

SAFARI Research Group

Introduction & Research

Onur Mutlu

omutlu@gmail.com

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

19 October 2021

EFCL Huawei Day

SAFARI

ETH zürich

Carnegie Mellon

Brief Self Introduction



■ Onur Mutlu

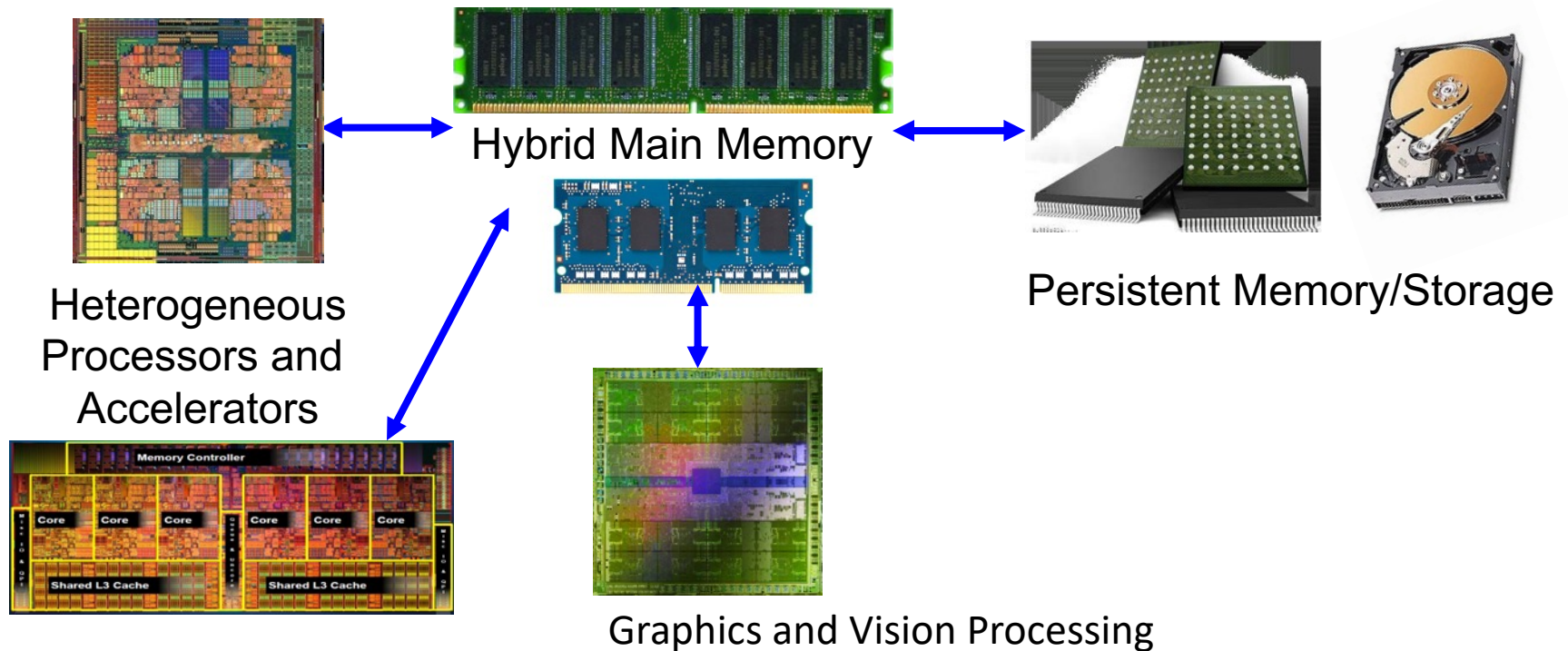
- ❑ Full Professor @ ETH Zurich ITET (INFK), since September 2015
- ❑ Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-...
- ❑ PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
- ❑ <https://people.inf.ethz.ch/omutlu/>
- ❑ omutlu@gmail.com (Best way to reach me)
- ❑ <https://people.inf.ethz.ch/omutlu/projects.htm>

■ Research and Teaching in:

- ❑ Computer architecture, computer systems, hardware security, bioinformatics
- ❑ Memory and storage systems
- ❑ Hardware security, safety, predictability
- ❑ Fault tolerance
- ❑ Hardware/software cooperation
- ❑ Architectures for bioinformatics, health, medicine
- ❑ ...

Current Research Mission

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

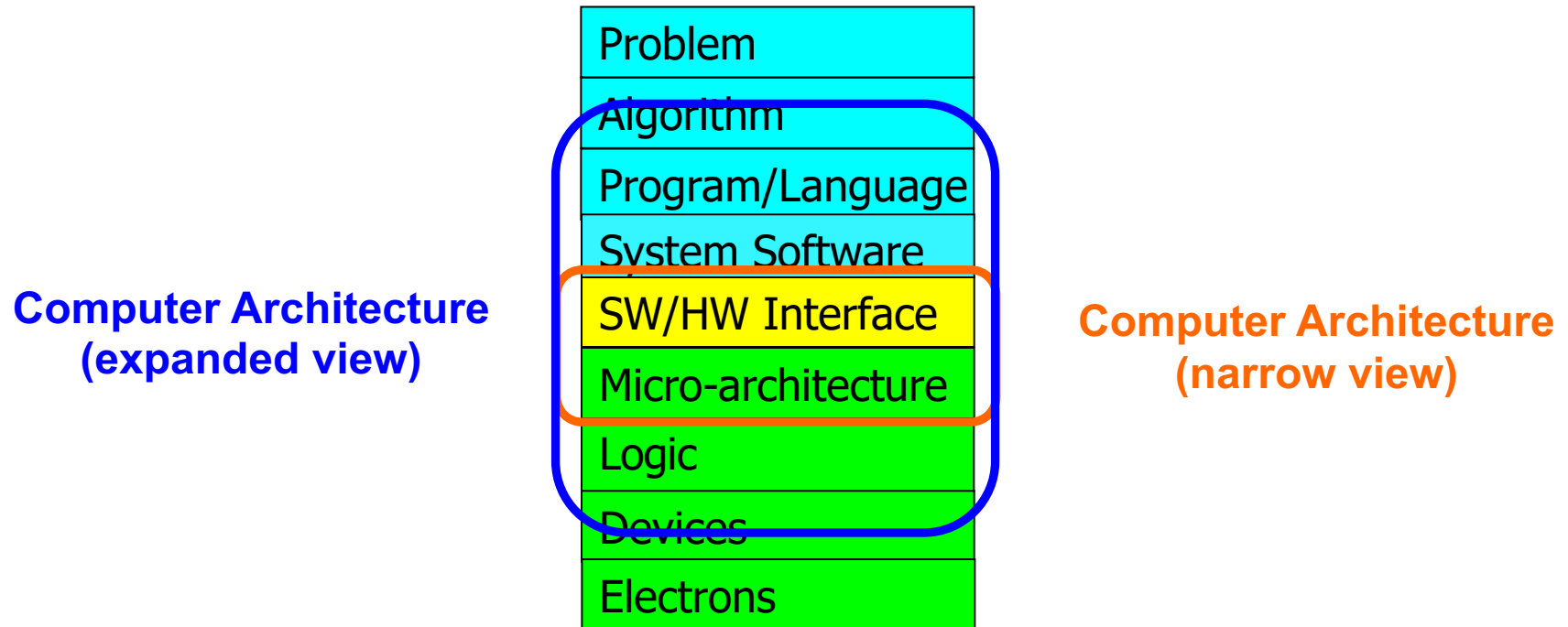


Build fundamentally better architectures

Four Key Current Directions

- Fundamentally **Secure/Reliable/Safe** Architectures
- Fundamentally **Energy-Efficient** Architectures
 - **Memory-centric** (Data-centric) Architectures
- Fundamentally **Low-Latency and Predictable** Architectures
- Architectures for **AI/ML, Genomics, Medicine, Health, ...**

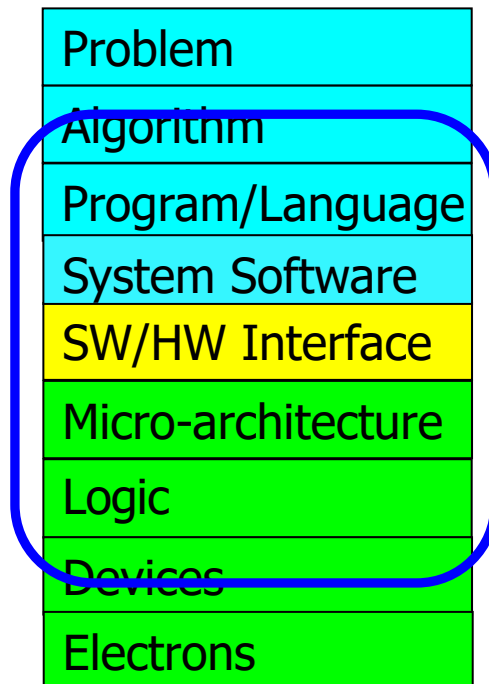
The Transformation Hierarchy



Axiom

To achieve the highest **energy efficiency** and **performance**:

we must take the expanded view
of computer architecture

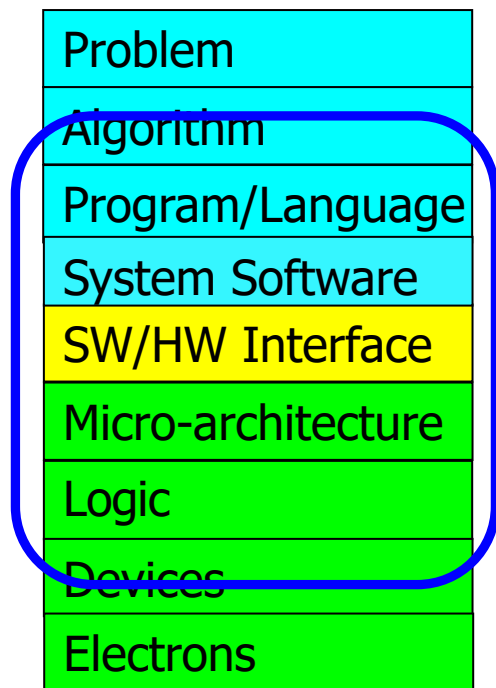


Co-design across the hierarchy:
Algorithms to devices

Specialize as much as possible
within the design goals

Current Research Mission & Major Topics

Build fundamentally better architectures



**Broad research
spanning apps, systems, logic
with architecture at the center**

- Data-centric arch. for low energy & high perf.
 - Proc. in Mem/DRAM, NVM, unified mem/storage
- Low-latency & predictable architectures
 - Low-latency, low-energy yet low-cost memory
 - QoS-aware and predictable memory systems
- Fundamentally secure/reliable/safe arch.
 - Tolerating all bit flips; patchable HW; secure mem
- Architectures for ML/AI/Genomics/Health/Med
 - Algorithm/arch./logic co-design; full heterogeneity
- Data-driven and data-aware architectures
 - ML/AI-driven architectural controllers and design
 - Expressive memory and expressive systems

SAFARI

SAFARI Research Group

safari.ethz.ch

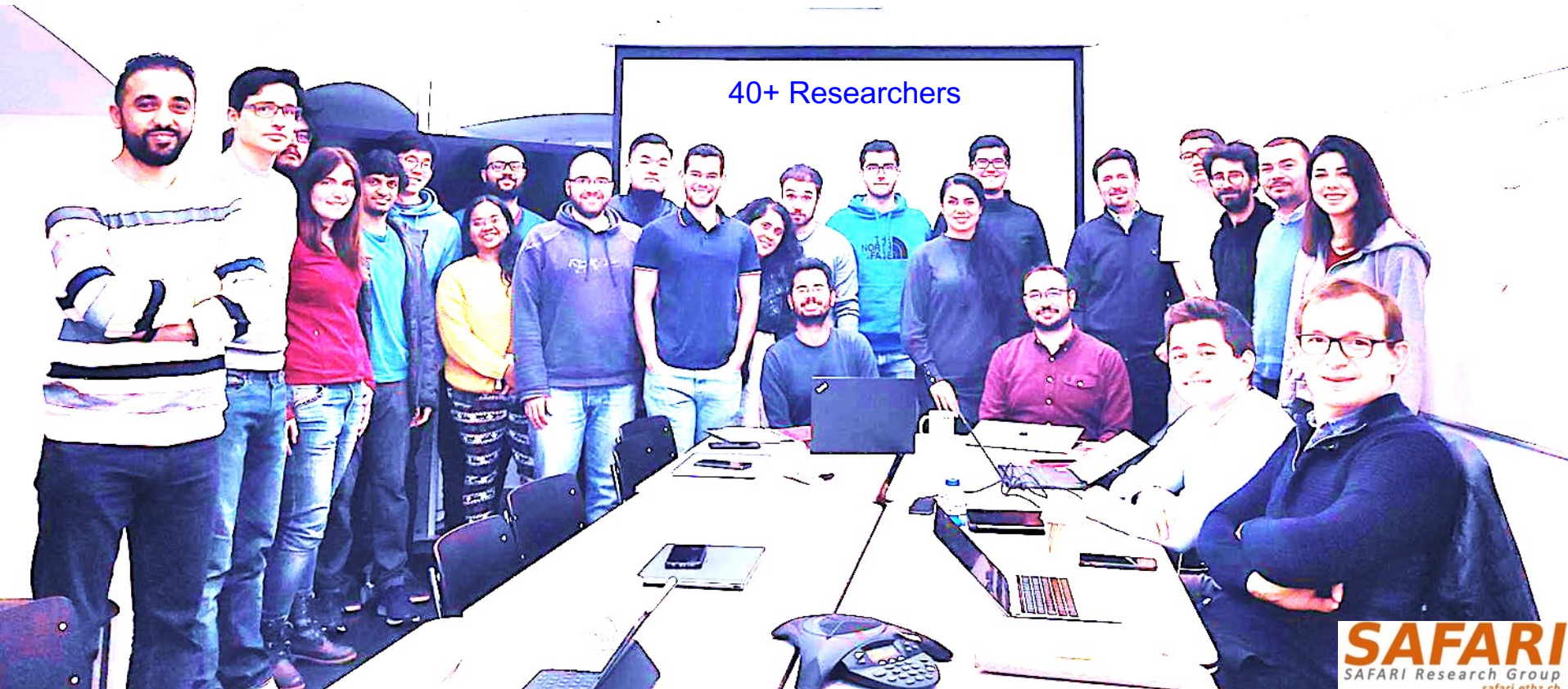
Think BIG, Aim HIGH!

<https://safari.ethz.ch>

Onur Mutlu's SAFARI Research Group

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

<https://safari.ethz.ch/safari-newsletter-april-2020/>



SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

SAFARI

<https://safari.ethz.ch>

SAFARI Newsletter January 2021 Edition

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



SAFARI
SAFARI Research Group

Newsletter
January 2021

*Think Big, Aim High, and
Have a Wonderful 2021!*



Dear SAFARI friends,

Happy New Year! We are excited to share our group highlights with you in this second edition of the SAFARI newsletter (You can find the first edition from April 2020 [here](#)). 2020 has

SAFARI PhD and Post-Doc Alumni

- <https://safari.ethz.ch/safari-alumni/>
- Damla Senol Cali (Bionano Genomics)
- Nastaran Hajinazar (ETH Zurich)
- Gagandeep Singh (ETH Zurich)
- Amirali Boroumand (Stanford Univ → Google)
- Jeremie Kim (ETH Zurich)
- Nandita Vijaykumar (Univ. of Toronto, Assistant Professor)
- Kevin Hsieh (Microsoft Research, Senior Researcher)
- Justin Meza (Facebook)
- Mohammed Alser (ETH Zurich)
- Yixin Luo (Google)
- Kevin Chang (Facebook)
- Rachata Ausavarungrun (KMUNTB, Assistant Professor)
- Gennady Pekhimenko (Univ. of Toronto, Assistant Professor)
- Vivek Seshadri (Microsoft Research)
- Donghyuk Lee (NVIDIA Research, Senior Researcher)
- Yoongu Kim (Software Robotics → Google)
- Lavanya Subramanian (Intel Labs → Facebook)
- Samira Khan (Univ. of Virginia, Assistant Professor)
- Saugata Ghose (Univ. of Illinois, Assistant Professor)
- Jawad Haj-Yahya (Huawei Research Zurich, Principal Researcher)

Principle: Teaching and Research

...

Teaching drives Research

Research drives Teaching

...

Research & Teaching: Some Overview Talks

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

■ Future Computing Architectures

- https://www.youtube.com/watch?v=kgiZISOcGFM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=1

■ Enabling In-Memory Computation

- https://www.youtube.com/watch?v=njX_14584Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=16

■ Accelerating Genome Analysis

- https://www.youtube.com/watch?v=r7sn41IH-4A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=41

■ Rethinking Memory System Design

- https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=3

■ Intelligent Architectures for Intelligent Machines

- https://www.youtube.com/watch?v=c6_LgzuNdkw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=25

■ The Story of RowHammer

- https://www.youtube.com/watch?v=sgd7PHQQ1AI&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=39

Online Courses & Lectures

■ **First Computer Architecture & Digital Design Course**

- ❑ Digital Design and Computer Architecture
- ❑ Spring 2021 Livestream Edition:
https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN

■ **Advanced Computer Architecture Course**

- ❑ Computer Architecture
- ❑ Fall 2020 Edition:
<https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN>



Onur Mutlu Lectures

16.9K subscribers

CUSTOMIZE CHANNEL

MANAGE VIDEOS

HOME

VIDEOS

PLAYLISTS

COMMUNITY

CHANNELS

ABOUT



Popular uploads ▶ PLAY ALL

How Computers Work (from the ground up)

1:33:25

Digital Design & Computer Architecture: Lecture 1: Introduction and...

49K views • 1 year ago

Computer Architecture - Lecture 1: Introduction and...

36K views • 3 years ago

Computer Architecture - Lecture 1: Introduction and...

31K views • 1 year ago

Computer Architecture - Lecture 1: Introduction and...

30K views • 8 months ago

Design of Digital Circuits - Lecture 1: Introduction and...

22K views • 2 years ago

Computer Architecture - Lecture 2: Fundamentals,...

17K views • 3 years ago

First Course in Computer Architecture & Digital Design 2021-2013

Livestream - Digital Design and Computer Architecture - ETH...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Digital Design & Computer Architecture - ETH Zürich...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Design of Digital Circuits - ETH Zürich - Spring 2019

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Design of Digital Circuits - ETH Zürich - Spring 2018

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Digital Circuits and Computer Architecture - ETH Zurich --...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Spring 2015 -- Computer Architecture Lectures --...

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

Advanced Computer Architecture Courses 2020-2012

Computer Architecture - ETH Zürich - Fall 2020

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Computer Architecture - ETH Zürich - Fall 2019

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Computer Architecture - ETH Zürich - Fall 2018

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Computer Architecture - ETH Zürich - Fall 2017

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Fall 2015 - 740 Computer Architecture

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

Fall 2013 - 740 Computer Architecture - Carnegie Mellon

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

Special Courses on Memory Systems

Memory Technology Lectures

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Champéry Winter School 2020 - Memory Systems and Memory...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Perugia NIPS Summer School 2019

Onur Mutlu Lectures

VIEW FULL PLAYLIST

SAMOS Tutorial 2019 - Memory Systems

Onur Mutlu Lectures

VIEW FULL PLAYLIST

TU Wien 2019 - Memory Systems and Memory-Centric...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

ACACES 2018 Lectures -- Memory Systems and Memory...

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Research Talks

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

SAFARI

DDCA (Spring 2021)



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

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

Trace: · schedule

Home

Announcements

Materials

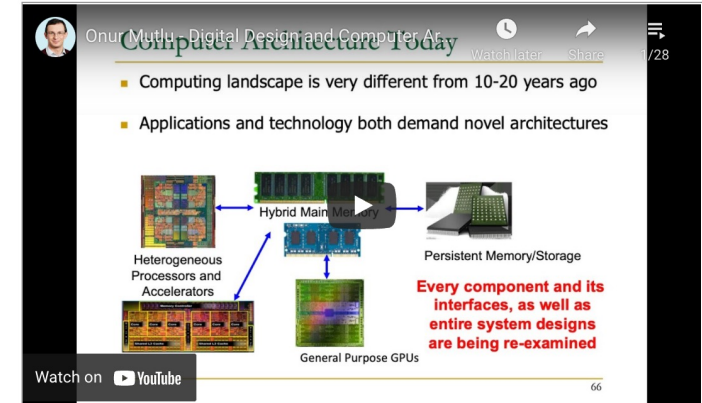
- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Optional HWs
- Labs
- Extra Assignments
- Exams
- Technical Docs

Resources

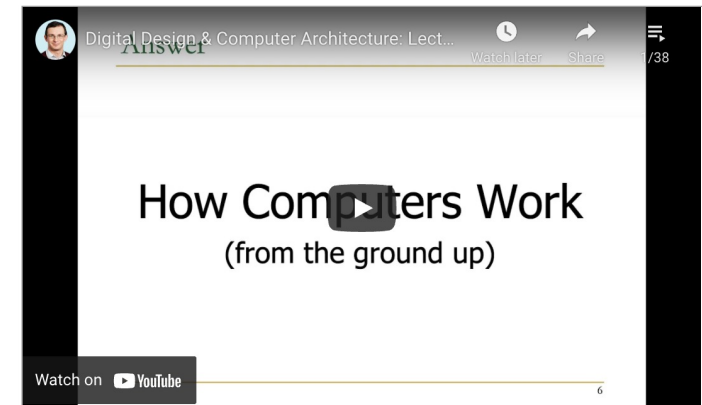
- Computer Architecture (CMU) SS15: Lecture Videos
- Computer Architecture (CMU) SS15: Course Website
- Digitaltechnik SS18: Lecture Videos
- Digitaltechnik SS18: Course Website
- Digitaltechnik SS19: Lecture Videos
- Digitaltechnik SS19: Course Website
- Digitaltechnik SS20: Lecture Videos
- Digitaltechnik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

Lecture Video Playlist on YouTube

Livestream Lecture Playlist



Recorded Lecture Playlist




Spring 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	25.02 Thu.	YouTube Live	L1: Introduction and Basics Q238 (PDF) Q240 (PPT)	Required Suggested Mentioned		
	26.02 Fri.	YouTube Live	L2a: Tradeoffs, Metrics, Mindset Q238 (PDF) Q240 (PPT)	Required		
			L2b: Mysteries in Computer Architecture Q238 (PDF) Q240 (PPT)	Required Suggested Mentioned		
W2	04.03 Thu.	YouTube Live	L3a: Mysteries in Computer Architecture II Q238 (PDF) Q240 (PPT)	Required Suggested Mentioned		

Comp Arch (Fall 2020)

- <https://safari.ethz.ch/architecture/fall2020/doku.php?id=schedule>
- <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



Computer Architecture - Fall 2020

Recent Changes Media Manager Sitemap

Trace: start schedule

Home

Announcements

Materials


- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

Resources

- Computer Architecture FS19: Course Webpage
- Computer Architecture FS19: Lecture Videos
- Digitaltechnik SS20: Course Webpage
- Digitaltechnik SS20: Lecture Videos
- Moodle
- Piazza (Q&A)
- HotCRP
- Verilog Practice Website (HDLBits)

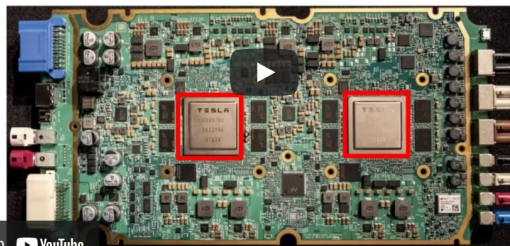
Lecture Video Playlist on YouTube

Lecture Playlist



Computer Architecture - Lecture: Introduction

- ML accelerator: 260 mm², 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.



