

# Microarchitecture Research Panel @ MICRO 2021

Onur Mutlu

[omutlu@gmail.com](mailto:omutlu@gmail.com)

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

20 October 2021

MICRO 2021 Panel

**SAFARI**

**ETH** zürich

# Why Do We Do Computing?

# To Solve Problems

# To Gain Insight

To Enable  
a Better Life & Future

# How Does a Computer Solve Problems?

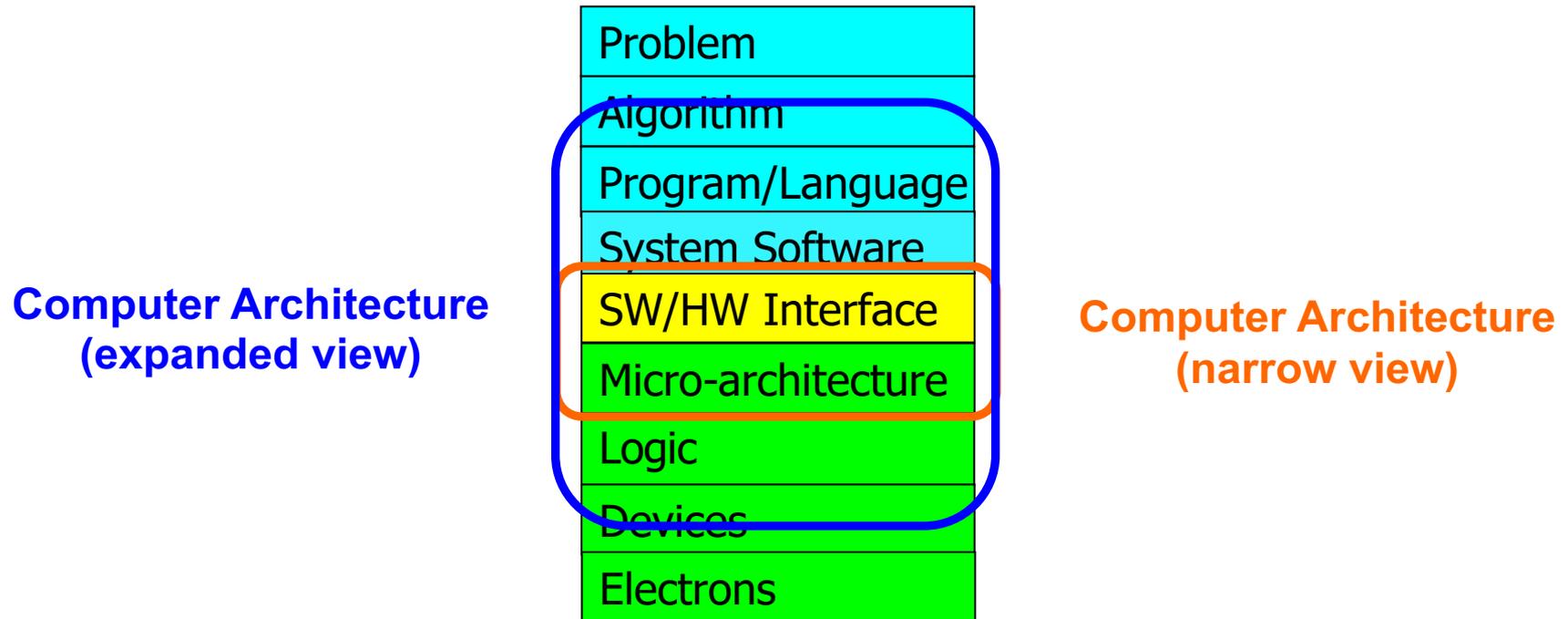
# Orchestrating Electrons

In today's dominant technologies

# How Do Problems Get Solved by Electrons?

# The Transformation Hierarchy

---

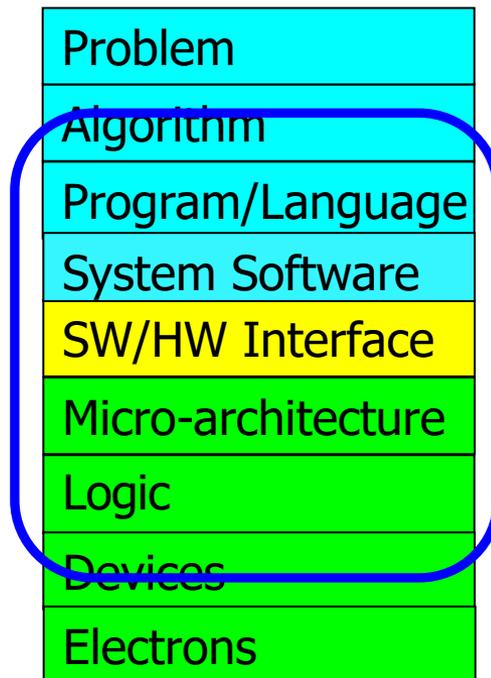


# Axiom

---

To achieve the highest **energy efficiency and performance**  
(and also dependability, security, safety):

