Memory-Centric Computing

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
27 September 2022
Microsoft Azure AHA Learning Session 2022

SAFARI
ETH Zürich
Carnegie Mellon
The Problem

Computing

is Bottlenecked by Data
Important workloads are all data intensive

They require rapid and efficient processing of large amounts of data

Data is increasing
  - We can generate more than we can process
Huge Demand for Performance & Efficiency

Exponential Growth of Neural Networks

1800x more compute
In just 2 years

Tomorrow, multi-trillion parameter models

https://www.youtube.com/watch?v=x2-qB0J7KHW
Data is Key for Future Workloads

In-memory Databases
[Ma'o+, EuroSys’12; Clapp+ (Intel), IISWC’15]

Graph/Tree Processing
[Xu+, IISWC’12; Umuroglu+, FPL’15]

In-Memory Data Analytics
[Clapp+ (Intel), IISWC’15; Awan+, BDCloud’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

VP9
Video Playback
Google’s video codec

VP9
Video Capture
Google’s video codec
Data Overwhelms Modern Machines

Data → performance & energy bottleneck

Chrome

TensorFlow Mobile

VP9

Video Playback

Google’s video codec

Video Capture

Google’s video codec
Data is Key for Future Workloads

Development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

Genome Analysis

1 Sequencing
2 Read Mapping
3 Variant Calling
4 Scientific Discovery

Data → performance & energy bottleneck
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 ▼

[Open arxiv.org version]
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

Damla Senol Cali, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

Briefings in Bioinformatics, bby017, https://doi.org/10.1093/bib/bby017
Published: 02 April 2018 Article history ▼

Oxford Nanopore MinION

Data → performance & energy bottleneck
One Problem with (Genome) Analysis Today

Special-Purpose Machine for Data Generation

FAST

General-Purpose Machine for Data Analysis

SLOW

Slow and inefficient processing capability

SAFARI This picture is similar for many “data generators & analyzers” today
Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"
[Slides (pptx)(pdf)]
[Talk Video (1 hour 2 minutes)]
FPGA-based Near-Memory Analytics


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
Data Overwhelms Modern Machines …

- Storage/memory capability
- Communication capability
- Computation capability

- Greatly impacts robustness, energy, performance, cost
A Computing System

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

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

Perils of Processor-Centric Design

Most of the system is dedicated to storing and moving data Yet, system is still bottlenecked by memory
Deeper and Larger Memory Hierarchies

Core Count: 8 cores/16 threads

L1 Caches: 32 KB per core

L2 Caches: 512 KB per core

L3 Cache: 32 MB shared

AMD Ryzen 5000, 2020

AMD 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)

A lot of SRAM
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


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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

- Amirali Boroumand
- Rachata Ausavarungnirun
- Aki Kuusela
- Allan Knies

- Saugata Ghose
- Eric Shiu

- Youngsok Kim
- Rahul Thakur
- Parthasarathy Ranganathan

- Daehyun Kim
- Onur Mutlu

SAFARI
An Intelligent Architecture

Handles Data Well
How to Handle Data Well

- **Ensure data does not overwhelm** the components
  - via intelligent algorithms
  - via intelligent architectures
  - via whole 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
Architectures for Intelligent Machines

Data-centric

Data-driven

Data-aware
Onur Mutlu,  
"Intelligent Architectures for Intelligent Computing Systems"  
[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
We Need to Revisit the Entire Stack

We can get there step by step
Data-Centric (Memory-Centric) Architectures
Data-Centric Architectures: Properties

- **Process data where it resides** *(where it makes sense)*
  - Processing in and near memory structures

- **Low-latency and low-energy data access**
  - Low latency memory
  - Low energy memory

- **Low-cost data storage and 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)

https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php
Processing in Memory: An Old Idea


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.

Fig. 1. Cellular sorting array I.

https://doi.org/10.1109/T-C.1969.222754
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?

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

- **Pull from Applications & Systems**
  - Data access is the major bottleneck
  - Systems are energy & power limited
  - Data movement much more energy-hungry than computation
  - Need all at the same time: performance, energy, sustainability
  - We can improve all metrics by minimizing data movement

- **Designs are squeezed in the middle**
Processing-in-Memory Landscape Today

And, many other experimental chips and startups
Memory Scaling Issues Are Real

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

Memory Scaling: A Systems Architecture Perspective

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

Memory Scaling Issues Are Real

- Onur Mutlu,
  "The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"
  [Slides (pptx) (pdf)]

---

The RowHammer Problem
and Other Issues We May Face as Memory Becomes Denser

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch
https://people.inf.ethz.ch/omutlu

SAFARI  https://people.inf.ethz.ch/omutlu/pub/rowhammer-and-other-memory-issues_date17.pdf  40
Memory Scaling Issues Are Real

- Slides from COSADE 2019 (pptx)
- Slides from VLSI-SOC 2020 (pptx) (pdf)
- Talk Video (1 hr 15 minutes, with Q&A)

RowHammer: A Retrospective

Onur Mutlu§‡
§ETH Zürich

Jeremie S. Kim‡§
‡Carnegie Mellon University
The Story of RowHammer Tutorial …

Onur Mutlu,
"Security Aspects of DRAM: The Story of RowHammer"

[Slides (pptx)(pdf)]
[Tutorial Video (57 minutes)]
10 Years of RowHammer in 20 Minutes

- Onur Mutlu, "The Story of RowHammer"
  Invited Talk at the Workshop on Robust and Safe Software 2.0 (RSS2), held with the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, 28 February 2022.
  [Slides (pptx) (pdf)]

[Video link: https://www.youtube.com/watch?v=ctKTRyi96Bk]
Main Memory Needs
Intelligent Controllers
Emerging Memories Also Need Intelligent Controllers

- Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger,
  "Architecting Phase Change Memory as a Scalable DRAM Alternative"
  One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight. 2022 Persistent Impact Prize.

Architecting Phase Change Memory as a Scalable DRAM Alternative

Benjamin C. Lee†  Engin Ipek†  Onur Mutlu‡  Doug Burger†

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

‡Computer Architecture Laboratory
Carnegie Mellon University
Pittsburgh, PA
onur@cmu.edu
Intelligent Memory Controllers Can Avoid Many Failures & Enable Better Scaling
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?

Source: V. Milutinovic
Or This?

Source: V. Milutinovic
Challenge and Opportunity for Future

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)
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.

Today’s Computing Systems

- Processor centric

- All data processed in the processor → at great system cost
Yet …

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

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

The Performance Perspective

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


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

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

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

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

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

‡Desktop Platforms Group
Intel Corporation
chris.wilkerson@intel.com
The Performance Perspective (Today)

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

![Pipeline slot breakdown graph]

The Energy Perspective

Communication Dominates Arithmetic

Dally, HiPEAC 2015

- 64-bit DP 20pJ
- 256-bit buses
- 256-bit access 8 kB SRAM
- DRAM Rd/Wr
- Efficient off-chip link

- 20mm

- 26 pJ
- 256 pJ
- 16 nJ
- 500 pJ
- 50 pJ
- 1 nJ
A memory access consumes $\sim 100\text{-}1000\times$ the energy of a complex addition.
Data Movement vs. Computation Energy

Energy for a 32-bit Operation (log scale)

- ADD (int)
- ADD (float)
- Register File
- MULT (int)
- MULT (float)
- SRAM Cache
- DRAM

Energy (pJ)

ADD (int) Relative Cost

A memory access consumes 6400X the energy of a simple integer addition.
62.7% of the total system energy is spent on data movement
We Do Not Want to Move Data!

