Reinventing Computing & Storage

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
7 April 2023
OlympusMons Award Keynote Speech
The Problem

Computing is Bottlenecked by Data
Data is Key for AI, ML, Genomics, …

- Important workloads are all data intensive

- They require rapid and efficient processing of large amounts of data

- Data is increasing
  - We can generate more than we can process
  - We need to perform more sophisticated analyses on more data
Huge Demand for Performance & Efficiency

Exponential Growth of Neural Networks

1800x more compute
In just 2 years

Tomorrow, multi-trillion parameter models

Source: https://youtu.be/Bh13Idwcb0Q?t=283
Data is Key for Future Workloads

In-memory Databases
[Mao+, 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

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

Graph/Tree Processing

Datacenter Workloads
[Kanev+ (Google), ISCA’15]

Data → performance & energy bottleneck
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

Video Capture
Google’s video codec
Data Overwhelms Modern Machines

Chrome

TensorFlow Mobile

Data → performance & energy bottleneck

VP9

Video Playback

Google’s video codec

VP9

Video Capture

Google’s video codec
Data is Key for Future Workloads

Development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

Source: Illumina

1 Sequencing

Genome Analysis

2 Read Mapping

Data → performance & energy bottleneck

3 Variant Calling

Scientific Discovery

4

Read4: CGCTTCCT
read5: CCATGACGC
read6: TTCCATGAC

Billions of Short Reads

Read Alignment

Reference Genome
We Need Faster & Scalable Genome Analysis

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

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

Rapid surveillance of **disease outbreaks**

Developing **personalized medicine**

And, many, many other applications ...
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]
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 ▼

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

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

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

**FAST**

**SLOW**

Slow and inefficient processing capability
Large amounts of data movement
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)]
Beginner Reading on Genome Analysis

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

“From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”
Computational and Structural Biotechnology Journal, 2022

[Source code]

Review

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

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

ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland

FPGA-based Near-Memory Analytics


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

Gagandeep Singh\(^{\diamond}\)  Mohammed Alser\(^{\diamond}\)  Damla Senol Cali\(^{\times}\)
Dionysios Diamantopoulos\(^{\nabla}\)  Juan Gómez-Luna\(^{\diamond}\)
Henk Corporaal\(^{*}\)  Onur Mutlu\(^{\diamond\times}\)

\(^{\diamond}\)ETH Zürich  \(^{\times}\)Carnegie Mellon University
\(^{*}\)Eindhoven University of Technology  \(^{\nabla}\)IBM Research Europe
Near-Memory Acceleration using FPGAs

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
GenASMFramewo [MICRO 2020]


[Lighting Talk Video (1.5 minutes)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
[Slides (pptx) (pdf)]
In-Storage Genome Filtering [ASPLOS 2022]

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

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


[Talk Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Talk Video (17 minutes)]

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

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

¹ETH Zürich  ²Bionano Genomics  ³KMUTNB  ⁴University of Toronto
Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, 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\textsuperscript{1}  Konstantinos Kanellopoulos\textsuperscript{2}  Joël Lindegger\textsuperscript{2}  Zülal 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 Ghiasi\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

- 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

A Framework for Designing Efficient Deep Learning-Based Genomic Basecallers

Gagandeep Singh\textsuperscript{a} Mohammed Alser\textsuperscript{*a} Alireza Khodamoradi\textsuperscript{*b}
Kristof Denolf\textsuperscript{b} Can Firtina\textsuperscript{a} Meryem Banu Cavlak\textsuperscript{a}
Henk Corporaal\textsuperscript{c} Onur Mutlu\textsuperscript{a}
\textsuperscript{a}ETH Zürich \textsuperscript{b}AMD \textsuperscript{c}Eindhoven University of Technology

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

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


MinION from ONT

Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

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

Accelerating Genome Analysis
A Primer on an Ongoing Journey

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

https://www.youtube.com/watch?v=NCagwf0ivT0
Detailed Lectures on Genome Analysis

- Computer Architecture, Fall 2020, Lecture 3a
  - Introduction to Genome Sequence Analysis (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5

- Computer Architecture, Fall 2020, Lecture 8
  - Intelligent Genome Analysis (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14

- Computer Architecture, Fall 2020, Lecture 9a
  - GenASM: Approx. String Matching Accelerator (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=XoLpzmNPas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15

- Accelerating Genomics Project Course, Fall 2020, Lecture 1
  - Accelerating Genomics (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCgLgwiDRQDTyId

SAFARI

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

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

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

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV](https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NnPdOR1yRU_Cxxjw-u18](https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NnPdOR1yRU_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)
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.
Most of the system is dedicated to storing and moving data

Yet, system is still bottlenecked by memory & storage
Deeper and Larger Memory Hierarchies

Core Count:
8 cores/16 threads

L1 Caches:
32 KB per core

L2 Caches:
512 KB per core

L3 Cache:
32 MB shared

AMD Ryzen 5000, 2020

AMD’s 3D Last Level Cache (2021)

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

https://youtu.be/gqAYMx34euU
https://www.tech-critter.com/amd-keynote-computex-2021/
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)

https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php
Data Overwhelms Modern Machines

