

# Memory Systems and Memory-Centric Computing Systems

## Part 2: RowHammer

Prof. Onur Mutlu

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

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

3 February 2020

Champery Winter School

**SAFARI**

**ETH** zürich

**Carnegie Mellon**

# Four Key Directions

---

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

# The Story of RowHammer

- One can **predictably induce bit flips** in commodity DRAM chips
  - >80% of the tested DRAM chips are vulnerable
- First example of how a **simple hardware failure mechanism** can create a **widespread system security vulnerability**

**WIRED**

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS

CULTURE

DESIGN

GEAR

SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE



SHARE  
18276



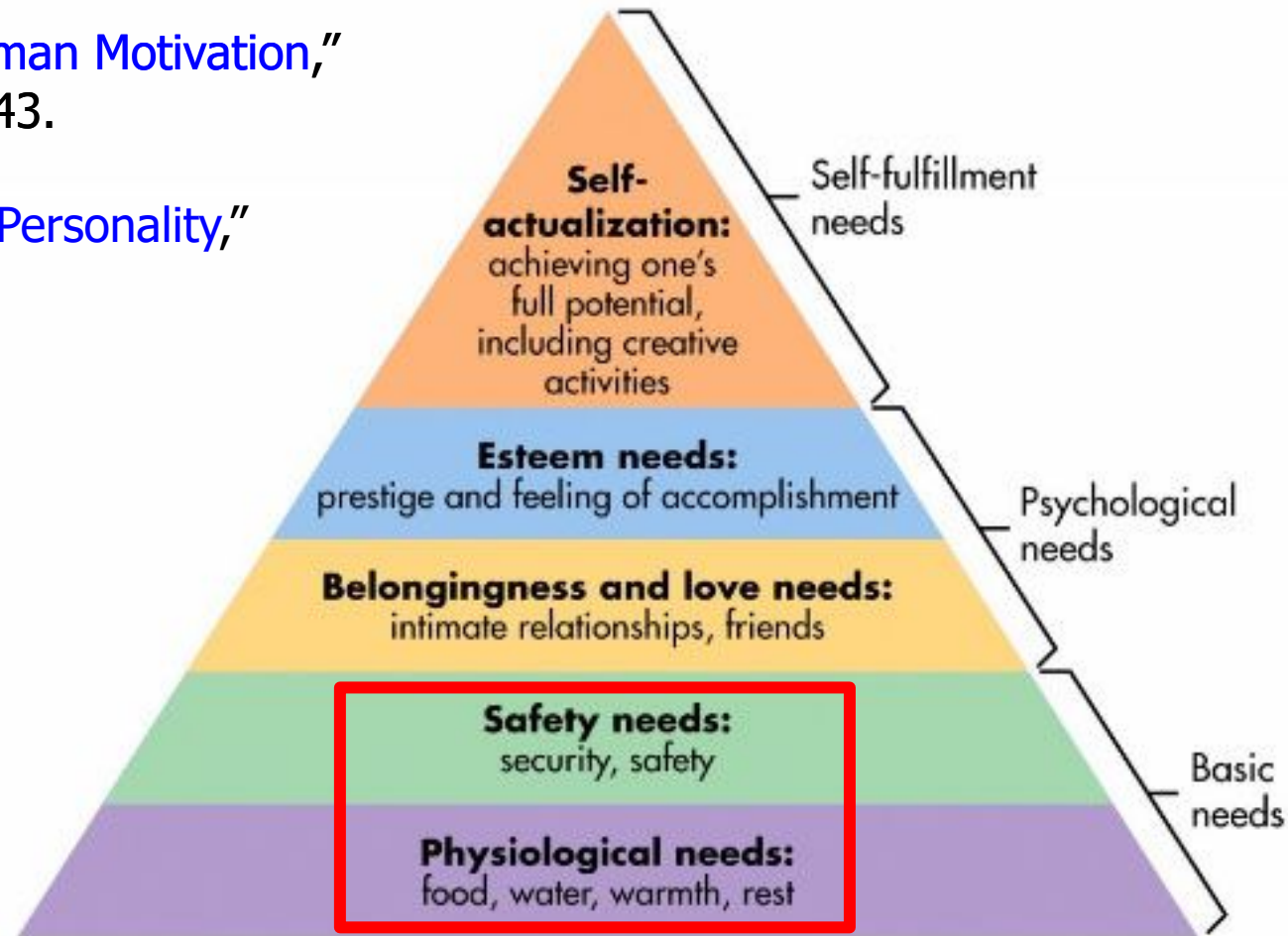
TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

# Maslow's (Human) Hierarchy of Needs

Maslow, "A Theory of Human Motivation,"  
Psychological Review, 1943.

Maslow, "Motivation and Personality,"  
Book, 1954-1970.



- We need to start with **reliability and security**...

# How Reliable/Secure/Safe is This Bridge?

---



# Collapse of the “Galloping Gertie”

---



# How Secure Are These People?

---

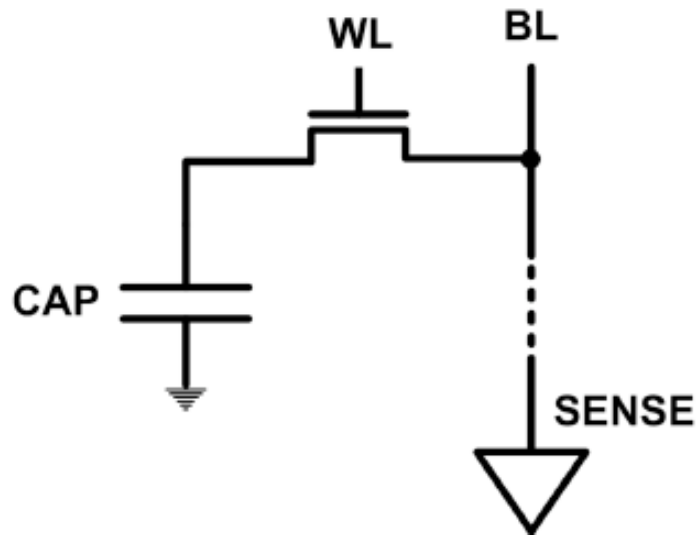


**Security is about preventing unforeseen consequences**

# 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]

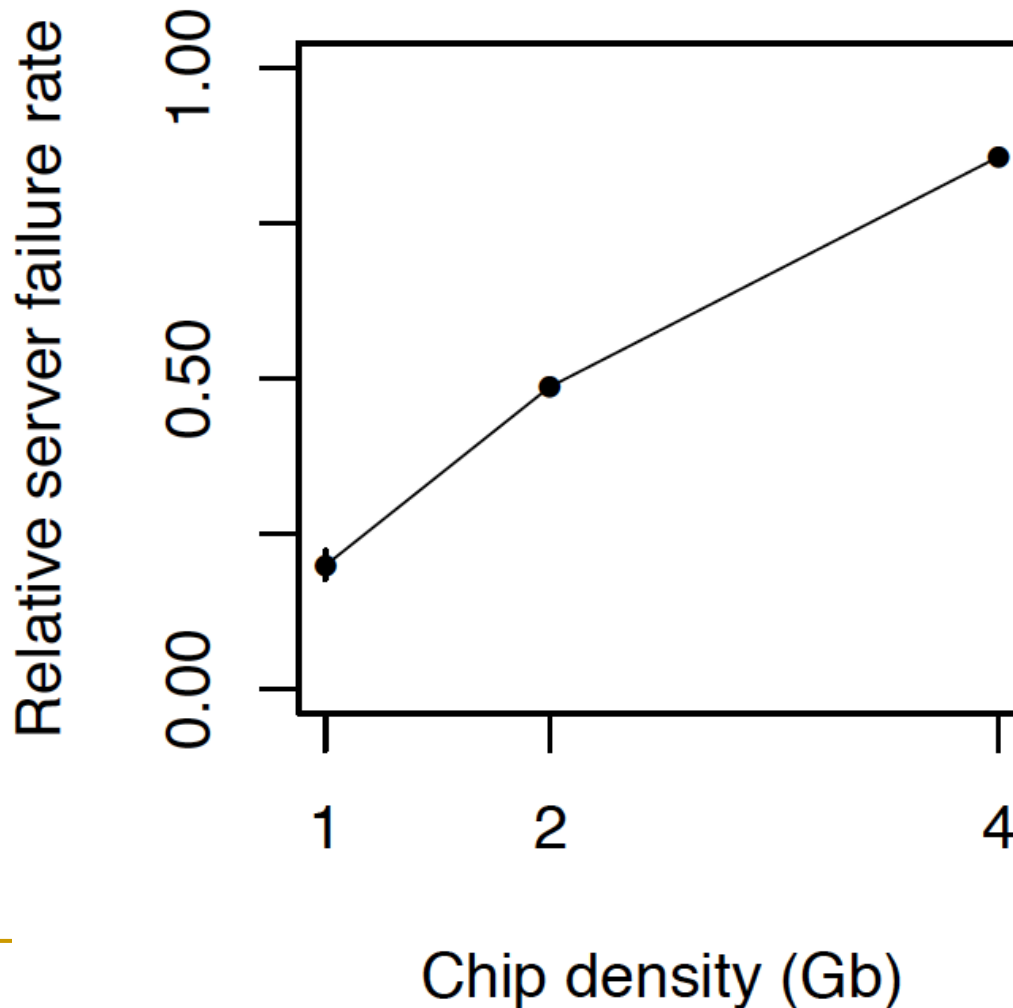


- DRAM capacity, cost, and energy/power hard to scale



# 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](#)]

## 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



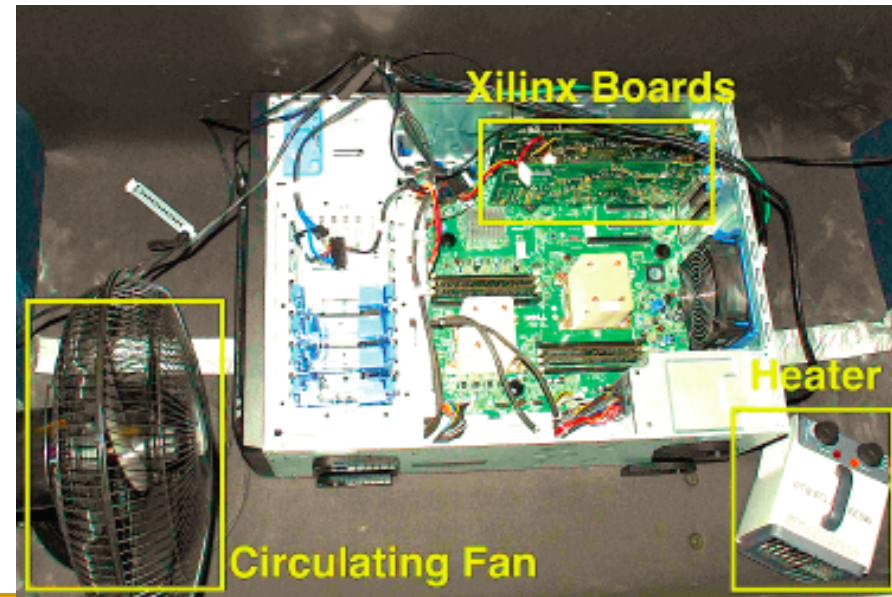
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

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

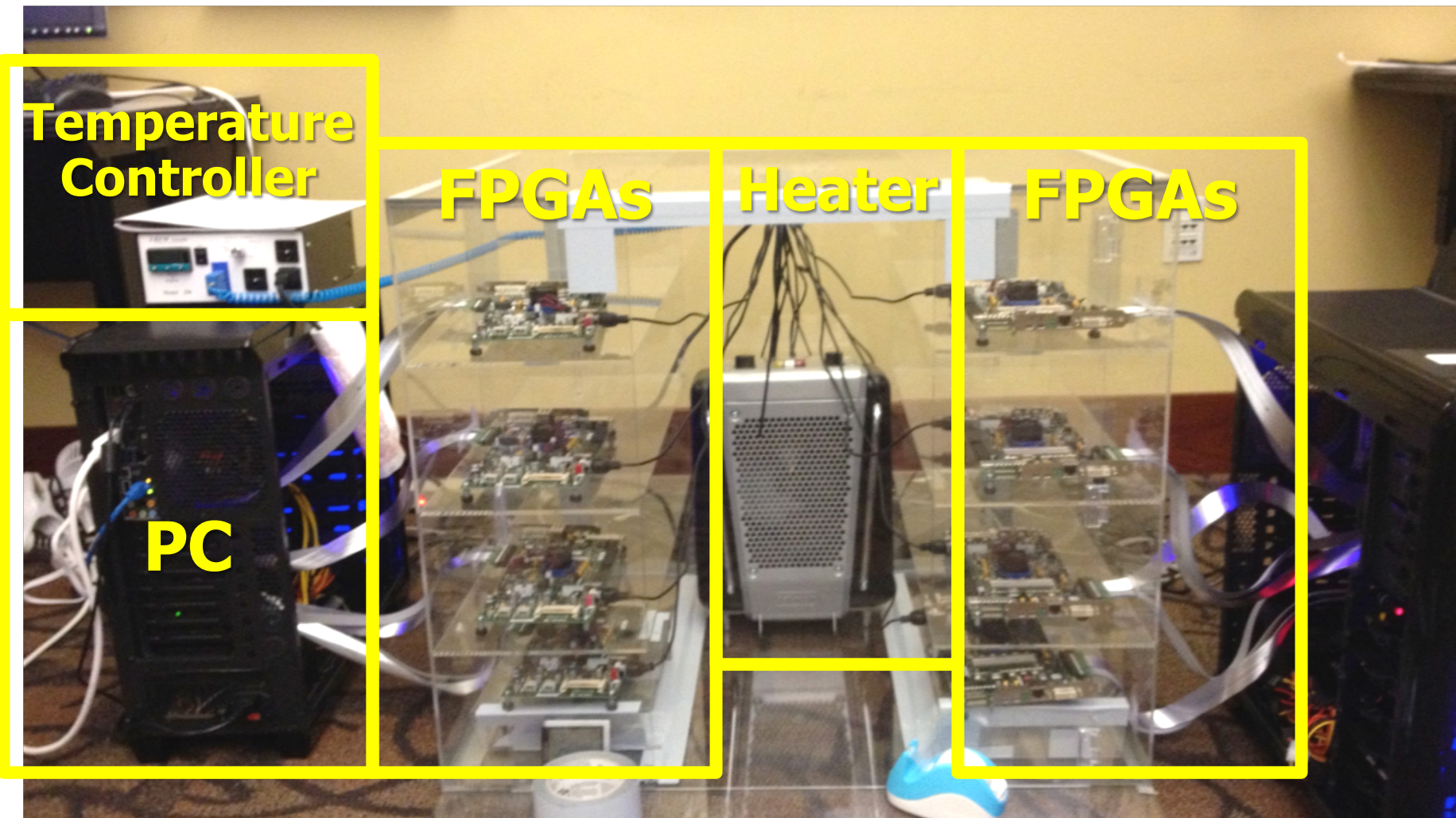
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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

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



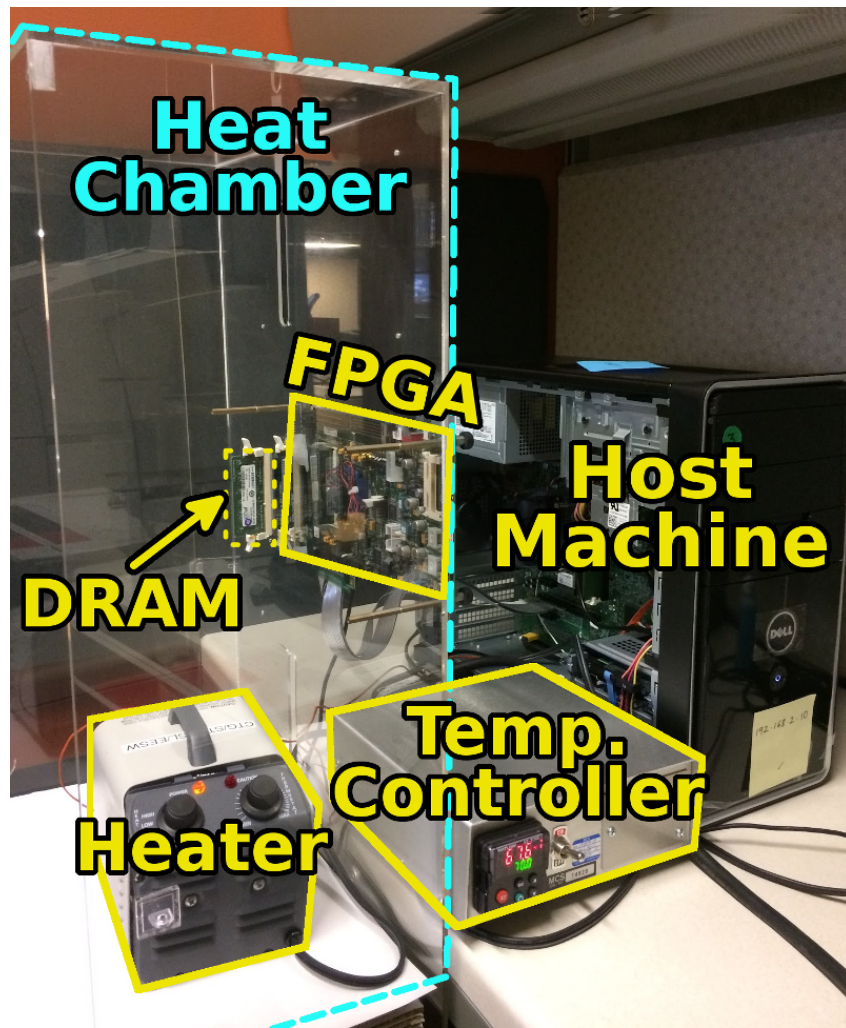
# Infrastructures to Understand Such Issues



# SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan et al., "**SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies,**" HPCA 2017.

- Flexible
- Easy to Use (C++ API)
- Open-source  
[github.com/CMU-SAFARI/SoftMC](https://github.com/CMU-SAFARI/SoftMC)



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

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

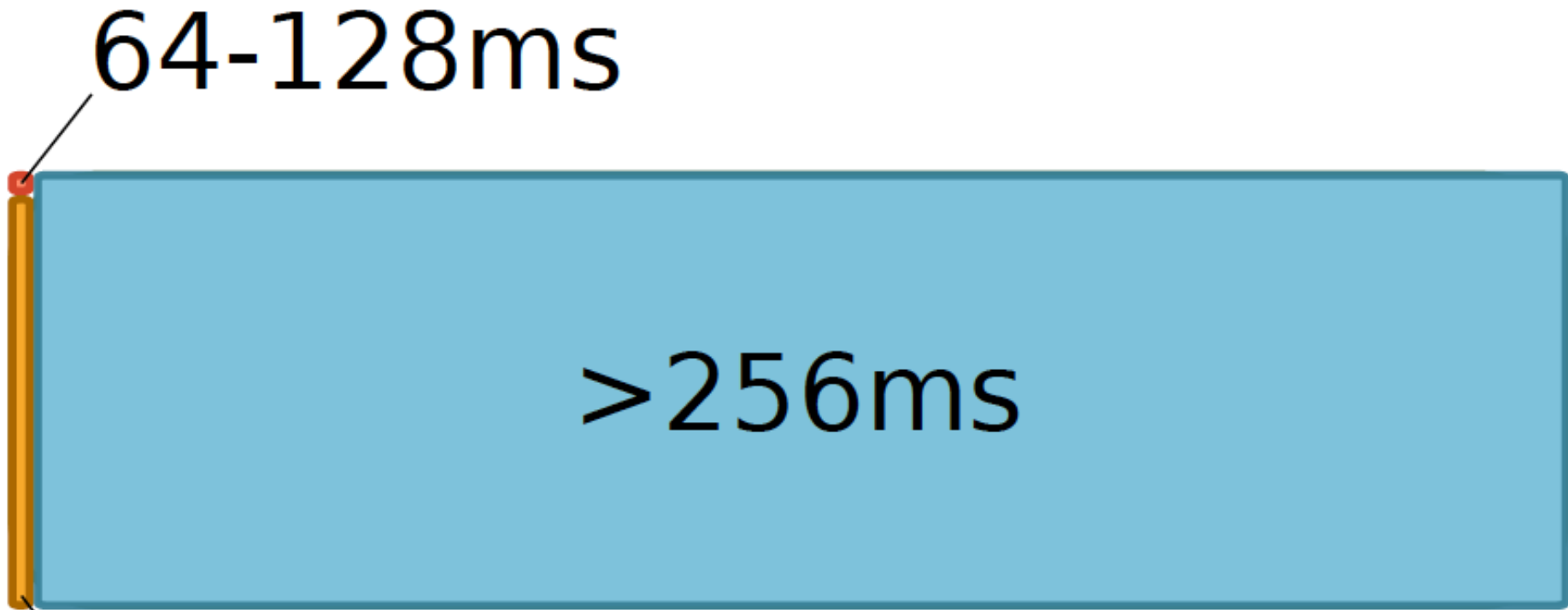
Hasan Hassan<sup>1,2,3</sup> Nandita Vijaykumar<sup>3</sup> Samira Khan<sup>4,3</sup> Saugata Ghose<sup>3</sup> Kevin Chang<sup>3</sup>  
Gennady Pekhimenko<sup>5,3</sup> Donghyuk Lee<sup>6,3</sup> Oguz Ergin<sup>2</sup> Onur Mutlu<sup>1,3</sup>

<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*

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

---

- Retention Time Profile of DRAM looks like this:



**Location** dependent  
**Stored value pattern** dependent  
**Time** dependent

# RAIDR: Heterogeneous Refresh [ISCA'12]

---

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

## **RAIDR: Retention-Aware Intelligent DRAM Refresh**

Jamie Liu    Ben Jaiyen    Richard Veras    Onur Mutlu  
Carnegie Mellon University

---



# Analysis of Data Retention Failures [ISCA'13]

---

- Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson, and Onur Mutlu, **"An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms"**  
*Proceedings of the 40th International Symposium on Computer Architecture (ISCA)*, Tel-Aviv, Israel, June 2013. [Slides \(ppt\)](#) [Slides \(pdf\)](#)

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

Jamie Liu\*

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

Ben Jaiyen\*

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

Yoongu Kim

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

Chris Wilkerson

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

Onur Mutlu

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

# Mitigation of Retention Issues [SIGMETRICS'14]

---

- Samira Khan, Donghyuk Lee, Yoongu Kim, Alaa Alameldeen, Chris Wilkerson, and Onur Mutlu,  
**"The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study"**  
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Austin, TX, June 2014. [[Slides \(pptx\)](#)] [[pdf](#)] [[Poster \(pptx\)](#)] [[pdf](#)] [[Full data sets](#)]

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

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

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

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

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

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

Onur Mutlu<sup>†</sup>  
onur@cmu.edu

<sup>†</sup>Carnegie Mellon University

<sup>\*</sup>Intel Labs

# Mitigation of Retention Issues [DSN'15]

---

- Moinuddin Qureshi, Dae Hyun Kim, Samira Khan, Prashant Nair, and Onur Mutlu,  
**"AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems"**  
*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Rio de Janeiro, Brazil, June 2015.  
[[Slides \(pptx\)](#)] [[pdf](#)]

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

Moinuddin K. Qureshi<sup>†</sup>      Dae-Hyun Kim<sup>†</sup>      Samira Khan<sup>‡</sup>      Prashant J. Nair<sup>†</sup>      Onur Mutlu<sup>‡</sup>  
<sup>†</sup>Georgia Institute of Technology      <sup>‡</sup>Carnegie Mellon University  
{*moin, dhkim, pnair6*}@ece.gatech.edu      {*samirakhan, onur*}@cmu.edu

# Mitigation of Retention Issues [DSN'16]

---

- Samira Khan, Donghyuk Lee, and Onur Mutlu,  
**"PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM"**  
*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Toulouse, France, June 2016.  
[\[Slides \(pptx\)\]](#) [\[pdf\]](#)

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

Samira Khan<sup>\*</sup>

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

Donghyuk Lee<sup>†‡</sup>

<sup>†</sup>Carnegie Mellon University

Onur Mutlu<sup>\*†</sup>

<sup>‡</sup>Nvidia

<sup>\*</sup>ETH Zürich

# Mitigation of Retention Issues [MICRO'17]

---

- Samira Khan, Chris Wilkerson, Zhe Wang, Alaa R. Alameldeen, Donghyuk Lee, and Onur Mutlu,  
**"Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content"**  
*Proceedings of the 50th International Symposium on Microarchitecture (MICRO), Boston, MA, USA, October 2017.*  
[\[Slides \(pptx\) \(pdf\)\]](#) [\[Lightning Session Slides \(pptx\) \(pdf\)\]](#) [\[Poster \(pptx\) \(pdf\)\]](#)

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

Samira Khan<sup>\*</sup> Chris Wilkerson<sup>†</sup> Zhe Wang<sup>†</sup> Alaa R. Alameldeen<sup>†</sup> Donghyuk Lee<sup>‡</sup> Onur Mutlu<sup>\*</sup>  
<sup>\*</sup>University of Virginia    <sup>†</sup>Intel Labs    <sup>‡</sup>Nvidia Research    <sup>\*</sup>ETH Zürich

# Mitigation of Retention Issues [ISCA'17]

---

- Minesh Patel, Jeremie S. Kim, and Onur Mutlu,  
**"The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions"**  
*Proceedings of the 44th International Symposium on Computer Architecture (ISCA)*, Toronto, Canada, June 2017.  
[[Slides \(pptx\)](#)] [[pdf](#)]  
[[Lightning Session Slides \(pptx\)](#)] [[pdf](#)]
- First experimental analysis of (mobile) LPDDR4 chips
- Analyzes the complex tradeoff space of retention time profiling
- Idea: enable fast and robust profiling at higher refresh intervals & temperatures

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

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

# Mitigation of Retention Issues [DSN'19]

---

- Minesh Patel, Jeremie S. Kim, Hasan Hassan, and Onur Mutlu,  
**"Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices"**  
*Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Portland, OR, USA, June 2019.  
[[Slides \(pptx\)](#)] [[pdf](#)]  
[[Talk Video](#) (26 minutes)]  
[[Full Talk Lecture](#) (29 minutes)]  
[[Source Code for EINSim, the Error Inference Simulator](#)]  
***Best paper award.***

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

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

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

# 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



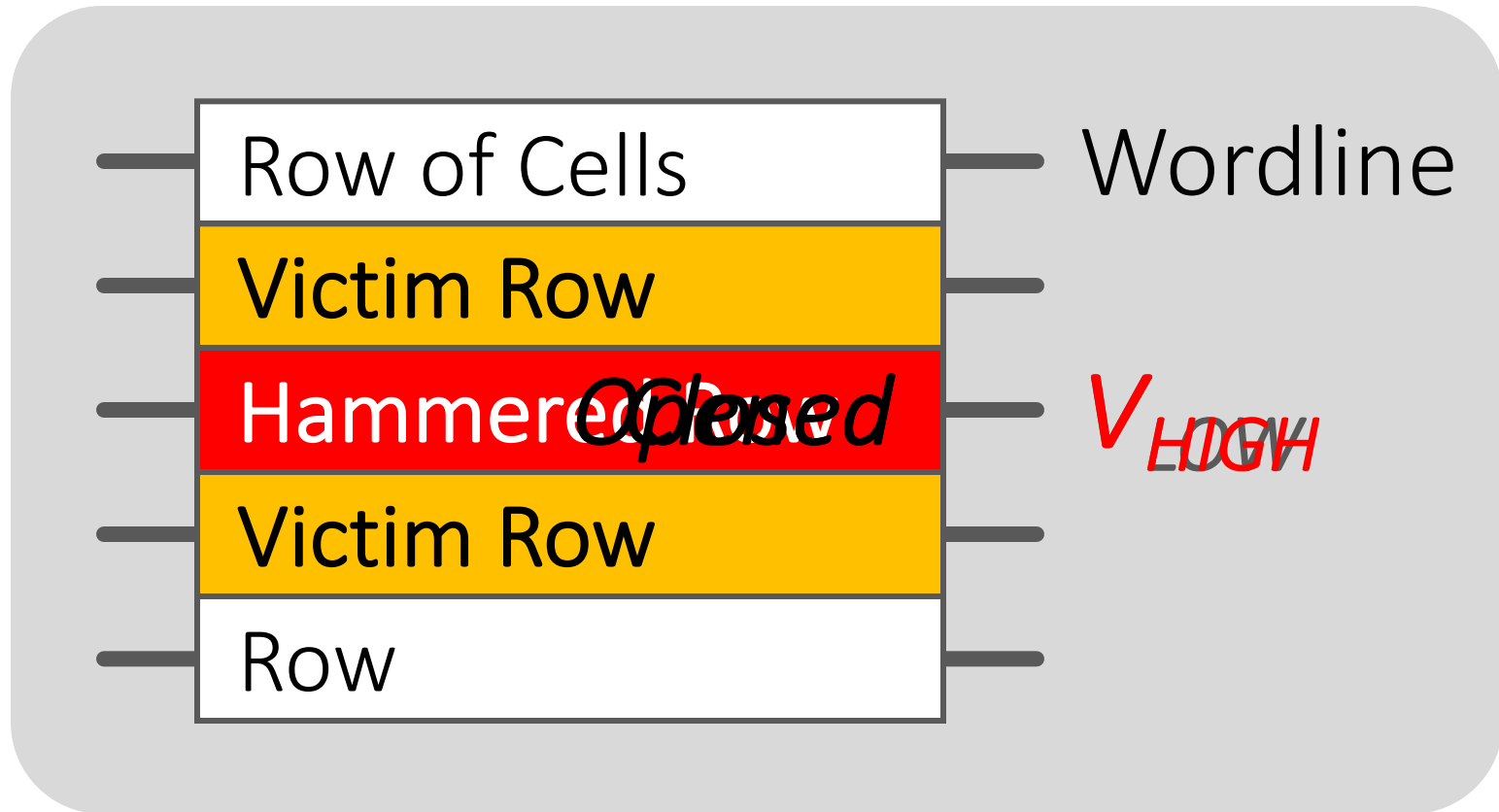
SHARE  
18276



TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

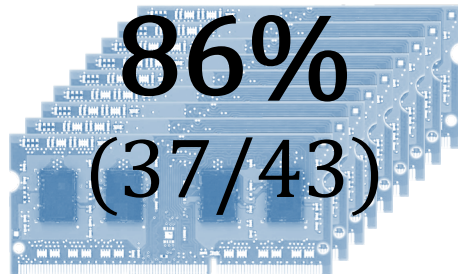
# Modern DRAM is Prone to Disturbance Errors



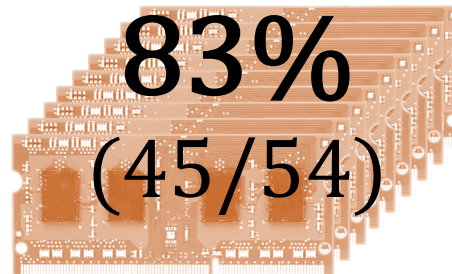
Repeatedly reading a row enough times (before memory gets refreshed) induces **disturbance errors** in adjacent rows in **most real DRAM chips you can buy today**

# Most DRAM Modules Are Vulnerable

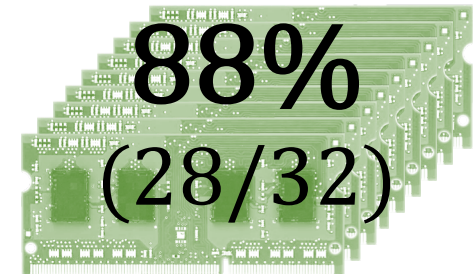
A company



B company



C company

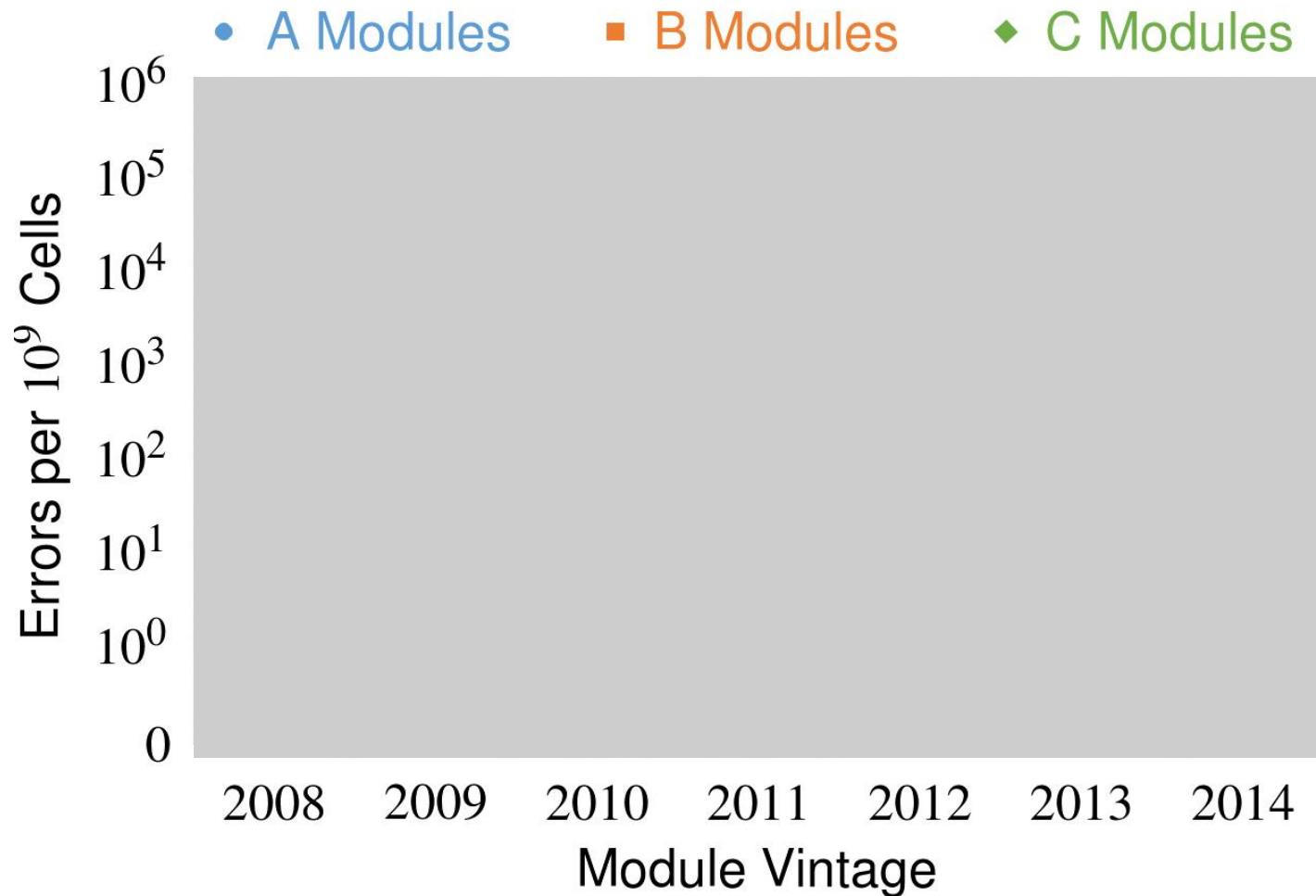


Up to  
 $1.0 \times 10^7$   
errors

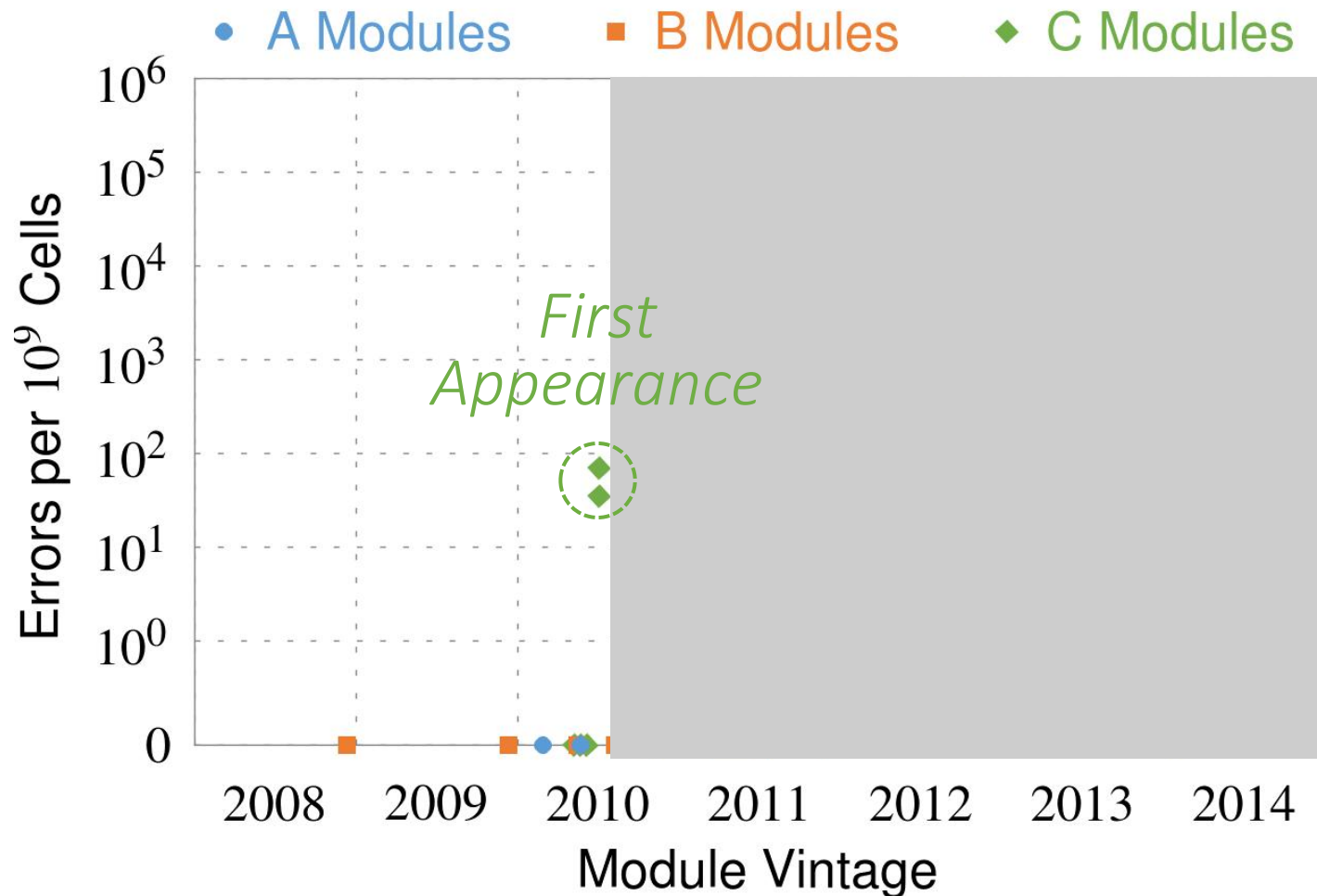
Up to  
 $2.7 \times 10^6$   
errors

Up to  
 $3.3 \times 10^5$   
errors

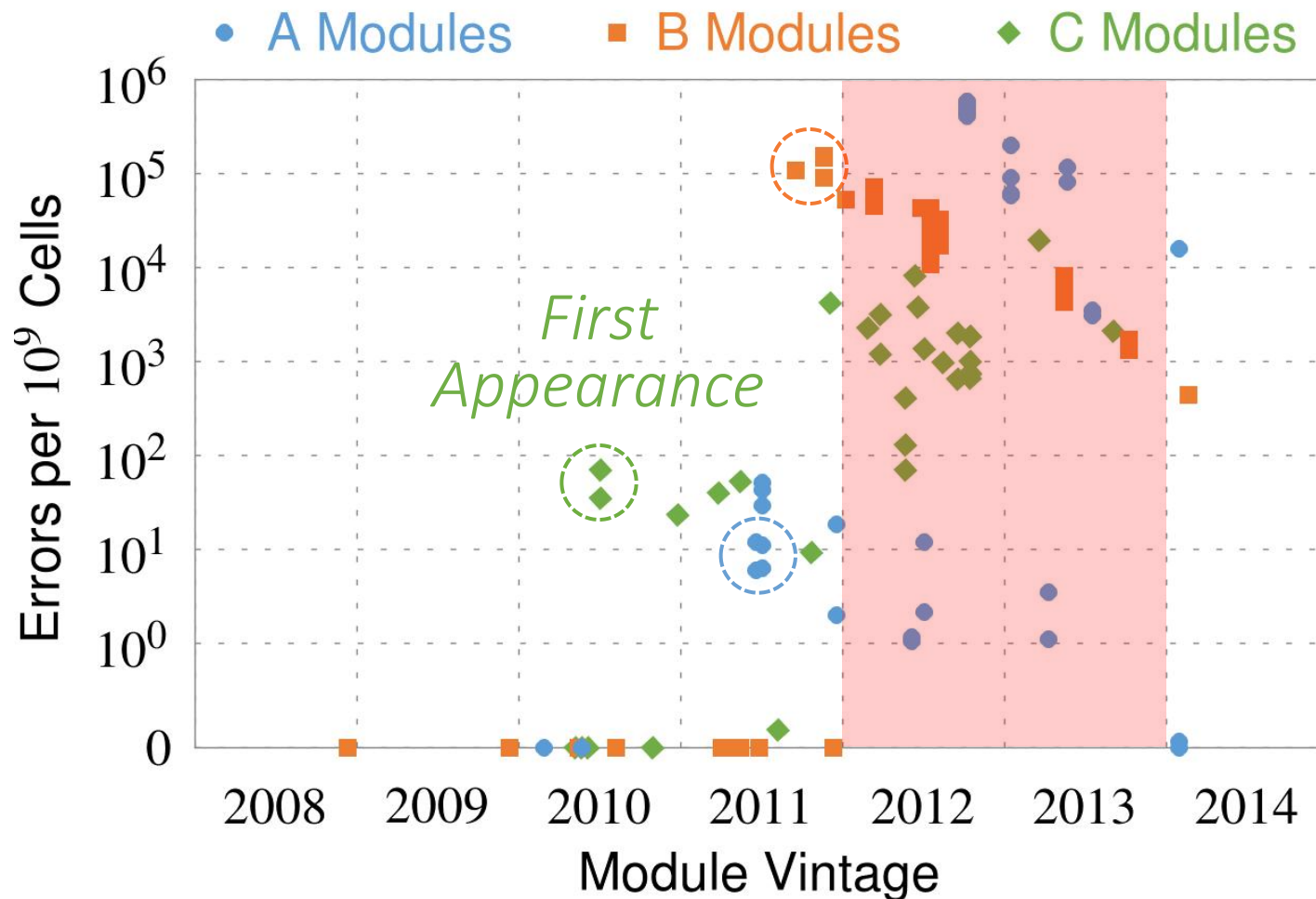
# Recent DRAM Is More Vulnerable



# Recent DRAM Is More Vulnerable



# Recent DRAM Is More Vulnerable



*All modules from 2012-2013 are vulnerable*

# Why Is This Happening?

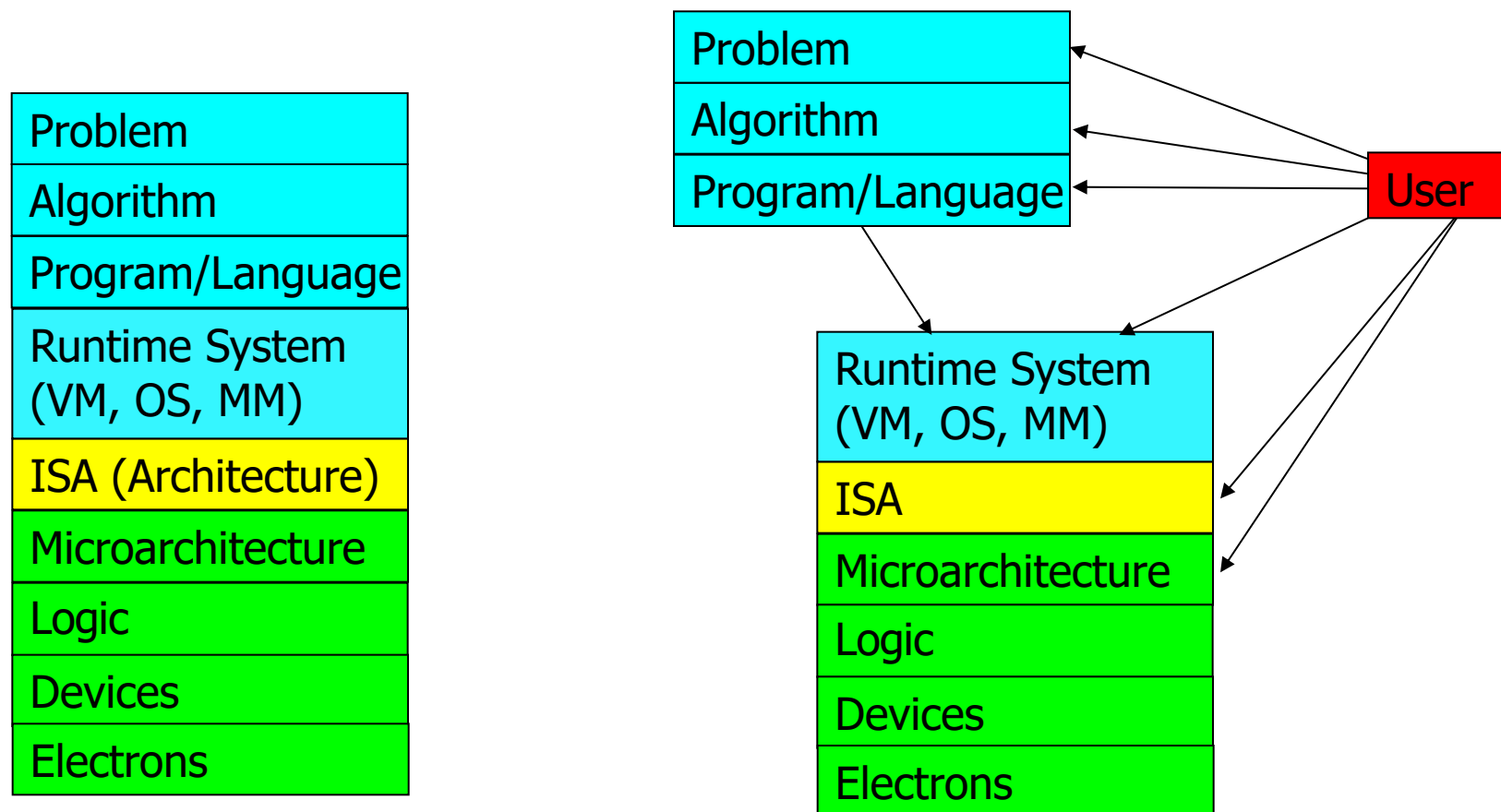
---

- DRAM cells are too close to each other!
  - They are not electrically isolated from each other
- Access to one cell affects the value in nearby cells
  - due to **electrical interference** between
    - the cells
    - wires used for accessing the cells
  - Also called cell-to-cell coupling/interference
- Example: When we activate (apply high voltage) to a row, an adjacent row gets slightly activated as well
  - Vulnerable cells in that slightly-activated row lose a little bit of charge
  - If row hammer happens enough times, charge in such cells gets drained