**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**

# Challenge and Opportunity for Future

---

Reliable, Secure, Safe

## Sustainable and Energy Efficient

## High Performance

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

## Personalized and Private

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

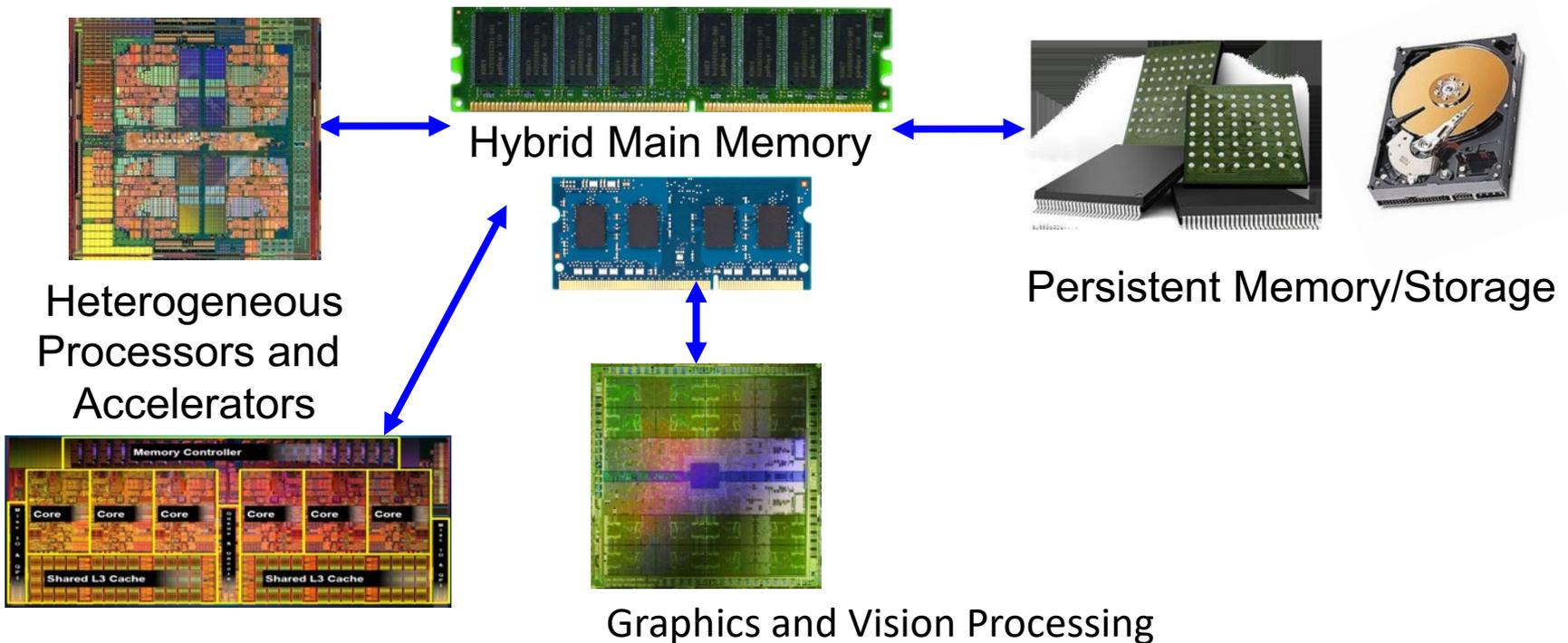
# Four Key Current Directions

---

- 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, ...**

# Current Research Mission

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



**Build fundamentally better architectures**

# Panel Question 1

---

- External or geographic influences on the microarchitecture research environment.
- What do you think is different about the environment in which you and your colleagues perform your research, and what do you think is similar across multiple regions of the globe.
- For example – academic research funding in different parts of the world is different; so are national or regional priorities which may influence industrial policy and hence the research environment.

# Panel Question 1

---

- Globally and ideally, we should be working together freely to solve the huge problems we face, exploring many diverse and big ideas, without borders or external barriers
- Unfortunately, funding issues & world+local politics & academic “merit” systems & review systems affect this goal
  - Collaboration
  - Immigration
  - Focus on large advances
  - Funding
  - Reviewing biases
- Switzerland & ETH: Relatively free, ample funding (so far), yet still affected by global+local politics

# Panel Question 2

---

- How do you see the similarities and/or difference influencing the microarchitecture research community?

# Panel Question 2

---

- Negative global+local politics is not good for collaboration, funding and progress
- Flawed academic merit systems are similarly problematic
- Ditto for issues in review systems
- All take useful cycles away from fast & large progress

# Panel Question 3

---

- What do you think are the most important microarchitecture research areas that researchers in your region should invest in, and why?

# Challenge and Opportunity for Future

---

**Reliable, Secure, Safe**

## Sustainable and Energy Efficient

## High Performance

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

## Personalized and Private

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

Hardware systems that

fundamentally

guarantee

robustness

(security, safety, reliability)

---

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

Computing Architectures

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

# Computing Architectures with Maximal Efficiency

Challenge and Opportunity for Future

---

Computing systems for

Genomics

Medicine

Health

Climate

...

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

## **Data-Aware (Expressive)**

## **Computing Architectures**

# Fundamentally Better Architectures

---

**Data-centric**

**Data-driven**

**Data-aware**

# Panel Question 4

---

- What areas you think are less important for researchers to investigate.

**Diversity is good**

**Avoid bias against  
any topic**

# Suggestions to Reviewers

---

- Be fair; you do not know it all
- Be open-minded; you do not know it all
- Be accepting of diverse research methods: there is no single way of doing research or writing papers
- Be constructive, not destructive
- Enable heterogeneity, but do **not** have double standards...

**Do not block or delay scientific progress for non-reasons**

---

# Microarchitecture Research Panel @ MICRO 2021

Onur Mutlu

[omutlu@gmail.com](mailto:omutlu@gmail.com)

