

Memory Systems

Fundamentals, Recent Research, Challenges, Opportunities

Lecture 2a: Memory Scaling: Reliability & Security

Prof. Onur Mutlu

omutlu@gmail.com

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

8 October 2018

Technion Fast Course 2018

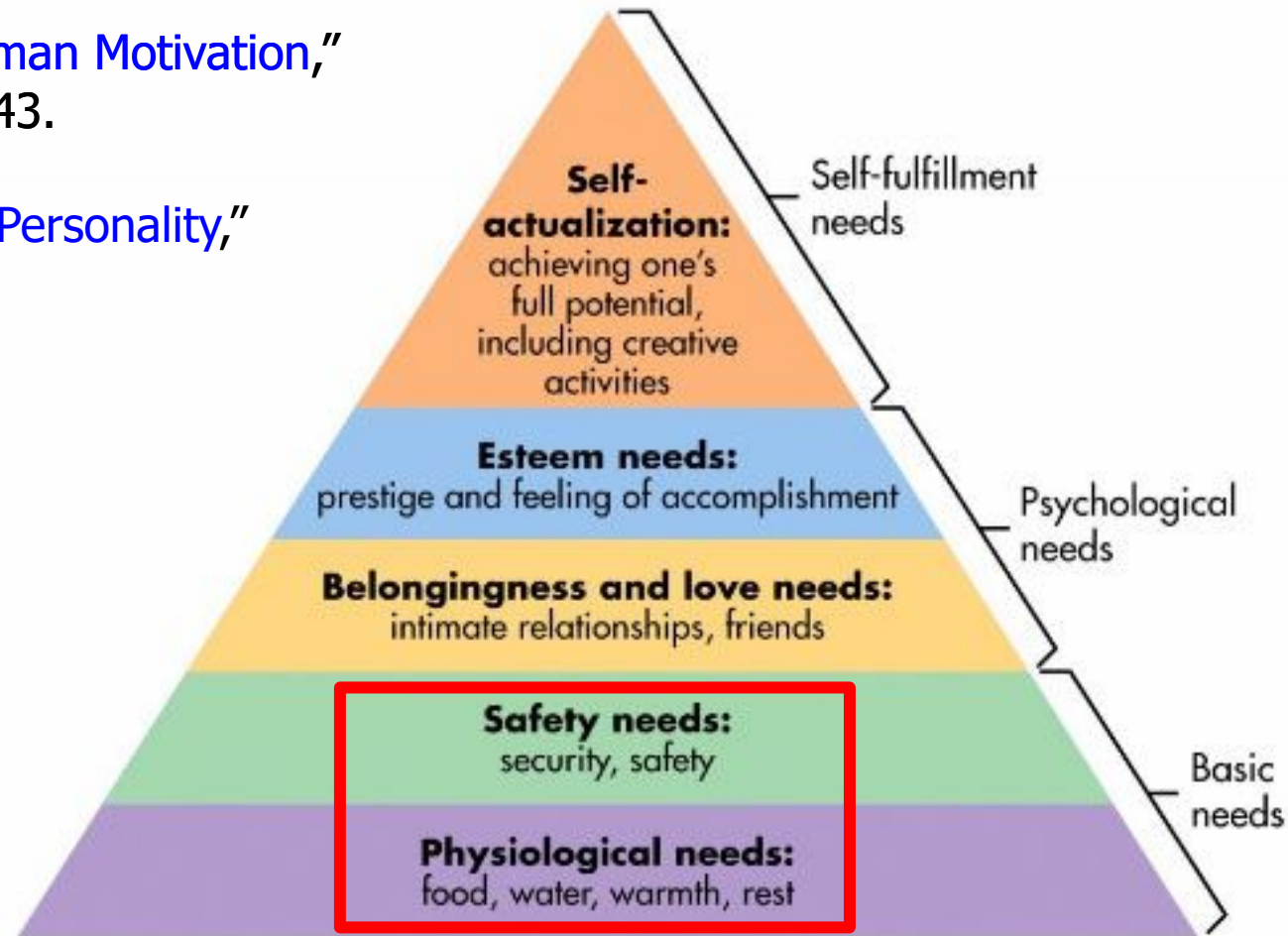
Four Key Directions

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

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”



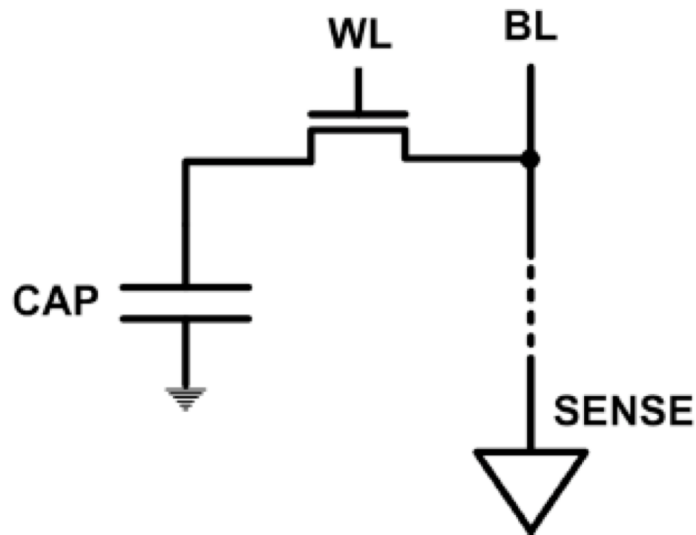
How Secure Are These People?



Security is about preventing unforeseen consequences

The DRAM Scaling Problem

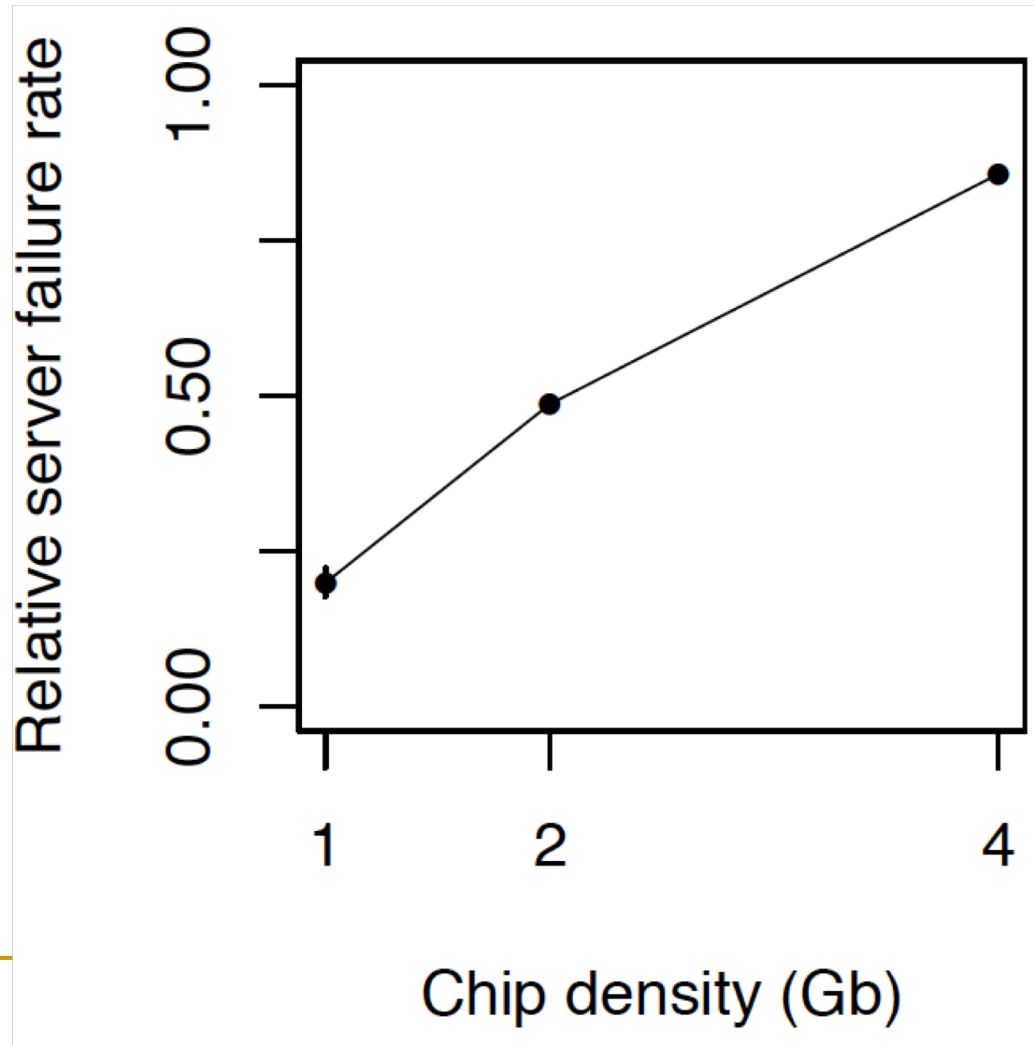
- DRAM stores charge in a capacitor (charge-based memory)
 - Capacitor must be large enough for reliable sensing
 - Access transistor should be large enough for low leakage and high retention time
 - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]



- 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



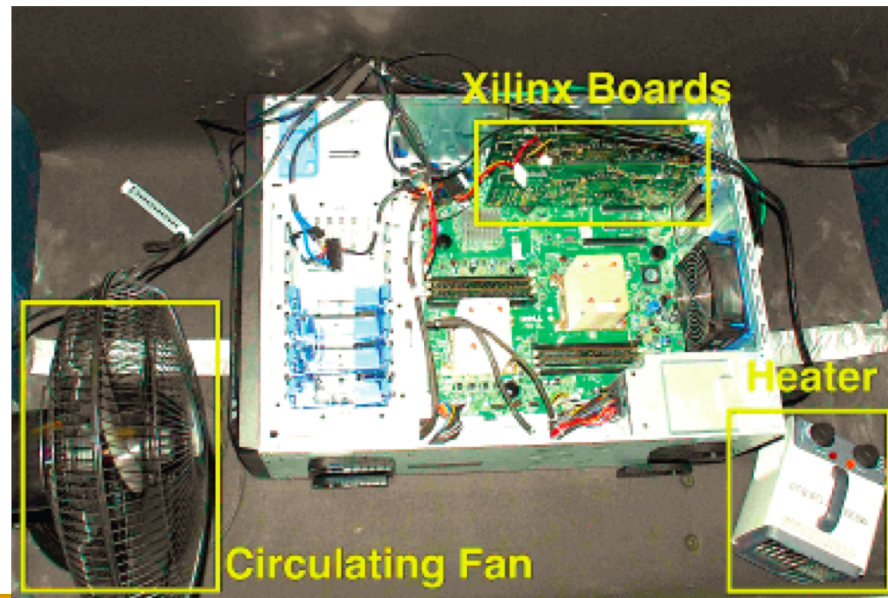
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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

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

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

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



Temperature Controller

PC

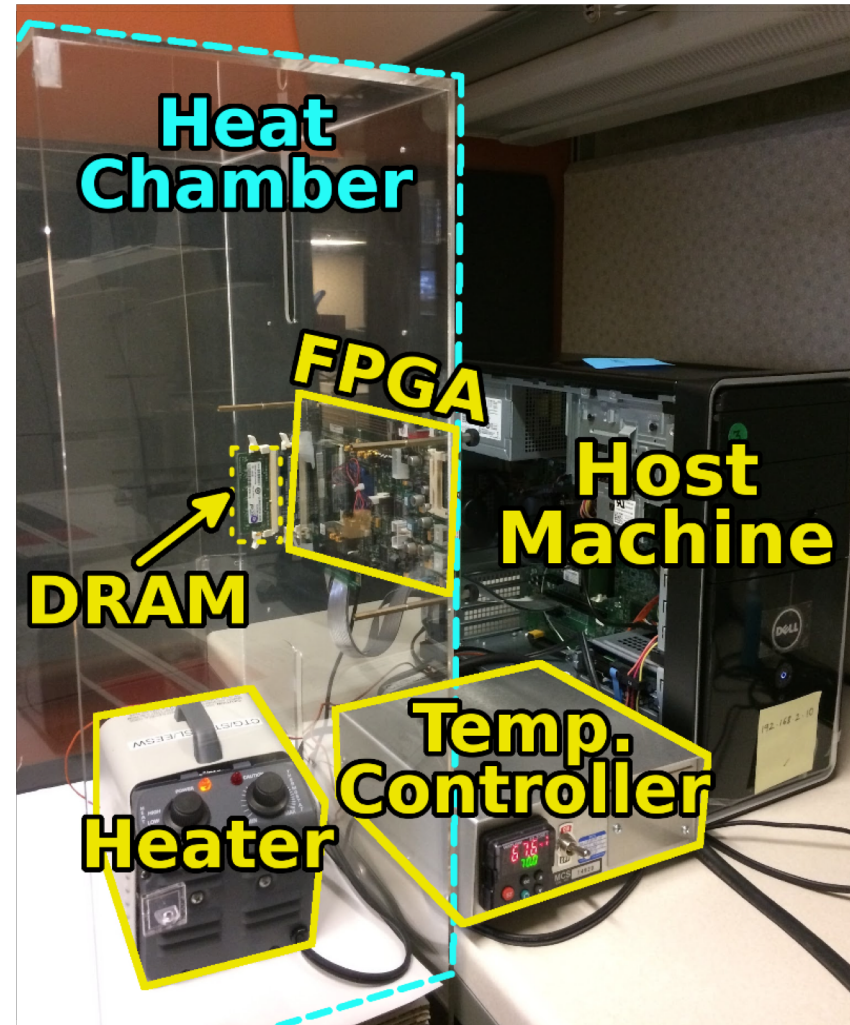
FPGAs

Heater

FPGAs

SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan et al., “**SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies**,” HPCA 2017.
- Flexible
- Easy to Use (C++ API)
- Open-source
github.com/CMU-SAFARI/SoftMC



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

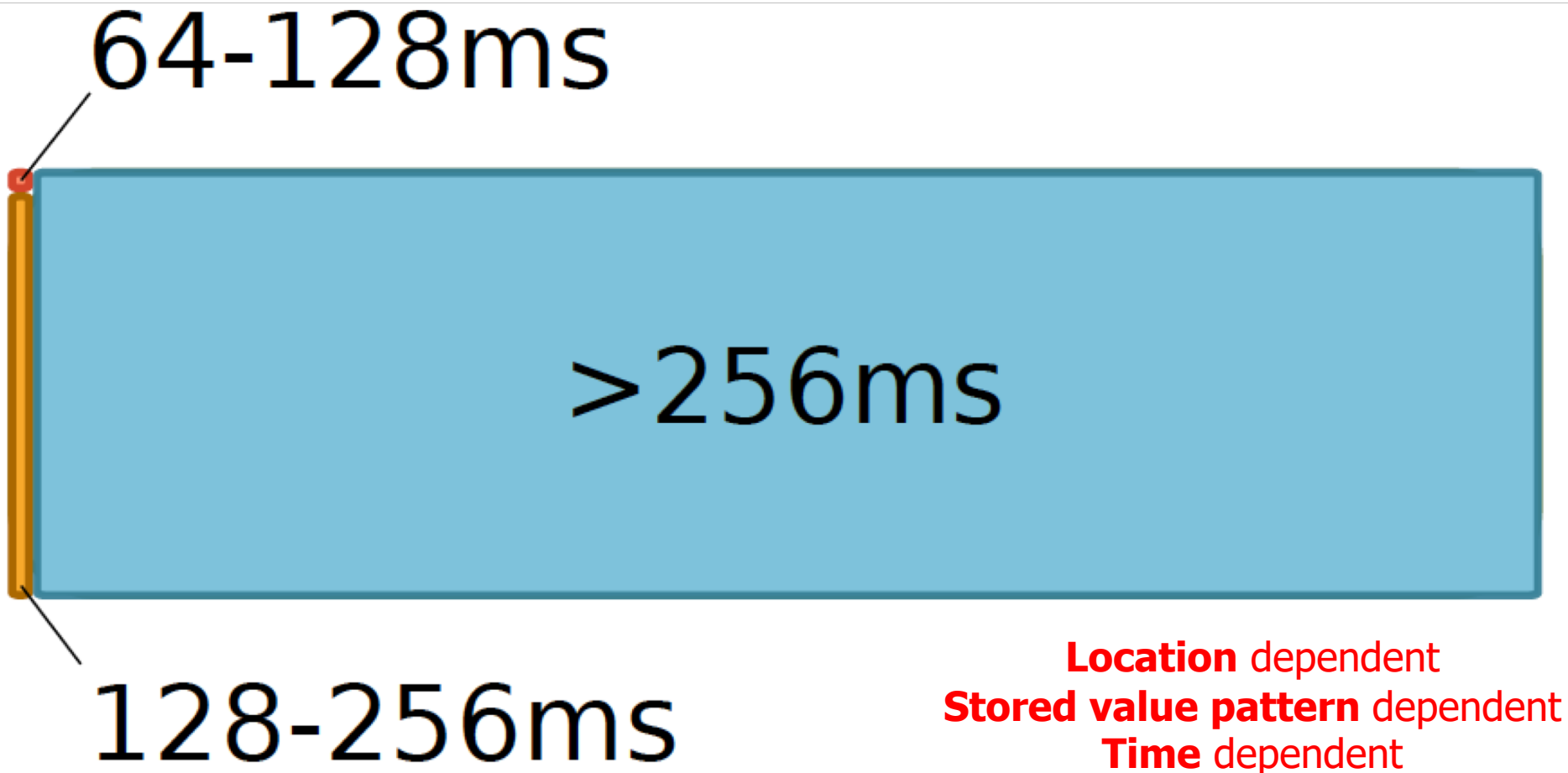
SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan^{1,2,3} Nandita Vijaykumar³ Samira Khan^{4,3} Saugata Ghose³ Kevin Chang³
Gennady Pekhimenko^{5,3} Donghyuk Lee^{6,3} Oguz Ergin² Onur Mutlu^{1,3}

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

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

- Retention Time Profile of DRAM looks like this:



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



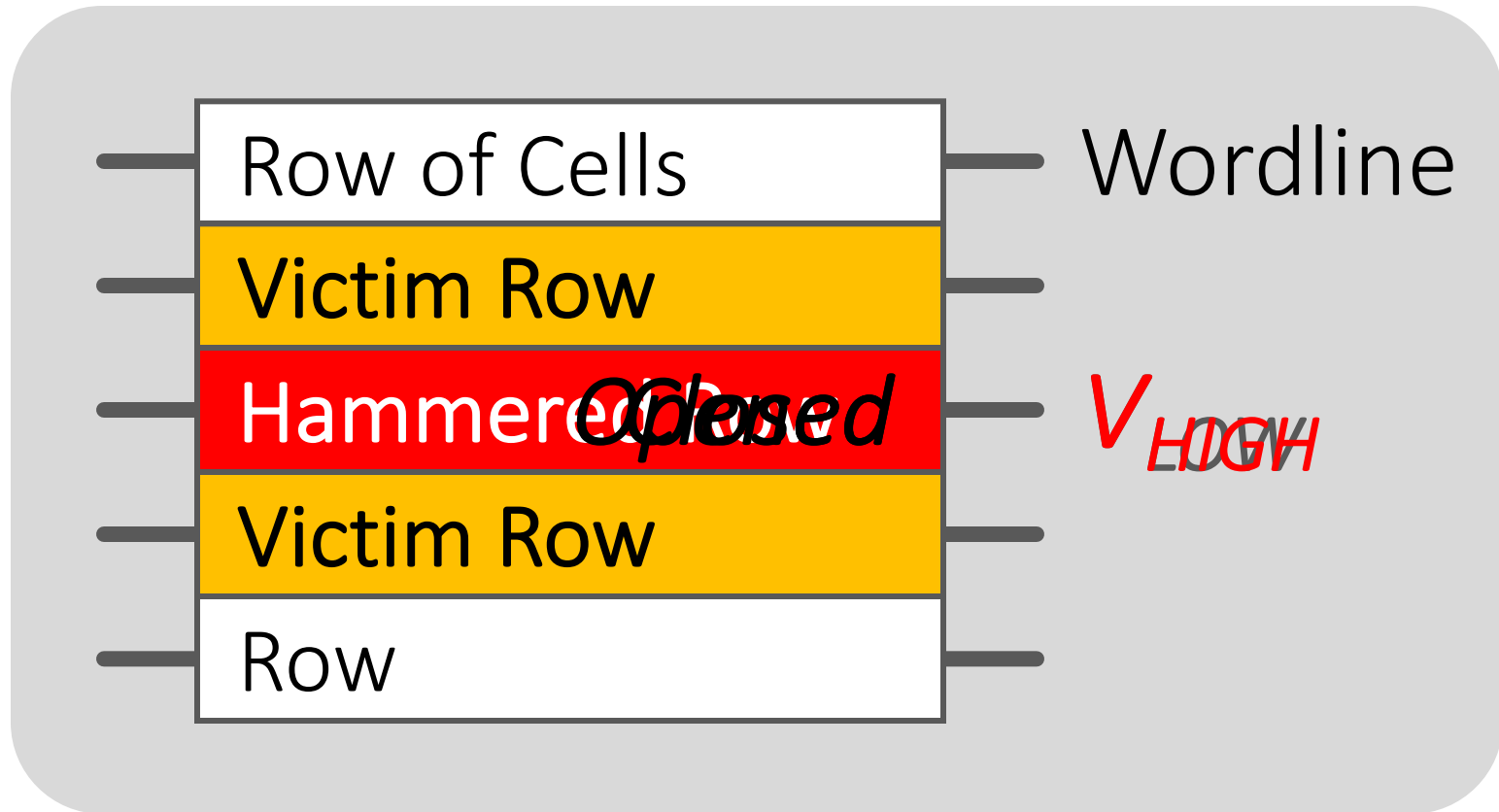
SHARE
18276



TWEET

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

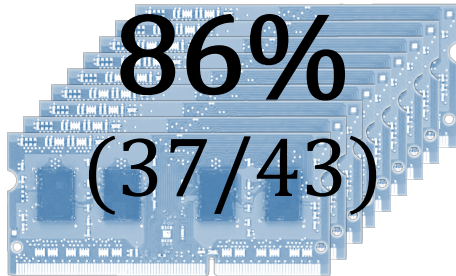
Modern DRAM is Prone to Disturbance Errors



