Tutorial on Memory-Centric Computing: Processing-Near-Memory

Geraldo F. Oliveira
Prof. Onur Mutlu

HEART 2024
21 June 2024
Agenda

- Introduction to Memory-Centric Computing Systems
- Real-World Processing-Near-Memory Systems
- Processing-Using-Memory Architectures for Bulk Bitwise Operations
- Lunch Break
- PIM Programming & Infrastructure for PIM Research
- Tentatively: Hands-on Lab on Programming and Understanding a Real Processing-in-Memory Architecture
Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory
When to Employ PNM

**Mobile consumer workloads**
(GoogleWL$^2$)

- **Graph processing**
  (Tesseract$^1$)

- **Databases**
  (Polynesia$^5$)

- **Neural networks**
  (GoogleWL$^2$)

- **Time series analysis**
  (NATSA$^6$)

- **DNA sequence mapping**
  (GenASM$^3$; GRIM-Filter$^4$)

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
  
<table>
<thead>
<tr>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

  +42% Speedup

  32 Cores
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

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

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

Host Processor

Memory-Mapped Accelerator Interface
Noncacheable, Physically Addressed)

Memory
Logic

Crossbar Network

In-Order Core

LP
PF Buffer
MTP

Message Queue

DRAM Controller

NI

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)
- Memory
- Logic
- Crossbar Network
- In-Order Core
- DRAM Controller
- NI
- Communications via Remote Function Calls
- Message Queue

SAFARI
Tesseract System for Graph Processing

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

Crossbar Network

Memory

Logic

Prefetching
LP
PF Buffer
MTP
Message Queue

DRAM Controller
NI

SAFARI
Evaluated Systems

**DDR3-OoO**
- 8 OoO 4GHz
- 102.4GB/s

**HMC-OoO**
- 8 OoO 4GHz
- 8 OoO 4GHz
- 128 In-Order 2GHz
- 640GB/s

**HMC-MC**
- 8 OoO 4GHz
- 8 OoO 4GHz
- 128 In-Order 2GHz
- 640GB/s

**Tesseract**
- 32 Tesseract Cores
- 8TB/s

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: 0x
- HMC-OoO: +56%
- HMC-MC: +25%
- Tesseract: 9.0x
- Tesseract-LP: 11.6x
- Tesseract-LP-MTP: 13.8x

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

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

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

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

Junwhan Ahn  Sungpack Hong§  Sungjoo Yoo  Onur Mutlu†  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
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

SAFARI

15
Genome Sequence Analysis

Data Movement from Storage

Storage System

Main Memory

Cache

Computation Unit (CPU or Accelerator)

Alignment

Computation overhead

Data movement overhead

SAFARI
Compute-Centric Accelerators

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

Heuristics

✓ Computation overhead

✗ Data movement overhead

SAFARI
Key Idea: In-Storage Filtering

Filter reads that do not require alignment inside the storage system

Filtered Reads

Exactly-matching reads
Do not need expensive approximate string matching during alignment

Non-matching reads
Do not have potential matching locations and can skip alignment
GenStore

Filter reads that do not require alignment inside the storage system

GenStore-Enabled Storage System

Main Memory

Cache

Computation Unit (CPU or Accelerator)

✓ Computation overhead

✓ Data movement overhead

GenStore provides significant speedup (1.4x - 33.6x) and energy reduction (3.9x – 29.2x) at low cost
In-Storage Genomic Data Filtering [ASPLOS 2022]

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

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

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

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

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

\textsuperscript{1}ETH Zürich \textsuperscript{2}Bionano Genomics \textsuperscript{3}KMUTNB \textsuperscript{4}University of Toronto
Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

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

Consumer devices are everywhere!

