

# Computer Architecture

## Lecture 2a: Memory Systems: Challenges and Opportunities

Prof. Onur Mutlu

ETH Zürich

Fall 2023

29 September 2023

# Four Key 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

# Memory & Storage

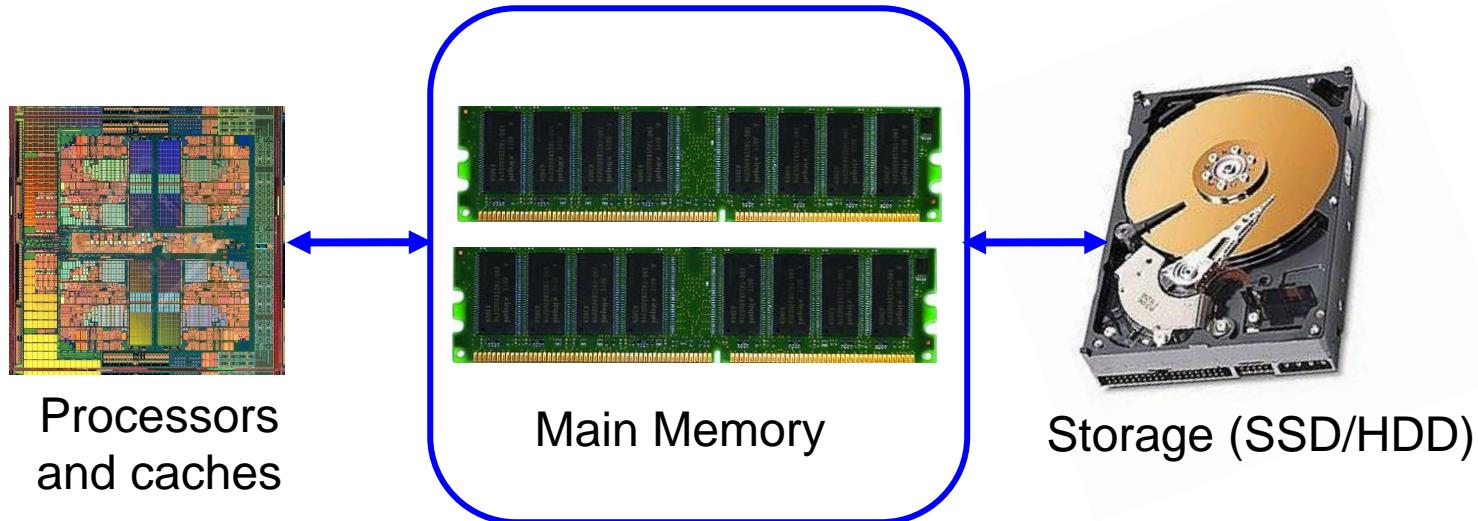
# Why Is Memory So Important? (Especially Today)

# Importance of Main Memory

---

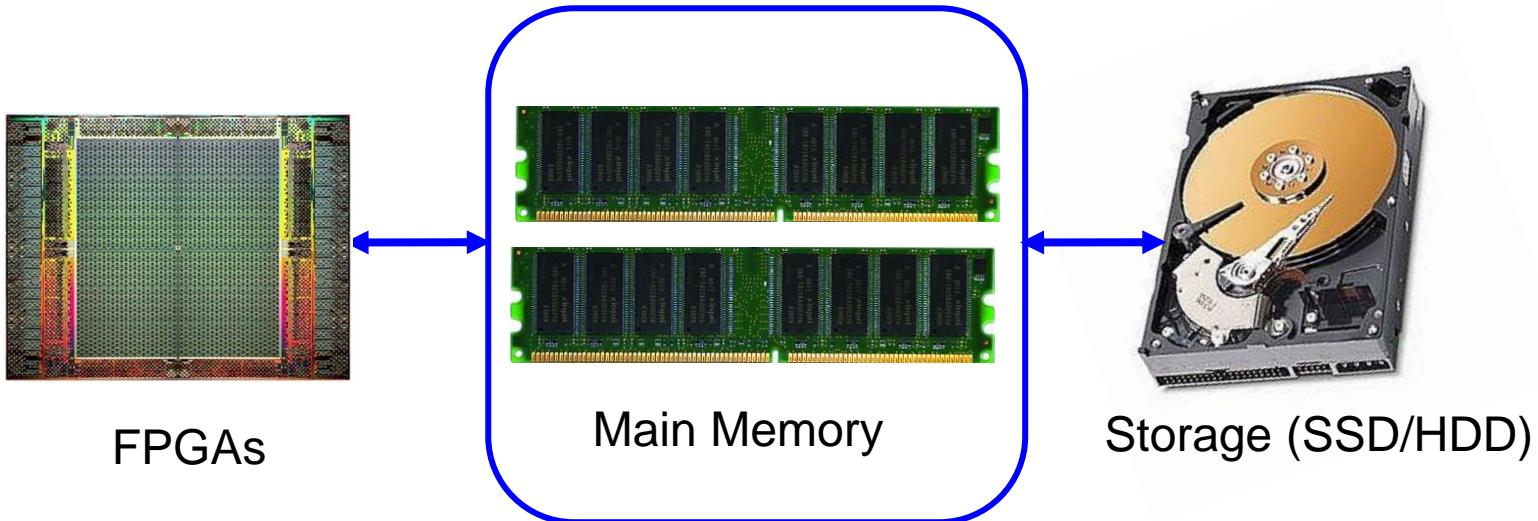
- The Performance Perspective
- The Energy Perspective
- The Scaling/Reliability/Security Perspective
- Trends/Challenges/Opportunities in Main Memory

# The Main Memory System



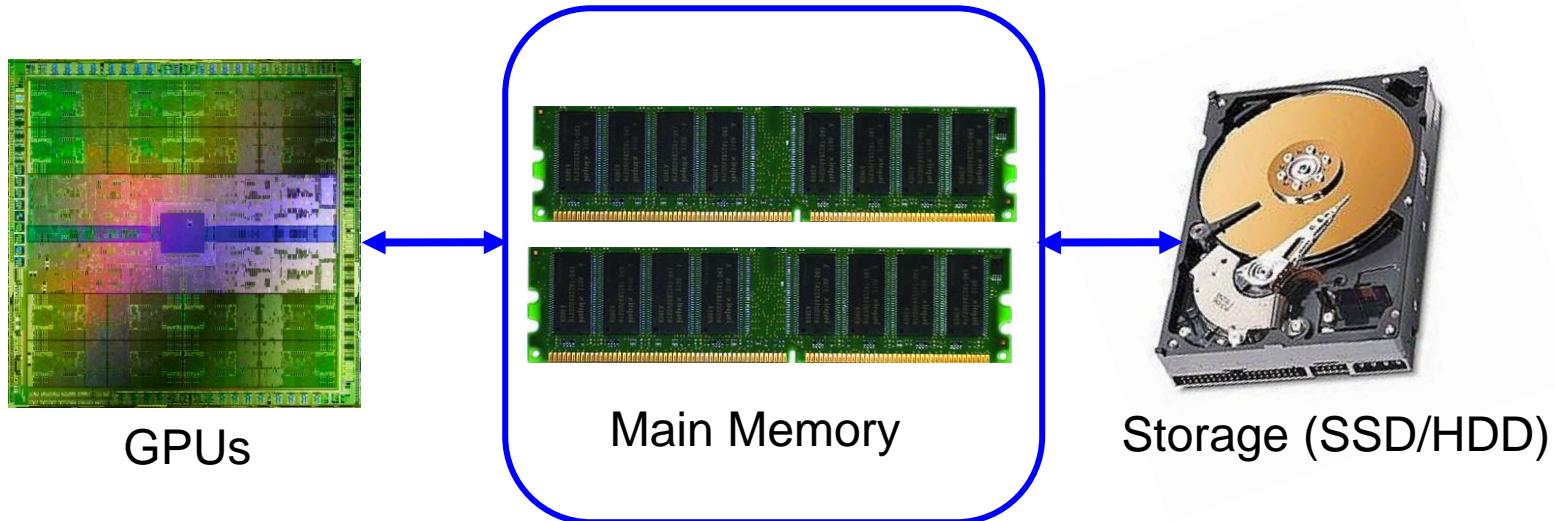
- Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor
- Main memory system must scale (*in size, technology, efficiency, cost, and management algorithms*) to maintain performance growth and technology scaling benefits

# The Main Memory System



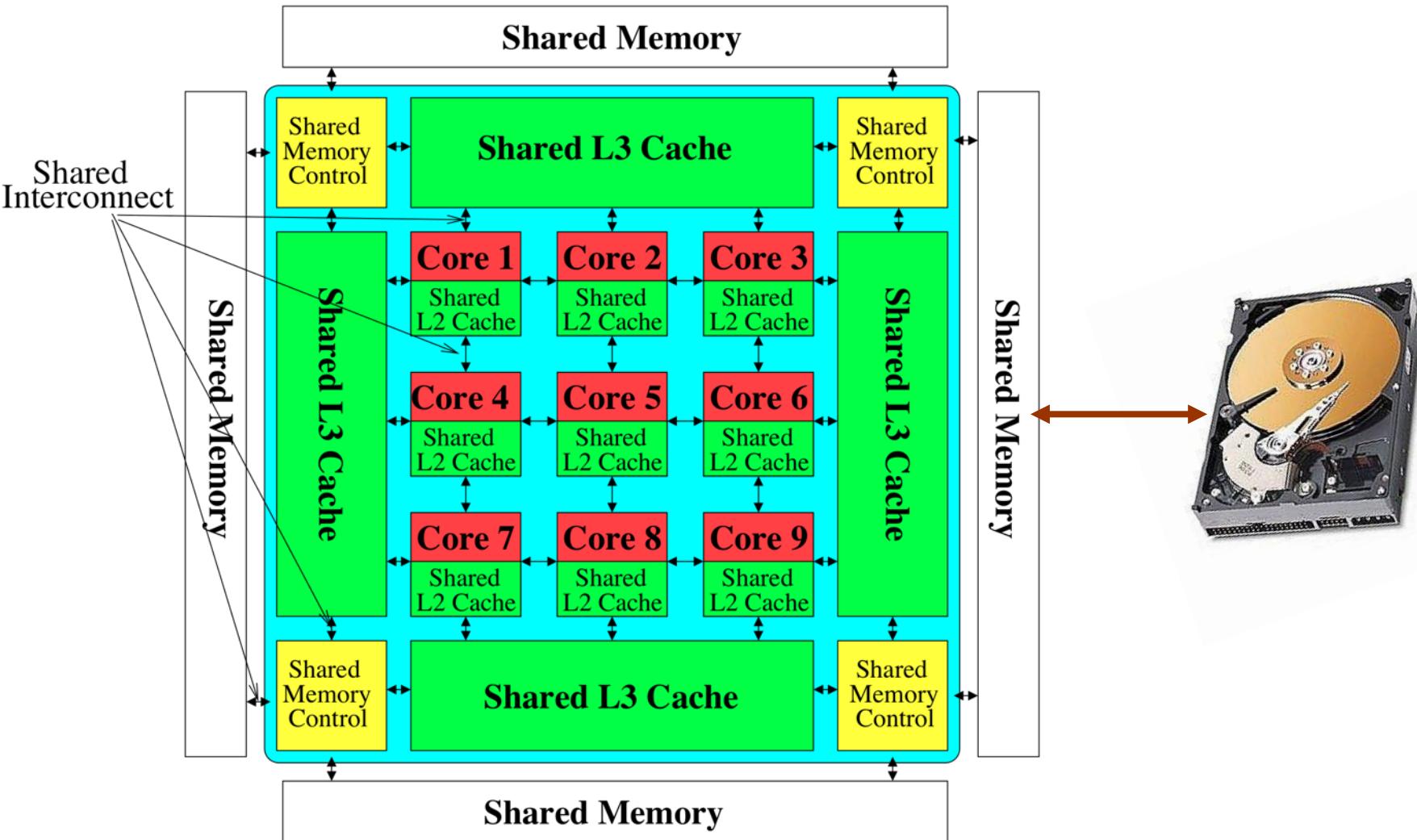
- Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor
- Main memory system must scale (*in size, technology, efficiency, cost, and management algorithms*) to maintain performance growth and technology scaling benefits

# The Main Memory System



- Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor
- Main memory system must scale (*in size, technology, efficiency, cost, and management algorithms*) to maintain performance growth and technology scaling benefits

# Perils of Processor-Centric Design



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

# State of the Main Memory System

---

- Recent technology, architecture, and application trends
  - lead to new requirements
  - exacerbate old requirements
- DRAM and memory controllers, as we know them today, are (will be) unlikely to satisfy all requirements
- Some emerging non-volatile memory technologies (e.g., PCM) enable new opportunities: memory+storage merging
- We need to rethink the main memory system
  - to fix DRAM issues and enable emerging technologies
  - to satisfy all requirements

# Major Trends Affecting Main Memory (I)

---

- Need for main memory capacity, bandwidth, QoS increasing
- Main memory energy/power is a key system design concern
- DRAM technology scaling is ending

# Major Trends Affecting Main Memory (II)

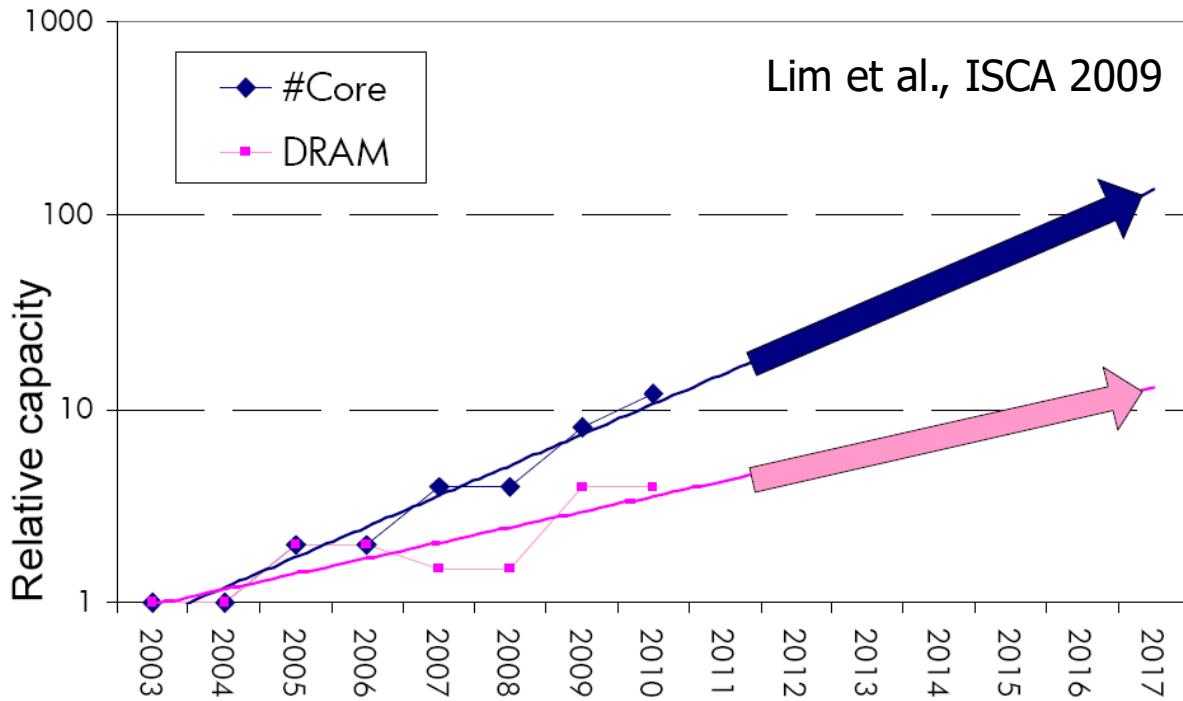
---

- Need for main memory capacity, bandwidth, QoS increasing
  - Multi-core: increasing number of cores/agents
  - Data-intensive applications: increasing demand/hunger for data
  - Consolidation: cloud computing, GPUs, mobile, heterogeneity
- Main memory energy/power is a key system design concern
- DRAM technology scaling is ending

# Consequence: The Memory Capacity Gap

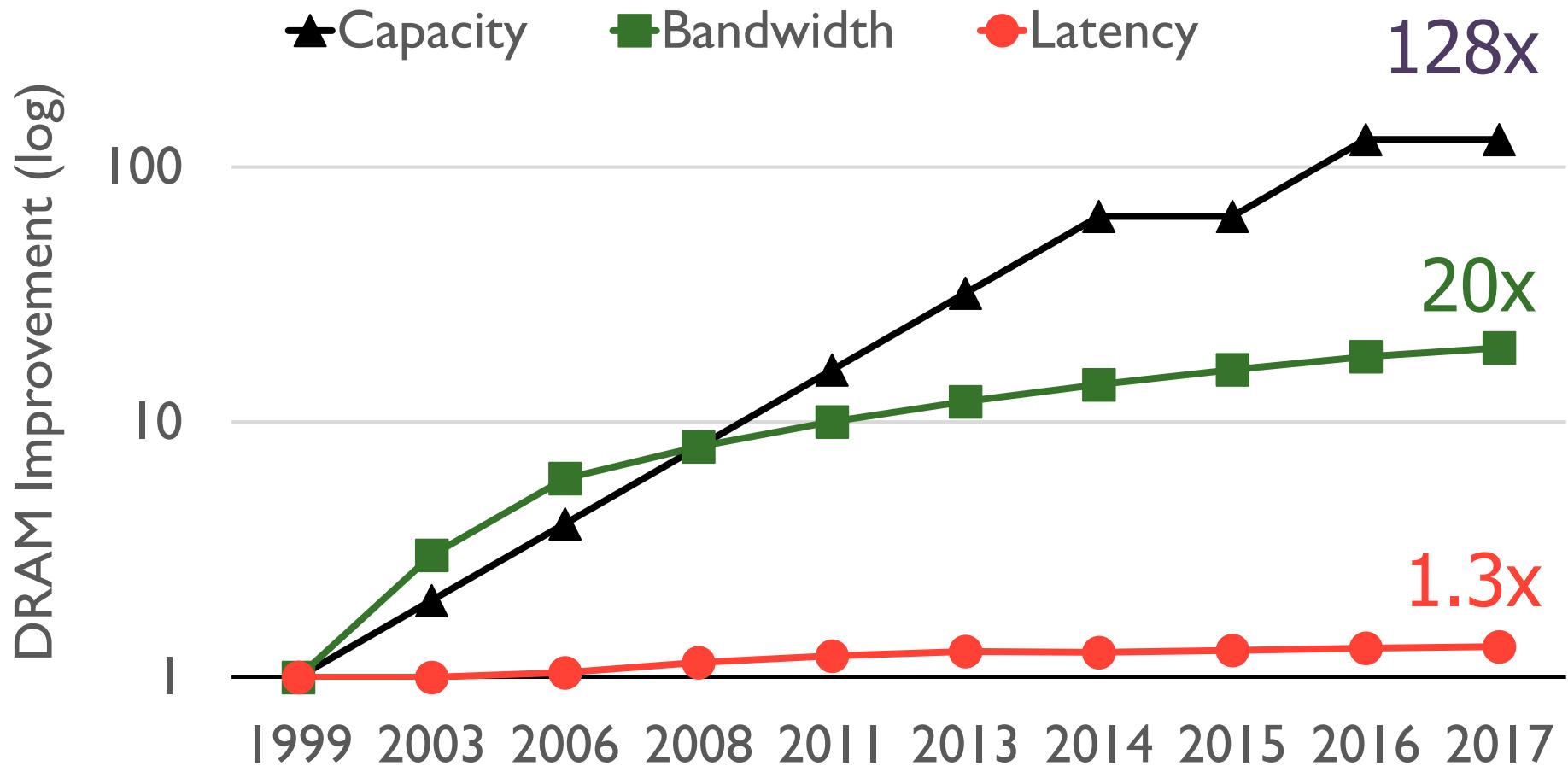
Core count doubling ~ every 2 years

DRAM DIMM capacity doubling ~ every 3 years



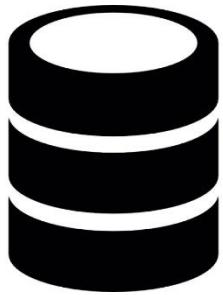
- *Memory capacity per core* expected to drop by 30% every two years
- Trends worse for *memory bandwidth per core!*

# DRAM Capacity, Bandwidth & Latency



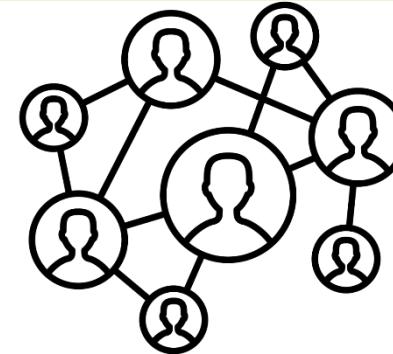
# Memory Is Critical for Performance

---



## In-memory Databases

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



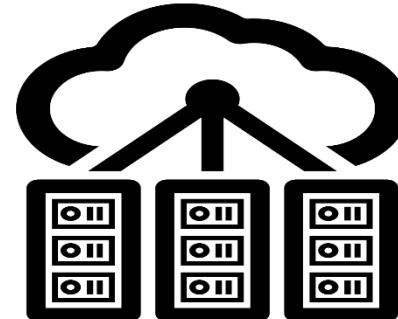
## Graph/Tree Processing

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



## In-Memory Data Analytics

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

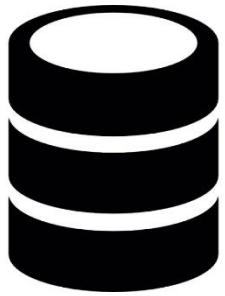


## Datacenter Workloads

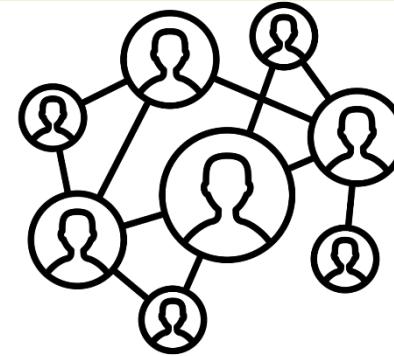
[Kanев+ (Google), ISCA'15]

# Memory Is Critical for Performance

---



**In-memory Databases**



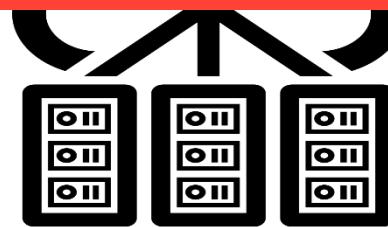
**Graph/Tree Processing**

Memory → performance bottleneck



**In-Memory Data Analytics**

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



**Datacenter Workloads**

[Kanев+ (Google), ISCA'15]

# Memory Is Critical for Performance



**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**

# Memory Is Critical for Performance



**Chrome**



**TensorFlow Mobile**

Memory → performance bottleneck



**Video Playback**

Google's **video codec**



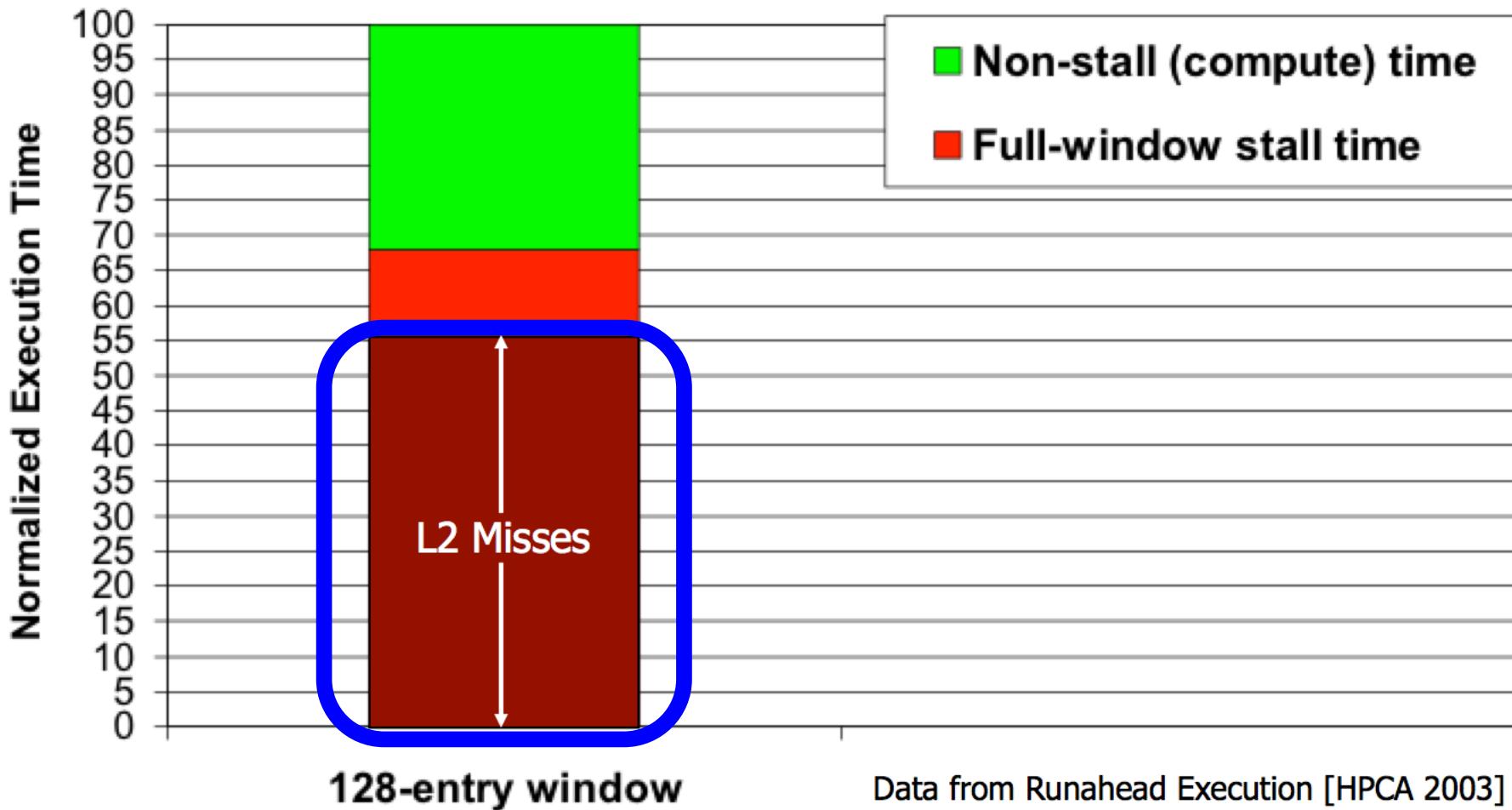
**Video Capture**

Google's **video codec**

# Memory Bottleneck

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 Memory Bottleneck

---

- 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.

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

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

[[Lecture Video \(1 hr 54 mins\)](#)]

[[Retrospective HPCA Test of Time Award Talk Slides \(pptx\) \(pdf\)](#)]

[[Retrospective HPCA Test of Time Award Talk Video \(14 minutes\)](#)]

***One of the 15 computer architecture papers of 2003 selected as Top Picks by IEEE Micro.  
HPCA Test of Time Award (awarded in 2021).***

## 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](mailto:jared.w.stark@intel.com)

‡Desktop Platforms Group

Intel Corporation

[chris.wilkerson@intel.com](mailto:chris.wilkerson@intel.com)

# The Memory Bottleneck

---

- Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt,  
**"Runahead Execution: An Effective Alternative to Large Instruction Windows"**

*IEEE Micro, Special Issue: Micro's Top Picks from Microarchitecture Conferences (**MICRO TOP PICKS**)*, Vol. 23, No. 6, pages 20-25, November/December 2003.

# RUNAHEAD EXECUTION: AN EFFECTIVE ALTERNATIVE TO LARGE INSTRUCTION WINDOWS

# More on Runahead Execution (I)

Review: Runahead Execution (Mutlu et al., HPCA 2003)

*Small Window:*

*Runahead:*

18

◀ ▶ ⏪ ⏩ 🔍 40:36 / 1:32:51

CC ⚙️ 🗑️ 🗒️ 🗑️

Computer Architecture - Lecture 19a: Execution-Based Prefetching (ETH Zürich, Fall 2020)

395 views • Nov 29, 2020

14 0 SHARE SAVE ...



Onur Mutlu Lectures  
16.5K subscribers

ANALYTICS EDIT VIDEO

# More on Runahead Execution (II)

## Runahead Execution in NVIDIA Denver

Reducing the effects of long cache-miss penalties has been a major focus of the micro-architecture, using techniques like prefetching and run-ahead. An aggressive hardware prefetcher implementation detects L2 cache requests and tracks up to 32 streams, each with complex stride patterns.

