suffer from **data movement bottlenecks**

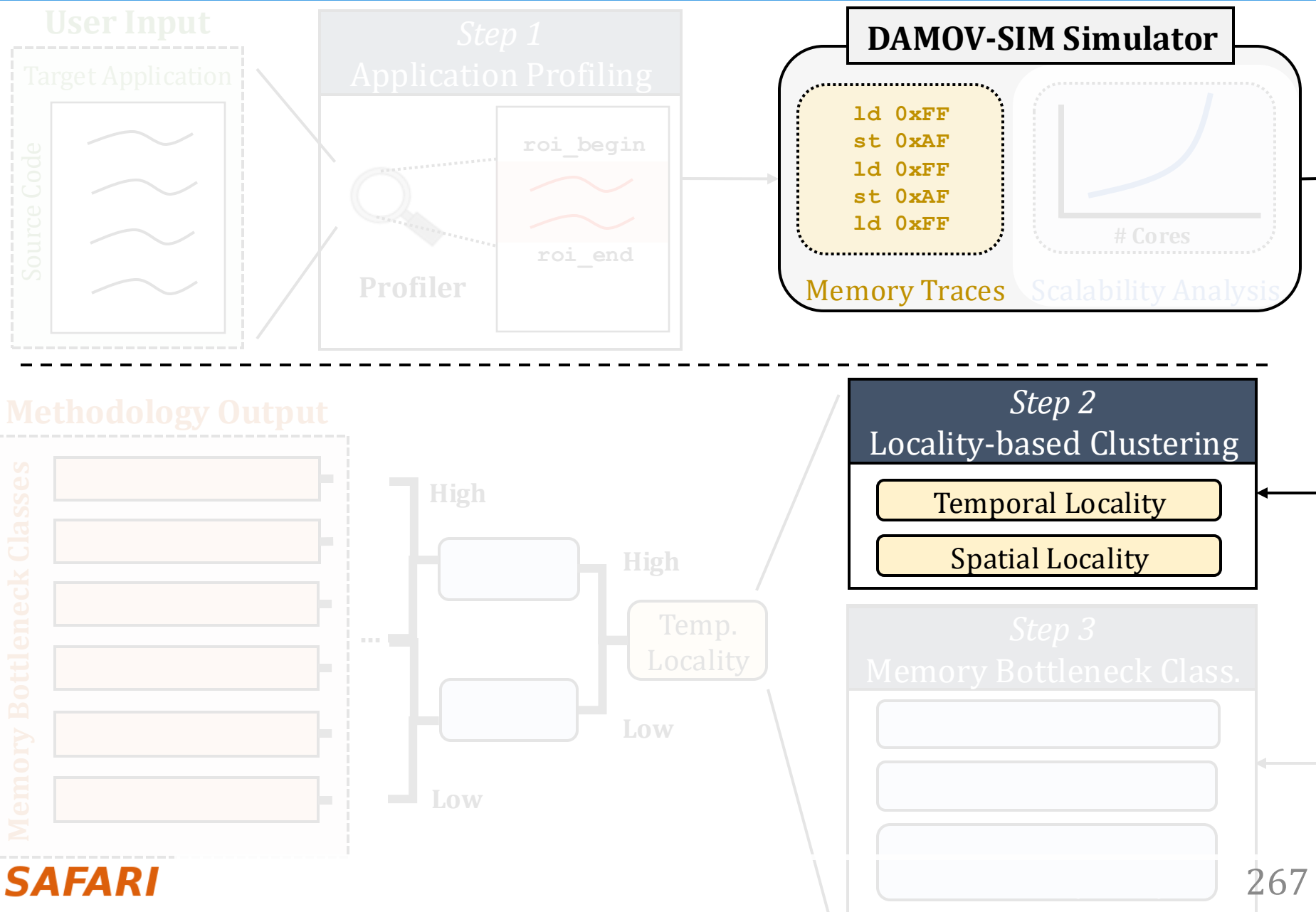


Hardware Profiling Tool:
Intel VTune



MemoryBound:
CPU is stalled due to load/store

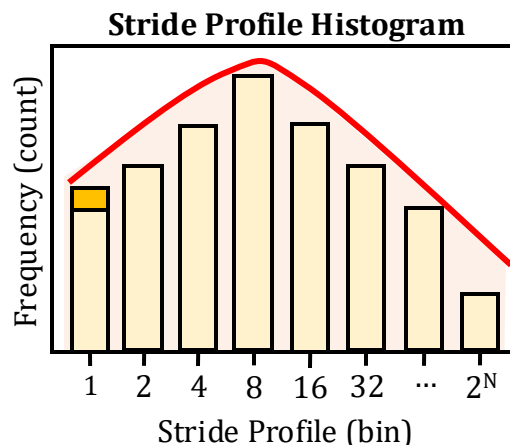
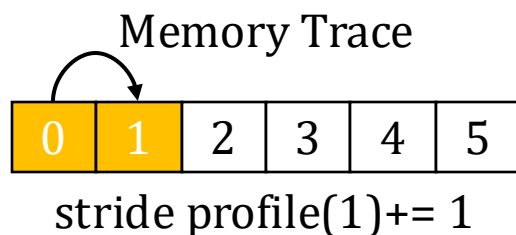
Methodology Overview