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

Chrome
Google’s web browser

TensorFlow Mobile
Google’s machine learning framework

VP9
Video Playback
Google’s video codec

VP9
Video Capture
Google’s video codec
**1st key observation:** 62.7% of the total system energy is spent on **data movement**

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

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

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

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

Small embedded low-power core

Small fixed-function accelerators

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

Chrome
Google’s web browser

TensorFlow Mobile
Google’s machine learning framework

VP9
YouTube Video Playback
Google’s video codec

VP9
YouTube Video Capture
Google’s video codec
57.3% of the inference energy is spent on **data movement**

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

<table>
<thead>
<tr>
<th></th>
<th>CPU-Only</th>
<th>PIM-Core</th>
<th>PIM-Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture Tiling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color Blitting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decompression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub-Pixel Interpolation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deblocking Filter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motion Estimation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PIM core and PIM accelerator reduce energy consumption on average by 49.1% and 55.4%
Offloading these kernels to **PIM core and PIM accelerator** reduces **program runtime** on average by **44.6%** and **54.2%**
More on PIM for Mobile Devices

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

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


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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand\textsuperscript{1} Rachata Ausavarungnirun\textsuperscript{1} Aki Kuusela\textsuperscript{3} Allan Knies\textsuperscript{3}  
Saugata Ghose\textsuperscript{1} Eric Shiu\textsuperscript{3}  
Youngsok Kim\textsuperscript{2} Rahul Thakur\textsuperscript{3}  
Daehyun Kim\textsuperscript{4,3} Parthasarathy Ranganathan\textsuperscript{3}  
Onur Mutlu\textsuperscript{5,1}

SAFARI

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

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

Kevin Hsieh† Eiman Ebrahimi† Gwangsun Kim* Niladrish Chatterjee† Mike O’Connor†
Nandita Vijaykumar‡ Onur Mutlu§‡ Stephen W. Keckler†

‡Carnegie Mellon University †NVIDIA *KAIST §ETH Zürich
Scheduling Techniques for GPU Architectures with Processing-In-Memory Capabilities

Ashutosh Pattnaik\textsuperscript{1} Xulong Tang\textsuperscript{1} Adwait Jog\textsuperscript{2} Onur Kayiran\textsuperscript{3}
Asit K. Mishra\textsuperscript{4} Mahmut T. Kandemir\textsuperscript{1} Onur Mutlu\textsuperscript{5,6} Chita R. Das\textsuperscript{1}

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

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

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

Kevin Hsieh†  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"
  Seoul, South Korea, June 2016.
  [Slides (pptx) (pdf)]
  [Lightning Session Slides (pptx) (pdf)]

Accelerating Dependent Cache Misses with an Enhanced Memory Controller

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

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

SAFARI
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 (p-px) (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
GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis

Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

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

[Lighting Talk Video (1.5 minutes)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
[Slides (pptx) (pdf)]

SAFARI
Accelerating Sequence-to-Graph Mapping

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

\(^1\)Bionano Genomics  \(^2\)ETH Zürich  \(^3\)Bilkent University  \(^4\)Intel Labs
\(^5\)Carnegie Mellon University  \(^6\)University of Illinois Urbana-Champaign

Accelerating Basecalling + Read Mapping

- Haiyu Mao, Mohammed Alser, Mohammad Sadrosadati, Can Firtina, Akanksha Baranwal, Damla Senol Cali, Aditya Manglik, Nour Almadhoun Alserr, and Onur Mutlu,

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

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

[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Lecture Video (25 minutes)]
[arXiv version]

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

Haiyu Mao¹  Mohammed Alser¹  Mohammad Sadrosadati¹  Can Firtina¹  Akanksha Baranwal¹
Damla Senol Cali²  Aditya Manglik¹  Nour Almadhoun Alserr¹  Onur Mutlu¹

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 Ausavarungrun, 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 Ausavarungrun³, 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