Watch on YouTube

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

48

Fall 2020 Lectures & Schedule

Week	Date	Lecture	Readings	Lab	HW
W1	17.09 Thu.	L1: Introduction and Basics CORA (PDF) PPT YouTube Video	Described Suggested		HW 0 Out
		L2a: Memory Performance Attacks CORA (PDF) PPT YouTube Video	Described Suggested	Lab 1 Out	
	18.09 Fri.	L2b: Data Retention and Memory Refresh CORA (PDF) PPT YouTube Video	Described Suggested		
		L2c: Course Logistics CORA (PDF) PPT YouTube Video			
W2	24.09 Thu.	L3a: Introduction to Genome Sequence Analysis CORA (PDF) PPT YouTube Video	Described Suggested		HW 1 Out
		L3b: Memory Systems: Challenges and Opportunities CORA (PDF) PPT YouTube Video	Described Suggested		
	25.09 Fri.	L4a: Memory Systems: Solution Directions CORA (PDF) PPT YouTube Video	Described Suggested		
		L4b: RowHammer CORA (PDF) PPT YouTube Video	Described Suggested		
W3	01.10 Thu.	L5a: RowHammer in 2020: TRRespass CORA (PDF) PPT YouTube Video	Described Suggested		
		L5b: RowHammer in 2020: Revisiting RowHammer CORA (PDF) PPT YouTube Video	Described Suggested		
		L5c: Secure and Reliable Memory CORA (PDF) PPT YouTube Video	Described		

Comp Arch (Current)


■ <https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule>

■ **Youtube Livestream:**

❑ https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILKTOF

■ **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


Computer Architecture - Fall 2021

Recent Changes
Media Manager
Sitemap

Trace:
readings
start
schedule

Home

Announcements

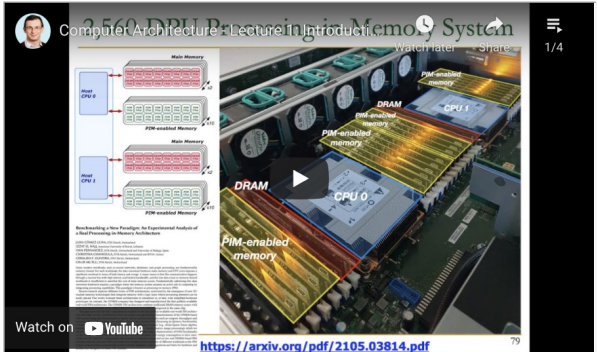
Materials

Resources

Lectures/Schedule
Lecture Buzzwords
Readings
HWs
Labs
Exams
Related Courses
Tutorials

Computer Architecture FS20: Course Webpage
Computer Architecture FS20: Lecture Videos
Digitaltechnik SS21: Course Webpage
Digitaltechnik SS21: Lecture Videos
Moodle
HotCRP
Verilog Practice Website (HDLBits)

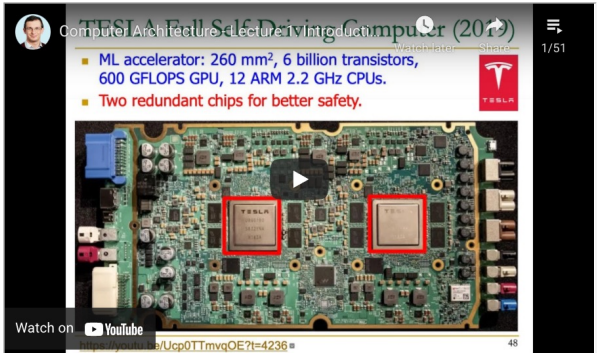
Lecture Video Playlist on YouTube
Livestream Lecture Playlist



Watch on

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

Recorded Lecture Playlist



Watch on

<https://www.youtube.com/watch?v=Ucp0TTmvqOE?e=4236>

Fall 2021 Lectures & Schedule


Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	30.09 Thu.		L1: Introduction and Basics L1a (PDF) L1b (PPT)	Required Mentioned	Lab 1 Out	HW 0 Out
	01.10 Fri.		L2: Trends, Tradeoffs and Design Fundamentals L2a (PDF) L2b (PPT)	Required Mentioned		
W2	07.10 Thu.		L3a: Memory Systems: Challenges and Opportunities L3a (PDF) L3a (PPT)	Described Suggested		HW 1 Out
			L3b: Course Info & Logistics L3b (PDF) L3b (PPT)			
			L3c: Memory Performance Attacks L3c (PDF) L3c (PPT)	Described Suggested		
	08.10 Fri.		L4a: Memory Performance Attacks L4a (PDF) L4a (PPT)	Described Suggested	Lab 2 Out	
			L4b: Data Retention and Memory Refresh L4b (PDF) L4b (PPT)	Described Suggested		
			L4c: RowHammer L4c (PDF) L4c (PPT)	Described Suggested		

Seminar (Spring'21)

■ https://safari.ethz.ch/architecture_seminar/spring2021/doku.php?id=schedule

■ https://www.youtube.com/watch?v=t3m93ZpLOyw&list=PL5Q2soXY2Zi_awYdjmWVIUegsbY7TPGW4

- Critical analysis course
 - Taken by Bachelor's/Masters/PhD students
 - Cutting-edge research topics + fundamentals in Computer Architecture
 - 20+ research papers, presentations, analyses


Seminar in Computer Architecture - Spring 2021

Recent Changes Media Manager Sitemap

Trace: - start - schedule

Home

Materials

- Announcements
- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Sessions
- Papers
- Synthesis Report
- Homework

Past Course Materials

- Fall 2020
- Spring 2020
- Fall 2019
- Spring 2019

Resources

Computer Architecture

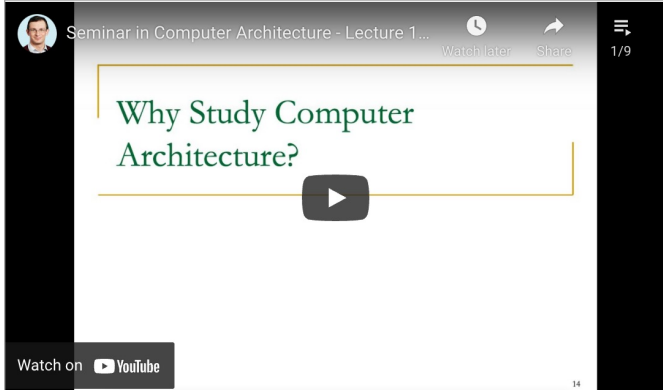
- Fall 2020
- Fall 2020: Lecture Videos
- Fall 2019
- Fall 2019: Lecture Videos
- Fall 2018
- Fall 2018: Lecture Videos

Digital Design and Computer Architecture

- Spring 2020
- Spring 2020: Lecture Videos
- Spring 2019
- Spring 2019: Lecture Videos

Lecture Video Playlist on YouTube

Lecture Playlist

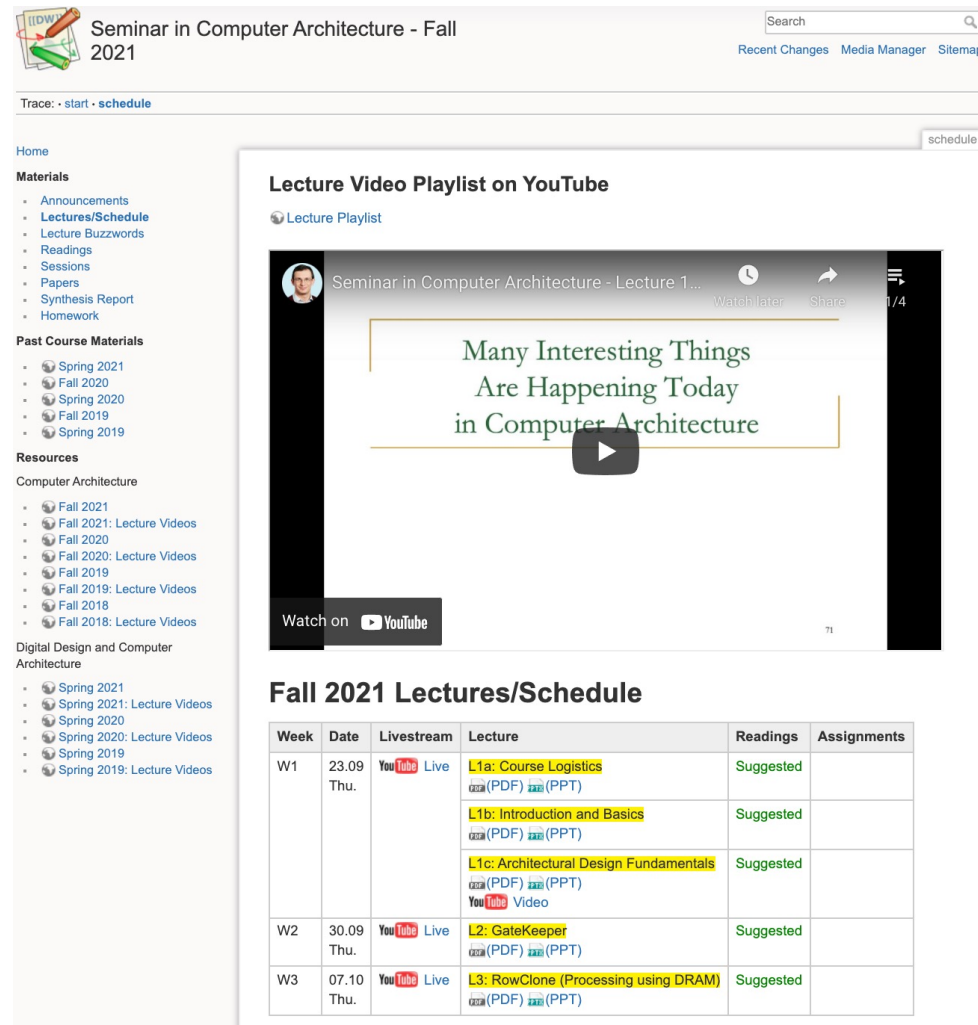


Spring 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Assignments
W1	25.02 Thu.	YouTube Live	L1a: Introduction and Basics <small>PDF (PPT)</small> Optional Lecture: Design Fundamentals <small>PDF (PPT)</small> L1b: Course Logistics <small>PDF (PPT)</small>	Suggested	
W2	04.03 Thu.	YouTube Live	L2: Example Review: RowClone <small>PDF (PPT)</small>	Suggested	
W3	11.03 Thu.	YouTube Live	L3: Example Review: Memory Channel Partitioning <small>PDF (PPT)</small>	Suggested	
W4	18.03 Thu.	YouTube Live	L4: Example Review: GateKeeper <small>PDF (PPT)</small>	Suggested	
W5	25.03 Thu.	YouTube Premiere	S1.1: Spectre Attacks: Exploiting Speculative Execution, S&P 2019 <small>PPT (PDF)</small> S1.2: BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows, HPCA 2021 <small>PPT (PDF)</small>	Mentioned	
W6	01.04 Thu.	YouTube Live	S2.1: D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput, HPCA 2019 <small>PPT (PDF)</small> S2.2: ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAMs, MICRO 2019 <small>PPT (PDF)</small>	Mentioned	
W7	15.04 Thu.		S3.1: PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture,	Mentioned	

Seminar (Current)

- https://safari.ethz.ch/architecture_seminar/fall2021/doku.php?id=schedule
- **Youtube Livestream:**
 - https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4
- Critical analysis course
 - Taken by Bachelor's/Masters/PhD students
 - Cutting-edge research topics + fundamentals in Computer Architecture
 - 20+ research papers, presentations, analyses



Seminar in Computer Architecture - Fall 2021

Trace: start - schedule

Home

Materials

- Announcements
- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Sessions
- Papers
- Synthesis Report
- Homework

Past Course Materials

- Spring 2021
- Fall 2020
- Spring 2020
- Fall 2019
- Spring 2019

Resources

Computer Architecture

- Fall 2021
- Fall 2021: Lecture Videos
- Fall 2020
- Fall 2020: Lecture Videos
- Fall 2019
- Fall 2019: Lecture Videos
- Fall 2018
- Fall 2018: Lecture Videos

Digital Design and Computer Architecture

- Spring 2021
- Spring 2021: Lecture Videos
- Spring 2020
- Spring 2020: Lecture Videos
- Spring 2019
- Spring 2019: Lecture Videos

Lecture Video Playlist on YouTube

Lecture Playlist

Seminar in Computer Architecture - Lecture 1...

Watch later Share 1/4

Many Interesting Things Are Happening Today in Computer Architecture

Watch on YouTube

Fall 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Assignments
W1	23.09 Thu.	YouTube Live	L1a: Course Logistics L1b: Introduction and Basics L1c: Architectural Design Fundamentals	Suggested	
W2	30.09 Thu.	YouTube Live	L2: GateKeeper	Suggested	
W3	07.10 Thu.	YouTube Live	L3: RowClone (Processing using DRAM)	Suggested	

SAFARI Live Seminars


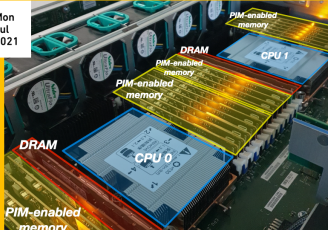
SAFARI Live Seminars in Computer Architecture

Dr. Juan Gómez Luna, ETH Zurich

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

SAFARI
SAFARI Research Group

12 Mon Jul 2021


SAFARI Live Seminars in Computer Architecture

Dr. Andrew Walker, Schiltron Corporation & Nexgen Power Systems

An Addition to Low Cost Per Memory Bit – How to Recognize It and What to Do About It

SAFARI
SAFARI Research Group

19 Mo Jul 2021





SAFARI Live Seminars in Computer Architecture

Geraldo F. Oliveira, ETH Zurich

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

SAFARI
SAFARI Research Group

22 Do Jul 2021



Near-Data Processing (2/2)

UPMEM (2019) Samsung HBM-PIM (2021)

Near-DRAM-banks processing for general-purpose computing

Near-DRAM-banks processing for neural networks

0.9 TOPS compute throughput¹ 1.2 TFLOPS compute throughput²

The goal of Near-Data Processing (NDP) is to mitigate data movement

SAFARI © 2021 ETH Zurich, The Data Processing & Research Institute, DSI, 2021. All rights reserved. This work is licensed under a Creative Commons Attribution 4.0 International License. For more information, see the License at <http://creativecommons.org/licenses/by/4.0/>.


SAFARI Live Seminars in Computer Architecture

Gennady Pekhimenko, University of Toronto

Efficient DNN Training at Scale: from Algorithms to Hardware

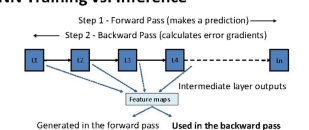
SAFARI
SAFARI Research Group

5 Do Aug 2021



DNN Training vs. Inference

Step 1 - Forward Pass (makes a prediction)
Step 2 - Backward Pass (calculates error gradients)



Generated in the forward pass Used in the backward pass

DNN training requires stashing feature maps for the backward pass (not required in Inference)


SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Power Management Mechanisms in Modern Microprocessors and Their Security Implications

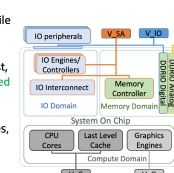
SAFARI
SAFARI Research Group

16 Mo Aug 2021



Overview of a Modern SoC Architecture

- 3 domains in modern thermally-constrained mobile SoC: Compute, Memory, IO
- Several voltage sources exist, and some of them are shared between domains
- IO controllers and engines, IO interconnect, memory controller, and DDRIO typically each has an independent clock




SAFARI Live Seminars in Computer Architecture

Ataberk Olgun, TOBB & ETH Zurich

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

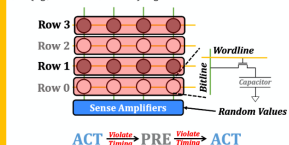
SAFARI
SAFARI Research Group

15 Mi Sep 2021



Using QUAC to Generate Random Values

Use QUAC to activate DRAM rows that are initialized with conflicting data (e.g., two '1's and two '0's) to generate random values



SAFARI © kasirga


SAFARI Live Seminars in Computer Architecture

Minesh Patel, ETH Zurich

Enabling Effective Error Mitigation in Memory Chips That Use On-Die ECCs

SAFARI
SAFARI Research Group

21 Tue Sep 2021



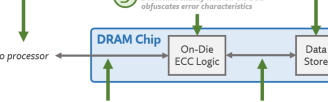
Position Paper (Ongoing) Arguing for increased transparency of DRAM reliability characteristics

REAPER (ISCA'17) Understand the basic properties of DRAM data-retention errors

BEER (MICRO'20, best paper) Determine exactly how on-die ECCs adjust error characteristics

HARP (MICRO'21) Understand how errors appear and how to identify at-risk bits

EIN (DSN'19, best paper) Understand and recover the error characteristics beneath on-die ECC




SAFARI Live Seminars in Computer Architecture

Christina Giannoula, National Technical University of Athens

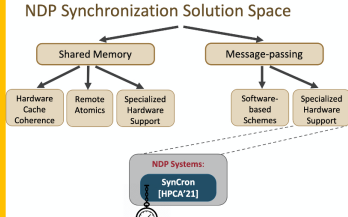
Efficient Synchronization Support for Near-Data-Processing Architectures

SAFARI
SAFARI Research Group

27 Mo Sep 2021



NDP Synchronization Solution Space




SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Security Implications of Power Management Mechanisms in Modern Processors, Current Studies and Future Trends

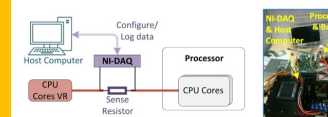
SAFARI
SAFARI Research Group

4 Mo Okt 2021

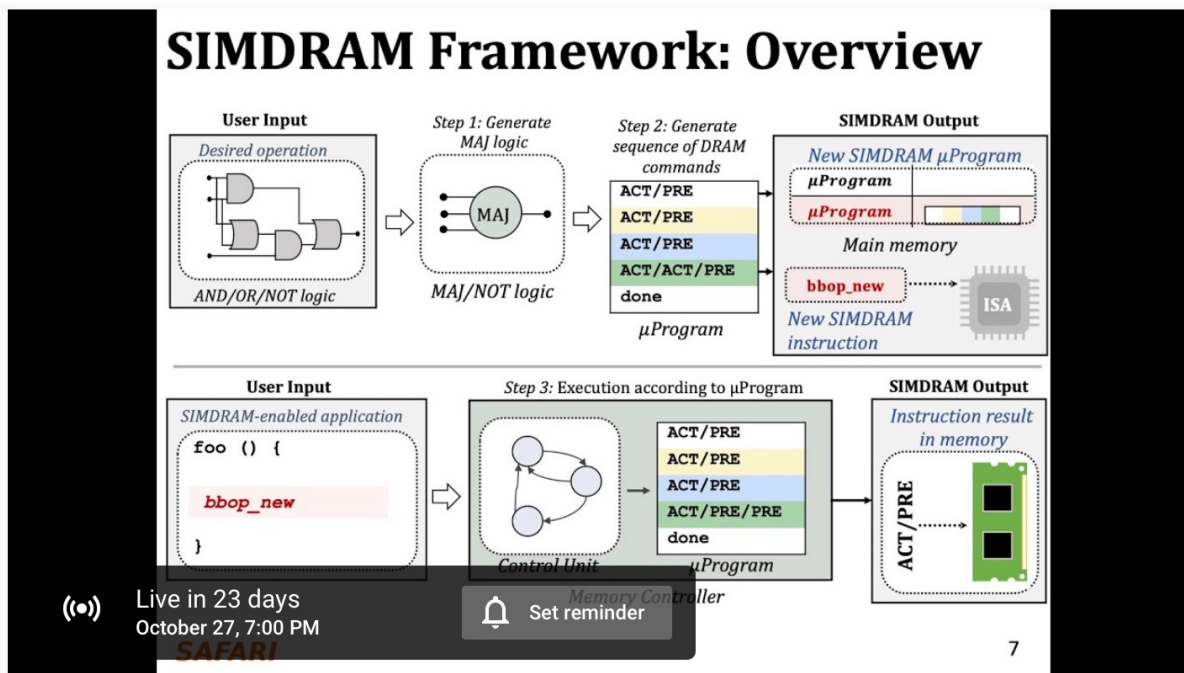


Experimental Methodology

- We experimentally study three modern Intel processors
 - Haswell, Coffee Lake, and Cannon Lake
- We measure voltage and current using a Data Acquisition card (NI-DAQ)



Upcoming SAFARI Live Seminar: Oct 27



SAFARI Live Seminar - Data-Centric & Data-Aware Frameworks for Fundamentally Efficient Data Handling

2 waiting • Scheduled for Oct 27, 2021

4 0 SHARE SAVE ...



Onur Mutlu Lectures
19K subscribers

SUBSCRIBED



Title: Data-Centric and Data-Aware Frameworks for Fundamentally Efficient Data Handling in Modern Computing Systems

Speaker: Nastaran Hajinazar, SAFARI Research Group, <https://www.linkedin.com/in/nastaran-...>

More on Our Research & Teaching



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

Applying to Grad School
& Doing Impactful Research

Onur Mutlu
omutlu@gmail.com
<https://people.inf.ethz.ch/omutlu>
13 June 2020
Undergraduate Architecture Mentoring Workshop @ ISCA 2021

Logos for SAFARI, ETH zürich, and Carnegie Mellon are displayed at the bottom of the slide.

Below the video player, the YouTube interface shows:

Arch. Mentoring Workshop @ISCA'21 - Applying to Grad School & Doing Impactful Research - Onur Mutlu
1,563 views • Premiered Jun 16, 2021

Onur Mutlu Lectures
17.2K subscribers

Panel talk at Undergraduate Architecture Mentoring Workshop at ISCA 2021
(<https://sites.google.com/wisc.edu/uar...>)


Engagement icons: 74 likes, 1 comment, SHARE, SAVE, and a menu icon.

Buttons: ANALYTICS, EDIT VIDEO

Open-Source Artifacts

<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.

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

[🏠 Overview](#) [💻 Repositories 55](#) [📦 Packages](#) [👤 People 40](#) [👥 Teams 1](#) [📁 Projects](#) [⚙ Settings](#)


Pinned

Customize your pins

**ramulator** Public ⋮


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

🔴 C++ ☆ 250 🍴 130

**prim-benchmarks** Public ⋮

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

⬛ C ☆ 18 🍴 8

**DAMOV** Public ⋮

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

🔴 C++ ☆ 12 🍴 1

📁 Repositories

Type ▾ Language ▾ Sort ▾ New

Pythia

A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning.

🔴 C++ ☆ 0 🍴 1 🔄 0 📄 0 Updated yesterday

BurstLink

☆ 0 🍴 0 🔄 0 📄 0 Updated 21 days ago

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

25

Repositories 45 Packages People 12 Projects

Find a repository...

Type

Language

Sort

COVIDHunter

COVIDHunter 🦠: An accurate and flexible COVID-19 outbreak simulation model that forecasts the strength of future mitigation measures and the numbers of cases, hospitalizations, and deaths for a given day, while considering the potential effect of environmental conditions. Described by Alser et al. (preliminary version at <https://arxiv.org/abs/2...>

simulation epidemiology covid-19 covid-19-data covid-19-tracker
 reproduction-number covidhunter

Swift MIT 1 5 0 0 Updated 9 hours ago

SNP-Selective-Hiding

An optimization-based mechanism 🧠 to selectively hide the minimum number of overlapping SNPs among the family members 👨 who participated in the genomic studies (i.e. GWAS). Our goal is to distort the dependencies among the family members in the original database for achieving better privacy without significantly degrading the data utility.

gwas genomics data-privacy differential-privacy
 genomic-data-analysis laplace-distribution genomic-privacy

MATLAB 0 0 0 0 Updated 10 hours ago

SneakySnake

SneakySnake 🐍 is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude) expedites sequence alignment calculation for both short and long reads. Described in the Bioinformatics (2020) by Alser et al. <https://arxiv.org/abs...>

fpga gpu smith-waterman needleman-wunsch
 sequence-alignment long-reads minimap2

VHDL GPL-3.0 6 31 0 1 Updated on May 12

ramulator

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 IEEE CAL 2015 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/ramulator_dram_simulator-ieee-cal15.pdf

C++ MIT 121 237 47 4 Updated on May 11

Top languages

C++ C C# AGS Script
 VHDL

Most used topics

dram reliability
 error-correcting-codes
 experimental-data
 pre-alignment-filtering

People

12 >



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

Papers, Talks, Videos, Artifacts

- All are available at

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

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

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

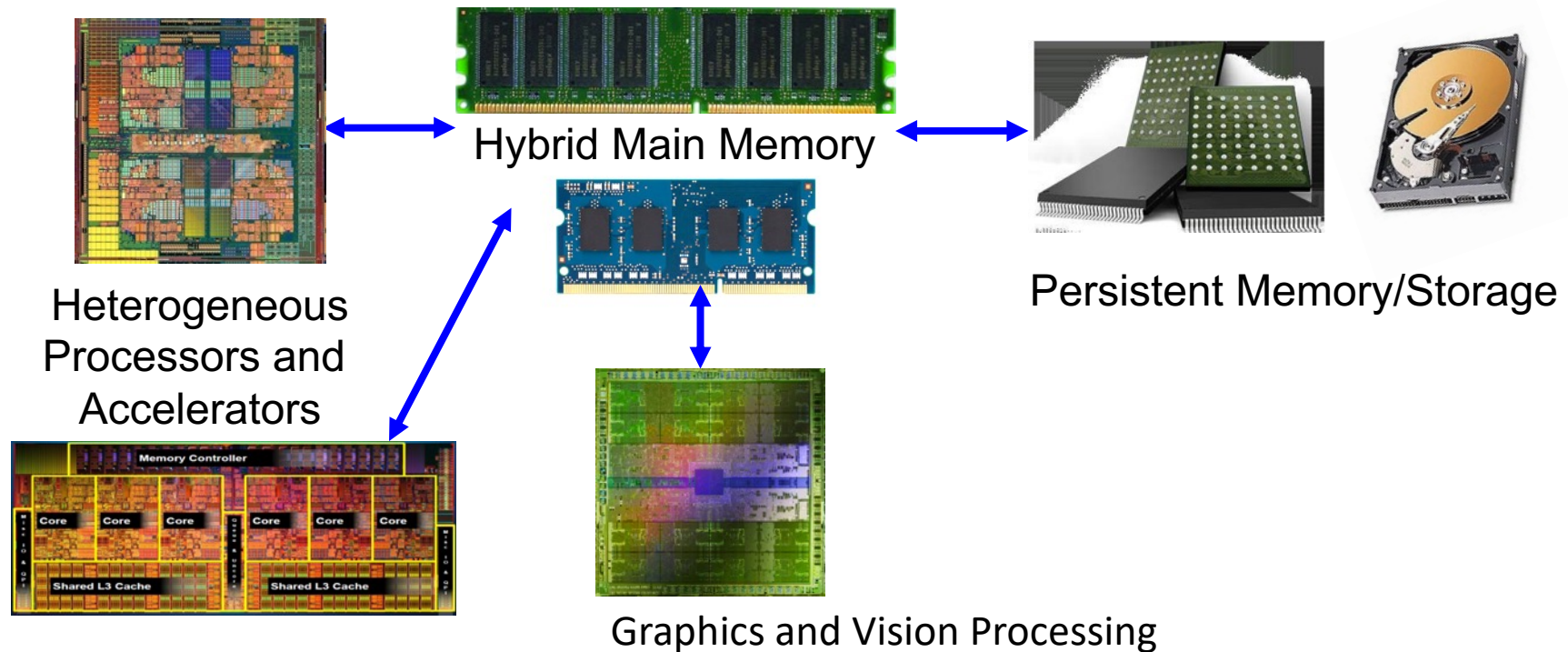
Funding Acknowledgments

- Alibaba, AMD, ASML, [Google](#), Facebook, [Hi-Silicon](#), HP Labs, [Huawei](#), IBM, [Intel](#), [Microsoft](#), Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, [VMware](#)
- NSF
- NIH
- GSRC
- [SRC](#)
- CyLab

Example Research Topics: Quick Overview

Current Research Mission

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



Build fundamentally better architectures

Four Key Issues in Future Platforms

- Fundamentally **Secure/Reliable/Safe** Architectures
- Fundamentally **Energy-Efficient** Architectures
 - **Memory-centric** (Data-centric) Architectures
- Fundamentally **Low-Latency and Predictable** Architectures
- Architectures for **AI/ML, Genomics, Medicine, Health**

Data-centric

Data-driven

Data-aware

Current EFCL Projects

- “A New Methodology and Open-Source Benchmark Suite for Evaluating Data Movement Bottlenecks: A Processing-in-Memory Case Study”
 - Data-centric
- “Machine-Learning-Assisted Intelligent Microarchitectures to Reduce Memory Access Latency”
 - Data-driven
- “Cross-layer Hardware/Software Techniques to Enable Powerful Computation and Memory Optimizations”
 - Data-aware

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

Data is Key for Future Workloads



In-memory Databases

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



In-Memory Data Analytics

[Clapp+ (Intel), IISWC'15;
Awan+, BDCloud'15]



Graph/Tree Processing

[Xu+, IISWC'12; Umuroglu+, FPL'15]



Datacenter Workloads

[Kanev+ (Google), ISCA'15]

Data Overwhelms Modern Machines



In-memory Databases



Graph/Tree Processing

Data → performance & energy bottleneck



In-Memory Data Analytics

[Clapp+ (Intel), IISWC'15;
Awan+, BDCloud'15]



Datacenter Workloads

[Kanev+ (Google), ISCA'15]