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

20 October 2021

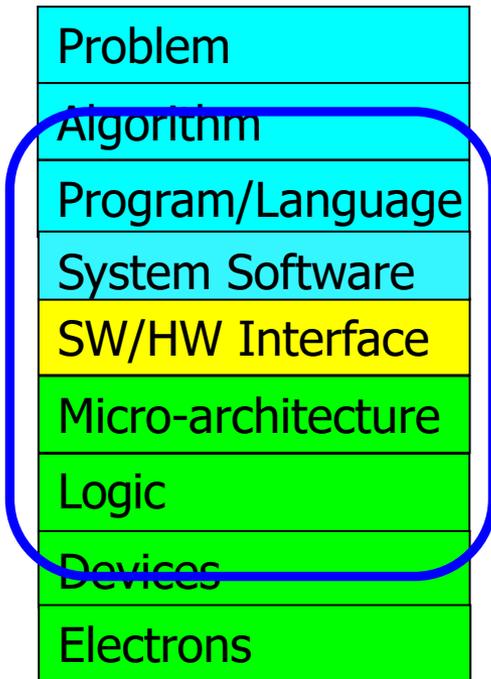
MICRO 2021 Panel

**SAFARI**

**ETH** zürich

# 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

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

---



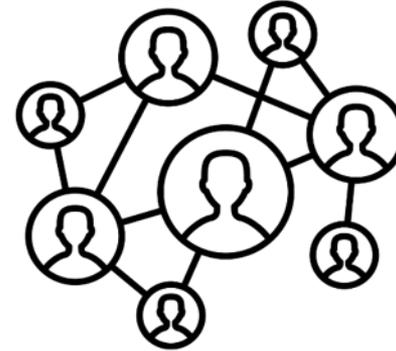
## In-memory Databases

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



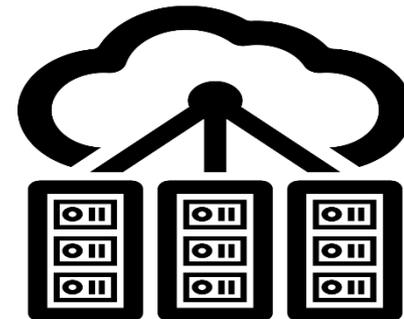
## In-Memory Data Analytics

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



## Graph/Tree Processing

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

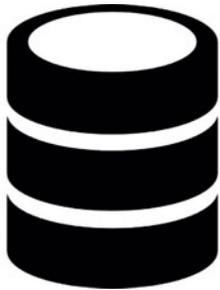


## Datacenter Workloads

[Kanev+ (Google), ISCA'15]