# Higher-Level Implications

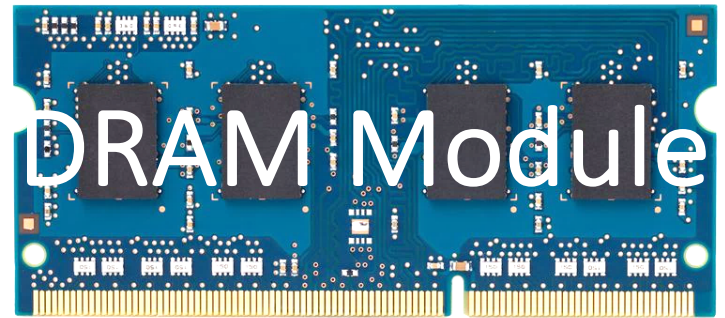
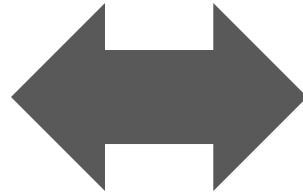
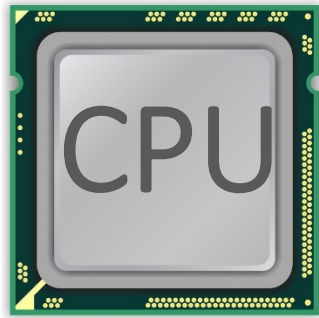
---

- This simple circuit level failure mechanism has enormous implications on upper layers of the transformation hierarchy

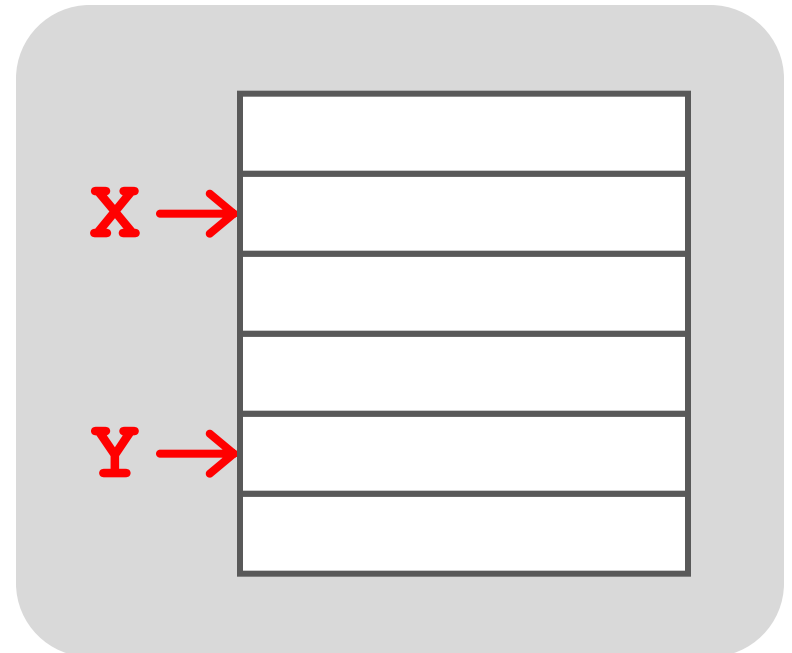




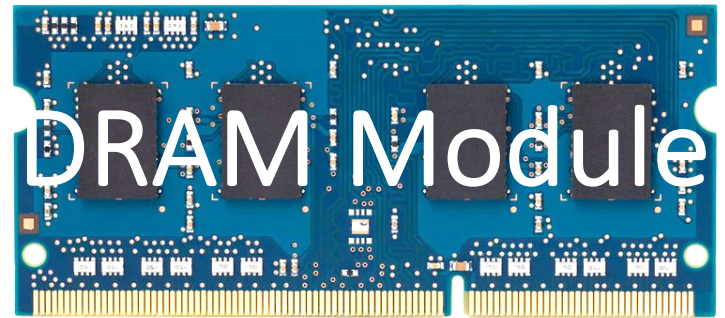
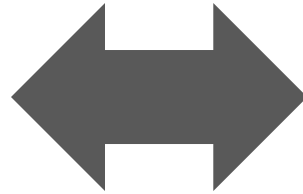
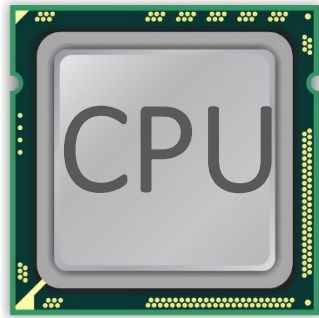
# A Simple Program Can Induce Many Errors



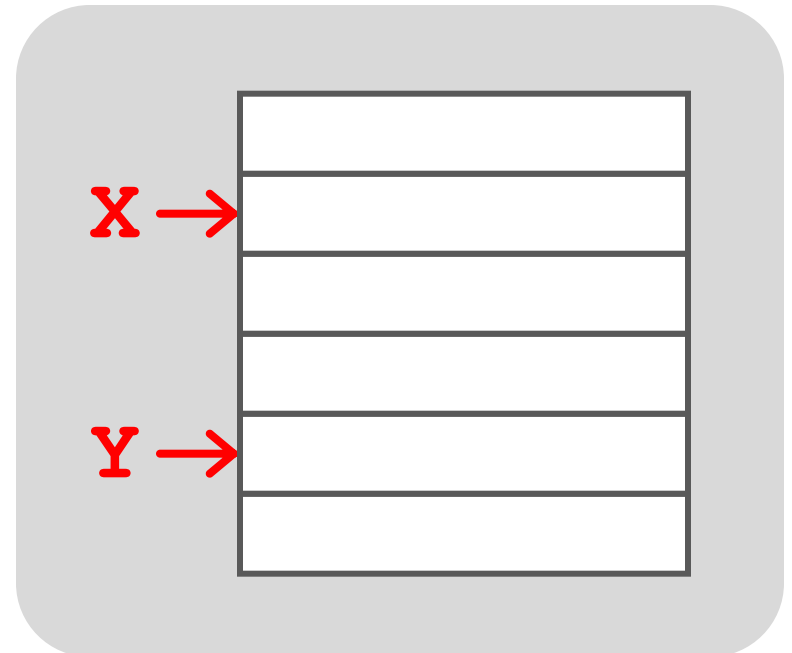
```
loop:  
  mov  (X), %eax  
  mov  (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp  loop
```



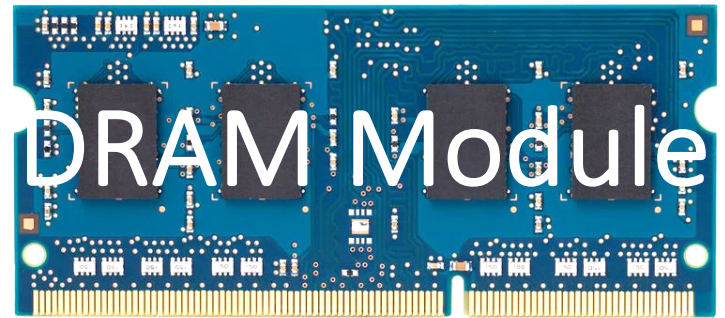
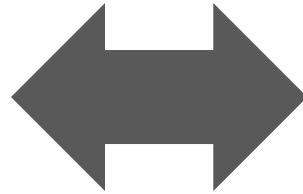
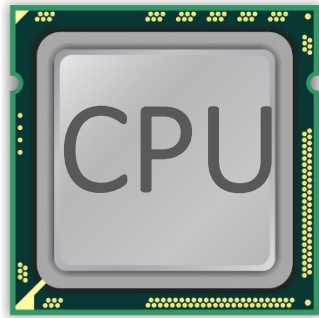
# A Simple Program Can Induce Many Errors



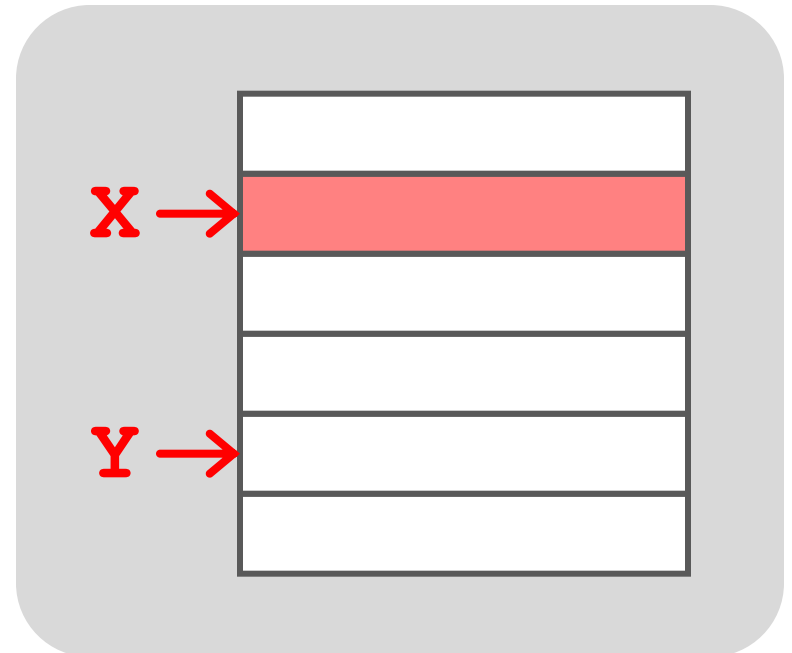
1. Avoid *cache hits*
  - Flush **X** from cache
2. Avoid *row hits* to **X**
  - Read **Y** in another row



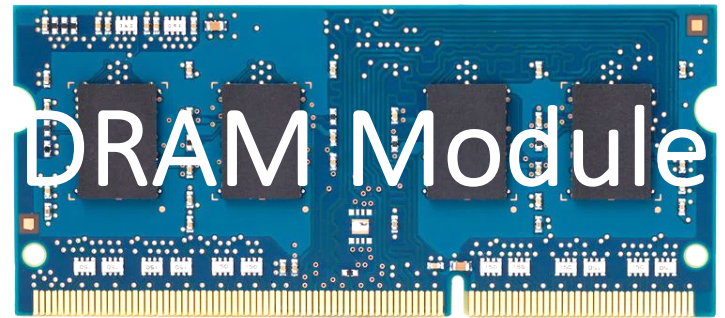
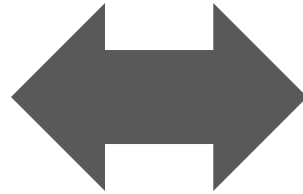
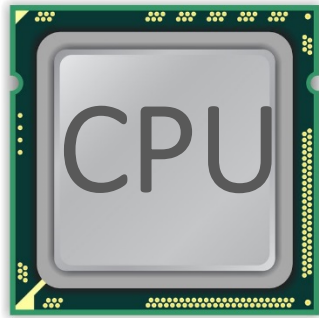
# A Simple Program Can Induce Many Errors



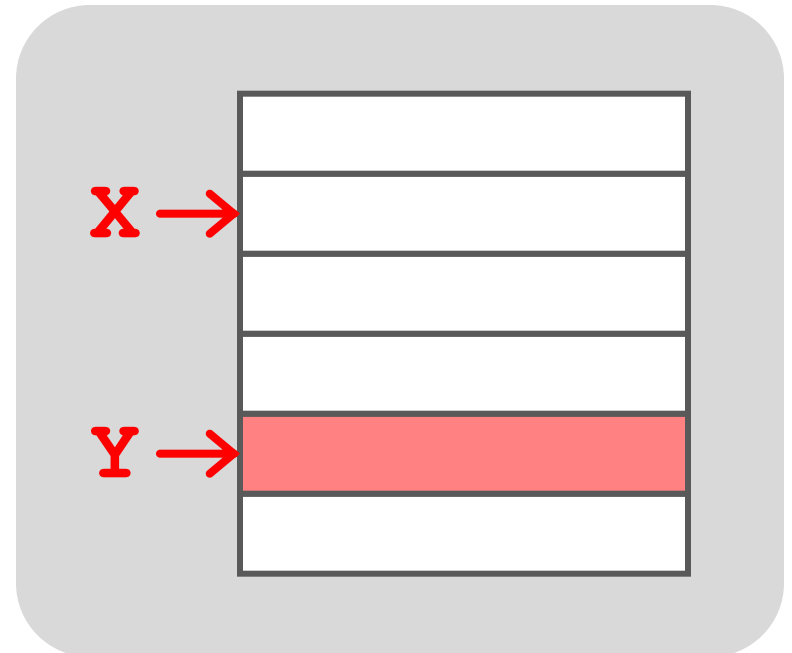
```
loop:  
  mov  (X), %eax  
  mov  (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp  loop
```



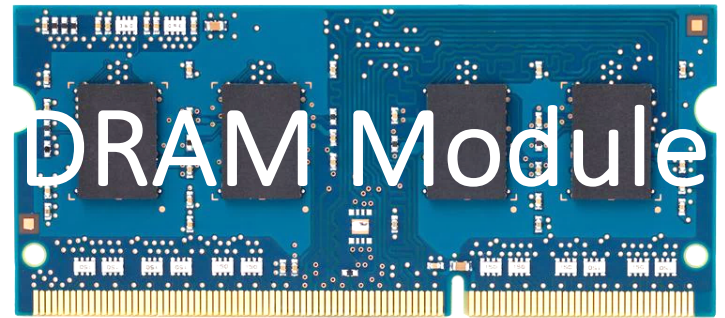
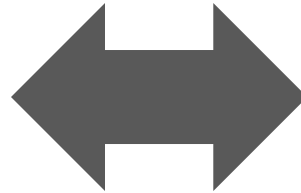
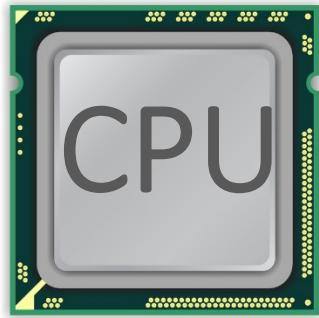
# A Simple Program Can Induce Many Errors



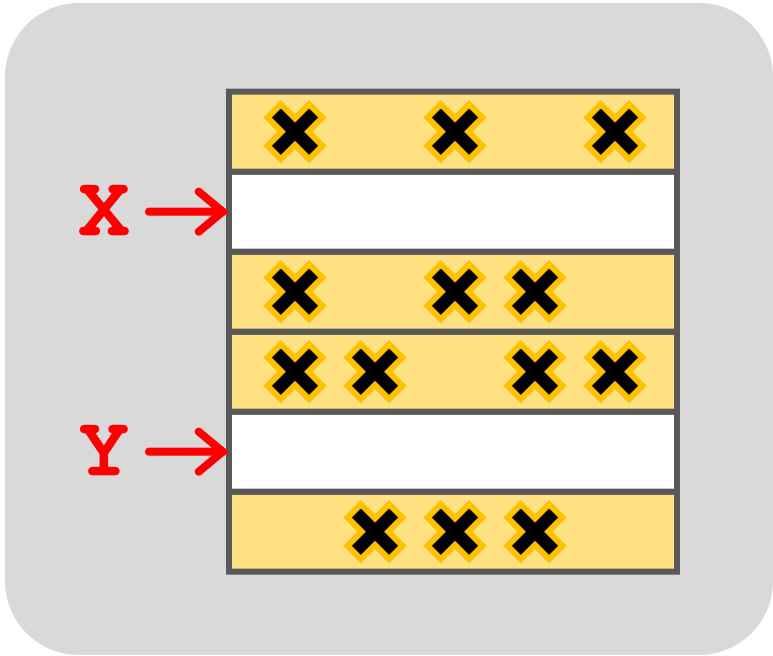
```
loop:  
  mov (X), %eax  
  mov (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp loop
```



# A Simple Program Can Induce Many Errors



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



# Observed Errors in Real Systems

CPU Architecture	Errors	Access-Rate
Intel Haswell (2013)	22.9K	12.3M/sec
Intel Ivy Bridge (2012)	20.7K	11.7M/sec
Intel Sandy Bridge (2011)	16.1K	11.6M/sec
AMD Piledriver (2012)	59	6.1M/sec

**A real reliability & security issue**

# One Can Take Over an Otherwise-Secure System

---

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

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

## Project Zero

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

News and updates from the Project Zero team at Google

[Exploiting the DRAM rowhammer bug to  
gain kernel privileges](#) (Seaborn, 2015)

Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

# RowHammer Security Attack Example

---

- “Rowhammer” is a problem with some recent DRAM devices in which repeatedly accessing a row of memory can cause bit flips in adjacent rows (Kim et al., ISCA 2014).
  - Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)
- We tested a selection of laptops and found that a subset of them exhibited the problem.
- We built two working privilege escalation exploits that use this effect.
  - Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn+, 2015)
- One exploit uses rowhammer-induced bit flips to gain kernel privileges on x86-64 Linux when run as an unprivileged userland process.
- When run on a machine vulnerable to the rowhammer problem, the process was able to induce bit flips in page table entries (PTEs).
- It was able to use this to gain write access to its own page table, and hence gain read-write access to all of physical memory.



# Security Implications



# Security Implications



**Rowhammer**

It's like breaking into an apartment by repeatedly slamming a neighbor's door until the vibrations open the door you were after

# Selected Readings on RowHammer (I)

---

- Our first detailed study: Rowhammer analysis and solutions (June 2014)
  - 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](#)]
- Our Source Code to Induce Errors in Modern DRAM Chips (June 2014)
  - <https://github.com/CMU-SAFARI/rowhammer>
- Google Project Zero's Attack to Take Over a System (March 2015)
  - [Exploiting the DRAM rowhammer bug to gain kernel privileges](#) (Seaborn+, 2015)
  - <https://github.com/google/rowhammer-test>
  - **Double-sided Rowhammer**

# Selected Readings on RowHammer (II)

---

- **Remote RowHammer Attacks via JavaScript** (July 2015)
  - <http://arxiv.org/abs/1507.06955>
  - <https://github.com/IAIK/rowhammerjs>
  - Gruss et al., DIMVA 2016.
  - **CLFLUSH-free Rowhammer**
  - “A fully automated attack that requires nothing but a website with JavaScript to **trigger faults on remote hardware.**”
  - “We can gain unrestricted access to systems of website visitors.”
  
- **ANVIL: Software-Based Protection Against Next-Generation Rowhammer Attacks** (March 2016)
  - <http://dl.acm.org/citation.cfm?doid=2872362.2872390>
  - Aweke et al., ASPLOS 2016
  - **CLFLUSH-free Rowhammer**
  - Software based monitoring for rowhammer detection

# Selected Readings on RowHammer (III)

---

- **Dedup Est Machina: Memory Deduplication as an Advanced Exploitation Vector** (May 2016)
  - <https://www.ieee-security.org/TC/SP2016/papers/0824a987.pdf>
  - Bosman et al., IEEE S&P 2016.
  - Exploits Rowhammer and Memory Deduplication to overtake a browser
  - “We report on the **first reliable remote exploit for the Rowhammer vulnerability** running entirely in Microsoft Edge.”
  - “[an attacker] ... can reliably “own” a system with all defenses up, even if the software is entirely free of bugs.”
  
- **CAN't Touch This: Software-only Mitigation against Rowhammer Attacks targeting Kernel Memory** (August 2017)
  - <https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-brasser.pdf>
  - Brasser et al., USENIX Security 2017.
  - Partitions physical memory into security domains, user vs. kernel; limits rowhammer-induced bit flips to the user domain.

# Selected Readings on RowHammer (IV)

---

- [A New Approach for Rowhammer Attacks](#) (May 2016)
  - <https://ieeexplore.ieee.org/document/7495576>
  - Qiao et al., HOST 2016
  - **CLFLUSH-free RowHammer**
  - “Libc functions memset and memcpy are found capable of rowhammer.”
  - Triggers RowHammer with malicious inputs but benign code
- [One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation](#) (August 2016)
  - [https://www.usenix.org/system/files/conference/usenixsecurity16/sec16\\_paper\\_xiao.pdf](https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_xiao.pdf)
  - Xiao et al., USENIX Security 2016.
  - **“Technique that allows a malicious guest VM to have read and write accesses to arbitrary physical pages on a shared machine.”**
  - Graph-based algorithm to reverse engineer mapping of physical addresses in DRAM

# Selected Readings on RowHammer (V)

---

- **Curious Case of RowHammer: Flipping Secret Exponent Bits using Timing Analysis** (August 2016)
  - [https://link.springer.com/content/pdf/10.1007%2F978-3-662-53140-2\\_29.pdf](https://link.springer.com/content/pdf/10.1007%2F978-3-662-53140-2_29.pdf)
  - Bhattacharya et al., CHES 2016
  - Combines timing analysis to perform **rowhammer on cryptographic keys** stored in memory
  
- **DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks** (August 2016)
  - [https://www.usenix.org/system/files/conference/usenixsecurity16/sec16\\_paper\\_pessl.pdf](https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_pessl.pdf)
  - Pessl et al., USENIX Security 2016
  - **Shows RowHammer failures on DDR4 devices despite TRR solution**
  - Reverse engineers address mapping functions to improve existing RowHammer attacks

# Selected Readings on RowHammer (VI)

---

- Flip Feng Shui: Hammering a Needle in the Software Stack (August 2016)
  - [https://www.usenix.org/system/files/conference/usenixsecurity16/sec16\\_paper\\_razavi.pdf](https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_razavi.pdf)
  - Razavi et al., USENIX Security 2016.
  - Combines memory deduplication and RowHammer
  - **"A malicious VM can gain unauthorized access to a co-hosted VM running OpenSSH."**
  - Breaks OpenSSH public key authentication
  
- Drammer: Deterministic Rowhammer Attacks on Mobile Platforms (October 2016)
  - <http://dl.acm.org/citation.cfm?id=2976749.2978406>
  - Van Der Veen et al., ACM CCS 2016
  - **Can take over an ARM-based Android system deterministically**
  - Exploits predictable physical memory allocator behavior
    - Can deterministically place security-sensitive data (e.g., page table) in an attacker-chosen, vulnerable location in memory



# Selected Readings on RowHammer (VII)

---

- **When Good Protections go Bad: Exploiting anti-DoS Measures to Accelerate Rowhammer Attacks** (May 2017)
  - <https://web.eecs.umich.edu/~misiker/resources/HOST-2017-Misiker.pdf>
  - Aga et al., HOST 2017
  - "A virtual-memory based cache-flush free attack that is sufficiently fast to **rowhammer with double rate refresh.**"
  - Enabled by Cache Allocation Technology
- **SGX-Bomb: Locking Down the Processor via Rowhammer Attack** (October 2017)
  - <https://dl.acm.org/citation.cfm?id=3152709>
  - Jang et al., SysTEX 2017
  - "Launches the Rowhammer attack against enclave memory to trigger the processor lockdown."
  - **Running unknown enclave programs on the cloud can shut down servers shared with other clients.**

# Selected Readings on RowHammer (VIII)

---

- [Another Flip in the Wall of Rowhammer Defenses](#) (May 2018)
  - <https://arxiv.org/pdf/1710.00551.pdf>
  - Gruss et al., IEEE S&P 2018
  - **A new type of Rowhammer attack which only hammers one single address**, which can be done without knowledge of physical addresses and DRAM mappings
  - Defeats static analysis and performance counter analysis defenses by running inside an SGX enclave
  
- [GuardION: Practical Mitigation of DMA-Based Rowhammer Attacks on ARM](#) (June 2018)
  - [https://link.springer.com/chapter/10.1007/978-3-319-93411-2\\_5](https://link.springer.com/chapter/10.1007/978-3-319-93411-2_5)
  - Van Der Veen et al., DIMVA 2018
  - Presents RAMPAGE, a DMA-based RowHammer attack against the latest Android OS

# Selected Readings on RowHammer (IX)

---

- Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU (May 2018)
  - <https://www.vusec.net/wp-content/uploads/2018/05/glitch.pdf>
  - Frigo et al., IEEE S&P 2018.
  - The first end-to-end remote Rowhammer exploit on mobile platforms that use our GPU-based primitives in orchestration to **compromise browsers on mobile devices in under two minutes.**
  
- Throwhammer: Rowhammer Attacks over the Network and Defenses (July 2018)
  - [https://www.cs.vu.nl/~herbertb/download/papers/throwhammer\\_atc18.pdf](https://www.cs.vu.nl/~herbertb/download/papers/throwhammer_atc18.pdf)
  - Tatar et al., USENIX ATC 2018.
  - “[We] show that **an attacker can trigger and exploit Rowhammer bit flips directly from a remote machine by only sending network packets.**”

# Selected Readings on RowHammer (X)

---

- **Nethammer: Inducing Rowhammer Faults through Network Requests** (July 2018)
  - <https://arxiv.org/pdf/1805.04956.pdf>
  - Lipp et al., arxiv.org 2018.
  - “Nethammer is the first truly **remote Rowhammer attack**, without a single attacker-controlled line of code on the targeted system.”
  
- **ZebRAM: Comprehensive and Compatible Software Protection Against Rowhammer Attacks** (October 2018)
  - <https://www.usenix.org/system/files/osdi18-konoth.pdf>
  - Konoth et al., OSDI 2018
  - A new pure-software protection mechanism against RowHammer.

# Selected Readings on RowHammer (XI.A)

---

- PassMark Software, memtest86, since 2014
  - <https://www.memtest86.com/troubleshooting.htm#hammer>

## Why am I only getting errors during Test 13 Hammer Test?

The Hammer Test is designed to detect RAM modules that are susceptible to disturbance errors caused by charge leakage. This phenomenon is characterized in the research paper **Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors** by Yoongu Kim et al. According to the research, a significant number of RAM modules manufactured 2010 or newer are affected by this defect. In simple terms, susceptible RAM modules can be subjected to disturbance errors when repeatedly accessing addresses in the same memory bank but different rows in a short period of time. Errors occur when the repeated access causes charge loss in a memory cell, before the cell contents can be refreshed at the next DRAM refresh interval.

Starting from MemTest86 v6.2, the user may see a warning indicating that the RAM may be vulnerable to high frequency row hammer bit flips. This warning appears when errors are detected during the first pass (maximum hammer rate) but no errors are detected during the second pass (lower hammer rate). See **MemTest86 Test Algorithms** for a description of the two passes that are performed during the Hammer Test (Test 13). When performing the second pass, address pairs are hammered only at the rate deemed as the maximum allowable by memory vendors (200K accesses per 64ms). Once this rate is exceeded, the integrity of memory contents may no longer be guaranteed. If errors are detected in both passes, errors are reported as normal.

The errors detected during Test 13, albeit exposed only in extreme memory access cases, are most certainly real errors. During typical home PC usage (eg. web browsing, word processing, etc.), it is less likely that the memory usage pattern will fall into the extreme case that make it vulnerable to disturbance errors. It may be of greater concern if you were running highly sensitive equipment such as medical equipment, aircraft control systems, or bank database servers. It is impossible to predict with any accuracy if these errors will occur in real life applications. One would need to do a major scientific study of 1000 of computers and their usage patterns, then do a forensic analysis of each application to study how it makes use of the RAM while it executes. To date, we have only seen 1-bit errors as a result of running the Hammer Test.

# Selected Readings on RowHammer (XI.B)

- PassMark Software, memtest86, since 2014
  - <https://www.memtest86.com/troubleshooting.htm#hammer>

## Detection and mitigation of row hammer errors

The ability of MemTest86 to detect and report on row hammer errors depends on several factors and what mitigations are in place. To generate errors adjacent memory rows must be repeatedly accessed. But hardware features such as multiple channels, interleaving, **scrambling**, Channel Hashing, NUMA & XOR schemes make it nearly impossible (for an arbitrary CPU & RAM stick) to know which memory addresses correspond to which rows in the RAM. Various mitigations might also be in place. Different BIOS firmware might set the refresh interval to different values (tREFI). The shorter the interval the more resistant the RAM will be to errors. But shorter intervals result in higher power consumption and increased processing overhead. Some CPUs also support pseudo target row refresh (pTRR) that can be used in combination with pTRR-compliant RAM. This field allows the RAM stick to indicate the MAC (Maximum Active Count) level which is the RAM can support. A typical value might be 200,000 row activations. Some CPUs also support the Joint Electron Design Engineering Council (JEDEC) Targeted Row Refresh (TRR) algorithm. The TRR is an improved version of the previously implemented pTRR algorithm and does not inflict any performance drop or additional power usage. As a result the row hammer test implemented in MemTest86 maybe not be the worst case possible and vulnerabilities in the underlying RAM might be undetectable due to the mitigations in place in the BIOS and CPU.



The screenshot displays the PassMark Software website. At the top left is the "PASSMARK SOFTWARE" logo. To the right is a shopping cart icon and a search bar. Below the logo is a navigation menu with links for Home, Software, Hardware, Benchmarks, Services, Store, Support, Forums, and About Us. The main content area features the MemTest86 logo, which includes a stylized blue head with circuit lines, and the text "MemTest86™" in orange and blue. Below the logo is the tagline "The original industry standard memory diagnostic utility". At the bottom of the page is a secondary navigation menu with links for Overview, Features, Technical Info, Screenshots, Download, Purchase, and Help.

# Security Implications (ISCA 2014)

- *Breach of memory protection*
  - OS page (4KB) fits inside DRAM row (8KB)
  - Adjacent DRAM row → Different OS page
- *Vulnerability: disturbance attack*
  - By accessing its own page, a program could corrupt pages belonging to another program
- *We constructed a proof-of-concept*
  - Using only user-level instructions

# More Security Implications (I)

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

www.iaik.tugraz.at

Not there yet, but ...



ROOT privileges for web apps!

29

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



GATED  
COMMUNITIES

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



# More Security Implications (II)

**"Can gain control of a smart phone deterministically"**



Drammer: Deterministic Rowhammer  
Attacks on Mobile Platforms, CCS'16 <sup>57</sup>

# More Security Implications (III)

- Using an integrated GPU in a mobile system to remotely escalate privilege via the WebGL interface



BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

"GRAND PWNING UNIT" —

## Drive-by Rowhammer attack uses GPU to compromise an Android phone

JavaScript based GLitch pwns browsers by flipping bits inside memory chips.

DAN GOODIN - 5/3/2018, 12:00 PM

## Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU

Pietro Frigo  
Vrije Universiteit  
Amsterdam  
p.frigo@vu.nl

Cristiano Giuffrida  
Vrije Universiteit  
Amsterdam  
giuffrida@cs.vu.nl

Herbert Bos  
Vrije Universiteit  
Amsterdam  
herbertb@cs.vu.nl

Kaveh Razavi  
Vrije Universiteit  
Amsterdam  
kaveh@cs.vu.nl

# More Security Implications (IV)

---

- Rowhammer over RDMA (I)



BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

THROWHAMMER —

## Packets over a LAN are all it takes to trigger serious Rowhammer bit flips

The bar for exploiting potentially serious DDR weakness keeps getting lower.