Data is Key for Future Workloads



Chrome

Google's web browser



TensorFlow Mobile

Google's machine learning
framework



Video Playback

Google's **video codec**



Video Capture

Google's **video codec**

Data Overwhelms Modern Machines



Chrome



TensorFlow Mobile

Data → performance & energy bottleneck

VP9



Video Playback

Google's **video codec**

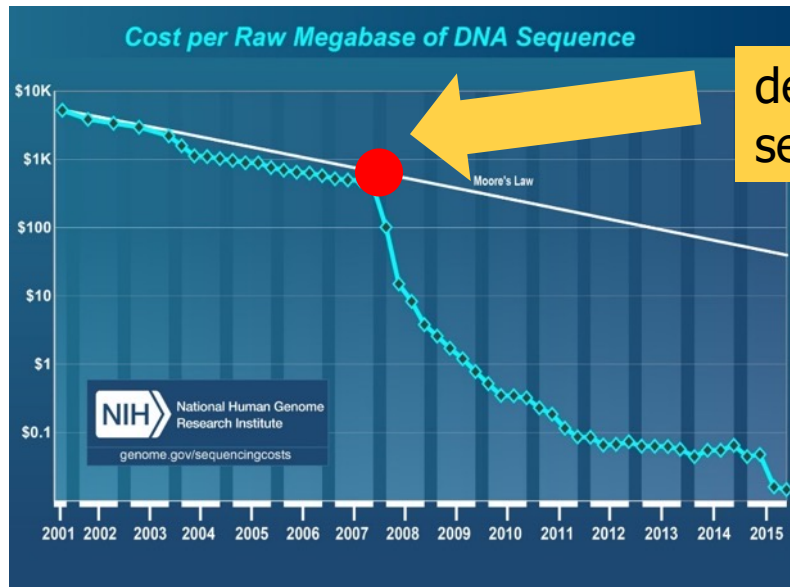
VP9



Video Capture

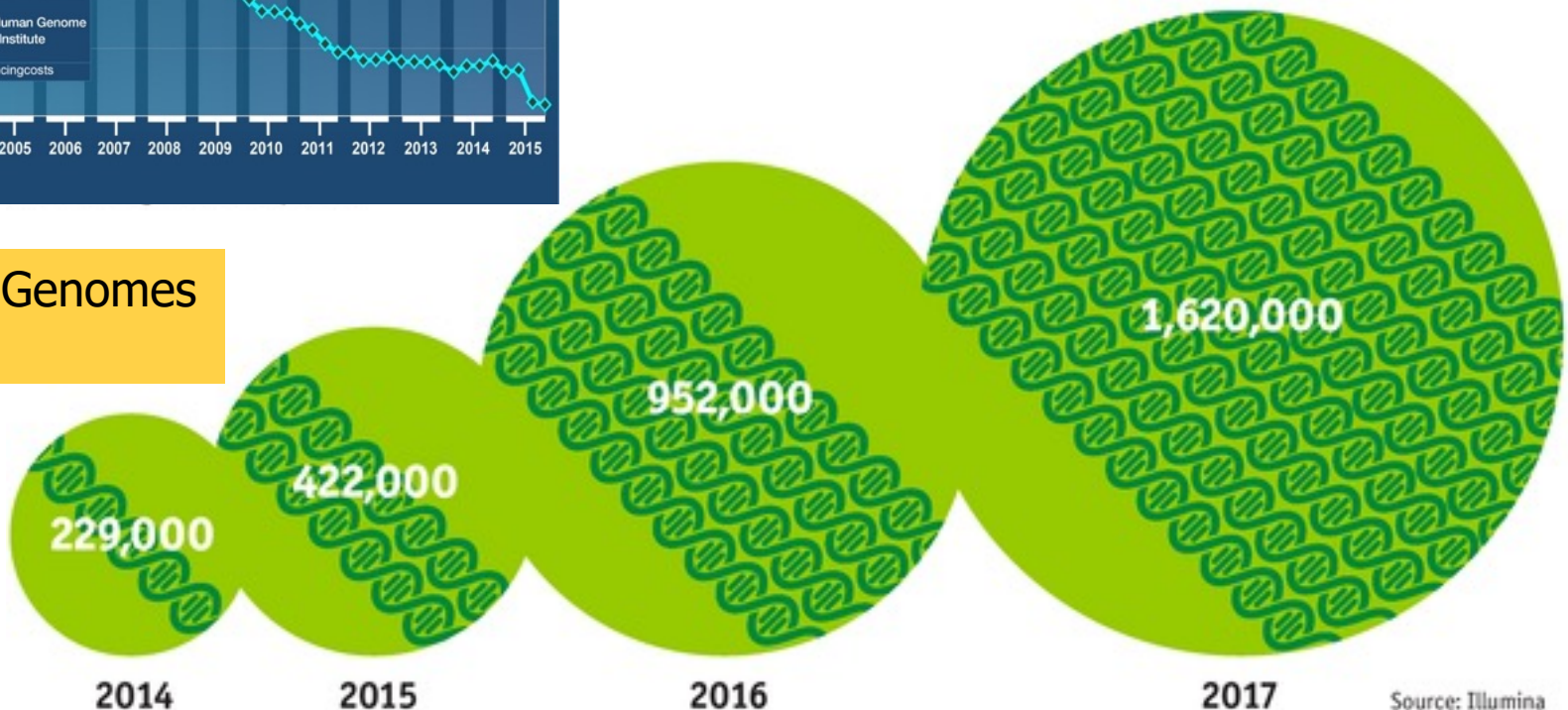
Google's **video codec**

Data is Key for Future Workloads

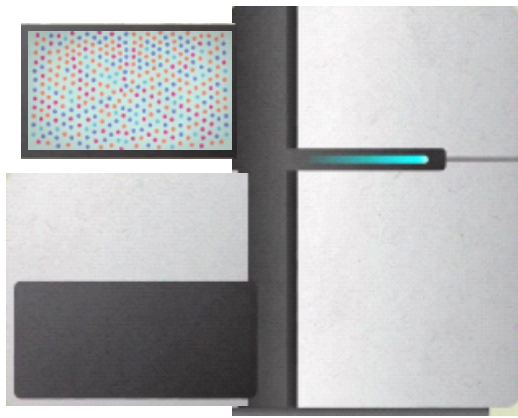


development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced



The Economist



Billions of Short Reads

ATATATACGTACTAGTACGT
 TTTAGTACGTACGT
 ATACGTACTAGTACGT
 CGCCCCTACGTA
 ACGTACTAGTACGT
 TTAGTACGTACGT
 TACGTACTAAAGTACGT
 TACGTACTAGTACGT
 TTTAAACGTA
 CGTACTAGTACGT
 GGGAGTACGTACGT



1 Sequencing

Genome Analysis

2 Read Mapping

Data → performance & energy bottleneck

read4: CGCTTCCAT
 read5: CCATGACGC
 read6: TTCCATGAC



3 Variant Calling

4 Scientific Discovery

New Genome Sequencing Technologies

Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

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

Briefings in Bioinformatics, bby017, <https://doi.org/10.1093/bib/bby017>

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



Oxford Nanopore MinION

Senol Cali+, “**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**,” *Briefings in Bioinformatics*, 2018.

[[Open arxiv.org version](#)]

New Genome Sequencing Technologies

Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

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

Briefings in Bioinformatics, bby017, <https://doi.org/10.1093/bib/bby017>

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



Oxford Nanopore MinION

Data → performance & energy bottleneck

Accelerating Genome Analysis [IEEE MICRO 2020]

- Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#)
[IEEE Micro \(IEEE MICRO\)](#), Vol. 40, No. 5, pages 65-75, September/October 2020.
[\[Slides \(pptx\)\(pdf\)\]](#)
[\[Talk Video \(1 hour 2 minutes\)\]](#)

Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser
ETH Zürich

Zülal Bingöl
Bilkent University

Damla Senol Cali
Carnegie Mellon University

Jeremie Kim
ETH Zurich and Carnegie Mellon University

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

Can Alkan
Bilkent University

Onur Mutlu
ETH Zurich, Carnegie Mellon University, and
Bilkent University

GenASM Framework [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zülal Bingöl, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"**
Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.
[[Lightning Talk Video](#) (1.5 minutes)]
[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (18 minutes)]
[[Slides \(pptx\)](#) ([pdf](#))]

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

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

FPGA-based Processing Near Memory

- Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gómez-Luna, Henk Corporaal, and Onur Mutlu, ["FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications"](#) *IEEE Micro* (**IEEE MICRO**), 2021.

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

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

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

Henk Corporaal[★] Onur Mutlu^{◇✕}

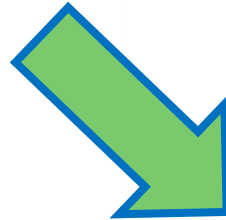
[◇]*ETH Zürich* [✕]*Carnegie Mellon University*

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

Future of Genome Sequencing & Analysis



MinION from ONT



SmidgION from ONT

More on Fast & Efficient Genome Analysis ...

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



Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

740 views • Premiered Feb 6, 2021

35 0 SHARE SAVE ...

SAFARI



Onur Mutlu Lectures
15.9K subscribers

ANALYTICS

EDIT VIDEO

Detailed Lectures on Genome Analysis

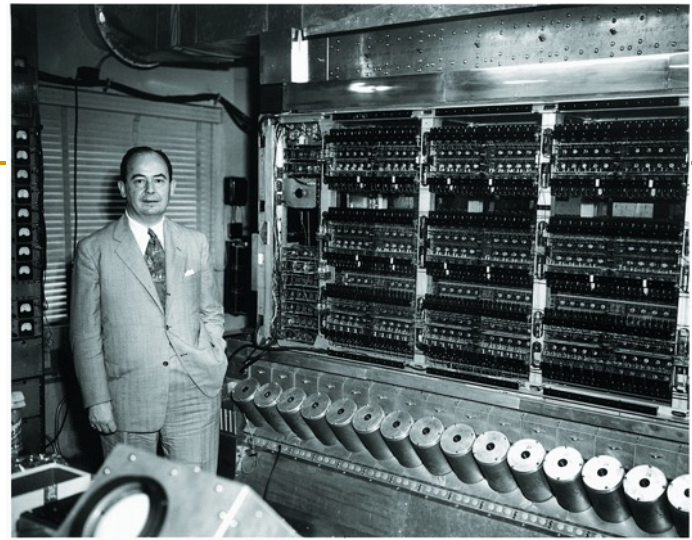
- **Computer Architecture, Fall 2020, Lecture 3a**
 - **Introduction to Genome Sequence Analysis** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5>
- **Computer Architecture, Fall 2020, Lecture 8**
 - **Intelligent Genome Analysis** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14>
- **Computer Architecture, Fall 2020, Lecture 9a**
 - **GenASM: Approx. String Matching Accelerator** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15>
- **Accelerating Genomics Project Course, Fall 2020, Lecture 1**
 - **Accelerating Genomics** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId>

Data Overwhelms Modern Machines ...

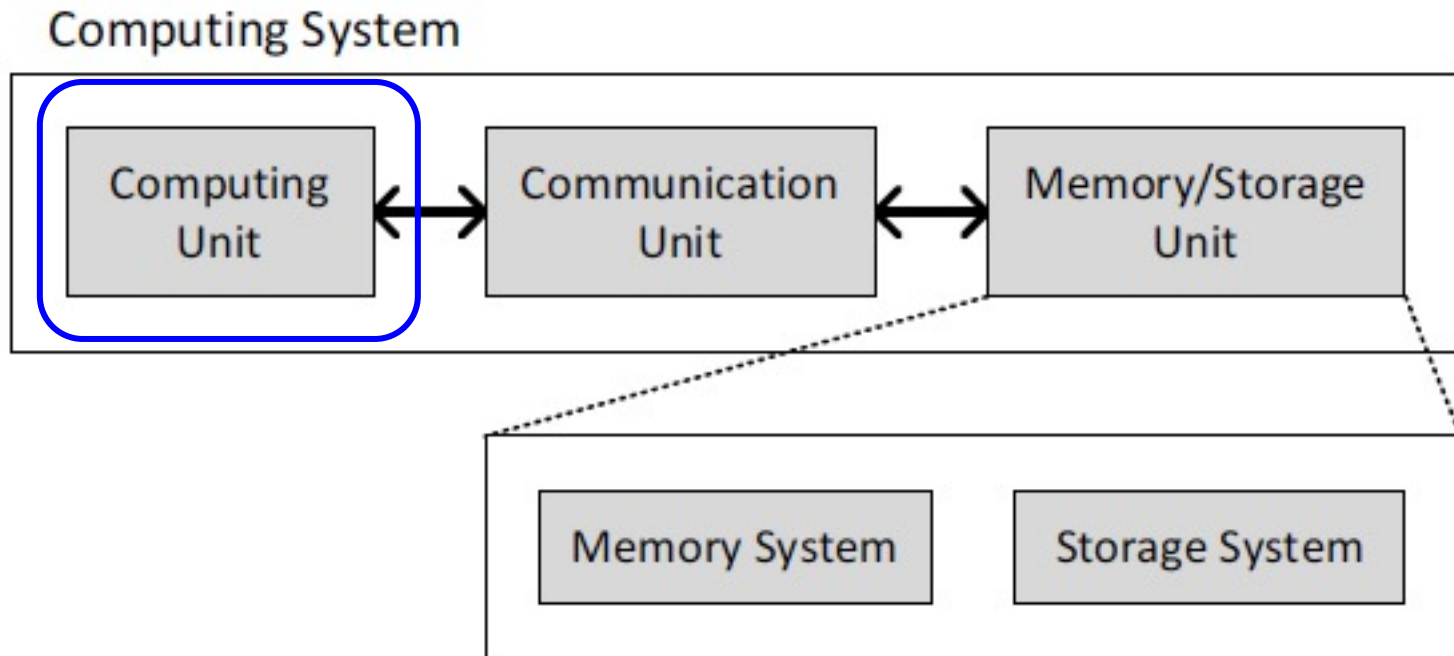
- Storage/memory capability
- Communication capability
- Computation capability
- Greatly impacts robustness, energy, performance, cost

A Computing System

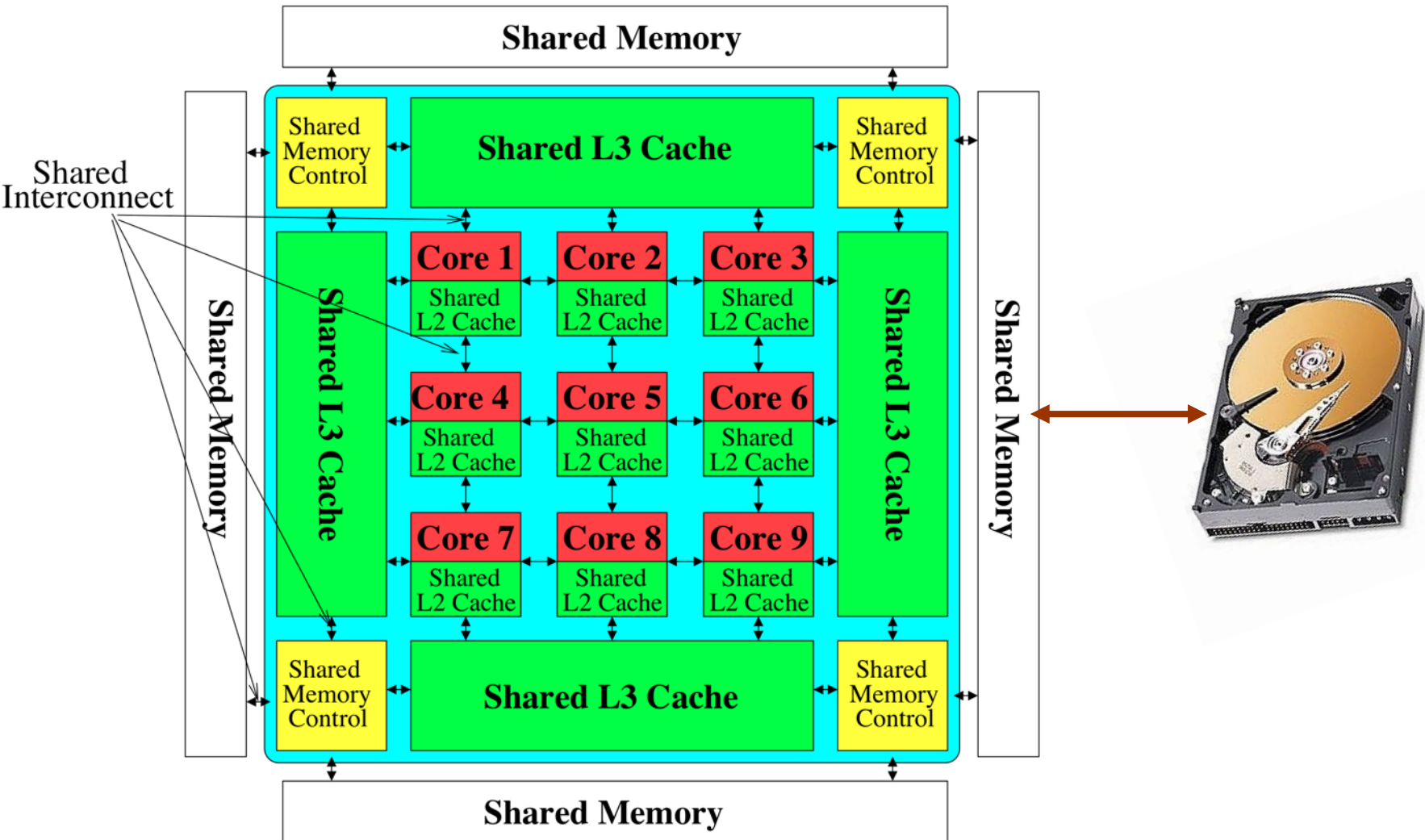
- Three key components
- Computation
- Communication
- Storage/memory



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



Perils of Processor-Centric Design



Most of the system is dedicated to storing and moving data

Yet, system is still bottlenecked by memory

Data Overwhelms Modern Machines



Chrome



TensorFlow Mobile

Data → performance & energy bottleneck

VP9



Video Playback

Google's **video codec**

VP9



Video Capture

Google's **video codec**

Data Movement Overwhelms Modern Machines

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

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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

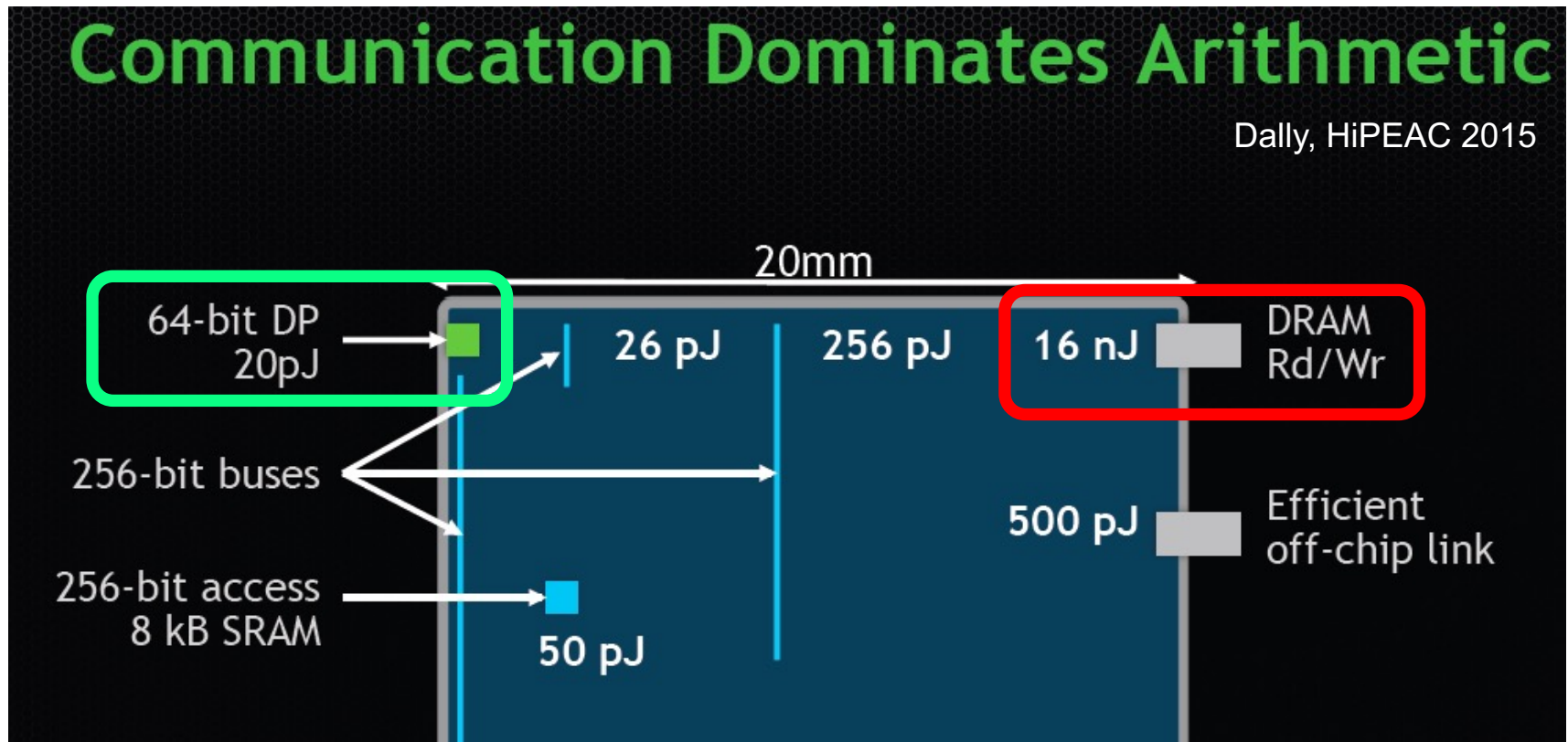
Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

Data Movement vs. Computation Energy

Communication Dominates Arithmetic

Dally, HiPEAC 2015



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

An Intelligent Architecture Handles Data Well

How to Handle Data Well

- Ensure data does not overwhelm the components
 - via intelligent algorithms
 - via intelligent architectures
 - via whole system designs: algorithm-architecture-devices
- Take advantage of vast amounts of data and metadata
 - to improve architectural & system-level decisions
- Understand and exploit properties of (different) data
 - to improve algorithms & architectures in various metrics

Corollaries: Architectures Today ...

- Architectures are **terrible at dealing with data**
 - ❑ Designed to mainly store and move data vs. to compute
 - ❑ They are **processor-centric** as opposed to **data-centric**
- Architectures are **terrible at taking advantage of vast amounts of data** (and metadata) available to them
 - ❑ Designed to make simple decisions, ignoring lots of data
 - ❑ They make **human-driven decisions vs. data-driven**
- Architectures are **terrible at knowing and exploiting different properties of application data**
 - ❑ Designed to treat all data as the same
 - ❑ They make **component-aware decisions vs. data-aware**

Data-Centric (Memory-Centric) Architectures

Data-Centric Architectures: Properties

- **Process data where it resides** (where it makes sense)
 - Processing in and near memory structures
- **Low-latency and low-energy data access**
 - Low latency memory
 - Low energy memory
- **Low-cost data storage and processing**
 - High capacity memory at low cost: hybrid memory, compression
- **Intelligent data management**
 - Intelligent controllers handling robustness, security, cost

Processing Data Where It Makes Sense

The Problem

Data access is the major performance and energy bottleneck

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

The Problem

Processing of data
is performed
far away from the data

We Need A Paradigm Shift To ...

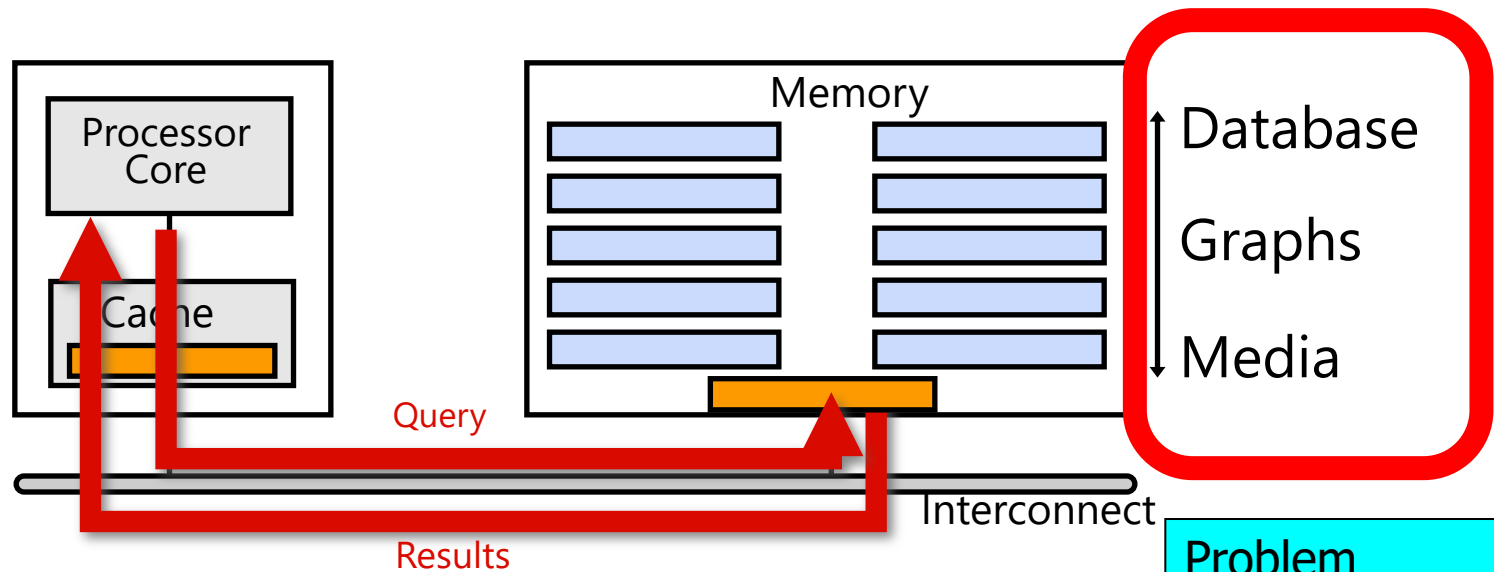
- Enable computation with minimal data movement
- Compute where it makes sense (where data resides)
- Make computing architectures more data-centric