Communication Dominates Arithmetic

Dally, HiPEAC 2015

A memory access consumes \( \sim 100\text{-}1000X \) 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

Many questions ... How do we design the:

- compute-capable memory & controllers?
- processors & communication units?
- software & hardware interfaces?
- system software, compilers, languages?
- algorithms & theoretical foundations?
A Modern Primer on Processing in Memory

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

SAFARI Research Group

\textsuperscript{a}ETH Zürich
\textsuperscript{b}Carnegie Mellon University
\textsuperscript{c}University of Illinois at Urbana-Champaign
\textsuperscript{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"

SAFARI

We Need to Think Differently from the Past Approaches
Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory
Memory similar to a “conventional” accelerator
Accelerating In-Memory Graph Processing

- Large graphs are everywhere (circa 2015)
  - 36 Million Wikipedia Pages
  - 1.4 Billion Facebook Users
  - 300 Million Twitter Users
  - 30 Billion Instagram Photos

- Scalable large-scale graph processing is challenging
  - Speedup
    - 32 Cores: 0
    - 128...: +42%
Key Bottlenecks in Graph Processing

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

1. Frequent random memory accesses
2. Little amount of computation
Opportunity: 3D-Stacked Logic+Memory

Hybrid Memory Cube Consortium

Other “True 3D” technologies under development
Tesseract System for Graph Processing

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

SAFARI Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
Tesseract System for Graph Processing

Host Processor
Memory-Mapped Accelerator Interface
(Noncacheable, Physically Addressed)

Crossbar Network

In-Order Core

Communications via Remote Function Calls

Message Queue

DRAM

NI
Tesseract System for Graph Processing

Host Processor
Memory-Mapped Accelerator Interface
(Noncacheable, Physically Addressed)

Memory

Logic

Crossbar Network

Prefetching

LP
PF Buffer
MTP

Message Queue

DRAM Controller
NI
Evaluated Systems

<table>
<thead>
<tr>
<th>DDR3-OoO</th>
<th>HMC-OoO</th>
<th>HMC-MC</th>
<th>Tesseract</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
<td>8 OoO 8GHz</td>
<td>32 Tesseract Cores</td>
</tr>
<tr>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
<td>128 In-Order 2GHz</td>
<td></td>
</tr>
<tr>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
<td>128 In-Order 2GHz</td>
<td></td>
</tr>
<tr>
<td>102.4GB/s</td>
<td>640GB/s</td>
<td>640GB/s</td>
<td>8TB/s</td>
</tr>
</tbody>
</table>

SAFARI Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
Tesseract Graph Processing Performance

>13X Performance Improvement

On five graph processing algorithms

- DDR3-OoO
- HMC-OoO
- HMC-MC
- Tesseract
- Tesseract-LP
- Tesseract-LP-MTP

>13X Performance Improvement on five graph processing algorithms

Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
Tesseract Graph Processing System Energy

HMC-OoO

Tesseract with Prefetching

> 8X Energy Reduction

Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
More on Tesseract


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

Junwhan Ahn, Sungpack Hong$, Sungjoo Yoo, Onur Mutlu$, Kiyoungh Choi
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr
Seoul National University $Oracle Labs $Carnegie Mellon University
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"


[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
Eliminating the Adoption Barriers

Processing-in-Memory in the Real World
Processing-in-Memory Landscape Today

This does not include many experimental chips and startups
UPMEMP Processing-in-DRAM Engine (2019)

- Processing in DRAM Engine
  - Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.

- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth

UPMEM Memory Modules

- E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
- P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz
2,560-DPU Processing-in-Memory System

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

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
IZZAT EL HAJI, American University of Beirut, Lebanon
IVAN FERNÁNDEZ, ETH Zürich and University of Málaga, Spain
CHRISTINA GIANNOULI, ETH Zürich, Switzerland and NTUA, Greece
GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland

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

Recent research exploits different forms of PIM architecture, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware prototypes. In contrast, the UPNEM company has designed and manufactured the first publicly available real-world PIM architecture. The UPNEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM Processing Units (DPUs), integrated in the same chip.

This paper provides the first comprehensive analysis of the first publicly available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPNEM board to analyze performance and energy consumption compared to a state-of-the-art CPU and GPU counterparts. An extensive evaluation conducted on two real UPNEM-board PIM systems with 34 and 2,560 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.

More on the UPMEM PIM System

https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidy1gBxUz7xRPS-wisBN&index=26

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

1,120 views • Oct 31, 2020

Onur Mutlu Lectures
16.7K subscribers

https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidy1gBxUz7xRPS-wisBN&index=26
Experimental Analysis of the UPMEM PIM Engine

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

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

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

Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware prototypes. In contrast, the UPMEM company has designed and manufactured the first publicly-available real-world PIM architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called *DRAM Processing Units (DPUs)*, integrated in the same chip.

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

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

Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, Onur Mutlu

https://github.com/CMU-SAFARI/prim-benchmarks
Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware

Year: 2021, Pages: 1-7
DOI Bookmark: 10.1109/IGSC54211.2021.9651614

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

https://www.youtube.com/watch?v=nphV36SrysA


Based on two major works

}$\text{Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware}$

Dr. Juan Gomez-Luna
The throughput saturation point is as low as \( \frac{1}{4} \) OP/B, i.e., 1 integer addition per every 32-bit element fetched.

**KEY TAKEAWAY 1**

The UPMEM PIM architecture is fundamentally compute bound. As a result, the most suitable workloads are memory-bound.
**Key Takeaway 2**

The most well-suited workloads for the UPMEM PIM architecture use no arithmetic operations or use only simple operations (e.g., bitwise operations and integer addition/subtraction).
The most well-suited workloads for the UPMEM PIM architecture require little or no communication across DPUs (inter-DPU communication).
Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, Onur Mutlu

el1goluj@gmail.com

https://github.com/CMU-SAFARI/prim-benchmarks
UPMEM PIM System Summary & Analysis

- Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu,
"Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware"

[arXiv version]
[PrIM Benchmarks Source Code]
[Slides (pptx) (pdf)]
[Talk Video (37 minutes)]
[Lightning Talk Video (3 minutes)]

Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware
## PrIM Benchmarks: Application Domains

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

- All microbenchmarks, benchmarks, and scripts
- [https://github.com/CMU-SAFARI/prim-benchmarks](https://github.com/CMU-SAFARI/prim-benchmarks)

PrIM (Processing-In-Memory Benchmarks)

PrIM is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publicly-available real-world processing-in-memory (PIM) architecture, the UPMEM PIM architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM Processing Units (DPUs), integrated in the same chip.

PrIM provides a common set of workloads to evaluate the UPMEM PIM architecture with and can be useful for programming, architecture and system researchers all alike to improve multiple aspects of future PIM hardware and software. The workloads have different characteristics, exhibiting heterogeneity in their memory access patterns, operations and data types, and communication patterns. This repository also contains baseline CPU and GPU implementations of PrIM benchmarks for comparison purposes.

PrIM also includes a set of microbenchmarks can be used to assess various architecture limits such as compute throughput and memory bandwidth.
Understanding a Modern PIM Architecture

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

JUAN GÓMEZ-LUNA¹, IZZAT EL HAJJ², IVAN FERNANDEZ¹,³, CHRISTINA GIANNOULA¹,⁴, GERALDO F. OLIVEIRA¹, AND ONUR MUTLU¹

¹ETH Zürich
²American University of Beirut
³University of Malaga
⁴National Technical University of Athens

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

https://github.com/CMU-SAFARI/prim-benchmarks
Understanding a Modern PIM Architecture

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

Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, Onur Mutlu

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

Onur Mutlu Lectures
18.7K subscribers

https://www.youtube.com/watch?v=D8Hiy2lU9I4&list=PL5Q2soXY2Z1_tOTAYm--dYByNPL7JhwR9
More on Analysis of the UPMEM PIM Engine

Inter-DPU Communication

- There is no direct communication channel between DPUs

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

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

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

Onur Mutlu Lectures
17.6K subscribers

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

https://www.youtube.com/watch?v=D8Hjy2iU9I4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9
More on Analysis of the UPMEM PIM Engine

Data Movement in Computing Systems

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

---

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

3,482 views • Premiered Jul 25, 2021

https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXYZi8_VVChACnON4sfh2bJ5JrD&index=159
ML Training on a Real PIM System