DAN GOODIN - 5/10/2018, 5:26 PM

### Throwhammer: Rowhammer Attacks over the Network and Defenses

Andrei Tatar  
*VU Amsterdam*

Radhesh Krishnan  
*VU Amsterdam*

Elias Athanasopoulos  
*University of Cyprus*

Cristiano Giuffrida  
*VU Amsterdam*

Herbert Bos  
*VU Amsterdam*

Kaveh Razavi  
*VU Amsterdam*

# More Security Implications (V)

- Rowhammer over RDMA (II)



Nethammer—Exploiting DRAM Rowhammer Bug Through Network Requests



## Nethammer: Inducing Rowhammer Faults through Network Requests

Moritz Lipp  
Graz University of Technology

Daniel Gruss  
Graz University of Technology

Misiker Tadesse Aga  
University of Michigan

Clémentine Maurice  
Univ Rennes, CNRS, IRISA

Michael Schwarz  
Graz University of Technology

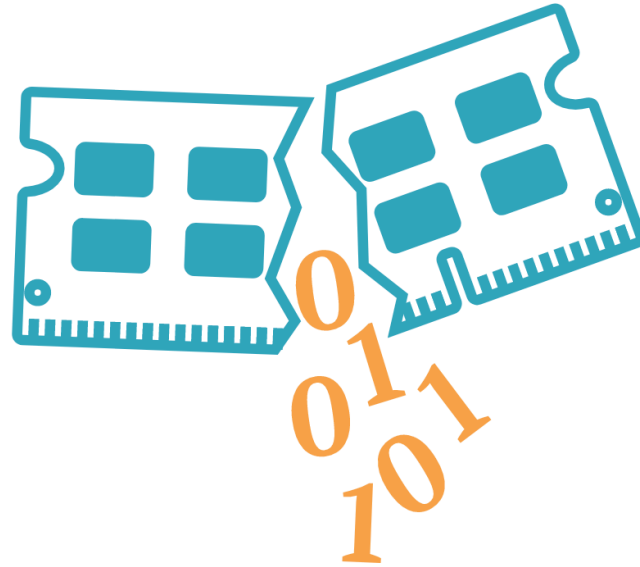
Lukas Raab  
Graz University of Technology

Lukas Lamster  
Graz University of Technology

# More Security Implications (VI)

---

- IEEE S&P 2020



RAMBleed

**RAMBleed: Reading Bits in Memory Without Accessing Them**

Andrew Kwong  
*University of Michigan*  
[ankwong@umich.edu](mailto:ankwong@umich.edu)

Daniel Genkin  
*University of Michigan*  
[genkin@umich.edu](mailto:genkin@umich.edu)

Daniel Gruss  
*Graz University of Technology*  
[daniel.gruss@iaik.tugraz.at](mailto:daniel.gruss@iaik.tugraz.at)

Yuval Yarom  
*University of Adelaide and Data61*  
[yval@cs.adelaide.edu.au](mailto:yval@cs.adelaide.edu.au)

# More Security Implications (VII)

---

- Rowhammer on MLC NAND Flash (based on [Cai+, HPCA 2017])



Security

## Rowhammer RAM attack adapted to hit flash storage

Project Zero's two-year-old dog learns a new trick

By [Richard Chirgwin](#) 17 Aug 2017 at 04:27

17 SHARE ▼

**From random block corruption to privilege escalation:  
A filesystem attack vector for rowhammer-like attacks**

Anil Kurmus

Nikolas Ioannou

Matthias Neugschwandtner

Nikolaos Papandreou

Thomas Parnell

*IBM Research – Zurich*

# More Security Implications?

---



# Understanding RowHammer



# Root Causes of Disturbance Errors

- *Cause 1: Electromagnetic coupling*
  - Toggling the wordline voltage briefly increases the voltage of adjacent wordlines
  - Slightly opens adjacent rows → Charge leakage
- *Cause 2: Conductive bridges*
- *Cause 3: Hot-carrier injection*

*Confirmed by at least one manufacturer*

# Experimental DRAM Testing Infrastructure



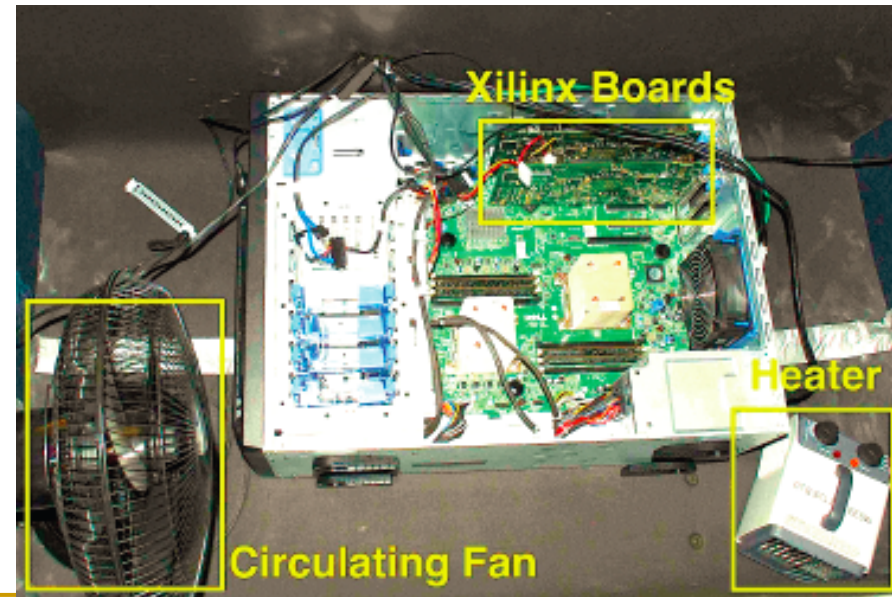
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

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

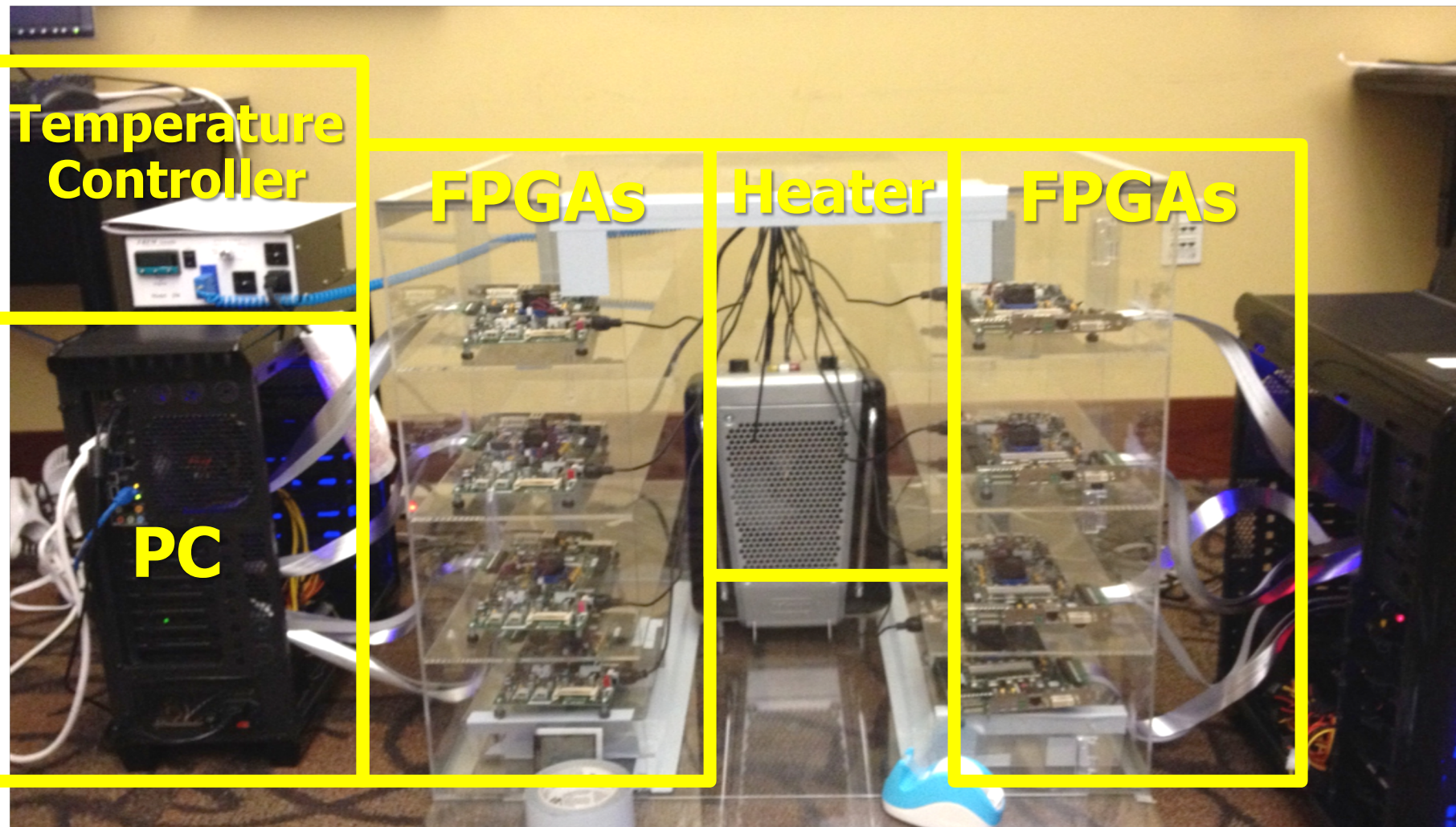
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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

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



# Experimental DRAM Testing Infrastructure



# Tested DRAM Modules (129 total)

Manufacturer	Module	Date* (yy-ww)	Timing†		Organization		Chip			Victims-per-Module			RI <sub>th</sub> (ms)
			Freq (MT/s)	t <sub>RC</sub> (ns)	Size (GB)	Chips	Size (Gb)‡	Pins	DieVersion§	Average	Minimum	Maximum	Min
A	A <sub>1</sub>	10-08	1066	50.625	0.5	4	1	×16	B	0	0	0	–
	A <sub>2</sub>	10-20	1066	50.625	1	8	1	×8	F	0	0	0	–
	A <sub>3-5</sub>	10-20	1066	50.625	0.5	4	1	×16	B	0	0	0	–
	A <sub>6-7</sub>	11-24	1066	49.125	1	4	2	×16	D	7.8 × 10 <sup>1</sup>	5.2 × 10 <sup>1</sup>	1.0 × 10 <sup>2</sup>	21.3
	A <sub>8-12</sub>	11-26	1066	49.125	1	4	2	×16	D	2.4 × 10 <sup>2</sup>	5.4 × 10 <sup>1</sup>	4.4 × 10 <sup>2</sup>	16.4
	A <sub>13-14</sub>	11-50	1066	49.125	1	4	2	×16	D	8.8 × 10 <sup>1</sup>	1.7 × 10 <sup>1</sup>	1.6 × 10 <sup>2</sup>	26.2
	A <sub>15-16</sub>	12-22	1600	50.625	1	4	2	×16	D	9.5	9	1.0 × 10 <sup>1</sup>	34.4
	A <sub>17-18</sub>	12-26	1600	49.125	2	8	2	×8	M	1.2 × 10 <sup>2</sup>	3.7 × 10 <sup>1</sup>	2.0 × 10 <sup>2</sup>	21.3
	A <sub>19-30</sub>	12-40	1600	48.125	2	8	2	×8	K	8.6 × 10 <sup>6</sup>	7.0 × 10 <sup>6</sup>	1.0 × 10 <sup>7</sup>	8.2
	A <sub>31-34</sub>	13-02	1600	48.125	2	8	2	×8	–	1.8 × 10 <sup>6</sup>	1.0 × 10 <sup>6</sup>	3.5 × 10 <sup>6</sup>	11.5
	A <sub>35-36</sub>	13-14	1600	48.125	2	8	2	×8	–	4.0 × 10 <sup>1</sup>	1.9 × 10 <sup>1</sup>	6.1 × 10 <sup>1</sup>	21.3
	A <sub>37-38</sub>	13-20	1600	48.125	2	8	2	×8	K	1.7 × 10 <sup>6</sup>	1.4 × 10 <sup>6</sup>	2.0 × 10 <sup>6</sup>	9.8
	A <sub>39-40</sub>	13-28	1600	48.125	2	8	2	×8	K	5.7 × 10 <sup>4</sup>	5.4 × 10 <sup>4</sup>	6.0 × 10 <sup>4</sup>	16.4
	A <sub>41</sub>	14-04	1600	49.125	2	8	2	×8	–	2.7 × 10 <sup>5</sup>	2.7 × 10 <sup>5</sup>	2.7 × 10 <sup>5</sup>	18.0
	A <sub>42-43</sub>	14-04	1600	48.125	2	8	2	×8	K	0.5	0	1	62.3
	B	B <sub>1</sub>	08-49	1066	50.625	1	8	1	×8	D	0	0	0
B <sub>2</sub>		09-49	1066	50.625	1	8	1	×8	E	0	0	0	–
B <sub>3</sub>		10-19	1066	50.625	1	8	1	×8	F	0	0	0	–
B <sub>4</sub>		10-31	1333	49.125	2	8	2	×8	C	0	0	0	–
B <sub>5</sub>		11-13	1333	49.125	2	8	2	×8	C	0	0	0	–
B <sub>6</sub>		11-16	1066	50.625	1	8	1	×8	F	0	0	0	–
B <sub>7</sub>		11-19	1066	50.625	1	8	1	×8	F	0	0	0	–
B <sub>8</sub>		11-25	1333	49.125	2	8	2	×8	C	0	0	0	–
B <sub>9</sub>		11-37	1333	49.125	2	8	2	×8	D	1.9 × 10 <sup>6</sup>	1.9 × 10 <sup>6</sup>	1.9 × 10 <sup>6</sup>	11.5
B <sub>10-12</sub>		11-46	1333	49.125	2	8	2	×8	D	2.2 × 10 <sup>6</sup>	1.5 × 10 <sup>6</sup>	2.7 × 10 <sup>6</sup>	11.5
B <sub>13</sub>		11-49	1333	49.125	2	8	2	×8	C	0	0	0	–
B <sub>14</sub>		12-01	1866	47.125	2	8	2	×8	D	9.1 × 10 <sup>5</sup>	9.1 × 10 <sup>5</sup>	9.1 × 10 <sup>5</sup>	9.8
B <sub>15-31</sub>		12-10	1866	47.125	2	8	2	×8	D	9.8 × 10 <sup>5</sup>	7.8 × 10 <sup>5</sup>	1.2 × 10 <sup>6</sup>	11.5
B <sub>32</sub>		12-25	1600	48.125	2	8	2	×8	E	7.4 × 10 <sup>5</sup>	7.4 × 10 <sup>5</sup>	7.4 × 10 <sup>5</sup>	11.5
B <sub>33-42</sub>		12-28	1600	48.125	2	8	2	×8	E	5.2 × 10 <sup>5</sup>	1.9 × 10 <sup>5</sup>	7.3 × 10 <sup>5</sup>	11.5
B <sub>43-47</sub>		12-31	1600	48.125	2	8	2	×8	E	4.0 × 10 <sup>5</sup>	2.9 × 10 <sup>5</sup>	5.5 × 10 <sup>5</sup>	13.1
B <sub>48-51</sub>	13-19	1600	48.125	2	8	2	×8	E	1.1 × 10 <sup>5</sup>	7.4 × 10 <sup>4</sup>	1.4 × 10 <sup>5</sup>	14.7	
B <sub>52-53</sub>	13-40	1333	49.125	2	8	2	×8	D	2.6 × 10 <sup>4</sup>	2.3 × 10 <sup>4</sup>	2.9 × 10 <sup>4</sup>	21.3	
B <sub>54</sub>	14-07	1333	49.125	2	8	2	×8	D	7.5 × 10 <sup>3</sup>	7.5 × 10 <sup>3</sup>	7.5 × 10 <sup>3</sup>	26.2	
C	C <sub>1</sub>	10-18	1333	49.125	2	8	2	×8	A	0	0	0	–
	C <sub>2</sub>	10-20	1066	50.625	2	8	2	×8	A	0	0	0	–
	C <sub>3</sub>	10-22	1066	50.625	2	8	2	×8	A	0	0	0	–
	C <sub>4-5</sub>	10-26	1333	49.125	2	8	2	×8	B	8.9 × 10 <sup>2</sup>	6.0 × 10 <sup>2</sup>	1.2 × 10 <sup>3</sup>	29.5
	C <sub>6</sub>	10-43	1333	49.125	1	8	1	×8	T	0	0	0	–
	C <sub>7</sub>	10-51	1333	49.125	2	8	2	×8	B	4.0 × 10 <sup>2</sup>	4.0 × 10 <sup>2</sup>	4.0 × 10 <sup>2</sup>	29.5
	C <sub>8</sub>	11-12	1333	46.25	2	8	2	×8	B	6.9 × 10 <sup>2</sup>	6.9 × 10 <sup>2</sup>	6.9 × 10 <sup>2</sup>	21.3
	C <sub>9</sub>	11-19	1333	46.25	2	8	2	×8	B	9.2 × 10 <sup>2</sup>	9.2 × 10 <sup>2</sup>	9.2 × 10 <sup>2</sup>	27.9
	C <sub>10</sub>	11-31	1333	49.125	2	8	2	×8	B	3	3	3	39.3
	C <sub>11</sub>	11-42	1333	49.125	2	8	2	×8	B	1.6 × 10 <sup>2</sup>	1.6 × 10 <sup>2</sup>	1.6 × 10 <sup>2</sup>	39.3
	C <sub>12</sub>	11-48	1600	48.125	2	8	2	×8	C	7.1 × 10 <sup>4</sup>	7.1 × 10 <sup>4</sup>	7.1 × 10 <sup>4</sup>	19.7
	C <sub>13</sub>	12-08	1333	49.125	2	8	2	×8	C	3.9 × 10 <sup>4</sup>	3.9 × 10 <sup>4</sup>	3.9 × 10 <sup>4</sup>	21.3
	C <sub>14-15</sub>	12-12	1333	49.125	2	8	2	×8	C	3.7 × 10 <sup>4</sup>	2.1 × 10 <sup>4</sup>	5.4 × 10 <sup>4</sup>	21.3
	C <sub>16-18</sub>	12-20	1600	48.125	2	8	2	×8	C	3.5 × 10 <sup>3</sup>	1.2 × 10 <sup>3</sup>	7.0 × 10 <sup>3</sup>	27.9
	C <sub>19</sub>	12-23	1600	48.125	2	8	2	×8	E	1.4 × 10 <sup>5</sup>	1.4 × 10 <sup>5</sup>	1.4 × 10 <sup>5</sup>	18.0
	C <sub>20</sub>	12-24	1600	48.125	2	8	2	×8	C	6.5 × 10 <sup>4</sup>	6.5 × 10 <sup>4</sup>	6.5 × 10 <sup>4</sup>	21.3
	C <sub>21</sub>	12-26	1600	48.125	2	8	2	×8	C	2.3 × 10 <sup>4</sup>	2.3 × 10 <sup>4</sup>	2.3 × 10 <sup>4</sup>	24.6
C <sub>22</sub>	12-32	1600	48.125	2	8	2	×8	C	1.7 × 10 <sup>4</sup>	1.7 × 10 <sup>4</sup>	1.7 × 10 <sup>4</sup>	22.9	
C <sub>23-24</sub>	12-37	1600	48.125	2	8	2	×8	C	2.3 × 10 <sup>4</sup>	1.1 × 10 <sup>4</sup>	3.4 × 10 <sup>4</sup>	18.0	
C <sub>25-30</sub>	12-41	1600	48.125	2	8	2	×8	C	2.0 × 10 <sup>4</sup>	1.1 × 10 <sup>4</sup>	3.2 × 10 <sup>4</sup>	19.7	
C <sub>31</sub>	13-11	1600	48.125	2	8	2	×8	C	3.3 × 10 <sup>5</sup>	3.3 × 10 <sup>5</sup>	3.3 × 10 <sup>5</sup>	14.7	
C <sub>32</sub>	13-35	1600	48.125	2	8	2	×8	C	3.7 × 10 <sup>4</sup>	3.7 × 10 <sup>4</sup>	3.7 × 10 <sup>4</sup>	21.3	

\* We report the manufacture date marked on the chip packages, which is more accurate than other dates that can be gleaned from a module.

† We report timing constraints stored in the module's on-board ROM [33], which is read by the system BIOS to calibrate the memory controller.

‡ The maximum DRAM chip size supported by our testing platform is 2Gb.

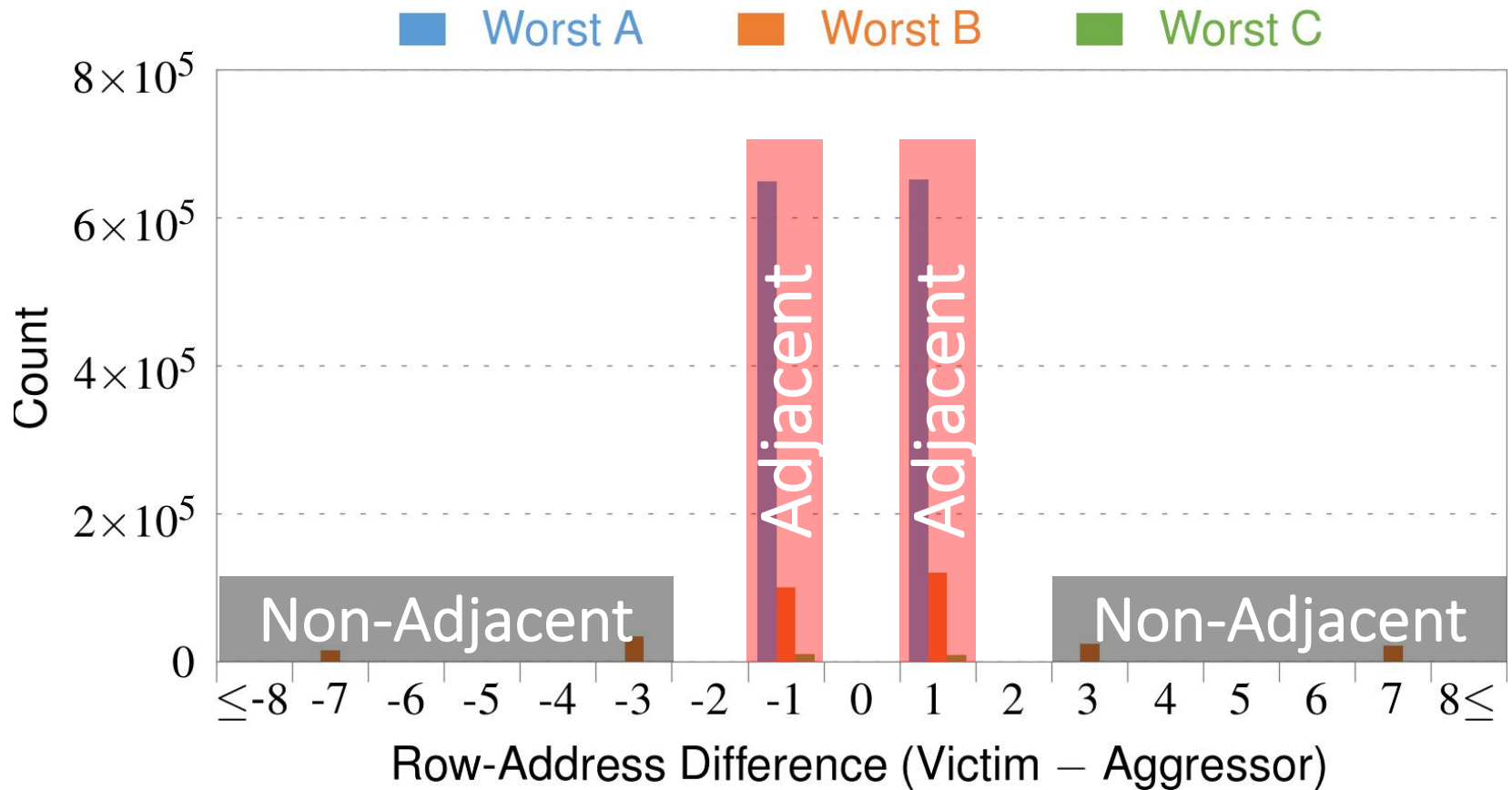
§ We report DRAM die versions marked on the chip packages, which typically progress in the following manner:  $\mathcal{M} \rightarrow \mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C} \rightarrow \dots$ .

Table 3. Sample population of 129 DDR3 DRAM modules, categorized by manufacturer and sorted by manufacture date

# RowHammer Characterization Results

1. Most Modules Are at Risk
2. Errors vs. Vintage
3. Error = Charge Loss
4. Adjacency: Aggressor & Victim
5. Sensitivity Studies
6. Other Results in Paper
7. Solution Space

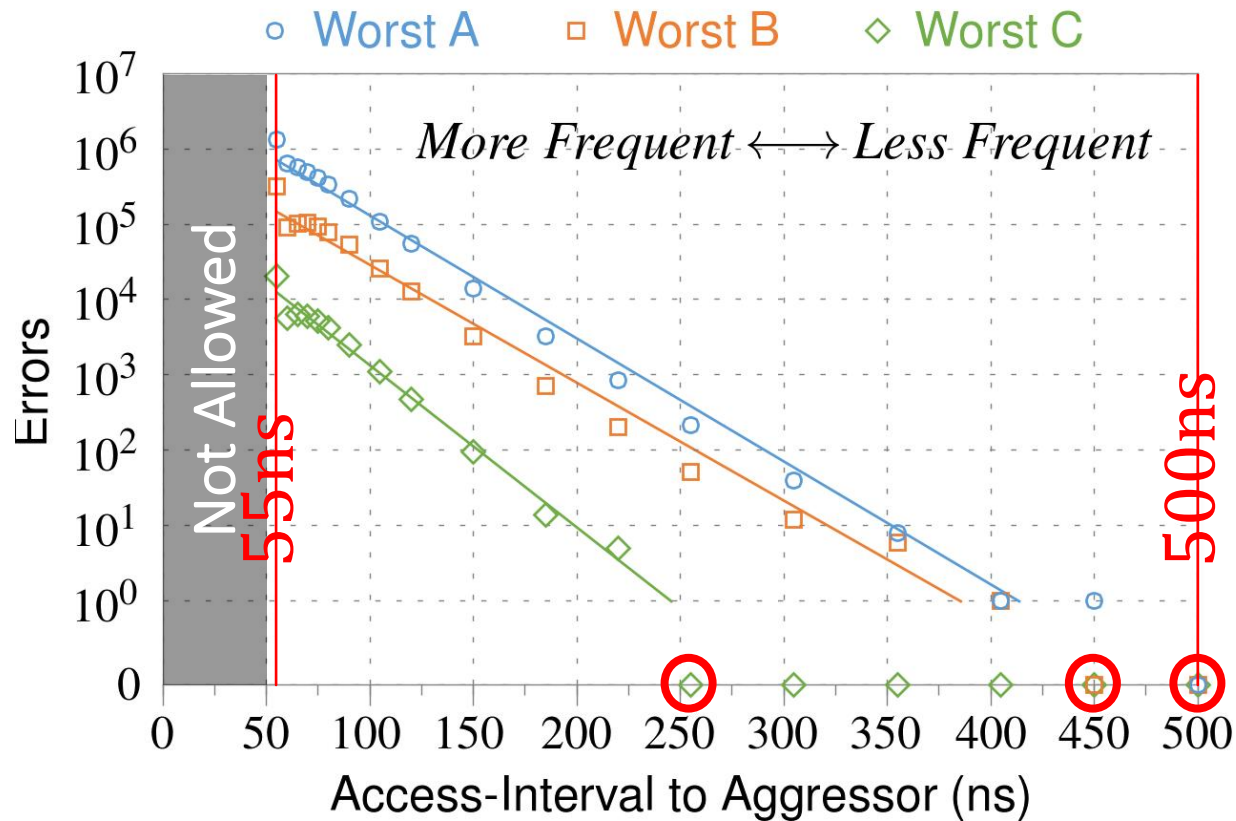
# 4. Adjacency: Aggressor & Victim



Note: For three modules with the most errors (only first bank)

*Most aggressors & victims are adjacent*

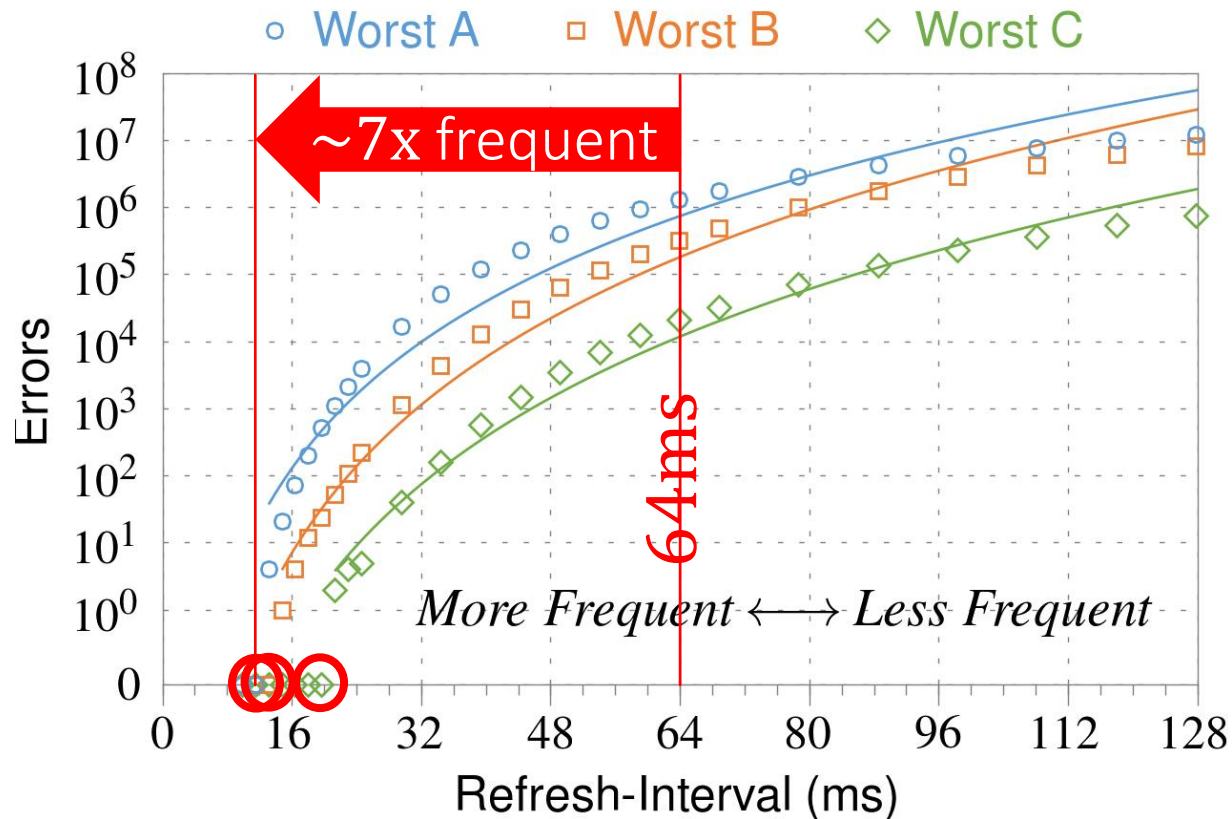
# 1 Access Interval (Aggressor)



Note: For three modules with the most errors (only first bank)

*Less frequent accesses  $\rightarrow$  Fewer errors*

## 2 Refresh Interval

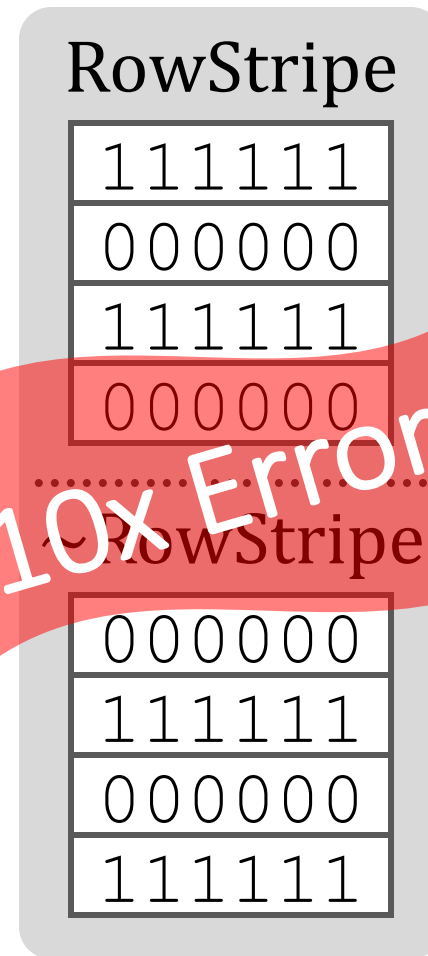
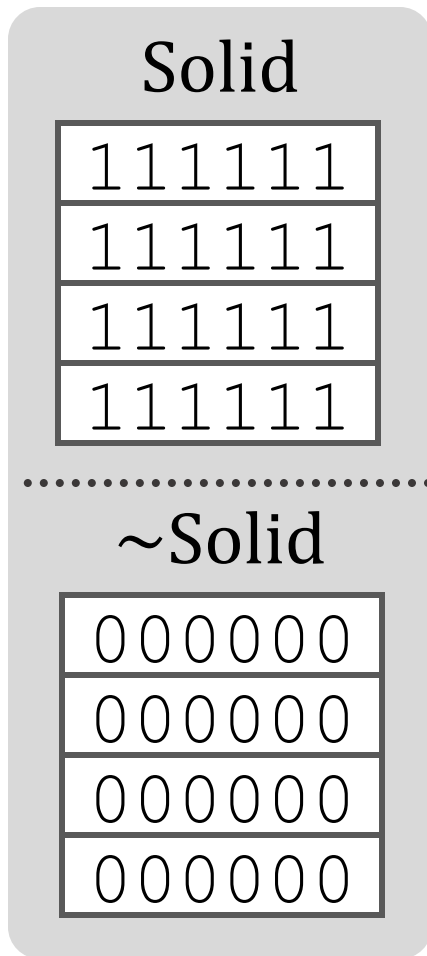


Note: Using three modules with the most errors (only first bank)

*More frequent refreshes → Fewer errors*



### 3 Data Pattern



10x Errors

*Errors affected by data stored in other cells*

# 6. Other Results (in Paper)

- *Victim Cells  $\neq$  Weak Cells (i.e., leaky cells)*
  - Almost no overlap between them
- *Errors not strongly affected by temperature*
  - Default temperature: 50°C
  - At 30°C and 70°C, number of errors changes <15%
- *Errors are repeatable*
  - Across ten iterations of testing, >70% of victim cells had errors in every iteration

# 6. Other Results (in Paper) cont'd

- *As many as 4 errors per cache-line*
  - Simple ECC (e.g., SECDED) cannot prevent all errors
- *Number of cells & rows affected by aggressor*
  - Victims cells per aggressor:  $\leq 110$
  - Victims rows per aggressor:  $\leq 9$
- *Cells affected by two aggressors on either side*
  - Very small fraction of victim cells ( $< 100$ ) have an error when either one of the aggressors is toggled

# More on RowHammer Analysis

---

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

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

Yoongu 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

# Retrospective on RowHammer & Future

---

- Onur Mutlu,  
**"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"**  
*Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (DATE), Lausanne, Switzerland, March 2017.*  
[[Slides \(pptx\)](#) ([pdf](#))]

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

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

# A More Recent RowHammer Retrospective

---

- Onur Mutlu and Jeremie Kim,  
**"RowHammer: A Retrospective"**  
*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.*  
[\[Preliminary arXiv version\]](#)

## RowHammer: A Retrospective

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

# RowHammer Solutions

# Two Types of RowHammer Solutions

---

## ■ Immediate

- ❑ To protect the vulnerable DRAM chips in the field
- ❑ Limited possibilities

## ■ Longer-term

- ❑ To protect future DRAM chips
- ❑ Wider range of protection mechanisms

## ■ Our ISCA 2014 paper proposes both types of solutions

- ❑ Seven solutions in total
- ❑ PARA proposed as best solution → already employed in the field



# Some Potential Solutions

---

- Make better DRAM chips

Cost

- Refresh frequently

Power, Performance

- Sophisticated ECC

Cost, Power

- Access counters

Cost, Power, Complexity

# Naive Solutions

## 1 *Throttle accesses to same row*

- Limit access-interval:  $\geq 500\text{ns}$
- Limit number of accesses:  $\leq 128\text{K}$  (=64ms/500ns)

## 2 *Refresh more frequently*

- Shorten refresh-interval by  $\sim 7\text{x}$

*Both naive solutions introduce significant overhead in performance and power*

# Apple's Patch for RowHammer

---

- <https://support.apple.com/en-gb/HT204934>

Available for: OS X Mountain Lion v10.8.5, OS X Mavericks v10.9.5

Impact: A malicious application may induce memory corruption to escalate privileges

Description: A disturbance error, also known as Rowhammer, exists with some DDR3 RAM that could have led to memory corruption. This issue was mitigated by increasing memory refresh rates.

CVE-ID

CVE-2015-3693 : Mark Seaborn and Thomas Dullien of Google, working from original research by Yoongu Kim et al (2014)

HP, Lenovo, and other vendors released similar patches

---

# Our Solution to RowHammer

- PARA: *Probabilistic Adjacent Row Activation*
- Key Idea
  - After closing a row, we activate (i.e., refresh) one of its neighbors with a low probability:  $p = 0.005$
- Reliability Guarantee
  - When  $p=0.005$ , errors in one year:  $9.4 \times 10^{-14}$
  - By adjusting the value of  $p$ , we can vary the strength of protection against errors

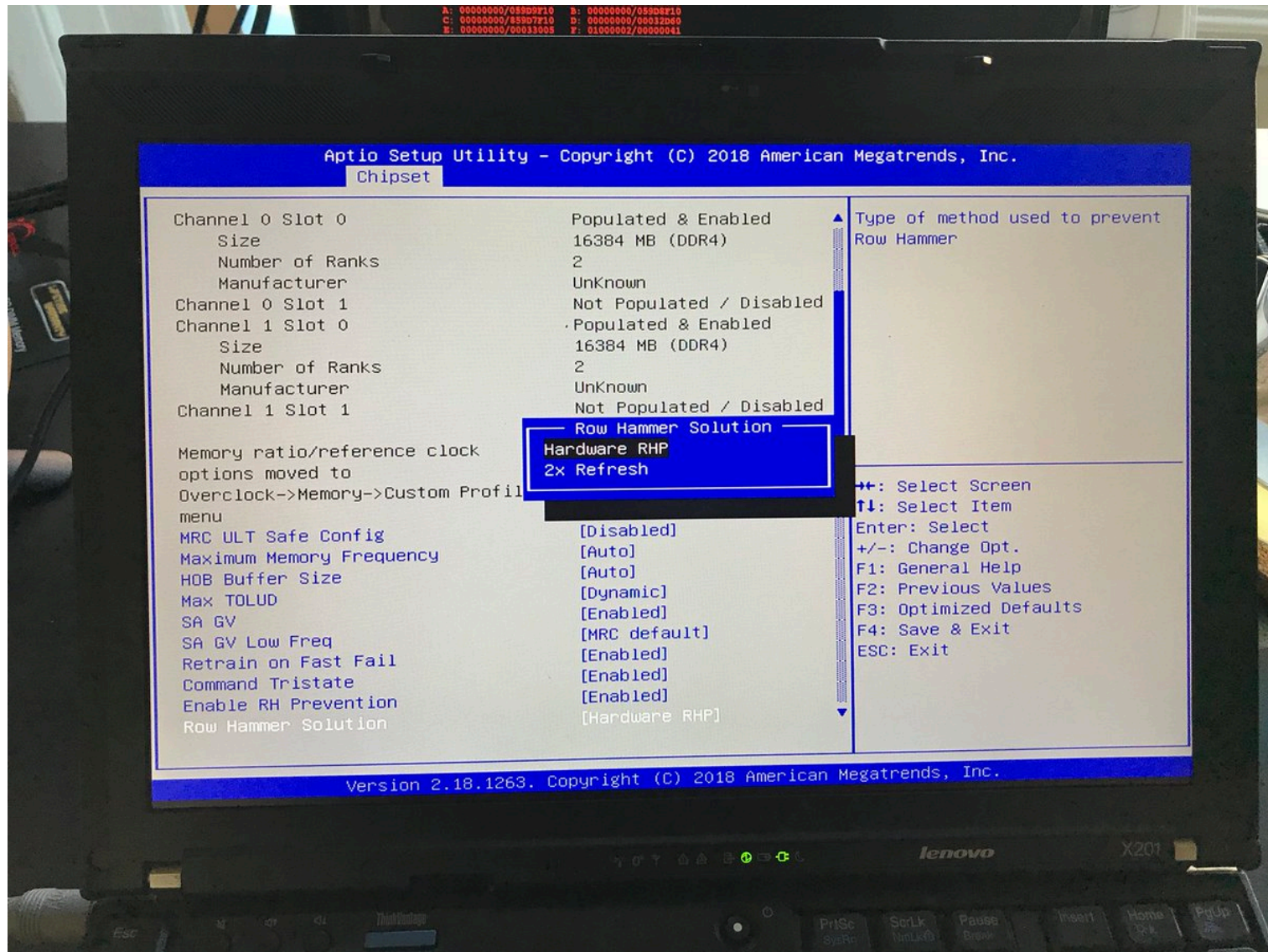
# Advantages of PARA

- *PARA refreshes rows infrequently*
  - Low power
  - Low performance-overhead
    - Average slowdown: **0.20%** (for 29 benchmarks)
    - Maximum slowdown: **0.75%**
- *PARA is stateless*
  - Low cost
  - Low complexity
- *PARA is an effective and low-overhead solution to prevent disturbance errors*

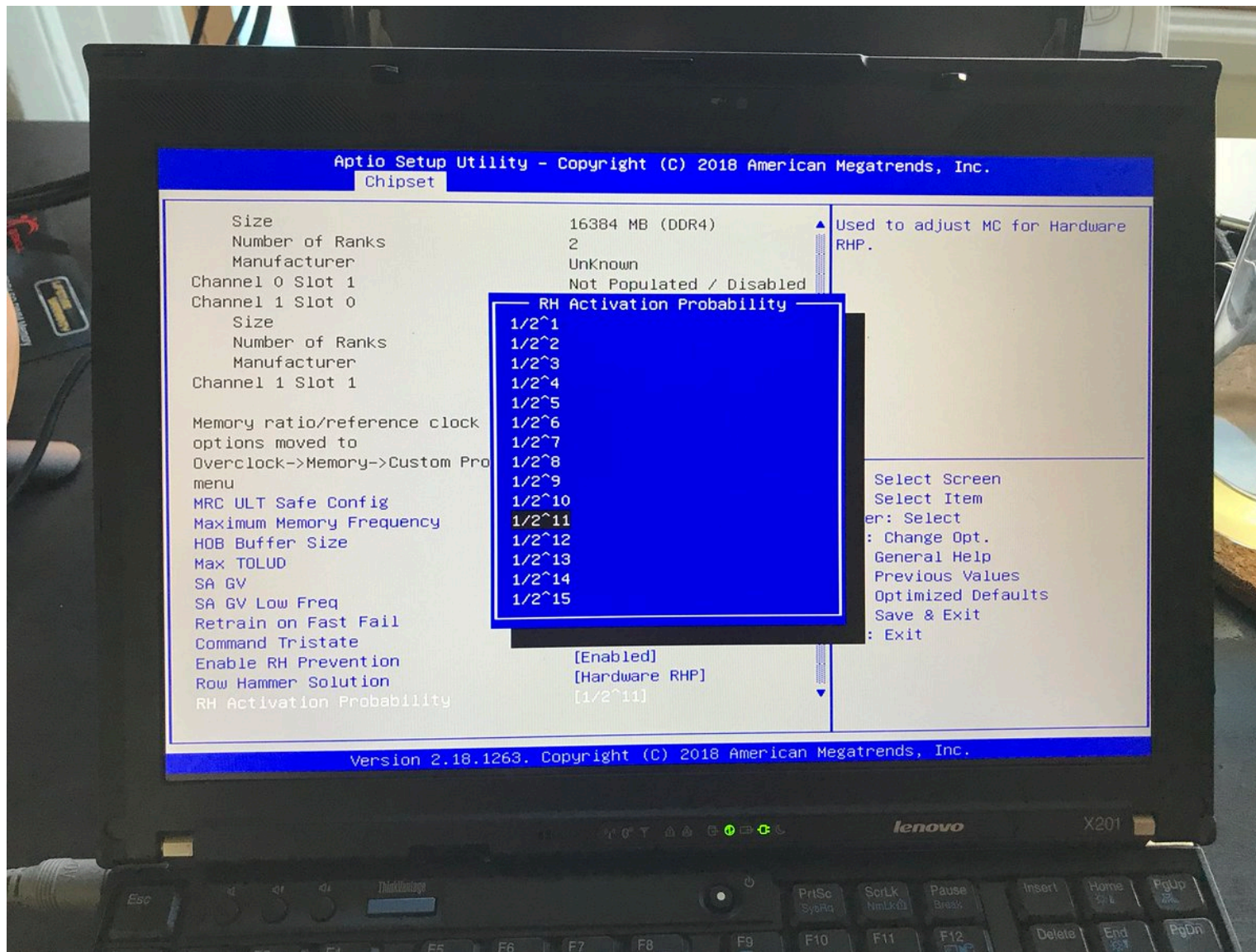
# Requirements for PARA

- If implemented in **DRAM chip** (done today)
  - Enough slack in timing parameters
  - Plenty of slack today:
    - Lee et al., “**Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common Case**,” HPCA 2015.
    - Chang et al., “**Understanding Latency Variation in Modern DRAM Chips**,” SIGMETRICS 2016.
    - Lee et al., “**Design-Induced Latency Variation in Modern DRAM Chips**,” SIGMETRICS 2017.
    - Chang et al., “**Understanding Reduced-Voltage Operation in Modern DRAM Devices**,” SIGMETRICS 2017.
    - Ghose et al., “**What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study**,” SIGMETRICS 2018.
- If implemented in **memory controller**
  - Better coordination between memory controller and DRAM
  - Memory controller should know which rows are physically adjacent