# Data Overwhelms Modern Machines

---



**In-memory Databases**



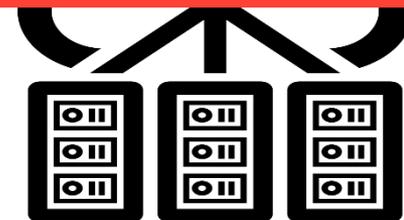
**Graph/Tree Processing**

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

# VP9



## Video Playback

Google's **video codec**

# VP9



## Video Capture

Google's **video codec**

# Data Overwhelms Modern Machines



**Chrome**



**TensorFlow Mobile**

Data → performance & energy bottleneck

**VP9**



**Video Playback**

Google's **video codec**

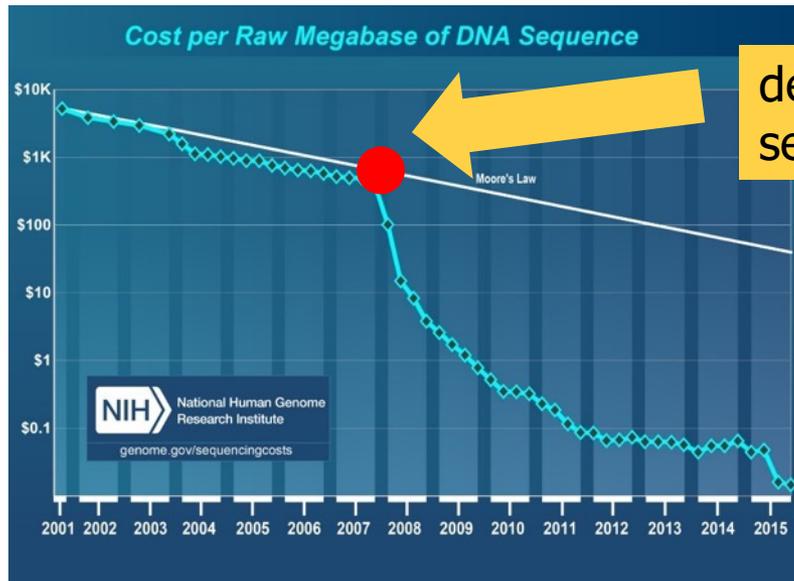
**VP9**



**Video Capture**

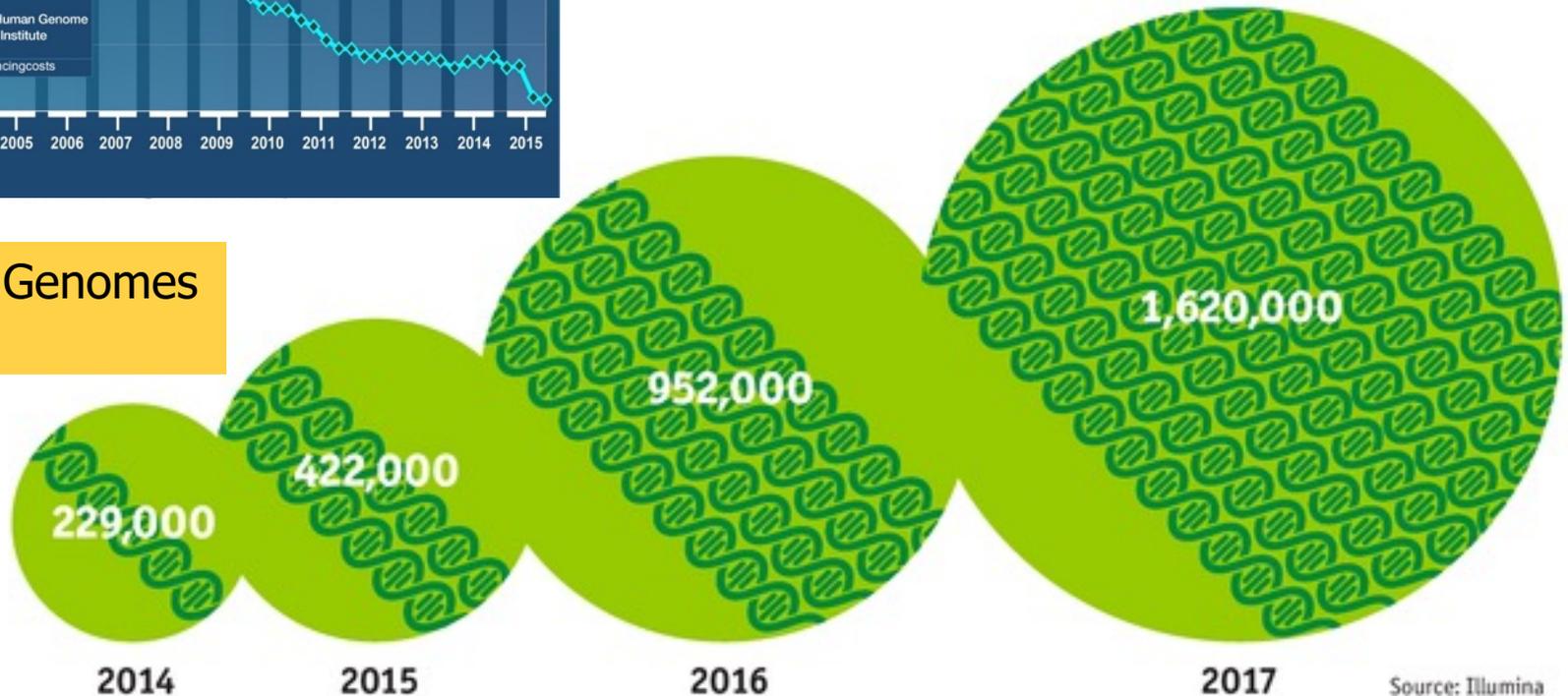
Google's **video codec**

# Data is Key for Future Workloads



development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

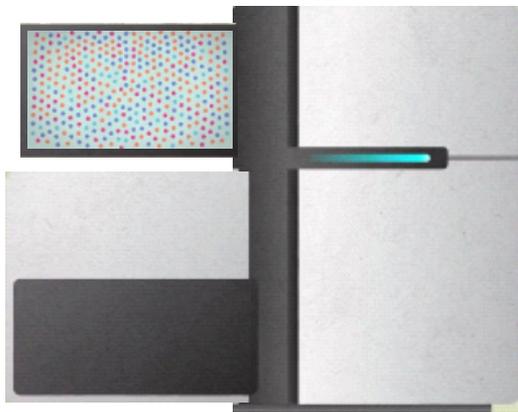


The Economist

**SAFARI**

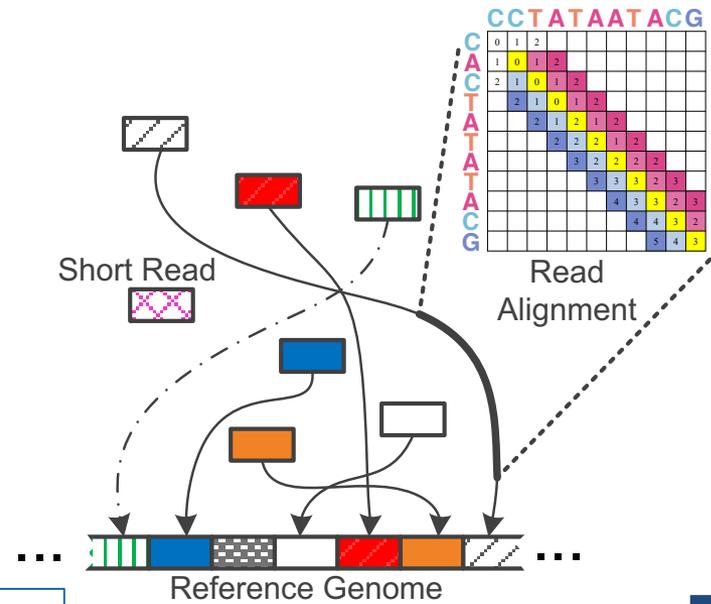
<http://www.economist.com/news/21631808-so-much-genetic-data-so-many-uses-genes-unzipped>

Source: Illumina



Billions of Short Reads

ATATATACGTA  
 TTTAGTACGTACGT  
 ATACGTA  
 CG CCCCTACGTA  
 CGTACTAGTACGT  
 TTAGTACGTACGT  
 TACGTA  
 TACGTA  
 TTTAAACGTA  
 CGTACTAGTACGT  
 GGGAGTACGTACGT



# 1 Sequencing

# Genome Analysis

# 2 Read Mapping

reference: TTTATCGCTTCCATGACGCAG

read1: ATCGCATCC

read2: TATCGCATC

read3: CATCCATGA

read4: CGCTTCCAT

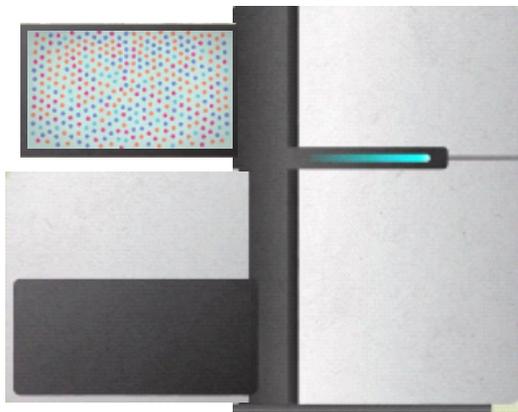
read5: CCATGACGC

read6: TTCCATGAC



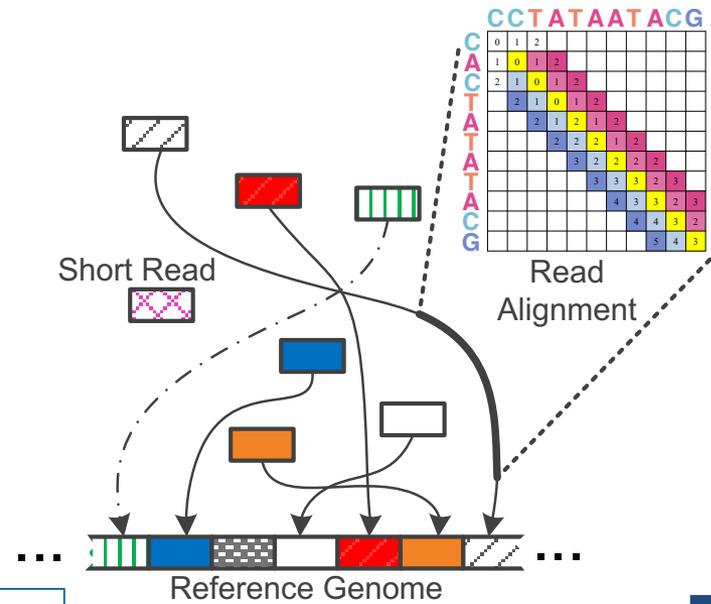
# 3 Variant Calling

# 4 Scientific Discovery



Billions of Short Reads

ATATATACGTA  
 TTTAGTACGTACGT  
 ATACGTA  
 CG CCCCTACGTA  
 CGTACTAGTACGT  
 TTAGTACGTACGT  
 TACGTA  
 TACGTA  
 TTTAAACGTA  
 CGTACTAGTACGT  
 GGGAGTACGTACGT