Computing Architectures with Minimal Data Movement

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

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

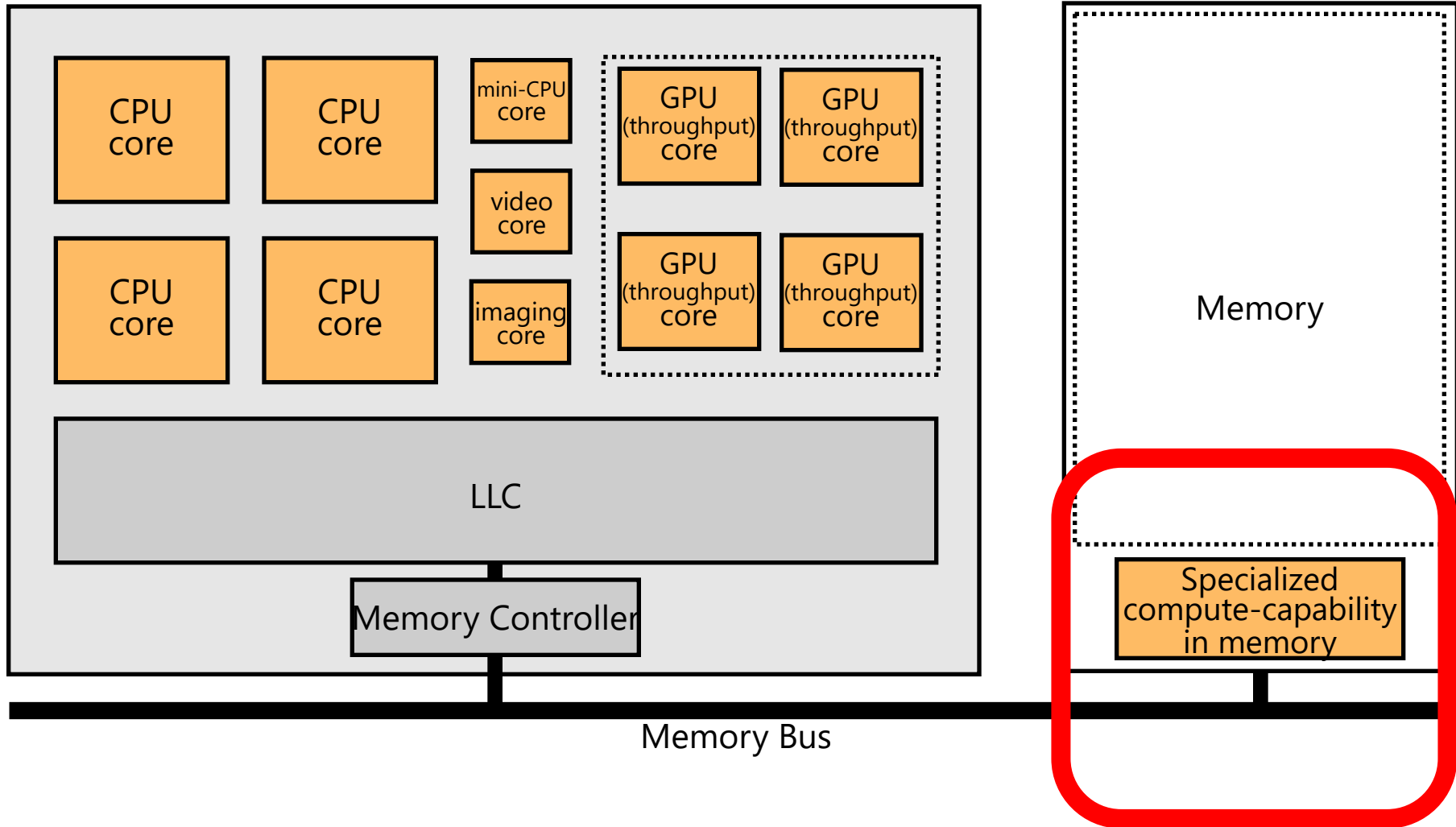
Goal: Processing Inside Memory



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

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

Mindset: Memory-Centric Computing



Memory similar to a “conventional” accelerator

PIM Review and Open Problems

A Modern Primer on Processing in Memory

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

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

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

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

"A Modern Primer on Processing in Memory"

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

A Modern Primer on Processing in Memory

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

SAFARI Research Group

^aETH Zürich

^bCarnegie Mellon University

^cUniversity of Illinois at Urbana-Champaign

^dKing Mongkut's University of Technology North Bangkok

Abstract

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

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

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

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

1 Introduction	2
2 Major Trends Affecting Main Memory	4
3 The Need for Intelligent Memory Controllers to Enhance Memory Scaling	6
4 Perils of Processor-Centric Design	9
5 Processing-in-Memory (PIM): Technology Enablers and Two Approaches	12
5.1 New Technology Enablers: 3D-Stacked Memory and Non-Volatile Memory . . .	12
5.2 Two Approaches: Processing Using Memory (PUM) vs. Processing Near Memory (PNM)	13
6 Processing Using Memory (PUM)	14
6.1 RowClone	14
6.2 Ambit	15
6.3 Gather-Scatter DRAM	17
6.4 In-DRAM Security Primitives	17
7 Processing Near Memory (PNM)	18
7.1 Tesseract: Coarse-Grained Application-Level PNM Acceleration of Graph Processing	19
7.2 Function-Level PNM Acceleration of Mobile Consumer Workloads	20
7.3 Programmer-Transparent Function-Level PNM Acceleration of GPU Applications	21
7.4 Instruction-Level PNM Acceleration with PIM-Enabled Instructions (PEI) . .	21
7.5 Function-Level PNM Acceleration of Genome Analysis Workloads	22
7.6 Application-Level PNM Acceleration of Time Series Analysis	23
8 Enabling the Adoption of PIM	24
8.1 Programming Models and Code Generation for PIM	24
8.2 PIM Runtime: Scheduling and Data Mapping	25
8.3 Memory Coherence	27
8.4 Virtual Memory Support	27
8.5 Data Structures for PIM	28
8.6 Benchmarks and Simulation Infrastructures	29
8.7 Real PIM Hardware Systems and Prototypes	30
8.8 Security Considerations	30
9 Conclusion and Future Outlook	31

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

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

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

PIM Review and Open Problems (II)

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

Saugata Ghose[†] Amirali Boroumand[†] Jeremie S. Kim^{†§} Juan Gómez-Luna[§] Onur Mutlu^{§†}

[†]*Carnegie Mellon University*

[§]*ETH Zürich*

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu,

"Processing-in-Memory: A Workload-Driven Perspective"

Invited Article in IBM Journal of Research & Development, Special Issue on Hardware for Artificial Intelligence, to appear in November 2019.

[Preliminary arXiv version]

Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory

Processing using Memory

- We can support in-DRAM AND, OR, NOT, MAJ
- At low cost
- Using analog computation capability of DRAM
 - Idea: activating multiple rows performs computation
- 30-60X performance and energy improvement
 - Seshadri+, “Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology,” MICRO 2017.
- New memory technologies enable even more opportunities
 - Memristors, resistive RAM, phase change mem, STT-MRAM, ...
 - Can operate on data with minimal movement

Ambit: Bulk-Bitwise in-DRAM Computation

- Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,
"Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology"
Proceedings of the 50th International Symposium on Microarchitecture (MICRO), Boston, MA, USA, October 2017.
[[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Session Slides \(pptx\)](#)] [[pdf](#)] [[Poster \(pptx\)](#)] [[pdf](#)]

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

Vivek Seshadri^{1,5} Donghyuk Lee^{2,5} Thomas Mullins^{3,5} Hasan Hassan⁴ Amirali Boroumand⁵
Jeremie Kim^{4,5} Michael A. Kozuch³ Onur Mutlu^{4,5} Phillip B. Gibbons⁵ Todd C. Mowry⁵

¹Microsoft Research India ²NVIDIA Research ³Intel ⁴ETH Zürich ⁵Carnegie Mellon University

In-DRAM Bulk Bitwise Execution Paradigm

- Vivek Seshadri and Onur Mutlu,
"In-DRAM Bulk Bitwise Execution Engine"
Invited Book Chapter in Advances in Computers, 2020.
[Preliminary arXiv version]

In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri
Microsoft Research India
visesha@microsoft.com

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

SIMDRAM Framework

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu, [**"SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM"**](#) *Proceedings of the 26th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Virtual, March-April 2021.
[[2-page Extended Abstract](#)]
[[Short Talk Slides \(pptx\)](#) ([pdf](#))]
[[Talk Slides \(pptx\)](#) ([pdf](#))]
[[Short Talk Video](#) (5 mins)]
[[Full Talk Video](#) (27 mins)]

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

*Nastaran Hajinazar ^{1,2}	*Geraldo F. Oliveira ¹	Sven Gregorio ¹	João Dinis Ferreira ¹
Nika Mansouri Ghiasi ¹	Minesh Patel ¹	Mohammed Alser ¹	Saugata Ghose ³
	Juan Gómez-Luna ¹	Onur Mutlu ¹	

¹ETH Zürich

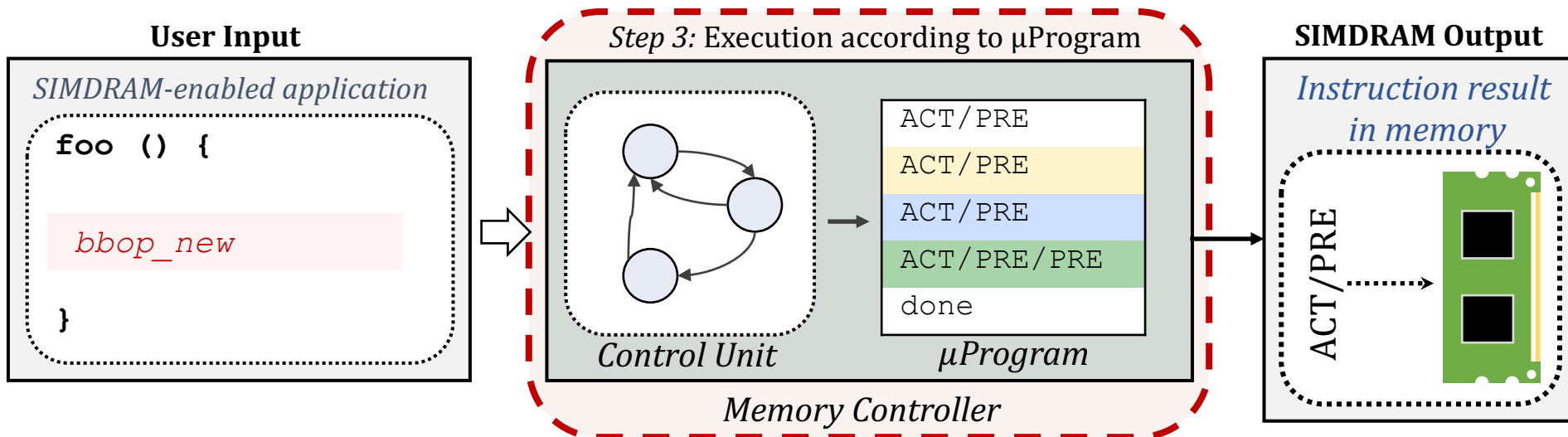
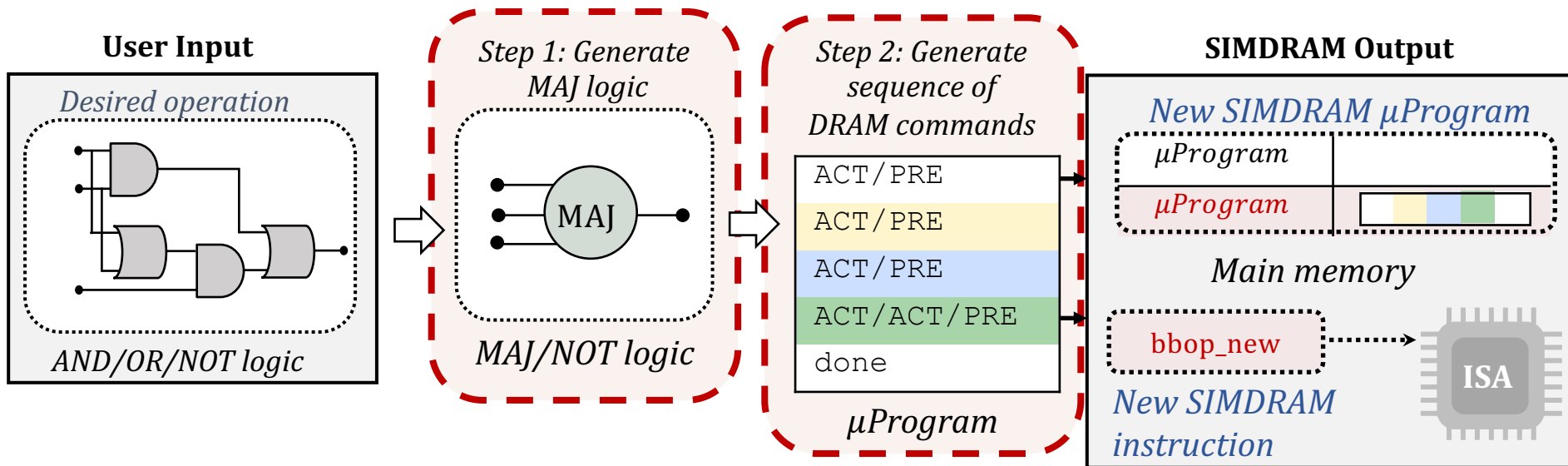
²Simon Fraser University

³University of Illinois at Urbana–Champaign

SIMDRAM Key Idea

- **SIMDRAM**: An end-to-end processing-using-DRAM framework that provides the **programming interface**, the **ISA**, and the **hardware support** for:
 - **Efficiently** computing **complex** operations in DRAM
 - Providing the ability to implement **arbitrary** operations as required
 - Using an **in-DRAM massively-parallel SIMD substrate** that requires **minimal** changes to DRAM architecture

SIMDRAM Framework: Overview



SIMDRAM Key Results

Evaluated on:

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

SIMDRAM provides:

- **88×** and **5.8×** the **throughput** of a **CPU** and a **high-end GPU**, respectively, over **16 operations**
- **257×** and **31×** the **energy efficiency** of a **CPU** and a **high-end GPU**, respectively, over **16 operations**
- **21×** and **2.1×** the **performance** of a **CPU** and a **high-end GPU**, over **seven real-world applications**

SIMDRAM Conclusion

- **SIMDRAM:**

- Enables **efficient** computation of a **flexible** set and wide range of operations in a PuM **massively parallel** SIMD substrate
- Provides the hardware, programming, and ISA support, to:
 - Address key **system integration** challenges
 - Allow programmers to define and employ **new operations** without hardware changes

SIMDRAM is a promising PuM framework

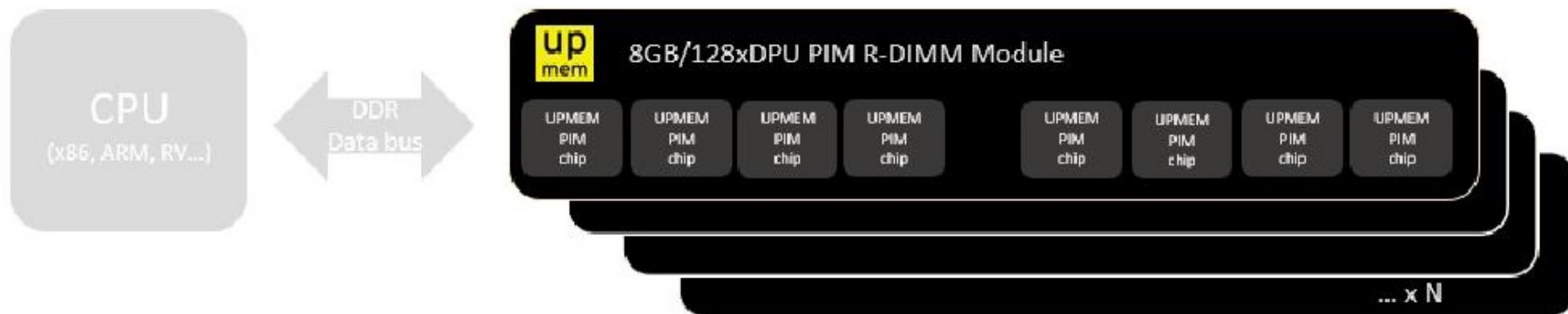
- Can **ease the adoption** of processing-using-DRAM architectures
- Improves the **performance** and **efficiency** of processing-using-memory architectures

Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory

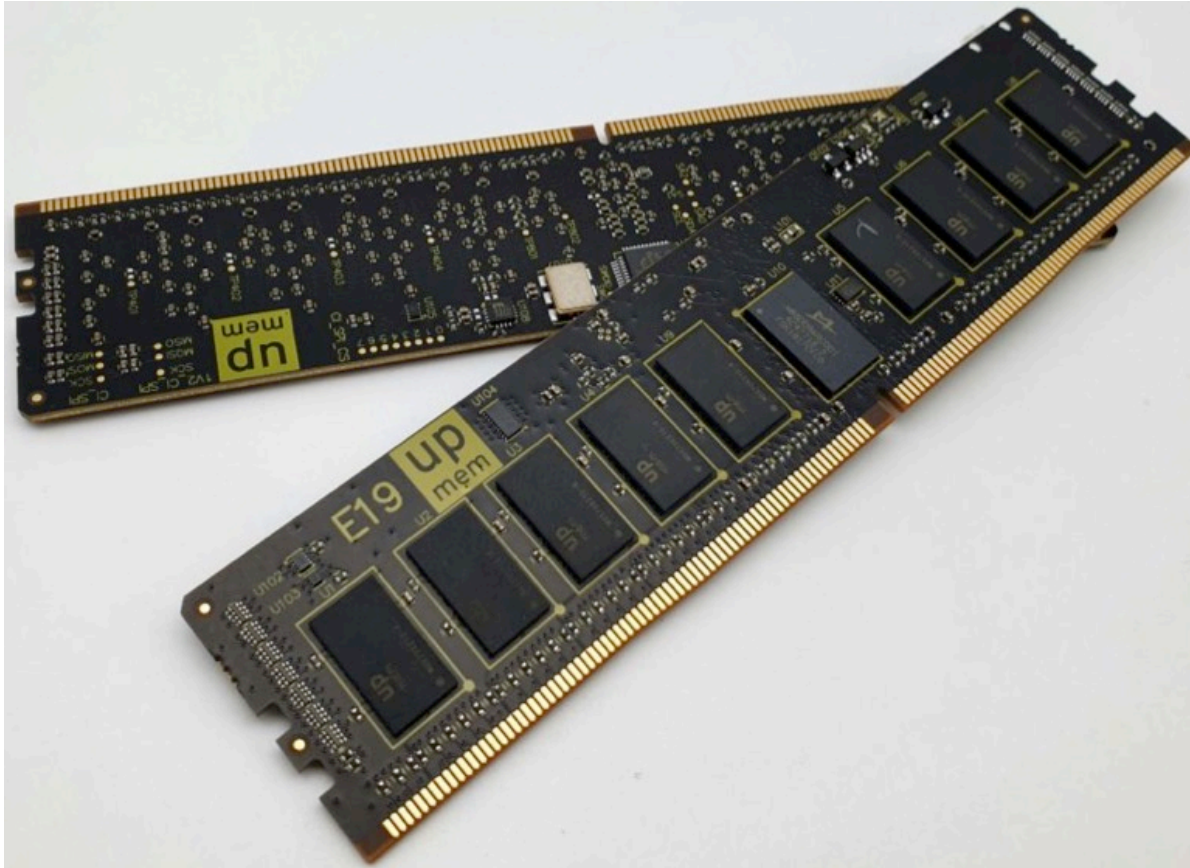
UPMEM Processing-in-DRAM Engine (2019)

- **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

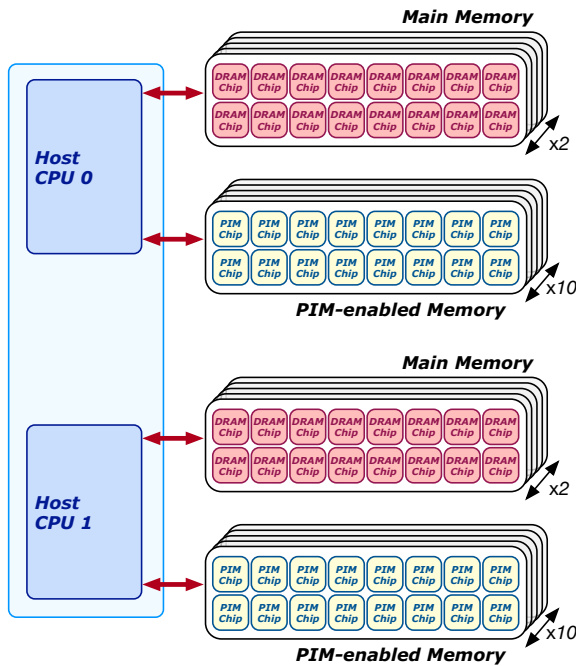


UPMEM Memory Modules

- 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 FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
 CHRISTINA GIANNOULA, ETH Zürich, Switzerland and NTUA, Greece
 GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
 ONUR MUTLU, ETH Zürich, Switzerland

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

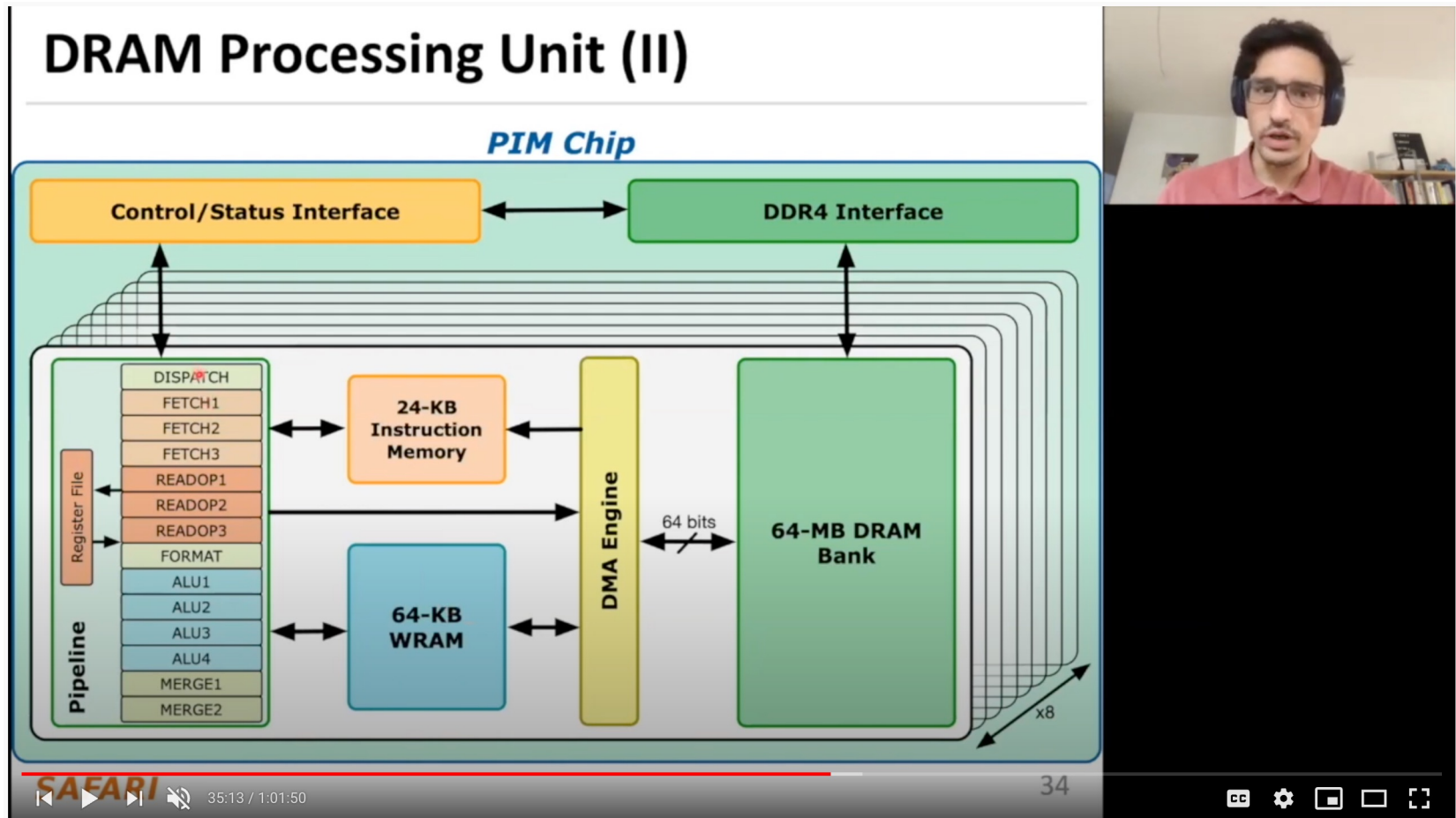
Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware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

This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present *PrIM (Processing-In-Memory benchmarks)*, a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 640 and 2,560 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.



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

More on the UPMEM PIM System



ETH ZÜRICH HAUPTGEBÄUDE

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

1,120 views • Oct 31, 2020

30 0 SHARE SAVE ...



Onur Mutlu Lectures
16.7K subscribers

ANALYTICS

EDIT VIDEO

<https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=26>

Experimental Analysis of the UPMEM PIM Engine

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

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

IZZAT EL HAJJ, American University of Beirut, Lebanon

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

CHRISTINA GIANNOULA, ETH Zürich, Switzerland and NTUA, Greece

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

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

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

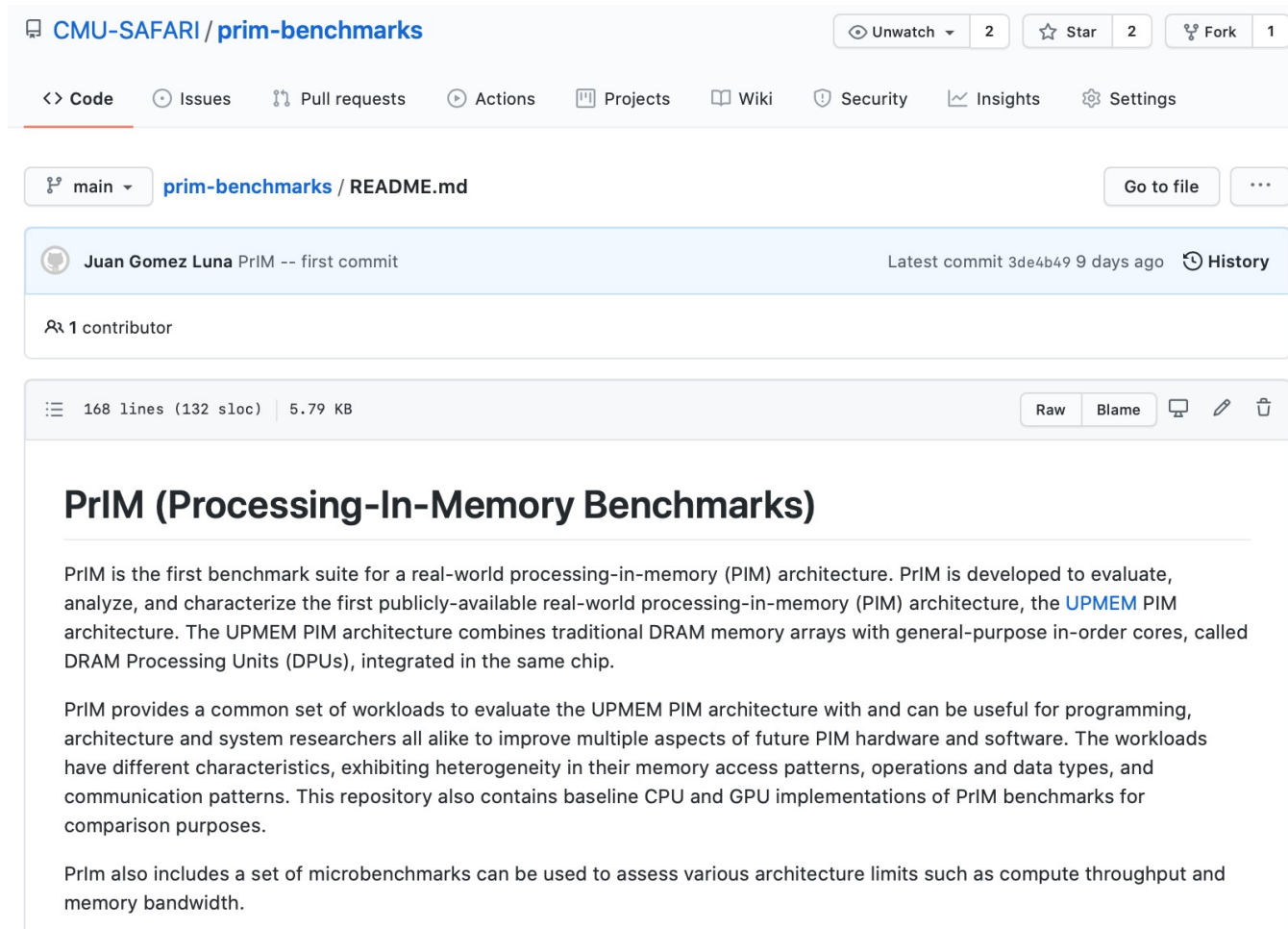
This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute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

PrIM Benchmarks: Application Domains

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

PrIM Benchmarks are Open Source

- All microbenchmarks, benchmarks, and scripts
- <https://github.com/CMU-SAFARI/prim-benchmarks>



CMU-SAFARI / **prim-benchmarks** Unwatch 2 Star 2 Fork 1

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main prim-benchmarks / README.md Go to file ...