# Probabilistic Activation in Real Life (I)



# Probabilistic Activation in Real Life (II)





# Seven RowHammer Solutions Proposed

---

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

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

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

<sup>1</sup>Carnegie Mellon University

<sup>2</sup>Intel Labs

# Main Memory Needs Intelligent Controllers for Security

# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

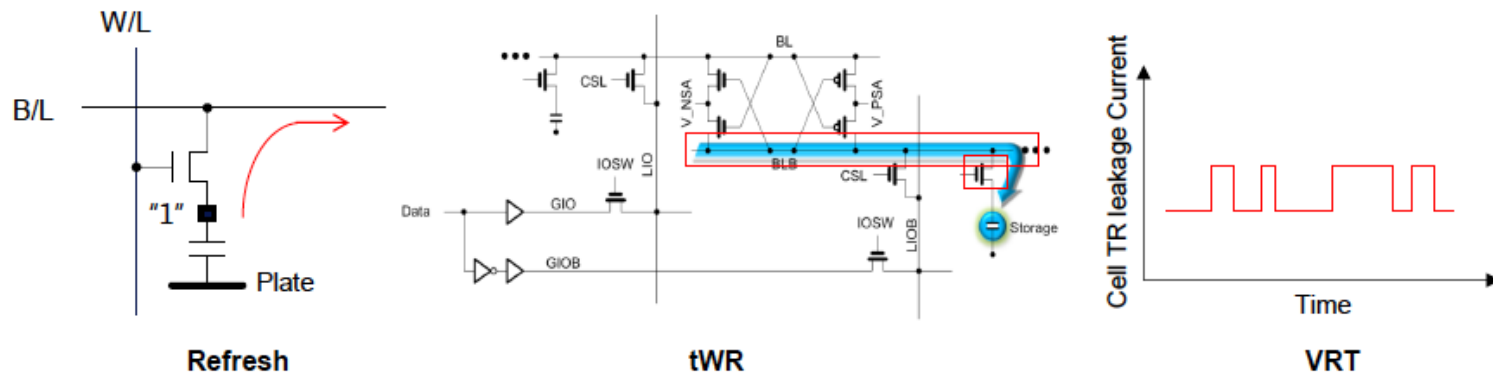
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

### ❖ $t_{WR}$

- 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

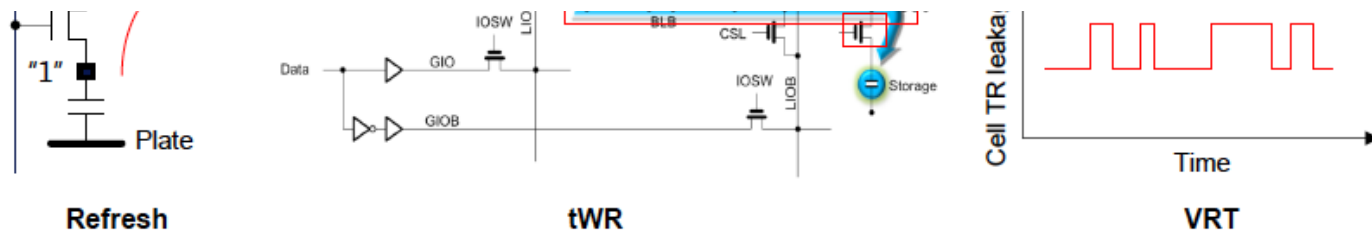
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

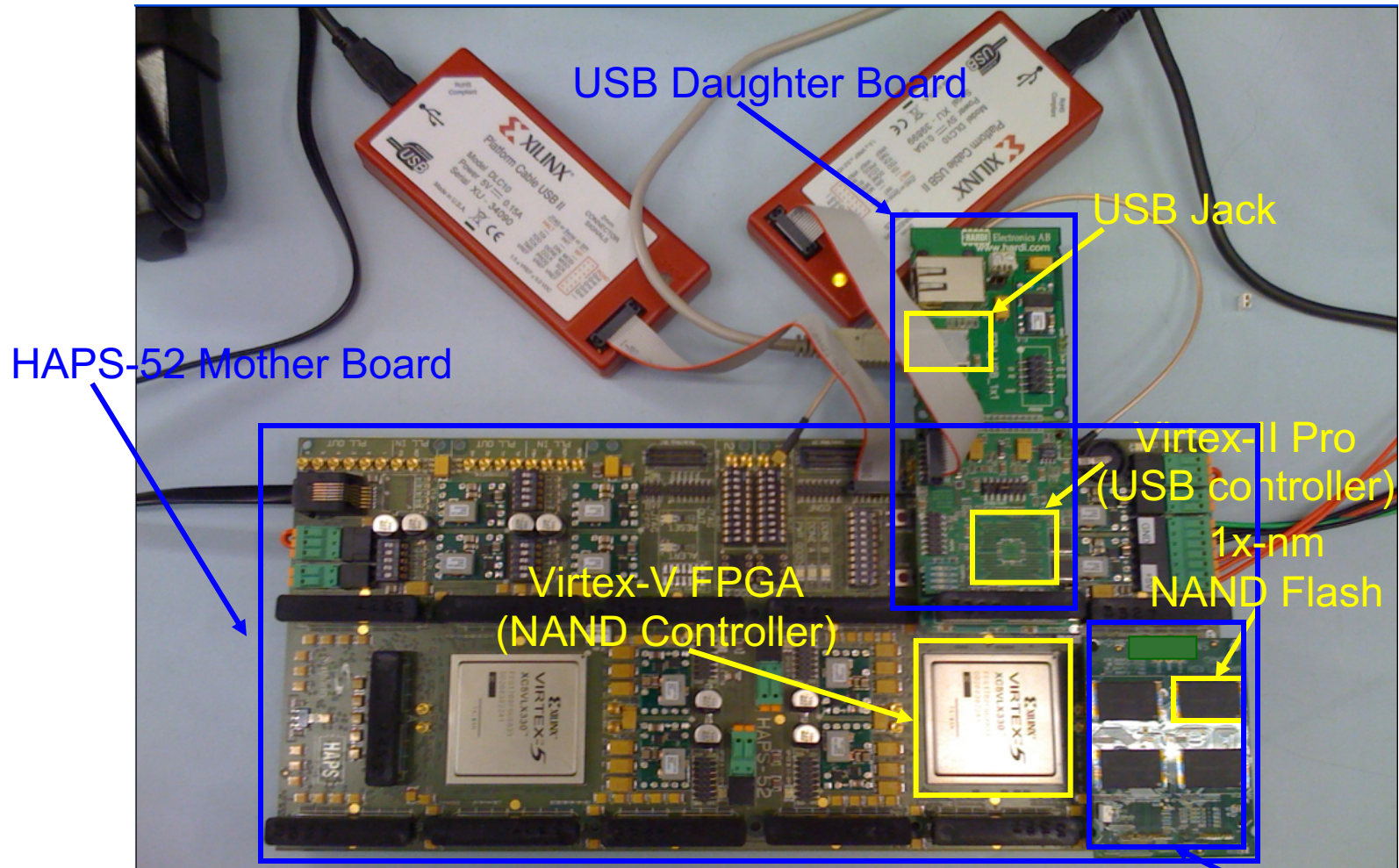
# Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

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*



# Aside: Intelligent Controller for NAND Flash



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

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



*Proceedings of the IEEE, Sept. 2017*



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

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

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

<https://arxiv.org/pdf/1706.08642>

**Main Memory Needs  
Intelligent Controllers**

# Future Memory Reliability/Security Challenges



# Future of Main Memory

---

- DRAM is becoming less reliable → more vulnerable

# Large-Scale Failure Analysis of DRAM Chips

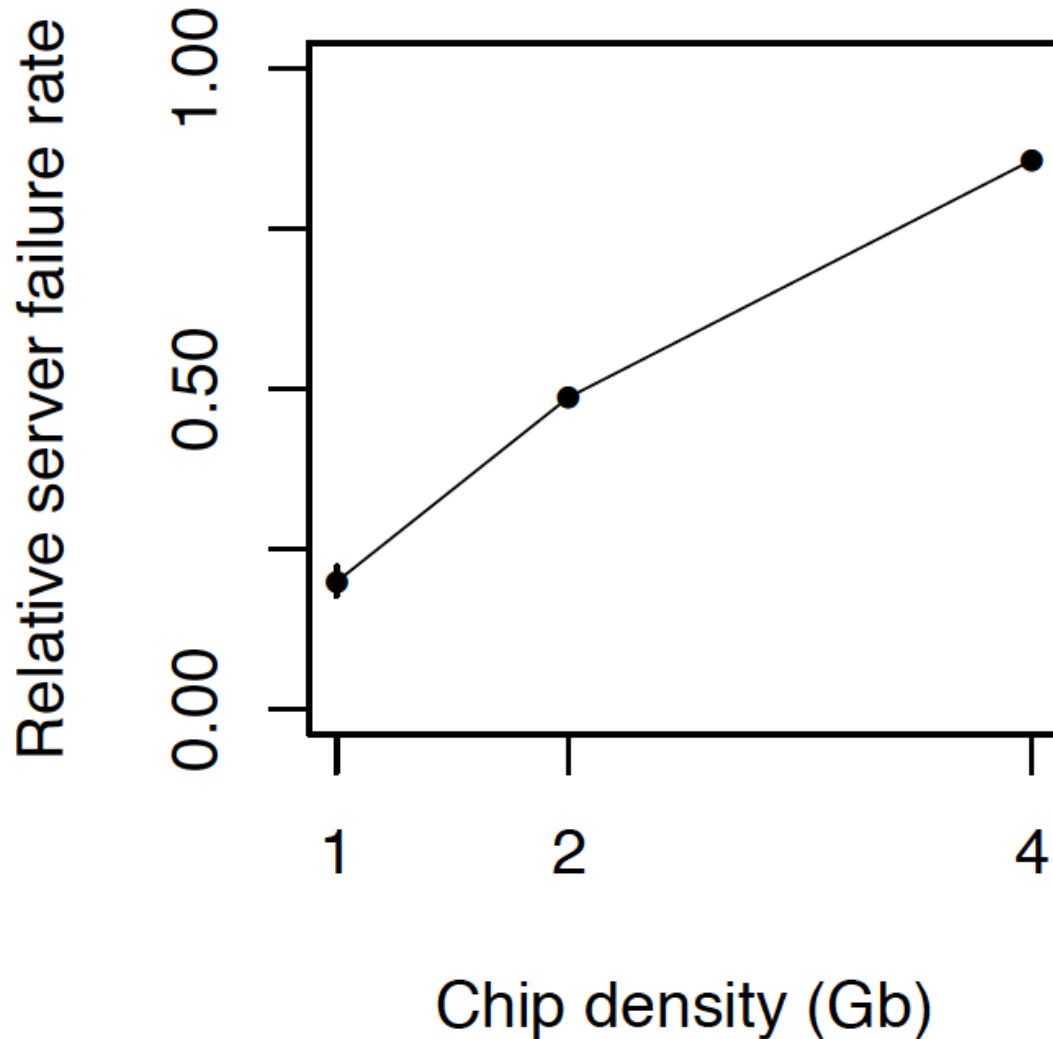
---

- 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.

# DRAM Reliability Reducing



*Intuition:  
quadratic  
increase in  
capacity*

# Aside: SSD Error Analysis in the Field

---

- First large-scale field study of flash memory errors
- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu, **"A Large-Scale Study of Flash Memory Errors in the Field"** *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Portland, OR, June 2015.  
[[Slides \(pptx\)](#)] [[pdf](#)] [[Coverage at ZDNet](#)]

## A Large-Scale Study of Flash Memory Failures in the Field

Justin Meza  
Carnegie Mellon University  
meza@cmu.edu

Qiang Wu  
Facebook, Inc.  
qwu@fb.com

Sanjeev Kumar  
Facebook, Inc.  
skumar@fb.com

Onur Mutlu  
Carnegie Mellon University  
onur@cmu.edu

# Future of Main Memory

---

- DRAM is becoming less reliable → more vulnerable
- Due to difficulties in DRAM scaling, other problems may also appear (or they may be going unnoticed)
- Some errors may already be slipping into the field
  - Read disturb errors (Rowhammer)
  - Retention errors
  - Read errors, write errors
  - ...
- These errors can also pose security vulnerabilities

# DRAM Data Retention Time Failures

---

- Determining the data retention time of a cell/row is getting more difficult
- Retention failures may already be slipping into the field

# Analysis of Data Retention Failures [ISCA'13]

---

- Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson, and Onur Mutlu, **"An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms"**  
*Proceedings of the 40th International Symposium on Computer Architecture (ISCA)*, Tel-Aviv, Israel, June 2013. [Slides \(ppt\)](#) [Slides \(pdf\)](#)

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

Jamie Liu\*

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

Ben Jaiyen\*

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

Yoongu Kim

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

Chris Wilkerson

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

Onur Mutlu

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

# Mitigation of Retention Issues [SIGMETRICS'14]

---

- Samira Khan, Donghyuk Lee, Yoongu Kim, Alaa Alameldeen, Chris Wilkerson, and Onur Mutlu,  
**"The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study"**  
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Austin, TX, June 2014. [[Slides \(pptx\)](#)] [[pdf](#)] [[Poster \(pptx\)](#)] [[pdf](#)] [[Full data sets](#)]

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

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

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

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

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

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

Onur Mutlu<sup>†</sup>  
onur@cmu.edu

<sup>†</sup>Carnegie Mellon University

<sup>\*</sup>Intel Labs



# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

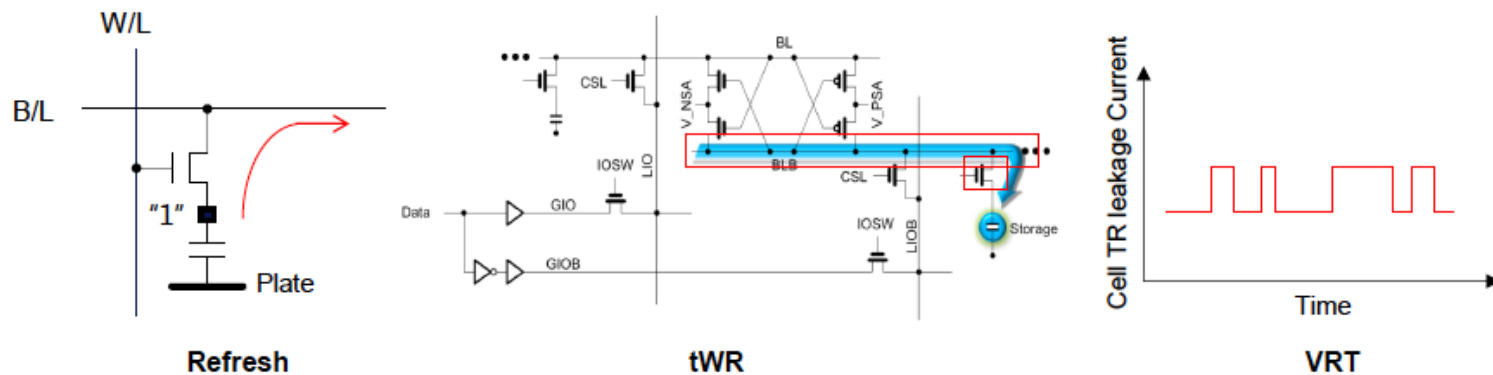
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

### ❖ 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



# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

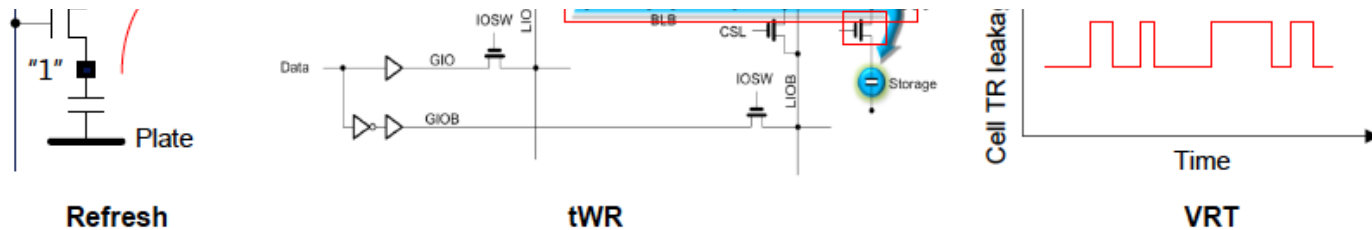
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

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*



# Mitigation of Retention Issues [DSN'15]

---

- Moinuddin Qureshi, Dae Hyun Kim, Samira Khan, Prashant Nair, and Onur Mutlu,  
**"AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems"**  
*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Rio de Janeiro, Brazil, June 2015.  
[[Slides \(pptx\)](#)] [[pdf](#)]

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

Moinuddin K. Qureshi<sup>†</sup>      Dae-Hyun Kim<sup>†</sup>      Samira Khan<sup>‡</sup>      Prashant J. Nair<sup>†</sup>      Onur Mutlu<sup>‡</sup>  
<sup>†</sup>Georgia Institute of Technology      <sup>‡</sup>Carnegie Mellon University  
{*moin, dhkim, pnair6*}@ece.gatech.edu      {*samirakhan, onur*}@cmu.edu

# Mitigation of Retention Issues [DSN'16]

---

- Samira Khan, Donghyuk Lee, and Onur Mutlu,  
**"PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM"**  
*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Toulouse, France, June 2016.  
[\[Slides \(pptx\)\]](#) [\[pdf\]](#)

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

Samira Khan<sup>\*</sup>

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

Donghyuk Lee<sup>†‡</sup>

<sup>†</sup>Carnegie Mellon University

Onur Mutlu<sup>\*†</sup>

<sup>‡</sup>Nvidia

<sup>\*</sup>ETH Zürich

# Mitigation of Retention Issues [MICRO'17]

---

- Samira Khan, Chris Wilkerson, Zhe Wang, Alaa R. Alameldeen, Donghyuk Lee, and Onur Mutlu,  
**"Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content"**  
*Proceedings of the 50th International Symposium on Microarchitecture (MICRO)*, Boston, MA, USA, October 2017.  
[\[Slides \(pptx\) \(pdf\)\]](#) [\[Lightning Session Slides \(pptx\) \(pdf\)\]](#) [\[Poster \(pptx\) \(pdf\)\]](#)

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

Samira Khan<sup>\*</sup> Chris Wilkerson<sup>†</sup> Zhe Wang<sup>†</sup> Alaa R. Alameldeen<sup>†</sup> Donghyuk Lee<sup>‡</sup> Onur Mutlu<sup>\*</sup>  
<sup>\*</sup>University of Virginia    <sup>†</sup>Intel Labs    <sup>‡</sup>Nvidia Research    <sup>\*</sup>ETH Zürich

# Mitigation of Retention Issues [ISCA'17]

---

- Minesh Patel, Jeremie S. Kim, and Onur Mutlu,  
**"The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions"**  
*Proceedings of the 44th International Symposium on Computer Architecture (ISCA)*, Toronto, Canada, June 2017.  
[[Slides \(pptx\)](#)] [[pdf](#)]  
[[Lightning Session Slides \(pptx\)](#)] [[pdf](#)]
- First experimental analysis of (mobile) LPDDR4 chips
- Analyzes the complex tradeoff space of retention time profiling
- Idea: enable fast and robust profiling at higher refresh intervals & temperatures

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

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

# Mitigation of Retention Issues [DSN'19]

---

- Minesh Patel, Jeremie S. Kim, Hasan Hassan, and Onur Mutlu, **"Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices"**  
*Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Portland, OR, USA, June 2019.  
[[Slides \(pptx\)](#)] [[pdf](#)]  
[[Talk Video](#) (26 minutes)]  
[[Full Talk Lecture](#) (29 minutes)]  
[[Source Code for EINSim, the Error Inference Simulator](#)]  
***Best paper award.***

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

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

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

**Main Memory Needs**  
**Intelligent Controllers**



# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

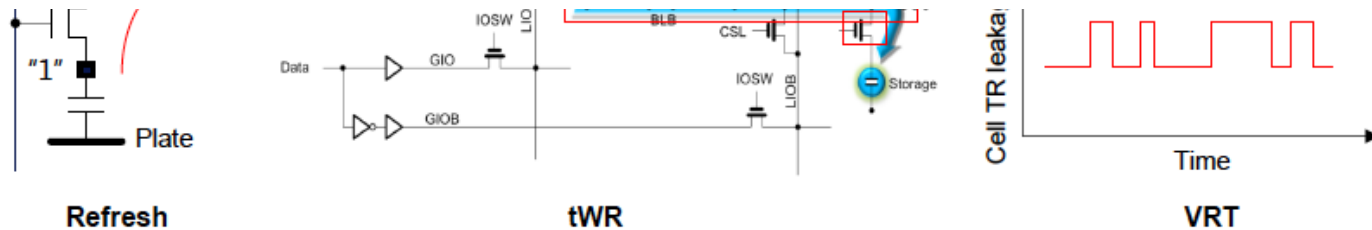
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

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*



# 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/>

# Keeping Future Memory Secure

# How Do We Keep Memory Secure?

---

- DRAM
- Flash memory
- Emerging Technologies
  - Phase Change Memory
  - STT-MRAM
  - RRAM, memristors
  - ...

## Fundamentally Secure, Reliable, Safe Computing Architectures

# Solution Direction: Principled Designs

---

**Design fundamentally secure  
computing architectures**

**Predict and prevent  
such safety issues**

# Architecting for Security

---

- **Understand:** Methods for vulnerability modeling & discovery
  - Modeling and prediction based on real (device) data and analysis
  - Understanding vulnerabilities
  - Developing reliable metrics
- **Architect:** Principled architectures with security as key concern
  - Good partitioning of duties across the stack
  - Cannot give up performance and efficiency
  - Patch-ability in the field
- **Design & Test:** Principled design, automation, (online) testing
  - Design for security
  - High coverage and good interaction with system reliability methods

# Understand and Model with Experiments (DRAM)

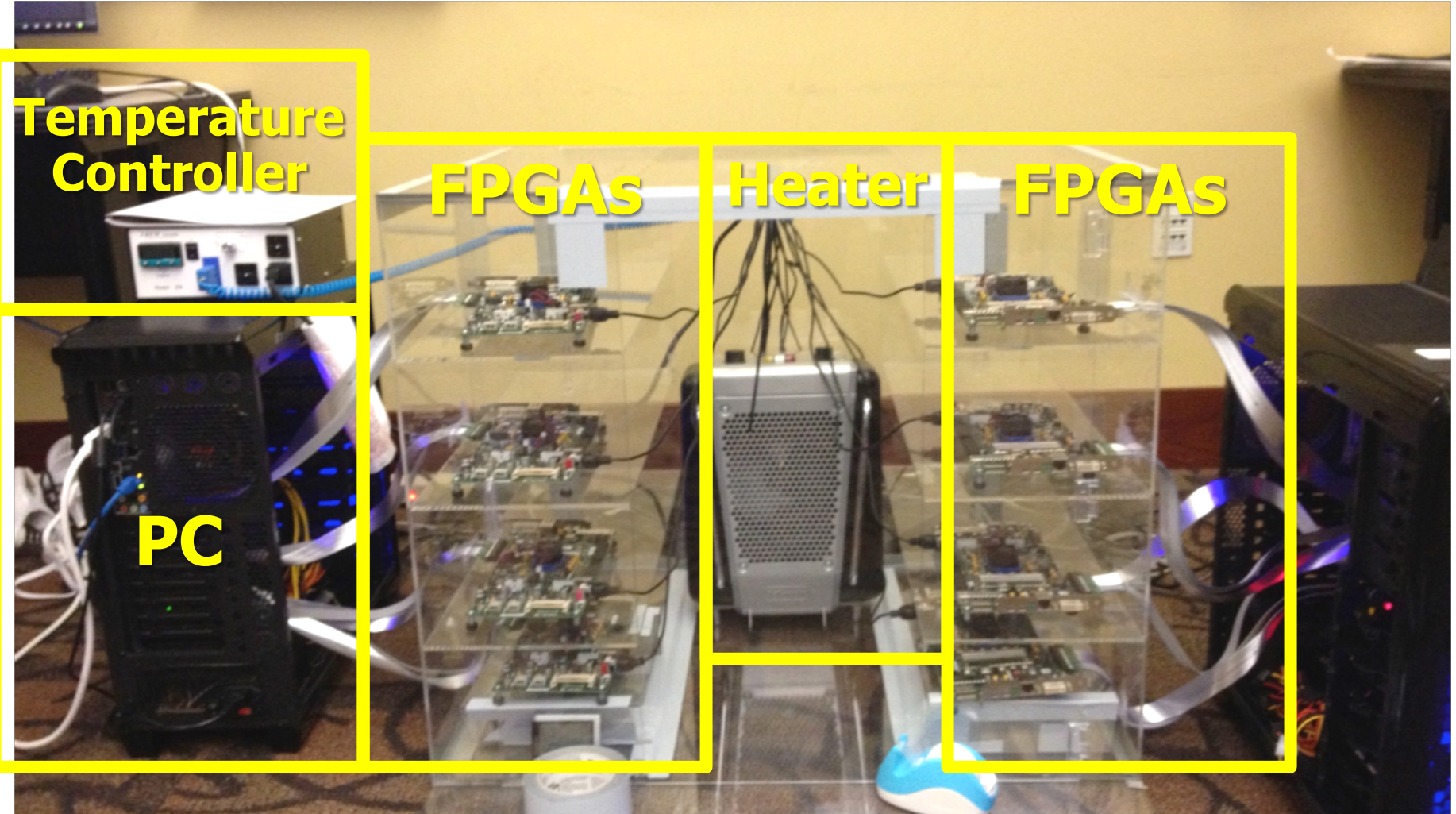
Temperature  
Controller

FPGAs

Heater

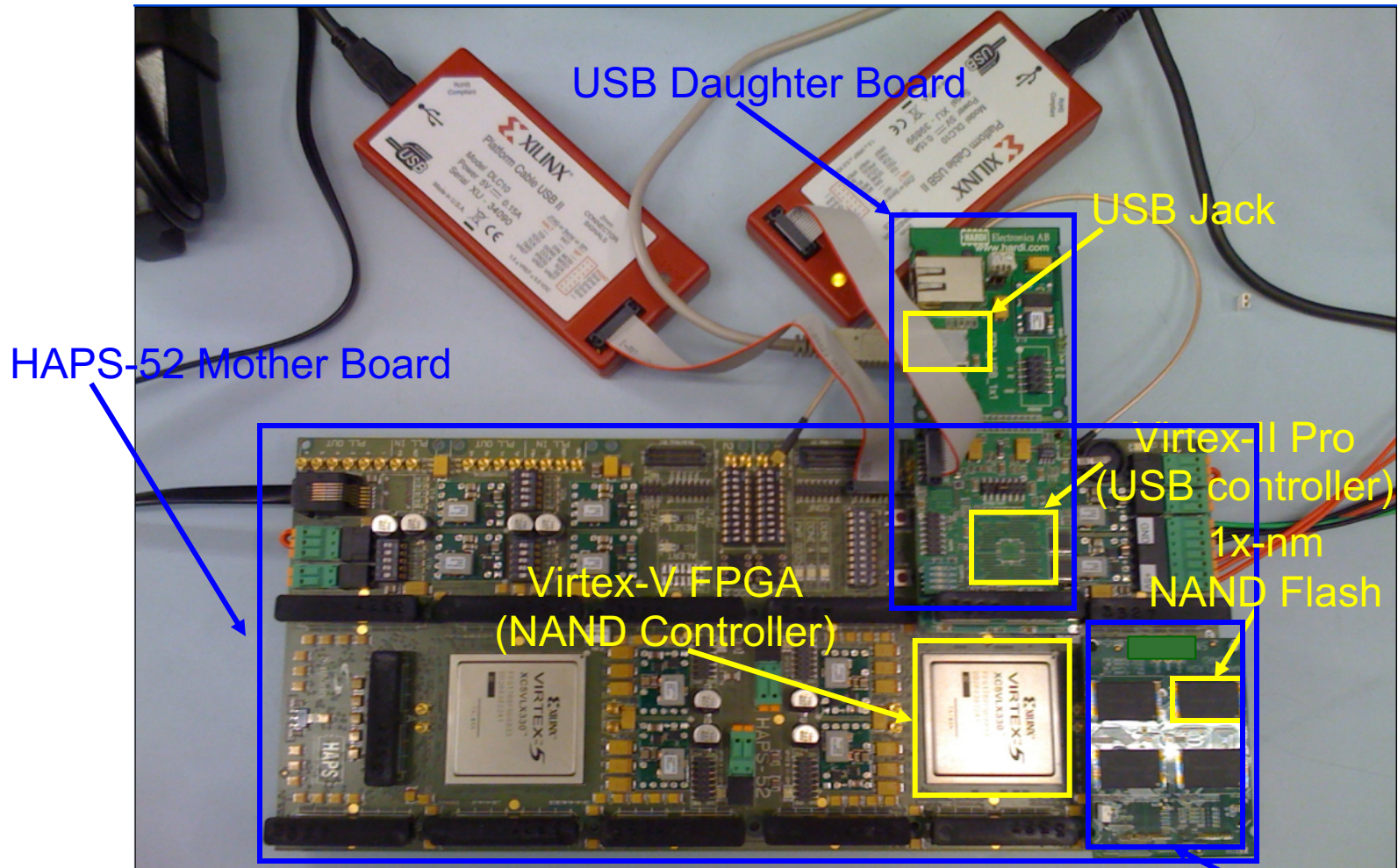
FPGAs

PC





# Understand and Model with Experiments (Flash)



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

NAND Daughter Board

# Recall: Collapse of the “Galloping Gertie”

---



# Another Example (1994)



# Yet Another Example (2007)

---



Source: Morry Gash/AP,  
<https://www.npr.org/2017/08/01/540669701/10-years-after-bridge-collapse-america-is-still-crumbling?t=1535427165809>

# A More Recent Example (2018)

---



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

# Final Thoughts on RowHammer

# Some Thoughts on RowHammer

---

- A simple hardware failure mechanism can create a widespread system security vulnerability
- How to exploit and fix the vulnerability requires a strong understanding across the transformation layers
  - And, a strong understanding of tools available to you
- Fixing needs to happen for two types of chips
  - Existing chips (already in the field)
  - Future chips
- Mechanisms for fixing are different between the two types



# Aside: Byzantine Failures

---

- This class of failures is known as **Byzantine failures**
- Characterized by
  - **Undetected erroneous computation**
  - Opposite of “fail fast (with an error or no result)”
- “erroneous” can be “malicious” (intent is the only distinction)
- Very difficult to detect and confine Byzantine failures
- **Do all you can to avoid them**
- Lamport et al., “The Byzantine Generals Problem,” ACM TOPLAS 1982.

# RowHammer, Revisited

- One can **predictably induce bit flips** in commodity DRAM chips
  - >80% of the tested DRAM chips are vulnerable
- First example of how a **simple hardware failure mechanism** can create a **widespread system security vulnerability**

**WIRED**

Forget Software—Now Hackers Are Exploiting Physics

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

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE



SHARE  
18276



TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

# RowHammer: Retrospective

---

- New mindset that has enabled a renewed interest in HW security attack research:
  - ❑ Real (memory) chips are vulnerable, in a simple and widespread manner → this causes real security problems
  - ❑ Hardware reliability → security connection is now mainstream discourse
- Many new RowHammer attacks...
  - ❑ Tens of papers in top security venues
  - ❑ **More to come** as RowHammer is getting worse (DDR4 & beyond)
- Many new RowHammer solutions...
  - ❑ Apple security release; Memtest86 updated
  - ❑ Many solution proposals in top venues (latest in ISCA 2019)
  - ❑ Principled system-DRAM co-design (in original RowHammer paper)
  - ❑ **More to come...**

# Perhaps Most Importantly...

---

- RowHammer enabled a shift of mindset in mainstream security researchers
  - **General-purpose hardware is fallible**, in a widespread manner
  - Its problems are exploitable
- This mindset has enabled many systems security researchers to examine hardware in more depth
  - And understand HW's inner workings and vulnerabilities
- It is no coincidence that two of the groups that discovered Meltdown and Spectre heavily worked on RowHammer attacks before
  - **More to come...**

# Summary: RowHammer

---

- DRAM reliability is reducing
- Reliability issues open up security vulnerabilities
  - Very hard to defend against
- **Rowhammer is a prime example**
  - First example of how a simple hardware failure mechanism can create a widespread system security vulnerability
  - Its implications on system security research are tremendous & exciting
- Bad news: RowHammer is getting worse.
- **Good news: We have a lot more to do.**
  - We are now fully aware hardware is easily fallible.
  - We are developing both attacks and solutions.
  - We are developing principled models, methodologies, solutions.

# For More on RowHammer...

---

- Onur Mutlu and Jeremie Kim,  
**"RowHammer: A Retrospective"**  
*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.*  
[\[Preliminary arXiv version\]](#)

## RowHammer: A Retrospective

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



Rowhammer

# Memory Systems and Memory-Centric Computing Systems

## Part 2: RowHammer

Prof. Onur Mutlu

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

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

3 February 2020

Champery Winter School

**SAFARI**

**ETH** zürich

**Carnegie Mellon**



# Future Memory Reliability/Security Challenges

# Future of Main Memory

---

- DRAM is becoming less reliable → more vulnerable

# Large-Scale Failure Analysis of DRAM Chips

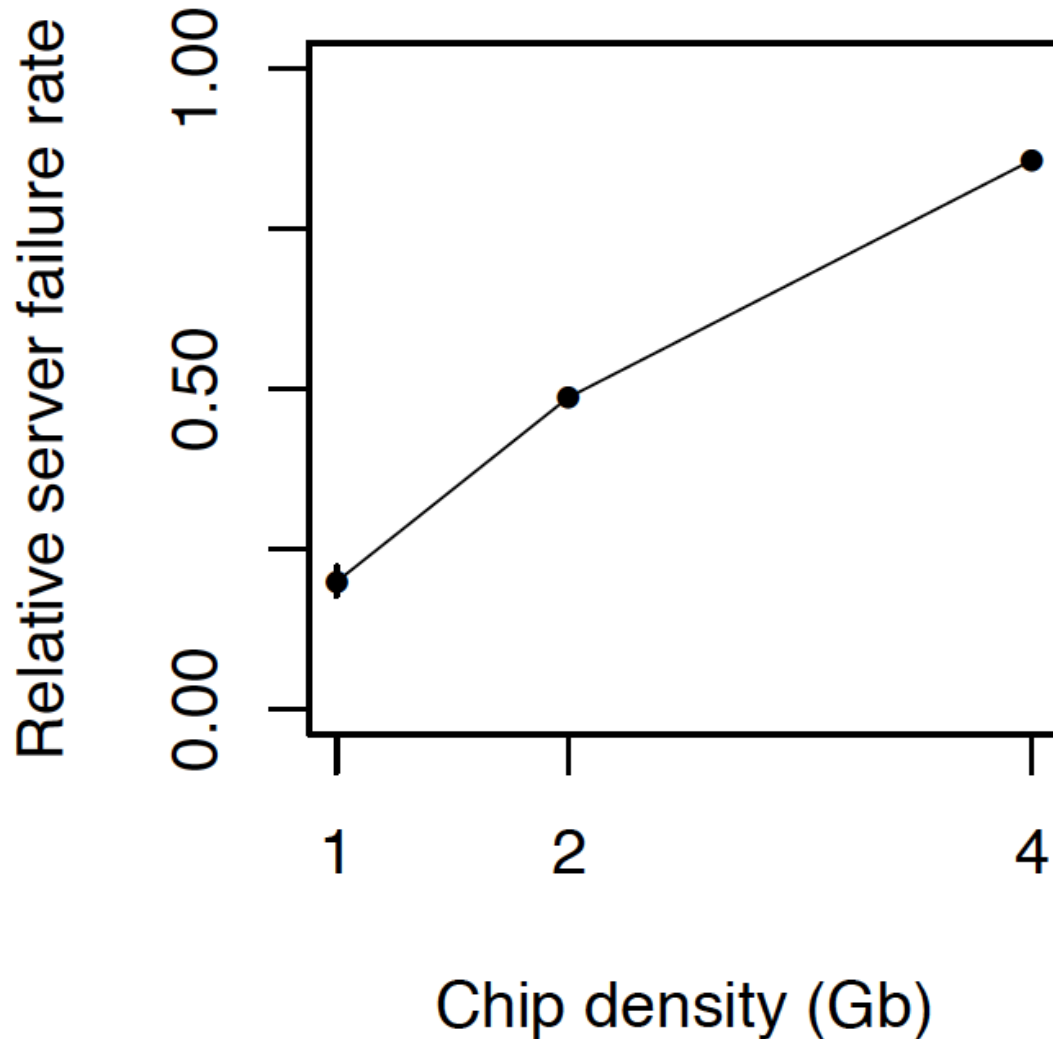
---

- 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.