**1 Sequencing**

**Genome Analysis**

**Read Mapping 2**

Data → performance & energy bottleneck

read4: CGCTTCCAT  
 read5: CCATGACGC  
 read6: TTCCATGAC



**3 Variant Calling**

**Scientific Discovery 4**

# 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.  
[[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<sup>†</sup><sup>✕</sup> Gurpreet S. Kalsi<sup>✕</sup> Zülal Bingöl<sup>∇</sup> Can Firtina<sup>◇</sup> Lavanya Subramanian<sup>‡</sup> Jeremie S. Kim<sup>◇</sup><sup>†</sup>  
Rachata Ausavarungnirun<sup>○</sup> Mohammed Alser<sup>◇</sup> Juan Gomez-Luna<sup>◇</sup> Amirali Boroumand<sup>†</sup> Anant Nori<sup>✕</sup>  
Allison Scibisz<sup>†</sup> Sreenivas Subramoney<sup>✕</sup> Can Alkan<sup>∇</sup> Saugata Ghose<sup>\*†</sup> Onur Mutlu<sup>◇</sup><sup>∇</sup>  
<sup>†</sup>Carnegie Mellon University <sup>✕</sup>Processor Architecture Research Lab, Intel Labs <sup>∇</sup>Bilkent University <sup>◇</sup>ETH Zürich  
<sup>‡</sup>Facebook <sup>○</sup>King Mongkut's University of Technology North Bangkok <sup>\*</sup>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<sup>◇</sup> Mohammed Alser<sup>◇</sup> Damla Senol Cali<sup>✕</sup>

Dionysios Diamantopoulos<sup>▽</sup> Juan Gómez-Luna<sup>◇</sup>

Henk Corporaal<sup>\*</sup> Onur Mutlu<sup>◇✕</sup>

<sup>◇</sup>*ETH Zürich*    <sup>✕</sup>*Carnegie Mellon University*

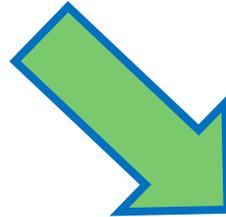
<sup>\*</sup>*Eindhoven University of Technology*    <sup>▽</sup>*IBM Research Europe*

# Future of Genome Sequencing & Analysis

---



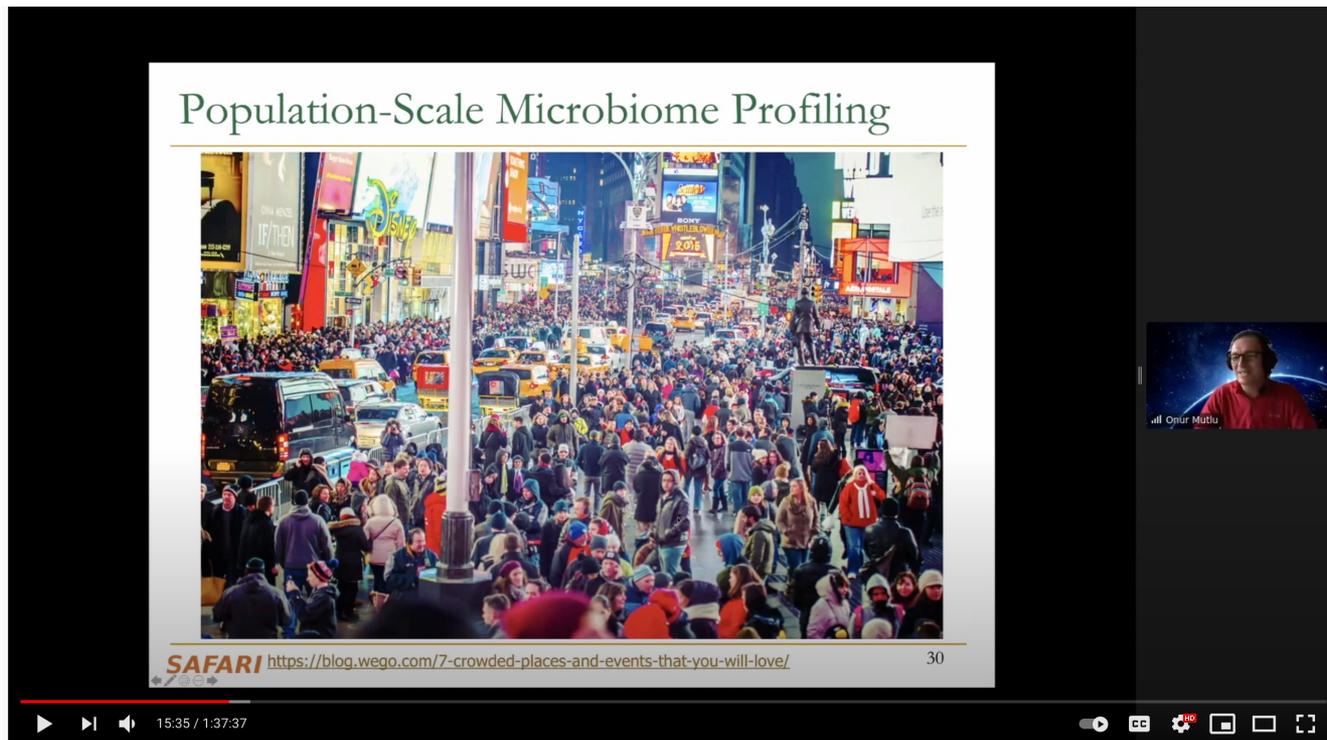
MinION from ONT



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 \(at IEEE Micro, 2020\)](#)]



The screenshot shows a video player interface. The main content is a slide titled "Population-Scale Microbiome Profiling" which features a photograph of a very crowded city street at night, likely Times Square in New York City, with many people and cars. Below the photo is a URL: <https://blog.wego.com/7-crowded-places-and-events-that-you-will-love/>. The slide number "30" is visible in the bottom right corner. On the right side of the video player, there is a small inset video of the speaker, Onur Mutlu, with his name "Onur Mutlu" displayed below it. The video player controls at the bottom show a play button, a progress bar at 15:35 / 1:37:37, and various icons for volume, full screen, and sharing.

