

# Intelligent Architectures for Intelligent Machines

Onur Mutlu

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

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

3 May 2021

TU Vienna MiM (Mondays in Memory) Webinar

**SAFARI**

**ETH** zürich

Carnegie Mellon

## The Problem

---

Computing  
is Bottlenecked by Data

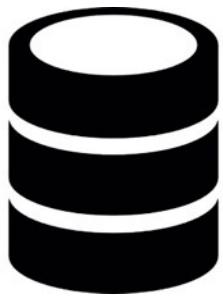
# Data is Key for AI, ML, Genomics, ...

---

- Important workloads are all data intensive
- They require rapid and efficient processing of large amounts of data
- Data is increasing
  - We can generate more than we can process

# Data is Key for Future Workloads

---



## In-memory Databases

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



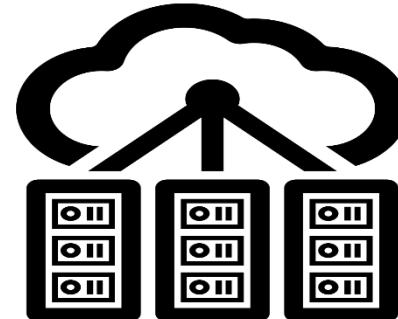
## Graph/Tree Processing

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



## In-Memory Data Analytics

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

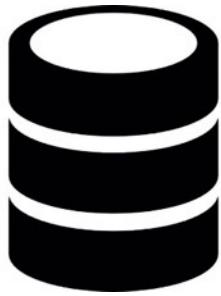


## Datacenter Workloads

[Kanев+ (Google), ISCA'15]

# Data Overwhelms Modern Machines

---



**In-memory Databases**



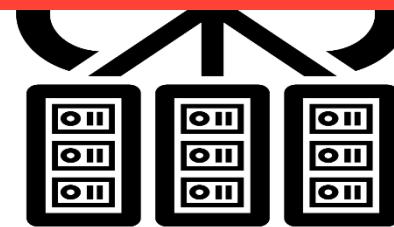
**Graph/Tree Processing**

Data → performance & energy bottleneck



**In-Memory Data Analytics**

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



**Datacenter Workloads**

[Kanев+ (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**

**SAFARI**

# Data Overwhelms Modern Machines



**Chrome**



**TensorFlow Mobile**

Data → performance & energy bottleneck



**Video Playback**

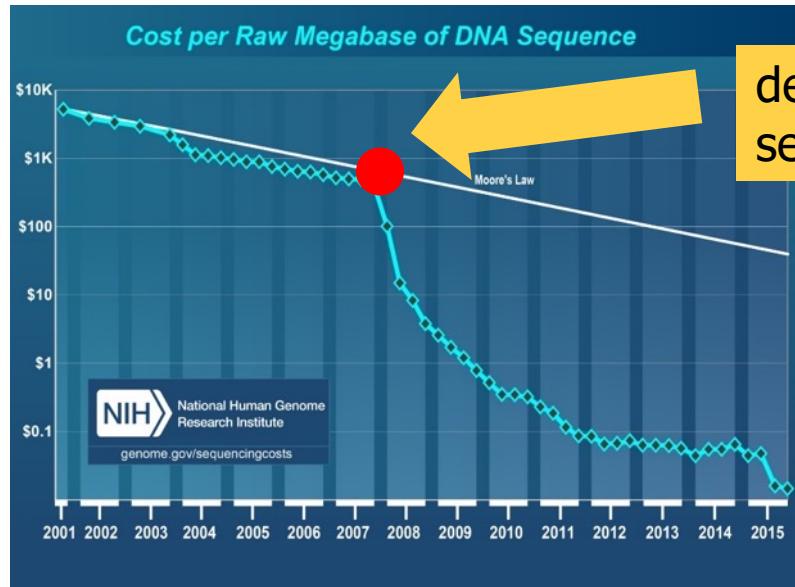
Google's **video codec**



**Video Capture**

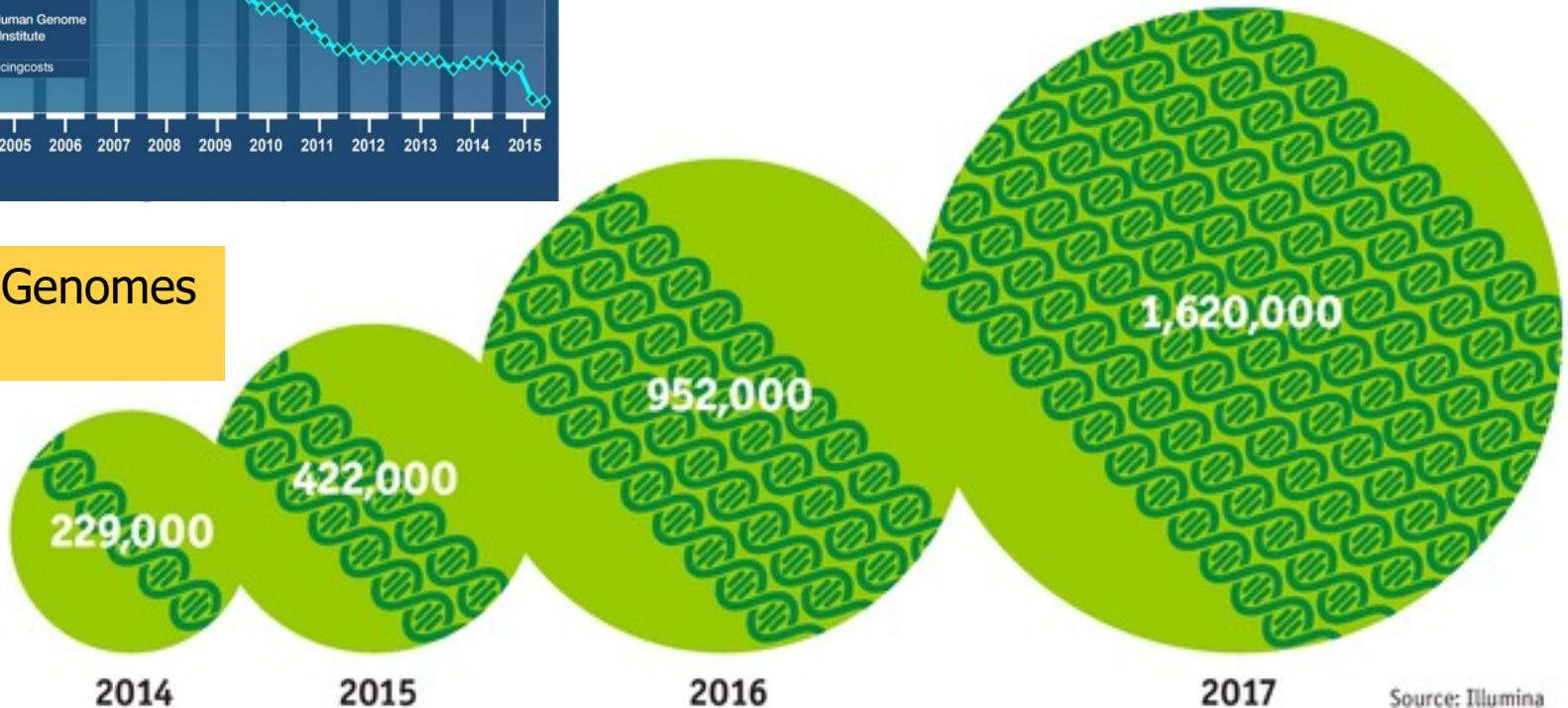
Google's **video codec**

# Data is Key for Future Workloads



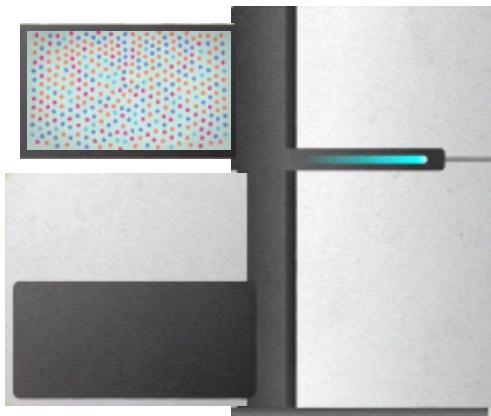
development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced



The Economist

Source: Illumina

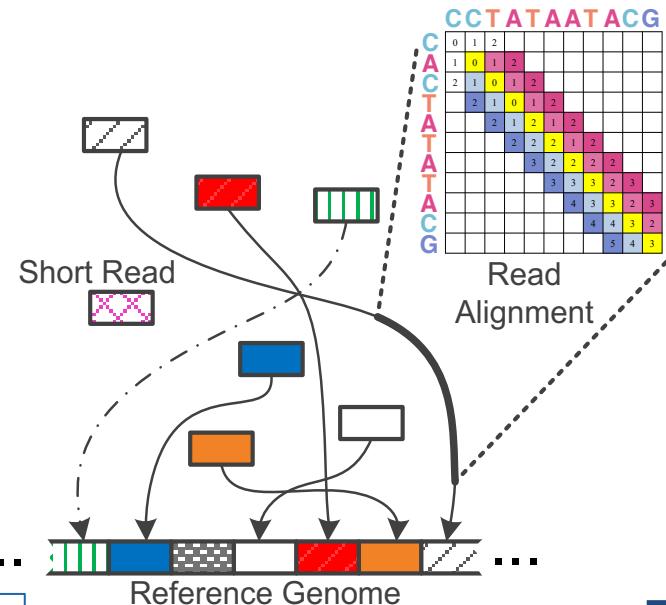


Billions of Short Reads

```
ATATATAACGTACGTACGT  
TTTAGTACGTACGTACGT  
ATACGTACTAGTACGTACGT  
ACGCCCCTACGTA  
ACGTACTAGTACGT  
TTAGTACGTACGT  
TACGTACTAAAGTACGT  
TACGTACTAGTACGT  
TTTAAAAACGTA  
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.  
[\[Open arxiv.org version\]](https://arxiv.org/abs/1804.00701)

# 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

---

- Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,

**"Accelerating Genome Analysis: A Primer on an Ongoing Journey"**

*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.*

[[Lightning Talk Video](#) (1.5 minutes)]

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

[[Talk Video](#) (18 minutes)]

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

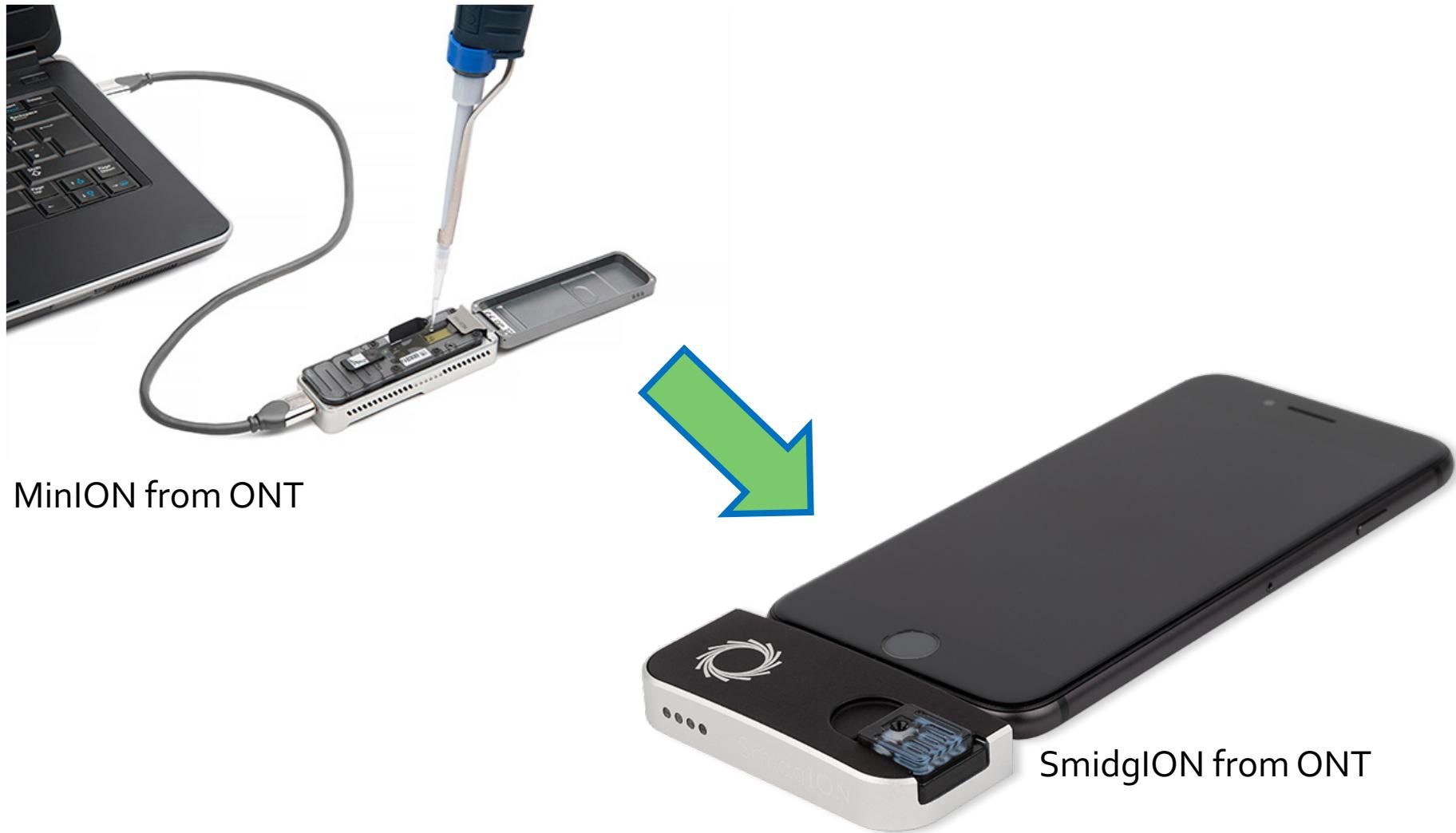
## **GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis**

Damla Senol Cali<sup>†✉</sup> Gurpreet S. Kalsi<sup>✉</sup> Zülal Bingöl<sup>▽</sup> Can Firtina<sup>◊</sup> Lavanya Subramanian<sup>‡</sup> Jeremie S. Kim<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>*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*

# Future of Genome Sequencing & Analysis



# More on Fast & Efficient Genome Analysis ...

- Onur Mutlu,

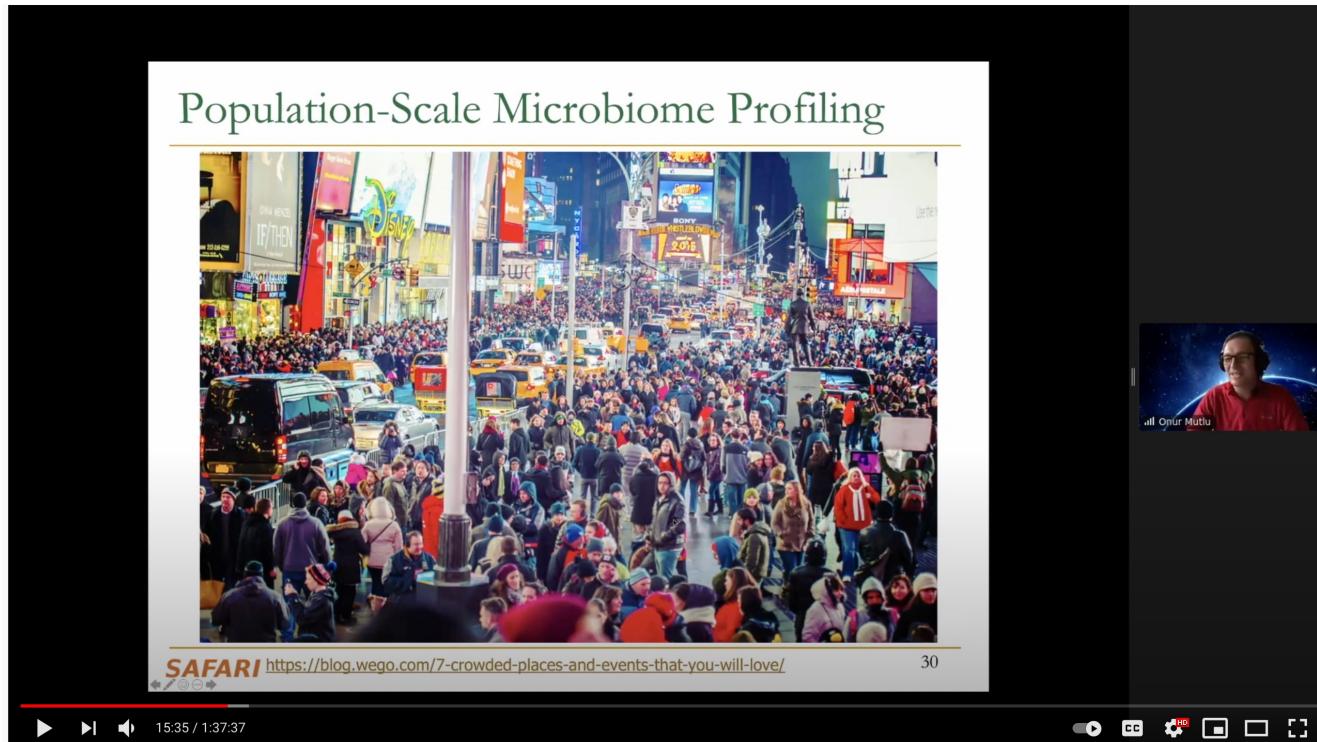
**"Accelerating Genome Analysis: A Primer on an Ongoing Journey"**

*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

1 like 35 dislike 0 SHARE SAVE ...



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>

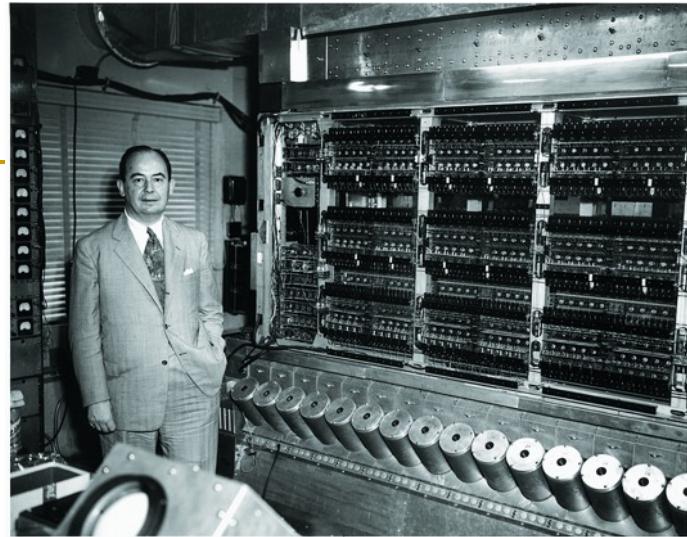
# Data Overwhelms Modern Machines ...

---

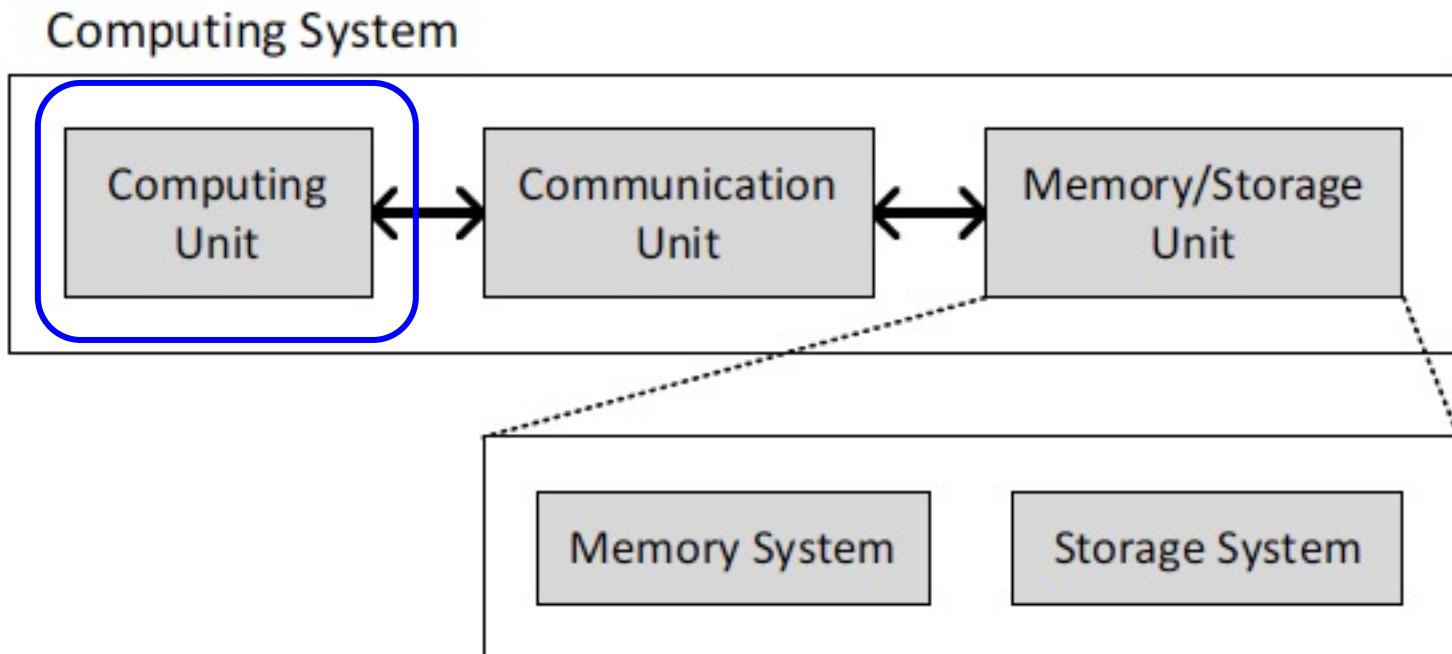
- Storage/memory capability
- Communication capability
- Computation capability
- Greatly impacts robustness, energy, performance, cost

# A Computing System

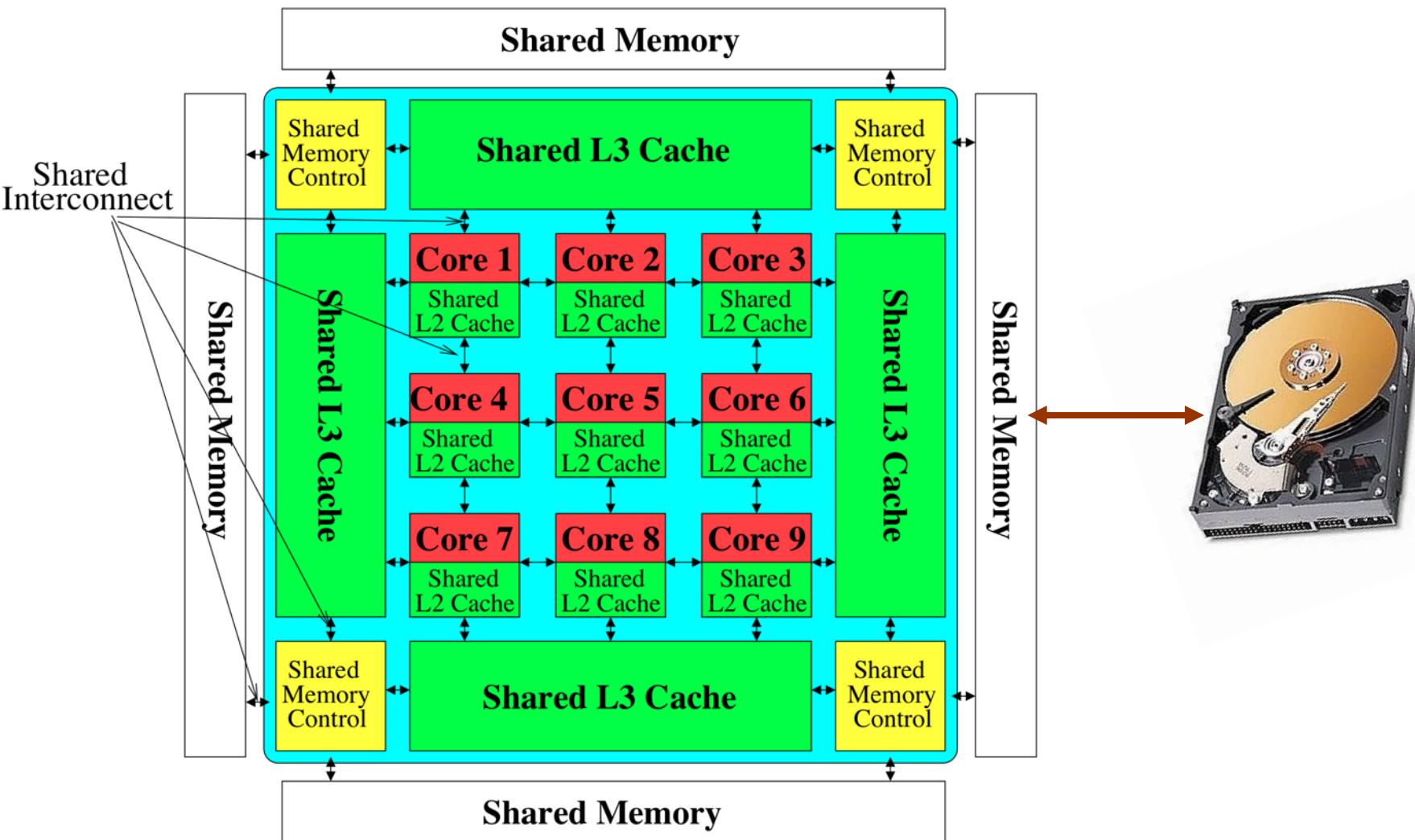
- Three key components
- Computation
- Communication
- Storage/memory



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**

# Data Overwhelms Modern Machines



**Chrome**



**TensorFlow Mobile**

Data → performance & energy bottleneck



**Video Playback**

Google's **video codec**



**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>

Rachata Ausavarungnirun<sup>1</sup>

Aki Kuusela<sup>3</sup>

Saugata Ghose<sup>1</sup>

Eric Shiu<sup>3</sup>

Allan Knies<sup>3</sup>

Youngsok Kim<sup>2</sup>

Rahul Thakur<sup>3</sup>

Parthasarathy Ranganathan<sup>3</sup>

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

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

# 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

# Processing in/near Memory: An Old Idea

---

- Stone, “[A Logic-in-Memory Computer](#),” IEEE TC 1970.

## A Logic-in-Memory Computer

HAROLD S. STONE

*Abstract*—If, as presently projected, the cost of microelectronic arrays in the future will tend to reflect the number of pins on the array rather than the number of gates, the logic-in-memory array is an extremely attractive computer component. Such an array is essentially a microelectronic memory with some combinational logic associated with each storage element.

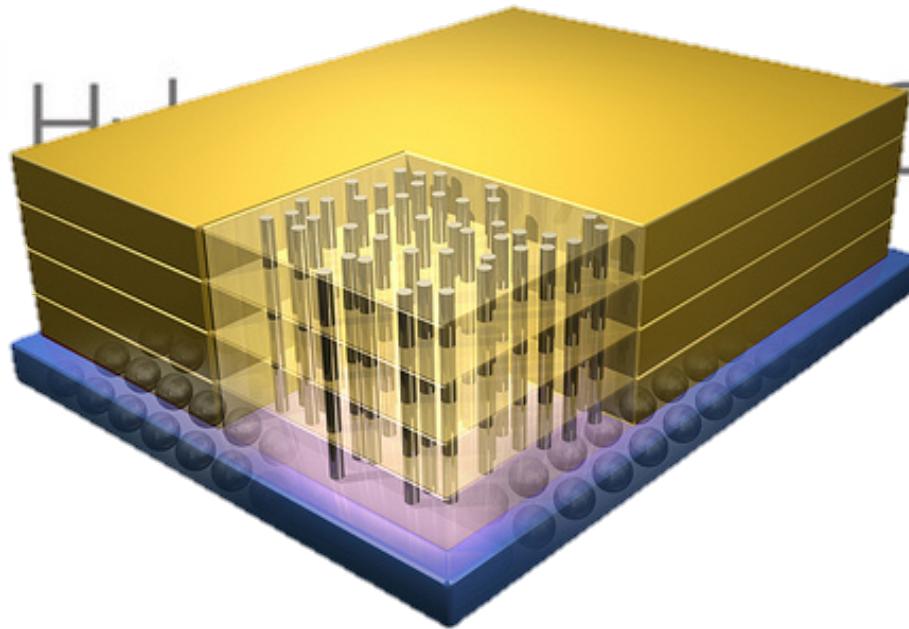
# Why In-Memory Computation Today?

---

- Push from Technology
  - DRAM Scaling at jeopardy
    - Controllers close to DRAM
    - Industry open to new memory architectures

# Why In-Memory Computation Today?

---



# Memory Scaling Issues Were Real

---

- Onur Mutlu,

**"Memory Scaling: A Systems Architecture Perspective"**

*Proceedings of the 5th International Memory*

*Workshop (IMW)*, Monterey, CA, May 2013. Slides

(pptx) (pdf)

EETimes Reprint

## Memory Scaling: A Systems Architecture Perspective

Onur Mutlu

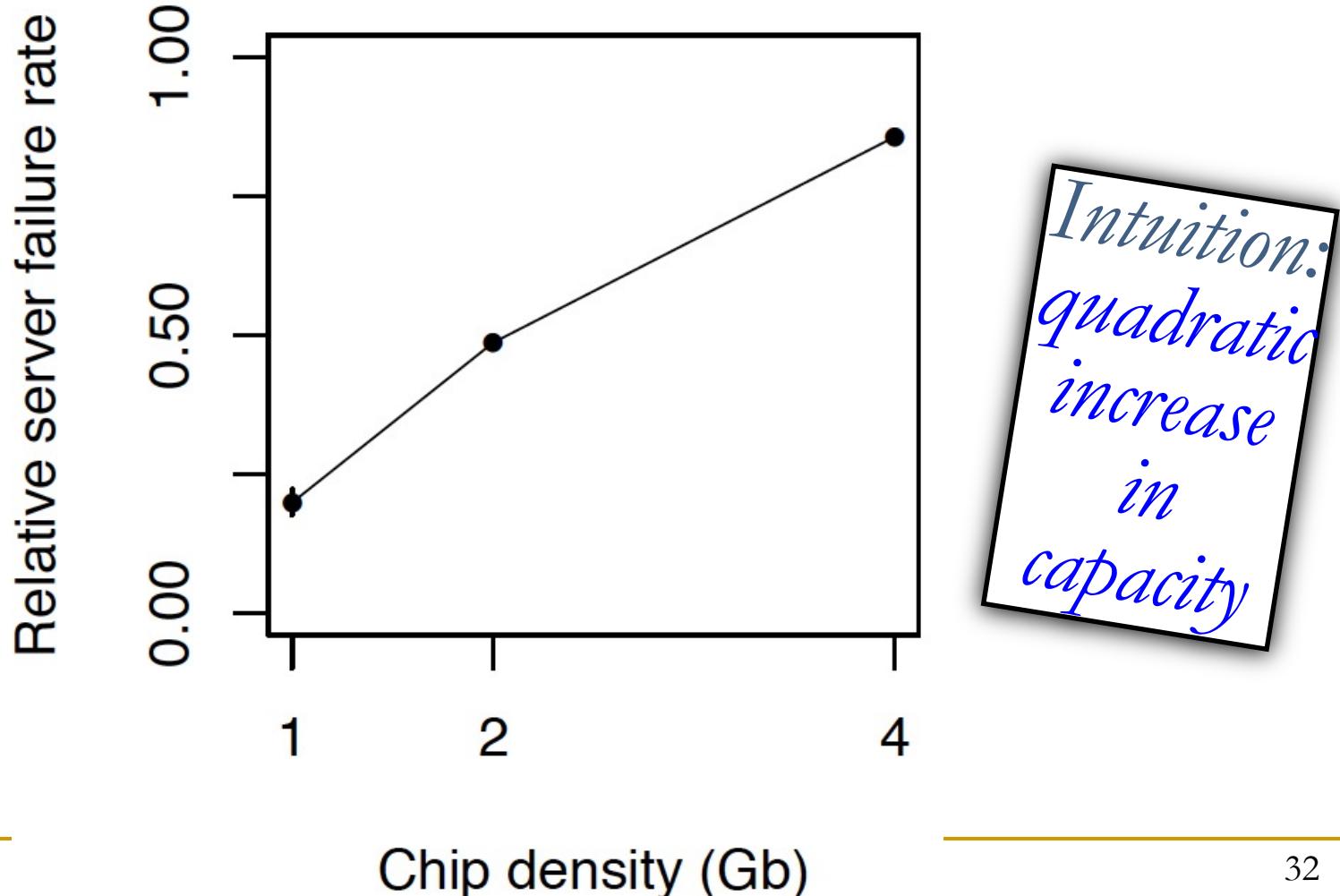
Carnegie Mellon University

onur@cmu.edu

<http://users.ece.cmu.edu/~omutlu/>

# As Memory Scales, It Becomes Unreliable

- Data from all of Facebook's servers worldwide
- Meza+, "Revisiting Memory Errors in Large-Scale Production Data Centers," DSN'15.



# Large-Scale Failure Analysis of DRAM Chips

---

- Analysis and modeling of memory errors found in all of Facebook's server fleet
- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu,  
**"Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field"**  
*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Rio de Janeiro, Brazil, June 2015.  
[[Slides \(pptx\)](#) ([pdf](#))] [[DRAM Error Model](#)]

Revisiting Memory Errors in Large-Scale Production Data Centers:  
Analysis and Modeling of New Trends from the Field

Justin Meza   Qiang Wu \*   Sanjeev Kumar \*   Onur Mutlu  
Carnegie Mellon University   \* Facebook, Inc.

# Infrastructures to Understand Such Issues



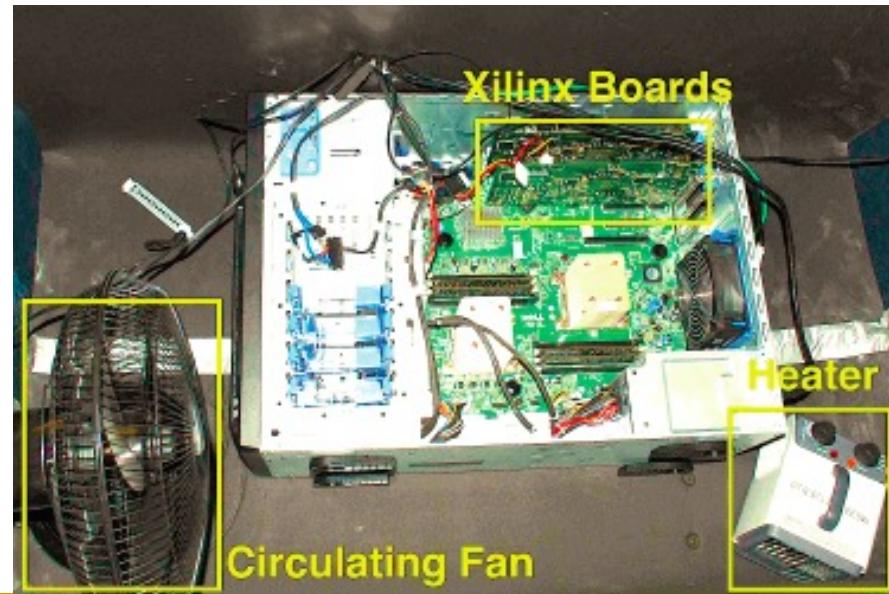
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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

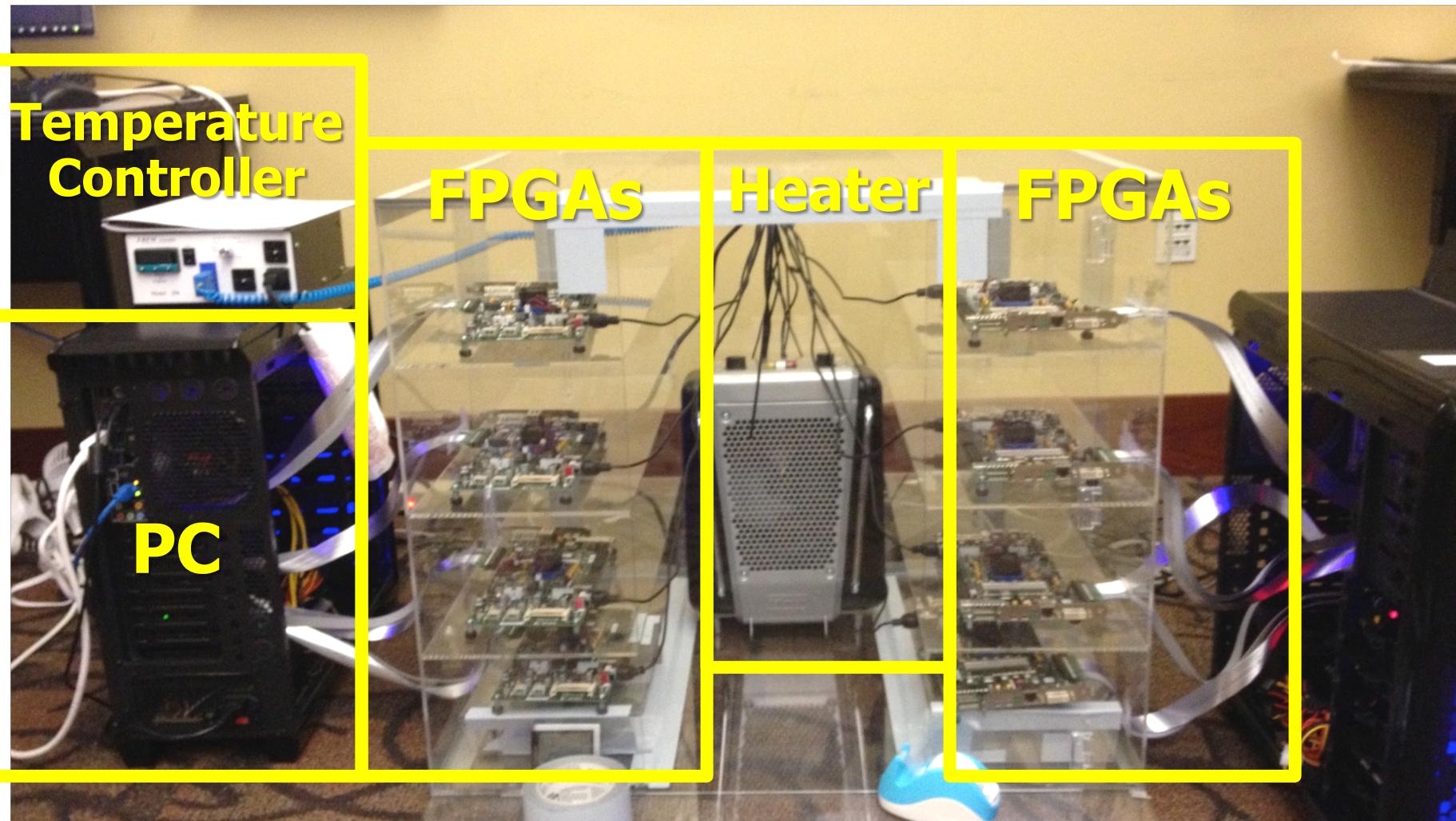
AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)

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)

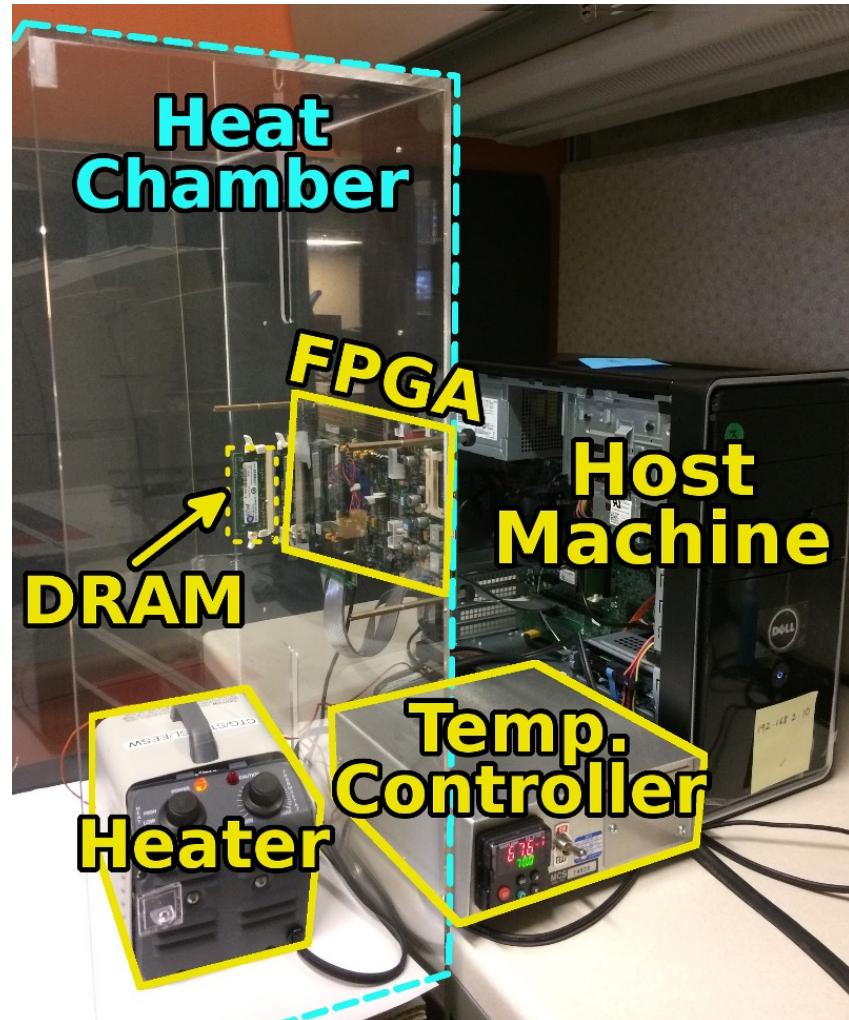


# Infrastructures to Understand Such Issues



# 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<sup>1,2,3</sup> Nandita Vijaykumar<sup>3</sup> Samira Khan<sup>4,3</sup> Saugata Ghose<sup>3</sup> Kevin Chang<sup>3</sup>  
Gennady Pekhimenko<sup>5,3</sup> Donghyuk Lee<sup>6,3</sup> Oguz Ergin<sup>2</sup> Onur Mutlu<sup>1,3</sup>

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

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

---

One can  
predictably induce errors  
in most DRAM memory chips

# The Story of RowHammer

- One can predictably induce bit flips in commodity DRAM chips
  - >80% of the tested DRAM chips are vulnerable
- First example of how 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

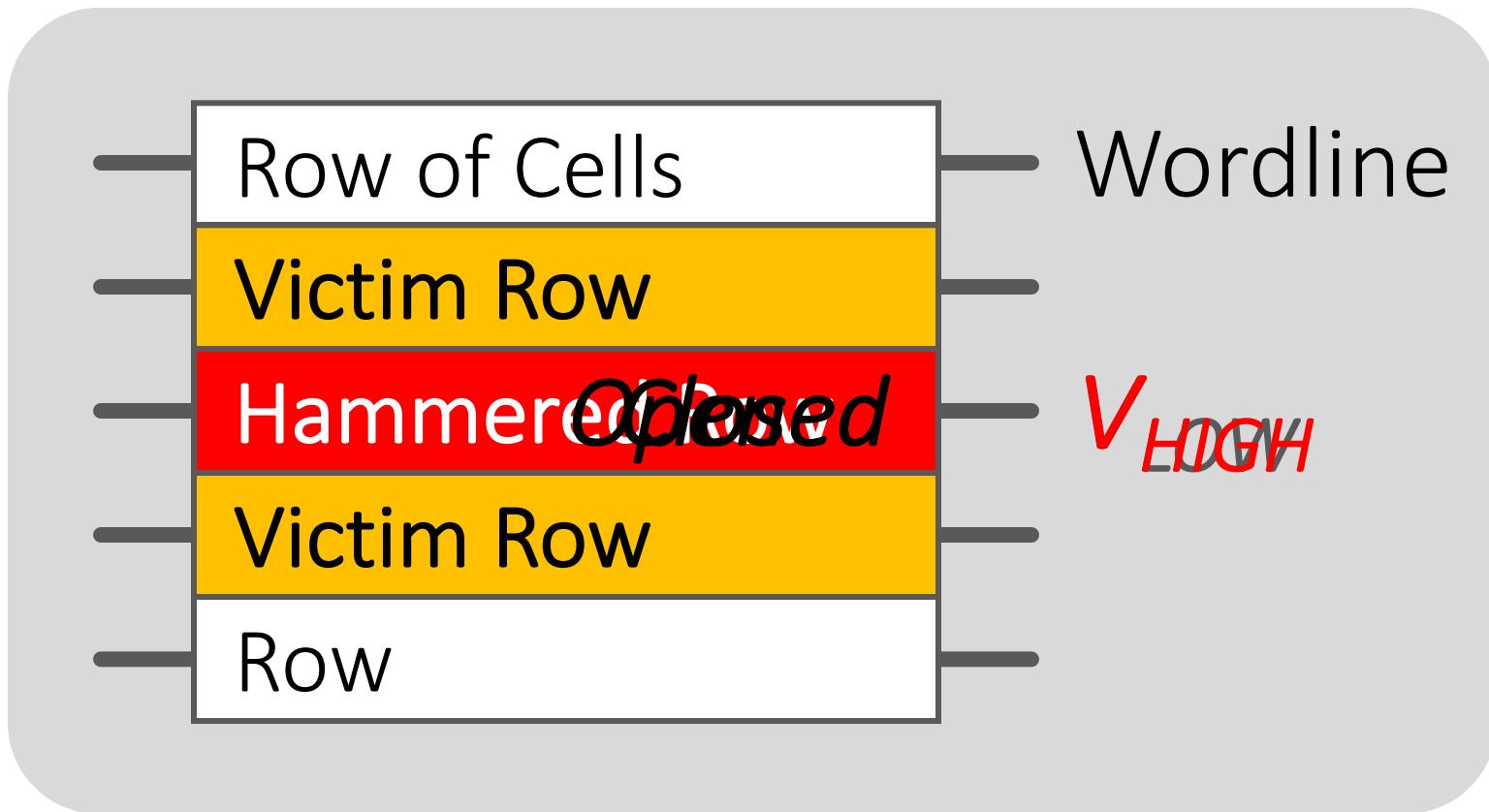
SHARE

f SHARE  
18276

tweet TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

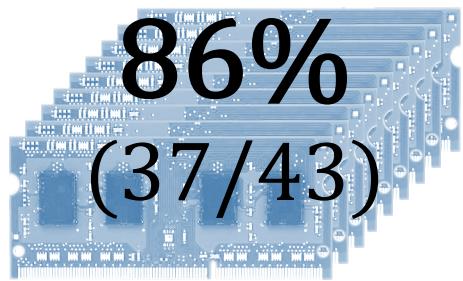
# Modern DRAM is Prone to Disturbance Errors



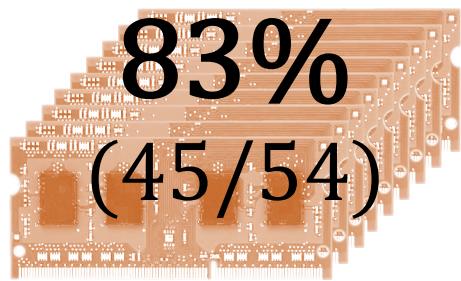
Repeatedly reading a row enough times (before memory gets refreshed) induces **disturbance errors** in adjacent rows in **most real DRAM chips you can buy today**

# Most DRAM Modules Are Vulnerable

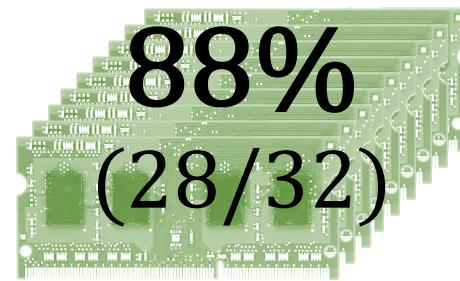
A company



B company



C company

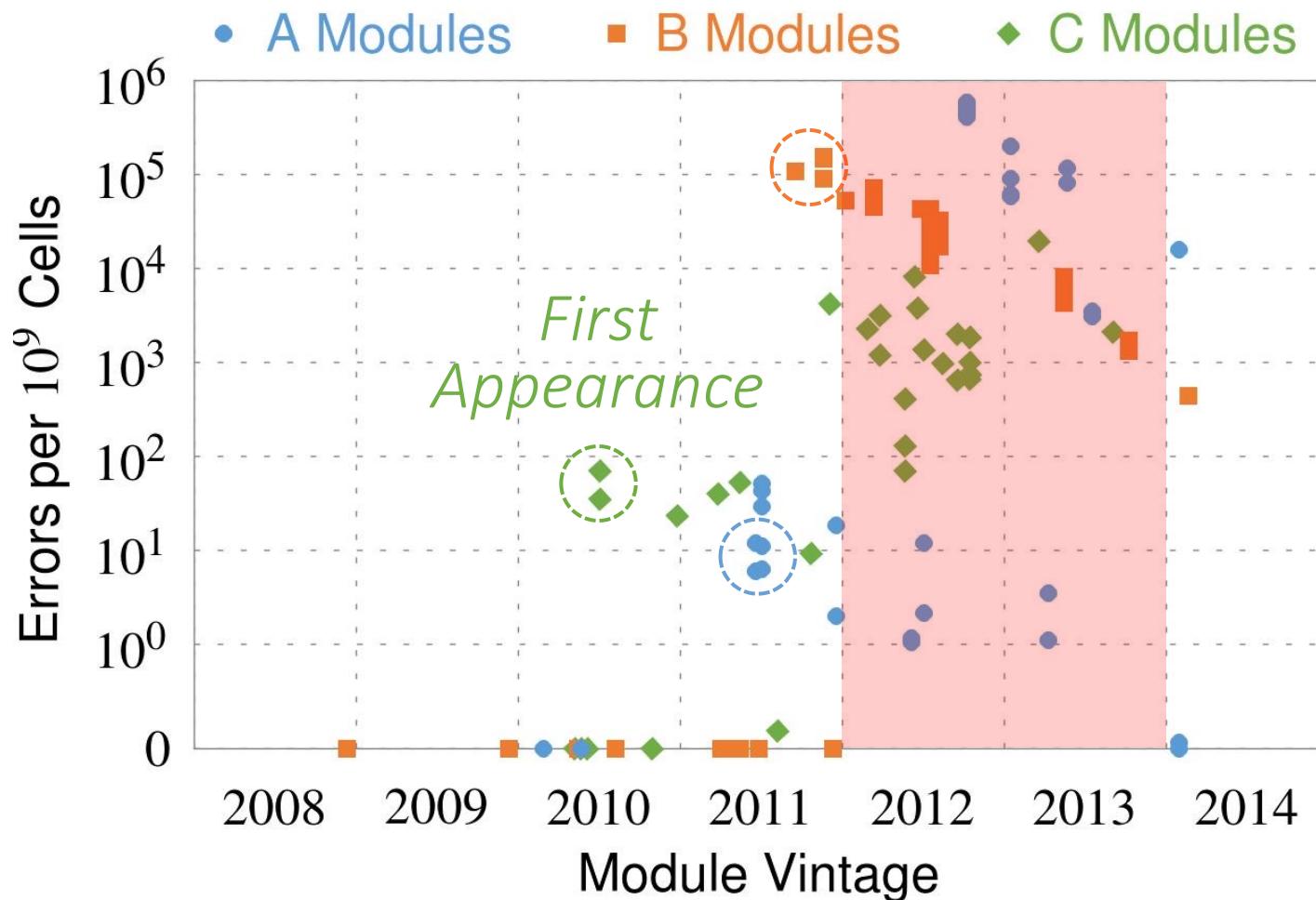


Up to  
 $1.0 \times 10^7$   
errors

Up to  
 $2.7 \times 10^6$   
errors

Up to  
 $3.3 \times 10^5$   
errors

# Recent DRAM Is More Vulnerable



*All modules from 2012–2013 are vulnerable*

# 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

# More Security Implications (I)

**"We can gain unrestricted access to systems of website visitors."**

www.iaik.tugraz.at ■

Not there yet, but ...



ROOT privileges for web apps!



29

Daniel Gruss (@lavados), Clémentine Maurice (@BloodyTangerine),  
December 28, 2015 — 32c3, Hamburg, Germany

Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript (DIMVA'16)

# More Security Implications (II)

**“Can gain control of a smart phone deterministically”**



Drammer: Deterministic Rowhammer  
Attacks on Mobile Platforms, CCS'16 45

# More Security Implications (VII)

- USENIX Security 2019

## Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks

Sanghyun Hong, Pietro Frigo<sup>†</sup>, Yiğitcan Kaya, Cristiano Giuffrida<sup>†</sup>, Tudor Dumitraş

*University of Maryland, College Park*

*†Vrije Universiteit Amsterdam*



### A Single Bit-flip Can Cause Terminal Brain Damage to DNNs

*One specific bit-flip in a DNN's representation leads to accuracy drop over 90%*

Our research found that a specific bit-flip in a DNN's bitwise representation can cause the accuracy loss up to 90%, and the DNN has 40-50% parameters, on average, that can lead to the accuracy drop over 10% when individually subjected to such single bitwise corruptions...

[Read More](#)

# More Security Implications (VIII)

## ■ USENIX Security 2020

### DeepHammer: Depleting the Intelligence of Deep Neural Networks through Targeted Chain of Bit Flips

Fan Yao

*University of Central Florida*  
*fan.yao@ucf.edu*

Adnan Siraj Rakin

*Arizona State University*  
*asrakin@asu.edu*

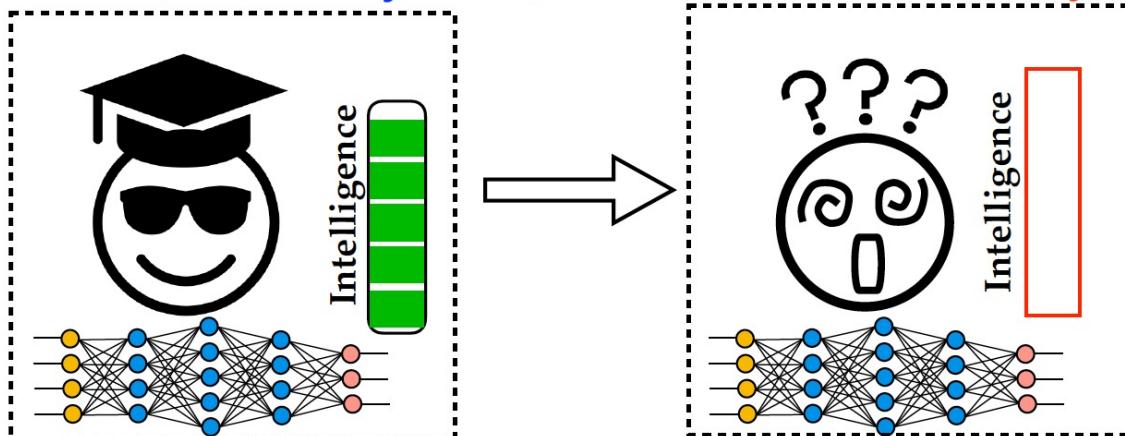
Deliang Fan

*dfan@asu.edu*

Degrade the inference accuracy to the level of Random Guess

Example: ResNet-20 for CIFAR-10, 10 output classes

Before attack, **Accuracy: 90.2%** After attack, **Accuracy: ~10% (1/10)**



# Memory Scaling Issues Are Real

---

- 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 Kim<sup>1</sup>   Ross Daly\*   Jeremie Kim<sup>1</sup>   Chris Fallin\*   Ji Hye Lee<sup>1</sup>  
Donghyuk Lee<sup>1</sup>   Chris Wilkerson<sup>2</sup>   Konrad Lai   Onur Mutlu<sup>1</sup>

<sup>1</sup>Carnegie Mellon University

<sup>2</sup>Intel Labs

# Memory Scaling Issues Are Real

---

- 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]

## RowHammer: A Retrospective

Onur Mutlu<sup>§‡</sup>

<sup>§</sup>ETH Zürich

Jeremie S. Kim<sup>†§</sup>

<sup>†</sup>Carnegie Mellon University

## Main Memory Needs Intelligent Controllers

# 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<sup>§†</sup>      Minesh Patel<sup>§</sup>      A. Giray Yağlıkçı<sup>§</sup>  
Hasan Hassan<sup>§</sup>      Roknoddin Azizi<sup>§</sup>      Lois Orosa<sup>§</sup>      Onur Mutlu<sup>§†</sup>

<sup>§</sup>*ETH Zürich*

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

# Key Takeaways from 1580 Chips

- Newer DRAM chips are more vulnerable to RowHammer
- There are chips today whose weakest cells fail after **only 4800 hammers**
- Chips of newer DRAM technology nodes can exhibit RowHammer bit flips 1) in **more rows** and 2) **farther away** from the victim row.
- Existing mitigation mechanisms are NOT effective

# 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 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<sup>§†</sup>, Minesh Patel<sup>§</sup>, Lillian Tsai<sup>‡</sup>,  
Stefan Saroiu, Alec Wolman, and Onur Mutlu<sup>§†</sup>  
Microsoft Research, <sup>§</sup>ETH Zürich, <sup>†</sup>CMU, <sup>‡</sup>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 Yağlıkçı<sup>1</sup> Minesh Patel<sup>1</sup> Jeremie S. Kim<sup>1</sup> Roknoddin Azizi<sup>1</sup> Ataberk Olgun<sup>1</sup> Lois Orosa<sup>1</sup>  
Hasan Hassan<sup>1</sup> Jisung Park<sup>1</sup> Konstantinos Kanellopoulos<sup>1</sup> Taha Shahroodi<sup>1</sup> Saugata Ghose<sup>2</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>*ETH Zürich*

<sup>2</sup>*University of Illinois at Urbana-Champaign*

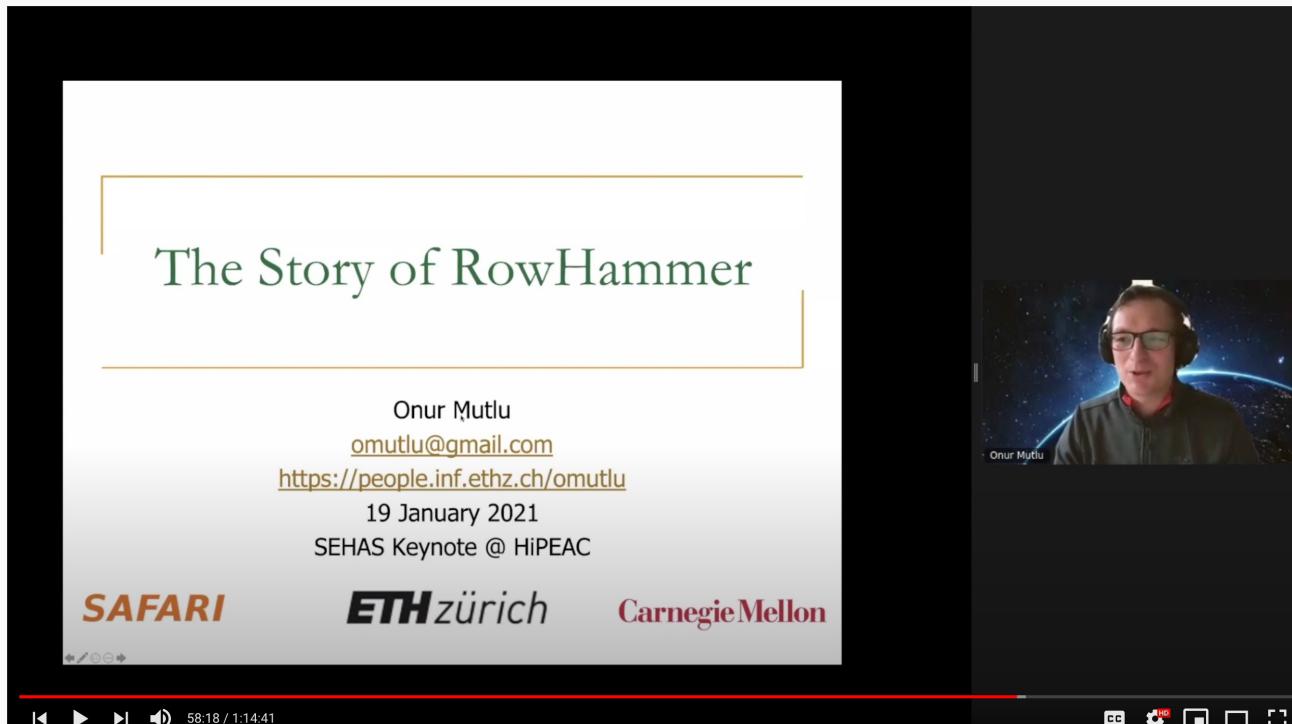