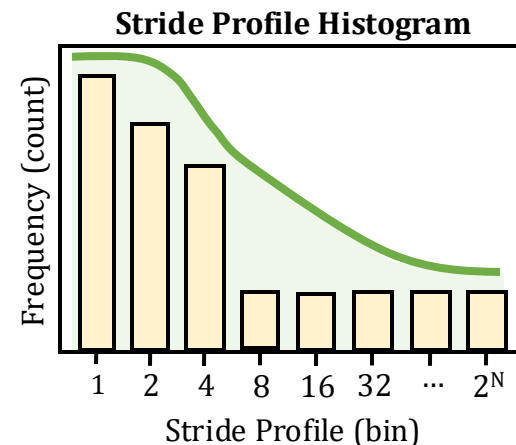
Step 2: Locality-Based Clustering

- **Goal:** analyze application's memory characteristics

Spatial Locality⁷



Low spatial locality

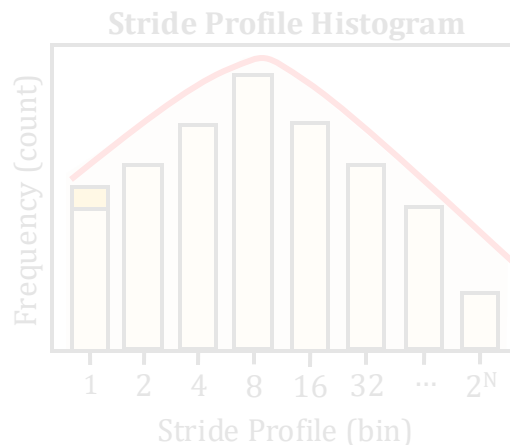
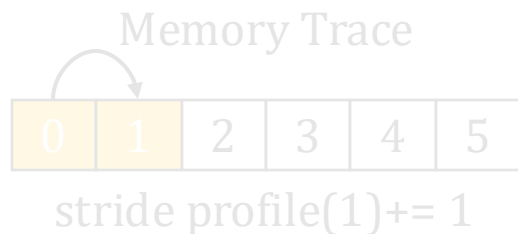


High spatial locality

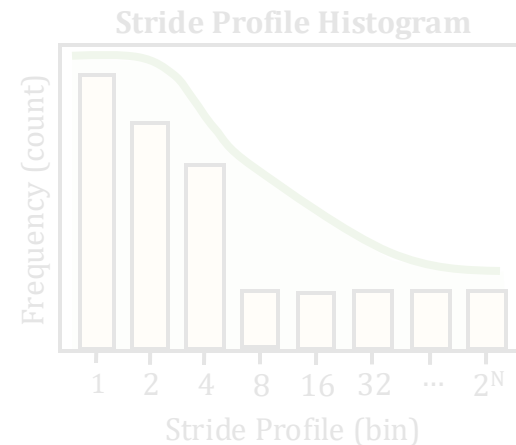
Step 2: Locality-Based Clustering

- **Goal:** analyze application's memory characteristics

Spatial Locality⁷

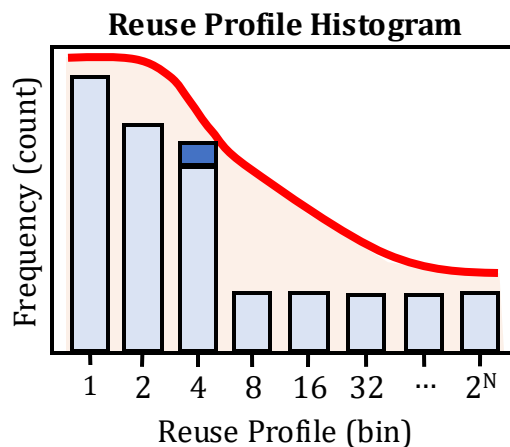
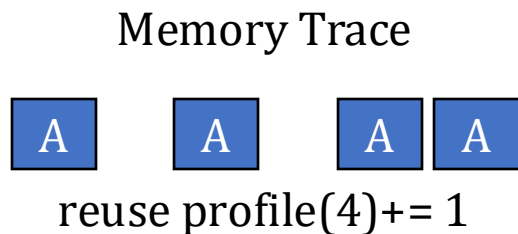