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**

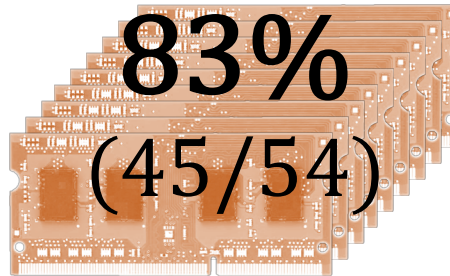
Most DRAM Modules Are Vulnerable

A company



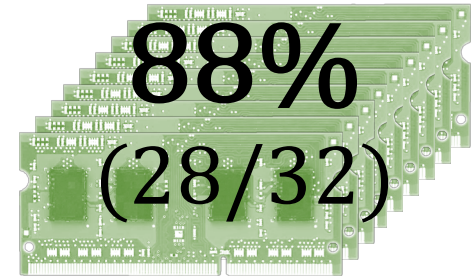
Up to
 1.0×10^7
errors

B company



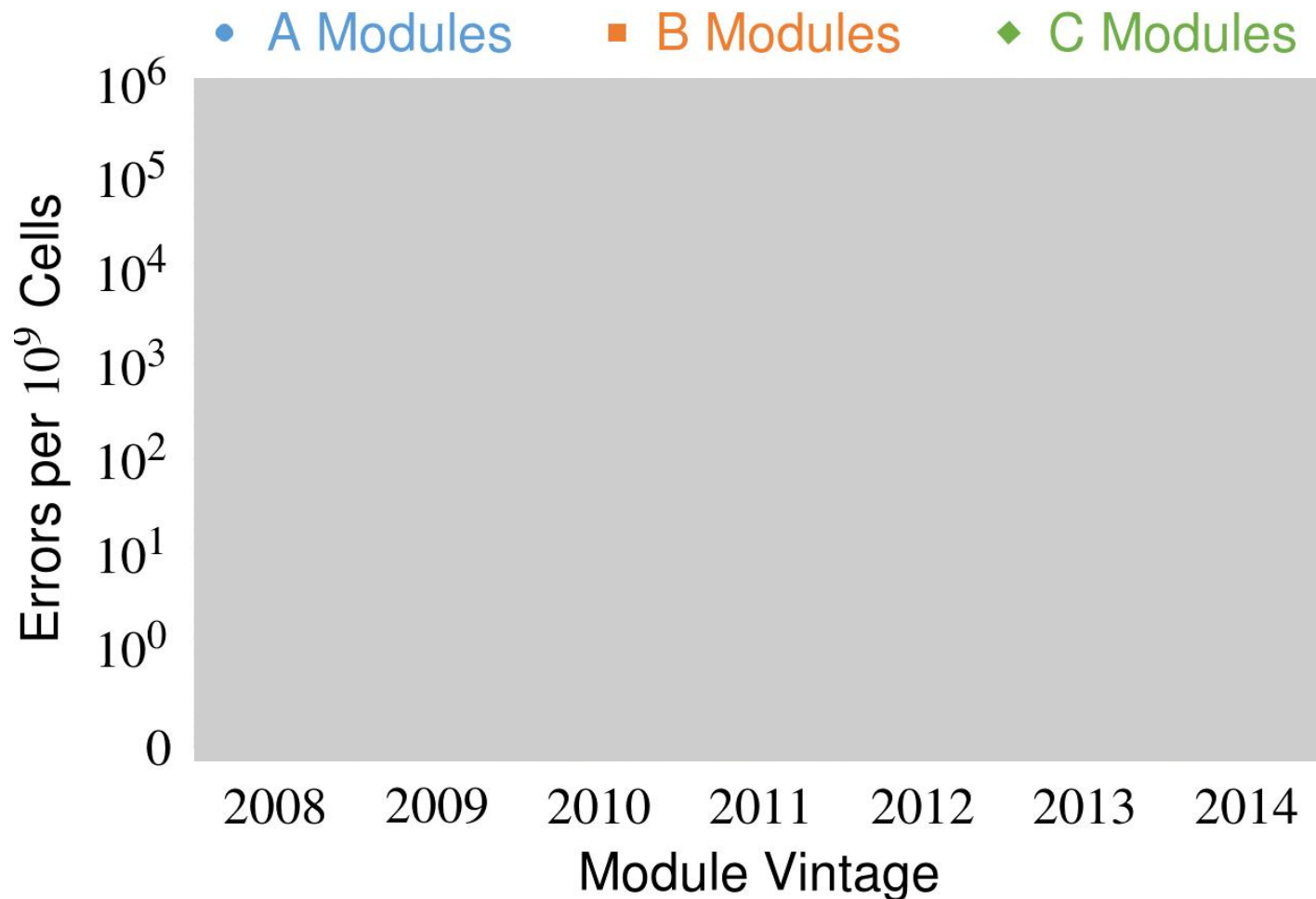
Up to
 2.7×10^6
errors

C company

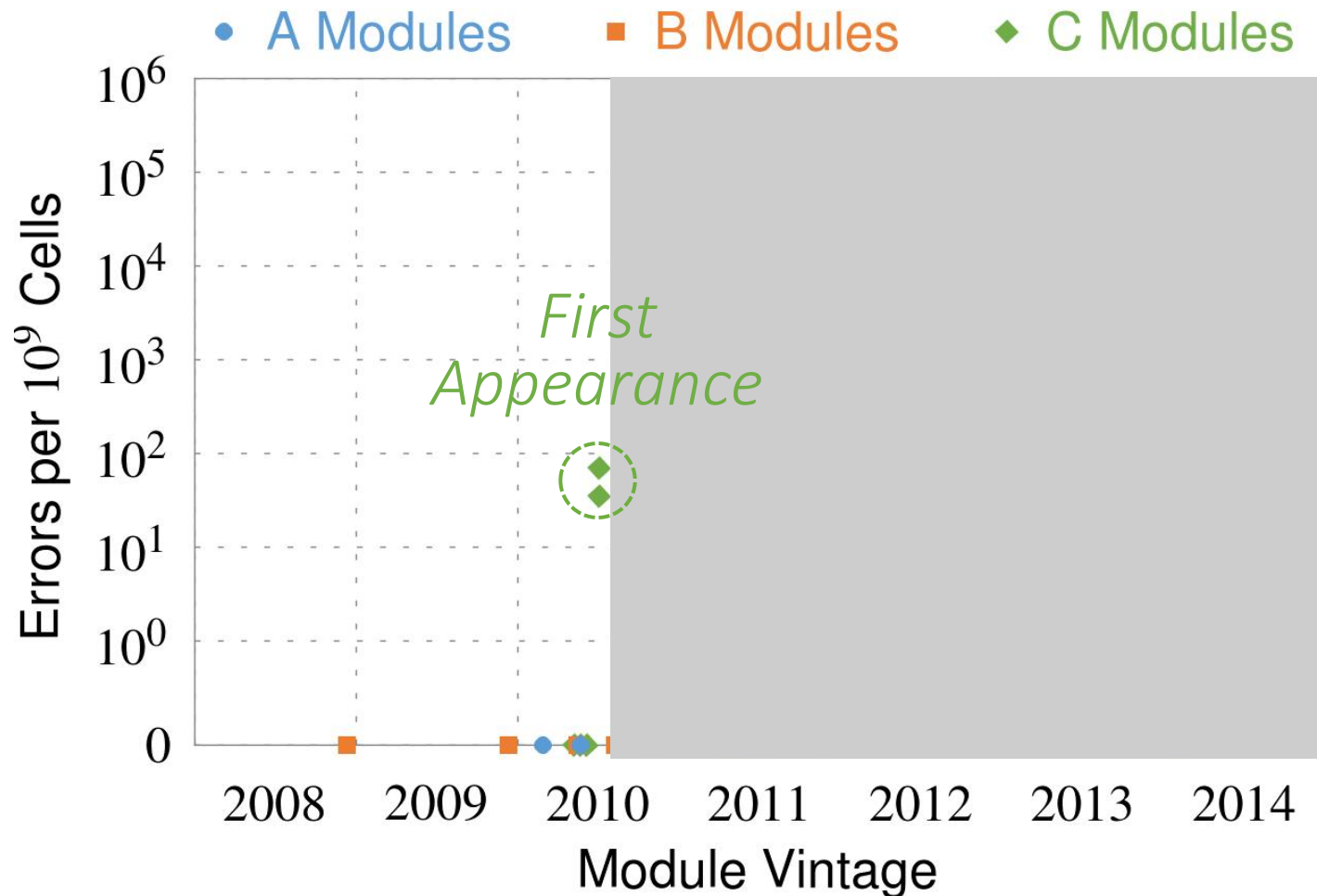


Up to
 3.3×10^5
errors

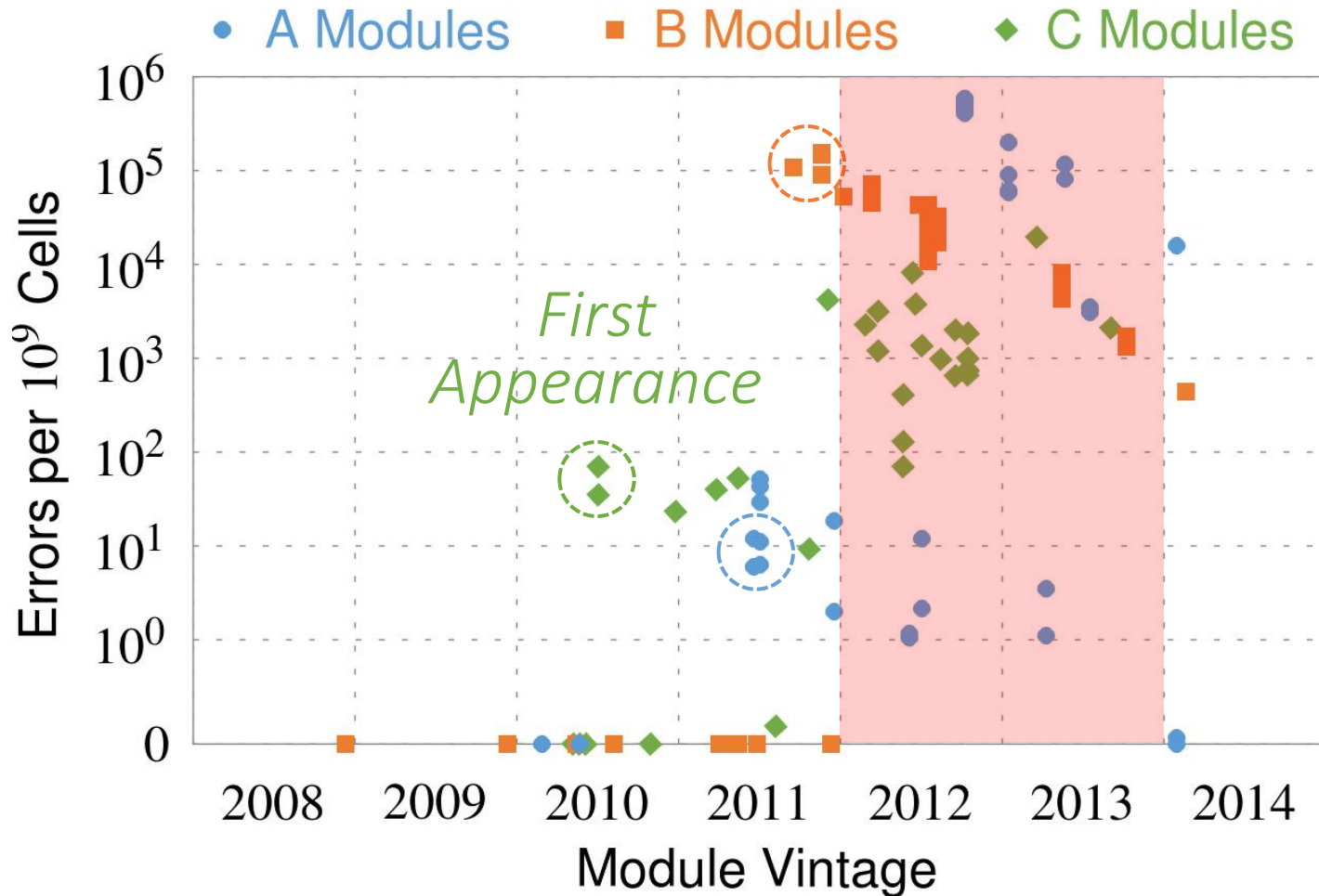
Recent DRAM Is More Vulnerable



Recent DRAM Is More Vulnerable



Recent DRAM Is More Vulnerable



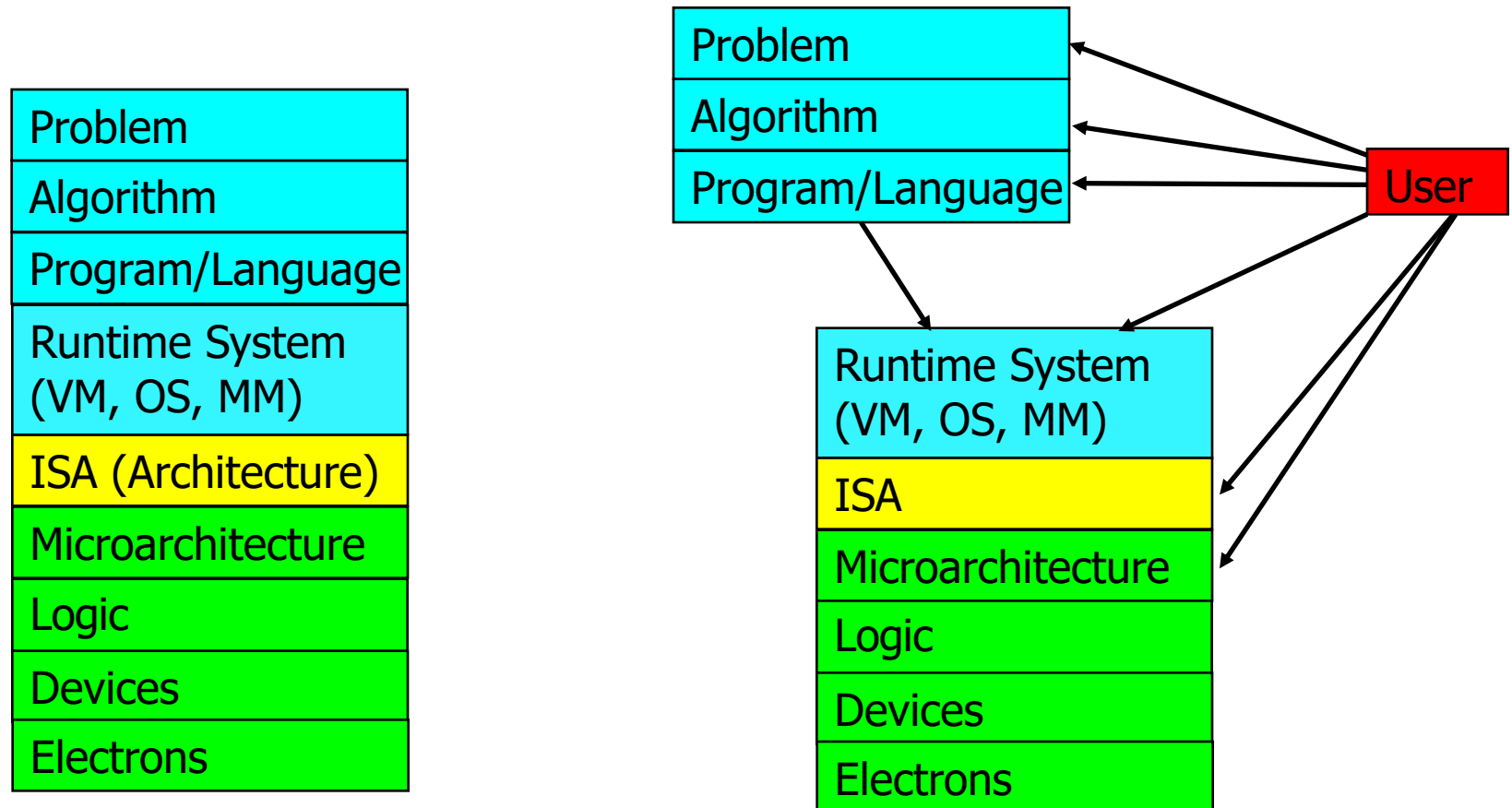
All modules from 2012-2013 are vulnerable

Why Is This Happening?

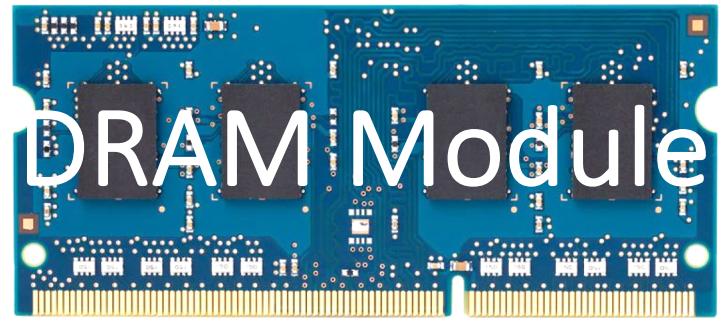
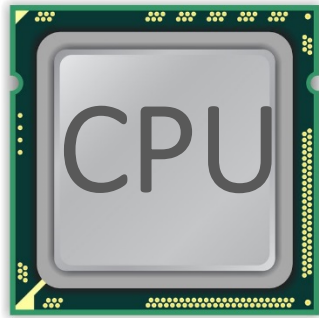
- DRAM cells are too close to each other!
 - They are not electrically isolated from each other
- Access to one cell affects the value in nearby cells
 - due to **electrical interference** between
 - the cells
 - wires used for accessing the cells
 - Also called cell-to-cell coupling/interference
- Example: 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

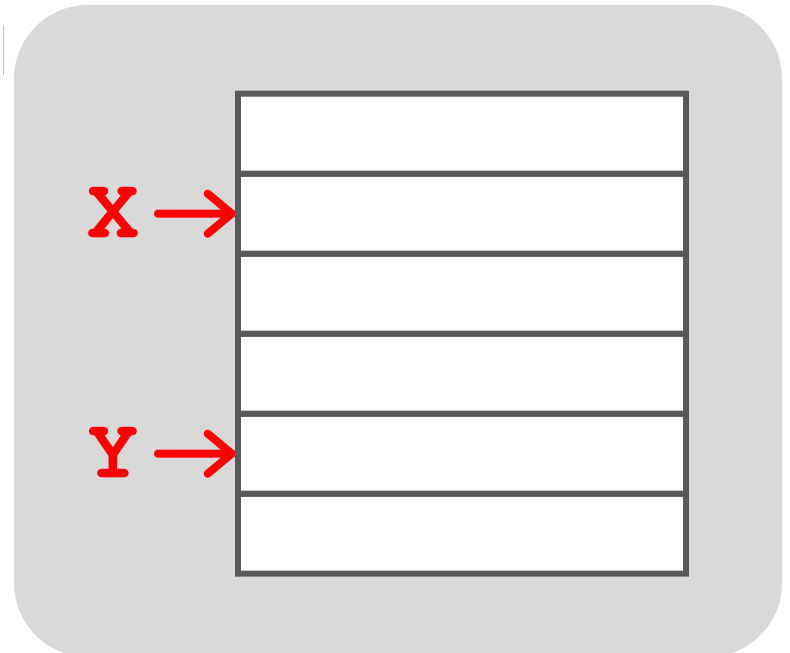
- This simple circuit level failure mechanism has enormous implications on upper layers of the transformation hierarchy



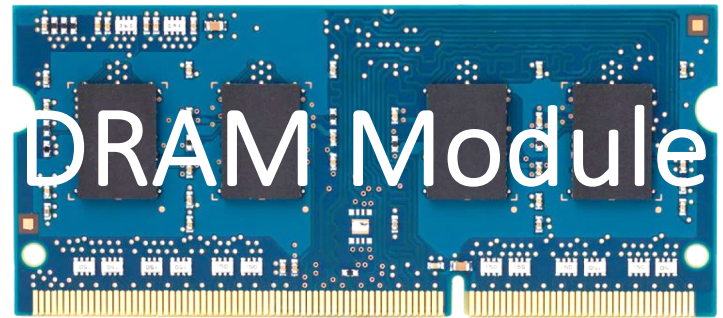
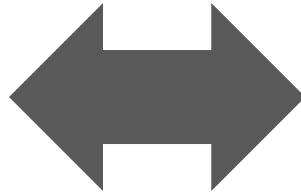
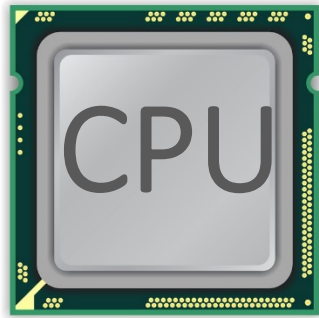
A Simple Program Can Induce Many Errors



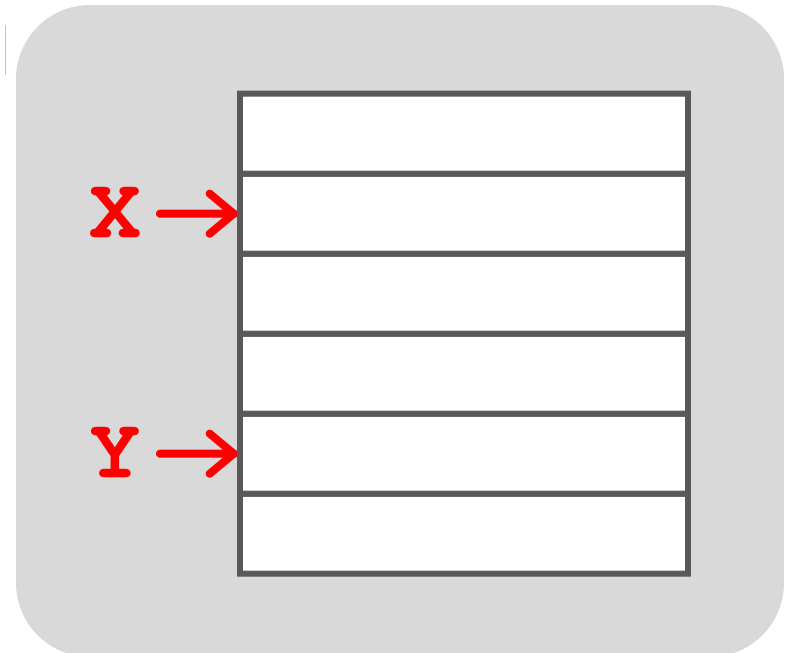
```
loop:  
  mov  (X), %eax  
  mov  (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp  loop
```



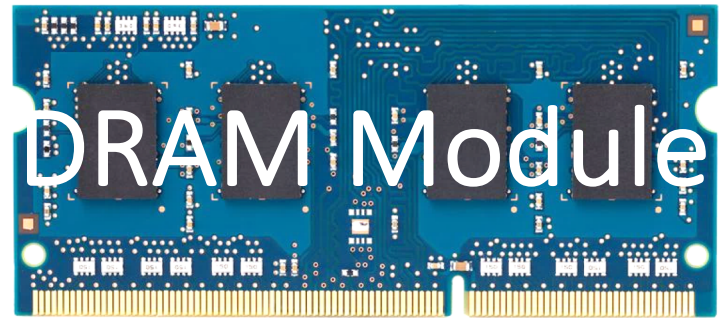
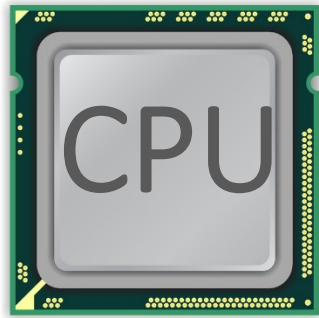
A Simple Program Can Induce Many Errors



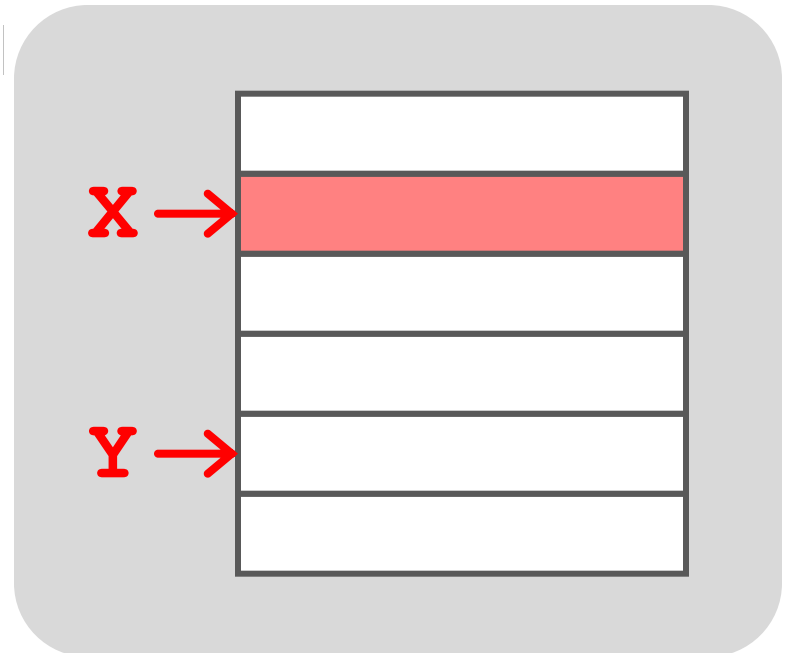
1. Avoid *cache hits*
 - Flush **X** from cache
2. Avoid *row hits* to **X**
 - Read **Y** in another row



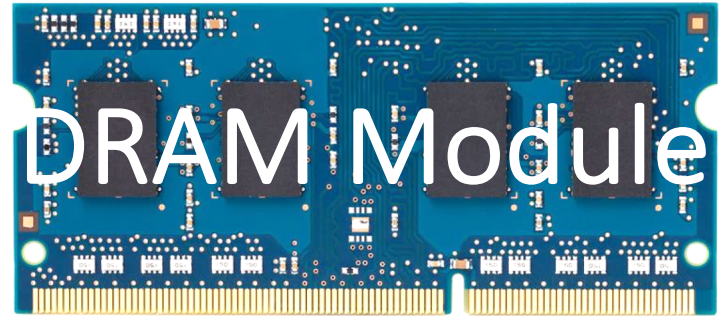
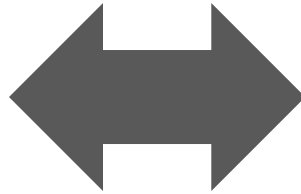
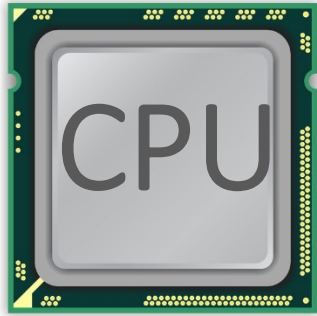
A Simple Program Can Induce Many Errors



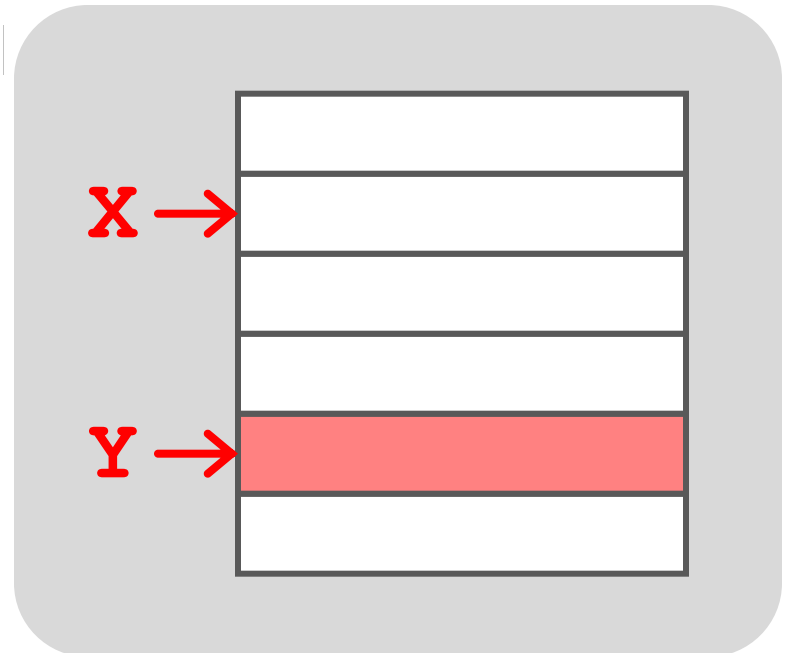
```
loop:  
  mov  (X), %eax  
  mov  (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp  loop
```



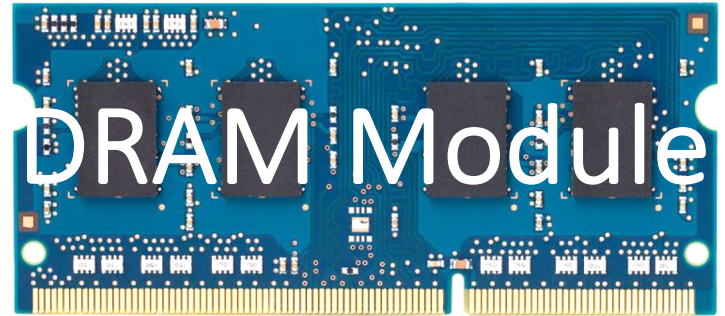
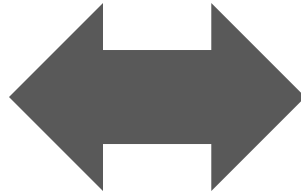
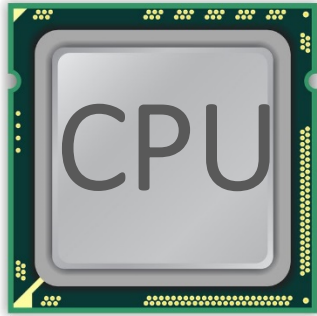
A Simple Program Can Induce Many Errors



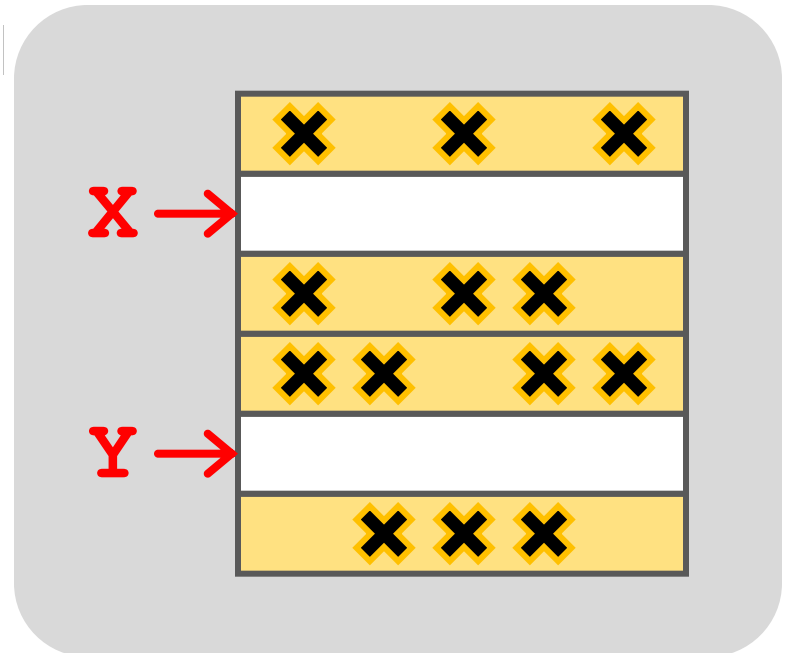
```
loop:  
  mov  (X), %eax  
  mov  (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp  loop
```



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.”

Selected Readings on RowHammer (IV)

- Flip Feng Shui: Hammering a Needle in the Software Stack (August 2016)
 - 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., 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 (V)

- 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
 - 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 (VI)

- 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.”

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)

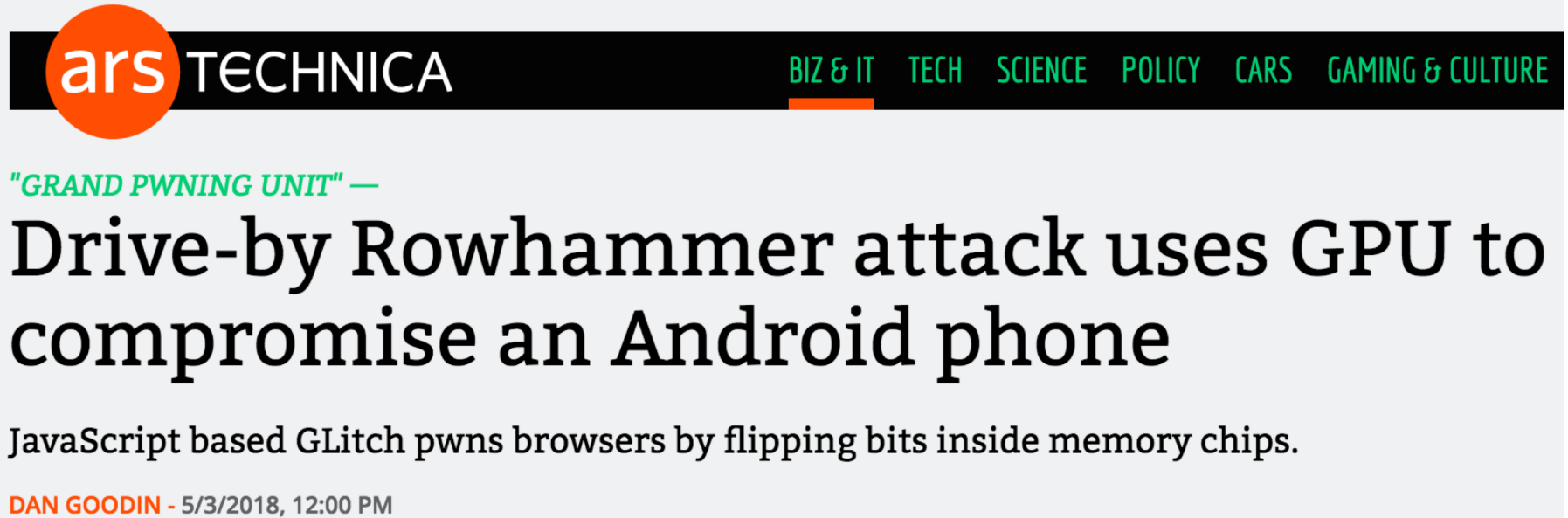
"Can gain control of a smart phone deterministically"



Drammer: Deterministic Rowhammer
Attacks on Mobile Platforms, CCS'16 41

More Security Implications (III)

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



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)



TECHNICA

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

Misiker Tadesse Aga
University of Michigan

Michael Schwarz
Graz University of Technology

Daniel Gruss
Graz University of Technology

Clémentine Maurice
Univ Rennes, CNRS, IRISA

Lukas Raab
Graz University of Technology

Lukas Lamster
Graz University of Technology

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



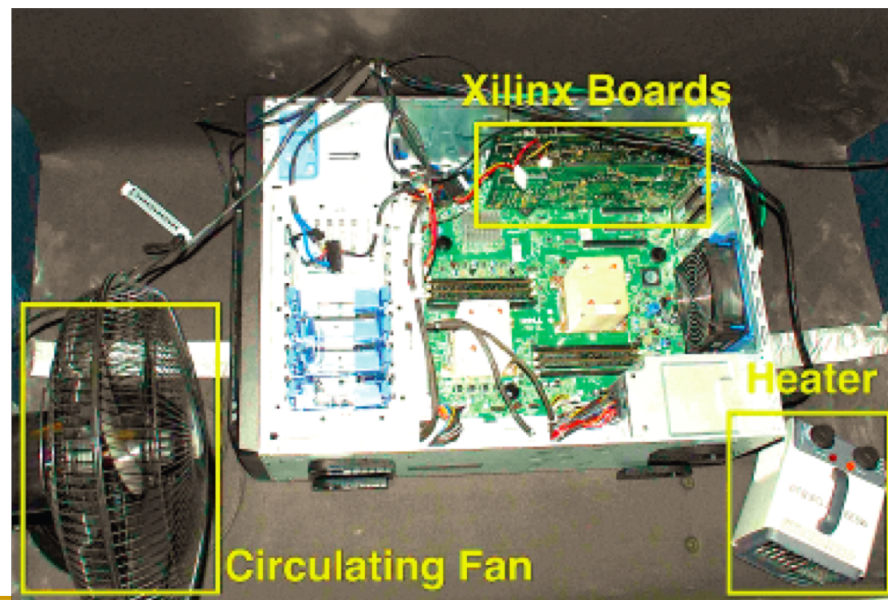
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

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

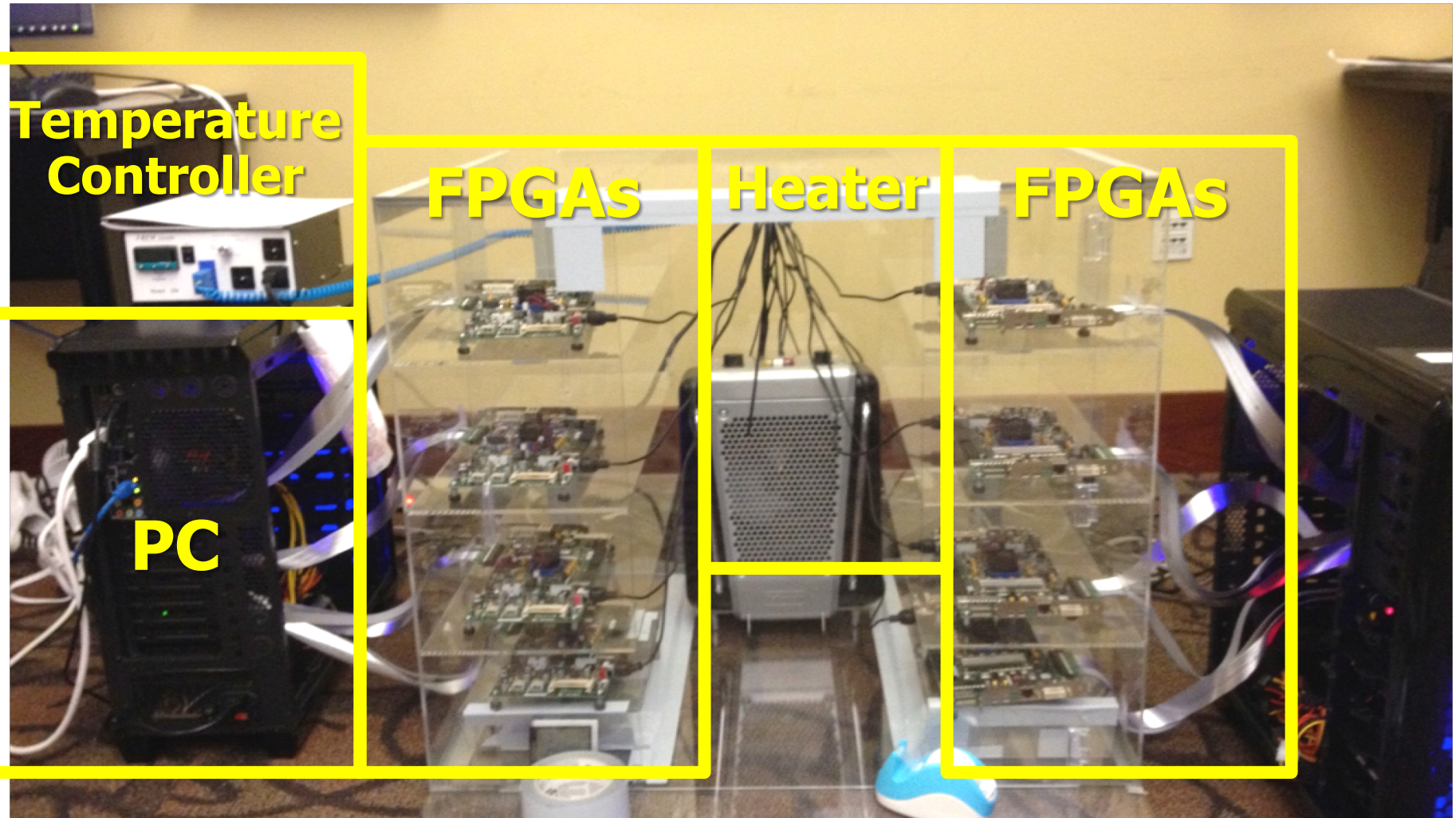
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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)



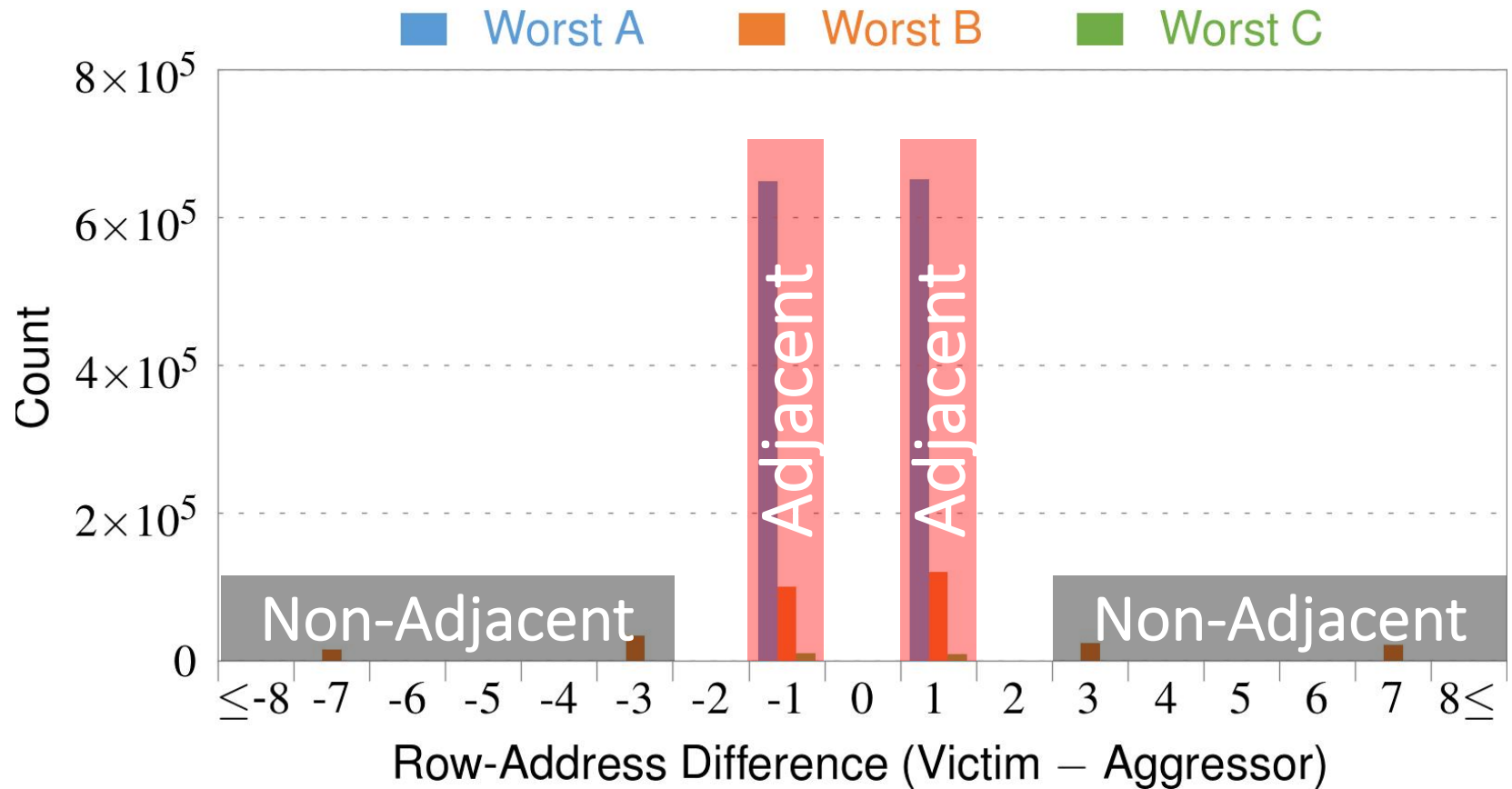
Experimental DRAM Testing Infrastructure