# DRAM Reliability Reducing



*Intuition:  
quadratic  
increase in  
capacity*

# Aside: SSD Error Analysis in the Field

---

- First large-scale field study of flash memory errors
- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu, **"A Large-Scale Study of Flash Memory Errors in the Field"** *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Portland, OR, June 2015.  
[[Slides \(pptx\)](#)] [[pdf](#)] [[Coverage at ZDNet](#)]

## A Large-Scale Study of Flash Memory Failures in the Field

Justin Meza  
Carnegie Mellon University  
meza@cmu.edu

Qiang Wu  
Facebook, Inc.  
qwu@fb.com

Sanjeev Kumar  
Facebook, Inc.  
skumar@fb.com

Onur Mutlu  
Carnegie Mellon University  
onur@cmu.edu

# Future of Main Memory

---

- DRAM is becoming less reliable → more vulnerable
- Due to difficulties in DRAM scaling, other problems may also appear (or they may be going unnoticed)
- Some errors may already be slipping into the field
  - Read disturb errors (Rowhammer)
  - Retention errors
  - Read errors, write errors
  - ...
- These errors can also pose security vulnerabilities

# DRAM Data Retention Time Failures

---

- Determining the data retention time of a cell/row is getting more difficult
- Retention failures may already be slipping into the field

# Analysis of Data Retention Failures [ISCA'13]

---

- Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson, and Onur Mutlu, **"An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms"**  
*Proceedings of the 40th International Symposium on Computer Architecture (ISCA)*, Tel-Aviv, Israel, June 2013. [Slides \(ppt\)](#) [Slides \(pdf\)](#)

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

Jamie Liu\*

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

Ben Jaiyen\*

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

Yoongu Kim

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

Chris Wilkerson

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

Onur Mutlu

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

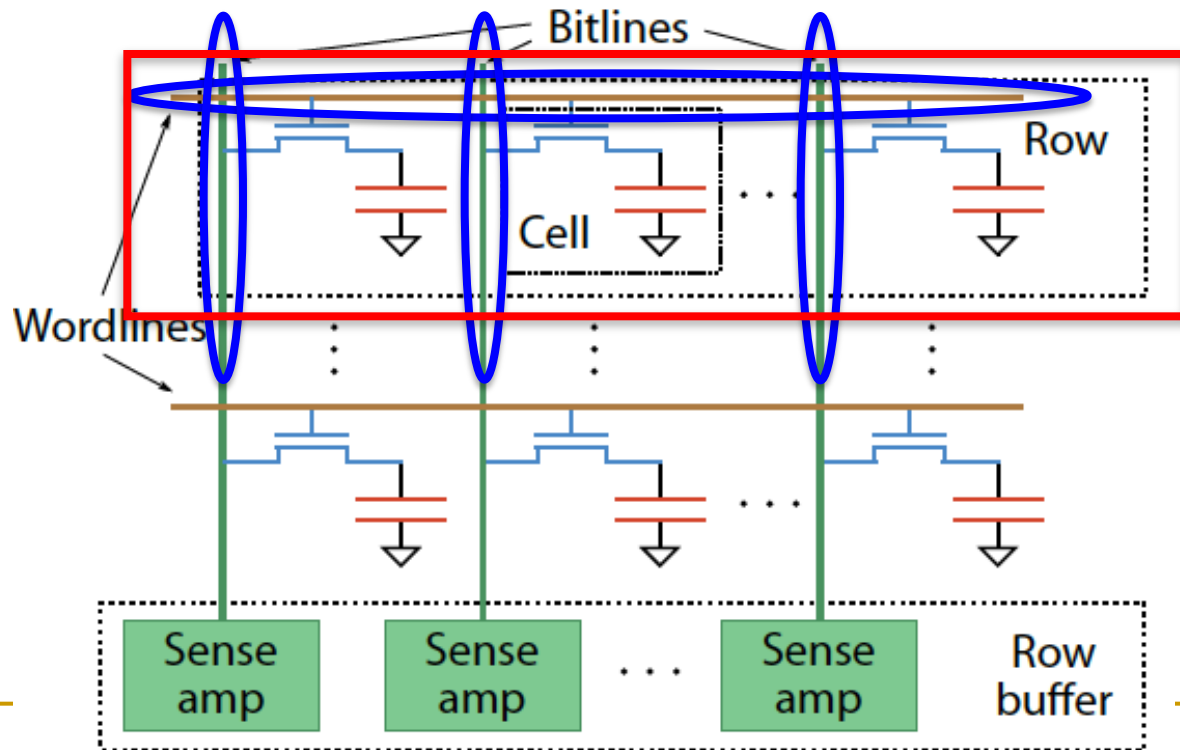




# Two Challenges to Retention Time Profiling

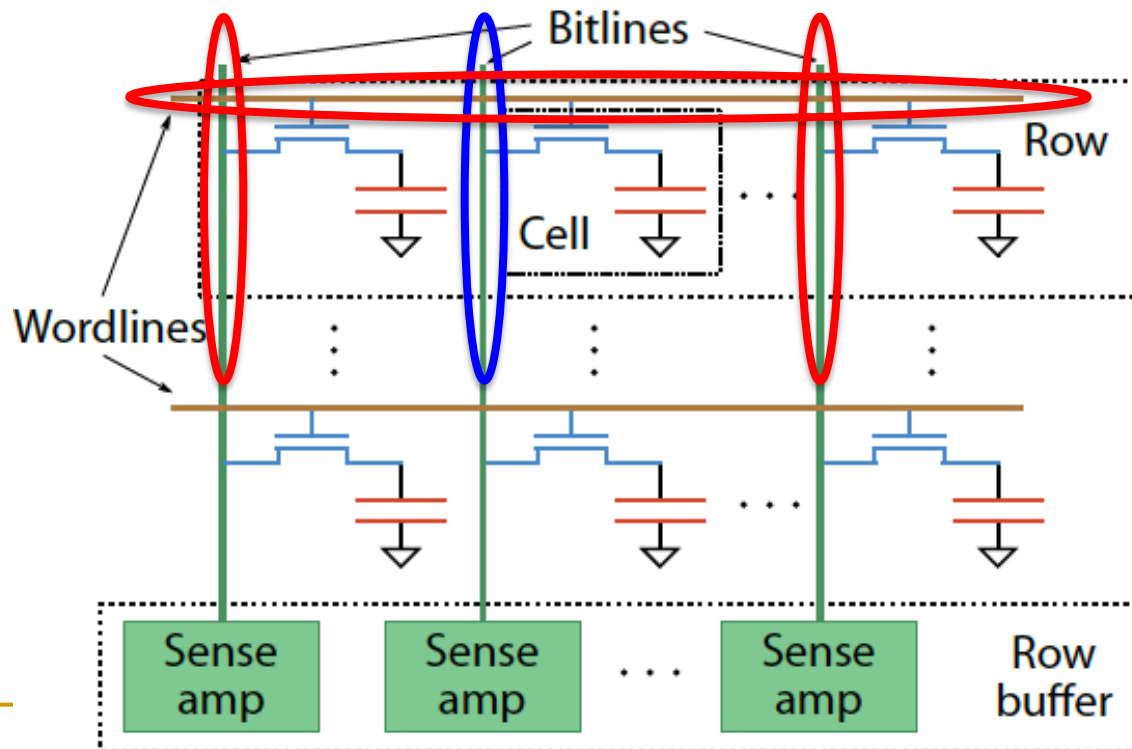
## ■ Challenge 1: Data Pattern Dependence (DPD)

- Retention time of a DRAM cell depends on its value and the values of cells nearby it
- When a row is activated, all bitlines are perturbed simultaneously



# Data Pattern Dependence

- Electrical noise on the bitline affects reliable sensing of a DRAM cell
- The magnitude of this noise is affected by values of nearby cells via
  - Bitline-bitline coupling → electrical coupling between adjacent bitlines
  - Bitline-wordline coupling → electrical coupling between each bitline and the activated wordline



# Data Pattern Dependence

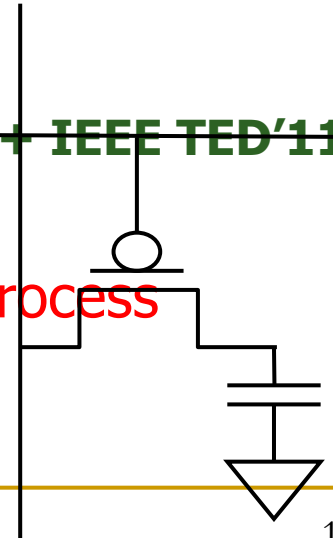
---

- Electrical noise on the bitline affects reliable sensing of a DRAM cell
- The magnitude of this noise is affected by values of nearby cells via
  - Bitline-bitline coupling → electrical coupling between adjacent bitlines
  - Bitline-wordline coupling → electrical coupling between each bitline and the activated wordline
  
- Retention time of a cell depends on data patterns stored in nearby cells
  - need to find the worst data pattern to find worst-case retention time
  - this pattern is location dependent

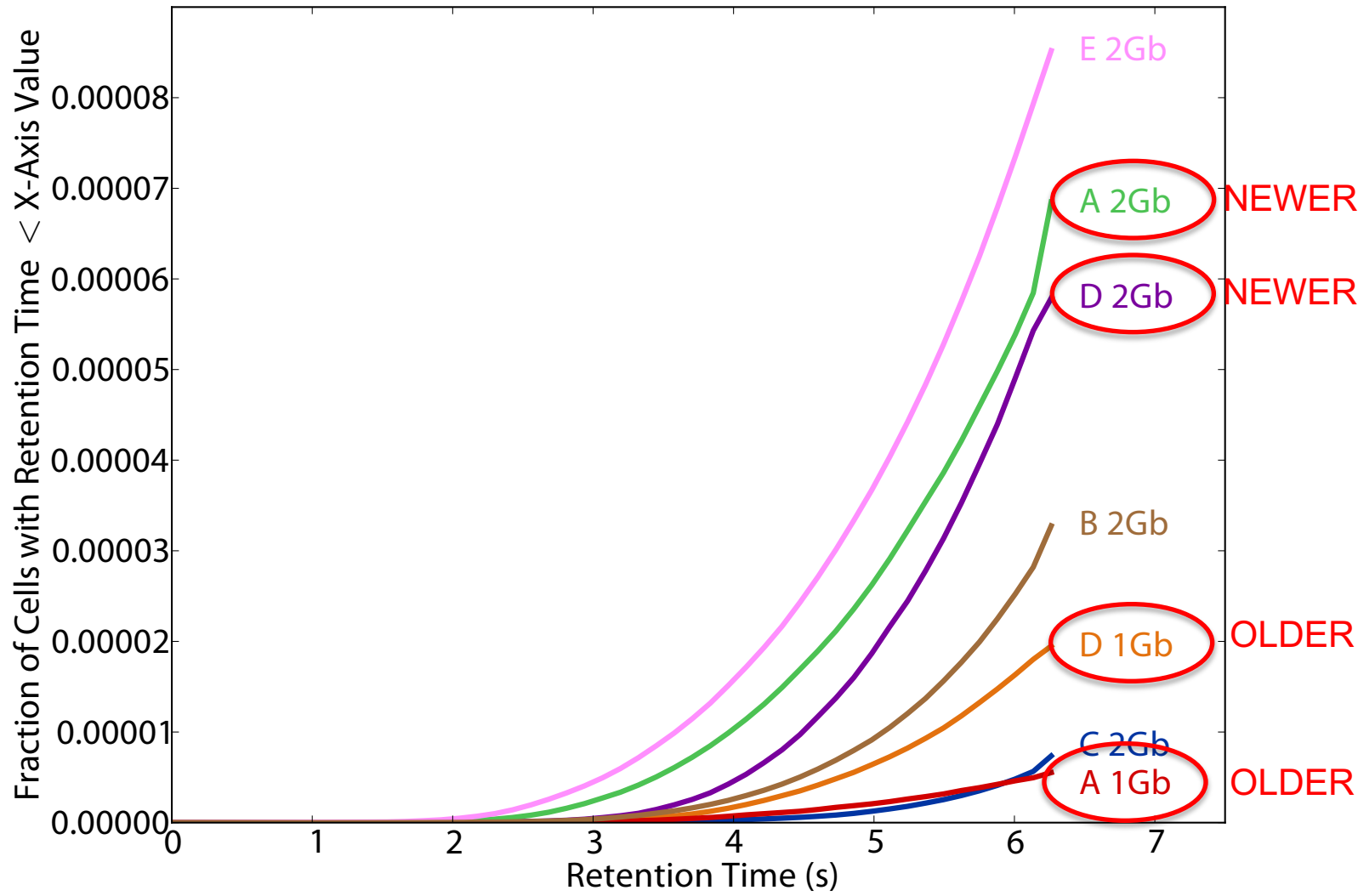
# Two Challenges to Retention Time Profiling

## ■ Challenge 2: Variable Retention Time (VRT)

- Retention time of a DRAM cell changes randomly over time
  - a cell alternates between multiple retention time states
- Leakage current of a cell changes sporadically due to a charge trap in the gate oxide of the DRAM cell access transistor
- When the trap becomes occupied, charge leaks more readily from the transistor's drain, leading to a short retention time
  - Called *Trap-Assisted Gate-Induced Drain Leakage*
- This process appears to be a random process [Kim+ IEEE TED'11]
- Worst-case retention time depends on a random process
  - need to find the worst case despite this

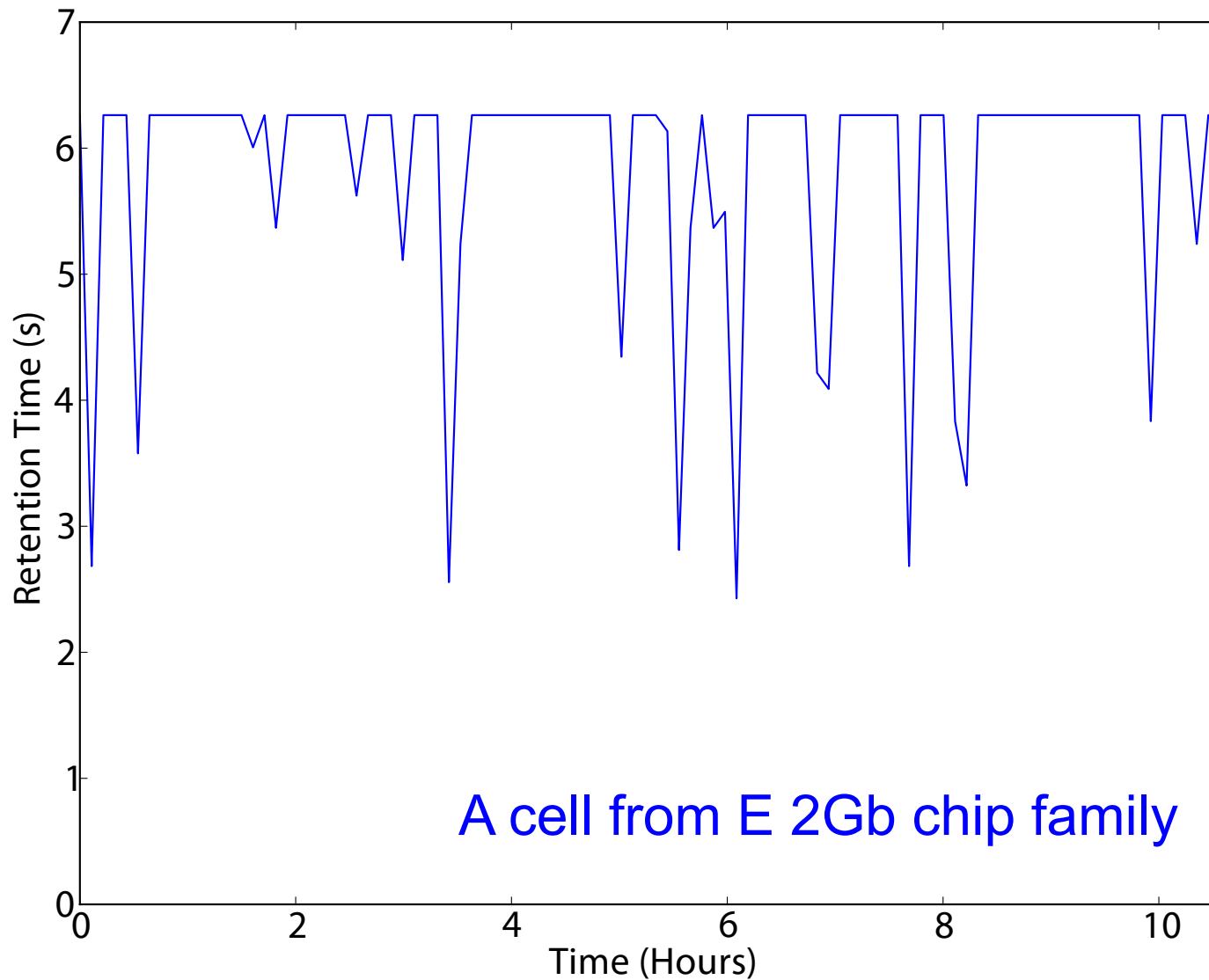


# Modern DRAM Retention Time Distribution

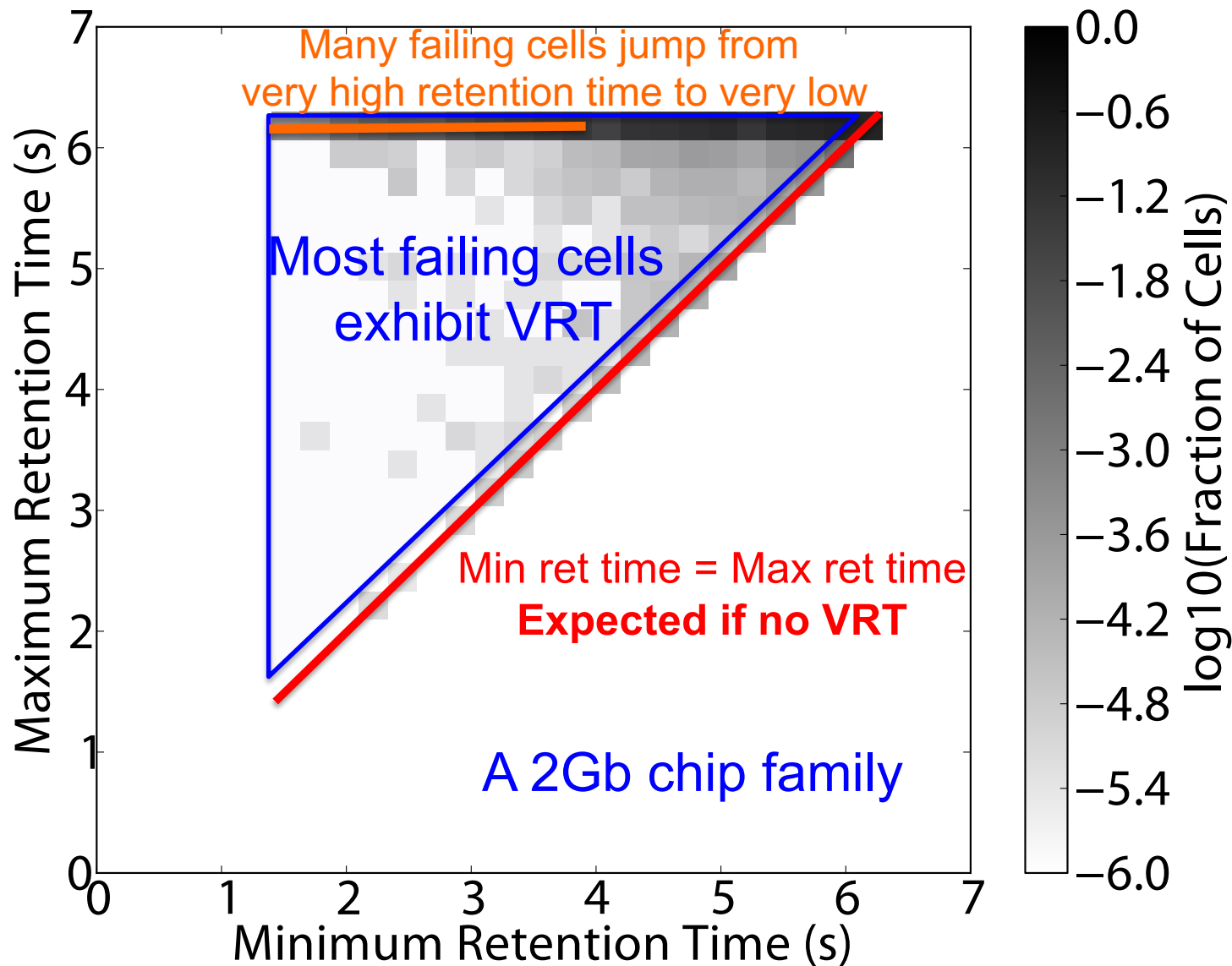


**Newer device families have more weak cells than older ones**  
**Likely a result of technology scaling**

# An Example VRT Cell



# Variable Retention Time





# More on Data Retention Failures [ISCA'13]

---

- Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson, and Onur Mutlu, **"An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms"**  
*Proceedings of the 40th International Symposium on Computer Architecture (ISCA)*, Tel-Aviv, Israel, June 2013. [Slides \(ppt\)](#) [Slides \(pdf\)](#)

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

Jamie Liu\*

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

Ben Jaiyen\*

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

Yoongu Kim

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

Chris Wilkerson

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

Onur Mutlu

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

# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

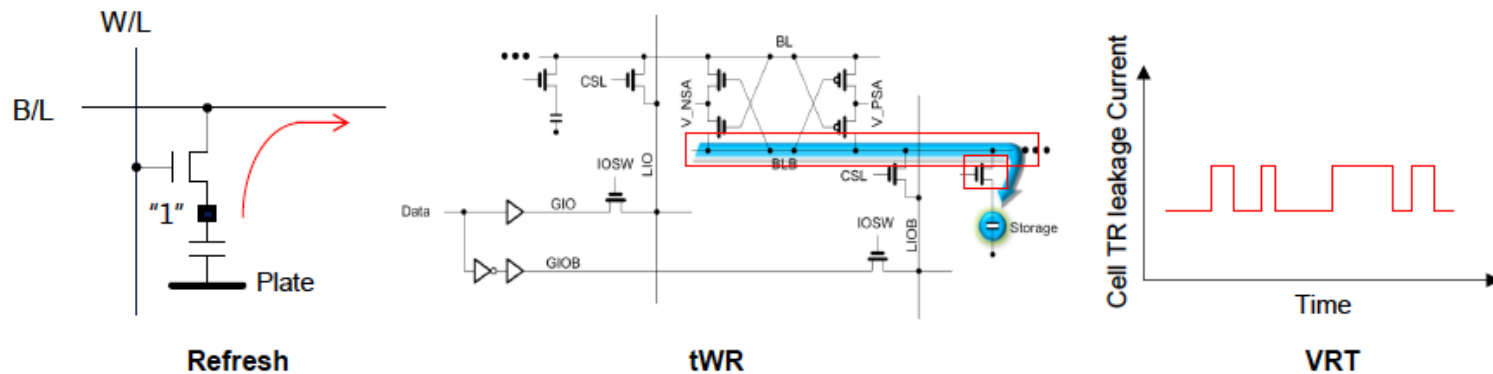
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

### ❖ 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



# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

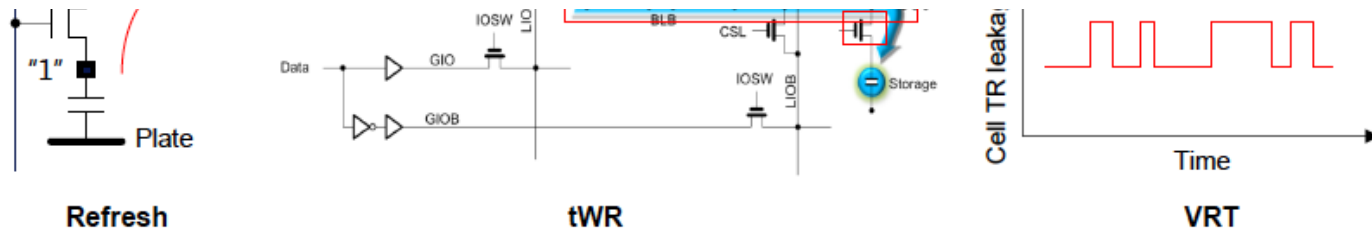
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

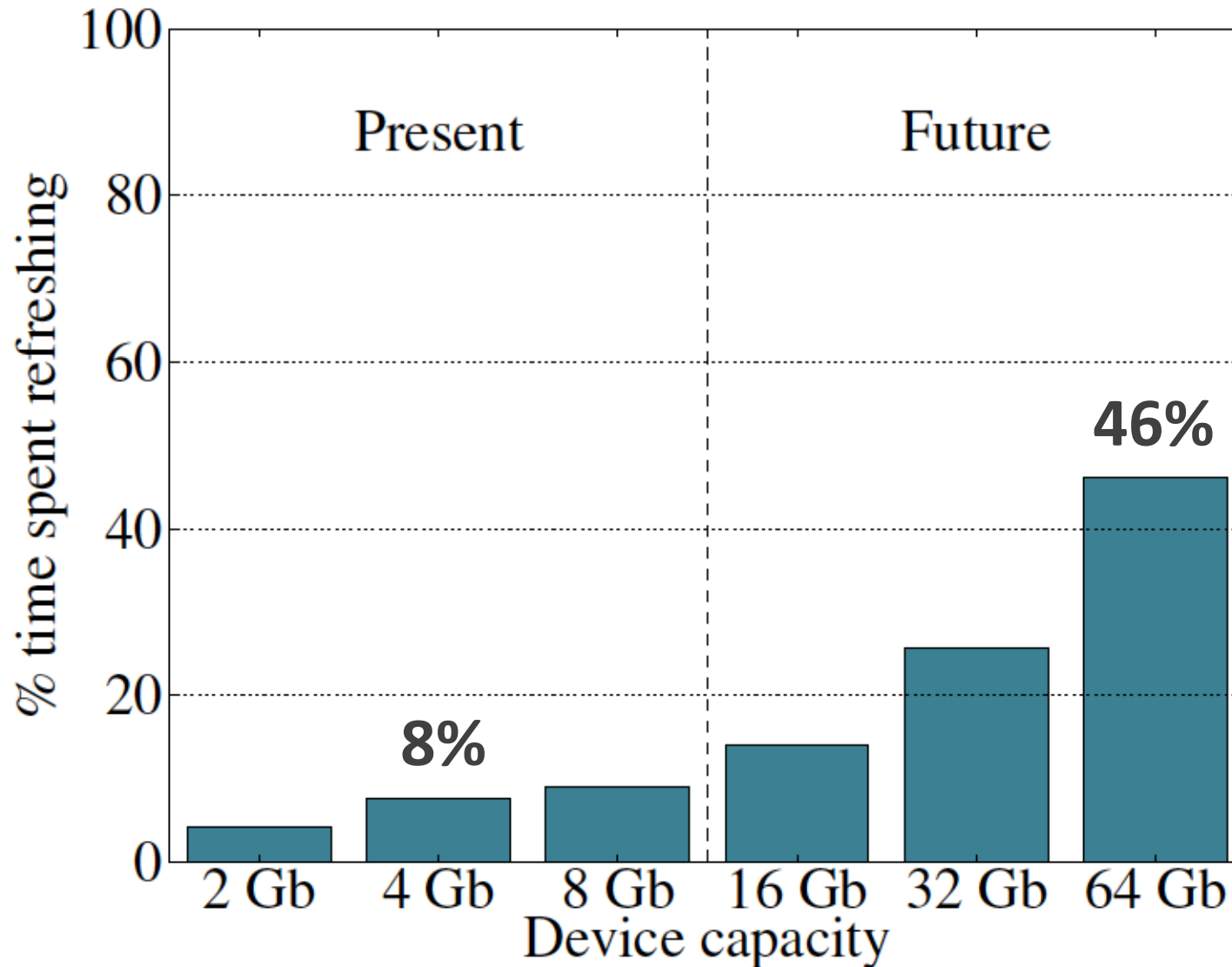
# Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

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

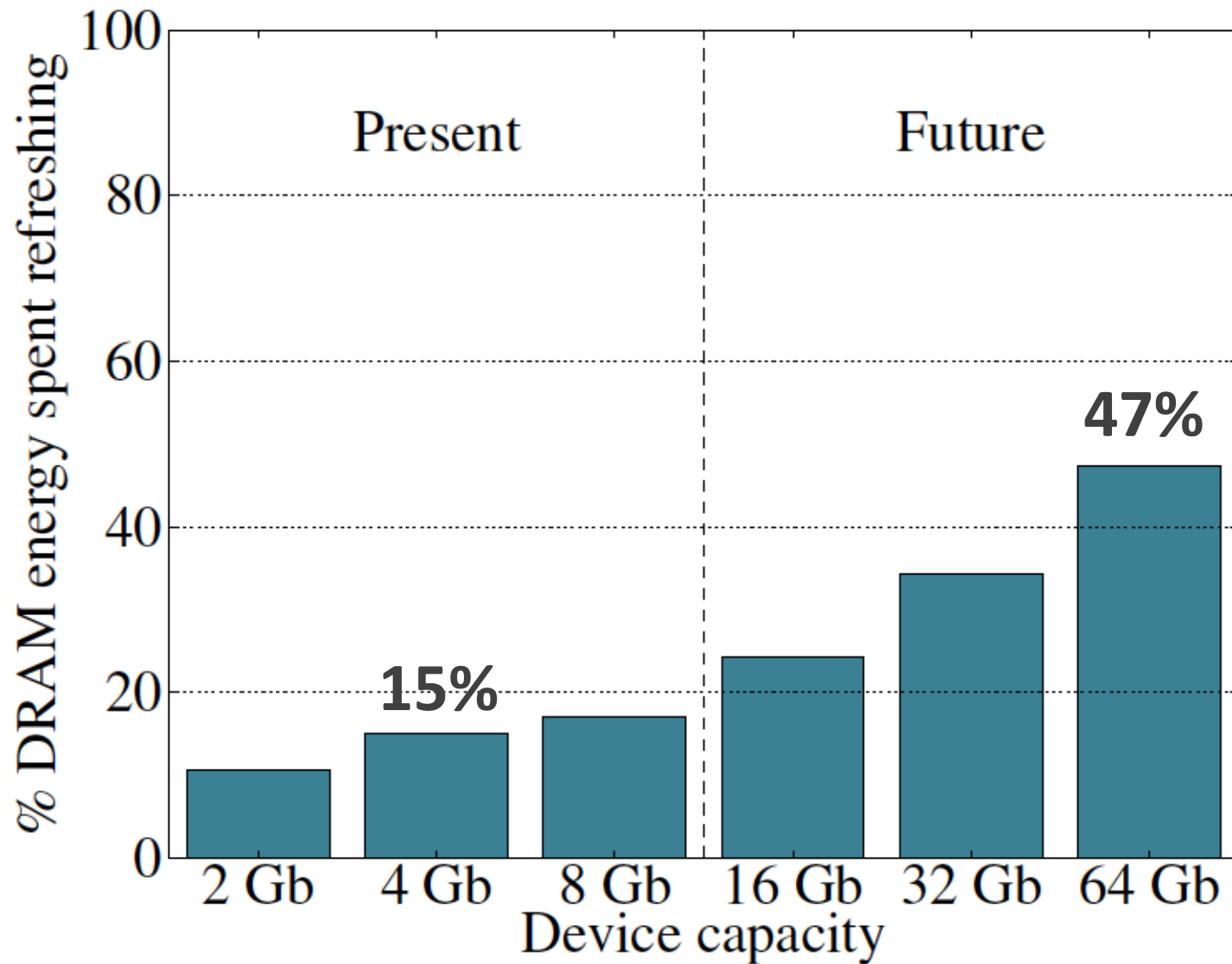
*Samsung Electronics, Hwasung, Korea / \*Samsung Electronics, San Jose / \*\*Intel*



# Refresh Overhead: Performance



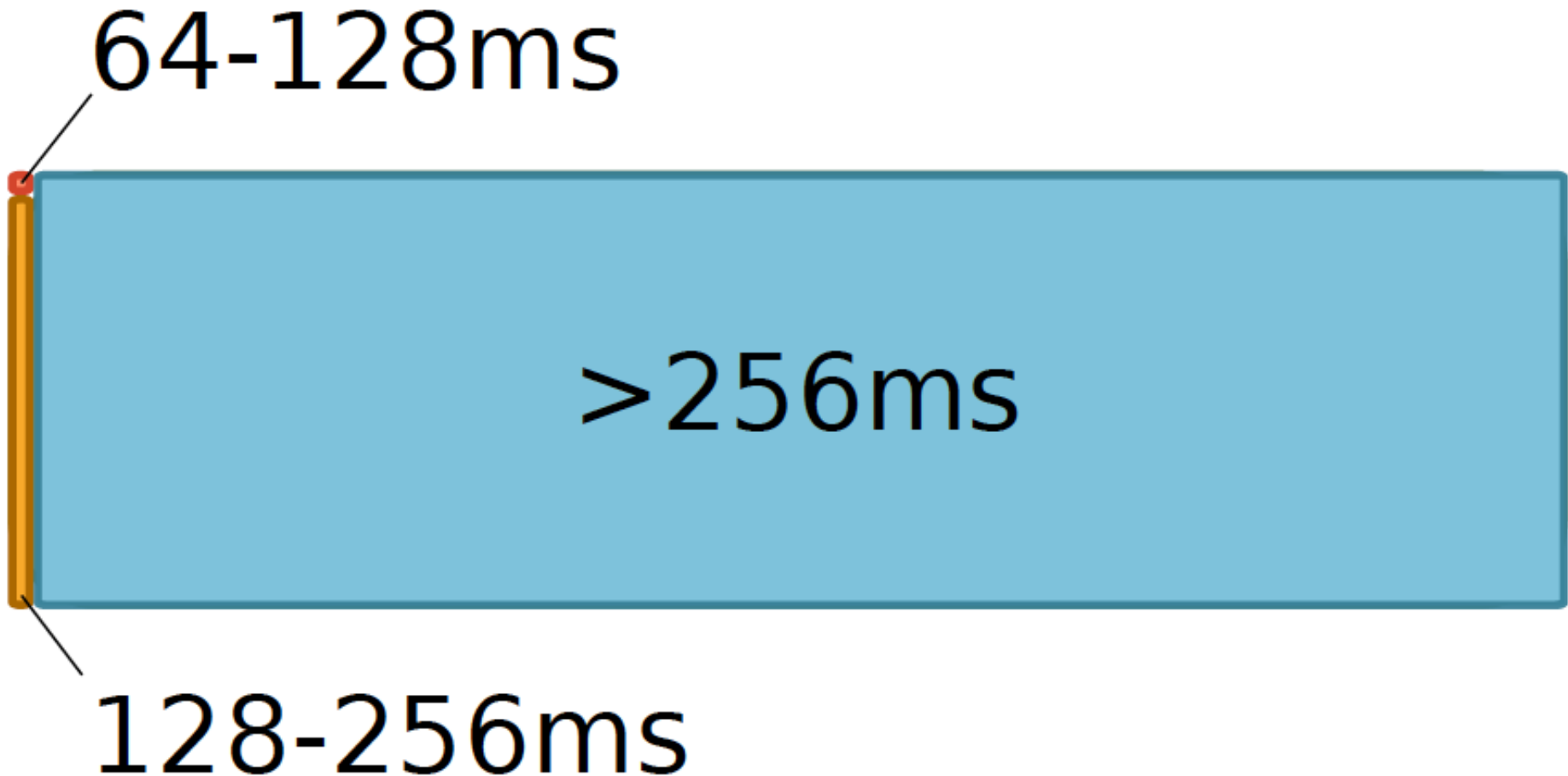
# Refresh Overhead: Energy



# Most Refreshes Are Unnecessary

---

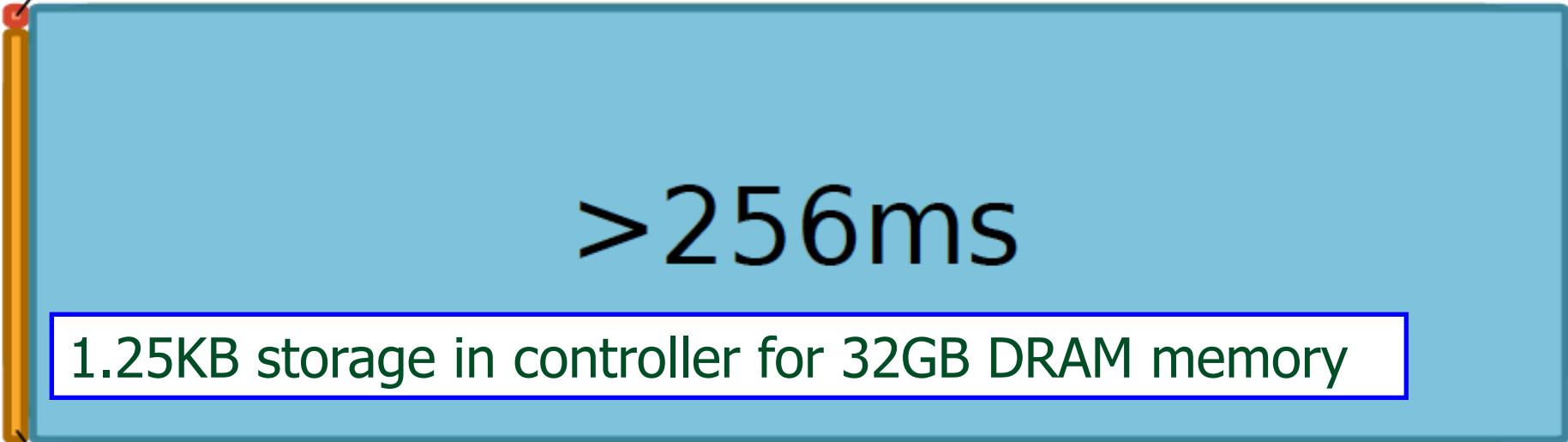
- Retention Time Profile of DRAM looks like this:



# RAIDR: Eliminating Unnecessary Refreshes

---

64-128ms



>256ms

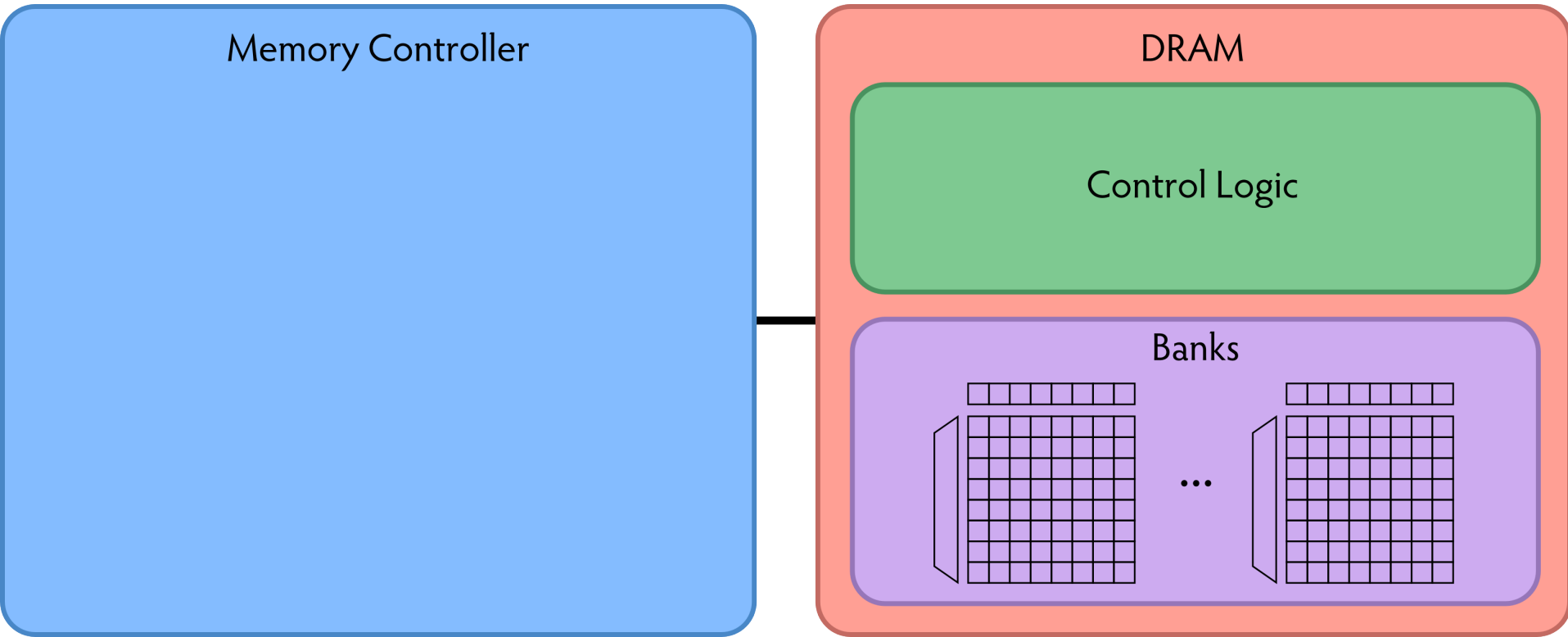
1.25KB storage in controller for 32GB DRAM memory

128-256ms

Can reduce refreshes by  $\sim 75\%$   
→ reduces energy consumption and improves performance

# RAIDR: Baseline Design

---

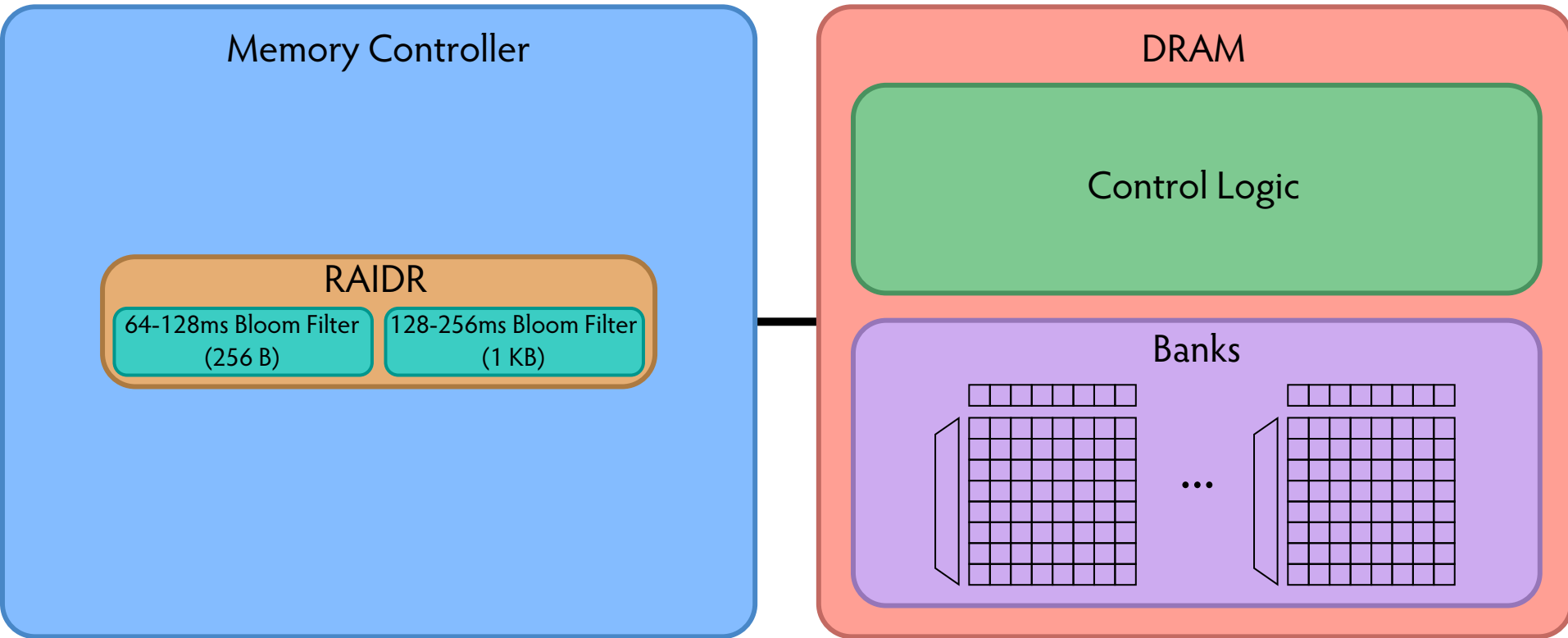


Refresh control is in DRAM in today's auto-refresh systems

RAIDR can be implemented in either the controller or DRAM

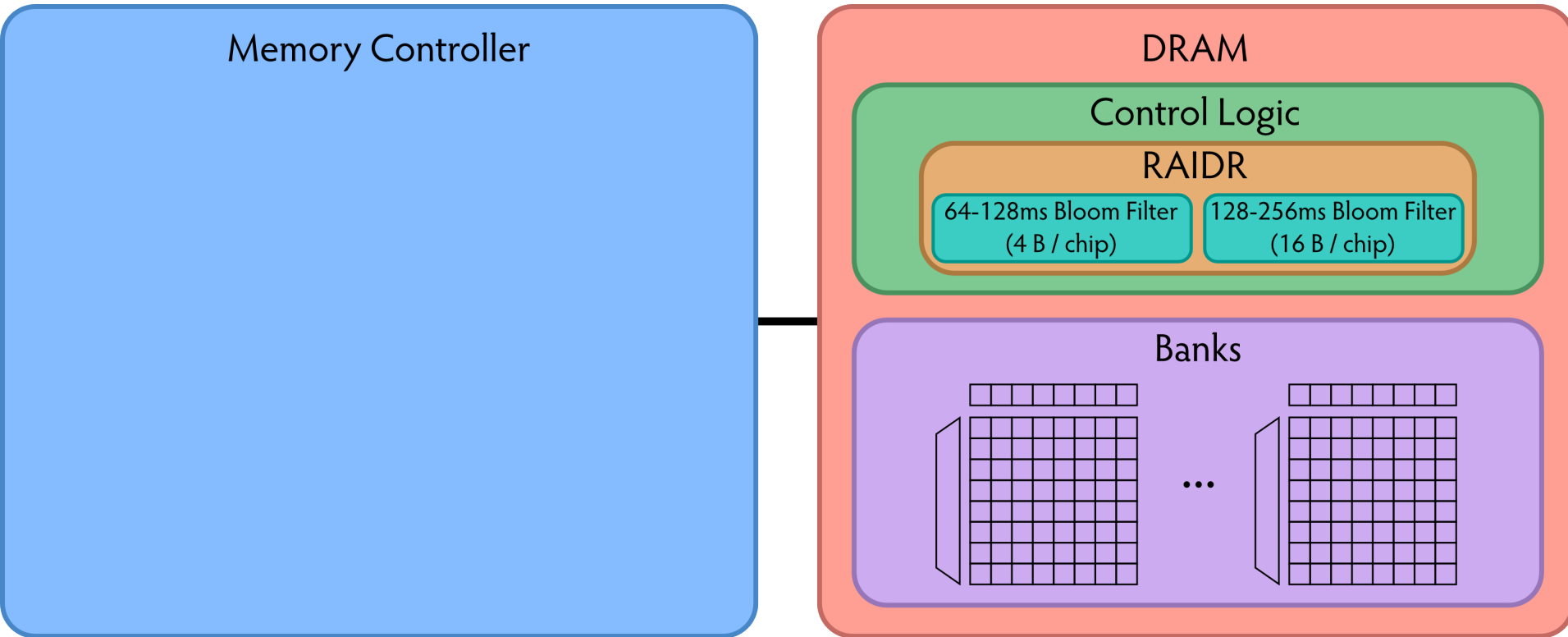


# RAIDR in Memory Controller: Option 1



Overhead of RAIDR in DRAM controller:  
1.25 KB Bloom Filters, 3 counters, additional commands  
issued for per-row refresh (all accounted for in evaluations)