Chrome

TensorFlow Mobile

Data → performance & energy bottleneck

Video Playback

Video Capture

VP9

Google’s video codec

Google’s video codec
62.7% of the total system energy is spent on data movement

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand1 Rachata Ausavarungnirun1 Saugata Ghose1 Youngsok Kim2 Eric Shiu3 Rahul Thakur3 Parthasarathy Ranganathan3 Daehyun Kim4,3 Onur Mutlu5,1

SAFARI
Data Movement Overwhelms Accelerators

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

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

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

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

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

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

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

†Carnegie Mellon Univ.  ‡Stanford Univ.  ‡Univ. of Illinois Urbana-Champaign  §Google  *ETH Zürich
An Intelligent Architecture
Handles Data Well
How to Handle Data Well

- **Ensure data does not overwhelm the components**
  - via intelligent algorithms, architectures & system designs: algorithm-architecture-devices

- **Take advantage of vast amounts of data and metadata**
  - to improve architectural & system-level decisions

- **Understand and exploit properties of (different) data**
  - to improve algorithms & architectures in various metrics
Corollaries: Computing Systems Today …

- Are processor-centric vs. data-centric

- Make designer-dictated decisions vs. data-driven

- Make component-based myopic decisions vs. data-aware
Fundamentally Better Architectures

Data-centric

Data-driven

Data-aware
We Need to Revisit the Entire Stack

We can get there step by step
A Blueprint for Fundamentally Better Architectures

  - Slides (pptx) (pdf)
  - IEDM Tutorial Slides (pptx) (pdf)
  - Short DATE Talk Video (11 minutes)
  - Longer IEDM Tutorial Video (1 hr 51 minutes)

Intelligent Architectures for Intelligent Computing Systems

Onur Mutlu
ETH Zurich
omutlu@gmail.com
Data-Driven (Self-Optimizing) Architectures
System Architecture Design Today

- Human-driven
  - Humans design the policies (how to do things)

- Many (too) simple, short-sighted policies all over the system

- No automatic data-driven policy learning

- (Almost) no learning: cannot take lessons from past actions

Can we design fundamentally intelligent architectures?
An Intelligent Architecture

- Data-driven
  - Machine learns the “best” policies (how to do things)

- Sophisticated, workload-driven, changing, far-sighted policies

- Automatic data-driven policy learning

- All controllers are intelligent data-driven agents

We need to rethink design (of all controllers)
Self-Optimizing Memory Controllers


Self-Optimizing Memory Controllers: A Reinforcement Learning Approach

Engin İpek\textsuperscript{1,2} Onur Mutlu\textsuperscript{2} José F. Martínez\textsuperscript{1} Rich Caruana\textsuperscript{1}

\textsuperscript{1}Cornell University, Ithaca, NY 14850 USA
\textsuperscript{2}Microsoft Research, Redmond, WA 98052 USA
Self-Optimizing Memory Prefetchers

Rahul Bera, Konstantinos Kanellopoulos, Anant Nori, Taha Shahroodi, Sreenivas Subramoney, and Onur Mutlu,
"Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning"
Proceedings of the 54th International Symposium on Microarchitecture (MICRO), Virtual, October 2021.
[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (20 minutes)]
[Lightning Talk Video (1.5 minutes)]
[Pythia Source Code (Officially Artifact Evaluated with All Badges)]
[arXiv version]
Officially artifact evaluated as available, reusable and reproducible.

Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera\textsuperscript{1}  Konstantinos Kanellopoulos\textsuperscript{1}  Anant V. Nori\textsuperscript{2}  Taha Shahroodi\textsuperscript{3,1}
Sreenivas Subramoney\textsuperscript{2}  Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich  \textsuperscript{2}Processor Architecture Research Labs, Intel Labs  \textsuperscript{3}TU Delft

Learning-Based Off-Chip Load Predictors

- Rahul Bera, Konstantinos Kanellopoulos, Shankar Balachandran, David Novo, Ataberk Olgun, Mohammad Sadrosadati, and Onur Mutlu,

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

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

[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Talk Video (12 minutes)]
[Lecture Video (25 minutes)]
[arXiv version]
[Source Code (Officially Artifact Evaluated with All Badges)]

Officially artifact evaluated as available, reusable and reproducible. Best paper award at MICRO 2022.

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

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

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

Self-Optimizing Hybrid SSD Controllers

Gagandeep Singh, Rakesh Nadig, Jisung Park, Rahul Bera, Nastaran Hajinazar, David Novo, Juan Gomez-Luna, Sander Stuijk, Henk Corporaal, and Onur Mutlu,
*Sibyl: Adaptive and Extensible Data Placement in Hybrid Storage Systems Using Online Reinforcement Learning*
[Slides (pptx) (pdf)]
[arXiv version]
[Sibyl Source Code]
[Talk Video (16 minutes)]

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

Gagandeep Singh¹  Rakesh Nadig¹  Jisung Park¹  Rahul Bera¹  Nastaran Hajinazar¹
David Novo³  Juan Gómez-Luna¹  Sander Stuijk²  Henk Corporaal²  Onur Mutlu¹

¹ETH Zürich  ²Eindhoven University of Technology  ³LIRMM, Univ. Montpellier, CNRS

