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
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.
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.
Memory System: A *Shared Resource* View

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

- Memory capacity per core expected to drop by 30% every two years
- Trends worse for memory bandwidth per core!
DRAM Capacity, Bandwidth & Latency

- **Capacity**
- **Bandwidth**
- **Latency**

- 128x improvement
- 20x improvement
- 1.3x improvement
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**
[Kanev+ (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
[Kanev+ (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

VP9

Video Playback

Video Capture

Google’s video codec

Google’s video codec
Memory Bottleneck

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

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

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),
  [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

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


Runahead Execution: An Effective Alternative to Large Instruction Windows
More on Runahead Execution (I)

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

Small Window:
- Load 1 Miss
- Load 2 Miss

Runahead:
- Load 1 Miss
- Load 2 Miss
- Load 1 Hit
- Load 2 Hit

Saved Cycles

https://www.youtube.com/watch?v=zPewo6IaJ_8&list=PL5Q2soXY2Zl9xidylGxBxUz7xRPS-wisBN&index=34
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. 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.


Gwennap, “NVIDIA’s First CPU is a Winner,” MPR 2014.

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

https://www.youtube.com/watch?v=KFCOecRQTlc
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× total). We guestimated about 10× would come from CPU clock improvement, 10× from multiple instruction issue, and 10× from multiple processors.
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):

![Graph showing cache-bound cycles as a percentage for various services.]

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

Deeper and Larger Cache Hierarchies

Source: https://www.anandtech.com/show/16252/mac-mini-apple-m1-tested
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

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

Nvidia Ampere, 2020

Memory System: Most of the Platform

Most of the system is dedicated to storing and moving data

Yet, system is still bottlenecked by memory
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
A memory access consumes \( \sim 100-1000X \) the energy of a complex addition.
Energy Waste in Mobile Devices


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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹
Rachata Ausavarungnirun¹
Aki Kuusela³
Allan Knies³
Saugata Ghose¹
Eric Shiu³
Rahul Thakur³
Parthasarathy Ranganathan³
Youngsok Kim²
Daehyun Kim⁴,³
Onur Mutlu⁵,¹

SAFARI
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.

Intuition: quadratic increase in capacity
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]
Infrastructures to Understand Such Issues

SoftMC: Open Source DRAM Infrastructure


- Flexible
- Easy to Use (C++ API)
- Open-source

github.com/CMU-SAFARI/SoftMC
SoftMC

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

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

Hasan Hassan$^{1,2,3}$ Nandita Vijaykumar$^3$ Samira Khan$^{4,3}$ Saugata Ghose$^3$ Kevin Chang$^3$
Gennady Pekhimenko$^{5,3}$ Donghyuk Lee$^{6,3}$ Oguz Ergin$^2$ Onur Mutlu$^{1,3}$

$^1$ETH Zürich  $^2$TOBB University of Economics & Technology  $^3$Carnegie Mellon University
$^4$University of Virginia  $^5$Microsoft Research  $^6$NVIDIA Research
One can predictably induce errors in most DRAM memory chips
A simple hardware failure mechanism can create a widespread system security vulnerability.
RowHammer: Seven Years Ago…

- 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"
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data]
Recent DRAM Is More Vulnerable
Recent DRAM Is More Vulnerable

![Graph showing the errors per $10^9$ cells for different module vintages. The A Modules show a significant increase in errors starting in 2010, indicating a first appearance of vulnerability.]}
Recent DRAM Is More Vulnerable

All modules from 2012–2013 are vulnerable
Onur Mutlu,
"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"
[Slides (pptx) (pdf)]
RowHammer: Seven Years Ago…

- 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"


[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).*
RowHammer: 2019 and Beyond…

- Onur Mutlu and Jeremie Kim, "RowHammer: A Retrospective"
  [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§‡
§ETH Zürich

Jeremie S. Kim‡§
‡Carnegie Mellon University
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"


[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (20 minutes)]
[Lightning Talk Video (3 minutes)]

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

Jeremie S. Kim$†, Minesh Patel$, A. Giray Yaglıkçı$, Hasan Hassan$, Roknoddin Azizi$, Lois Orosa$, Onur Mutlu$†

$ETH Zürich †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, $\text{HC}_{\text{first}}$ 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 (III)

- Lucian Cojocar, Jeremie Kim, Minesh Patel, Lillian Tsai, Stefan Saroiu, Alec Wolman, and Onur Mutlu,
"Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers"
[Slides (pptx) (pdf)]
[Talk Video (17 minutes)]

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

Lucian Cojocar, Jeremie Kim§†, Minesh Patel§, Lillian Tsai‡,
Stefan Saroiu, Alec Wolman, and Onur Mutlu§†
Microsoft Research, §ETH Zürich, †CMU, ‡MIT
RowHammer in 2020 (II)

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

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


[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*

*Vrije Universiteit Amsterdam
§ETH Zürich
¶Qualcomm Technologies Inc.
Two Upcoming RowHammer Papers at MICRO 2021

- Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, Onur Mutlu, "A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses" MICRO 2021
Two Upcoming RowHammer Papers at MICRO 2021

- Hasan Hassan, Yahya Can Tuğrul, Jeremie S. Kim, Victor van der Veen, Kaveh Razavi, Onur Mutlu,

"Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications"

MICRO 2021
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

<table>
<thead>
<tr>
<th>3D-Stacked DRAM</th>
<th>higher bandwidth</th>
<th>smaller capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reduced-Latency DRAM</strong> (e.g., RL/TL-DRAM, FLY-RAM)</td>
<td>lower latency</td>
<td>higher cost</td>
</tr>
<tr>
<td><strong>Low-Power DRAM</strong> (e.g., LPDDR3, LPDDR4, Voltron)</td>
<td>lower power</td>
<td>higher latency higher cost</td>
</tr>
<tr>
<td><strong>Non-Volatile Memory (NVM)</strong> (e.g., PCM, STTRAM, ReRAM, 3D Xpoint)</td>
<td>larger capacity</td>
<td>higher latency higher dynamic power lower endurance</td>
</tr>
</tbody>
</table>
Major Trend: Hybrid Main Memory

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

Yoon+, “Row Buffer Locality Aware Caching Policies for Hybrid Memories,” ICCD 2012 Best Paper Award.
One Foreshadowing

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
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

Refesh

tWR

VRT

Call TR leak

Time
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?
One Foreshadowing

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 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**
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

- "Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost," DAC 2015.
- "A Suitable Pseudo-Memory Accelerator for Parallel Graph Processing," ISCA 2019.
- "Repair Chiplet Storage: A TinyDRAM Translation to Improve the Spatial Locality of Rarely-Used Memory," MICRO 2015.
- "Detecting and Mitigating Data Dependence Failures by Exploiting Current Memory Content," MICRO 2017.
- Kim, "The DDRIF Latency and a Theory of Predictability of the Memory Patterns in Modern DDR4 Devices," HPCA 2018.
- Han, "Fau: A Reach-Driven Mechanism for High-Throughput Memory," ISCA 2017.
- "A Low-Cost Memory Accelerator for Parallel Graph Processing," ISCA 2019.
- "Immediate Compression: Practical Data Compression for On-Demand DRAM Copy and Initialization of Bulk Data," HPCA 2019.
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

- Zhao+, “FIRM: Fair and High-Performance Memory Control for Persistent Memory Systems,” MICRO 2014.
Combination: Hybrid Memory Systems

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

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

- **Vulnerable data**
- **Tolerant data**
- **Reliable memory**
- **Low-cost memory**

On Microsoft’s Web Search workload:

- Reduces server hardware **cost** by **4.7 %**
- Achieves single server **availability** target of **99.90 %**

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

**Step 1:** Characterize and classify application memory error tolerance

**Step 2:** Map application data to the HRM system enabled by *SW/HW cooperative solutions*

Reliable memory + software recovery (Par+R)

Parity memory

Low-cost memory
Evaluation Results

- **Typical Server**
- **Consumer PC**
- **HRM**
- **Less-Tested (L)**
- **HRM/L**

Bigger area means better tradeoff

Crashes/server/month

Server HW cost savings (%)

Memory cost savings (%)

Outer is better

Inner is worse
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]
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 the while providing **high system performance**

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 …
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
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)]
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)]
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)
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"

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

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

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

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

SAFARI
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)
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 Future

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

Engin İpek$^{1,2}$  Onur Mutlu$^2$  José F. Martínez$^1$  Rich Caruana$^1$

$^1$Cornell University, Ithaca, NY 14850 USA
$^2$Microsoft Research, Redmond, WA 98052 USA
The Future

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**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>Program/Language</th>
<th>System Software</th>
<th>SW/HW Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-architecture</td>
<td>Logic</td>
<td>Devices</td>
<td>Electrons</td>
<td></td>
</tr>
</tbody>
</table>

**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)

https://www.youtube.com/watch?v=8ffSEKZhmvo
Recommended Interview

- **Computing Research and Education (@ ISCA 2019)**
  - https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2s0XY2Zi_4oP9LdL3cc8G6NIjD2Ydz

- **Maurice Wilkes Award Speech (10 minutes)**
  - https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2s0XY2Zi8D_5MGV6EnXEJHnV2YFBJl&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

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

Chapter 6
Main Memory Scaling: Challenges and Solution Directions
Onur Mutlu, Carnegie Mellon University

Part of your Homework 1 assignment
A Recent Retrospective Paper [TCAD’19]

- Onur Mutlu and Jeremie Kim, "RowHammer: A Retrospective"
  [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$^{§‡}$

§ETH Zürich

Jeremie S. Kim$^{‡§}$

‡Carnegie Mellon University
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 3a: Memory Systems: Solution Directions

Prof. Onur Mutlu
ETH Zürich
Fall 2021
7 October 2021