# Detailed Lectures on RowHammer

---

- 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](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 - Secure Hardware, Architectures, and Operating Systems Keynote - Onur Mutlu

1,293 views • Premiered Feb 2, 2021

144 0 SHARE SAVE ...



Onur Mutlu Lectures  
13.9K subscribers

ANALYTICS EDIT VIDEO

# Rowhammer



Computing Systems  
Need

Intelligent Memories

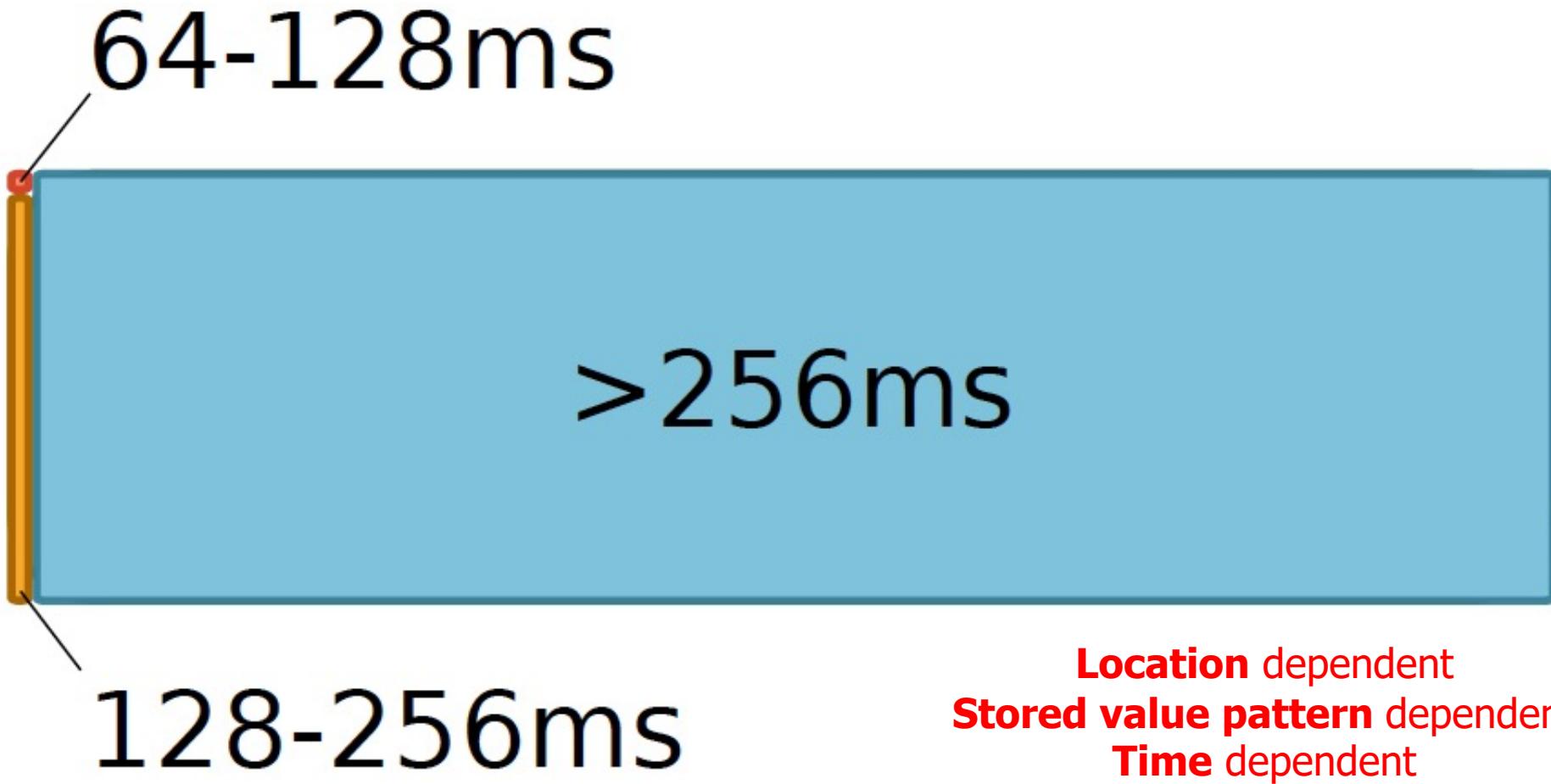
# The Takeaway, Again

---

In-Field Patch-ability  
(Intelligent Memory)  
Can Avoid Many Failures

# Data Retention in Memory [Liu et al., ISCA 2013]

- Retention Time Profile of DRAM looks like this:



# More on DRAM Refresh (I)

---

- Jamie Liu, Ben Jaiyen, Richard Veras, and Onur Mutlu,  
**"RAIDR: Retention-Aware Intelligent DRAM Refresh"**  
*Proceedings of the 39th International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2012.  
Slides (pdf)

## RAIDR: Retention-Aware Intelligent DRAM Refresh

Jamie Liu    Ben Jaiyen    Richard Veras    Onur Mutlu  
Carnegie Mellon University

# More on DRAM Refresh (II)

---

- Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson, and Onur Mutlu,  
**"An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms"**

*Proceedings of the 40th International Symposium on Computer Architecture (ISCA)*, Tel-Aviv, Israel, June 2013. [Slides \(ppt\)](#) [Slides \(pdf\)](#)

## An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms

Jamie Liu\*

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
jamiel@alumni.cmu.edu

Ben Jaiyen\*

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
bjaiyen@alumni.cmu.edu

Yoongu Kim

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
yoonguk@ece.cmu.edu

Chris Wilkerson

Intel Corporation  
2200 Mission College Blvd.  
Santa Clara, CA 95054  
chris.wilkerson@intel.com

Onur Mutlu

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
onur@cmu.edu

# More on DRAM Refresh (III)

- Samira Khan, Donghyuk Lee, Yoongu Kim, Alaa Alameldeen, Chris Wilkerson, and Onur Mutlu,

## **"The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study"**

*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Austin, TX, June 2014. [[Slides \(pptx\)](#) ([pdf](#))] [[Poster \(pptx\)](#) ([pdf](#))] [[Full data sets](#)]

## **The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study**

Samira Khan<sup>†\*</sup>  
samirakhan@cmu.edu

Donghyuk Lee<sup>†</sup>  
donghyuk1@cmu.edu

Yoongu Kim<sup>†</sup>  
yoongukim@cmu.edu

Alaa R. Alameldeen<sup>\*</sup>  
alaa.r.alameldeen@intel.com

Chris Wilkerson<sup>\*</sup>  
chris.wilkerson@intel.com

Onur Mutlu<sup>†</sup>  
onur@cmu.edu

<sup>†</sup>Carnegie Mellon University

<sup>\*</sup>Intel Labs

# More on DRAM Refresh (IV)

---

- Moinuddin Qureshi, Dae Hyun Kim, Samira Khan, Prashant Nair, and Onur Mutlu,  
**"AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems"**

*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Rio de Janeiro, Brazil, June 2015.  
[Slides (pptx) (pdf)]

## AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems

Moinuddin K. Qureshi<sup>†</sup>

<sup>†</sup>Georgia Institute of Technology

{moin, dhkim, pnair6}@ece.gatech.edu

Dae-Hyun Kim<sup>†</sup>

Samira Khan<sup>‡</sup>

Prashant J. Nair<sup>†</sup>

Onur Mutlu<sup>‡</sup>

<sup>‡</sup>Carnegie Mellon University

{samirakhan, onur}@cmu.edu

# More on DRAM Refresh (V)

---

- Samira Khan, Donghyuk Lee, and Onur Mutlu,  
**"PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM"**

*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Toulouse, France, June 2016.*  
[Slides (pptx) (pdf)]

## PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM

Samira Khan\*

\*University of Virginia

Donghyuk Lee<sup>†‡</sup>

<sup>†</sup>Carnegie Mellon University

Onur Mutlu<sup>\*†</sup>

<sup>‡</sup>Nvidia

<sup>\*</sup>ETH Zürich

# More on DRAM Refresh (VI)

---

- Samira Khan, Chris Wilkerson, Zhe Wang, Alaa R. Alameldeen, Donghyuk Lee, and Onur Mutlu,

**"Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content"**

*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](#))]

## Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content

Samira Khan<sup>\*</sup> Chris Wilkerson<sup>†</sup> Zhe Wang<sup>†</sup> Alaa R. Alameldeen<sup>†</sup> Donghyuk Lee<sup>‡</sup> Onur Mutlu<sup>\*</sup>

<sup>\*</sup>University of Virginia

<sup>†</sup>Intel Labs

<sup>‡</sup>Nvidia Research

<sup>\*</sup>ETH Zürich

# More on DRAM Refresh (VII)

---

- Minesh Patel, Jeremie S. Kim, and Onur Mutlu,

**"The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions"**

*Proceedings of the 44th International Symposium on Computer Architecture (ISCA)*, Toronto, Canada, June 2017.

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

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

- First experimental analysis of (mobile) LPDDR4 chips
- Analyzes the complex tradeoff space of retention time profiling
- Idea: enable fast and robust profiling at higher refresh intervals & temperatures

## The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions

Minesh Patel<sup>§‡</sup>    Jeremie S. Kim<sup>‡§</sup>    Onur Mutlu<sup>§‡</sup>  
<sup>§</sup>ETH Zürich    <sup>‡</sup>Carnegie Mellon University

# More on DRAM Refresh (VIII)

---

- Minesh Patel, Jeremie S. Kim, Hasan Hassan, and Onur Mutlu,  
**"Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices"**  
*Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Portland, OR, USA, June 2019.

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

[[Talk Video](#) (26 minutes)]

[[Full Talk Lecture](#) (29 minutes)]

[[Source Code for EINSim, the Error Inference Simulator](#)]

***Best paper award.***

## Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices

Minesh Patel<sup>†</sup>    Jeremie S. Kim<sup>‡†</sup>    Hasan Hassan<sup>†</sup>    Onur Mutlu<sup>†‡</sup>

<sup>†</sup>*ETH Zürich*    <sup>‡</sup>*Carnegie Mellon University*

# More on DRAM Refresh (IX)

---

- Minesh Patel, Jeremie S. Kim, Taha Shahroodi, Hasan Hassan, and Onur Mutlu,  
**"Bit-Exact ECC Recovery (BEER): Determining DRAM On-Die ECC Functions by Exploiting DRAM Data Retention Characteristics"**

*Proceedings of the 53rd International Symposium on Microarchitecture (**MICRO**), Virtual, October 2020.*

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

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

[[Talk Video](#) (15 minutes)]

[[Lightning Talk Video](#) (1.5 minutes)]

***Best paper award.***

## Bit-Exact ECC Recovery (BEER): Determining DRAM On-Die ECC Functions by Exploiting DRAM Data Retention Characteristics

Minesh Patel<sup>†</sup>   Jeremie S. Kim<sup>‡†</sup>   Taha Shahroodi<sup>†</sup>   Hasan Hassan<sup>†</sup>   Onur Mutlu<sup>†‡</sup>

<sup>†</sup>*ETH Zürich*   <sup>‡</sup>*Carnegie Mellon University*

## Main Memory Needs Intelligent Controllers

# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

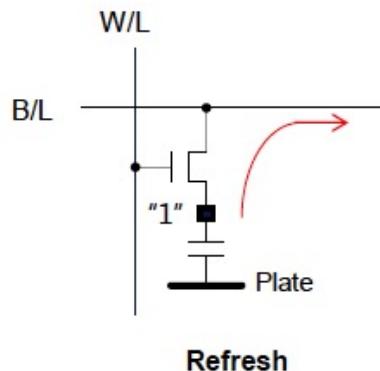
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

### ❖ tWR

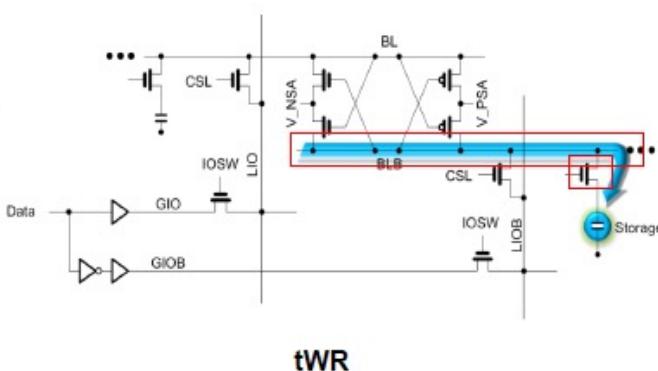
- Contact resistance between the cell capacitor and access transistor increasing
- On-current of the cell access transistor decreasing
- Bit-line resistance increasing

### ❖ VRT

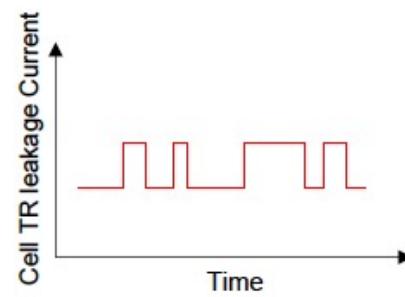
- Occurring more frequently with cell capacitance decreasing



Refresh



tWR



VRT

# Call for Intelligent Memory Controllers

## DRAM Process Scaling Challenges

### ❖ Refresh

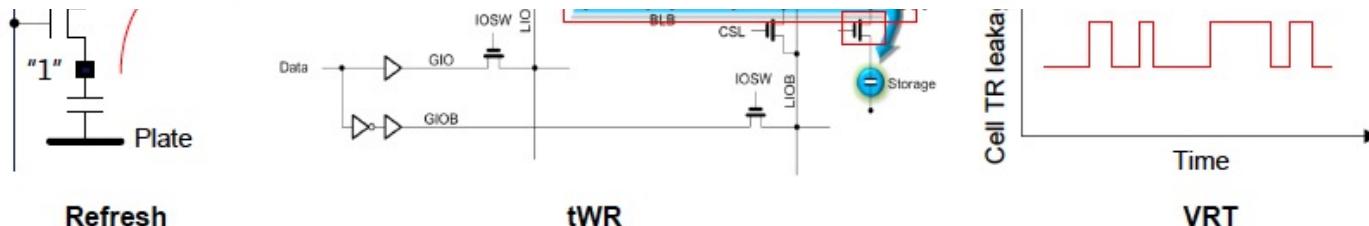
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

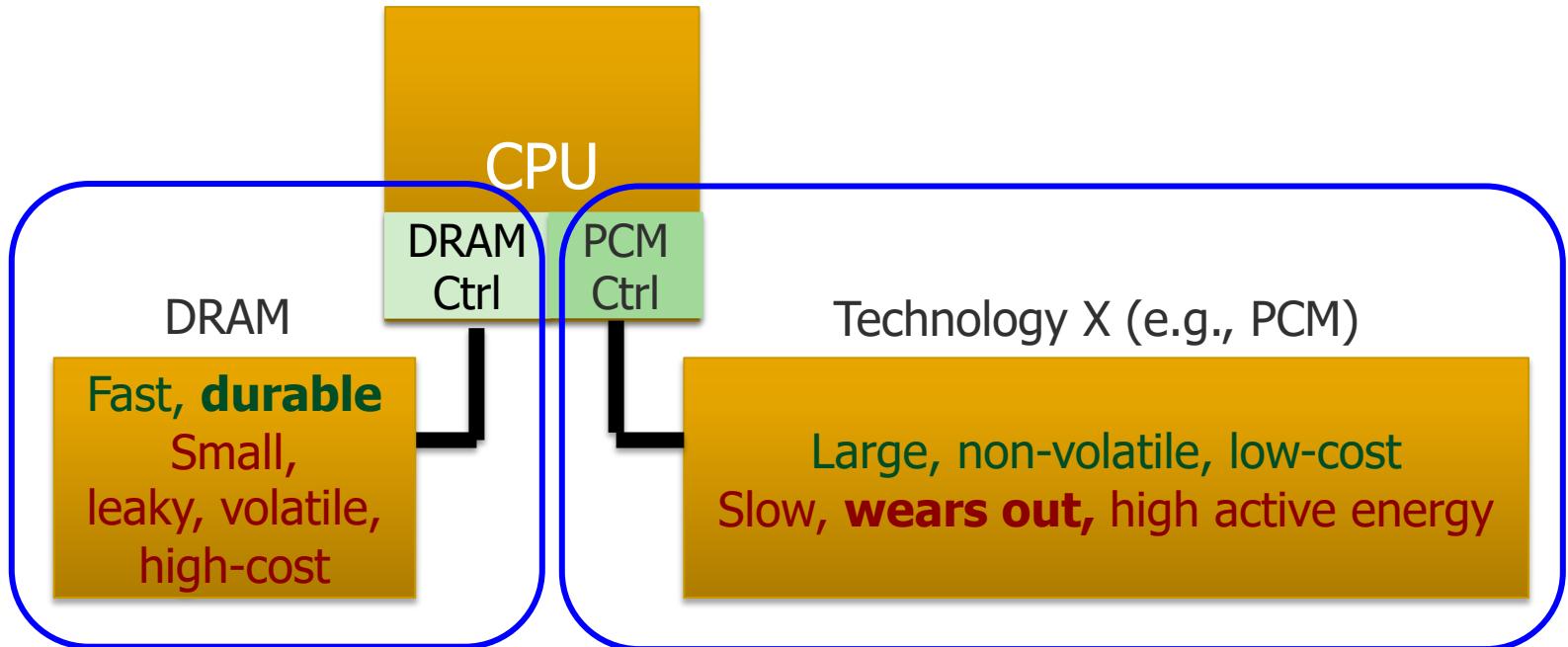
## Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

Uksong Kang, Hak-soo Yu, Churoo Park, \*Hongzhong Zheng,  
\*\*John Halbert, \*\*Kuljit Bains, SeongJin Jang, and Joo Sun Choi

*Samsung Electronics, Hwasung, Korea / \*Samsung Electronics, San Jose / \*\*Intel*



# Hybrid Memory Systems



Hardware/software manage data allocation and movement  
to achieve the best of multiple technologies

Meza+, “[Enabling Efficient and Scalable Hybrid Memories](#),” IEEE Comp. Arch. Letters, 2012.

Yoon, Meza et al., “[Row Buffer Locality Aware Caching Policies for Hybrid Memories](#),” ICCD 2012 Best Paper Award.

## Main Memory Needs Intelligent Controllers

# Why In-Memory Computation Today?

---

- Push from Technology
  - DRAM Scaling at jeopardy
    - Controllers close to DRAM
    - Industry open to new memory architectures
  
- Pull from Systems and Applications
  - Data access is a major system and application bottleneck
  - Systems are energy limited
  - Data movement much more energy-hungry than computation

# Three Key Systems Trends

---

## 1. Data access is a major bottleneck

- ❑ Applications are increasingly data hungry

## 2. Energy consumption is a key limiter

## 3. Data movement energy dominates compute

- ❑ Especially true for off-chip to on-chip movement

# Do We Want This?

---



# Or This?

---



# Challenge and Opportunity for Future

---

High Performance,  
Energy Efficient,  
Sustainable

# The Problem

---

Data access is the major performance and energy bottleneck

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

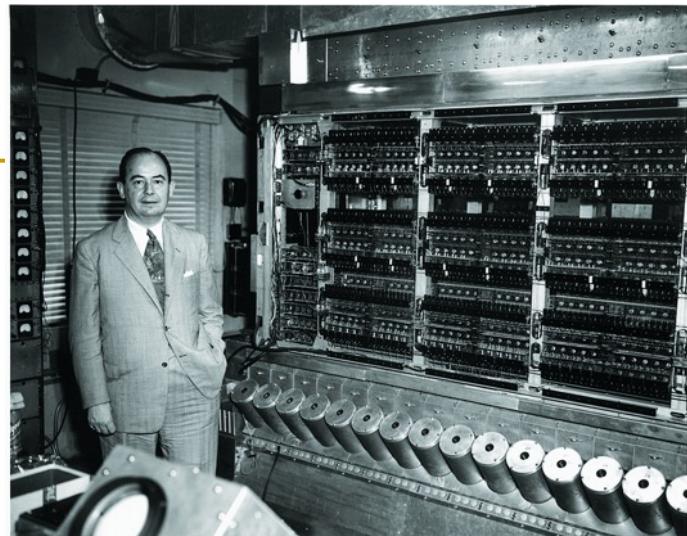
# The Problem

---

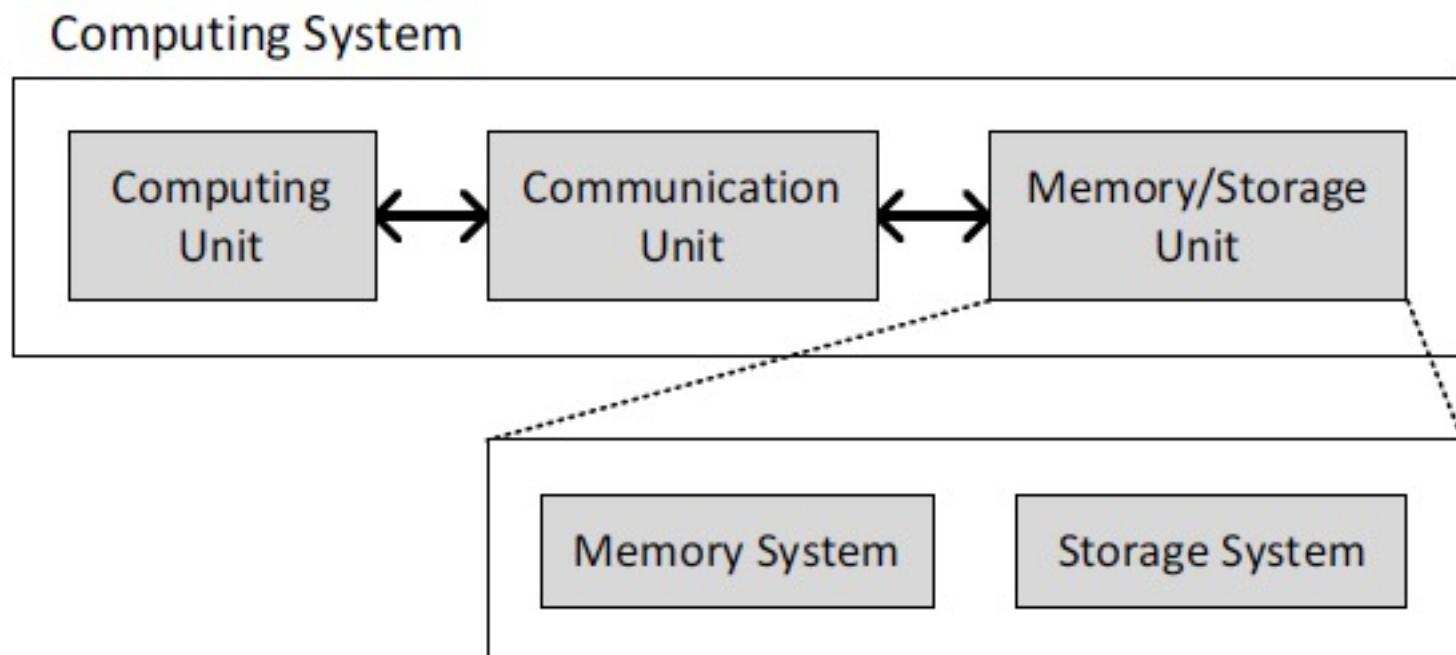
Processing of data  
is performed  
far away from the data

# A Computing System

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

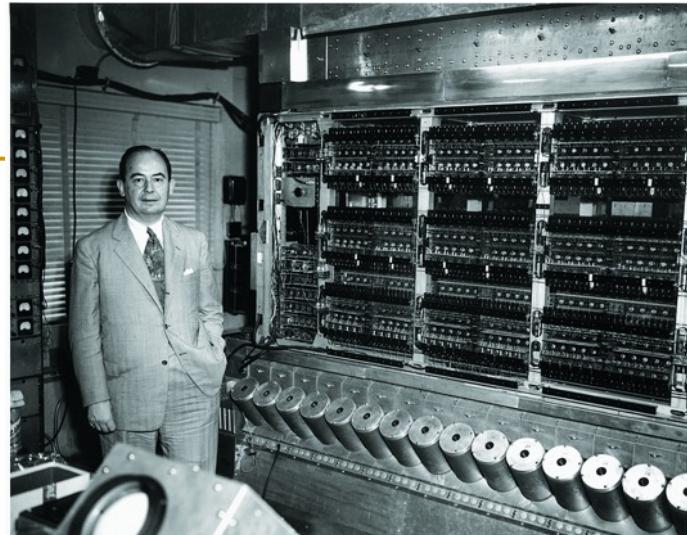


Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.

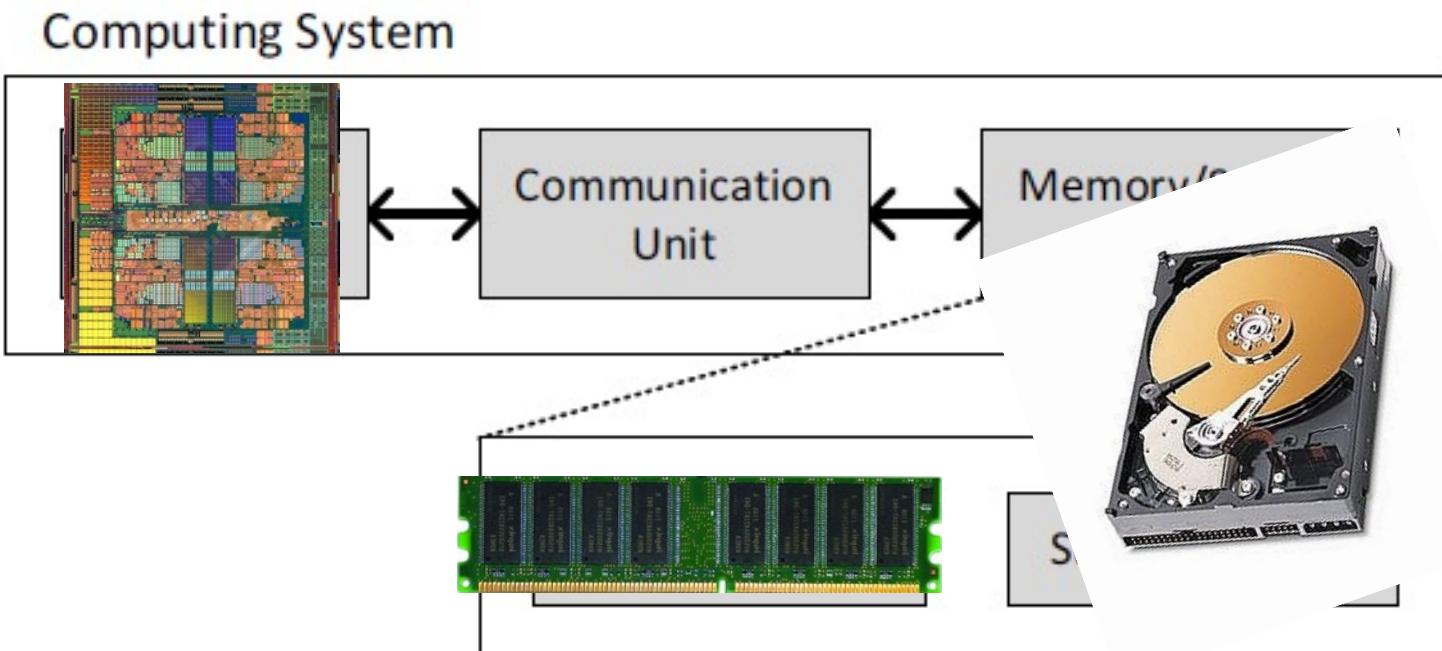


# A Computing System

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

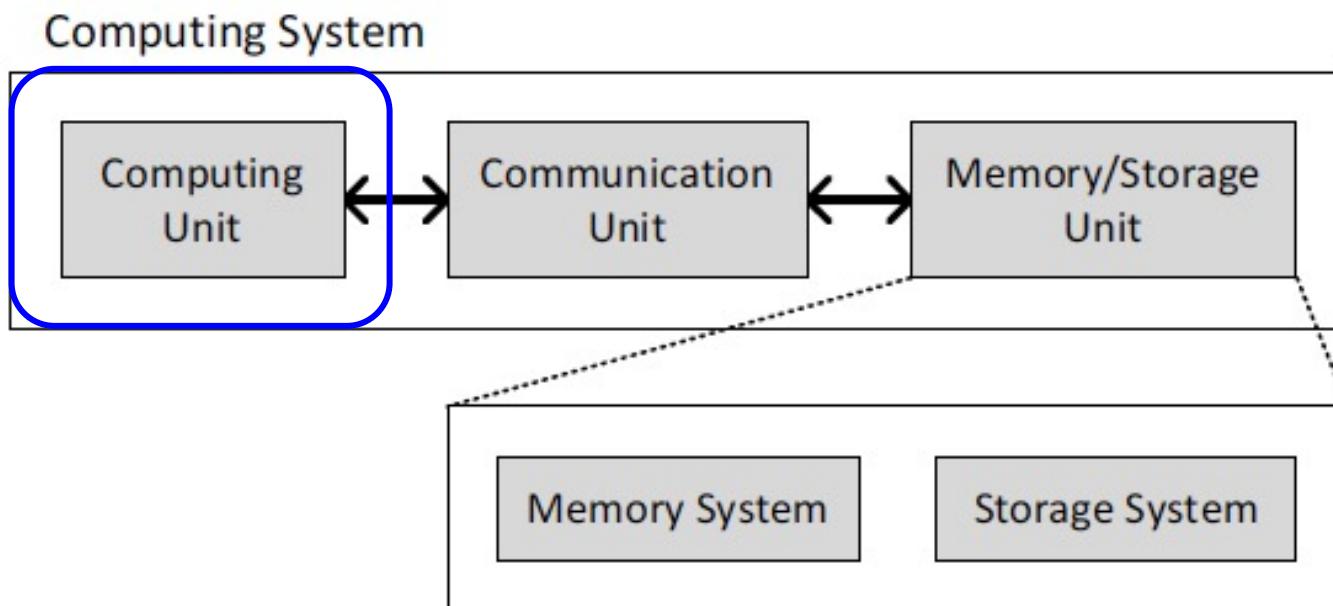


Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.



# Today's Computing Systems

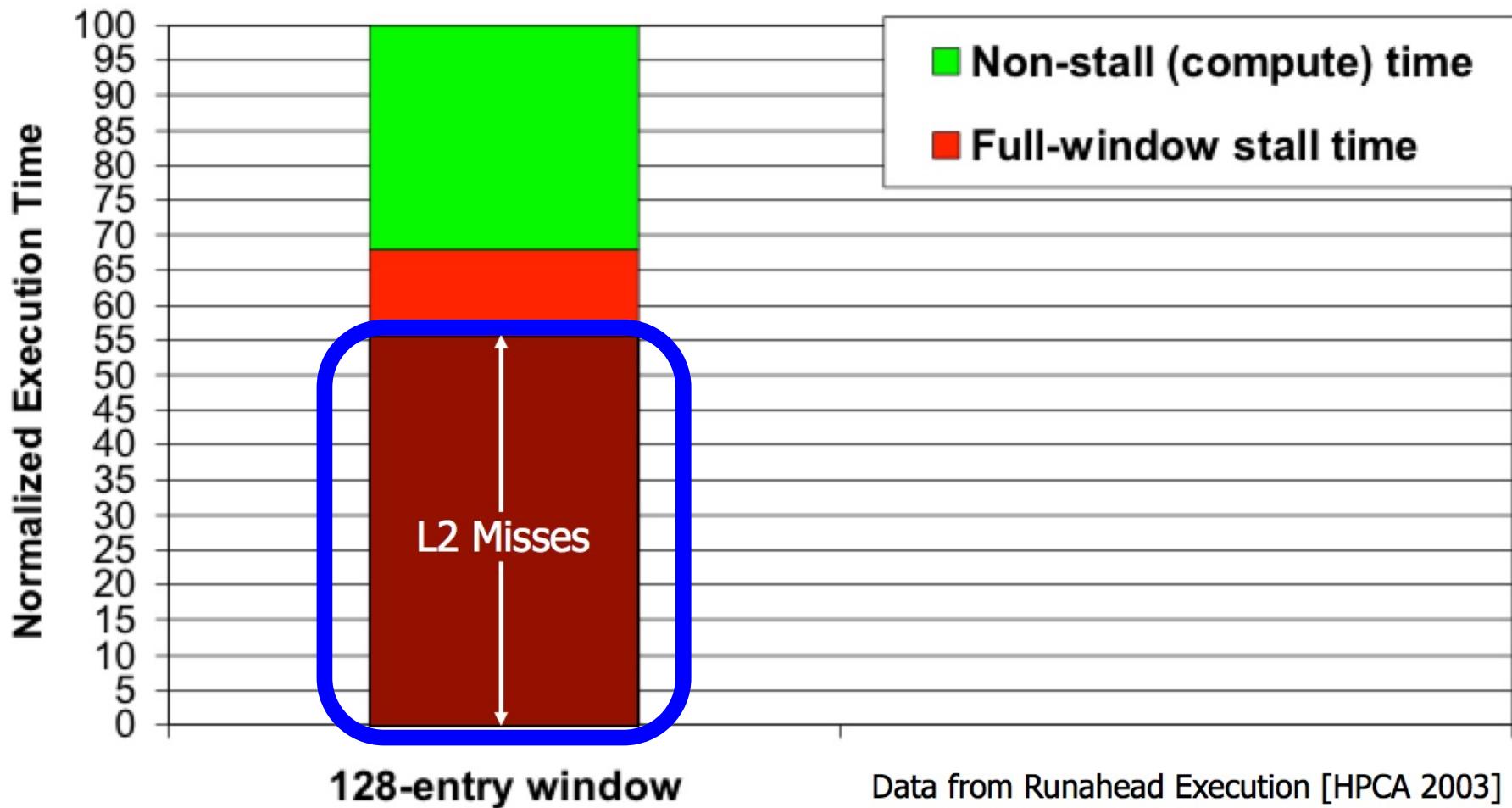
- Are overwhelmingly processor centric
- All data processed in the processor → at great system cost
- Processor is heavily optimized and is considered the master
- Data storage units are dumb and are largely unoptimized (except for some that are on the processor die)



# Yet ...

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

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



# The Performance Perspective

---

- Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt,  
**"Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors"**  
*Proceedings of the 9th International Symposium on High-Performance Computer Architecture (HPCA)*, pages 129-140, Anaheim, CA, February 2003. [Slides \(pdf\)](#)

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