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>

Our Goal

---

Computing systems for

Genomics

Medicine

Health

AI/ML

---

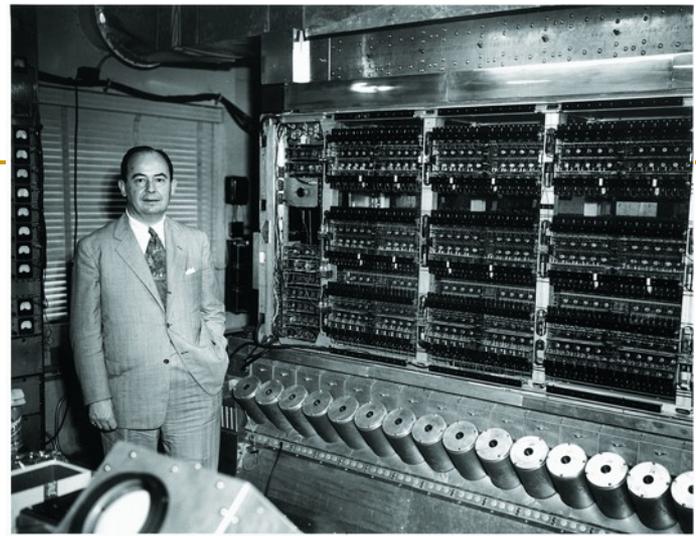
# Data Overwhelms Modern Machines ...

---

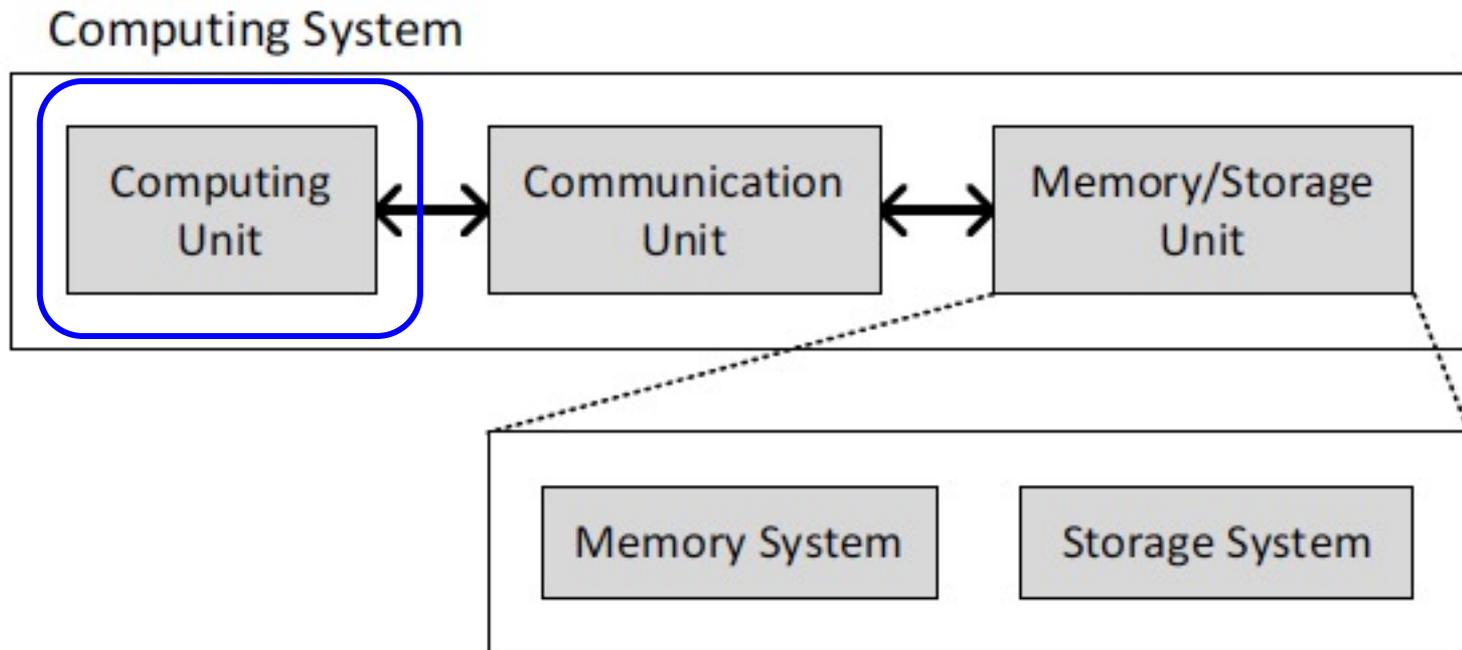
- Storage/memory capability
- Communication capability
- Computation capability
- Greatly impacts robustness, energy, performance, cost