Run-ahead uses the idle time that a CPU spends waiting on a long latency operation to discover cache and DTLB misses further down the instruction stream and generates prefetch requests for these misses.<sup>1</sup> These prefetch requests warm up the data cache and DTLB well before the actual execution of the instructions that require the data. Run-ahead complements the hardware prefetcher because it's better at prefetching nonstrided streams, and it trains the hardware prefetcher faster than normal execution to yield a combined benefit of 13 percent on SPECint2000 and up to 60 percent on SPECfp2000.

Boggs+, "Denver: NVIDIA's First 64-Bit ARM Processor," IEEE Micro 2015.

Gwennap, "NVIDIA's First CPU is a Winner," MPR 2014.

The core includes a hardware prefetch unit that Boggs describes as "aggressive" in preloading the data cache but less aggressive in preloading the instruction cache. It also implements a "run-ahead" feature that continues to execute microcode speculatively after a data-cache miss; this execution can trigger additional cache misses that resolve in the shadow of the first miss. Once the data from the original miss returns, the results of this speculative execution are discarded and execution restarts with the bundle containing the original miss, but run-ahead can preload subsequent data into the cache, thus avoiding a string of time-wasting cache misses. These and other features help Denver out-score Cortex-A15 by more than 2.6x on a memory-read test even when both use the same SoC framework (Tegra K1).

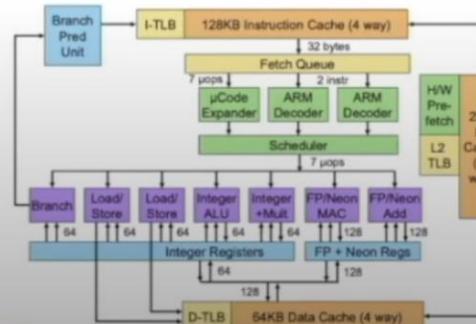
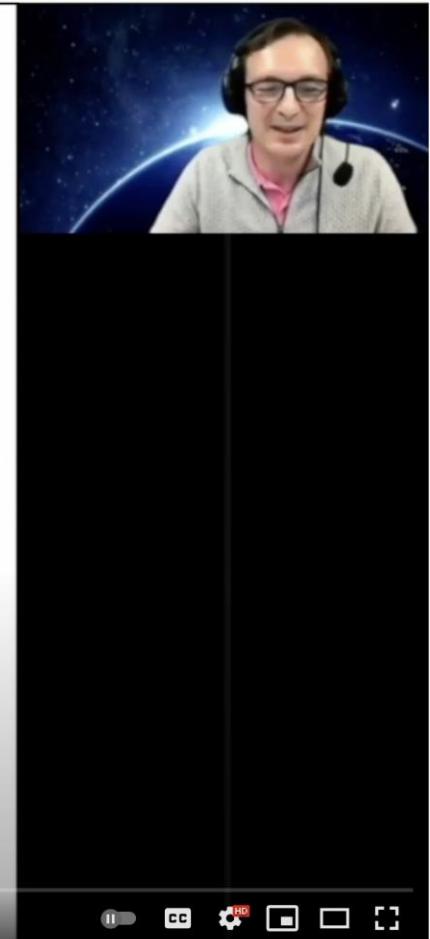


Figure 3. Denver CPU microarchitecture. This design combines a fairly



### Onur Mutlu - Runahead Execution: A Short Retrospective (HPCA Test of Time Award Talk @ HPCA 2021)

1,162 views • Premiered Mar 6, 2021



Onur Mutlu Lectures  
16.5K subscribers

ANALYTICS

EDIT VIDEO

# It's the Memory, Stupid!

---

RICHARD SITES

## It's the Memory, Stupid!

When we started the Alpha architecture design in 1988, we estimated a 25-year lifetime and a relatively modest 32% per year compounded performance improvement of implementations over that lifetime (1,000 $\times$  total). We guestimated about 10 $\times$  would come from CPU clock improvement, 10 $\times$  from multiple instruction issue, and 10 $\times$  from multiple processors.

5 , 1996



MICROPROCESSOR REPORT

# An Informal Interview on Memory

---

- Madeleine Gray and Onur Mutlu,

**"It's the memory, stupid': A conversation with Onur Mutlu"**

*HiPEAC info 55, HiPEAC Newsletter*, October 2018.

[Shorter Version in Newsletter]

[Longer Online Version with References]

'It's the memory, stupid': A conversation with Onur Mutlu

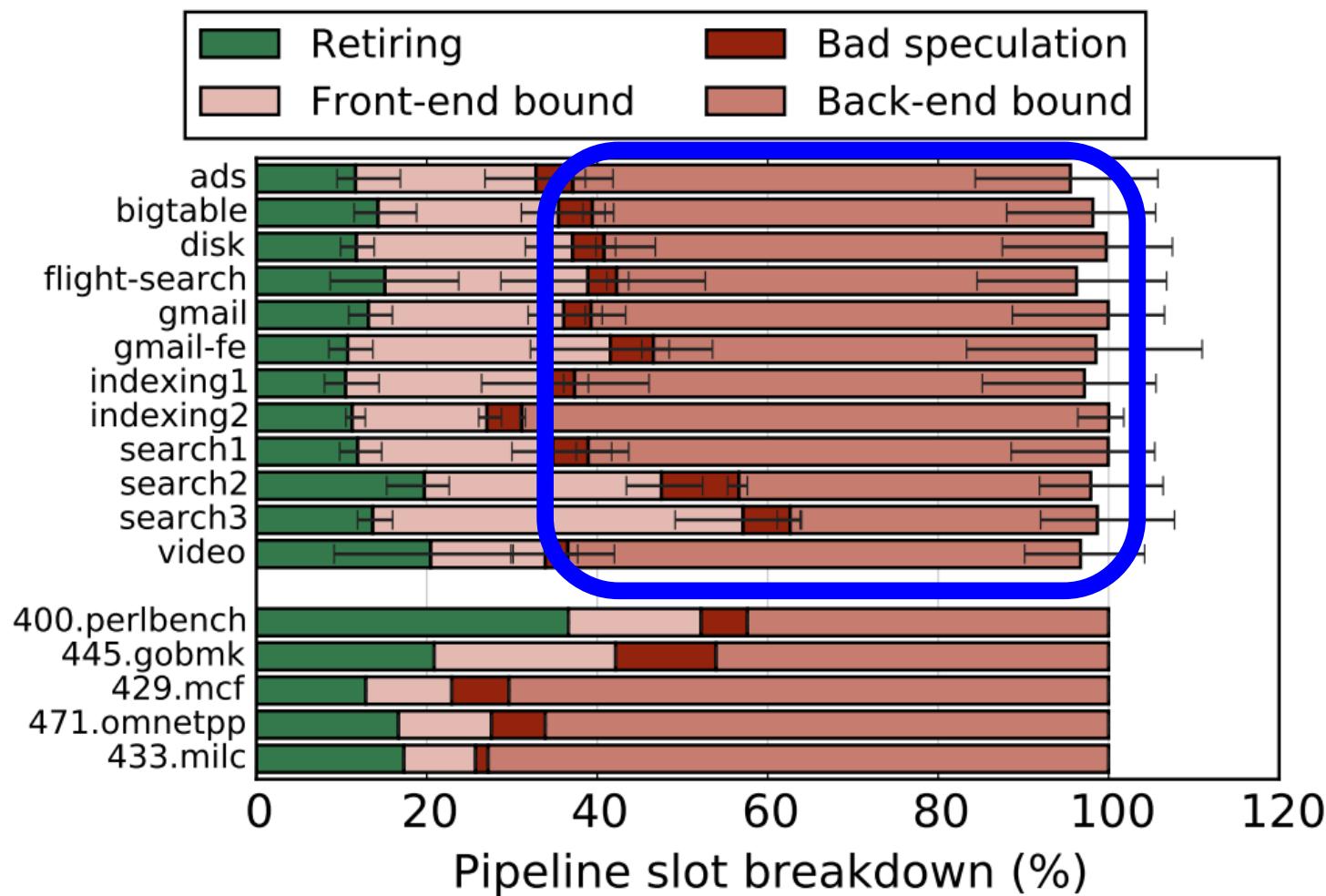
'We're beyond computation; we know how to do computation really well, we can optimize it, we can build all sorts of accelerators ... but the memory – how to feed the data, how to get the data into the accelerators – is a huge problem.'

This was how ETH Zürich and Carnegie Mellon Professor Onur Mutlu opened his course on memory systems and memory-centric computing systems at HiPEAC's summer school, ACACES18. A prolific publisher – he recently bagged the top spot on the International Symposium on Computer Architecture (ISCA) hall of fame – Onur is passionate about computation and communication that are efficient and secure by design. In advance of our Computing Systems Week focusing on data centres, storage, and networking, which takes place next week in Heraklion, HiPEAC picked his brains on all things data-based.



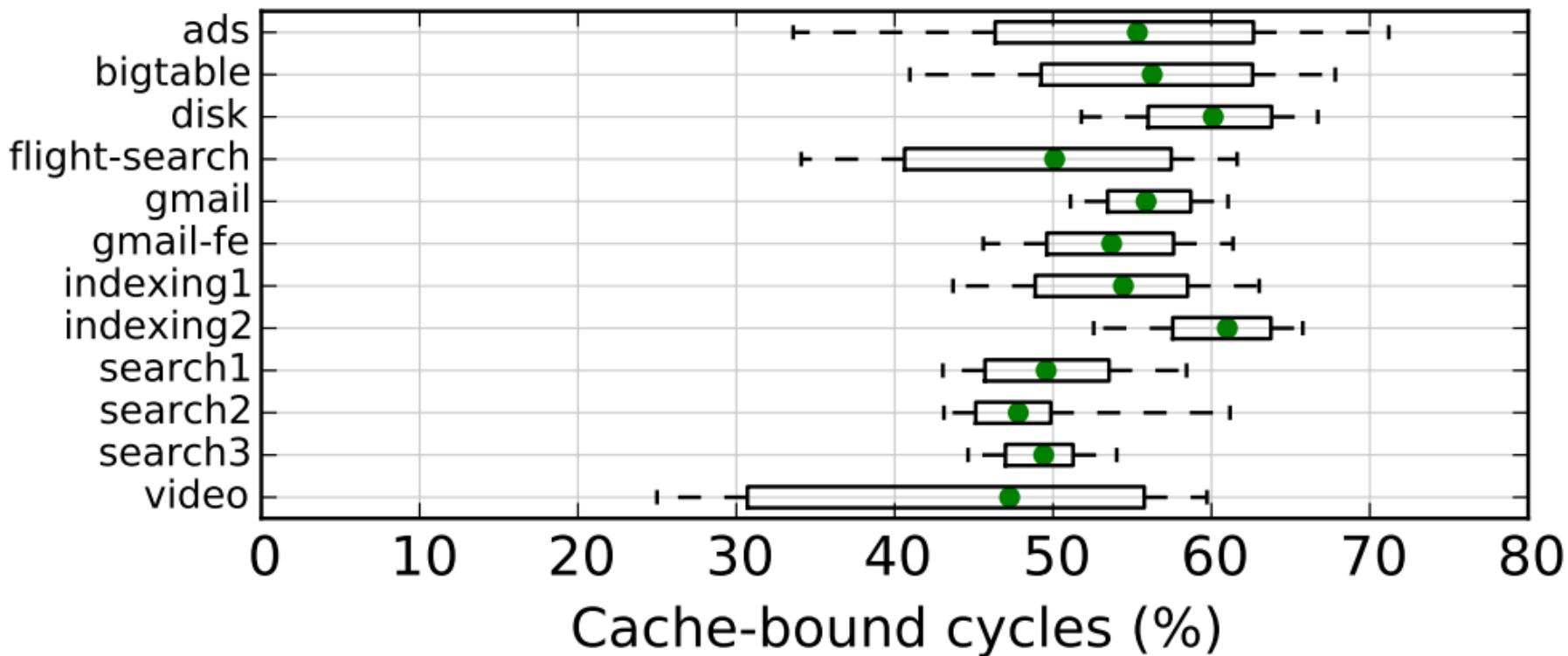
# The Memory Bottleneck

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



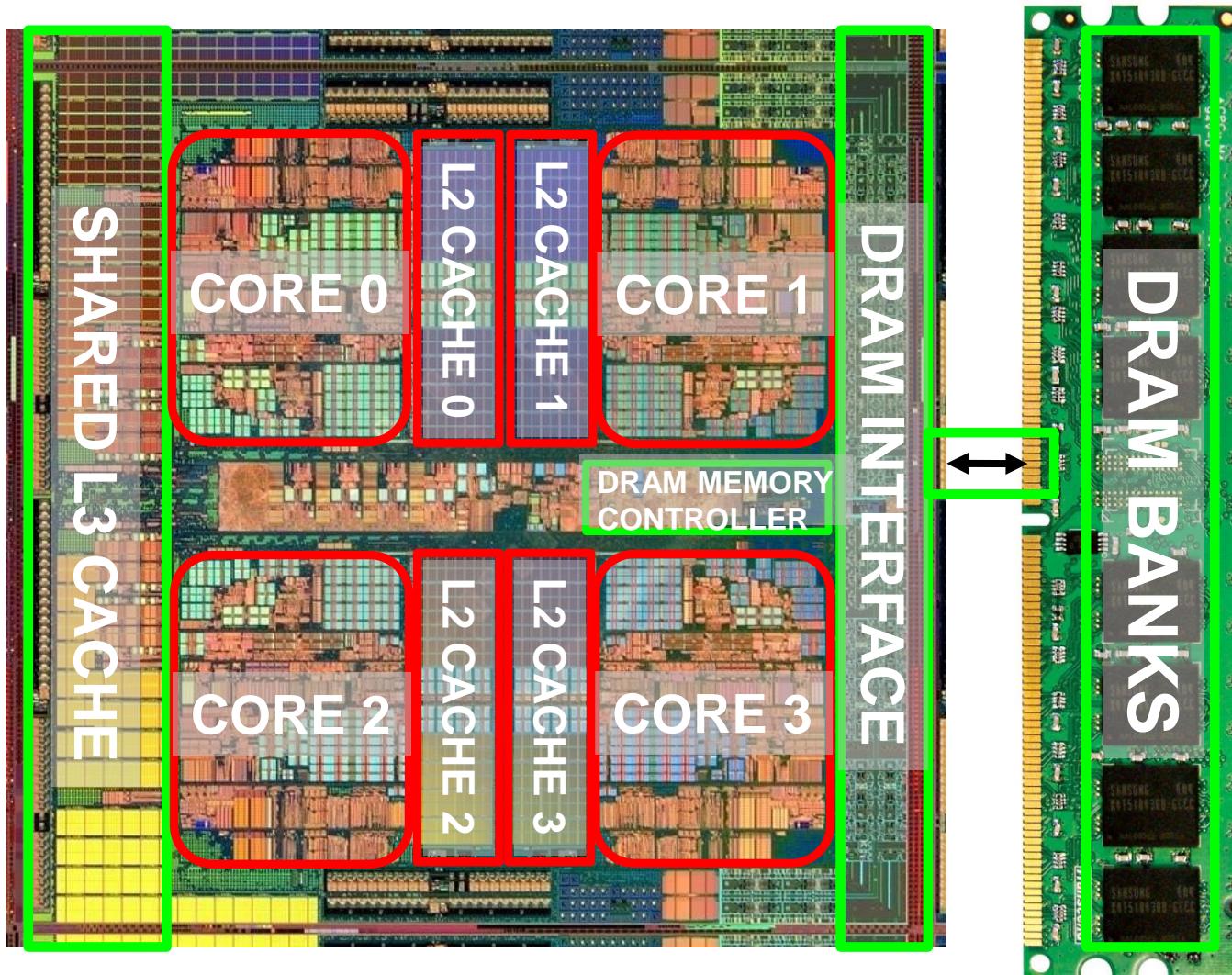
# The Memory Bottleneck

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



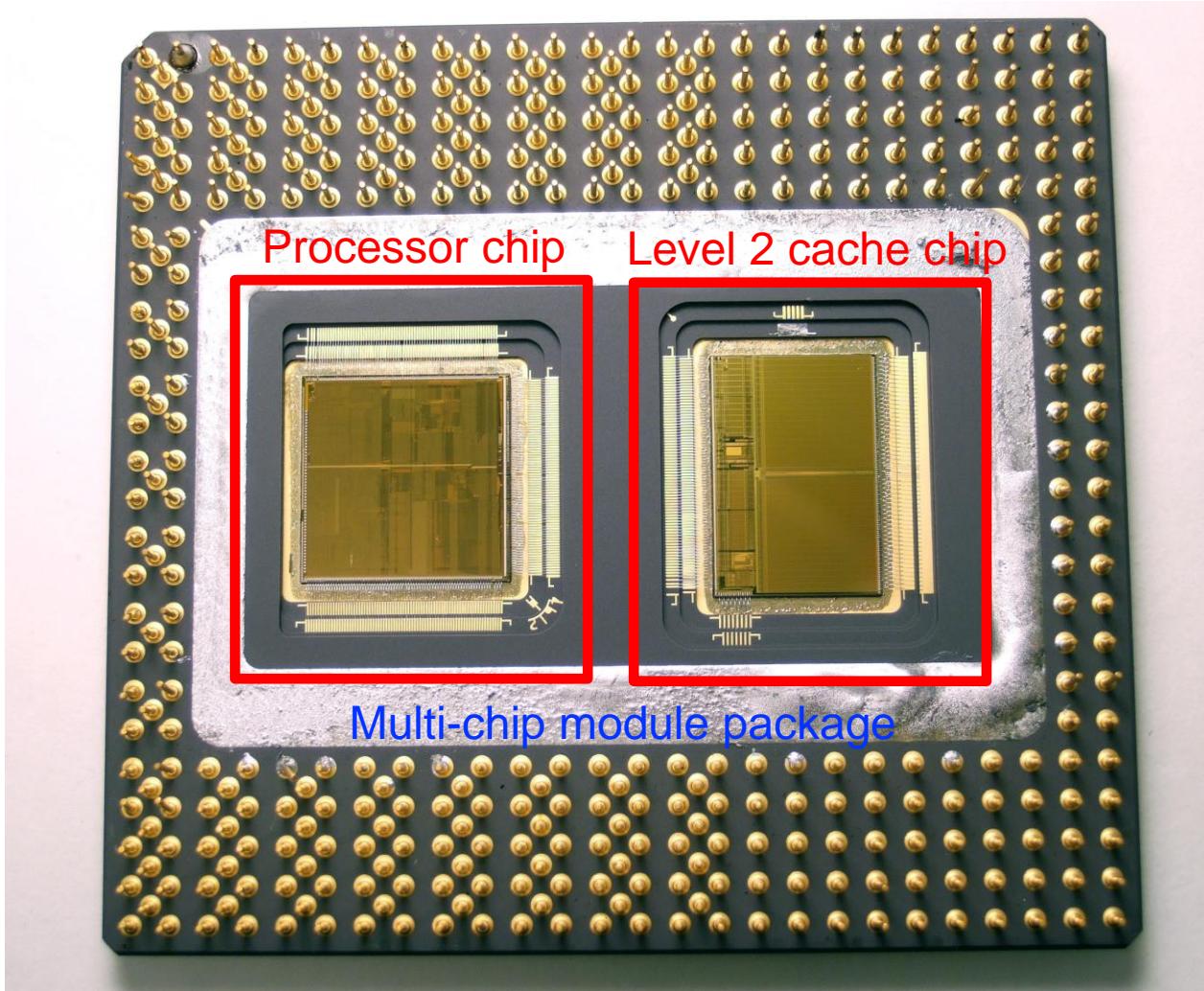
**Figure 11: Half of cycles are spent stalled on caches.**