Juan Gomez Luna PrIM -- first commit Latest commit 3de4b49 9 days ago History

1 contributor

168 lines (132 sloc) 5.79 KB Raw Blame

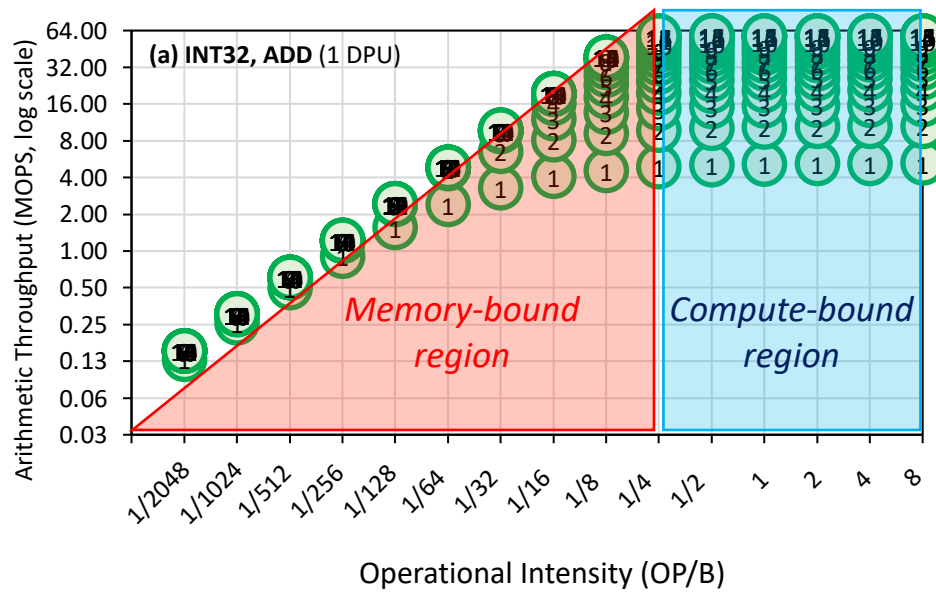
PrIM (Processing-In-Memory Benchmarks)

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.

Key Takeaway 1

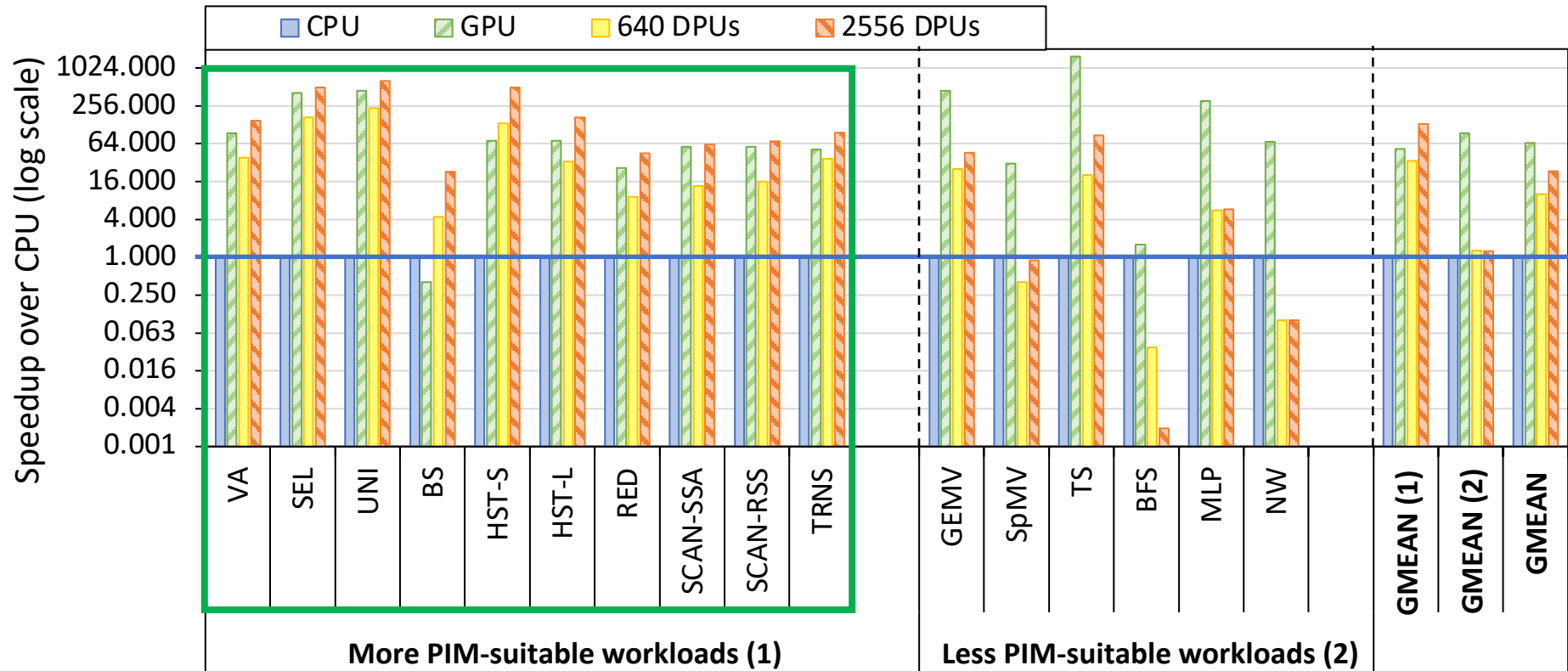


The throughput saturation point is as low as $\frac{1}{4}$ OP/B, i.e., 1 integer addition per every 32-bit element fetched

KEY TAKEAWAY 1

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

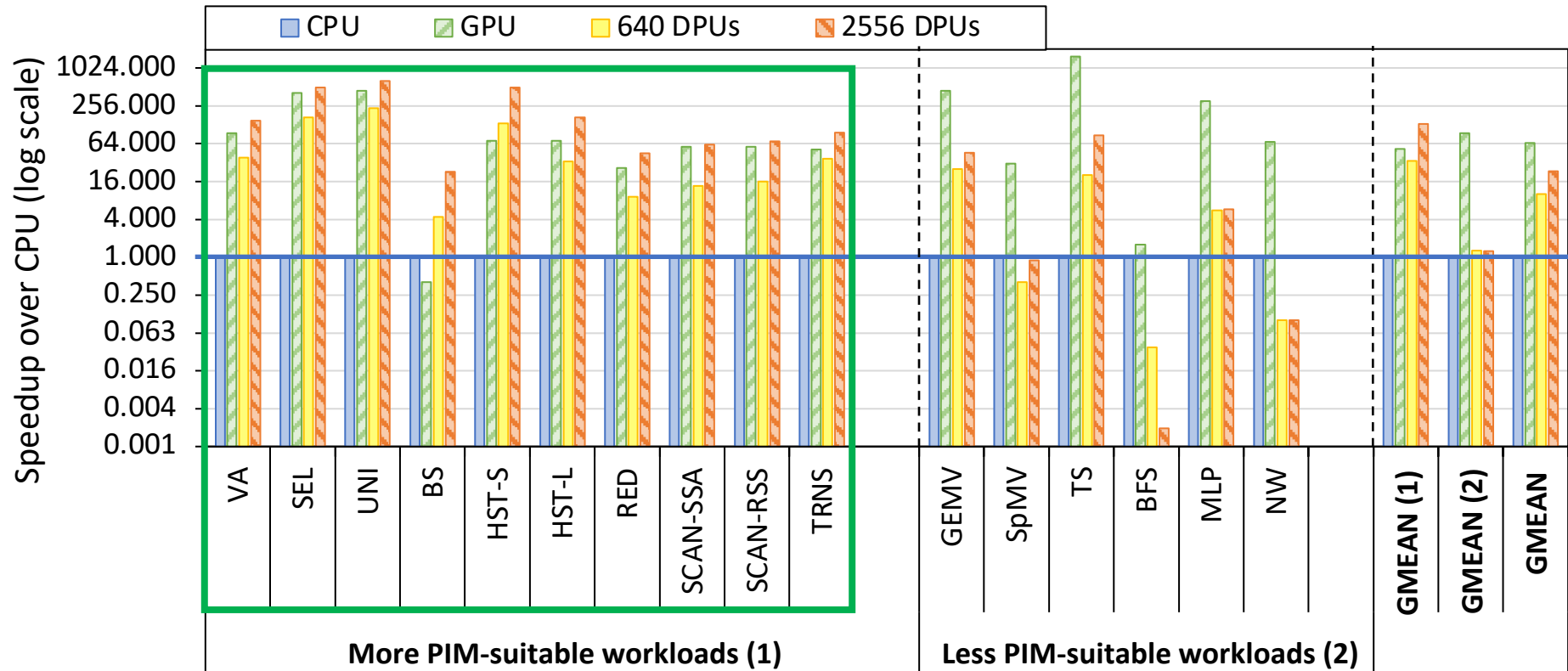
Key Takeaway 2



KEY TAKEAWAY 2

The most well-suited workloads for the UPMEM PIM architecture use no arithmetic operations or use only simple operations (e.g., bitwise operations and integer addition/subtraction).

Key Takeaway 3



KEY TAKEAWAY 3

The most well-suited workloads for the UPMEM PIM architecture require little or no communication across DPUs (inter-DPU communication).

Key Takeaway 4

KEY TAKEAWAY 4

- UPMEM-based PIM systems **outperform state-of-the-art CPUs in terms of performance and energy efficiency on most of PrIM benchmarks.**
- UPMEM-based PIM systems **outperform state-of-the-art GPUs on a majority of PrIM benchmarks**, and the outlook is even more positive for future PIM systems.
- UPMEM-based PIM systems are **more energy-efficient than state-of-the-art CPUs and GPUs on workloads that they provide performance improvements** over the CPUs and the GPUs.

More on UPMEM System & Analysis

- Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu, [**"Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture"**](#)
Preprint in [arXiv](#), 9 May 2021.
[\[arXiv preprint\]](#)
[\[PrIM Benchmarks Source Code\]](#)
[\[Slides \(pptx\) \(pdf\)\]](#)
[\[Long Talk Slides \(pptx\) \(pdf\)\]](#)
[\[Short Talk Slides \(pptx\) \(pdf\)\]](#)
[\[SAFARI Live Seminar Slides \(pptx\) \(pdf\)\]](#)
[\[SAFARI Live Seminar Video \(2 hrs 57 mins\)\]](#)
[\[Lightning Talk Video \(3 minutes\)\]](#)
[\[Short Talk Video \(21 minutes\)\]](#)
[\[1-hour Talk Video \(58 minutes\)\]](#)

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

Juan Gómez-Luna¹ Izzat El Hajj² Ivan Fernandez^{1,3} Christina Giannoula^{1,4}
Geraldo F. Oliveira¹ Onur Mutlu¹

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

Understanding a Modern PIM Architecture



The video player shows a lecture titled "Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization". The speaker is Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu. The video is from the "SAFARI Live Seminar" series. The player includes a progress bar at 2:26 / 2:57:10, a volume icon, and a "SAFARI" logo. The video description below the player states: "SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture", "2,579 views • Streamed live on Jul 12, 2021", and "Onur Mutlu Lectures 18.7K subscribers". The video player interface includes a progress bar at 2:26 / 2:57:10, a volume icon, and a "SAFARI" logo. The video description below the player states: "SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture", "2,579 views • Streamed live on Jul 12, 2021", and "Onur Mutlu Lectures 18.7K subscribers".

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

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

<https://arxiv.org/pdf/2105.03814.pdf>
<https://github.com/CMU-SAFARI/prim-benchmarks>

ETH Zürich SAFARI

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

2,579 views • Streamed live on Jul 12, 2021

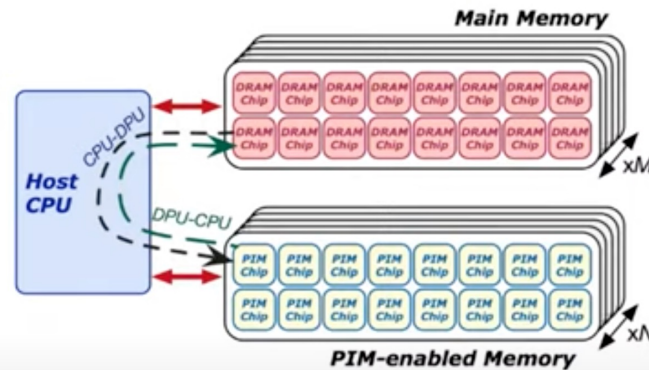
Onur Mutlu Lectures
18.7K subscribers

SUBSCRIBED

More on Analysis of the UPMEM PIM Engine

Inter-DPU Communication

- There is **no direct communication channel between DPUs**



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



Onur Mutlu

zoom

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

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

81 0 SHARE SAVE ...



Onur Mutlu Lectures
17.6K subscribers

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

ANALYTICS

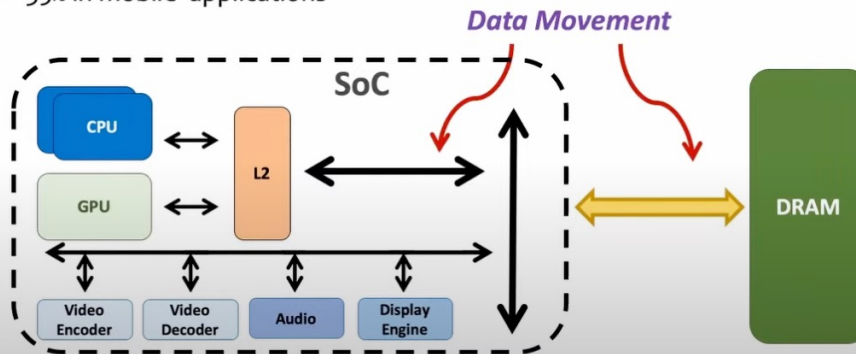
EDIT VIDEO

https://www.youtube.com/watch?v=D8Hjy2IU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9

More on Analysis of the UPMEM PIM Engine

Data Movement in Computing Systems

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



* Boroumand et al., "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS 2018

* Kestor et al., "Quantifying the Energy Cost of Data Movement in Scientific Applications," IISWC 2013

* Pandiyan and Wu, "Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms," IISWC 2014

SAFARI

3

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

3,482 views • Premiered Jul 25, 2021

38 0 SHARE SAVE ...



Onur Mutlu Lectures

17.9K subscribers

ANALYTICS

EDIT VIDEO

https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXY2Zi8_VVChACnON4sfh2bJ5IrD&index=159

FPGA-based Processing Near Memory

- Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gómez-Luna, Henk Corporaal, and Onur Mutlu, ["FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications"](#) *IEEE Micro* (**IEEE MICRO**), to appear, 2021.

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

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

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

Henk Corporaal^{*} Onur Mutlu^{◇✕}

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

^{*}*Eindhoven University of Technology* [▽]*IBM Research Europe*

DAMOV Analysis Methodology & Workloads

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

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

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

NANDITA VIJAYKUMAR, University of Toronto, Canada

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

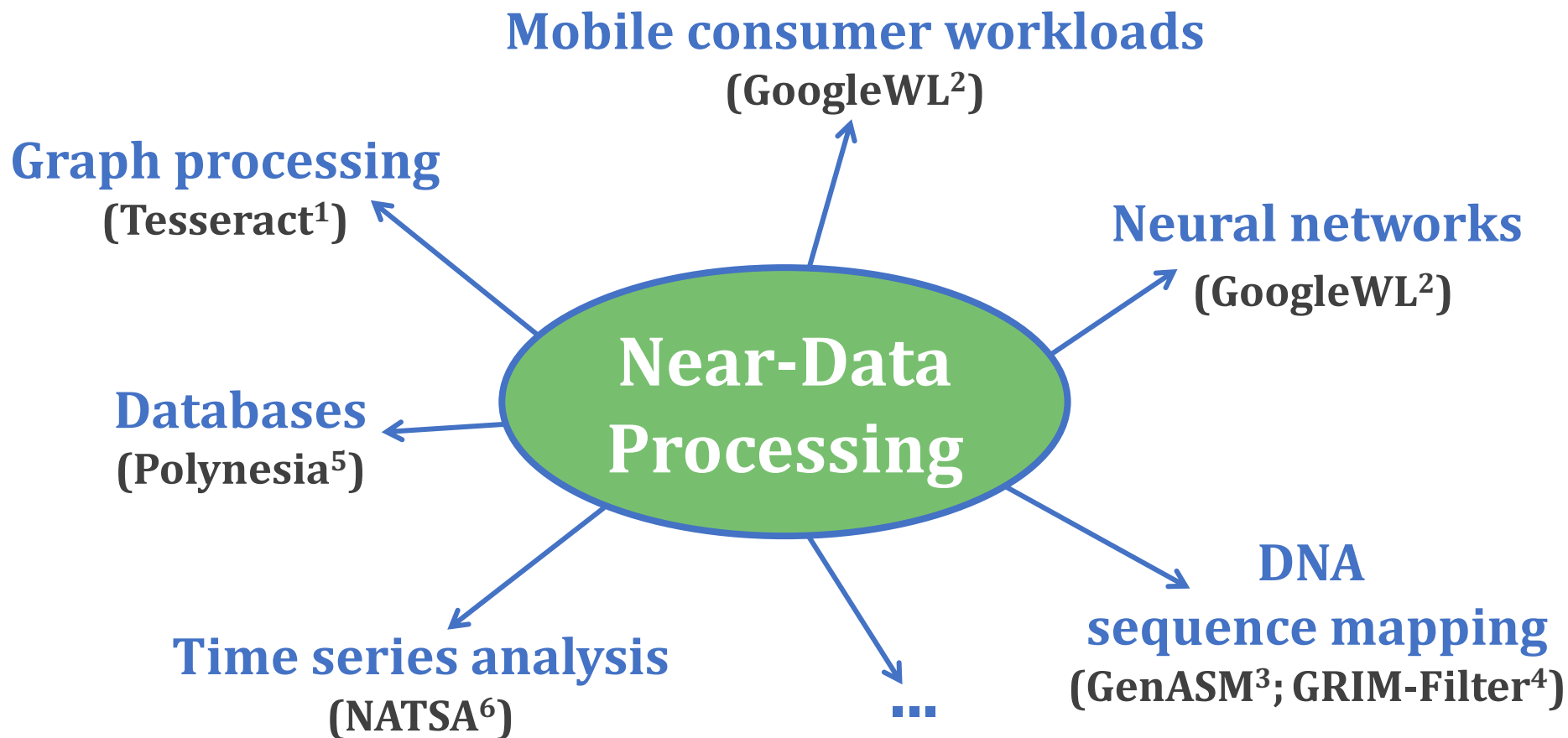
MOHAMMAD SADROSADATI, Institute for Research in Fundamental Sciences (IPM), Iran & ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

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

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

When to Employ Near-Data Processing?



[1] Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing," ISCA, 2015

[2] Boroumand+, "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS, 2018

[3] Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis," MICRO, 2020

[4] Kim+, "GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies," BMC Genomics, 2018

[5] Boroumand+, "Polynesia: Enabling Effective Hybrid Transactional/Analytical Databases with Specialized Hardware/Software Co-Design," arXiv:2103.00798 [cs.AR], 2021

[6] Fernandez+, "NATSA: A Near-Data Processing Accelerator for Time Series Analysis," ICCD, 2020

Key Approach

- New **workload characterization methodology** to analyze:
 - data movement bottlenecks
 - suitability of different data movement mitigation mechanisms
- Two main profiling strategies:

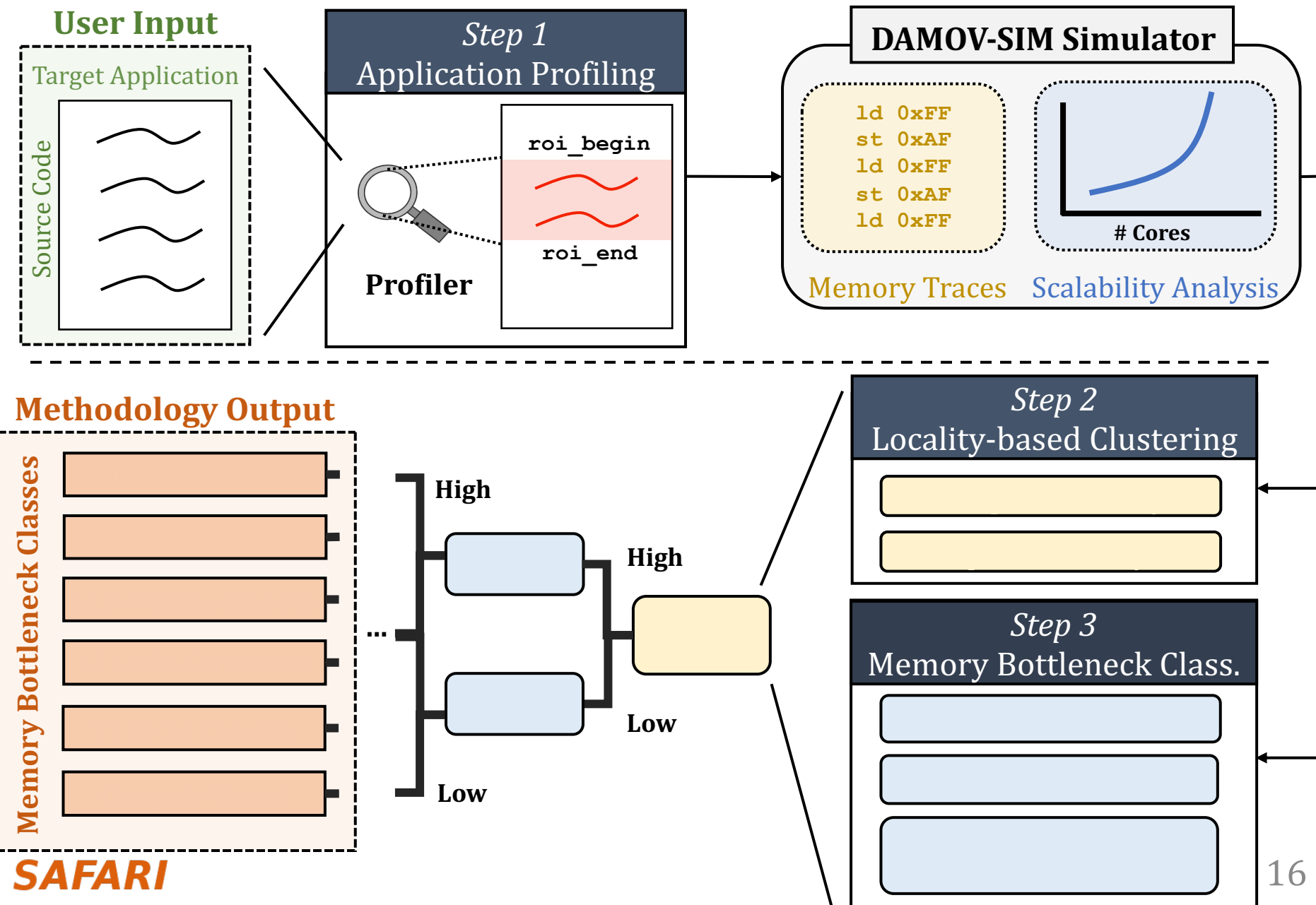
Architecture-independent profiling:

characterizes the memory behavior **independently**
of the underlying **hardware**

Architecture-dependent profiling:

evaluates the **impact of the system configuration**
on the memory behavior

Methodology Overview



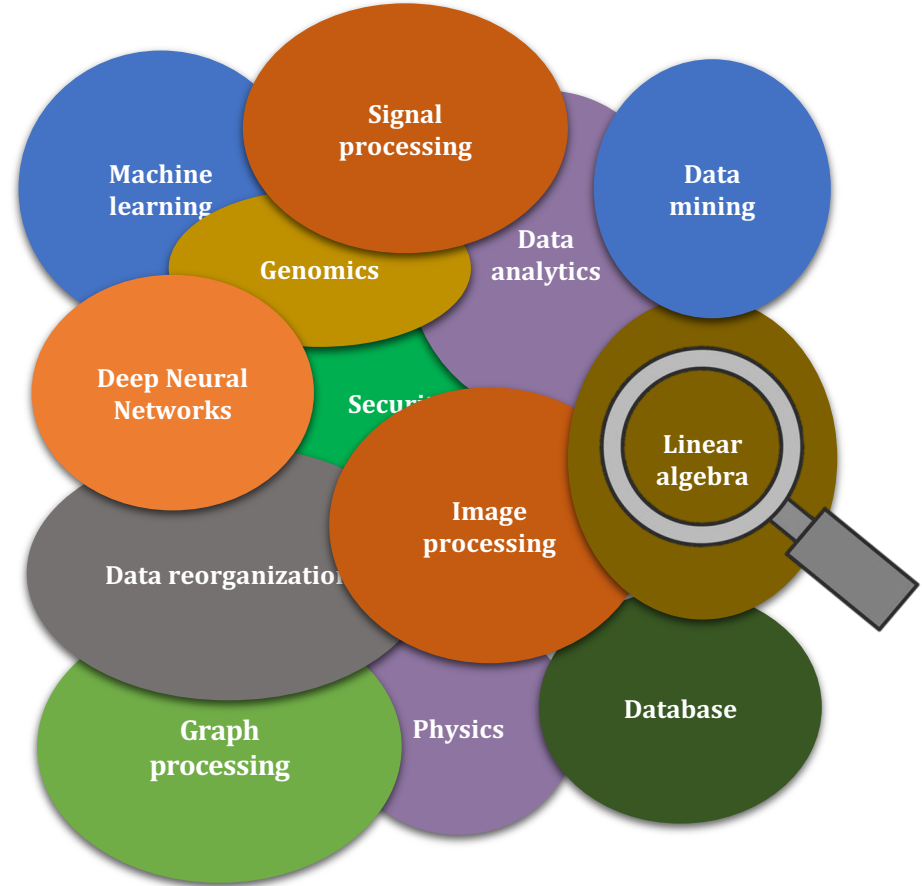
Step 1: Application Profiling

- We analyze 345 applications from distinct domains:

- Graph Processing
- Deep Neural Networks
- Physics
- High-Performance Computing
- Genomics
- Machine Learning
- Databases
- Data Reorganization
- Image Processing
- Map-Reduce
- Benchmarking
- Linear Algebra

...