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

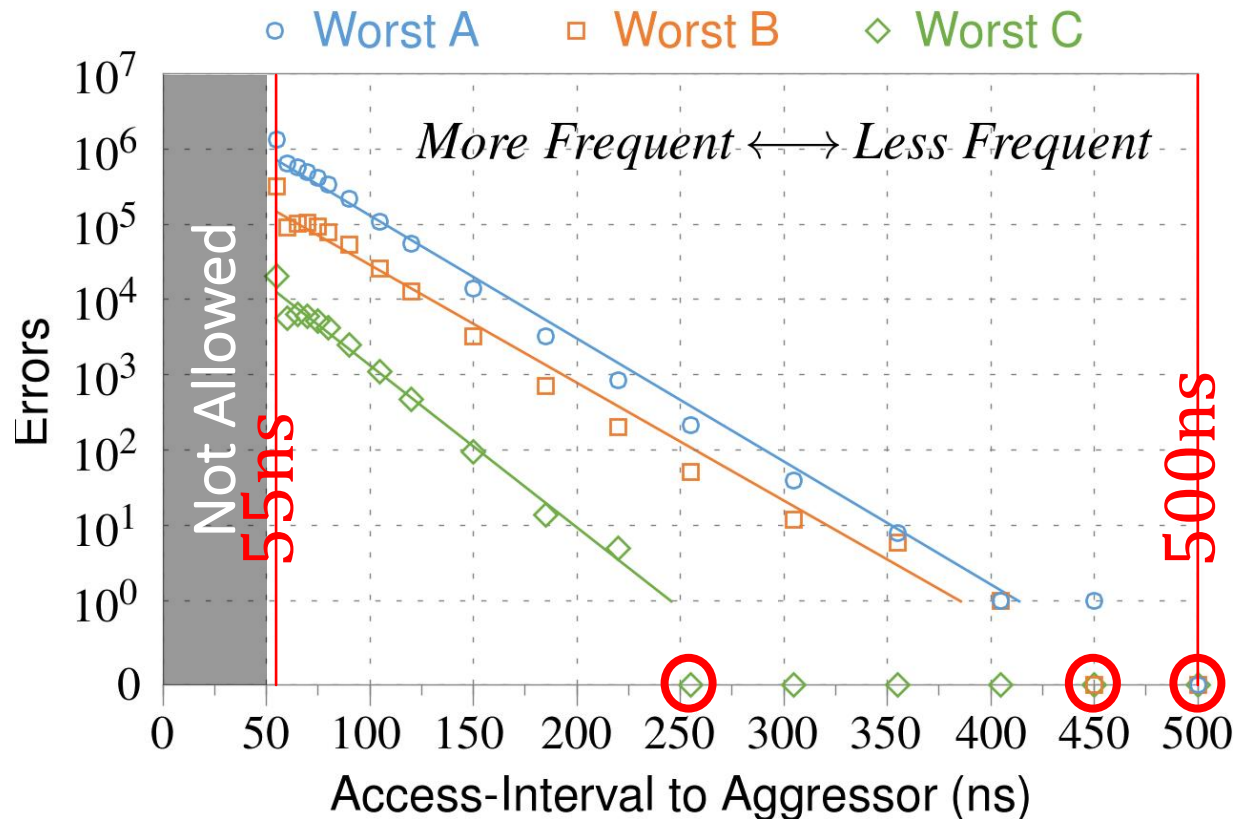
4. Adjacency: Aggressor & Victim



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

Most aggressors & victims are adjacent

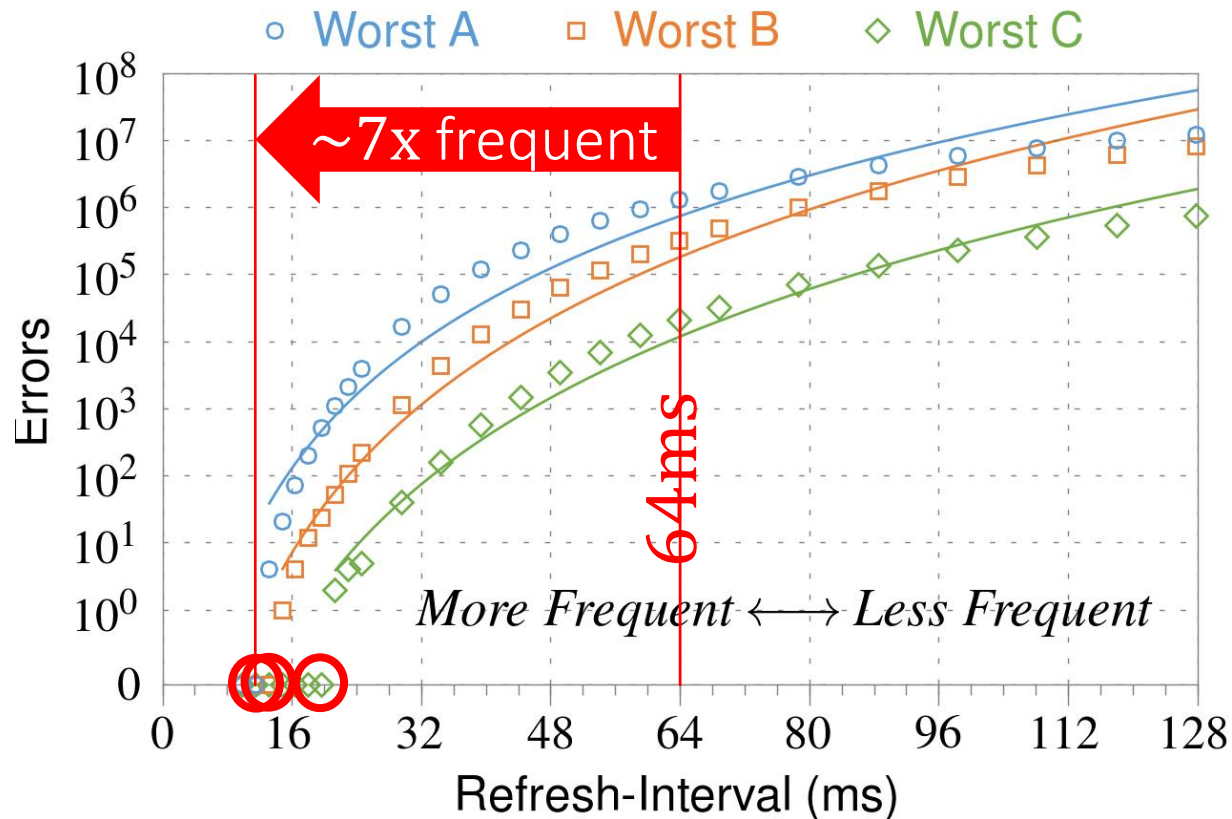
① Access Interval (Aggressor)



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

Less frequent accesses \rightarrow Fewer errors

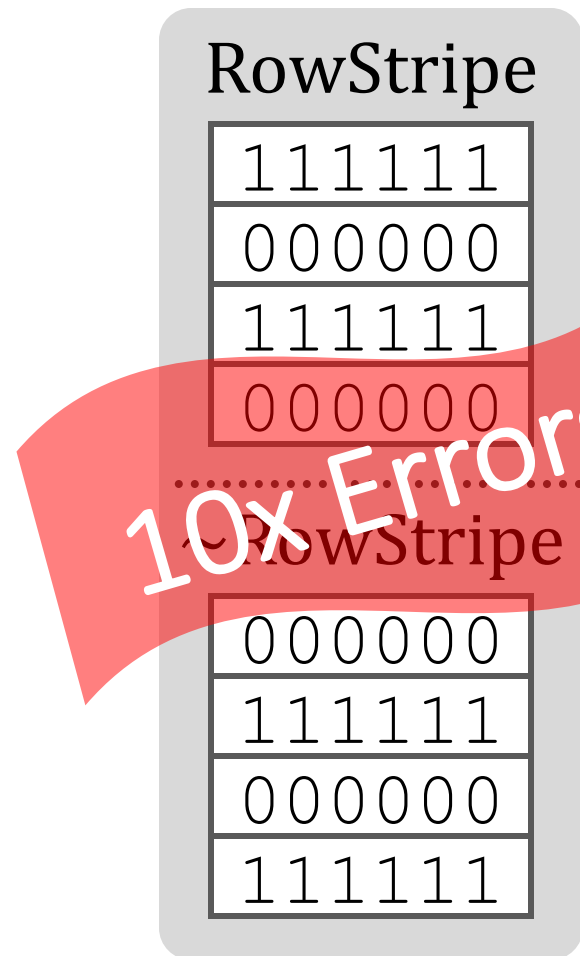
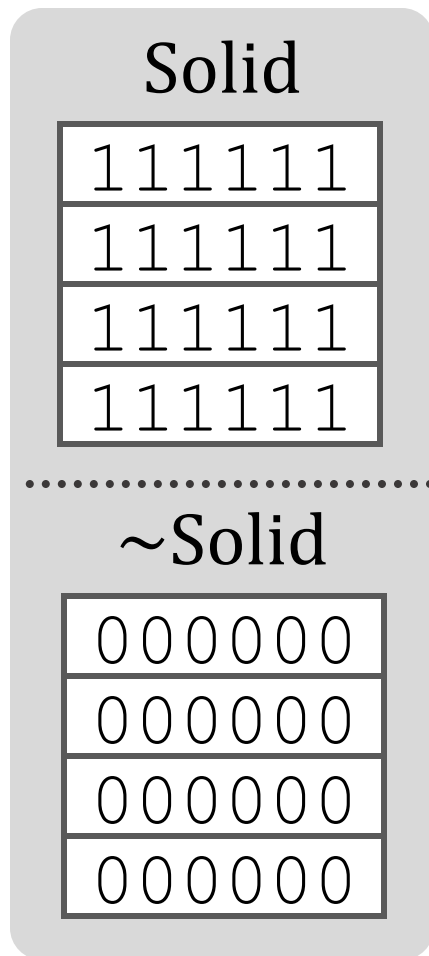
2 Refresh Interval



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

More frequent refreshes \rightarrow 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: ≤ 110
 - Victims rows per aggressor: ≤ 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 Kim¹ Ross Daly* Jeremie Kim¹ Chris Fallin* Ji Hye Lee¹
Donghyuk Lee¹ Chris Wilkerson² Konrad Lai Onur Mutlu¹

¹Carnegie Mellon University ²Intel 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>

RowHammer Solutions

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: $\leq 128\text{K}$ ($=64\text{ms}/500\text{ns}$)

2 *Refresh more frequently*

- Shorten refresh-interval by $\sim 7\times$

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: 9.4×10^{-14}
 - By adjusting the value of p , we can vary the strength of protection against errors

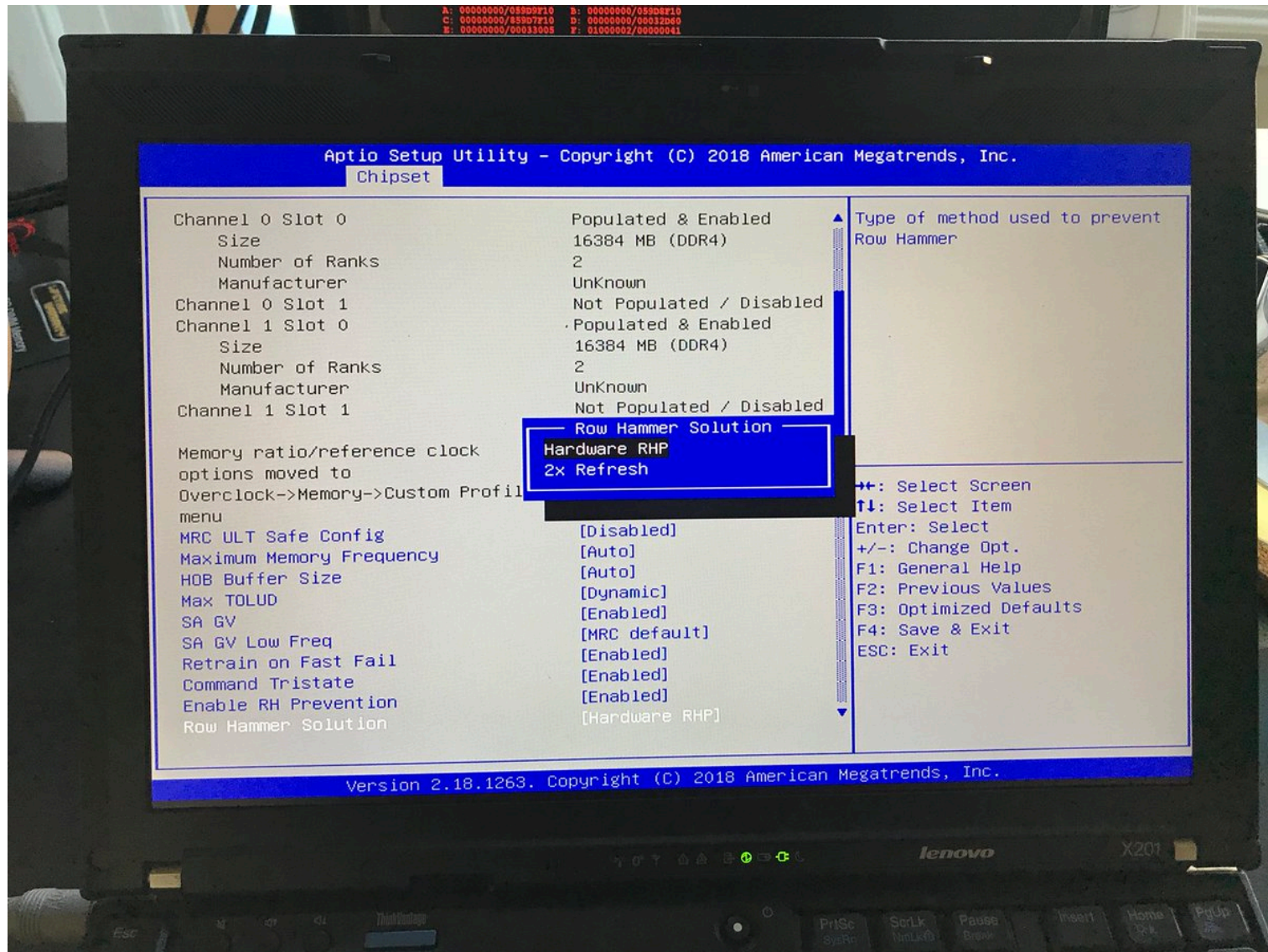
Advantages of PARA

- *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*

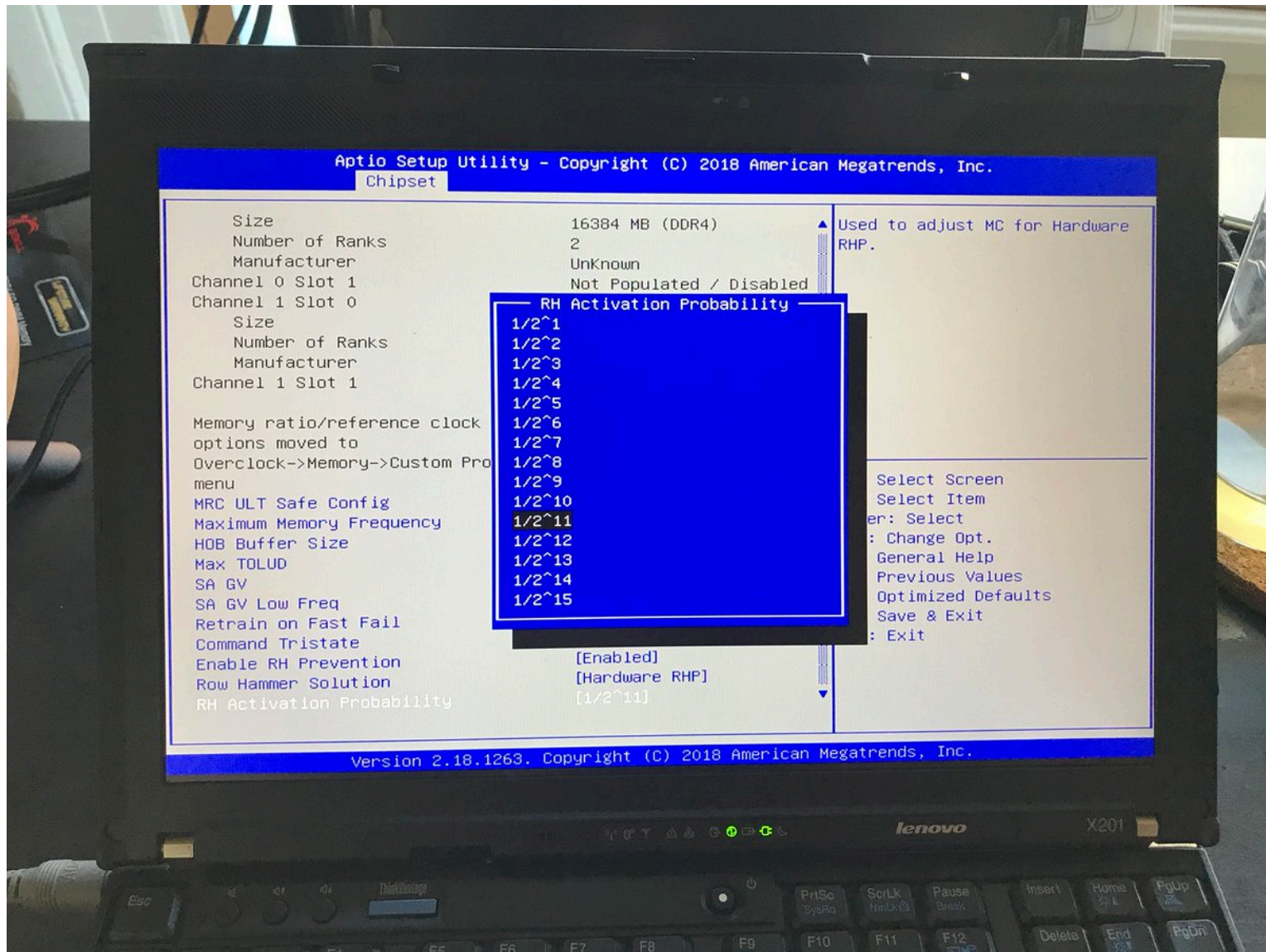
Requirements for PARA

- If implemented in **DRAM chip**
 - 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)



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 Kim¹ Ross Daly* Jeremie Kim¹ Chris Fallin* Ji Hye Lee¹
Donghyuk Lee¹ Chris Wilkerson² Konrad Lai Onur Mutlu¹

¹Carnegie Mellon University ²Intel Labs

Future of Memory Reliability

- 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>

Industry Is Writing Papers About It, Too

DRAM Process Scaling Challenges

❖ Refresh

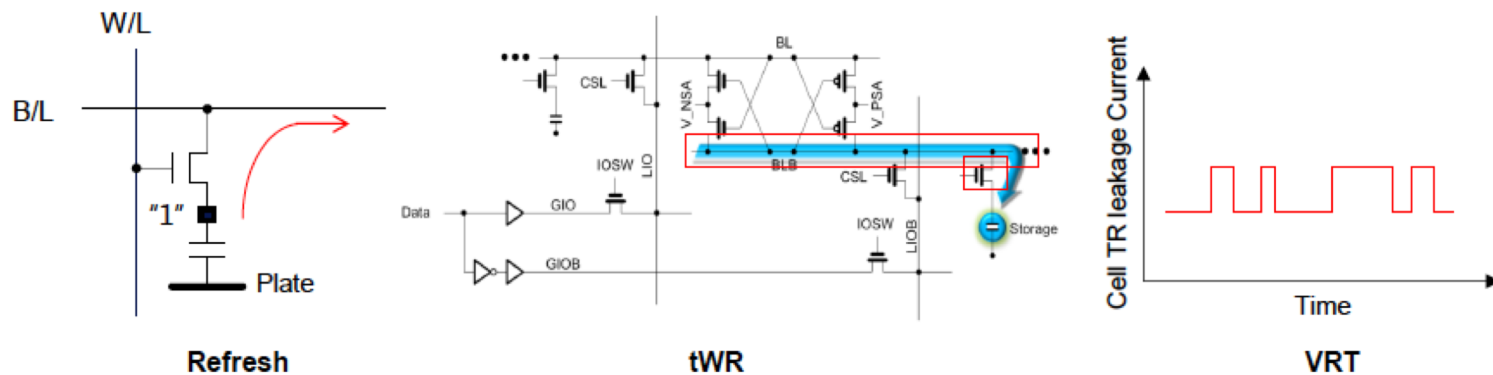
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

❖ tWR

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

❖ VRT

- Occurring more frequently with cell capacitance decreasing



Call for Intelligent Memory Controllers

DRAM Process Scaling Challenges

❖ Refresh

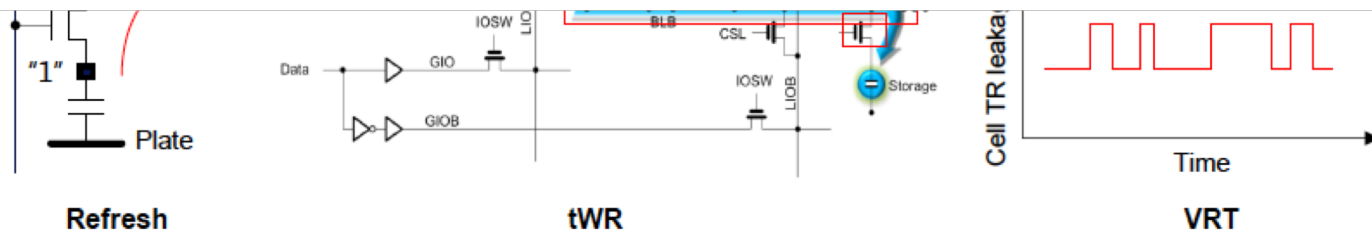
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

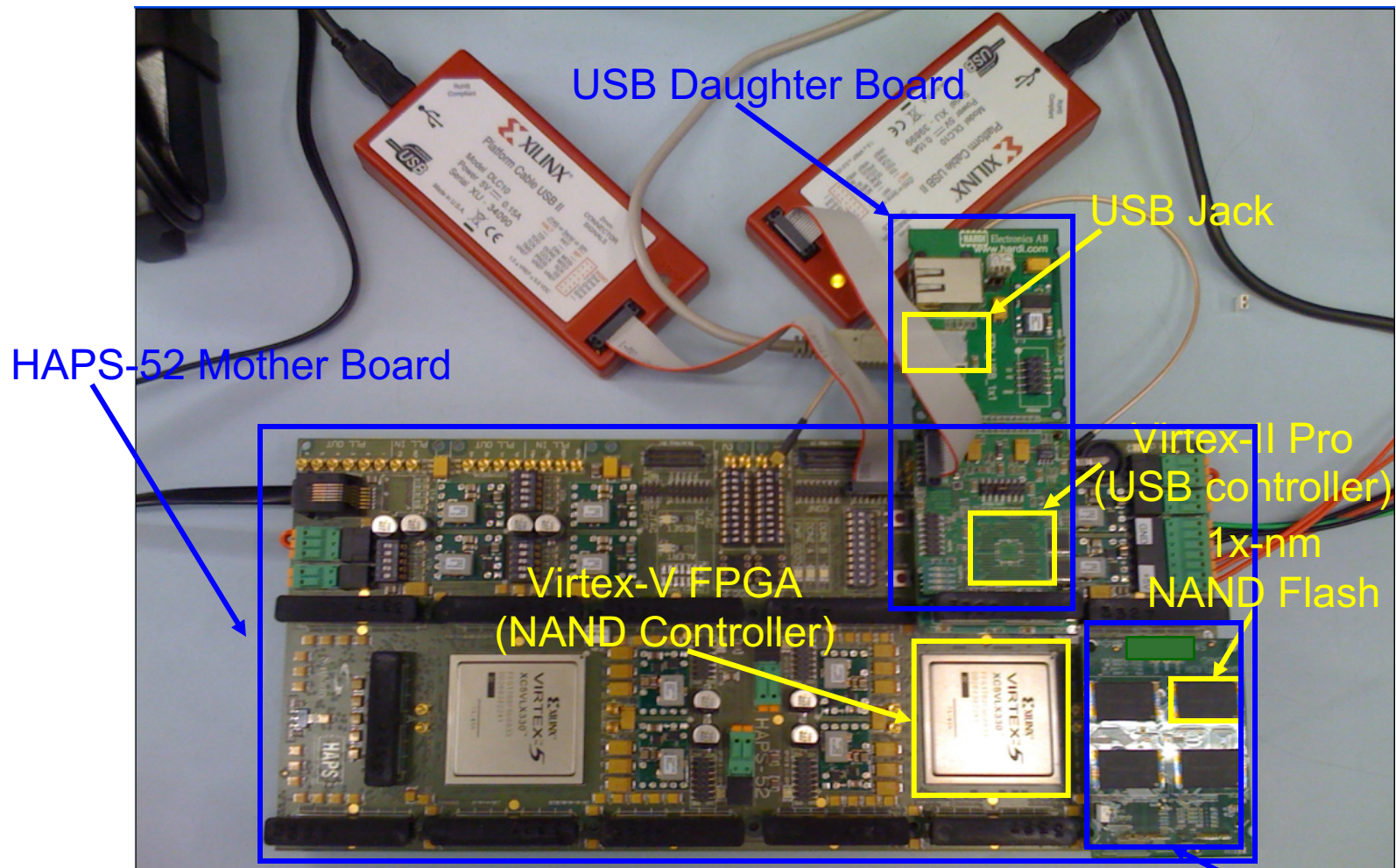
Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

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

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



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**

Fundamentally Secure, Reliable, Safe Computing Architectures

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.

Future Memory Reliability/Security Challenges

Future of Main Memory

- DRAM is becoming less reliable → more vulnerable

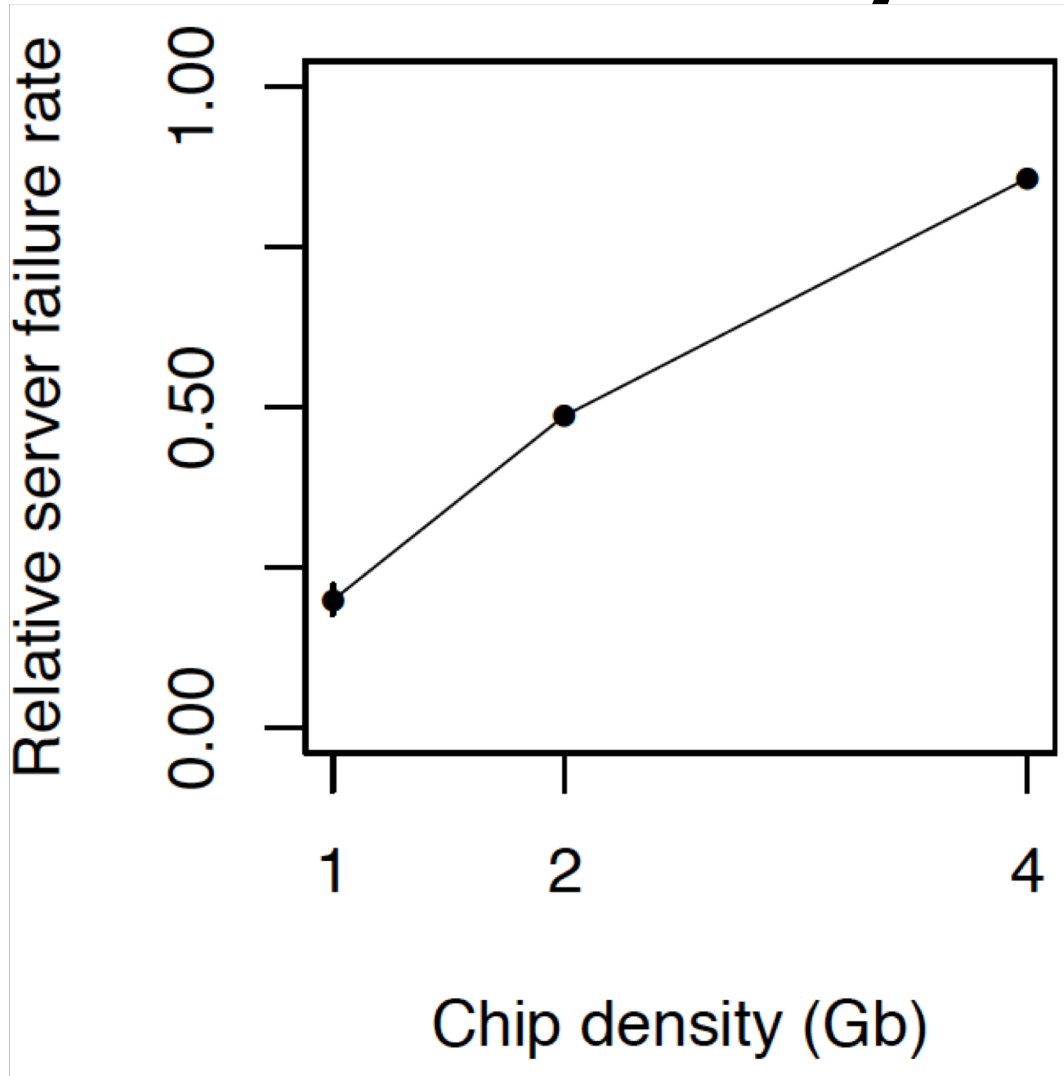
Large-Scale Failure Analysis of DRAM Chips

- 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

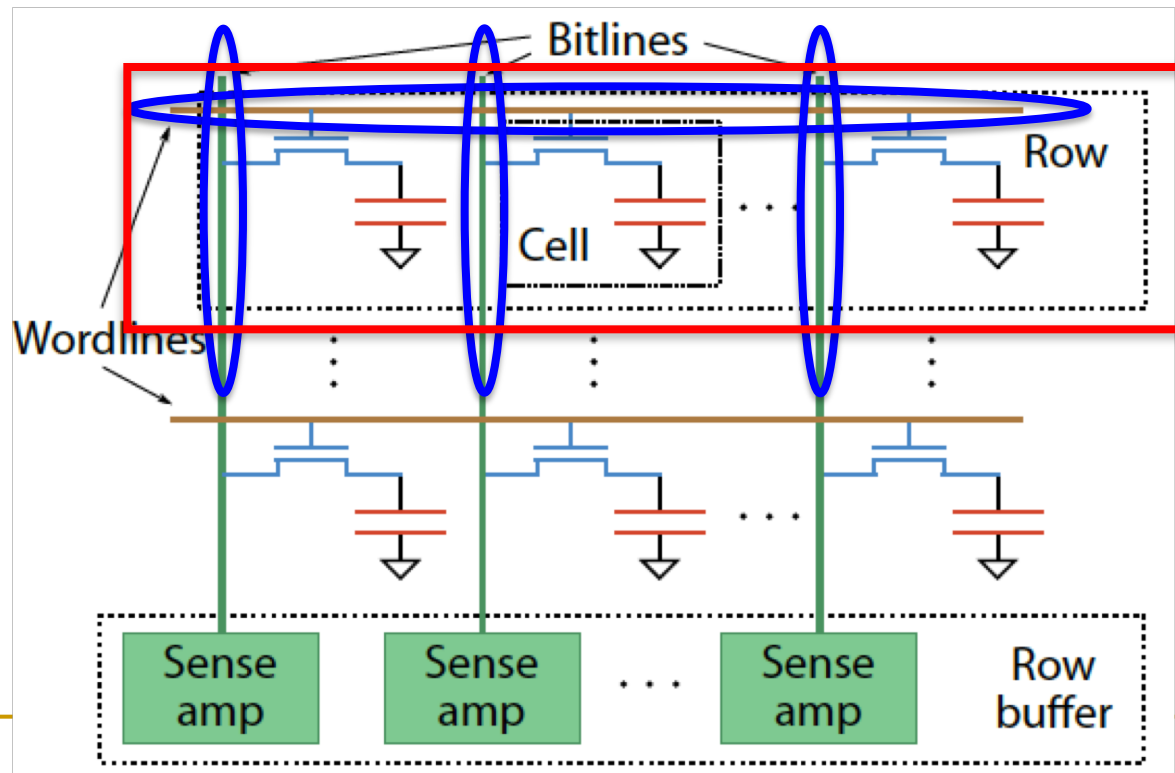
Onur Mutlu
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
onur@cmu.edu