# Memory in a Modern System



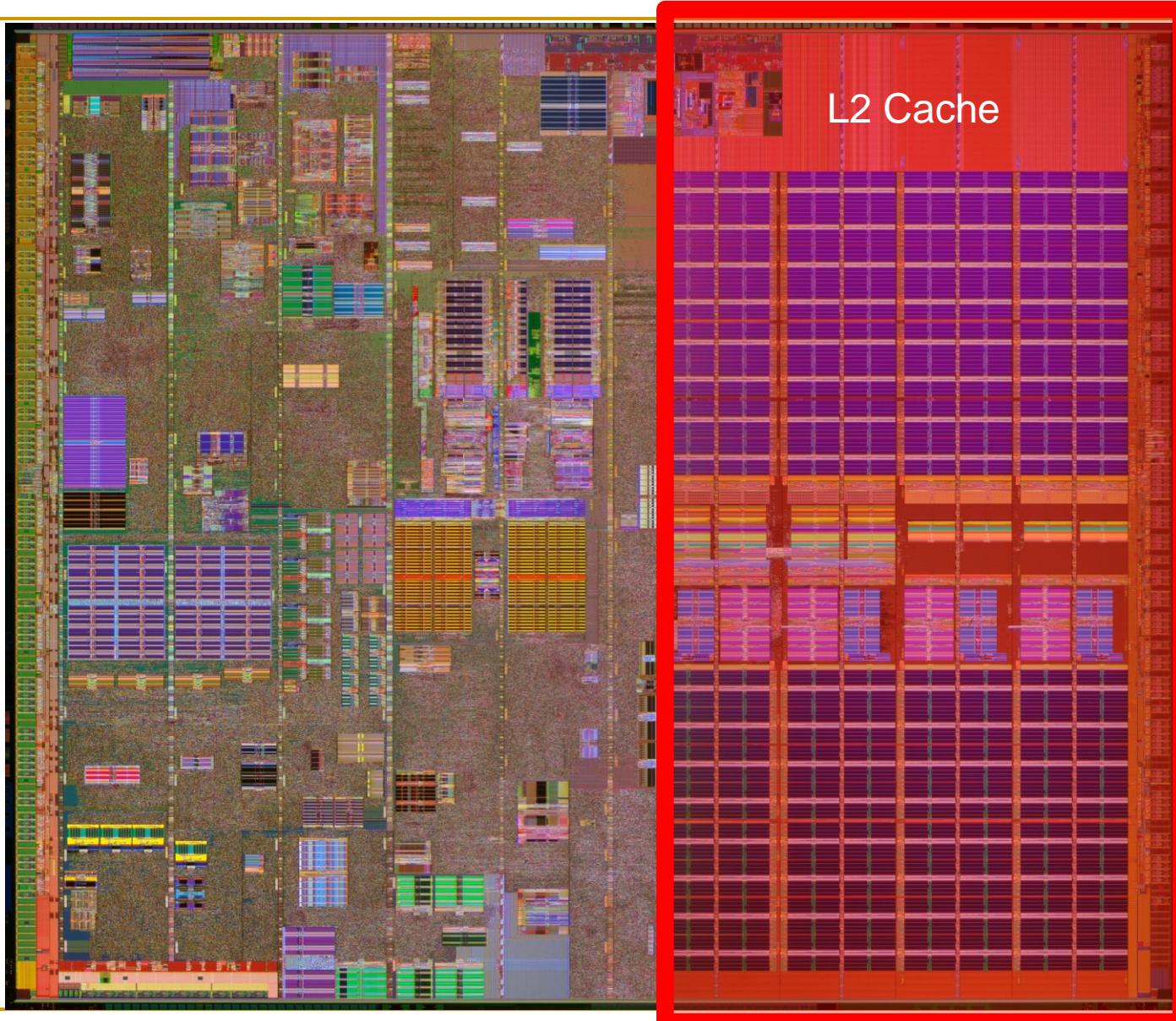
AMD Barcelona, 2006

# A Large Fraction of Modern Systems is Memory

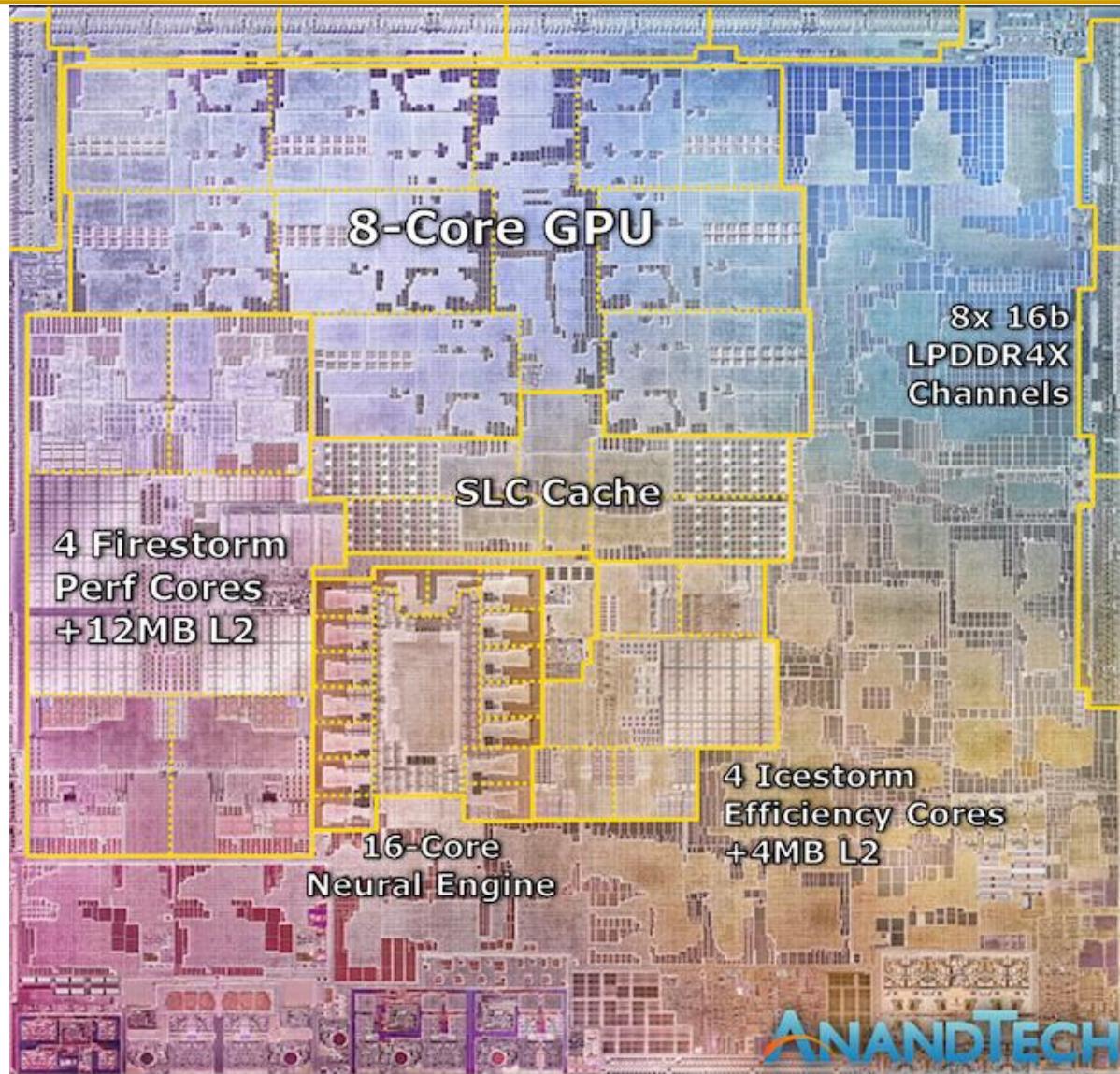


Intel Pentium Pro, 1995

# A Large Fraction of Modern Systems is Memory

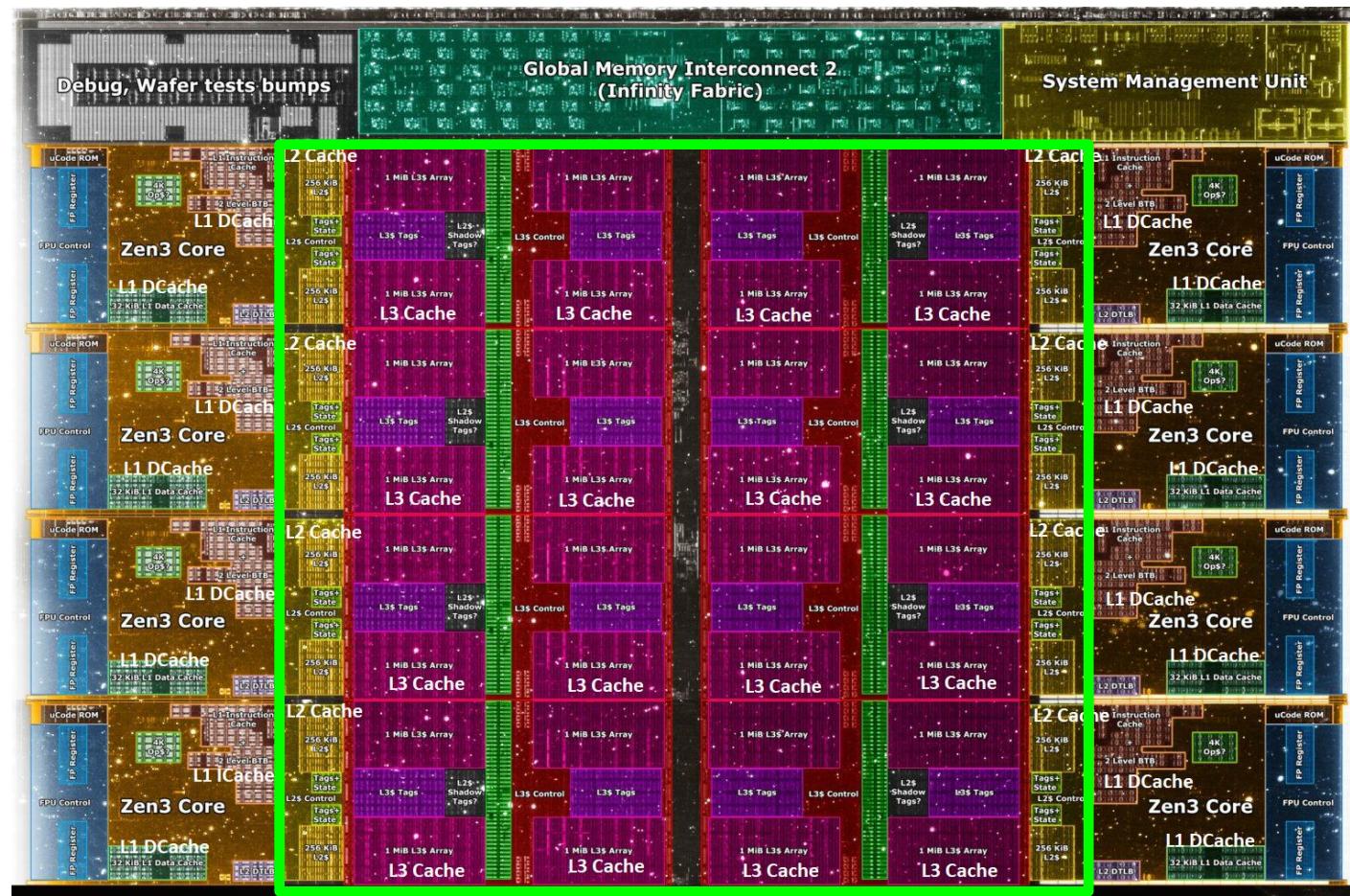


# Deeper and Larger Cache Hierarchies



Apple M1,  
2021

# Deeper and Larger Cache Hierarchies



Core Count:  
8 cores/16 threads

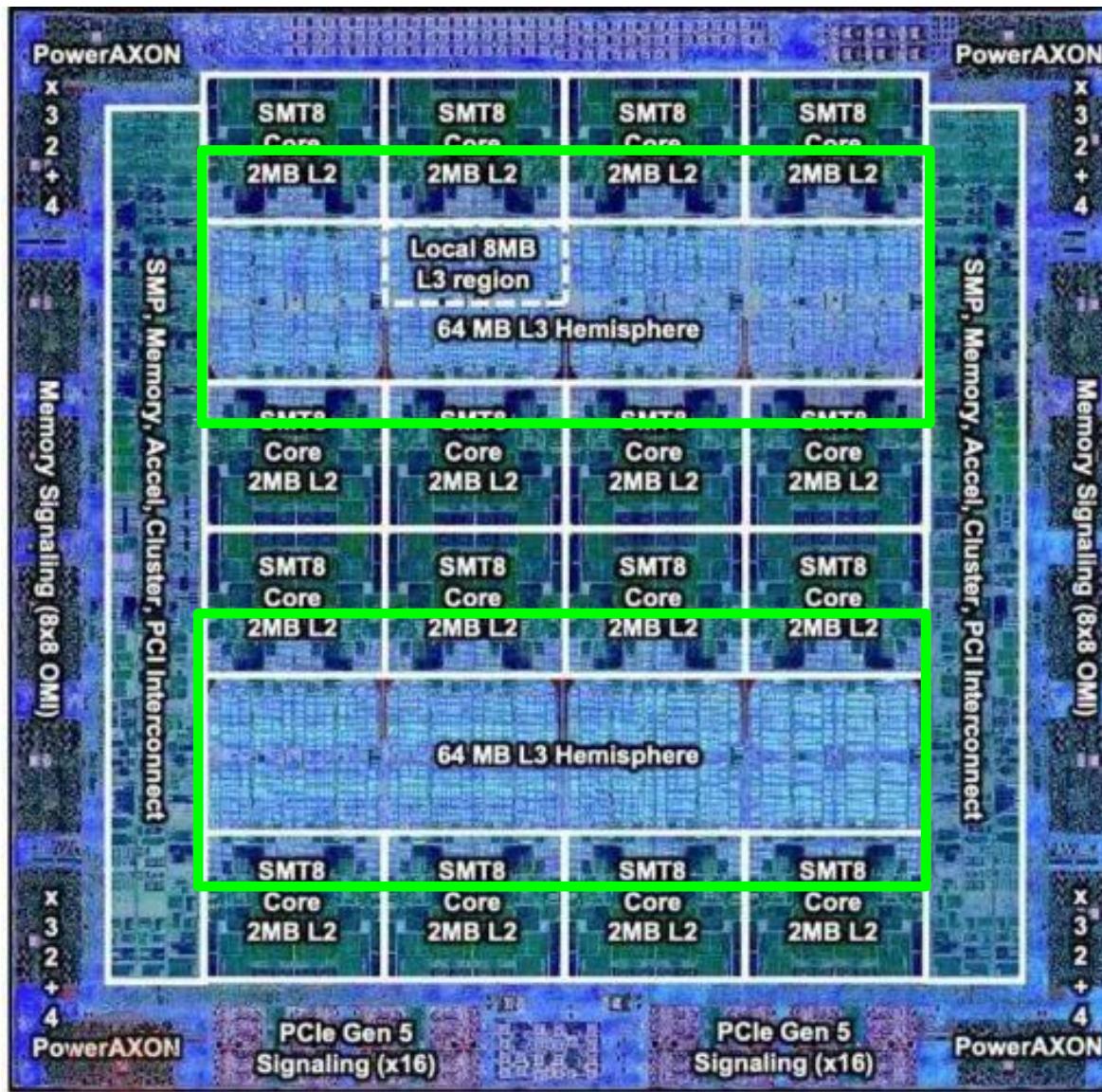
L1 Caches:  
32 KB per core

L2 Caches:  
512 KB per core

L3 Cache:  
32 MB shared

AMD Ryzen 5000, 2020

# Deeper and Larger Cache Hierarchies



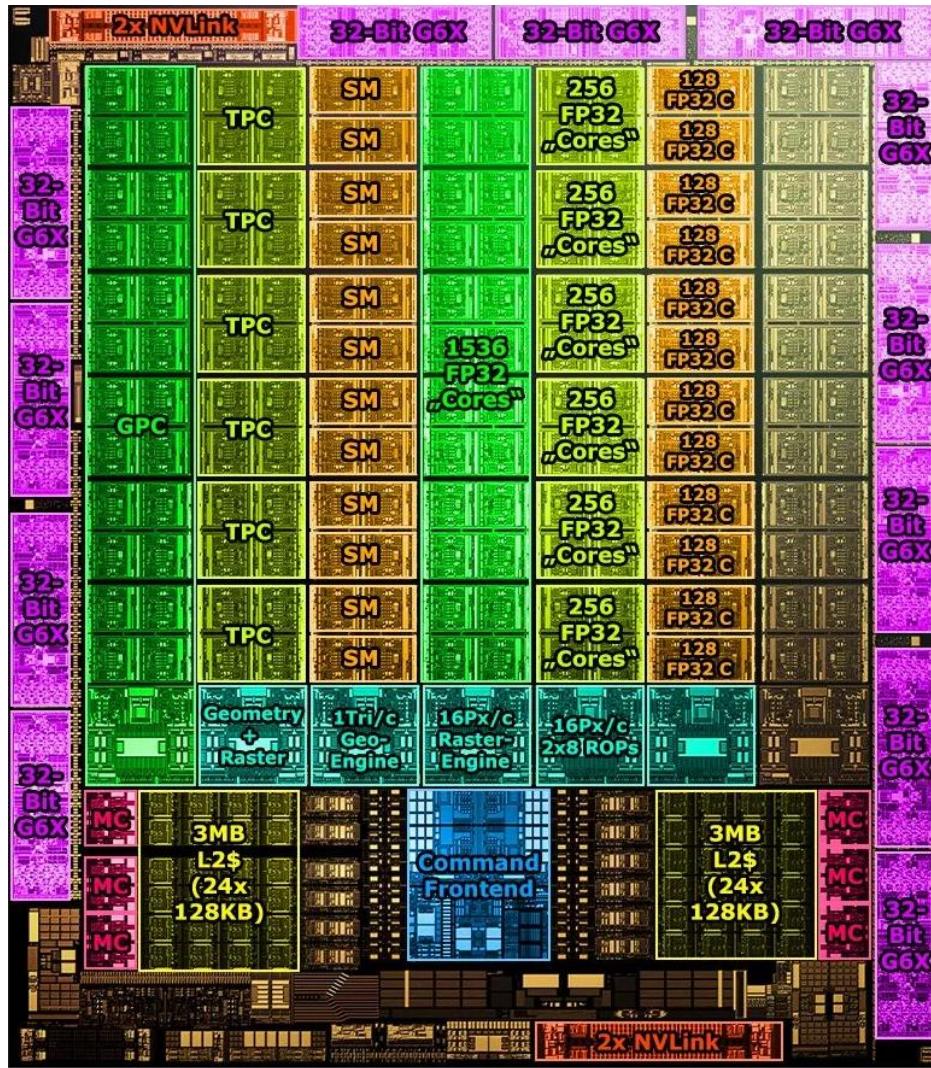
IBM POWER10,  
2020

Cores:  
15-16 cores,  
8 threads/core

L2 Caches:  
2 MB per core

L3 Cache:  
120 MB shared

# Deeper and Larger Cache Hierarchies



Nvidia Ampere, 2020

Cores:

128 Streaming Multiprocessors

L1 Cache or  
Scratchpad:

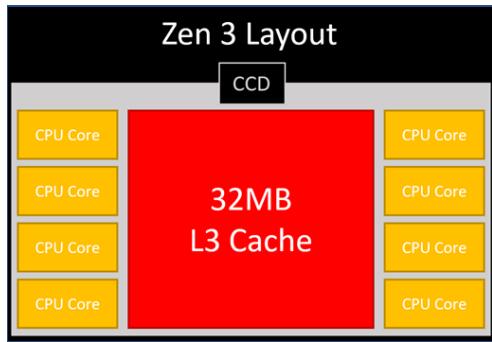
192KB per SM

Can be used as L1 Cache  
and/or Scratchpad

L2 Cache:

40 MB shared

# AMD's 3D Last Level Cache (2021)

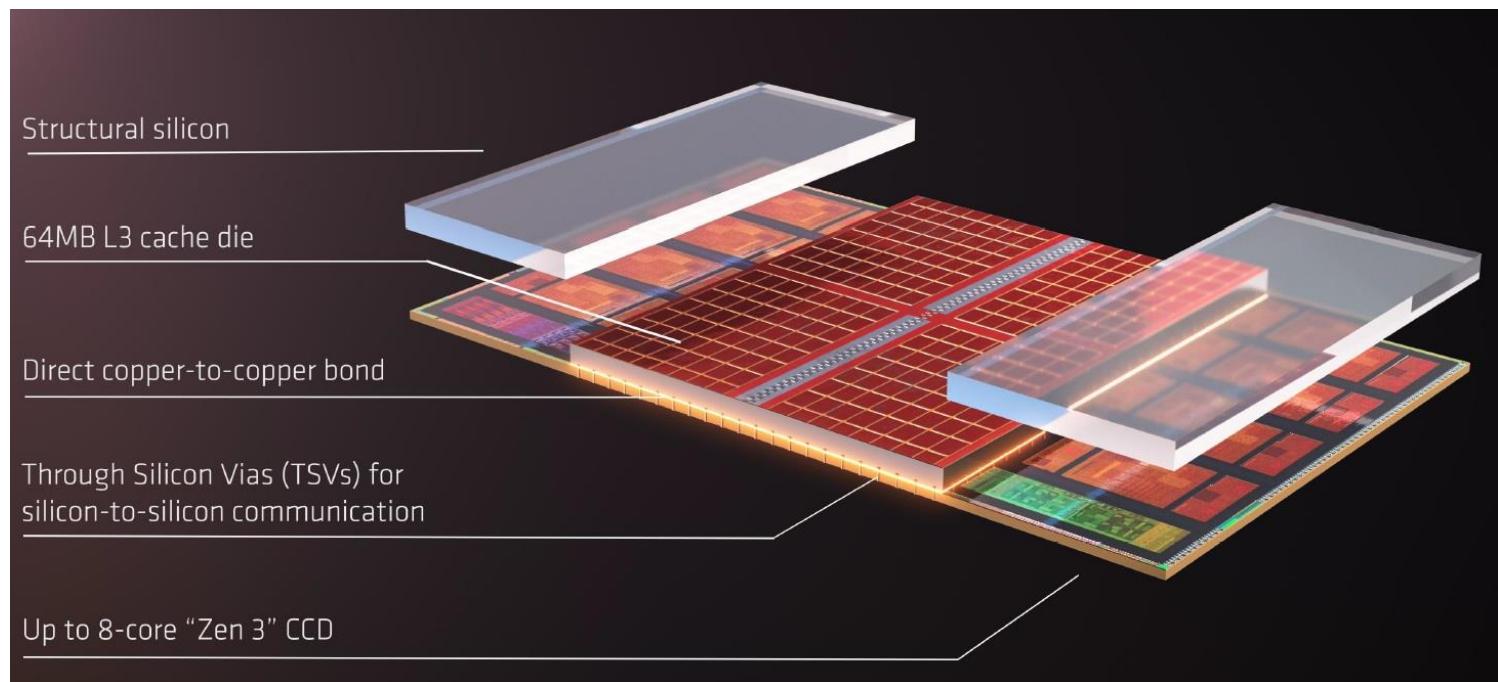


<https://community.microcenter.com/discussion/5134/comparing-zen-3-to-zen-2>

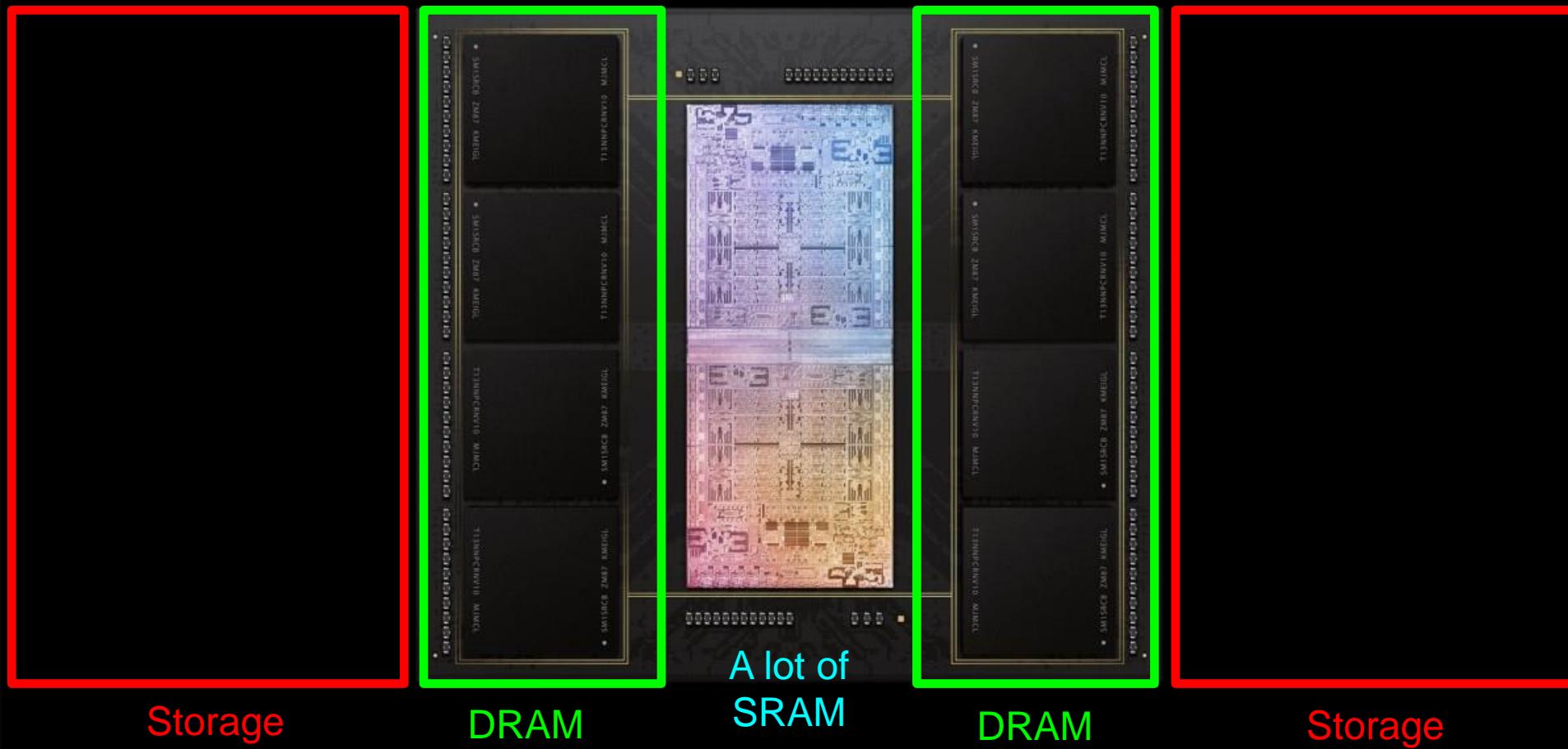
AMD increases the L3 size of their 8-core Zen 3 processors from 32 MB to 96 MB

**Additional 64 MB L3 cache die stacked on top of the processor die**

- Connected using Through Silicon Vias (TSVs)
- Total of 96 MB L3 cache

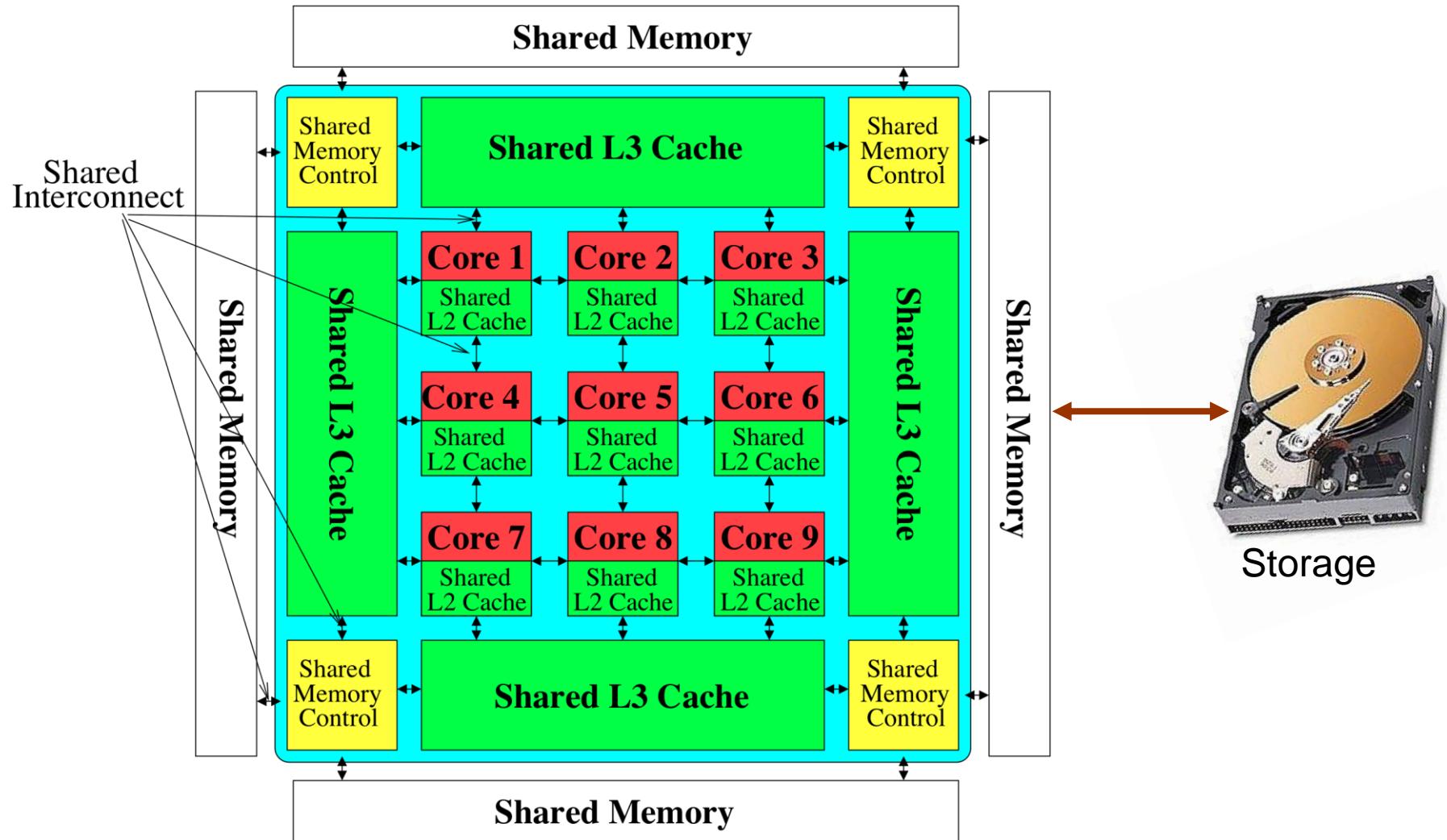


# Deeper and Larger Memory Hierarchies



Apple M1 Ultra System (2022)

# Memory System: Most of the Platform



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

# Major Trends Affecting Main Memory (III)

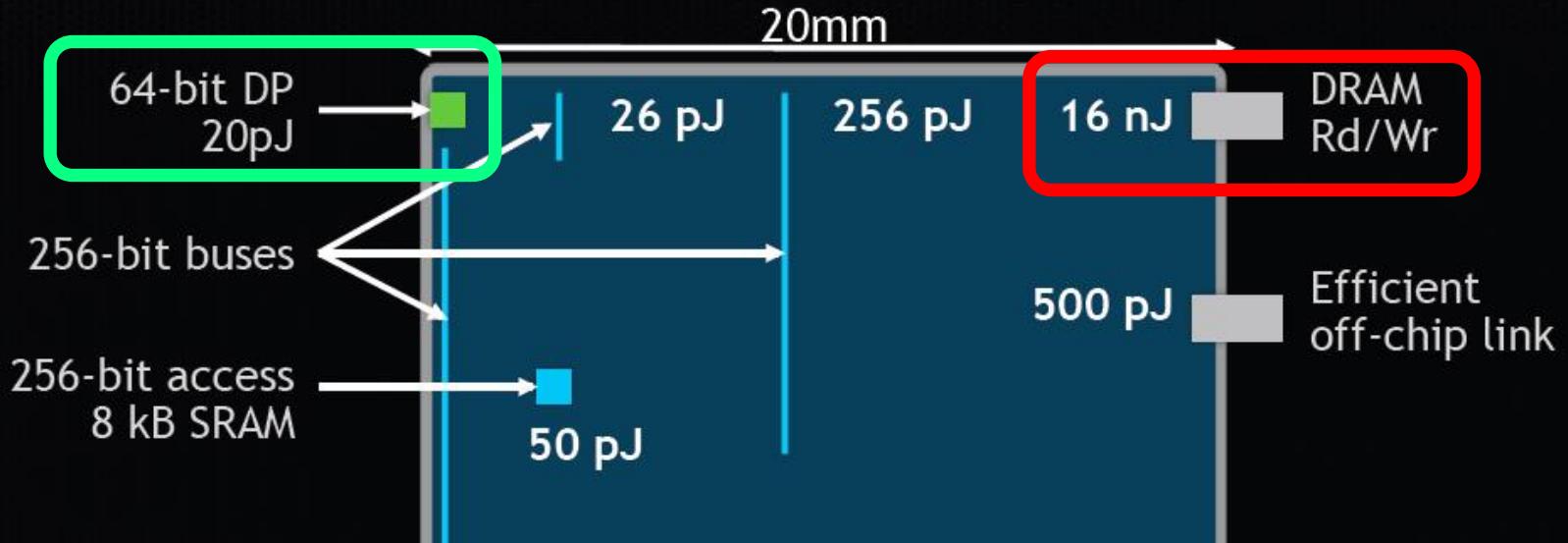
---

- Need for main memory capacity, bandwidth, QoS increasing
- Main memory energy/power is a key system design concern
  - ~40-50% energy spent in off-chip memory hierarchy [Lefurgy, IEEE Computer'03] >40% power in DRAM [Ware, HPCA'10][Paul,ISCA'15]
  - DRAM consumes power even when not used (periodic refresh)
- DRAM technology scaling is ending