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

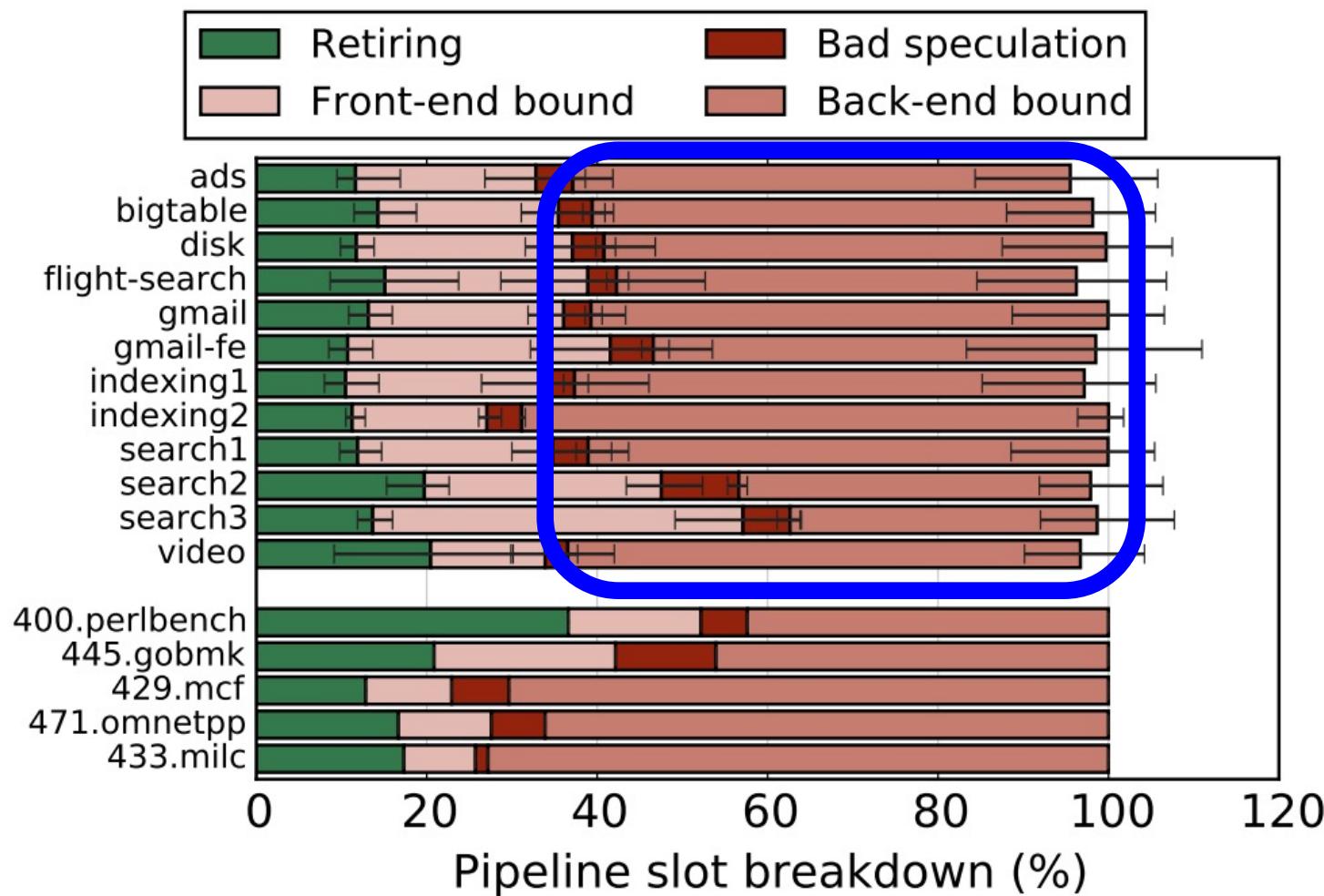
§ECE Department  
The University of Texas at Austin  
{onur,patt}@ece.utexas.edu

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

‡Desktop Platforms Group  
Intel Corporation  
chris.wilkerson@intel.com

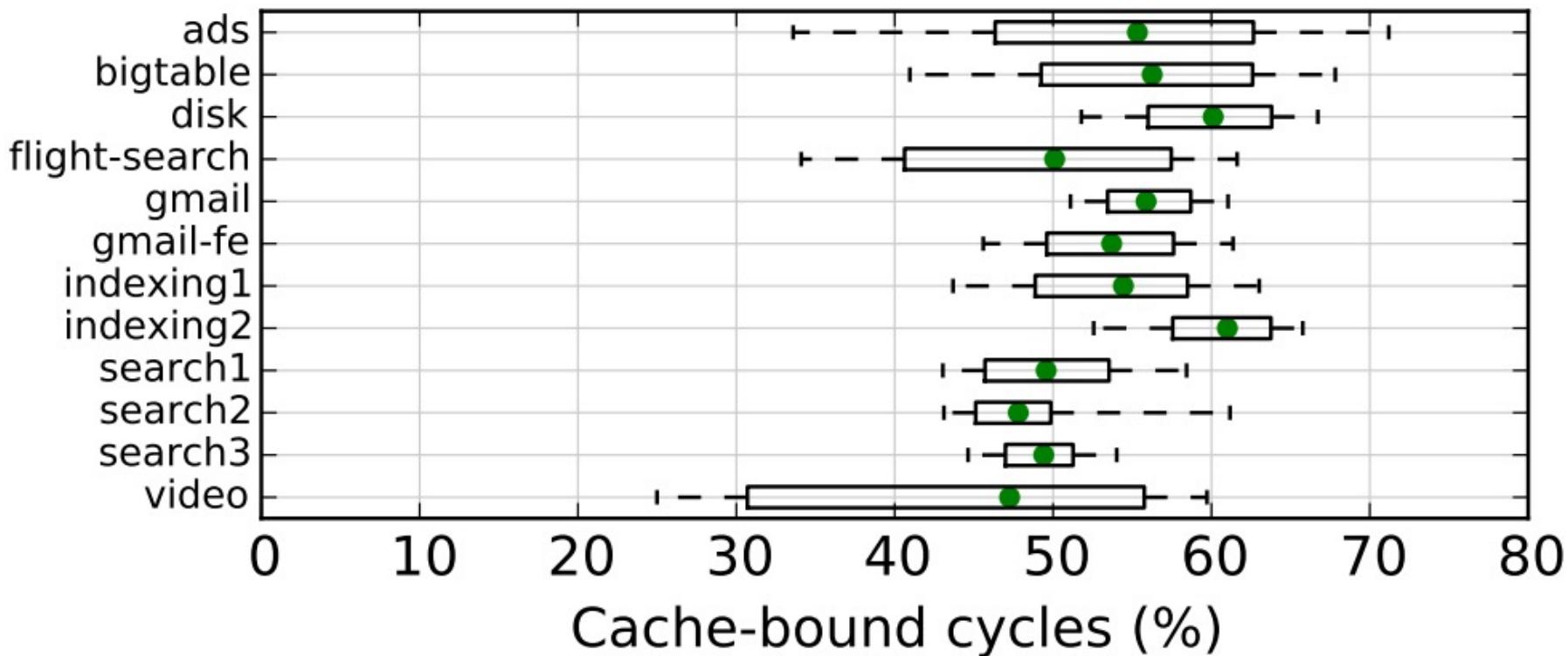
# The Performance Perspective (Today)

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



# The Performance Perspective (Today)

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



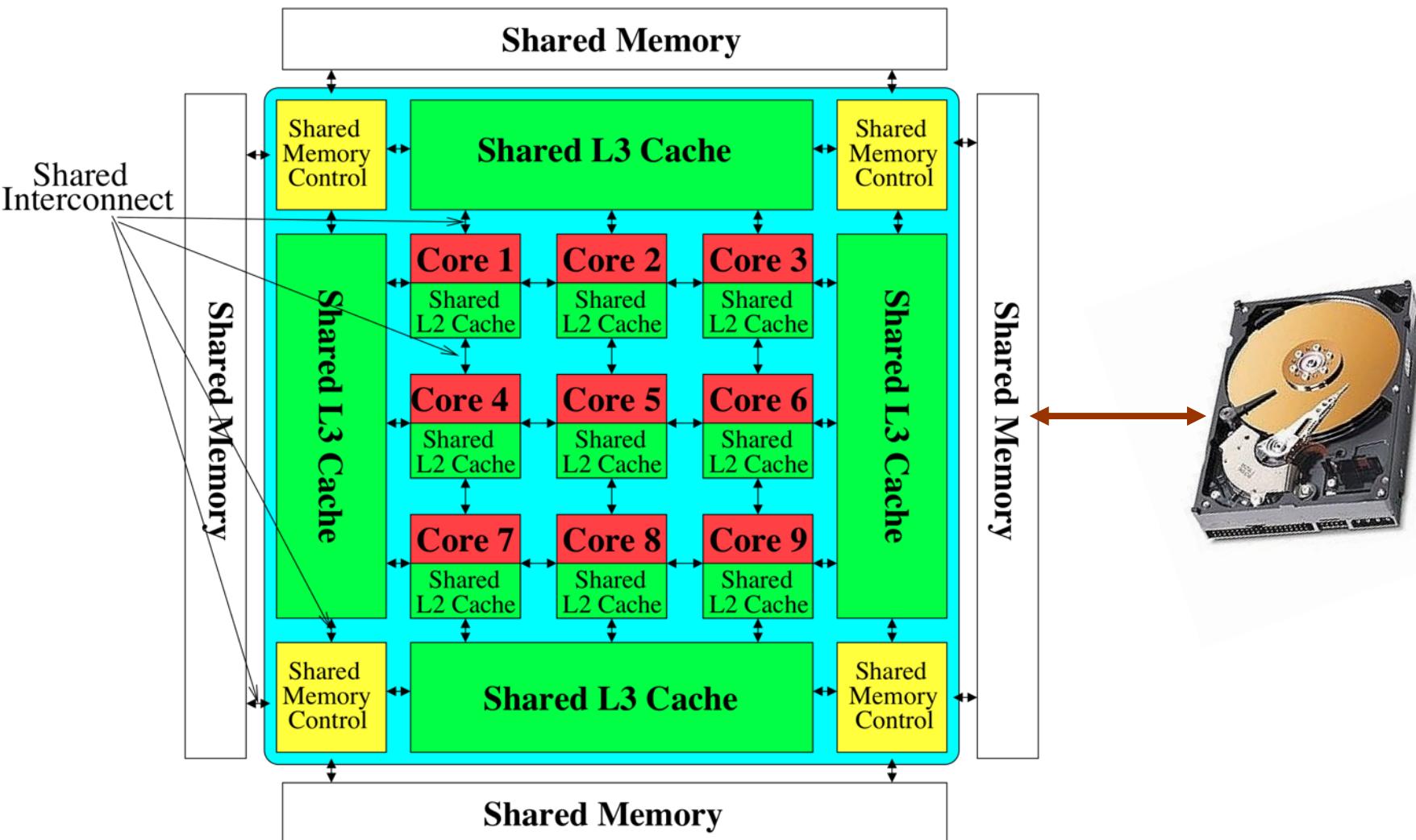
**Figure 11: Half of cycles are spent stalled on caches.**

# Perils of Processor-Centric Design

---

- Grossly-imbalanced systems
  - Processing done only in **one place**
  - Everything else just stores and moves data: **data moves a lot**
    - Energy inefficient
    - Low performance
    - Complex
- Overly complex and bloated processor (and accelerators)
  - To tolerate data access from memory
  - Complex hierarchies and mechanisms
    - Energy inefficient
    - Low performance
    - Complex

# Perils of Processor-Centric Design

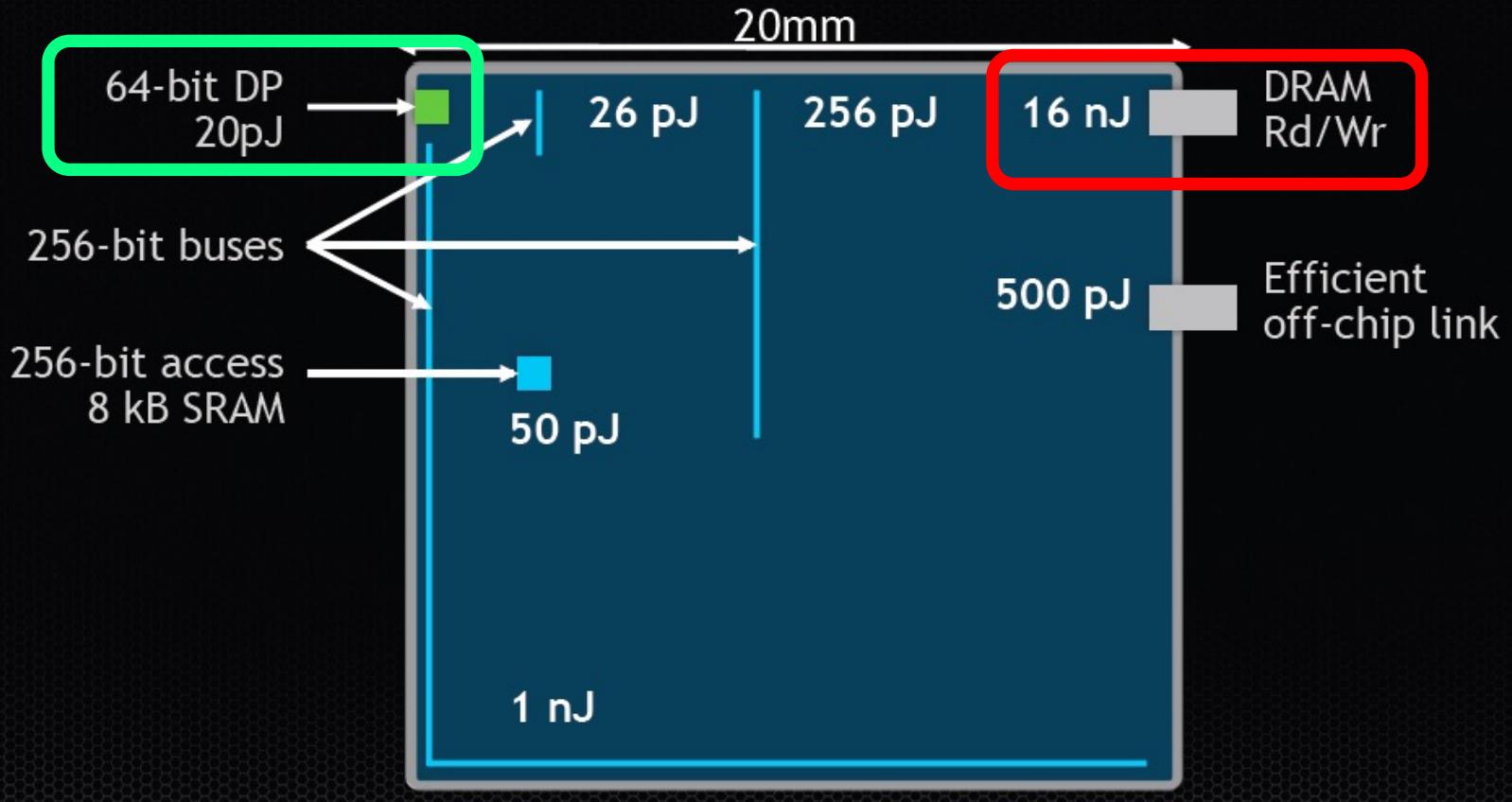


**Most of the system is dedicated to storing and moving data**

# The Energy Perspective

## Communication Dominates Arithmetic

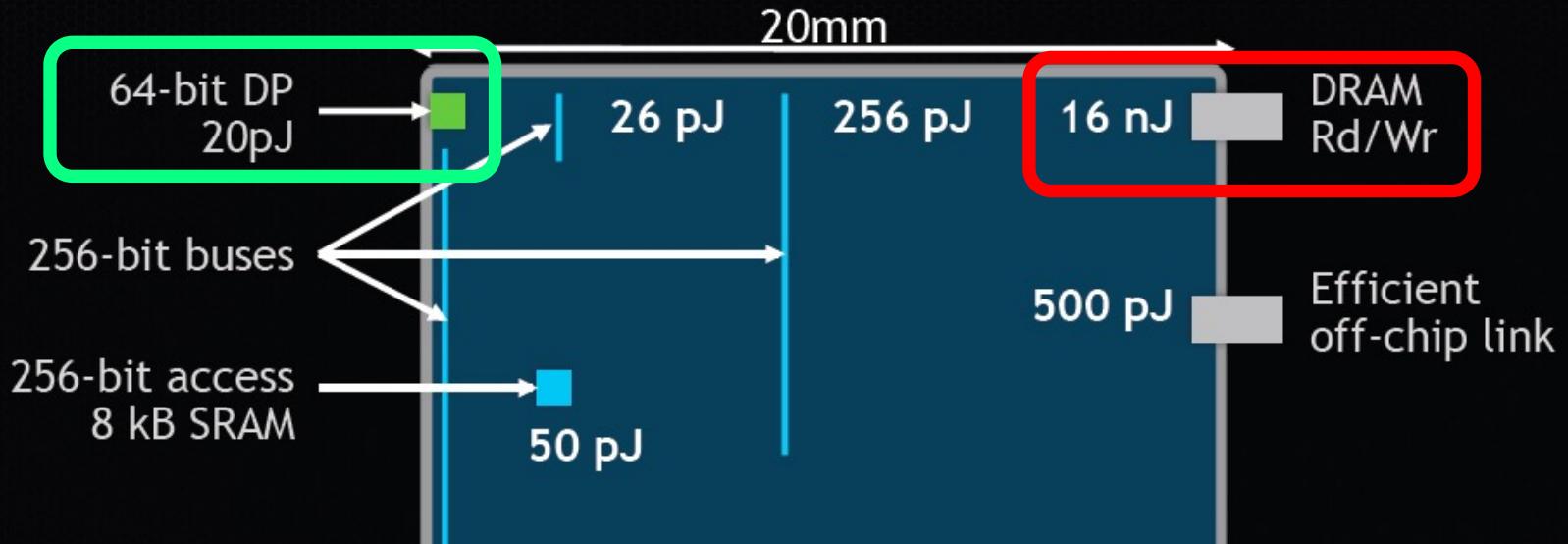
Dally, HiPEAC 2015



# Data Movement vs. Computation Energy

## Communication Dominates Arithmetic

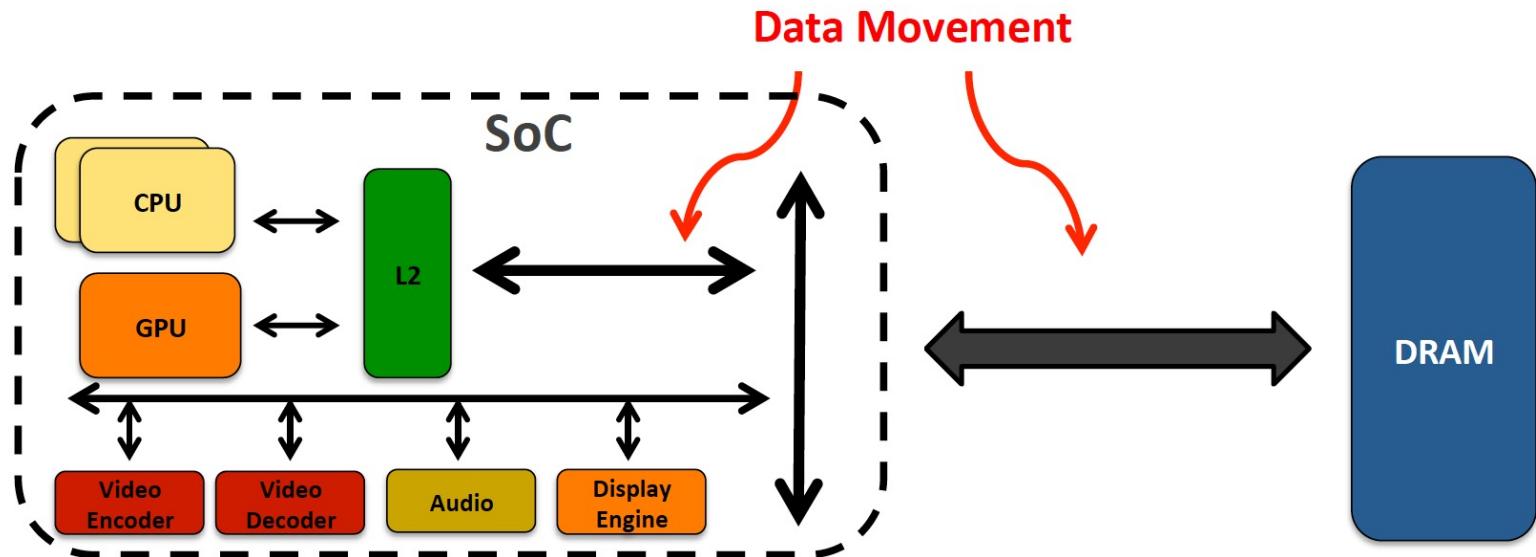
Dally, HiPEAC 2015



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

# Data Movement vs. Computation Energy

- Data movement is a major system energy bottleneck
  - Comprises 41% of mobile system energy during web browsing [2]
  - Costs ~115 times as much energy as an ADD operation [1, 2]



[1]: Reducing data Movement Energy via Online Data Clustering and Encoding (MICRO'16)

[2]: Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms (IISWC'14)

# 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 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>

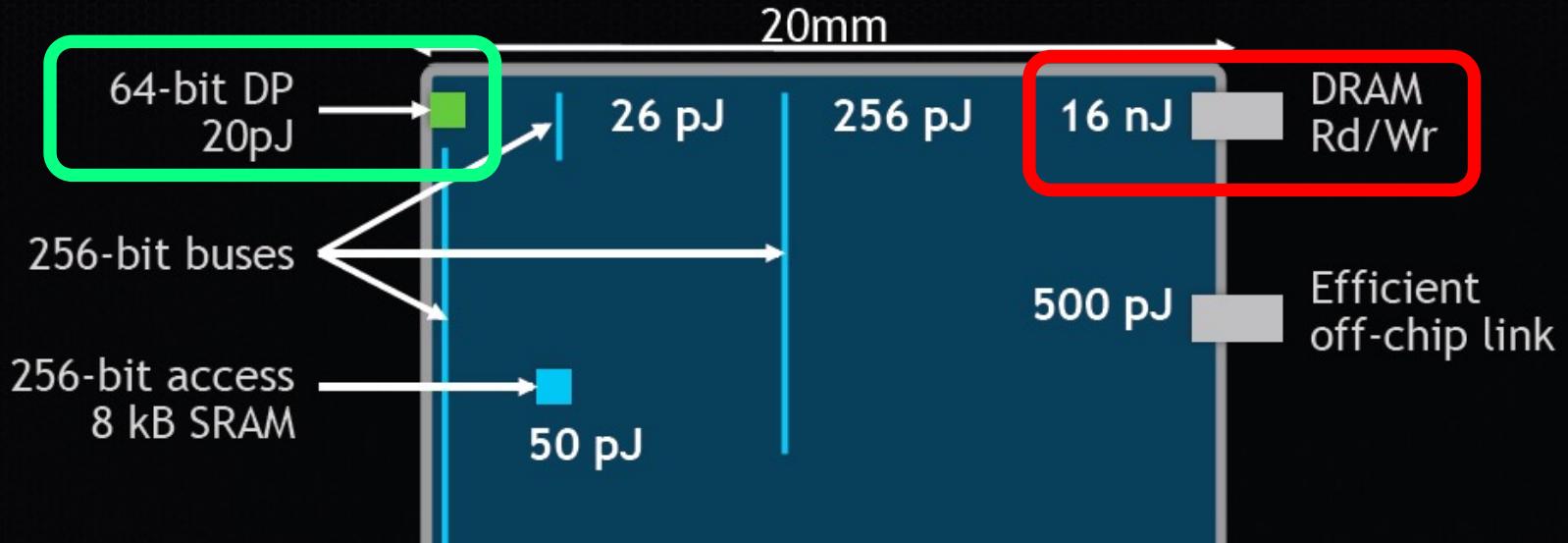
Parthasarathy Ranganathan<sup>3</sup>

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

# We Do Not Want to Move Data!

## Communication Dominates Arithmetic

Dally, HiPEAC 2015



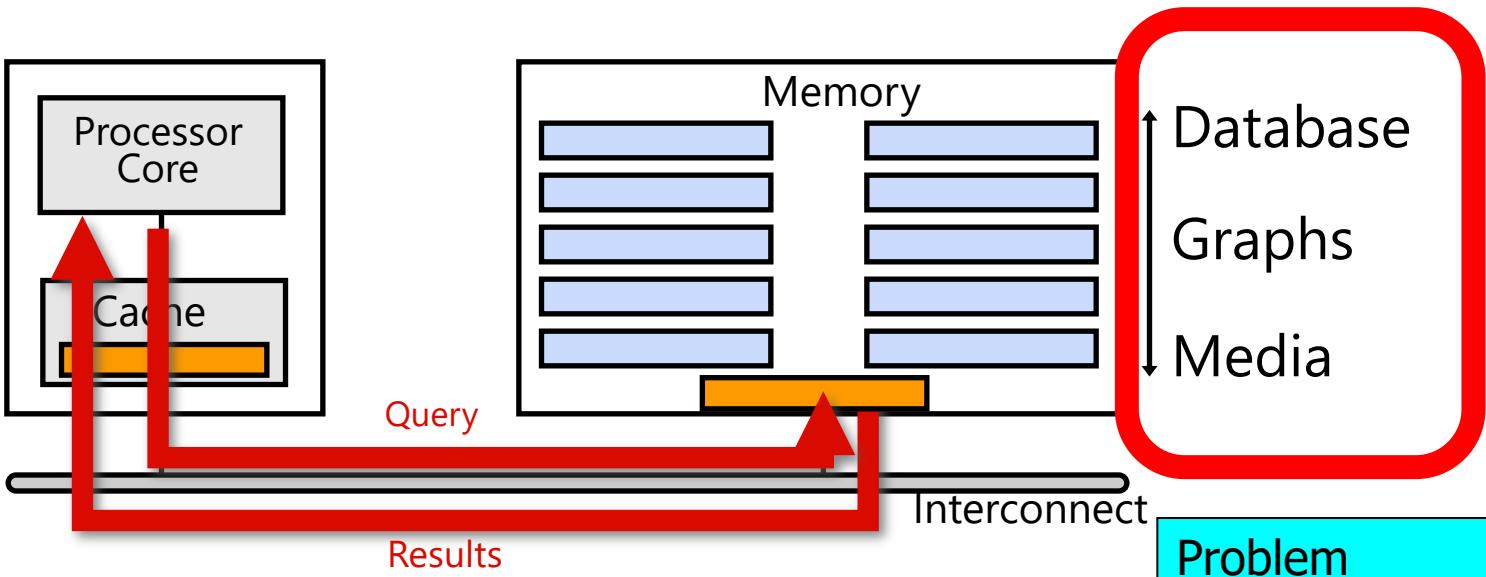
A memory access consumes  $\sim$ 100-1000X  
the energy of a complex addition

# We Need A Paradigm Shift To ...

---

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

# Goal: Processing Inside Memory



- 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?

# Processing in Memory: Two Approaches

1. Minimally changing memory chips
2. Exploiting 3D-stacked memory

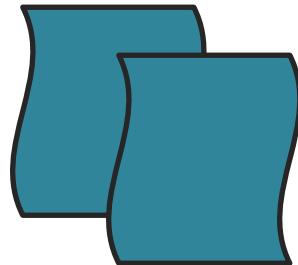
# Approach 1: Minimally Changing Memory

---

- DRAM has great capability to perform **bulk data movement** and **computation** internally with small changes
  - Can **exploit internal connectivity** to move data
  - Can **exploit analog computation capability**
  - ...
- Examples: RowClone, In-DRAM AND/OR, Gather/Scatter DRAM
  - RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data (Seshadri et al., MICRO 2013)
  - Fast Bulk Bitwise AND and OR in DRAM (Seshadri et al., IEEE CAL 2015)
  - Gather-Scatter DRAM: In-DRAM Address Translation to Improve the Spatial Locality of Non-unit Strided Accesses (Seshadri et al., MICRO 2015)
  - "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology" (Seshadri et al., MICRO 2017)

# Starting Simple: Data Copy and Initialization

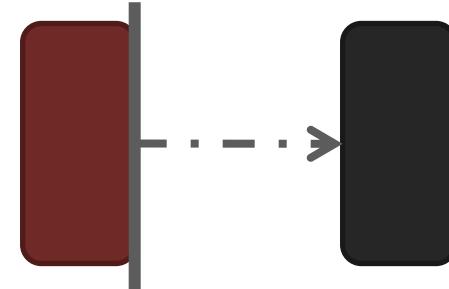
*memmove & memcp $\gamma$ : 5% cycles in Google's datacenter [Kanев+ ISCA'15]*



**Forking**



**Zero initialization**  
(e.g., security)



**Checkpointing**



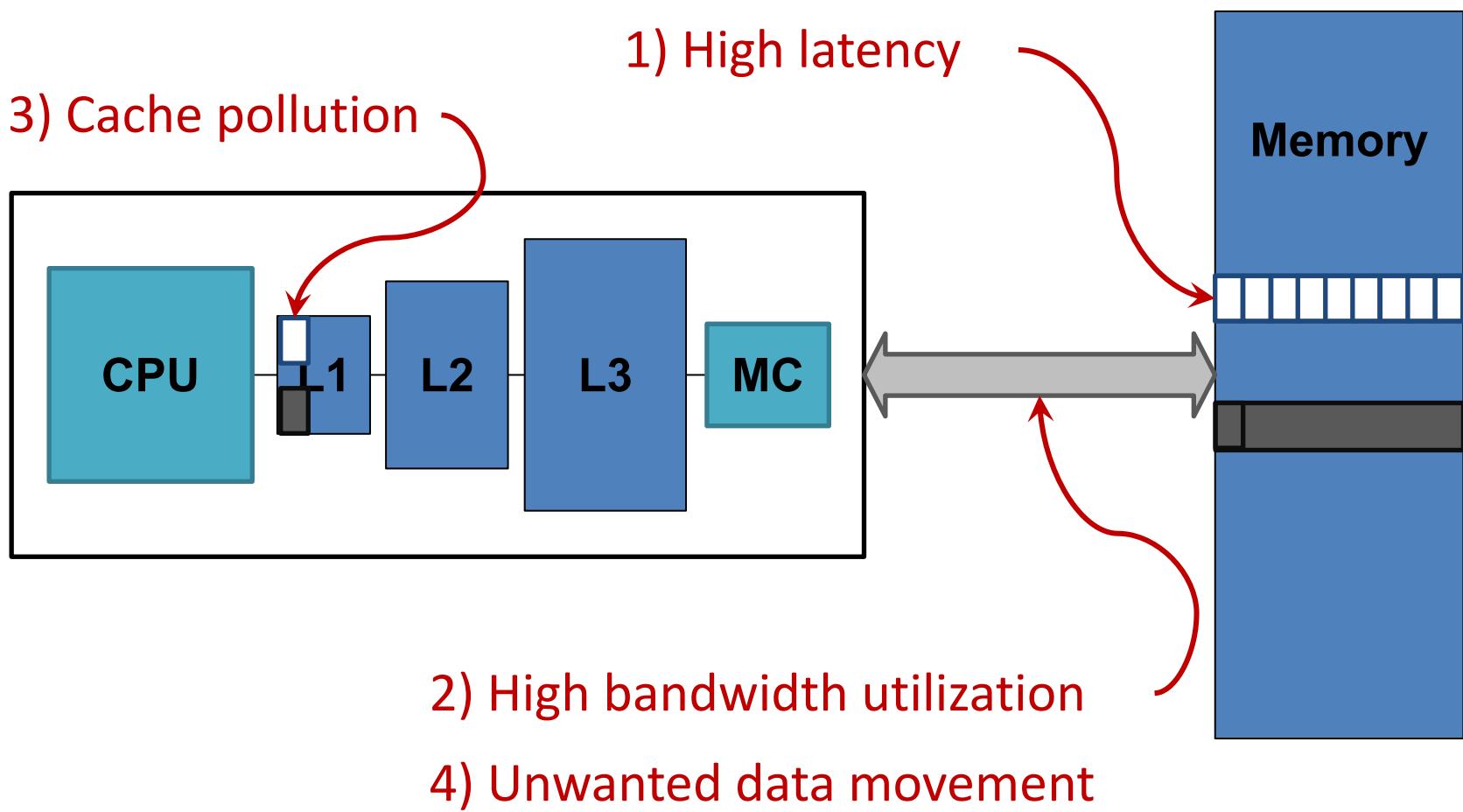
**VM Cloning  
Deduplication**



**Page Migration**

...  
Many more

# Today's Systems: Bulk Data Copy

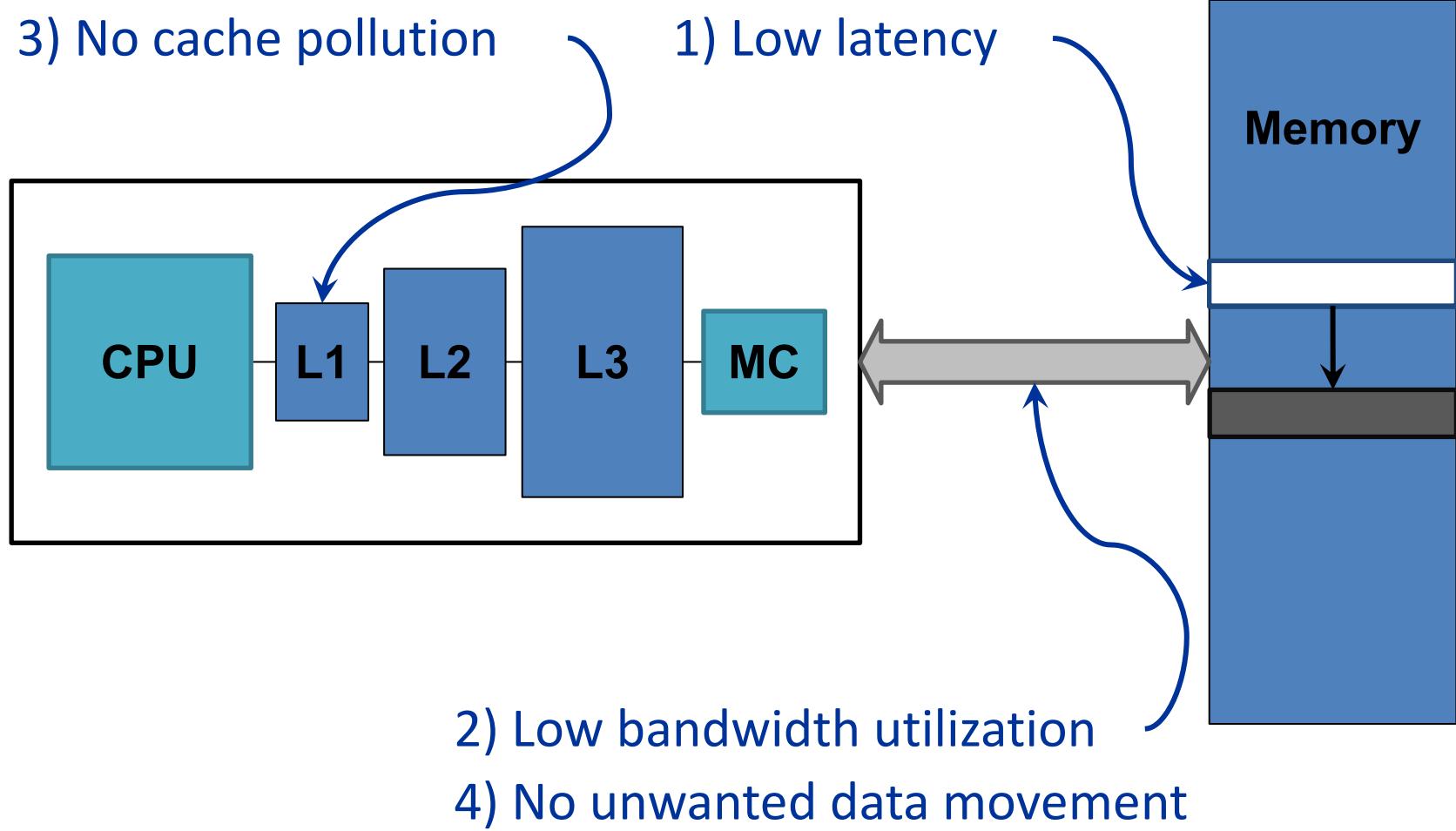


1046ns, 3.6uJ (for 4KB page copy via DMA)

# Future Systems: In-Memory Copy

3) No cache pollution

1) Low latency

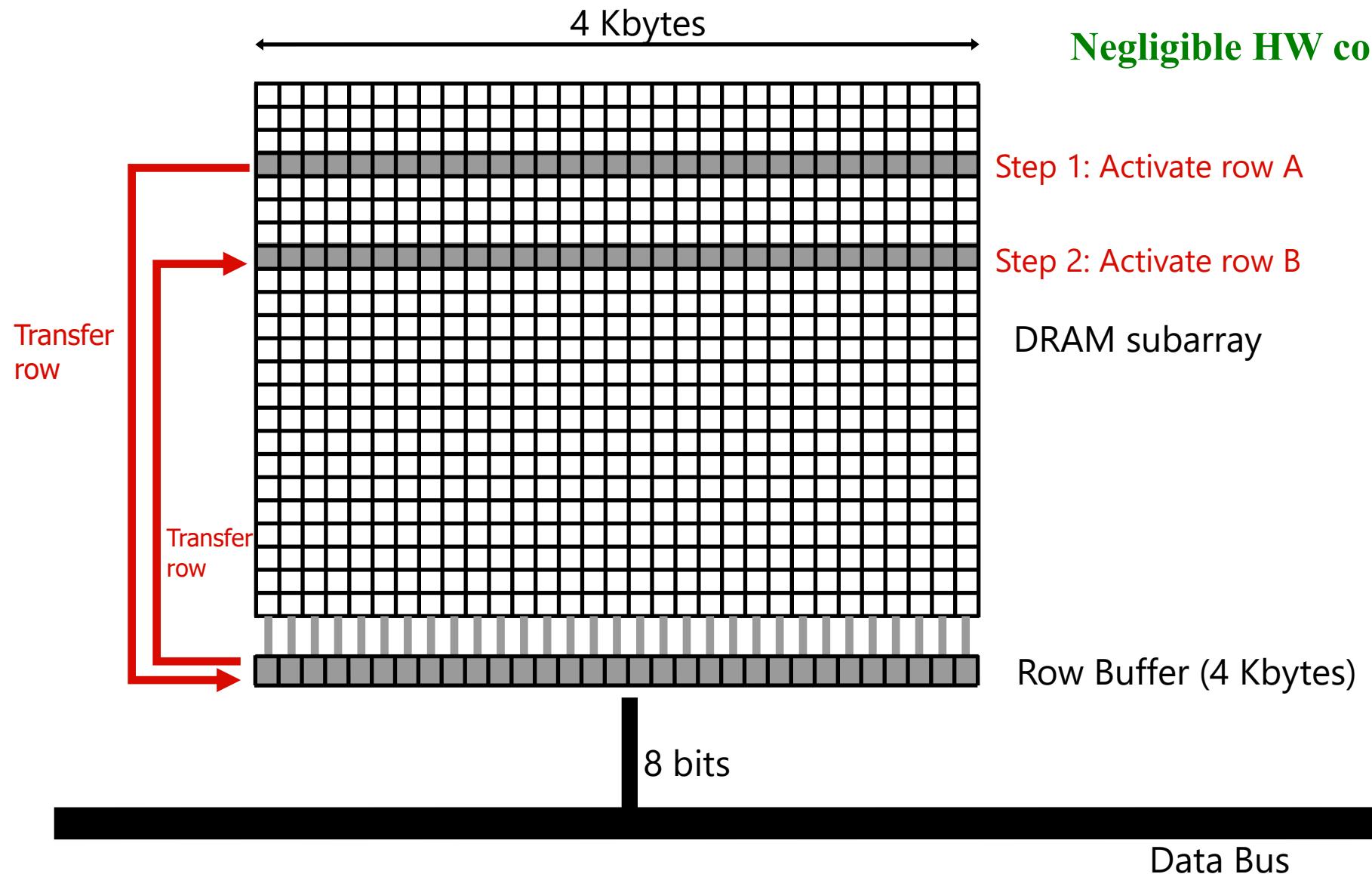


1046ns, 3.6uJ → 90ns, 0.04uJ

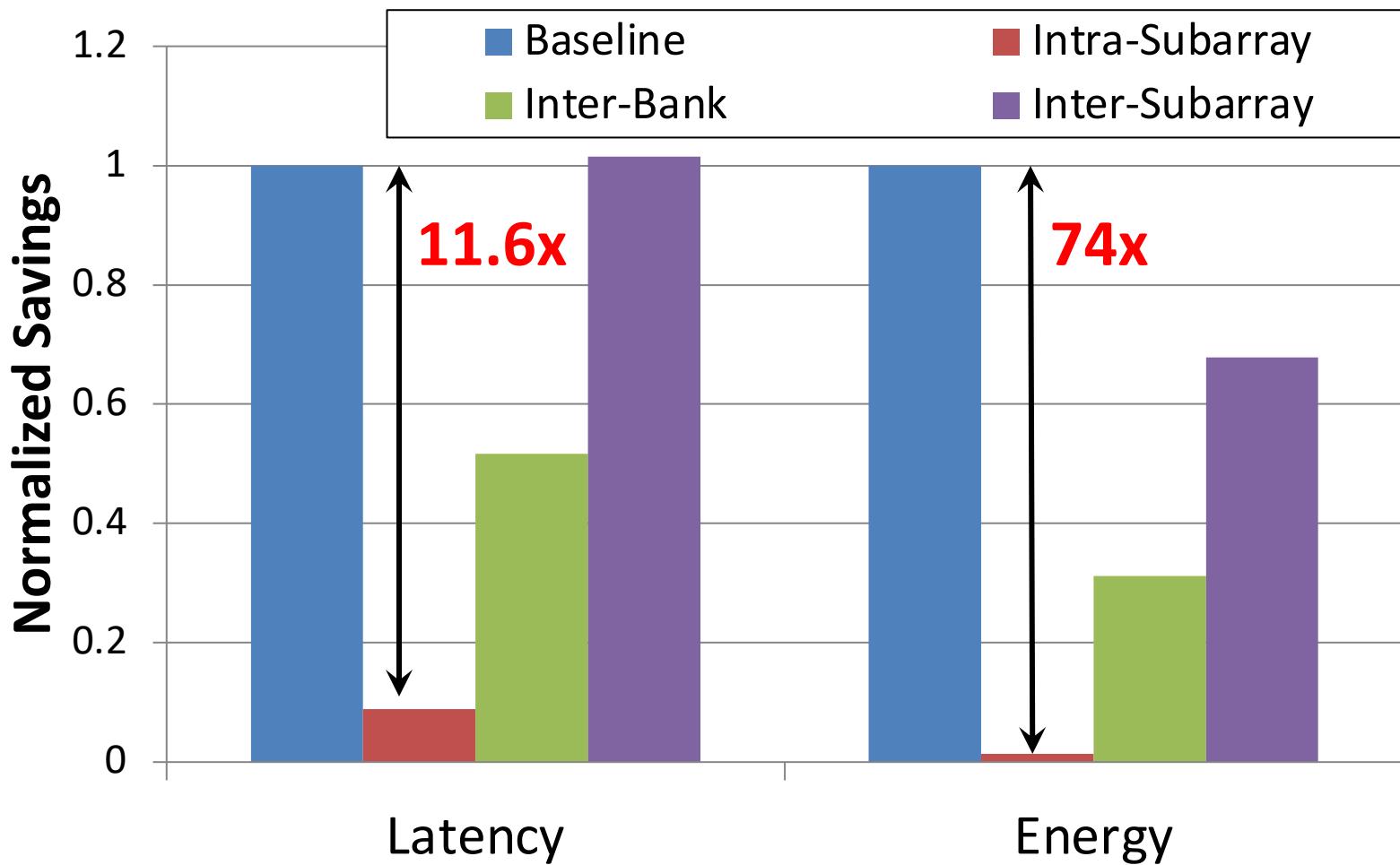
# RowClone: In-DRAM Row Copy

Idea: Two consecutive ACTivates

Negligible HW cost



# RowClone: Latency and Energy Savings



Seshadri et al., "RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.

# More on RowClone

---

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

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

*Proceedings of the 46th International Symposium on Microarchitecture (MICRO)*, Davis, CA, December 2013. [[Slides \(pptx\)](#) [\(pdf\)](#)] [[Lightning Session Slides \(pptx\)](#) [\(pdf\)](#)] [[Poster \(pptx\)](#) [\(pdf\)](#)]

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

Vivek Seshadri      Yoongu Kim      Chris Fallin\*      Donghyuk Lee  
[vseshadr@cs.cmu.edu](mailto:vseshadr@cs.cmu.edu)    [yoongukim@cmu.edu](mailto:yoongukim@cmu.edu)    [cfallin@c1f.net](mailto:cfallin@c1f.net)    [donghyuk1@cmu.edu](mailto:donghyuk1@cmu.edu)

Rachata Ausavarungnirun      Gennady Pekhimenko      Yixin Luo  
[rachata@cmu.edu](mailto:rachata@cmu.edu)    [gpekhime@cs.cmu.edu](mailto:gpekhime@cs.cmu.edu)    [yixinluo@andrew.cmu.edu](mailto:yixinluo@andrew.cmu.edu)

Onur Mutlu      Phillip B. Gibbons<sup>†</sup>      Michael A. Kozuch<sup>†</sup>      Todd C. Mowry  
[onur@cmu.edu](mailto:onur@cmu.edu)    [phillip.b.gibbons@intel.com](mailto:phillip.b.gibbons@intel.com)    [michael.a.kozuch@intel.com](mailto:michael.a.kozuch@intel.com)    [tcm@cs.cmu.edu](mailto:tcm@cs.cmu.edu)

# RowClone Extensions and Follow-Up Work

---

- Can this be improved to do faster inter-subarray copy?
  - Yes, see LISA [Chang et al., HPCA 2016]
- Can we enable data movement at smaller granularities within a bank?
  - Yes, see FIGARO [Wang et al., MICRO 2020]
- Can this be improved to do better inter-bank copy?
  - Yes, see Network-on-Memory [CAL 2020]
- Can similar ideas and DRAM properties be used to perform computation on data?
  - Yes, see Ambit [Seshadri et al., CAL 2015, MICRO 2017]

# LISA: Increasing Connectivity in DRAM

---

- Kevin K. Chang, Prashant J. Nair, Saugata Ghose, Donghyuk Lee, Moinuddin K. Qureshi, and Onur Mutlu,

## **"Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM"**

*Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA), Barcelona, Spain, March 2016.*

[Slides (pptx) (pdf)]

[Source Code]

## **Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM**

Kevin K. Chang<sup>†</sup>, Prashant J. Nair<sup>\*</sup>, Donghyuk Lee<sup>†</sup>, Saugata Ghose<sup>†</sup>, Moinuddin K. Qureshi<sup>\*</sup>, and Onur Mutlu<sup>†</sup>

<sup>†</sup>*Carnegie Mellon University*    <sup>\*</sup>*Georgia Institute of Technology*

# FIGARO: Fine-Grained In-DRAM Copy

---

- Yaohua Wang, Lois Orosa, Xiangjun Peng, Yang Guo, Saugata Ghose, Minesh Patel, Jeremie S. Kim, Juan Gómez Luna, Mohammad Sadrosadati, Nika Mansouri Ghiasi, and Onur Mutlu,  
**"FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching"**

*Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.*

## **FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching**

Yaohua Wang<sup>\*</sup> Lois Orosa<sup>†</sup> Xiangjun Peng<sup>○\*</sup> Yang Guo<sup>\*</sup> Saugata Ghose<sup>◊‡</sup> Minesh Patel<sup>†</sup>  
Jeremie S. Kim<sup>†</sup> Juan Gómez Luna<sup>†</sup> Mohammad Sadrosadati<sup>§</sup> Nika Mansouri Ghiasi<sup>†</sup> Onur Mutlu<sup>†‡</sup>

<sup>\*</sup>National University of Defense Technology   <sup>†</sup>ETH Zürich   <sup>○</sup>Chinese University of Hong Kong

<sup>◊</sup>University of Illinois at Urbana-Champaign   <sup>‡</sup>Carnegie Mellon University   <sup>§</sup>Institute of Research in Fundamental Sciences

# Network-On-Memory: Fast Inter-Bank Copy

---

- Seyyed Hossein SeyyedAghaei Rezaei, Mehdi Modarressi, Rachata Ausavarungnirun, Mohammad Sadrosadati, Onur Mutlu, and Masoud Daneshtalab,

**"NoM: Network-on-Memory for Inter-Bank Data Transfer in Highly-Banked Memories"**

*IEEE Computer Architecture Letters (CAL)*, to appear in 2020.

## NoM: NETWORK-ON-MEMORY FOR INTER-BANK DATA TRANSFER IN HIGHLY-BANKED MEMORIES

Seyyed Hossein SeyyedAghaei Rezaei<sup>1</sup>  
Mohammad Sadrosadati<sup>3</sup>

Mehdi Modarressi<sup>1,3</sup>  
Onur Mutlu<sup>4</sup>

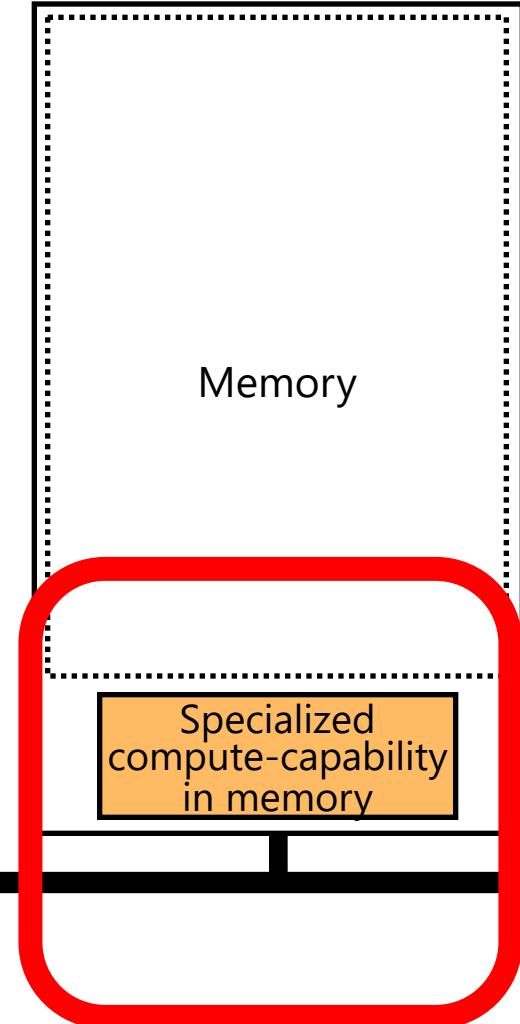
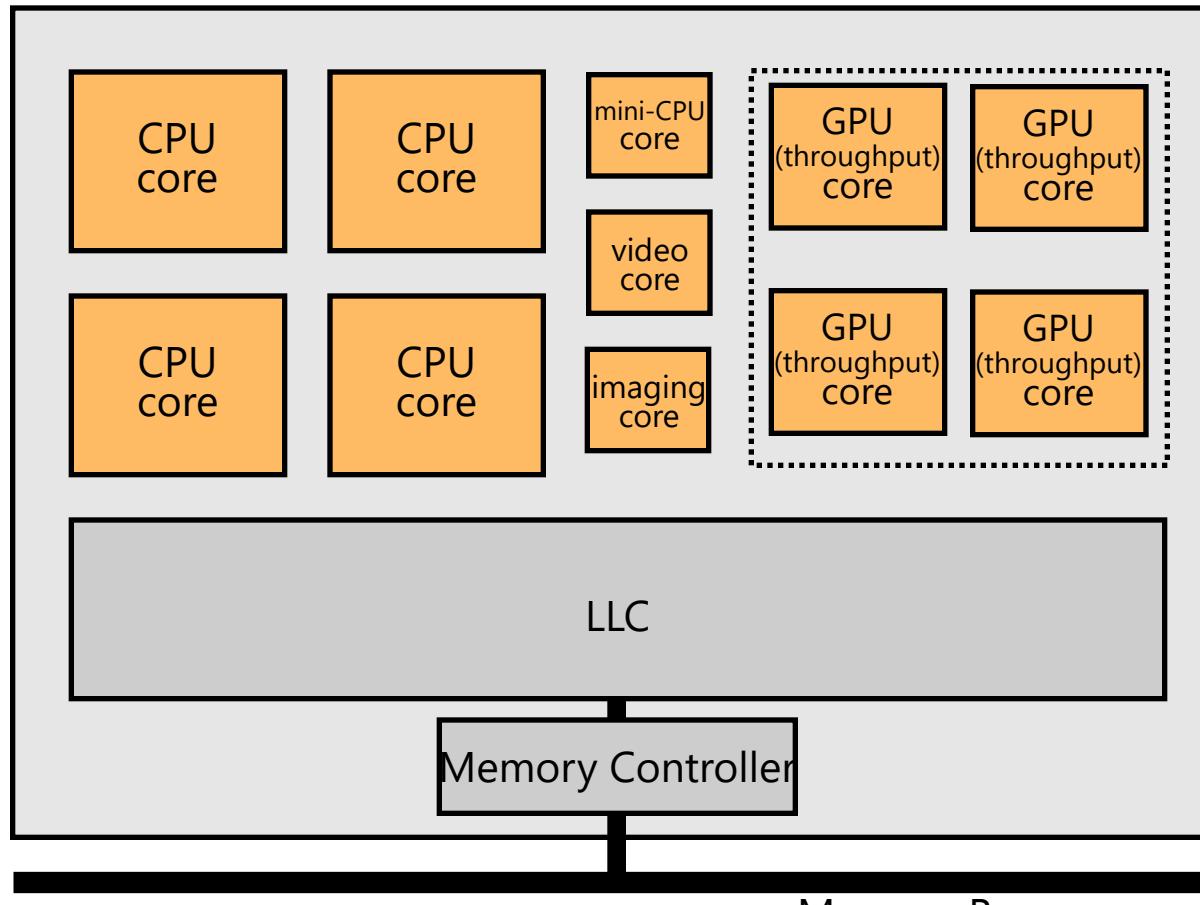
Rachata Ausavarungnirun<sup>2</sup>  
Masoud Daneshtalab<sup>5</sup>

<sup>1</sup>University of Tehran

<sup>2</sup>King Mongkut's University of Technology North Bangkok  
<sup>4</sup>ETH Zürich

<sup>3</sup>Institute for Research in Fundamental Sciences  
<sup>5</sup>Mälardalens University

# Memory as an Accelerator



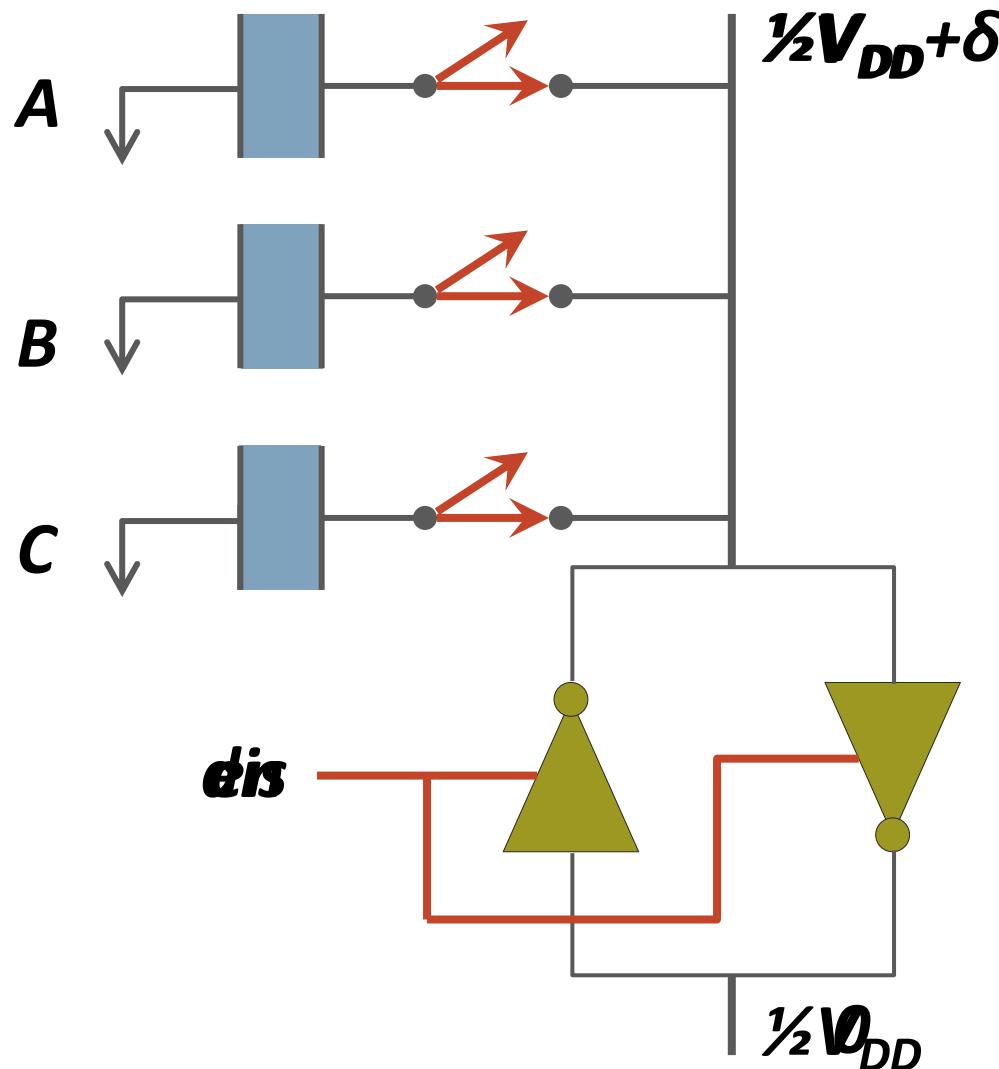
**Memory similar to a “conventional” accelerator**

# In-Memory Bulk Bitwise Operations

---

- We can support **in-DRAM COPY, ZERO, 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**

# In-DRAM AND/OR: Triple Row Activation



Final State  
 $AB + BC + AC$

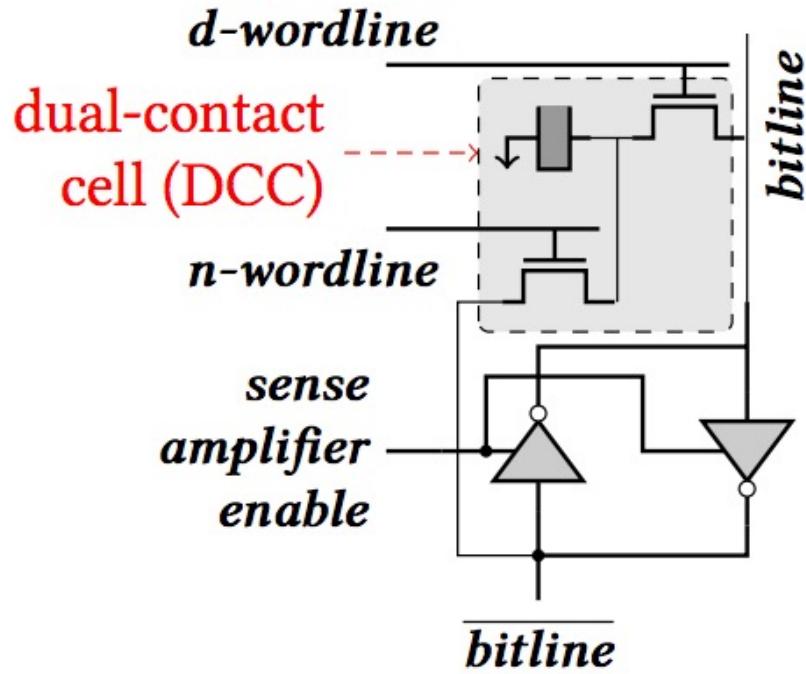
$C(A + B) +$   
 $\sim C(AB)$

# In-DRAM Bulk Bitwise AND/OR Operation

---

- **BULKAND A, B → C**
  - Semantics: Perform a bitwise AND of two rows A and B and store the result in row C
  - R0 – reserved zero row, R1 – reserved one row
  - D1, D2, D3 – Designated rows for triple activation
1. RowClone A into D1
  2. RowClone B into D2
  3. RowClone R0 into D3
  4. ACTIVATE D1,D2,D3
  5. RowClone Result into C

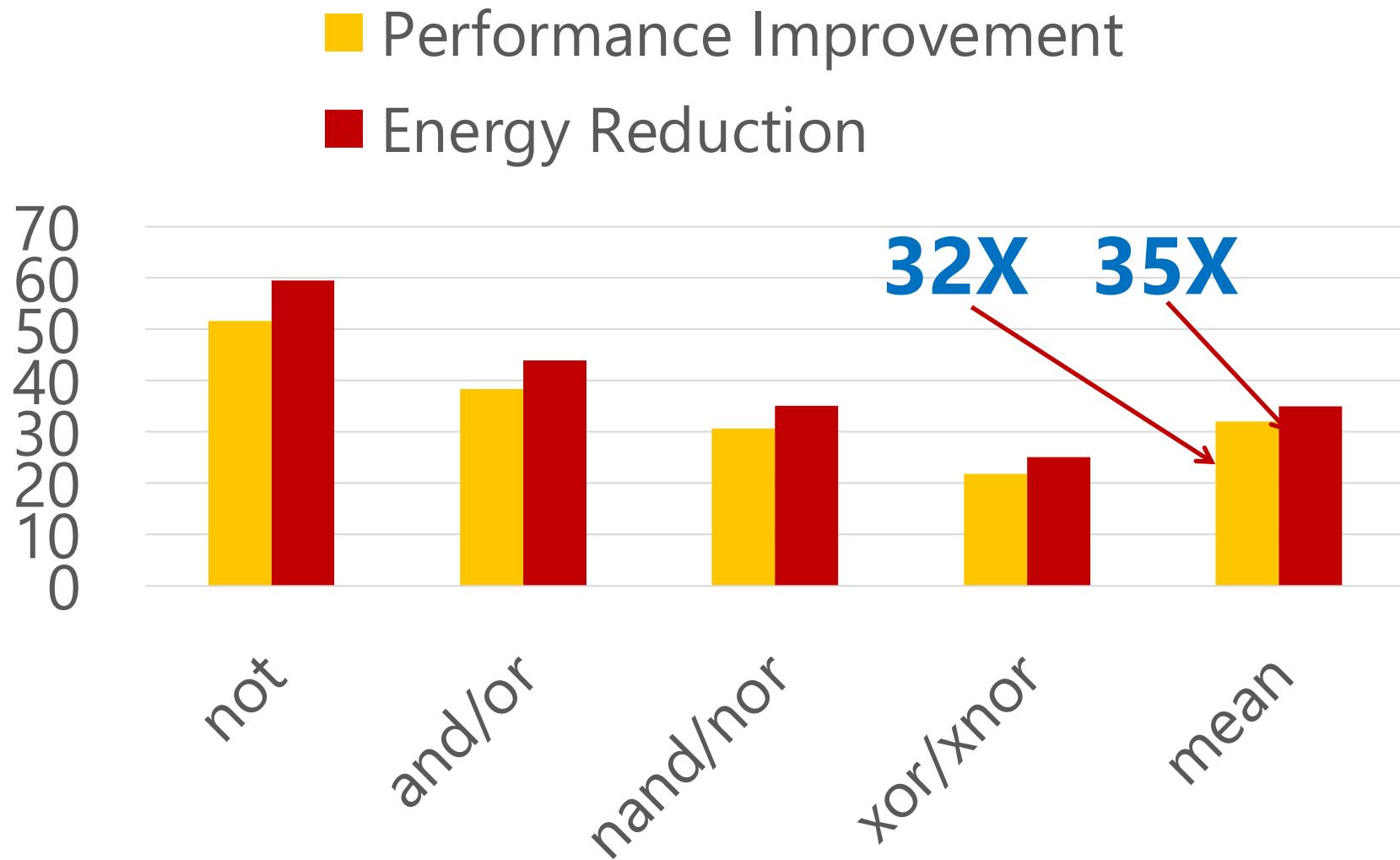
# In-DRAM NOT: Dual Contact Cell



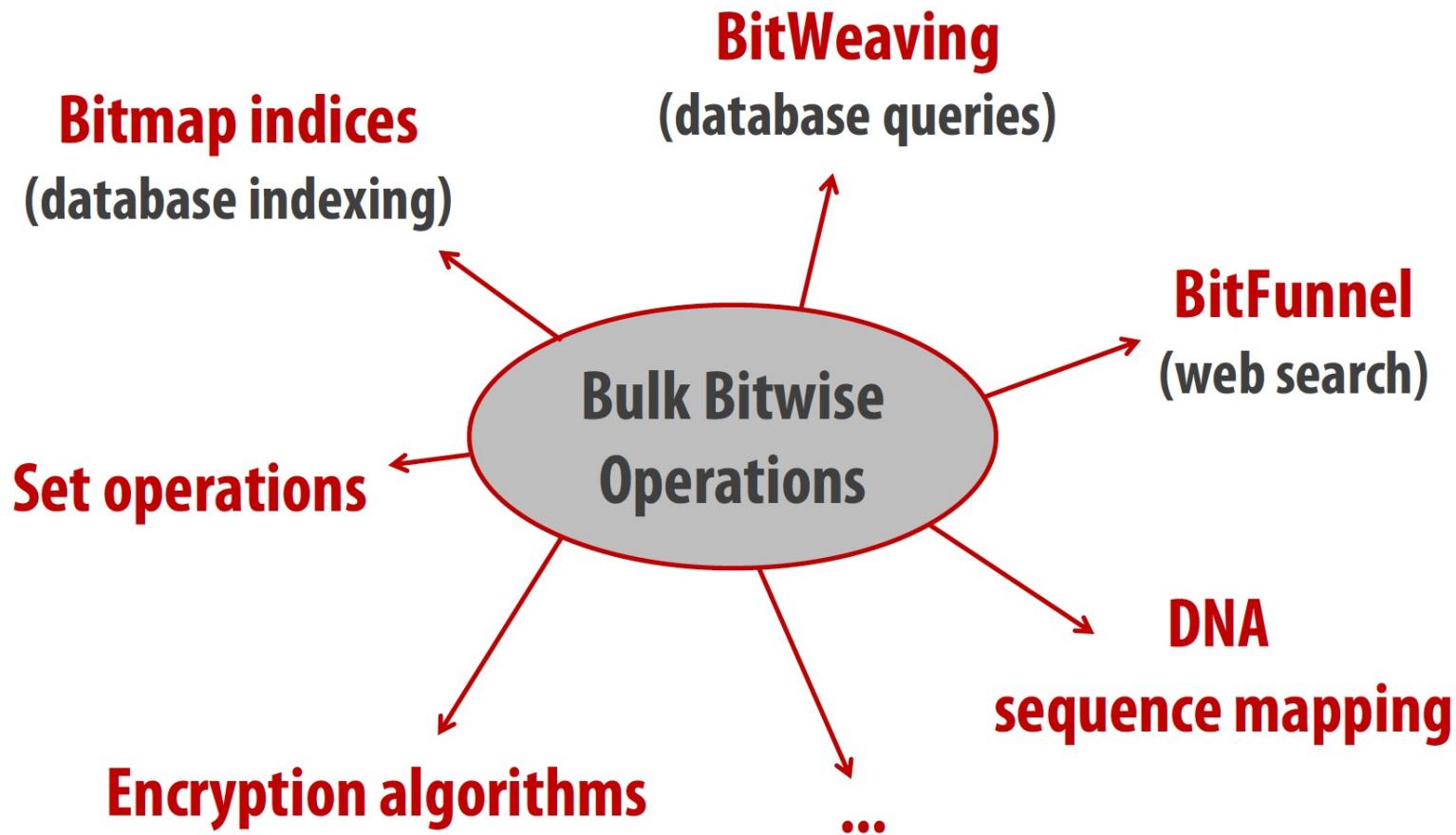
**Figure 5:** A dual-contact cell connected to both ends of a sense amplifier

Idea:  
Feed the negated value  
in the sense amplifier  
into a special row

# Ambit vs. DDR3: Performance and Energy



# Bulk Bitwise Operations in Workloads



# Performance: Bitmap Index on Ambit

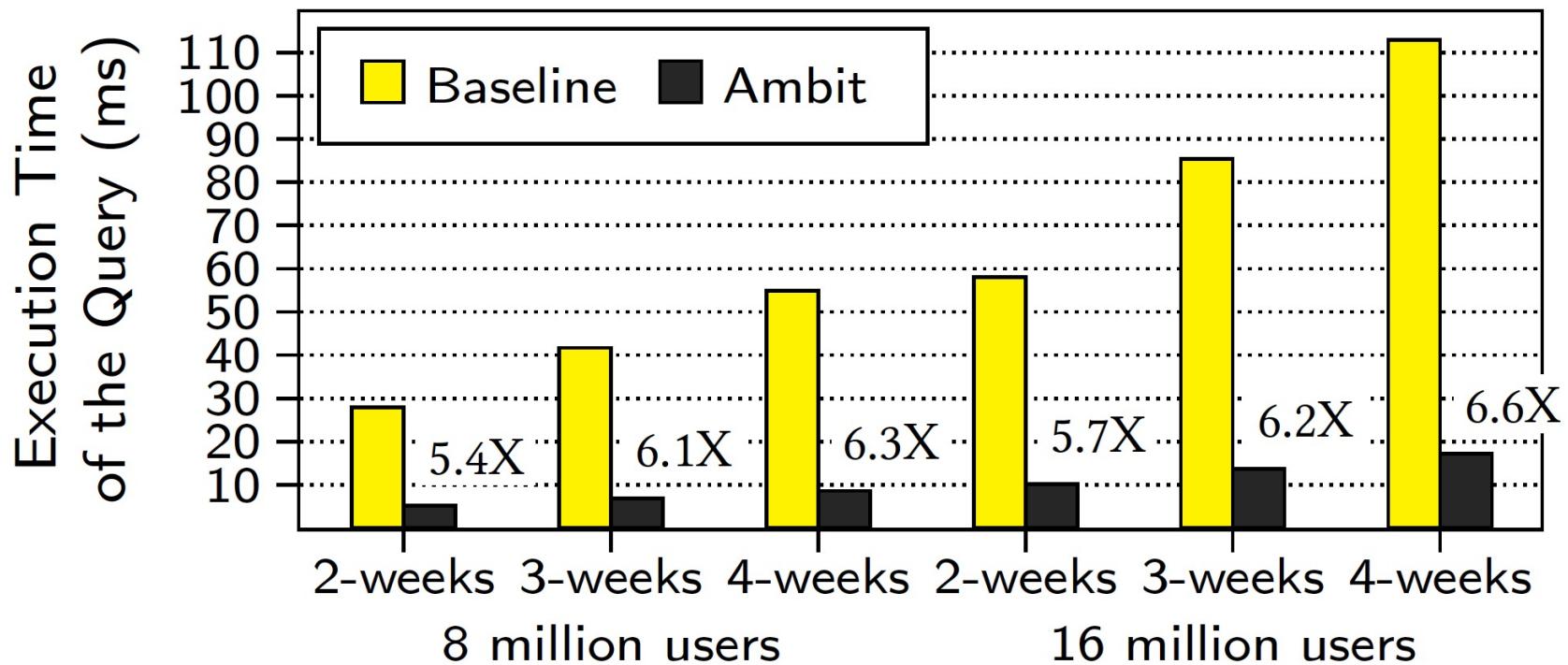


Figure 10: Bitmap index performance. The value above each bar indicates the reduction in execution time due to Ambit.

**>5.4-6.6X Performance Improvement**

Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations using Commodity DRAM Technology," MICRO 2017.

# Performance: BitWeaving on Ambit

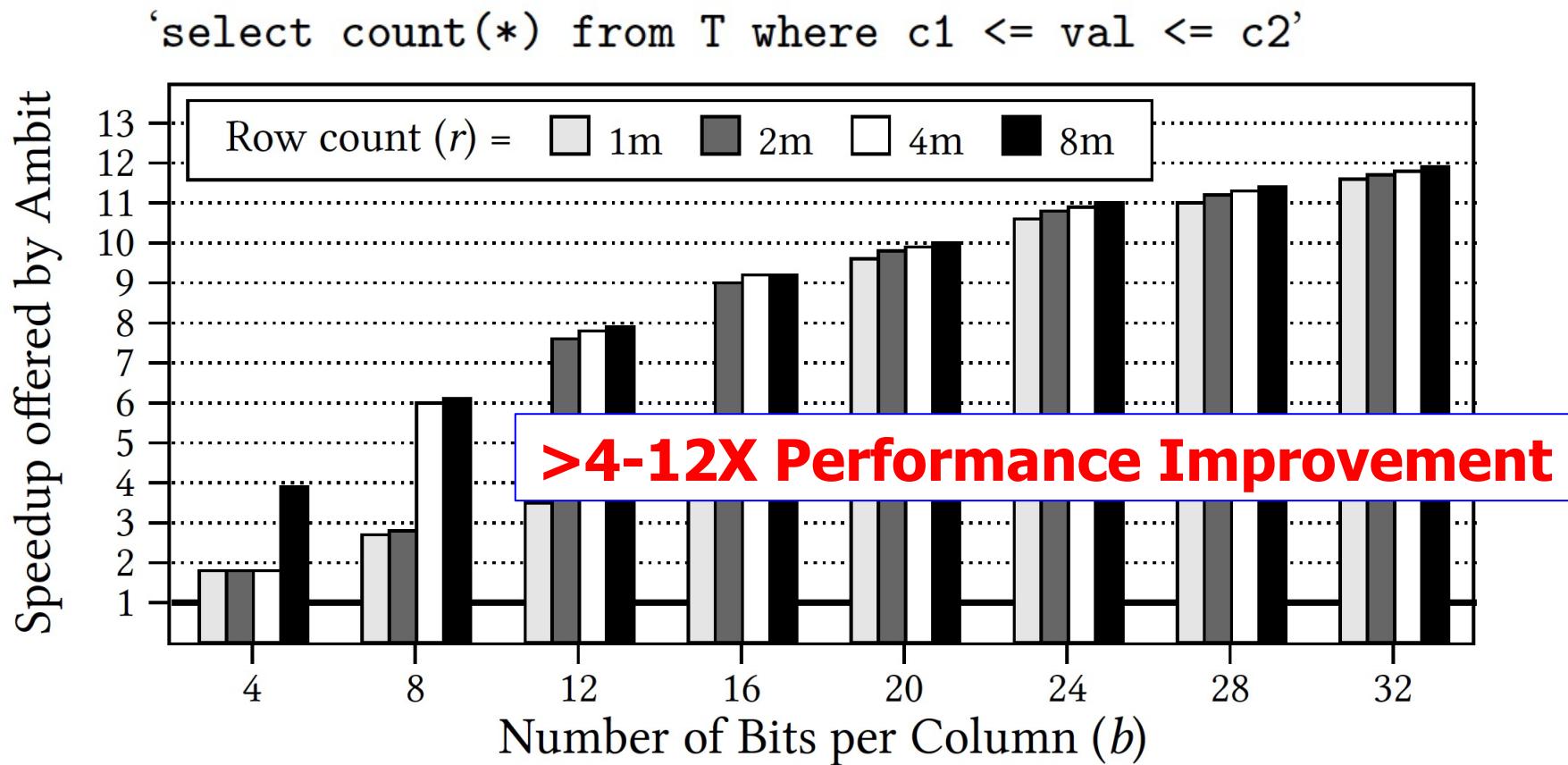


Figure 11: Speedup offered by Ambit over baseline CPU with SIMD for BitWeaving

Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations using Commodity DRAM Technology," MICRO 2017.

# More on In-DRAM Bulk AND/OR

---

- Vivek Seshadri, Kevin Hsieh, Amirali Boroumand, Donghyuk Lee, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,