Two Challenges to Retention Time Profiling

- Data Pattern Dependence (DPD) of retention time
- Variable Retention Time (VRT) phenomenon

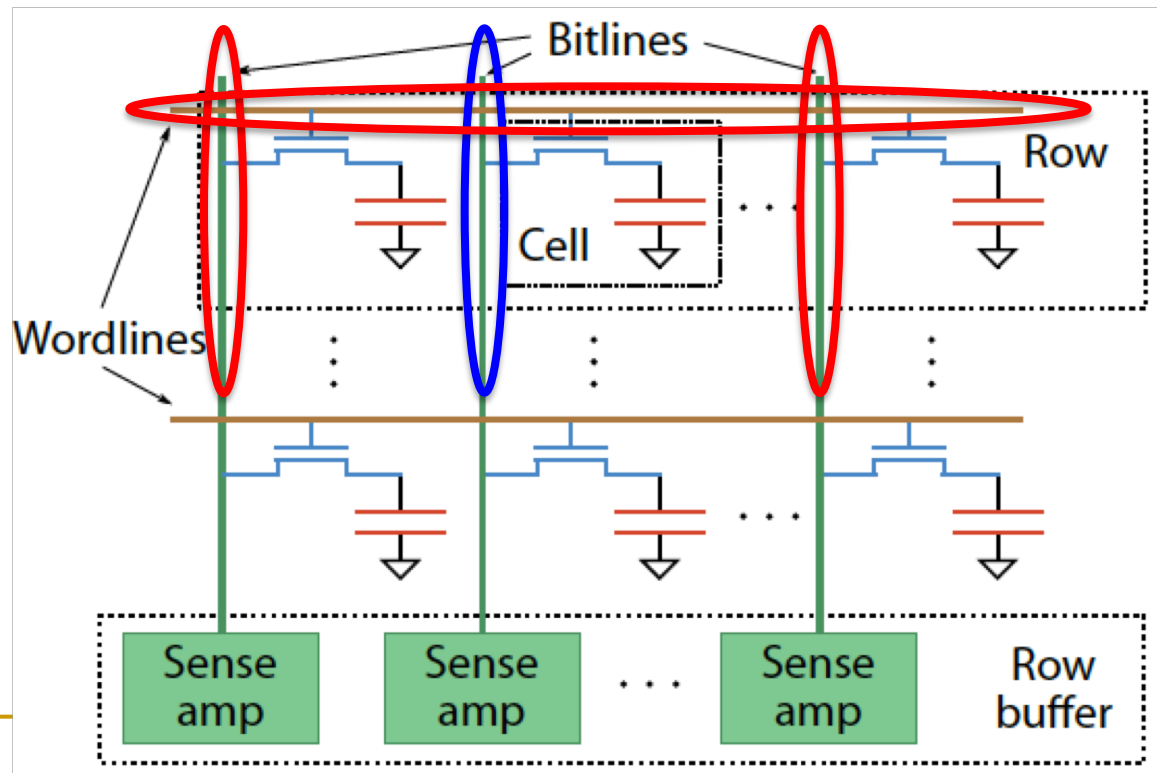
Two Challenges to Retention Time Profiling

- **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



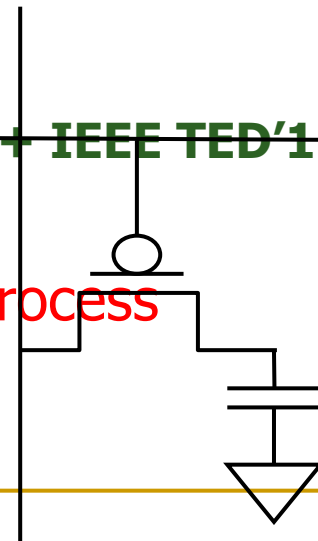
Data Pattern Dependence

- 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

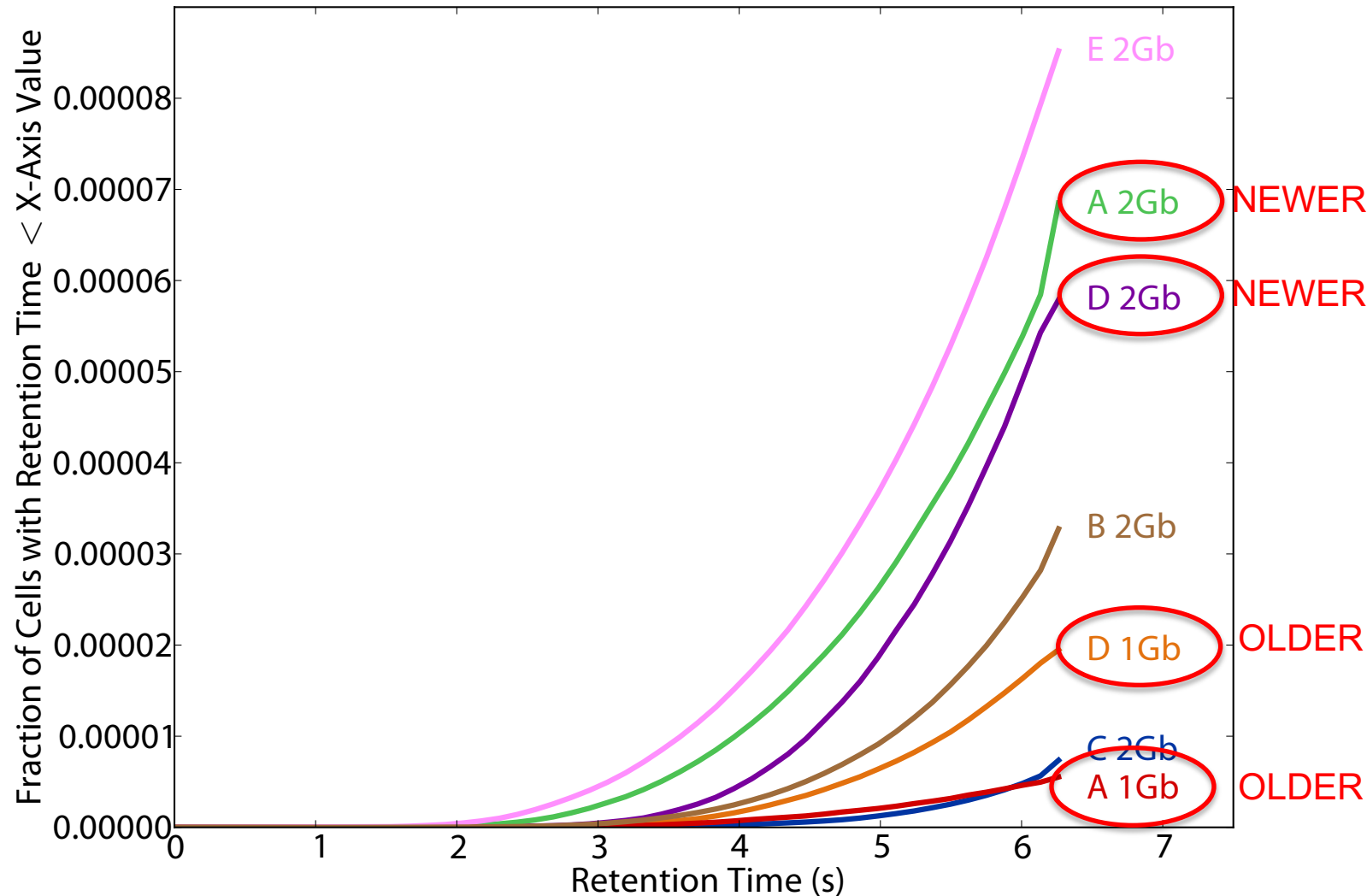
Two Challenges to Retention Time Profiling

■ 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

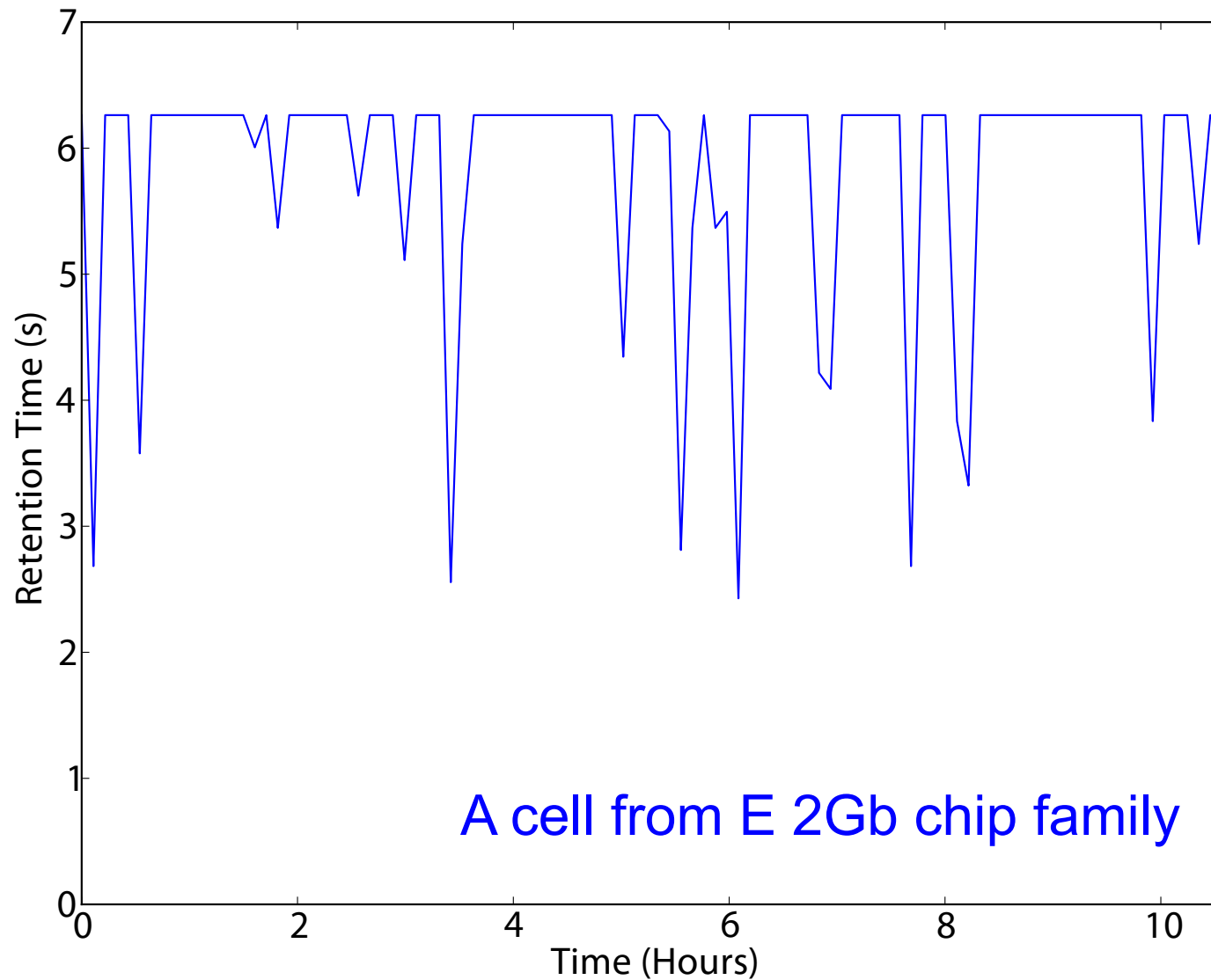


Modern DRAM Retention Time Distribution

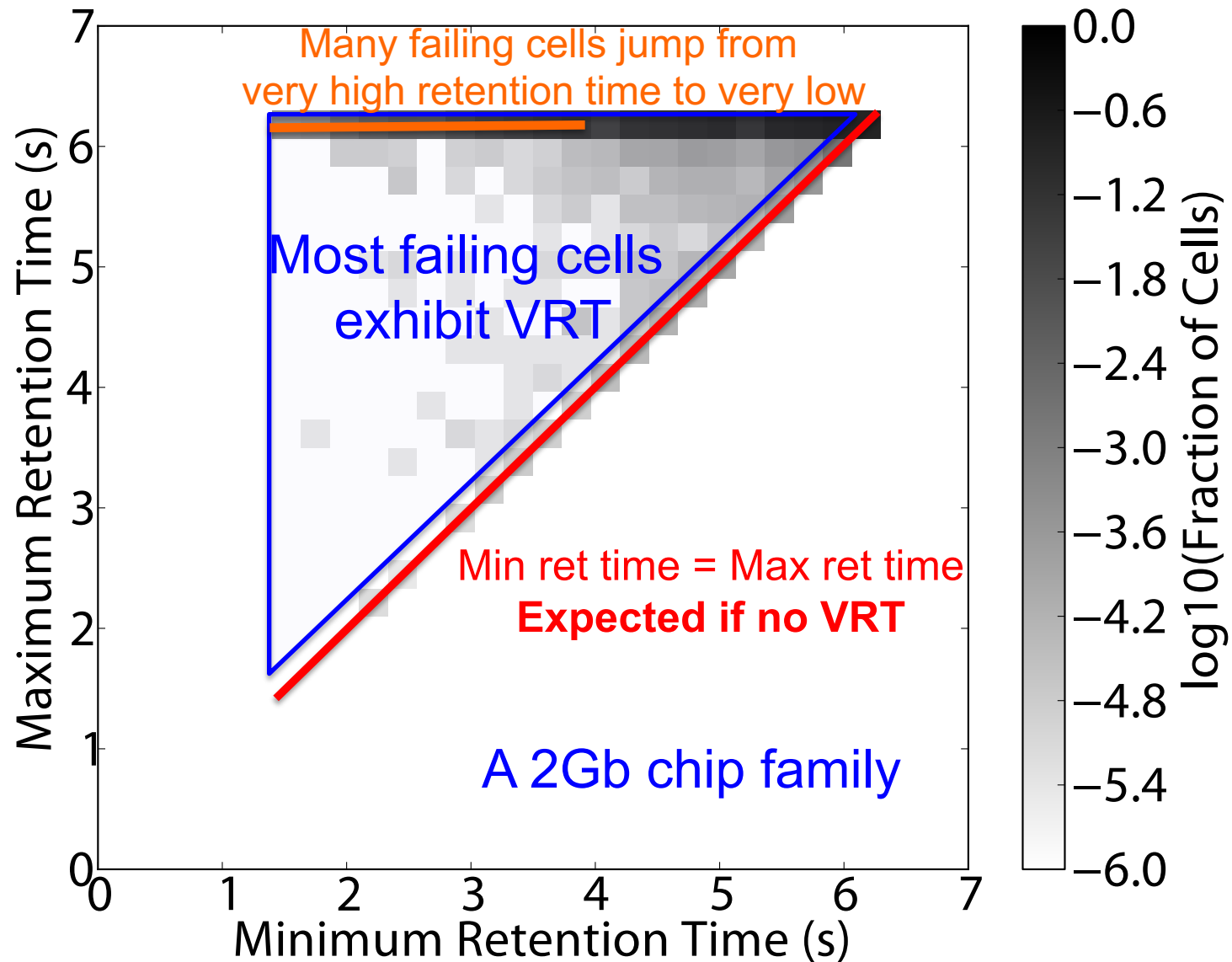


Newer device families have more weak cells than older ones
Likely a result of technology scaling

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

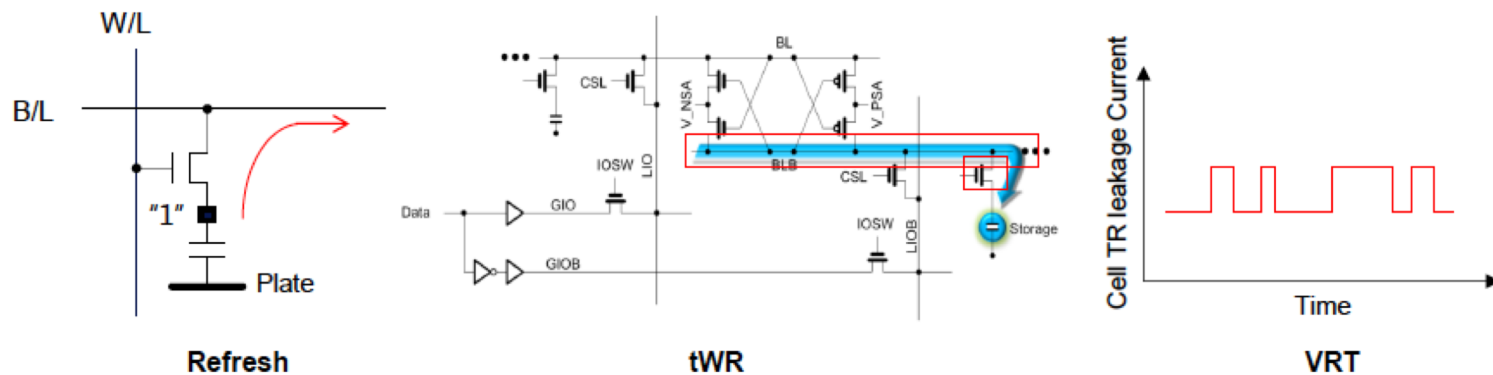
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

❖ 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

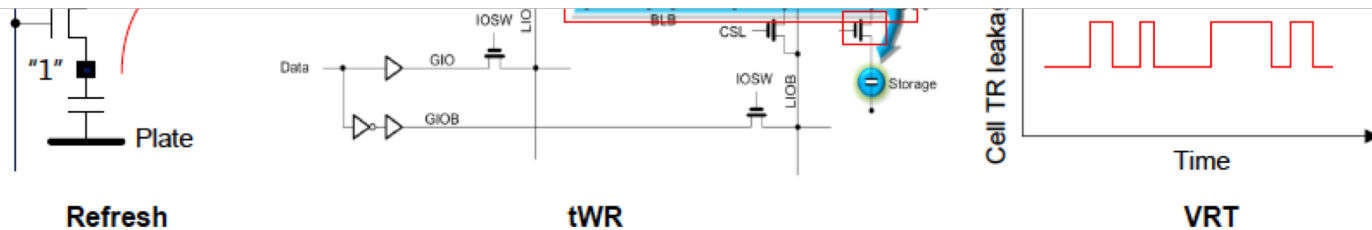
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

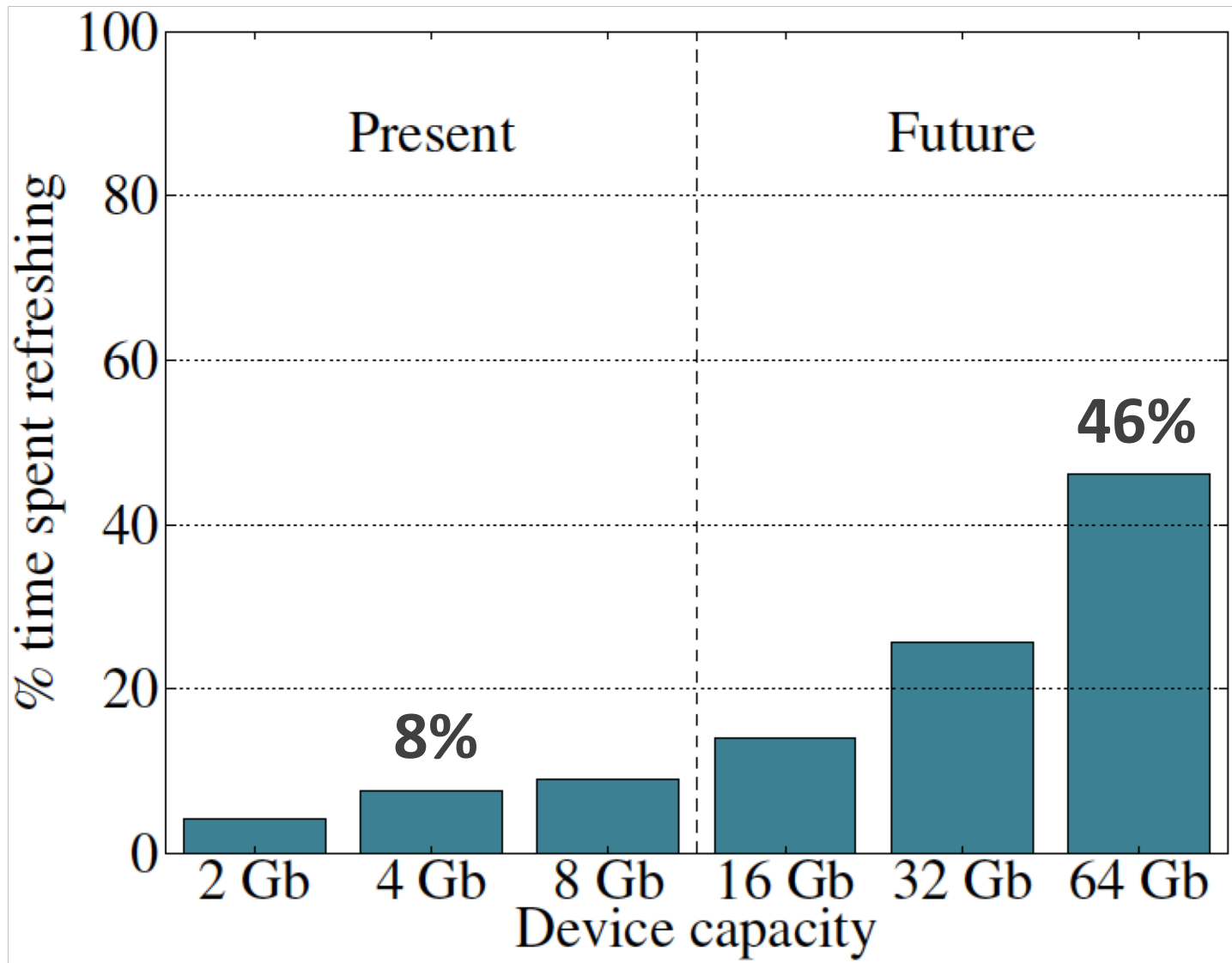
Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

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

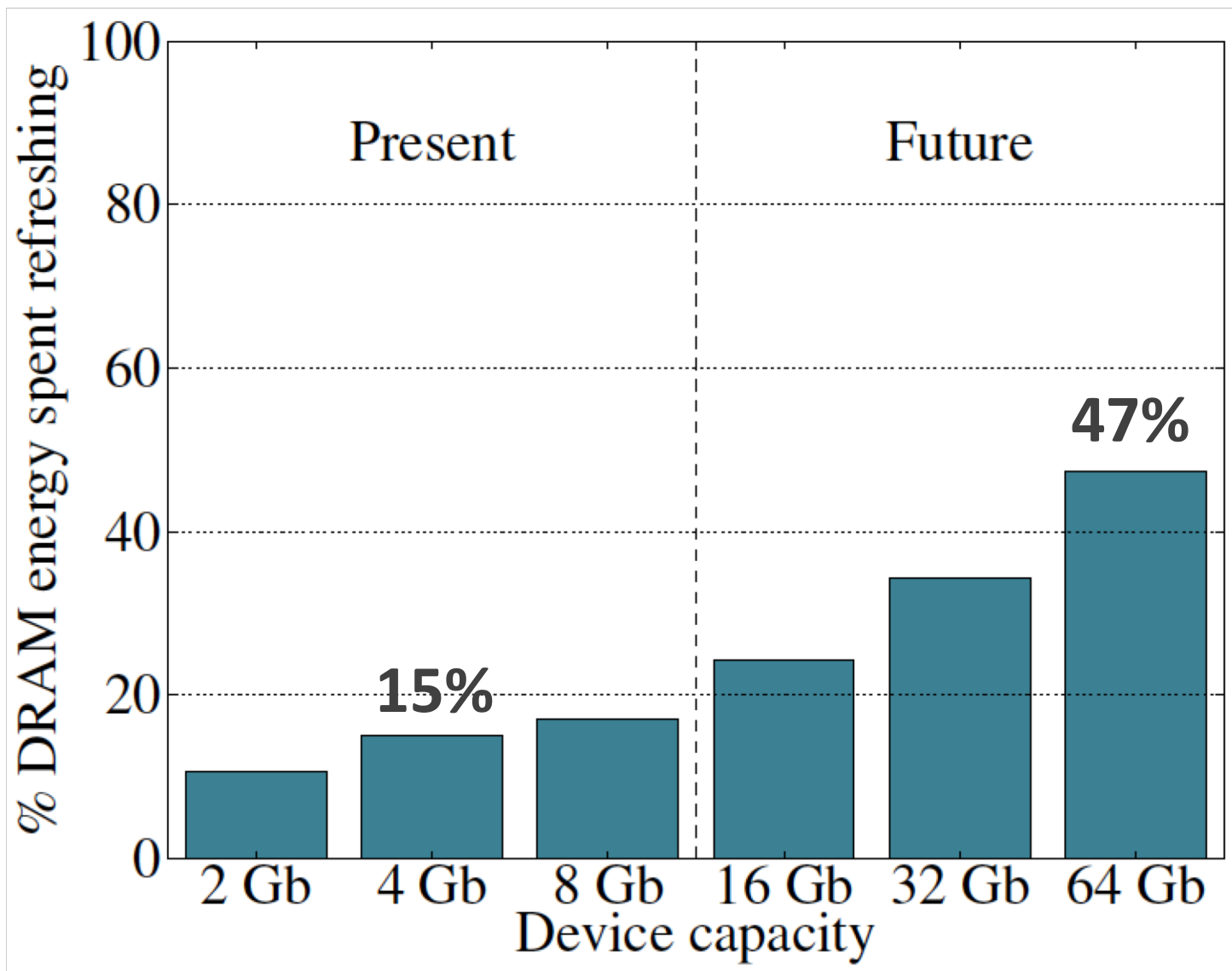
*Samsung Electronics, Hwasung, Korea / *Samsung Electronics, San Jose / **Intel*



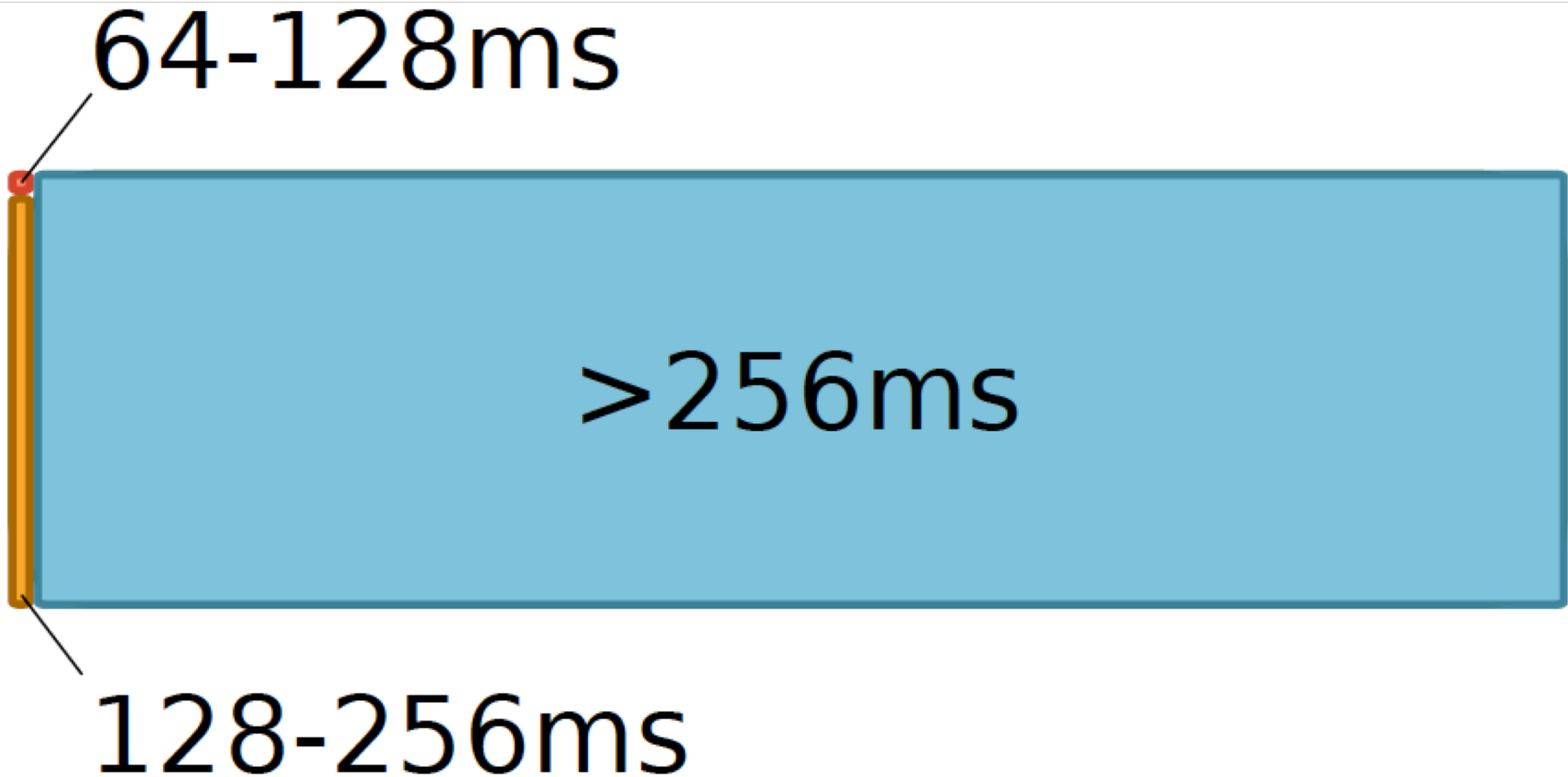
Refresh Overhead: Performance



Refresh Overhead: Energy



Retention Time Profile of DRAM



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

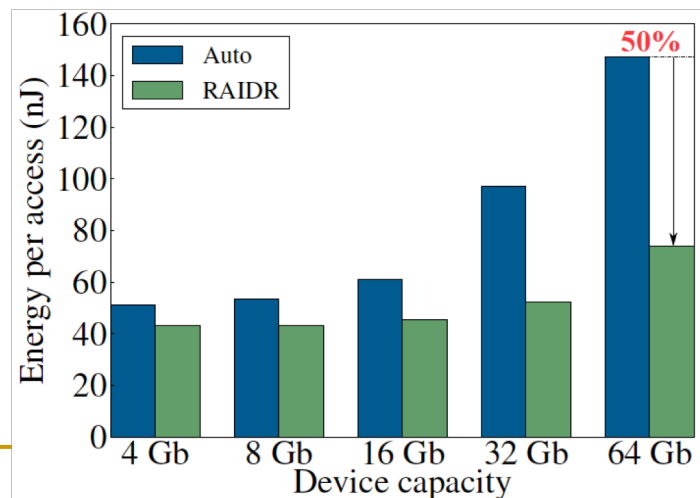
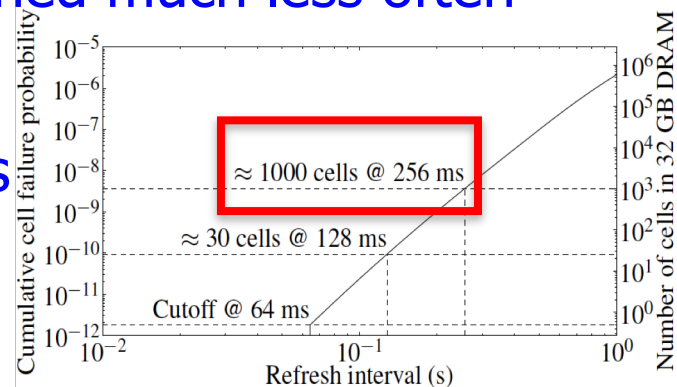
2. **Binning:** Store rows into bins by retention time in memory controller