# Data Movement vs. Computation Energy

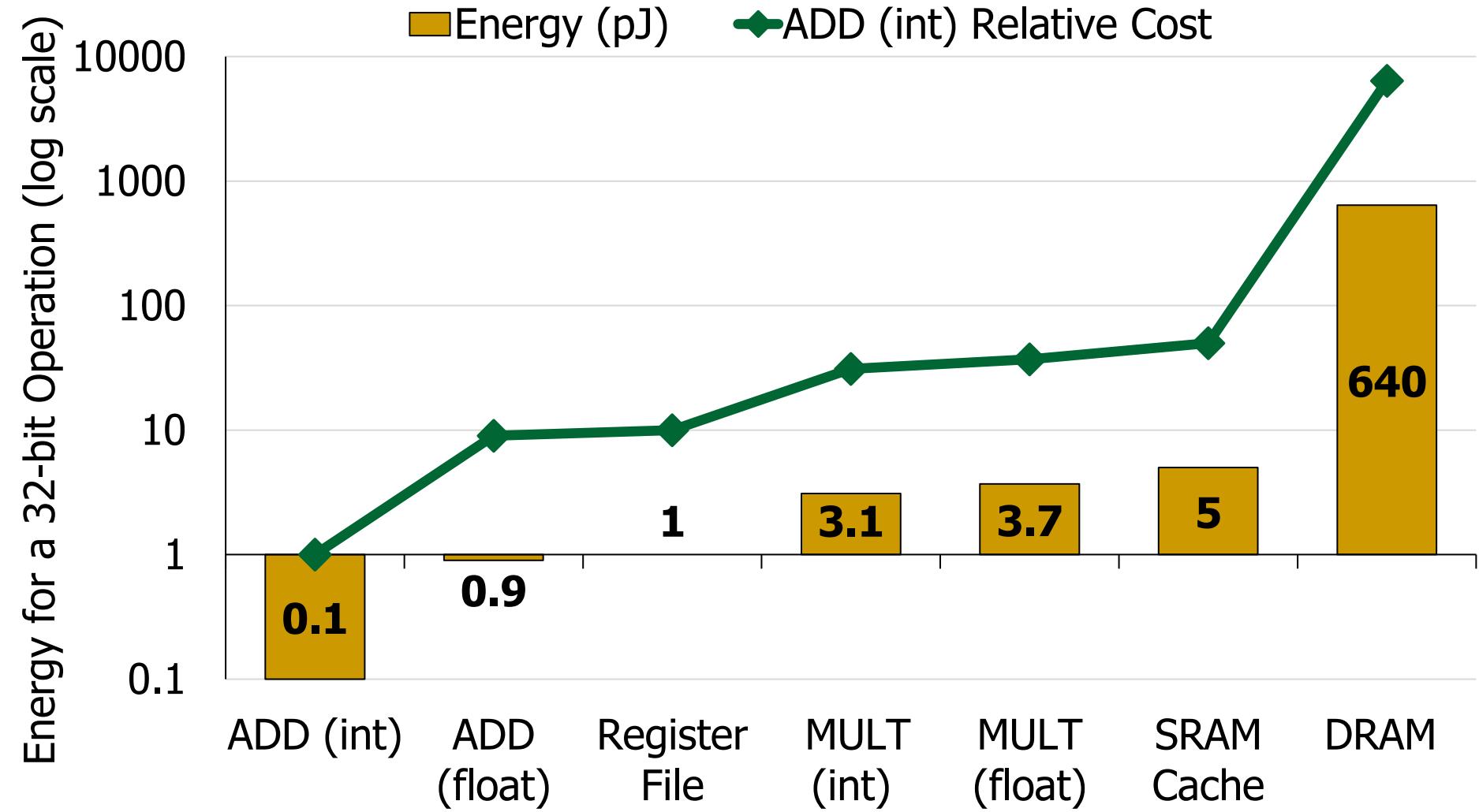
## Communication Dominates Arithmetic

Dally, HiPEAC 2015

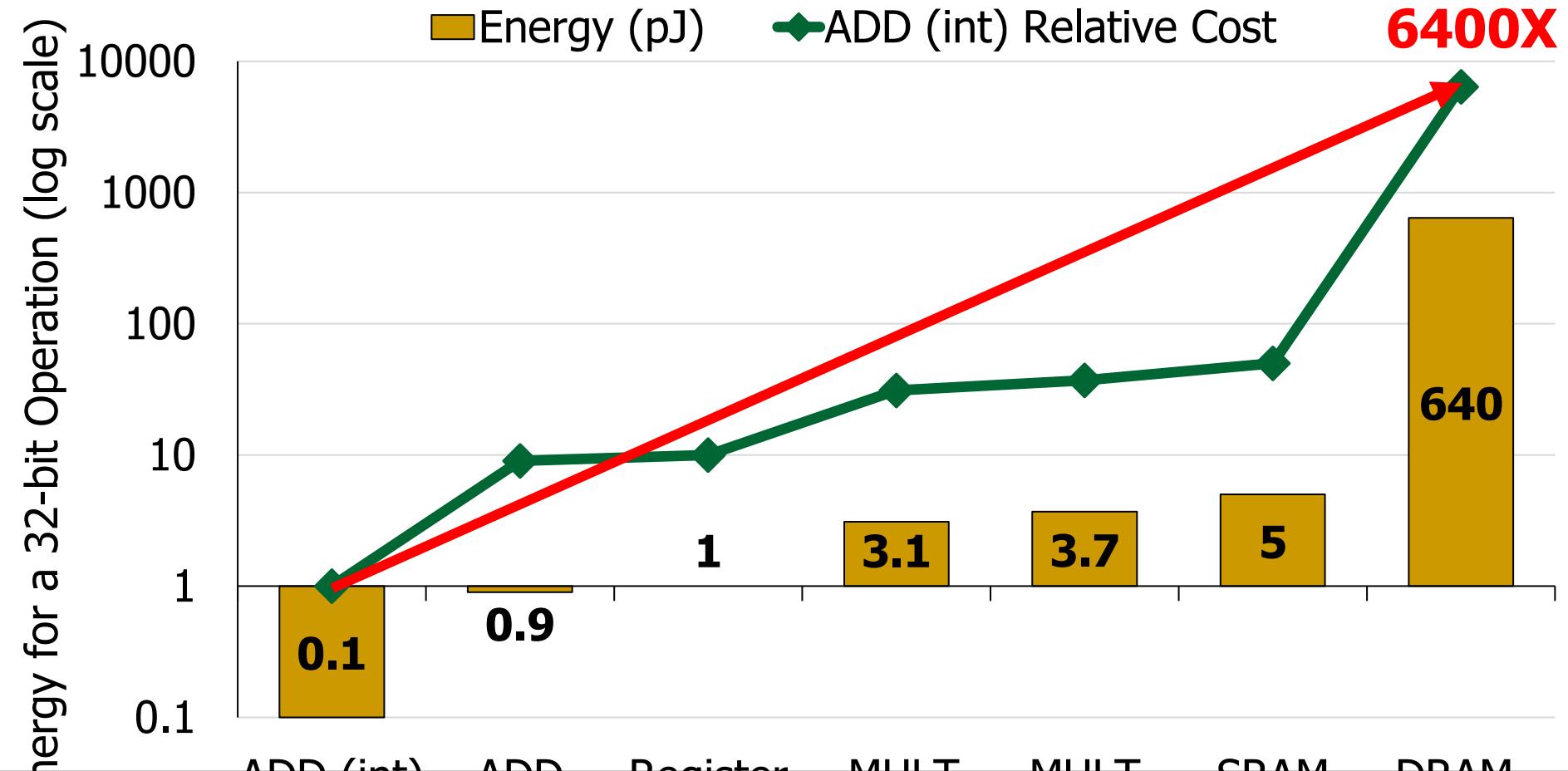


A memory access consumes  $\sim$ 100-1000X  
the energy of a complex addition

# Data Movement vs. Computation Energy



# Data Movement vs. Computation Energy



A memory access consumes 6400X  
the energy of a simple integer addition

# 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>

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>

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

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

# Major Trends Affecting Main Memory (IV)

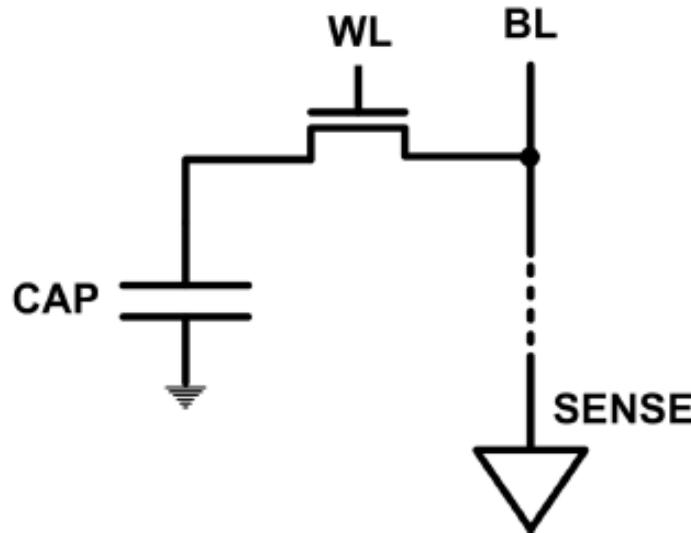
---

- Need for main memory capacity, bandwidth, QoS increasing
- Main memory energy/power is a key system design concern
- DRAM technology scaling is ending
  - ITRS projects DRAM will not scale easily below X nm
  - Scaling has provided many benefits:
    - higher capacity (density), lower cost, lower energy

# The DRAM Scaling Problem

---

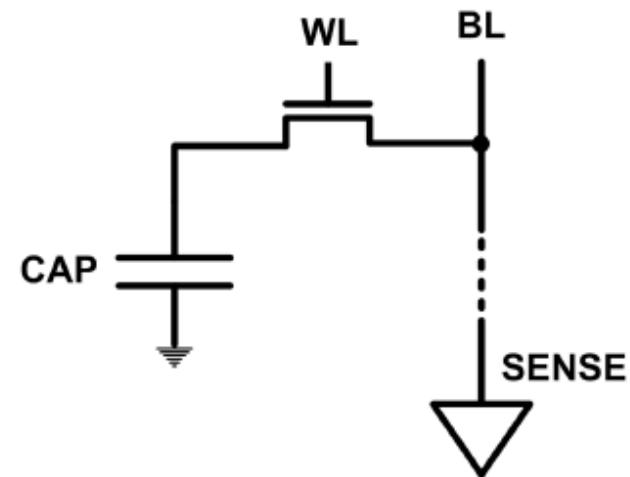
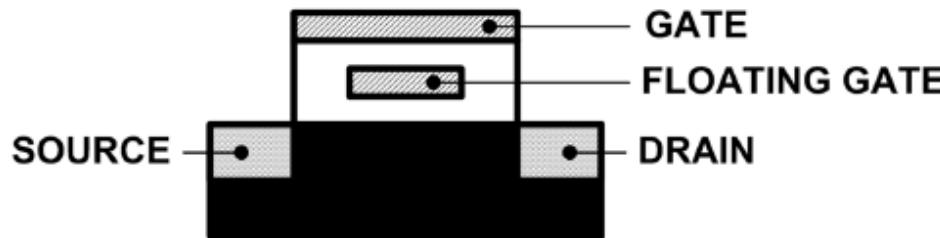
- DRAM stores charge in a capacitor (charge-based memory)
  - Capacitor must be large enough for reliable sensing
  - Access transistor should be large enough for low leakage and high retention time
  - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]



- DRAM capacity, cost, and energy/power hard to scale

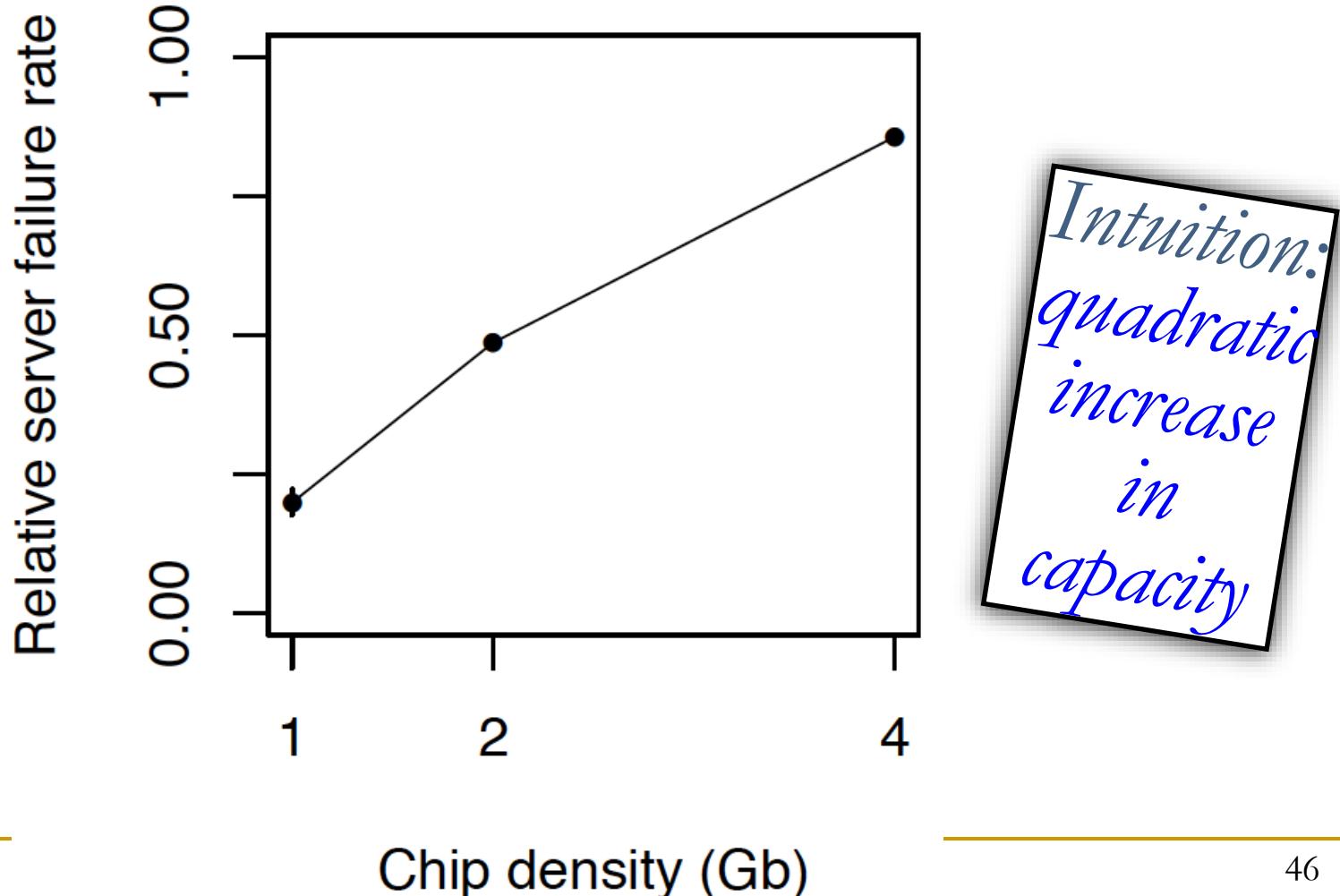
# Limits of Charge Memory

- Difficult charge placement and control
  - Flash: floating gate charge
  - DRAM: capacitor charge, transistor leakage
- Data retention and reliable sensing becomes difficult as charge storage unit size reduces



# 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



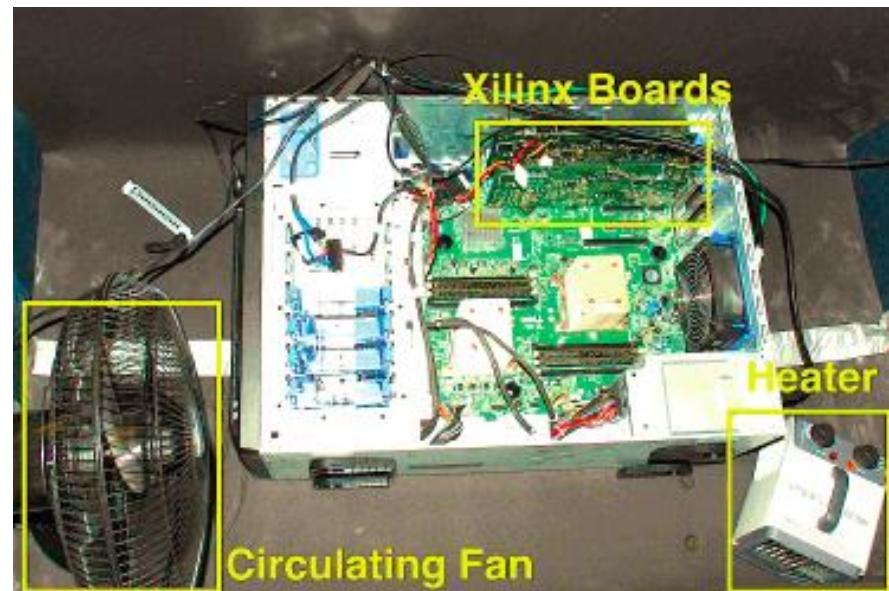
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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

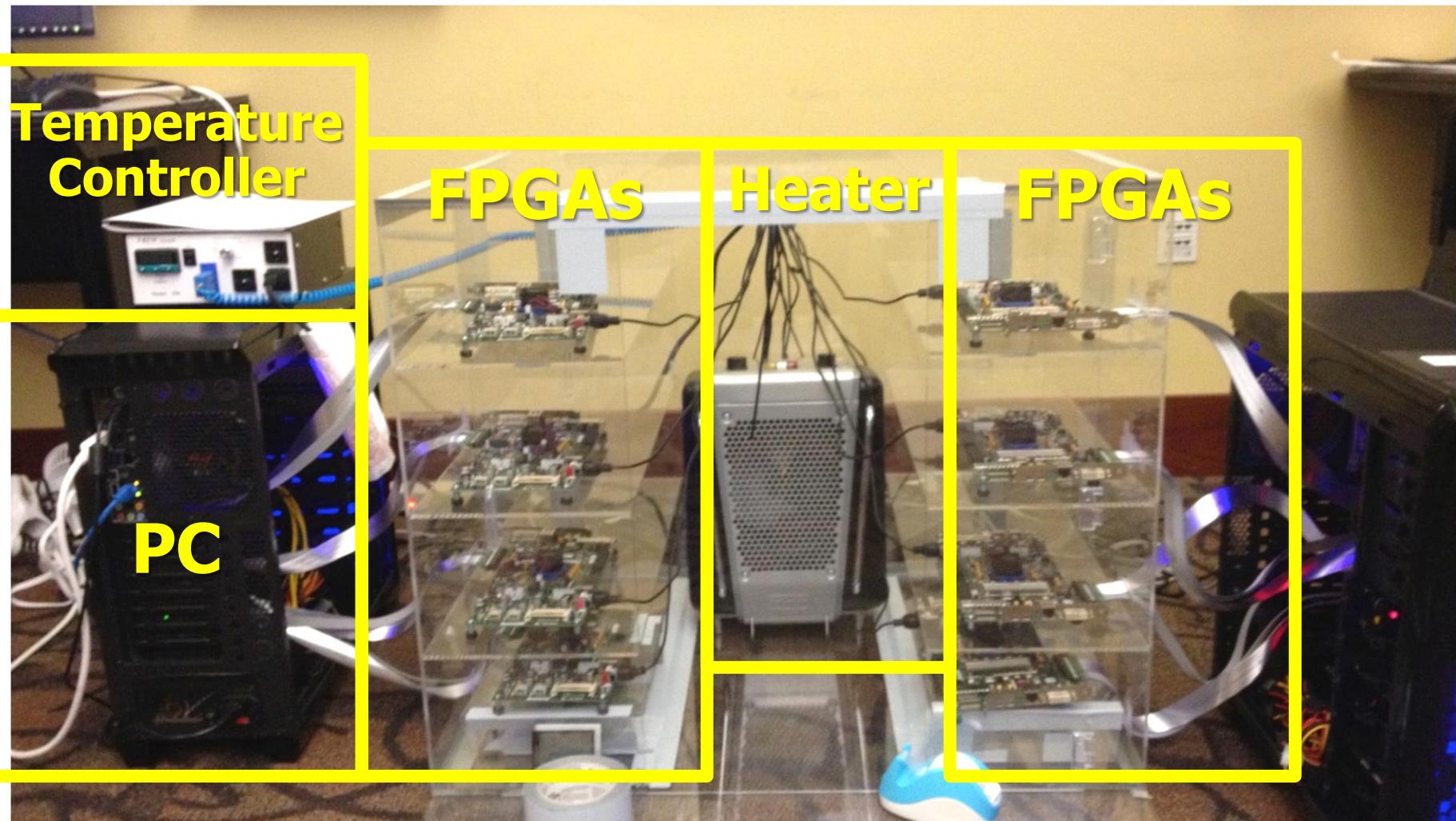
AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)

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

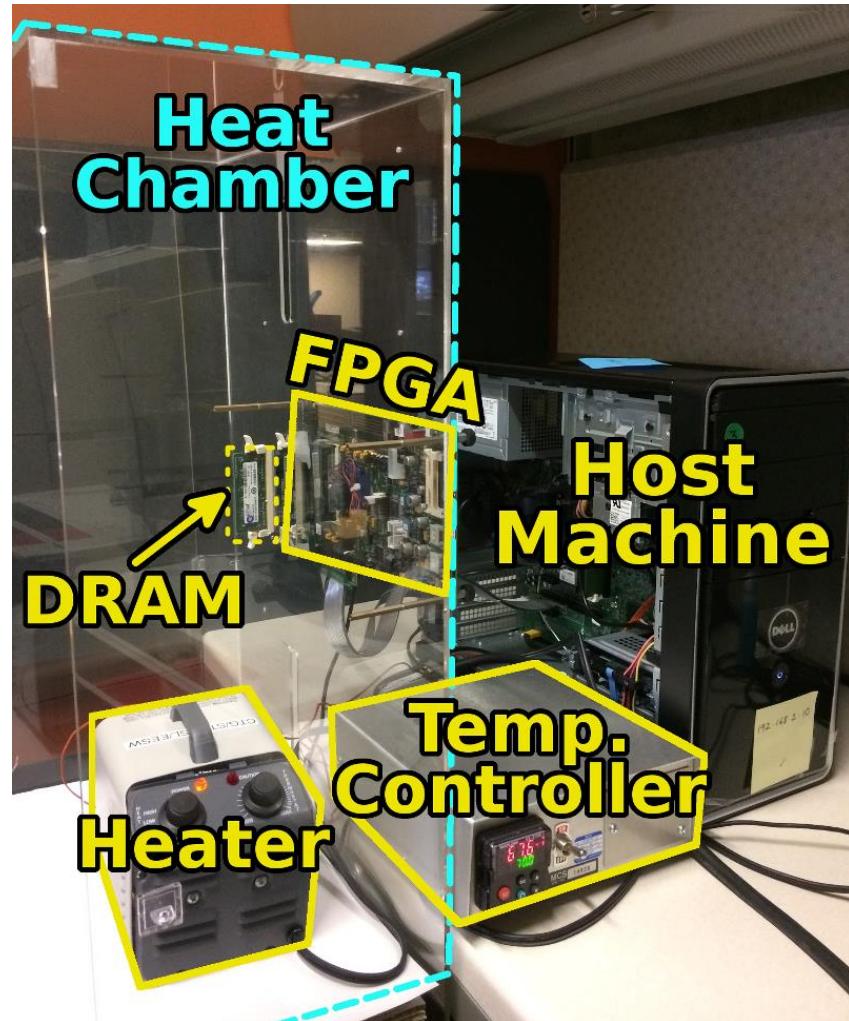


# SoftMC: Open Source DRAM Infrastructure

- 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)*



# SoftMC: Open Source DRAM Infrastructure

---

- <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

# DRAM RowHammer

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

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS

CULTURE

DESIGN

GEAR

SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE

 SHARE  
18276

 TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

# First RowHammer Analysis

---

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,

## **["Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"](#)**

*Proceedings of the [41st International Symposium on Computer Architecture \(ISCA\)](#), Minneapolis, MN, June 2014.*

[\[Slides \(pptx\) \(pdf\)\]](#) [\[Lightning Session Slides \(pptx\) \(pdf\)\]](#) [\[Source Code and Data\]](#) [\[Lecture Video \(1 hr 49 mins\), 25 September 2020\]](#)

***One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD ([link](#)).***

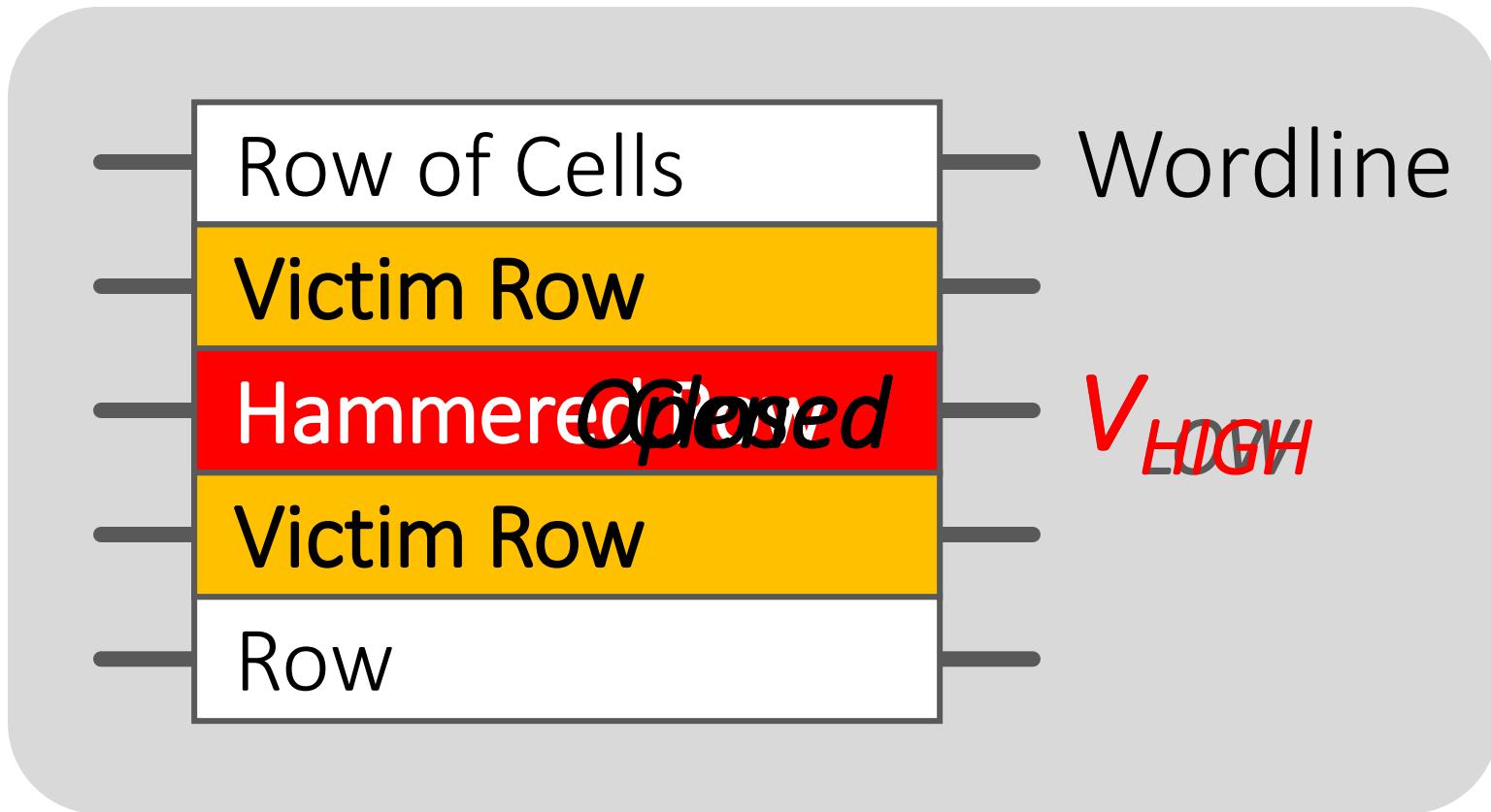
## Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

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

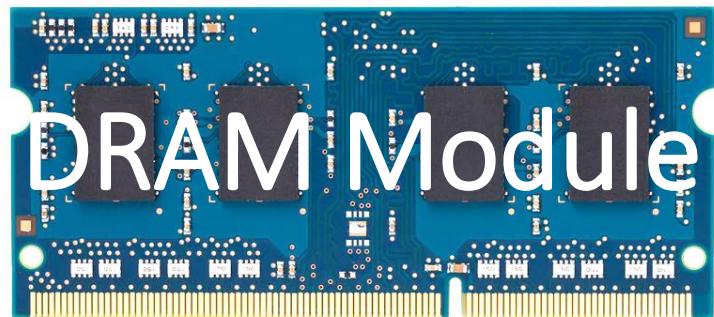
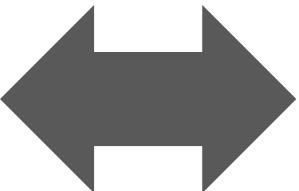
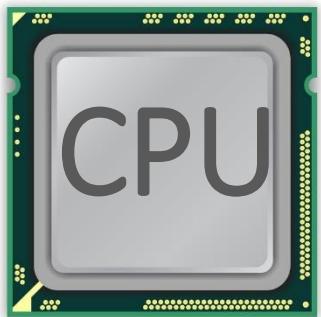
<sup>2</sup>Intel Labs

# Modern DRAM is Prone to Disturbance Errors

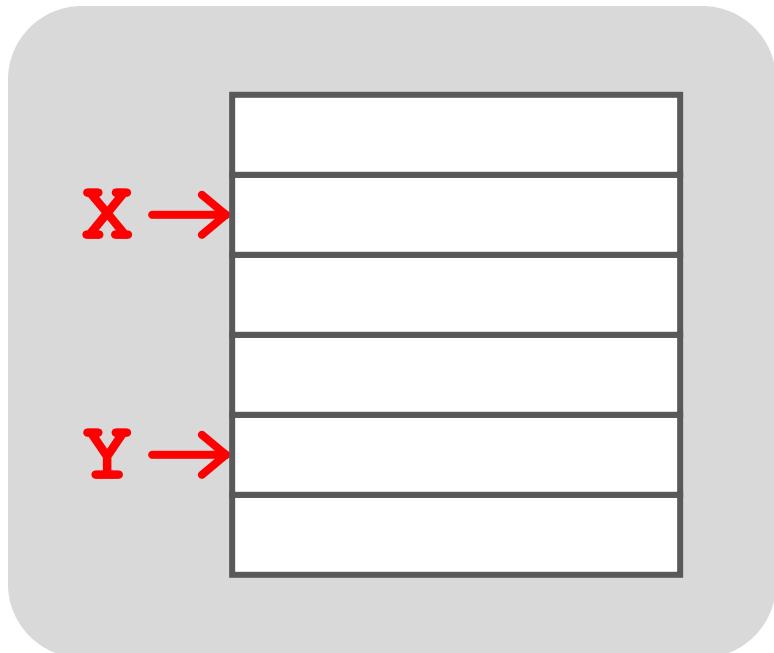


Repeatedly reading a row enough times (before memory gets refreshed) induces **disturbance errors** in adjacent rows in **most real DRAM chips you can buy today**