# RAIDR in DRAM Chip: Option 2



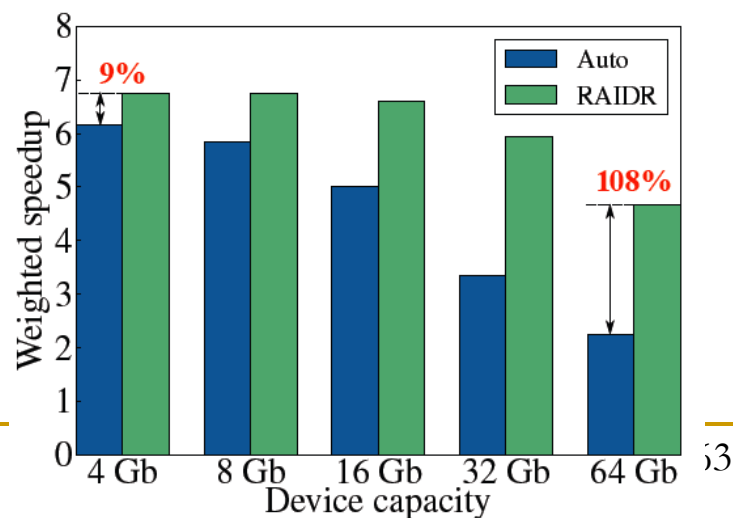
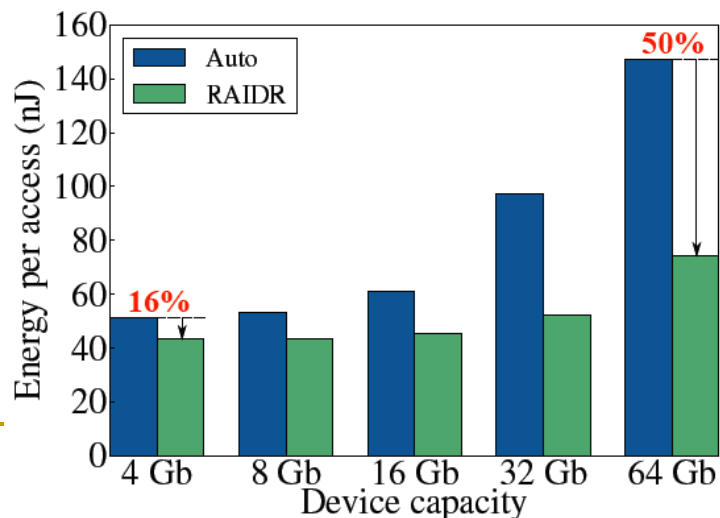
Overhead of RAIDR in DRAM chip:

Per-chip overhead: 20B Bloom Filters, 1 counter (4 Gbit chip)

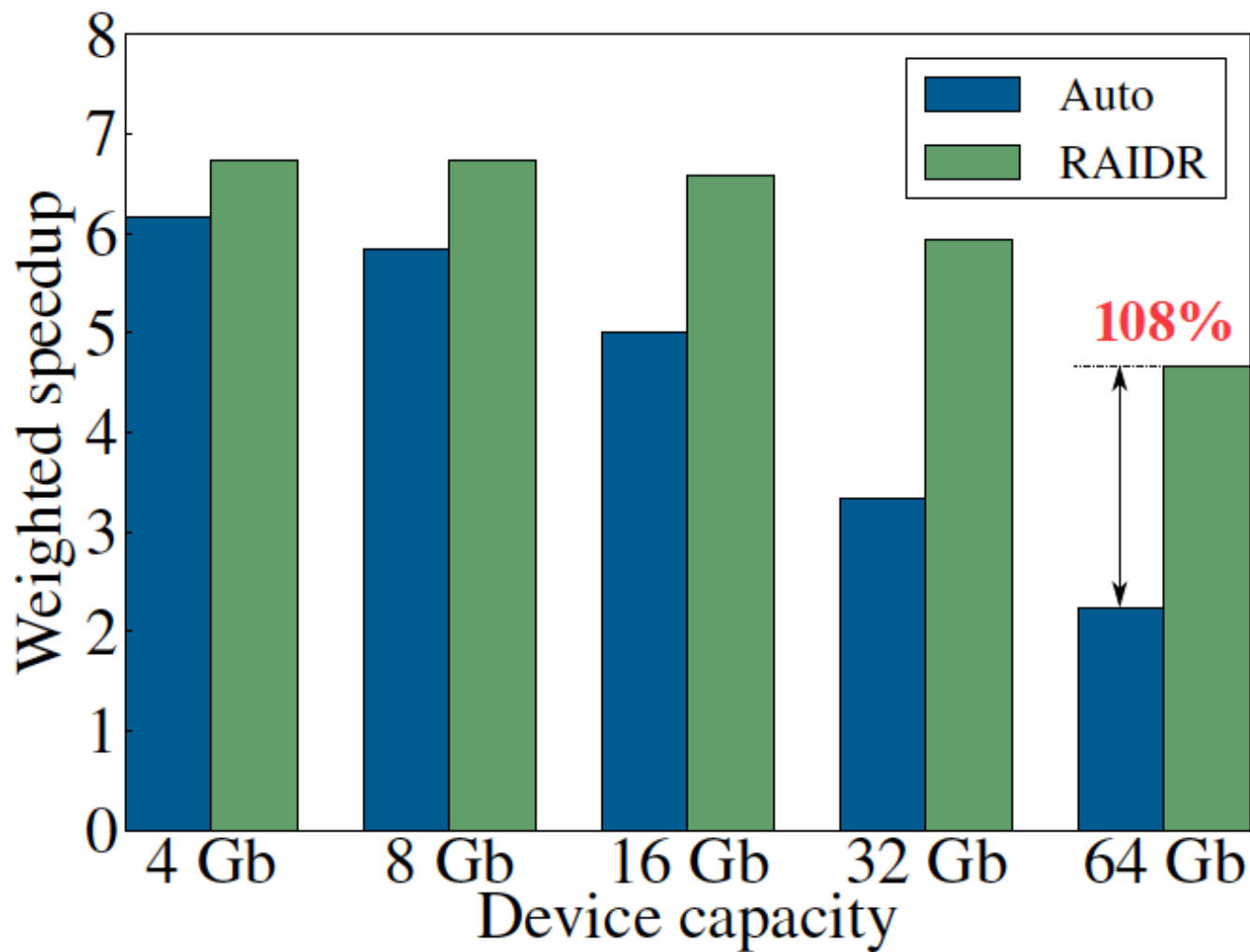
Total overhead: 1.25KB Bloom Filters, 64 counters (32 GB DRAM)

# RAIDR: Results and Takeaways

- System: 32GB DRAM, 8-core; SPEC, TPC-C, TPC-H workloads
- RAIDR hardware cost: 1.25 kB (2 Bloom filters)
- Refresh reduction: 74.6%
- Dynamic DRAM energy reduction: 16%
- Idle DRAM power reduction: 20%
- Performance improvement: 9%
- Benefits increase as DRAM scales in density

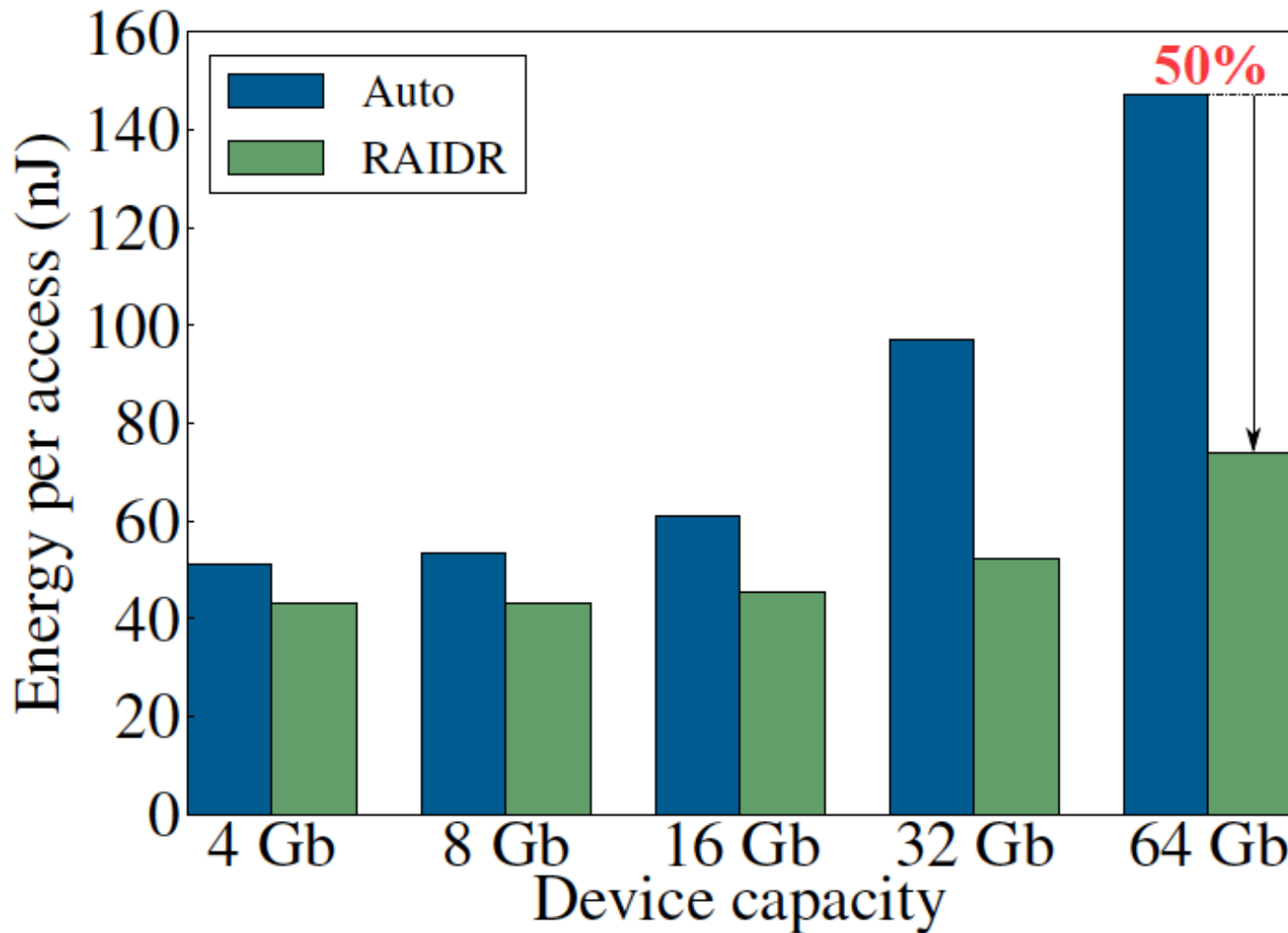


# DRAM Device Capacity Scaling: Performance



RAIDR performance benefits increase with DRAM chip capacity

# DRAM Device Capacity Scaling: Energy



RAIDR energy benefits increase with DRAM chip capacity

# RAIDR: Eliminating Unnecessary Refreshes

■ Observation: Most DRAM rows can be refreshed much less often without losing data [Kim+, EDL'09][Liu+ ISCA'13]

■ Key idea: Refresh rows containing weak cells more frequently, other rows less frequently

1. Profiling: Profile retention time of all rows

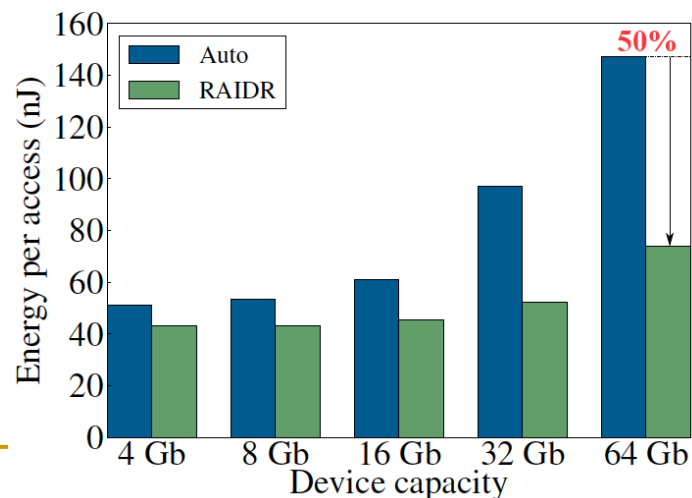
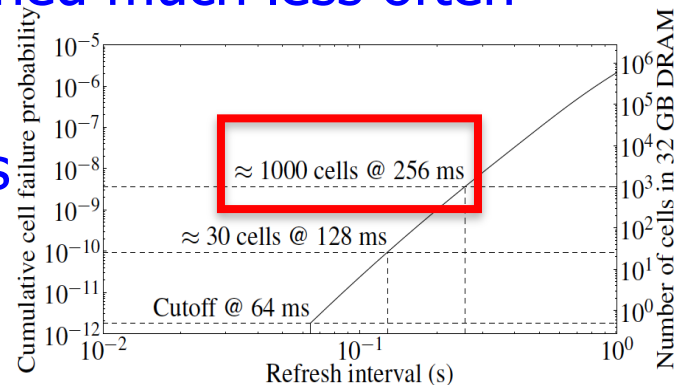
2. Binning: Store rows into bins by retention time in memory controller

*Efficient storage with Bloom Filters (only 1.25KB for 32GB memory)*

3. Refreshing: Memory controller refreshes rows in different bins at different rates

■ Results: 8-core, 32GB, SPEC, TPC-C, TPC-H

- 74.6% refresh reduction @ 1.25KB storage
- ~16%/20% DRAM dynamic/idle power reduction
- ~9% performance improvement
- Benefits increase with DRAM capacity



# More on RAIDR: Perf+Energy Perspective

---

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

## **RAIDR: Retention-Aware Intelligent DRAM Refresh**

Jamie Liu    Ben Jaiyen    Richard Veras    Onur Mutlu  
Carnegie Mellon University

# Finding DRAM Retention Failures

---

- How can we reliably find the retention time of all DRAM cells?
- Goals: so that we can
  - Make DRAM reliable and secure
  - Make techniques like RAIDR work
    - improve performance and energy



# Mitigation of Retention Issues [SIGMETRICS'14]

---

- Samira Khan, Donghyuk Lee, Yoongu Kim, Alaa Alameldeen, Chris Wilkerson, and Onur Mutlu,  
**"The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study"**  
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Austin, TX, June 2014. [[Slides \(pptx\)](#)] [[pdf](#)] [[Poster \(pptx\)](#)] [[pdf](#)] [[Full data sets](#)]

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

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

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

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

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

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

Onur Mutlu<sup>†</sup>  
onur@cmu.edu

<sup>†</sup>Carnegie Mellon University

<sup>\*</sup>Intel Labs

# Towards an Online Profiling System

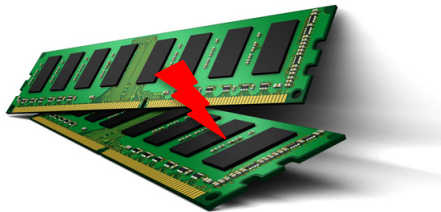
## *Key Observations:*

- **Testing** alone **cannot detect** all possible failures
- **Combination** of ECC and other mitigation techniques is much more **effective**
  - **But degrades performance**
- **Testing** can help to reduce the **ECC strength**
  - Even when starting with a **higher strength ECC**

# Towards an Online Profiling System

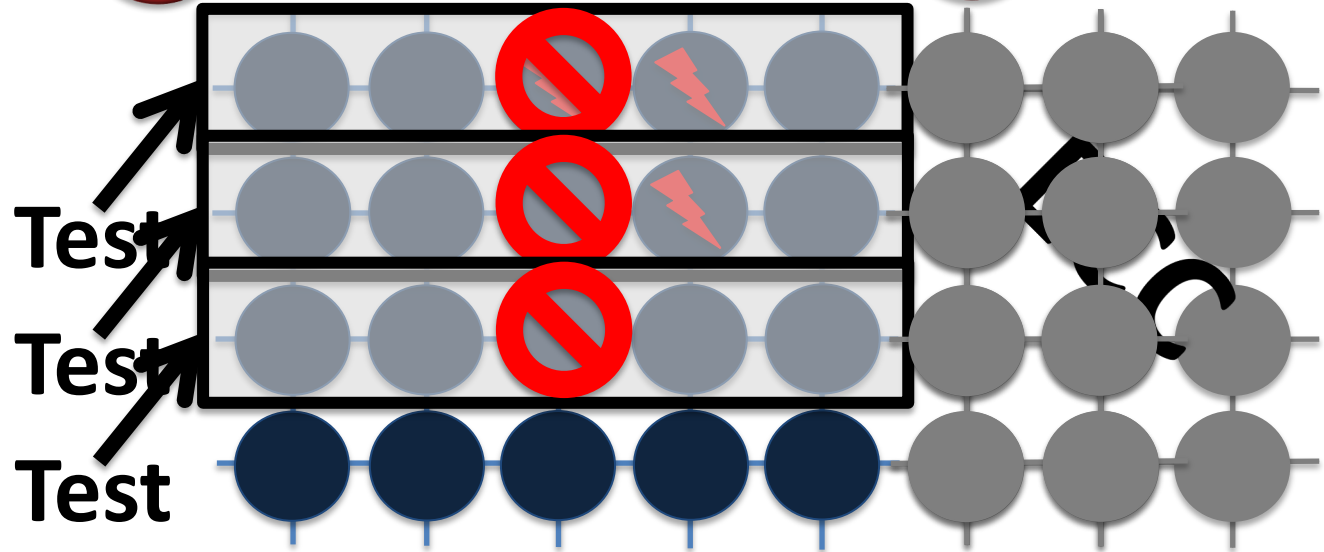
Initially Protect DRAM  
with Strong ECC

1



Periodically Test  
Parts of DRAM

2



Mitigate errors and  
reduce ECC

3

Run tests periodically after a short interval  
at smaller regions of memory

# Handling Variable Retention Time [DSN'15]

---

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

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

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

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

Moinuddin K. Qureshi<sup>†</sup>      Dae-Hyun Kim<sup>†</sup>      Samira Khan<sup>‡</sup>      Prashant J. Nair<sup>†</sup>      Onur Mutlu<sup>‡</sup>  
<sup>†</sup>Georgia Institute of Technology      <sup>‡</sup>Carnegie Mellon University  
{*moin, dhkim, pnair6*}@ece.gatech.edu      {*samirakhan, onur*}@cmu.edu

# AVATAR

**Insight:** Avoid retention failures → Upgrade row on ECC error

**Observation:** Rate of VRT  $\gg$  Rate of soft error (50x-2500x)

Scrub  
(15 min)



DRAM Rows

ECC	A
ECC	B
ECC	C
ECC	D
ECC	E
ECC	F
ECC	G
ECC	H

Ref. Rate Table

0
0
1
0
0
0
1
1

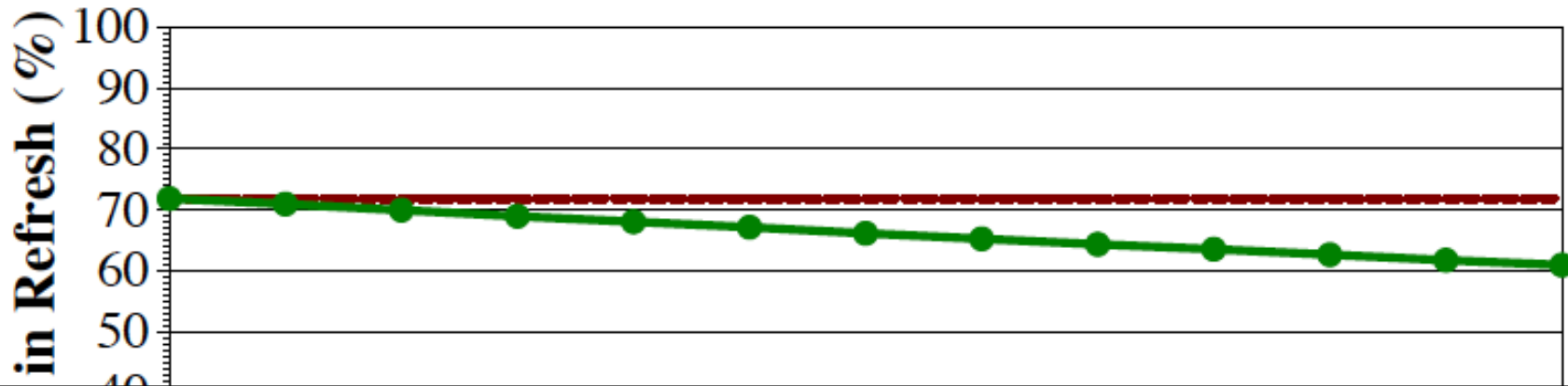
Weak Cell

RETENTION  
PROFILING

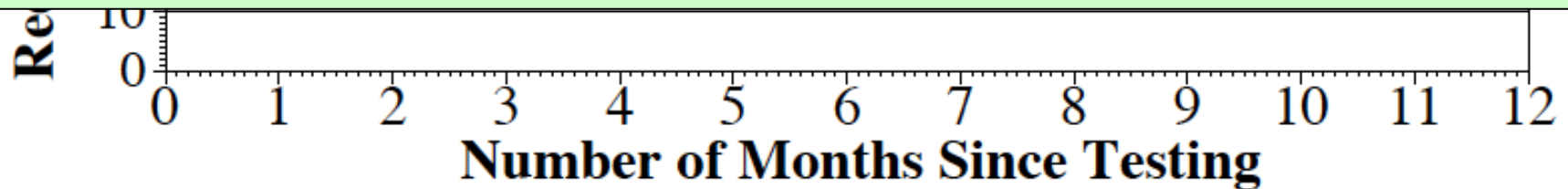
Row protected from  
future  
retention failures

**AVATAR mitigates VRT by increasing refresh rate on error**

# RESULTS: REFRESH SAVINGS

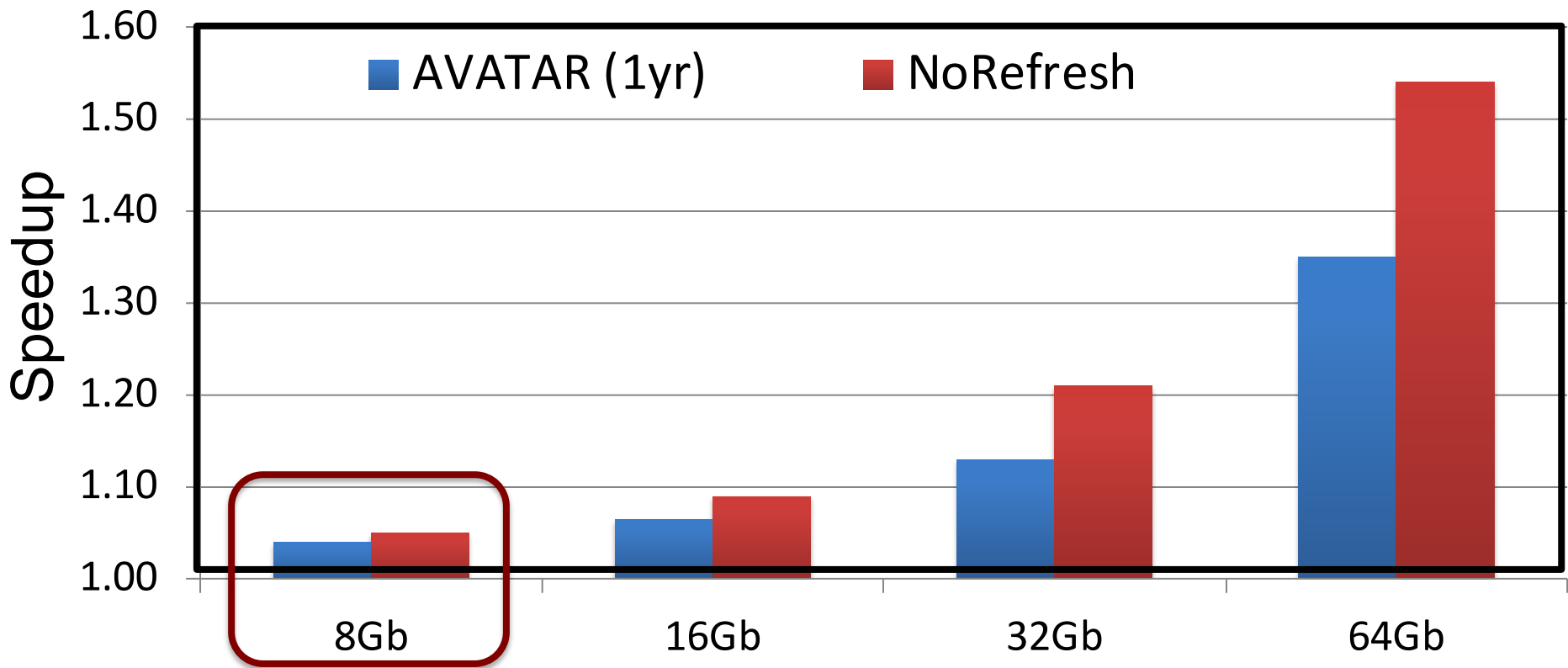


**Retention Testing Once a Year can revert refresh saving from 60% to 70%**



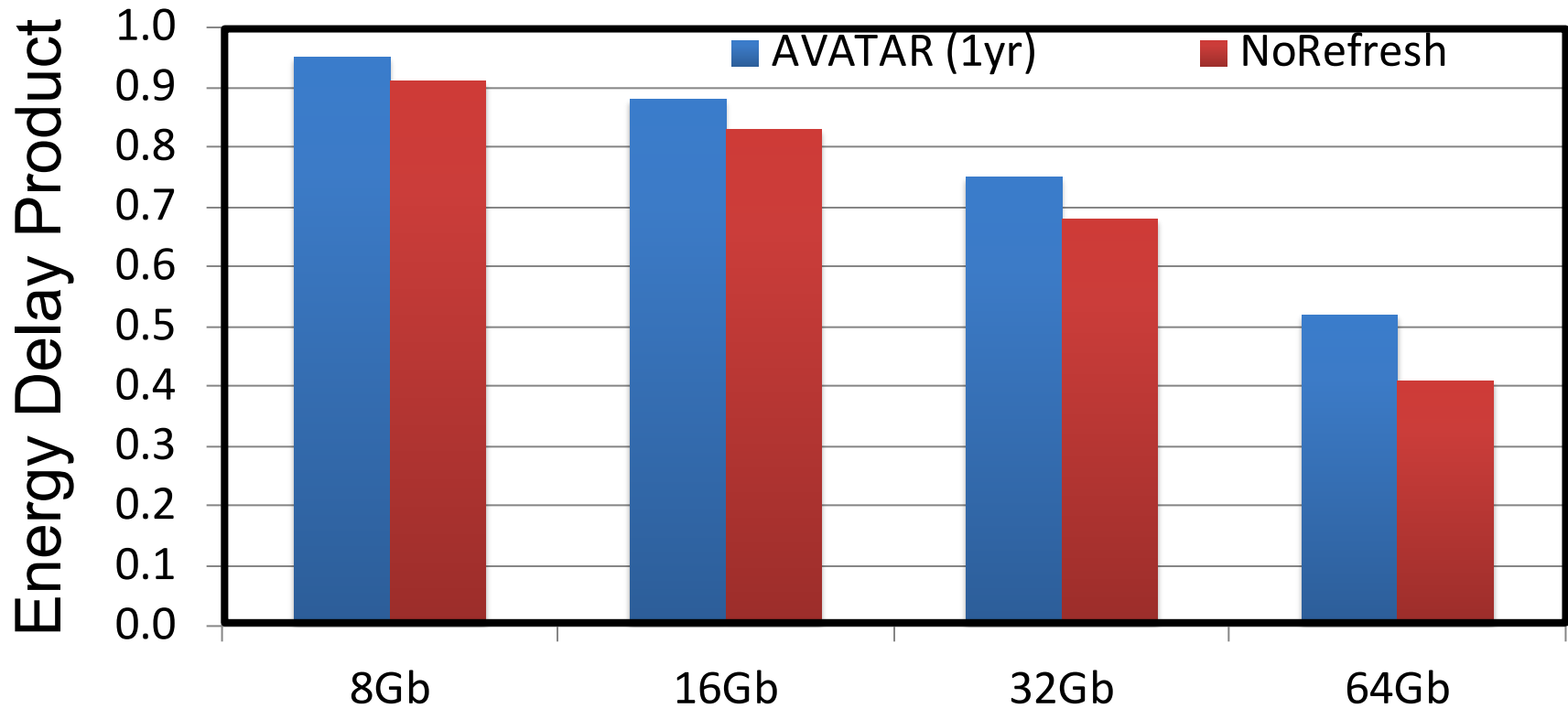
**AVATAR reduces refresh by 60%-70%, similar to multi rate refresh but with VRT tolerance**

# SPEEDUP



**AVATAR gets 2/3<sup>rd</sup> the performance of NoRefresh. More gains at higher capacity nodes**

# ENERGY DELAY PRODUCT



**AVATAR reduces EDP,  
Significant reduction at higher capacity nodes**



# Handling Data-Dependent Failures [DSN'16]

---

- Samira Khan, Donghyuk Lee, and Onur Mutlu,  
**"PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM"**  
*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Toulouse, France, June 2016.  
[\[Slides \(pptx\)\]](#) [\[pdf\]](#)

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

Samira Khan<sup>\*</sup>

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

Donghyuk Lee<sup>†‡</sup>

<sup>†</sup>Carnegie Mellon University

Onur Mutlu<sup>\*†</sup>

<sup>‡</sup>Nvidia

<sup>\*</sup>ETH Zürich

# Handling Data-Dependent Failures [MICRO'17]

---

- Samira Khan, Chris Wilkerson, Zhe Wang, Alaa R. Alameldeen, Donghyuk Lee, and Onur Mutlu,  
**"Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content"**  
*Proceedings of the 50th International Symposium on Microarchitecture (MICRO), Boston, MA, USA, October 2017.*  
[\[Slides \(pptx\) \(pdf\)\]](#) [\[Lightning Session Slides \(pptx\) \(pdf\)\]](#) [\[Poster \(pptx\) \(pdf\)\]](#)

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

Samira Khan<sup>\*</sup> Chris Wilkerson<sup>†</sup> Zhe Wang<sup>†</sup> Alaa R. Alameldeen<sup>†</sup> Donghyuk Lee<sup>‡</sup> Onur Mutlu<sup>\*</sup>  
<sup>\*</sup>University of Virginia    <sup>†</sup>Intel Labs    <sup>‡</sup>Nvidia Research    <sup>\*</sup>ETH Zürich

# Handling Both DPD and VRT [ISCA'17]

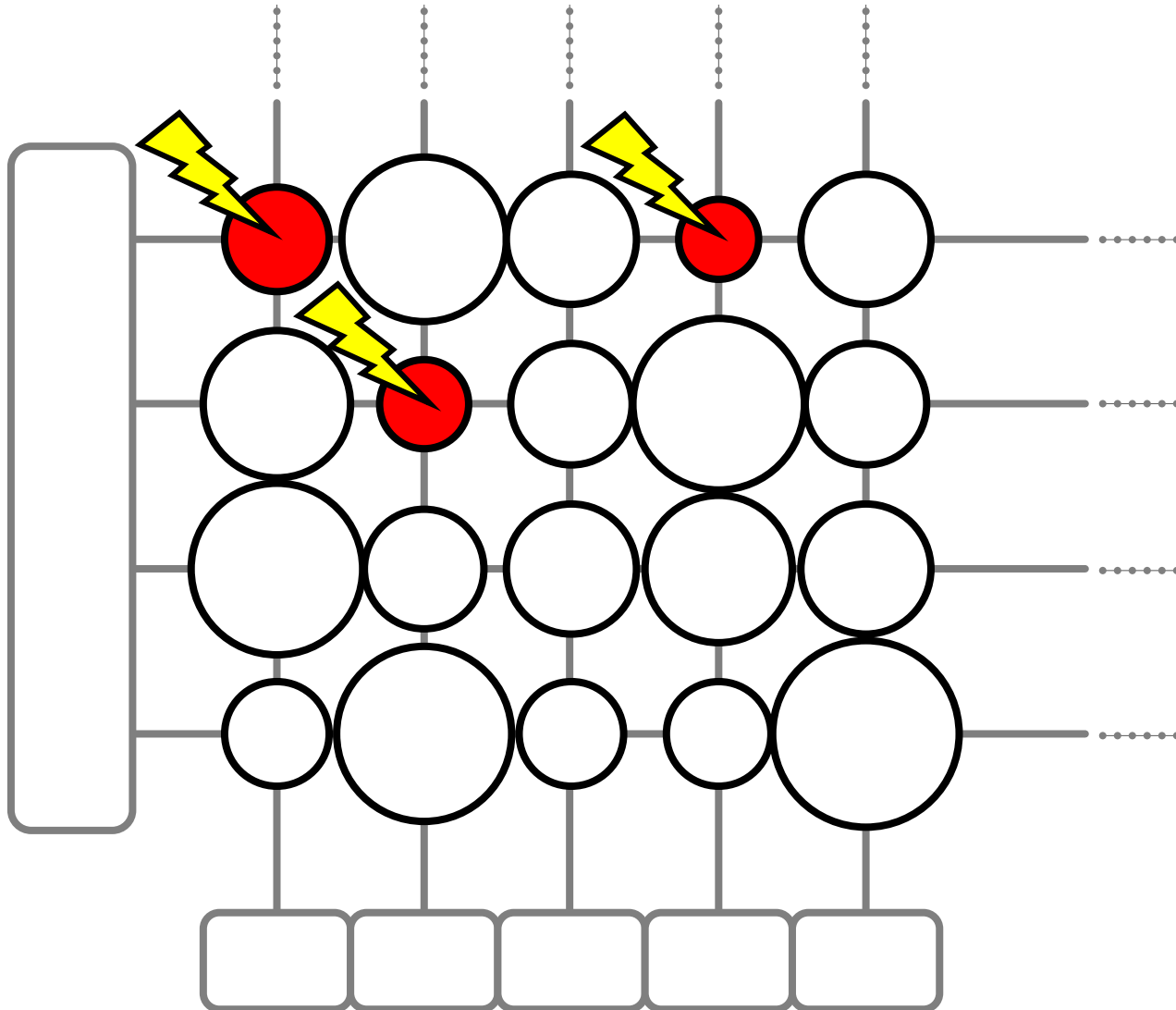
---

- Minesh Patel, Jeremie S. Kim, and Onur Mutlu,  
**"The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions"**  
*Proceedings of the 44th International Symposium on Computer Architecture (ISCA)*, Toronto, Canada, June 2017.  
[[Slides \(pptx\)](#)] [[pdf](#)]  
[[Lightning Session Slides \(pptx\)](#)] [[pdf](#)]
- First experimental analysis of (mobile) LPDDR4 chips
- Analyzes the complex tradeoff space of retention time profiling
- Idea: enable fast and robust profiling at higher refresh intervals & temperatures

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

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

**Goal:** find *all* retention failures for a refresh interval  $T >$  default (64ms)



**Process, voltage, temperature**

**Variable retention time**

**Data pattern dependence**

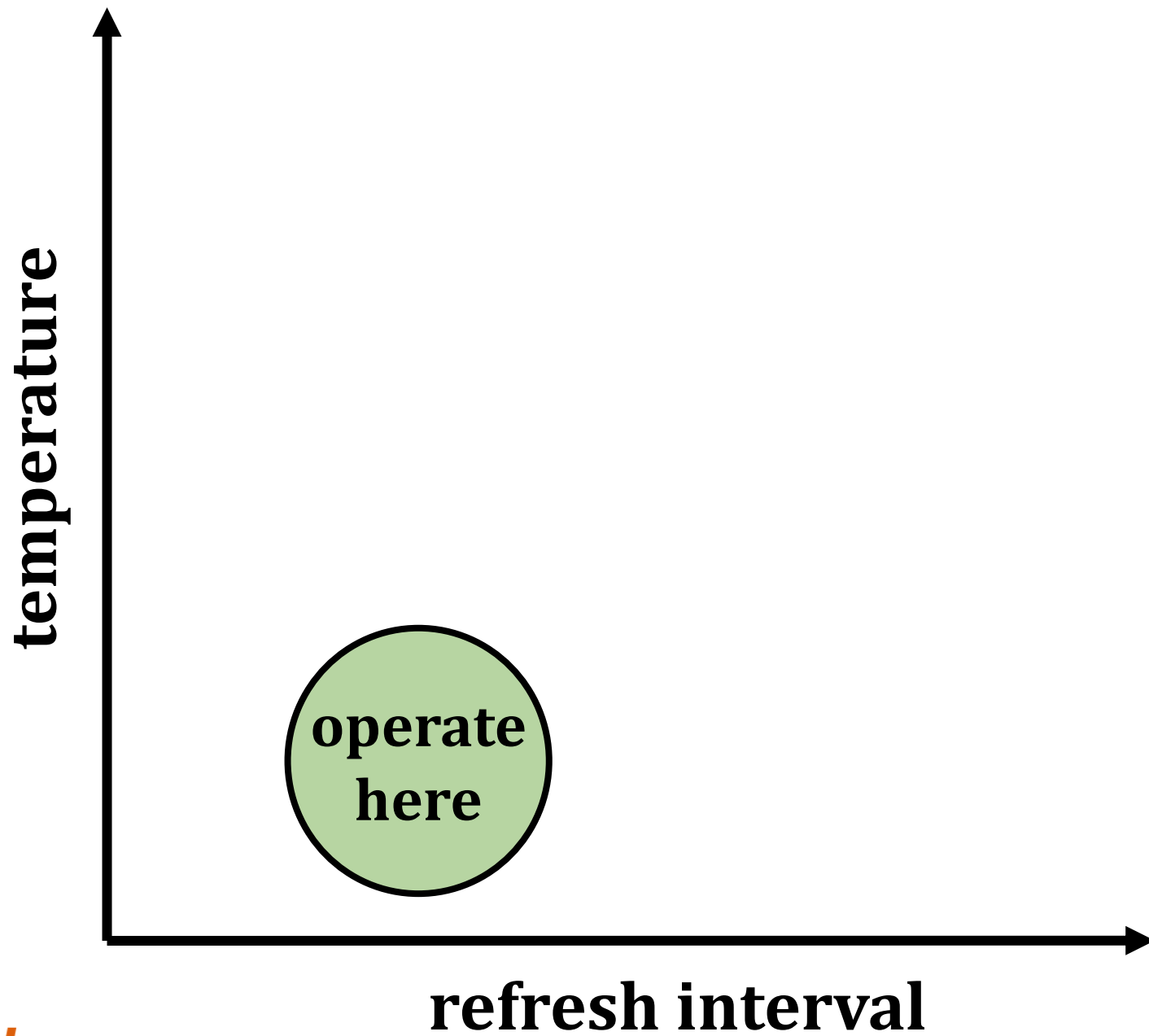
# Characterization of 368 LPDDR4 DRAM Chips