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

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

¹ETH Zürich  ²UPMEM

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

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

¹ETH Zürich  ²UPMEM

https://www.youtube.com/watch?v=qeukNs5Xl3g&t=11226s
ML Training on a Real PIM System

• Need to optimize data representation
  (1) fixed-point
  (2) quantization
  (3) hybrid precision

• Use lookup tables (LUTs) to implement complex functions (e.g., sigmoid)

• Optimize data placement & layout for streaming

• Large speedups: 2.8X/27X vs. CPU, 1.3x/3.2x vs. GPU
ML Training on Real PIM Talk Video

https://www.youtube.com/watch?v=qeukNs5XI3g&t=11226s
SpMV Multiplication on Real PIM Systems

- Appears in SIGMETRICS 2022

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

CHRISTINA GIANNOUNLA, ETH Zürich, Switzerland and National Technical University of Athens, Greece
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
NECTARIOS KOZIRIS, National Technical University of Athens, Greece
GEORGIOS GOUMAS, National Technical University of Athens, Greece
ONUR MUTLU, ETH Zürich, Switzerland

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

SAFARI: https://www.youtube.com/watch?v=5kaOsJKlGrE
SparseP
Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures

Christina Giannoula
Ivan Fernandez, Juan Gomez-Luna,
Nectarios Koziris, Georgios Goumas, Onur Mutlu
SparseP: Key Contributions

1. Efficient SpMV kernels for current & future PIM systems
   • SparseP library = 25 SpMV kernels
     • Compression, data types, data partitioning, synchronization, load balancing

SparseP is Open-Source
SparseP: https://github.com/CMU-SAFARI/SparseP

2. Comprehensive analysis of SpMV on the first commercially-available real PIM system
   • 26 sparse matrices
   • Comparisons to state-of-the-art CPU and GPU systems
   • Recommendations for software, system and hardware designers

Recommendations for Architects and Programmers
SparseP Talk Video

Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures

Christina Giannoula
Ivan Fernandez, Juan Gomez-Luna,
Nectarios Koziris, Georgios Goumas, Onur Mutlu

Processing-in-Memory Course: Lecture 11: SpMV on a Real PIM Architecture - Spring 2022

https://www.youtube.com/watch?v=5kaOsJKlGrE
Samsung Develops Industry’s First High Bandwidth Memory with AI Processing Power

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

Samsung Electronics, the world leader in advanced memory technology, today announced that it has developed the industry’s first High Bandwidth Memory (HBM) integrated with artificial intelligence (AI) processing power – the HBM-PIM. The new processing-in-memory (PIM) architecture brings powerful AI computing capabilities inside high-performance memory, to accelerate large-scale processing in data centers, high performance computing (HPC) systems and AI-enabled mobile applications.

Kwangil Park, senior vice president of Memory Product Planning at Samsung Electronics stated, “Our groundbreaking HBM-PIM is the industry’s first programmable PIM solution tailored for diverse AI-driven workloads such as HPC, training and inference. We plan to build upon this breakthrough by further collaborating with AI solution providers for even more advanced PIM-powered applications.”
Samsung Function-in-Memory DRAM (2021)

- FIMDRAM based on HBM2

![3D Chip Structure of HBM with FIMDRAM]

**Chip Specification**

- 128DQ / 8CH / 16 banks / BL4
- 32 PCU blocks (1 FIM block/2 banks)
- 1.2 TFLOPS (4H)
- FP16 ADD / Multiply (MUL) / Multiply-Accumulate (MAC) / Multiply-and-Add (MAD)

---

**ISSCC 2021 / SESSION 25 / DRAM / 25.4**

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

Young-Choon Kwon¹, Suk Han Lee¹, Jae hoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seonggil O¹, Hak-soo Yu¹, Hassen Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyoon Choi¹, Hyun-Sung Shin¹, Jin Kim¹, Beng Cheng Phuah¹, Hyo Young Min Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, Soo Young Kim¹, Eun-Bong Kim¹, David Wang¹, Shinhuan Kang¹, Yu hwan Ro¹, Seung woo Seo¹, Joo Ho Song¹, Jaryoung Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

¹Samsung Electronics, Hwasung, Korea
²Samsung Electronics, San Jose, CA
³Samsung Electronics, Suwon, Korea
Programmable Computing Unit

- Configuration of PCU block
  - Interface unit to control data flow
  - Execution unit to perform operations
  - Register group
    - 32 entries of CRF for instruction memory
    - 16 GRF for weight and accumulation
    - 16 SRF to store constants for MAC operations

[Block diagram of PCU in FIMDRAM]
### Available instruction list for FIM operation

<table>
<thead>
<tr>
<th>Type</th>
<th>CMD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Floating Point</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td></td>
<td>FP16 addition</td>
</tr>
<tr>
<td>MUL</td>
<td></td>
<td>FP16 multiplication</td>
</tr>
<tr>
<td>MAC</td>
<td></td>
<td>FP16 multiply-accumulate</td>
</tr>
<tr>
<td>MAD</td>
<td></td>
<td>FP16 multiply and add</td>
</tr>
<tr>
<td><strong>Data Path</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOVE</td>
<td></td>
<td>Load or store data</td>
</tr>
<tr>
<td>FILL</td>
<td></td>
<td>Copy data from bank to GRFs</td>
</tr>
<tr>
<td><strong>Control Path</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td>Do nothing</td>
</tr>
<tr>
<td>JUMP</td>
<td></td>
<td>Jump instruction</td>
</tr>
<tr>
<td>EXIT</td>
<td></td>
<td>Exit instruction</td>
</tr>
</tbody>
</table>

*ISSCC 2021 / SESSION 25 / DRAM / 25.4*

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

Young-Chae Ko, Seungjae Lee, Jooheon Lee, Sang-Hyuk Kim, Jae Min Ryu, Jung-Phill Son, Seung Gil Lee, Hak-Soo Yu, Haeook Lee, Soo Young Kim, Youngmin Cho, Jin Suk Kim, Jeongyeon Che, Hyeon-Sung Shin, Jin Kwon, Bongilang Phak, HyungMin Kim, Myung-Jin Sung, Min Choi, DaeHo Kim, Deok Young Kim, Eun-Bong Kim, Daewon Kang, Shihuihui Heo, Seongwook Song, Jeonho Song, Jeonjuu Yool, Kyoungh Seol, Nam Sung Kim

*Samsung Electronics, Suwon, Korea*  
*Samsung Electronics, San Jose, CA*  
*Samsung Electronics, Suwon, Korea*
Chip Implementation

- Mixed design methodology to implement FIMDRAM
  - Full-custom + Digital RTL
Samsung AxDIMM (2021)

- DDRx-PIM
  - Deep learning recommendation system

SK Hynix Develops PIM, Next-Generation AI Accelerator

February 16, 2022

Seoul, February 16, 2022

SK hynix (or "the Company", www.skhynix.com) announced on February 16 that it has developed PIM*, a next-generation memory chip with computing capabilities.

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

It has been generally accepted that memory chips store data and CPU or GPU, like human brain, process data. SK hynix, following its challenge to such notion and efforts to pursue innovation in the next-generation smart memory, has found a breakthrough solution with the development of the latest technology.

SK hynix plans to showcase its PIM development at the world’s most prestigious semiconductor conference, 2022 ISSCC**, in San Francisco at the end of this month. The company expects continued efforts for innovation of this technology to bring the memory-centric computing, in which semiconductor memory plays a central role, a step closer to the reality in devices such as smartphones.

**ISSCC: The International Solid-State Circuits Conference will be held virtually from Feb. 20 to Feb. 24 this year with a theme of “Intelligent Silicon for a Sustainable World”

For the first product that adopts the PIM technology, SK hynix has developed a sample of GDDR6-AIM (Accelerator in memory). The GDDR6-AIM adds computational functions to GDDR6 memory chips, which process data at 16Gbps. A combination of GDDR6-AIM with CPU or GPU instead of a typical DRAM makes certain computation speed 16 times faster. GDDR6-AIM is widely expected to be adopted for machine learning, high-performance computing, and big data computation and storage.