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^{†*}
samirakhan@cmu.edu

Donghyuk Lee[†]
donghyuk1@cmu.edu

Yoongu Kim[†]
yoongukim@cmu.edu

Alaa R. Alameldeen^{*}
alaa.r.alameldeen@intel.com

Chris Wilkerson^{*}
chris.wilkerson@intel.com

Onur Mutlu[†]
onur@cmu.edu

[†]Carnegie Mellon University

^{*}Intel Labs

Towards an Online Profiling System

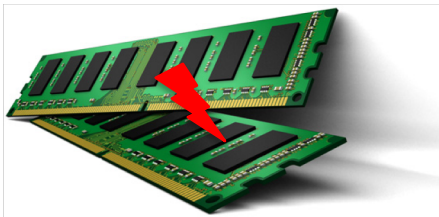
Key Observations:

- **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

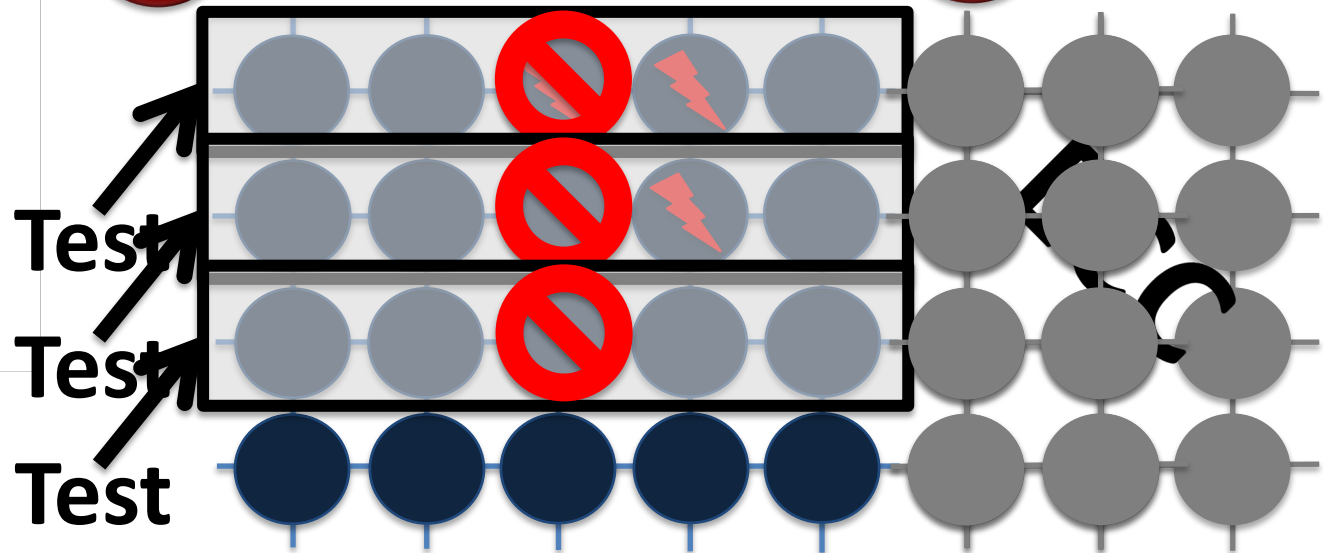
Initially Protect DRAM
with Strong ECC

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](#)]

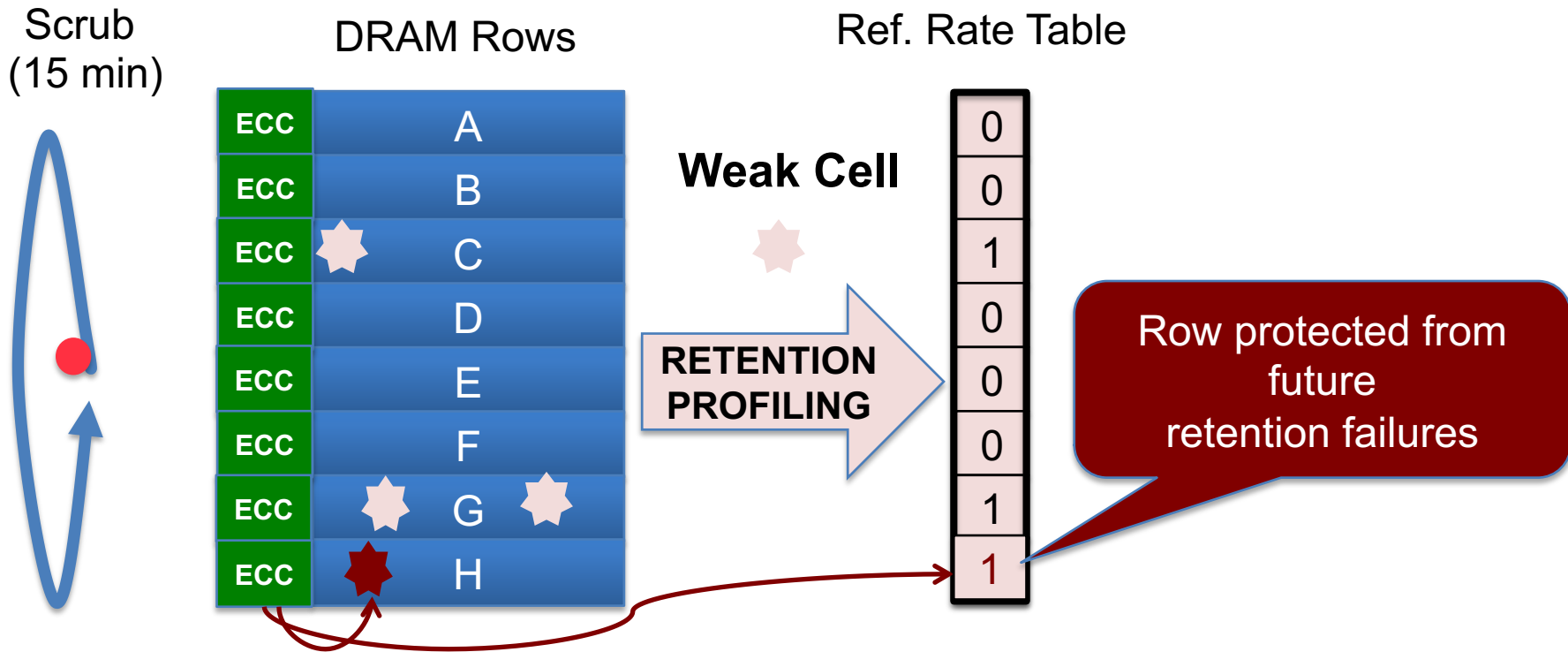
AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems

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

AVATAR

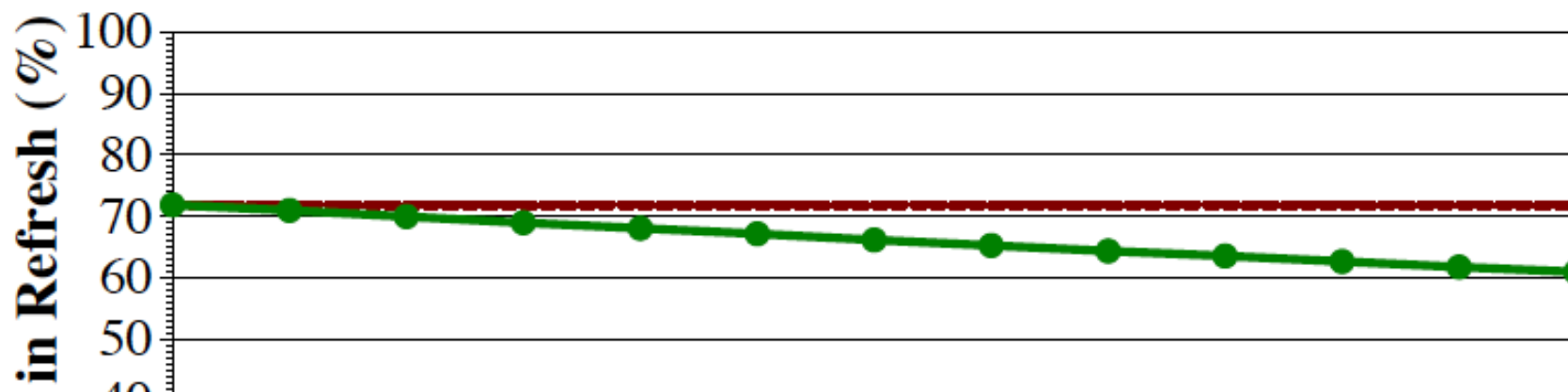
Insight: Avoid retention failures → Upgrade row on ECC error

Observation: Rate of VRT >> Rate of soft error (50x-2500x)

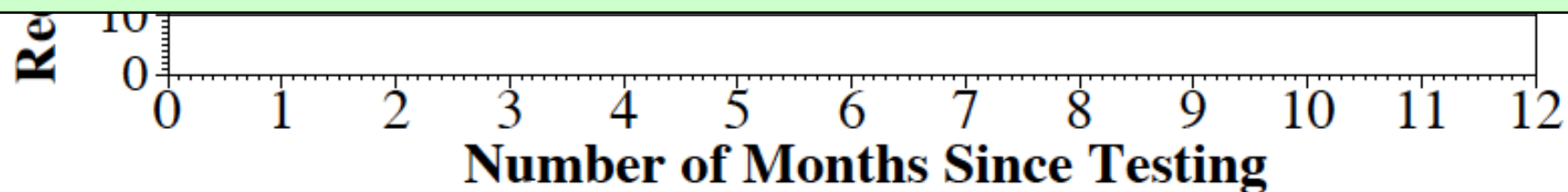


AVATAR mitigates VRT by increasing refresh rate on error

RESULTS: REFRESH SAVINGS

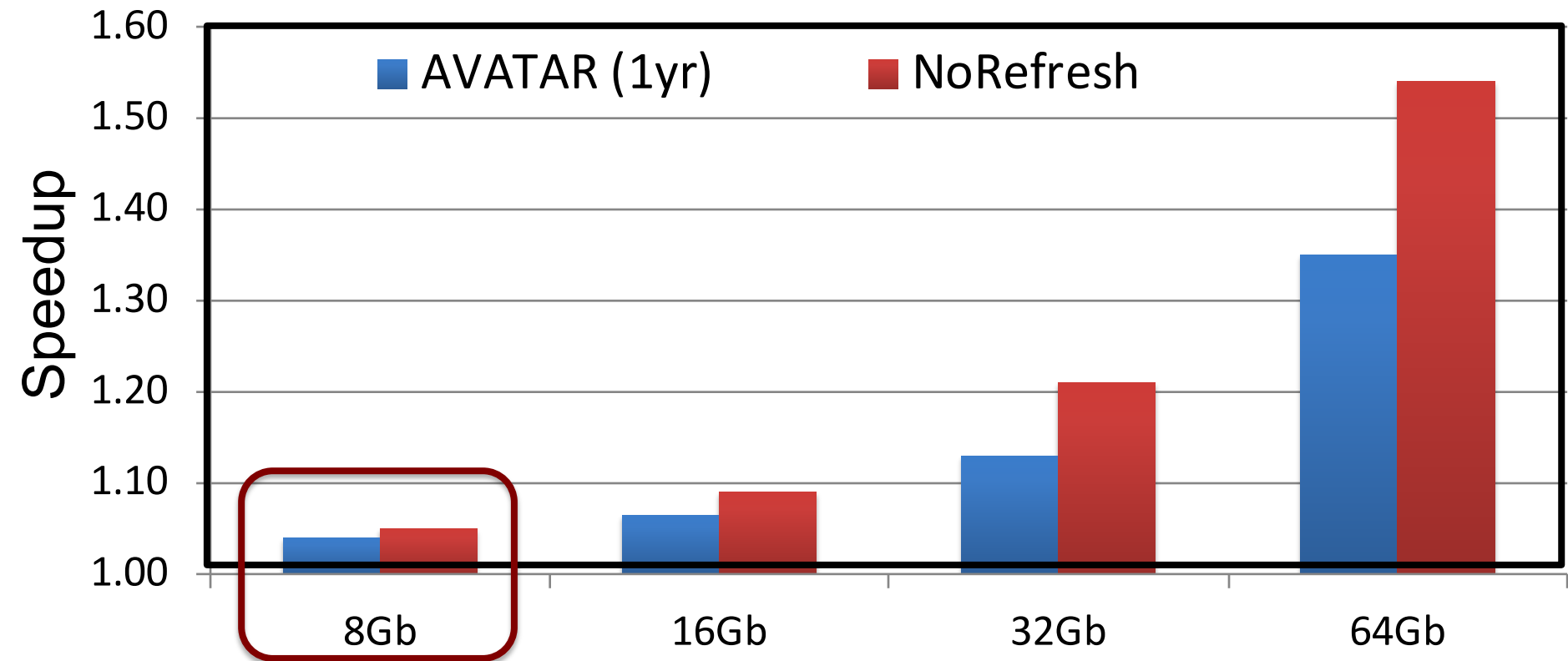


Retention Testing Once a Year can revert refresh saving from 60% to 70%



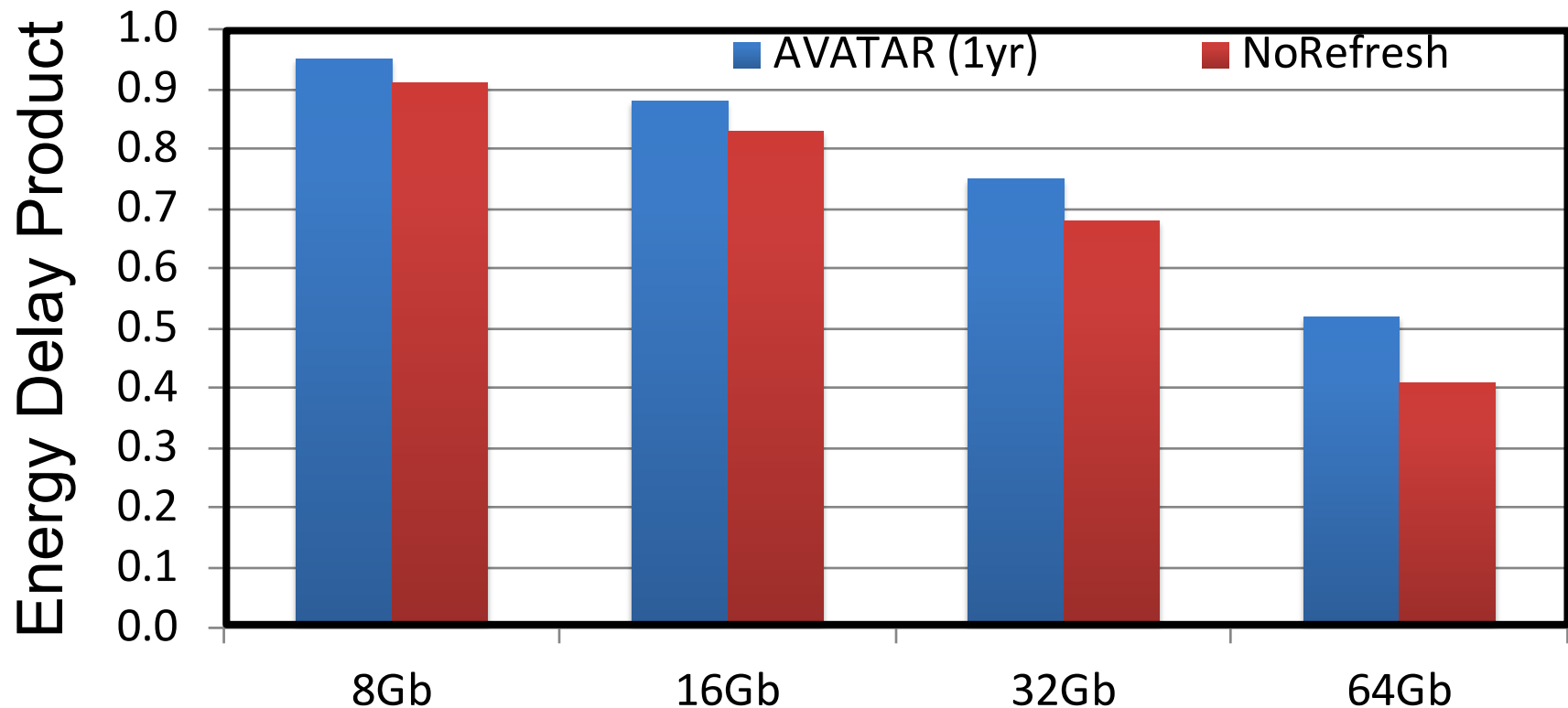
AVATAR reduces refresh by 60%-70%, similar to multi rate refresh but with VRT tolerance

SPEEDUP



AVATAR gets 2/3rd 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^{*}

^{*}University of Virginia

Donghyuk Lee^{†‡}

[†]Carnegie Mellon University

Onur Mutlu^{*†}

[‡]Nvidia

^{*}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^{*} Chris Wilkerson[†] Zhe Wang[†] Alaa R. Alameldeen[†] Donghyuk Lee[‡] Onur Mutlu^{*}
^{*}University of Virginia [†]Intel Labs [‡]Nvidia Research ^{*}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^{§‡} Jeremie S. Kim^{‡§} Onur Mutlu^{§‡}
[§]ETH Zürich [‡]Carnegie Mellon University

The Reach Profiler (REAPER):

Enabling the Mitigation of DRAM Retention Failures
via Profiling at Aggressive Conditions

Minesh Patel

Jeremie S. Kim

Onur Mutlu

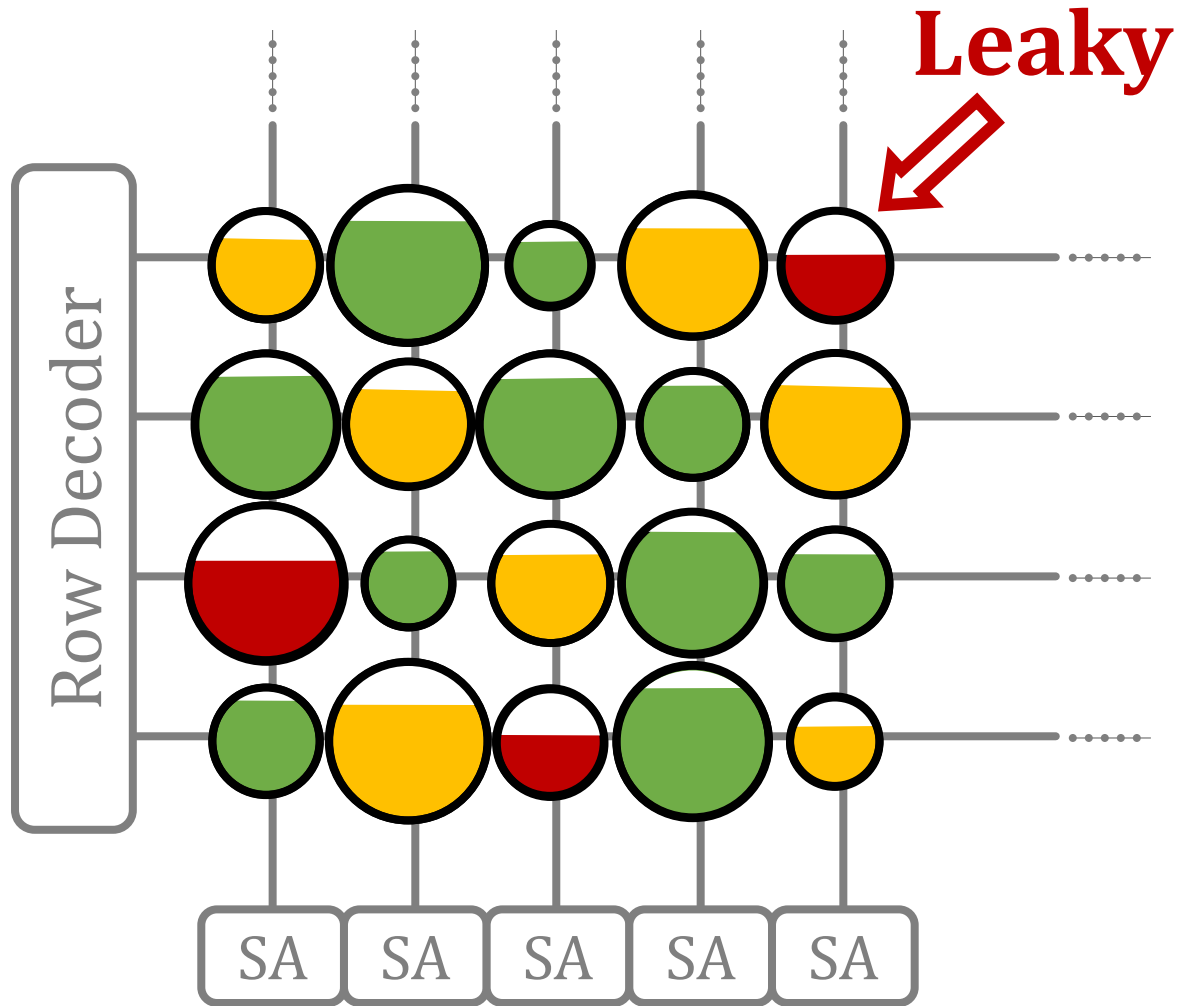
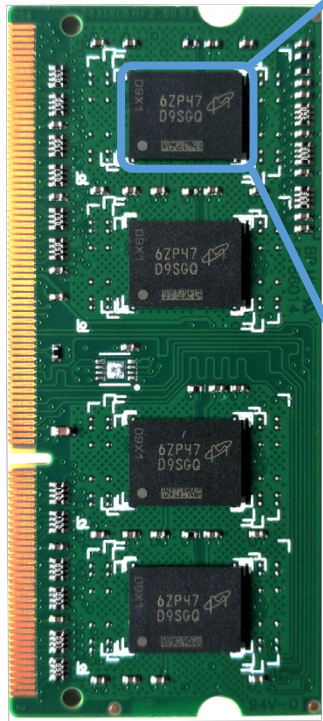


SAFARI

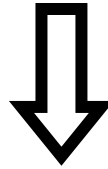
ETH zürich

Carnegie Mellon

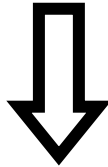
DRAM



Leaky Cells

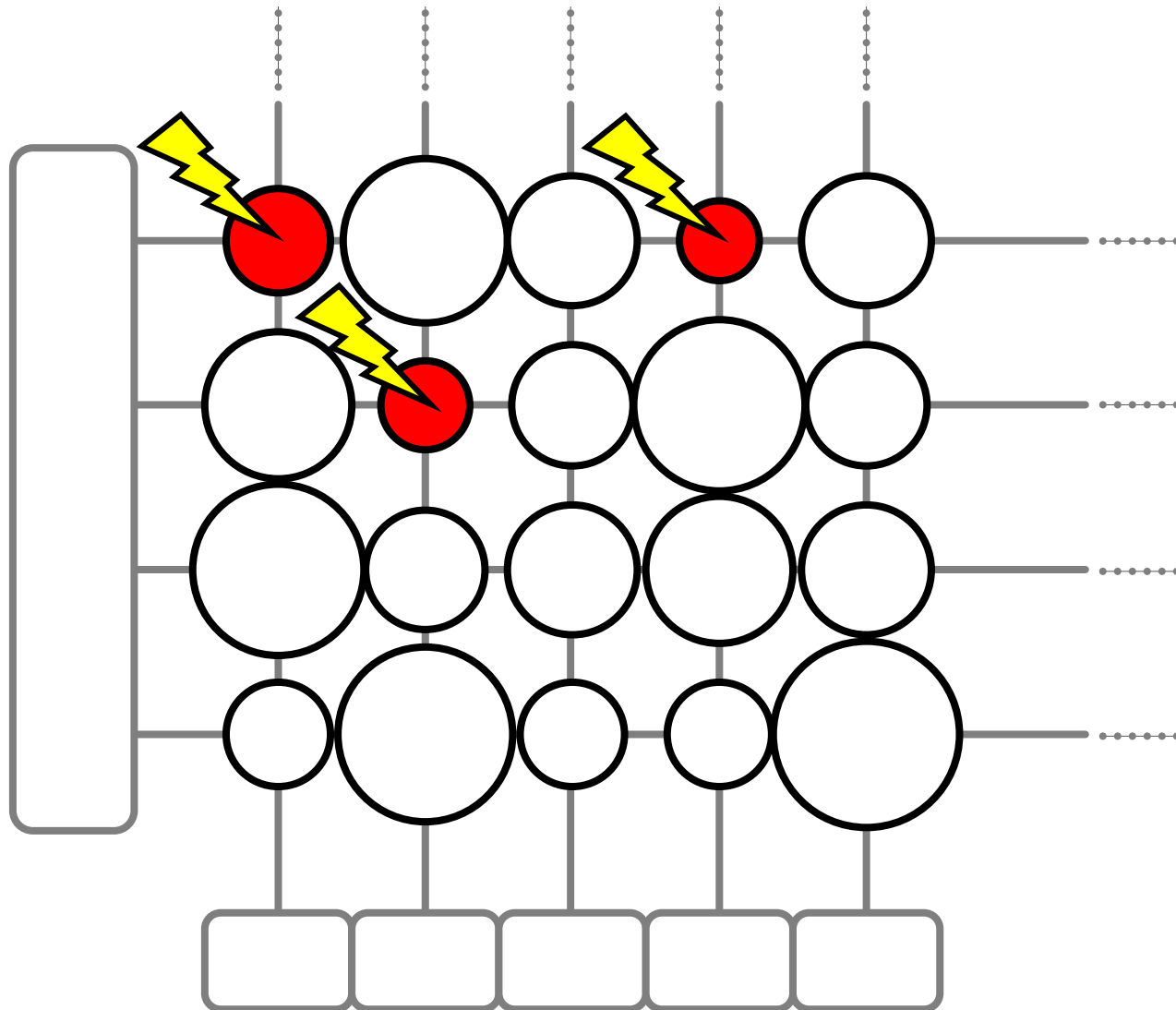


Periodic DRAM Refresh



Performance + Energy Overhead

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



Process, voltage, temperature

Variable retention time

Data pattern dependence

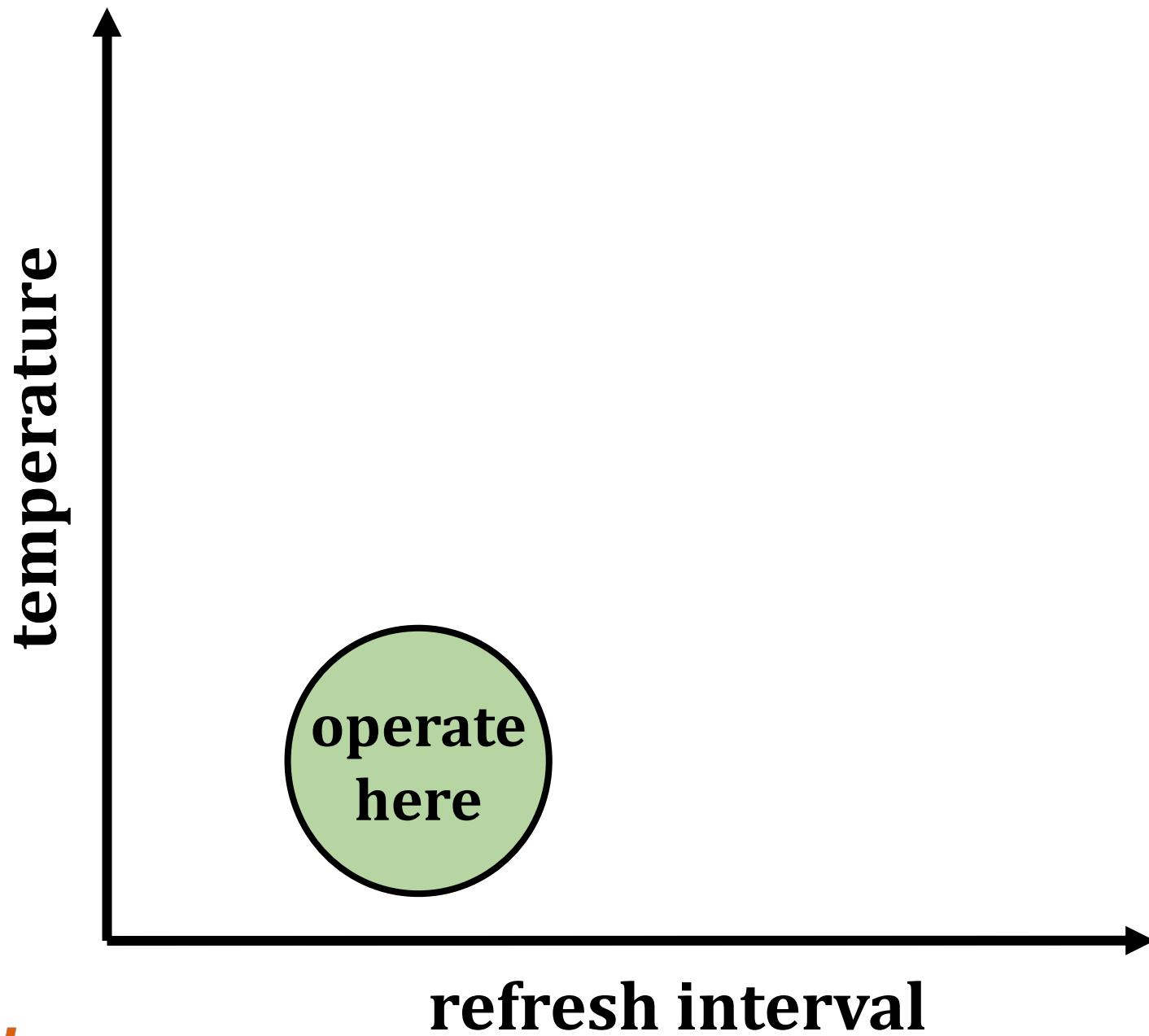
Characterization of 368 LPDDR4 DRAM Chips