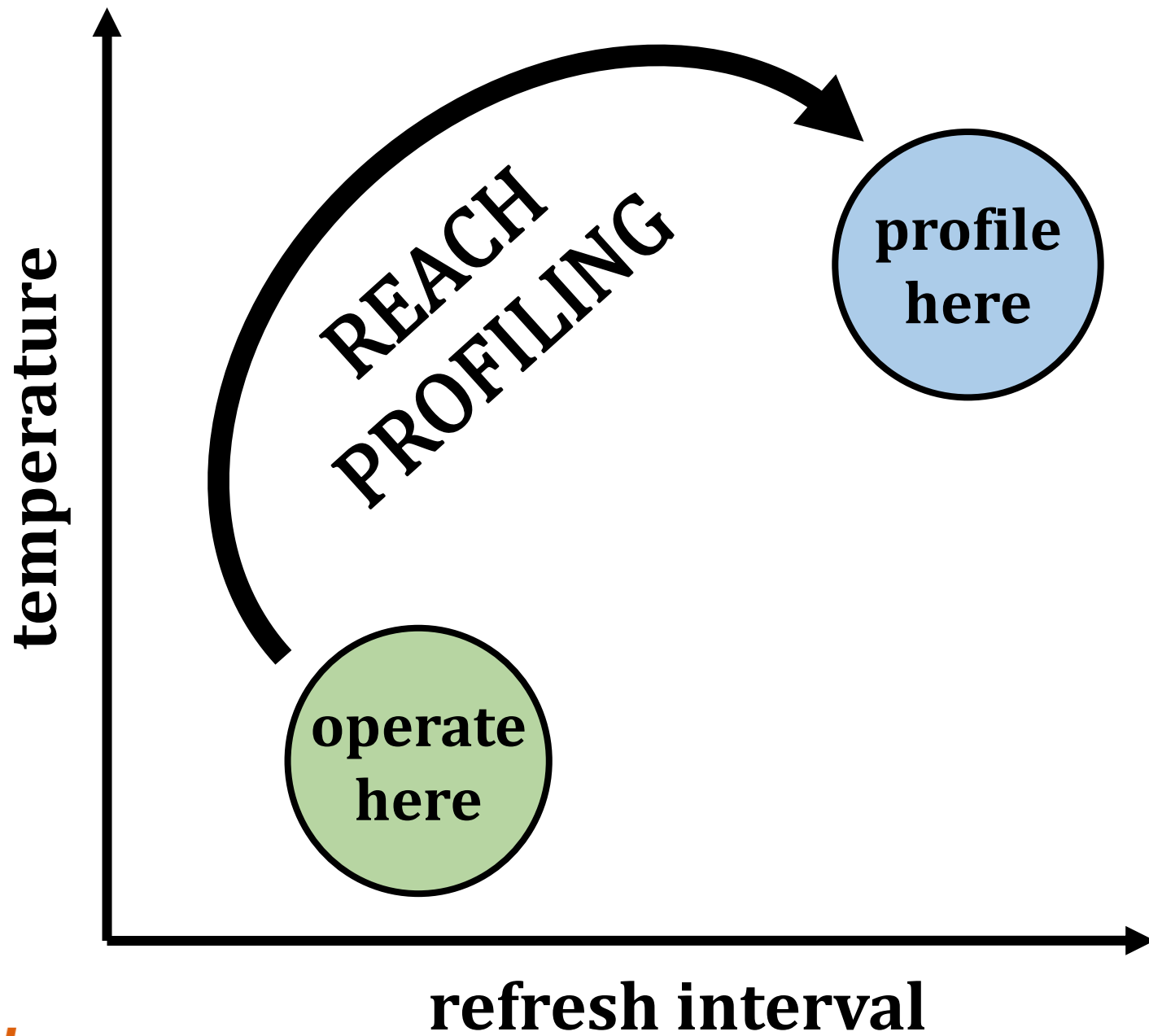
①

Cells are **more likely to fail** at an **increased (refresh interval | temperature)**

②

**Complex tradeoff space** between profiling  
**(speed & coverage & false positives)**







# Reach Profiling

**A new DRAM retention failure profiling methodology**

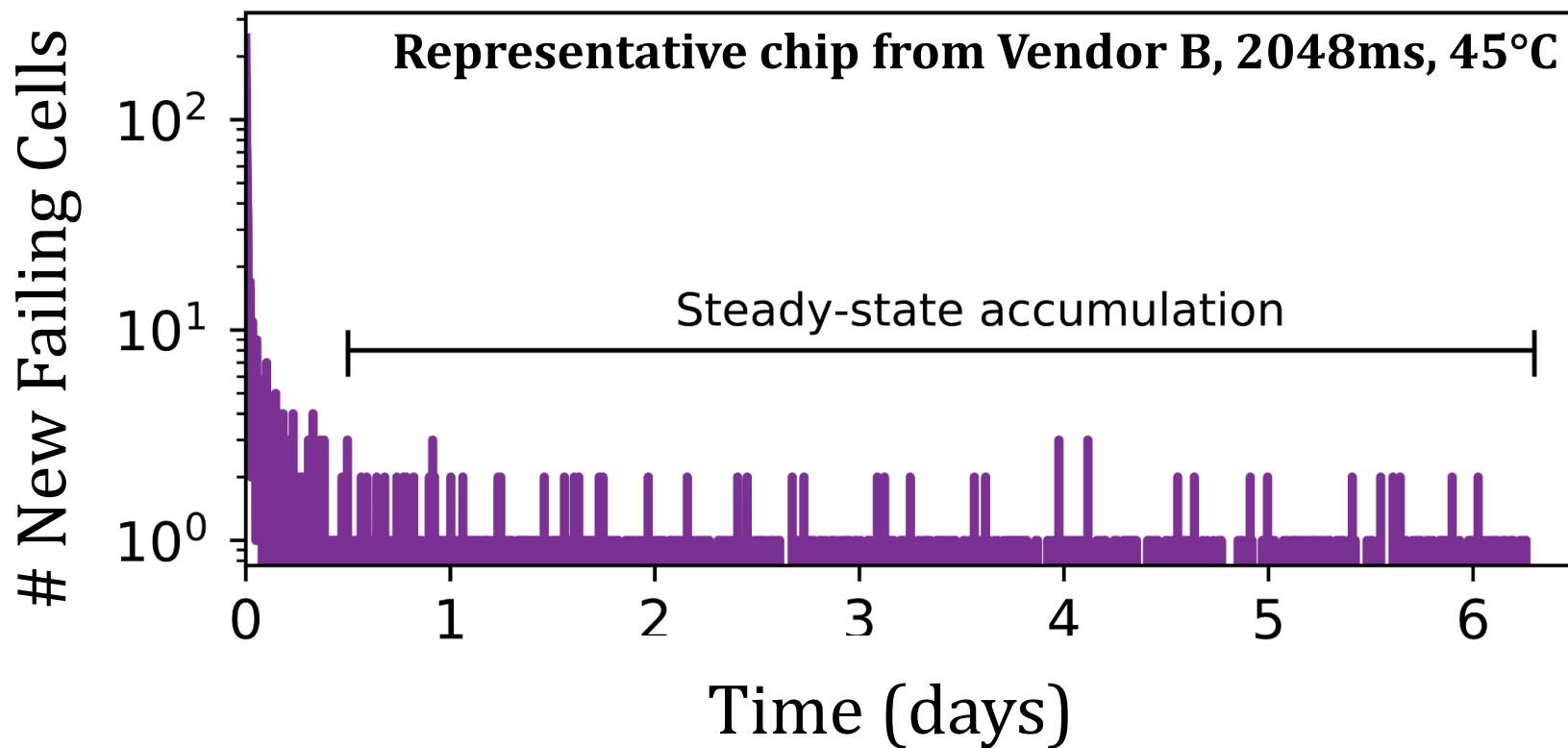
+ **Faster** and **more reliable**  
than current approaches

+ Enables **longer refresh intervals**

# LPDDR4 Studies

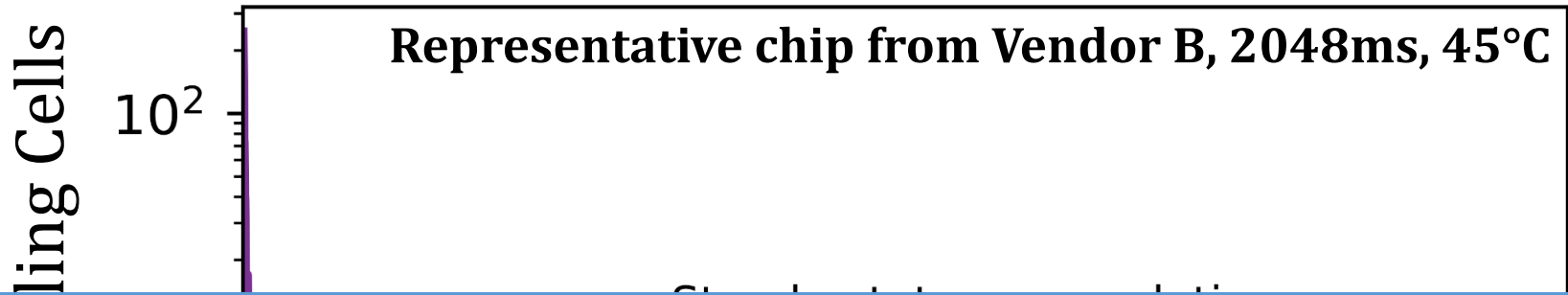
1. Temperature
2. Data Pattern Dependence
3. Retention Time Distributions
- 4. Variable Retention Time**
- 5. Individual Cell Characterization**

# Long-term Continuous Profiling



- New failing cells continue to appear over time
  - Attributed to **variable retention time (VRT)**
- The set of failing cells changes over time

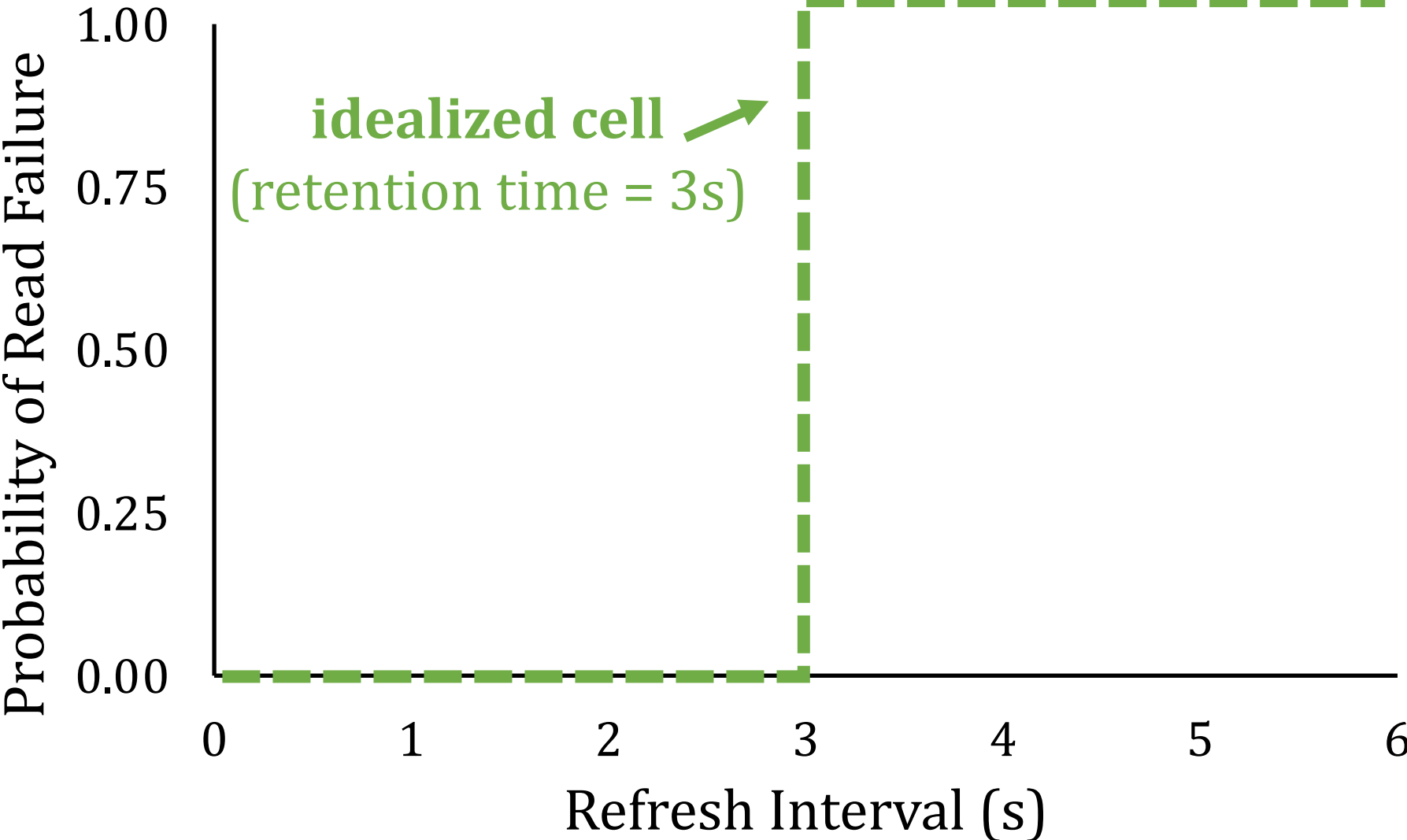
# Long-term Continuous Profiling



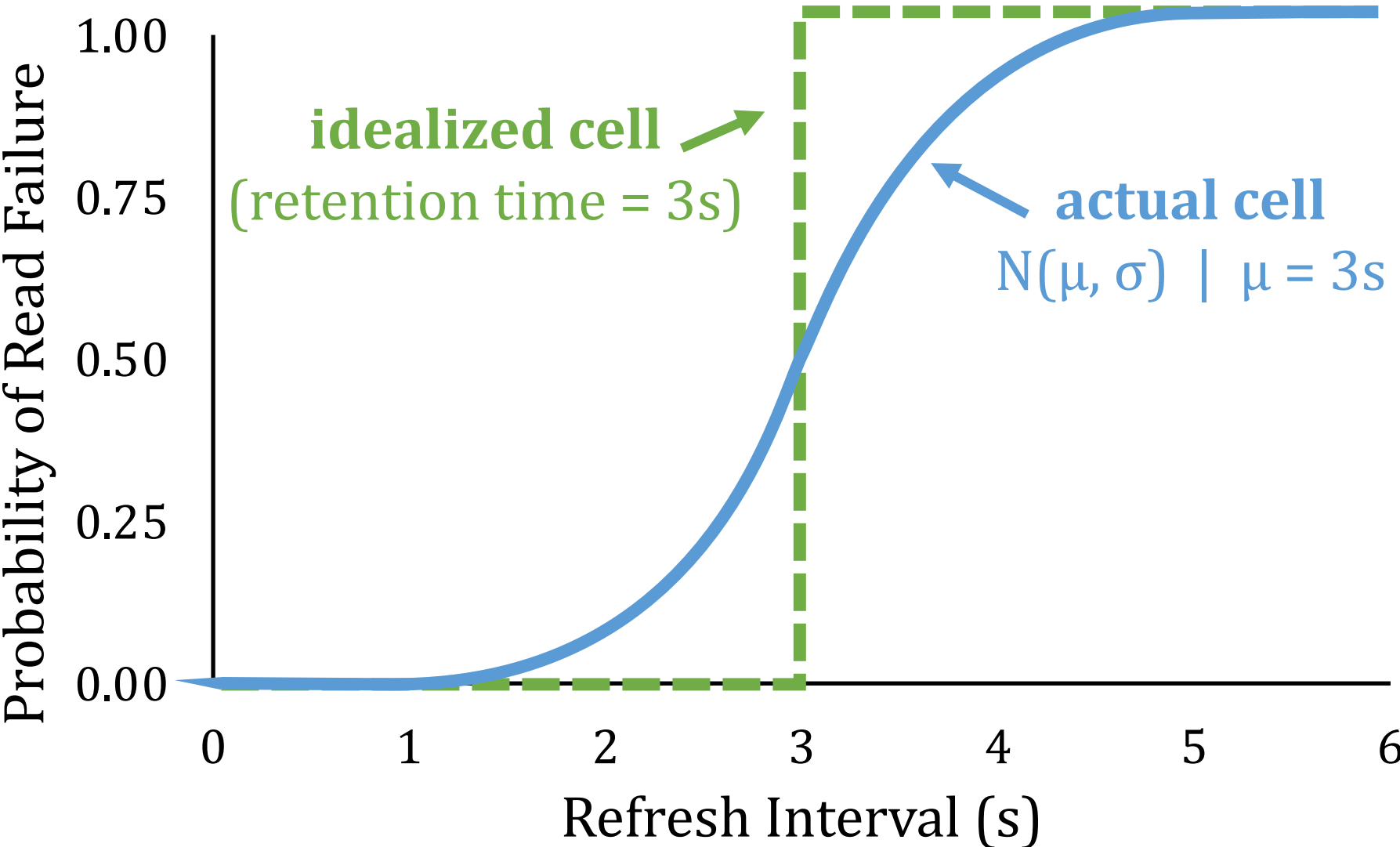
**Error correction codes (ECC)  
and online profiling are *necessary*  
to manage new failing cells**

- New failing cells continue to appear over time
  - Attributed to **variable retention time (VRT)**
- The set of failing cells changes over time

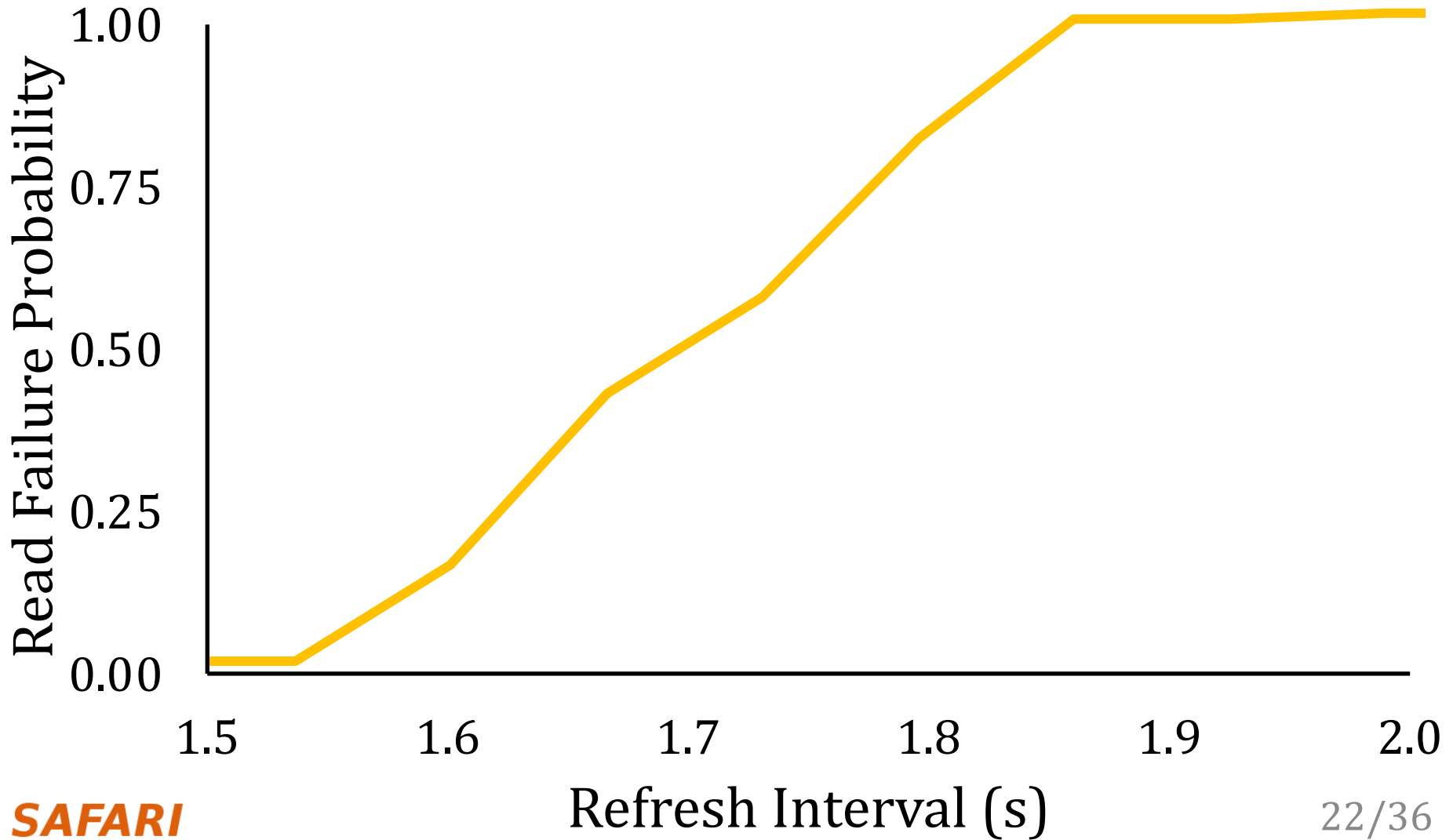
# Single-cell Failure Probability (Cartoon)



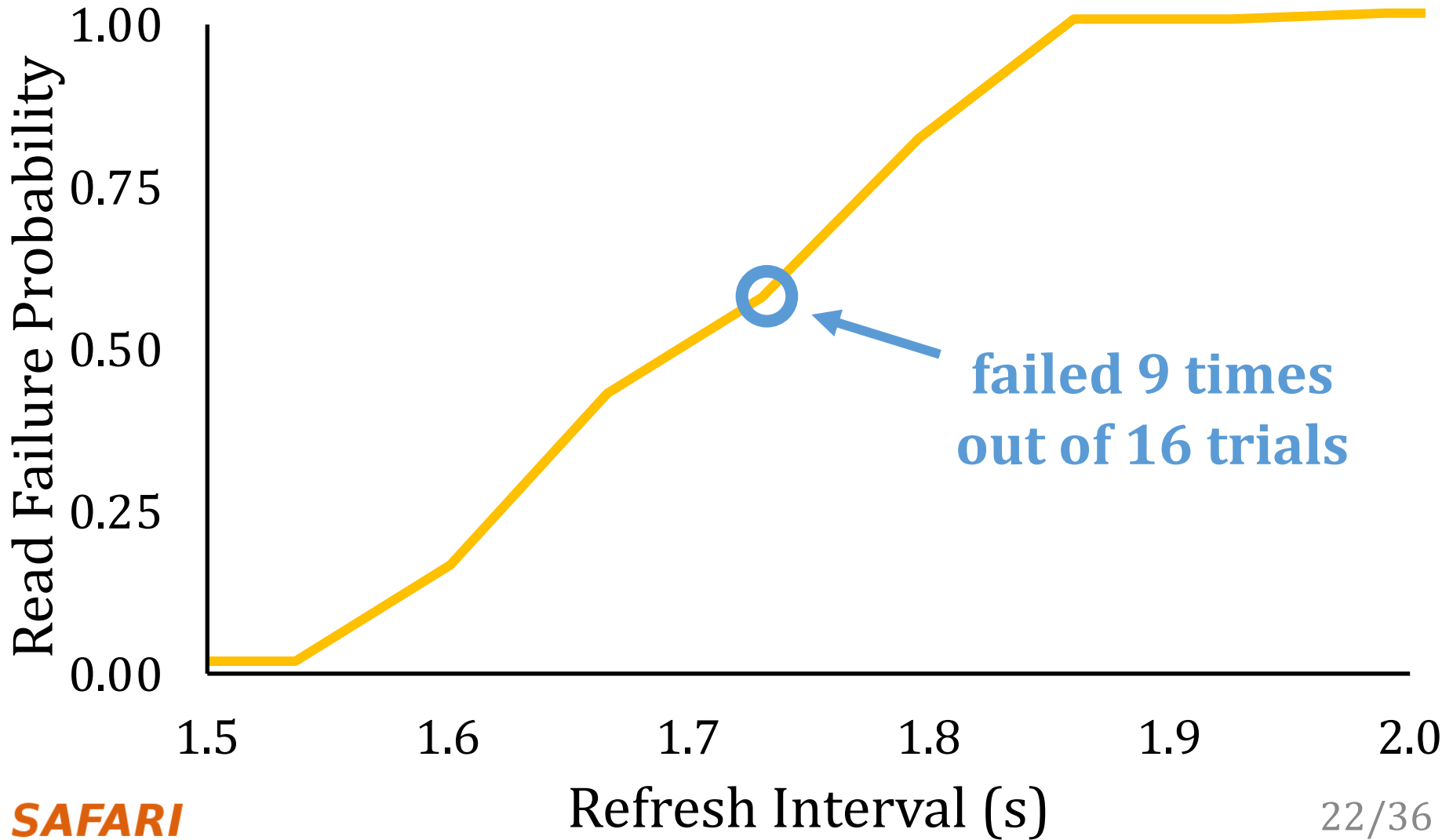
# Single-cell Failure Probability (Cartoon)



# Single-cell Failure Probability (Real)

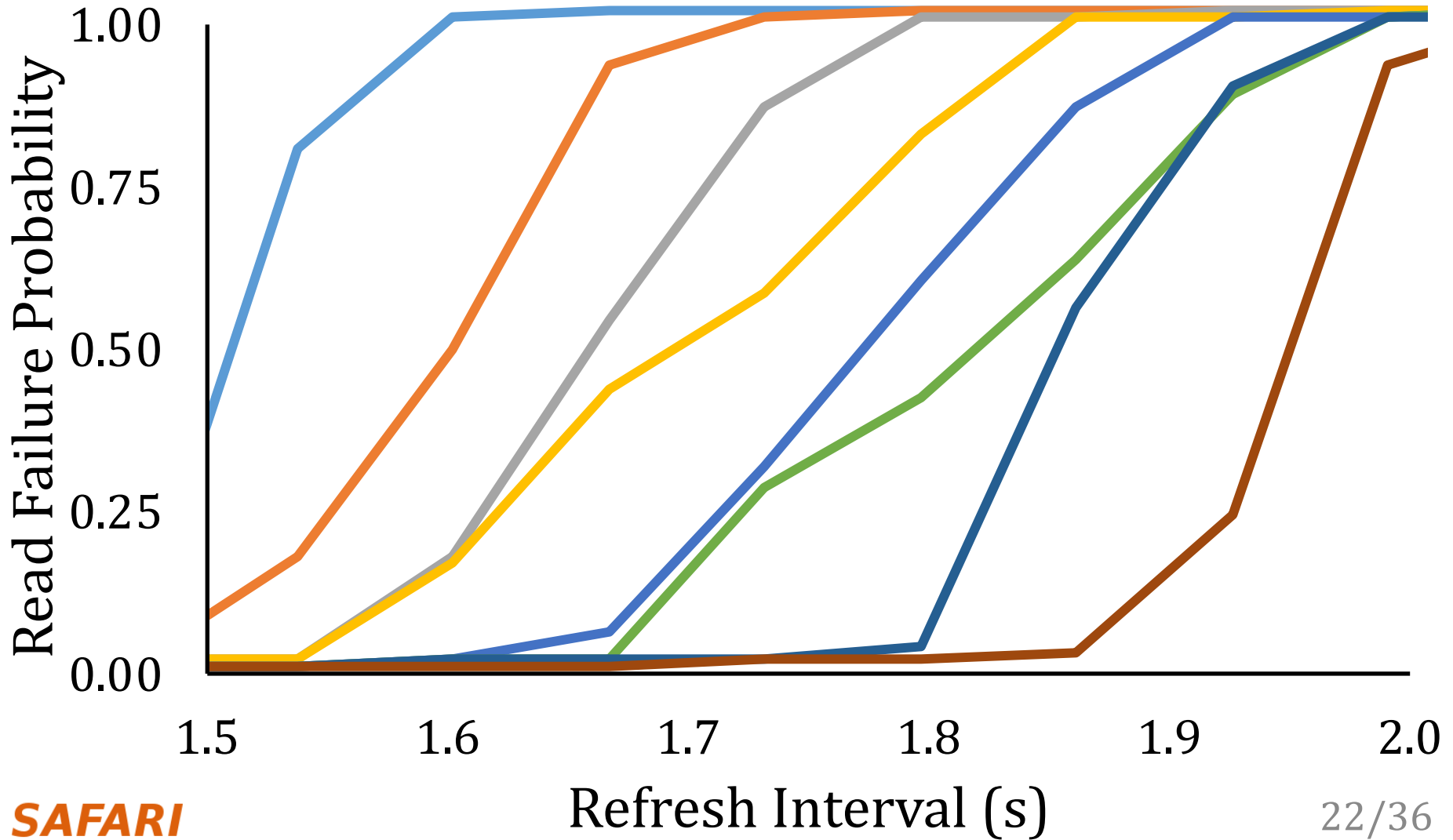


# Single-cell Failure Probability (Real)

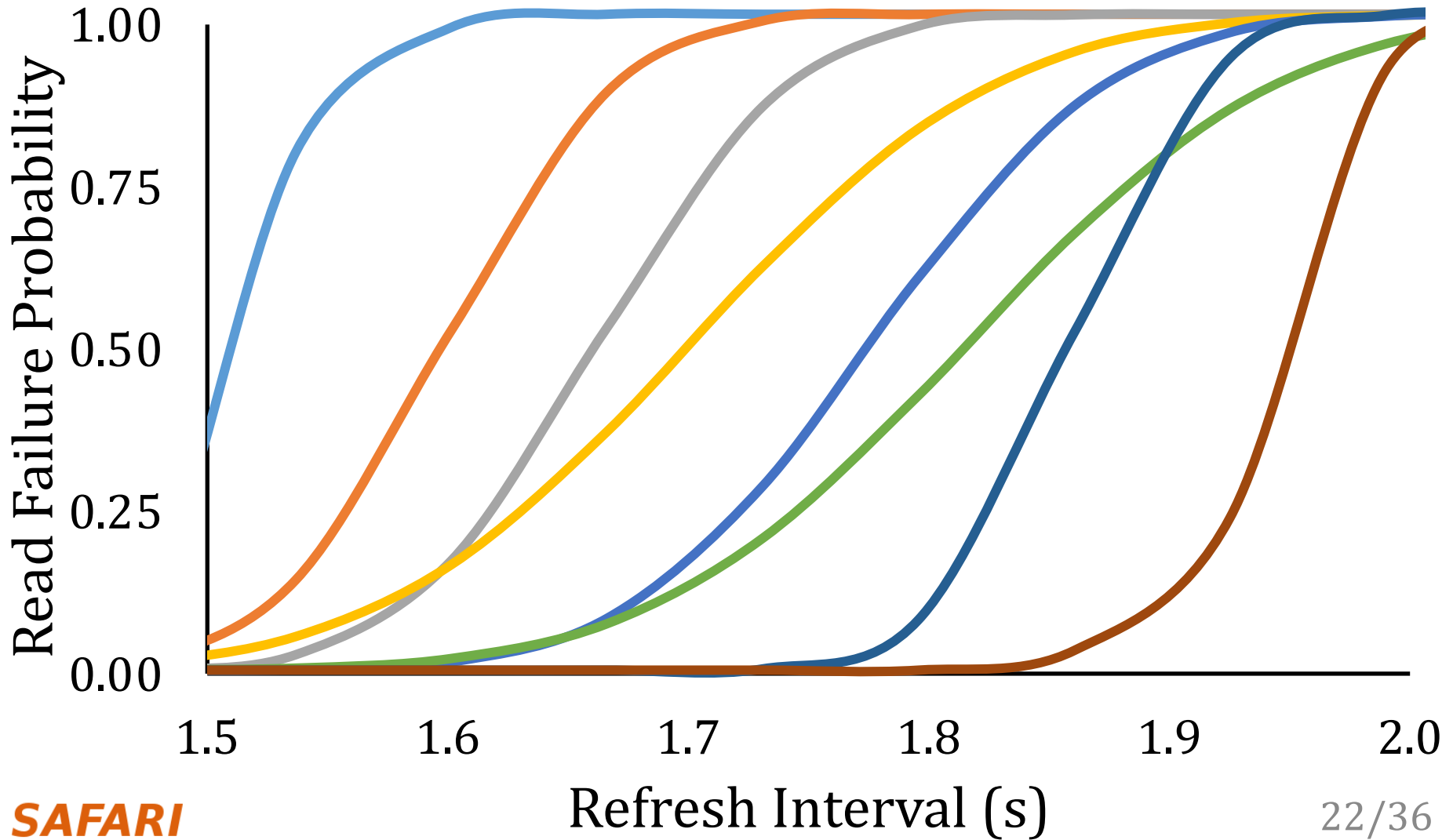




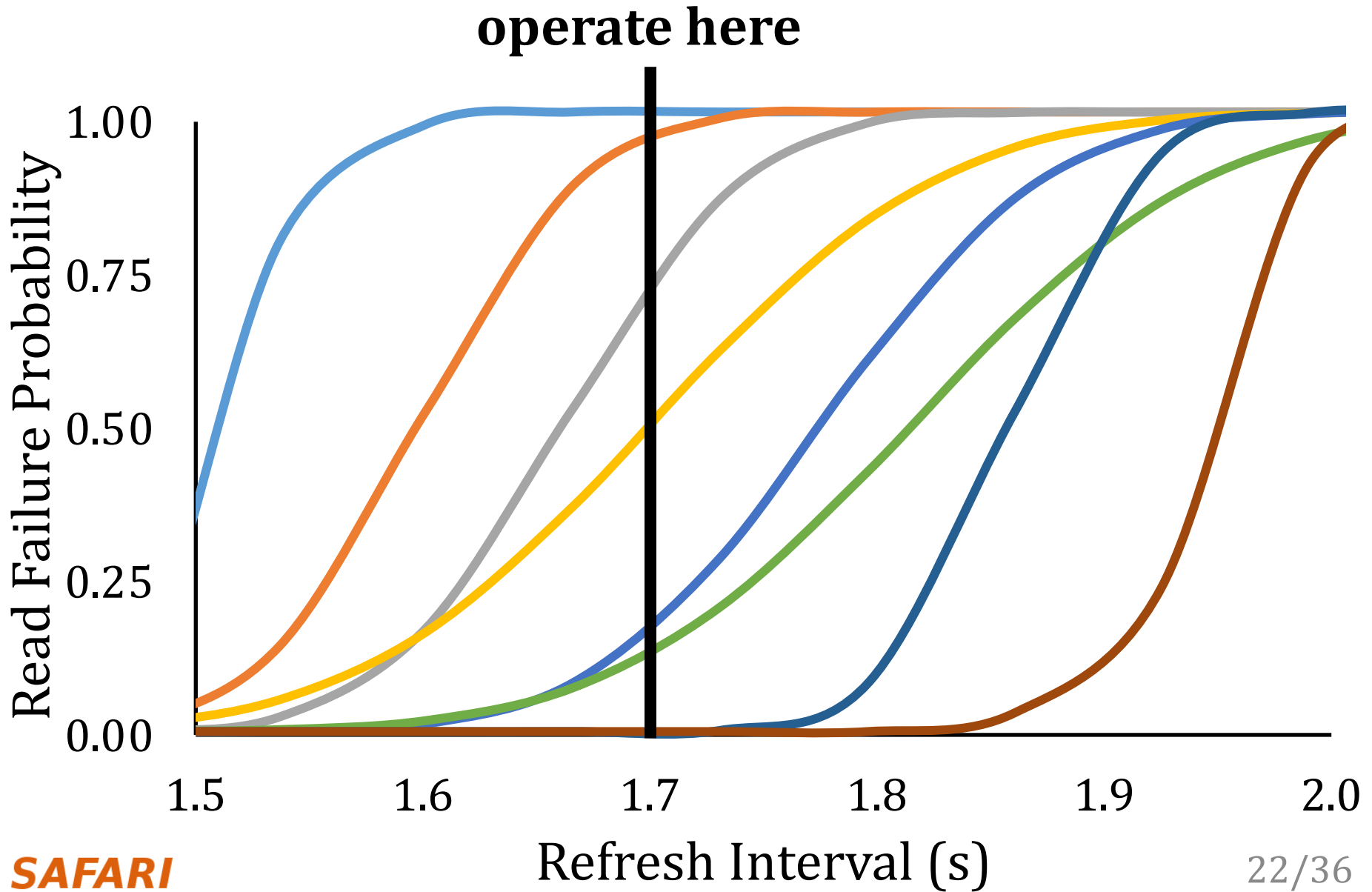
# Single-cell Failure Probability (Real)



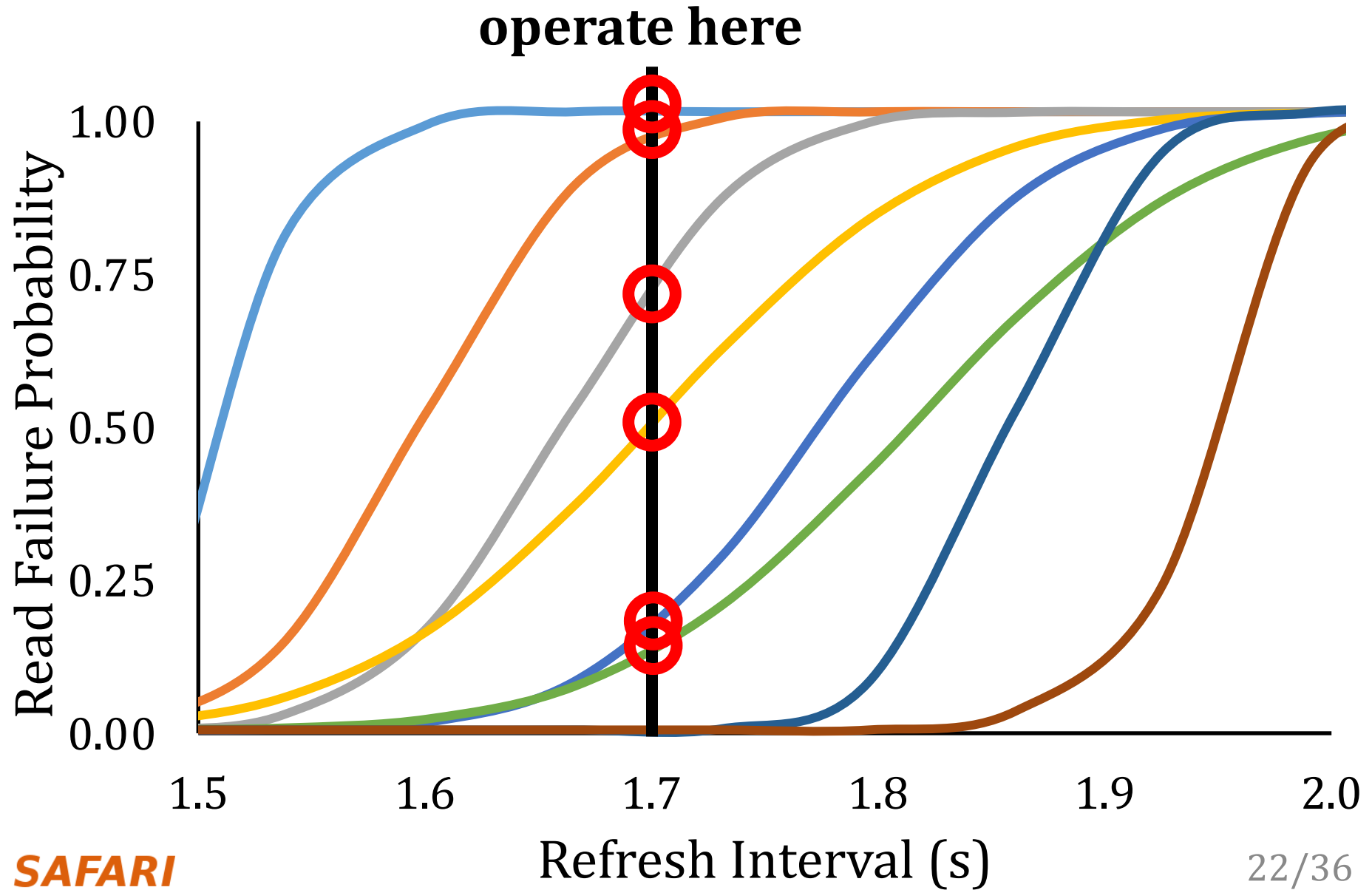
# Single-cell Failure Probability (Real)