SAFARI



Step 3: Memory Bottleneck Analysis

**Six classes of
data movement bottlenecks:**

each class \leftrightarrow data movement
mitigation mechanism

Memory Bottleneck Class

1a: *DRAM
Bandwidth*

1b: *DRAM Latency*

1c: *L1/L2
Cache Capacity*

2a: *L3 Cache
Contention*

2b: *L1 Cache
Capacity*

2c: *Compute-Bound*

DAMOV is Open Source

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

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About



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

Readme

Releases

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

Packages

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

Languages



omutlu Update README.md

ce1b4ea 17 days ago 5 commits

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

README.md

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

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

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

DAMOV-SIM

DAMOV
Benchmarks

DAMOV is Open Source

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

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About

DAMOV is a benchmark suite and a

Get DAMOV at:

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

README.md

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

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

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

Readme

Releases

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

Packages

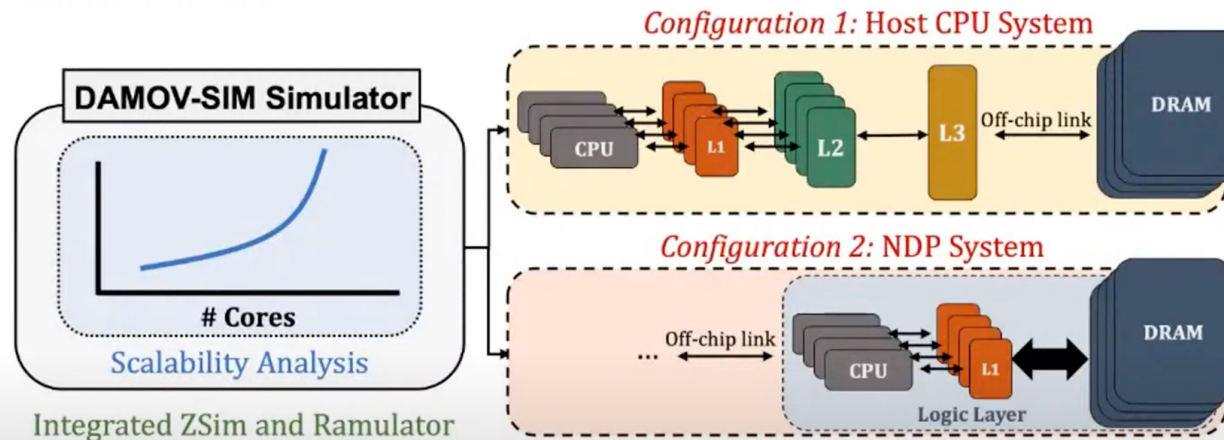
No packages published
[Publish your first package](#)

Languages

More on DAMOV Analysis Methodology & Workloads

Step 3: Memory Bottleneck Classification (2/)

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



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

SAFARI DAMOV-SIM: <https://github.com/CMU-SAFARI/DAMOV> 30

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

352 views • Streamed live on Jul 22, 2021

18 0 SHARE SAVE ...



Onur Mutlu Lectures
17.7K subscribers

ANALYTICS

EDIT VIDEO

https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=3

More on DAMOV

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

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

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

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

NANDITA VIJAYKUMAR, University of Toronto, Canada

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

MOHAMMAD SADROSADATI, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

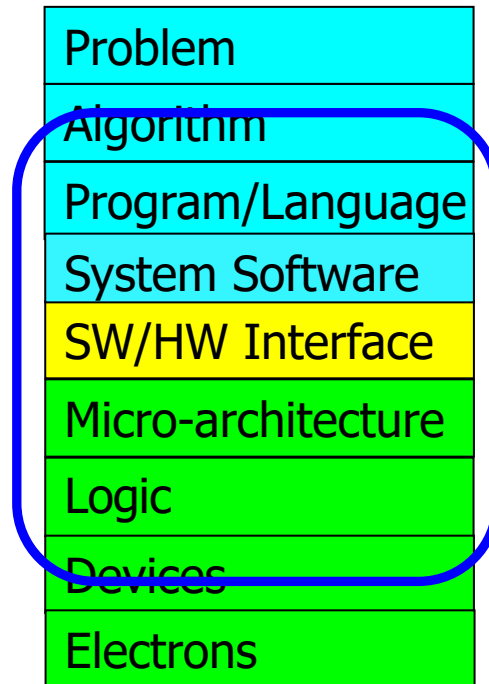
How to Enable Adoption of Processing in Memory

Potential Barriers to Adoption of PIM

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

All can be solved with change of mindset

We Need to Revisit the Entire Stack



We can get there step by step

Data-Driven **(Self-Optimizing)** **Computing Architectures**

Data-Aware (Expressive)

Computing Architectures

More Info in This Tutorial...

- Onur Mutlu,

"Memory-Centric Computing Systems"

Invited Tutorial at *66th International Electron Devices Meeting (IEDM)*, Virtual, 12 December 2020.

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

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

[[Tutorial Video](#) (1 hour 51 minutes)]

[[Executive Summary Video](#) (2 minutes)]

[[Abstract and Bio](#)]

[[Related Keynote Paper from VLSI-DAT 2020](#)]

[[Related Review Paper on Processing in Memory](#)]

<https://www.youtube.com/watch?v=H3sEaINPBOE>

Memory-Centric Computing Systems



Onur Mutlu

omutlu@gmail.com

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

12 December 2020

IEDM Tutorial

SAFARI

ETH zürich

Carnegie Mellon



0:06 / 1:51:05



IEDM 2020 Tutorial: Memory-Centric Computing Systems, Onur Mutlu, 12 December 2020

1,641 views • Dec 23, 2020

48 0 SHARE SAVE ...



Onur Mutlu Lectures
13.9K subscribers

ANALYTICS

EDIT VIDEO

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

Data-centric

Data-driven

Data-aware



SAFARI Research Group

Introduction & Research

Onur Mutlu

omutlu@gmail.com

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

19 October 2021

EFCL Huawei Day

SAFARI

ETH zürich

Carnegie Mellon

More Detailed Research Overview

Slides from ISCA 2021

Mentoring Workshop Panel

Onur Mutlu,

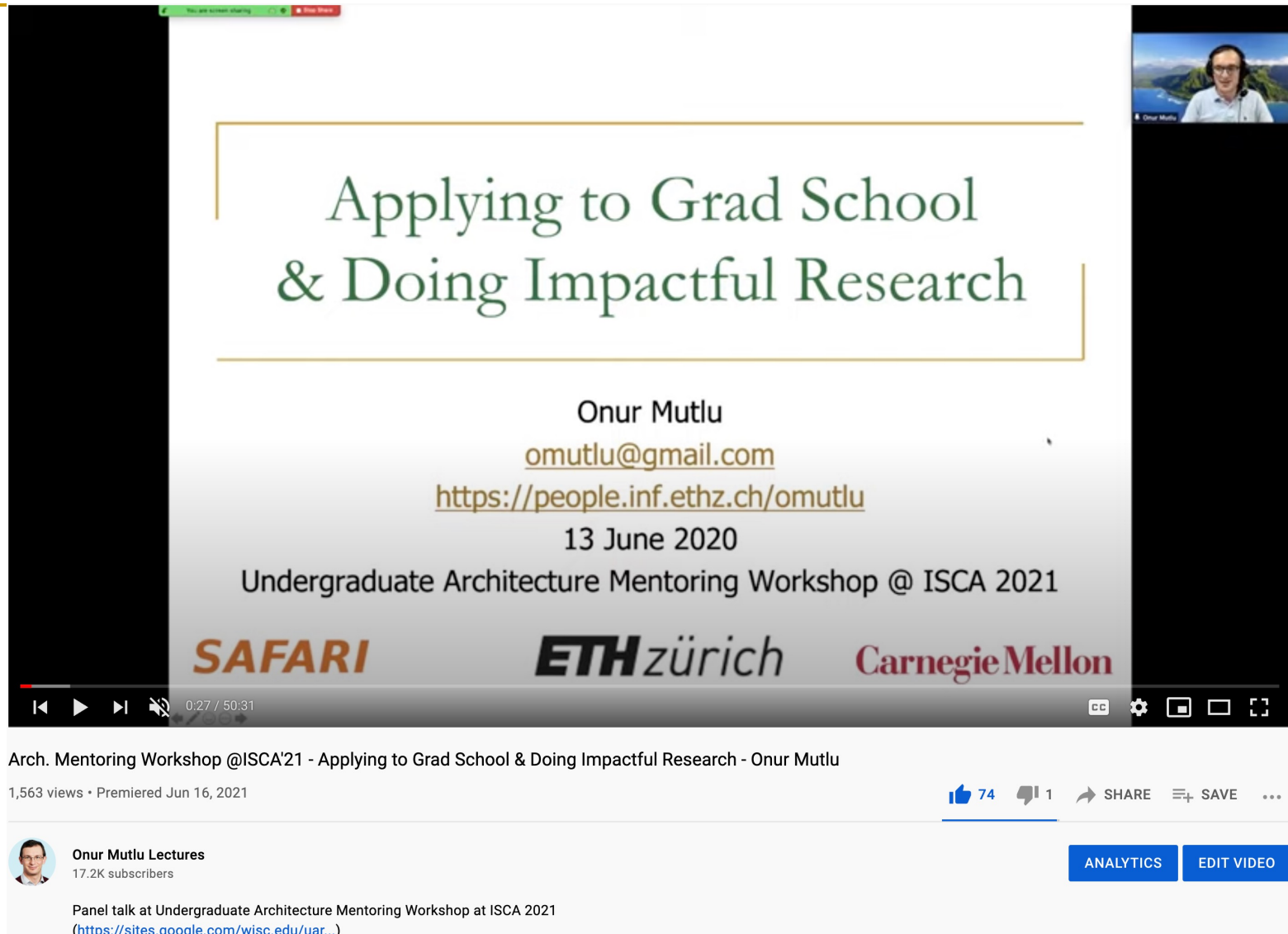
"Applying to Graduate School & Doing Impactful Research"

*Invited Panel Talk at the 3rd Undergraduate Mentoring Workshop,
held with the 48th International Symposium on Computer
Architecture (**ISCA**), Virtual, 18 June 2021.*

[Slides (pptx) (pdf)]

[Talk Video (50 minutes)]

A Talk on Impactful Research & Teaching



The video player shows a presentation slide with the title "Applying to Grad School & Doing Impactful Research" in a green serif font, enclosed in a thin gold border. Below the title, the speaker's name "Onur Mutlu" is listed, followed by his email "omutlu@gmail.com" and his website "https://people.inf.ethz.ch/omutlu". The date "13 June 2020" and the event "Undergraduate Architecture Mentoring Workshop @ ISCA 2021" are also displayed. At the bottom of the slide, the logos for "SAFARI", "ETH zürich", and "Carnegie Mellon" are shown. The video player interface includes a progress bar at 0:27 / 50:31, a small video feed of the speaker in the top right corner, and a bottom bar with engagement metrics (74 likes, 1 comment), share, save, and analytics/edit video buttons.

Applying to Grad School
& Doing Impactful Research

Onur Mutlu
omutlu@gmail.com
<https://people.inf.ethz.ch/omutlu>
13 June 2020
Undergraduate Architecture Mentoring Workshop @ ISCA 2021

SAFARI ETH zürich Carnegie Mellon

Arch. Mentoring Workshop @ISCA'21 - Applying to Grad School & Doing Impactful Research - Onur Mutlu
1,563 views • Premiered Jun 16, 2021

Onur Mutlu Lectures
17.2K subscribers

Panel talk at Undergraduate Architecture Mentoring Workshop at ISCA 2021
(<https://sites.google.com/wisc.edu/uar...>)

Example Research Topics: Quick Overview

High Performance

(to solve
the **toughest & all** problems)

Personalized and Private

(in every aspect of life:
health, medicine,
spaces, devices, robotics, ...)

Accelerating Genome Analysis

- Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#)
[IEEE Micro \(IEEE MICRO\)](#), Vol. 40, No. 5, pages 65-75, September/October 2020.
[[Slides \(pptx\)\(pdf\)](#)]
[[Talk Video \(1 hour 2 minutes\)](#)]

Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser
ETH Zürich

Zülal Bingöl
Bilkent University

Damla Senol Cali
Carnegie Mellon University

Jeremie Kim
ETH Zurich and Carnegie Mellon University

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

Can Alkan
Bilkent University

Onur Mutlu
ETH Zurich, Carnegie Mellon University, and
Bilkent University

GenASM Framework [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"**
Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.
[[Lighting Talk Video](#) (1.5 minutes)]
[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (18 minutes)]
[[Slides \(pptx\)](#) ([pdf](#))]

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

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

New Genome Sequencing Technologies

Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

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

Briefings in Bioinformatics, bby017, <https://doi.org/10.1093/bib/bby017>

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



Oxford Nanopore MinION

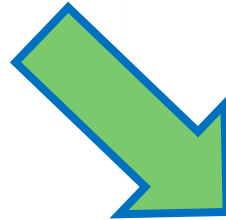
Senol Cali+, “**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**,” *Briefings in Bioinformatics*, 2018.

[[Preliminary arxiv.org version](#)]

Future of Genome Sequencing & Analysis



MinION from ONT



SmidgION from ONT

More on Fast & Efficient Genome Analysis

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

Insight: Shifting a String Helps Similarity Search

7 matches 1 mismatch

81

Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

566 views · Premiered Feb 6, 2021

31 0 SHARE SAVE ...

Onur Mutlu Lectures
13.9K subscribers

ANALYTICS EDIT VIDEO

Detailed Lectures on Genome Analysis

- **Computer Architecture, Fall 2020, Lecture 3a**
 - **Introduction to Genome Sequence Analysis** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5>
- **Computer Architecture, Fall 2020, Lecture 8**
 - **Intelligent Genome Analysis** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14>
- **Computer Architecture, Fall 2020, Lecture 9a**
 - **GenASM: Approx. String Matching Accelerator** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15>
- **Accelerating Genomics Project Course, Fall 2020, Lecture 1**
 - **Accelerating Genomics** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId>

Computing

is Bottlenecked by Data

Modern Systems are Bottlenecked by Data Storage and Movement

Modern Systems are
Bottlenecked by
Memory

An “Early” Overview Paper...

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

Memory Scaling: A Systems Architecture Perspective

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

Fundamentally Secure, Reliable, Safe Computing Architectures

Infrastructures to Understand Such Issues



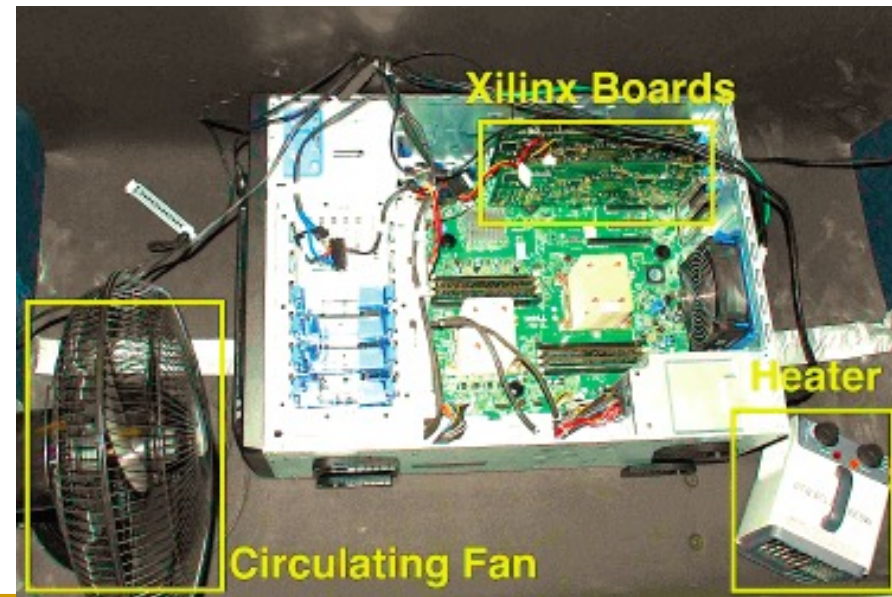
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study (Khan et al., SIGMETRICS 2014)

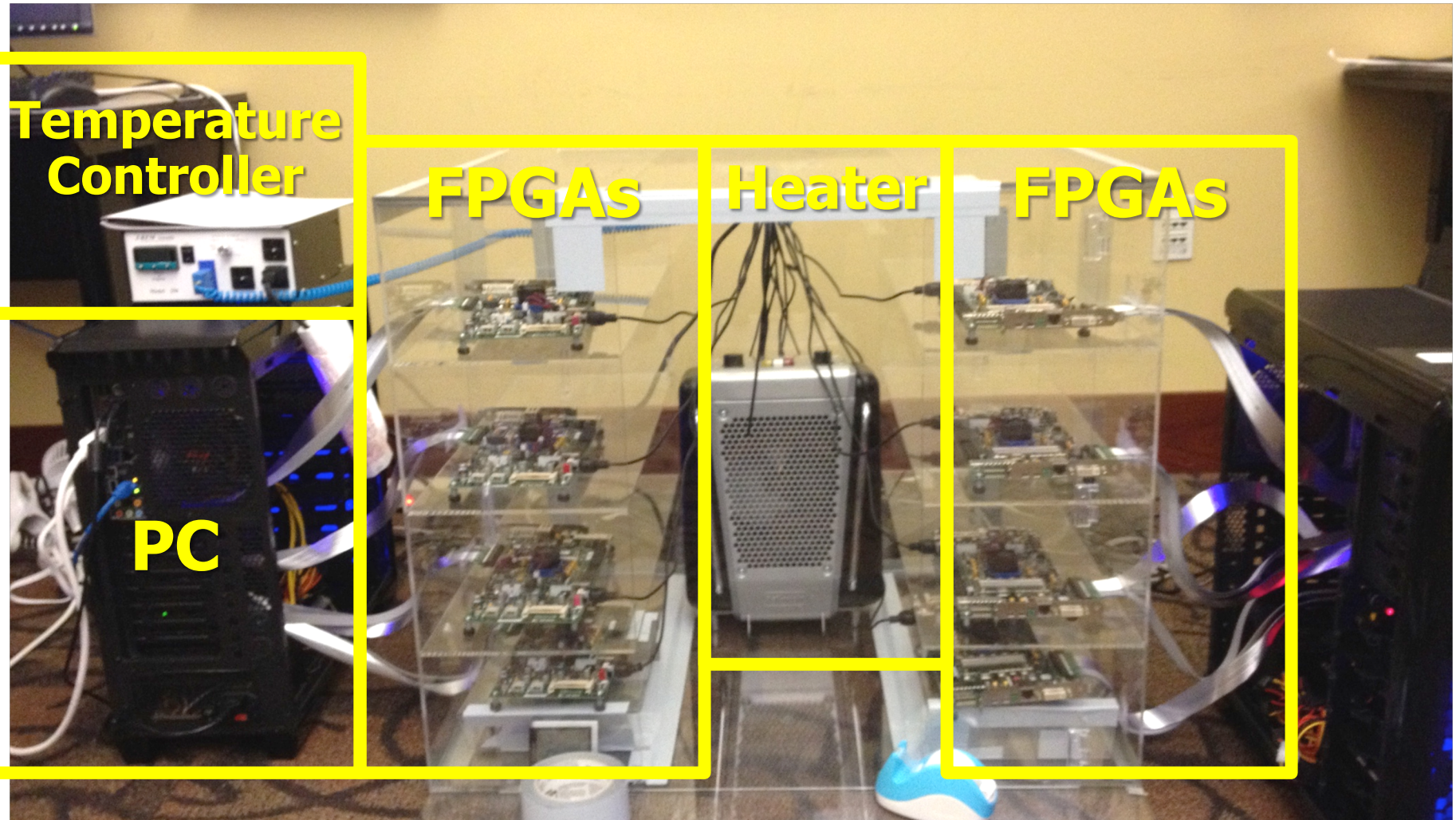
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case (Lee et al., HPCA 2015)

AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)



Infrastructures to Understand Such Issues



SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan et al., “[SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies](#),” HPCA 2017.
- Flexible
- Easy to Use (C++ API)
- Open-source
[*github.com/CMU-SAFARI/SoftMC*](https://github.com/CMU-SAFARI/SoftMC)



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

SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan^{1,2,3} Nandita Vijaykumar³ Samira Khan^{4,3} Saugata Ghose³ Kevin Chang³
Gennady Pekhimenko^{5,3} Donghyuk Lee^{6,3} Oguz Ergin² Onur Mutlu^{1,3}

¹*ETH Zürich* ²*TOBB University of Economics & Technology* ³*Carnegie Mellon University*
⁴*University of Virginia* ⁵*Microsoft Research* ⁶*NVIDIA Research*

A Curious Discovery [Kim et al., ISCA 2014]

One can
predictably induce errors
in most DRAM memory chips

DRAM RowHammer

A simple hardware failure mechanism
can create a widespread
system security vulnerability

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS	CULTURE	DESIGN	GEAR	SCIENCE
----------	---------	--------	------	---------

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE



SHARE
18276



TWEET

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

One Can Take Over an Otherwise-Secure System

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Abstract. Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology

Project Zero

Flipping Bits in Memory Without Accessing Them:
An Experimental Study of DRAM Disturbance Errors
(Kim et al., ISCA 2014)

News and updates from the Project Zero team at Google

Exploiting the DRAM rowhammer bug to
gain kernel privileges (Seaborn+, 2015)

Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges



Rowhammer

First RowHammer Analysis

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"
Proceedings of the 41st International Symposium on Computer Architecture (ISCA), Minneapolis, MN, June 2014.
[\[Slides \(pptx\) \(pdf\)\]](#) [\[Lightning Session Slides \(pptx\) \(pdf\)\]](#) [\[Source Code and Data\]](#)

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim¹ Ross Daly* Jeremie Kim¹ Chris Fallin* Ji Hye Lee¹
Donghyuk Lee¹ Chris Wilkerson² Konrad Lai Onur Mutlu¹

¹Carnegie Mellon University ²Intel Labs

Future of Memory Reliability/Security

- Onur Mutlu,
"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"

*Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (**DATE**), Lausanne, Switzerland, March 2017.*

[Slides (pptx) (pdf)]

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

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

A More Recent RowHammer Retrospective

- Onur Mutlu and Jeremie Kim,
["RowHammer: A Retrospective"](#)
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.
[[Preliminary arXiv version](#)]
[[Slides from COSADE 2019 \(pptx\)](#)]
[[Slides from VLSI-SOC 2020 \(pptx\) \(pdf\)](#)]
[[Talk Video](#) (30 minutes)]

RowHammer: A Retrospective

Onur Mutlu^{§‡} Jeremie S. Kim^{‡§}
§ETH Zürich ‡Carnegie Mellon University

RowHammer in 2020

RowHammer in 2020 (I)

- Jeremie S. Kim, Minesh Patel, A. Giray Yaglikci, Hasan Hassan, Roknoddin Azizi, Lois Orosa, and Onur Mutlu,
"Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques"
Proceedings of the 47th International Symposium on Computer Architecture (ISCA), Valencia, Spain, June 2020.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (20 minutes)]
[[Lightning Talk Video](#) (3 minutes)]

Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques

Jeremie S. Kim^{§†} Minesh Patel[§] A. Giray Yağlıkçı[§]
Hasan Hassan[§] Roknoddin Azizi[§] Lois Orosa[§] Onur Mutlu^{§†}
[§]*ETH Zürich* [†]*Carnegie Mellon University*

RowHammer in 2020 (II)

- Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi,
"TRRespass: Exploiting the Many Sides of Target Row Refresh"
Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P), San Francisco, CA, USA, May 2020.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Lecture Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#)] (17 minutes)
[[Lecture Video](#)] (59 minutes)
[[Source Code](#)]
[[Web Article](#)]
Best paper award.
Pwnie Award 2020 for Most Innovative Research. [Pwnie Awards 2020](#)

TRRespass: Exploiting the Many Sides of Target Row Refresh

Pietro Frigo^{*†} Emanuele Vannacci^{*†} Hasan Hassan[§] Victor van der Veen[¶]
Onur Mutlu[§] Cristiano Giuffrida^{*} Herbert Bos^{*} Kaveh Razavi^{*}

RowHammer is still
an open problem

Security by obscurity
is likely not a good solution

RowHammer in 2020 (III)

- Lucian Cojocar, Jeremie Kim, Minesh Patel, Lillian Tsai, Stefan Saroiu, Alec Wolman, and Onur Mutlu,

"Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers"

Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P), San Francisco, CA, USA, May 2020.

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

[[Talk Video](#) (17 minutes)]

Are We Susceptible to Rowhammer?

An End-to-End Methodology for Cloud Providers

Lucian Cojocar, Jeremie Kim^{§†}, Minesh Patel[§], Lillian Tsai[‡],
Stefan Saroiu, Alec Wolman, and Onur Mutlu^{§†}
Microsoft Research, [§]ETH Zürich, [†]CMU, [‡]MIT

BlockHammer Solution in 2021

- A. Giray Yaglikci, Minesh Patel, Jeremie S. Kim, Roknoddin Azizi, Ataberk Olgun, Lois Orosa, Hasan Hassan, Jisung Park, Konstantinos Kanellopoulos, Taha Shahroodi, Saugata Ghose, and Onur Mutlu,

"BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows"

Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.