**"Fast Bulk Bitwise AND and OR in DRAM"**

IEEE Computer Architecture Letters (CAL), April 2015.

## Fast Bulk Bitwise AND and OR in DRAM

Vivek Seshadri\*, Kevin Hsieh\*, Amirali Boroumand\*, Donghyuk Lee\*,  
Michael A. Kozuch†, Onur Mutlu\*, Phillip B. Gibbons†, Todd C. Mowry\*

\*Carnegie Mellon University      †Intel Pittsburgh

# More on Ambit

---

- 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<sup>1,5</sup> Donghyuk Lee<sup>2,5</sup> Thomas Mullins<sup>3,5</sup> Hasan Hassan<sup>4</sup> Amirali Boroumand<sup>5</sup>  
Jeremie Kim<sup>4,5</sup> Michael A. Kozuch<sup>3</sup> Onur Mutlu<sup>4,5</sup> Phillip B. Gibbons<sup>5</sup> Todd C. Mowry<sup>5</sup>

<sup>1</sup>Microsoft Research India <sup>2</sup>NVIDIA Research <sup>3</sup>Intel <sup>4</sup>ETH Zürich <sup>5</sup>Carnegie Mellon University

# In-DRAM Bulk Bitwise Execution

---

- 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](mailto:visesha@microsoft.com)

Onur Mutlu  
ETH Zürich  
[onur.mutlu@inf.ethz.ch](mailto: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<sup>1,2</sup>

Nika Mansouri Ghiasi<sup>1</sup>

\*Geraldo F. Oliveira<sup>1</sup>

Minesh Patel<sup>1</sup>

Juan Gómez-Luna<sup>1</sup>

Sven Gregorio<sup>1</sup>

Mohammed Alser<sup>1</sup>

Onur Mutlu<sup>1</sup>

João Dinis Ferreira<sup>1</sup>

Saugata Ghose<sup>3</sup>

<sup>1</sup>ETH Zürich

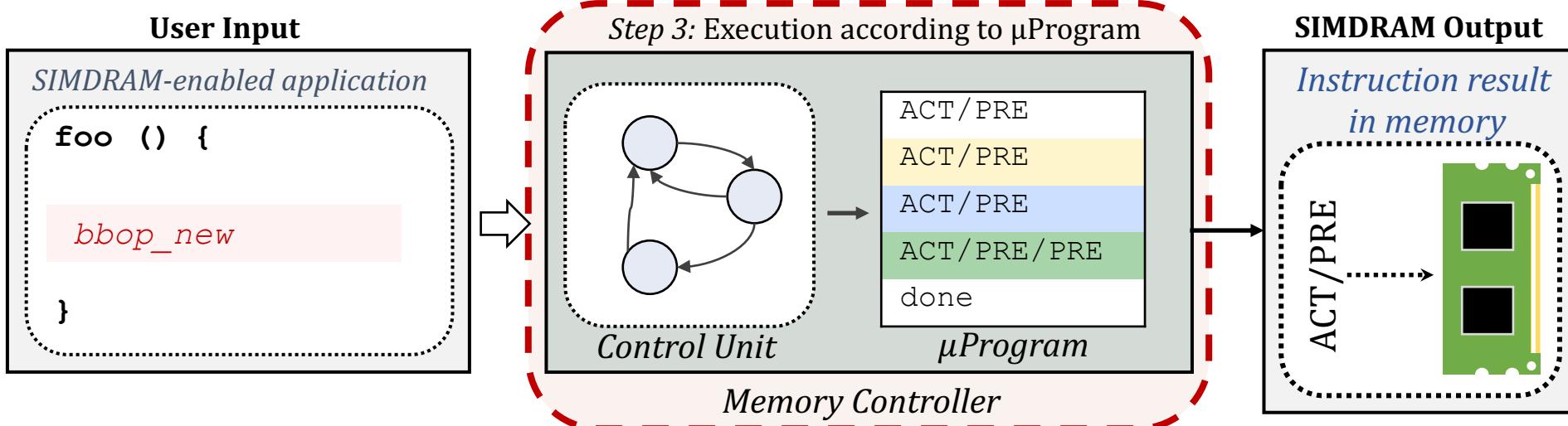
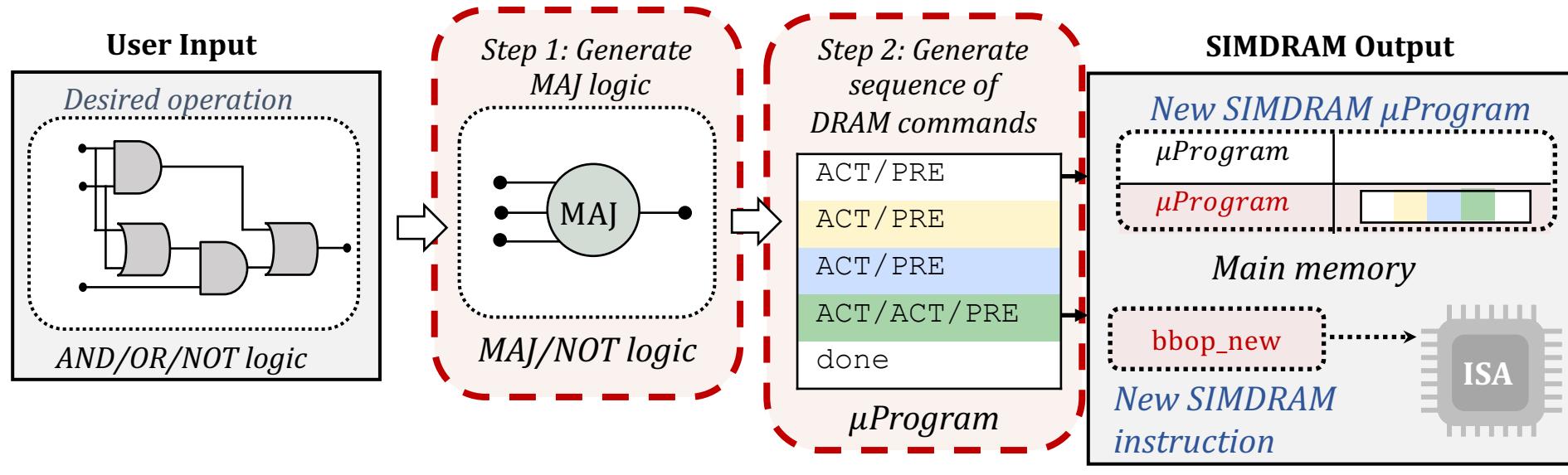
<sup>2</sup>Simon Fraser University

<sup>3</sup>University 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



# More on the 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<sup>1,2</sup>

Nika Mansouri Ghiasi<sup>1</sup>

\*Geraldo F. Oliveira<sup>1</sup>

Minesh Patel<sup>1</sup>

Juan Gómez-Luna<sup>1</sup>

Sven Gregorio<sup>1</sup>

Mohammed Alser<sup>1</sup>

Onur Mutlu<sup>1</sup>

João Dinis Ferreira<sup>1</sup>

Saugata Ghose<sup>3</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Simon Fraser University

<sup>3</sup>University of Illinois at Urbana–Champaign

# RowClone & Bitwise Ops in Real DRAM Chips

---

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

Fei Gao

[feig@princeton.edu](mailto:feig@princeton.edu)

Department of Electrical Engineering  
Princeton University

Georgios Tziantzioulis

[georgios.tziantzioulis@princeton.edu](mailto:georgios.tziantzioulis@princeton.edu)

Department of Electrical Engineering  
Princeton University

David Wentzlaff

[wentzlaf@princeton.edu](mailto:wentzlaf@princeton.edu)

Department of Electrical Engineering  
Princeton University

# Pinatubo: RowClone and Bitwise Ops in PCM

---

## **Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-volatile Memories**

Shuangchen Li<sup>1\*</sup>, Cong Xu<sup>2</sup>, Qiaosha Zou<sup>1,5</sup>, Jishen Zhao<sup>3</sup>, Yu Lu<sup>4</sup>, and Yuan Xie<sup>1</sup>

University of California, Santa Barbara<sup>1</sup>, Hewlett Packard Labs<sup>2</sup>  
University of California, Santa Cruz<sup>3</sup>, Qualcomm Inc.<sup>4</sup>, Huawei Technologies Inc.<sup>5</sup>  
{shuangchenli, yuanxie}@ece.ucsb.edu<sup>1</sup>

# 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<sup>†§</sup>      Minesh Patel<sup>§</sup>      Hasan Hassan<sup>§</sup>      Onur Mutlu<sup>§†</sup>  
<sup>†</sup>Carnegie Mellon University      <sup>§</sup>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<sup>†§</sup>

Minesh Patel<sup>§</sup>

Hasan Hassan<sup>§</sup>

Lois Orosa<sup>§</sup>

Onur Mutlu<sup>§‡</sup>

<sup>†</sup>Carnegie Mellon University

<sup>§</sup>ETH Zürich

# Processing in Memory: Two Approaches

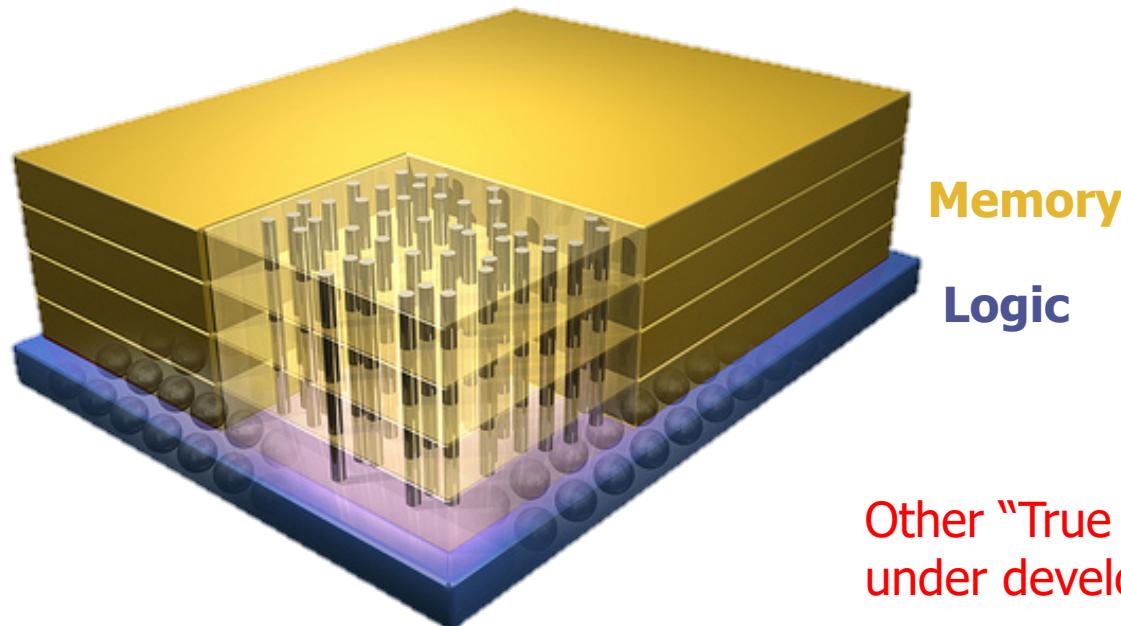
1. Minimally changing memory chips
2. Exploiting 3D-stacked memory

# Opportunity: 3D-Stacked Logic+Memory

---



Hybrid Memory Cube  
C O N S O R T I U M



Other “True 3D” technologies  
under development

# DRAM Landscape (circa 2015)

<i>Segment</i>	<i>DRAM Standards &amp; Architectures</i>
Commodity	DDR3 (2007) [14]; DDR4 (2012) [18]
Low-Power	LPDDR3 (2012) [17]; LPDDR4 (2014) [20]
Graphics	GDDR5 (2009) [15]
Performance	eDRAM [28], [32]; RLDRAM3 (2011) [29]
3D-Stacked	WIO (2011) [16]; WIO2 (2014) [21]; MCDRAM (2015) [13]; HBM (2013) [19]; HMC1.0 (2013) [10]; HMC1.1 (2014) [11]
Academic	SBA/SSA (2010) [38]; Staged Reads (2012) [8]; RAIDR (2012) [27]; SALP (2012) [24]; TL-DRAM (2013) [26]; RowClone (2013) [37]; Half-DRAM (2014) [39]; Row-Buffer Decoupling (2014) [33]; SARP (2014) [6]; AL-DRAM (2015) [25]

Table 1. Landscape of DRAM-based memory

Kim+, "Ramulator: A Flexible and Extensible DRAM Simulator", IEEE CAL 2015.

# Two Key Questions in 3D-Stacked PIM

---

- What are the performance and energy benefits of using 3D-stacked memory as a coarse-grained accelerator?
  - By changing the entire system
  - By performing simple function offloading
  
- What is the minimal processing-in-memory support we can provide?
  - With minimal changes to system and programming

# Graph Processing

---

- Large graphs are everywhere (circa 2015)



36 Million  
Wikipedia Pages



1.4 Billion  
Facebook Users

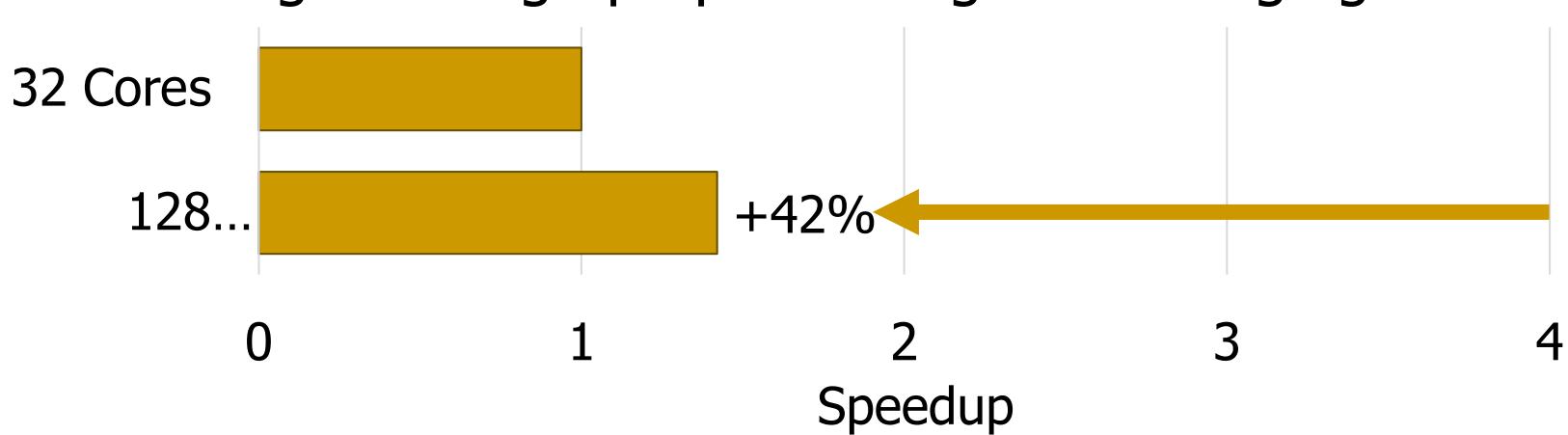


300 Million  
Twitter Users



30 Billion  
Instagram Photos

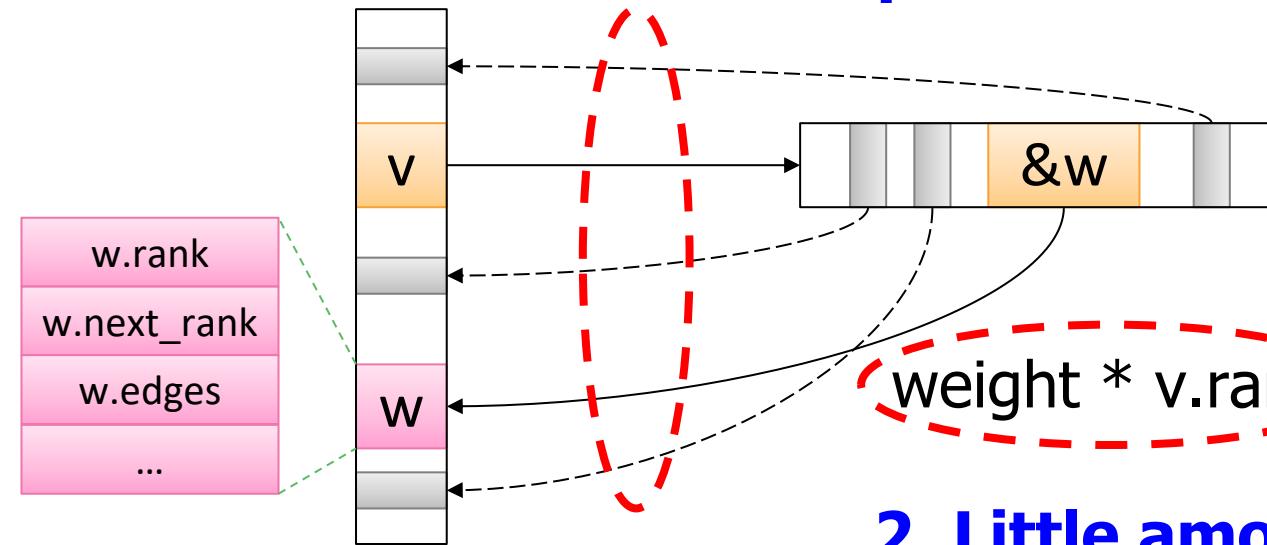
- Scalable large-scale graph processing is challenging



# Key Bottlenecks in Graph Processing

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

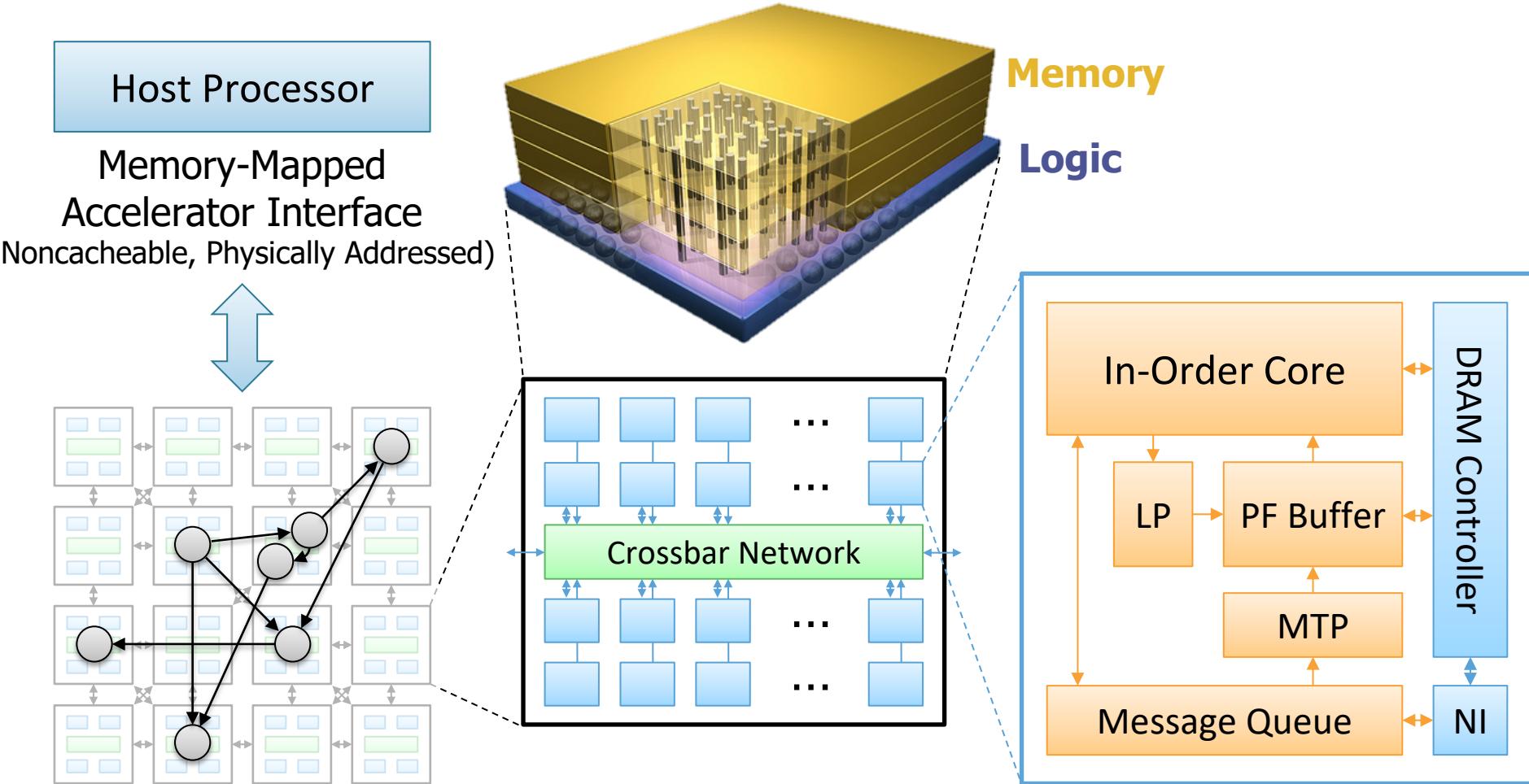
## 1. Frequent random memory accesses



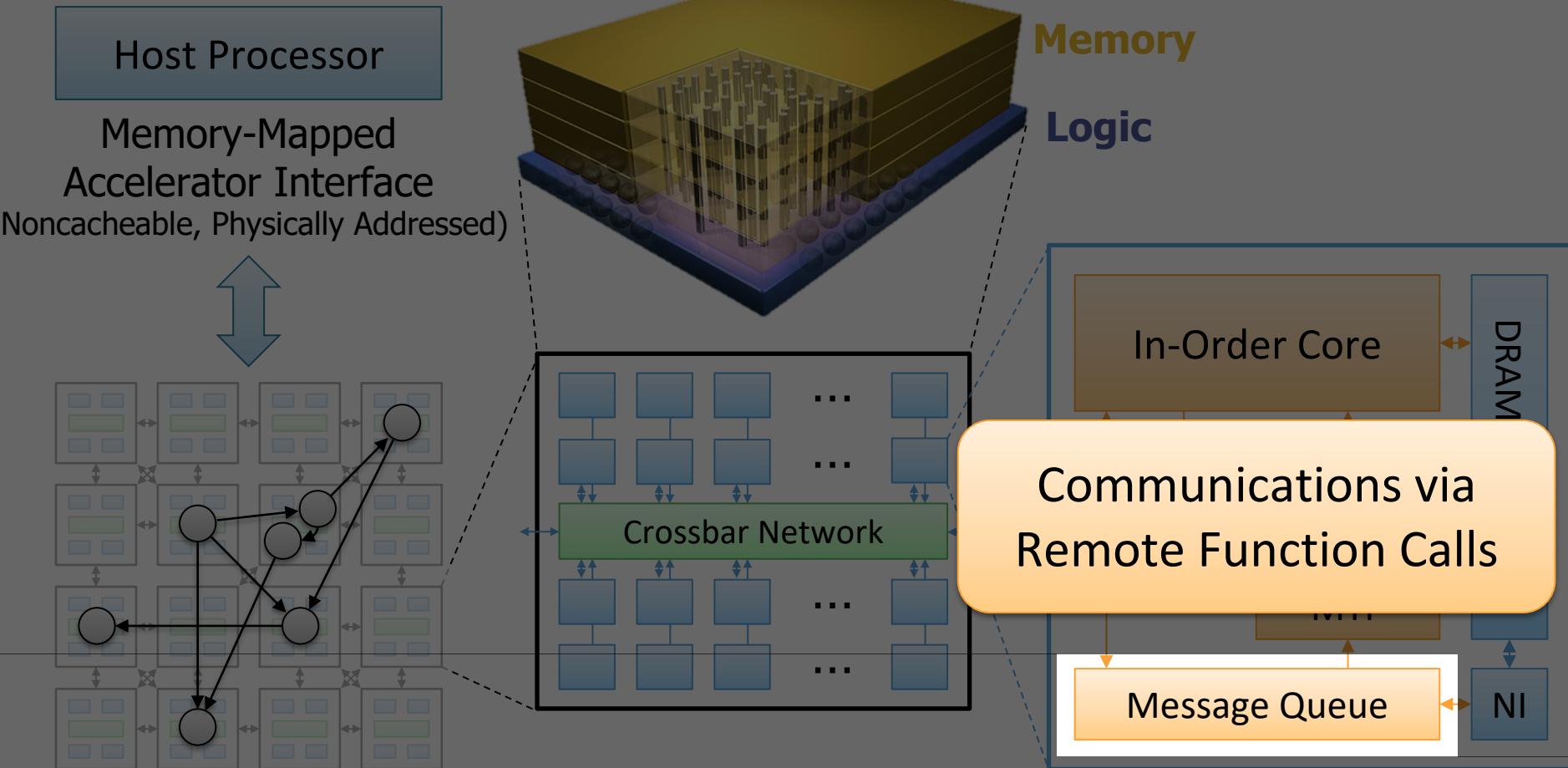
## 2. Little amount of computation

# Tesseract System for Graph Processing

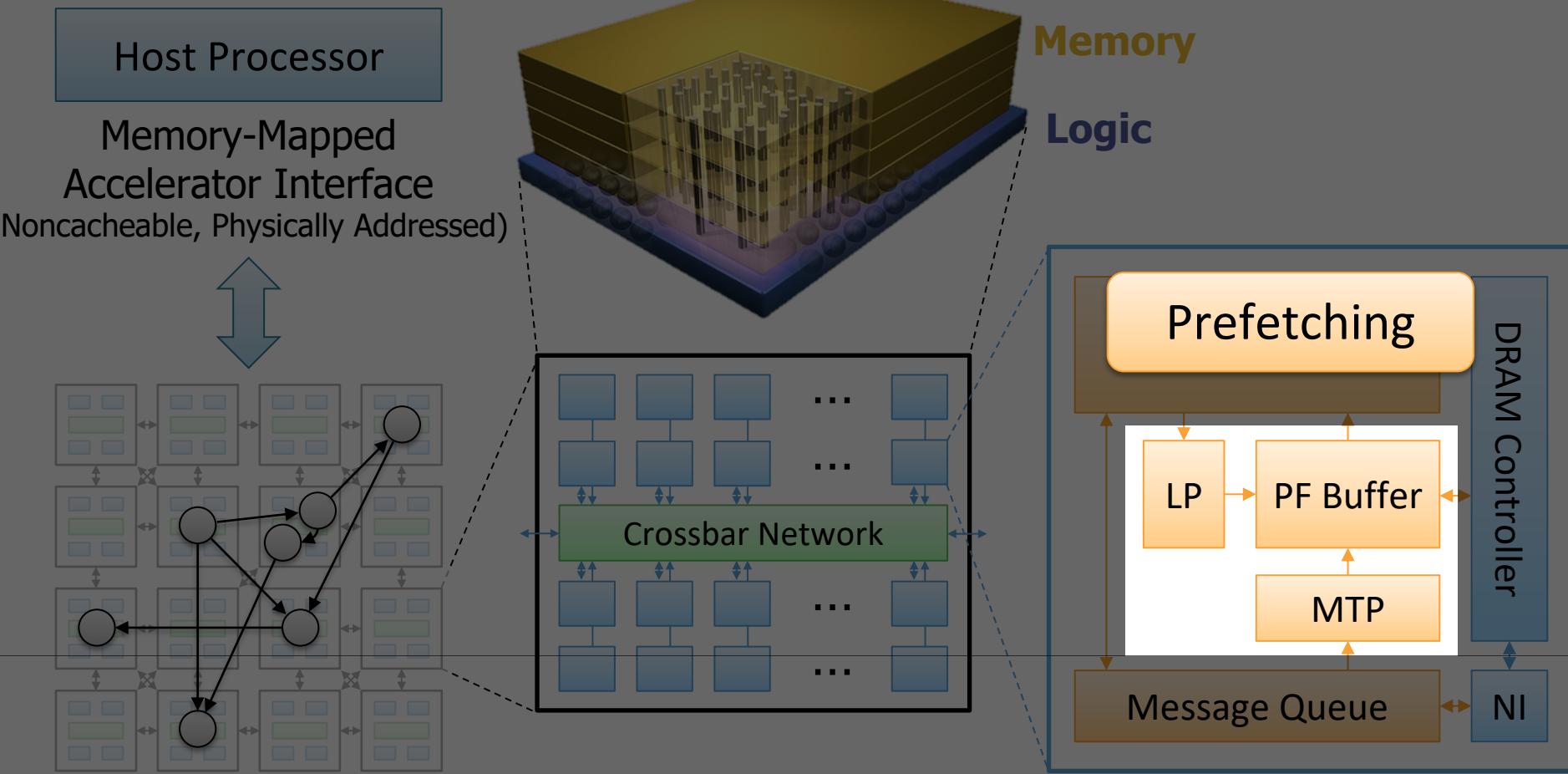
Interconnected set of 3D-stacked memory+logic chips with simple cores



# Tesseract System for Graph Processing

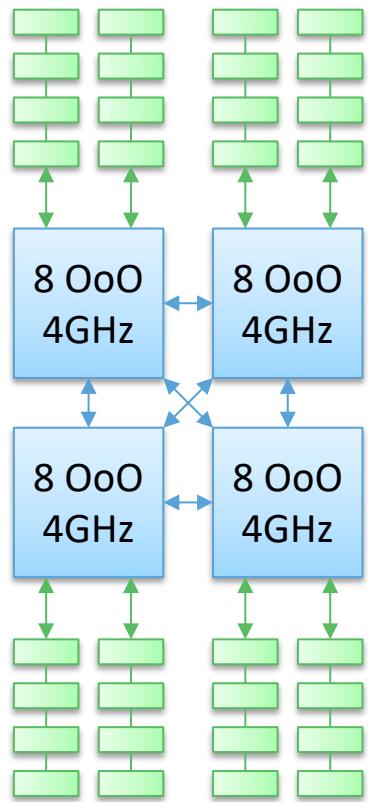


# Tesseract System for Graph Processing

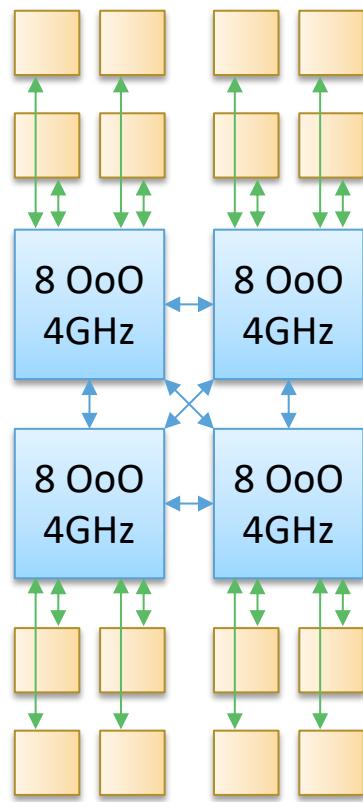


# Evaluated Systems

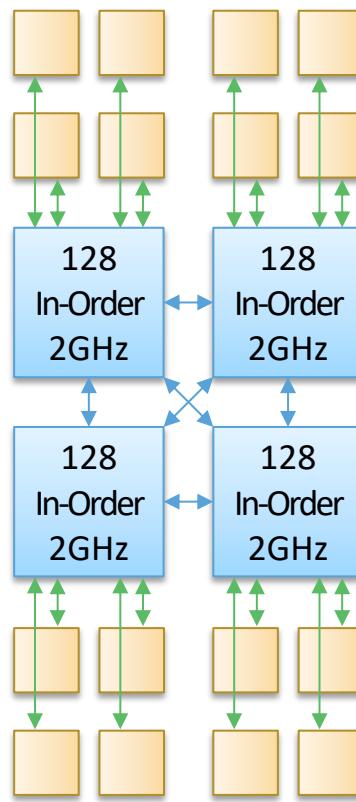
DDR3-OoO



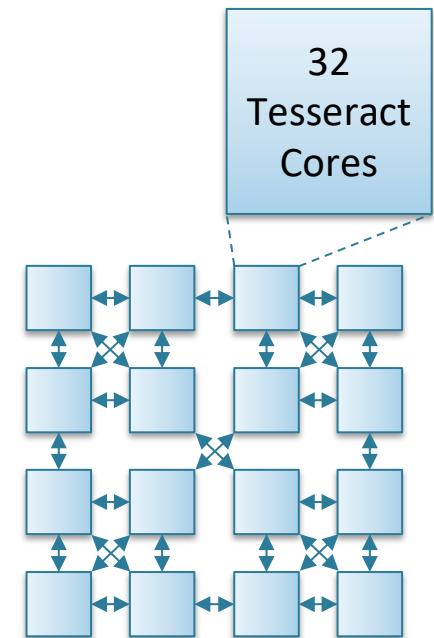
HMC-OoO



HMC-MC



**Tesseract**



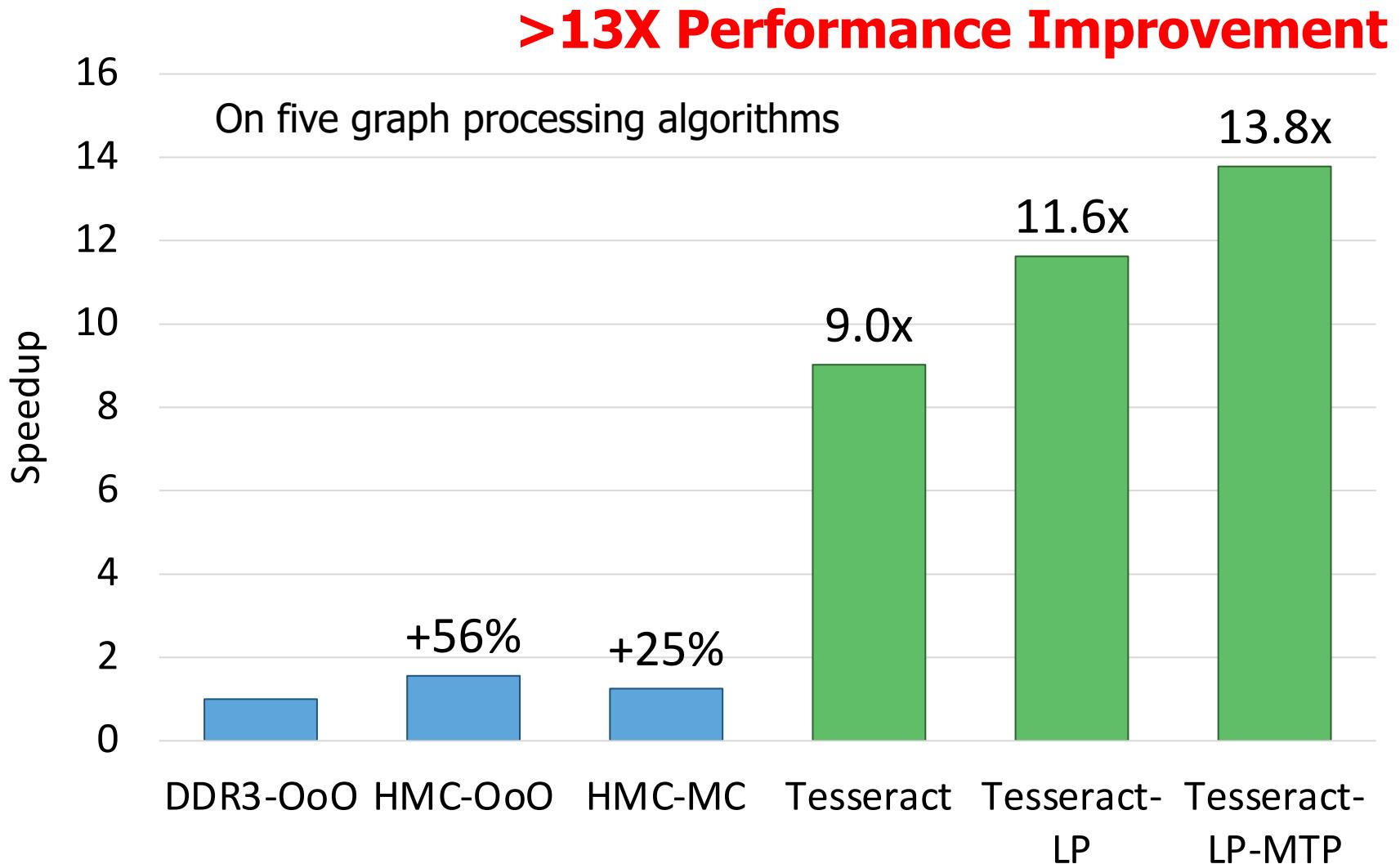
102.4GB/s

640GB/s

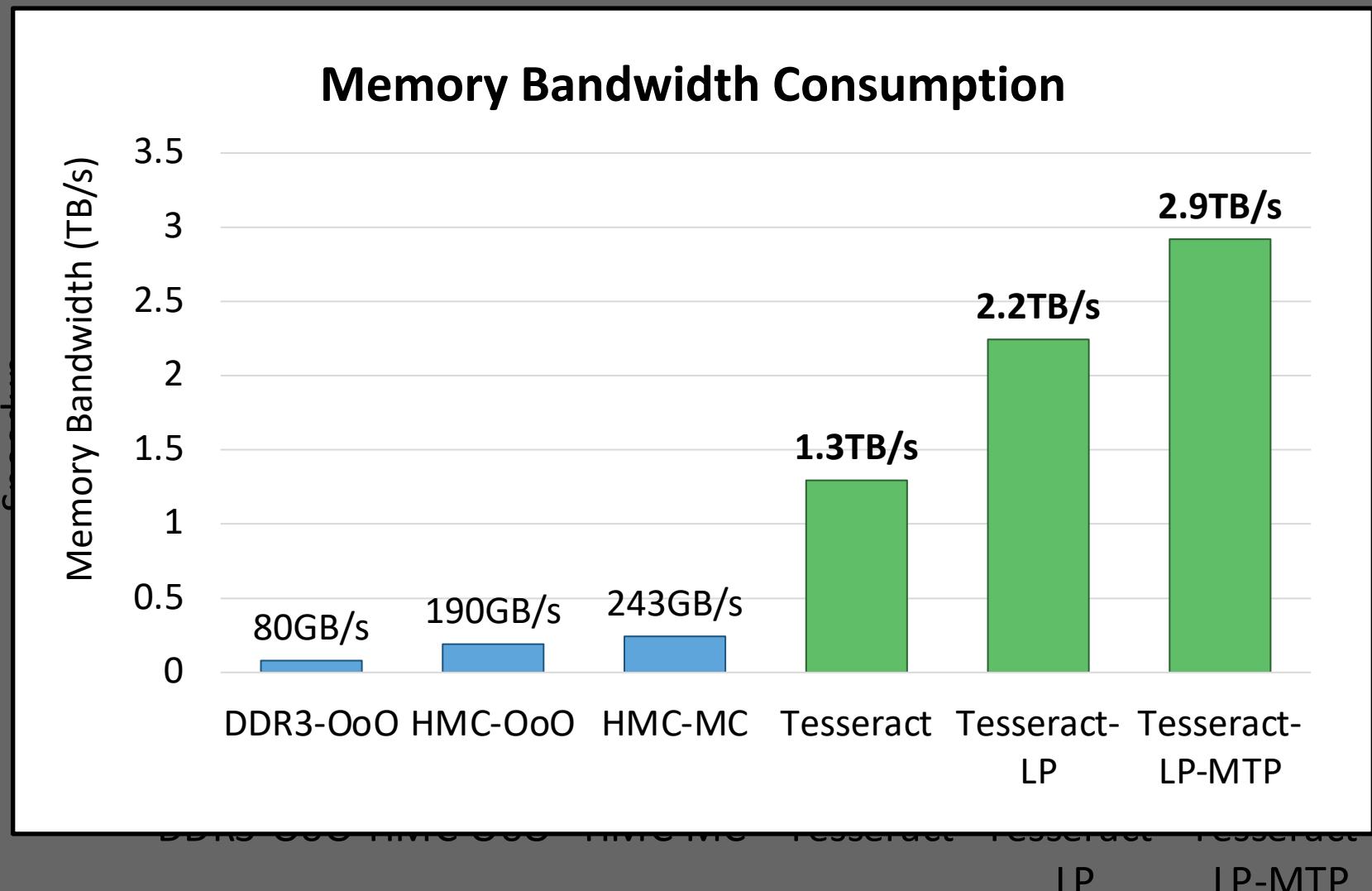
640GB/s

**8TB/s**

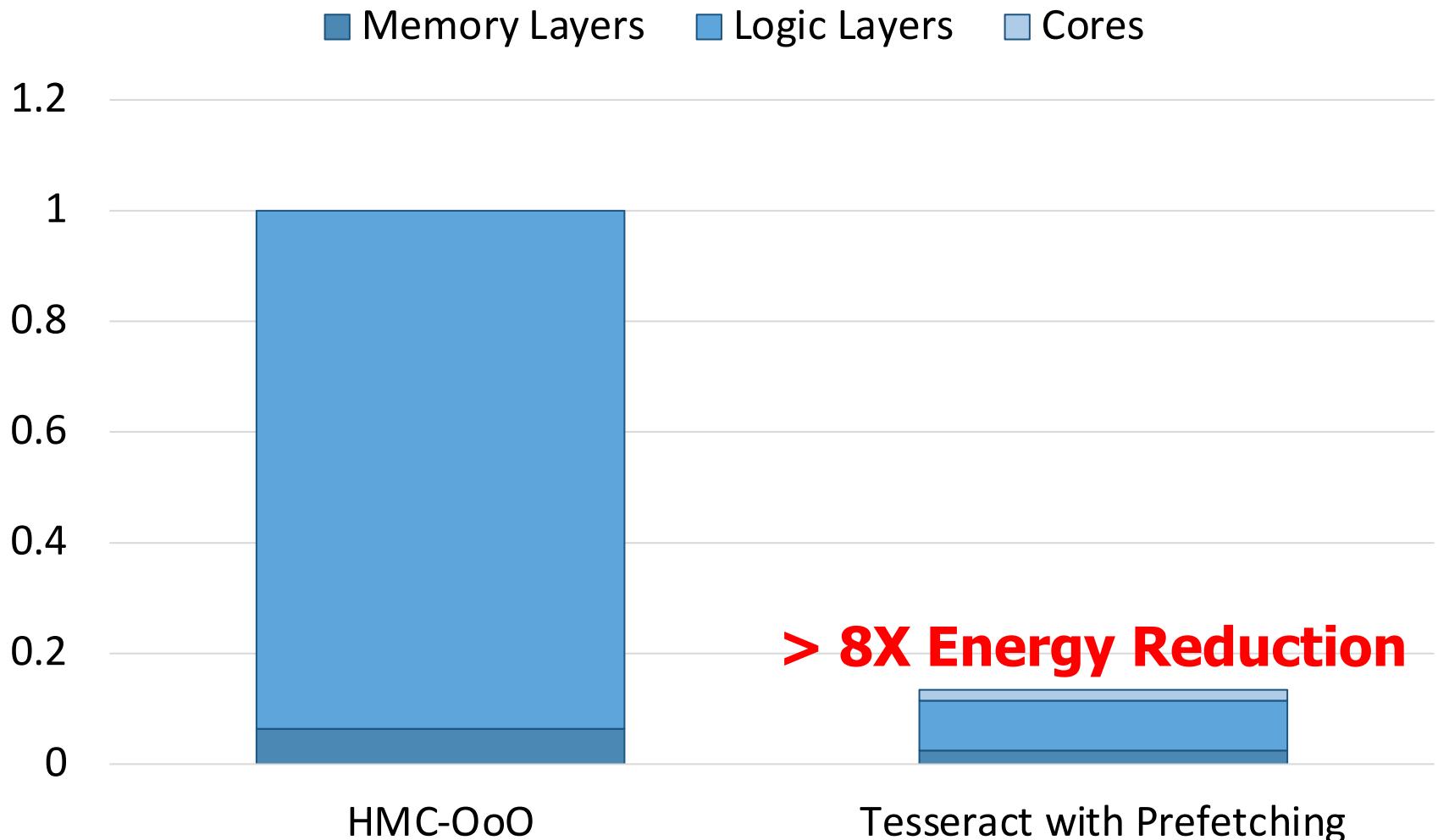
# Tesseract Graph Processing Performance



# Tesseract Graph Processing Performance



# Tesseract Graph Processing System Energy



# More on Tesseract

---

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung 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<sup>§</sup>   Sungjoo Yoo   Onur Mutlu<sup>†</sup>   Kiyoung Choi

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

Seoul National University

<sup>§</sup>Oracle Labs

<sup>†</sup>Carnegie Mellon University

# Two Key Questions in 3D-Stacked PIM

---

- What are the performance and energy benefits of using 3D-stacked memory as a coarse-grained accelerator?
  - By changing the entire system
  - By performing simple function offloading
  
- What is the minimal processing-in-memory support we can provide?
  - With minimal changes to system and programming

# PIM on 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.*

## 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>

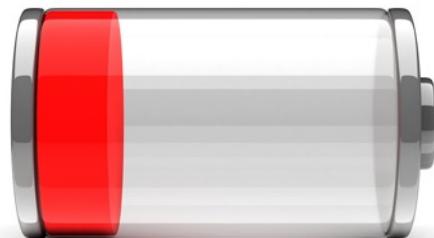
Onur Mutlu<sup>5,1</sup>

# Consumer Devices



**Consumer devices are everywhere!**

**Energy consumption is  
a first-class concern in consumer devices**



# Four Important Workloads



**Chrome**

Google's web browser



**TensorFlow Mobile**

Google's machine learning  
framework

**VP9**



**Video Playback**

Google's **video codec**

**VP9**

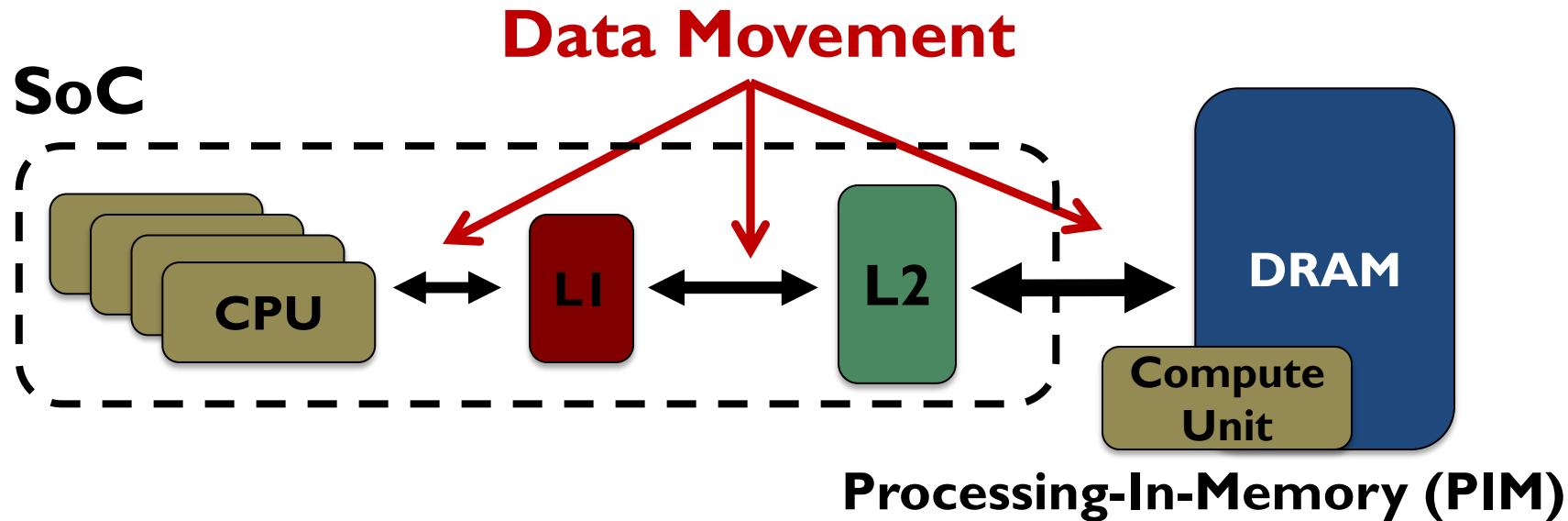


**Video Capture**

Google's **video codec**

# Energy Cost of Data Movement

1<sup>st</sup> key observation: **62.7% of the total system energy is spent on data movement**



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

Challenge: limited area and energy budget

# Using PIM to Reduce Data Movement

**2<sup>nd</sup> key observation:** a significant fraction of the **data movement** often comes from **simple functions**

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

**Small embedded  
low-power core**



**Small fixed-function  
accelerators**



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

# Workload Analysis



**Chrome**

Google's web browser



**TensorFlow Mobile**

Google's machine learning  
framework

**VP9**



**Video Playback**

Google's **video codec**

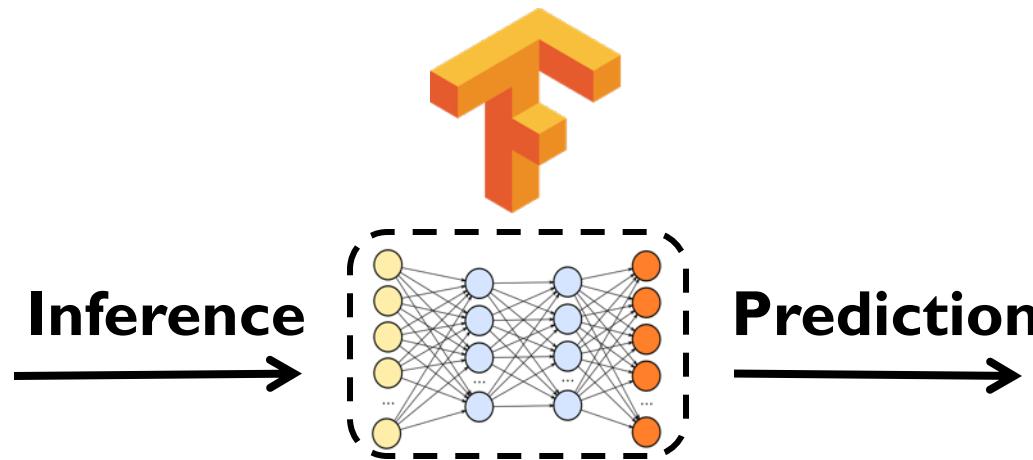
**VP9**



**Video Capture**

Google's **video codec**

# TensorFlow Mobile



**57.3% of the inference energy is spent on  
data movement**



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

# Packing



**Reorders** elements of matrices to minimize  
**cache misses** during **matrix multiplication**



Up to **40%** of the  
inference **energy** and **31%** of  
inference **execution time**



**Packing's data movement**  
accounts for up to  
**35.3%** of the inference **energy**

A simple **data reorganization** process  
that requires **simple arithmetic**

# Quantization



Converts 32-bit floating point to 8-bit integers to improve inference execution time and energy consumption



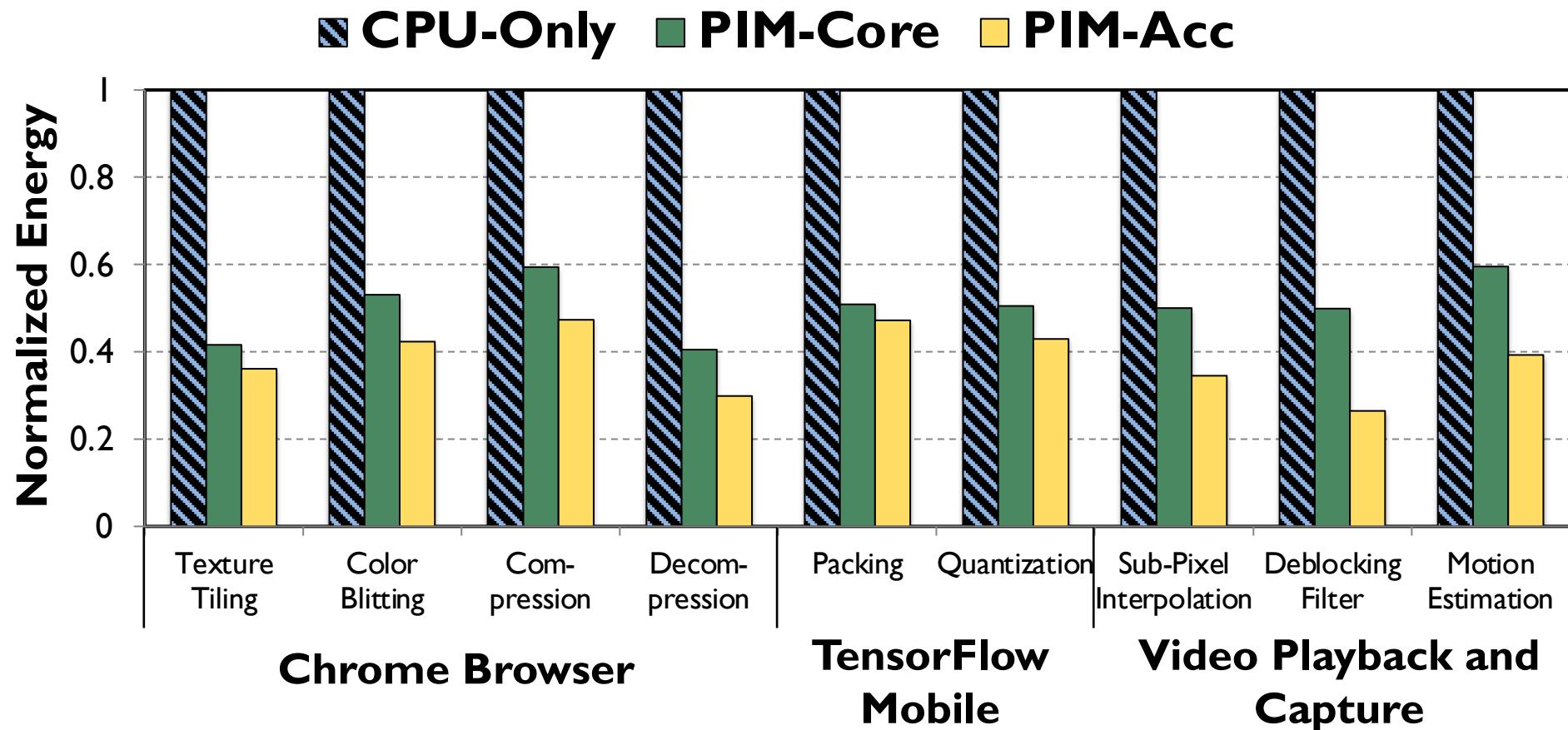
Up to **16.8%** of the inference **energy** and **16.1%** of inference **execution time**



Majority of **quantization energy** comes from **data movement**

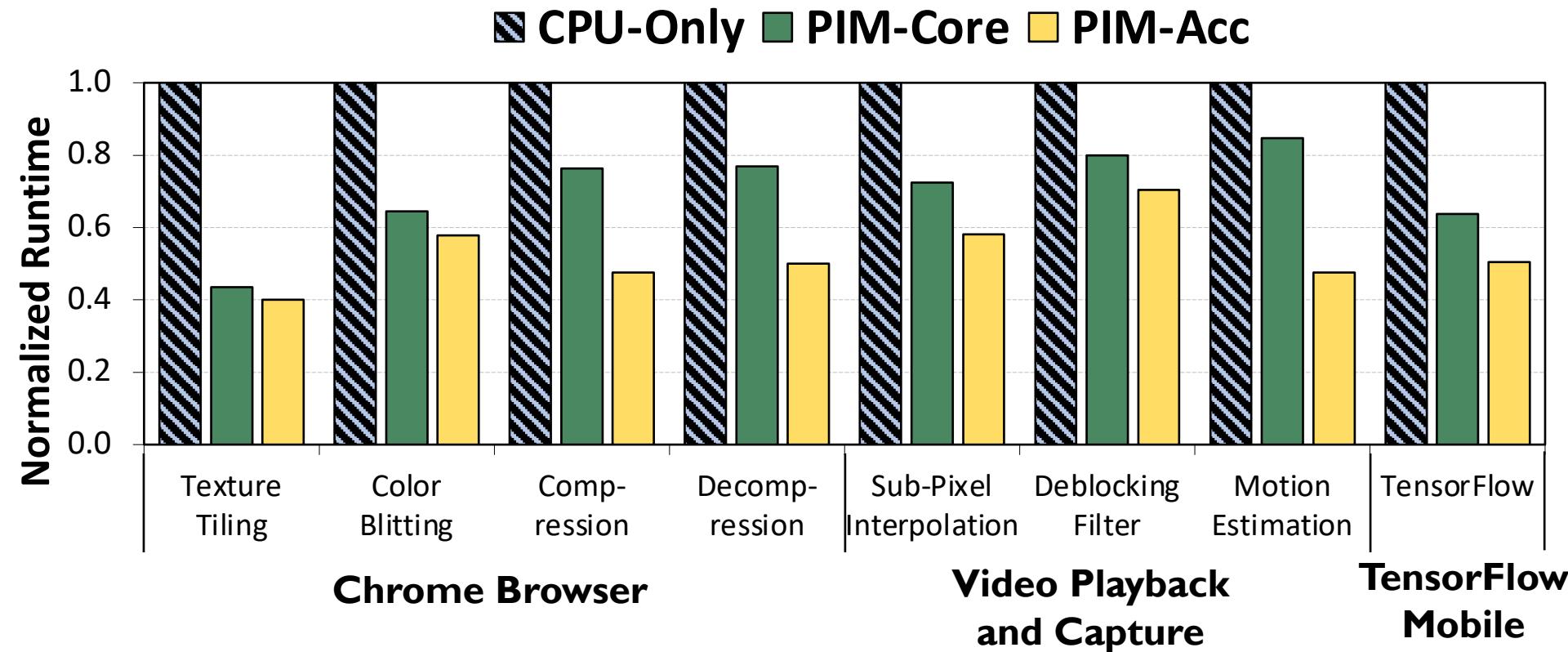
A simple **data conversion** operation that requires **shift**, **addition**, and **multiplication** operations

# Normalized Energy



**PIM core and PIM accelerator reduce  
energy consumption on average by 49.1% and 55.4%**

# Normalized Runtime



Offloading these kernels to **PIM core** and **PIM accelerator** reduces **program runtime** on average by **44.6%** and **54.2%**

# More on PIM for Mobile Devices

---

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

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

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

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

# Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand<sup>1</sup>

Rachata Ausavarungnirun<sup>1</sup>

Aki Kuusela<sup>3</sup>

Saugata Ghose<sup>1</sup>

Eric Shiu<sup>3</sup>

Allan Knies<sup>3</sup>

Youngsok Kim<sup>2</sup>

Rahul Thakur<sup>3</sup>

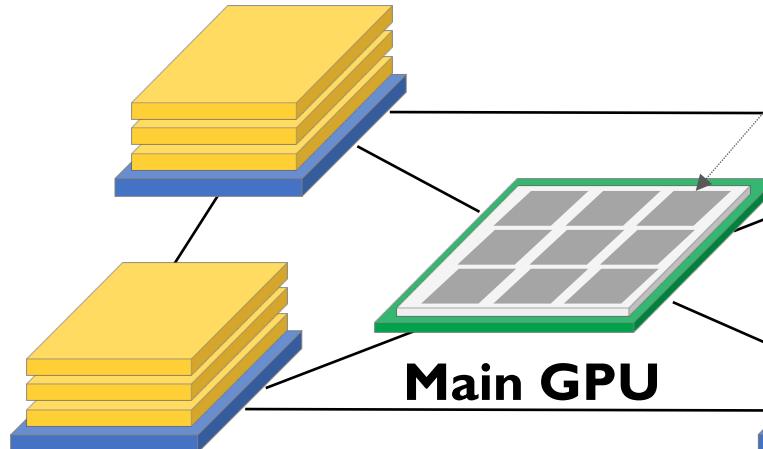
Parthasarathy Ranganathan<sup>3</sup>

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

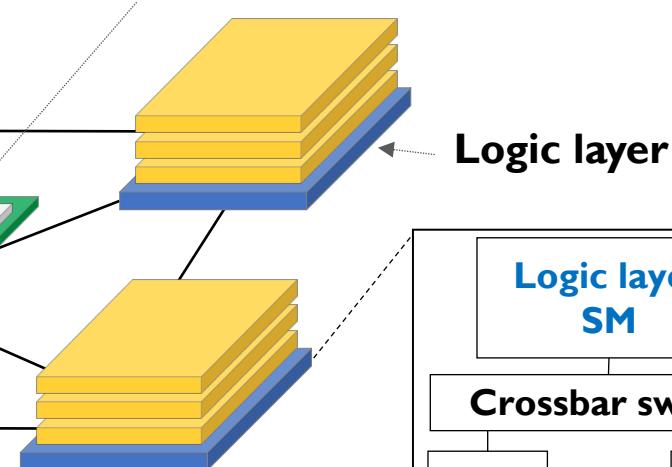
Onur Mutlu<sup>5,1</sup>

# Truly Distributed GPU Processing with PIM?

**3D-stacked memory  
(memory stack)**



**SM (Streaming Multiprocessor)**

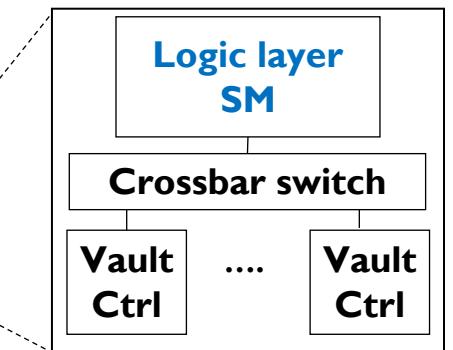