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

Gagandeep Singh, Rakesh Nadig, Jisung Park, Rahul Bera, Nastaran Hajinazar, David Novo, Juan Gómez Luna, Sander Stuijk, Henk Corporaal, Onur Mutlu
Hybrid Storage System Basics

Address Space (Application/File System View)

Logical Pages

Read

Write

Storage Management Layer

Read

Write

Read

Write

Promotion

Eviction

Fast Device

Slow Device

Hybrid Storage System
Hybrid Storage System Basics

Performance of a hybrid storage system highly depends on the ability of the storage management layer.
Key Shortcomings in Prior Techniques

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

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

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

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

1. **Adaptivity**, by continuously learning and adapting to the application and underlying device characteristics

2. **Easy extensibility** to incorporate a wide range of hybrid storage configurations
Our Proposal

Sibyl

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

Sybil is an oracle that makes accurate prophecies
https://en.wikipedia.org/wiki/Sibyl
Basics of Reinforcement Learning (RL)

Agent learns to take an **action** in a given **state** to maximize a numerical **reward**
Formulating Data Placement as RL

Agent

State ($S_t$)

Reward ($R_{t+1}$)

Action ($A_t$)

Environment

Sibyl

Features of the current request and system

Request latency (of last served request)

Select storage device to place the current page

Hybrid Storage System

Features of the current request and system

Request latency (of last served request)

Select storage device to place the current page
Sibyl Execution

- RL Training Thread
  - Periodic Policy Weight Update
  - State, Reward, and Action Information

- RL Decision Thread
  - Data Placement Decision
  - Storage Request (from OS)

Asynchronous Execution

Sibyl
Sibyl Design: Overview

- **Training Network**
  - Periodic Policy Weight Update

- **Inference Network**
  - Max

- **Sibyl Policy**
  - Observation Vector
  - State

- **Experience Buffer** (in host DRAM)
  - Collect Experiences
  - HSS
  - Reward

- **Training Dataset**
  - Batch

- **RL Training Thread**
  - Periodic Policy Weight Update

- **RL Decision Thread**
  - Observation Vector
  - State
  - Action

- **Storage Request** (from OS)
Evaluation Methodology (1/3)

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

High-end SSD | Low-end HDD

Performance-Oriented HSS Configuration

High-end SSD | Middle-end SSD
Evaluation Methodology (3/3)

• **18 different workloads** from:
  - MSR Cambridge and Filebench Suites

• **Four** state-of-the-art data placement baselines:
  - HPS [Meswani+, HPCA’15]
  - Archivist [Ren+, ICCD’19] **Learning-based**
  - RNN-HSS [Doudali+, HPDC’19]
Performance Analysis

Cost-Oriented HSS Configuration

Normalized Average Request Latency

- Slow-Only
- CDE
- HPS
- Archivist
- RNN-HSS
- Sibyl
- Oracle

66
Performance Analysis

Cost-Oriented HSS Configuration

Sibyl consistently outperforms all the baselines for all the workloads
Performance Analysis

Sibyl achieves **80% of the performance** of an oracle policy that has complete knowledge of future access patterns.
Performance on Tri-HSS

Extending Sibyl for more devices:
1. Add a new action

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

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

• We introduced Sibyl, the first reinforcement learning-based data placement technique in hybrid storage systems that provides
  - Adaptivity
  - Easily extensibility
  - Ease of design and implementation

• We evaluated Sibyl on real systems using many different workloads
  - In a tri-HSS configuration, Sibyl outperforms the state-of-the-art data placement policy by 48.2%
  - Sibyl achieves 80% of the performance of an oracle policy with a storage overhead of only 124.4 KiB

[GitHub Link: https://github.com/CMU-SAFARI/Sibyl]
Challenge and Opportunity for Future Data-Driven (Self-Optimizing) Computing Architectures
Data-Characteristic-Aware Architectures
Data-Aware Architectures

- A data-aware architecture understands what it can do with and to each piece of data.

- It makes use of different properties of data to improve performance, efficiency and other metrics:
  - Compressibility
  - Approximability
  - Locality
  - Sparsity
  - Criticality for Computation
  - Access Semantics
  - ...

One Problem: Limited Expressiveness

Higher-level information is not visible to HW

Software

Hardware

Data Structures

Code Optimizations

Access Patterns

Data Type

Integer

Float

Char

Instructions

Memory Addresses

100011111...

101010011...
A Solution: More Expressive Interfaces
Expressive (Memory) Interfaces


[Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video]
Expressive (Memory) Interfaces for GPUs


---

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

Nandita Vijaykumar†§ Eiman Ebrahimi‡ Kevin Hsieh†
Phillip B. Gibbons† Onur Mutlu§†
†Carnegie Mellon University ‡NVIDIA §ETH Zürich
[arXiv version]
Presented at the 18th HiPEAC Conference, Toulouse, France, January 2023.
[Slides (pptx) (pdf)]
[Preliminary Talk Video (14 minutes)]
[SAFARI Live Seminar Video (1 hour 26 minutes)]
[MetaSys Source Code]

Best paper award at HiPEAC 2023.