29.1 184QPS/W 64Mb/mm² 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System

Dimin Niu¹, Shuangchen Li¹, Yuhao Wang¹, Wei Han¹, Zhe Zhang², Yijin Guan², Tianchan Guan³, Fei Sun¹, Fei Xue¹, Lide Duan¹, Yuanwei Fang¹, Hongzhong Zheng¹, Xiping Jiang⁴, Song Wang⁴, Fengguo Zuo⁴, Yubing Wang⁴, Bing Yu⁴, Qiwei Ren⁴, Yuan Xie¹
Eliminating the Adoption Barriers

Processing-in-Memory in the Real World
PIM Course (Spring 2022)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory)

- **Youtube Livestream:**
  - [https://www.youtube.com/watch?v=9e4Chnwdowo&list=PL5Q2soXY2Zi841fUYYYUK9EsXKhQKRPyX](https://www.youtube.com/watch?v=9e4Chnwdowo&list=PL5Q2soXY2Zi841fUYYYUK9EsXKhQKRPyX)

- **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](https://www.youtube.com/onurmutlulectures)
Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory
Mindset: Memory as an Accelerator

Memory similar to a “conventional” accelerator
Starting Simple: Data Copy and Initialization

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

Forking

Zero initialization
(e.g., security)

Checkpointing

VM Cloning
Deduplication

Page Migration

Many more
Future Systems: In-Memory Copy

3) No cache pollution
1) Low latency
2) Low bandwidth utilization
4) No unwanted data movement

1046ns, 3.6uJ  →  90ns, 0.04uJ
RowClone: In-DRAM Row Copy

Idea: Two consecutive ACTivates

Negligible HW cost

Step 1: Activate row A

Step 2: Activate row B

DRAM subarray

Row Buffer (4 Kbytes)

4 Kbytes

Transfer row

Transfer row

8 bits

Data Bus
RowClone: Latency and Energy Savings

More on RowClone

- Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry,

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

Proceedings of the 46th International Symposium on Microarchitecture (MICRO), Davis, CA, December 2013. [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]
RowClone in Off-the-Shelf DRAM Chips

- Idea: Violate DRAM timing parameters to mimic RowClone

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

Fei Gao
feig@princeton.edu
Department of Electrical Engineering
Princeton University

Georgios Tziantzioulis
georgios.tziantzioulis@princeton.edu
Department of Electrical Engineering
Princeton University

David Wentzlaff
wentzlaf@princeton.edu
Department of Electrical Engineering
Princeton University

Real Processing Using Memory Prototype

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

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

Ataberk Olgun$\dagger$  Juan Gómez Luna$\$  Konstantinos Kanellopoulos$\$  Behzad Salami$\$$
Hasan Hassan$\$  Oğuz Ergin$\dagger$  Onur Mutlu$\$  *
$\$ETH Zürich  $\dagger$TOBB ETÜ  BSC

[Links]
- https://github.com/cmu-safari/pidram
- https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s
Real Processing Using Memory Prototype

https://github.com/cmu-safari/pidram
https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s
Building a PiDRAM Prototype

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

Rebuilding Steps

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

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

 Generating DDR3 Controller IP sources

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

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

https://github.com/cmu-safari/pidram
https://www.youtube.com/watch?v=qeuKnS5XI3g&t=4192s
In-DRAM Copy and Initialization improve throughput by 119x and 89x
PiDRAM is the first flexible end-to-end framework that enables system integration studies and evaluation of real Processing-using-Memory techniques. PiDRAM, at a high level, comprises a RISC-V system and a custom memory controller that can perform PuM operations in real DDR3 chips. This repository contains all sources required to build PiDRAM and develop its prototype on the Xilinx ZC706 FPGA boards.

https://github.com/CMU-SAFARI/PiDRAM
PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun, Juan Gómez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oğuz Ergin, Onur Mutlu

Processing-using-memory (PuM) techniques leverage the analog operation of memory cells to perform computation. Several recent works have demonstrated PuM techniques in off-the-shelf DRAM devices. Since DRAM is the dominant memory technology as main memory in current computing systems, these PuM techniques represent an opportunity for alleviating the data movement bottleneck at very low cost. However, system integration of PuM techniques imposes non-trivial challenges that are yet to be solved. Design space exploration of potential solutions to the PuM integration challenges requires appropriate tools to develop necessary hardware and software components. Unfortunately, current specialized DRAM-testing platforms, or system simulators do not provide the flexibility and/or the holistic system view that is necessary to deal with PuM integration challenges.

We design and develop PiDRAM, the first flexible end-to-end framework that enables system integration studies and evaluation of real PuM techniques. PiDRAM provides software and hardware components to rapidly integrate PuM techniques across the whole system software and hardware stack (e.g., necessary modifications in the operating system, memory controller). We implement PiDRAM on an FPGA-based platform along with an open-source RISC-V system. Using PiDRAM, we implement and evaluate two state-of-the-art PuM techniques: in-DRAM (i) copy and initialization, (ii) true random number generation. Our results show that in-memory copy and initialization techniques can improve the performance of bulk copy operations by 12.6x and bulk initialization operations by 14.6x on a real system. Implementing the true random number generator requires only 190 lines of Verilog and 74 lines of C code using PiDRAM’s software and hardware components.

Comments: 15 pages, 12 figures
Subjects: Hardware Architecture (cs.AR)
Cite as: arXiv:2111.00082 [cs.AR]
(or arXiv:2111.00082v3 [cs.AR] for this version)
https://doi.org/10.48550/arXiv.2111.00082
Long Talk + Tutorial on Youtube

https://youtu.be/s_z_S6FYpC8
(Truly) In-Memory Computation

- We can support in-DRAM AND, OR, NOT, MAJ
- At low cost
- Using analog computation capability of DRAM
  - Idea: activating multiple rows performs computation
- 30-60X performance and energy improvement

- New memory technologies enable even more opportunities
  - Memristors, resistive RAM, phase change mem, STT-MRAM, ...
  - Can operate on data with minimal movement
In-DRAM AND/OR: Triple Row Activation

Final State

\[ AB + BC + AC \]

\[ C(A + B) + \sim C(AB) \]

Bulk Bitwise Operations in Workloads

- Bitmap indices (database indexing)
- BitWeaving (database queries)
- Set operations
- Encryption algorithms
- BitFunnel (web search)
- DNA sequence mapping

[1] Li and Patel, BitWeaving, SIGMOD 2013
In-DRAM Acceleration of Database Queries

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

More on Ambit

- Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,

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

Proceedings of the 50th International Symposium on Microarchitecture (MICRO), Boston, MA, USA, October 2017.