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

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

Amirali Boroumand\textsuperscript{†,○}  Saugata Ghose\textsuperscript{‡}  Berkin Akin\textsuperscript{§}  Ravi Narayanaswami\textsuperscript{§}
Geraldo F. Oliveira\textsuperscript{*}  Xiaoyu Ma\textsuperscript{§}  Eric Shiu\textsuperscript{§}  Onur Mutlu\textsuperscript{*†}
\textsuperscript{†}Carnegie Mellon Univ.  \textsuperscript{○}Stanford Univ.  \textsuperscript{‡}Univ. of Illinois Urbana-Champaign  \textsuperscript{§}Google  \textsuperscript{*}ETH Zürich
Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

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

PACT 2021

SAFARI

Carnegie Mellon  University of Illinois  Google  ETH Zürich
Executive Summary

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

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

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

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

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

We analyze inference execution using 24 edge NN models

- 6 RNN Transducers
- 13 CNN
- 2 LSTMs
- 3 RCNN

Google Edge TPU

Speech Recognition
Face Detection
Language Translation
Image Captioning
Insight 1: there is significant variation in terms of layer characteristics across the models.
Diversity Within the Models

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

For example, our analysis of edge CNN models shows:

- Variation in **MAC intensity:** up to 200x across layers.
- Variation in **FLOP/Byte:** up to 244x across layers.

**MACs (M)**

**FLOP/Byte**
Mensa High-Level Overview

**Edge TPU Accelerator**

- Model A
- Model B
- Model C

![Diagram of Edge TPU Accelerator](image)

**Mensa**

- Model A
- Model B
- Model C

![Diagram of Mensa Accelerators](image)

**Heterogeneous Accelerators**

- Acc. 1
- Acc. 2
- Acc. 3

- CPU
- 3D-Stacked DRAM
- NoC
- Buffer

**Monolithic Accelerator**

![Diagram of Monolithic Accelerator](image)
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

SAFARI
Mensa: Throughput Improvement

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

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

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

Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.
[Slides (pptx) (pdf)]
[Talk Video (14 minutes)]

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

Amirali Boroumand\textsuperscript{†,○} \quad Saugata Ghose\textsuperscript{‡} \quad Berkin Akin\textsuperscript{§} \quad Ravi Narayanaswami\textsuperscript{§}
Geraldo F. Oliveira\textsuperscript{*} \quad Xiaoyu Ma\textsuperscript{§} \quad Eric Shiu\textsuperscript{§} \quad Onur Mutlu\textsuperscript{*†}

\textsuperscript{†}Carnegie Mellon Univ. \quad \textsuperscript{○}Stanford Univ. \quad \textsuperscript{‡}Univ. of Illinois Urbana-Champaign \quad \textsuperscript{§}Google \quad \textsuperscript{*}ETH Zürich
Accelerating Data-Intensive Workloads

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoungh Choi, "PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"
  [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
FPGA-based Processing Near Memory


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

Gagandeep Singh\(\Diamond\)  Mohammed Alser\(\Diamond\)  Damla Senol Cali\(\clubsuit\)
Dionysis Diamantopoulos\(\checkmark\)  Juan Gómez-Luna\(\Diamond\)
Henk Corporaal\(\star\)  Onur Mutlu\(\clubsuit\)

\(\Diamond\)ETH Zürich  \(\checkmark\)Carnegie Mellon University
\(\star\)Eindhoven University of Technology  \(\checkmark\)IBM Research Europe

SAFARI
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
UPMEM Processing-in-DRAM Engine (2019)

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

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

UPMEM Memory Modules

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

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

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
IZZAT EL HAJJ, American University of Beirut, Lebanon
IVAN FERNÁNDEZ, ETH Zürich 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 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 UPHENM company has designed and manufactured the first publicly available real-world PIM architecture. The UPHENM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM-MoSoNing Chips (DPU), integrated in the same die.