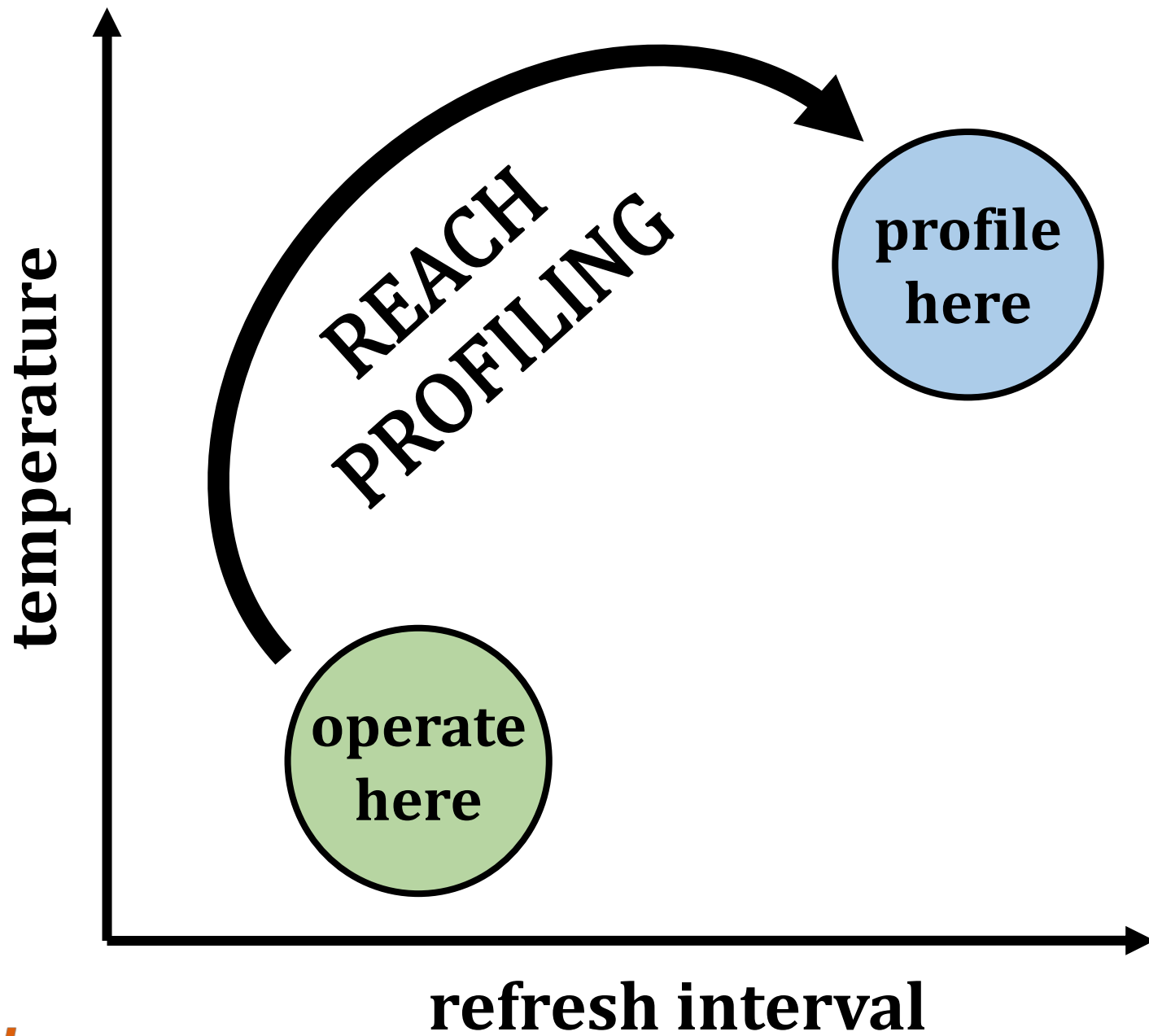
①

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**

REAPER Outline

1. DRAM Refresh Background

2. Failure Profiling Challenges

3. Current Approaches

4. LPDDR4 Characterization

5. Reach Profiling

6. End-to-end Evaluation

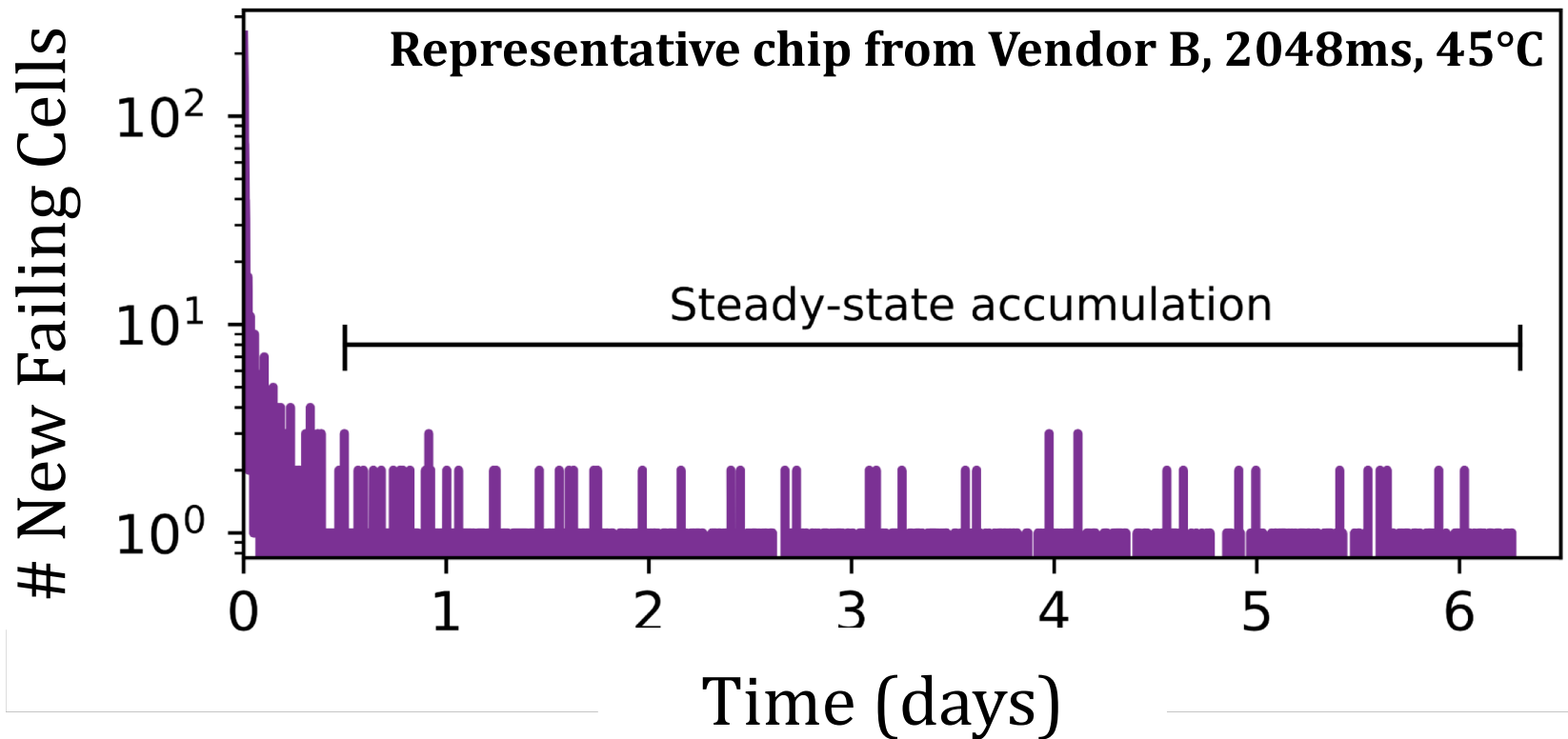
Experimental Infrastructure

- **368 2y-nm LPDDR4 DRAM chips**
 - 4Gb chip size
 - From 3 major DRAM vendors
- **Thermally controlled testing chamber**
 - Ambient temperature range: $\{40^{\circ}\text{C} - 55^{\circ}\text{C}\} \pm 0.25^{\circ}\text{C}$
 - DRAM temperature is held at 15°C above ambient

LPDDR4 Studies

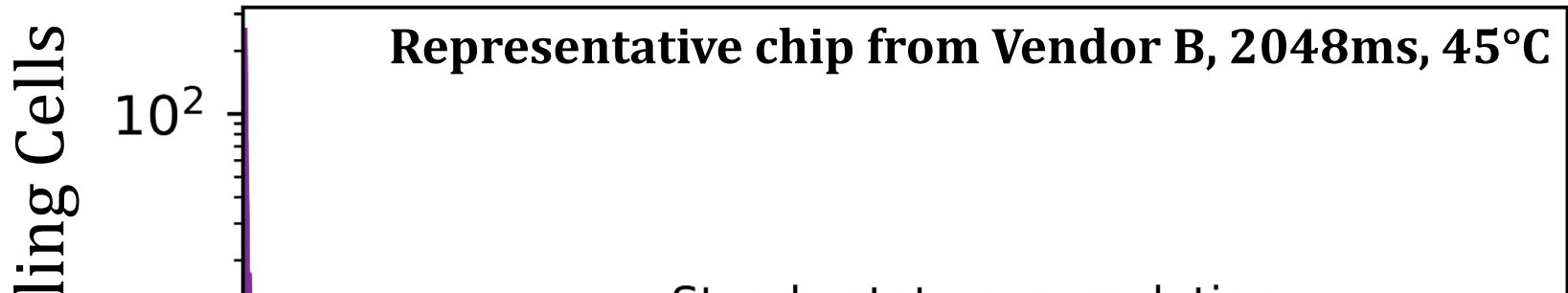
1. Temperature
2. Data Pattern Dependence
3. Retention Time Distributions
- 4. Variable Retention Time**
- 5. Individual Cell Characterization**

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

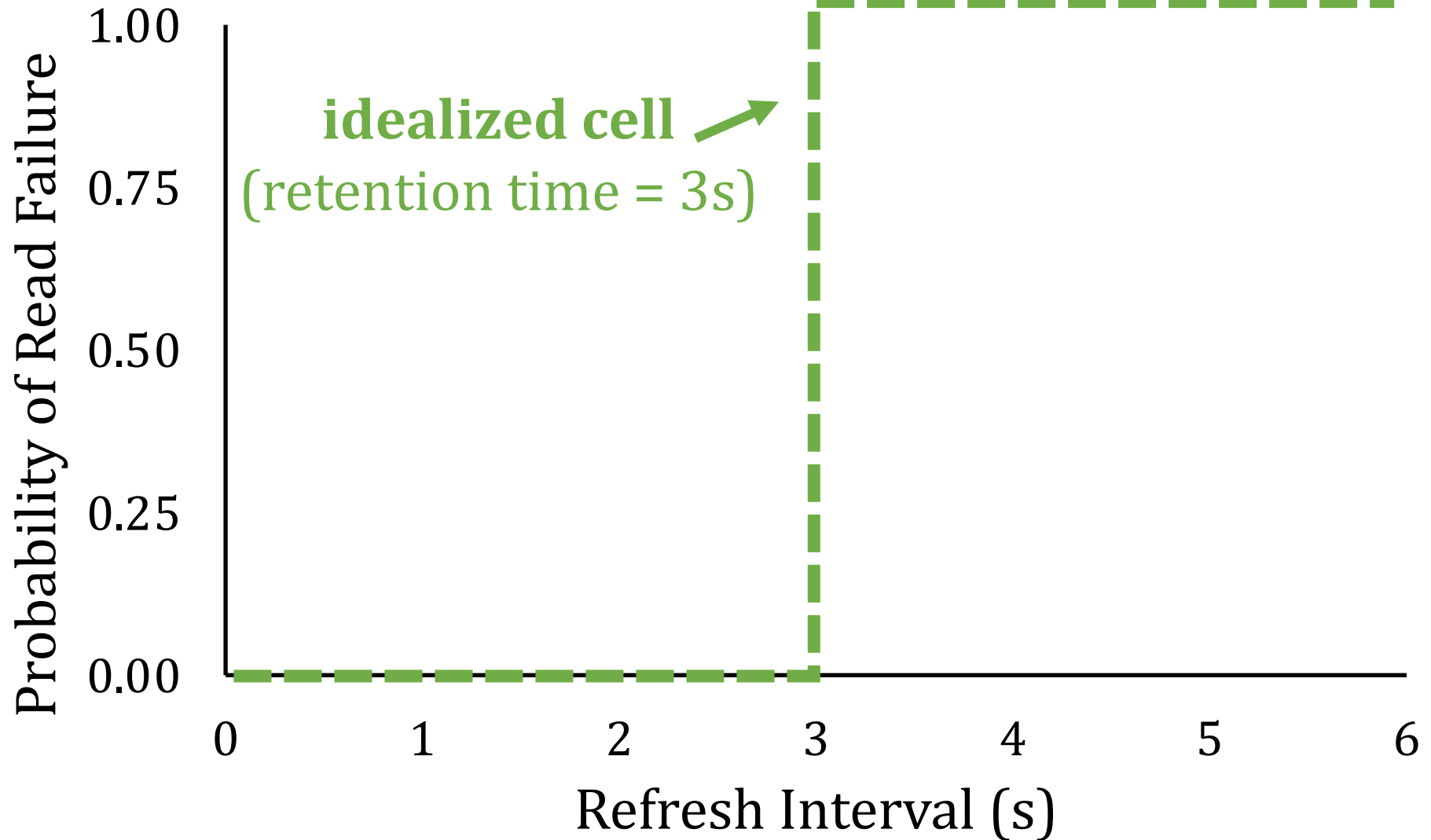
Long-term Continuous Profiling



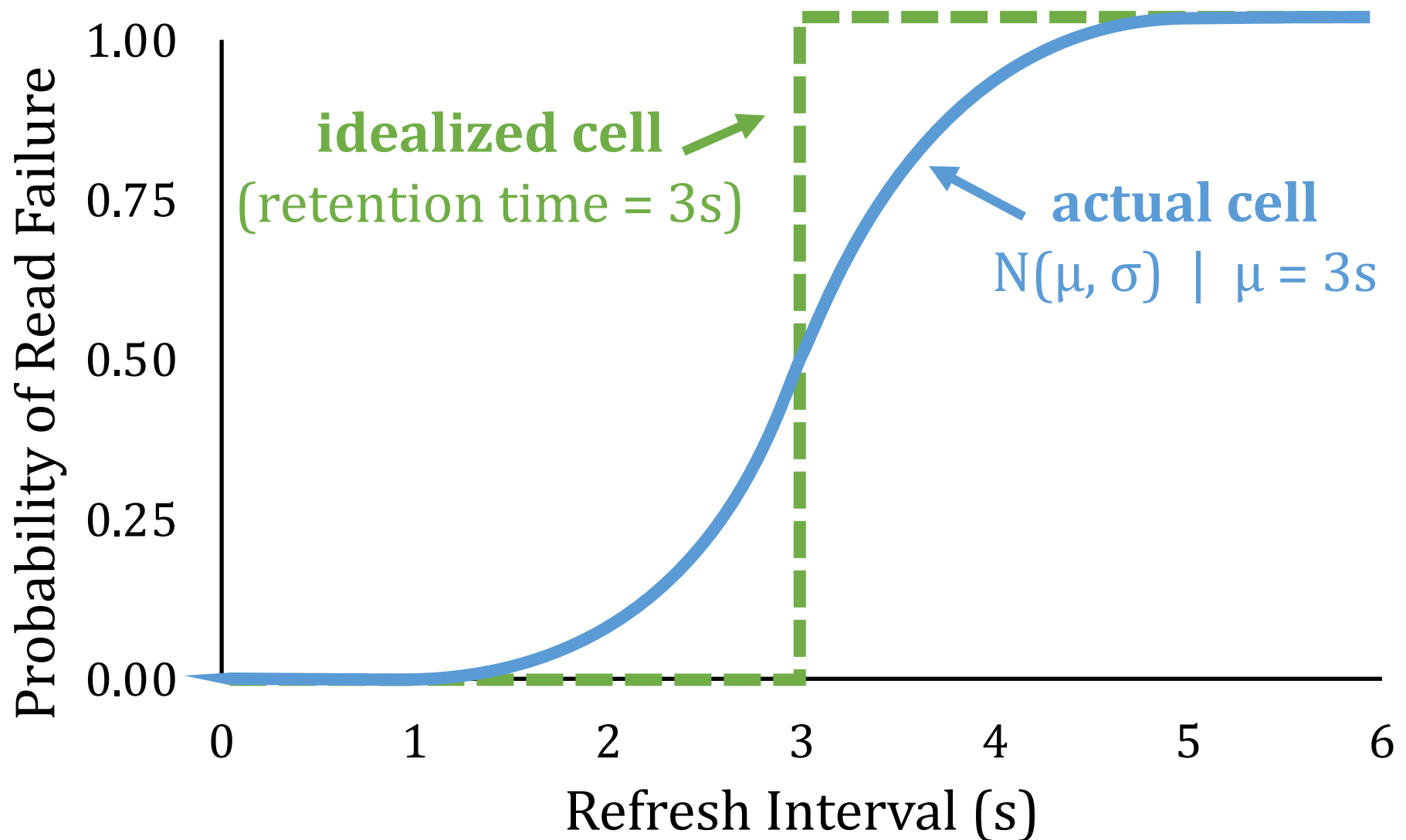
Error correction codes (ECC)
and online profiling are *necessary*
to manage new failing cells

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

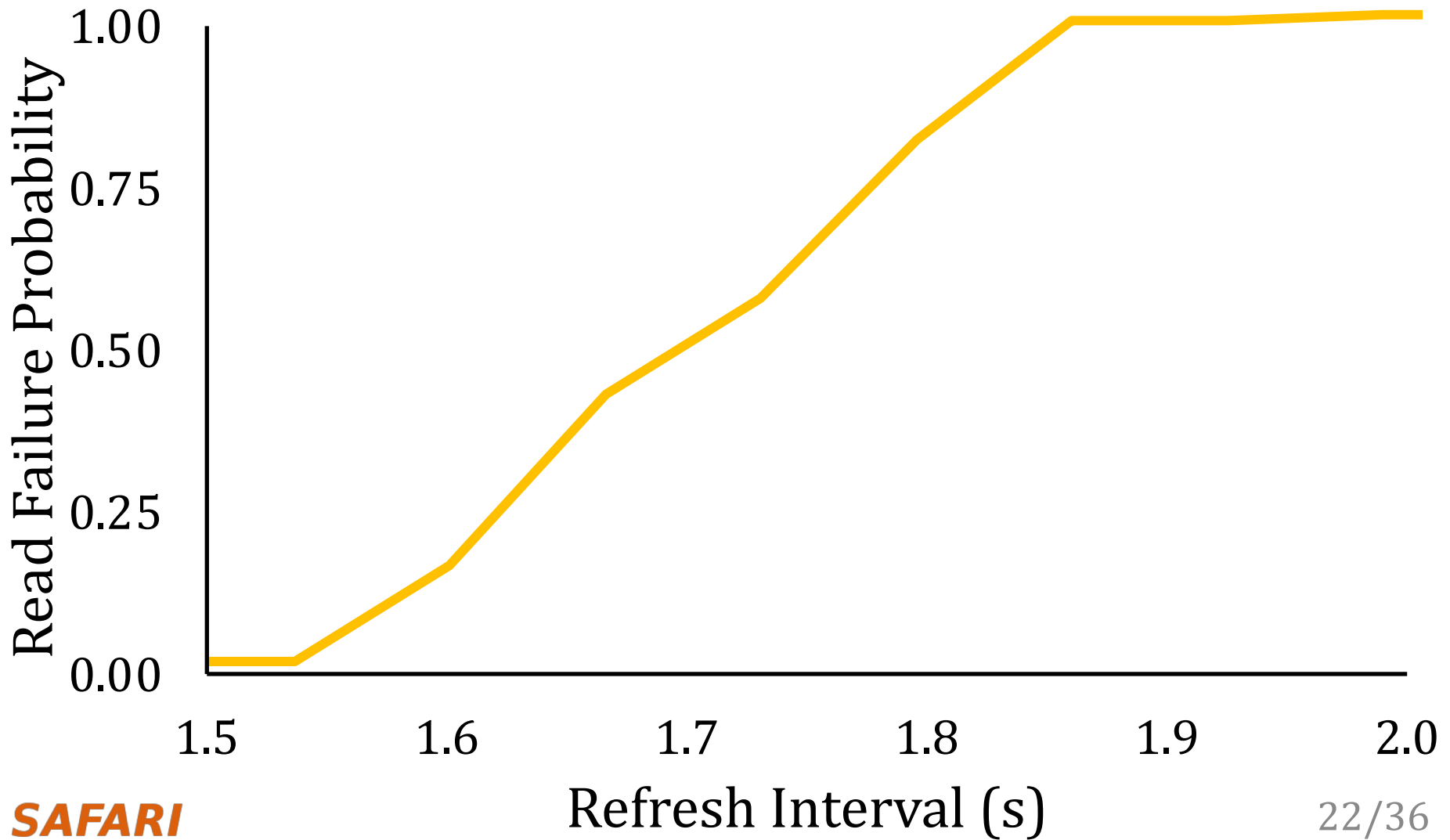
Single-cell Failure Probability (Cartoon)



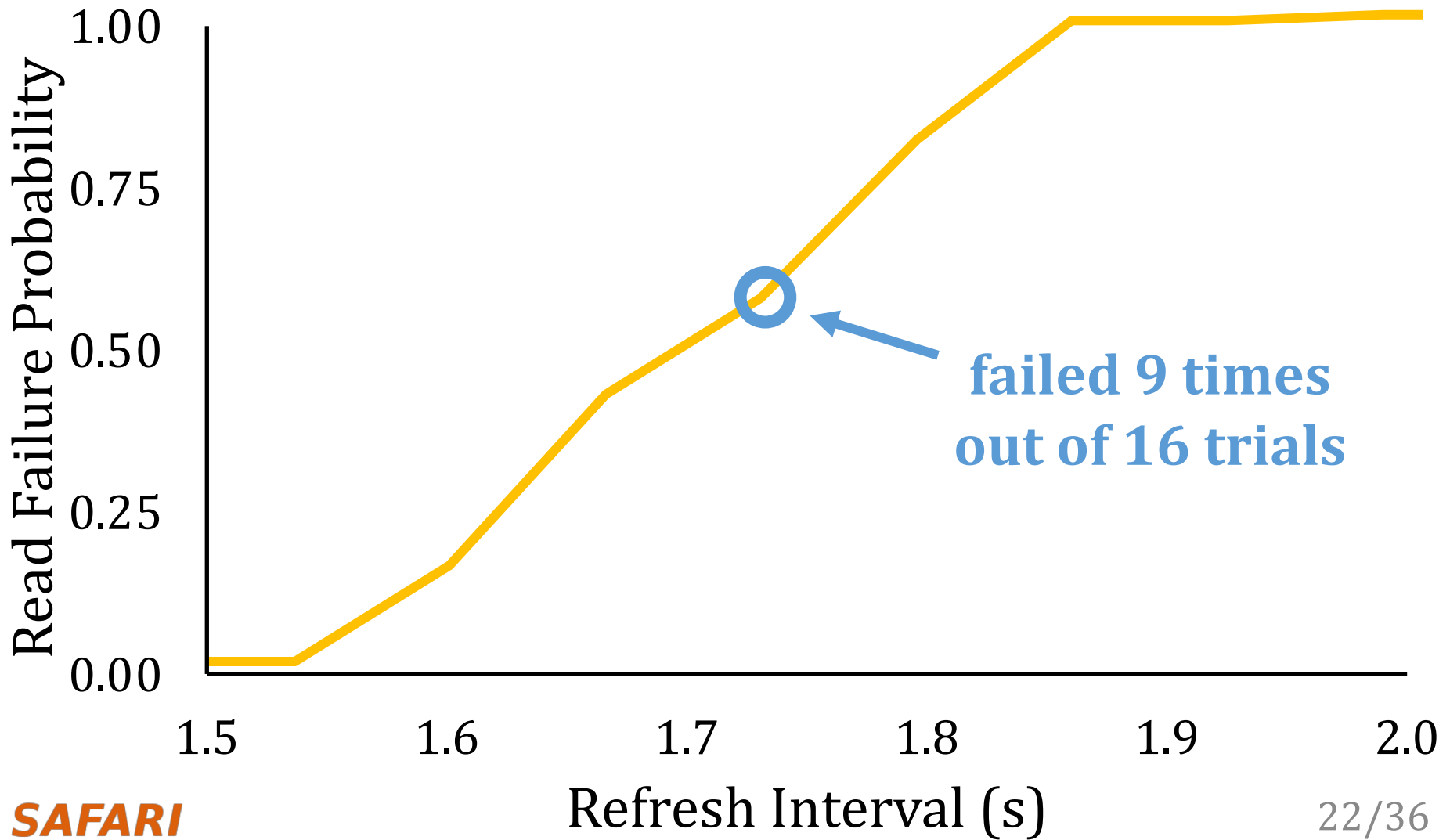
Single-cell Failure Probability (Cartoon)



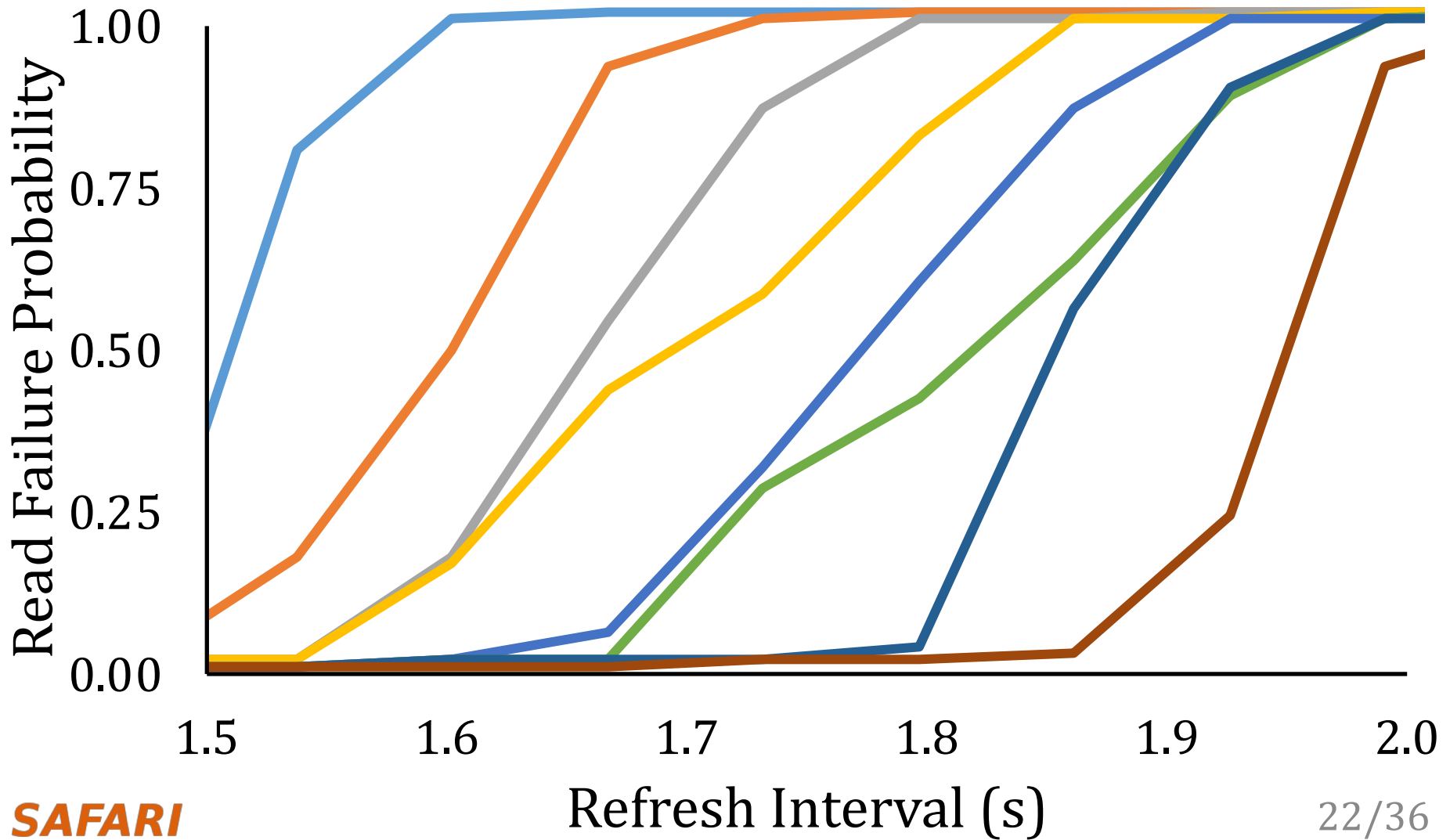
Single-cell Failure Probability (Real)



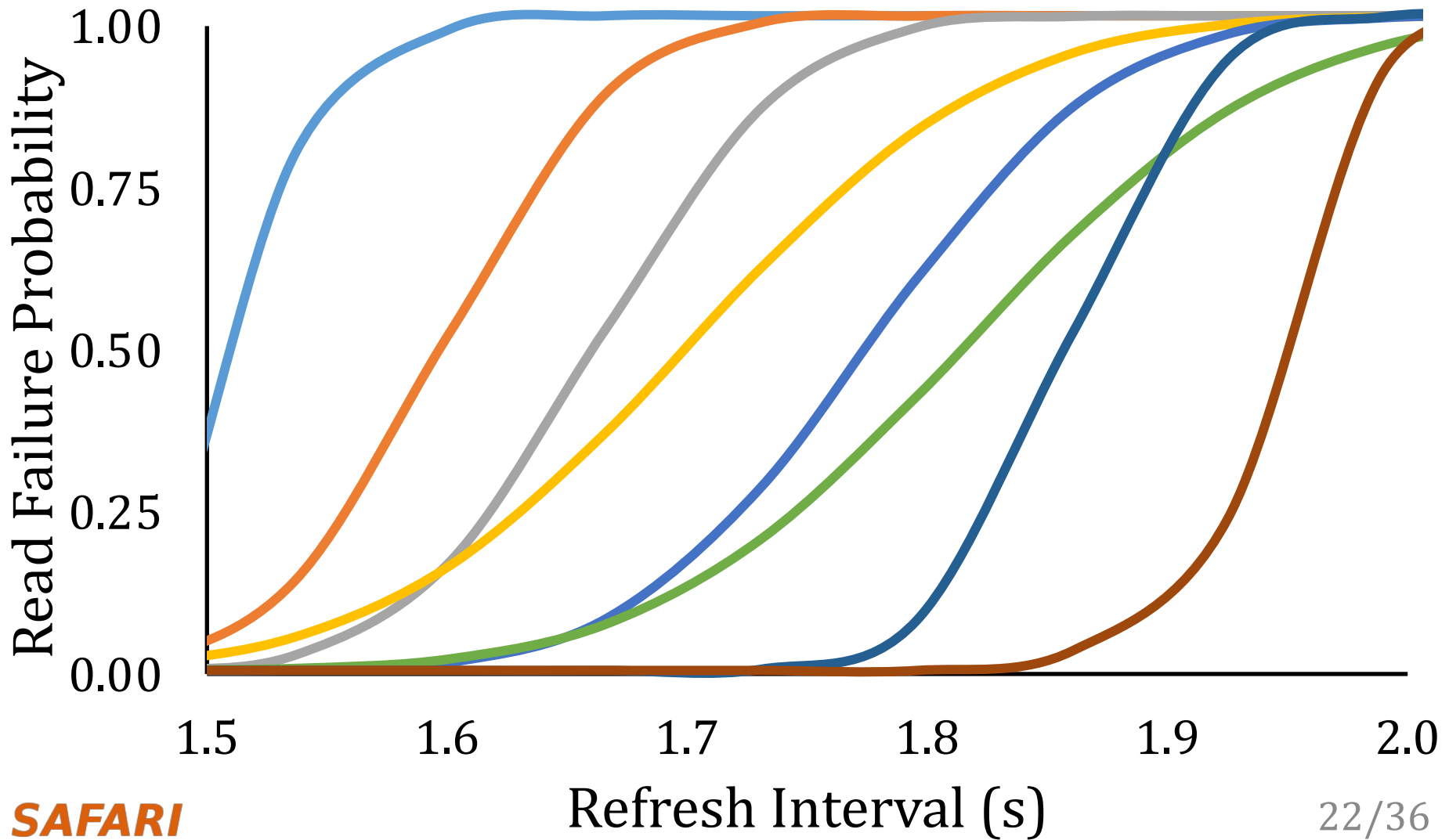
Single-cell Failure Probability (Real)



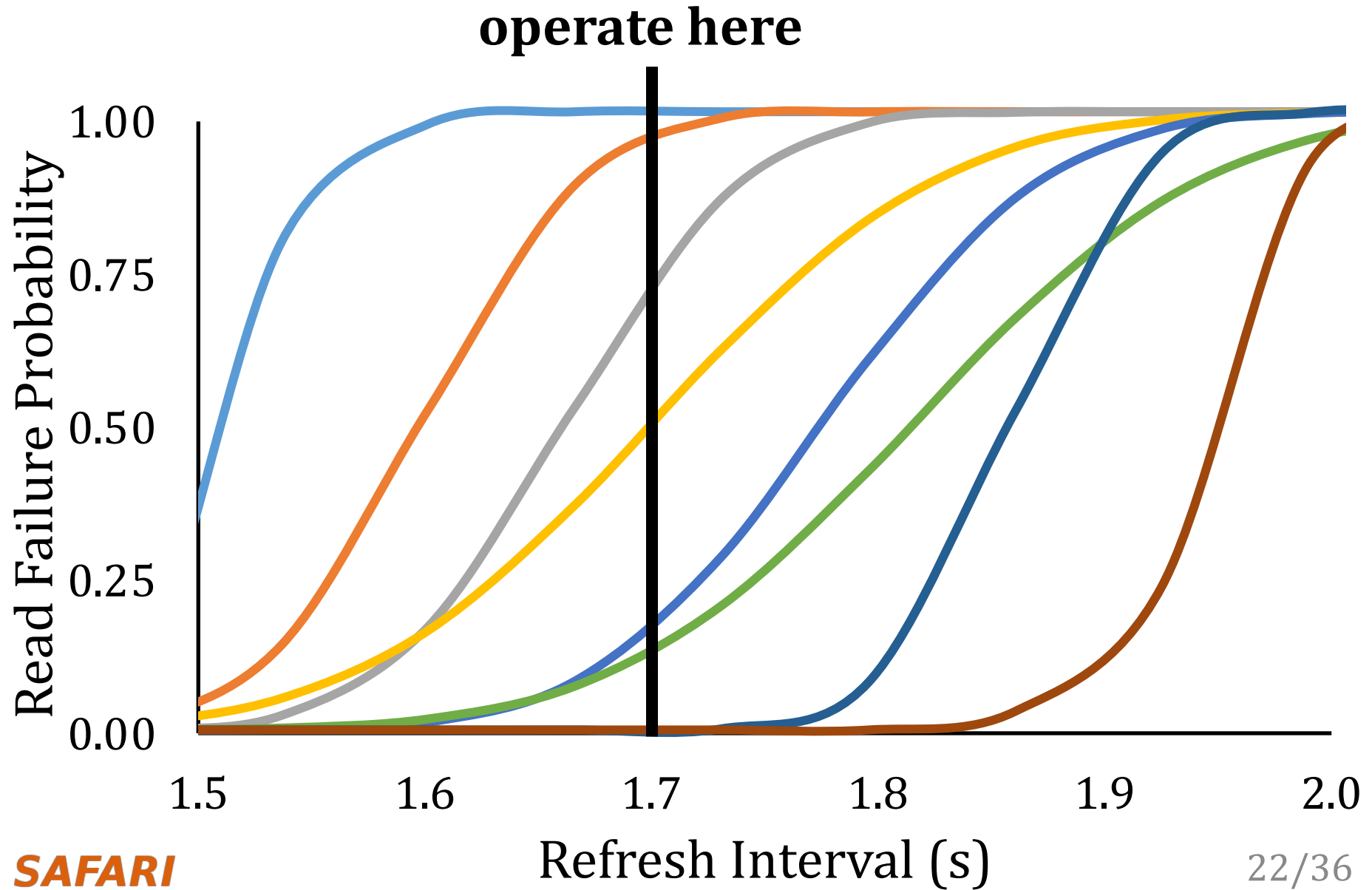
Single-cell Failure Probability (Real)