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]

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

Vivek Seshadri\textsuperscript{1,5} Donghyuk Lee\textsuperscript{2,5} Thomas Mullins\textsuperscript{3,5} Hasan Hassan\textsuperscript{4} Amirali Boroumand\textsuperscript{5}
Jeremie Kim\textsuperscript{4,5} Michael A. Kozuch\textsuperscript{3} Onur Mutlu\textsuperscript{4,5} Phillip B. Gibbons\textsuperscript{5} Todd C. Mowry\textsuperscript{5}

\textsuperscript{1}Microsoft Research India \textsuperscript{2}NVIDIA Research \textsuperscript{3}Intel \textsuperscript{4}ETH Zürich \textsuperscript{5}Carnegie Mellon University
In-DRAM Bulk Bitwise Execution

  [Preliminary arXiv version]

In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri  
Microsoft Research India  
visesha@microsoft.com

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

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu, *SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM*


[2-page Extended Abstract]
[Short Talk Slides (pptx) (pdf)]
[Talk Slides (pptx) (pdf)]
[Short Talk Video (5 mins)]
[Full Talk Video (27 mins)]

---

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

*Nastaran Hajinazar*¹,²
Nika Mansouri Ghiasi¹

*Geraldo F. Oliveira*¹
Minesh Patel¹
Juan Gómez-Luna¹

Sven Gregorio¹
Mohammed Alser¹
Onur Mutlu¹

João Dinis Ferreira¹
Saugata Ghose³

¹ETH Zürich ²Simon Fraser University ³University of Illinois at Urbana–Champaign
SIMDRAM Framework: Overview

User Input

Desired operation

AND/OR/NOT logic

Step 1: Generate MAJ logic

MAJ

MAJ/NOT logic

Step 2: Generate sequence of DRAM commands

<table>
<thead>
<tr>
<th>ACT/PRE</th>
<th>ACT/PRE</th>
<th>ACT/PRE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>done</td>
<td></td>
</tr>
</tbody>
</table>

μProgram

SIMDRAM Output

New SIMDRAM μProgram

μProgram

Main memory

bbop_new

New SIMDRAM instruction

User Input

SIMDRAM-enabled application

foo () {
    bbop_new
}

Step 3: Execution according to μProgram

Control Unit

μProgram

Memory Controller

SIMDRAM Output

Instruction result in memory

ACT/PRE

ISA

SAFARI
SIMDRAM Key Results

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

SIMDRAM provides:

• 88× and 5.8× the throughput of a CPU and a high-end GPU, respectively, over 16 operations

• 257× and 31× the energy efficiency of a CPU and a high-end GPU, respectively, over 16 operations

• 21× and 2.1× the performance of a CPU an a high-end GPU, over seven real-world applications
More on SIMDREAM

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu,
  "SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM"
  [2-page Extended Abstract]
  [Short Talk Slides (pptx) (pdf)]
  [Talk Slides (pptx) (pdf)]
  [Short Talk Video (5 mins)]
  [Full Talk Video (27 mins)]

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

*Nastaran Hajinazar
Nika Mansouri Ghiasi

*Geraldo F. Oliveira
Minesh Patel
Juan Gómez-Luna

Sven Gregorio
Mohammed Alser
Onur Mutlu
João Dinis Ferreira
Saugata Ghose

1ETH Zürich
2Simon Fraser University
3University of Illinois at Urbana–Champaign
In-DRAM Physical Unclonable Functions

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu,
  "The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices"
  [Lightning Talk Video]
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  [Full Talk Lecture Video] (28 minutes)

The DRAM Latency PUF:
Quickly Evaluating Physical Unclonable Functions
by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim†§ Minesh Patel§ Hasan Hassan§ Onur Mutlu§†
†Carnegie Mellon University §ETH Zürich
In-DRAM True Random Number Generation

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, Lois Orosa, and Onur Mutlu, "D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput"


[Slides (pptx) (pdf)]
[Full Talk Video (21 minutes)]
[Full Talk Lecture Video (27 minutes)]

Top Picks Honorable Mention by IEEE Micro.

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

Jeremie S. Kim‡§ Minesh Patel§ Hasan Hassan§ Lois Orosa§ Onur Mutlu§†
‡Carnegie Mellon University §ETH Zürich

SAFARI
In-DRAM True Random Number Generation


[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Talk Video (25 minutes)]
[SAFARI Live Seminar Video (1 hr 26 mins)]

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

Ataberk Olgun$\dagger$, Minesh Patel$\S$, A. Giray Yağılkıçı$\S$, Haocong Luo$\S$
Jeremie S. Kim$\S$, F. Nisa Bostancı$\dagger$, Nandita Vijaykumar$\S\circ$, Oğuz Ergin$\dagger$, Onur Mutlu$\S$

$\S$ETH Zürich $\dagger$TOBB University of Economics and Technology $\circ$University of Toronto

SAFARI
In-DRAM True Random Number Generation

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

"DR-STRaNGe: End-to-End System Design for DRAM-based True Random Number Generators"
Proceedings of the 28th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, April 2022.
[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]

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

F. Nisa Bostancı†§
Jeremie S. Kim§
Ataberk Olgun†§
Hasan Hassan§
Lois Orosa§
A. Giray Yağlıkçı§
Oğuz Ergin†
Onur Mutlu§

†TOBB University of Economics and Technology §ETH Zürich

In-Flash Bulk Bitwise Execution

To appear at MICRO 2022

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

In-DRAM Lookup-Table Based Execution

- To appear at MICRO 2022

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

João Dinis Ferreira§
Lois Orosa§ ▽

Gabriel Falcao†
Mohammad Sadrosadati§
Taha Shahroodi‡

Juan Gómez-Luna§
Jeremie S. Kim§
Anant Nori*

Mohammed Alser§
Geraldo F. Oliveira§
Onur Mutlu§

‡ETH Zürich  †IT, University of Coimbra  ▽Galicia Supercomputing Center  ‡TU Delft  *Intel

Processing in Memory: Adoption Challenges

1. Processing near Memory
2. Processing using Memory
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
Challenge and Opportunity for Future

Fundamentally Energy-Efficient (Data-Centric) Computing Architectures
Challenge and Opportunity for Future

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
Fundamentally Better Architectures

Data-centric

Data-driven

Data-aware
We Need to Revisit the Entire Stack

We can get there step by step
We Need to Exploit Good Principles

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

Open minds
A Modern Primer on Processing in Memory

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

SAFARI Research Group

\textsuperscript{a}ETH Zürich
\textsuperscript{b}Carnegie Mellon University
\textsuperscript{c}University of Illinois at Urbana-Champaign
\textsuperscript{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"

\url{https://arxiv.org/pdf/1903.03988.pdf}
Funding Acknowledgments

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

Thank you!
Acknowledgments

SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

https://safari.ethz.ch
Onur Mutlu’s SAFARI Research Group

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

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

Think BIG, Aim HIGH!
SAFARI Newsletter December 2021 Edition

https://safari.ethz.ch/safari-newsletter-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/
### SAFARI Research Group at ETH Zurich and Carnegie Mellon University

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

- ETH Zurich and Carnegie Mellon University
- https://safari.ethz.ch/
- omutlu@gmail.com

#### Overview

<table>
<thead>
<tr>
<th>Repository</th>
<th>Stars</th>
<th>Forks</th>
<th>Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ramulator</strong></td>
<td>311</td>
<td>161</td>
<td>C++</td>
<td>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...</td>
</tr>
<tr>
<td><strong>prim-benchmarks</strong></td>
<td>53</td>
<td>21</td>
<td>C</td>
<td>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...</td>
</tr>
<tr>
<td><strong>DAMOV</strong></td>
<td>26</td>
<td>4</td>
<td>C++</td>
<td>DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processin...</td>
</tr>
<tr>
<td><strong>SneakySnake</strong></td>
<td>41</td>
<td>8</td>
<td>VHDL</td>
<td>SneakySnake is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude...</td>
</tr>
<tr>
<td><strong>MQSim</strong></td>
<td>146</td>
<td>93</td>
<td>C++</td>
<td>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...</td>
</tr>
</tbody>
</table>

[https://github.com/CMU-SAFARI/](https://github.com/CMU-SAFARI/)
Special Research Sessions & Courses

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

https://www.youtube.com/watch?v=qeukNs5XI3g
Comp Arch (Fall 2021)

- **Fall 2021 Edition:**
  - [https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule](https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule)

- **Fall 2020 Edition:**

- **Youtube Livestream (2021):**
  - [https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF](https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF)

- **Youtube Livestream (2020):**
  - [https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN](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](https://www.youtube.com/onurmutlulectures)
DDCA (Spring 2022)

- **Spring 2022 Edition:**

- **Spring 2021 Edition:**

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=cpXdE3HWykK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6](https://www.youtube.com/watch?v=cpXdE3HWykK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6)

- **Youtube Livestream (Spring 2021):**
  - [https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7L1N](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7L1N)

- 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](https://www.youtube.com/onurmutlulectures)
PIM Course (Spring 2022)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory

- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi841fUYUK9EsXKhQKPryX

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Processing-in-Memory lectures
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures
Genomics (Spring 2022)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics)

- **Youtube Livestream:**
  - [https://www.youtube.com/watch?v=DEL5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18](https://www.youtube.com/watch?v=DEL5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18)

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Hetero. Systems (Spring’22)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=heterogeneous_systems](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](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](https://www.youtube.com/onurmutlulectures)
HW/SW Co-Design (Spring 2022)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_co_design](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_co_design)

- **Youtube Livestream:**
  - [https://youtube.com/playlist?list=PL5Q2soXY2ZizH7un3ghD2nutKWWDk-NK](https://youtube.com/playlist?list=PL5Q2soXY2ZizH7un3ghD2nutKWWDk-NK)

- Project course
  - Taken by Bachelor’s/Master’s students
  - HW/SW co-design lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
SSD Course (Spring 2022)

- **Spring 2022 Edition:**

- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=_q4rm71DsY4&list=PL5Q2soXY2Zi8vabcse1kL22DEcgMI2RAq

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - SSD Basics and Advanced Topics
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures
Projects & Seminars: SoftMC
FPGA-Based Exploration of DRAM and RowHammer (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=softmc](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=softmc)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=r5QxuoJWttg&list=PL5Q2soXY2Zi_1trfCckr6PTN8WR72icUO](https://www.youtube.com/watch?v=r5QxuoJWttg&list=PL5Q2soXY2Zi_1trfCckr6PTN8WR72icUO)

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Projects & Seminars: Ramulator
Exploration of Emerging Memory Systems (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=aMllXRQd3s&list=PL5Q2soXY2Zi_TlmLGw_Z8hBo2925ZApqV](https://www.youtube.com/watch?v=aMllXRQd3s&list=PL5Q2soXY2Zi_TlmLGw_Z8hBo2925ZApqV)

- 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](https://www.youtube.com/onurmutlulectures)
Memory-Centric Computing

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
27 September 2022
Microsoft Azure AHA Learning Session 2022
Backup Slides
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"
[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

Storage System

Main Memory

Cache

Alignment

Computation Unit (CPU or Accelerator)

X Computation overhead

X Data movement overhead
Accelerating Genome Sequence Analysis

- Storage System
- Main Memory
- Cache
- Computation Unit (CPU or Accelerator)
- Heuristics
- Accelerators
- Filters

✓ Computation overhead

✗ Data movement overhead
**Key Idea**

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

GenStore-EM: Not Finding a Match

Sorted Read Table

<table>
<thead>
<tr>
<th>Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAA</td>
</tr>
<tr>
<td>AAAAAG</td>
</tr>
<tr>
<td>AAAACT</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Sorted K-mer Index

<table>
<thead>
<tr>
<th>K-mer</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAA</td>
</tr>
<tr>
<td>AAAAAC</td>
</tr>
<tr>
<td>AAAAAT</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Comparator

Read < K-mer

https://www.youtube.com/watch?v=kIU44FxjbFk
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"
[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
Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand

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

SAFARI  Carnegie Mellon  Google

SAMSUNG  SEOUL NATIONAL UNIVERSITY  ETH Zürich
Consumer Devices

Consumer devices are everywhere!

Energy consumption is a first-class concern in consumer devices.
Popular Consumer Workloads

Chrome
Google’s web browser

TensorFlow Mobile
Google’s machine learning framework

VP9
YouTube Video Playback
Google’s video codec

VP9
YouTube Video Capture
Google’s video codec
Energy Cost of Data Movement

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

Potential solution: move computation close to data

Challenge: limited area and energy budget
Using PIM to Reduce Data Movement

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

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

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

Chrome
Google's web browser

TensorFlow Mobile
Google's machine learning framework

VP9
Google's video codec

Video Playback

Video Capture
TensorFlow Mobile

57.3% of the inference energy is spent on data movement

54.4% of the data movement energy comes from packing/unpacking and quantization
More on PIM for Mobile Devices

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

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


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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹
Rachata Ausavarungnirun¹
Aki Kuusela³

Saugata Ghose¹
Eric Shiu³
Parthasarathy Ranganathan³

Youngsok Kim²
Rahul Thakur³

Daehyun Kim⁴,³

Onur Mutlu⁵,¹

SAFARI
Truly Distributed GPU Processing with PIM

```c
__global__
void applyScaleFactorsKernel( uint8_t * const out,
                             uint8_t const * const in,
                             const double *factor,
                             size_t const numRows, size_t const numCols )
{
    // Work out which pixel we are working on.
    const int rowIdx = blockIdx.x * blockDim.x + threadIdx.x;
    const int colIdx = blockIdx.y;
    const int sliceIdx = threadIdx.z;

    // Check this thread isn't off the image
    if( rowIdx >= numRows ) return;

    // Compute the index of my element
    size_t linearIdx = rowIdx + colIdx*numRows +
                        sliceIdx*numRows*numCols;
```
Accelerating GPU Execution with PIM (I)

[Slides (pptx) (pdf)]
[Lightning Session Slides (pptx) (pdf)]
Accelerating GPU Execution with PIM (II)


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

Ashutosh Pattnaik¹, Xulong Tang¹, Adwait Jog², Onur Kayiran³
Asit K. Mishra⁴, Mahmut T. Kandemir¹, Onur Mutlu⁵,⁶, Chita R. Das¹

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

Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,

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

Proceedings of the 34th IEEE International Conference on Computer Design (ICCD), Phoenix, AZ, USA, October 2016.

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

Kevin Hsieh†  Samira Khan‡  Nandita Vijaykumar†
Kevin K. Chang†  Amirali Boroumand†  Saugata Ghose†  Onur Mutlu§†
†Carnegie Mellon University  ‡University of Virginia  §ETH Zürich
Accelerating Dependent Cache Misses

- Milad Hashemi, Khubaib, Eiman Ebrahimi, Onur Mutlu, and Yale N. Patt, "Accelerating Dependent Cache Misses with an Enhanced Memory Controller"
  [Slides (pptx) (pdf)]
  [Lightning Session Slides (pptx) (pdf)]

Accelerating Dependent Cache Misses with an Enhanced Memory Controller

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

*The University of Texas at Austin  †Apple  ‡NVIDIA  §ETH Zürich & Carnegie Mellon University
Accelerating Runahead Execution

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

Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads

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

*The University of Texas at Austin  §ETH Zürich
Accelerating Climate Modeling

- Gagandeep Singh, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gómez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal,

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

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

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (23 minutes)]

Nominated for the Stamatis Vassiliadis Memorial Award.

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

Gagandeep Singh\textsuperscript{a,b,c} Dionysios Diamantopoulos\textsuperscript{c} Christoph Hagleitner\textsuperscript{c} Juan Gómez-Luna\textsuperscript{b}
Sander Stuijk\textsuperscript{a} Onur Mutlu\textsuperscript{b} Henk Corporaal\textsuperscript{a}
\textsuperscript{a}Eindhoven University of Technology \textsuperscript{b}ETH Zürich \textsuperscript{c}IBM Research Europe, Zurich
Accelerating Approximate String Matching

  - [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

SAFARI
Accelerating Sequence-to-Graph Mapping


[arXiv version]

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

Damla Senol Cali\textsuperscript{1} Konstantinos Kanellopoulos\textsuperscript{2} Joël Lindegger\textsuperscript{2} Züalal Bingöl\textsuperscript{3} Gurpreet S. Kalsi\textsuperscript{4} Ziyi Zuo\textsuperscript{5} Can Firtina\textsuperscript{2} Meryem Banu Cavlak\textsuperscript{2} Jeremie Kim\textsuperscript{2} Nika Mansouri Ghias\textsuperscript{2} Gagandeep Singh\textsuperscript{2} Juan Gómez-Luna\textsuperscript{2} Nour Almadhoun Alserr\textsuperscript{2} Mohammed Alser\textsuperscript{2} Sreenivas Subramoney\textsuperscript{4} Can Alkan\textsuperscript{3} Saugata Ghose\textsuperscript{6} Onur Mutlu\textsuperscript{2} 