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 UPHENM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present PIM benchmark suite of 6 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 PIM benchmarks on the UPHENM 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 UPHENM-based PIM systems with 140 and 2,560 DPU provides new insights about the 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
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 GIANNELLA¹,⁴, 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
## 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.
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
SpMV Multiplication on Real PIM Systems

- Appears in SIGMETRICS 2022

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

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

[https://github.com/CMU-SAFARI/SparseP](https://github.com/CMU-SAFARI/SparseP)

**SAFARI**
[https://www.youtube.com/watch?v=5kaOsJKIGrE](https://www.youtube.com/watch?v=5kaOsJKIGrE)
A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab 1 Amir Nassereldine 1 Mohammed Alser 2 Juan Gómez Luna 2
Onur Mutlu 2 Izzat El Hajj 1

1American University of Beirut 2ETH Zürich

https://github.com/CMU-SAFARI/alignment-in-memory

Summary

- Sequence alignment on traditional systems is limited by the **memory bandwidth bottleneck**.

- **Processing-in-memory (PIM)** overcomes this bottleneck by placing cores near the memory.

- Our framework, **Alignment-in-Memory (AIM)**, is a PIM framework that supports multiple alignment algorithms (NW, SWG, GenASM, WFA).
  - Implemented on UPMEM, the first real PIM system.

- Results show **substantial speedups over both CPUs (1.8X-28X) and GPUs (1.2X-2.7X)**.

- AIM is available at:
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

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)

[3D Chip Structure of HBM with FIMDRAM]
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
## Available instruction list for FIM operation

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

- Mixed design methodology to implement FIMDRAM
  - Full-custom + Digital RTL

TPV & Peri Control Block

[TSV & Peri Control Block diagram]

[Digital RTL design for PCU block]
Main target: transformer decoders used in ChatGPT, GPT-3
- Compute-bound step: Summarization
- Memory-bound step: Generation
  - Most of the execution time is spent on the memory copy from the host CPU memory to the CPU memory

GEMV portion can be 60%-80% of total generation latency, which is the target of PIM/PNM
Solution I: Samsung’s HBM-PIM (2023)

- **AMD MI100 GPUs** fabricated with **HBM-PIM**

- Experimental setup: GPT-J (6B, 32 input tokens), single AMD MI100-PIM GPU

- GPT can be accelerated by more than 2x over baseline

Solution II: Samsung’s LPDDR-PIM (2023)

- PIM for on-device generative AI
  - Datacenter costs and power consumption are increasing due to the growing demand for cloud AI

- LPDDR-PIM improves battery life by preventing memory over-provisioning just for bandwidth

- 4.47x performance gains and 70.6% energy reduction in GPT-2

Solution III: Samsung’s CXL-PNM (2023)

- A CXL-based processing-near-memory solution
  - Improves capacity, bandwidth, and power
  - Large-scale large-language models are often capacity-bound

Multiple CXL-PNM can offer 4.4x higher energy efficiency and 53% higher throughput than multiple GPUs

Samsung AxDIMM (2021)

- DDRx-PIM
  - Deep learning recommendation system

SK Hynix AiM: Chip Implementation (2022)

- 4 Gb AiM die with 16 processing units (PUs)

AiM Die Photograph

1 Process Unit (PU) Area

<table>
<thead>
<tr>
<th>Component</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>0.19mm²</td>
</tr>
<tr>
<td>MAC</td>
<td>0.11mm²</td>
</tr>
<tr>
<td>Activation Function (AF)</td>
<td>0.02mm²</td>
</tr>
<tr>
<td>Reservoir Cap.</td>
<td>0.05mm²</td>
</tr>
<tr>
<td>Etc.</td>
<td>0.01mm²</td>
</tr>
</tbody>
</table>

MAC 58%

Reservoir Cap. 26%

AF 11%

Etc. 5%
SK Hynix AiM: System Organization (2022)

- GDDR6-based AiM architecture

Lee et al., A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications, ISSCC 2022
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
Tutorial on Memory-Centric Computing: Processing-Near-Memory

Geraldo F. Oliveira
Prof. Onur Mutlu

HEART 2024
21 June 2024