Low spatial locality

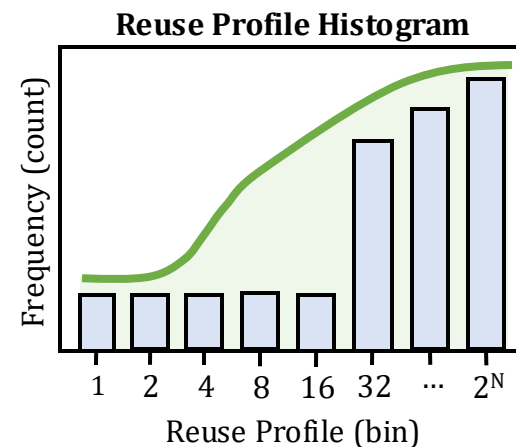


High spatial locality

Temporal Locality⁷

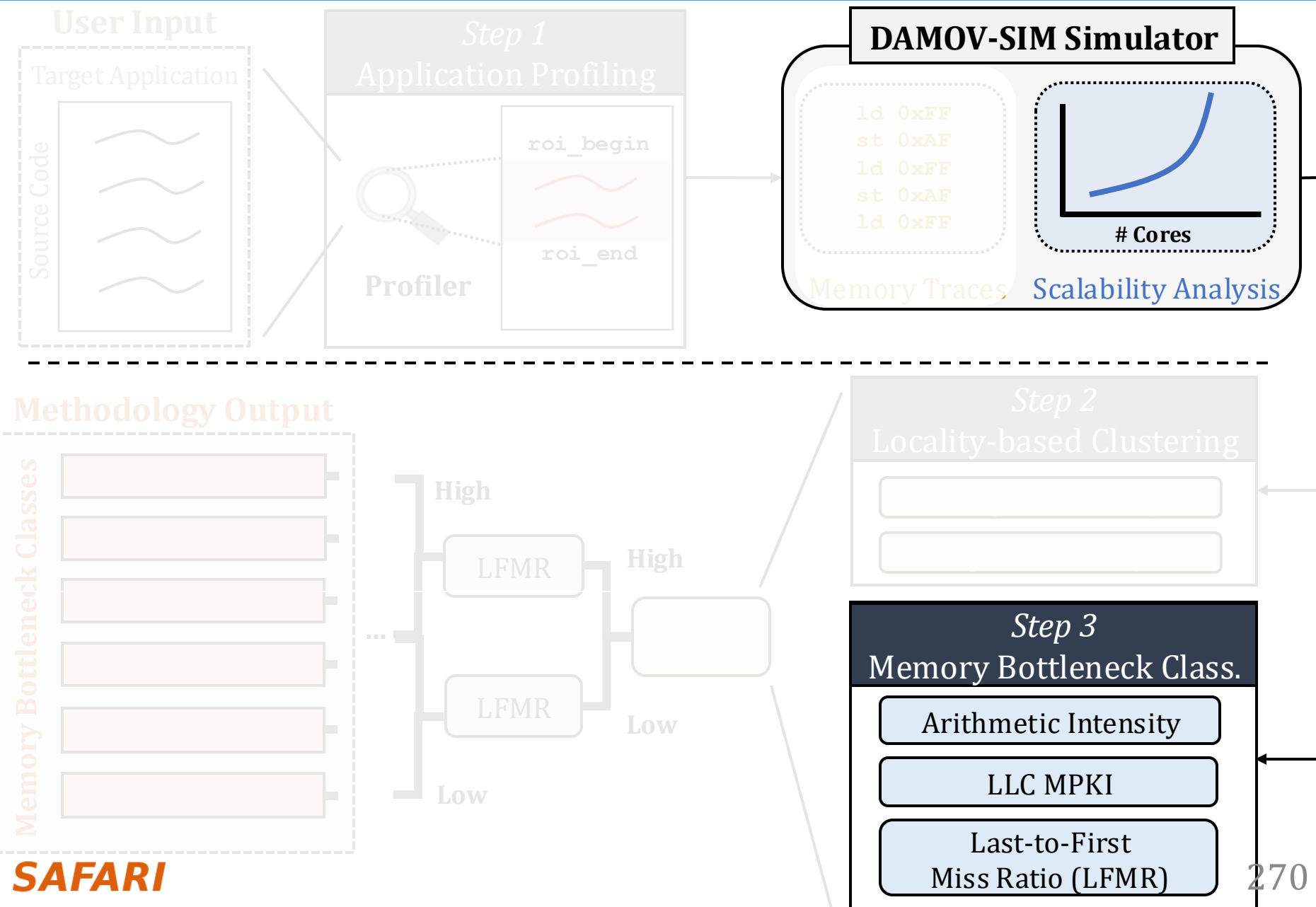


Low temporal locality



High temporal locality

Methodology Overview



Step 3: Memory Bottleneck Classification (1/2)

Arithmetic Intensity (AI)

- floating-point/arithmetic operations per L1 cache lines accessed
→ shows **computational intensity** per memory request

LLC Misses-per-Kilo-Instructions (MPKI)

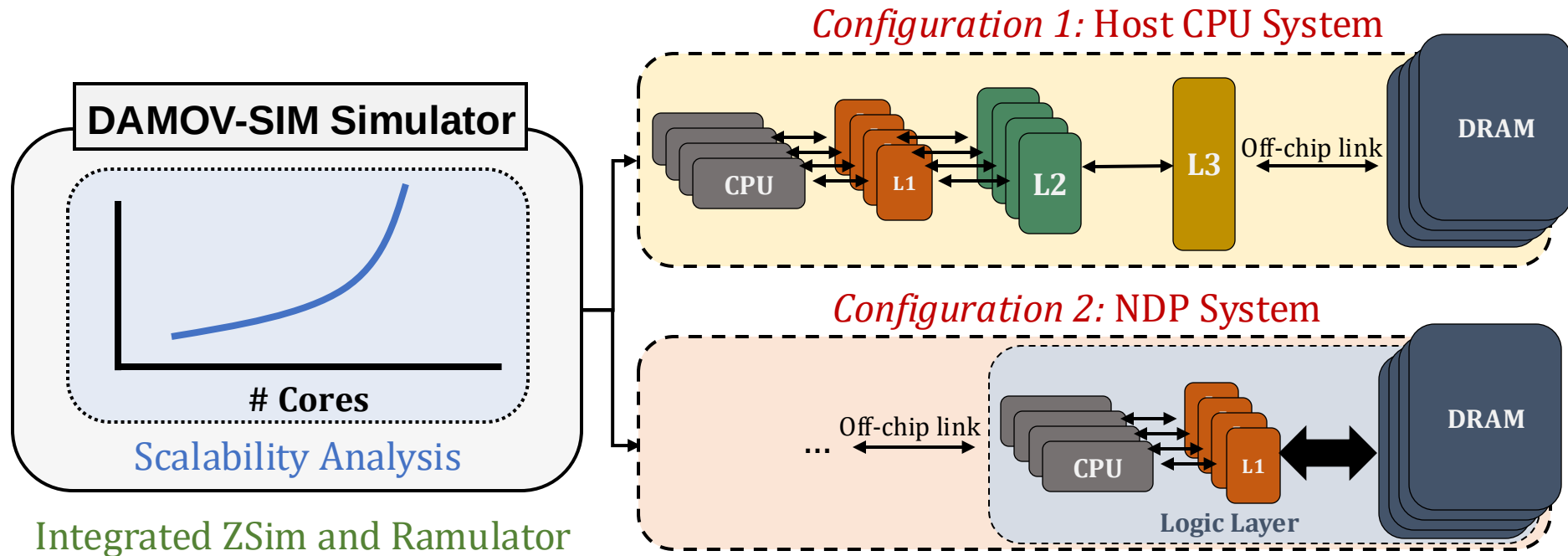
- LLC misses per one thousand instructions
→ shows **memory intensity**