```
global_
void applyScaleFactorsKernel( uint8_T * const out,
    uint8_T const * const in, const double *factor,
    size_t const numRows, size_t const numCols )
{
    // Work out which pixel we are working on.
    const int rowIdx = blockIdx.x * blockDim.x + threadIdx.x;
    const int colIdx = blockIdx.y;
    const int sliceIdx = threadIdx.z;

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

    // Compute the index of my element
    size_t linearIdx = rowIdx + colIdx*numRows +
        sliceIdx*numRows*numCols;
```

**Logic layer**



# Accelerating GPU Execution with PIM (I)

---

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler,

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

*Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.*

[Slides (pptx) (pdf)]

[Lightning Session Slides (pptx) (pdf)]

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

Kevin Hsieh<sup>‡</sup> Eiman Ebrahimi<sup>†</sup> Gwangsun Kim<sup>\*</sup> Niladrish Chatterjee<sup>†</sup> Mike O'Connor<sup>†</sup>  
Nandita Vijaykumar<sup>‡</sup> Onur Mutlu<sup>§‡</sup> Stephen W. Keckler<sup>†</sup>

<sup>‡</sup>Carnegie Mellon University    <sup>†</sup>NVIDIA    <sup>\*</sup>KAIST    <sup>§</sup>ETH Zürich

# Accelerating GPU Execution with PIM (II)

---

- Ashutosh Pattnaik, Xulong Tang, Adwait Jog, Onur Kayiran, Asit K. Mishra, Mahmut T. Kandemir, Onur Mutlu, and Chita R. Das,  
**"Scheduling Techniques for GPU Architectures with Processing-In-Memory Capabilities"**

*Proceedings of the 25th International Conference on Parallel Architectures and Compilation Techniques (PACT), Haifa, Israel, September 2016.*

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

Ashutosh Pattnaik<sup>1</sup>    Xulong Tang<sup>1</sup>    Adwait Jog<sup>2</sup>    Onur Kayıran<sup>3</sup>  
Asit K. Mishra<sup>4</sup>    Mahmut T. Kandemir<sup>1</sup>    Onur Mutlu<sup>5,6</sup>    Chita R. Das<sup>1</sup>

<sup>1</sup>Pennsylvania State University    <sup>2</sup>College of William and Mary

<sup>3</sup>Advanced Micro Devices, Inc.    <sup>4</sup>Intel Labs    <sup>5</sup>ETH Zürich    <sup>6</sup>Carnegie Mellon University

# Accelerating Linked Data Structures

---

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

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

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

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

Kevin Hsieh<sup>†</sup>   Samira Khan<sup>‡</sup>   Nandita Vijaykumar<sup>†</sup>

Kevin K. Chang<sup>†</sup>   Amirali Boroumand<sup>†</sup>   Saugata Ghose<sup>†</sup>   Onur Mutlu<sup>§†</sup>

<sup>†</sup>*Carnegie Mellon University*   <sup>‡</sup>*University of Virginia*   <sup>§</sup>*ETH Zürich*

# Accelerating Dependent Cache Misses

---

- Milad Hashemi, Khubaib, Eiman Ebrahimi, Onur Mutlu, and Yale N. Patt,  
**"Accelerating Dependent Cache Misses with an Enhanced Memory Controller"**

*Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.*

[Slides (pptx) (pdf)]

[Lightning Session Slides (pptx) (pdf)]

## Accelerating Dependent Cache Misses with an Enhanced Memory Controller

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

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

# Accelerating Runahead Execution

---

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

## Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads

Milad Hashemi\*, Onur Mutlu<sup>§</sup>, Yale N. Patt\*

\**The University of Texas at Austin*    <sup>§</sup>*ETH Zürich*

# Accelerating Climate Modeling

---

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

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

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

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

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

[[Talk Video](#) (23 minutes)]

***Nominated for the Stamatis Vassiliadis Memorial Award.***

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

Gagandeep Singh<sup>a,b,c</sup>

Dionysios Diamantopoulos<sup>c</sup>

Christoph Hagleitner<sup>c</sup>

Juan Gómez-Luna<sup>b</sup>

Sander Stuijk<sup>a</sup>

Onur Mutlu<sup>b</sup>

Henk Corporaal<sup>a</sup>

<sup>a</sup>Eindhoven University of Technology

<sup>b</sup>ETH Zürich

<sup>c</sup>IBM Research Europe, Zurich

# Accelerating Approximate String Matching

- 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.

- [[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<sup>†✉</sup> Gurpreet S. Kalsi<sup>✉</sup> Zülal Bingöl<sup>▽</sup> Can Firtina<sup>◊</sup> Lavanya Subramanian<sup>‡</sup> Jeremie S. Kim<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>*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*

# Accelerating Time Series Analysis

---

- Ivan Fernandez, Ricardo Quislant, Christina Giannoula, Mohammed Alser, Juan Gómez-Luna, Eladio Gutiérrez, Oscar Plata, and Onur Mutlu,  
**"NATSA: A Near-Data Processing Accelerator for Time Series Analysis"**  
*Proceedings of the 38th IEEE International Conference on Computer Design (ICCD)*, Virtual, October 2020.  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (10 minutes)]

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

Ivan Fernandez<sup>§</sup>

Ricardo Quislant<sup>§</sup>

Christina Giannoula<sup>†</sup>

Mohammed Alser<sup>‡</sup>

Juan Gómez-Luna<sup>‡</sup>

Eladio Gutiérrez<sup>§</sup>

Oscar Plata<sup>§</sup>

Onur Mutlu<sup>‡</sup>

<sup>§</sup>*University of Malaga*

<sup>†</sup>*National Technical University of Athens*

<sup>‡</sup>*ETH Zürich*

# Two Key Questions in 3D-Stacked PIM

---

- What are the performance and energy benefits of using 3D-stacked memory as a coarse-grained accelerator?
  - By changing the entire system
  - By performing simple function offloading
- What is the minimal processing-in-memory support we can provide?
  - With minimal changes to system and programming

# PEI: PIM-Enabled Instructions (Ideas)

---

- **Goal:** Develop mechanisms to get the most out of near-data processing with **minimal cost**, **minimal changes to the system**, no changes to the programming model
- **Key Idea 1:** Expose each PIM operation as a **cache-coherent, virtually-addressed host processor instruction** (called PEI) that operates on **only a single cache block**
  - e.g., `__pim_add(&w.next_rank, value) → pim.add r1, (r2)`
  - No changes sequential execution/programming model
  - No changes to virtual memory
  - Minimal changes to cache coherence
  - No need for data mapping: Each PEI restricted to a single memory module
- **Key Idea 2: Dynamically decide where to execute a PEI** (i.e., the host processor or PIM accelerator) based on simple locality characteristics and simple hardware predictors
  - Execute each operation at the location that provides the best performance

# PEI: PIM-Enabled Instructions (Example)