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

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

*University of Toronto §ETH Zürich †Carnegie Mellon University
Heterogeneous-Reliability Memory

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

"Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory"

Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, GA, June 2014. [Summary]

[Slides (pptx) (pdf)] [Coverage on ZDNet]
EDEN: Data-Aware Efficient DNN Inference

- Skanda Koppula, Lois Orosa, A. Giray Yaglikci, Roknoddin Azizi, Taha Shahroodi, Konstantinos Kanellopoulos, and Onur Mutlu,

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

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

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Poster (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Full Talk Lecture (38 minutes)]

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

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

ETH Zürich
SMASH: SW/HW Indexing Acceleration

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

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

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

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Poster (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Full Talk Lecture (30 minutes)]

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

Konstantinos Kanellopoulos¹ Nandita Vijaykumar²,¹ Christina Giannoula¹,³ Roknoddin Azizi¹ Skanda Koppula¹ Nika Mansouri Ghiasi¹ Taha Shahroodi¹ Juan Gomez Luna¹ Onur Mutlu¹,²

¹ETH Zürich ²Carnegie Mellon University ³National Technical University of Athens
Rethinking Virtual Memory

- Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarunngirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,

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


[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[ARM Research Summit Poster (pptx) (pdf)]
[Talk Video (26 minutes)]
[Lightning Talk Video (3 minutes)]

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

Nastaran Hajinazar*† Pratyush Patel‡ Minesh Patel* Konstantinos Kanellopoulos* Saugata Ghose† Rachata Ausavarunngirun© Geraldo F. Oliveira* Jonathan Appavoo◊ Vivek Seshadri▼ Onur Mutlu*‡

*ETH Zürich †Simon Fraser University ‡University of Washington ▼Carnegie Mellon University
©King Mongkut’s University of Technology North Bangkok ◊Boston University ▼Microsoft Research India
Data-Characteristic-Aware Computing Architectures
Data-Centric (Memory-Centric) Architectures
Data-Centric Architectures: Properties

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

- **Low-latency & low-energy data access**

- **Low-cost data storage & processing**
  - High capacity memory at low cost: hybrid memory, compression

- **Intelligent data management**
  - Intelligent controllers handling robustness, security, cost, perf.
Processing Data
Where It Makes Sense
Process Data Where It Makes Sense

Apple M1 Ultra System (2022)

https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php
Processing in/near 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.

CELL EQUATIONS: \( \hat{x} = \bar{w}x + wy \)
\( s_y = wcx, r_y = wc\bar{x} \)
\( \hat{z} = M(x, y, z) = x\bar{y} + z(x + \bar{y}) \)

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?

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

- **Huge demand from Applications & Systems**
  - Data access bottleneck
  - Energy & power bottlenecks
  - Data movement energy dominates computation energy
  - Need all at the same time: performance, energy, sustainability
  - We can improve all metrics by minimizing data movement

- **Designs are squeezed in the middle**
Three Key Systems & Application Trends

1. Data access is the major bottleneck
   - Applications are increasingly data hungry

2. Energy consumption is a key limiter

3. Data movement energy dominates compute
   - Especially true for off-chip to on-chip movement
Do We Want This?
Or This?

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 $\Rightarrow$ at great system cost
Energy Waste in Mobile Devices

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

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

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

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} Rahul Thakur\textsuperscript{3} Parthasarathy Ranganathan\textsuperscript{3} 
Youngsok Kim\textsuperscript{2} 
Daehyun Kim\textsuperscript{4,3} Onur Mutlu\textsuperscript{5,1}

SAFARI
Energy Waste in Accelerators

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

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

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

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

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

SAFARI
We Do Not Want to Move Data!

Communication Dominates Arithmetic

A memory access consumes $\sim 100$-1000X the energy of a complex addition.
We Need A **Paradigm Shift** To …

- Enable computation with **minimal data movement**

- **Compute where it makes sense** *(where data resides)*

- Make computing architectures more **data-centric**
Goal: Processing Inside Memory/Storage

Many questions ... How do we design the:

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

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

Abstract

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

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

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

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

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

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

The cost of data movement is a fundamental issue with the processor-centric nature of contemporary computer systems. The CPU is considered to be the master in the system, and computation is performed only in the processor (and accelerators). In contrast, data storage and communication units, including the main memory, are treated as unintelligent workers that are incapable of computation. As a result of this processor-centric design paradigm, data moves a lot in the system between the computation units and communication/storage units so that computation can be done on it. With the increasingly data-centric nature of contemporary and emerging appli-
PIM Course (Fall 2022)

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

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

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0G0BnK](https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0G0BnK)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX](https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX)

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
SSD Course (Spring 2023)

- **Spring 2023 Edition:**

- **Fall 2022 Edition:**

- **Youtube Livestream (Spring 2023):**
  - [https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB](https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB)

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&pp=iAQB](https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&pp=iAQB)

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Genomics Course (Fall 2022)

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

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

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV](https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDqOR1yRU_Cxxjw-u18](https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDqOR1yRU_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)
We Need to Think Differently from the Past Approaches
Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory
In-Storage Genomic Data Filtering [ASPLOS 2022]

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

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


[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
Genome Sequence Analysis

Data Movement from Storage

Storage System

Main Memory

Cache

Computation Unit
(CPU or Accelerator)

Alignment

X

Computation overhead

X

Data movement overhead
Compute-Centric Accelerators

- Heuristics
- Accelerators
- Filters

Storage System

Main Memory

Cache

Computation Unit (CPU or Accelerator)

✓ Computation overhead

✗ Data movement overhead
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

Computation Unit (CPU or Accelerator)

Main Memory

Cache

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

¹ETH Zürich  ²Bionano Genomics  ³KMUTNB  ⁴University of Toronto
Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory
In-Flash Bulk Bitwise Execution

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungsuk Kim, and Onur Mutlu,

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

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

[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Lecture Video (44 minutes)]
[arXiv version]

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

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

§ETH Zürich ▼ POSTECH †LIRMM, Univ. Montpellier, CNRS ‡Kyungpook National University

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

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

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

Low-cost & requires no changes to flash cell arrays.
Conventional systems: Outside-storage processing (OSP) that must move the entire data to CPUs/GPUs through the memory hierarchy.