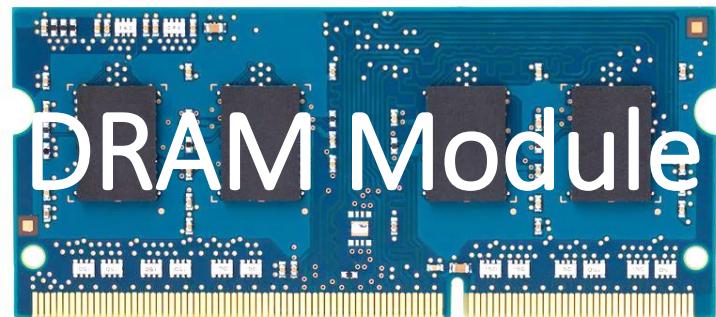
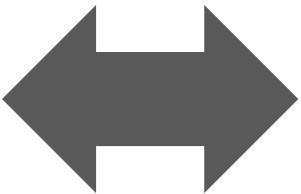
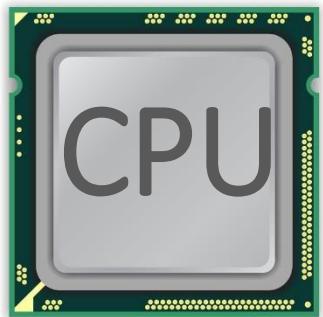
# A Simple Program Can Induce Many Errors



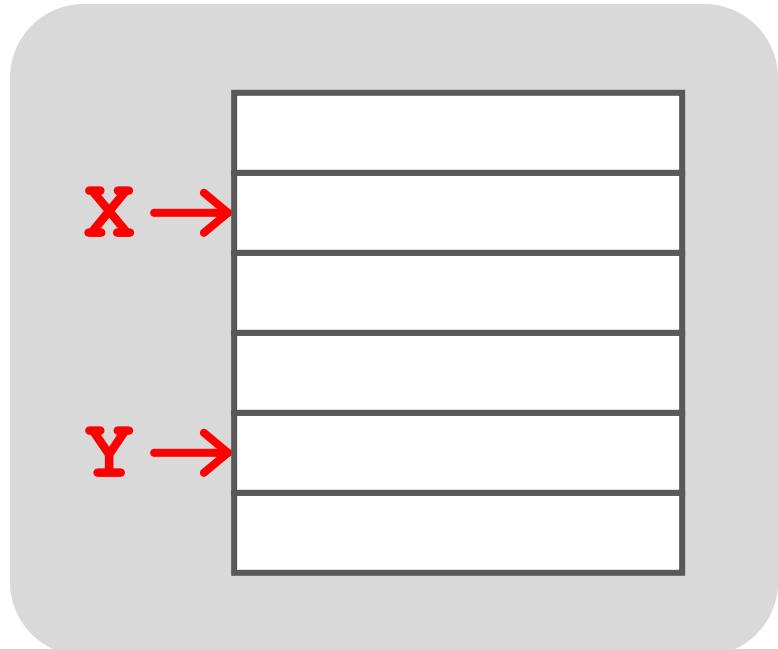
```
loop:  
    mov (%X), %eax  
    mov (%Y), %ebx  
    clflush (%X)  
    clflush (%Y)  
    mfence  
    jmp loop
```



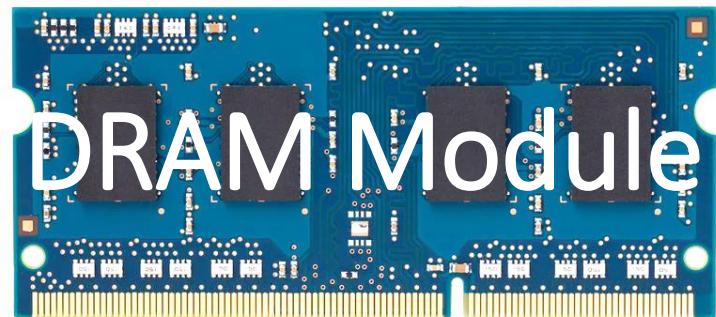
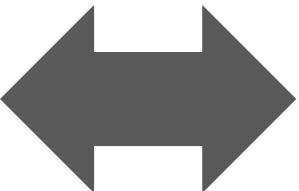
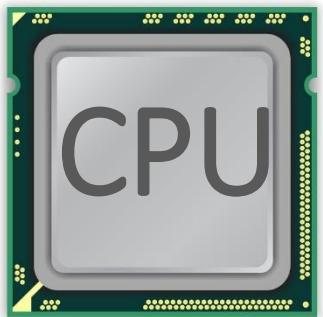
# A Simple Program Can Induce Many Errors



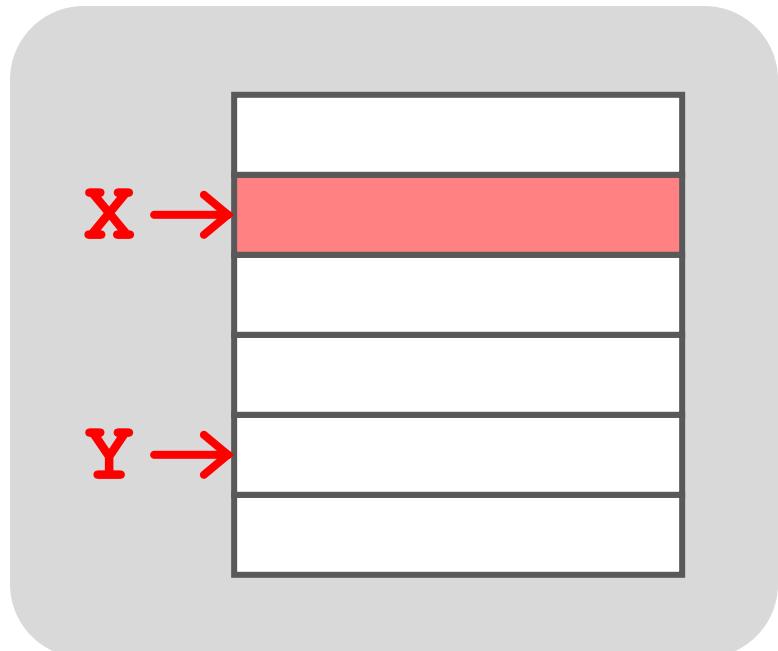
1. Avoid *cache hits*
  - Flush **X** from cache
2. Avoid *row hits* to **X**
  - Read **Y** in another row



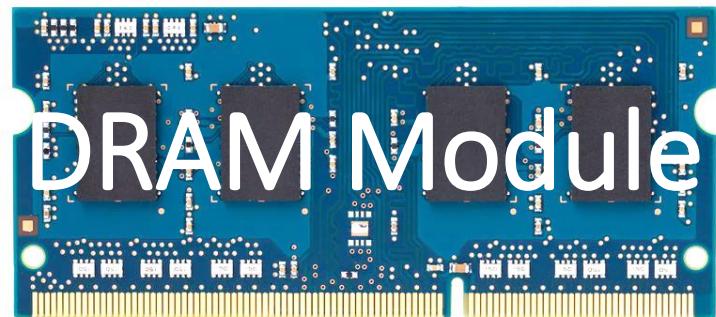
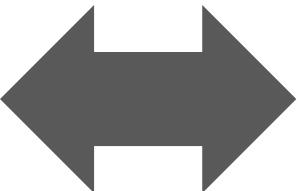
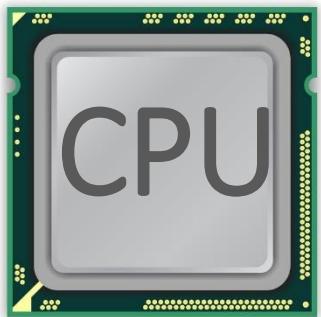
# A Simple Program Can Induce Many Errors



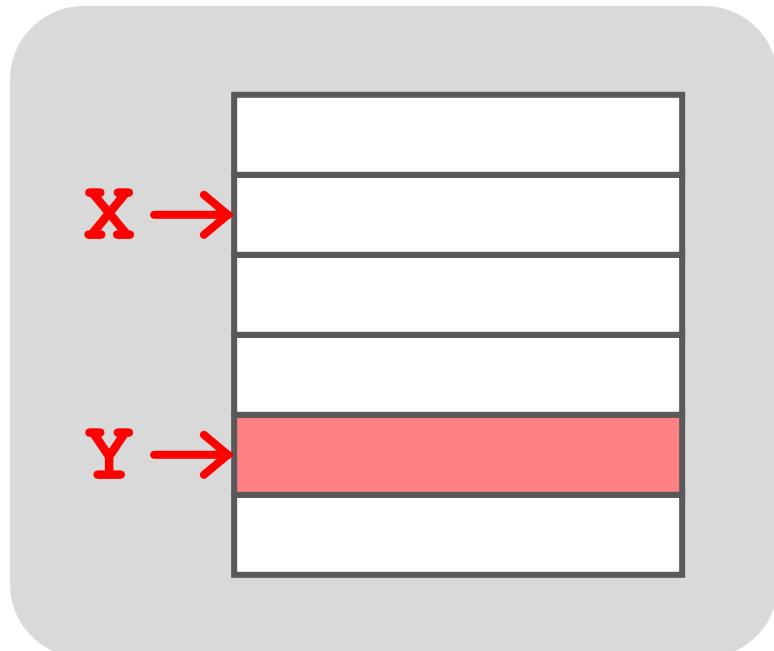
```
loop:  
    mov (%X), %eax  
    mov (%Y), %ebx  
    clflush (%X)  
    clflush (%Y)  
    mfence  
    jmp loop
```



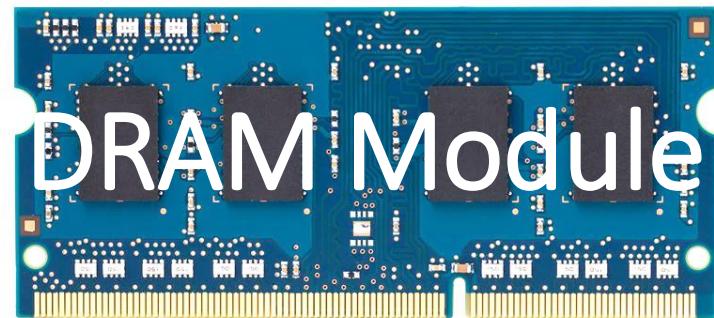
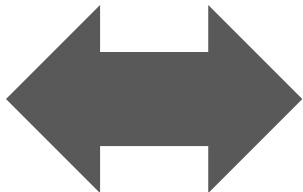
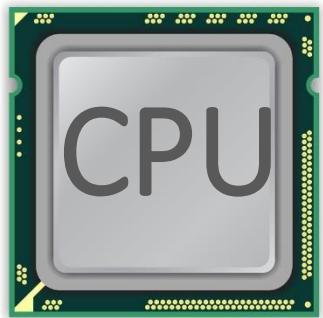
# A Simple Program Can Induce Many Errors



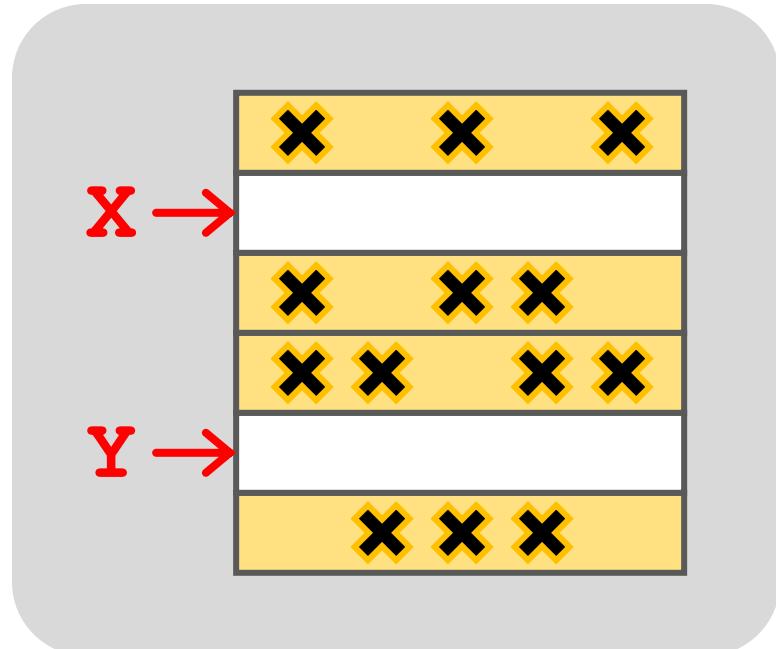
```
loop:  
    mov (%X), %eax  
    mov (%Y), %ebx  
    clflush (%X)  
    clflush (%Y)  
    mfence  
    jmp loop
```



# A Simple Program Can Induce Many Errors

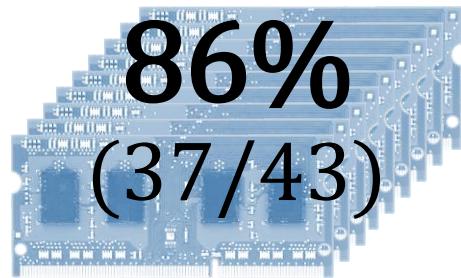


```
loop:  
    mov (%X), %eax  
    mov (%Y), %ebx  
    clflush (%X)  
    clflush (%Y)  
    mfence  
    jmp loop
```

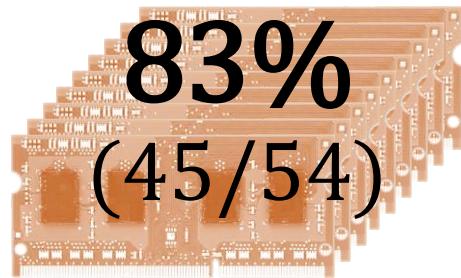


# Most DRAM Modules Are Vulnerable

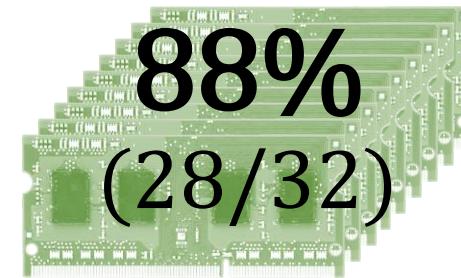
A company



B company



C company

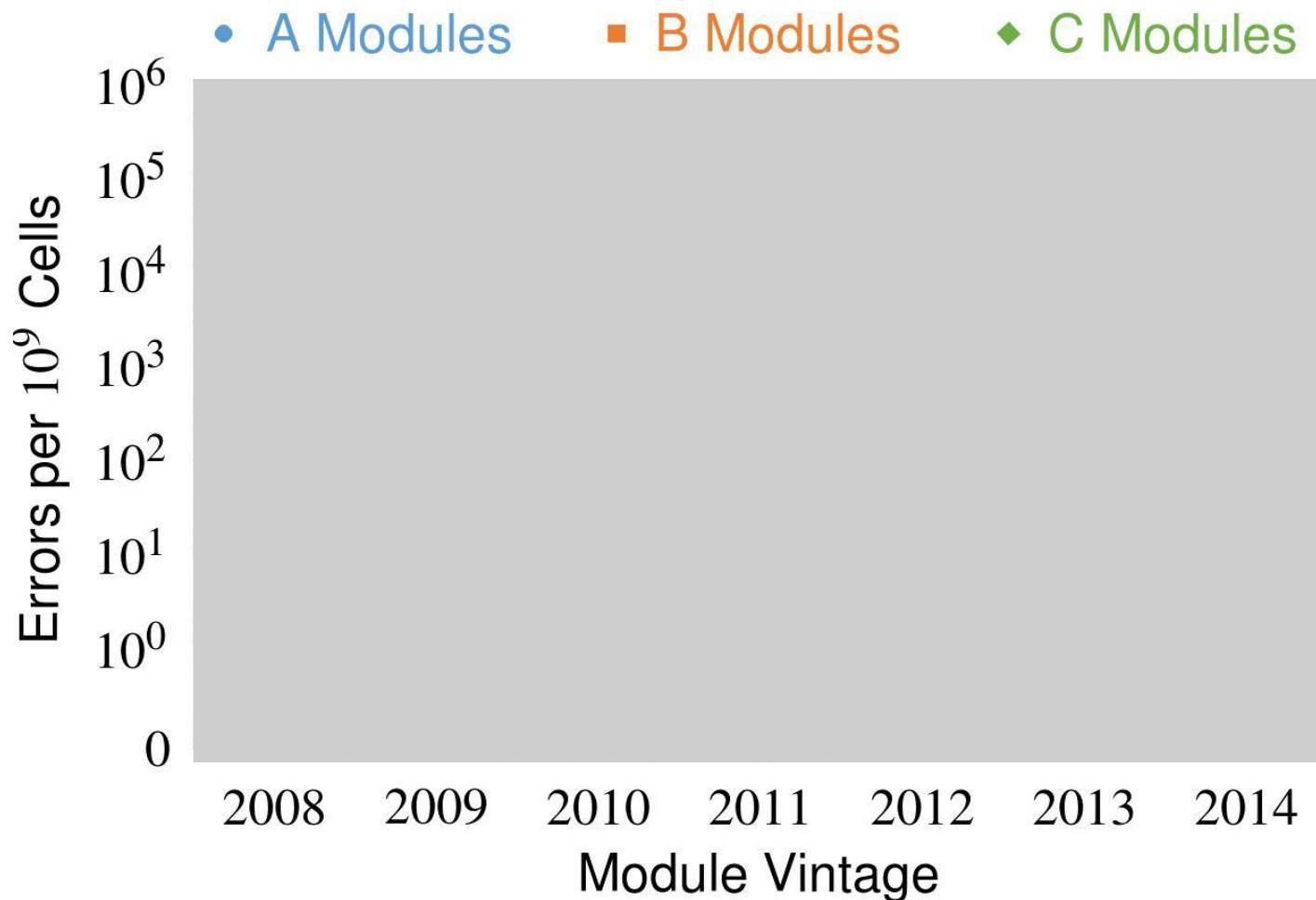


Up to  
 $1.0 \times 10^7$   
errors

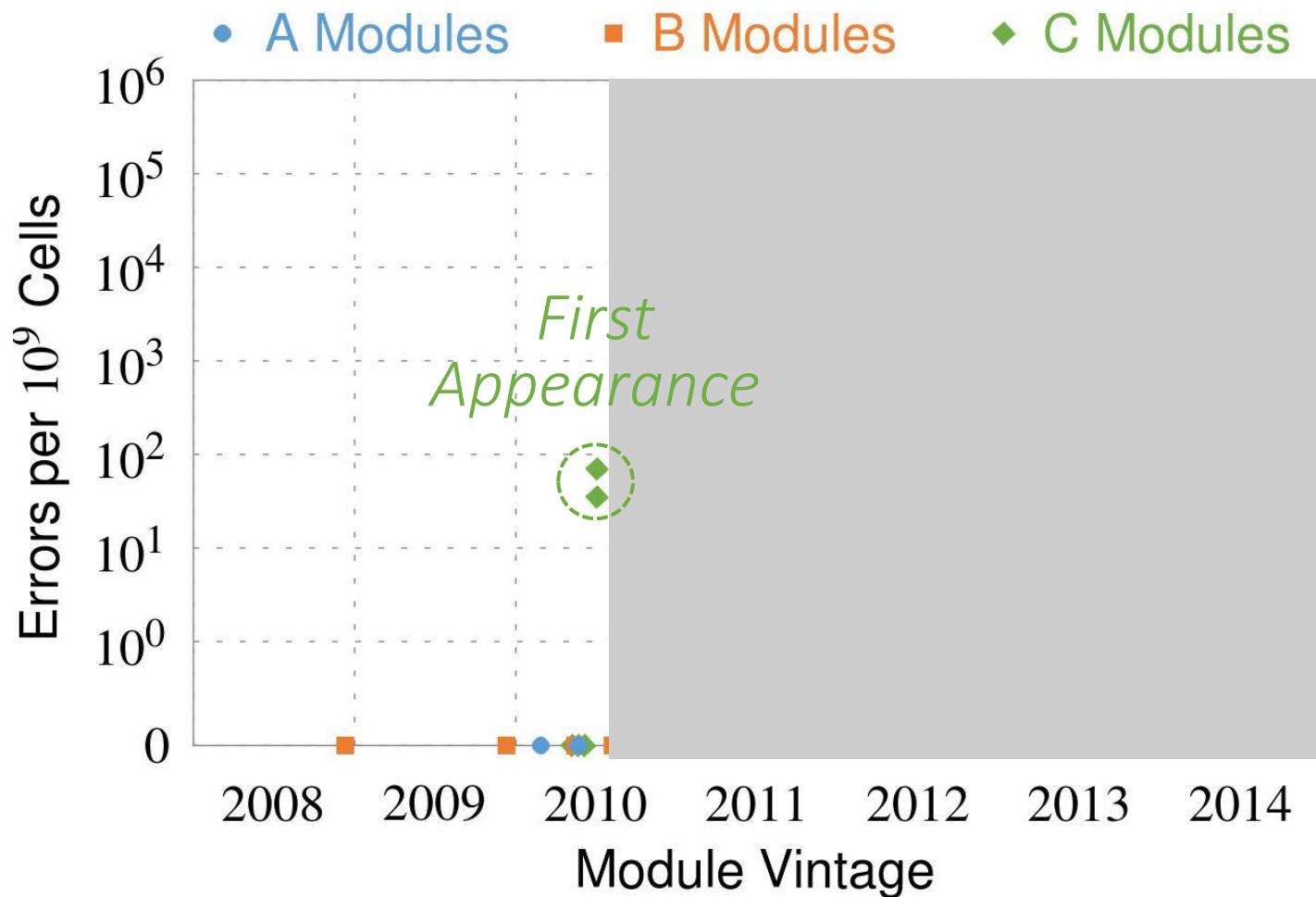
Up to  
 $2.7 \times 10^6$   
errors

Up to  
 $3.3 \times 10^5$   
errors

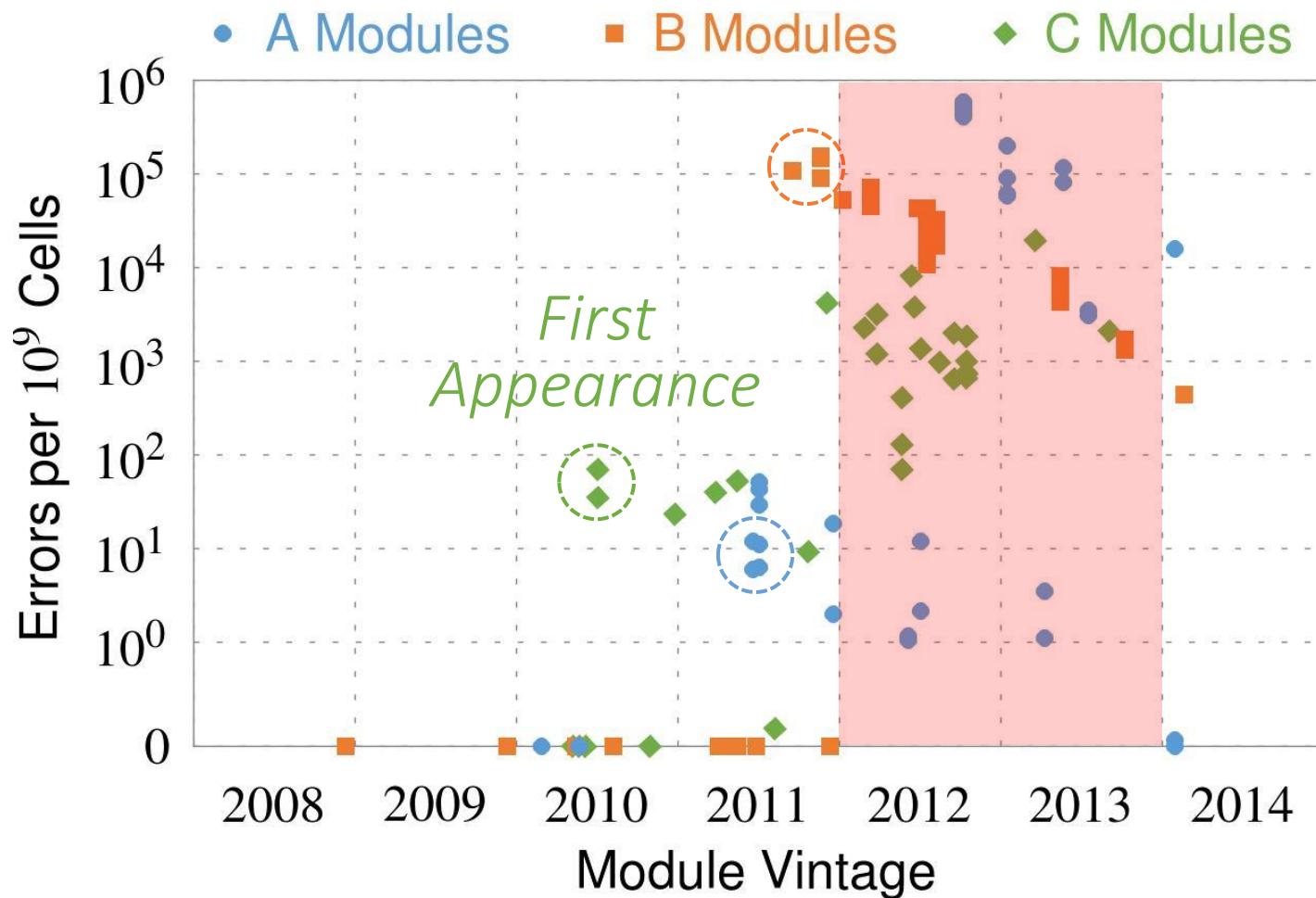
# Recent DRAM Is More Vulnerable



# Recent DRAM Is More Vulnerable



# Recent DRAM Is More Vulnerable



*All modules from 2012–2013 are vulnerable*

# The Reliability & Security Perspectives

---

- 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>

# First RowHammer Analysis

---

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,

## **["Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"](#)**

*Proceedings of the [41st International Symposium on Computer Architecture \(ISCA\)](#)*, Minneapolis, MN, June 2014.