# A Computing System

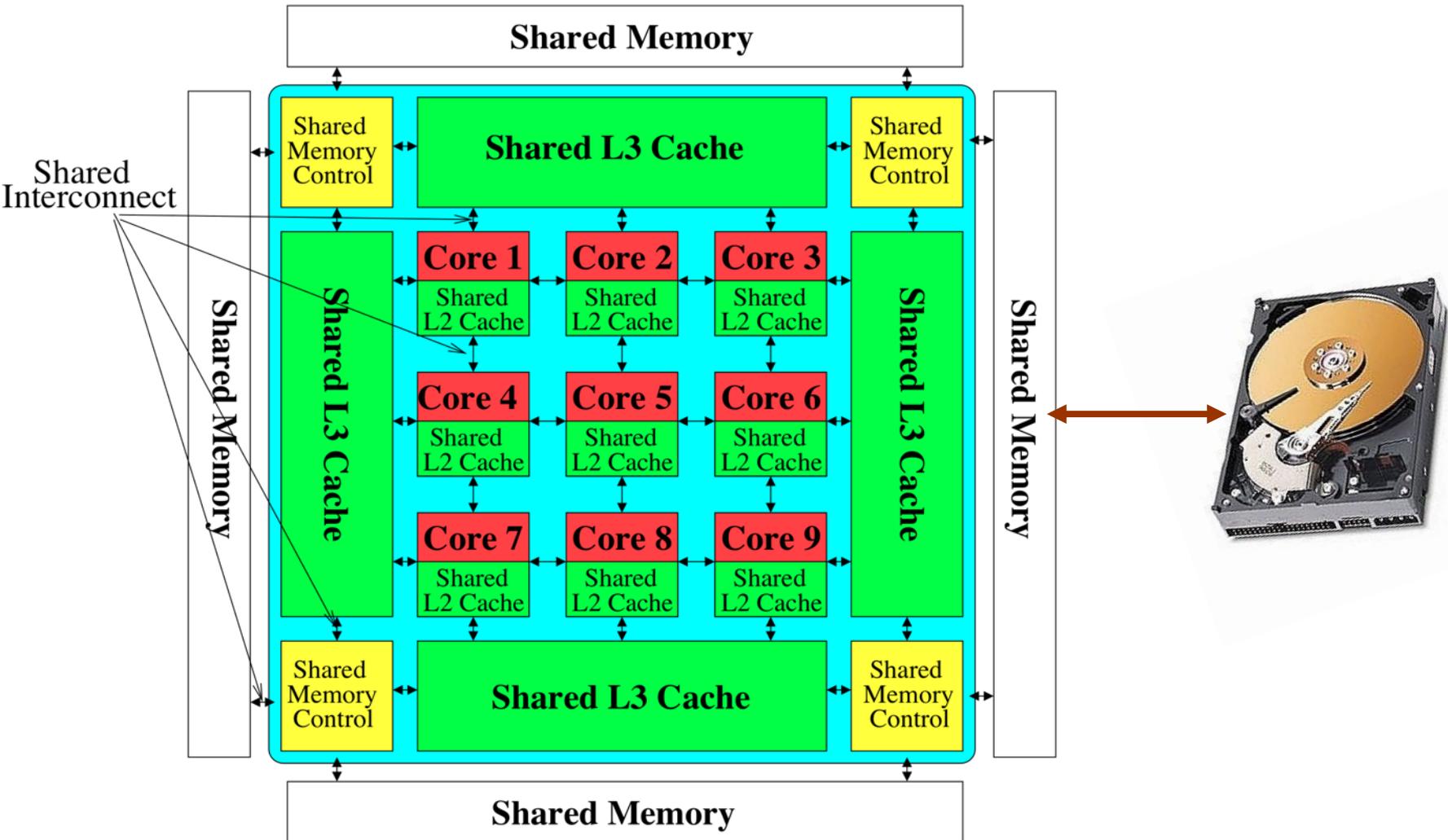
- Three key components
- Computation
- Communication
- Storage/memory



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 Boroumand<sup>1</sup>

Saugata Ghose<sup>1</sup>

Youngsok Kim<sup>2</sup>

Rachata Ausavarungnirun<sup>1</sup>

Eric Shiu<sup>3</sup>

Rahul Thakur<sup>3</sup>

Daehyun Kim<sup>4,3</sup>

Aki Kuusela<sup>3</sup>

Allan Knies<sup>3</sup>

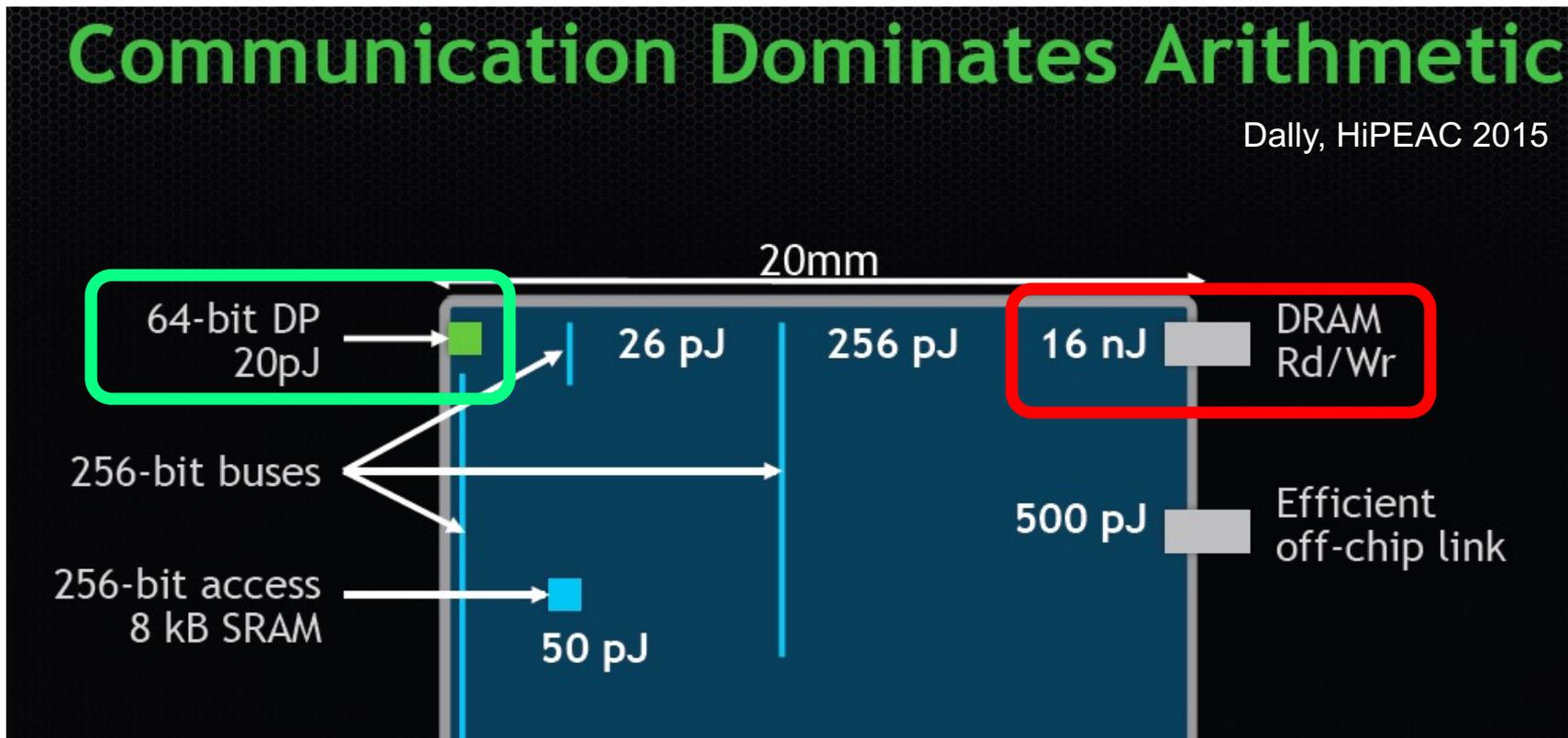
Parthasarathy Ranganathan<sup>3</sup>

Onur Mutlu<sup>5,1</sup>

# Data Movement vs. Computation Energy

## Communication Dominates Arithmetic

Dally, HiPEAC 2015



A memory access consumes  $\sim 100-1000X$  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**