**Memory bandwidth:**

- ~40 GB/s

**Storage I/O bandwidth:**

- ~8 GB/s

**External I/O bandwidth** of storage systems is the main bottleneck in conventional systems (OSP).
Near-Data Processing for Bitwise Operations

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

**In-Storage Processing Unit** (e.g., Biscuit\(^4\))

**SRAM-Based Cache** (e.g., Compute Cache\(^1\))

**DRAM-Based Main Memory** (e.g., Ambit\(^2\))

**NAND Flash-Based Storage** (e.g., ParaBit\(^5\))

**NVM-Based Main Memory** (e.g., Pinatubo\(^3\))

---

5. Gao+, “ParaBit: Processing Parallel Bitwise Operations in NAND Flash Memory Based SSDs,” MICRO, 2021
Near-Data Processing for Bitwise Operations

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

**In-Storage Processing Unit** (e.g., Biscuit\(^4\))

**Near-Data Processing (NDP)**

- SRAM-Based Cache (e.g., Compute Cache\(^1\))
- DRAM-Based Main Memory (e.g., Ambit\(^2\))
- NVM-Based Main Memory (e.g., Pinatubo\(^3\))

**Our focus:**
Large data sets that do not fit in main memory

NAND Flash-Based Storage (e.g., ParaBit\(^5\))

---

\(^1\)Aga+, “Compute Caches,” HPCA, 2017
\(^5\)Gao+, “ParaBit: Processing Parallel Bitwise Operations in NAND Flash Memory Based SSDs,” MICRO, 2021
In-Storage Processing (ISP)

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

Memory bandwidth: 
~ 40 GB/s

CPU/GPU
$ (SRAM)

Main Memory (DRAM)

Storage (NAND Flash-Based SSD)

Storage I/O bandwidth: 
~ 8 GB/s
In-Storage Processing (ISP)

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

**Memory bandwidth:**
- ~ 40 GB/s

**Storage I/O bandwidth:**
- ~ 8 GB/s
In-Storage Processing (ISP)

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

**Memory bandwidth:**
~ 40 GB/s

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

ISP can **mitigate** data movement overhead by **reducing** SSD-external data data movement.
In-Storage Processing (ISP)

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

**Memory bandwidth:**
~ 40 GB/s

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

SSD-internal bandwidth becomes the **new bottleneck** in ISP
In-Flash Processing (IFP)

- Performs computation inside NAND flash chips

**Memory bandwidth:**
~ 40 GB/s

**Results only**

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

**In-flash processing**

**SSD internal I/O bandwidth:**
~ 10 GB/s
In-Flash Processing (IFP)

- Performs computation inside NAND flash chips

Memory bandwidth: ~ 40 GB/s

Storage I/O bandwidth: ~ 8 GB/s

SSD internal I/O bandwidth: ~ 10 GB/s

IFP fundamentally mitigates data movement
**Our Proposal: Flash-Cosmos**

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

![Diagram of NAND Flash Chip with Bitlines (BLs) and Operands O₁, O₂, O₃, ..., O₃₂]

**Simultaneous sensing**
Key Ideas of Flash-Cosmos

Multi-Wordline Sensing (MWS) to enable in-flash bulk bitwise operations via a single sensing operation

Enhanced SLC-Mode Programming (ESP) to eliminate raw bit errors in stored data (and thus in computation results)
Multi-Wordline Sensing (MWS): Bitwise AND

- **Intra-Block MWS:**
  - Simultaneously activates multiple WLs in the same block
  - → Bitwise AND of the stored data in the WLs
Multi-Wordline Sensing (MWS): Bitwise AND

- **Intra-Block MWS:**
  Simultaneously activates multiple WLs in the same block → Bitwise AND of the stored data in the WLs

![Diagram of Multi-Wordline Sensing (MWS)]

Non-Target Cells: *Operate as resistors*
Multi-Wordline Sensing (MWS): Bitwise AND

- **Intra-Block MWS:**
  - Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs

**Target Cells:**
- Operate as resistors (1)
- or open switches (0)

**Non-Target Cells:**
- Operate as resistors

Result: 0 0 0 1
Multi-Wordline Sensing (MWS): Bitwise AND

- **Intra-Block MWS:**
  Simultaneously activates multiple WLs in the same block → Bitwise AND of the stored data in the WLs

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

Non-Target Cell: *Operate as a resistance*

Target Cell:

**Result:** 0 0 1 1
**Multi-Wordline Sensing (MWS): Bitwise AND**

- **Intra-Block MWS:**
  Simultaneously activates multiple WLs in the same block

→ Bitwise AND of the stored data in the WLs

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

**Result:**
0 0 0 0
Multi-Wordline Sensing (MWS): Bitwise AND

- **Intra-Block MWS:**
  - Simultaneously activates multiple WLs in the same block
  - Bitwise AND of the stored data in the WLs

A bitline reads as ‘1’ only when all the target cells store ‘1’

→ Equivalent to the bitwise AND of all the target cells

---

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

Result: 0 0 0 0
Multi-Wordline Sensing (MWS): Bitwise AND

- **Intra-Block MWS:**
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs

Flash-Cosmos (Intra-Block MWS) enables bitwise AND of multiple pages in the same block via a single sensing operation
• Inter-Block MWS: Simultaneously activates multiple WLs in different blocks
  → Bitwise OR of the stored data in the WLs
**Multi-Wordline Sensing (MWS): Bitwise OR**

- **Inter-Block MWS:**
  - Simultaneously activates multiple WLs in different blocks
  - Bitwise OR of the stored data in the WLs

```
WL_x in Block_1

WL_y in Block_i

Result: 1 1 1 0
```
Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:**
  Simultaneously activates multiple WLs in different blocks
  → Bitwise OR of the stored data in the WLs

A bitline reads as ‘0’ only when all the target cells store ‘0’
→ Equivalent to the bitwise OR of all the target cells
Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:**
  Simultaneously activates multiple WLs in different blocks → Bitwise OR of the stored data in the WLs

![Diagram of Multi-Wordline Sensing (MWS): Bitwise OR](image)
Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:**
  Simultaneously activates multiple WLs in different blocks → Bitwise OR of the stored data in the WLs

A bitline reads as ‘0’ only when all the target cells store ‘0’ → Equivalent to the bitwise OR of all the target cells
Multi-Wordline Sensing (MWS): Bitwise OR

- **Inter-Block MWS:** Simultaneously activates multiple WLs in different blocks → Bitwise OR of the stored data in the WLs

Flash-Cosmos (Inter-Block MWS) enables bitwise OR of multiple pages in different blocks via a single sensing operation
Flash-Cosmos also enables other types of bitwise operations (NOT/NAND/NOR/XOR/XNOR) leveraging existing features of NAND flash memory.

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

Jisung Park$\S$ Roknoddin Azizi$\S$ Geraldo F. Oliveira$\S$ Mohammad Sadrosadati$\S$
Rakesh Nadig$\S$ David Novo$\dagger$ Juan Gómez-Luna$\S$ Myungsuk Kim$\ddagger$ Onur Mutlu$\S$

$\S$ETH Zürich $\S$POSTECH $\dagger$LIRMM, Univ. Montpellier, CNRS $\ddagger$Kyungpook National University

Key Ideas

Multi-Wordline Sensing (MWS) to enable in-flash bulk bitwise operations via a single sensing operation

Enhanced SLC-Mode Programming (ESP) to eliminate raw bit errors in stored data (and thus in computation results)
Goal: eliminate raw bit errors in stored data (and computation results)

Key ideas
- Programs only a single bit per cell (SLC-mode programming)
  - Trades storage density for reliable computation
- Performs more precise programming of the cells
  - Trades programming latency for reliable computation

Maximizes the reliability margin between the different states of flash cells
Enhanced SLC-Mode Programming (ESP)

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

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

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

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

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

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

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

ESP significantly improves the reliability of computation results (no observed bit error in the tested flash cells).
Results: Performance & Energy

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

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

SAFARI
Summary: Flash-Cosmos

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

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

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

Low-cost & requires no changes to flash cell arrays.
In-Flash Bulk Bitwise Execution

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungsuk Kim, and Onur Mutlu,

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

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

[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Lecture Video (44 minutes)]
[arXiv version]

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

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

§ETH Zürich ▼ POSTECH †LIRMM, Univ. Montpellier, CNRS ‡Kyungpook National University

Processing in Memory: Two Approaches

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

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

Fundamentally
High-Performance
(Data-Centric)
Computing Architectures
Challenge and Opportunity for Future Computing Architectures with Minimal Data Movement
Storage-Centric Computing (I)

Storage system is a heterogeneous computing device with hybrid memory

Storage system enables data-centric design of systems & workloads

Application-driven customization enables a powerful data-centric engine

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


Storage-Centric Computing (II)

Workload-customized storage-centric computing
Workload-Customized Storage-Centric Computing

- Software and hardware customized for major workloads
  - Genomics
  - Video analytics
  - Data & graph analytics
  - Machine learning
  - ...

- Data-centric (processing capability in all memories)
- Data-driven (design & decision making)
- Data-aware (optimization & design)

- Unified interfaces for efficient & fast communication
Concluding Remarks
Concluding Remarks

- We must design systems to be **balanced, high-performance, energy-efficient** (all at the same time) → intelligent systems
  - **Data-centric, data-driven, data-aware**

- Enable computation capability inside and close to memory

- **This** can
  - Lead to **orders-of-magnitude** improvements
  - **Enable new applications & computing platforms**
  - **Enable better understanding of nature**
  - ...