Last-to-First Miss Ratio (LFMR)

- LLC misses per L1 misses
→ shows if an application **benefits from L2/L3 caches**

Step 3: Memory Bottleneck Classification (2/2)

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



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

Step 3: Memory Bottleneck Analysis

**Six classes of
data movement bottlenecks:**

each class $\leftrightarrow$ data movement
mitigation mechanism

Memory Bottleneck Class

1a: *DRAM
Bandwidth*

1b: *DRAM Latency*

1c: *L1/L2
Cache Capacity*

2a: *L3 Cache
Contention*

2b: *L1 Cache
Capacity*

2c: *Compute-Bound*

DAMOV is Open Source

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

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About



omutlu Update README.md

ce1b4ea 17 days ago 5 commits

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

README.md

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

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

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

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

Readme

Releases

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

Packages

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

Languages



DAMOV-SIM

DAMOV
Benchmarks

SAFARI

DAMOV is Open Source

- We open-source our [benchmark suite](https://github.com/CMU-SAFARI/DAMOV) and our [toolchain](https://github.com/CMU-SAFARI/DAMOV)

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About

DAMOV is a benchmark suite and a

Get DAMOV at:

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

README.md

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

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

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

Readme

Releases

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

Packages

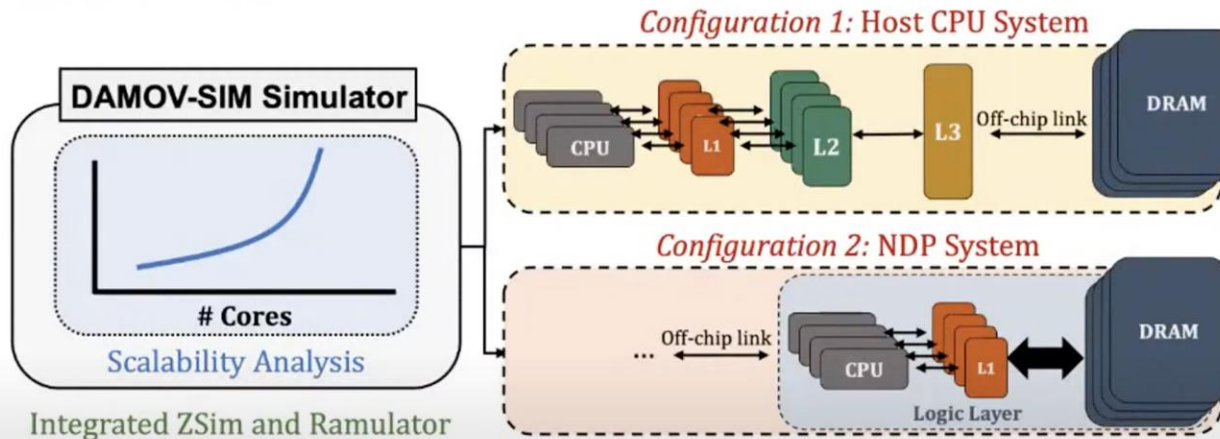
No packages published
[Publish your first package](#)

Languages

More on DAMOV Analysis Methodology & Workloads

Step 3: Memory Bottleneck Classification (2/)

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



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

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

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

352 views • Streamed live on Jul 22, 2021

18 0 SHARE SAVE ...



Onur Mutlu Lectures
17.7K subscribers

ANALYTICS

EDIT VIDEO

More on DAMOV Methods & Benchmarks

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

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

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

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

NANDITA VIJAYKUMAR, University of Toronto, Canada

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

MOHAMMAD SADROSADATI, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Fundamentally Energy-Efficient (Data-Centric) Computing Architectures

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

Computing Architectures with Minimal Data Movement

Concluding Remarks

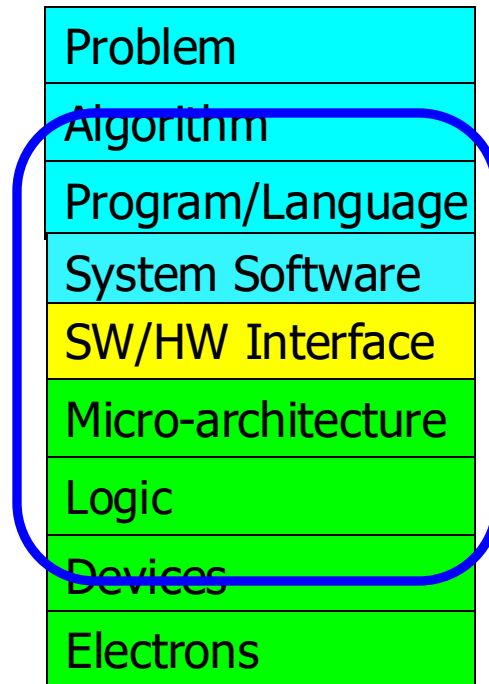
- We must design systems to be **balanced, high-performance, energy-efficient** (all at the same time) → intelligent systems
 - **Data-centric, data-driven, data-aware**
- Enable computation capability inside and close to memory
- This can
 - Lead to **orders-of-magnitude** improvements
 - **Enable new applications & computing platforms**
 - **Enable better understanding of nature**
 - ...
- Future of **truly memory-centric computing** is bright
 - We need to do research & design across the computing stack