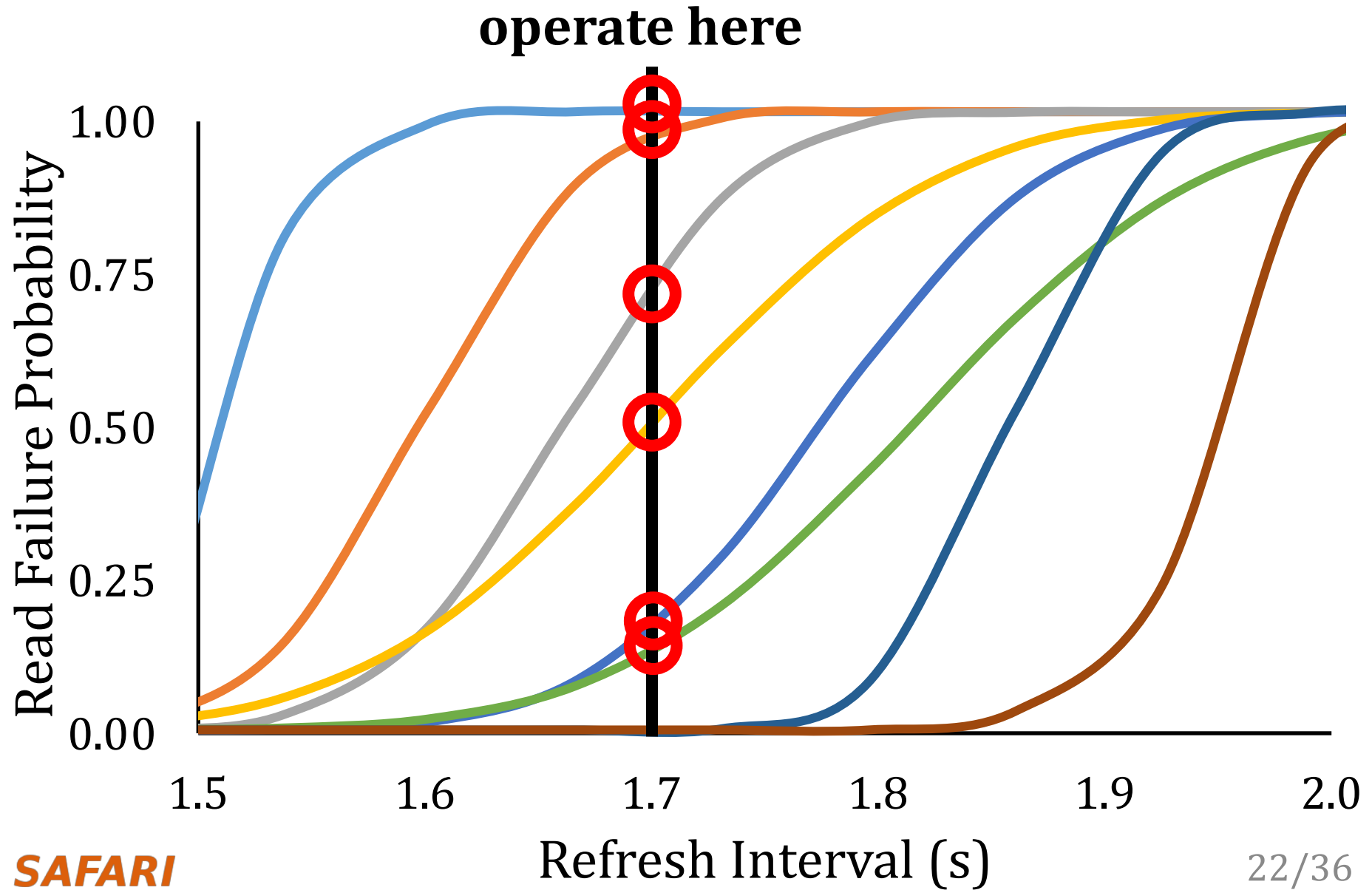
Single-cell Failure Probability (Real)



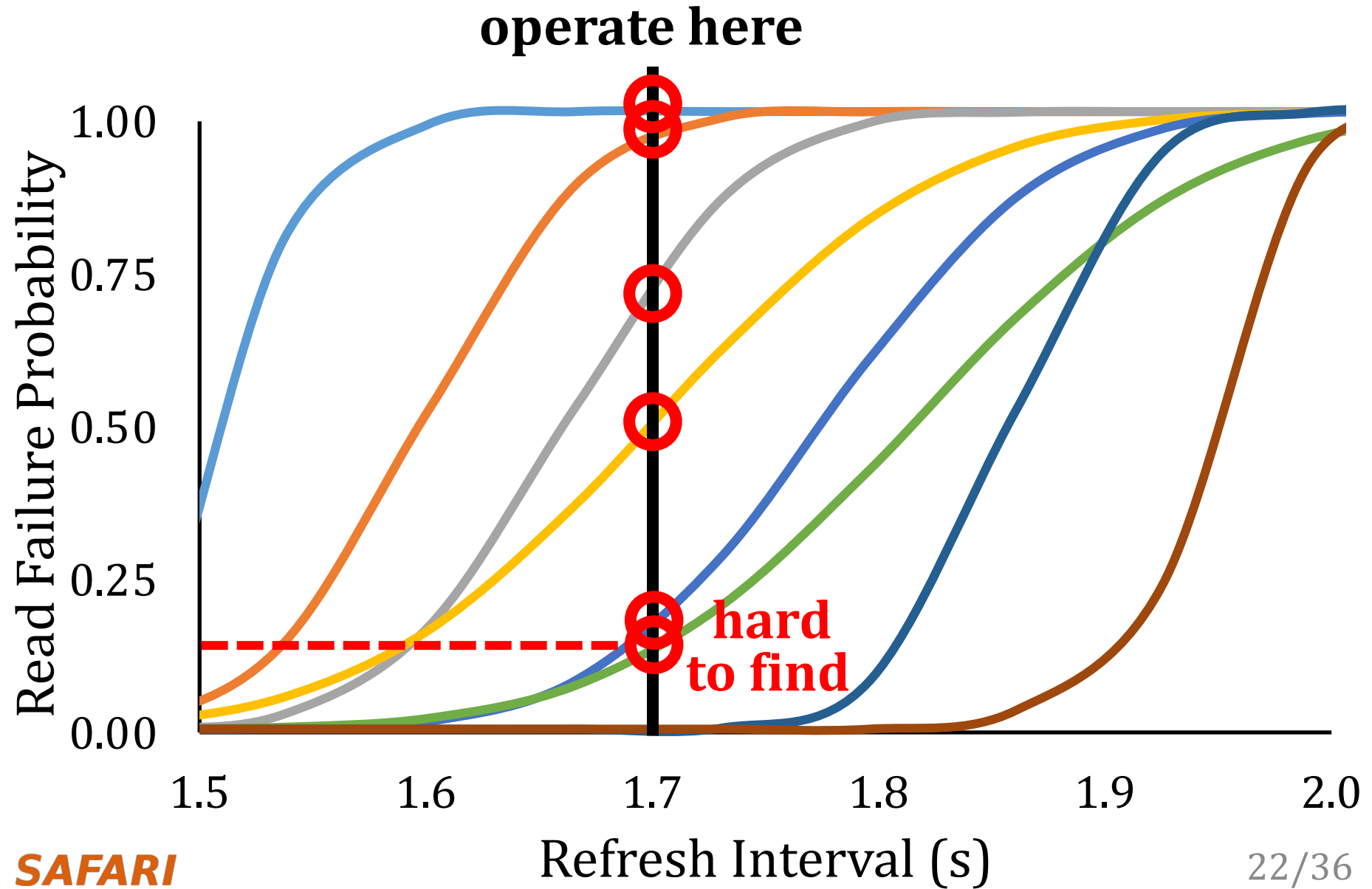
Single-cell Failure Probability (Real)



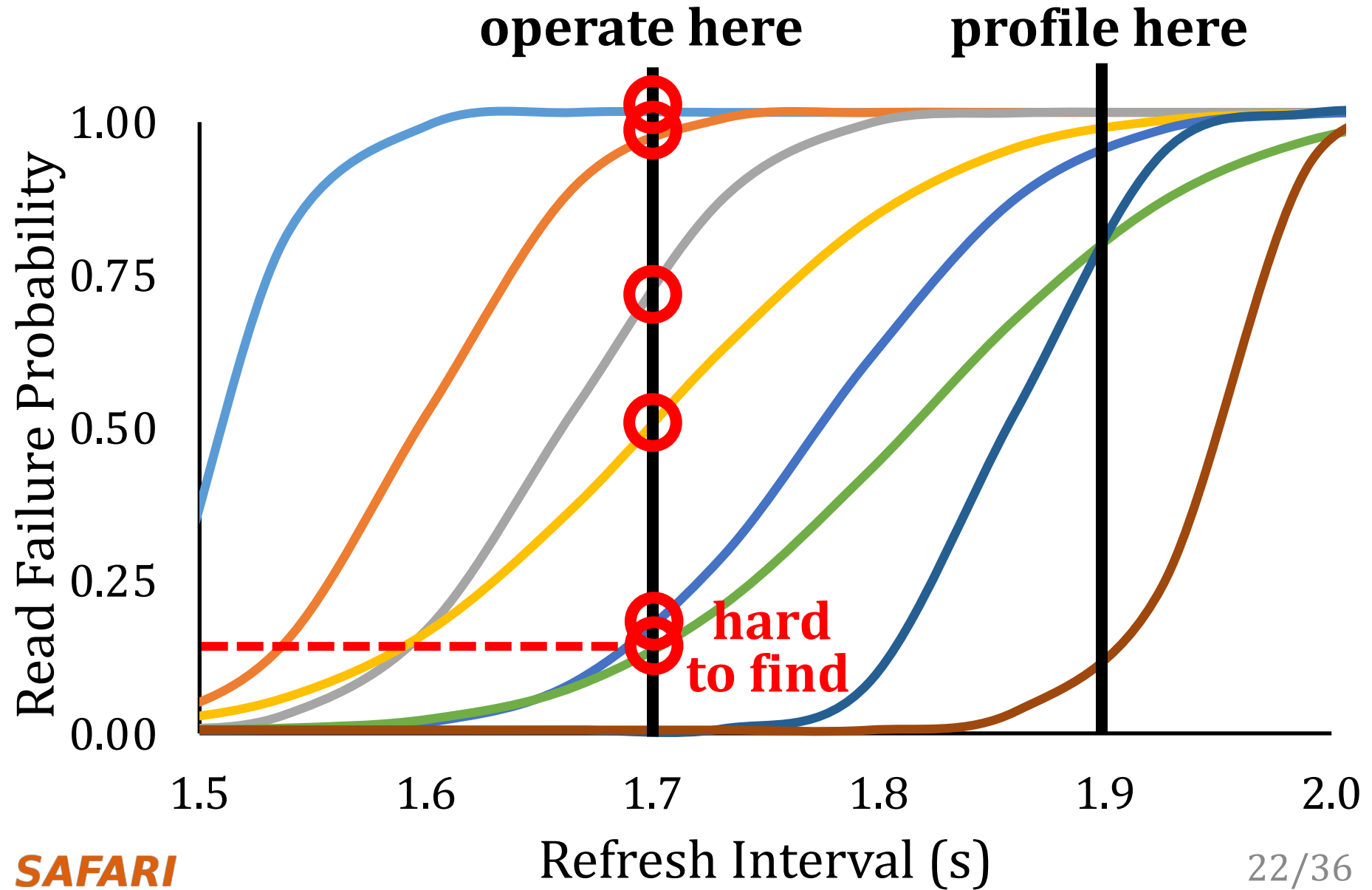
Single-cell Failure Probability (Real)



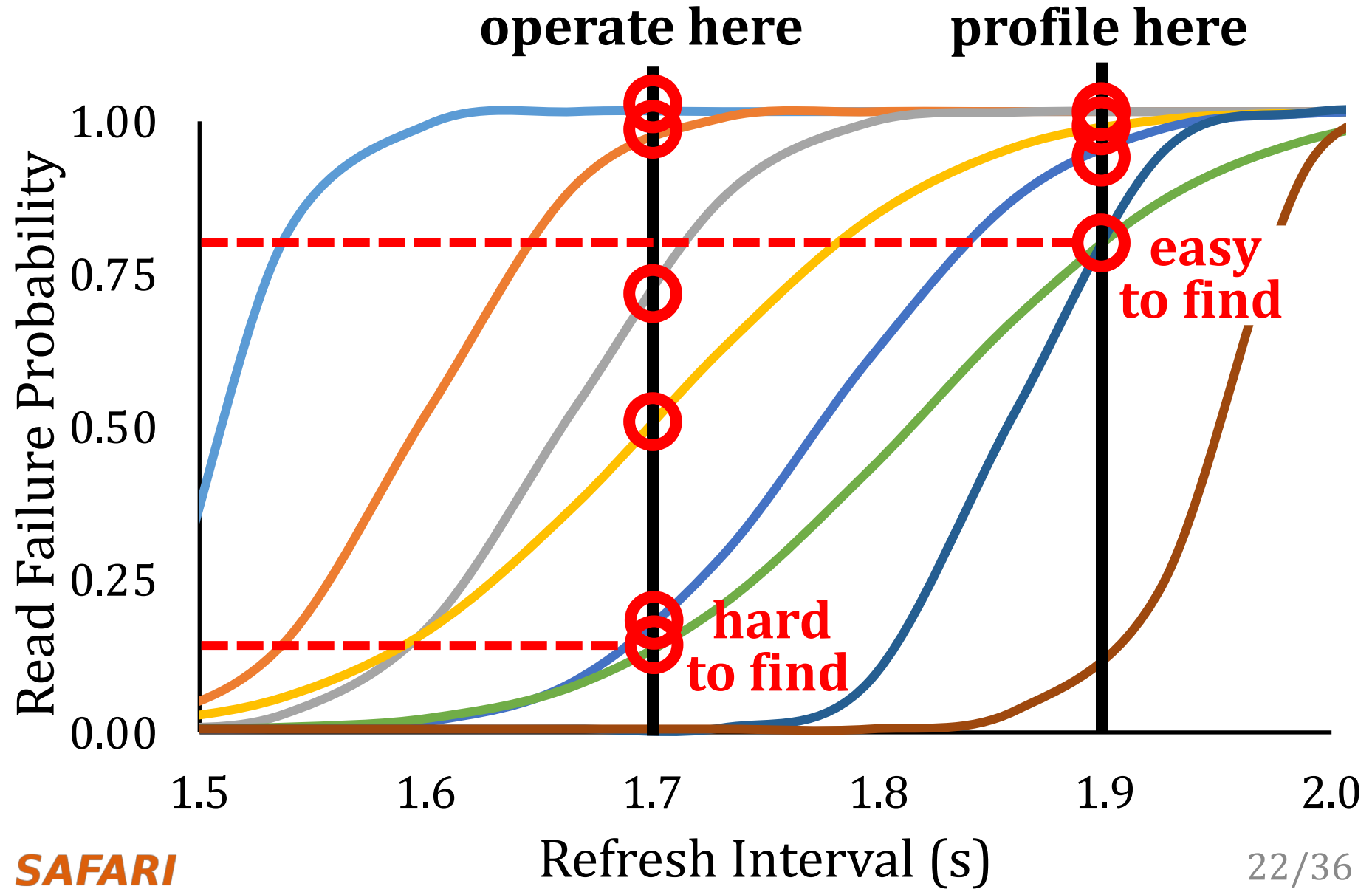
Single-cell Failure Probability (Real)



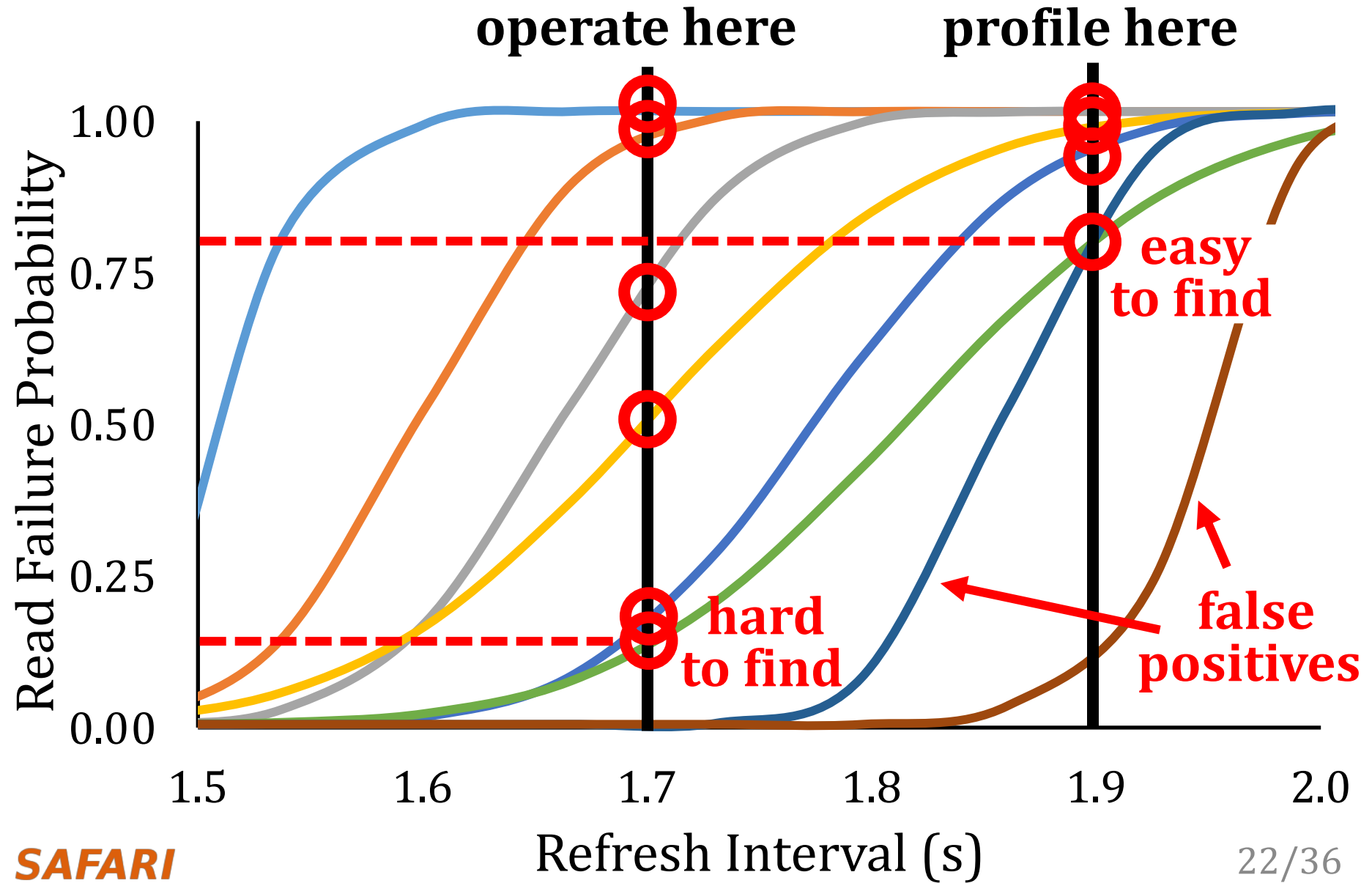
Single-cell Failure Probability (Real)



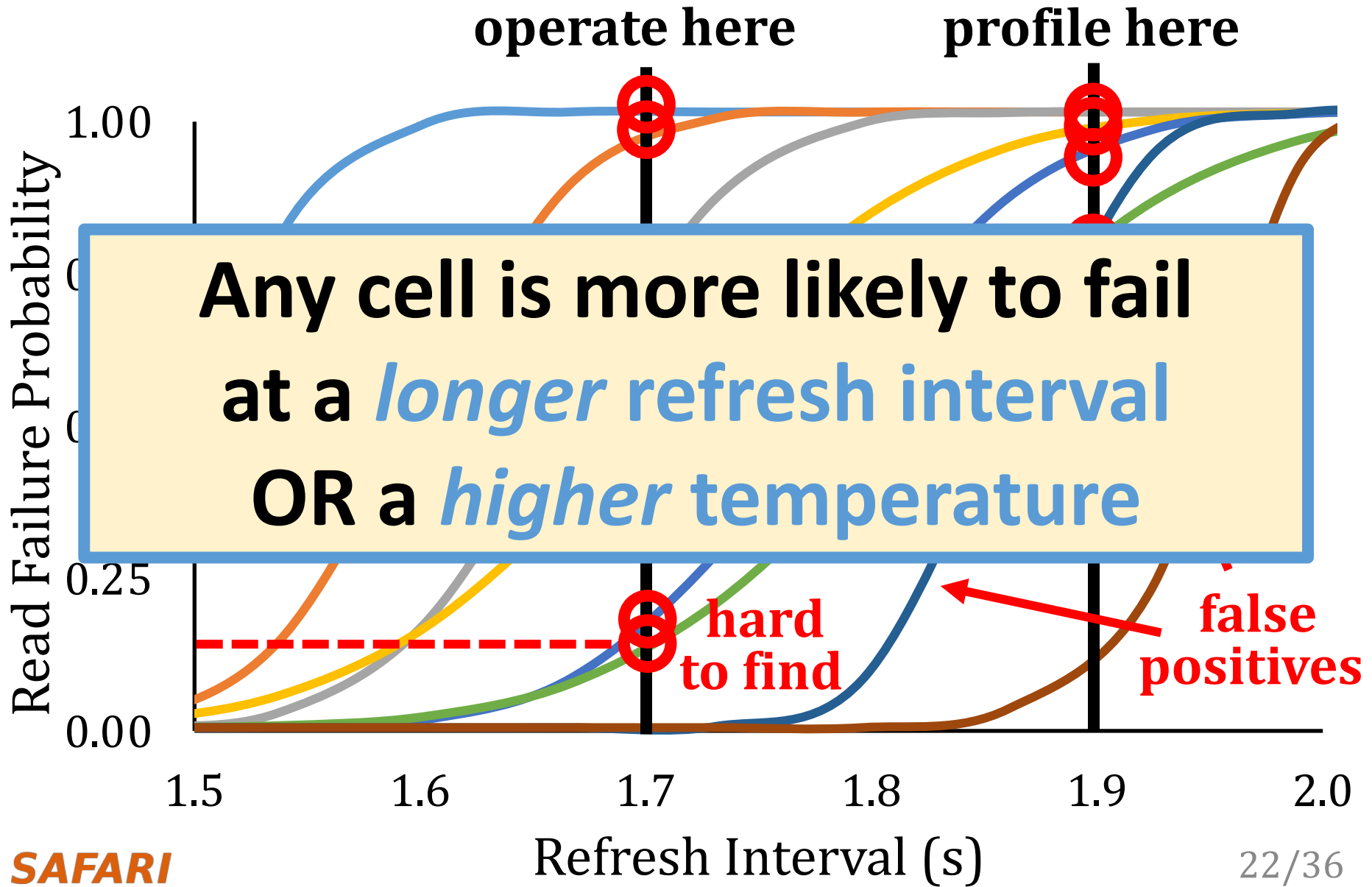
Single-cell Failure Probability (Real)



Single-cell Failure Probability (Real)



Single-cell Failure Probability (Real)



REAPER Outline

1. DRAM Refresh Background

2. Failure Profiling Challenges

3. Current Approaches

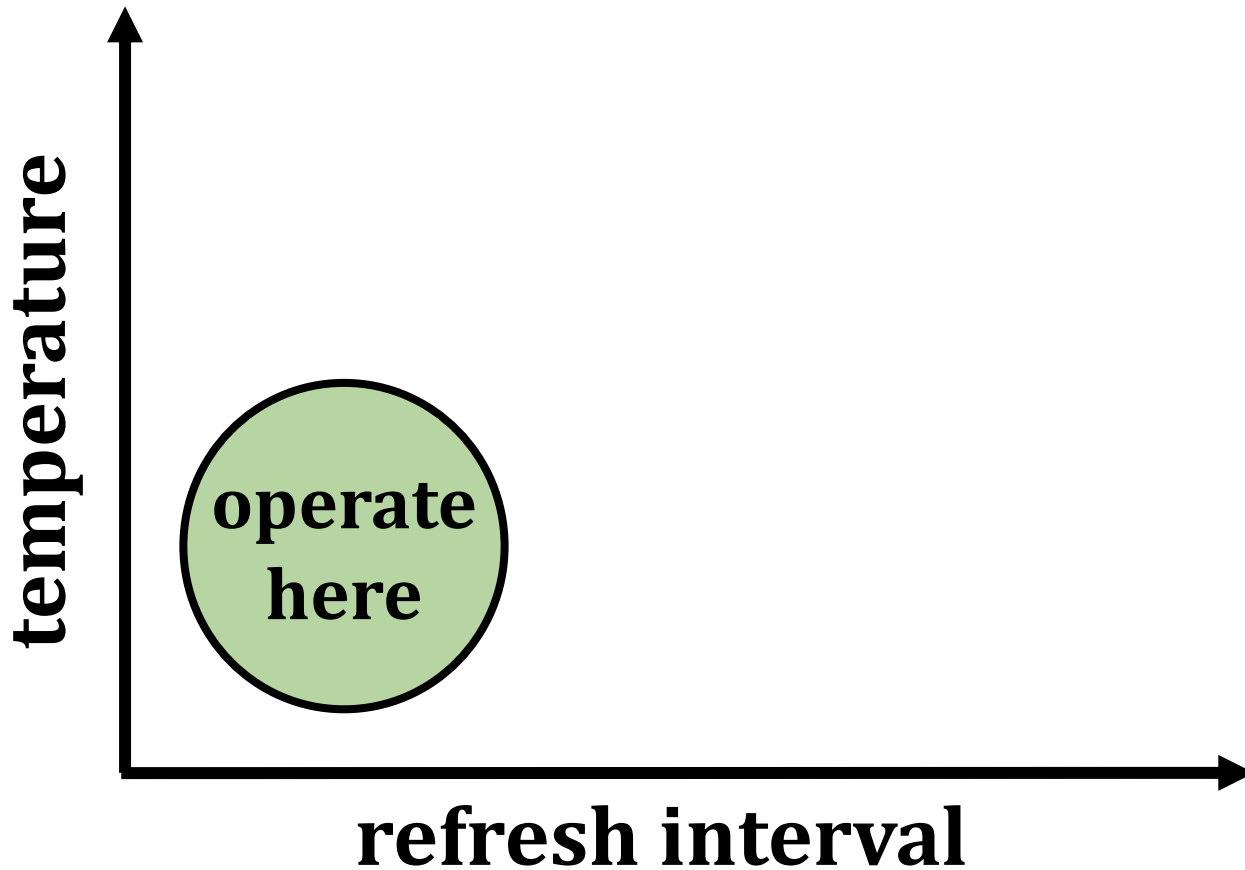
4. LPDDR4 Characterization

5. Reach Profiling

6. End-to-end Evaluation

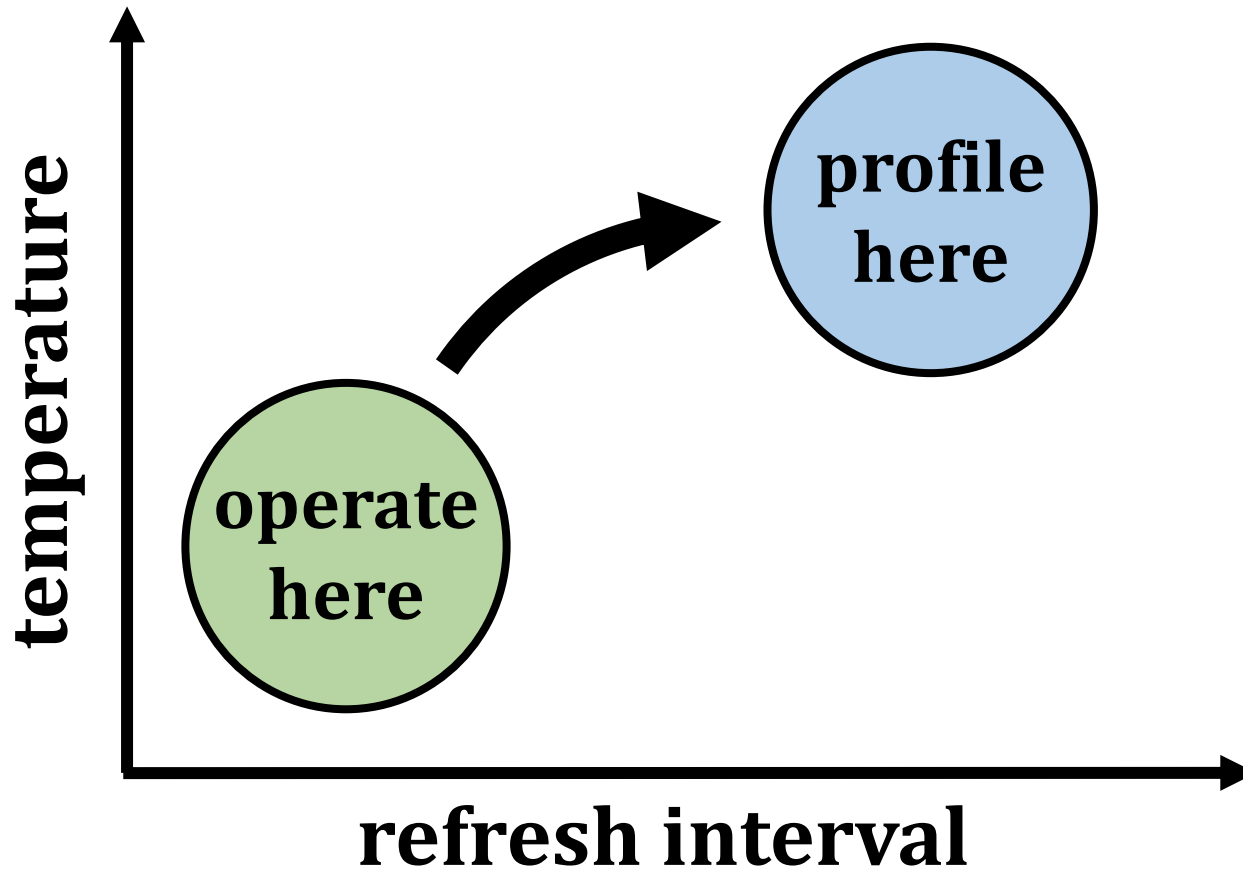
Reach Profiling

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



Reach Profiling

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



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

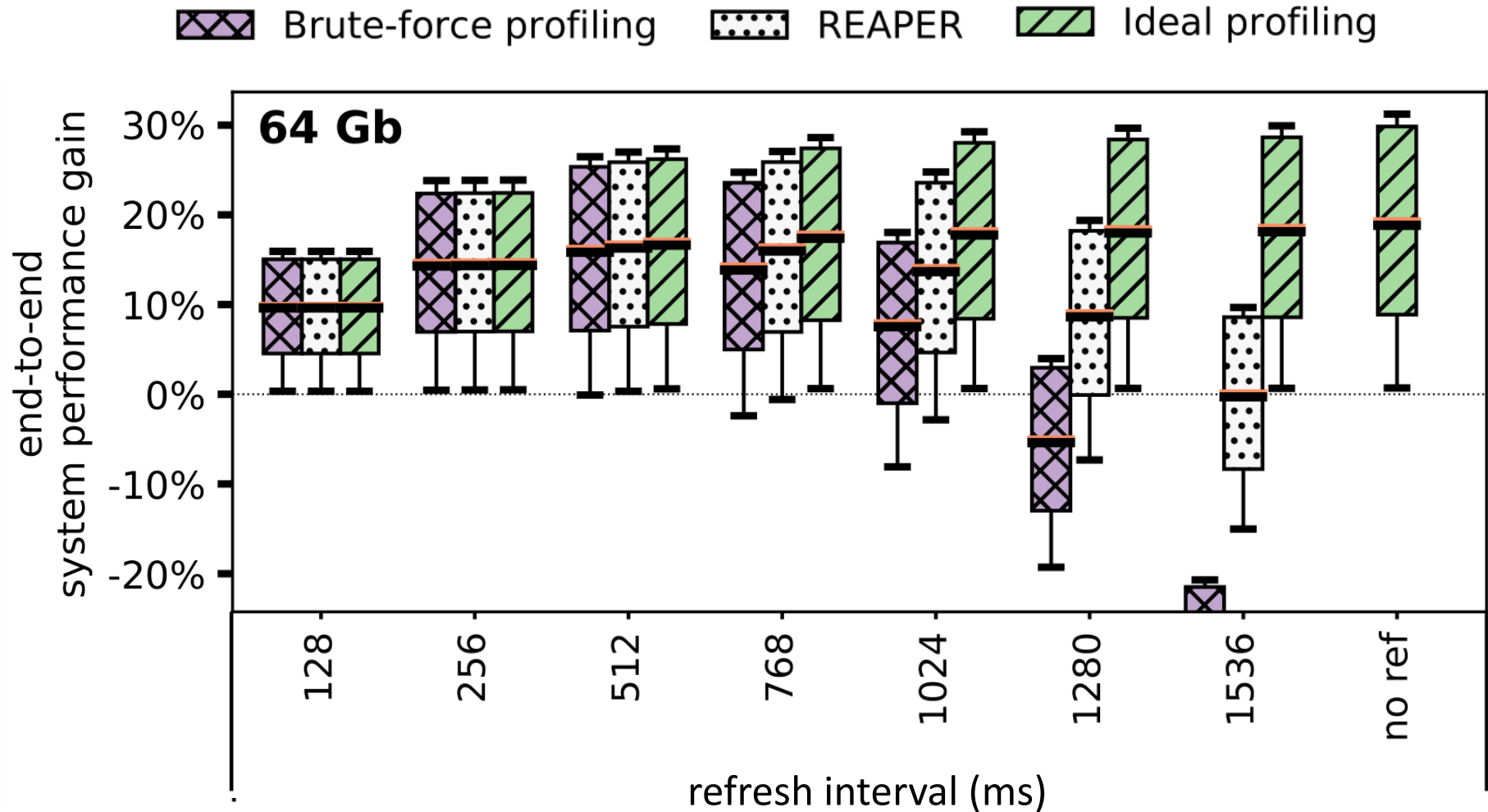
- 1. Runtime:** how long profiling takes
- 2. Coverage:** portion of all possible failures discovered by profiling