\textsuperscript{1}Bionano Genomics \textsuperscript{2}ETH Zürich \textsuperscript{3}Bilkent University \textsuperscript{4}Intel Labs \textsuperscript{5}Carnegie Mellon University \textsuperscript{6}University of Illinois Urbana-Champaign

Accelerating Basecalling + Read Mapping

- To appear at MICRO 2022

**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 Alseri¹  Onur Mutlu¹

¹ETH Zürich  ²Bionano Genomics

Accelerating Time Series Analysis

- Ivan Fernandez, Ricardo Quislant, Christina Giannoula, Mohammed Alser, Juan Gómez-Luna, Eladio Gutiérrez, Oscar Plata, and Onur Mutlu,
  "NATSA: A Near-Data Processing Accelerator for Time Series Analysis"
  [Slides (pptx) (pdf)]
  [Talk Video (10 minutes)]
  [Source Code]

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

Ivan Fernandez§, Ricardo Quislant§, Christina Giannoula†, Mohammed Alser‡
Juan Gómez-Luna‡, Eladio Gutiérrez§, Oscar Plata§, Onur Mutlu‡

§University of Malaga  †National Technical University of Athens  ‡ETH Zürich
Accelerating Graph Pattern Mining

Maciej Besta, Raghavendra Kanakagiri, Grzegorz Kwasniewski, Rachata Ausavarungrunrun, Jakub Beránek, Konstantinos Kanellopoulos, Kacper Janda, Zur Vonarburg-Shmaria, Lukas Gianinazzi, Ioana Stefan, Juan Gómez-Luna, Marcin Copik, Lukas Kapp-Schwoerer, Salvatore Di Girolamo, Nils Blach, Marek Konieczny, Onur Mutlu, and Torsten Hoefler,
"SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems"
Proceedings of the 54th International Symposium on Microarchitecture (MICRO), Virtual, October 2021.
[Slides (pdf)]
[Talk Video (22 minutes)]
[Lightning Talk Video (1.5 minutes)]
[Full arXiv version]

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

Maciej Besta¹, Raghavendra Kanakagiri², Grzegorz Kwasniewski¹, Rachata Ausavarungrunrun³, Jakub Beránek⁴, Konstantinos Kanellopoulos¹, Kacper Janda⁵, Zur Vonarburg-Shmaria¹, Lukas Gianinazzi¹, Ioana Stefan¹, Juan Gómez-Luna¹, Marcin Copik¹, Lukas Kapp-Schwoerer¹, Salvatore Di Girolamo¹, Nils Blach¹, Marek Konieczny⁵, Onur Mutlu¹, Torsten Hoefler¹

¹ETH Zurich, Switzerland  ²IIT Tirupati, India  ³King Mongkut’s University of Technology North Bangkok, Thailand  ⁴Technical University of Ostrava, Czech Republic  ⁵AGH-UST, Poland
Accelerating HTAP Database Systems


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

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

Accelerating Neural Network Inference

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,

"Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"
Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.
[Slides (pptx) (pdf)]
[Talk Video (14 minutes)]

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

Amirali Boroumand†○
Geraldo F. Oliveira*
Saugata Ghose‡
Xiaoyu Ma§
Berkin Akin§
Eric Shiu§
Ravi Narayanaswami§
Onur Mutlu*†

†Carnegie Mellon Univ. ○Stanford Univ. ‡Univ. of Illinois Urbana-Champaign §Google *ETH Zürich

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

Amirali Boroumand  Saugata Ghose  Berkin Akin
Ravi Narayanaswami  Geraldo F. Oliveira  Xiaoyu Ma
Eric Shiu  Onur Mutlu

PACT 2021
Executive Summary

Context: We extensively analyze a state-of-the-art edge ML accelerator (Google Edge TPU) using 24 Google edge models
- Wide range of models (CNNs, LSTMs, Transducers, RCNNs)

Problem: The Edge TPU accelerator suffers from three challenges:
- It operates significantly below its peak throughput
- It operates significantly below its theoretical energy efficiency
- It inefficiently handles memory accesses

Key Insight: These shortcomings arise from the monolithic design of the Edge TPU accelerator
- The Edge TPU accelerator design does not account for layer heterogeneity

Key Mechanism: A new framework called Mensa
- Mensa consists of heterogeneous accelerators whose dataflow and hardware are specialized for specific families of layers

Key Results: We design a version of Mensa for Google edge ML models
- Mensa improves performance and energy by 3.0X and 3.1X
- Mensa reduces cost and improves area efficiency
Google Edge Neural Network Models

We analyze inference execution using 24 edge NN models

- Speech Recognition
- Face Detection
- Language Translation
- Image Captioning

Google Edge TPU

- 6 RNN Transducers
- 13 CNN
- 2 LSTMs
- 3 RCNN
Diversity Across the Models

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

- FLOP/Byte
- Parameter Footprint (MB)

Layers from CNNs and RCNNs:
- CNN3
- CNN4
- CNN11
- CNN9
- CNN13

Layers from LSTMs and Transducers:
- LSTM1

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

For example, our analysis of edge CNN models shows:

Variation in MAC intensity: up to 200x across layers

Variation in FLOP/Byte: up to 244x across layers
Mensa High-Level Overview

**Edge TPU Accelerator**

Model A  Model B  Model C

---

**Mensa**

Model A  Model B  Model C

---

Family 1  Family 2  Family 3

---

Runtime

---

Family 1

---

Family 2

---

Family 3

---

CPU

---

3D-Stacked DRAM

---

Acc. 1  Acc. 2  Acc. 3

---

Heterogeneous Accelerators

---

SAFARI
Identifying Layer Families

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

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

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

Mensa-G reduces energy consumption by 3.0X compared to the baseline Edge TPU.
Mensa: Throughput Improvement

Mensa-G improves inference throughput by 3.1X compared to the baseline Edge TPU
Mensa: Highly-Efficient ML Inference

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,

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

Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.

[Slides (pptx) (pdf)]
[Talk Video (14 minutes)]
FPGA-based Processing Near Memory


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

IBM POWER9 CPU

HBM-based FPGA board

Near-HBM FPGA-based accelerator

Two communication technologies: CAPI2 and OCAPI
Two memory technologies: DDR4 and HBM
Two workloads: Weather Modeling and Genome Analysis
Performance & Energy Greatly Improve

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

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

HBM alleviates memory bandwidth contention vs. DDR4
We Need to Revisit the Entire Stack

We can get there step by step
PIM Review and Open Problems

A Modern Primer on Processing in Memory

Onur Mutlu$^{a,b}$, 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"

SAFARI

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

Saugata Ghose† Amirali Boroumand† Jeremie S. Kim†§ Juan Gómez-Luna§ Onur Mutlu§†

†Carnegie Mellon University §ETH Zürich

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu, "Processing-in-Memory: A Workload-Driven Perspective"
[Preliminary arXiv version]
Processing in Memory: Adoption Challenges

1. Processing using Memory
2. Processing near Memory
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
Mohammed Alser, Zulal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"
[Slides (pptx)(pdf)]
[Talk Video (1 hour 2 minutes)]

Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser
ETH Zürich

Zülal Bingöl
Bilkent University

Damla Senol Cali
Carnegie Mellon University

Jeremie Kim
ETH Zurich and Carnegie Mellon University

Saugata Ghose
University of Illinois at Urbana–Champaign and Carnegie Mellon University

Can Alkan
Bilkent University

Onur Mutlu
ETH Zurich, Carnegie Mellon University, and Bilkent University
Adoption: Accelerating Key Applications (II)

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,

"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"


[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
Adoption: Accelerating Key Applications (III)

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi,
  "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
Top Picks Honorable Mention by IEEE Micro.