Data-centric

Data-driven

Data-aware

We Need to Revisit the Entire Stack



We can get there step by step

We Need to Exploit Good Principles

- Data-centric system design
- All components intelligent
- Better (cross-layer) communication, better interfaces
- Better-than-worst-case design
- Heterogeneity
- Flexibility, adaptability

Open minds

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*

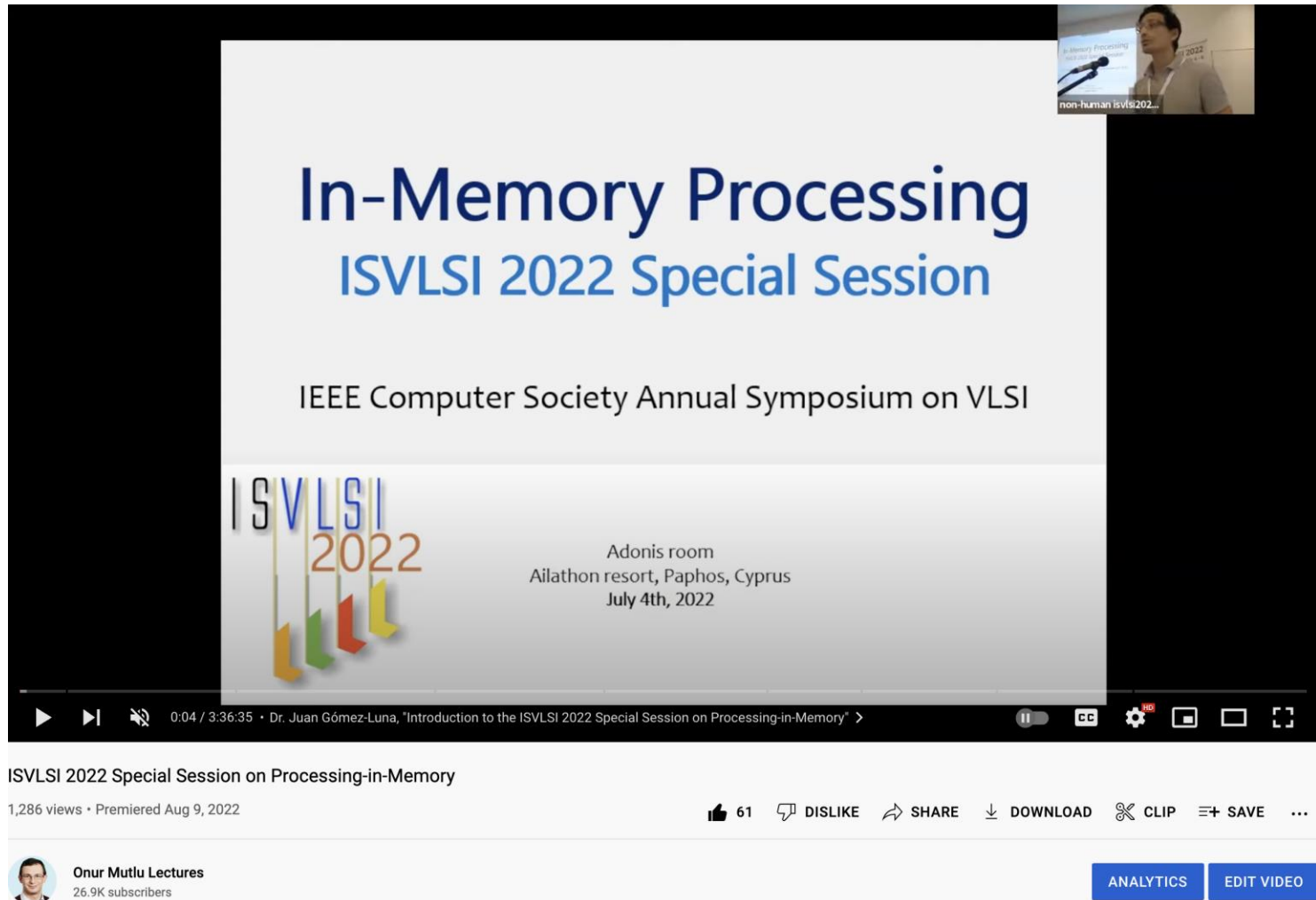
Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,

"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.*

Special Research Sessions & Courses (I)