- Future of **truly memory-centric computing** is bright
  - **We need to do research & design across the computing stack**
Fundamentally Better Architectures

Data-centric

Data-driven

Data-aware
We Need to Revisit the Entire Stack

We can get there step by step
A Blueprint for Fundamentally Better Architectures

Onur Mutlu,
"Intelligent Architectures for Intelligent Computing Systems"
Invited Paper in Proceedings of the Design, Automation, and Test in
Europe Conference (DATE), Virtual, February 2021.

[Slides (pptx) (pdf)]
[IEDM Tutorial Slides (pptx) (pdf)]
[Short DATE Talk Video (11 minutes)]
[Longer IEDM Tutorial Video (1 hr 51 minutes)]

Intelligent Architectures for Intelligent Computing Systems

Onur Mutlu
ETH Zurich
omutlu@gmail.com
Acknowledgments

SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

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

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

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

Think BIG, Aim HIGH!

SAFARI

https://safari.ethz.ch
Referenced Papers, Talks, Artifacts

- All are available at

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

  https://www.youtube.com/onurmutlulectures

  https://github.com/CMU-SAFARI/
Open Source Tools: SAFARI GitHub

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

210 followers  ETH Zurich and Carnegie Mellon U...  https://safari.ethz.ch/  omutlu@gmail.com

Overview  Repositories 77  Projects  Packages  People 13

Pinned

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

  - C++  405  184

- **MQSim**  Public
MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

  - C++  181  108

- **prim-benchmarks**  Public
PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publi...

  - C  79  33

- **rowhammer**  Public

  - C  199  40

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

  - C  50  11

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

  - Verilog  94  26

https://github.com/CMU-SAFARI/
Reinventing Computing & Storage

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
7 April 2023
OlympusMons Award Keynote Speech
Backup Slides
Executive Summary

**Background**: A hybrid storage system (HSS) uses multiple different storage devices to provide high and scalable storage capacity at high performance.

**Problem**: Two key shortcomings of prior data placement policies:
- Lack of adaptivity to:
  - Workload changes
  - Changes in device types and configurations
- Lack of extensibility to more devices

**Goal**: Design a data placement technique that provides:
- Adaptivity, by continuously learning and adapting to the application and underlying device characteristics
- Easy extensibility to incorporate a wide range of hybrid storage configurations

**Contribution**: Sibyl, the first reinforcement learning-based data placement technique in hybrid storage systems that:
- Provides adaptivity to changing workload demands and underlying device characteristics
- Can easily extend to any number of storage devices
- Provides ease of design and implementation that requires only a small computation overhead

**Key Results**: Evaluate on real systems using a wide range of workloads
- Sibyl improves performance by 21.6% compared to the best previous data placement technique in dual-HSS configuration
- In a tri-HSS configuration, Sibyl outperforms the state-of-the-art-policy policy by 48.2%
- Sibyl achieves 80% of the performance of an oracle policy with storage overhead of only 124.4 KiB

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

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

https://www.youtube.com/watch?v=qeukNs5XI3g
Special Research Sessions & Courses (II)

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

Livestream - P&I Data-Centric Architectures: Fundamentally...

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

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

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

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

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

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

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

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

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

https://www.youtube.com/playlist?list=PL5Q2soXY2Zi8KzG2CQYRNQQOVD0GOBrnKy
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_5EFg0&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF](https://www.youtube.com/watch?v=4yfkM_5EFg0&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:
- https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule

Spring 2021 Edition:

Youtube Livestream (Spring 2022):
- https://www.youtube.com/watch?v=cpXdE3HwyK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6

Youtube Livestream (Spring 2021):
- https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN

Bachelor’s course
- 2nd semester at ETH Zurich
- Rigorous introduction into “How Computers Work”
- Digital Design/Logic
- Computer Architecture
- 10 FPGA Lab Assignments

https://www.youtube.com/onurmutlulectures
Processing-in-Memory Course (Fall 2022)

- Short weekly lectures
- Hands-on projects

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

Course Description

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

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

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

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

https://youtube.com/playlist?list=PL5Q2soXY2Zi8KzG2CQYRNQ0VD0GOBrnKy
PIM Course (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory

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

- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQ0VD0GOBrnKy

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX

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

https://www.youtube.com/onurmutlulectures
Real PIM Tutorial (HPCA 2023)

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

https://www.youtube.com/watch?v=f5-nT1tbz5w
https://events.safari.ethz.ch/real-pim-tutorial/
Real PIM Tutorial (ASPLOS 2023)

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

---

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

- **Important note about registration**

**Tutorial Description**

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

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

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

**2.560-DPU Processing-in-Memory System**

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) communicate successful stories for both hardware and software.