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

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

Seoul National University  §Oracle Labs  †Carnegie Mellon University

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

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

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

[Slides (pdf)]
[Talk Video (22 minutes)]
[Lightning Talk Video (1.5 minutes)]
[Full arXiv version]

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

Maciej Besta¹, Raghavendra Kanakagiri², Grzegorz Kwasniewski¹, Rachata Ausavarungrunr³, Jakub Beránek⁴, Konstantinos Kanellopoulos¹, Kacper Janda⁵, Zur Vonarburg-Shmaria¹, Lukas Gianinazzi¹, Ioana Stefan¹, Juan Gómez-Luna¹, Marcin Copik¹, Lukas Kapp-Schwoerer¹, Salvatore Di Girolamo¹, Nils Blach¹, Marek Konieczny⁵, Onur Mutlu¹, Torsten Hoefler¹

¹ETH Zurich, Switzerland  ²IIT Tirupati, India  ³King Mongkut’s University of Technology North Bangkok, Thailand  ⁴Technical University of Ostrava, Czech Republic  ⁵AGH-UST, Poland
Adoption: Accelerating Key Applications (V)

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,

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

Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.

[Slides (pptx) (pdf)]
[Talk Video (14 minutes)]

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

Amirali Boroumand†○
Saugata Ghose‡
Geraldo F. Oliveira*
Berkin Akin§
Ravi Narayanaswami§
Xiaoyu Ma§
Eric Shiu§
Onur Mutlu*†

†Carnegie Mellon Univ. ○Stanford Univ. ‡Univ. of Illinois Urbana-Champaign §Google *ETH Zürich
Adoption: Accelerating Key Applications (VI)

- Gagandeep Singh, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gómez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal,

"NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling"
Proceedings of the 30th International Conference on Field-Programmable Logic and Applications (FPL), Gothenburg, Sweden, September 2020.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (23 minutes)]
Nominated for the Stamatis Vassiliadis Memorial Award.

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

Gagandeep Singh\textsuperscript{a,b,c} Dionysios Diamantopoulos\textsuperscript{c} Christoph Hagleitner\textsuperscript{c} Juan Gómez-Luna\textsuperscript{b}
Sander Stuijk\textsuperscript{a} Onur Mutlu\textsuperscript{b} Henk Corporaal\textsuperscript{a}
\textsuperscript{a}Eindhoven University of Technology \textsuperscript{b}ETH Zürich \textsuperscript{c}IBM Research Europe, Zurich
Adoption: Accelerating Key Applications (VII)

- Ivan Fernandez, Ricardo Quislant, Christina Giannoula, Mohammed Alser, Juan Gómez-Luna, Eladio Gutiérrez, Oscar Plata, and Onur Mutlu,

"NATSA: A Near-Data Processing Accelerator for Time Series Analysis"
[Slides (pptx) (pdf)]
[Talk Video (10 minutes)]
[Source Code]

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

Ivan Fernandez§ Ricardo Quislant§ Christina Giannoula† Mohammed Alser‡
Juan Gómez-Luna‡ Eladio Gutiérrez§ Oscar Plata§ Onur Mutlu‡

§University of Malaga †National Technical University of Athens ‡ETH Zürich
GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis

Damla Senol Cali\textsuperscript{†} M, Gurpreet S. Kalsi\textsuperscript{M}, Züülal Bingöl\textsuperscript{v}, Can Firtina\textsuperscript{c}, Lavanya Subramanian\textsuperscript{†}, Jeremie S. Kim\textsuperscript{†}
Rachata Ausavarungnirun\textsuperscript{c}, Mohammed Alser\textsuperscript{c}, Juan Gomez-Luna\textsuperscript{c}, Amirali Boroumand\textsuperscript{†}, Anant Nori\textsuperscript{M}
Allison Scibisz\textsuperscript{†}, Sreenivas Subramoney\textsuperscript{M}, Can Alkan\textsuperscript{v}, Saugata Ghose\textsuperscript{v}, Onur Mutlu\textsuperscript{v, M}

\textsuperscript{†}Carnegie Mellon University \textsuperscript{M}Processor Architecture Research Lab, Intel Labs \textsuperscript{v}Bilkent University \textsuperscript{c}ETH Zürich
\textsuperscript{v}Facebook \textsuperscript{c}King Mongkut’s University of Technology North Bangkok \textsuperscript{c}University of Illinois at Urbana–Champaign
Adoption: Accelerating Key Applications (IX)

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhiasi, 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"

[arXiv version]

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

Damla Senol Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Züalal 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


[arXiv version]
[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
Adoption: Accelerating Key Applications (XI)

- To appear at MICRO 2022

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

Haiyu Mao\textsuperscript{1}  Mohammed Alser\textsuperscript{1}  Mohammad Sadrosadati\textsuperscript{1}  Can Firtina\textsuperscript{1}  Akanksha Baranwal\textsuperscript{1}  Damla Senol Cali\textsuperscript{2}  Aditya Manglik\textsuperscript{1}  Nour Almadhoun Alser\textsuperscript{1}  Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich  \textsuperscript{2}Bionano Genomics

Adoption: How to Keep It Simple?

  [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

SAFARI
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"


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

The graph shows the speedup of various applications under different coherence levels for Hybrid CPU-PIM architectures. The x-axis represents different applications: Components, Radii, PageRank, Components, Radii, PageRank, Components, Radii, PageRank, HTAP-256, HTAP-128, and GMean. The y-axis represents speedup, ranging from 0.00 to 2.00.

- **CPU-only**: Traditional coherence
- **FG**
- **CG**
- **NC**
- **LazyPIM**
- **Ideal-PIM**

The chart indicates the performance comparison between traditional coherence and no coherence overhead for different applications. The Ideal-PIM approach shows significant improvements in speedup compared to other coherence levels.
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"


CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators

Amirali Boroumand† Brandon Lucia†
Saugata Ghose† Rachata Ausavarungnirun†‡
Minesh Patel* Kevin Hsieh†
Hasan Hassan* Hongzhong Zheng§
Nastaran Hajinazar♦† Krishna T. Malladi§
Onur Mutlu*†

†Carnegie Mellon University
♦Simon Fraser University
§Samsung Semiconductor, Inc.
*ETH Zürich
‡KMUTNB
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"
[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?


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

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
LOIS OROSA, ETH Zürich, Switzerland
SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA
NANDITA VIJAYKUMAR, University of Toronto, Canada
IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland
MOHAMMAD SADROSADATI, Institute for Research in Fundamental Sciences (IPM), Iran & ETH Zürich, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland

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

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

When to Employ Near-Data Processing?

Mobile consumer workloads
(GoogleWL²)

Graph processing
(Tesseract¹)

Databases
(Polynesia⁵)

Time series analysis
(NATSA⁶)

Neural networks
(GoogleWL²)

DNA sequence mapping
(GenASM³; GRIM-Filter⁴)

Near-Data Processing


SAFARI
Step 1: Application Profiling

- We analyze **345 applications** from distinct domains:
  - Graph Processing
  - Deep Neural Networks
  - Physics
  - High-Performance Computing
  - Genomics
  - Machine Learning
  - Databases
  - Data Reorganization
  - Image Processing
  - Map-Reduce
  - Benchmarking
  - Linear Algebra
  ...
Step 3: Memory Bottleneck Analysis

Six classes of data movement bottlenecks:

each class ↔ 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

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. Described by Oliveira et al. (preliminary version at https://arxiv.org/pdf/2105.03725.pdf)

DAMOV-SIM

DAMOV Benchmarks

SAFARI
DAMOV is Open Source

• We open-source our benchmark suite and our toolchain

Get DAMOV at:

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

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.
More on DAMOV Analysis Methodology & Workloads

Step 3: Memory Bottleneck Classification (2/3)

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

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

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

352 views • Streamed live on Jul 22, 2021

SAFARI

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

Onur Mutlu Lectures
17.7K subscribers

https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2ZJ_tOTAYm--dYByNPL7JhwR9&index=3
More on DAMOV Methods & Benchmarks

  - [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 GÔMEZ-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