# Fundamentally Better Architectures

---

**Data-centric**

**Data-driven**

**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

# Do We Want This?

---

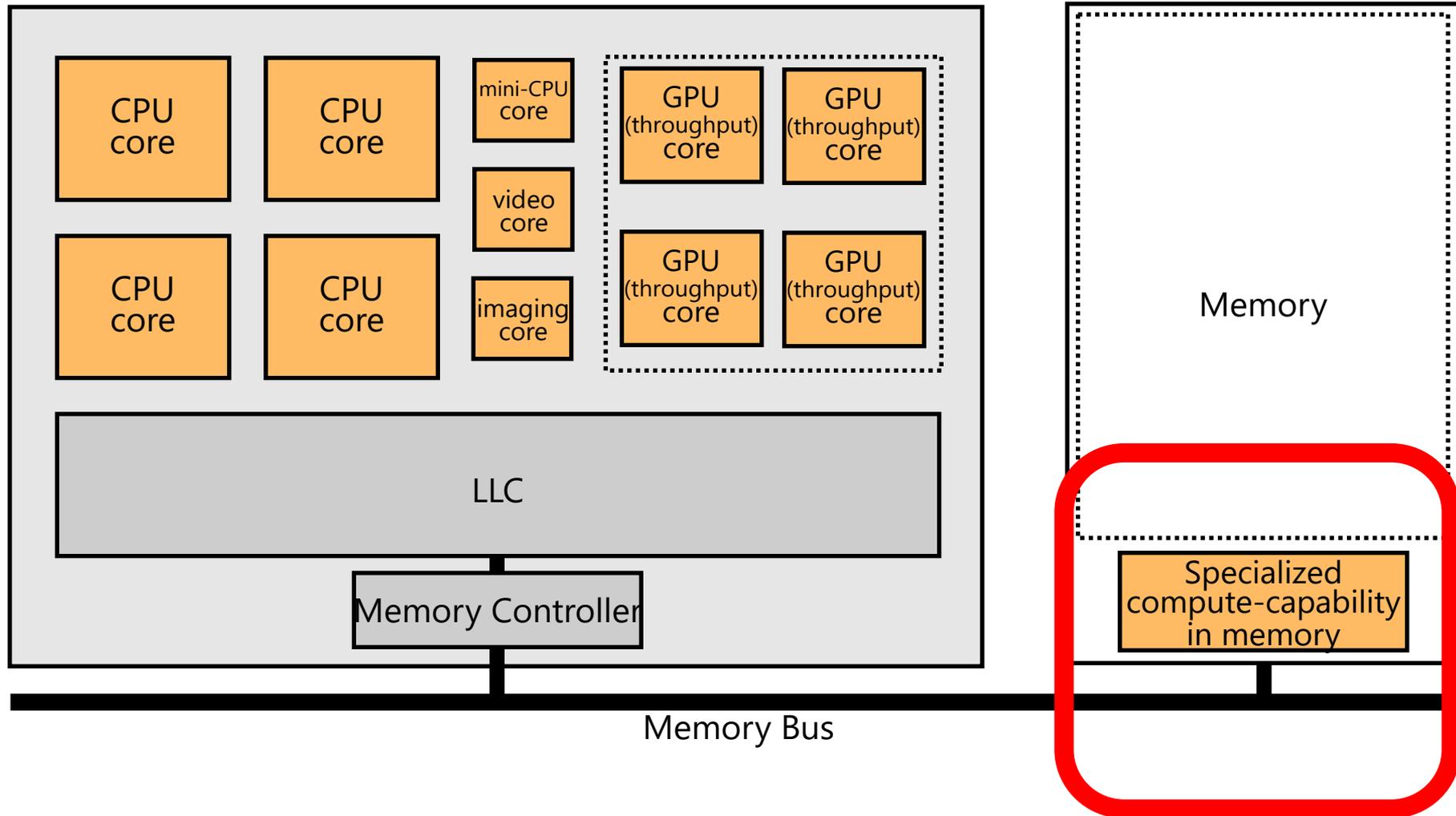


# Or This?

---



# Mindset: Memory-Centric Computing



**Memory similar to a "conventional" accelerator**

# PIM Review and Open Problems

---

## A Modern Primer on Processing in Memory

Onur Mutlu<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirun<sup>d</sup>

*SAFARI Research Group*

<sup>a</sup>*ETH Zürich*

<sup>b</sup>*Carnegie Mellon University*

<sup>c</sup>*University of Illinois at Urbana-Champaign*

<sup>d</sup>*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<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirun<sup>d</sup>

SAFARI Research Group

<sup>a</sup>ETH Zürich

<sup>b</sup>Carnegie Mellon University

<sup>c</sup>University of Illinois at Urbana-Champaign

<sup>d</sup>King Mongkut's University of Technology North Bangkok

---

## Abstract

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

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

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

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

<b>1 Introduction</b>	<b>2</b>
<b>2 Major Trends Affecting Main Memory</b>	<b>4</b>
<b>3 The Need for Intelligent Memory Controllers to Enhance Memory Scaling</b>	<b>6</b>
<b>4 Perils of Processor-Centric Design</b>	<b>9</b>
<b>5 Processing-in-Memory (PIM): Technology Enablers and Two Approaches</b>	<b>12</b>
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
<b>6 Processing Using Memory (PUM)</b>	<b>14</b>
6.1 RowClone . . . . .	14
6.2 Ambit . . . . .	15
6.3 Gather-Scatter DRAM . . . . .	17
6.4 In-DRAM Security Primitives . . . . .	17
<b>7 Processing Near Memory (PNM)</b>	<b>18</b>
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
<b>8 Enabling the Adoption of PIM</b>	<b>24</b>
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
<b>9 Conclusion and Future Outlook</b>	<b>31</b>

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-