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

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

[[Talk Video](#) (22 minutes)]

[[Short Talk Video](#) (7 minutes)]

BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows

A. Giray Yağlıkçı¹ Minesh Patel¹ Jeremie S. Kim¹ Roknoddin Azizi¹ Ataberk Olgun¹ Lois Orosa¹
Hasan Hassan¹ Jisung Park¹ Konstantinos Kanellopoulos¹ Taha Shahroodi¹ Saugata Ghose² Onur Mutlu¹

¹ETH Zürich

²University of Illinois at Urbana–Champaign

Detailed Lectures on RowHammer

■ Computer Architecture, Fall 2020, Lecture 4b

- RowHammer (ETH Zürich, Fall 2020)
- <https://www.youtube.com/watch?v=KDy632z23UE&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=8>

■ Computer Architecture, Fall 2020, Lecture 5a

- RowHammer in 2020: TRRespass (ETH Zürich, Fall 2020)
- https://www.youtube.com/watch?v=pwRw7QqK_qA&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=9

■ Computer Architecture, Fall 2020, Lecture 5b

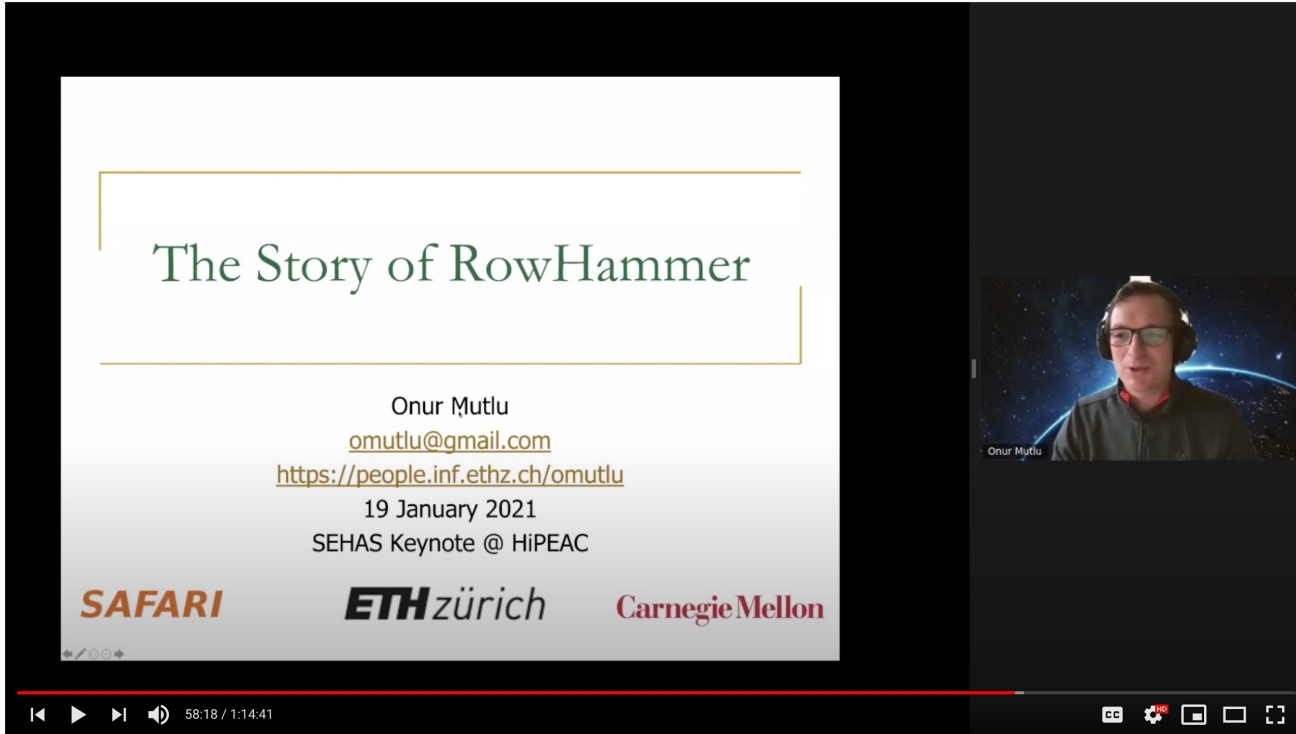
- RowHammer in 2020: Revisiting RowHammer (ETH Zürich, Fall 2020)
- <https://www.youtube.com/watch?v=gR7XR-Eepcg&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=10>

■ Computer Architecture, Fall 2020, Lecture 5c

- Secure and Reliable Memory (ETH Zürich, Fall 2020)
- <https://www.youtube.com/watch?v=HvswnsfG3oQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=11>

The Story of RowHammer Lecture ...

- Onur Mutlu,
["The Story of RowHammer"](#)
Keynote Talk at [Secure Hardware, Architectures, and Operating Systems Workshop \(SeHAS\)](#), held with [HiPEAC 2021 Conference](#), Virtual, 19 January 2021.
[[Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (1 hr 15 minutes, with Q&A)]



The Story of RowHammer

Onur Mutlu
omutlu@gmail.com
<https://people.inf.ethz.ch/omutlu>
19 January 2021
SEHAS Keynote @ HiPEAC

SAFARI ETH zürich Carnegie Mellon

1,293 views • Premiered Feb 2, 2021

64 0 SHARE SAVE ...

ANALYTICS EDIT VIDEO



Proceedings of the IEEE, Sept. 2017

Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD's reliability and lifetime.

By YU CAI, SAUGATA GHOSE, ERICH F. HARATSCH, YIXIN LUO, AND ONUR MUTLU

Understand and Model with Experiments (Flash)



[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018, SIGMETRICS 2018]

NAND Daughter Board

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

Main Memory Needs
Intelligent Controllers

High Performance,
Energy Efficient,
Sustainable

The Problem

Processing of data
is performed
far away from the data

Energy Waste in Mobile Devices

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

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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

The Problem

Data access is the major performance and energy bottleneck

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

We Need A Paradigm Shift To ...

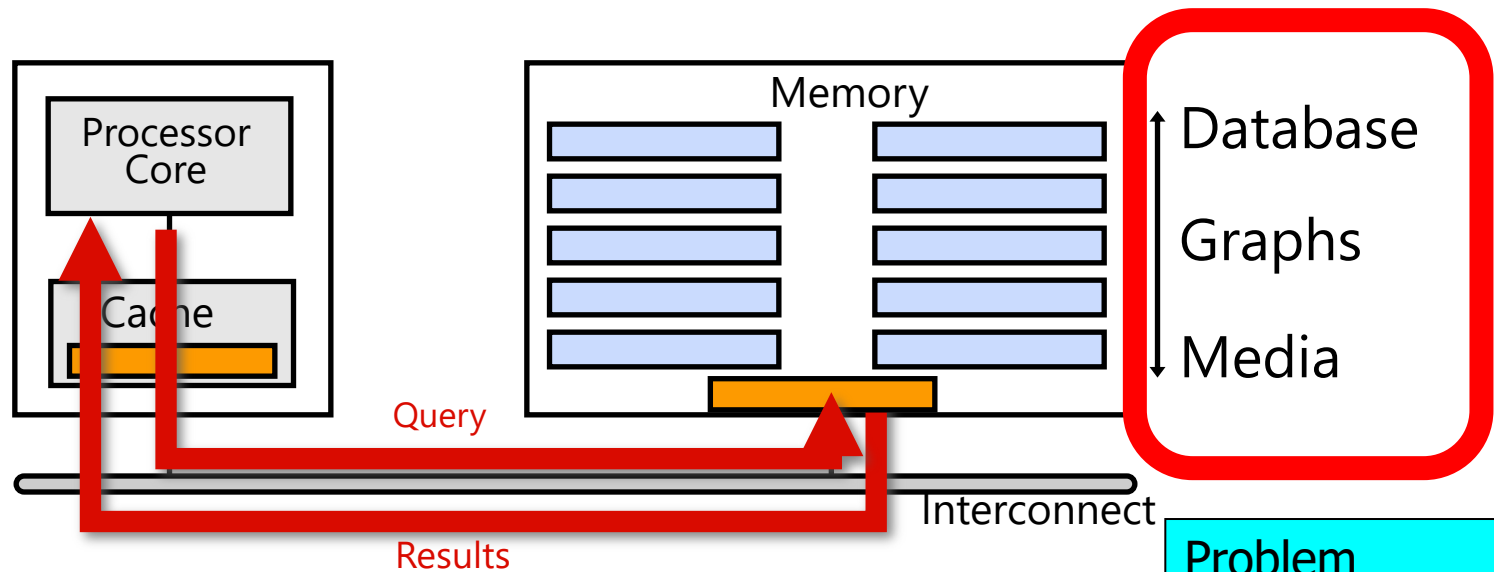
- Enable computation with minimal data movement
- Compute where it makes sense (where data resides)
- Make computing architectures more data-centric

Computing Architectures with Minimal Data Movement

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

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

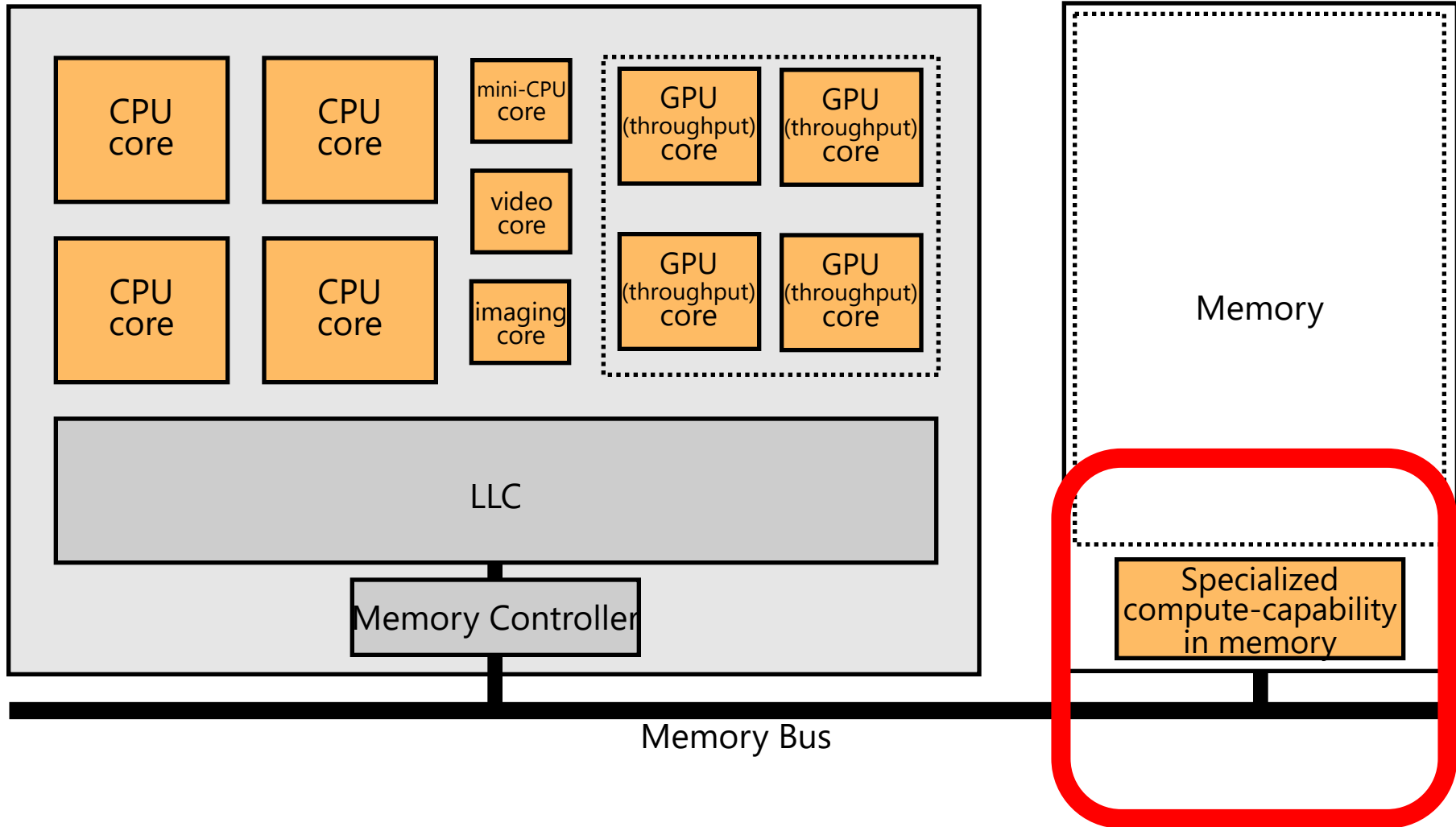
Goal: Processing Inside Memory



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

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

Memory as an Accelerator



Memory similar to a "conventional" accelerator

Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory

PIM Review and Open Problems

A Modern Primer on Processing in Memory

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

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

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

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

"A Modern Primer on Processing in Memory"

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

PIM Review and Open Problems (II)

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

Saugata Ghose[†] Amirali Boroumand[†] Jeremie S. Kim^{†§} Juan Gómez-Luna[§] Onur Mutlu^{§†}

[†]*Carnegie Mellon University*

[§]*ETH Zürich*

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu,

"Processing-in-Memory: A Workload-Driven Perspective"

Invited Article in IBM Journal of Research & Development, Special Issue on Hardware for Artificial Intelligence, to appear in November 2019.

[Preliminary arXiv version]

More on Processing in Memory

- Vivek Seshadri et al., “[Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology](#),” MICRO 2017.

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

Vivek Seshadri^{1,5} Donghyuk Lee^{2,5} Thomas Mullins^{3,5} Hasan Hassan⁴ Amirali Boroumand⁵
Jeremie Kim^{4,5} Michael A. Kozuch³ Onur Mutlu^{4,5} Phillip B. Gibbons⁵ Todd C. Mowry⁵

¹Microsoft Research India ²NVIDIA Research ³Intel ⁴ETH Zürich ⁵Carnegie Mellon University

More on Processing in Memory

- Vivek Seshadri and Onur Mutlu,
"In-DRAM Bulk Bitwise Execution Engine"
Invited Book Chapter in Advances in Computers, to appear
in 2020.
[[Preliminary arXiv version](#)]

In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri
Microsoft Research India
visesha@microsoft.com

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

More on Processing in Memory (II)

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu, **["SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM"](#)** *Proceedings of the 26th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Virtual, March-April 2021.
[[2-page Extended Abstract](#)]
[[Short Talk Slides \(pptx\)](#) ([pdf](#))]
[[Talk Slides \(pptx\)](#) ([pdf](#))]
[[Short Talk Video](#) (5 mins)]
[[Full Talk Video](#) (27 mins)]

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

*Nastaran Hajinazar ^{1,2}	*Geraldo F. Oliveira ¹	Sven Gregorio ¹	João Dinis Ferreira ¹
Nika Mansouri Ghiasi ¹	Minesh Patel ¹	Mohammed Alser ¹	Saugata Ghose ³
	Juan Gómez-Luna ¹	Onur Mutlu ¹	

¹ETH Zürich

²Simon Fraser University

³University of Illinois at Urbana–Champaign

More on Processing in Memory (III)

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoun Choi,
"A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"
Proceedings of the 42nd International Symposium on Computer Architecture (ISCA), Portland, OR, June 2015.
[Slides (pdf)] [Lightning Session Slides (pdf)]

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

Junwhan Ahn Sungpack Hong[§] Sungjoo Yoo Onur Mutlu[†] Kiyoun 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

More on Processing in Memory (IV)

- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu, ["Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks"](#)

*Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**), Williamsburg, VA, USA, March 2018.*

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

More on Processing in Memory (V)

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

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

Junwhan Ahn Sungjoo Yoo Onur Mutlu[†] Kiyoun Choi

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

Seoul National University

[†]Carnegie Mellon University

In-DRAM Physical Unclonable Functions

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu,
"The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices"
Proceedings of the 24th International Symposium on High-Performance Computer Architecture (HPCA), Vienna, Austria, February 2018.
[[Lightning Talk Video](#)]
[[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Session Slides \(pptx\)](#)] [[pdf](#)]
[[Full Talk Lecture Video](#) (28 minutes)]

The DRAM Latency PUF:

Quickly Evaluating Physical Unclonable Functions

by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim^{†§}

Minesh Patel[§]

Hasan Hassan[§]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[§]ETH Zürich

In-DRAM True Random Number Generation

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

Proceedings of the 25th International Symposium on High-Performance Computer Architecture (HPCA), Washington, DC, USA, February 2019.

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

[[Full Talk Video](#) (21 minutes)]

[[Full Talk Lecture Video](#) (27 minutes)]

Top Picks Honorable Mention by IEEE Micro.

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

Jeremie S. Kim^{‡§}

Minesh Patel[§]

Hasan Hassan[§]

Lois Orosa[§]

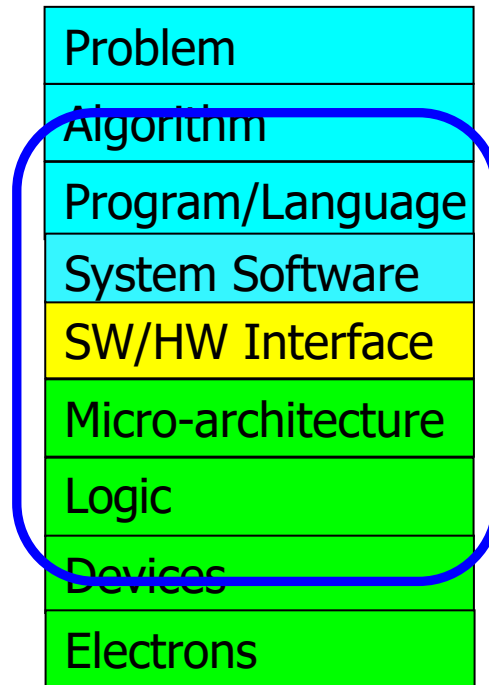
Onur Mutlu^{§‡}

[‡]Carnegie Mellon University

[§]ETH Zürich

How to Enable Adoption of Processing in Memory

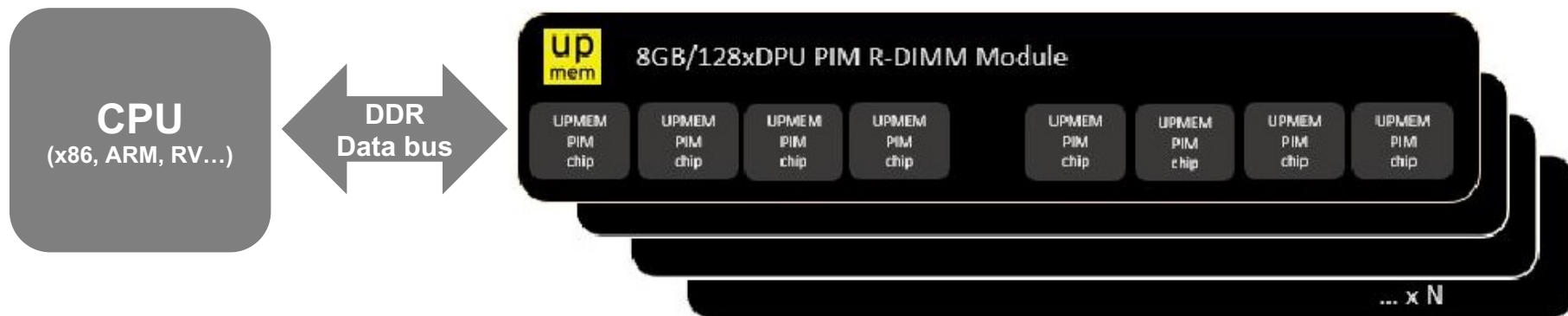
We Need to Revisit the Entire Stack



We can get there step by step

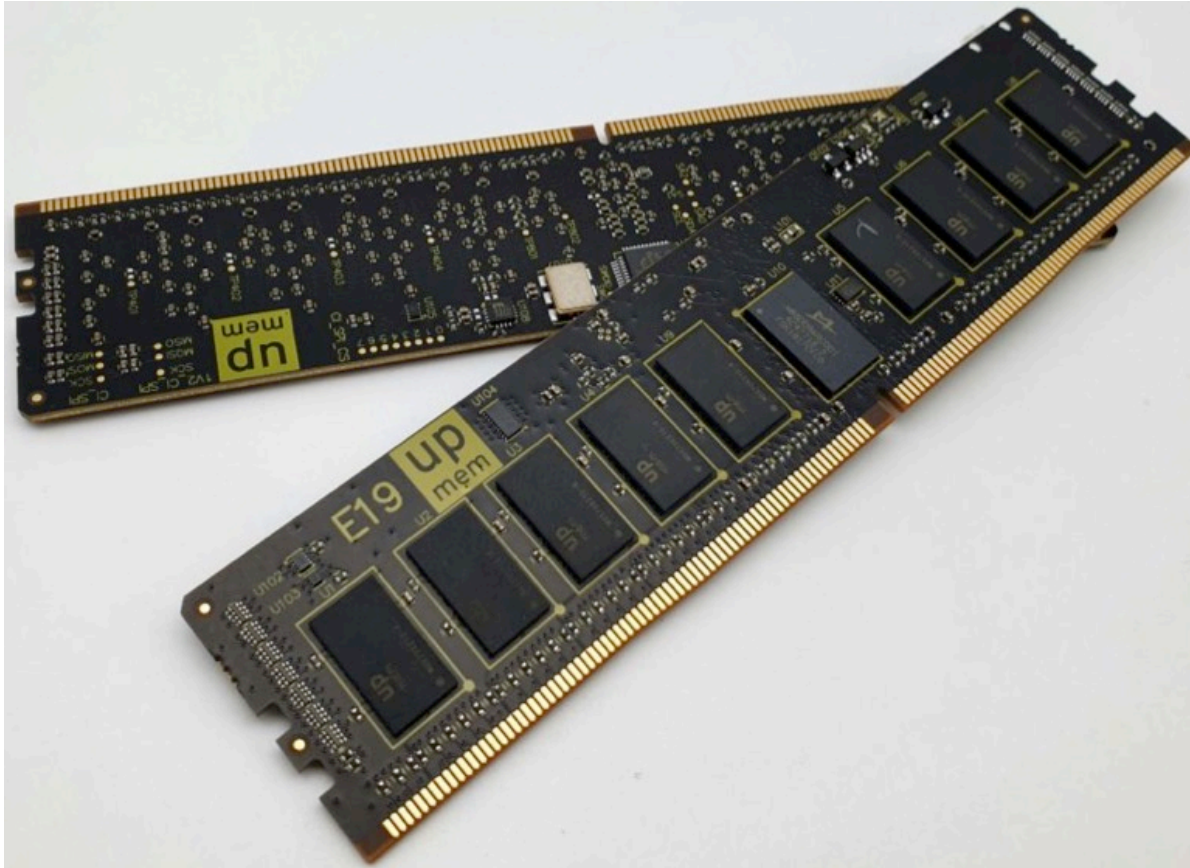
UPMEM Processing-in-DRAM Engine (2019)

- **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



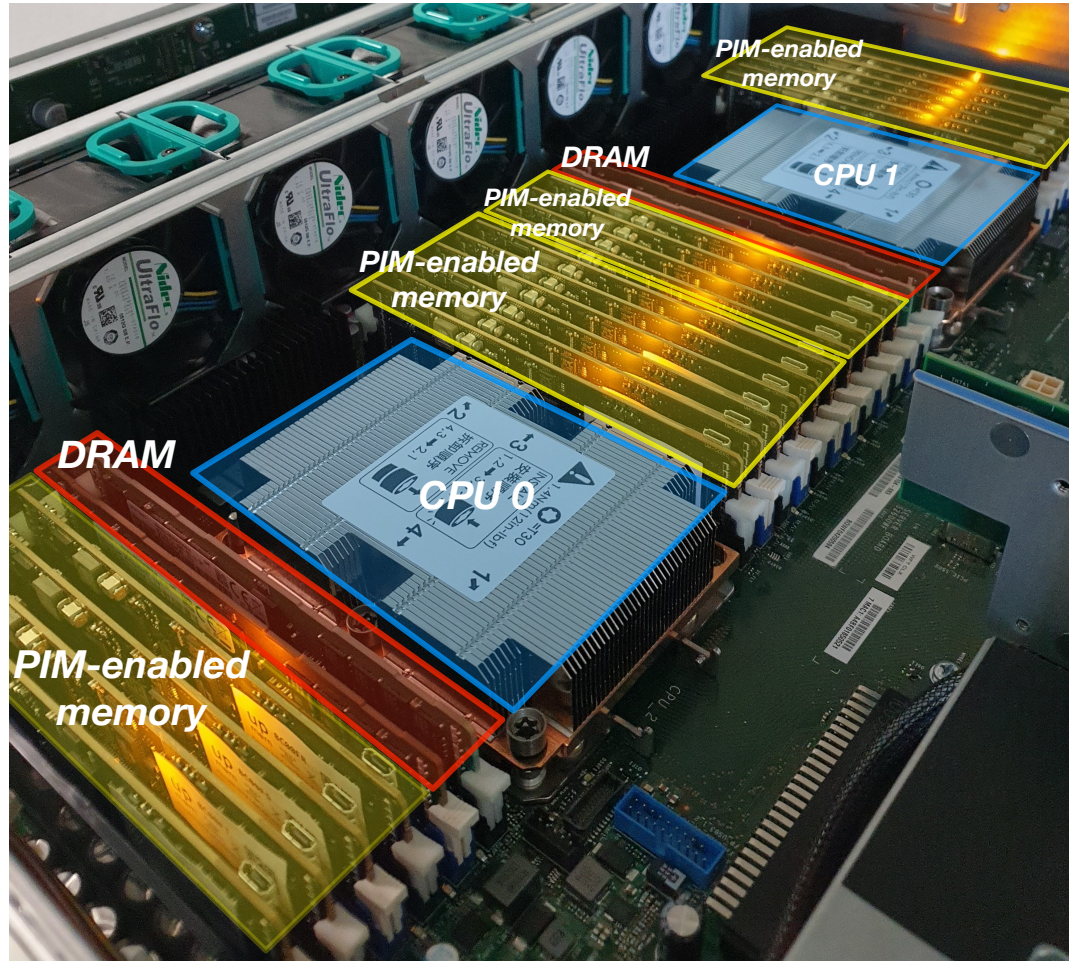
UPMEM Memory Modules

- E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
- P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz

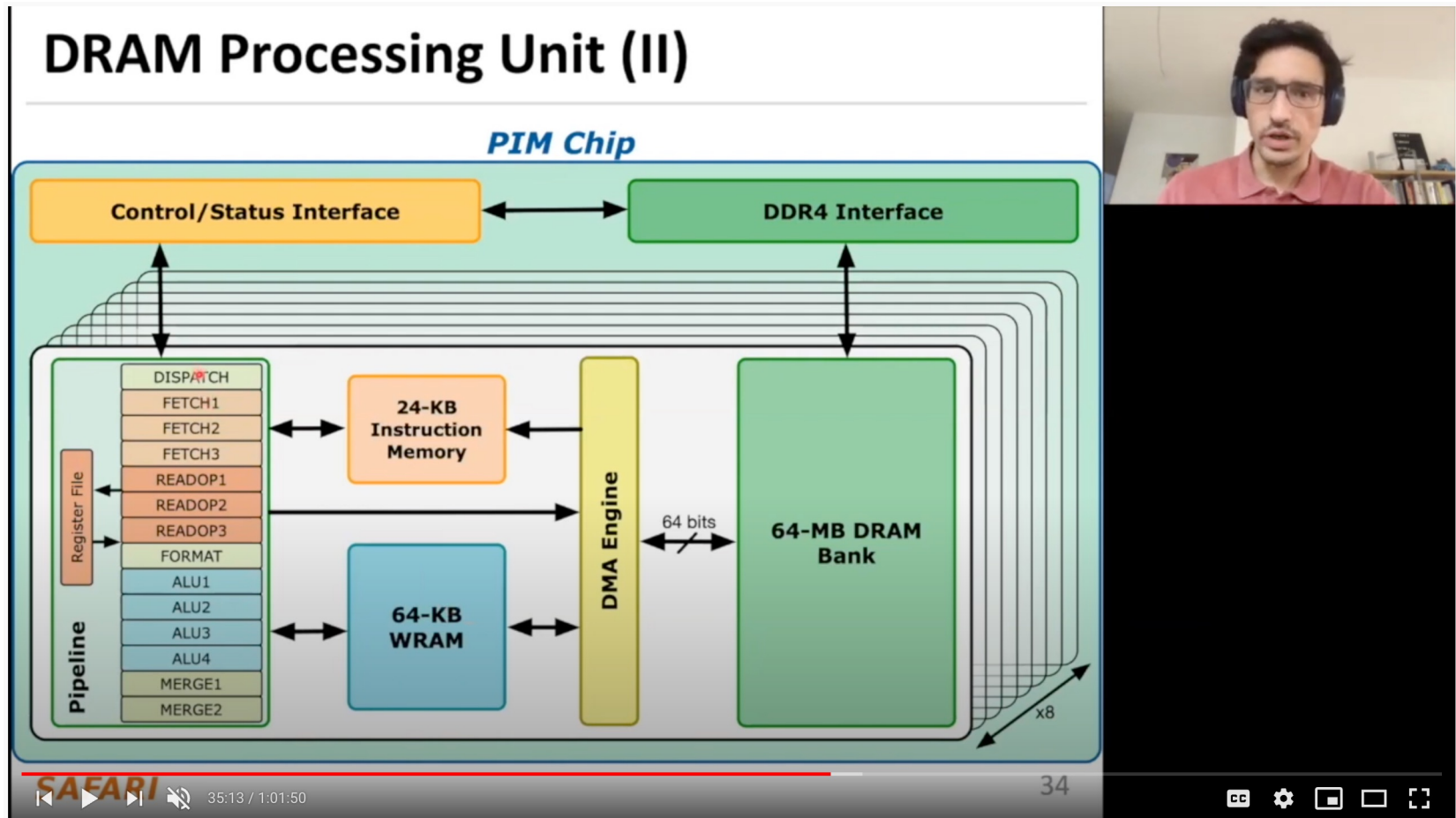


PIM System Organization

- UPMEM-based PIM system with 20 UPMEM memory modules of 16 chips each (40 ranks) → 2560 DPUs



More on the UPMEM PIM System



ETH ZÜRICH HAUPTGEBÄUDE

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

1,120 views • Oct 31, 2020

30 0 SHARE SAVE ...