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



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

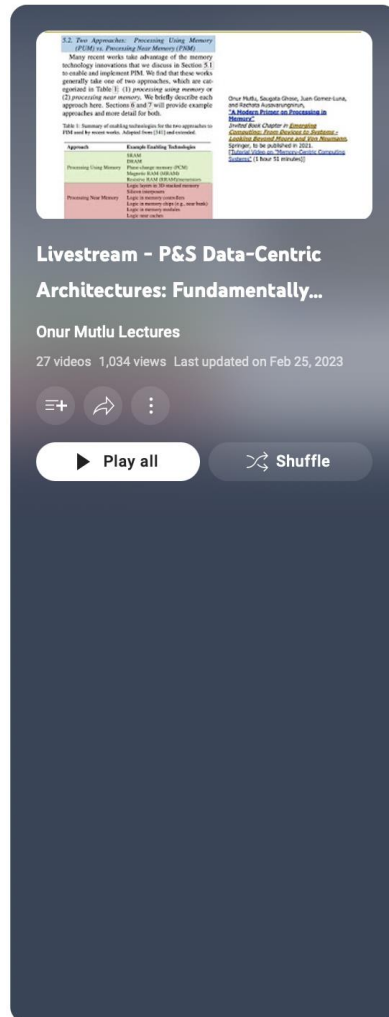
61 DISLIKE SHARE DOWNLOAD CLIP SAVE

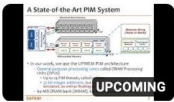
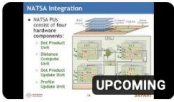
Onur Mutlu Lectures
26.9K subscribers

ANALYTICS EDIT VIDEO

Special Research Sessions & Courses (II)

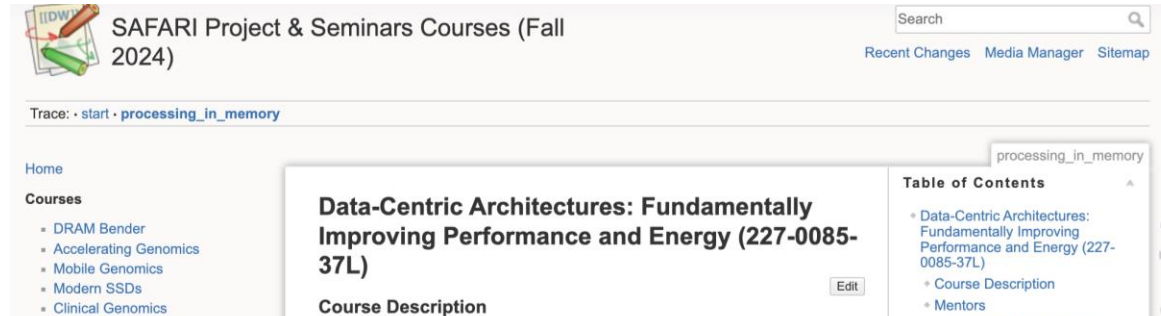
■ Special Session at ISVLSI 2022: 9 cutting-edge talks



- 19  **GenStore: In-Storage Filtering for High-Performance and Energy-Efficient Genome Analysis**
Onur Mutlu Lectures • Premieres 3/12/23, 7:00 PM
- 20  **Introduction to the ISVLSI 2022 Special Session on Processing-in-Memory**
Onur Mutlu Lectures • 286 views • 2 days ago
- 21  **Heterogeneous Data-Centric Architectures for Data-Intensive Applications: Case Studies in ML and DB**
Onur Mutlu Lectures • 2 waiting • Premieres 3/10/23, 7:00 PM
- 22  **Machine Learning Training on a Real Processing-In-Memory System**
Onur Mutlu Lectures • Premieres 3/14/23, 7:00 PM
- 23  **Exploiting Near-Data Processing to Accelerate Time Series Analysis**
Onur Mutlu Lectures • Premieres 3/11/23, 7:00 PM
- 24  **PiDRAM: An FPGA-Based Framework for End-To-End Evaluation of Processing-In-DRAM Techniques**
Onur Mutlu Lectures • Premieres 3/9/23, 7:00 PM
- 25  **The Road to Widely Deploying Processing-In-Memory: Challenges and Opportunities**
Onur Mutlu Lectures • 399 views • 1 day ago
- 26  **SparseP: Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures**
Onur Mutlu Lectures • 1 waiting • Premieres 3/13/23, 7:00 PM
- 27  **HPCA 2023 Tutorial: Real-World Processing-in-Memory Architectures**
Onur Mutlu Lectures • 1.6K views • Streamed 10 days ago

Processing-in-Memory Course (Fall 2024)

- Short weekly lectures
- Hands-on projects



SAFARI Project & Seminars Courses (Fall 2024)

Trace: • start • processing_in_memory

Home

Courses

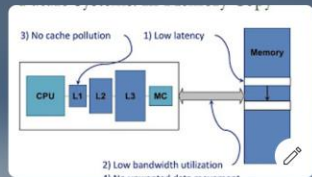
- DRAM Bender
- Accelerating Genomics
- Mobile Genomics
- Modern SSDs
- Clinical Genomics

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

Course Description

Table of Contents

- Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)
- Course Description
- Mentors
- Lecture Video Playlist on YouTube
- Fall 2024 Meetings/Schedule
- Past Lecture Video Playlists on YouTube
- Learning Materials
- Assignments



Livestream - Data-Centric Architectures: ...