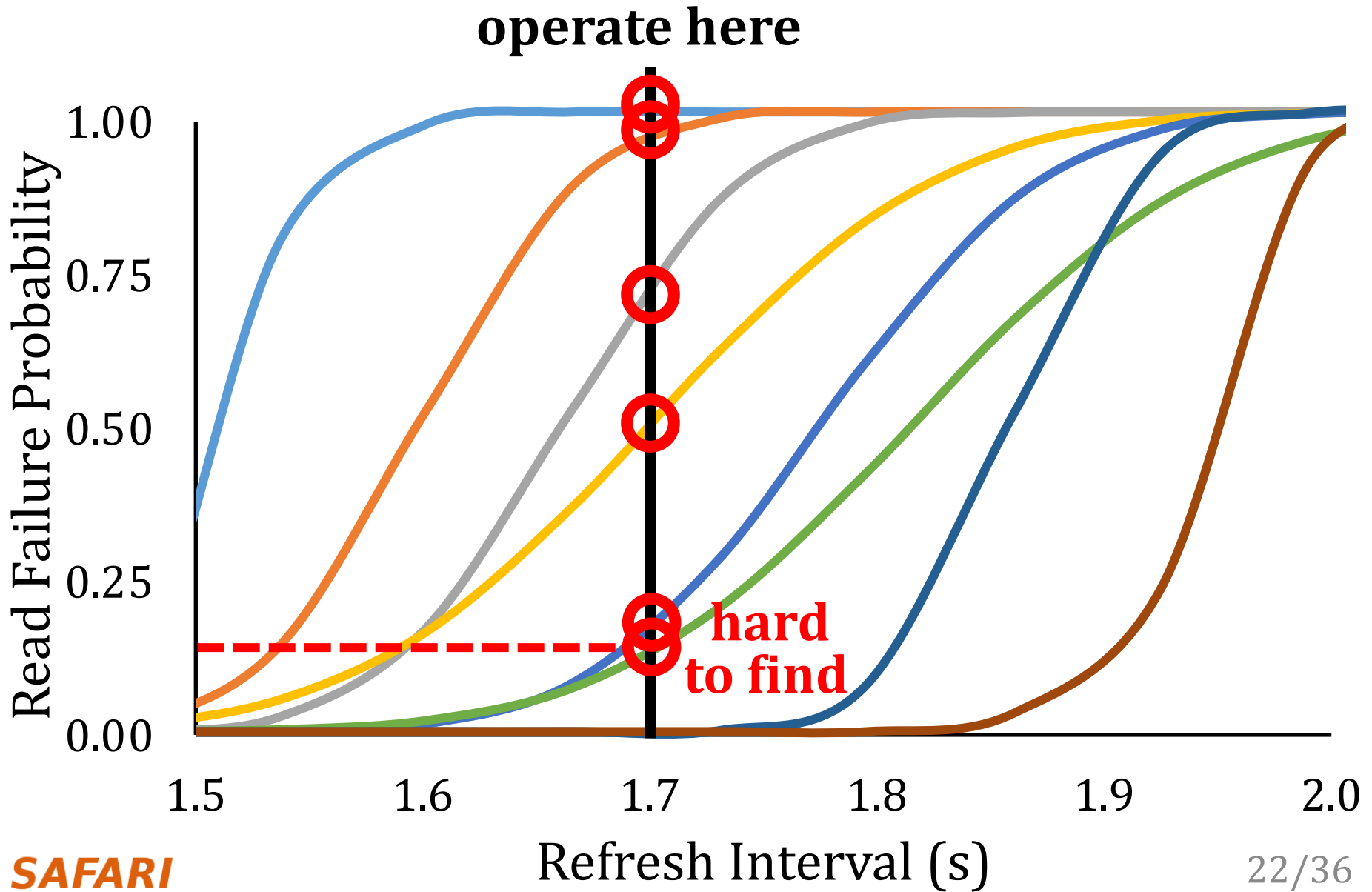
# Single-cell Failure Probability (Real)



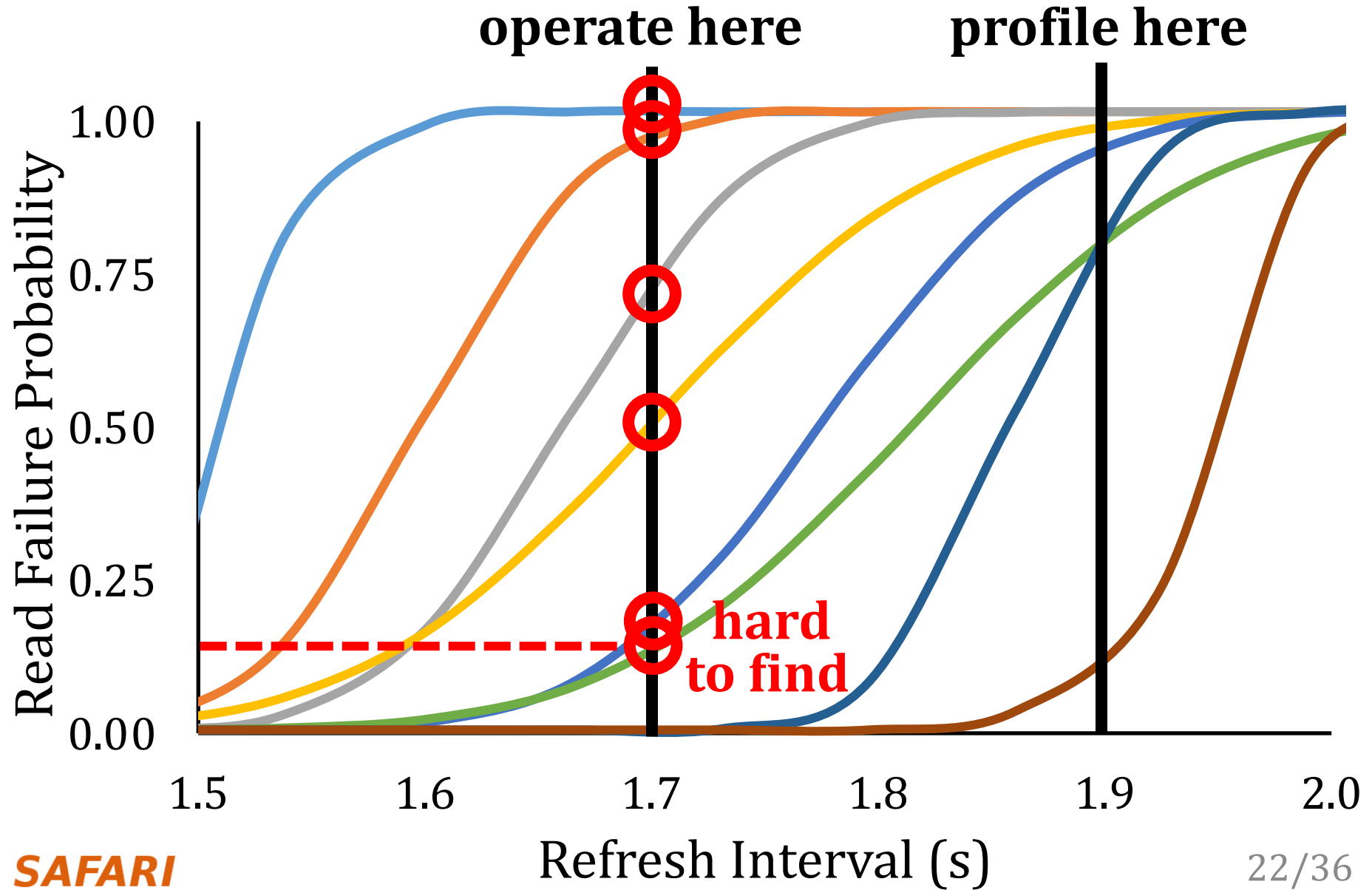
# Single-cell Failure Probability (Real)



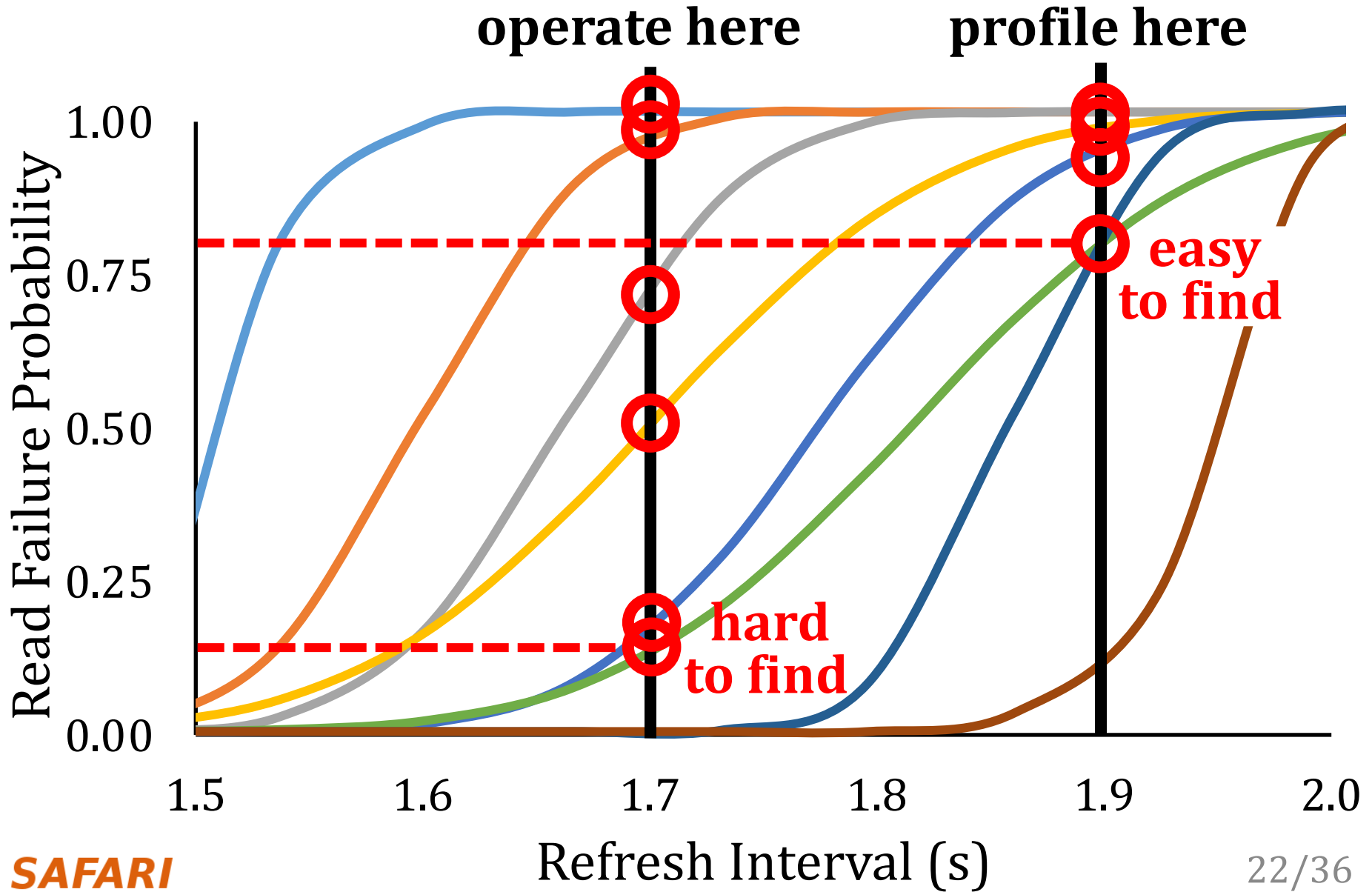
# Single-cell Failure Probability (Real)



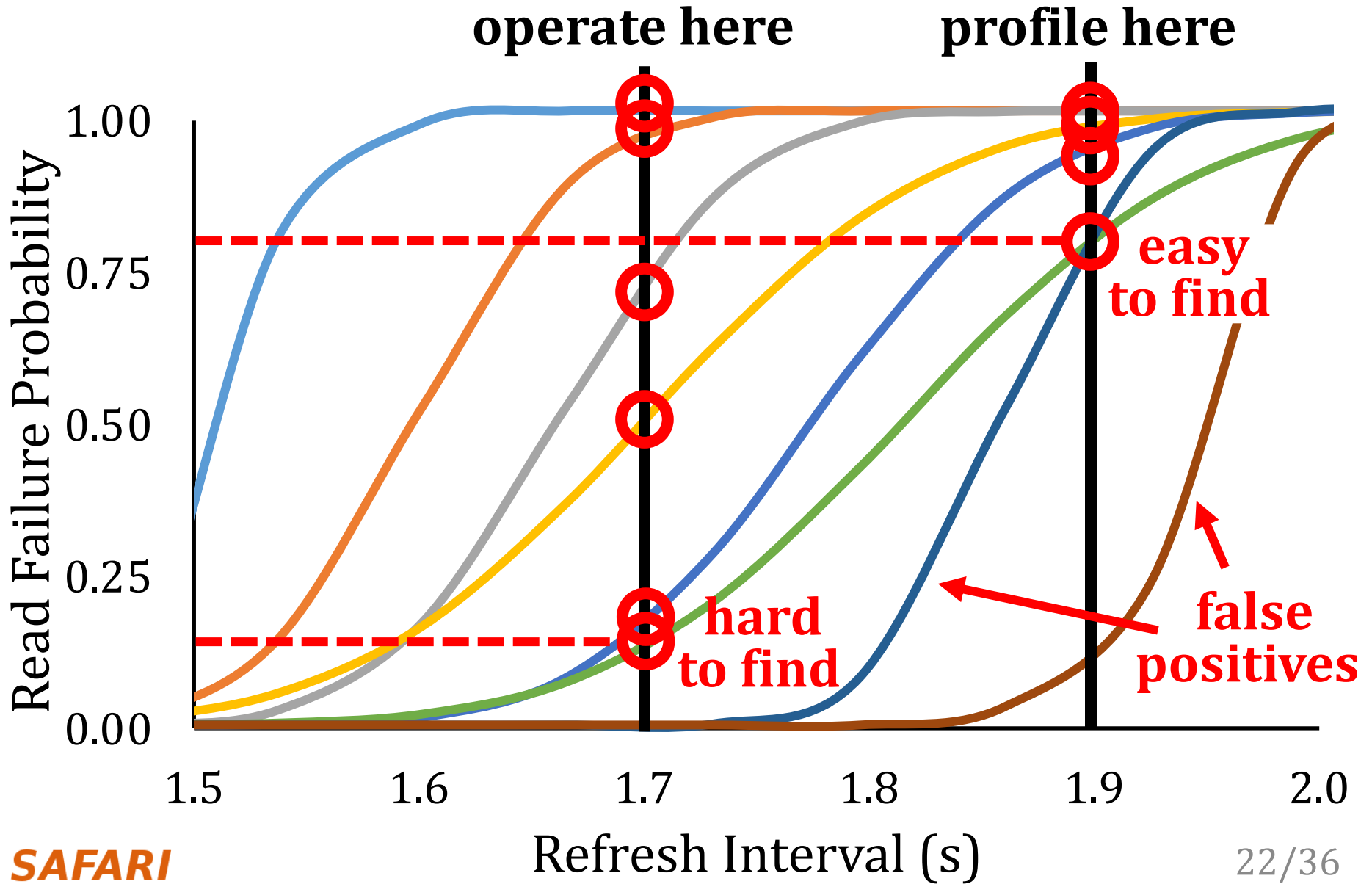
# Single-cell Failure Probability (Real)



# Single-cell Failure Probability (Real)

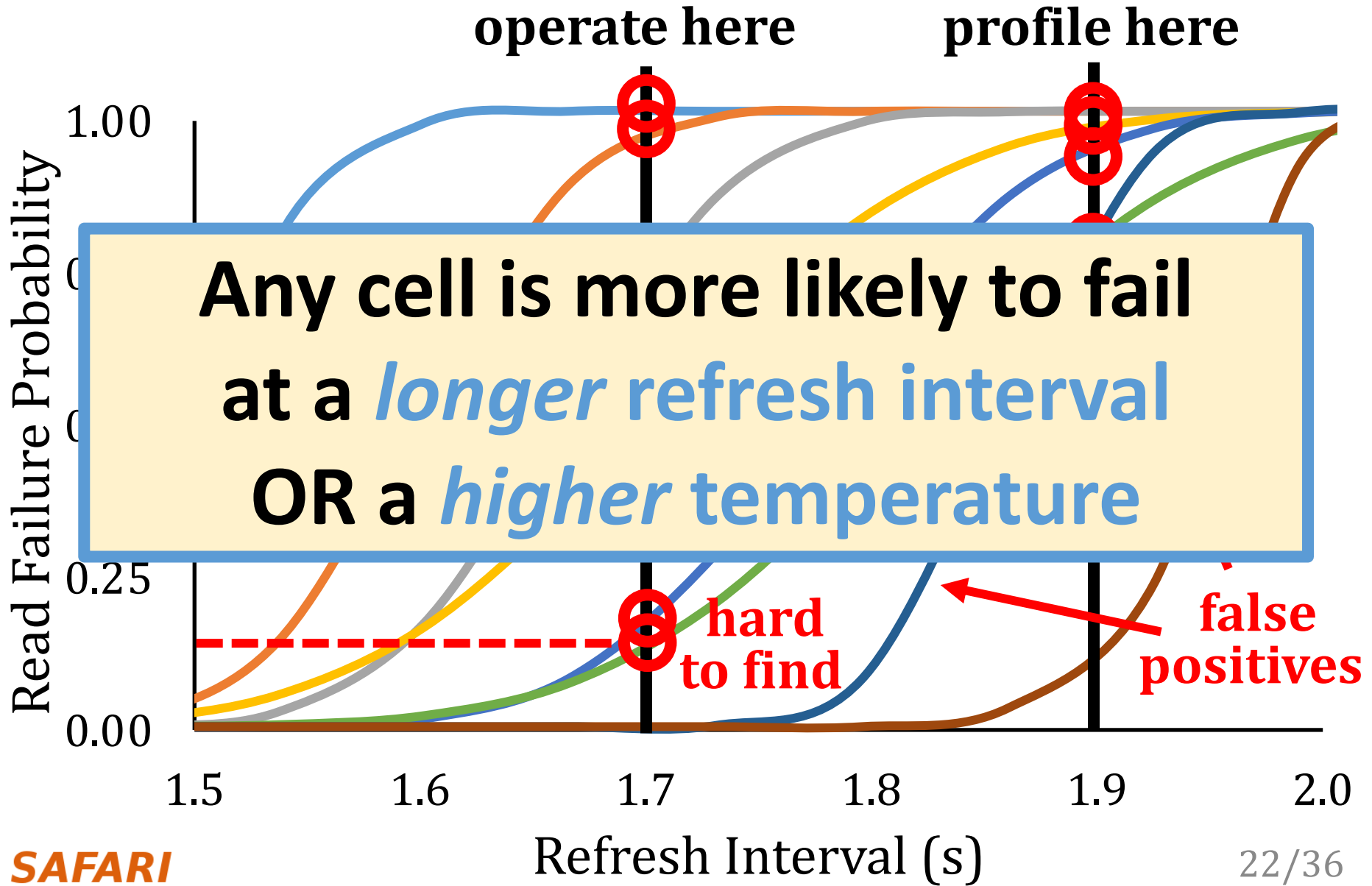


# Single-cell Failure Probability (Real)





# Single-cell Failure Probability (Real)



# Reach Profiling

**Key idea:** profile at a *longer refresh interval* and/or a *higher temperature*

- **Pros**

- **Fast + Reliable:** reach profiling searches for cells *where they are most likely to fail*

- **Cons**

- **False Positives:** profiler may identify cells that fail under profiling conditions, but not under operating conditions

# Towards an Implementation

Reach profiling is a **general methodology**

3 key questions for an implementation:

**What are desirable profiling conditions?**

**How often should the system profile?**

**What information does the profiler need?**

# Three Key Profiling Metrics

- 1. Runtime:** how long profiling takes
- 2. Coverage:** portion of all possible failures discovered by profiling
- 3. False positives:** number of cells observed to fail during profiling but never during actual operation

# Three Key Profiling Metrics

- 1. Runtime:** how long profiling takes
- 2. Coverage:** portion of all possible failures discovered by profiling

We explore how these three metrics change under **many** different profiling conditions

# Simulated End-to-end Performance

 Brute-force profiling    REAPER    Ideal profiling

**On average, REAPER enables:**

**16.3% system performance improvement**

**36.4% DRAM power reduction**



**REAPER enables longer refresh intervals, which are unreasonable using brute-force profiling**

refresh intervals

rarely

refresh intervals

often

# REAPER Summary

## Problem:

- DRAM refresh performance and energy overhead is high
- Current approaches to retention failure profiling are slow or unreliable

## Goals:

1. Thoroughly analyze profiling tradeoffs
2. Develop a **fast** and **reliable** profiling mechanism

## Key Contributions:

1. **First** detailed characterization of 368 LPDDR4 DRAM chips
2. **Reach profiling:** Profile at a **longer refresh interval** or **higher temperature** than target conditions, where cells are more likely to fail

## Evaluation:

- **2.5x** faster profiling with **99%** coverage and **50%** false positives
- REAPER enables **16.3% system performance improvement** and **36.4% DRAM power reduction**
- Enables longer refresh intervals that were previously unreasonable

## Main Memory Needs Intelligent Controllers for Reliability & Security



# Understanding In-DRAM ECC

---

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

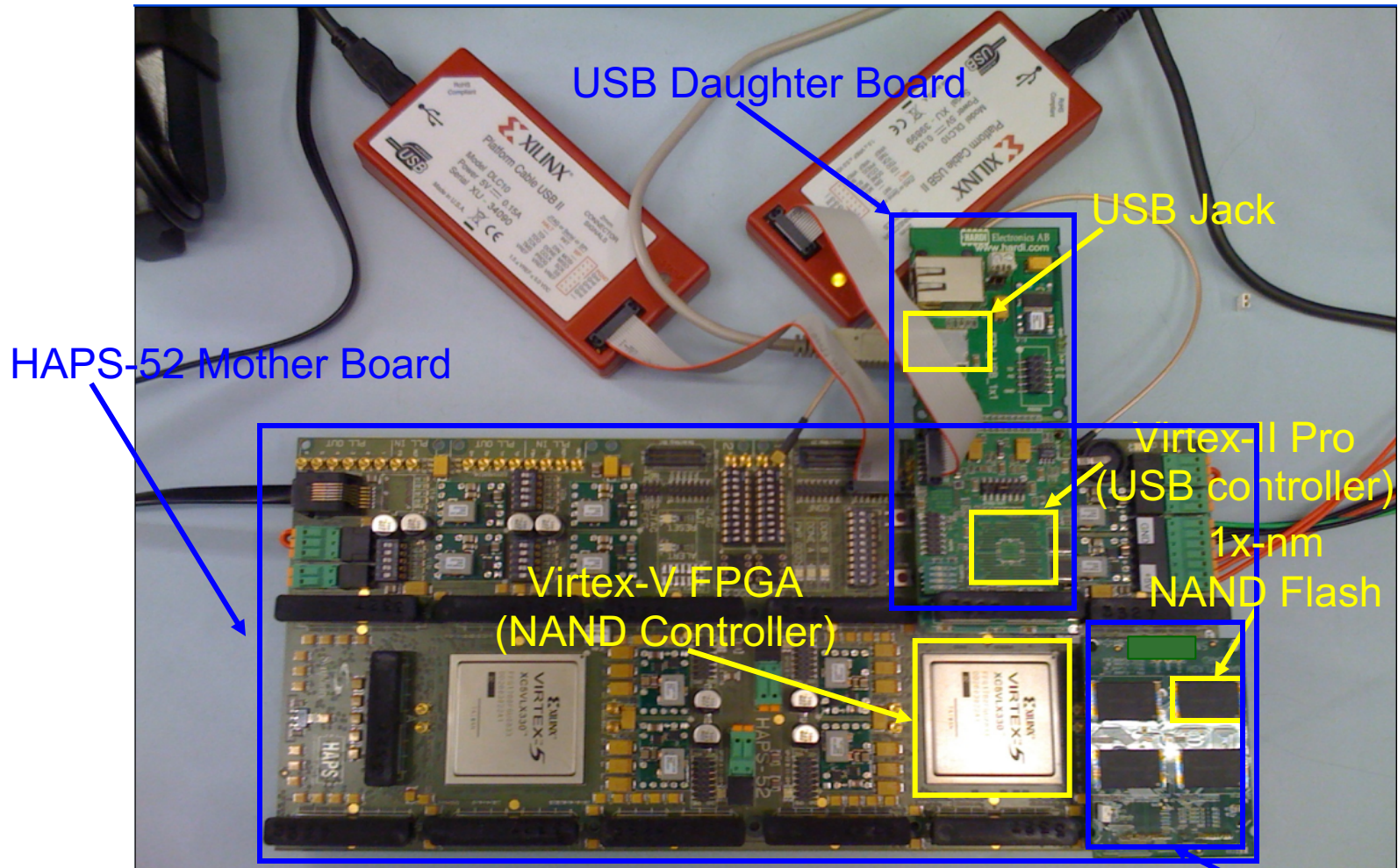
## Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices

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

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

# Understanding Flash Memory Vulnerabilities

# Understand and Model with Experiments (Flash)



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

NAND Daughter Board

# Understanding Flash Memory Reliability

---



*Proceedings of the IEEE, Sept. 2017*

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

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

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

# Understanding Flash Memory Reliability

---

- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu,  
**"A Large-Scale Study of Flash Memory Errors in the Field"**  
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Portland, OR, June 2015.  
[[Slides \(pptx\)](#)] [[pdf](#)] [[Coverage at ZDNet](#)] [[Coverage on The Register](#)]  
[[Coverage on TechSpot](#)] [[Coverage on The Tech Report](#)]

## A Large-Scale Study of Flash Memory Failures in the Field

Justin Meza  
Carnegie Mellon University  
meza@cmu.edu

Qiang Wu  
Facebook, Inc.  
qwu@fb.com

Sanjeev Kumar  
Facebook, Inc.  
skumar@fb.com

Onur Mutlu  
Carnegie Mellon University  
onur@cmu.edu

# NAND Flash Vulnerabilities [HPCA'17]

*HPCA, Feb. 2017*

## Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

Yu Cai<sup>†</sup>   Saugata Ghose<sup>†</sup>   Yixin Luo<sup>††</sup>   Ken Mai<sup>†</sup>   Onur Mutlu<sup>§†</sup>   Erich F. Haratsch<sup>‡</sup>  
<sup>†</sup>Carnegie Mellon University   <sup>‡</sup>Seagate Technology   <sup>§</sup>ETH Zürich

*Modern NAND flash memory chips provide high density by storing two bits of data in each flash cell, called a multi-level cell (MLC). An MLC partitions the threshold voltage range of a flash cell into four voltage states. When a flash cell is programmed, a high voltage is applied to the cell. Due to parasitic capacitance coupling between flash cells that are physically close to each other, flash cell programming can lead to cell-to-cell program interference, which introduces errors into neighboring flash cells. In order to reduce the impact of cell-to-cell interference on the reliability of MLC NAND flash memory, flash manufacturers adopt a two-step programming method, which programs the MLC in two separate steps. First, the flash memory partially programs the least significant bit of the MLC to some intermediate threshold voltage. Second, it programs the most significant bit to bring the MLC up to its full voltage state.*

*In this paper, we demonstrate that two-step programming exposes new reliability and security vulnerabilities. We expe-*

*belongs to a different flash memory page (the unit of data programmed and read at the same time), which we refer to, respectively, as the least significant bit (LSB) page and the most significant bit (MSB) page [5].*

*A flash cell is programmed by applying a large voltage on the control gate of the transistor, which triggers charge transfer into the floating gate, thereby increasing the threshold voltage. To precisely control the threshold voltage of the cell, the flash memory uses *incremental step pulse programming* (ISPP) [12, 21, 25, 41]. ISPP applies multiple short pulses of the programming voltage to the control gate, in order to increase the cell threshold voltage by some small voltage amount ( $V_{step}$ ) after each step. Initial MLC designs programmed the threshold voltage in *one shot*, issuing all of the pulses back-to-back to program *both* bits of data at the same time. However, as flash memory scales down, the distance between neighboring flash cells decreases, which*

[https://people.inf.ethz.ch/omutlu/pub/flash-memory-programming-vulnerabilities\\_hpca17.pdf](https://people.inf.ethz.ch/omutlu/pub/flash-memory-programming-vulnerabilities_hpca17.pdf)

# 3D NAND Flash Reliability I [HPCA'18]

---

- Yixin Luo, Saugata Ghose, Yu Cai, Erich F. Haratsch, and Onur Mutlu, **"HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature-Awareness"**

*Proceedings of the 24th International Symposium on High-Performance Computer Architecture (HPCA)*, Vienna, Austria, February 2018.

[[Lightning Talk Video](#)]

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

## HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature Awareness

Yixin Luo<sup>†</sup>

Saugata Ghose<sup>†</sup>

Yu Cai<sup>‡</sup>

Erich F. Haratsch<sup>‡</sup>

Onur Mutlu<sup>§†</sup>

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

<sup>‡</sup>*Seagate Technology*

<sup>§</sup>*ETH Zürich*

# 3D NAND Flash Reliability II [SIGMETRICS'18]

---

- Yixin Luo, Saugata Ghose, Yu Cai, Erich F. Haratsch, and Onur Mutlu,  
**"Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation"**

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

[[Abstract](#)]

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

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

## **Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation**

Yixin Luo<sup>†</sup>

Saugata Ghose<sup>†</sup>

Yu Cai<sup>†</sup>

Erich F. Haratsch<sup>‡</sup>

Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>Seagate Technology

<sup>§</sup>ETH Zürich



# Another Talk: NAND Flash Memory Robustness

---

- Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu, **"Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives"**

*to appear in Proceedings of the IEEE, 2017.*

Cai+, "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis," DATE 2012.

Cai+, "Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime," ICCD 2012.

Cai+, "Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling," DATE 2013.

Cai+, "Error Analysis and Retention-Aware Error Management for NAND Flash Memory," Intel Technology Journal 2013.

Cai+, "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation," ICCD 2013.

Cai+, "Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories," SIGMETRICS 2014.

Cai+, "Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery," HPCA 2015.

Cai+, "Read Disturb Errors in MLC NAND Flash Memory: Characterization and Mitigation," DSN 2015.

Luo+, "WARM: Improving NAND Flash Memory Lifetime with Write-hotness Aware Retention Management," MSST 2015.

Meza+, "A Large-Scale Study of Flash Memory Errors in the Field," SIGMETRICS 2015.

Luo+, "Enabling Accurate and Practical Online Flash Channel Modeling for Modern MLC NAND Flash Memory," IEEE JSAC 2016.

Cai+, "Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques," HPCA 2017.

Fukami+, "Improving the Reliability of Chip-Off Forensic Analysis of NAND Flash Memory Devices," DFRWS EU 2017.

Luo+, "HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature-Awareness," HPCA 2018.

Luo+, "Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation," SIGMETRICS 2018.

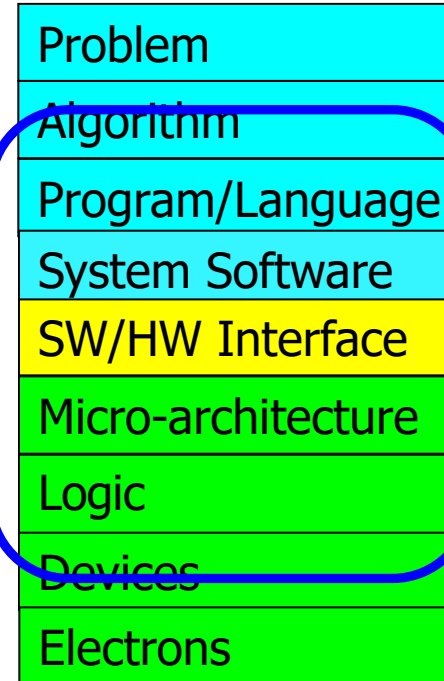
---

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

# Two Other Solution Directions

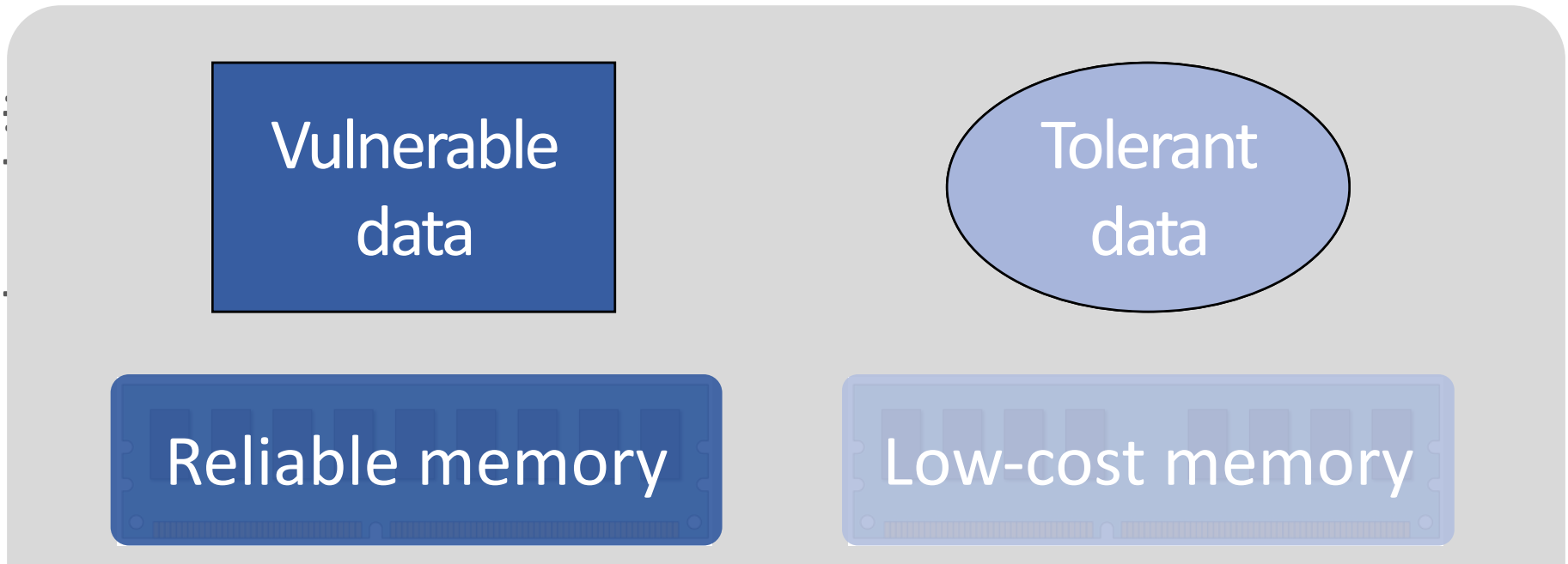
# There are Two Other Solution Directions

- **New Technologies:** Replace or (more likely) augment DRAM with a different technology
  - Non-volatile memories
- **Embracing Un-reliability:**  
Design memories with different reliability and store data intelligently across them  
**[Luo+ DSN 2014]**
- ...



**Fundamental solutions to security  
require co-design across the hierarchy**

# Exploiting Memory Error Tolerance with Hybrid Memory Systems



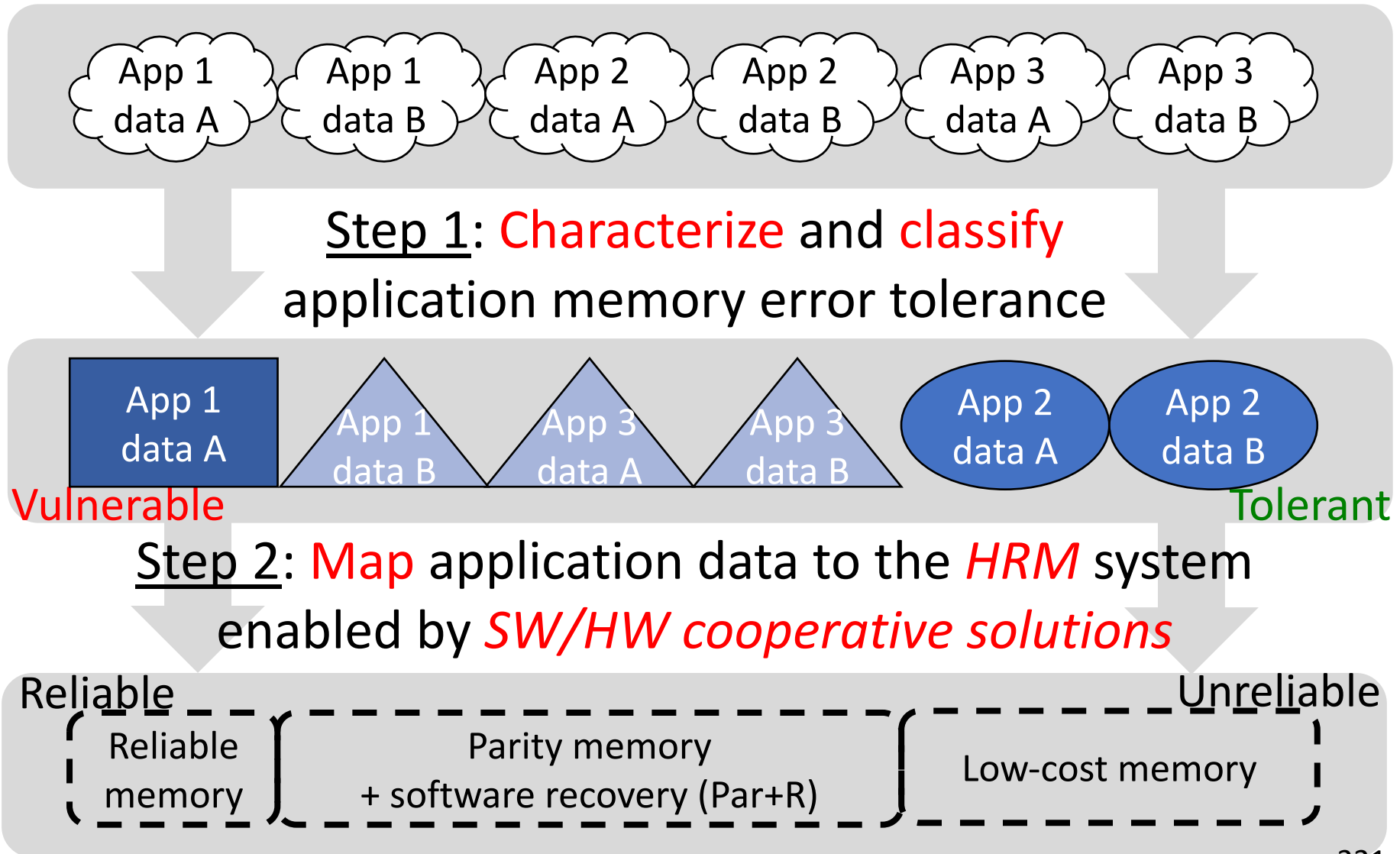
On Microsoft's Web Search workload

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

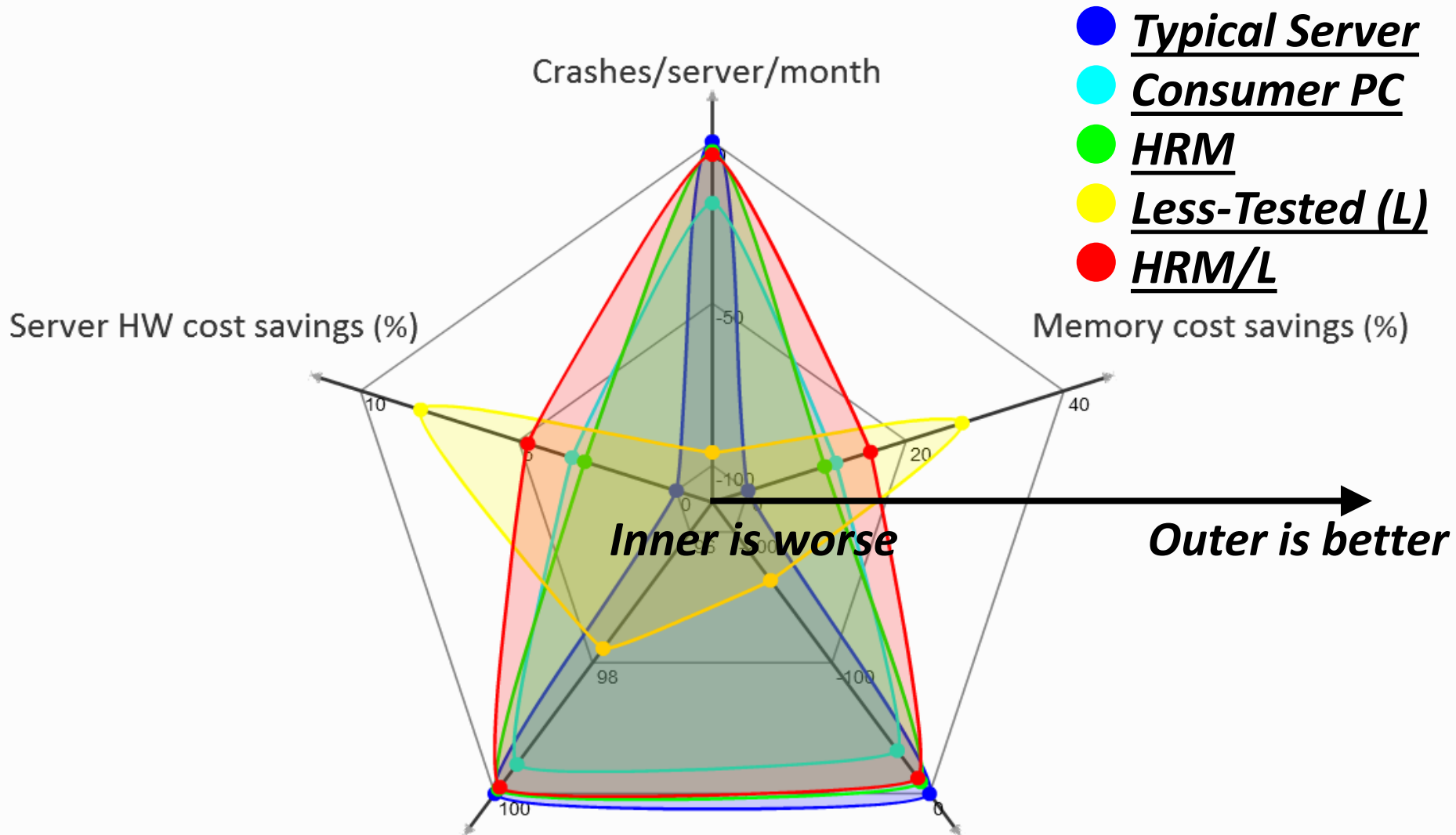
Achieves single server **availability** target of **99.90 %**

**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, {meza, onur}@cmu.edu

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