```
for (v: graph.vertices) {  
    value = weight * v.rank;  
    for (w: v.successors) {  
        __pim_add(&w.next_rank, value);  
    }  
}  
pfence();
```

pim.add r1, (r2)

Table 1: Summary of Supported PIM Operations

Operation	R	W	Input	Output	Applications
8-byte integer increment	O	O	0 bytes	0 bytes	AT
8-byte integer min	O	O	8 bytes	0 bytes	BFS, SP, WCC
Floating-point add	O	O	8 bytes	0 bytes	PR
Hash table probing	O	X	8 bytes	9 bytes	HJ
Histogram bin index	O	X	1 byte	16 bytes	HG, RP
Euclidean distance	O	X	64 bytes	4 bytes	SC
Dot product	O	X	32 bytes	8 bytes	SVM

- Executed either in memory or in the processor: dynamic decision
  - Low-cost locality monitoring for a single instruction
- Cache-coherent, virtually-addressed, single cache block only
- Atomic between different PEIs
- Not* atomic with normal instructions (use *pfence* for ordering)

# PEI: Initial Evaluation Results

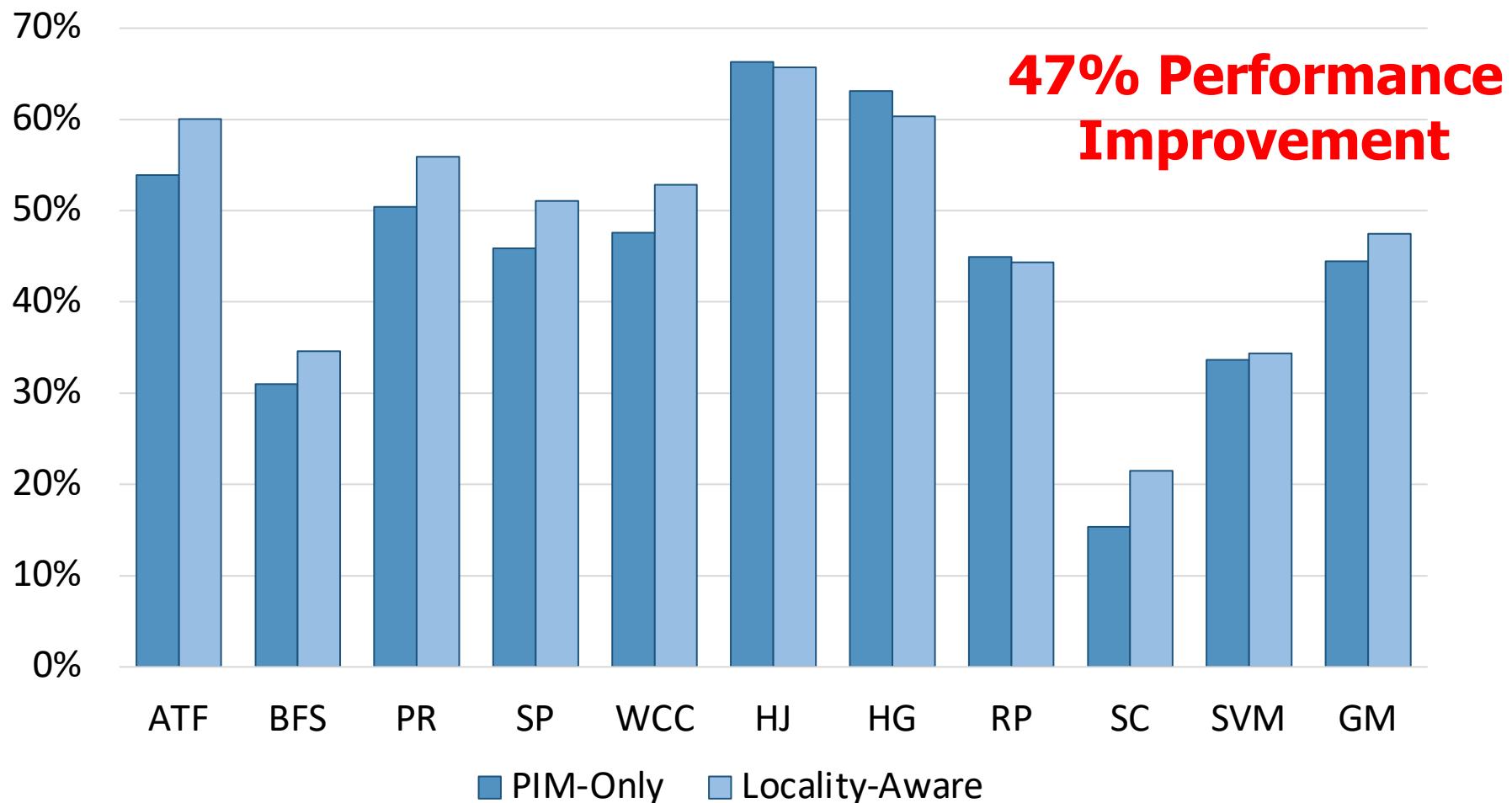
- Initial evaluations with **10 emerging data-intensive workloads**
  - Large-scale graph processing
  - In-memory data analytics
  - Machine learning and data mining
  - Three input sets (small, medium, large) for each workload to analyze the impact of data locality
- Pin-based cycle-level x86-64 simulation
- **Performance Improvement and Energy Reduction:**
  - 47% average speedup with large input data sets
  - 32% speedup with small input data sets
  - 25% avg. energy reduction in a single node with large input data sets

**Table 2: Baseline Simulation Configuration**

Component	Configuration
Core	16 out-of-order cores, 4 GHz, 4-issue
L1 I/D-Cache	Private, 32 KB, 4/8-way, 64 B blocks, 16 MSHRs
L2 Cache	Private, 256 KB, 8-way, 64 B blocks, 16 MSHRs
L3 Cache	Shared, 16 MB, 16-way, 64 B blocks, 64 MSHRs
On-Chip Network	Crossbar, 2 GHz, 144-bit links
Main Memory	32 GB, 8 HMCs, daisy-chain (80 GB/s full-duplex)
HMC	4 GB, 16 vaults, 256 DRAM banks [20]
– DRAM	FR-FCFS, tCL = tRCD = tRP = 13.75 ns [27]
– Vertical Links	64 TSVs per vault with 2 Gb/s signaling rate [23]

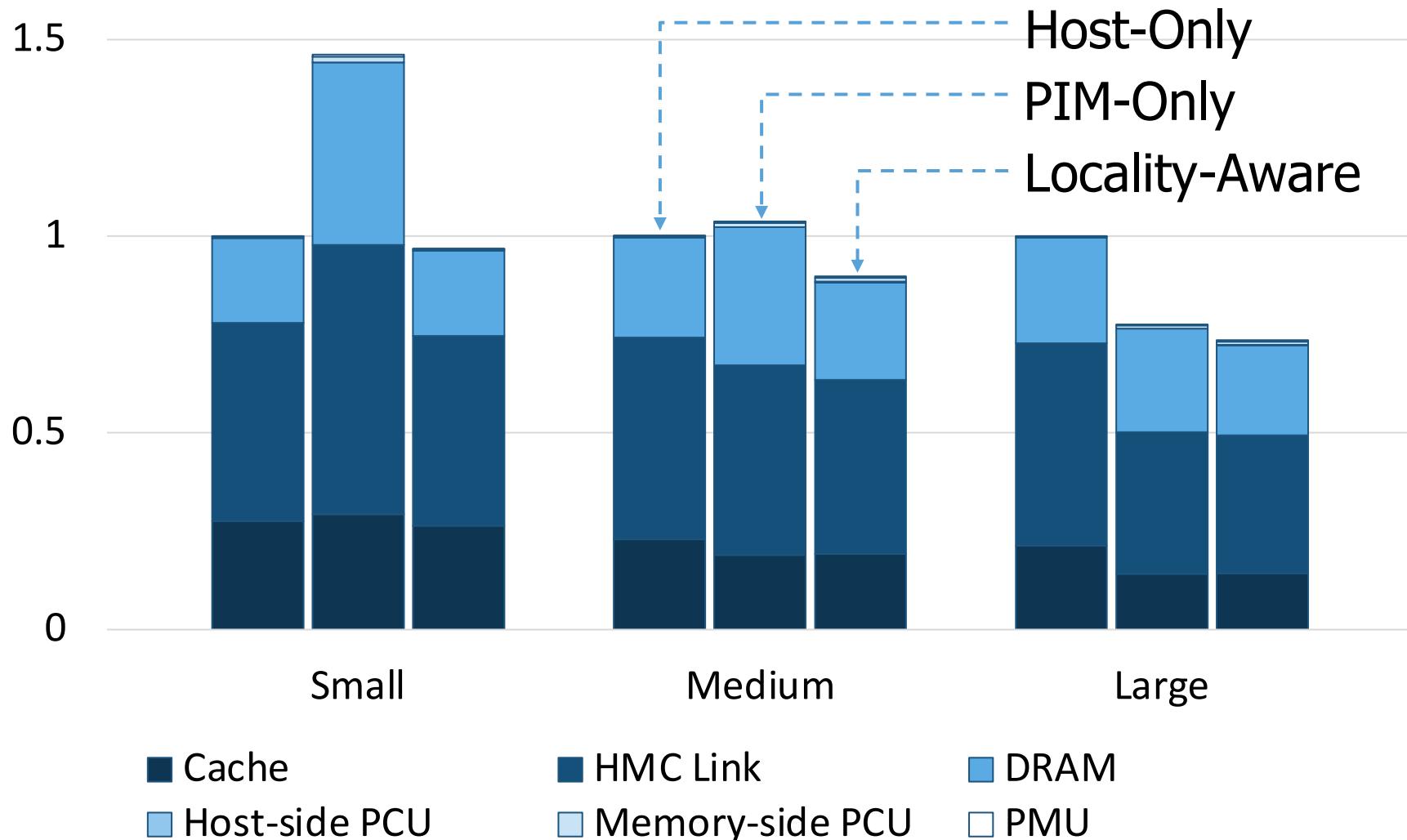
# PEI Performance Delta: Large Data Sets

(Large Inputs, Baseline: Host-Only)



# PEI Energy Consumption

**25% Energy Reduction**



# Simpler PIM: PIM-Enabled Instructions

---

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoung 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<sup>†</sup>   Kiyoung Choi

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

Seoul National University

<sup>†</sup>Carnegie Mellon University

# Eliminating the Adoption Barriers

---

## How to Enable Adoption of Processing in Memory

# Barriers to Adoption of PIM

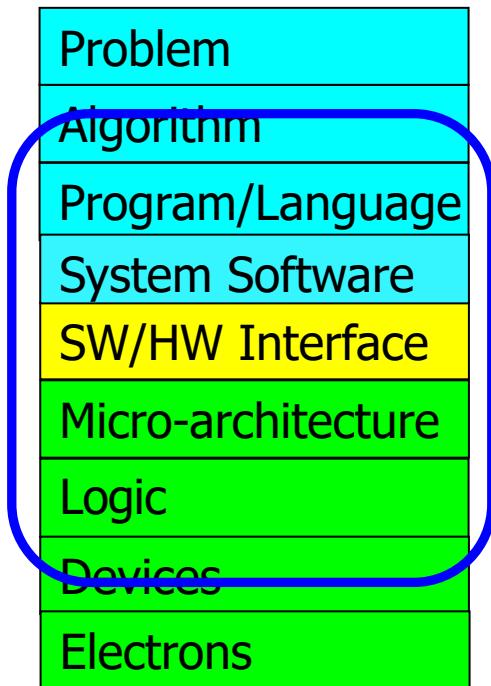
---

1. Functionality of and applications & software for PIM
2. Ease of programming (interfaces and compiler/HW support)
3. System support: coherence & 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**

# DAMOV Methodology & Workloads [SIGMETRICS'21]

---

## 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>.

# 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.*

# PIM Review and Open Problems (II)

---

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

Saugata Ghose<sup>†</sup>      Amirali Boroumand<sup>†</sup>      Jeremie S. Kim<sup>†\\$</sup>      Juan Gómez-Luna<sup>\\$</sup>      Onur Mutlu<sup>\\$†</sup>

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

<sup>\\$</sup>*ETH Zürich*

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

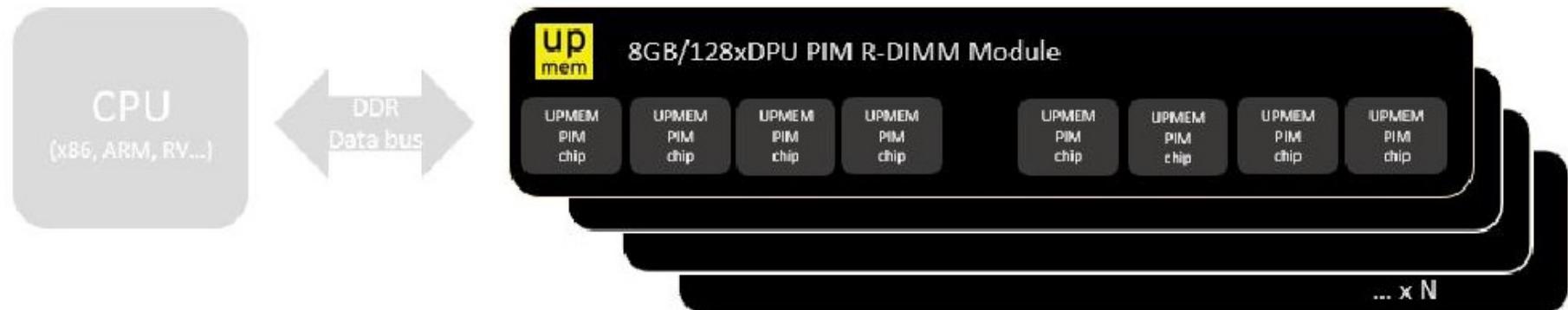
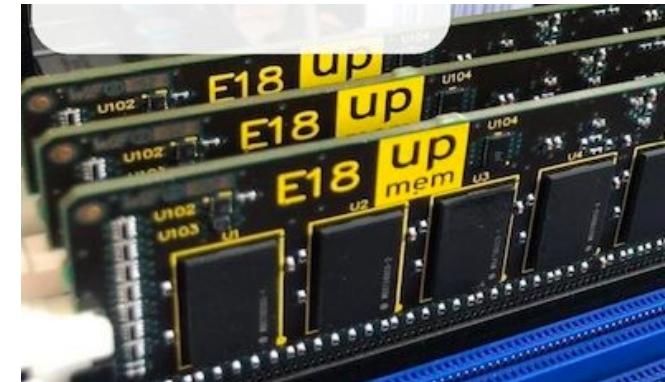
### "Processing-in-Memory: A Workload-Driven Perspective"

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

[Preliminary arXiv version]

# 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



# Samsung Function-in-Memory DRAM (2021)



## Samsung Develops Industry's First High Bandwidth Memory with AI Processing Power

Korea on February 17, 2021

Audio



Share



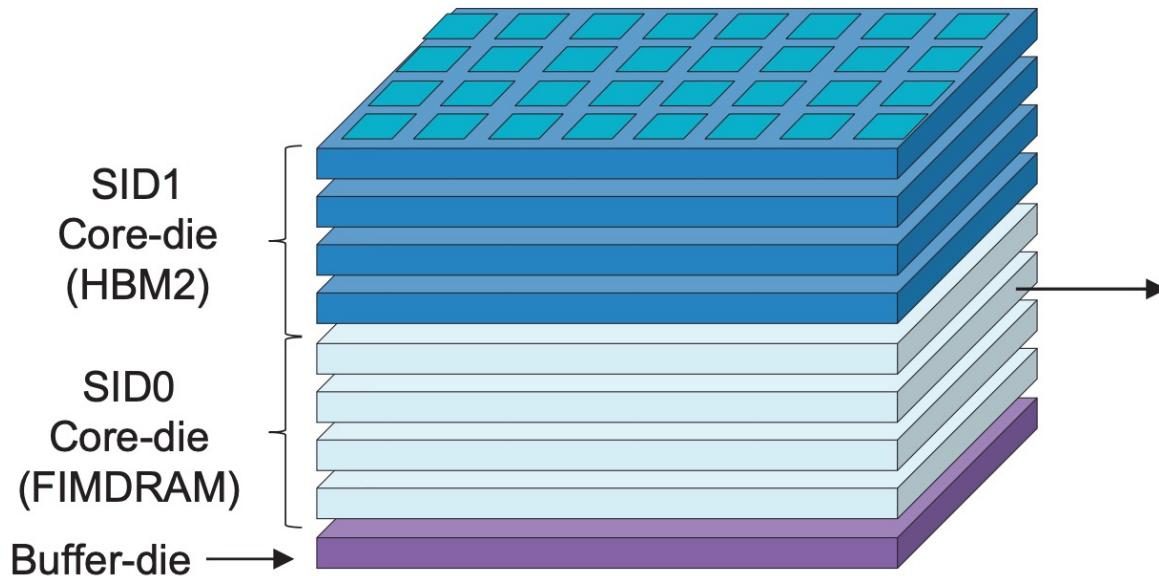
*The new architecture will deliver over twice the system performance and reduce energy consumption by more than 70%*

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

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

# Samsung Function-in-Memory DRAM (2021)

## ■ FIMDRAM based on HBM2



[3D Chip Structure of HBM with FIMDRAM]

### Chip Specification

128DQ / 8CH / 16 banks / BL4

32 PCU blocks (1 FIM block/2 banks)

1.2 TFLOPS (4H)

**FP16 ADD /  
Multiply (MUL) /  
Multiply-Accumulate (MAC) /  
Multiply-and- Add (MAD)**

ISSCC 2021 / SESSION 25 / DRAM / 25.4

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

Young-Cheon Kwon<sup>1</sup>, Suk Han Lee<sup>1</sup>, Jaehoon Lee<sup>1</sup>, Sang-Hyuk Kwon<sup>1</sup>, Je Min Ryu<sup>1</sup>, Jong-Pil Son<sup>1</sup>, Seongil Oh<sup>1</sup>, Hak-Soo Yu<sup>1</sup>, Haesuk Lee<sup>1</sup>, Soo Young Kim<sup>1</sup>, Younghmin Cho<sup>1</sup>, Jin Guk Kim<sup>1</sup>, Jongyoon Choi<sup>1</sup>, Hyun-Sung Shin<sup>1</sup>, Jin Kim<sup>1</sup>, BengSeng Phuah<sup>2</sup>, Hyo Young Kim<sup>1</sup>, Myeong Jun Song<sup>1</sup>, Ahn Choi<sup>1</sup>, Daeho Kim<sup>1</sup>, Soo Young Kim<sup>1</sup>, Eun-Bong Kim<sup>1</sup>, David Wang<sup>2</sup>, Shinhaeng Kang<sup>1</sup>, Yuhwan Ro<sup>3</sup>, Seungwoo Seo<sup>3</sup>, JoonHo Song<sup>3</sup>, Jaeyoun Youn<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

<sup>1</sup>Samsung Electronics, Hwaseong, Korea

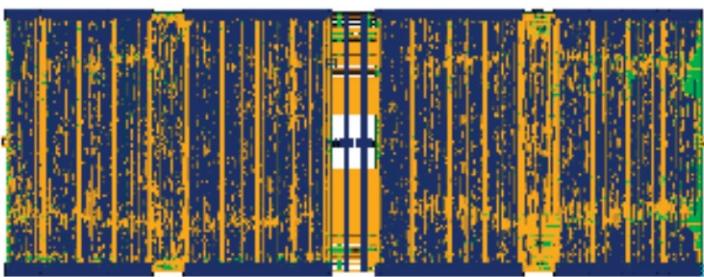
<sup>2</sup>Samsung Electronics, San Jose, CA

<sup>3</sup>Samsung Electronics, Suwon, Korea

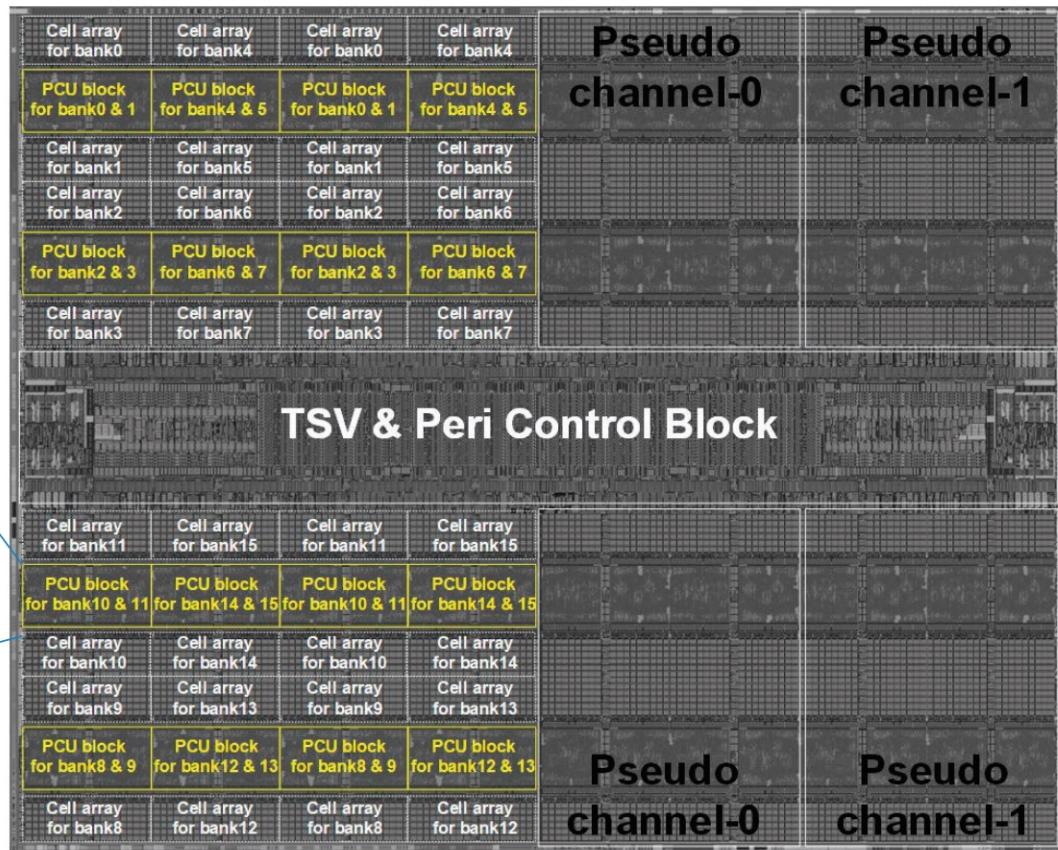
# Samsung Function-in-Memory DRAM (2021)

## Chip Implementation

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



[Digital RTL design for PCU block]



ISSCC 2021 / SESSION 25 / DRAM / 25.4

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

Young-Cheon Kwon<sup>1</sup>, Suk Han Lee<sup>1</sup>, Jaehoon Lee<sup>1</sup>, Sang-Hyuk Kwon<sup>1</sup>, Je Min Ryu<sup>1</sup>, Jong-Pil Son<sup>1</sup>, Seongil O<sup>1</sup>, Hak-Soo Yu<sup>1</sup>, Haesuk Lee<sup>1</sup>, Soo Young Kim<sup>1</sup>, Youngmin Cho<sup>1</sup>, Jin Guk Kim<sup>1</sup>, Jongyoon Choi<sup>1</sup>, Hyun-Sung Shin<sup>1</sup>, Jin Kim<sup>1</sup>, BengSeng Phua<sup>2</sup>, HyoungMin Kim<sup>1</sup>, Myeong Jun Song<sup>1</sup>, Ahn Choi<sup>1</sup>, Daeho Kim<sup>1</sup>, SooYoung Kim<sup>1</sup>, Eun-Bong Kim<sup>1</sup>, David Wang<sup>1</sup>, Shinhwang Kang<sup>1</sup>, Yuhwan Ro<sup>1</sup>, Seungwoo Seo<sup>1</sup>, JoonHo Song<sup>1</sup>, Jaeyoun Youn<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

<sup>1</sup>Samsung Electronics, Hwasung, Korea  
<sup>2</sup>Samsung Electronics, San Jose, CA  
<sup>3</sup>Samsung Electronics, Suwon, Korea

# 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](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](mailto: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

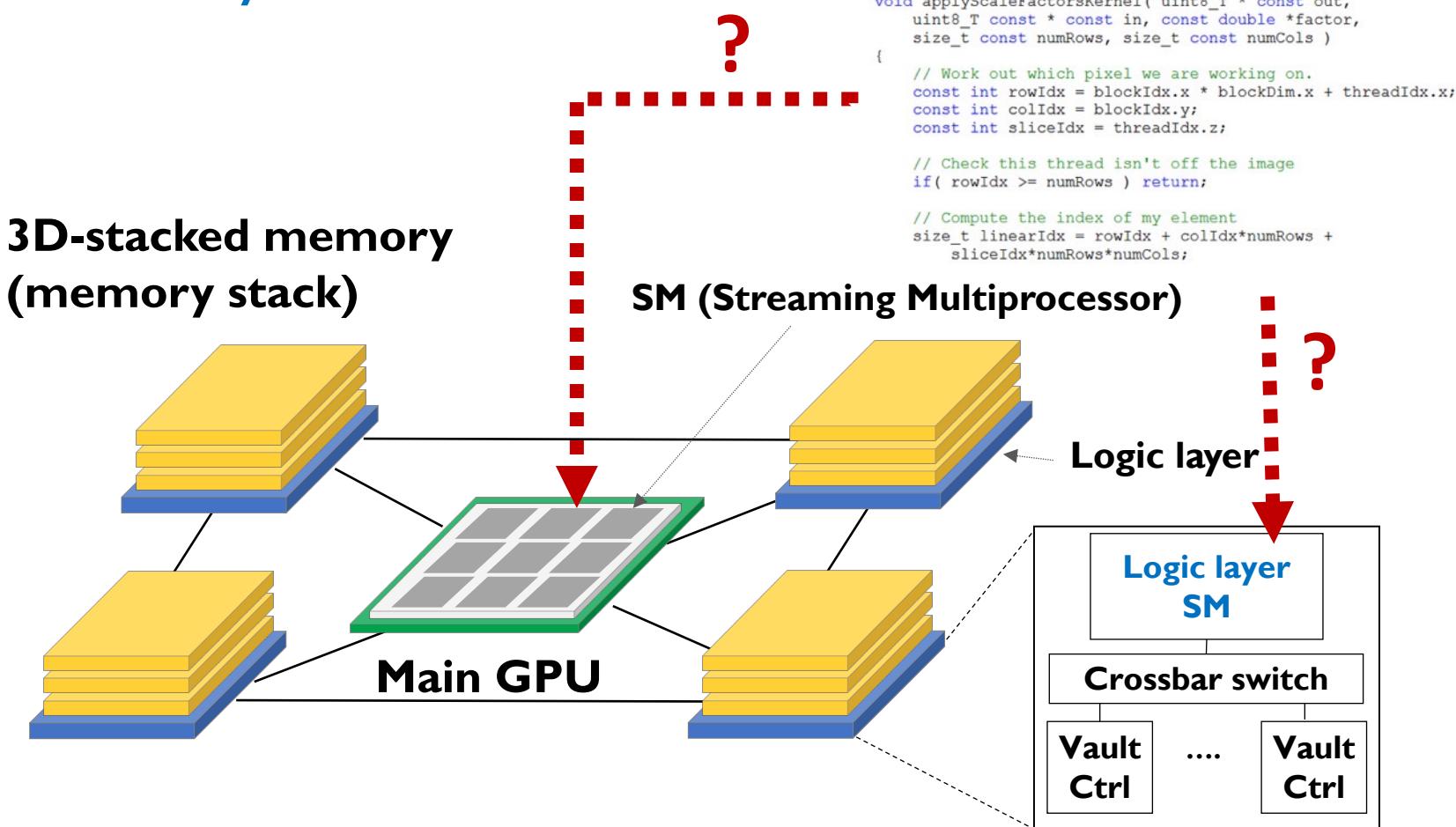
ANALYTICS

EDIT VIDEO

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

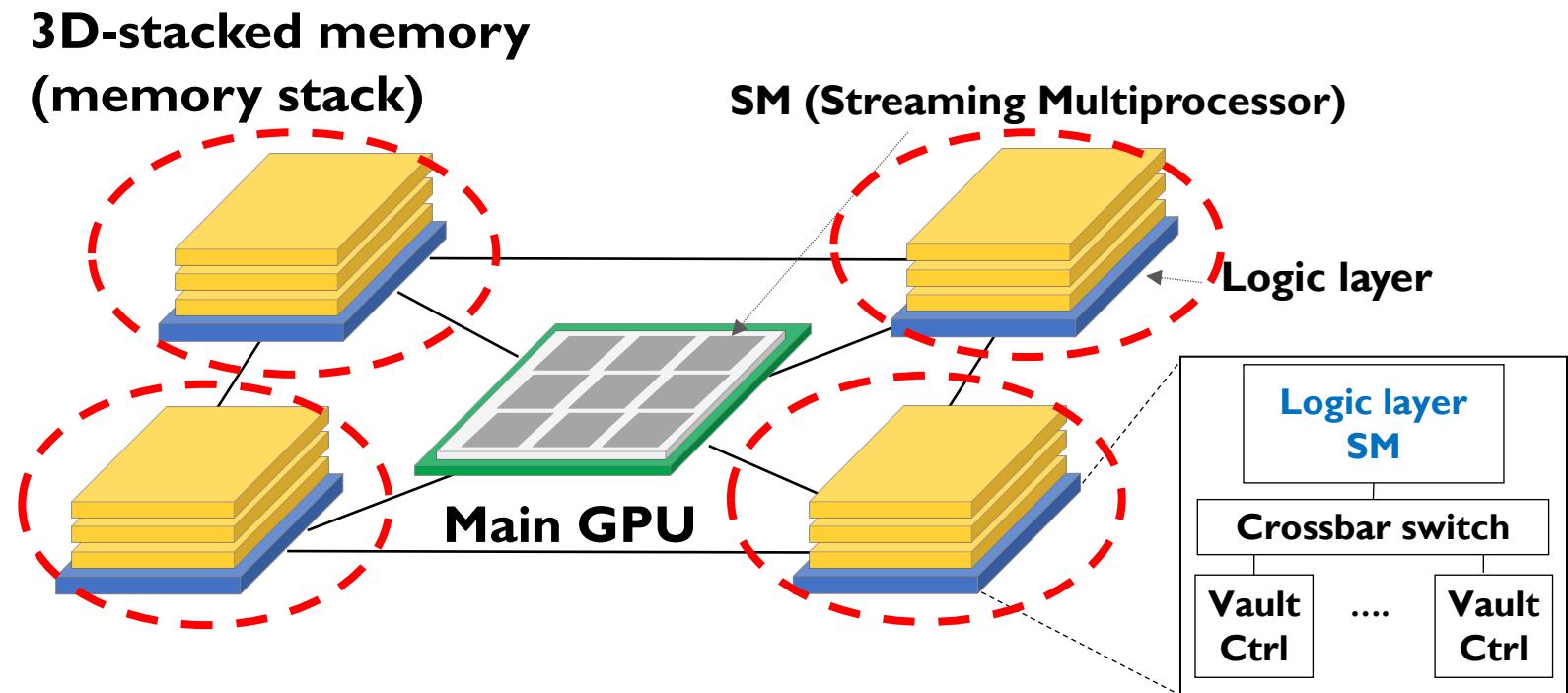
# Key Challenge 1: Code Mapping

- Challenge 1: Which operations should be executed in memory vs. in CPU?



# Key Challenge 2: Data Mapping

- **Challenge 2:** How should data be mapped to different 3D memory stacks?



# How to Do the Code and Data Mapping?

---

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler,  
**"Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems"**

*Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.*

[Slides (pptx) (pdf)]

[Lightning Session Slides (pptx) (pdf)]

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

Kevin Hsieh<sup>‡</sup> Eiman Ebrahimi<sup>†</sup> Gwangsun Kim<sup>\*</sup> Niladrish Chatterjee<sup>†</sup> Mike O'Connor<sup>†</sup>  
Nandita Vijaykumar<sup>‡</sup> Onur Mutlu<sup>§‡</sup> Stephen W. Keckler<sup>†</sup>

<sup>‡</sup>Carnegie Mellon University    <sup>†</sup>NVIDIA    <sup>\*</sup>KAIST    <sup>§</sup>ETH Zürich

# How to Schedule Code? (I)

---

- Ashutosh Pattnaik, Xulong Tang, Adwait Jog, Onur Kayiran, Asit K. Mishra, Mahmut T. Kandemir, Onur Mutlu, and Chita R. Das,  
**"Scheduling Techniques for GPU Architectures with Processing-In-Memory Capabilities"**

*Proceedings of the 25th International Conference on Parallel Architectures and Compilation Techniques (PACT), Haifa, Israel, September 2016.*

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

Ashutosh Pattnaik<sup>1</sup>    Xulong Tang<sup>1</sup>    Adwait Jog<sup>2</sup>    Onur Kayıran<sup>3</sup>  
Asit K. Mishra<sup>4</sup>    Mahmut T. Kandemir<sup>1</sup>    Onur Mutlu<sup>5,6</sup>    Chita R. Das<sup>1</sup>

<sup>1</sup>Pennsylvania State University    <sup>2</sup>College of William and Mary

<sup>3</sup>Advanced Micro Devices, Inc.    <sup>4</sup>Intel Labs    <sup>5</sup>ETH Zürich    <sup>6</sup>Carnegie Mellon University

# How to Schedule Code? (II)

---

- Milad Hashemi, Khubaib, Eiman Ebrahimi, Onur Mutlu, and Yale N. Patt,  
**"Accelerating Dependent Cache Misses with an Enhanced Memory Controller"**

*Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.*

[Slides (pptx) (pdf)]

[Lightning Session Slides (pptx) (pdf)]

## Accelerating Dependent Cache Misses with an Enhanced Memory Controller

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

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

# How to Schedule Code? (III)

---

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

## Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads

Milad Hashemi\*, Onur Mutlu<sup>§</sup>, Yale N. Patt\*

\**The University of Texas at Austin*    <sup>§</sup>*ETH Zürich*

# How to Maintain Coherence? (I)

---

- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,  
**"LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory"**  
*IEEE Computer Architecture Letters (CAL)*, June 2016.

## **LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory**

Amirali Boroumand<sup>†</sup>, Saugata Ghose<sup>†</sup>, Minesh Patel<sup>†</sup>, Hasan Hassan<sup>†§</sup>, Brandon Lucia<sup>†</sup>,  
Kevin Hsieh<sup>†</sup>, Krishna T. Malladi<sup>\*</sup>, Hongzhong Zheng<sup>\*</sup>, and Onur Mutlu<sup>‡†</sup>

<sup>†</sup>*Carnegie Mellon University*   <sup>\*</sup>*Samsung Semiconductor, Inc.*   <sup>§</sup>*TOBB ETÜ*   <sup>‡</sup>*ETH Zürich*

# How to Maintain Coherence? (II)

---

- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,  
**"CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators"**

*Proceedings of the 46th International Symposium on Computer Architecture (ISCA), Phoenix, AZ, USA, June 2019.*

## **CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators**

Amirali Boroumand<sup>†</sup>

Brandon Lucia<sup>†</sup>

Nastaran Hajinazar<sup>◦†</sup>

Saugata Ghose<sup>†</sup>

Rachata Ausavarungnirun<sup>†‡</sup>

Krishna T. Malladi<sup>§</sup>

Minesh Patel<sup>\*</sup>

Kevin Hsieh<sup>†</sup>

Hongzhong Zheng<sup>§</sup>

Hasan Hassan<sup>\*</sup>

Onur Mutlu<sup>★†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>◦</sup>Simon Fraser University

<sup>\*</sup>ETH Zürich

<sup>‡</sup>KMUTNB

<sup>§</sup>Samsung Semiconductor, Inc.

# How to Support Synchronization?

---

- Christina Giannoula, Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas, Ivan Fernandez, Juan Gómez-Luna, Lois Orosa, Nectarios Koziris, Georgios Goumas, Onur Mutlu,

## **"SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures"**

*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 (21 minutes)]

[Short Talk Video (7 minutes)]

## ***SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures***

Christina Giannoula<sup>†‡</sup> Nandita Vijaykumar<sup>\*‡</sup> Nikela Papadopoulou<sup>†</sup> Vasileios Karakostas<sup>†</sup> Ivan Fernandez<sup>§‡</sup>  
Juan Gómez-Luna<sup>‡</sup> Lois Orosa<sup>‡</sup> Nectarios Koziris<sup>†</sup> Georgios Goumas<sup>†</sup> Onur Mutlu<sup>‡</sup>

<sup>†</sup>*National Technical University of Athens*    <sup>‡</sup>*ETH Zürich*    <sup>\*</sup>*University of Toronto*    <sup>§</sup>*University of Malaga*

# How to Support Virtual Memory?

---

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

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

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

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

Kevin Hsieh<sup>†</sup>   Samira Khan<sup>‡</sup>   Nandita Vijaykumar<sup>†</sup>  
Kevin K. Chang<sup>†</sup>   Amirali Boroumand<sup>†</sup>   Saugata Ghose<sup>†</sup>   Onur Mutlu<sup>§†</sup>  
<sup>†</sup>*Carnegie Mellon University*   <sup>‡</sup>*University of Virginia*   <sup>§</sup>*ETH Zürich*

# How to Design Data Structures for PIM?

---

- Zhiyu Liu, Irina Calciu, Maurice Herlihy, and Onur Mutlu,  
**"Concurrent Data Structures for Near-Memory Computing"**  
*Proceedings of the 29th ACM Symposium on Parallelism in Algorithms  
and Architectures (SPAA)*, Washington, DC, USA, July 2017.  
[[Slides \(pptx\)](#) [\(pdf\)](#)]

## Concurrent Data Structures for Near-Memory Computing

Zhiyu Liu

Computer Science Department

Brown University

[zhiyu\\_liu@brown.edu](mailto:zhiyu_liu@brown.edu)

Maurice Herlihy

Computer Science Department

Brown University

[mph@cs.brown.edu](mailto:mph@cs.brown.edu)

Irina Calciu

VMware Research Group

[icalciu@vmware.com](mailto:icalciu@vmware.com)

Onur Mutlu

Computer Science Department

ETH Zürich

[onur.mutlu@inf.ethz.ch](mailto:onur.mutlu@inf.ethz.ch)

# Simulation Infrastructures for PIM

---

- Ramulator extended for PIM
  - Flexible and extensible DRAM simulator
  - Can model many different memory standards and proposals
  - Kim+, "**Ramulator: A Fast and Extensible DRAM Simulator**", IEEE CAL 2015.
  - <https://github.com/CMU-SAFARI/ramulator-pim>
  - <https://github.com/CMU-SAFARI/ramulator>
  - [\[Source Code for Ramulator-PIM\]](#)

## Ramulator: A Fast and Extensible DRAM Simulator

Yoongu Kim<sup>1</sup>    Weikun Yang<sup>1,2</sup>    Onur Mutlu<sup>1</sup>

<sup>1</sup>Carnegie Mellon University    <sup>2</sup>Peking University

# Performance & Energy Models for PIM

---

- Gagandeep Singh, Juan Gomez-Luna, Giovanni Mariani, Geraldo F. Oliveira, Stefano Corda, Sander Stuijk, Onur Mutlu, and Henk Corporaal,  
**"NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning"**

*Proceedings of the 56th Design Automation Conference (DAC), Las Vegas, NV, USA, June 2019.*

[Slides (pptx) (pdf)]

[Poster (pptx) (pdf)]

[Source Code for Ramulator-PIM]

## **NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning**

Gagandeep Singh<sup>a,c</sup>  
Stefano Corda<sup>a,c</sup>

<sup>a</sup>Eindhoven University of Technology

Juan Gómez-Luna<sup>b</sup>  
Sander Stuijk<sup>a</sup>

Giovanni Mariani<sup>c</sup>  
Onur Mutlu<sup>b</sup>  
<sup>b</sup>ETH Zürich

Geraldo F. Oliveira<sup>b</sup>  
Henk Corporaal<sup>a</sup>  
<sup>c</sup>IBM Research - Zurich

# DAMOV Methodology & Workloads [SIGMETRICS'21]

---

## 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>.

Challenge and Opportunity for Future

---

# Computing Architectures with Minimal Data Movement

# Challenge and Opportunity for Future

---

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

Computing Architectures

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

# Computing Architectures

# 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** decisions
- 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**

# Exploiting Data to Design Intelligent Architectures

# System Architecture Design Today

---

- Human-driven
  - Humans design the policies (how to do things)
- Many (too) simple, short-sighted policies all over the system
- No automatic data-driven policy learning
- (Almost) no learning: cannot take lessons from past actions

**Can we design  
fundamentally intelligent architectures?**

# An Intelligent Architecture

---

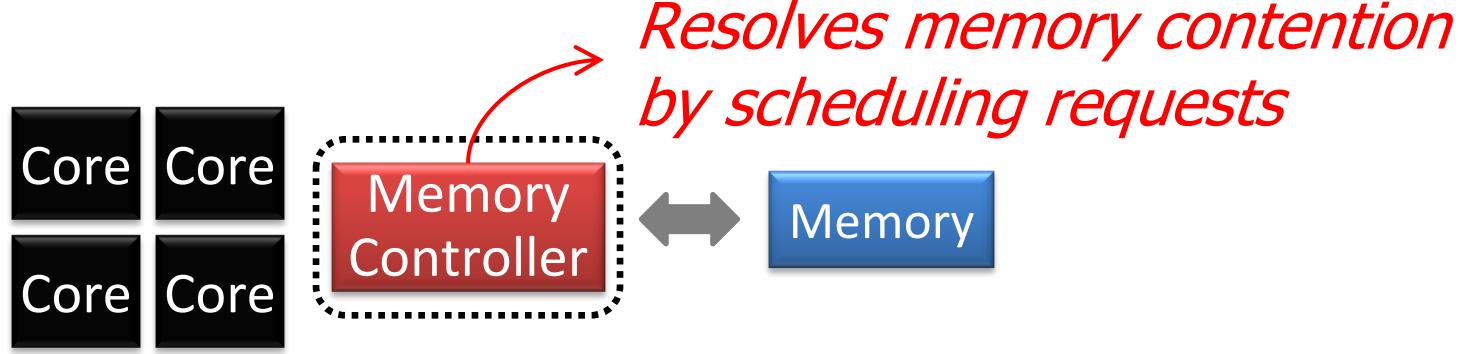
- Data-driven
  - Machine learns the “best” policies (how to do things)
- Sophisticated, workload-driven, changing, far-sighted policies
- Automatic data-driven policy learning
- All controllers are intelligent data-driven agents

**How do we start?**

# Self-Optimizing Memory Controllers

# Memory Controller

---



How to schedule requests to maximize system performance?

# Why are Memory Controllers Difficult to Design?

---

- Need to obey **DRAM timing constraints** for correctness
  - There are many (50+) timing constraints in DRAM
  - tWTR: Minimum number of cycles to wait before issuing a read command after a write command is issued
  - tRC: Minimum number of cycles between the issuing of two consecutive activate commands to the same bank
  - ...
- Need to **keep track of many resources** to prevent conflicts
  - Channels, banks, ranks, data bus, address bus, row buffers, ...
- Need to handle **DRAM refresh**
- Need to **manage power consumption**
- Need to **optimize performance & QoS** (in the presence of constraints)
  - Reordering is not simple
  - Fairness and QoS needs complicates the scheduling problem
- ...

# Many Memory Timing Constraints

---

Latency	Symbol	DRAM cycles	Latency	Symbol	DRAM cycles
Precharge	$t_{RP}$	11	Activate to read/write	$t_{RCD}$	11
Read column address strobe	$CL$	11	Write column address strobe	$CWL$	8
Additive	$AL$	0	Activate to activate	$t_{RC}$	39
Activate to precharge	$t_{RAS}$	28	Read to precharge	$t_{RTP}$	6
Burst length	$t_{BL}$	4	Column address strobe to column address strobe	$t_{CCD}$	4
Activate to activate (different bank)	$t_{RRD}$	6	Four activate windows	$t_{FAW}$	24
Write to read	$t_{WTR}$	6	Write recovery	$t_{WR}$	12

Table 4. DDR3 1600 DRAM timing specifications

- From Lee et al., “DRAM-Aware Last-Level Cache Writeback: Reducing Write-Caused Interference in Memory Systems,” HPS Technical Report, April 2010.

# Many Memory Timing Constraints

- Kim et al., “A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM,” ISCA 2012.
- Lee et al., “Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture,” HPCA 2013.

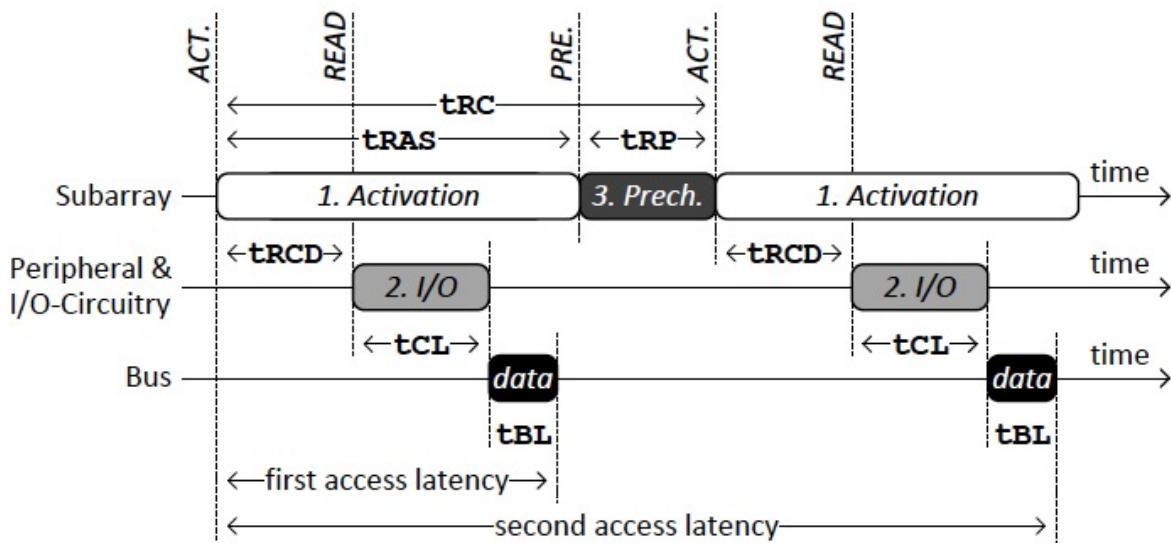
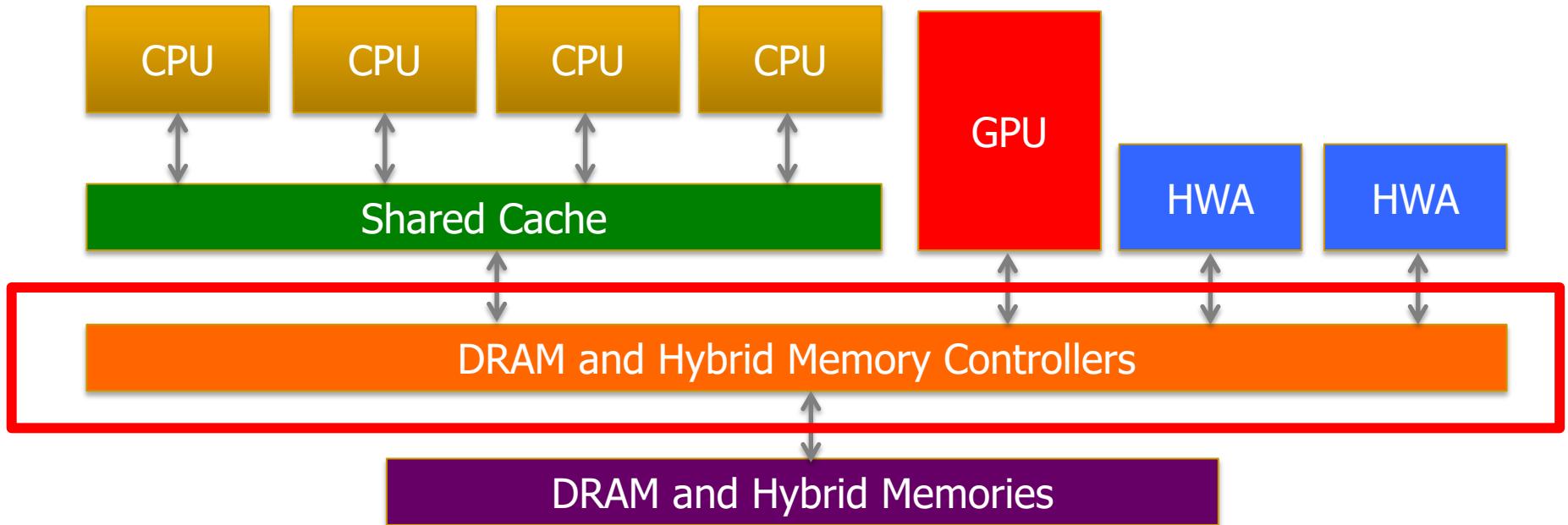


Figure 5. Three Phases of DRAM Access

Table 2. Timing Constraints (DDR3-1066) [43]

Phase	Commands	Name	Value
1	ACT → READ	tRCD	15ns
	ACT → WRITE	tRAS	37.5ns
2	ACT → PRE	tCL	15ns
	READ → data	tCWL	11.25ns
	data burst	tBL	7.5ns
3	PRE → ACT	tRP	15ns
1 & 3	ACT → ACT	tRC (tRAS+tRP)	52.5ns

# Memory Controller Design Is Becoming More Difficult



- Heterogeneous agents: CPUs, GPUs, and HWAs
- Main memory interference between CPUs, GPUs, HWAs
- Many timing constraints for various memory types
- Many goals at the same time: performance, fairness, QoS, energy efficiency, ...

# Reality and Dream

---

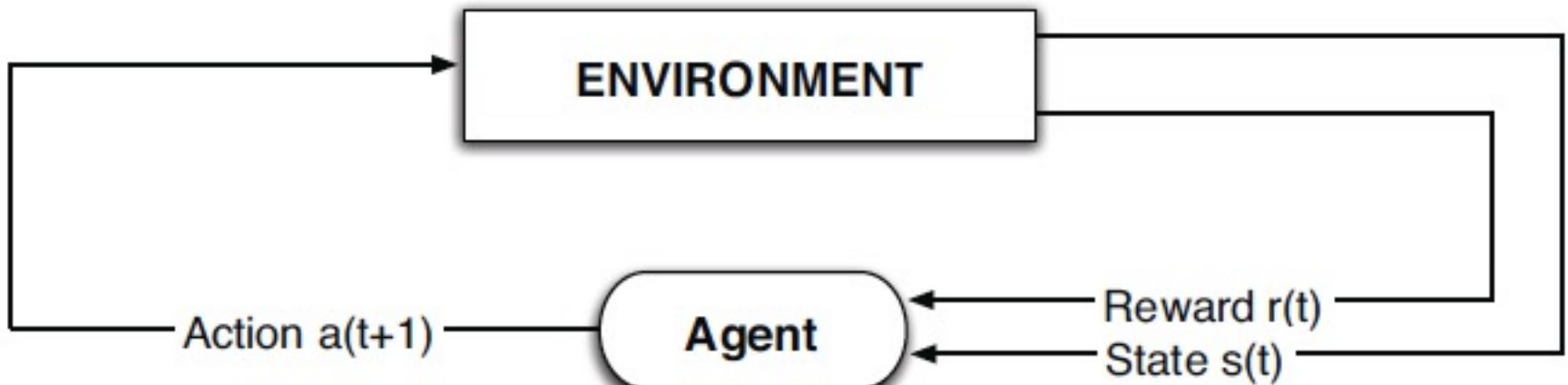
- Reality: It difficult to design a policy that maximizes performance, QoS, energy-efficiency, ...
  - Too many things to think about
  - Continuously changing workload and system behavior
- Dream: Wouldn't it be nice if the DRAM controller automatically found a good scheduling policy on its own?

# Self-Optimizing DRAM Controllers

---

- Problem: DRAM controllers are difficult to design
  - It is difficult for human designers to design a policy that can adapt itself very well to different workloads and different system conditions
- Idea: A memory controller that adapts its scheduling policy to workload behavior and system conditions using machine learning.
- Observation: Reinforcement learning maps nicely to memory control.
- Design: Memory controller is a reinforcement learning agent
  - It dynamically and continuously learns and employs the best scheduling policy to maximize long-term performance.

# Self-Optimizing DRAM Controllers

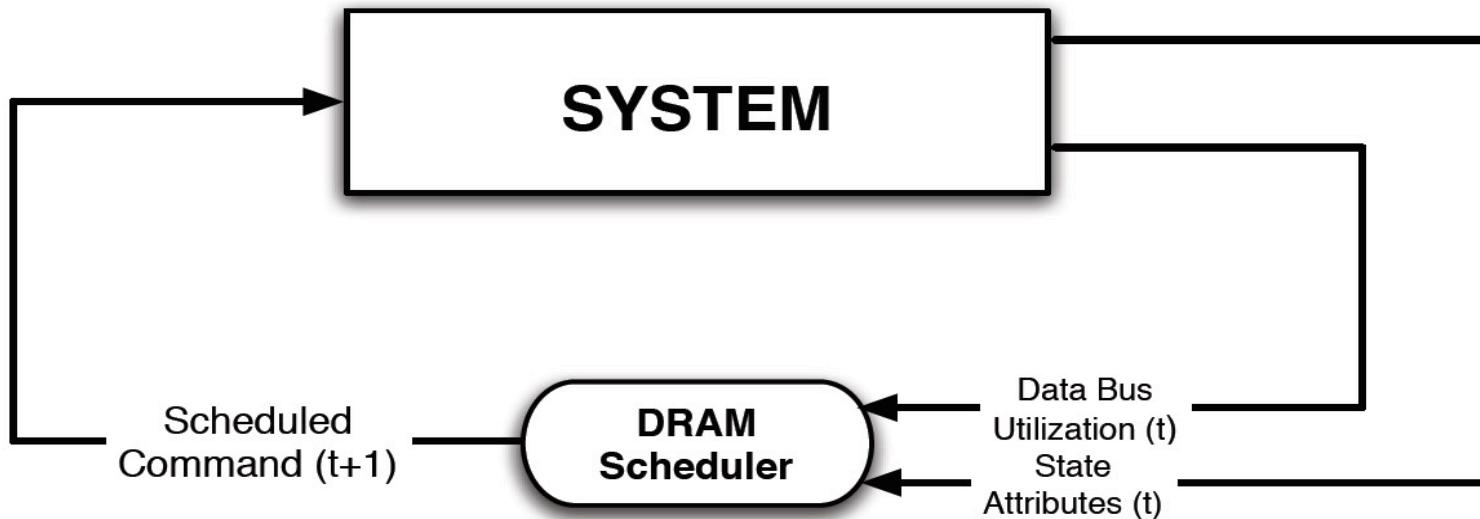


Goal: Learn to choose actions to maximize  $r_0 + \gamma r_1 + \gamma^2 r_2 + \dots$  ( $0 \leq \gamma < 1$ )

**Figure 2:** (a) Intelligent agent based on reinforcement learning principles;

# Self-Optimizing DRAM Controllers

- Dynamically adapt the memory scheduling policy via interaction with the system at runtime
  - Associate system states and actions (commands) with long term reward values: each action at a given state leads to a learned reward
  - Schedule command with highest estimated long-term reward value in each state
  - Continuously update reward values for <state, action> pairs based on feedback from system



# Self-Optimizing DRAM Controllers

- Engin Ipek, Onur Mutlu, José F. Martínez, and Rich Caruana,  
**"Self Optimizing Memory Controllers: A Reinforcement Learning Approach"**

*Proceedings of the 35th International Symposium on Computer Architecture (ISCA)*, pages 39-50, Beijing, China, June 2008.

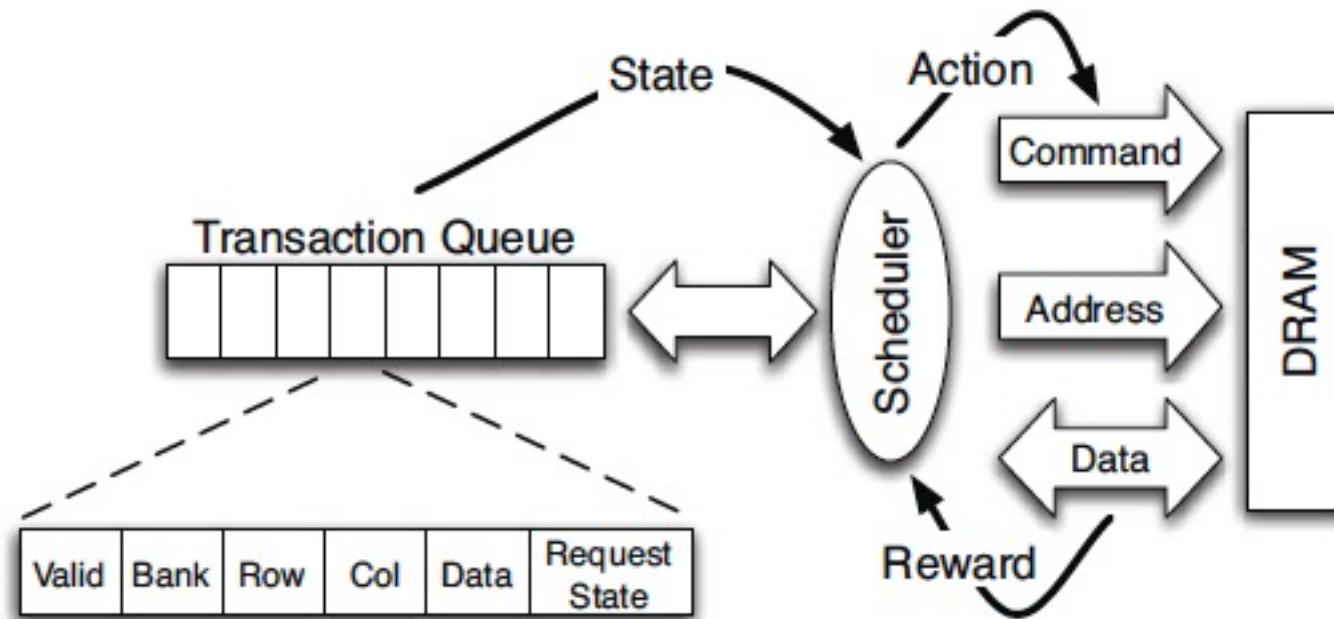


Figure 4: High-level overview of an RL-based scheduler.

# States, Actions, Rewards

---

## ❖ Reward function

- +1 for scheduling Read and Write commands
- 0 at all other times

Goal is to maximize long-term data bus utilization

## ❖ State attributes

- Number of reads, writes, and load misses in transaction queue
- Number of pending writes and ROB heads waiting for referenced row
- Request's relative ROB order

## ❖ Actions

- Activate
- Write
- Read - load miss
- Read - store miss
- Precharge - pending
- Precharge - preemptive
- NOP

# Performance Results

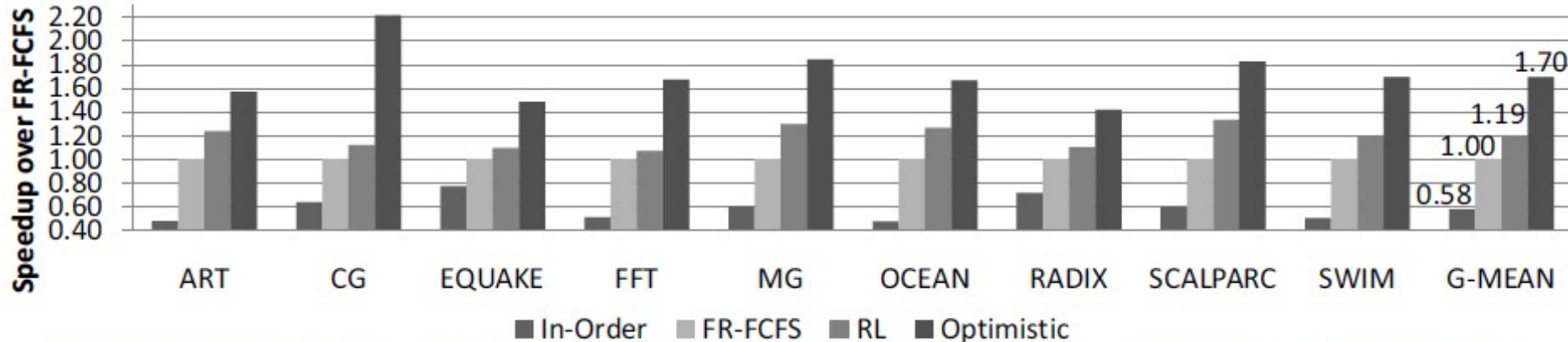


Figure 7: Performance comparison of in-order, FR-FCFS, RL-based, and optimistic memory controllers

**Large, robust performance improvements over many human-designed policies**

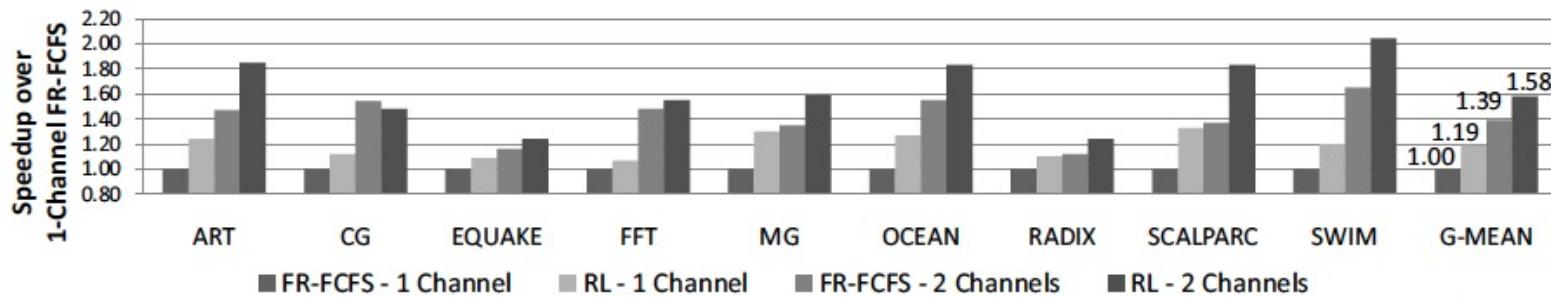


Figure 15: Performance comparison of FR-FCFS and RL-based memory controllers on systems with 6.4GB/s and 12.8GB/s peak DRAM bandwidth

# Self Optimizing DRAM Controllers

---

- + Continuous learning in the presence of changing environment
  - + Reduced designer burden in finding a good scheduling policy.  
Designer specifies:
    - 1) What system variables might be useful
    - 2) What target to optimize, but not how to optimize it
  - How to specify different objectives? (e.g., fairness, QoS, ...)
  - Hardware complexity?
  - Design mindset and flow
-

# More on Self-Optimizing DRAM Controllers

---

- Engin Ipek, Onur Mutlu, José F. Martínez, and Rich Caruana,  
**"[Self Optimizing Memory Controllers: A Reinforcement Learning Approach](#)"**

*Proceedings of the [35th International Symposium on Computer Architecture \(ISCA\)](#)*, pages 39-50, Beijing, China, June 2008.

## Self-Optimizing Memory Controllers: A Reinforcement Learning Approach

**Engin İpek**<sup>1,2</sup>    **Onur Mutlu**<sup>2</sup>    **José F. Martínez**<sup>1</sup>    **Rich Caruana**<sup>1</sup>

<sup>1</sup>Cornell University, Ithaca, NY 14850 USA

<sup>2</sup> Microsoft Research, Redmond, WA 98052 USA

# An Intelligent Architecture

---

- Data-driven
  - Machine learns the “best” policies (how to do things)
- Sophisticated, workload-driven, changing, far-sighted policies
- Automatic data-driven policy learning
- All controllers are intelligent data-driven agents

**We need to rethink design  
(of all controllers)**

# Data-Driven (Self-Optimizing) Computing Architectures

# 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** decisions
- 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-Aware Architectures

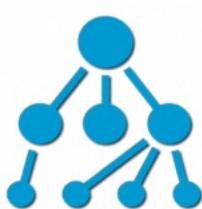
---

- A data-aware architecture **understands what it can do with and to each piece of data**
- It makes use of different properties of data to improve performance, efficiency and other metrics
  - Compressibility
  - Approximability
  - Locality
  - Sparsity
  - Criticality for Computation X
  - Access Semantics
  - ...

# One Problem: Limited Expressiveness

---

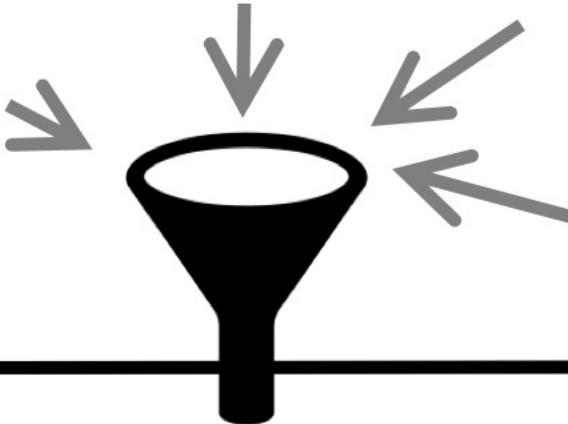
**Higher-level information is not visible to HW**



*Software*



**Code Optimizations**

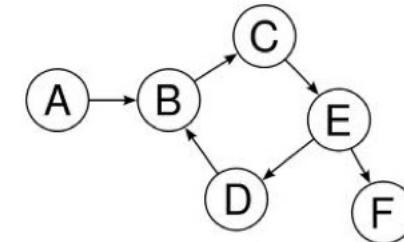


100011111...

101010011...

**Instructions**

**Memory Addresses**



**Access Patterns**

**Integer**

**Float**

**Data Type**

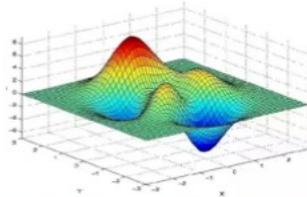
**Char**

*Hardware*

# A Solution: More Expressive Interfaces

Performance

Software



Functionality

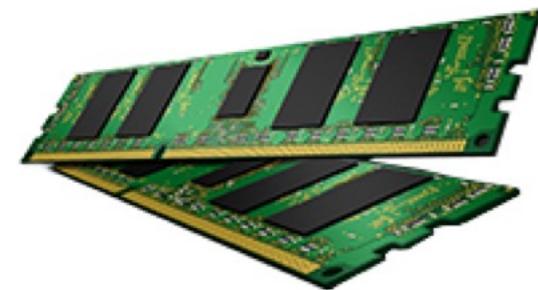
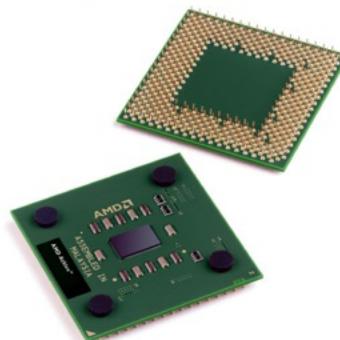


Higher-level  
Program  
Semantics

ISA  
Virtual Memory

Expressive  
Memory  
“XMem”

Hardware



# Expressive (Memory) Interfaces

---

- Nandita Vijaykumar, Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko, Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons and Onur Mutlu,  
**"A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory"**

*Proceedings of the 45th International Symposium on Computer Architecture (ISCA)*,  
Los Angeles, CA, USA, June 2018.

[[Slides \(pptx\)](#) ([pdf](#))] [[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]  
[[Lightning Talk Video](#)]

## A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory

Nandita Vijaykumar<sup>†§</sup> Abhilasha Jain<sup>†</sup> Diptesh Majumdar<sup>†</sup> Kevin Hsieh<sup>†</sup> Gennady Pekhimenko<sup>‡</sup>  
Eiman Ebrahimi<sup>¶</sup> Nastaran Hajinazar<sup>†</sup> Phillip B. Gibbons<sup>†</sup> Onur Mutlu<sup>§†</sup>

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

<sup>+</sup>**Simon Fraser University**

<sup>‡</sup>**University of Toronto**

<sup>§</sup>**ETH Zürich**

<sup>¶</sup>**NVIDIA**

# X-MeM Aids Many Optimizations

**Table 1: Summary of the example memory optimizations that XMem aids.**

Memory optimization	Example semantics provided by XMem (described in §3.3)	Example Benefits of XMem
Cache management	(i) Distinguishing between data structures or pools of similar data; (ii) Working set size; (iii) Data reuse	Enables: (i) applying different caching policies to different data structures or pools of data; (ii) avoiding cache thrashing by <i>knowing</i> the active working set size; (iii) bypassing/prioritizing data that has no/high reuse. (§5)
Page placement in DRAM e.g., [23, 24]	(i) Distinguishing between data structures; (ii) Access pattern; (iii) Access intensity	Enables page placement at the <i>data structure</i> granularity to (i) isolate data structures that have high row buffer locality and (ii) spread out concurrently-accessed irregular data structures across banks and channels to improve parallelism. (§6)
Cache/memory compression e.g., [25–32]	(i) Data type: integer, float, char; (ii) Data properties: sparse, pointer, data index	Enables using a <i>different compression algorithm</i> for each data structure based on data type and data properties, e.g., sparse data encodings, FP-specific compression, delta-based compression for pointers [27].
Data prefetching e.g., [33–36]	(i) Access pattern: strided, irregular, irregular but repeated (e.g., graphs), access stride; (ii) Data type: index, pointer	Enables (i) <i>highly accurate</i> software-driven prefetching while leveraging the benefits of hardware prefetching (e.g., by being memory bandwidth-aware, avoiding cache thrashing); (ii) using different prefetcher <i>types</i> for different data structures: e.g., stride [33], tile-based [20], pattern-based [34–37], data-based for indices/pointers [38, 39], etc.
DRAM cache management e.g., [40–46]	(i) Access intensity; (ii) Data reuse; (iii) Working set size	(i) Helps avoid cache thrashing by knowing working set size [44]; (ii) Better DRAM cache management via reuse behavior and access intensity information.
Approximation in memory e.g., [47–53]	(i) Distinguishing between pools of similar data; (ii) Data properties: tolerance towards approximation	Enables (i) each memory component to track how approximable data is (at a fine granularity) to inform approximation techniques; (ii) data placement in heterogeneous reliability memories [54].
Data placement: NUMA systems e.g., [55, 56]	(i) Data partitioning across threads (i.e., relating data to threads that access it); (ii) Read-Write properties	Reduces the need for profiling or data migration (i) to co-locate data with threads that access it and (ii) to identify Read-Only data, thereby enabling techniques such as replication.
Data placement: hybrid memories e.g., [16, 57, 58]	(i) Read-Write properties (Read-Only/Read-Write); (ii) Access intensity; (iii) Data structure size; (iv) Access pattern	Avoids the need for profiling/migration of data in hybrid memories to (i) effectively manage the asymmetric read-write properties in NVM (e.g., placing Read-Only data in the NVM) [16, 57]; (ii) make tradeoffs between data structure "hotness" and size to allocate fast/high bandwidth memory [14]; and (iii) leverage row-buffer locality in placement based on access pattern [45].
Managing NUCA systems e.g., [15, 59]	(i) Distinguishing pools of similar data; (ii) Access intensity; (iii) Read-Write or Private-Shared properties	(i) Enables using different cache policies for different data pools (similar to [15]); (ii) Reduces the need for reactive mechanisms that detect sharing and read-write characteristics to inform cache policies.

# Expressive (Memory) Interfaces for GPUs

---

- Nandita Vijaykumar, Eiman Ebrahimi, Kevin Hsieh, Phillip B. Gibbons and Onur Mutlu,  
**"The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs"**

*Proceedings of the 45th International Symposium on Computer Architecture (ISCA)*,  
Los Angeles, CA, USA, June 2018.

[[Slides \(pptx\)](#) ([pdf](#))] [[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]  
[[Lightning Talk Video](#)]

## The Locality Descriptor:

### A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs

Nandita Vijaykumar<sup>†§</sup>      Eiman Ebrahimi<sup>‡</sup>      Kevin Hsieh<sup>†</sup>

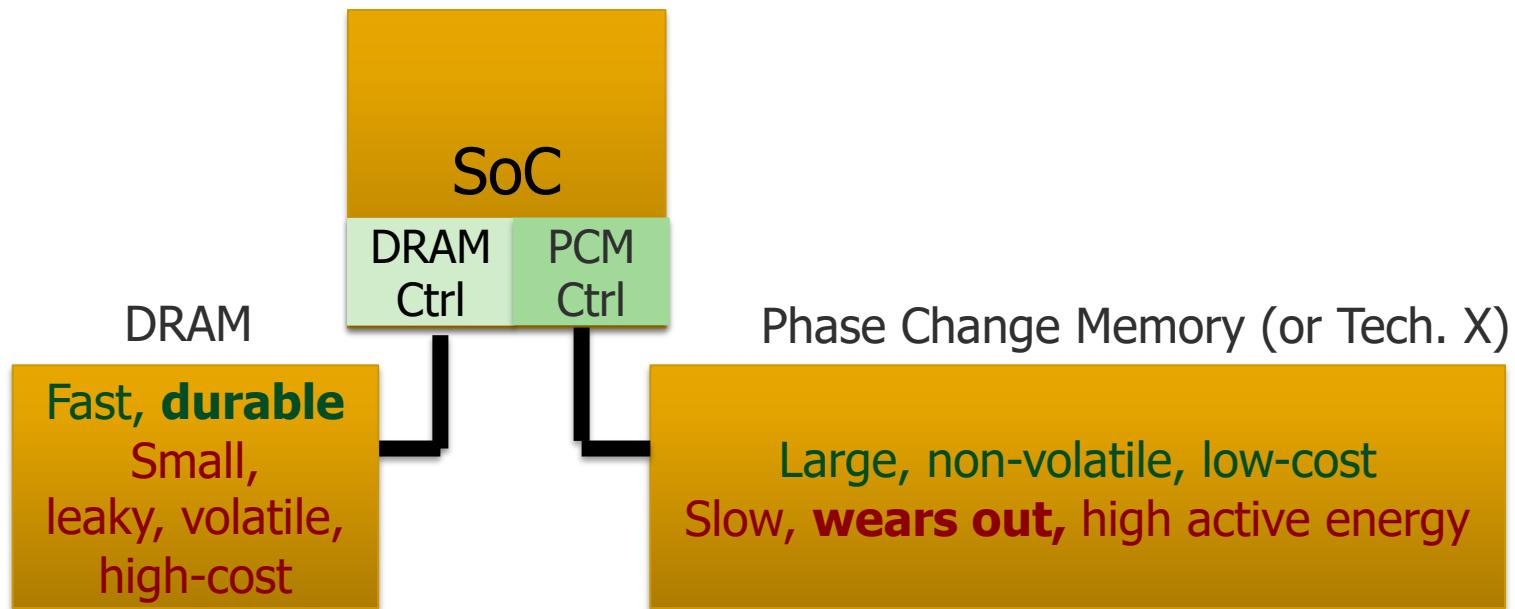
Phillip B. Gibbons<sup>†</sup>      Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>NVIDIA

<sup>§</sup>ETH Zürich

# An Example: Hybrid Memory Management



Hardware/software manage data allocation and movement  
to achieve the best of multiple technologies

Meza+, “[Enabling Efficient and Scalable Hybrid Memories](#),” IEEE Comp. Arch. Letters, 2012.

Yoon+, “[Row Buffer Locality Aware Caching Policies for Hybrid Memories](#),” ICCD 2012 Best Paper Award.

# An Example: Heterogeneous-Reliability Memory

---

- Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, and Onur Mutlu,  
**["Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory"](#)**  
*Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Atlanta, GA, June 2014. [Summary] [Slides (pptx) (pdf)] [Coverage on ZDNet]

## **Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory**

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

Carnegie Mellon University, [yixinluo@cs.cmu.edu](mailto:yixinluo@cs.cmu.edu), [{meza, onur}@cmu.edu](mailto:{meza, onur}@cmu.edu)

\*Microsoft Corporation, {srgovin, bsharma, marksan, kansal, jie.liu, bkhessib, kvaid}@microsoft.com

# Exploiting Memory Error Tolerance with Hybrid Memory Systems

Vulnerable  
data

Tolerant  
data

Reliable memory

Low-cost memory

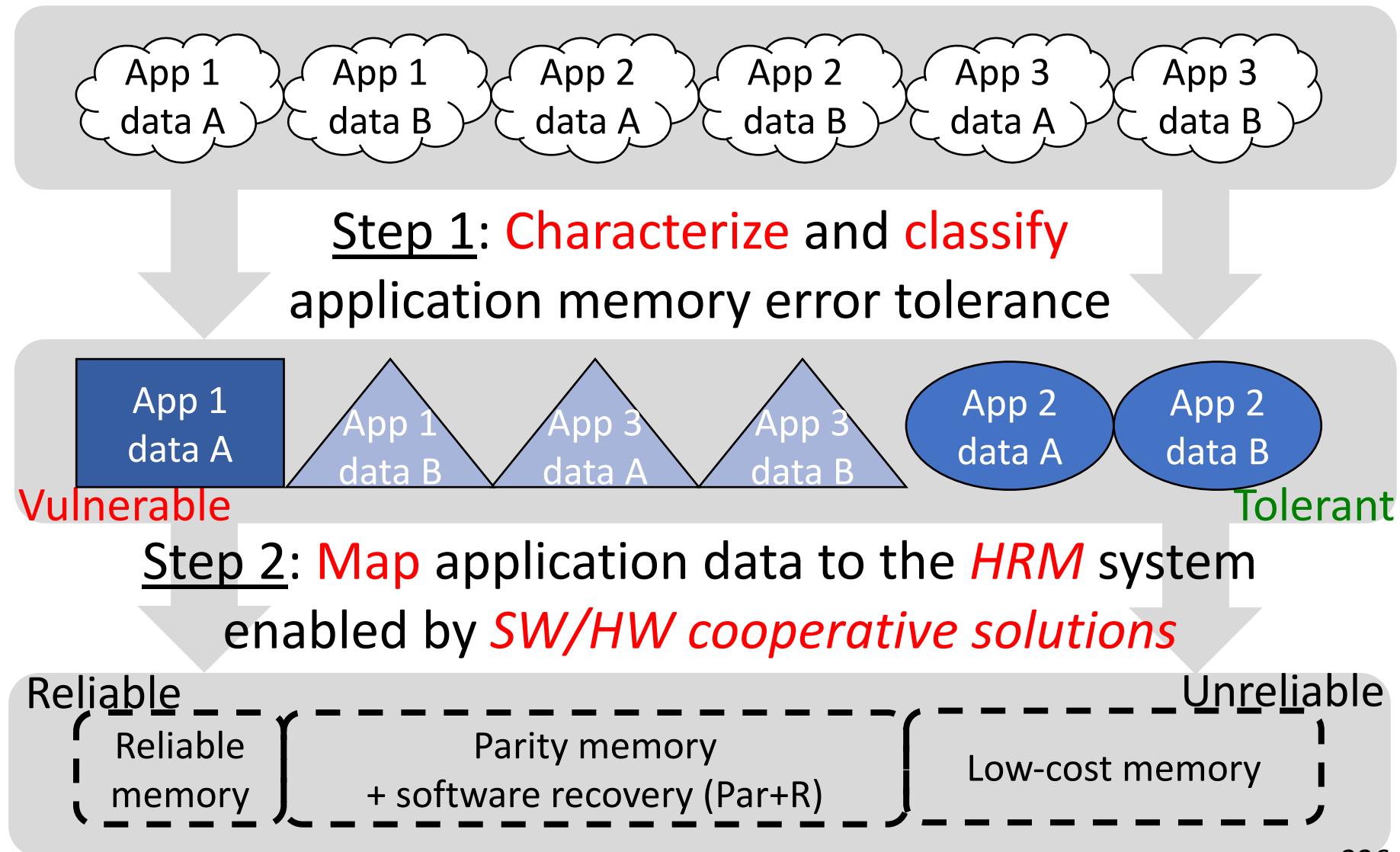
On Microsoft's Web Search workload

Reduces server hardware **cost** by **4.7 %**

Achieves single server **availability** target of **99.90 %**

**Heterogeneous-Reliability Memory [DSN 2014]**

# Heterogeneous-Reliability Memory



# More on Heterogeneous-Reliability Memory

---

- Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, and Onur Mutlu,  
**["Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory"](#)**  
*Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Atlanta, GA, June 2014. [[Summary](#)] [[Slides \(pptx\)](#)] [[pdf](#)] [[Coverage on ZDNet](#)]

## Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory

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

Carnegie Mellon University, [yixinluo@cs.cmu.edu](mailto:yixinluo@cs.cmu.edu), [{meza, onur}@cmu.edu](mailto:{meza, onur}@cmu.edu)

\*Microsoft Corporation, [{srgovin, bsharma, marksan, kansal, jie.liu, bkhessib, kvaid}@microsoft.com](mailto:{srgovin, bsharma, marksan, kansal, jie.liu, bkhessib, kvaid}@microsoft.com)

# Another Example: EDEN for DNNs

---

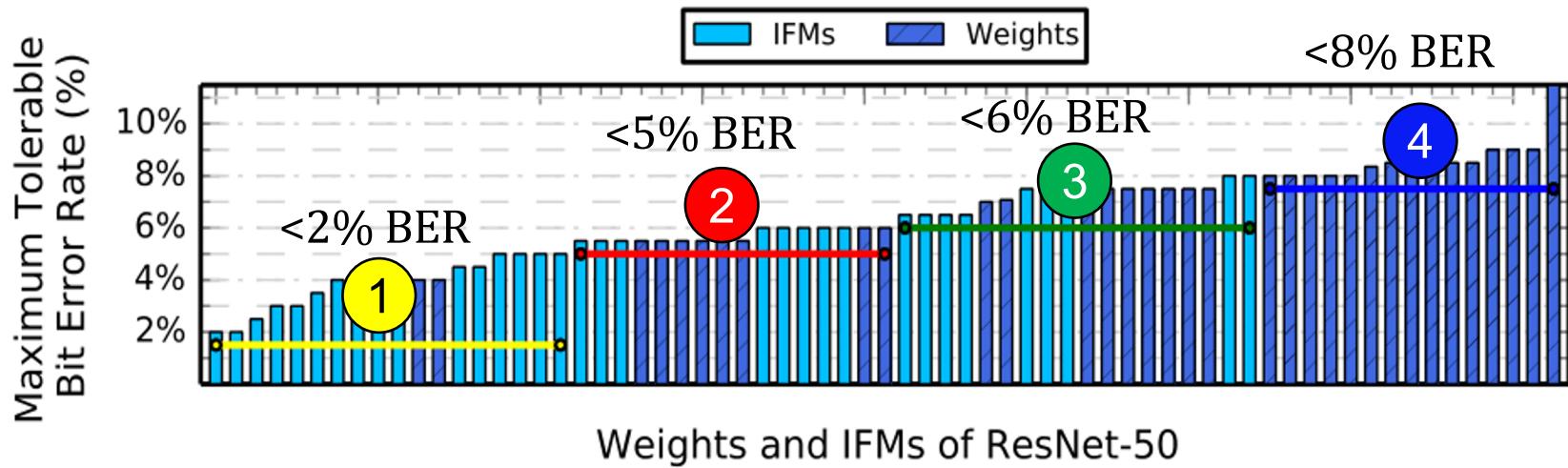
- Deep Neural Network evaluation is very DRAM-intensive (especially for large networks)
  1. Some data and layers in DNNs are very tolerant to errors
  2. Reduce DRAM latency and voltage on such data and layers
  3. While still achieving a user-specified DNN accuracy target by making training DRAM-error-aware

**Data-aware management of DRAM latency and voltage  
for Deep Neural Network Inference**

---

# Example DNN Data Type to DRAM Mapping

Mapping example of ResNet-50:



Map more error-tolerant DNN layers  
to DRAM partitions with lower voltage/latency

4 DRAM partitions with different error rates

# EDEN: Data-Aware Efficient DNN Inference

---

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

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

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

[Lightning Talk Slides (pptx) (pdf)]

[Lightning Talk Video (90 seconds)]

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

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

ETH Zürich

# SMASH: SW/HW Indexing Acceleration

---

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

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

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

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

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

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

[[Lightning Talk Video](#) (90 seconds)]

[[Full Talk Lecture](#) (30 minutes)]

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

Konstantinos Kanellopoulos<sup>1</sup> Nandita Vijaykumar<sup>2,1</sup> Christina Giannoula<sup>1,3</sup> Roknoddin Azizi<sup>1</sup>  
Skanda Koppula<sup>1</sup> Nika Mansouri Ghiasi<sup>1</sup> Taha Shahroodi<sup>1</sup> Juan Gomez Luna<sup>1</sup> Onur Mutlu<sup>1,2</sup>

# Data-Aware Virtual Memory Framework

---

Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,  
**"The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework"**

*Proceedings of the 47th International Symposium on Computer Architecture (ISCA)*, Virtual, June 2020.

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

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

[[ARM Research Summit Poster \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (26 minutes)]

[[Lightning Talk Video](#) (3 minutes)]

[[Lecture Video](#) (43 minutes)]

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

Nastaran Hajinazar<sup>\*†</sup> Pratyush Patel<sup>✉</sup> Minesh Patel<sup>\*</sup> Konstantinos Kanellopoulos<sup>\*</sup> Saugata Ghose<sup>‡</sup>  
Rachata Ausavarungnirun<sup>○</sup> Geraldo F. Oliveira<sup>\*</sup> Jonathan Appavoo<sup>◊</sup> Vivek Seshadri<sup>▽</sup> Onur Mutlu<sup>\*‡</sup>

<sup>\*</sup>*ETH Zürich*   <sup>†</sup>*Simon Fraser University*   <sup>✉</sup>*University of Washington*   <sup>‡</sup>*Carnegie Mellon University*

<sup>○</sup>*King Mongkut's University of Technology North Bangkok*   <sup>◊</sup>*Boston University*   <sup>▽</sup>*Microsoft Research India*

## Data-Aware (Expressive)

# Computing Architectures

# Concluding Remarks

# Recap: 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**

# Concluding Remarks

---

- It is time to design principled system architectures to solve the **data handling** (i.e., memory/storage) problem
- Design complete systems to be truly balanced, high-performance, and energy-efficient → intelligent architectures
  - **Data-centric, data-driven, data-aware**
- Enable computation capability inside and close to memory
- This can
  - Lead to **orders-of-magnitude** improvements
  - **Enable new applications & computing platforms**
  - **Enable better understanding of nature**
  - ...

# Architectures for Intelligent Machines

---

**Data-centric**

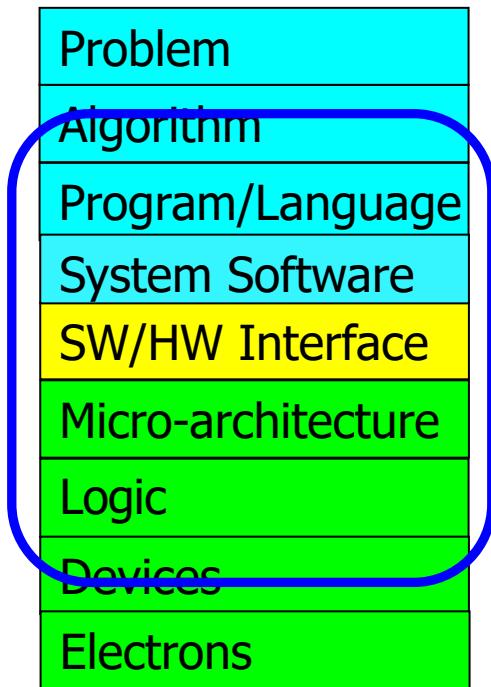
**Data-driven**

**Data-aware**



# We Need to Revisit the Entire Stack

---



**We can get there step by step**

# We Need to Exploit Good Principles

---

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

**Open minds**

# 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.*

# PIM Review and Open Problems (II)

---