[[Slides \(pptx\)](#) ([pdf](#))] [[Lightning Session Slides \(pptx\)](#) ([pdf](#))] [[Source Code and Data](#)] [[Lecture Video](#) (1 hr 49 mins), 25 September 2020]

***One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD ([link](#)).***

## 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

# A RowHammer Survey

---

- Onur Mutlu and Jeremie Kim,

## "RowHammer: A Retrospective"

*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.*

[Preliminary arXiv version]

[Slides from COSADE 2019 (pptx)]

[Slides from VLSI-SOC 2020 (pptx) (pdf)]

[Talk Video (1 hr 15 minutes, with Q&A)]

# RowHammer: A Retrospective

Onur Mutlu<sup>§‡</sup>

<sup>§</sup>ETH Zürich

Jeremie S. Kim<sup>†§</sup>

<sup>†</sup>Carnegie Mellon University

# A RowHammer Survey: Recent Update

---

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci,

## **"Fundamentally Understanding and Solving RowHammer"**

*Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, January 2023.*

[arXiv version]

[Slides (pptx) (pdf)]

[Talk Video (26 minutes)]

## Fundamentally Understanding and Solving RowHammer

Onur Mutlu

onur.mutlu@safari.ethz.ch

ETH Zürich

Zürich, Switzerland

Ataberk Olgun

ataberk.olgun@safari.ethz.ch

ETH Zürich

Zürich, Switzerland

A. Giray Yağlıkçı

giray.yaglikci@safari.ethz.ch

ETH Zürich

Zürich, Switzerland

**<https://arxiv.org/pdf/2211.07613.pdf>**

---

# 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)]

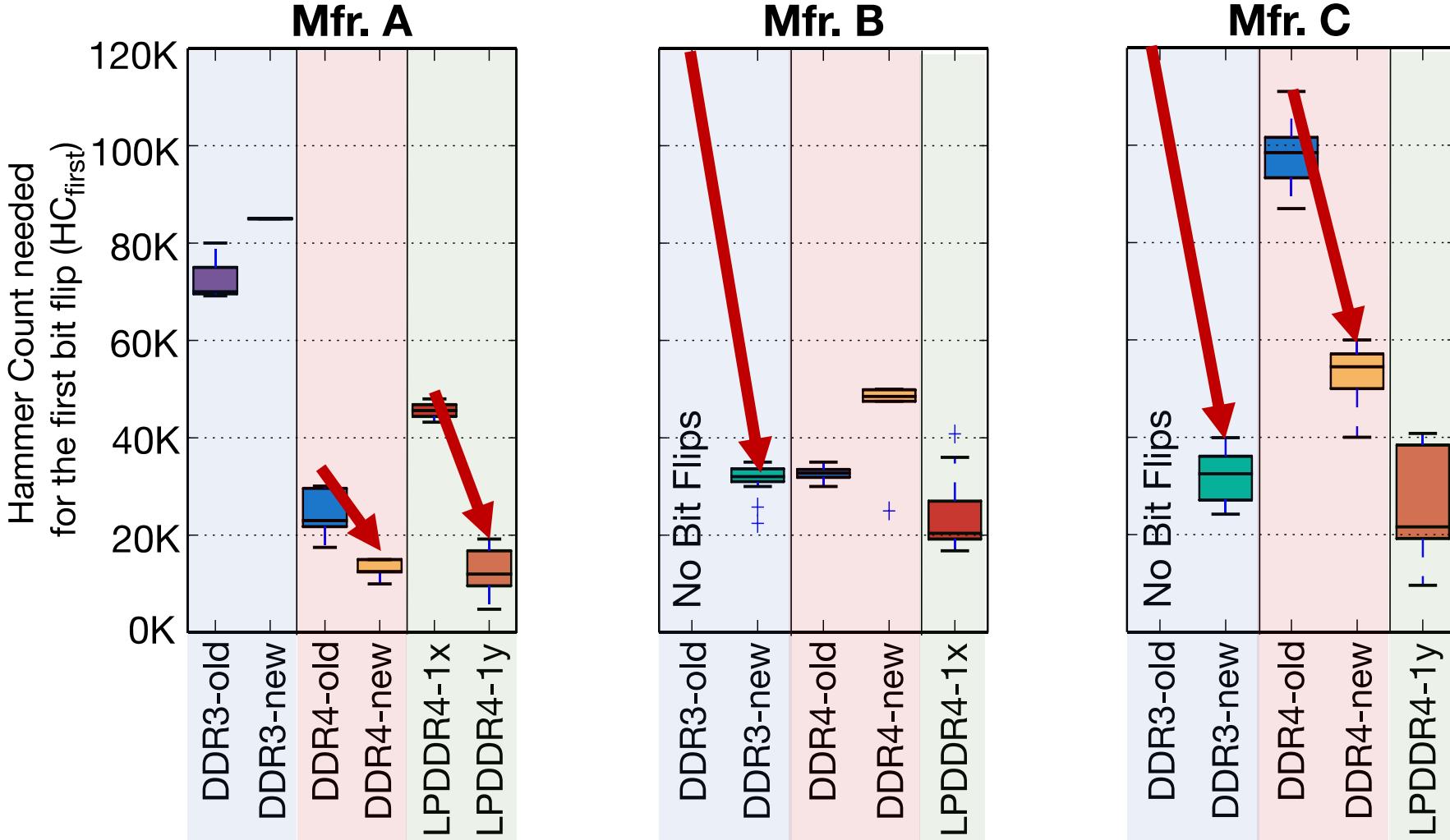
## Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques

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*

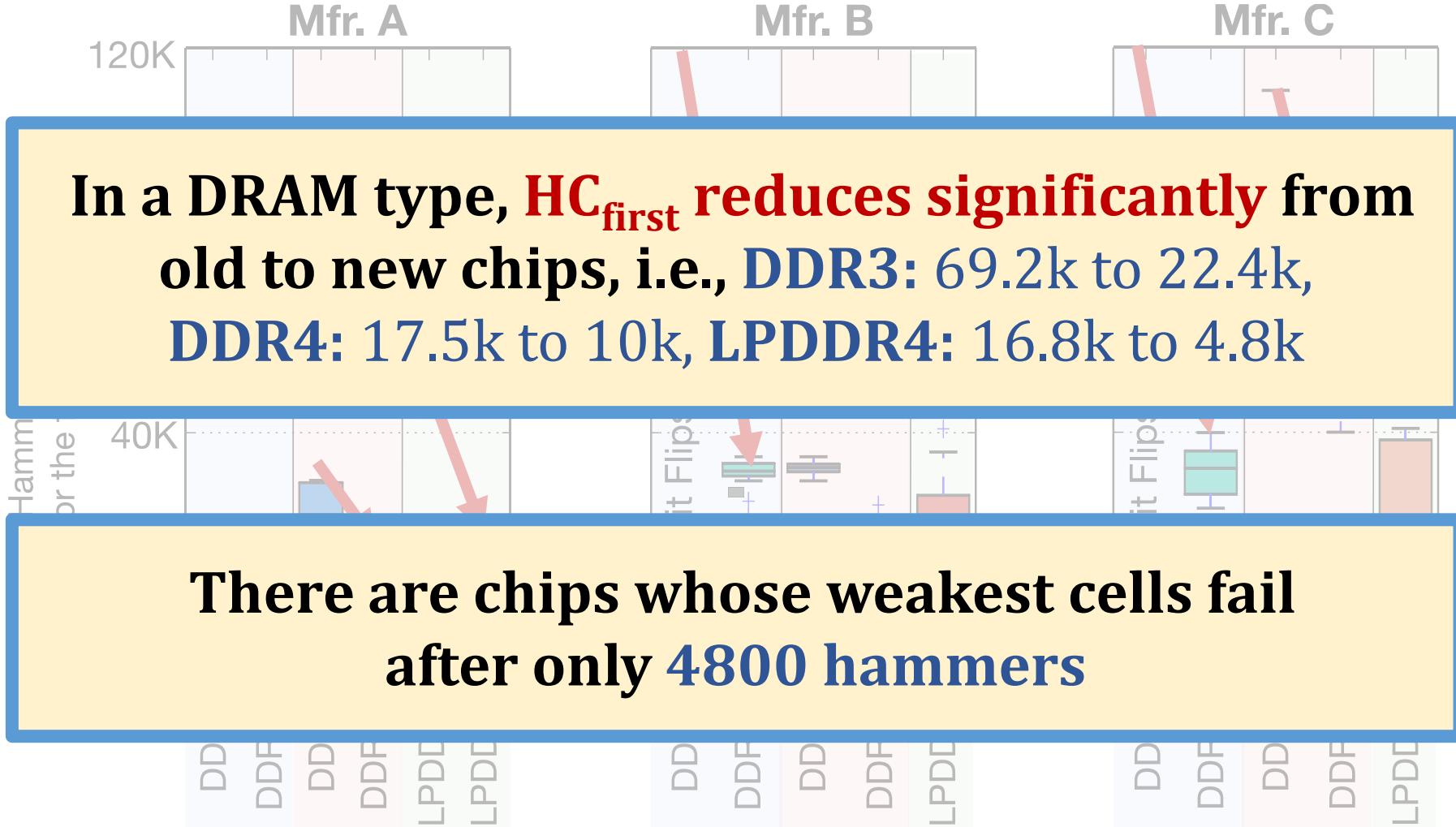
<sup>†</sup>*Carnegie Mellon University*

# 5. First RowHammer Bit Flips per Chip



Newer chips from a given DRAM manufacturer  
**more** vulnerable to RowHammer

# 5. First RowHammer Bit Flips per Chip



In a DRAM type,  $HC_{first}$  reduces significantly from old to new chips, i.e., DDR3: 69.2k to 22.4k, DDR4: 17.5k to 10k, LPDDR4: 16.8k to 4.8k

There are chips whose weakest cells fail after only 4800 hammers

Newer chips from a given DRAM manufacturer  
more vulnerable to RowHammer

# RowHammer in 2020 (II)

---

- 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

# RowHammer in 2020 (III)

---

- Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi,

## **"TRRespass: Exploiting the Many Sides of Target Row Refresh"**

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

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

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

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

[[Lecture Video](#) (59 minutes)]

[[Source Code](#)]

[[Web Article](#)]

**Best paper award.**

**Pwnie Award 2020 for Most Innovative Research.** [Pwnie Awards 2020](#)

# TRRespass: Exploiting the Many Sides of Target Row Refresh

Pietro Frigo\*†    Emanuele Vannacci\*†    Hasan Hassan§    Victor van der Veen¶  
Onur Mutlu§    Cristiano Giuffrida\*    Herbert Bos\*    Kaveh Razavi\*

# RowHammer is Getting Much Worse (2020)

---

- 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*

# RowHammer Has Many Dimensions (2021)

- Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, and Onur Mutlu,

## **"A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses"**

*Proceedings of the 54th International Symposium on Microarchitecture (MICRO)*, Virtual, October 2021.

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

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

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

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

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

[[arXiv version](#)]

## **A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses**

Lois Orosa\*  
ETH Zürich

A. Giray Yağlıkçı\*  
ETH Zürich

Haocong Luo  
ETH Zürich

Ataberk Olgun  
ETH Zürich, TOBB ETÜ

Jisung Park  
ETH Zürich

Hasan Hassan  
ETH Zürich

Minesh Patel  
ETH Zürich

Jeremie S. Kim  
ETH Zürich

Onur Mutlu  
ETH Zürich

# RowHammer vs. Wordline Voltage (2022)

---

- A. Giray Yağlıkçı, Haocong Luo, Geraldo F. de Oliviera, Ataberk Olgun, Minesh Patel, Jisung Park, Hasan Hassan, Jeremie S. Kim, Lois Orosa, and Onur Mutlu,

## **"Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices"**

*Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Baltimore, MD, USA, June 2022.

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

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

[[arXiv version](#)]

[[Talk Video](#) (34 minutes, including Q&A)]

[[Lightning Talk Video](#) (2 minutes)]

## **Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices**

A. Giray Yağlıkçı<sup>1</sup> Haocong Luo<sup>1</sup> Geraldo F. de Oliviera<sup>1</sup> Ataberk Olgun<sup>1</sup> Minesh Patel<sup>1</sup>  
Jisung Park<sup>1</sup> Hasan Hassan<sup>1</sup> Jeremie S. Kim<sup>1</sup> Lois Orosa<sup>1,2</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>*ETH Zürich*

<sup>2</sup>*Galicia Supercomputing Center (CESGA)*

# RowHammer in HBM Chips (2023)

---

- Ataberk Olgun, Majd Osseiran, A. Giray Yağlıkçı, Yahya Can Tuğrul, Haocong Luo, Steve Rhyner, Behzad Salami, Juan Gomez-Luna, and Onur Mutlu,

## **"An Experimental Analysis of RowHammer in HBM2 DRAM Chips"**

*Proceedings of the 53nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Disrupt Track (DSN Disrupt), Porto, Portugal, June 2023.*

[[arXiv version](#)]

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

[[Talk Video](#) (24 minutes, including Q&A)]

## An Experimental Analysis of RowHammer in HBM2 DRAM Chips

Ataberk Olgun<sup>1</sup> Majd Osseiran<sup>1,2</sup> A. Giray Yağlıkçı<sup>1</sup> Yahya Can Tuğrul<sup>1</sup>  
Haocong Luo<sup>1</sup> Steve Rhyner<sup>1</sup> Behzad Salami<sup>1</sup> Juan Gomez Luna<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>*SAFARI Research Group, ETH Zürich*      <sup>2</sup>*American University of Beirut*

# RowPress [ISCA 2023]



- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tugrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu,  
**"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"**

*Proceedings of the 50th International Symposium on Computer Architecture (ISCA)*, Orlando, FL, USA, June 2023.

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

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

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

[[RowPress Source Code and Datasets](#) (Officially Artifact Evaluated with All Badges)]

***Officially artifact evaluated as available, reusable and reproducible.  
Best artifact award at ISCA 2023.***

## RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

Haocong Luo   Ataberk Olgun   A. Giray Yağlıkçı   Yahya Can Tuğrul   Steve Rhyner  
Meryem Banu Cavlak   Joël Lindegger   Mohammad Sadrosadati   Onur Mutlu  
*ETH Zürich*

## Key Takeaways

---

RowHammer is still  
an open problem

Security by obscurity  
is likely not a good solution

# Major Trends Affecting Main Memory (V)

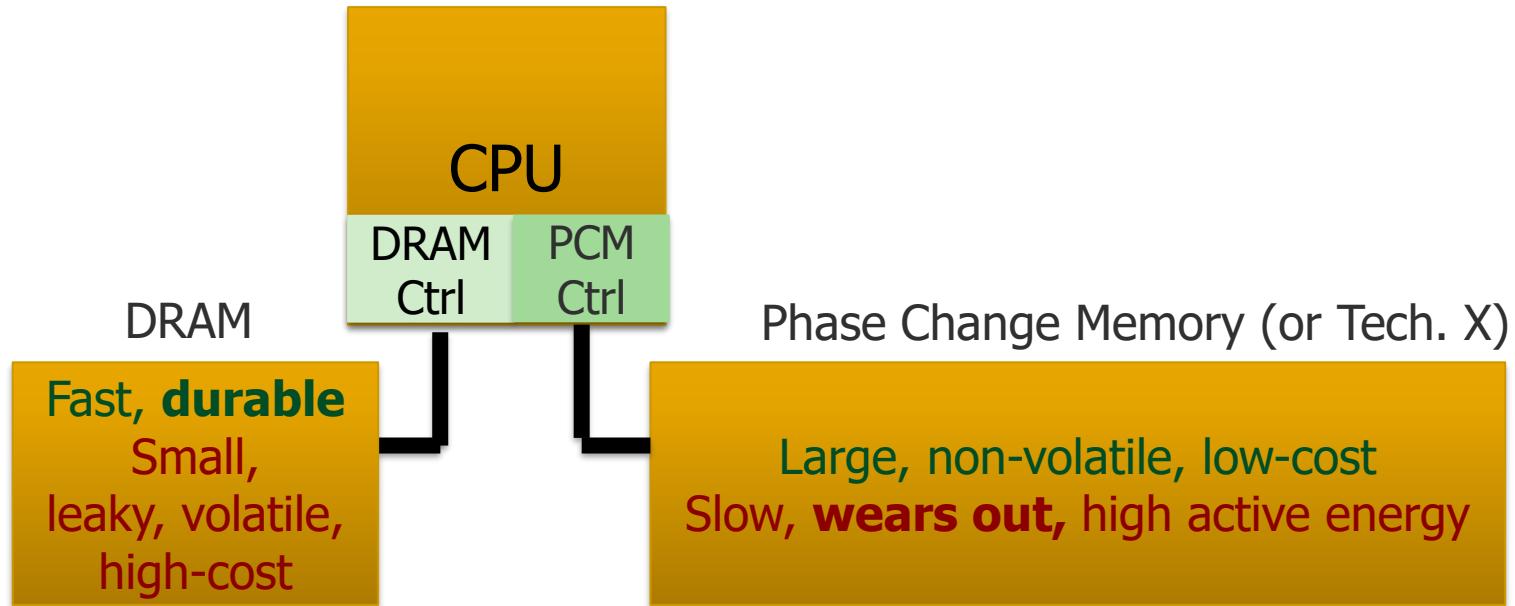
- DRAM scaling has already become increasingly difficult
  - Increasing cell leakage current, reduced cell reliability, increasing manufacturing difficulties [Kim+ ISCA 2014], [Liu+ ISCA 2013], [Mutlu IMW 2013], [Mutlu DATE 2017]
  - **Difficult to significantly improve capacity, energy**
- **Emerging memory technologies** are promising


# Major Trends Affecting Main Memory (V)

- DRAM scaling has already become increasingly difficult
  - Increasing cell leakage current, reduced cell reliability, increasing manufacturing difficulties [Kim+ ISCA 2014], [Liu+ ISCA 2013], [Mutlu IMW 2013], [Mutlu DATE 2017]
  - **Difficult to significantly improve capacity, energy**
- **Emerging memory technologies** are promising

<b>3D-Stacked DRAM</b>	higher bandwidth	smaller capacity
<b>Reduced-Latency DRAM</b> (e.g., RL/TL-DRAM, FLY-RAM)	lower latency	higher cost
<b>Low-Power DRAM</b> (e.g., LPDDR3, LPDDR4, Voltron)	lower power	higher latency higher cost
<b>Non-Volatile Memory (NVM)</b> (e.g., PCM, STTRAM, ReRAM, 3D Xpoint)	larger capacity	higher latency higher dynamic power lower endurance

# Major Trend: Hybrid Main Memory



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.

# Main Memory Needs Intelligent Controllers

# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

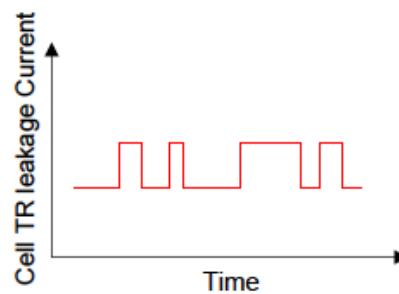
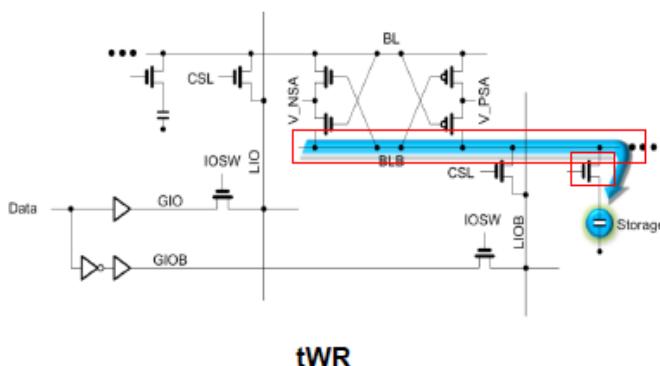
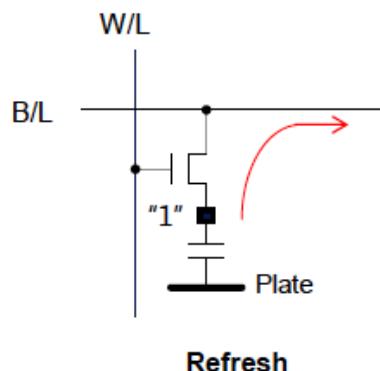
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

### ❖ 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



VRT

# Call for Intelligent Memory Controllers

## DRAM Process Scaling Challenges

### ❖ Refresh

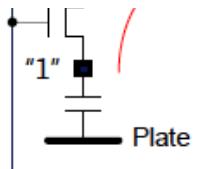
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

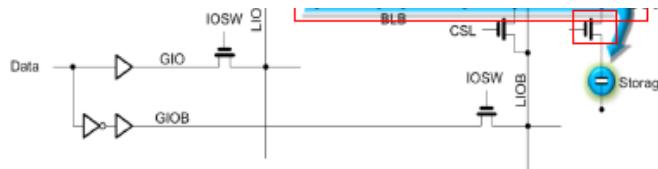
## Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

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*



Refresh



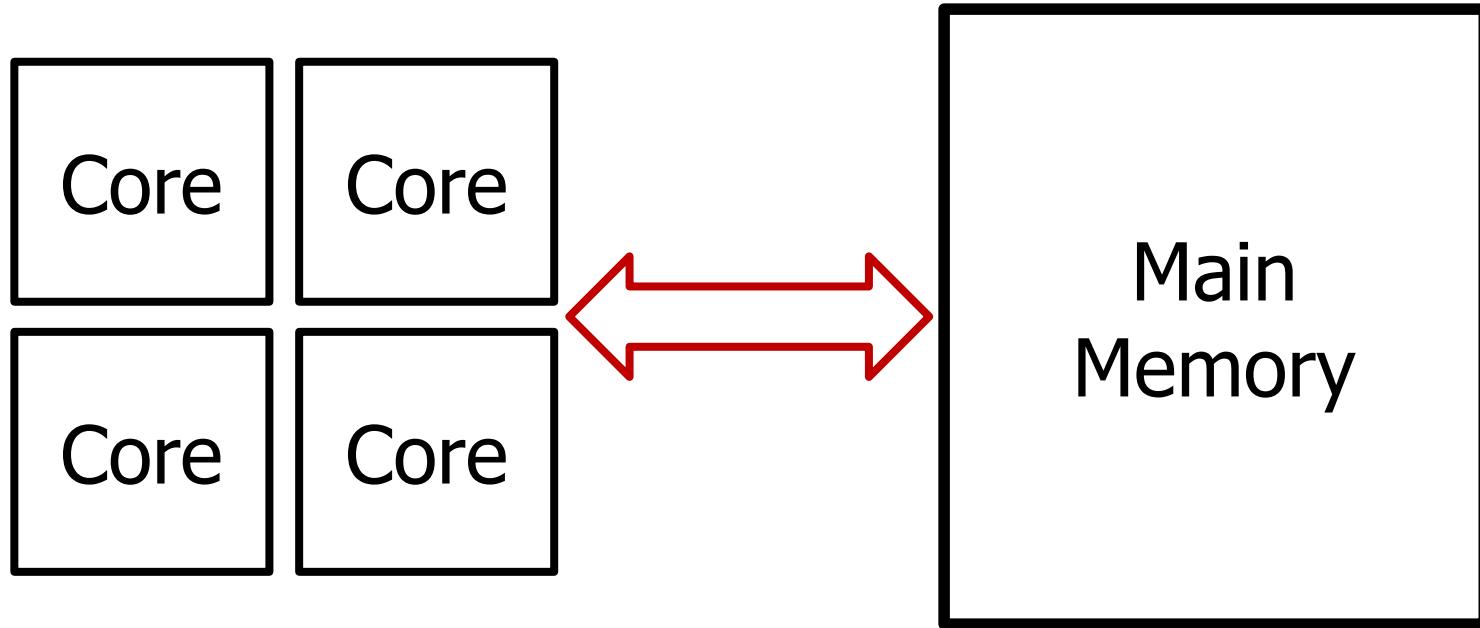
tWR



VRT

# An Orthogonal Issue: Memory Interference

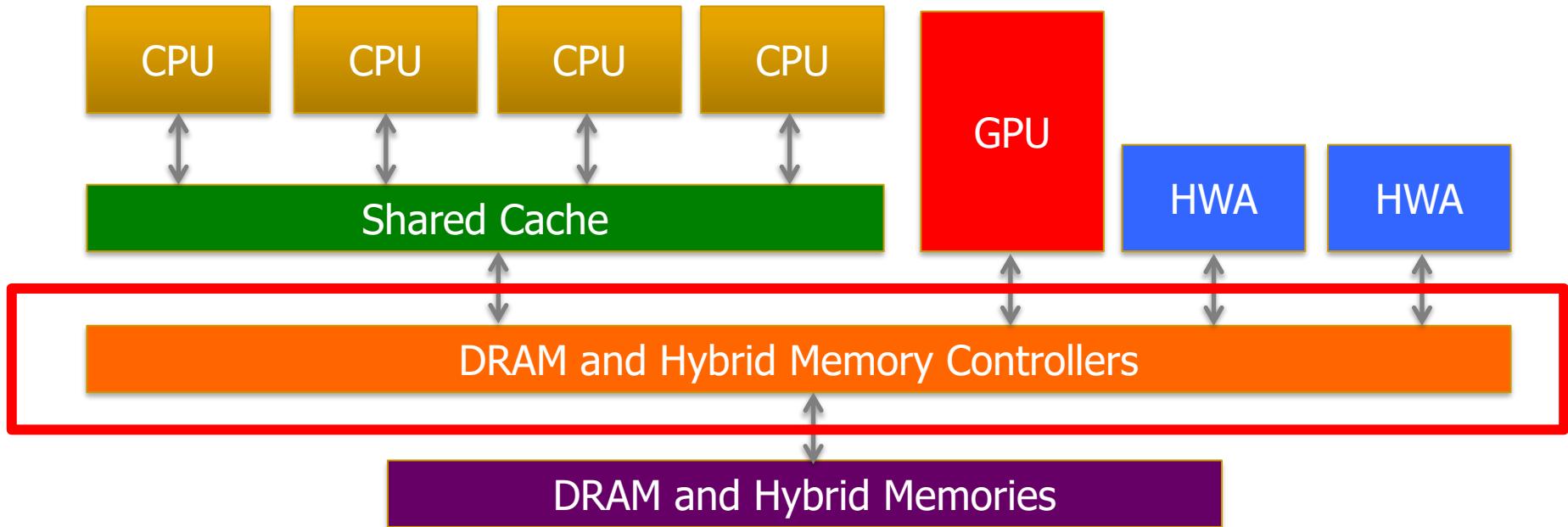
---



Cores' interfere with each other when accessing shared main memory  
Uncontrolled interference leads to many problems (QoS, performance)

---

# Goal: Predictable Performance in Complex Systems



- Heterogeneous agents: CPUs, GPUs, and HWAs
- Main memory interference between CPUs, GPUs, HWAs

How to allocate resources to heterogeneous agents  
to mitigate interference and provide predictable performance?

# Main Memory Needs Intelligent Controllers

# Solving the Memory Problem

# How Do We Solve The Memory Problem?

---

- **Fix it:** Make memory and controllers more intelligent
  - **New interfaces, functions, architectures:** system-mem codesign
- **Eliminate or minimize it:** Replace or (more likely) augment DRAM with a different technology
  - **New technologies and system-wide rethinking** of memory & storage
- **Embrace it:** Design heterogeneous memories (none of which are perfect) and map data intelligently across them
  - **New models for data management and maybe usage**
- ...