Onur Mutlu Lectures

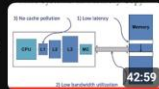
Playlist • Public • 4 videos • 179 views



Play all



Sort



PIM Course: Lecture 4: Processing-Using-Memory for Data Manipulation (Fall 2024)

Onur Mutlu Lectures • 338 views • Streamed 2 days ago



PIM Course: Lecture 3: Processing-Near-Memory (Fall 2024)

Onur Mutlu Lectures • 324 views • Streamed 9 days ago



PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks (Fall 2024)

Onur Mutlu Lectures • 434 views • Streamed 2 weeks ago



PIM Course: Lecture 1: Data-Centric Architectures: Improving Performance & Energy (Fall 2024)

Onur Mutlu Lectures • 337 views • Streamed 3 weeks ago

compute units of energy bottleneck. Recent costs dominate energy consumption. and the energy in consumer is a huge burden of modern computing systems. This phenomenon is and the processor, which leads to the data movement

the learning, computational biology, graph processing, suffer greatly from the data movement bottleneck. processes, relatively low data reuse, low cache line is per accessed byte), and large datasets that greatly workloads cannot usually compensate for the data nt bottleneck, we need a paradigm shift from the on takes place in the compute units, to a more data- oser to or inside where the data resides. This paradigm

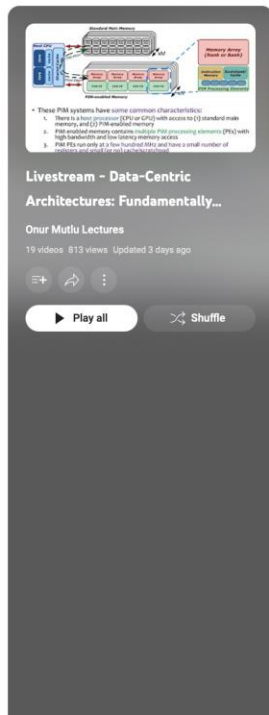
https://safari.ethz.ch/projects_and_seminars/fall2024/doku.php?id=processing_in_memory

<https://www.youtube.com/playlist?list=PL5Q2soXY2Zi9DSQg7OAlsNO8dOQKx>

F9sl

Processing-in-Memory Course (Spring 2023)

- Short weekly lectures
- Hands-on projects



- PIM Course: Lecture 1: Data-Centric Architectures: Improving Performance & Energy (Spring 2023)**
Onur Mutlu Lectures • 1.1K views • Streamed 3 months ago
1:14:16
- PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks (Spring 2023)**
Onur Mutlu Lectures • 332 views • 2 months ago
16:37
- ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads**
Onur Mutlu Lectures • 1.5K views • Streamed 2 months ago
6:27:39
- PIM Course: Lecture 3: Real-world PIM: UPMEM PIM (Spring 2023)**
Onur Mutlu Lectures • 411 views • 2 months ago
15:43
- PIM Course: Lecture 4: Real-world PIM: Microbenchmarking of UPMEM PIM (Spring 2023)**
Onur Mutlu Lectures • 188 views • 2 months ago
24:10
- Análisis Experimental de una Arquitectura PIM - Juan Gómez Luna - Lecture in Spanish @ U. de Córdoba**
Onur Mutlu Lectures • 169 views • 2 months ago
2:27:12
- PIM Course: Lecture 5: Real-world PIM: Samsung HBM-PIM (Spring 2023)**
Onur Mutlu Lectures • 483 views • 2 months ago
24:08
- PIM Course: Lecture 6: Real-world PIM: SK Hynix A1M (Spring 2023)**
Onur Mutlu Lectures • 573 views • 1 month ago
35:50
- PIM Course: Lecture 7: Real-world PIM: Samsung AxDIMM (Spring 2023)**
Onur Mutlu Lectures • 325 views • 1 month ago
21:32

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

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

Search

Recent Changes Media Manager Sitemap

Trace: • heterogeneous_systems • processing_in_memory

Home

Courses

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

processing_in_memory

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

Course Description

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

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

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

Prerequisites of the course:

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

Table of Contents

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

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

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>

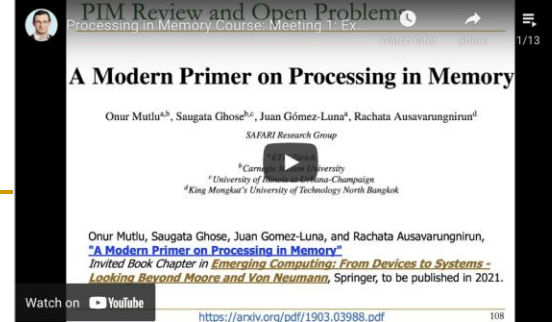
■ Youtube Livestream (Spring 2022):

- <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>



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 Tutorials [ISCA'23, ASPLOS'23, HPCA'23]

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



ISCA 2023 Real-World PIM Tutorial

Search

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

Trace: • [start](#)

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

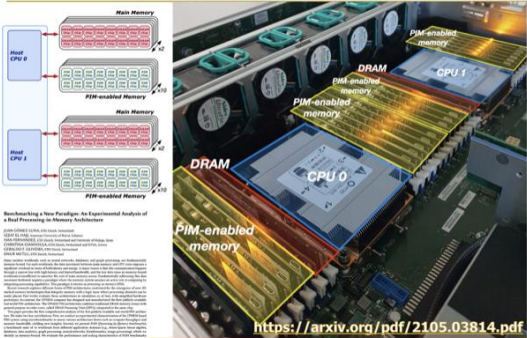
Tutorial Description

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

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

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

2,560-DPU Processing-in-Memory System



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

Table of Contents

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

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.

Real PIM Tutorial [ISCA 2023]

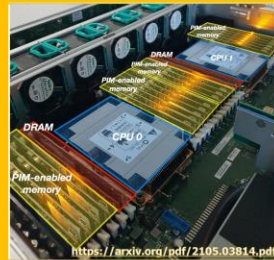
■ June 18: Lectures + Hands-on labs + Invited talks

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

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



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



International Symposium on Computer Architecture (ISCA)

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

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

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

SAFARI zoom

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

Onur Mutlu Lectures
33.9K subscribers

Subscribed

57

Share

Download

Clip

...