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

Saugata Ghose<sup>†</sup>      Amirali Boroumand<sup>†</sup>      Jeremie S. Kim<sup>†\\$</sup>      Juan Gómez-Luna<sup>\\$</sup>      Onur Mutlu<sup>\\$†</sup>

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

<sup>\\$</sup>*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]

# A Longer Version of This Talk

---

- 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](mailto: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>

# 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

# Acknowledgments

---

- My current and past students and postdocs
  - Rachata Ausavarungnirun, Abhishek Bhowmick, Amirali Boroumand, Rui Cai, Yu Cai, Kevin Chang, Saugata Ghose, Kevin Hsieh, Tyler Huberty, Ben Jaiyen, Samira Khan, Jeremie Kim, Yoongu Kim, Yang Li, Jamie Liu, Lavanya Subramanian, Donghyuk Lee, Yixin Luo, Justin Meza, Gennady Pekhimenko, Vivek Seshadri, Lavanya Subramanian, Nandita Vijaykumar, HanBin Yoon, Jishen Zhao, ...
- My collaborators
  - Can Alkan, Chita Das, Phil Gibbons, Sriram Govindan, Norm Jouppi, Mahmut Kandemir, Mike Kozuch, Konrad Lai, Ken Mai, Todd Mowry, Yale Patt, Moinuddin Qureshi, Partha Ranganathan, Bikash Sharma, Kushagra Vaid, Chris Wilkerson, ...

# Acknowledgments

---



Think BIG, Aim HIGH!

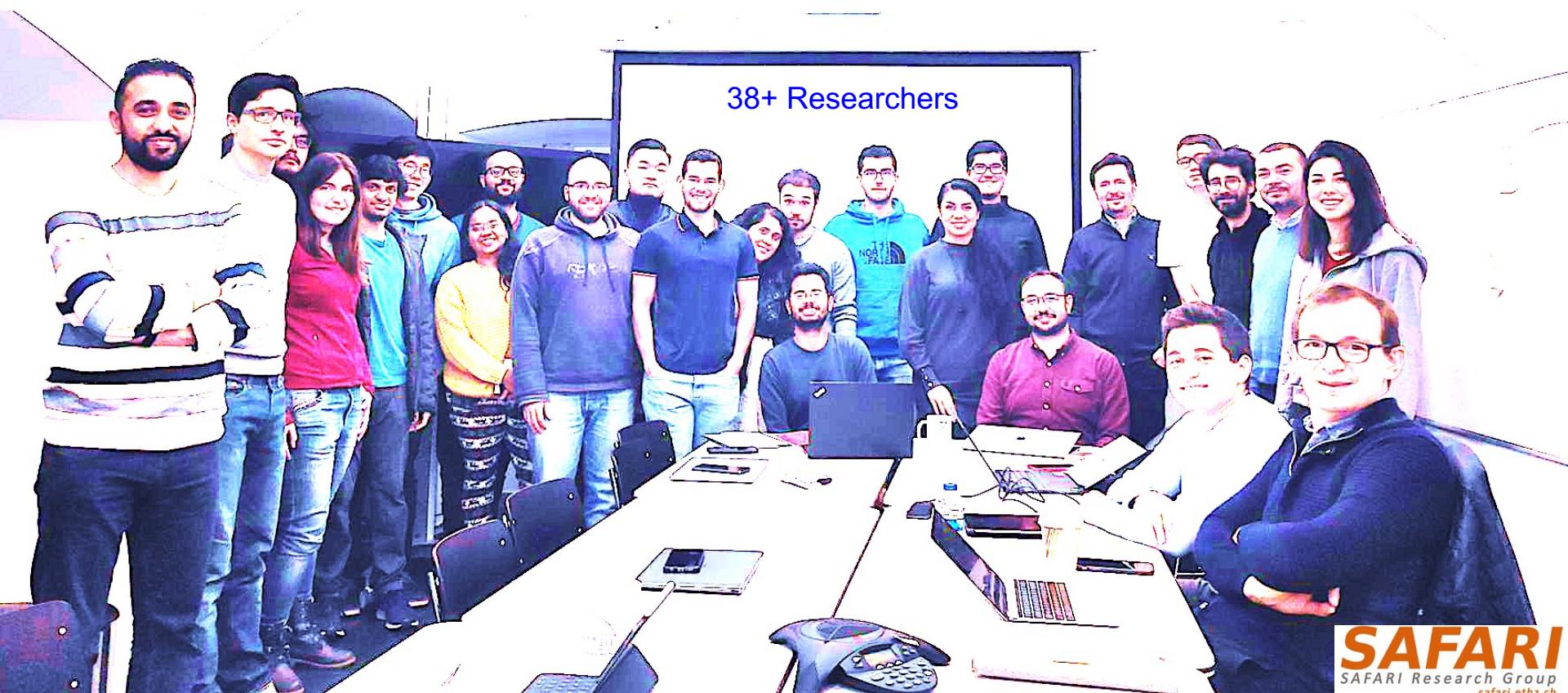
<https://safari.ethz.ch>

---

# Onur Mutlu's SAFARI Research Group

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

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



**SAFARI**  
SAFARI Research Group  
[safari.ethz.ch](http://safari.ethz.ch)

**Think BIG, Aim HIGH!**

**SAFARI**

<https://safari.ethz.ch>

# SAFARI Newsletter April 2020 Edition

---

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



Dear SAFARI friends,

---

2019 and the first three months of 2020 have been very positive eventful times for SAFARI.

# SAFARI Newsletter January 2021 Edition

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



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

# Referenced Papers, Talks, Artifacts

---

- All are available at

[\*\*https://people.inf.ethz.ch/omutlu/projects.htm\*\*](https://people.inf.ethz.ch/omutlu/projects.htm)

[\*\*http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en\*\*](http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en)

[\*\*https://www.youtube.com/onurmutlulectures\*\*](https://www.youtube.com/onurmutlulectures)

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

# Intelligent Architectures for Intelligent Machines

Onur Mutlu

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

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

3 May 2021

TU Vienna MiM (Mondays in Memory) Webinar

**SAFARI**

**ETH** zürich

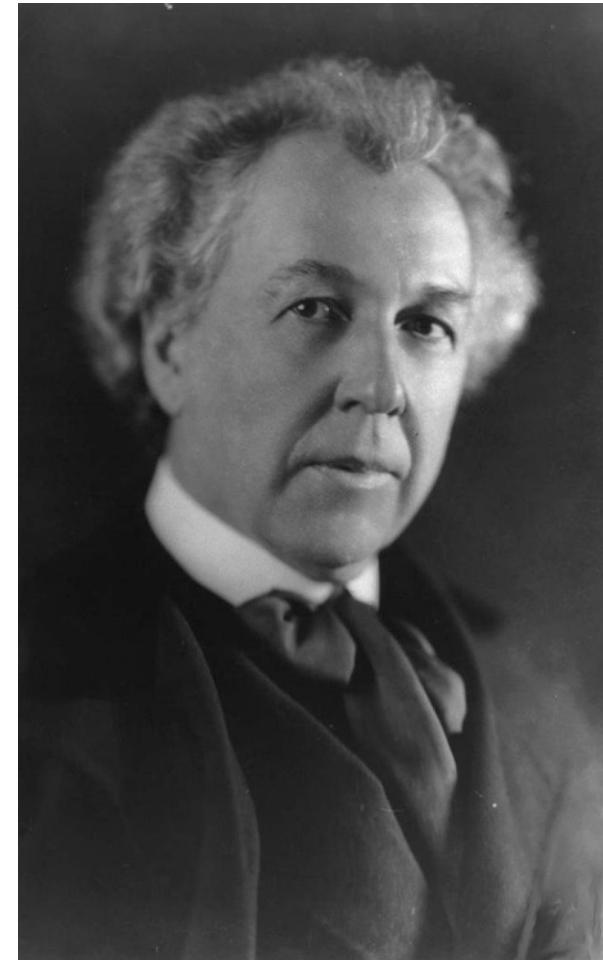
Carnegie Mellon

# Backup Slides

# A Quote from A Famous Architect

---

- “architecture [...] based upon principle, and not upon precedent”



# Precedent-Based Design?

---

- “architecture [...] based upon principle, and not upon precedent”



# Principled Design

---

- “architecture [...] based upon principle, and not upon precedent”





# The Overarching Principle

---

## Organic architecture

---

From Wikipedia, the free encyclopedia

**Organic architecture** is a philosophy of architecture which promotes harmony between human habitation and the natural world through design approaches so sympathetic and well integrated with its site, that buildings, furnishings, and surroundings become part of a unified, interrelated composition.

A well-known example of organic architecture is [Fallingwater](#), the residence Frank Lloyd Wright designed for the Kaufmann family in rural Pennsylvania. Wright had many choices to locate a home on this large site, but chose to place the home directly over the waterfall and creek creating a close, yet noisy dialog with the rushing water and the steep site. The horizontal striations of stone masonry with daring [cantilevers](#) of colored beige concrete blend with native rock outcroppings and the wooded environment.

---

# Another Example: Precedent-Based Design

---

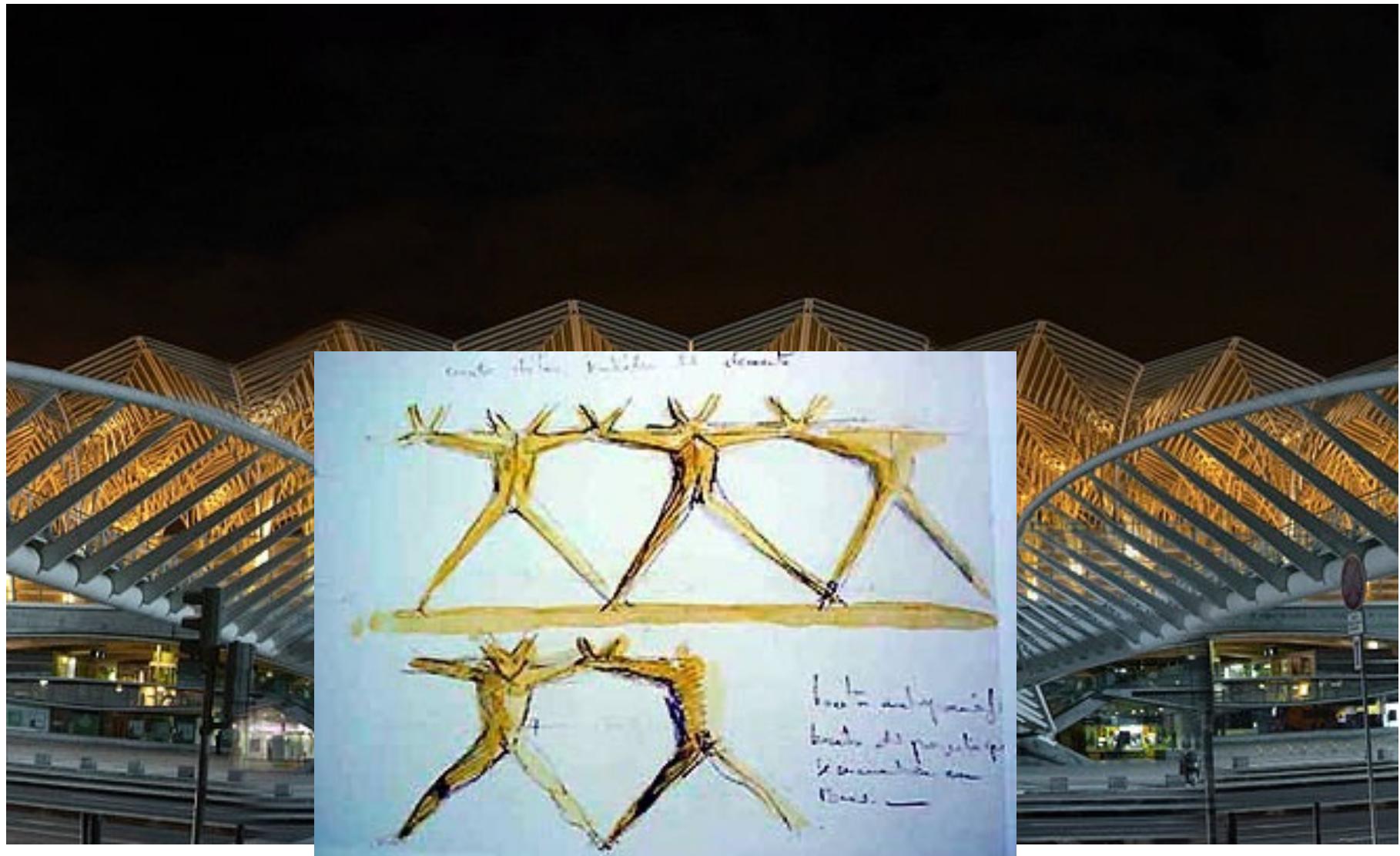


# Principled Design

---



# Another Principled Design



# Another Principled Design

---



# Principle Applied to Another Structure



# The Overarching Principle

---

## Zoomorphic architecture

---

From Wikipedia, the free encyclopedia

**Zoomorphic architecture** is the practice of using animal forms as the inspirational basis and blueprint for architectural design. "While animal forms have always played a role adding some of the deepest layers of meaning in architecture, it is now becoming evident that a new strand of **biomorphism** is emerging where the meaning derives not from any specific representation but from a more general allusion to biological processes."<sup>[1]</sup>

Some well-known examples of Zoomorphic architecture can be found in the [TWA Flight Center](#) building in [New York City](#), by [Eero Saarinen](#), or the [Milwaukee Art Museum](#) by [Santiago Calatrava](#), both inspired by the form of a bird's wings.<sup>[3]</sup>

---

# Overarching Principles for Computing?

---



# Readings, Videos, Reference Materials

# A Bit About Myself

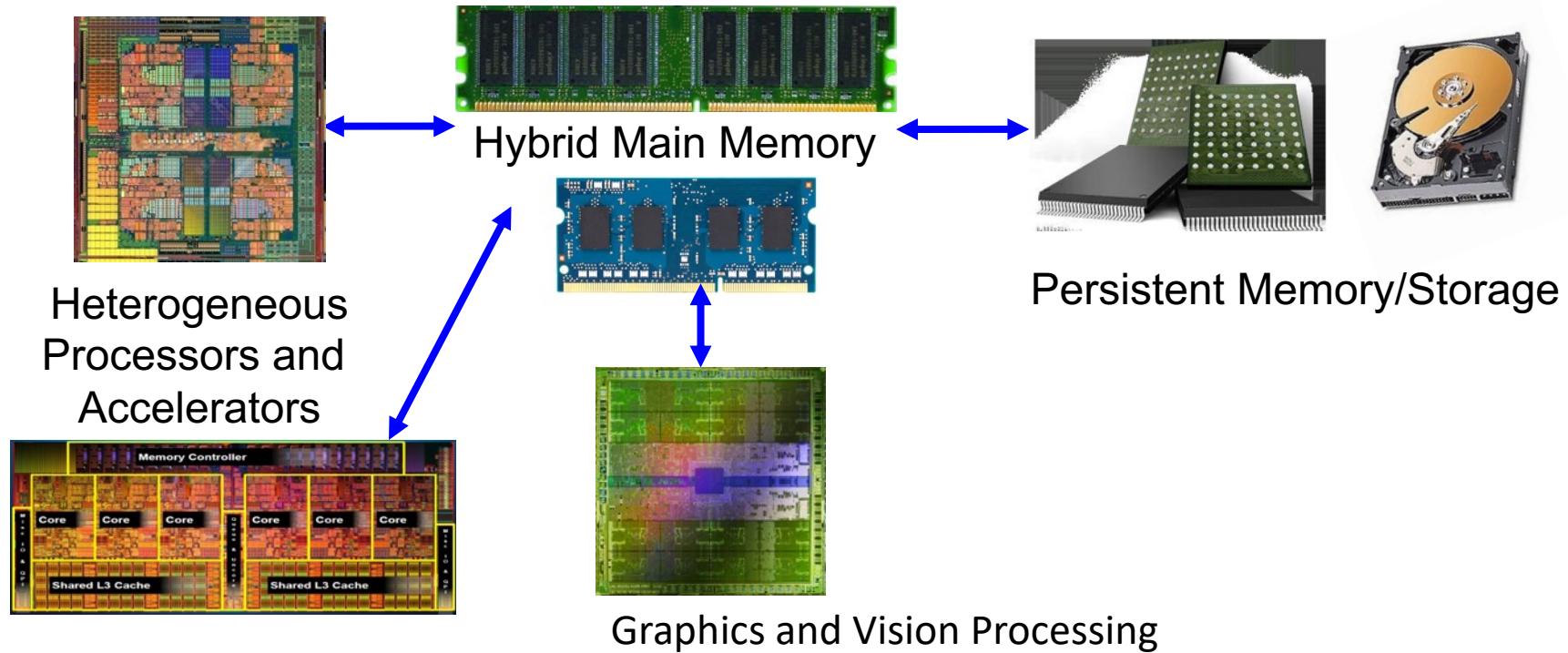
---



- Onur Mutlu
  - Full Professor @ ETH Zurich, 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](mailto: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

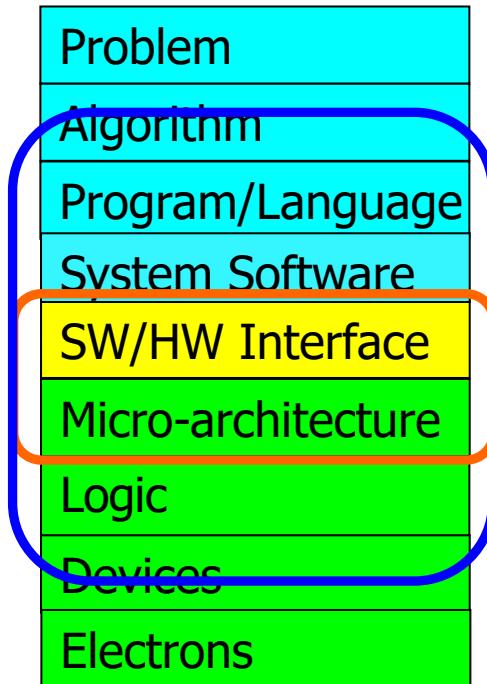
---

- 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

---

Computer Architecture  
(expanded view)

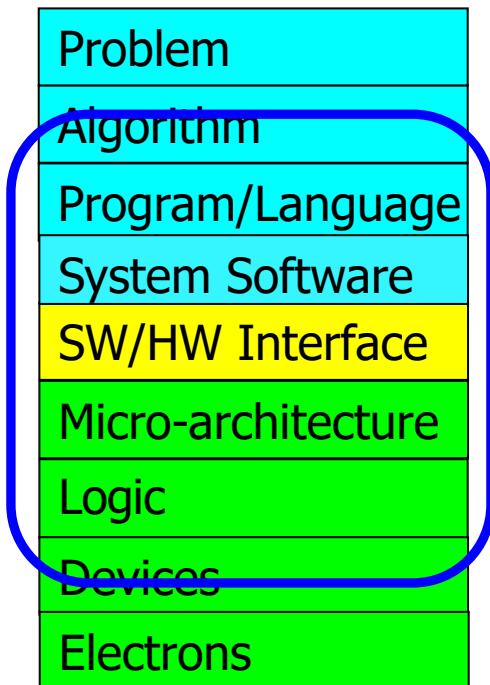


Computer Architecture  
(narrow view)

# Axiom

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

**we must take the expanded view**  
of computer architecture

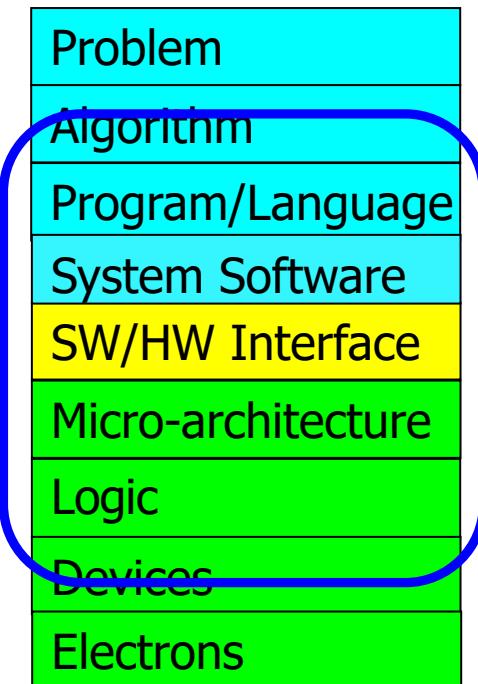


**Co-design across the hierarchy:  
Algorithms to devices**

**Specialize as much as possible  
within the design goals**

# Current Research Mission & Major Topics

## Build fundamentally better architectures



- 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/Graph/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

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

# Related Overview Talks

---

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

## ■ Future Computing Architectures

- [https://www.youtube.com/watch?v=kgiZISOcGFM&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJI&index=1](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](https://www.youtube.com/watch?v=njX_14584Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&index=16)

## ■ Accelerating Genome Analysis

- [https://www.youtube.com/watch?v=hPnSmfwu2-A&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJI&index=9](https://www.youtube.com/watch?v=hPnSmfwu2-A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&index=9)

## ■ Rethinking Memory System Design

- [https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJI&index=3](https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&index=3)

## ■ Intelligent Architectures for Intelligent Machines

- [https://www.youtube.com/watch?v=n8Aj\\_A0WSg8&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJI&index=22](https://www.youtube.com/watch?v=n8Aj_A0WSg8&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&index=22)

## ■ Revisiting RowHammer

- [https://www.youtube.com/watch?v=B58YT9hZM4g&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJI&index=25](https://www.youtube.com/watch?v=B58YT9hZM4g&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&index=25)

# Accelerated Memory Course (~6.5 hours)

---

- ACACES 2018
  - Memory Systems and Memory-Centric Computing Systems
  - Taught by Onur Mutlu July 9-13, 2018
  - ~6.5 hours of lectures
- Website for the Course including Videos, Slides, Papers
  - <https://people.inf.ethz.ch/omutlu/acaces2018.html>
  - <https://www.youtube.com/playlist?list=PL5Q2soXY2Zi-HXxomthrpDpMJm05P6J9x>
- All Papers are at:
  - <https://people.inf.ethz.ch/omutlu/projects.htm>
  - Final lecture notes and readings (for all topics)

# Longer Memory Course (~18 hours)

---

- TU Wien 2019
  - Memory Systems and Memory-Centric Computing Systems
  - Taught by Onur Mutlu June 12-19, 2019
  - ~18 hours of lectures
- Website for the Course including Videos, Slides, Papers
  - [https://safari.ethz.ch/memory\\_systems/TUWien2019](https://safari.ethz.ch/memory_systems/TUWien2019)
  - [https://www.youtube.com/playlist?list=PL5Q2soXY2Zi\\_gntM55VoMIKlw7YrXOhbl](https://www.youtube.com/playlist?list=PL5Q2soXY2Zi_gntM55VoMIKlw7YrXOhbl)
- All Papers are at:
  - <https://people.inf.ethz.ch/omutlu/projects.htm>
  - Final lecture notes and readings (for all topics)

# An Interview on Research and Education

---

- Computing Research and Education (@ ISCA 2019)
  - [https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi\\_4oP9LdL3cc8G6NIjD2Ydz](https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi_4oP9LdL3cc8G6NIjD2Ydz)
- Maurice Wilkes Award Speech (10 minutes)
  - [https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJI&index=15](https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&index=15)

# More Thoughts and Suggestions

---

- Onur Mutlu,  
**"Some Reflections (on DRAM)"**  
*Award Speech for ACM SIGARCH Maurice Wilkes Award, at the ISCA Awards Ceremony, Phoenix, AZ, USA, 25 June 2019.*  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Video of Award Acceptance Speech \(Youtube; 10 minutes\)](#) ([Youku; 13 minutes](#))]  
[[Video of Interview after Award Acceptance \(Youtube; 1 hour 6 minutes\)](#) ([Youku; 1 hour 6 minutes](#))]  
[[News Article on "ACM SIGARCH Maurice Wilkes Award goes to Prof. Onur Mutlu"](#)]
  
- Onur Mutlu,  
**"How to Build an Impactful Research Group"**  
*57th Design Automation Conference Early Career Workshop (DACE), Virtual, 19 July 2020.*  
[[Slides \(pptx\)](#) ([pdf](#))]

# Reference Overview Paper I

---

## Processing Data Where It Makes Sense: Enabling In-Memory Computation

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

<sup>a</sup>*ETH Zürich*

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

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

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,  
**"Processing Data Where It Makes Sense: Enabling In-Memory  
Computation"**

*Invited paper in Microprocessors and Microsystems (MICPRO), June 2019.  
[arXiv version]*

# Reference Overview Paper II

---

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

Saugata Ghose<sup>†</sup>      Amirali Boroumand<sup>†</sup>      Jeremie S. Kim<sup>†\\$</sup>      Juan Gómez-Luna<sup>\\$</sup>      Onur Mutlu<sup>\\$†</sup>

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

<sup>\\$</sup>*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]

# Reference Overview Paper III

---

## **Enabling the Adoption of Processing-in-Memory: Challenges, Mechanisms, Future Research Directions**

SAUGATA GHOSE, KEVIN HSIEH, AMIRALI BOROUMAND,  
RACHATA AUSAVARUNGNIRUN

Carnegie Mellon University

ONUR MUTLU

ETH Zürich and Carnegie Mellon University

Saugata Ghose, Kevin Hsieh, Amirali Boroumand, Rachata Ausavarungnirun, Onur Mutlu,  
**"Enabling the Adoption of Processing-in-Memory: Challenges, Mechanisms,  
Future Research Directions"**

*Invited Book Chapter*, to appear in 2018.

[Preliminary arxiv.org version]

# Reference Overview Paper IV

---

- Onur Mutlu and Lavanya Subramanian,  
**"Research Problems and Opportunities in Memory Systems"**  
*Invited Article in Supercomputing Frontiers and Innovations (SUPERFRI), 2014/2015.*

## Research Problems and Opportunities in Memory Systems

*Onur Mutlu<sup>1</sup>, Lavanya Subramanian<sup>1</sup>*

# Reference Overview Paper V

---

- 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](mailto:onur.mutlu@inf.ethz.ch)  
<https://people.inf.ethz.ch/omutlu>

# Reference Overview Paper VI

---

- Onur Mutlu,  
**"Memory Scaling: A Systems Architecture Perspective"**  
*Technical talk at MemCon 2013 (MEMCON), Santa Clara, CA, August 2013.* [Slides (pptx) (pdf)]  
[Video] [Coverage on StorageSearch]

## Memory Scaling: A Systems Architecture Perspective

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

# Reference Overview Paper VII

---



*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

# Reference Overview Paper VIII

---

- 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]

## RowHammer: A Retrospective

Onur Mutlu<sup>§‡</sup>      Jeremie S. Kim<sup>†§</sup>  
§ETH Zürich      †Carnegie Mellon University

# Related Videos and Course Materials (I)

---

- **Undergraduate Digital Design & Computer Architecture Course Lecture Videos** (2020, 2019, 2018, 2017, 2015, 2014, 2013)
- **Undergraduate Digital Design & Computer Architecture Course Materials** (2020, 2019, 2018, 2015, 2014, 2013)
- **Graduate Computer Architecture Course Lecture Videos** (2019, 2018, 2017, 2015, 2013)
- **Graduate Computer Architecture Course Materials** (2019, 2018, 2017, 2015, 2013)
- **Parallel Computer Architecture Course Materials (Lecture Videos)**

# Related Videos and Course Materials (II)

---

- **Seminar in Computer Architecture Course Lecture Videos** (Spring 2020, Fall 2019, Spring 2019, 2018)
- **Seminar in Computer Architecture Course Materials** (Spring 2020, Fall 2019, Spring 2019, 2018)
  
- **Memory Systems Course Lecture Videos** (Sept 2019, July 2019, June 2019, October 2018)
- **Memory Systems Short Course Lecture Materials** (Sept 2019, July 2019, June 2019, October 2018)
- **ACACES Summer School Memory Systems Course Lecture Videos** (2018, 2013)
- **ACACES Summer School Memory Systems Course Materials** (2018, 2013)

# 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/>
- <http://www.ece.cmu.edu/~safari/tools.html>

The screenshot shows the GitHub profile of the SAFARI Research Group. The main header reads "SAFARI Research Group at ETH Zurich and Carnegie Mellon University". Below it, a sub-header says "Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University." The GitHub interface includes a navigation bar with "Repositories 30", "People 27", "Teams 1", "Projects 0", and "Settings". Below the navigation bar are search and filter fields for "Search repositories...", "Type: All", "Language: All", and a "Customize pinned repositories" button. A green "New" button is also visible. On the left, a project card for "MQSim" is displayed, which is described as a fast and accurate simulator for modern multi-queue SSDs. It has 14 stars, 14 forks, and is MIT licensed, last updated 8 days ago. To the right of the project card are two boxes: "Top languages" (C++, C, C#, AGS Script, Verilog) and "Most used topics" (dram, reliability). The "Manage" link for topics is also visible.

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 14 MIT Updated 8 days ago

Top languages

- C++
- C
- C#
- AGS Script
- Verilog

Most used topics

dram reliability Manage

## 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++ ⚡ MIT ⚡ 93 ⚡ 170 ⚡ 37 ⚡ 2 Updated on Apr 13

## 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-asse...>

● 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](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](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 ⚡ 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

# Referenced Papers and Talks

---

- All are available at

[\*\*https://people.inf.ethz.ch/omutlu/projects.htm\*\*](https://people.inf.ethz.ch/omutlu/projects.htm)

[\*\*http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en\*\*](http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en)

[\*\*https://www.youtube.com/onurmutlulectures\*\*](https://www.youtube.com/onurmutlulectures)

# An Interview on Research and Education

---

- Computing Research and Education (@ ISCA 2019)
  - [https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi\\_4oP9LdL3cc8G6NIjD2Ydz](https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi_4oP9LdL3cc8G6NIjD2Ydz)
- Maurice Wilkes Award Speech (10 minutes)
  - [https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJI&index=15](https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&index=15)

# Low-Latency Memory

# Workload-DRAM Interaction Analysis

---

- Saugata Ghose, Tianshi Li, Nastaran Hajinazar, Damla Senol Cali, and Onur Mutlu,

## "Demystifying Workload–DRAM Interactions: An Experimental Study"

*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**)*, Phoenix, AZ, USA, June 2019.

[[Preliminary arXiv Version](#)]

[[Abstract](#)]

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

## Demystifying Complex Workload–DRAM Interactions: An Experimental Study

Saugata Ghose<sup>†</sup>

Tianshi Li<sup>†</sup>

Nastaran Hajinazar<sup>‡†</sup>

Damla Senol Cali<sup>†</sup>

Onur Mutlu<sup>§†</sup>

---

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>Simon Fraser University

<sup>§</sup>ETH Zürich

# Why Study Workload–DRAM Interactions?

SAFARI

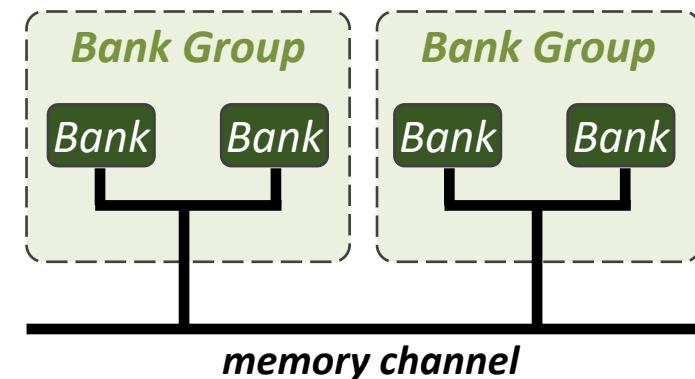
- Manufacturers are developing many new types of DRAM
  - DRAM limits performance, energy improvements:  
new types may overcome some limitations
  - Memory systems now serve a very diverse set of applications:  
can no longer take a one-size-fits-all approach
- So which DRAM type works best with which application?
  - Difficult to understand intuitively due to the complexity of the interaction
  - Can't be tested methodically on real systems: new type needs a new CPU
- We perform a wide-ranging experimental study to uncover the combined behavior of workloads and DRAM types
  - 115 prevalent/emerging applications and multiprogrammed workloads
  - 9 modern DRAM types: DDR3, DDR4, GDDR5, HBM, HMC, LPDDR3, LPDDR4, Wide I/O, Wide I/O 2

# Modern DRAM Types: Comparison to DDR3

SAFARI

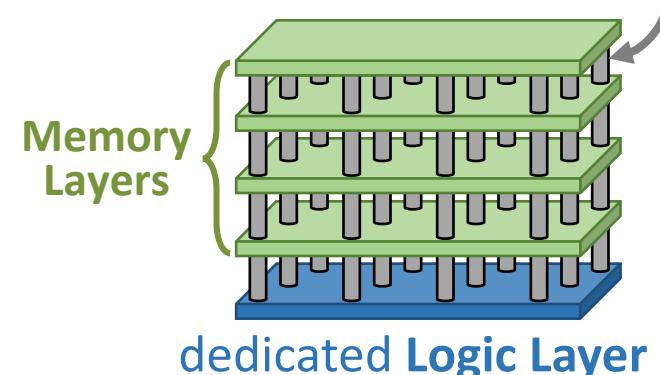
DRAM Type	Banks per Rank	Bank Groups	3D-Stacked	Low-Power
DDR3	8			
DDR4	16	✓	<i>increased latency</i>	
GDDR5	16	✓	<i>increased area/power</i>	
HBM High-Bandwidth Memory	16		✓	
HMC Hybrid Memory Cube	256	<i>narrower rows, higher latency</i>		✓
Wide I/O	4		✓	✓
Wide I/O 2	8		✓	✓
LPDDR3	8			✓
LPDDR4	16			✓

## ■ Bank groups



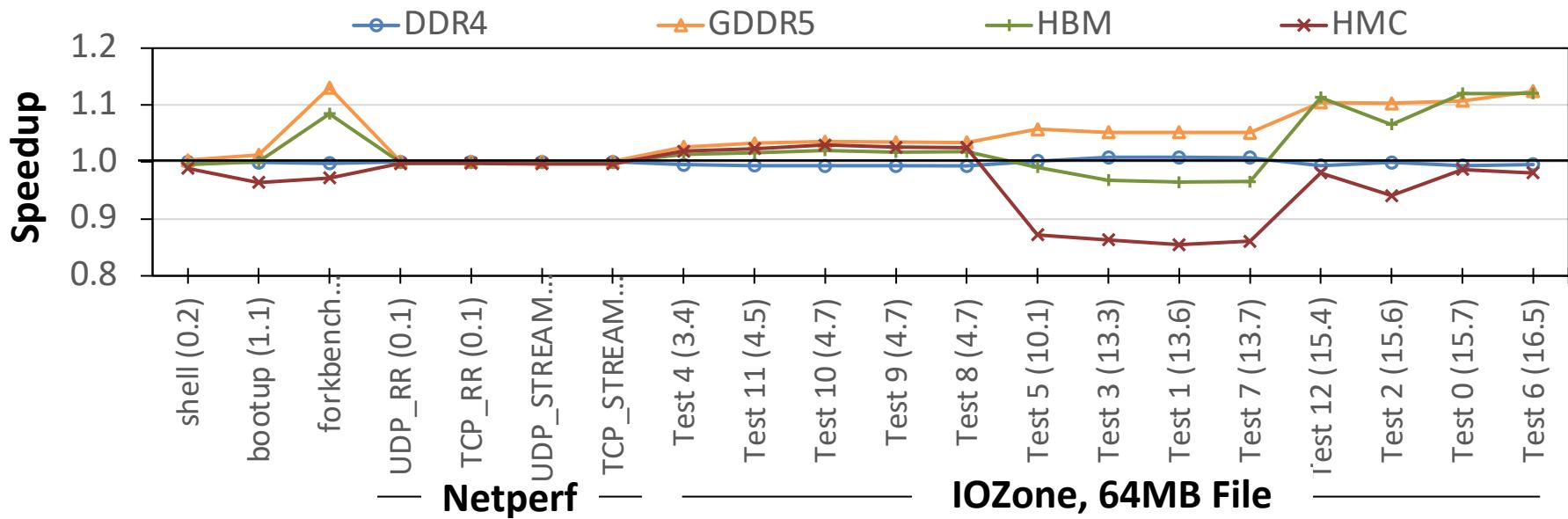
## ■ 3D-stacked DRAM

high bandwidth with Through-Silicon Vias (TSVs)



## 4. Need for Lower Access Latency: Performance

- New DRAM types often increase access latency in order to provide more banks, higher throughput
- Many applications can't make up for the increased latency
  - Especially true of common OS routines (e.g., file I/O, process forking)



- A variety of desktop/scientific, server/cloud, GPGPU applications

Several applications don't benefit from more parallelism

1. DRAM latency remains a critical bottleneck for many applications
2. Bank parallelism is not fully utilized by a wide variety of applications
3. Spatial locality continues to provide significant performance benefits if it is exploited by the memory subsystem
4. For some classes of applications, low-power memory can provide energy savings without sacrificing significant performance

- Manufacturers are developing many new types of DRAM
  - DRAM limits performance, energy improvements:  
new types may overcome some limitations
  - Memory systems now serve a very diverse set of applications:  
can no longer take a one-size-fits-all approach
  - Difficult to intuitively determine which DRAM–workload pair works best
- We perform a wide-ranging experimental study to uncover the combined behavior of workloads, DRAM types
  - 115 prevalent/emerging applications and multiprogrammed workloads
  - 9 modern DRAM types
- 12 key observations on DRAM–workload behavior

Open-source tools: <https://github.com/CMU-SAFARI/ramulator>

Full paper: <https://arxiv.org/pdf/1902.07609>

# The Memory Latency Problem

---

- High memory latency is a significant limiter of system performance and energy-efficiency
- It is becoming increasingly so with higher memory contention in multi-core and heterogeneous architectures
  - Exacerbating the bandwidth need
  - Exacerbating the QoS problem
- It increases processor design complexity due to the mechanisms incorporated to tolerate memory latency

# Retrospective: Conventional Latency Tolerance Techniques

- Caching [initially by Wilkes, 1965]
  - Widely used, simple, effective, but inefficient, passive
  - Not all applications/phases exhibit temporal or spatial locality
- Prefetching [initially in IBM 360/91, 1967]

**None of These  
Fundamentally Reduce  
Memory Latency**

ongoing research effort

- Out-of-order execution [initially by Tomasulo, 1967]
  - **Tolerates cache misses that cannot be prefetched**
  - Requires extensive hardware resources for tolerating long latencies

# Two Major Sources of Latency Inefficiency

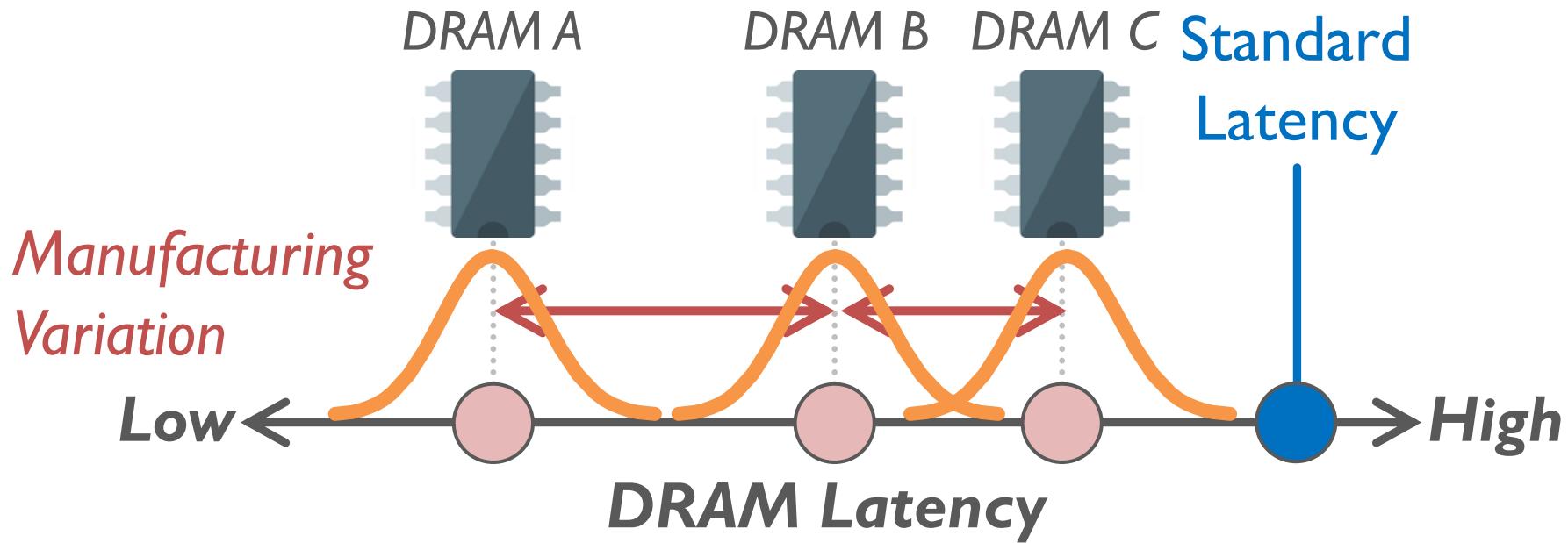
---

- Modern DRAM is **not** designed for low latency
  - Main focus is cost-per-bit (capacity)
- Modern DRAM latency is determined by **worst case** conditions and **worst case** devices
  - Much of memory latency is unnecessary

**Our Goal: Reduce Memory Latency  
at the Source of the Problem**

# Why is Memory Latency High?

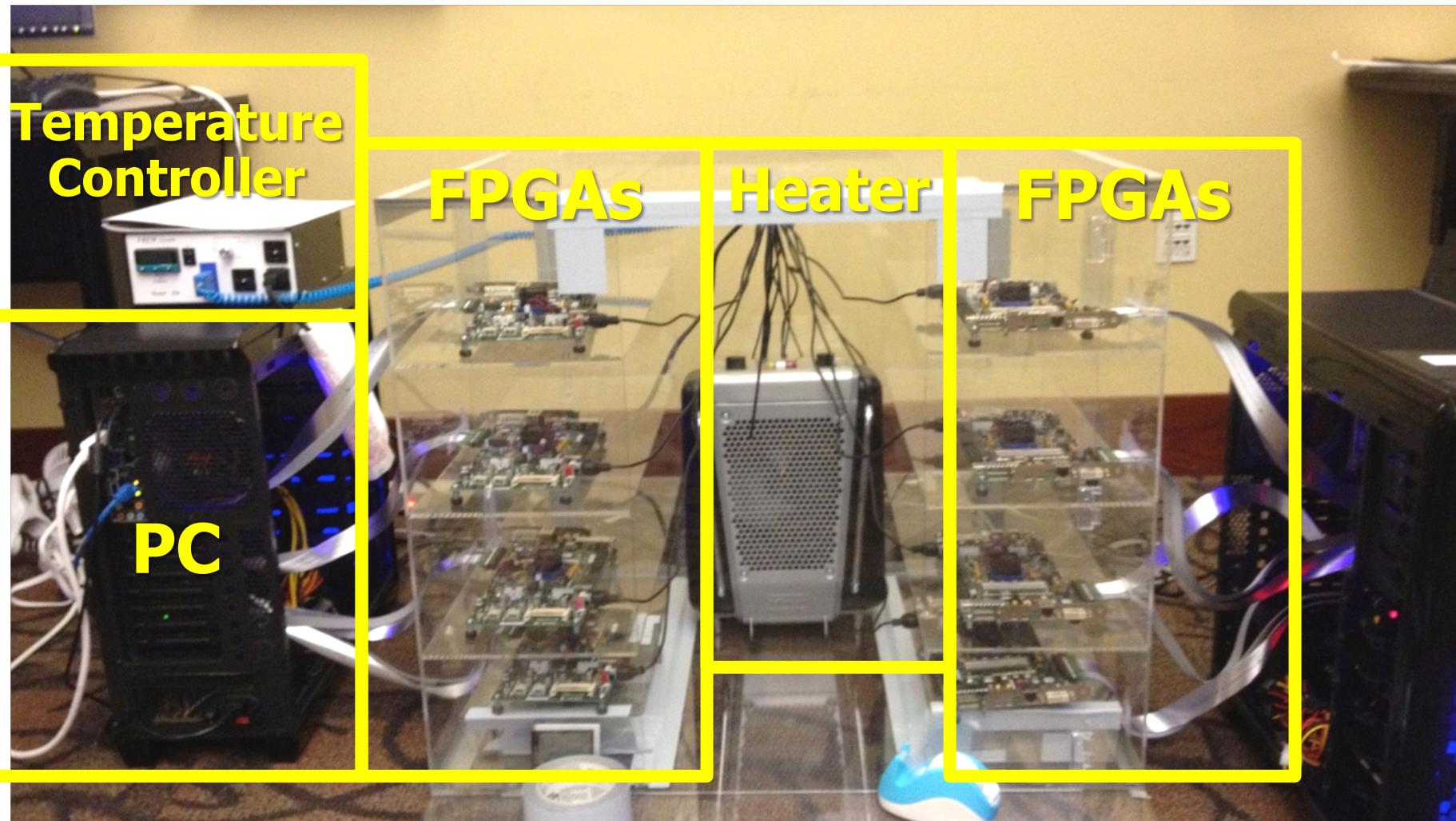
- DRAM latency: Delay as specified in DRAM standards
  - Doesn't reflect true DRAM device latency
- Imperfect manufacturing process → latency variation
- **High standard latency** chosen to increase yield



# Adaptive-Latency DRAM

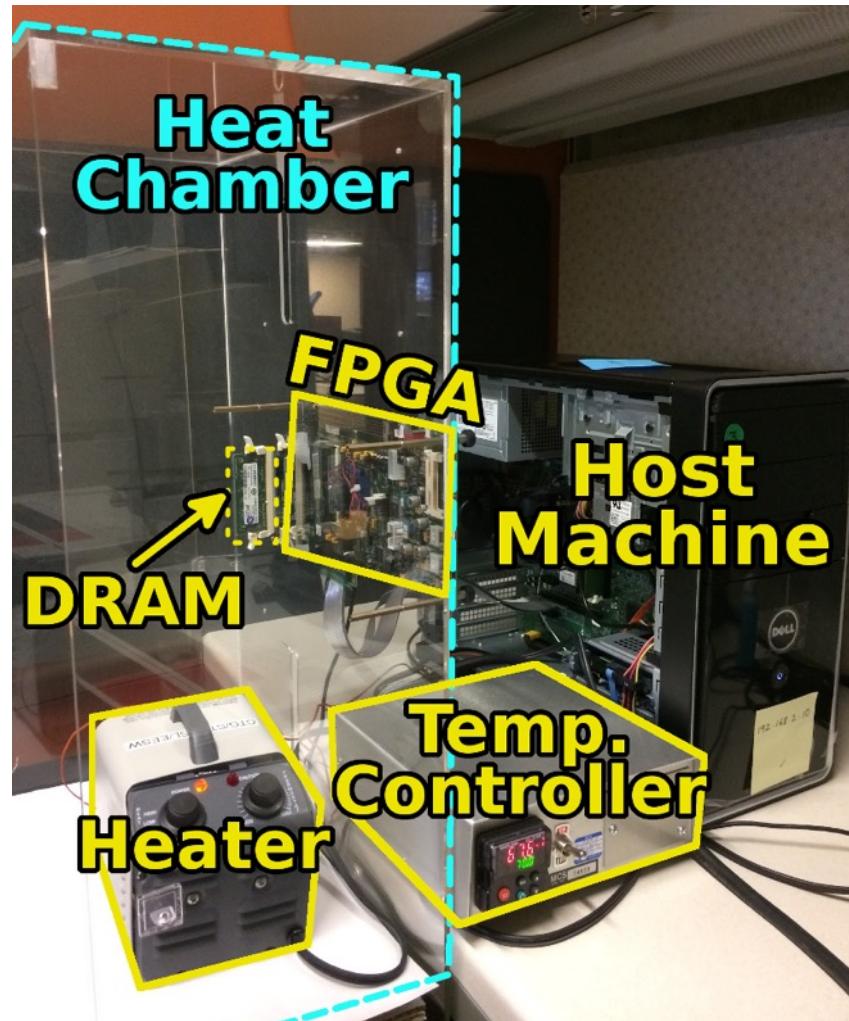
- *Key idea*
  - Optimize DRAM timing parameters online
- *Two components*
  - DRAM manufacturer provides multiple sets of reliable DRAM timing parameters at different temperatures for each DIMM
  - System monitors DRAM temperature & uses appropriate DRAM timing parameters

# 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<sup>1,2,3</sup> Nandita Vijaykumar<sup>3</sup> Samira Khan<sup>4,3</sup> Saugata Ghose<sup>3</sup> Kevin Chang<sup>3</sup>  
Gennady Pekhimenko<sup>5,3</sup> Donghyuk Lee<sup>6,3</sup> Oguz Ergin<sup>2</sup> Onur Mutlu<sup>1,3</sup>

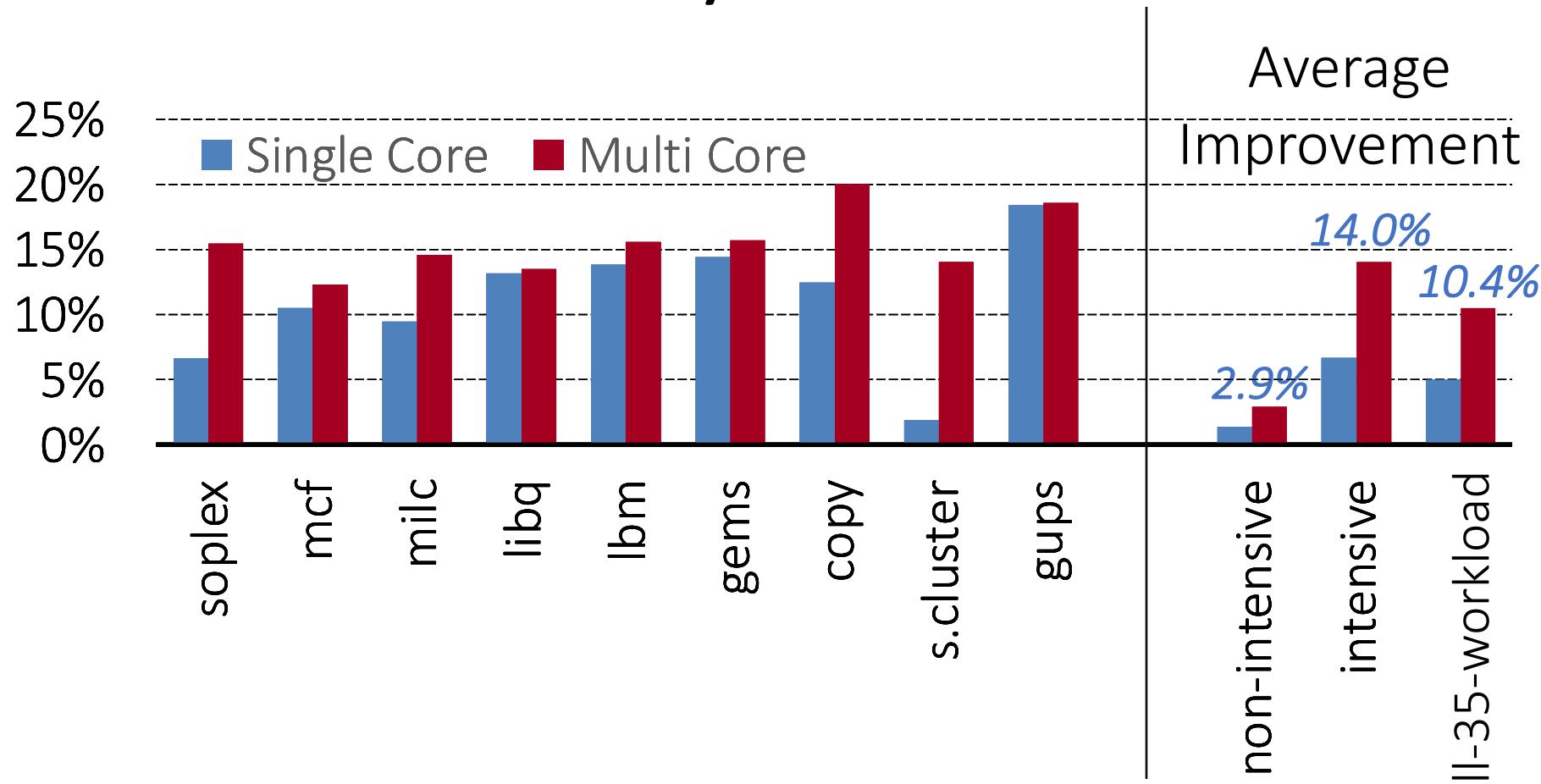
<sup>1</sup>*ETH Zürich*    <sup>2</sup>*TOBB University of Economics & Technology*    <sup>3</sup>*Carnegie Mellon University*  
<sup>4</sup>*University of Virginia*    <sup>5</sup>*Microsoft Research*    <sup>6</sup>*NVIDIA Research*

# Latency Reduction Summary of 115 DIMMs

- *Latency reduction for read & write (55°C)*
  - Read Latency: **32.7%**
  - Write Latency: **55.1%**
- *Latency reduction for each timing parameter (55°C)*
  - Sensing: **17.3%**
  - Restore: **37.3% (read), 54.8% (write)**
  - Precharge: **35.2%**

# AL-DRAM: Real-System Performance

*Performance Improvement*



*AL-DRAM provides high performance on  
memory-intensive workloads*

# Reducing Latency Also Reduces Energy

---

- AL-DRAM reduces DRAM power consumption
- Major reason: reduction in row activation time

# More on 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

# Tackling the Fixed Latency Mindset

---

- Reliable operation latency is actually very heterogeneous
  - Across temperatures, chips, parts of a chip, voltage levels, ...
- Idea: Dynamically find out and use the lowest latency one can reliably access a memory location with
  - Adaptive-Latency DRAM [HPCA 2015]
  - Flexible-Latency DRAM [SIGMETRICS 2016]
  - Design-Induced Variation-Aware DRAM [SIGMETRICS 2017]
  - Voltron [SIGMETRICS 2017]
  - DRAM Latency PUF [HPCA 2018]
  - DRAM Latency True Random Number Generator [HPCA 2019]
  - ...
- We would like to find sources of latency heterogeneity and exploit them to minimize latency (or create other benefits)

# 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. Chang<sup>1</sup>

Abhijith Kashyap<sup>1</sup>

Hasan Hassan<sup>1,2</sup>

Saugata Ghose<sup>1</sup> Kevin Hsieh<sup>1</sup> Donghyuk Lee<sup>1</sup> Tianshi Li<sup>1,3</sup>

Gennady Pekhimenko<sup>1</sup> Samira Khan<sup>4</sup> Onur Mutlu<sup>5,1</sup>

<sup>1</sup>Carnegie Mellon University

<sup>2</sup>TOBB ETÜ

<sup>3</sup>Peking University

<sup>4</sup>University of Virginia

<sup>5</sup>ETH 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: Exploiting Spatial Variation

---

- 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.

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

Jeremie S. Kim<sup>†§</sup>      Minesh Patel<sup>§</sup>      Hasan Hassan<sup>§</sup>      Onur Mutlu<sup>§‡</sup>  
<sup>†</sup>Carnegie Mellon University      <sup>§</sup>ETH Zürich

# 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<sup>†§</sup>      Minesh Patel<sup>§</sup>      Hasan Hassan<sup>§</sup>      Onur Mutlu<sup>§†</sup>  
<sup>†</sup>Carnegie Mellon University      <sup>§</sup>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<sup>†§</sup>

Minesh Patel<sup>§</sup>

Hasan Hassan<sup>§</sup>

Lois Orosa<sup>§</sup>

Onur Mutlu<sup>§‡</sup>

<sup>†</sup>Carnegie Mellon University

<sup>§</sup>ETH Zürich

# ChargeCache: Exploiting Access Patterns

---

- Hasan Hassan, Gennady Pekhimenko, Nandita Vijaykumar, Vivek Seshadri, Donghyuk Lee, Oguz Ergin, and Onur Mutlu,  
**"ChargeCache: Reducing DRAM Latency by Exploiting Row Access Locality"**

*Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (**HPCA**), Barcelona, Spain, March 2016.*

[Slides (pptx) (pdf)]

[Source Code]

## ChargeCache: Reducing DRAM Latency by Exploiting Row Access Locality

Hasan Hassan<sup>†\*</sup>, Gennady Pekhimenko<sup>†</sup>, Nandita Vijaykumar<sup>†</sup>  
Vivek Seshadri<sup>†</sup>, Donghyuk Lee<sup>†</sup>, Oguz Ergin<sup>\*</sup>, Onur Mutlu<sup>†</sup>

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

<sup>\*</sup>*TOBB University of Economics & Technology*

# Exploiting Subarray Level Parallelism

---

- Yoongu Kim, Vivek Seshadri, Donghyuk Lee, Jamie Liu, and Onur Mutlu,

**"A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM"**

*Proceedings of the 39th International Symposium on Computer Architecture (ISCA), Portland, OR, June 2012.* Slides (pptx)

## A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM

Yoongu Kim

Vivek Seshadri

Donghyuk Lee

Jamie Liu

Onur Mutlu

Carnegie Mellon University

# 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

# LISA: Low-cost Inter-linked Subarrays

---

- Kevin K. Chang, Prashant J. Nair, Saugata Ghose, Donghyuk Lee, Moinuddin K. Qureshi, and Onur Mutlu,

**"Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM"**

*Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA), Barcelona, Spain, March 2016.*

[Slides (pptx) (pdf)]

[Source Code]

## **Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM**

Kevin K. Chang<sup>†</sup>, Prashant J. Nair<sup>\*</sup>, Donghyuk Lee<sup>†</sup>, Saugata Ghose<sup>†</sup>, Moinuddin K. Qureshi<sup>\*</sup>, and Onur Mutlu<sup>†</sup>

<sup>†</sup>*Carnegie Mellon University*    <sup>\*</sup>*Georgia Institute of Technology*

# The CROW Substrate for DRAM

---

- Hasan Hassan, Minesh Patel, Jeremie S. Kim, A. Giray Yaglikci, Nandita Vijaykumar, Nika Mansouri Ghiasi, Saugata Ghose, and Onur Mutlu,

**"CROW: A Low-Cost Substrate for Improving DRAM Performance, Energy Efficiency, and Reliability"**

*Proceedings of the 46th International Symposium on Computer Architecture (ISCA), Phoenix, AZ, USA, June 2019.*

## CROW: A Low-Cost Substrate for Improving DRAM Performance, Energy Efficiency, and Reliability

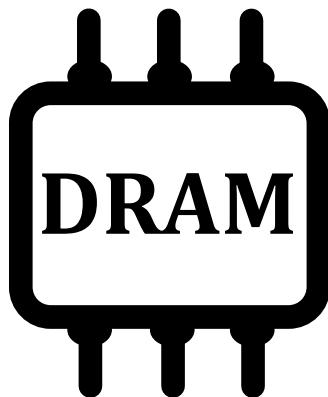
Hasan Hassan<sup>†</sup>    Minesh Patel<sup>†</sup>    Jeremie S. Kim<sup>†§</sup>    A. Giray Yaglikci<sup>†</sup>  
Nandita Vijaykumar<sup>†§</sup>    Nika Mansouri Ghiasi<sup>†</sup>    Saugata Ghose<sup>§</sup>    Onur Mutlu<sup>†§</sup>

<sup>†</sup>*ETH Zürich*    <sup>§</sup>*Carnegie Mellon University*

# CROW: The Copy Row Substrate

## [ISCA 2019]

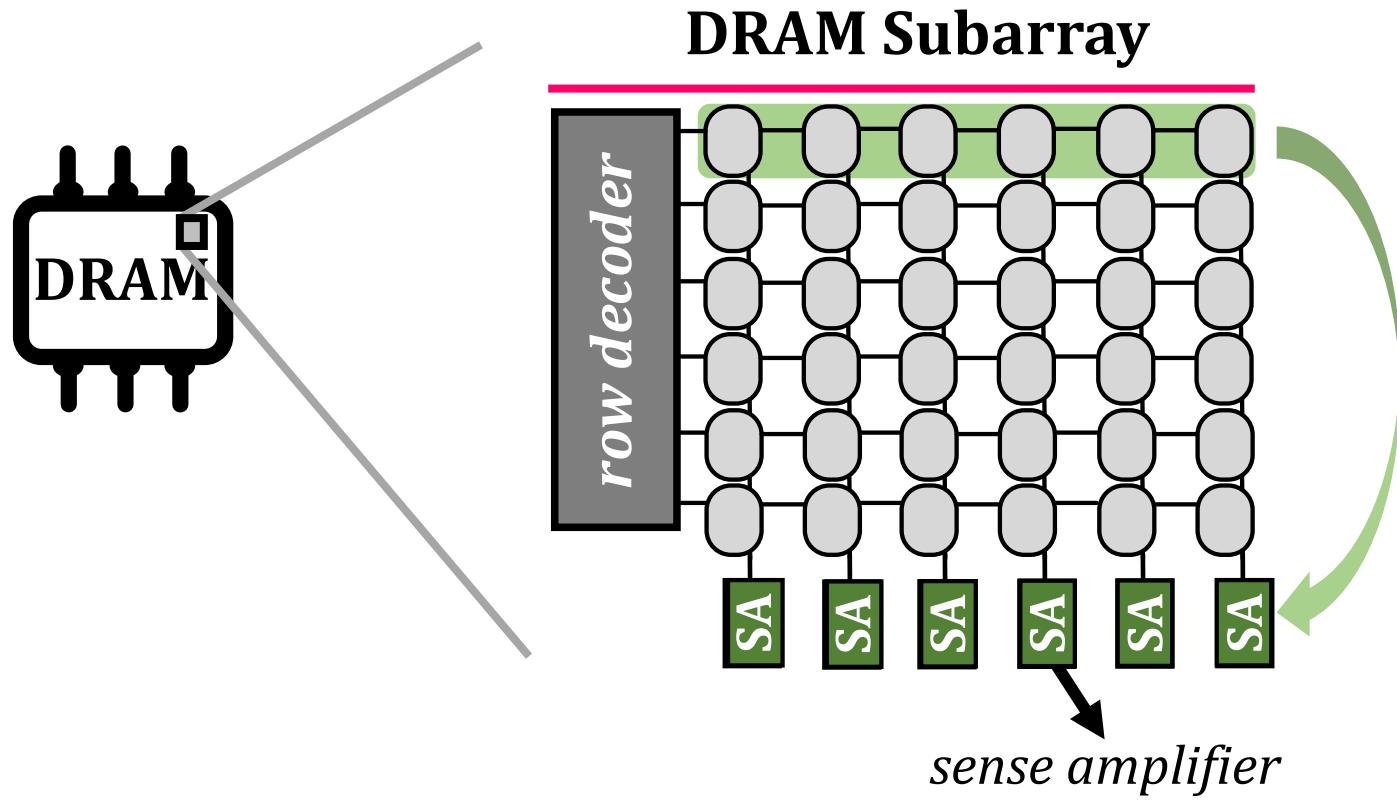
# Challenges of DRAM Scaling



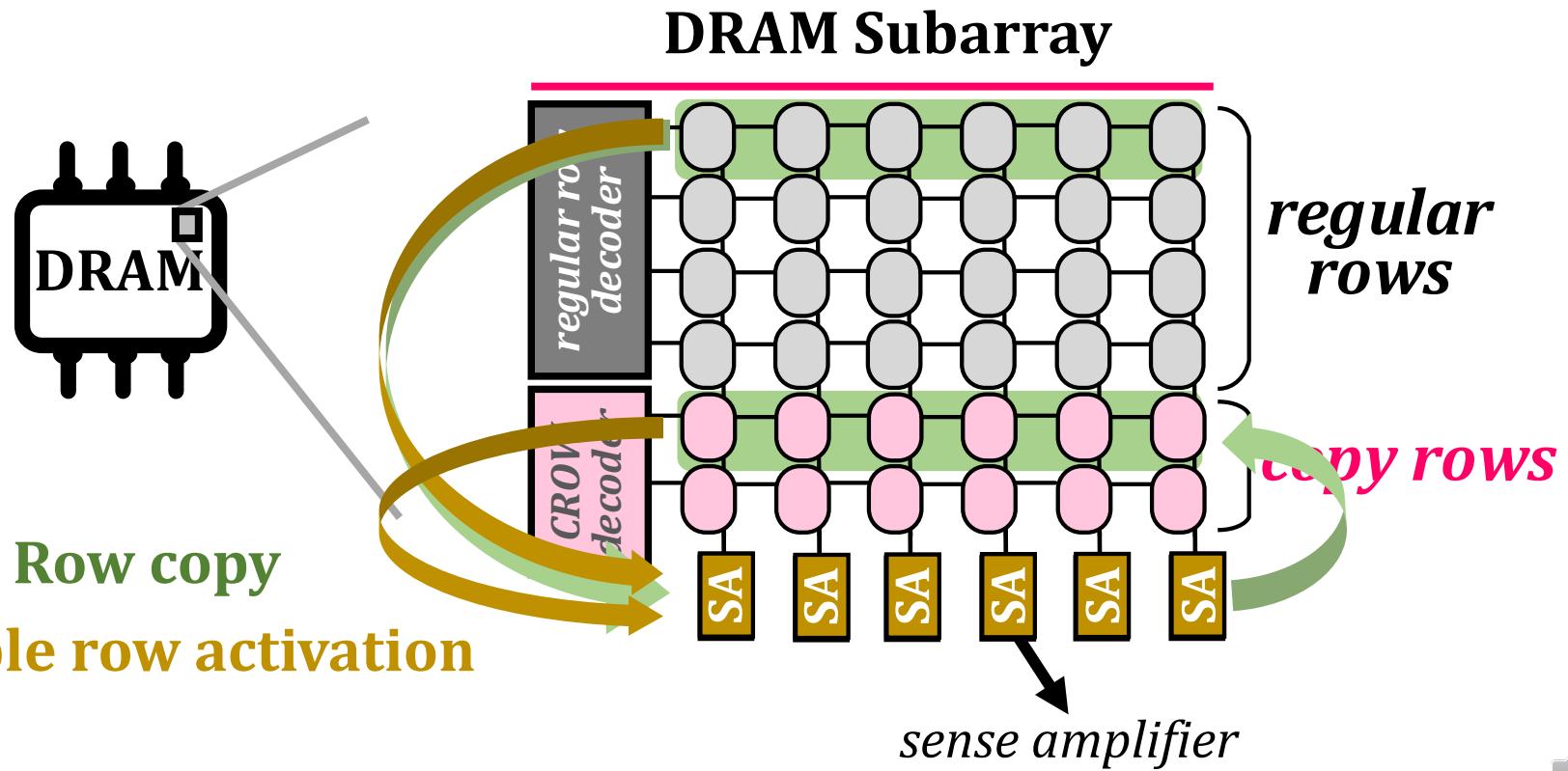
- 1 access latency
- 2 refresh overhead
- 3 exposure to vulnerabilities



# Conventional DRAM



# Copy Row DRAM (CROW)



# Use Cases of CROW

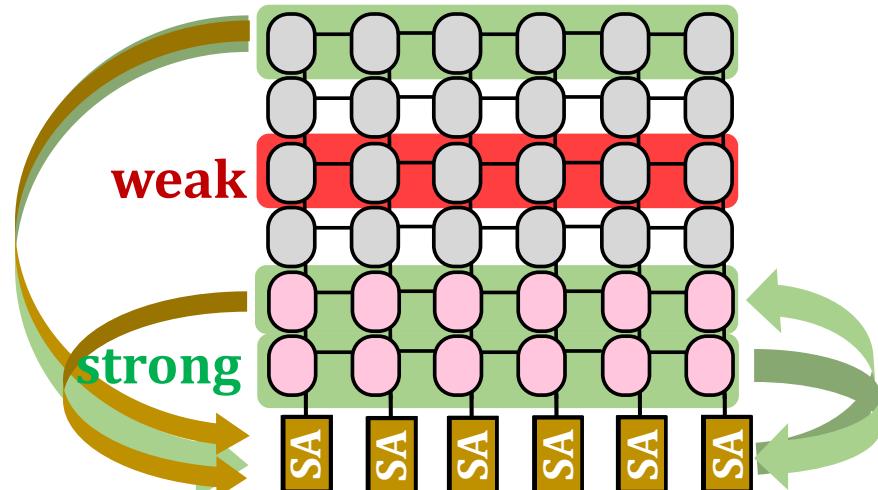
## ➤ CROW-cache

- ✓ reduces *access latency*

## ➤ CROW-ref

- ✓ reduces DRAM *refresh overhead*

## ➤ A mechanism for protecting against *RowHammer*



# Key Results

## CROW-cache + CROW-ref

- 20% speedup
- 22% less DRAM energy

## Hardware Overhead

- 0.5% DRAM chip area
- 1.6% DRAM capacity
- 11.3 KiB memory controller storage



# More on CROW

---

- Hasan Hassan, Minesh Patel, Jeremie S. Kim, A. Giray Yaglikci, Nandita Vijaykumar, Nika Mansouri Ghiasi, Saugata Ghose, and Onur Mutlu,

## **"CROW: A Low-Cost Substrate for Improving DRAM Performance, Energy Efficiency, and Reliability"**

*Proceedings of the 46th International Symposium on Computer Architecture (ISCA), Phoenix, AZ, USA, June 2019.*

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

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

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

[[Lightning Talk Video](#) (3 minutes)]

[[Full Talk Video](#) (16 minutes)]

[[Source Code for CROW](#) (Ramulator and Circuit Modeling)]

## **CROW: A Low-Cost Substrate for Improving DRAM Performance, Energy Efficiency, and Reliability**

Hasan Hassan<sup>†</sup>    Minesh Patel<sup>†</sup>    Jeremie S. Kim<sup>†§</sup>    A. Giray Yaglikci<sup>†</sup>

Nandita Vijaykumar<sup>†§</sup>    Nika Mansouri Ghiasi<sup>†</sup>    Saugata Ghose<sup>§</sup>    Onur Mutlu<sup>†§</sup>

<sup>†</sup>*ETH Zürich*    <sup>§</sup>*Carnegie Mellon University*

# 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<sup>§†</sup>

Taha Shahroodi<sup>§</sup>

Hasan Hassan<sup>§</sup>

Minesh Patel<sup>§</sup>

A. Giray Yağlıkçı<sup>§</sup>

Lois Orosa<sup>§</sup>

Jisung Park<sup>§</sup>

Onur Mutlu<sup>§</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*ShanghaiTech University*

# CLR-DRAM: Capacity-Latency Reconfigurable DRAM [ISCA 2020]

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

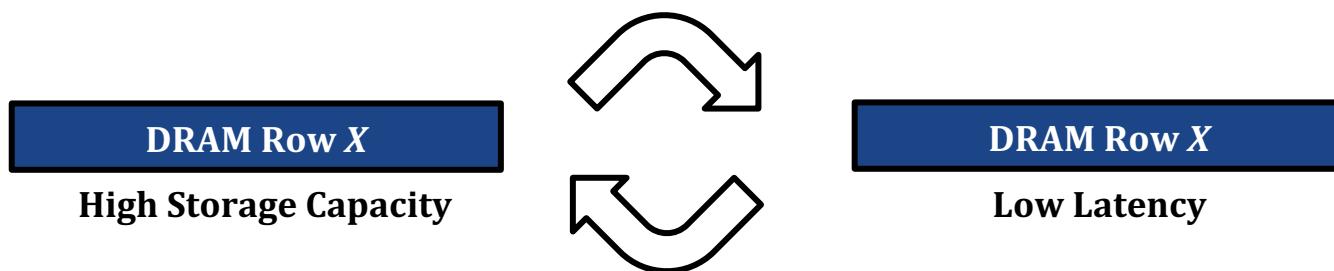
Haocong Luo Taha Shahroodi Hasan Hassan Minesh Patel  
A. Giray Yaglikcı Lois Orosa Jisung Park Onur Mutlu



上海科技大学  
ShanghaiTech University

# Motivation & Goal

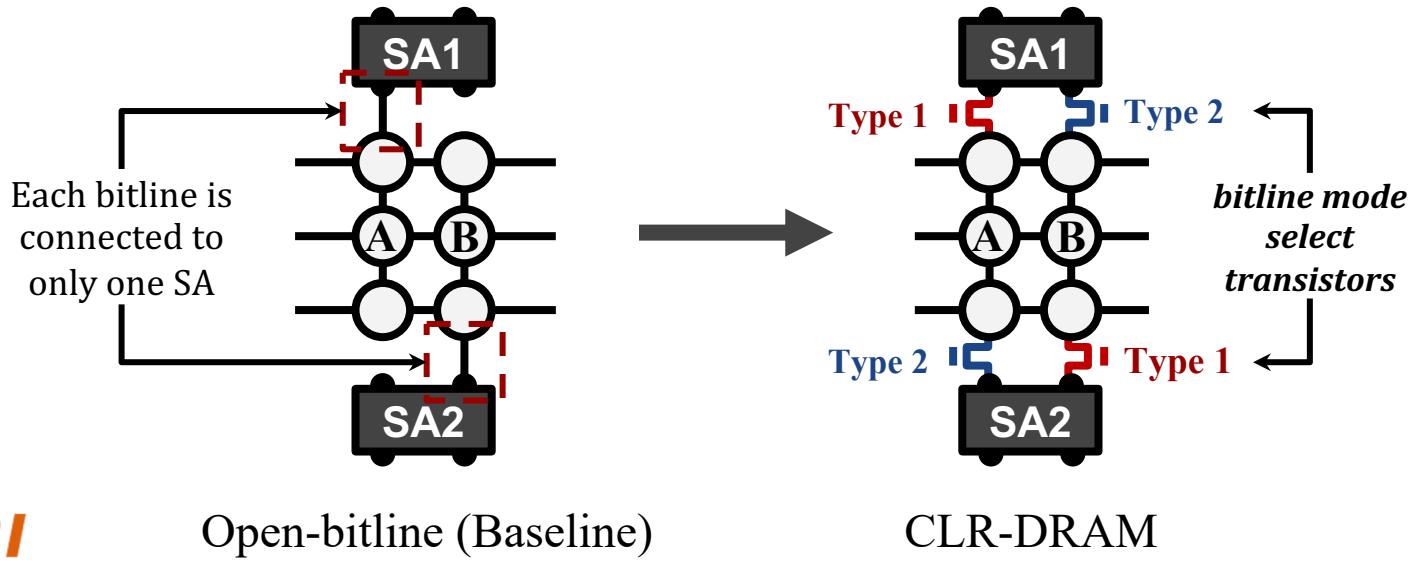
- Workloads and systems have **varying** main memory capacity and latency demands.
- Existing commodity DRAM makes **static** capacity-latency trade-off at **design time**.
- Systems miss opportunities to improve performance by adapting to changes in main memory capacity and latency demands.
- **Goal:** Design a low-cost DRAM architecture that can be **dynamically** configured to have high capacity or low latency at a fine granularity (i.e., at the granularity of a row).



# CLR-DRAM (Capacity-Latency-Reconfigurable DRAM)

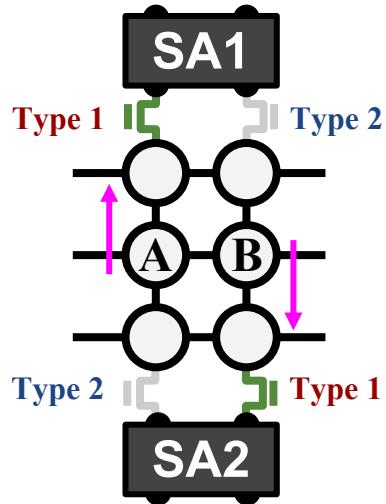
- **CLR-DRAM (Capacity-Latency-Reconfigurable DRAM):**
  - A **low cost** DRAM architecture that enables a single DRAM row to *dynamically* switch between **max-capacity mode** or **high-performance mode**.
- **Key Idea:**

*Dynamically* configure the connections between DRAM cells and sense amplifiers in the density-optimized open-bitline architecture.



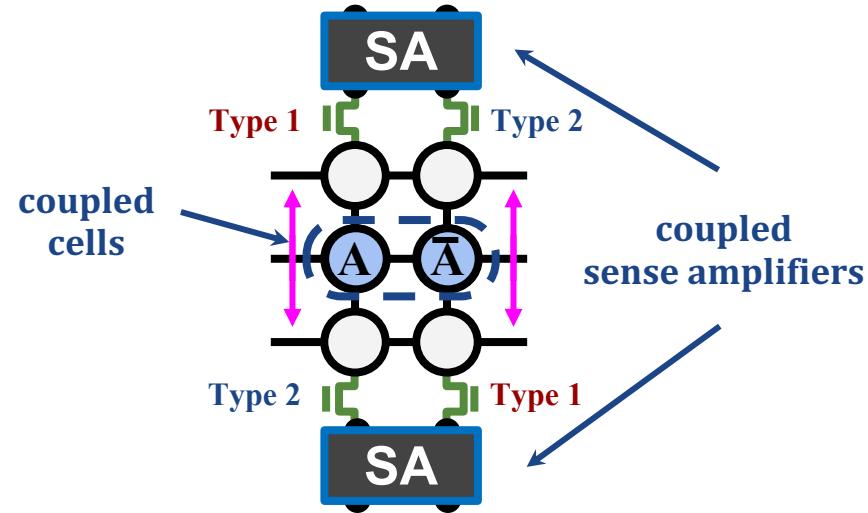
# CLR-DRAM (Capacity-Latency-Reconfigurable DRAM)

- Max-capacity mode



*mimics the cell-to-SA connections as in the open-bitline architecture*

- High-performance mode



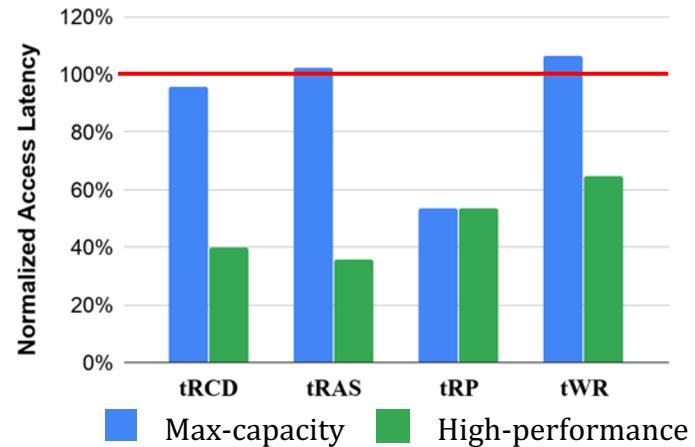
The same storage capacity as the conventional open-bitline architecture

Reduced latency and refresh overhead via coupled cell/SA operation

# Key Results

- **DRAM Latency Reduction:**

- Activation latency (**tRCD**) by **60.1%**
- Restoration latency (**tRAS**) by **64.2%**
- Precharge latency (**tRP**) by **46.4%**
- Write-recovery latency (**tWR**) by **35.2%**



- **System-level Benefits:**

- Performance improvement: **18.6%**
- DRAM energy reduction: **29.7%**
- DRAM refresh energy reduction: **66.1%**

We hope that CLR-DRAM can be exploited to develop more flexible systems that can adapt to the diverse and changing DRAM capacity and latency demands of workloads.

# More on CLR-DRAM

---

- 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<sup>§†</sup>    Taha Shahroodi<sup>§</sup>    Hasan Hassan<sup>§</sup>    Minesh Patel<sup>§</sup>  
A. Giray Yağlıkçı<sup>§</sup>    Lois Orosa<sup>§</sup>    Jisung Park<sup>§</sup>    Onur Mutlu<sup>§</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*ShanghaiTech University*

# Reducing Refresh Latency

---

- Anup Das, Hasan Hassan, and Onur Mutlu,  
**"VRL-DRAM: Improving DRAM Performance via  
Variable Refresh Latency"**  
*Proceedings of the 55th Design Automation  
Conference (DAC), San Francisco, CA, USA, June 2018.*

## VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency

Anup Das  
Drexel University  
Philadelphia, PA, USA  
anup.das@drexel.edu

Hasan Hassan  
ETH Zürich  
Zürich, Switzerland  
hhasan@ethz.ch

Onur Mutlu  
ETH Zürich  
Zürich, Switzerland  
omutlu@gmail.com

# Parallelizing Refreshes and Accesses

---

- Kevin Chang, Donghyuk Lee, Zeshan Chishti, Alaa Alameldeen, Chris Wilkerson, Yoongu Kim, and Onur Mutlu,  
**"Improving DRAM Performance by Parallelizing Refreshes with Accesses"**  
*Proceedings of the 20th International Symposium on High-Performance Computer Architecture (HPCA)*, Orlando, FL, February 2014.  
[Summary] [Slides (pptx) (pdf)]

## Reducing Performance Impact of DRAM Refresh by Parallelizing Refreshes with Accesses

Kevin Kai-Wei Chang   Donghyuk Lee   Zeshan Chishti†  
Alaa R. Alameldeen†   Chris Wilkerson†   Yoongu Kim   Onur Mutlu  
Carnegie Mellon University   †Intel Labs

# Eliminating Refreshes

---

- Jamie Liu, Ben Jaiyen, Richard Veras, and Onur Mutlu,  
**"RAIDR: Retention-Aware Intelligent DRAM Refresh"**  
*Proceedings of the 39th International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2012.  
[Slides \(pdf\)](#)

## RAIDR: Retention-Aware Intelligent DRAM Refresh

Jamie Liu    Ben Jaiyen    Richard Veras    Onur Mutlu  
Carnegie Mellon University

# Analysis of Latency-Voltage in DRAM Chips

---

- Kevin Chang, A. Giray Yaglikci, Saugata Ghose, Aditya Agrawal, Niladrish Chatterjee, Abhijith Kashyap, Donghyuk Lee, Mike O'Connor, Hasan Hassan, and Onur Mutlu,

## "Understanding Reduced-Voltage Operation in Modern DRAM Devices: Experimental Characterization, Analysis, and Mechanisms"

*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Urbana-Champaign, IL, USA, June 2017.

## **Understanding Reduced-Voltage Operation in Modern DRAM Chips: Characterization, Analysis, and Mechanisms**

Kevin K. Chang<sup>†</sup> Abdullah Giray Yağlıkçı<sup>†</sup> Saugata Ghose<sup>†</sup> Aditya Agrawal<sup>¶</sup> Niladrish Chatterjee<sup>¶</sup>  
Abhijith Kashyap<sup>†</sup> Donghyuk Lee<sup>¶</sup> Mike O'Connor<sup>¶,‡</sup> Hasan Hassan<sup>§</sup> Onur Mutlu<sup>§,†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>¶</sup>NVIDIA

<sup>‡</sup>The University of Texas at Austin

<sup>§</sup>ETH Zürich

# VAMPIRE DRAM Power Model

---

- Saugata Ghose, A. Giray Yaglikci, Raghav Gupta, Donghyuk Lee, Kais Kudrolli, William X. Liu, Hasan Hassan, Kevin K. Chang, Niladrish Chatterjee, Aditya Agrawal, Mike O'Connor, and Onur Mutlu,

## "What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study"

*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Irvine, CA, USA, June 2018.

[Abstract]

[POMACS Journal Version (same content, different format)]

[Slides (pptx) (pdf)]

[VAMPIRE DRAM Power Model]

## **What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study**

Saugata Ghose<sup>†</sup>      Abdullah Giray Yağlıkçı<sup>‡†</sup>      Raghav Gupta<sup>†</sup>      Donghyuk Lee<sup>§</sup>  
Kais Kudrolli<sup>†</sup>      William X. Liu<sup>†</sup>      Hasan Hassan<sup>‡</sup>      Kevin K. Chang<sup>†</sup>  
Niladrish Chatterjee<sup>§</sup>      Aditya Agrawal<sup>§</sup>      Mike O'Connor<sup>§¶</sup>      Onur Mutlu<sup>‡†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>ETH Zürich

<sup>§</sup>NVIDIA

<sup>¶</sup>University of Texas at Austin

We Can Reduce  
Memory Latency  
with Change of Mindset

# Main Memory Needs Intelligent Controllers to Reduce Latency