# How Do We Solve The Memory Problem?

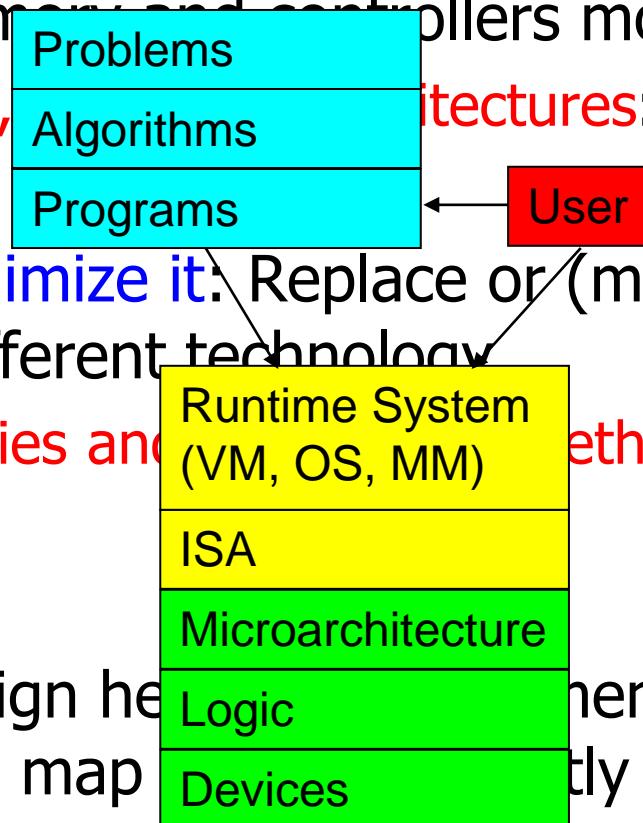
---

- Fix it: Make memory and controllers more intelligent
  - New interfaces, functions, architectures: system-mem codesign
- Eliminate or minimize it: Replace or (more likely) augment DRAM with a different technology
  - New technologies and system-wide rethinking of memory & storage
- Embrace it: Design heterogeneous memories (none of which are perfect) and map data intelligently across them
  - New models for data management and maybe usage

**Solutions (to memory scaling) require  
software/hardware/device cooperation**

# How Do We Solve The Memory Problem?

- Fix it: Make memory and controllers more intelligent
  - New interfaces, architectures: system-mem codesign
- Eliminate or minimize it: Replace or (more likely) augment DRAM with a different technology
  - New technologies and storage
- Embrace it: Design heterogeneous memories (none of which are perfect) and map them flexibly across them
  - New models for data management and maybe usage



**Solutions (to memory scaling) require software/hardware/device cooperation**

# Solution 1: New Memory Architectures

---

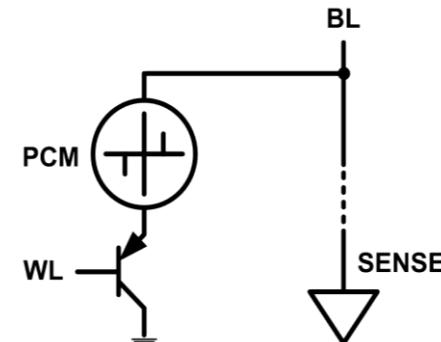
- Overcome memory shortcomings with
  - Memory-centric system design
  - Novel memory architectures, interfaces, functions
  - Better waste management (efficient utilization)
  
- Key issues to tackle
  - Enable reliability at low cost → high capacity
  - Reduce energy
  - Reduce latency
  - Improve bandwidth
  - Reduce waste (capacity, bandwidth, latency)
  - Enable computation close to data

# Solution 1: New Memory Architectures

- Liu+, "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.
- Kim+, "A Case for Exploiting Subarray-Level Parallelism in DRAM," ISCA 2012.
- Lee+, "Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture," HPCA 2013.
- Liu+, "An Experimental Study of Data Retention Behavior in Modern DRAM Devices," ISCA 2013.
- Seshadri+, "RowClone: Reducing Copy and Garbage Collection in Bulk Data," MICRO 2013.
- Pekhimenko+, "Memory Compression: A Major Step Forward in Compression Framework," MICRO 2013.
- Chang+, "Improving DRAM Performance by Parallelizing Refreshes with Accesses," HPCA 2014.
- Xuan+, "The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study," SIGMETRICS 2014.
- Luo+, "Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost," DSN 2014.
- Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.
- Lee+, "Adaptive Latency DRAM: Optimizing DRAM Timing for the Common-Case," HPCA 2015.
- Qureshi+, "AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems," DSN 2015.
- Mezav+, "Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field," DSN 2015.
- Kim+, "Ramulator: A Fast and Extensible DRAM Simulator," IEEE CAL 2015.
- Seshadri+, "Fast Bulk Bitwise AND and OR in DRAM," IEEE CAL 2015.
- Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing," ISCA 2015.
- Ahn+, "PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture," ISCA 2015.
- Lee+, "Decoupled Direct Memory Access: Isolating CPU and IO Traffic by Leveraging a Dual-Data-Port DRAM," PACT 2015.
- Seshadri+, "Gather-Scatter DRAM: In-DRAM Address Translation to Improve the Spatial Locality of Non-unit Strided Accesses," MICRO 2015.
- Lee+, "Simultaneous Multi-Layer Access: Improving 3D-Stacked Memory Bandwidth at Low Cost," TACO 2016.
- Hassan+, "ChargeCache: Reducing DRAM Latency by Exploiting Row Access Locality," HPCA 2016.
- Chang+, "Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Migration in DRAM," HPCA 2016.
- Chang+, "Understanding Latency Variation in Modern DRAM Chips Experimental Characterization, Analysis, and Optimization," SIGMETRICS 2016.
- Khan+, "PARBON: An Efficient System-Level Technique to Detect Data Dependent Failures in DRAM," DSN 2016.
- Heish+, "Transparent Offloading and Migration (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems," ISCA 2016.
- Hassan+, "Cooperative DRAM Cache Misses with an Enhanced Memory Controller," ISCA 2016.
- Boroumand+, "LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory," IEEE CAL 2016.
- Patrakai+, "Scaling Towards Scalable GPU Architectures with Processing-In-Memory Capabilities," PACT 2016.
- Heish+, "Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation," ICD 2016.
- Hoshimi+, "Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads," MICRO 2016.
- Khan+, "A Case for Memory Content-Based Detection and Mitigation of Data-Dependent Failures in DRAM," IEEE CAL 2016.
- Hassan+, "SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies," HPCA 2017.
- Mutu+, "The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser," DATE 2017.
- Lee+, "Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms," SIGMETRICS 2017.
- Chang+, "Understanding Reduced-Voltage Operation in Modern DRAM Devices: Experimental Characterization, Analysis, and Mechanisms," SIGMETRICS 2017.
- Patel+, "The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions," ISCA 2017.
- Seshadri and Mutu, "Simple Operations in Memory to Reduce Data Movement," ADCOM 2017.
- Liu+, "Concurrent Data Structures for Near-Memory Computing," SPAW 2017.
- Khan+, "Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content," MICRO 2017.
- Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO 2017.
- Kim+, "GRIM-Filter: Fair Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies," BMC Genomics 2018.
- Kim+, "DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices," HPCA 2018.
- Boroumand+, "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS 2018.
- Das+, "VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency," DAC 2018.
- Ghose+, "What Your DRAM Power Modes Are Not Telling You: Lessons from a Detailed Experimental Study," SIGMETRICS 2018.
- Kim+, "SolarDRAM: Reducing DRAM Power Consumption by Exploiting the Sun in Local Bins," ICD 2018.
- Wang+, "Performance of DRAM Cache via Change Level-Half-Size Look-Aside Partition Reservation," MICRO 2018.
- Kim+, "DR-NegC: Using Commodity DRAM Devices to Generate True Random Numbers with Low Power and High Throughput," HPCA 2019.
- Singh+, "NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning," DAC 2019.
- Ghose+, "Demystifying Workload-DRAM Interactions: An Experimental Study," SIGMETRICS 2019.
- Patel+, "Understanding and Modeling On-die Error Correction in Modern DRAM: An Experimental Study Using Real Devices," DSN 2019.
- Boroumand+, "CoIDA: Efficient Cache Coherence Support for Near-Die Accelerators," ISCA 2019.
- Hassan+, "CROW: A Low-Cost Substrate for Improving DRAM Performance, Energy Efficiency, and Reliability," ISCA 2019.
- Mutu and Kim, "RowHammer: A Retrospective," TACD 2019.
- Mutu+, "Processing Data Where It Makes Sense: Enabling In-Memory Computation," MICPRO 2019.
- Seshadri and Mutu, "In-DRAM Bulk Bitwise Execution Engine," ADCOM 2020.
- Koppula+, "EDEN: Energy-Efficient High-Performance Neural Network Inference Using Approximate DRAM," MICRO 2019.
- Zarei+, "NOM: Network-on-Memory for Inter-Bank Data Transfer in Highly-Banked Memories," CAL 2020.
- Frigio+, "TRRESPASS: Exploring the Many Sides of Target Row Refresh," S&P 2020.
- Cocjcar+, "Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers," S&P 2020.
- Luo+, "CLR-DRAM: A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-Off," ISCA 2020.
- Kim+, "Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques," ISCA 2020.
- Salami+, "An Experimental Study of Reduced-Voltage Operation in Modern FPGAs for Network Network Acceleration," DSN 2020.
- Fernandez+, "NTASA: A Near-Delay Processing Accelerator for Time Series Analyses," ICD 2020.
- Wang+, "FIGARD: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching," MICRO 2020.
- Patel+, "Bit-Exact ECC Recovery (BEER): Determining DRAM On-Die ECC Functions by Exploiting DRAM Data Retention Characteristics," MICRO 2020.
- Jain+, "Refresh Triggered Computation: Improving the Energy Efficiency of Convolutional Neural Network Accelerators," TC 2020.
- Latifi+, "Reducing DRAM Power Consumption and Redesign of High-Bandwidth Memory with Voltage Underclocking," DATE 2021.
- Yaglikci+, "BlockHammer: Preventing Rowhammer at Low Cost by Blocking Single-Block Access DRAM Cache," HPCA 2021.
- Giannoutsos+, "SyncOn: Efficient Synchronization Support for Near-Data Processing Architectures," HPCA 2021.
- Mutu+, "A Modern Primer on Processing in Memory," Invited Book Chapter 2021.
- Hajinazar+, "SiDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM," ASPLOS 2021.
- Orosa+, "CODEC: A Low-Cost Substrate for Enabling Custom In-DRAM Functionality and Optimizations," ISCA 2021.
- Olgun+, "QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips," ISCA 2021.
- Singh+, "TRGA-based Near-Memory Acceleration of Modern Data-Intensive Applications," IEEE Micro 2021.
- Oliviera+, "DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks," IEEE Access 2021.
- Gomez-Luna+, "Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture," Arxiv 2021.
- Boroumand+, "Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks," PACT 2021.
- Avoid DRAM:
- Seshadri+, "The Evicted-Address Filter: A Unified Mechanism to Address Both Cache Pollution and Thrashing," PACT 2012.
  - Pekhimenko+, "Base-Delta-Immediate Compression: Practical Data Compression for On-Chip Caches," PACT 2012.
  - Seshadri+, "The Dirty-Block Index," ISCA 2014.
  - Seshadri+, "Exploiting Compressed Block Size as an Indicator of Future Reuse," HPCA 2015.
  - Vijaykumar+, "A Case for Core-Assisted Bottleneck Acceleration in GPUs: Enabling Flexible Data Compression with Assist Warps," ISCA 2015.
  - Pekhimenko+, "Toggle-Aware Bandwidth Compression for GPUs," HPCA 2016.

# Solution 2: Emerging Memory Technologies

- Some emerging **resistive** memory technologies seem more scalable than DRAM (and they are non-volatile)
- Example: Phase Change Memory
  - Data stored by changing phase of material
  - Data read by detecting material's resistance
  - Expected to scale to 9nm (2022 [ITRS 2009])
  - Prototyped at 20nm (Raoux+, IBM JRD 2008)
  - Expected to be denser than DRAM: can store multiple bits/cell
- But, emerging technologies have (many) shortcomings
  - Can they be enabled to replace/augment/surpass DRAM?

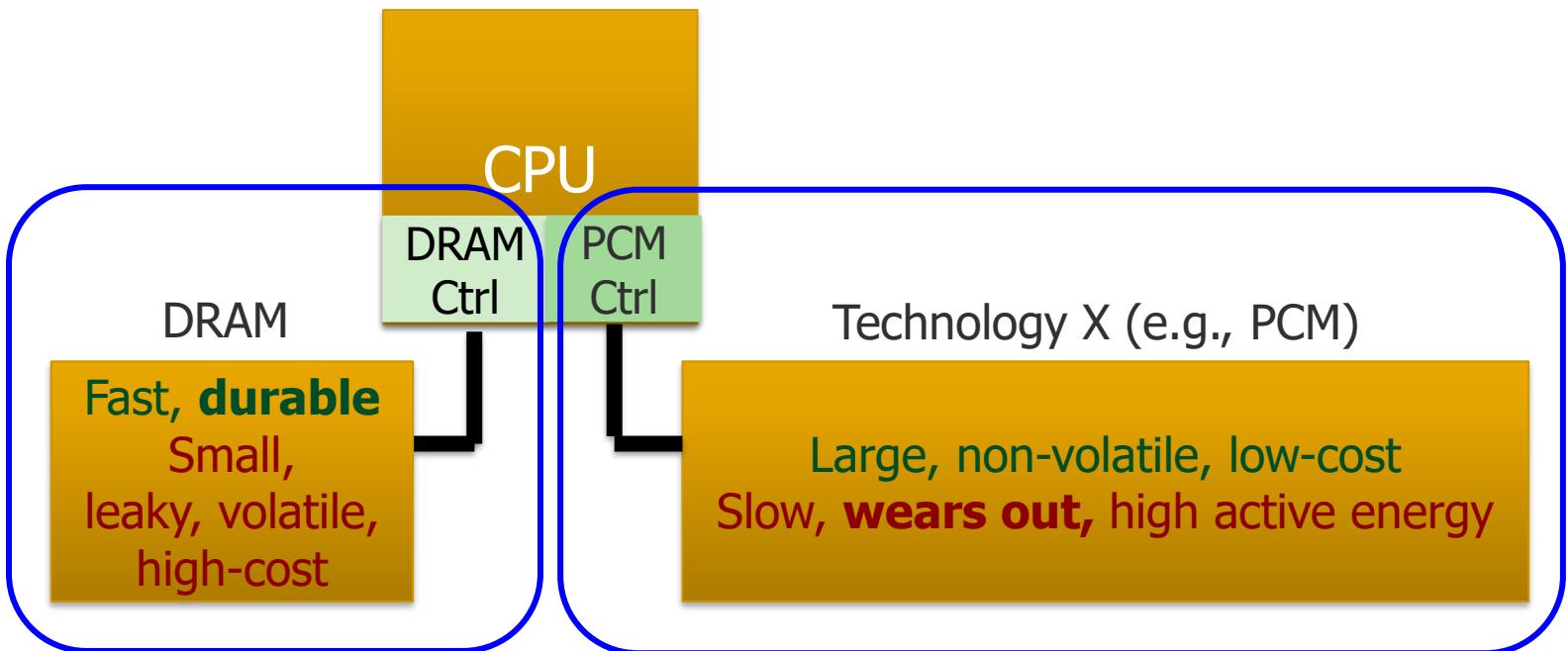


# Solution 2: Emerging Memory Technologies

---

- Lee+, “[Architecting Phase Change Memory as a Scalable DRAM Alternative](#),” ISCA’09, CACM’10, IEEE Micro’10.
- Meza+, “[Enabling Efficient and Scalable Hybrid Memories](#),” IEEE Comp. Arch. Letters 2012.
- Yoon, Meza+, “[Row Buffer Locality Aware Caching Policies for Hybrid Memories](#),” ICCD 2012.
- Kultursay+, “[Evaluating STT-RAM as an Energy-Efficient Main Memory Alternative](#),” ISPASS 2013.
- Meza+, “[A Case for Efficient Hardware-Software Cooperative Management of Storage and Memory](#),” WEED 2013.
- Lu+, “[Loose Ordering Consistency for Persistent Memory](#),” ICCD 2014.
- Zhao+, “[FIRM: Fair and High-Performance Memory Control for Persistent Memory Systems](#),” MICRO 2014.
- Yoon, Meza+, “[Efficient Data Mapping and Buffering Techniques for Multi-Level Cell Phase-Change Memories](#),” TACO 2014.
- Ren+, “[ThyNVM: Enabling Software-Transparent Crash Consistency in Persistent Memory Systems](#),” MICRO 2015.
- Chauhan+, “[NVMove: Helping Programmers Move to Byte-Based Persistence](#),” INFLOW 2016.
- Li+, “[Utility-Based Hybrid Memory Management](#),” CLUSTER 2017.
- Yu+, “[Banshee: Bandwidth-Efficient DRAM Caching via Software/Hardware Cooperation](#),” MICRO 2017.
- Tavakkol+, “[MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices](#),” FAST 2018.
- Tavakkol+, “[FLIN: Enabling Fairness and Enhancing Performance in Modern NVMe Solid State Drives](#),” ISCA 2018.
- Sadrosadati+. “[LTRF: Enabling High-Capacity Register Files for GPUs via Hardware/Software Cooperative Register Prefetching](#),” ASPLOS 2018.
- Salkhordeh+, “[An Analytical Model for Performance and Lifetime Estimation of Hybrid DRAM-NVM Main Memories](#),” TC 2019.
- Wang+, “[Panthera: Holistic Memory Management for Big Data Processing over Hybrid Memories](#),” PLDI 2019.
- Song+, “[Enabling and Exploiting Partition-Level Parallelism \(PALP\) in Phase Change Memories](#),” CASES 2019.
- Liu+, “[Binary Star: Coordinated Reliability in Heterogeneous Memory Systems for High Performance and Scalability](#),” MICRO’19.
- Song+, “[Improving Phase Change Memory Performance with Data Content Aware Access](#),” ISMM 2020.
- Yavits+, “[WoLFRaM: Enhancing Wear-Leveling and Fault Tolerance in Resistive Memories using Programmable Address Decoders](#),” ICCD 2020.
- Song+, “[Aging-Aware Request Scheduling for Non-Volatile Main Memory](#),” ASP-DAC 2021.

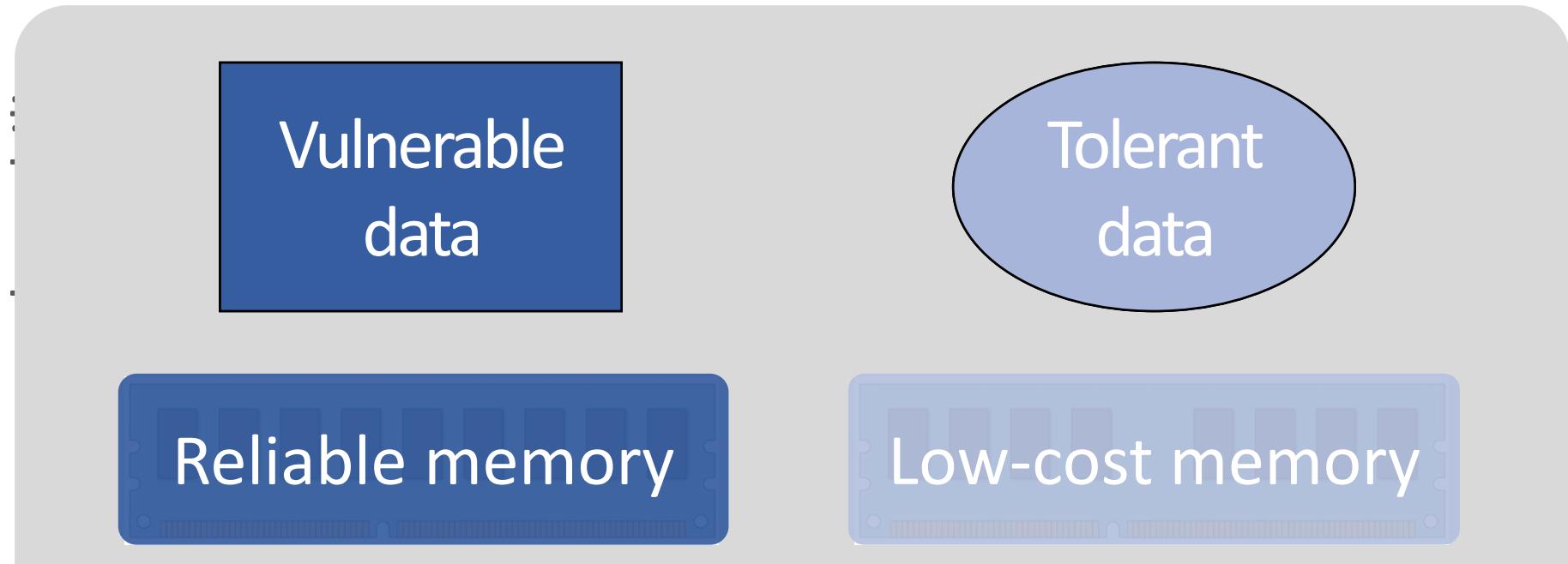
# Combination: Hybrid Memory Systems



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

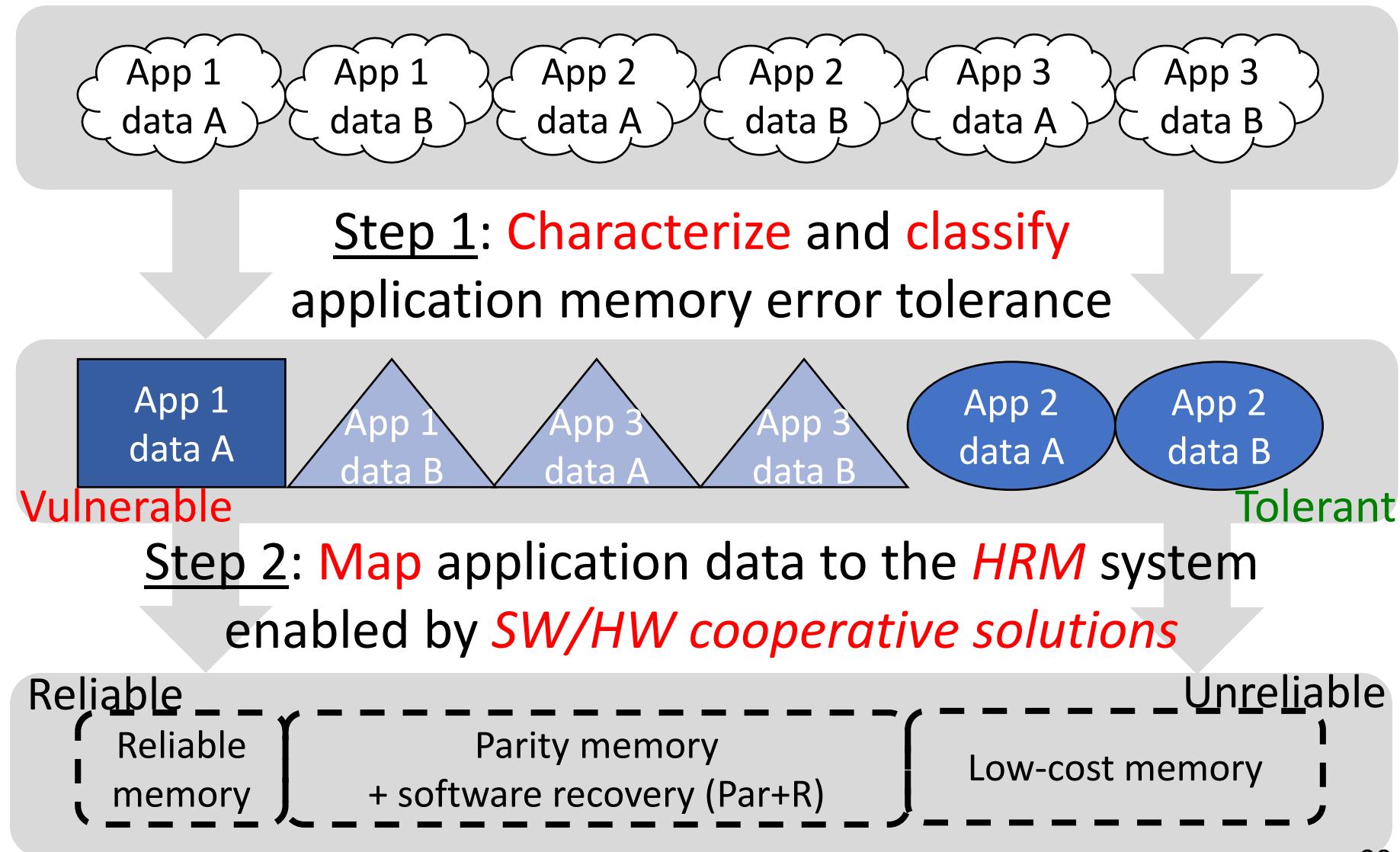
Meza+, "Enabling Efficient and Scalable Hybrid Memories," IEEE Comp. Arch. Letters, 2012.  
Yoon, Meza et al., "Row Buffer Locality Aware Caching Policies for Hybrid Memories," ICCD 2012 Best Paper Award.

# Exploiting Memory Error Tolerance with Hybrid Memory Systems

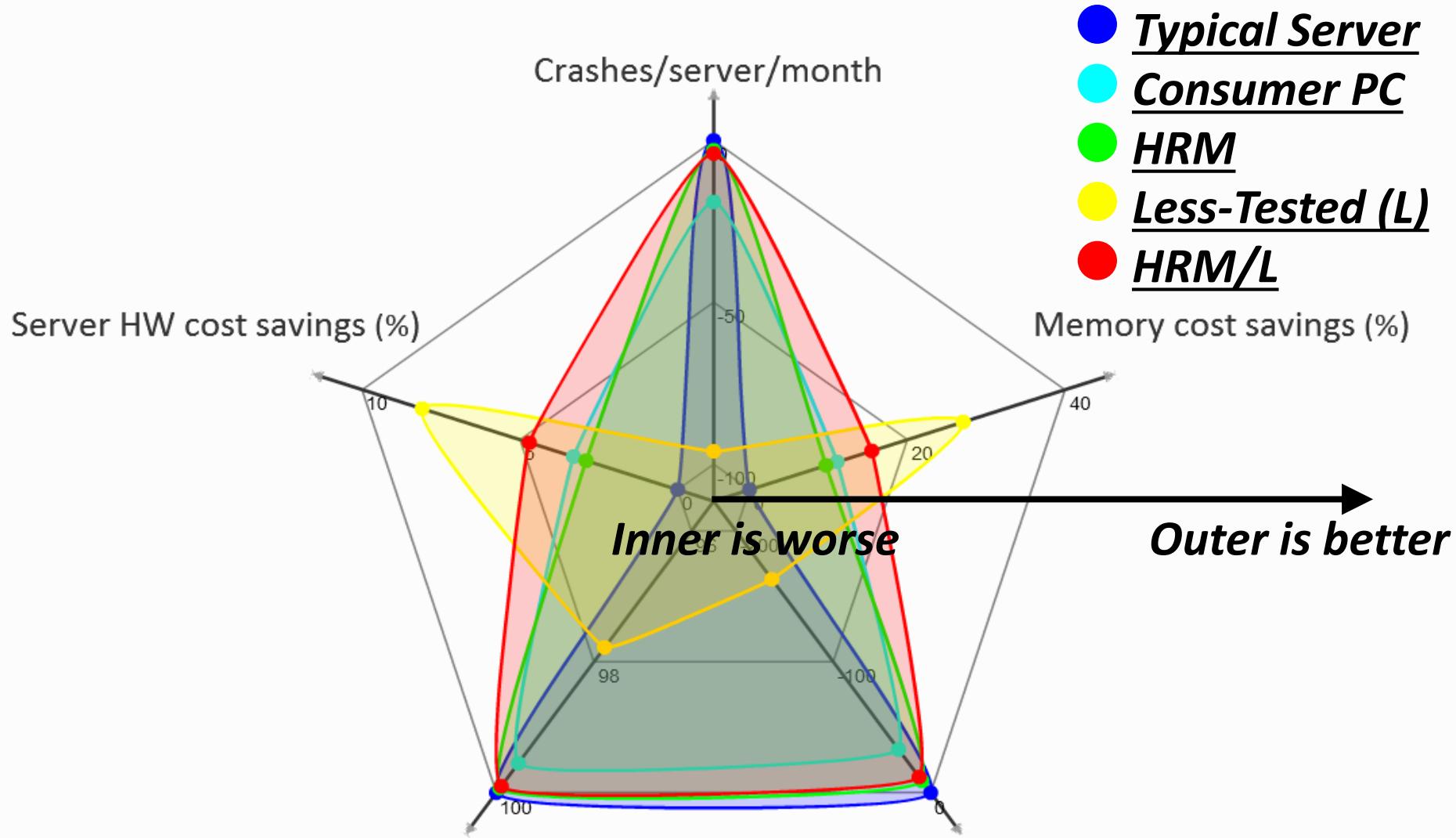


On Microsoft's Web Search workload  
Reduces server hardware **cost** by **4.7 %**  
Achieves single server **availability** target of **99.90 %**  
**Heterogeneous-Reliability Memory [DSN 2014]**

# Heterogeneous-Reliability Memory



# Evaluation Results



- ● Bigger area means better tradeoff

# 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

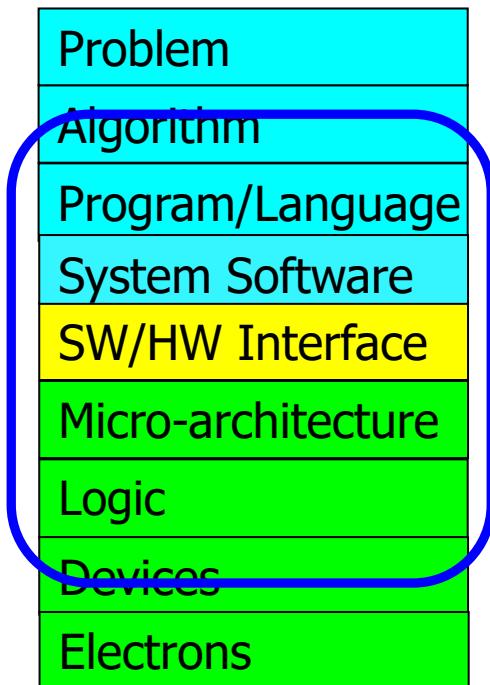
Carnegie Mellon University, [yixinluo@cs.cmu.edu](mailto:yixinluo@cs.cmu.edu), [{meza, onur}@cmu.edu](mailto:{meza, onur}@cmu.edu)

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

# HRM is an Example of Our 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**

# An Orthogonal Issue: Memory Interference

---

- Problem: **Memory interference between cores is uncontrolled**
  - unfairness, starvation, low performance
  - uncontrollable, unpredictable, vulnerable system
- Solution: **QoS-Aware Memory Systems**
  - **Hardware designed to provide a configurable fairness substrate**
    - Application-aware memory scheduling, partitioning, throttling
  - **Software designed to configure the resources to satisfy different QoS goals**
- **QoS-aware memory systems can provide predictable performance and higher efficiency**

# Strong Memory Service Guarantees

---

- Goal: Satisfy performance/SLA requirements in the presence of shared main memory, heterogeneous agents, and hybrid memory/storage
- Approach:
  - Develop techniques/models to accurately estimate the performance loss of an application/agent in the presence of resource sharing
  - Develop mechanisms (hardware and software) to enable the resource partitioning/prioritization needed to achieve the required performance levels for all applications
  - All them while providing high system performance
- Subramanian et al., "MISE: Providing Performance Predictability and Improving Fairness in Shared Main Memory Systems," HPCA 2013.
- Subramanian et al., "The Application Slowdown Model," MICRO 2015.