**Materials**

<table>
<thead>
<tr>
<th>Time</th>
<th>Speaker</th>
<th>Title</th>
<th>Materials</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00am-10:20am</td>
<td>Prof. Orur Mutlu</td>
<td>Memory-Centric Computing</td>
<td>(PDF)</td>
</tr>
<tr>
<td>10:40am-12:00pm</td>
<td>Dr. Juan Gómez Luna</td>
<td>Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM</td>
<td>(PDF)</td>
</tr>
<tr>
<td>1:40pm-2:20pm</td>
<td>Prof. Alexandra (Sasha) Fedorova (UBC)</td>
<td>Processing in Memory in the Wild</td>
<td>(PDF)</td>
</tr>
<tr>
<td>2:20pm-3:20pm</td>
<td>Dr. Juan Gómez Luna &amp; Alaber Koglo</td>
<td>Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components</td>
<td>(PDF) (PPT)</td>
</tr>
<tr>
<td>3:40pm-4:10pm</td>
<td>Dr. Juan Gómez Luna</td>
<td>Adoption issues: How to enable PIM? Accelerating Modern Workloads on a General-purpose PIM System</td>
<td>(PDF) (PPT)</td>
</tr>
<tr>
<td>4:10pm-4:50pm</td>
<td>Dr. Yongkie Kwon &amp; Eddy (Chunwook) Park (SK Hynix)</td>
<td>System Architecture and Software Stack for GDDR6-AIM</td>
<td>(PDF) (PPT)</td>
</tr>
<tr>
<td>4:50pm-5:00pm</td>
<td>Dr. Juan Gómez Luna</td>
<td>Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture</td>
<td>(Handout) (PDF) (PPT)</td>
</tr>
</tbody>
</table>

See more details and watch the tutorial videos: [https://www.youtube.com/watch?v=oYCaLcT0Kmo](https://www.youtube.com/watch?v=oYCaLcT0Kmo)
Upcoming Real PIM Tutorial (ISCA 2023)

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

https://events.safari.ethz.ch/isca-pim-tutorial/
SSD Course (Spring 2023)

- **Spring 2023 Edition:**

- **Fall 2022 Edition:**

- **Youtube Livestream (Spring 2023):**
  - [https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB](https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB)

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&pp=iAQB](https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&pp=iAQB)

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Genomics Course (Fall 2022)

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

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

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV](https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDqOR1yRU_Cxxjw-u18](https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDqOR1yRU_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)
**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=PL5Q2soXY2Zi9XrqXR38IM_FTjmY6h7Gzm](https://www.youtube.com/watch?v=oFO5fTrgFIY&list=PL5Q2soXY2Zi9XrqXR38IM_FTjmY6h7Gzm)

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

---

### Spring 2022 Meetings/Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Livestream</th>
<th>Meeting</th>
<th>Learning Materials</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>15.03Tue.</td>
<td>YouTube Premiere</td>
<td>M1: P&amp;S Course Presentation</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W2</td>
<td>22.03Tue.</td>
<td>YouTube Premiere</td>
<td>M2: SIMD Processing and GPUs</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W3</td>
<td>29.03Tue.</td>
<td>YouTube Premiere</td>
<td>M3: GPU Software Hierarchy</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W4</td>
<td>05.04Tue.</td>
<td>YouTube Premiere</td>
<td>M4: GPU Memory Hierarchy</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W5</td>
<td>12.04Tue.</td>
<td>YouTube Premiere</td>
<td>M5: GPU Performance Considerations</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W6</td>
<td>19.04Tue.</td>
<td>YouTube Premiere</td>
<td>M6: Parallel Patterns: Reduction</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W7</td>
<td>26.04Tue.</td>
<td>YouTube Premiere</td>
<td>M7: Parallel Patterns: Histogram</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W8</td>
<td>03.05Tue.</td>
<td>YouTube Premiere</td>
<td>M8: Parallel Patterns: Convolution</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W9</td>
<td>10.05Tue.</td>
<td>YouTube Premiere</td>
<td>M9: Parallel Patterns: Prefix Sum</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W10</td>
<td>17.05Tue.</td>
<td>YouTube Premiere</td>
<td>M10: Parallel Patterns: Sparse Matrices</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W11</td>
<td>24.05Tue.</td>
<td>YouTube Premiere</td>
<td>M11: Parallel Patterns: Graph Search</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W12</td>
<td>01.06Wed.</td>
<td>YouTube Premiere</td>
<td>M12: Parallel Patterns: Merge Sort</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W13</td>
<td>07.06Tue.</td>
<td>YouTube Premiere</td>
<td>M13: Dynamic Parallelism</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W14</td>
<td>14.06Wed.</td>
<td>YouTube Premiere</td>
<td>M14: Collaborative Computing</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W15</td>
<td>21.06Fri.</td>
<td>YouTube Premiere</td>
<td>M15: GPU Acceleration of Genome Sequence Alignment</td>
<td>[PDF] [PPT]</td>
<td>HW 0 Out</td>
</tr>
<tr>
<td>W16</td>
<td>28.06Thu.</td>
<td>YouTube Premiere</td>
<td>M16: Accelerating Agent-based Simulations</td>
<td>[PDF] [ODP]</td>
<td>HW 0 Out</td>
</tr>
</tbody>
</table>
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=PL5Q2soXY2Zi8nH7un3ghD2nutKWWDk-NK](https://youtube.com/playlist?list=PL5Q2soXY2Zi8nH7un3ghD2nutKWWDk-NK)

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
RowHammer & DRAM Exploration (Fall 2022)

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

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

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

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

https://www.youtube.com/onurmutlulectures
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=aMIIXRQd3s&list=PL5Q2soXY2Zi_TlmLGw_Z8hBo2925ZApqV](https://www.youtube.com/watch?v=aMIIXRQd3s&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)