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

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

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

Tutorial Materials

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

SAFARI

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

Tutorial Materials

The image shows a presentation slide from a video recording. The slide has a white background with a light blue header. The header text reads "ASPLOS 2023 Tutorial" and "Real-world Processing-in-Memory Systems for Modern Workloads". The main title of the presentation is "Accelerating Modern Workloads on a General-purpose PIM System" in a large, bold, blue font. Below the title, the presenters are listed as "Dr. Juan Gómez Luna" and "Professor Onur Mutlu". At the bottom of the slide, there are logos for "ETH zürich" and "SAFARI!". The video player interface at the bottom shows a red progress bar, a timestamp of 6:26:31 / 6:27:38, and the date "Sunday, March 26, 2023". On the right side of the video frame, there is a small inset window showing a person, likely the presenter, and a Zoom logo.

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/asplios-...>

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

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

Real PIM Tutorial [HPCA 2023]

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

Real-world Processing-in-Memory Architectures

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, Mythic) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Allbaba) 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. But, there is more to come: Academia and Industry are actively exploring other types of PIM by, e.g., exploiting the analog operation of DRAM, SRAM, flash memory and emerging non-volatile memories.

PIM can provide large improvements in both performance and energy consumption, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to examine and research adoption issues of PIM using especially learnings from real PIM systems that are available today.

This tutorial focuses on the latest advances in PIM technology. 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 using real PIM systems, and (4) shed light on how to enable the adoption of PIM in future computing systems.

2,560-DPU Processing-in-Memory System

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

Goal: Processing Inside Memory

Processor Core

Memory

Database

Graphs

Media

Query

Results

Interconnect

■ Many questions ... How do we design the:

- compute-capable memory & controllers?
- processors & communication units?
- software & hardware interfaces?
- system software, compilers, languages?
- algorithms & theoretical foundations?

HPCA 2023 Tutorial: Real-World Processing-in-Memory Architectures

Onur Mutlu Lectures

32.1K subscribers

1.8K views · Streamed 1 month ago · Livestream - P&S Data-Centric Architectures: Fundamentally Improving Performance and Energy (Fall 2022)

HPCA 2023 Tutorial: Real-World Processing-in-Memory Architectures

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

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/>

Real PIM Tutorial [MICRO 2023]

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

MICRO 2023 Real-World PIM Tutorial

Search

Recent Changes Media Manager Sitemap

Trace: - start

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

Tutorial Description

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

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

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

2,560-DPU Processing-in-Memory System

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

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

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

2,560-DPU Processing-in-Memory System

Host CPU 0, Host CPU 1, Main Memory, PIM-enabled Memory, DRAM, CPU 0, CPU 1, PIM-enabled memory.

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

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

Onur Mutlu Lectures

34.6K subscribers

Live in 74 days

October 29 at 6:00 PM

Notify me

5

Share

Save

Agenda (Tentative, October 29, 2023)

Lectures

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

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

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

PIM Tutorial at HEART 2024

HEART 2024 Memory-Centric Computing Systems Tutorial

Friday, June 21, Porto, Portugal

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

Program: <https://events.safari.ethz.ch/heart24-memorycentric-tutorial/>

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



PIM Tutorial at ISCA 2024

ISCA 2024 Memory-Centric Computing Systems Tutorial

Saturday, June 29, Buenos Aires, Argentina

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

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

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



PIM Tutorial at MICRO 2024

MICRO 2024 - Tutorial on Memory-Centric Computing Systems

Saturday, November 2nd, Austin, Texas, USA

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

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

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



PIM Tutorial at PPOPP 2025

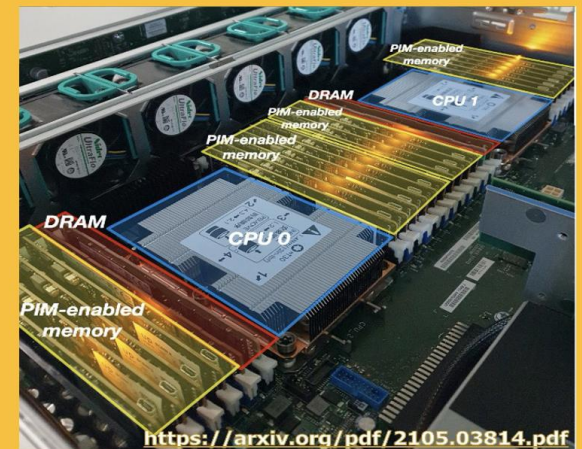
PPoPP 2025 - Tutorial on Memory-Centric Computing Systems

March 1st – March 5th, Las Vegas, Nevada, USA

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

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

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



Referenced Papers, Talks, Artifacts

- All are available at

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

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

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

Open Source Tools: SAFARI GitHub

**SAFARI Research Group at ETH Zurich and Carnegie Mellon University**
Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.
241 followers · ETH Zurich and Carnegie Mellon U... · <https://safari.ethz.ch/> · omutlu@gmail.com

[Overview](#) [Repositories 80](#) [Projects](#) [Packages](#) [People 13](#)

Pinned

[Customize pins](#)

 **ramulator** Public

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

C++ 442 195

 **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 100 38

 **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++ 213 120

 **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 208 41

 **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 104 26

 **Pythia** Public

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

C++ 85 25

1st Workshop on Memory-Centric Computing: Research Challenges & Closing Remarks

Geraldo F. Oliveira

<https://geraldofojunior.github.io>

ASPLOS 2025

30 March 2025