# MISE: Predictable Performance [HPCA'13]

---

- Lavanya Subramanian, Vivek Seshadri, Yoongu Kim, Ben Jaiyen, and Onur Mutlu,

## **"MISE: Providing Performance Predictability and Improving Fairness in Shared Main Memory Systems"**

*Proceedings of the 19th International Symposium on High-Performance Computer Architecture (HPCA), Shenzhen, China, February 2013.* [Slides \(pptx\)](#)

## MISE: Providing Performance Predictability and Improving Fairness in Shared Main Memory Systems

Lavanya Subramanian

Vivek Seshadri

Yoongu Kim

Ben Jaiyen

Onur Mutlu

Carnegie Mellon University

# ASM: Predictable Performance [MICRO'15]

---

- Lavanya Subramanian, Vivek Seshadri, Arnab Ghosh, Samira Khan, and Onur Mutlu,

**"The Application Slowdown Model: Quantifying and Controlling the Impact of Inter-Application Interference at Shared Caches and Main Memory"**

*Proceedings of the 48th International Symposium on Microarchitecture (MICRO)*, Waikiki, Hawaii, USA, December 2015.

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

[[Source Code](#)]

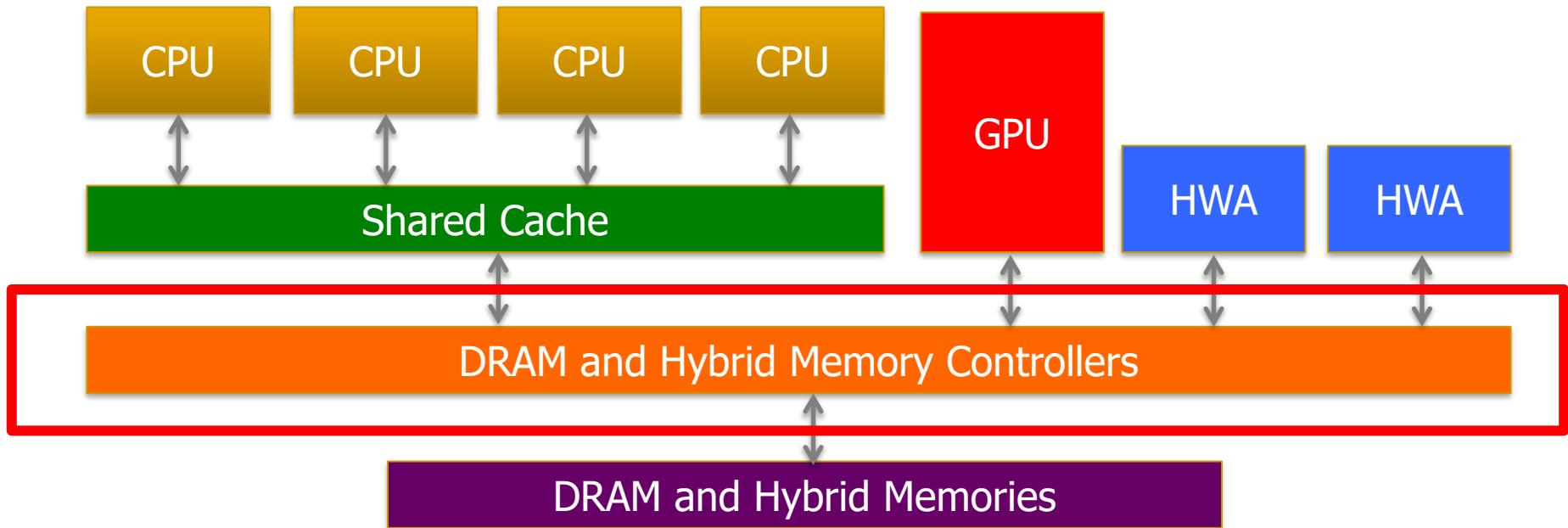
## The Application Slowdown Model: Quantifying and Controlling the Impact of Inter-Application Interference at Shared Caches and Main Memory

Lavanya Subramanian\*§      Vivek Seshadri\*      Arnab Ghosh\*†  
Samira Khan\*‡      Onur Mutlu\*

\*Carnegie Mellon University    §Intel Labs    †IIT Kanpur    ‡University of Virginia

# Memory Controllers

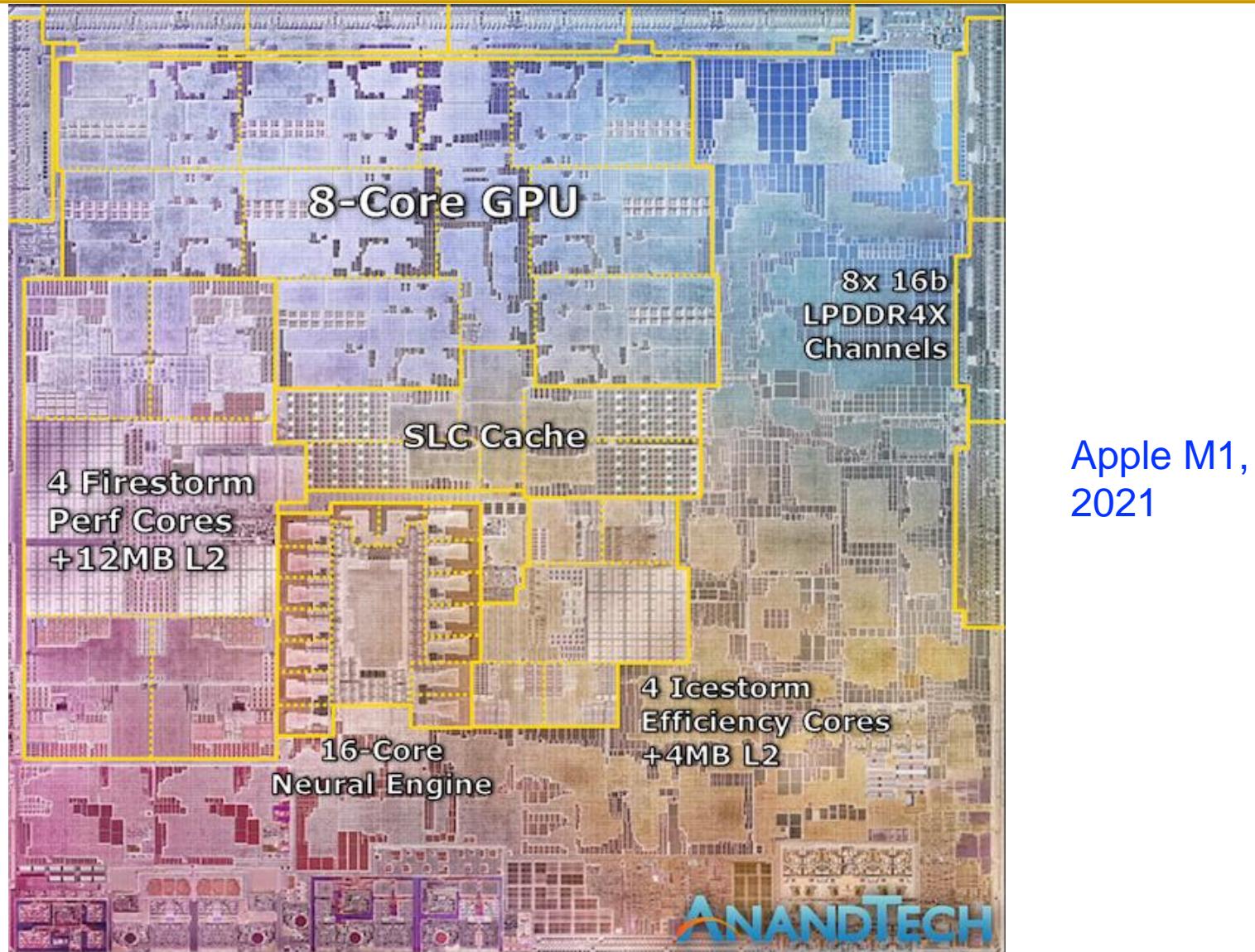
# Memory Control is Getting More Complex



- Heterogeneous agents: CPUs, GPUs, and HWAs
- Main memory interference between CPUs, GPUs, HWAs

**Many goals, many constraints, many metrics ...**

# A Similar Picture from Real Systems



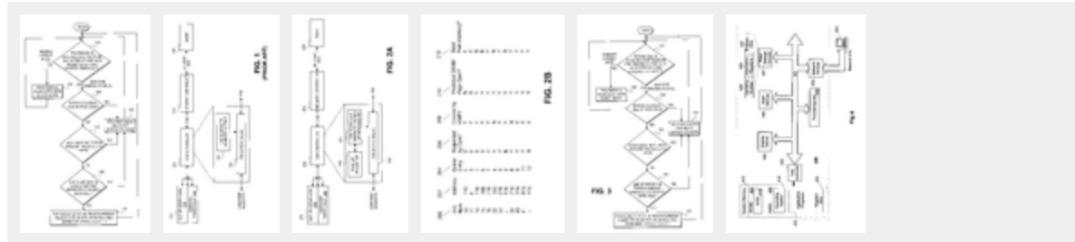
# It All Started with FSB Controllers (2001)

## Method and apparatus to control memory accesses

### Abstract

A method and apparatus for accessing memory comprising monitoring memory accesses from a hardware prefetcher and determining whether the memory accesses from the hardware prefetcher are used by an out-of-order core. A front side bus controller switches memory access modes from a minimize memory access latency mode to a maximize memory bus bandwidth mode if a percentage of the memory accesses generated by the hardware prefetcher are used by the out-of-order core.

### Images (6)



### Classifications

G06F12/0215 Addressing or allocation; Relocation with look ahead addressing means

US6799257B2

United States

[Download PDF](#)

[Find Prior Art](#)

[Similar](#)

Inventor: [Eric A. Sprangle, Onur Mutlu](#)

Current Assignee : [Intel Corp](#)

### Worldwide applications

2002 • US 2003 • AU JP DE KR CN WO GB TW 2004 • US  
2005 • HK

Application US10/079,967 events [?](#)

2002-02-21 • Application filed by Intel Corp

2002-02-21 • Priority to US10/079,967

2002-04-25 • Assigned to INTEL CORPORATION [?](#)

# Memory Performance Attacks [USENIX SEC'07]

---

- Thomas Moscibroda and Onur Mutlu,  
**"Memory Performance Attacks: Denial of Memory Service  
in Multi-Core Systems"**

*Proceedings of the 16th USENIX Security Symposium (**USENIX SECURITY**), pages 257-274, Boston, MA, August 2007.* Slides (ppt)

## **Memory Performance Attacks: Denial of Memory Service in Multi-Core Systems**

*Thomas Moscibroda    Onur Mutlu*

*Microsoft Research*

*{moscitho,onur}@microsoft.com*

# STFM [MICRO'07]

---

- Onur Mutlu and Thomas Moscibroda,  
**"Stall-Time Fair Memory Access Scheduling for Chip Multiprocessors"**

*Proceedings of the 40th International Symposium on Microarchitecture (**MICRO**)*, pages 146-158, Chicago, IL, December 2007. [[Summary](#)] [[Slides \(ppt\)](#)]

## Stall-Time Fair Memory Access Scheduling for Chip Multiprocessors

Onur Mutlu Thomas Moscibroda

Microsoft Research  
[{onur,moscitho}@microsoft.com](mailto:{onur,moscitho}@microsoft.com)

# PAR-Bs [ISCA'08]

---

- Onur Mutlu and Thomas Moscibroda,  
**"Parallelism-Aware Batch Scheduling: Enhancing both Performance and Fairness of Shared DRAM Systems"**  
*Proceedings of the 35th International Symposium on Computer Architecture (ISCA)*, pages 63-74, Beijing, China, June 2008.  
[Summary] [Slides (ppt)]

## Parallelism-Aware Batch Scheduling:

## Enhancing both Performance and Fairness of Shared DRAM Systems

Onur Mutlu Thomas Moscibroda  
Microsoft Research  
[{onur,moscitho}@microsoft.com](mailto:{onur,moscitho}@microsoft.com)

---

# On PAR-BS

---

- Variants implemented in Samsung SoC memory controllers

Effective platform level approach and DRAM accesses are crucial to system performance. This paper touches this topics and suggest a superior approach to current known techniques.

**Review from ISCA 2008**

# ATLAS Memory Scheduler [HPCA'10]

---

- Yoongu Kim, Dongsu Han, Onur Mutlu, and Mor Harchol-Balter,  
**"ATLAS: A Scalable and High-Performance Scheduling Algorithm for Multiple Memory Controllers"**

*Proceedings of the 16th International Symposium on High-Performance Computer Architecture (HPCA), Bangalore, India, January 2010.* Slides (pptx)

## ATLAS: A Scalable and High-Performance Scheduling Algorithm for Multiple Memory Controllers

Yoongu Kim   Dongsu Han   Onur Mutlu   Mor Harchol-Balter  
Carnegie Mellon University

---

# Thread Cluster Memory Scheduling [MICRO'10]

---

- Yoongu Kim, Michael Papamichael, Onur Mutlu, and Mor Harchol-Balter,

## **"Thread Cluster Memory Scheduling: Exploiting Differences in Memory Access Behavior"**

*Proceedings of the 43rd International Symposium on Microarchitecture (**MICRO**), pages 65-76, Atlanta, GA, December 2010.* [Slides \(pptx\)](#) [\(pdf\)](#)

## Thread Cluster Memory Scheduling: Exploiting Differences in Memory Access Behavior

Yoongu Kim

[yoonguk@ece.cmu.edu](mailto:yoonguk@ece.cmu.edu)

Michael Papamichael

[papamix@cs.cmu.edu](mailto:papamix@cs.cmu.edu)

Onur Mutlu

[onur@cmu.edu](mailto:onur@cmu.edu)

Mor Harchol-Balter

[harchol@cs.cmu.edu](mailto:harchol@cs.cmu.edu)

Carnegie Mellon University

# BLISS [ICCD'14, TPDS'16]

---

- Lavanya Subramanian, Donghyuk Lee, Vivek Seshadri, Harsha Rastogi, and Onur Mutlu,

## **"The Blacklisting Memory Scheduler: Achieving High Performance and Fairness at Low Cost"**

*Proceedings of the 32nd IEEE International Conference on Computer Design (ICCD), Seoul, South Korea, October 2014.*  
[Slides (pptx) (pdf)]

# The Blacklisting Memory Scheduler: Achieving High Performance and Fairness at Low Cost

Lavanya Subramanian, Donghyuk Lee, Vivek Seshadri, Harsha Rastogi, Onur Mutlu  
Carnegie Mellon University  
`{lsubrama,donghyu1,visesh,harshar,onur}@cmu.edu`

# Staged Memory Scheduling: CPU-GPU [ISCA'12]

---

- Rachata Ausavarungnirun, Kevin Chang, Lavanya Subramanian, Gabriel Loh, and Onur Mutlu,

**"Staged Memory Scheduling: Achieving High Performance and Scalability in Heterogeneous Systems"**

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

## Staged Memory Scheduling: Achieving High Performance and Scalability in Heterogeneous Systems

Rachata Ausavarungnirun<sup>†</sup> Kevin Kai-Wei Chang<sup>†</sup> Lavanya Subramanian<sup>†</sup> Gabriel H. Loh<sup>‡</sup> Onur Mutlu<sup>†</sup>

<sup>†</sup>Carnegie Mellon University

{rachata,kevincha,lsubrama,onur}@cmu.edu

<sup>‡</sup>Advanced Micro Devices, Inc.

gabe.loh@amd.com

# DASH: Heterogeneous Systems [TACO'16]

---

- Hiroyuki Usui, Lavanya Subramanian, Kevin Kai-Wei Chang, and Onur Mutlu,

**"DASH: Deadline-Aware High-Performance Memory Scheduler for Heterogeneous Systems with Hardware Accelerators"**

*ACM Transactions on Architecture and Code Optimization (TACO)*,  
Vol. 12, January 2016.

Presented at the 11th HiPEAC Conference, Prague, Czech Republic,  
January 2016.

[Slides (pptx) (pdf)]

[Source Code]

## DASH: Deadline-Aware High-Performance Memory Scheduler for Heterogeneous Systems with Hardware Accelerators

HIROYUKI USUI, LAVANYA SUBRAMANIAN, KEVIN KAI-WEI CHANG,  
and ONUR MUTLU, Carnegie Mellon University

---

# MISE: Predictable Performance [HPCA'13]

---

- Lavanya Subramanian, Vivek Seshadri, Yoongu Kim, Ben Jaiyen, and Onur Mutlu,

## **"MISE: Providing Performance Predictability and Improving Fairness in Shared Main Memory Systems"**

*Proceedings of the 19th International Symposium on High-Performance Computer Architecture (HPCA), Shenzhen, China, February 2013.* [Slides \(pptx\)](#)

## MISE: Providing Performance Predictability and Improving Fairness in Shared Main Memory Systems

Lavanya Subramanian

Vivek Seshadri

Yoongu Kim

Ben Jaiyen

Onur Mutlu

Carnegie Mellon University

# ASM: Predictable Performance [MICRO'15]

---

- Lavanya Subramanian, Vivek Seshadri, Arnab Ghosh, Samira Khan, and Onur Mutlu,

**"The Application Slowdown Model: Quantifying and Controlling the Impact of Inter-Application Interference at Shared Caches and Main Memory"**

*Proceedings of the 48th International Symposium on Microarchitecture (MICRO)*, Waikiki, Hawaii, USA, December 2015.

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

[[Source Code](#)]

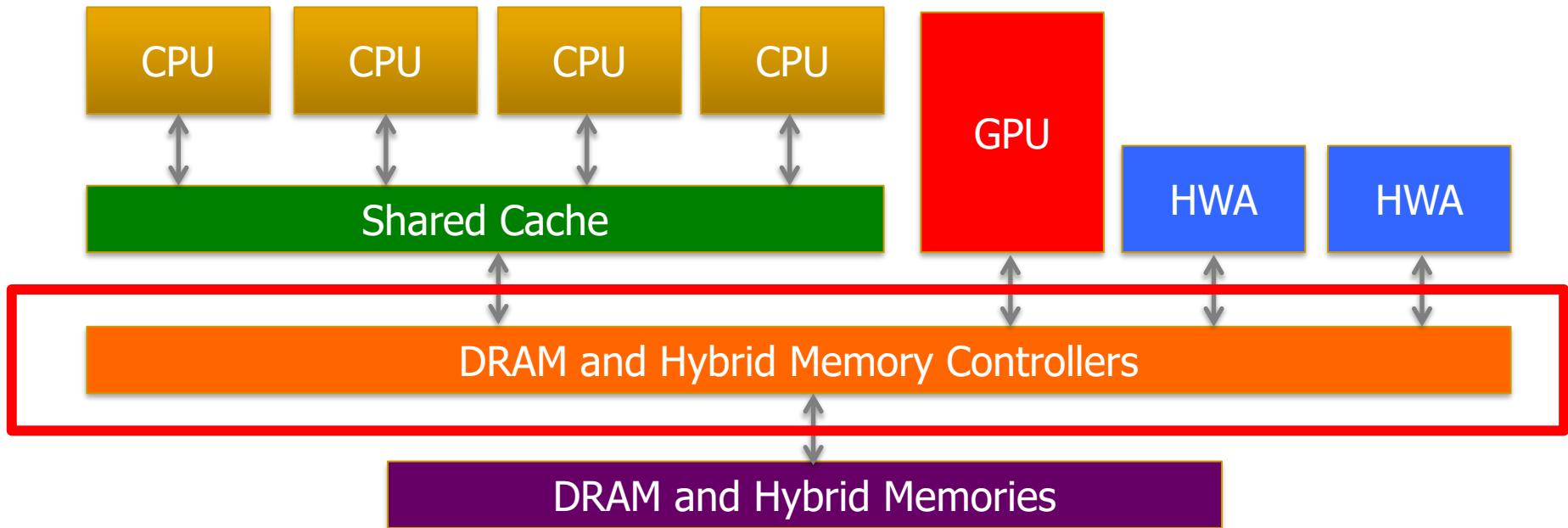
## The Application Slowdown Model: Quantifying and Controlling the Impact of Inter-Application Interference at Shared Caches and Main Memory

Lavanya Subramanian\*§      Vivek Seshadri\*      Arnab Ghosh\*†  
Samira Khan\*‡      Onur Mutlu\*

\*Carnegie Mellon University    §Intel Labs    †IIT Kanpur    ‡University of Virginia

**Memory Controllers  
are critical to research  
They will become  
even more important**

# Memory Control is Getting More Complex



- Heterogeneous agents: CPUs, GPUs, and HWAs
- Main memory interference between CPUs, GPUs, HWAs

**Many goals, many constraints, many metrics ...**

# Memory Control w/ Machine Learning [ISCA'08]

---

- 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.* [Slides \(pptx\)](#)

## 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

# Memory Controllers: Many New Problems

## Takeaway

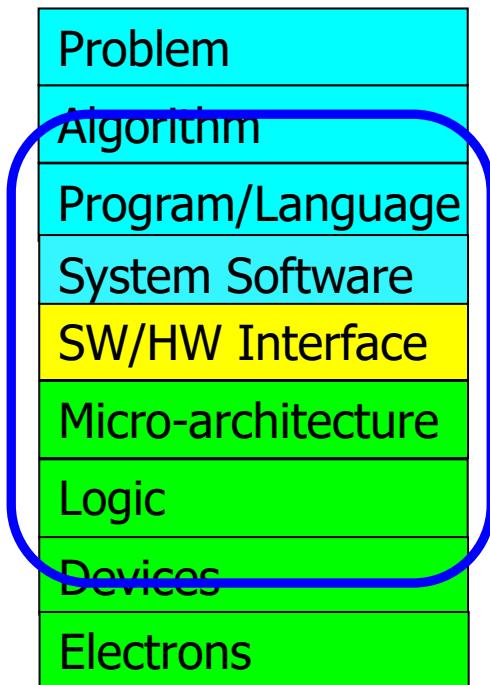
---

Main Memory Needs  
Intelligent Controllers

# We Will See More Examples of This

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**

# Recommended Interview



Interview with Onur Mutlu @ ISCA 2019 on computing research & education (after Maurice Wilkes Award)

6,749 views • Oct 19, 2019

195 ▾ 0 ▾ SHARE ▾ SAVE ▾ ...



Onur Mutlu Lectures  
19.1K subscribers

ANALYTICS

EDIT VIDEO

# Recommended Interview

---

- **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)
- 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"](#)]

# What We Will Cover in The Next Several Lectures

# Agenda for The Next Several Lectures

---

- Computation in Memory (Processing in/near Memory)
- Some Key Issues: Data Retention & Memory Interference
- RowHammer: Memory Reliability and Security
- Low-Latency Memory
- Data-Driven and Data-Aware Architectures
- Memory Controllers and Memory QoS
- Guiding Principles & Research Topics

# 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 2023*

# A Modern Primer on Processing in Memory

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

SAFARI Research Group

<sup>a</sup>ETH Zürich

<sup>b</sup>Carnegie Mellon University

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

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

---

## Abstract

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

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

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

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

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Major Trends Affecting Main Memory</b>	<b>4</b>
<b>3</b>	<b>The Need for Intelligent Memory Controllers to Enhance Memory Scaling</b>	<b>6</b>
<b>4</b>	<b>Perils of Processor-Centric Design</b>	<b>9</b>
<b>5</b>	<b>Processing-in-Memory (PIM): Technology Enablers and Two Approaches</b>	<b>11</b>
5.1	New Technology Enablers: 3D-Stacked Memory and Non-Volatile Memory . . . . .	12
5.2	Two Approaches: Processing Using Memory (PUM) vs. Processing Near Memory (PNM) . . . . .	13
<b>6</b>	<b>Processing Using Memory (PUM)</b>	<b>14</b>
6.1	RowClone . . . . .	14
6.2	Ambit . . . . .	15
6.3	SIMDRAM . . . . .	17
6.4	Gather-Scatter DRAM . . . . .	18
6.5	In-DRAM Security Primitives . . . . .	18
<b>7</b>	<b>Processing Near Memory (PNM)</b>	<b>20</b>
7.1	Tesseract: Coarse-Grained Application-Level PNM Acceleration of Graph Processing . . . . .	20
7.2	Function-Level PNM Acceleration of Mobile Consumer Workloads . . . . .	21
7.3	Programmer-Transparent Function-Level PNM Acceleration of GPU Applications . . . . .	22
7.4	Instruction-Level PNM Acceleration with PIM-Enabled Instructions (PEI) . . . . .	23
7.5	Function-Level PNM Acceleration of Genome Analysis Workloads . . . . .	24
7.6	Application-Level PNM Acceleration of Time Series Analysis . . . . .	26
<b>8</b>	<b>Enabling the Adoption of PIM</b>	<b>26</b>
8.1	Programming Models and Code Generation for PIM . . . . .	26
8.2	PIM Runtime: Scheduling and Data Mapping . . . . .	27
8.3	Memory Coherence . . . . .	29
8.4	Virtual Memory Support . . . . .	30
8.5	Data Structures for PIM . . . . .	30
8.6	Benchmarks and Simulation Infrastructures . . . . .	31
8.7	Real PIM Hardware Systems and Prototypes . . . . .	33
8.8	Security Considerations . . . . .	36
<b>9</b>	<b>Other Resources on PIM</b>	<b>37</b>
<b>10</b>	<b>Conclusion and Future Outlook</b>	<b>37</b>

---

## 1. Introduction

Main memory, built using the Dynamic Random Access Memory (DRAM) technology, is a major component in nearly all computing systems, including servers, cloud platforms, mobile/embedded devices, and sensor systems. Across all of these systems, the data working set sizes of modern applications are rapidly growing, while the need for fast analysis of such data is increasing. Thus, main memory is becoming an increasingly significant bottleneck across a wide variety of computing systems and applications [1–26]. Alleviating the main memory bottleneck requires the memory capacity, energy, cost, and performance to all scale in an efficient manner across technology generations. Unfortunately, it has become increasingly difficult in recent years, especially the past decade, to scale all of these dimensions [1, 2, 27–59], and thus the main memory bottleneck has been worsening.

A major reason for the main memory bottleneck is the high energy and latency cost associated with *data movement*. In modern computers, to perform any operation on data that resides in main memory, the processor must retrieve the data from main memory. This requires the memory controller to issue commands to a DRAM module across a relatively slow and power-hungry off-chip bus (known as the *memory channel*). The DRAM module sends the requested data across the memory channel, after which the data is placed in the caches and registers. The CPU can perform computation on the data once the data is in its registers. Data movement from the DRAM to the CPU incurs long latency and consumes a significant amount of energy [7–9, 60–64]. These costs are often exacerbated by the fact that much of the data brought into the caches is *not reused* by the CPU [62, 63, 65, 66], providing little benefit in return for the high latency and energy cost.

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

# 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 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

**SAFARI**

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

137

# An “Early” Position Paper [IMW’13]

---

- 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/>

# An Extended Version: Memory Scaling

---

- Onur Mutlu,  
**"Main Memory Scaling: Challenges and Solution Directions"**  
*Invited Book Chapter in More than Moore Technologies for Next Generation Computer Design, pp. 127-153, Springer, 2015.*

## Chapter 6

# Main Memory Scaling: Challenges and Solution Directions

*Onur Mutlu, Carnegie Mellon University*

**Part of your Homework 1 assignment**

---

# A RowHammer Survey

---

- Onur Mutlu and Jeremie Kim,

**"RowHammer: A Retrospective"**

*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.*

[Preliminary arXiv version]

[Slides from COSADE 2019 (pptx)]

[Slides from VLSI-SOC 2020 (pptx) (pdf)]

[Talk Video (1 hr 15 minutes, with Q&A)]

## RowHammer: A Retrospective

Onur Mutlu<sup>§‡</sup>

<sup>§</sup>ETH Zürich

Jeremie S. Kim<sup>†§</sup>

<sup>†</sup>Carnegie Mellon University

# A RowHammer Survey: Recent Update

---

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci,

## **"Fundamentally Understanding and Solving RowHammer"**

*Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, January 2023.*

[arXiv version]

[Slides (pptx) (pdf)]

[Talk Video (26 minutes)]

## Fundamentally Understanding and Solving RowHammer

Onur Mutlu

onur.mutlu@safari.ethz.ch

ETH Zürich

Zürich, Switzerland

Ataberk Olgun

ataberk.olgun@safari.ethz.ch

ETH Zürich

Zürich, Switzerland

A. Giray Yağlıkçı

giray.yaglikci@safari.ethz.ch

ETH Zürich

Zürich, Switzerland

**<https://arxiv.org/pdf/2211.07613.pdf>**

---

# Challenges in Memory Scaling

---

- Data retention (need for refresh)
- Reliability and vulnerabilities (e.g., RowHammer)
- Latency and parallelism (e.g., bank conflicts)
- Energy & power
- Memory's inability to do anything more than just store data

# Computer Architecture

## Lecture 2a: Memory Systems: Challenges and Opportunities

Prof. Onur Mutlu

ETH Zürich

Fall 2023

29 September 2023