Onur Mutlu Lectures
16.7K subscribers

ANALYTICS

EDIT VIDEO

<https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=26>

Experimental Analysis of the UPMEM PIM Engine

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

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

IZZAT EL HAJJ, American University of Beirut, Lebanon

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

CHRISTINA GIANNOULA, ETH Zürich, Switzerland and NTUA, Greece

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

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

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

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

DAMOV Methodology & Workloads

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

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

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

NANDITA VIJAYKUMAR, University of Toronto, Canada

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

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

ONUR MUTLU, ETH Zürich, Switzerland

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

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

Detailed Lectures on PIM (I)

- **Computer Architecture, Fall 2020, Lecture 6**
 - **Computation in Memory** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=oGcZAGwfEUE&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=12>
- **Computer Architecture, Fall 2020, Lecture 7**
 - **Near-Data Processing** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=j2GIigqn1Qw&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=13>
- **Computer Architecture, Fall 2020, Lecture 11a**
 - **Memory Controllers** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=TeG773OgiMQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=20>
- **Computer Architecture, Fall 2020, Lecture 12d**
 - **Real Processing-in-DRAM with UPMEM** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=25>

Detailed Lectures on PIM (II)

- **Computer Architecture, Fall 2020, Lecture 15**
 - **Emerging Memory Technologies** (ETH Zürich, Fall 2020)
 - https://www.youtube.com/watch?v=AIE1rD9G_YU&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=28
- **Computer Architecture, Fall 2020, Lecture 16a**
 - **Opportunities & Challenges of Emerging Memory Technologies** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=29>
- **Computer Architecture, Fall 2020, Guest Lecture**
 - **In-Memory Computing: Memory Devices & Applications** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=wNmQqHiEZnk&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=41>

A Tutorial on PIM

- Onur Mutlu,

"Memory-Centric Computing Systems"

Invited Tutorial at *66th International Electron Devices Meeting (IEDM)*, Virtual, 12 December 2020.

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

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

[[Tutorial Video](#) (1 hour 51 minutes)]

[[Executive Summary Video](#) (2 minutes)]

[[Abstract and Bio](#)]

[[Related Keynote Paper from VLSI-DAT 2020](#)]

[[Related Review Paper on Processing in Memory](#)]

<https://www.youtube.com/watch?v=H3sEaINPBOE>

Memory-Centric Computing Systems



Onur Mutlu

omutlu@gmail.com

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

12 December 2020

IEDM Tutorial

SAFARI

ETH zürich

Carnegie Mellon



0:06 / 1:51:05



IEDM 2020 Tutorial: Memory-Centric Computing Systems, Onur Mutlu, 12 December 2020

1,641 views • Dec 23, 2020

48 0 SHARE SAVE ...



Onur Mutlu Lectures
13.9K subscribers

ANALYTICS

EDIT VIDEO

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

PIM Can Enable New Medical Platforms

Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

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

Briefings in Bioinformatics, bby017, <https://doi.org/10.1093/bib/bby017>

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



Oxford Nanopore MinION

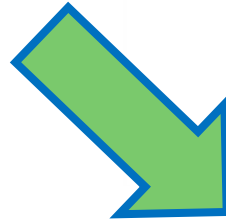
Senol Cali+, “**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**,” *Briefings in Bioinformatics*, 2018.

[[Preliminary arxiv.org version](#)]

Future of Genome Sequencing & Analysis



MinION from ONT



SmidgION from ONT

Accelerating Genome Analysis: Overview

- Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
[**"Accelerating Genome Analysis: A Primer on an Ongoing Journey"**](#)
[*IEEE Micro* \(**IEEE MICRO**\)](#), Vol. 40, No. 5, pages 65-75, September/October 2020.
[[Slides \(pptx\)\(pdf\)](#)]
[[Talk Video \(1 hour 2 minutes\)](#)]

Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser

ETH Zürich

Zülal Bingöl

Bilkent University

Damla Senol Cali

Carnegie Mellon University

Jeremie Kim

ETH Zurich and Carnegie Mellon University

Saugata Ghose

University of Illinois at Urbana–Champaign and
Carnegie Mellon University

Can Alkan

Bilkent University

Onur Mutlu

ETH Zurich, Carnegie Mellon University, and
Bilkent University

More on Fast Genome Analysis ...

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

Insight: Shifting a String Helps Similarity Search

7 matches 1 mismatch

ISTANBUL

ISTNBUL

ISTNBUL

81

46:08 / 1:37:37

Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

566 views · Premiered Feb 6, 2021

31 0 SHARE SAVE ...

Onur Mutlu Lectures
13.9K subscribers

ANALYTICS EDIT VIDEO

Detailed Lectures on Genome Analysis

- **Computer Architecture, Fall 2020, Lecture 3a**
 - **Introduction to Genome Sequence Analysis** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5>
- **Computer Architecture, Fall 2020, Lecture 8**
 - **Intelligent Genome Analysis** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14>
- **Computer Architecture, Fall 2020, Lecture 9a**
 - **GenASM: Approx. String Matching Accelerator** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15>
- **Accelerating Genomics Project Course, Fall 2020, Lecture 1**
 - **Accelerating Genomics** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId>

Fundamentally Low-Latency Computing Architectures

Truly Reducing Memory Latency

Tiered-Latency DRAM

- Donghyuk Lee, Yoongu Kim, Vivek Seshadri, Jamie Liu, Lavanya Subramanian, and Onur Mutlu,
["Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture"](#)
Proceedings of the 19th International Symposium on High-Performance Computer Architecture (HPCA), Shenzhen, China, February 2013. [Slides \(pptx\)](#)

Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture

Donghyuk Lee Yoongu Kim Vivek Seshadri Jamie Liu Lavanya Subramanian Onur Mutlu
Carnegie Mellon University

Adaptive-Latency DRAM

- Donghyuk Lee, Yoongu Kim, Gennady Pekhimenko, Samira Khan, Vivek Seshadri, Kevin Chang, and Onur Mutlu,
["Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case"](#)
Proceedings of the [21st International Symposium on High-Performance Computer Architecture \(HPCA\)](#), Bay Area, CA, February 2015.
[[Slides \(pptx\) \(pdf\)](#)] [[Full data sets](#)]

Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case

Donghyuk Lee Yoongu Kim Gennady Pekhimenko
Samira Khan Vivek Seshadri Kevin Chang Onur Mutlu
Carnegie Mellon University

Analysis of Latency Variation in DRAM Chips

- Kevin Chang, Abhijith Kashyap, Hasan Hassan, Samira Khan, Kevin Hsieh, Donghyuk Lee, Saugata Ghose, Gennady Pekhimenko, Tianshi Li, and Onur Mutlu,

"Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization"

*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**), Antibes Juan-Les-Pins, France, June 2016.*

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

[[Source Code](#)]

Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization

Kevin K. Chang¹

Abhijith Kashyap¹

Hasan Hassan^{1,2}

Saugata Ghose¹

Kevin Hsieh¹

Donghyuk Lee¹

Tianshi Li^{1,3}

Gennady Pekhimenko¹

Samira Khan⁴

Onur Mutlu^{5,1}

¹Carnegie Mellon University ²TOBB ETÜ ³Peking University ⁴University of Virginia ⁵ETH Zürich

Design-Induced Latency Variation in DRAM

- Donghyuk Lee, Samira Khan, Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Gennady Pekhimenko, Vivek Seshadri, and Onur Mutlu,
"Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms"
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**), Urbana-Champaign, IL, USA, June 2017.*

Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms

Donghyuk Lee, NVIDIA and Carnegie Mellon University

Samira Khan, University of Virginia

Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Carnegie Mellon University

Gennady Pekhimenko, Vivek Seshadri, Microsoft Research

Onur Mutlu, ETH Zürich and Carnegie Mellon University

Solar-DRAM: Putting It Together

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu,
**"Solar-DRAM: Reducing DRAM Access Latency by
Exploiting the Variation in Local Bitlines"**
*Proceedings of the 36th IEEE International Conference on
Computer Design (ICCD)*, Orlando, FL, USA, October 2018.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (16 minutes)]

Solar-DRAM: Reducing DRAM Access Latency by Exploiting the Variation in Local Bitlines

Jeremie S. Kim^{‡§} Minesh Patel[§] Hasan Hassan[§] Onur Mutlu^{§‡}
 ‡Carnegie Mellon University §ETH Zürich

CLR-DRAM: Capacity-Latency Reconfigurability

- Haocong Luo, Taha Shahroodi, Hasan Hassan, Minesh Patel, A. Giray Yaglikci, Lois Orosa, Jisung Park, and Onur Mutlu,
"CLR-DRAM: A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-Off"
Proceedings of the 47th International Symposium on Computer Architecture (ISCA), Valencia, Spain, June 2020.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (20 minutes)]
[[Lightning Talk Video](#) (3 minutes)]

CLR-DRAM: A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-Off

Haocong Luo^{§†} Taha Shahroodi[§] Hasan Hassan[§] Minesh Patel[§]
A. Giray Yağlıkçı[§] Lois Orosa[§] Jisung Park[§] Onur Mutlu[§]

[§]ETH Zürich

[†]ShanghaiTech University

Low-Latency Solid-State Drives (SSDs)

- Jisung Park, Myungsuk Kim, Myoungjun Chun, Lois Orosa, Jihong Kim, and Onur Mutlu,
[**"Reducing Solid-State Drive Read Latency by Optimizing Read-Retry"**](#)
Proceedings of the 26th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, March-April 2021.
[[2-page Extended Abstract](#)]
[[Short Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Full Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Short Talk Video](#) (5 mins)]
[[Full Talk Video](#) (19 mins)]

Reducing Solid-State Drive Read Latency by Optimizing Read-Retry

Jisung Park¹ Myungsuk Kim^{2,3} Myoungjun Chun² Lois Orosa¹ Jihong Kim² Onur Mutlu¹

¹ETH Zürich
Switzerland

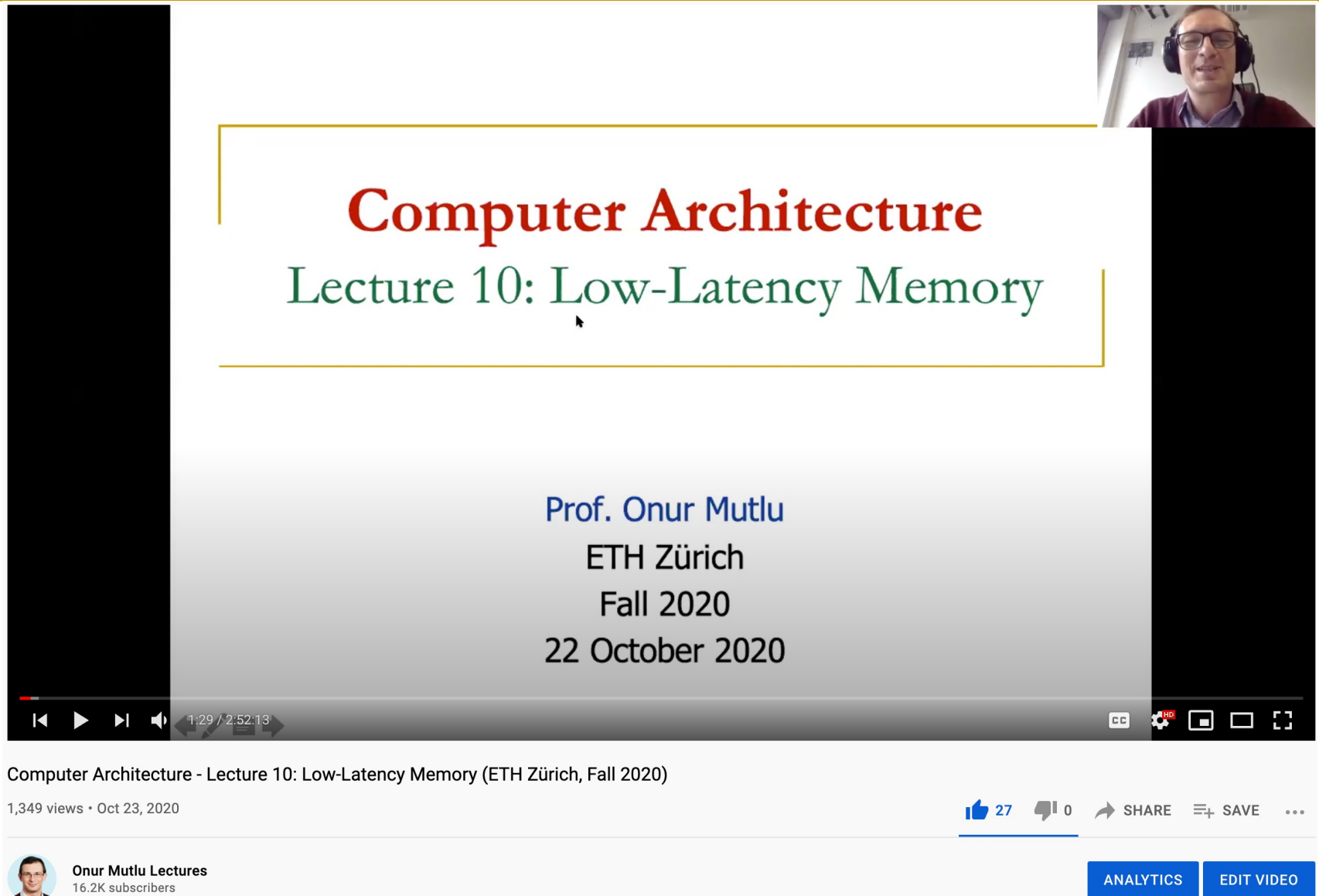
²Seoul National University
Republic of Korea

³Kyungpook National University
Republic of Korea

Lectures on Low-Latency Memory

- **Computer Architecture, Fall 2020, Lecture 10**
 - **Low-Latency Memory** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=vQd1YgOH1Mw&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=19>
- **Computer Architecture, Fall 2020, Lecture 12b**
 - **Capacity-Latency Reconfigurable DRAM** (ETH Zürich, Fall 2020)
 - <https://www.youtube.com/watch?v=DUtPFW3jxq4&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=23>
- **Computer Architecture, Fall 2019, Lecture 11a**
 - **DRAM Latency PUF** (ETH Zürich, Fall 2019)
 - https://www.youtube.com/watch?v=7gqnrTZpjxE&list=PL5Q2soXY2Zi-DyoI3HbqcdtUm9YWRR_z-&index=15
- **Computer Architecture, Fall 2019, Lecture 11b**
 - **DRAM True Random Number Generator** (ETH Zürich, Fall 2020)
 - https://www.youtube.com/watch?v=Y3hPv1I5f8Y&list=PL5Q2soXY2Zi-DyoI3HbqcdtUm9YWRR_z-&index=16

A Tutorial on Low-Latency Memory



The image shows a YouTube video player interface. The main video area displays a title slide for 'Computer Architecture' with 'Lecture 10: Low-Latency Memory' in green text. Below the title, it lists 'Prof. Onur Mutlu', 'ETH Zürich', 'Fall 2020', and '22 October 2020'. A small video inset in the top right corner shows the professor wearing headphones. The video player controls at the bottom show a progress bar at 1:29 / 2:52:13, along with icons for play, volume, and other controls. Below the video player, the video title 'Computer Architecture - Lecture 10: Low-Latency Memory (ETH Zürich, Fall 2020)' is displayed, followed by '1,349 views • Oct 23, 2020'. The channel name 'Onur Mutlu Lectures' with 16.2K subscribers is shown on the left, and buttons for 'ANALYTICS' and 'EDIT VIDEO' are on the right. Engagement icons for likes (27), comments (0), share, save, and a menu are also present.

Computer Architecture
Lecture 10: Low-Latency Memory

Prof. Onur Mutlu
ETH Zürich
Fall 2020
22 October 2020

1:29 / 2:52:13

Computer Architecture - Lecture 10: Low-Latency Memory (ETH Zürich, Fall 2020)

1,349 views • Oct 23, 2020

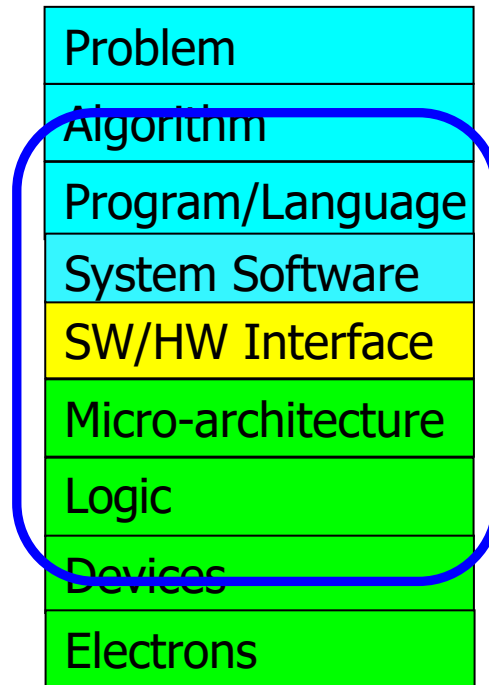
Onur Mutlu Lectures
16.2K subscribers

27 0 SHARE SAVE ...

ANALYTICS EDIT VIDEO

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

We Need to Revisit the Entire Stack



We can get there step by step

Open-Source Artifacts

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

Repositories 45 Packages People 12 Projects

Find a repository...

Type ▾

Language ▾

Sort ▾

COVIDHunter

COVIDHunter 🦠📊: An accurate and flexible COVID-19 outbreak simulation model that forecasts the strength of future mitigation measures and the numbers of cases, hospitalizations, and deaths for a given day, while considering the potential effect of environmental conditions. Described by Alser et al. (preliminary version at <https://arxiv.org/abs/2...>

simulation epidemiology covid-19 covid-19-data covid-19-tracker
 reproduction-number covidhunter

Swift MIT 1 5 0 0 Updated 9 hours ago

SNP-Selective-Hiding

An optimization-based mechanism 🧠🔒 to selectively hide the minimum number of overlapping SNPs among the family members 👨👩👧👦 who participated in the genomic studies (i.e. GWAS). Our goal is to distort the dependencies among the family members in the original database for achieving better privacy without significantly degrading the data utility.

gwas genomics data-privacy differential-privacy
 genomic-data-analysis laplace-distribution genomic-privacy

MATLAB 0 0 0 0 Updated 10 hours ago

SneakySnake

SneakySnake 🐍 is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude) expedites sequence alignment calculation for both short and long reads. Described in the Bioinformatics (2020) by Alser et al. <https://arxiv.org/abs...>

fpga gpu smith-waterman needleman-wunsch
 sequence-alignment long-reads minimap2

VHDL GPL-3.0 6 31 0 1 Updated on May 12

ramulator

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 IEEE CAL 2015 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/ramulator_dram_simulator-ieee-cal15.pdf

C++ MIT 121 237 47 4 Updated on May 11

Top languages

C++ C C# AGS Script
 VHDL

Most used topics

dram reliability
 error-correcting-codes
 experimental-data
 pre-alignment-filtering

People

12 >



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

Some Open Source Tools (I)

- Rowhammer – Program to Induce RowHammer Errors
 - <https://github.com/CMU-SAFARI/rowhammer>
- Ramulator – Fast and Extensible DRAM Simulator
 - <https://github.com/CMU-SAFARI/ramulator>
- MemSim – Simple Memory Simulator
 - <https://github.com/CMU-SAFARI/memsim>
- NOCulator – Flexible Network-on-Chip Simulator
 - <https://github.com/CMU-SAFARI/NOCulator>
- SoftMC – FPGA-Based DRAM Testing Infrastructure
 - <https://github.com/CMU-SAFARI/SoftMC>
- Other open-source software from my group
 - <https://github.com/CMU-SAFARI/>
 - <http://www.ece.cmu.edu/~safari/tools.html>

Some Open Source Tools (II)

- MQSim – A Fast Modern SSD Simulator
 - <https://github.com/CMU-SAFARI/MQSim>
- Mosaic – GPU Simulator Supporting Concurrent Applications
 - <https://github.com/CMU-SAFARI/Mosaic>
- IMPICA – Processing in 3D-Stacked Memory Simulator
 - <https://github.com/CMU-SAFARI/IMPICA>
- SMLA – Detailed 3D-Stacked Memory Simulator
 - <https://github.com/CMU-SAFARI/SMLA>
- HWASim – Simulator for Heterogeneous CPU-HWA Systems
 - <https://github.com/CMU-SAFARI/HWASim>
- Other open-source software from my group
 - <https://github.com/CMU-SAFARI/>
 - <http://www.ece.cmu.edu/~safari/tools.html>

More Open Source Tools (III)

- A lot more open-source software from my group
 - ❑ <https://github.com/CMU-SAFARI/>

The screenshot shows the GitHub profile for the SAFARI Research Group. The header includes the SAFARI logo, the group name, and a description: 'Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.' Below this, there are statistics for Repositories (30), People (27), Teams (1), and Projects (0). A search bar and filters for Type and Language are present. The main content area features the 'MQSim' repository, described as a fast and accurate simulator for SSDs. To the right, there are sections for 'Top languages' (C++, C, C#, AGS Script, Verilog) and 'Most used topics' (dram, reliability).

SAFARI SAFARI Research Group at ETH Zurich and Carnegie Mellon University

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

ETH Zurich and Carnegi... <http://www.ece.cmu.ed...> omutlu@gmail.com

Repositories 30 **People** 27 **Teams** 1 **Projects** 0 **Settings**

Search repositories... Type: All Language: All Customize pinned repositories **New**

MQSim

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 implementations, steady-state SSD conditions, and the full end-to-end latency of requests in modern SSDs. It is described in detail in the FAST 2018 paper by A...

C++ ★ 14 14 MIT Updated 8 days ago

Top languages

- C++ C C# AGS Script Verilog

Most used topics Manage

- dram reliability

ramulator-pim

A fast and flexible simulation infrastructure for exploring general-purpose processing-in-memory (PIM) architectures. Ramulator-PIM combines a widely-used simulator for out-of-order and in-order processors (ZSim) with Ramulator, a DRAM simulator with memory models for DDRx, LPDDRx, GDDRx, WIOx, HBMx, and HMCx. Ramulator is described in the IEEE ...

● C++ 🍴 11 ☆ 29 ⓘ 6 📄 0 Updated 19 days ago

SMASH

SMASH is a hardware-software cooperative mechanism that enables highly-efficient indexing and storage of sparse matrices. The key idea of SMASH is to compress sparse matrices with a hierarchical bitmap compression format that can be accelerated from hardware.

Described by Kanellopoulos et al. (MICRO '19)
<https://people.inf.ethz.ch/omutlu/pub/SMA...>

● C 🍴 1 ☆ 6 ⓘ 0 📄 0 Updated on May 17

MQSim

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 implementations, steady-state SSD conditions, and the full end-to-end latency of requests in modern SSDs. It is described in detail in the FAST 2018 paper by A...

● C++ 🍴 MIT 🍴 54 ☆ 62 ⓘ 10 📄 1 Updated on May 15

Apollo

Apollo is an assembly polishing algorithm that attempts to correct the errors in an assembly. It can take multiple set of reads in a single run and polish the assemblies of genomes of any size. Described in the Bioinformatics journal paper (2020) by Firtina et al. at

<https://people.inf.ethz.ch/omutlu/pub/apollo-technology-independent-genome-assem...>

● C++ 🍴 GPL-3.0 🍴 1 ☆ 12 ⓘ 0 📄 0 Updated on May 10

ramulator

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 IEEE CAL 2015 paper by Kim et al. at
http://users.ece.cmu.edu/~omutlu/pub/ramulator_dram_simulator-ieee-cal15.pdf

● C++ 🍴 MIT 🍴 93 ☆ 170 ⓘ 37 📄 2 Updated on Apr 13

Shifted-Hamming-Distance

Source code for the Shifted Hamming Distance (SHD) filtering mechanism for sequence alignment. Described in the Bioinformatics journal paper (2015) by Xin et al. at
http://users.ece.cmu.edu/~omutlu/pub/shifted-hamming-distance_bioinformatics15_proofs.pdf

● C 🍴 GPL-2.0 🍴 5 ☆ 20 ⓘ 0 📄 1 Updated on Mar 29

SneakySnake

The first and the only pre-alignment filtering algorithm that works on all modern high-performance computing architectures. It works efficiently and fast on CPU, FPGA, and GPU architectures and that greatly (by more than two orders of magnitude) expedites sequence alignment calculation. Described by Alser et al. (preliminary version at <https://a...>)

● VHDL 🍴 GPL-3.0 🍴 3 ☆ 11 ⓘ 0 📄 0 Updated on Mar 10

AirLift

AirLift is a tool that updates mapped reads from one reference genome to another. Unlike existing tools, It accounts for regions not shared between the two reference genomes and enables remapping across all parts of the references. Described by Kim et al. (preliminary version at <http://arxiv.org/abs/1912.08735>)

● C 🍴 0 ☆ 3 ⓘ 0 📄 0 Updated on Feb 19

GPGPUSim-Ramulator

The source code for GPGPUSim+Ramulator simulator. In this version, GPGPUSim uses Ramulator to simulate the DRAM. This simulator is used to produce some of the

End of Slides on More Detailed
Research Overview