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

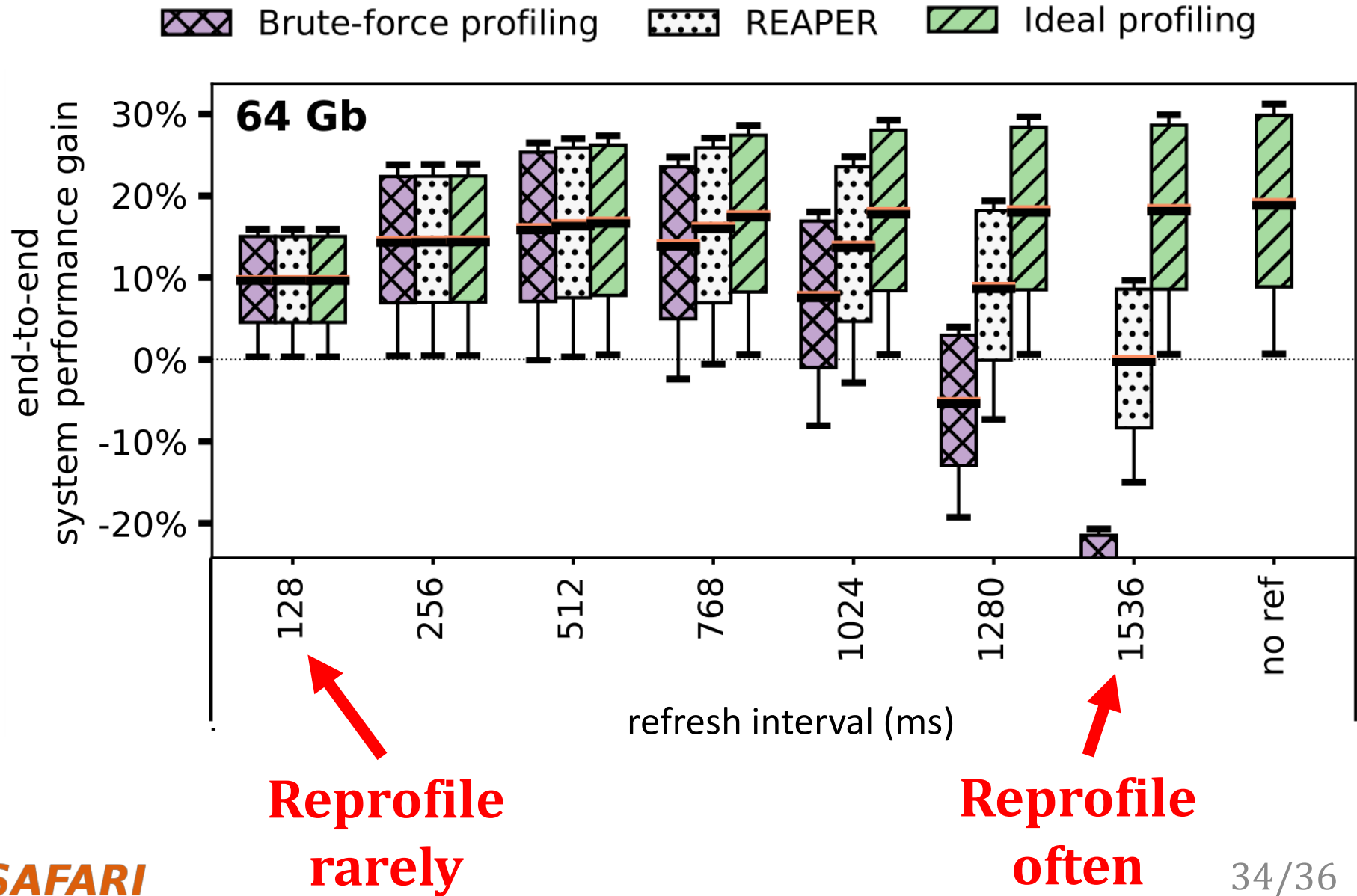
Evaluation Methodology

- Simulators
 - **Performance:** Ramulator [Kim+, CAL'15]
 - **Energy:** DRAMPower [Chandrasekar+, DSD'11]
- Configuration
 - 4-core (4GHz), 8MB LLC
 - LPDDR4-3200, 4 channels, 1 rank/channel
- Workloads
 - 20 random 4-core benchmark mixes
 - SPEC CPU2006 benchmark suite

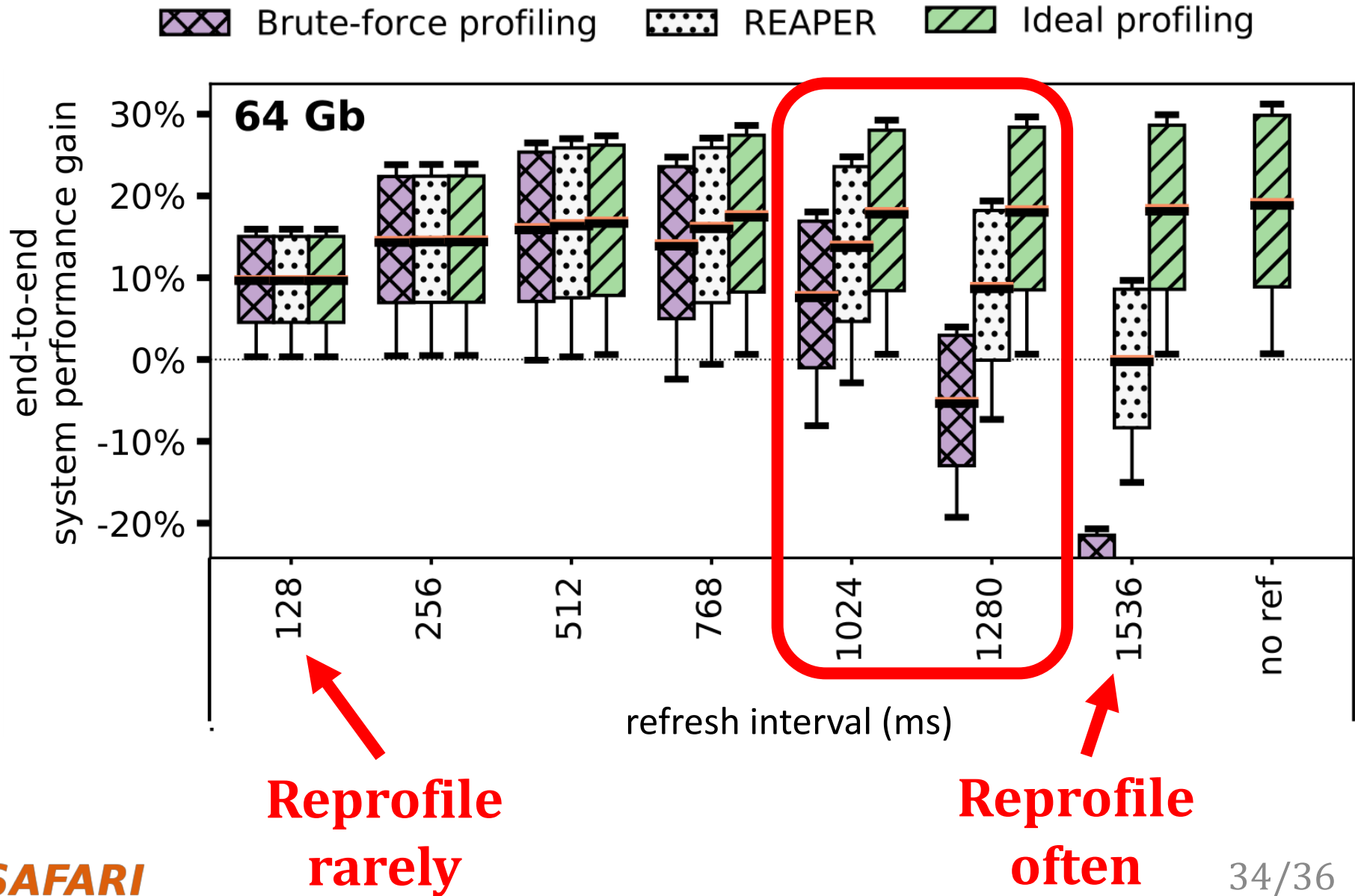
Simulated End-to-end Performance



Simulated End-to-end Performance



Simulated End-to-end Performance



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

Other Analyses in the Paper

- **Detailed LPDDR4 characterization data**
 - Temperature dependence effects
 - Retention time distributions
 - Data pattern dependence
 - Variable retention time
 - Individual cell failure distributions
- **Profiling tradeoff space characterization**
 - Runtime, coverage, and false positive rate
 - Temperature and refresh interval
- **Probabilistic model for tolerable failure rates**
- **Detailed results for end-to-end evaluations**

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

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^{§‡} Jeremie S. Kim^{‡§} Onur Mutlu^{§‡}
[§]ETH Zürich [‡]Carnegie Mellon University

The DRAM Latency PUF:

Quickly Evaluating Physical Unclonable Functions
by Exploiting the Latency-Reliability Tradeoff
in Modern Commodity DRAM Devices

Jeremie S. Kim Minesh Patel

Hasan Hassan Onur Mutlu



QR Code for the paper

https://people.inf.ethz.ch/omutlu/pub/dram-latency-puf_hpca18.pdf

HPCA 2018

SAFARI



ETH zürich

Carnegie Mellon

Keeping Future Memory Secure

How Do We Keep Memory Secure?

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

Solution Direction: Principled Designs

Design fundamentally secure
computing architectures

Predict and prevent
such safety issues

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-crumbing?t=1535427165809>

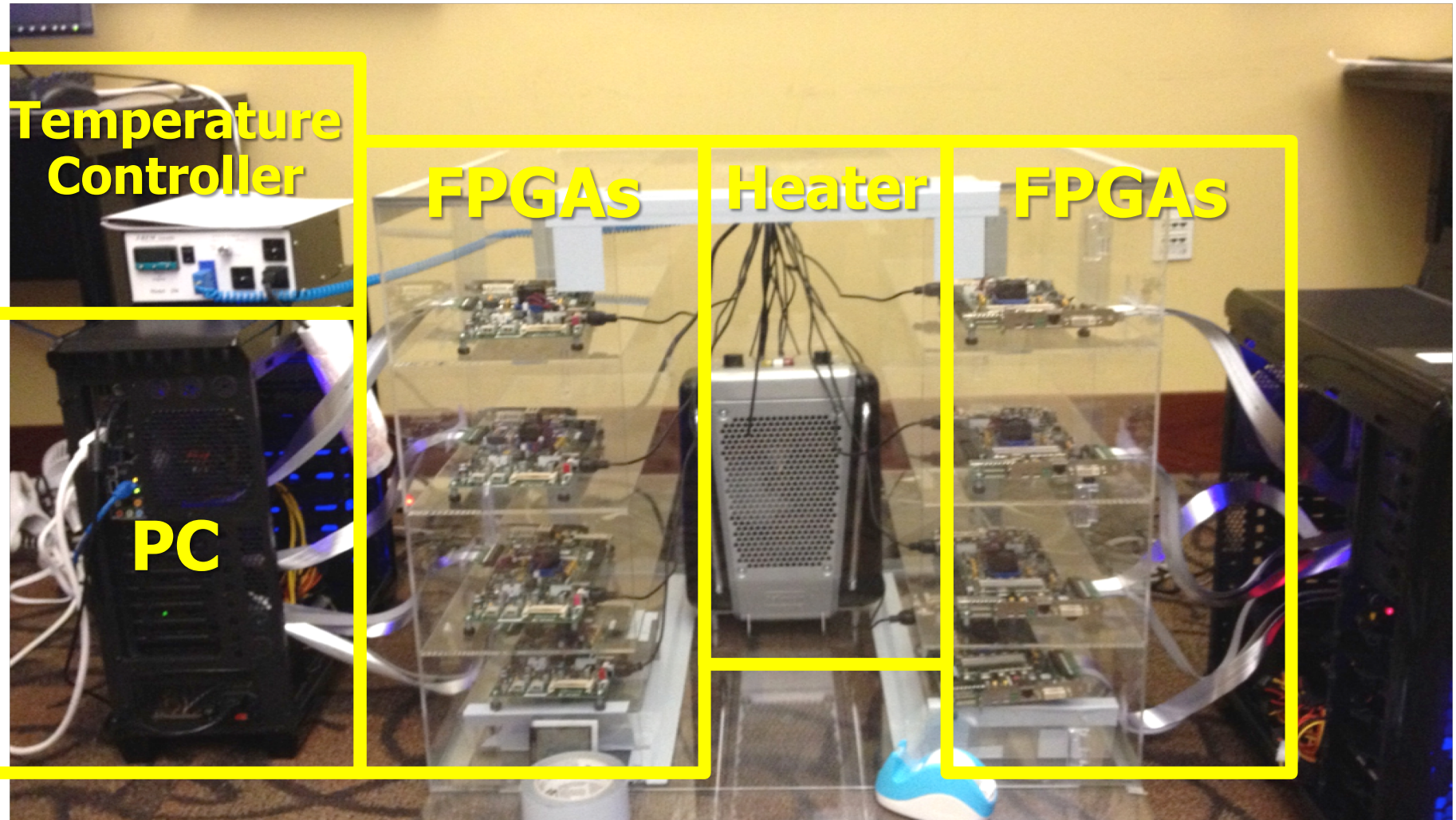
A More Recent Example (2018)



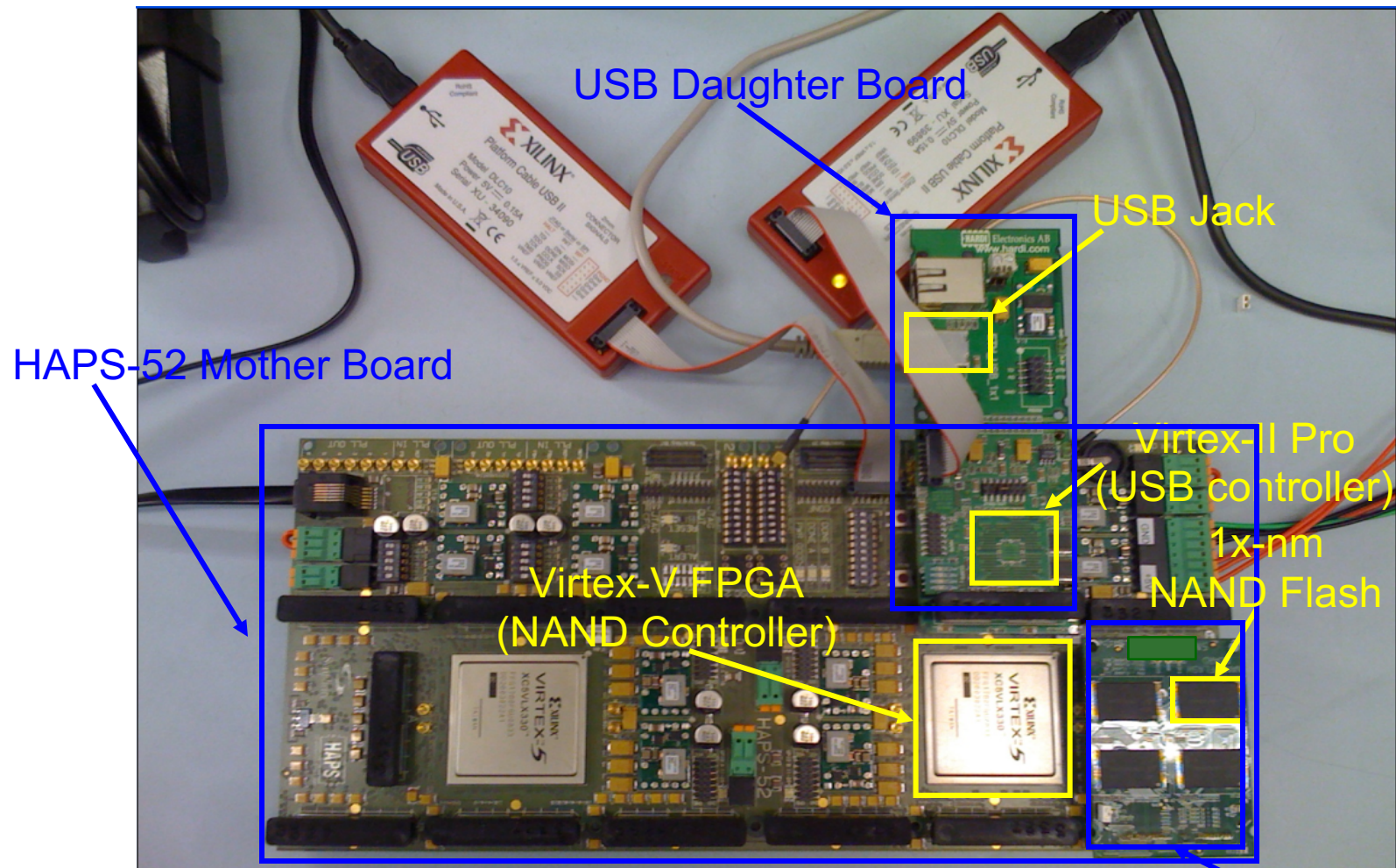
Architecting for Security

- **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)



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]

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

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.
qw@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[†] Saugata Ghose[†] Yixin Luo^{‡†} Ken Mai[†] Onur Mutlu^{§†} Erich F. Haratsch[‡]
[†]Carnegie Mellon University [‡]Seagate Technology [§]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

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[†] Saugata Ghose[†] Yu Cai[‡] Erich F. Haratsch[‡] Onur Mutlu^{§†}
[†]*Carnegie Mellon University* [‡]*Seagate Technology* [§]*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[†]

Saugata Ghose[†]

Yu Cai[†]

Erich F. Haratsch[‡]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[‡]Seagate Technology

[§]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.

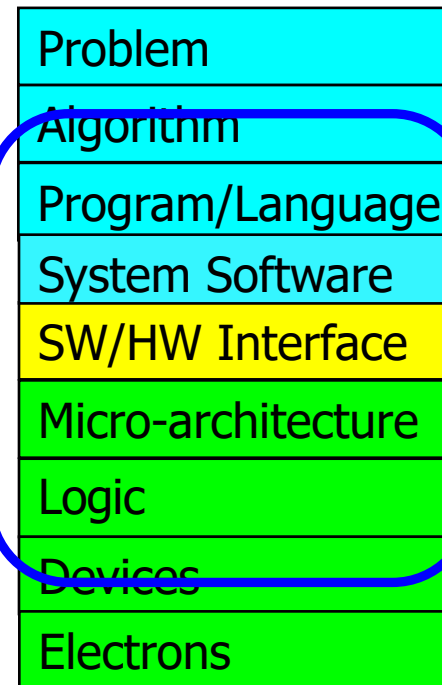
Luo+, "HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature-Awareness," HPCA 2018.

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.

There are Two Other Solution Directions

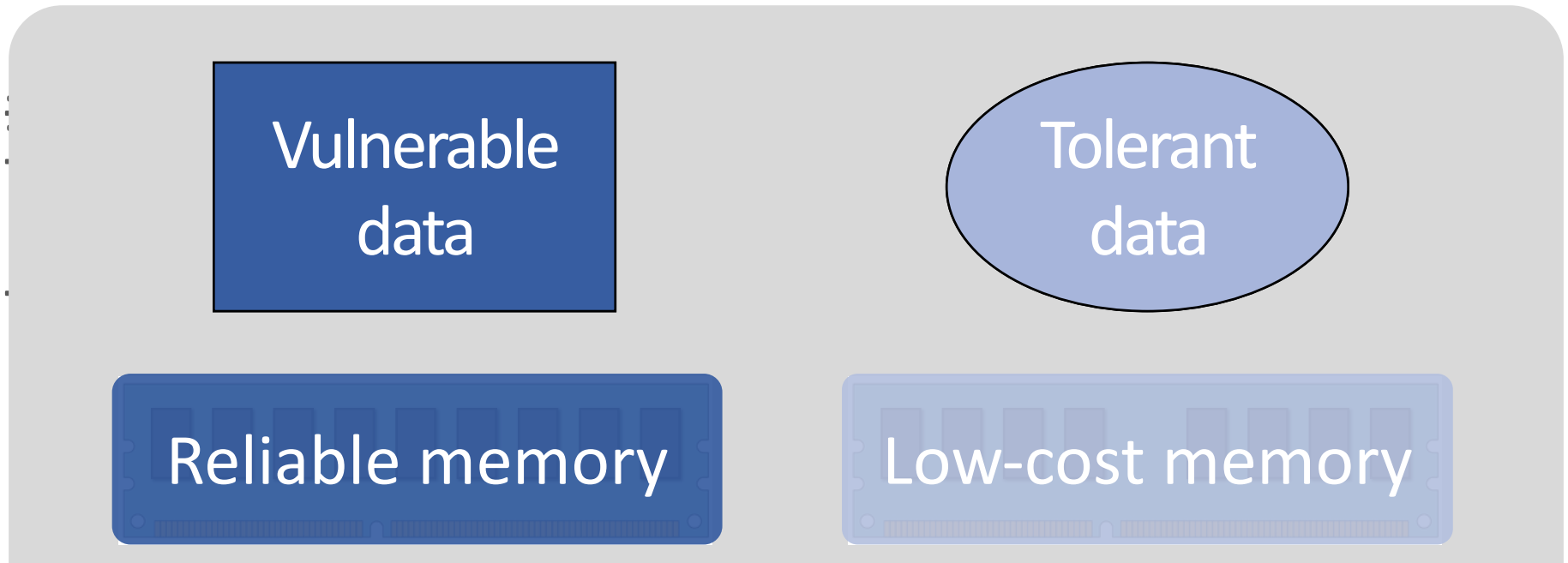
- **New Technologies:** Replace or (more likely) augment DRAM with a different technology
 - ❑ Non-volatile memories
- **Embracing Un-reliability:**
Design memories with different reliability and store data intelligently across them
[Luo+ DSN 2014]



■ ...

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

Exploiting Memory Error Tolerance with Hybrid Memory Systems



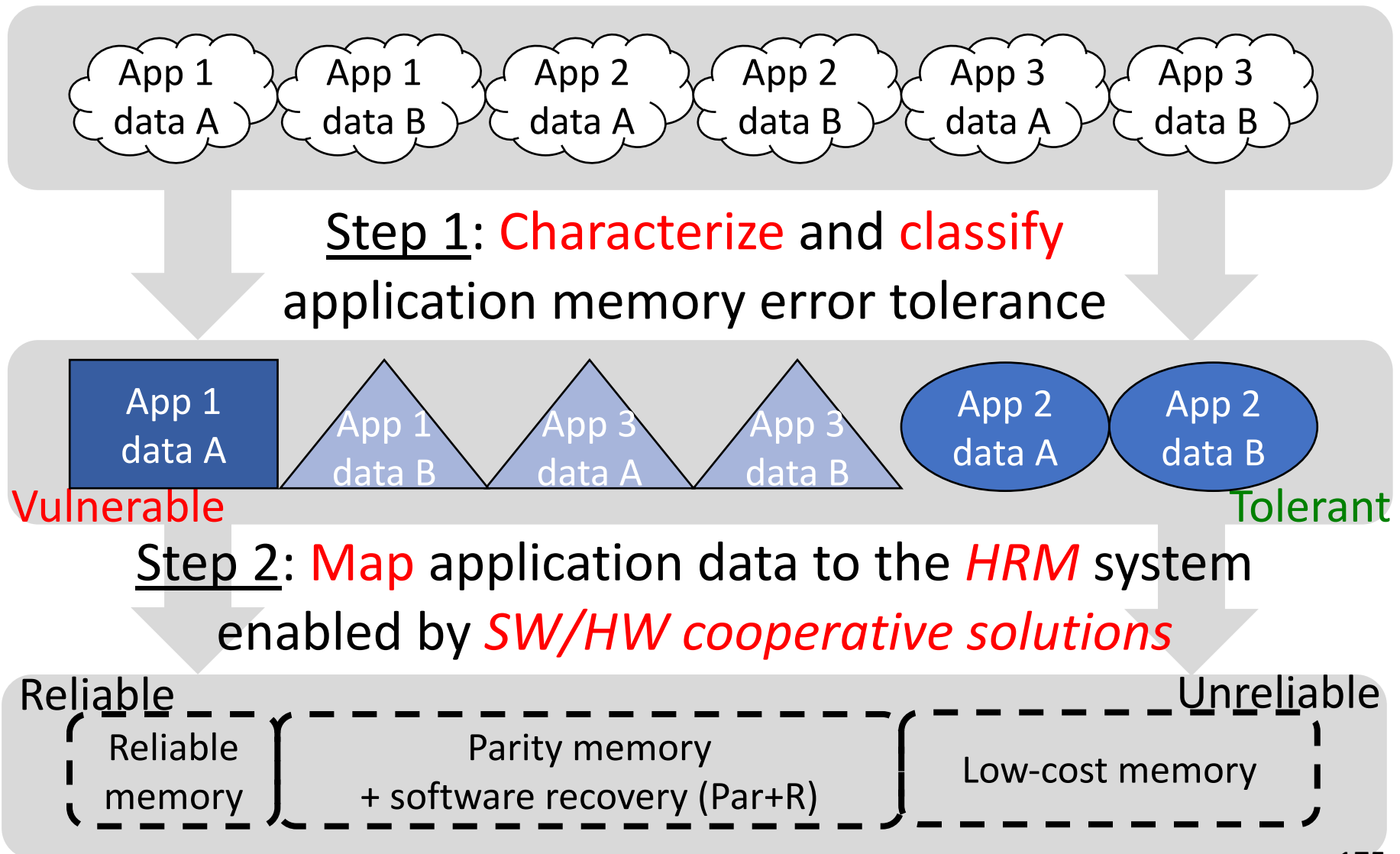
On Microsoft's Web Search workload

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

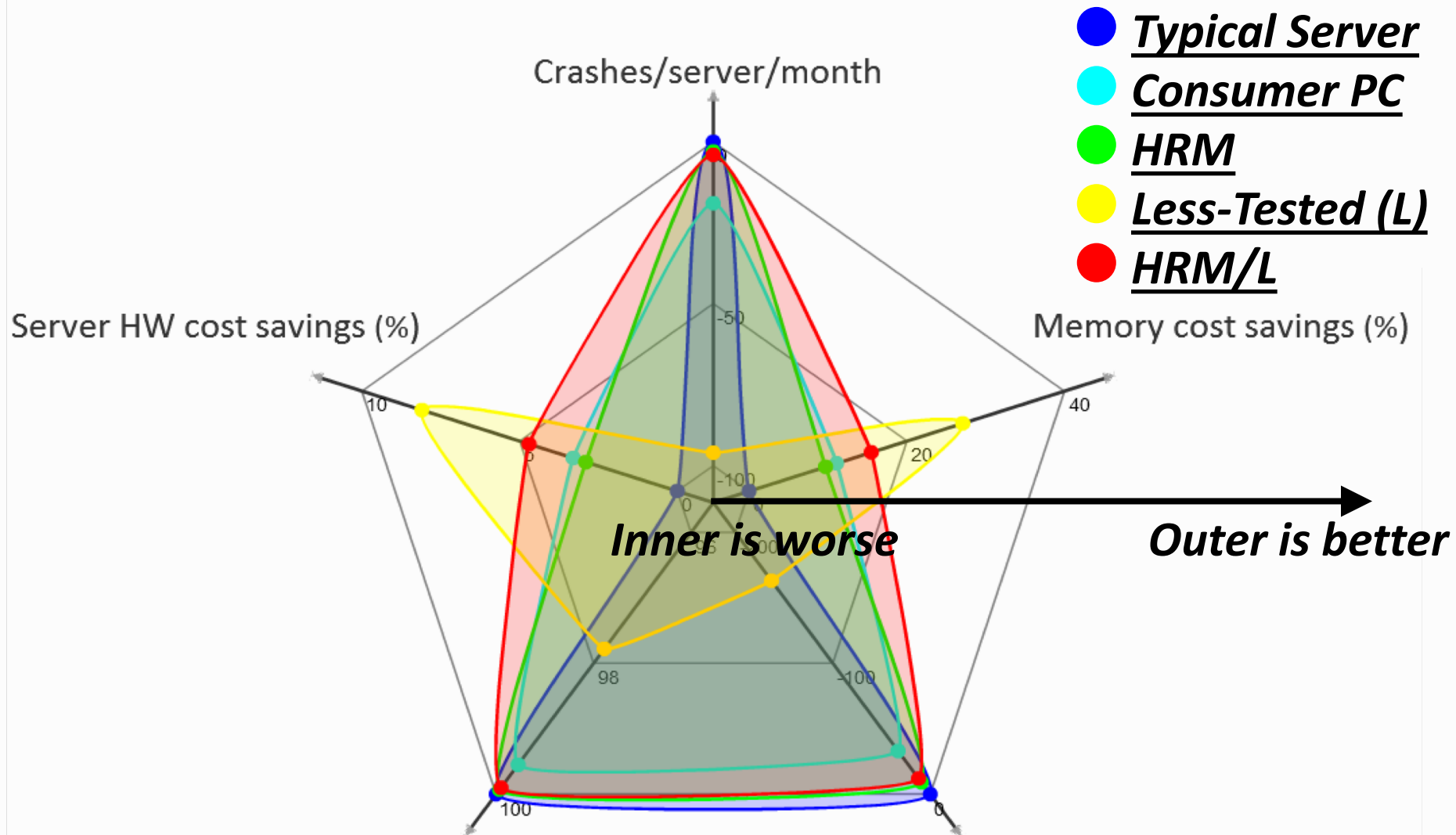
Achieves single server **availability** target of **99.90 %**

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, bknessib, kvaid}@microsoft.com

RowHammer Conclusion

Summary: Memory Reliability and Security

- **Memory reliability is reducing**
- Reliability issues open up security vulnerabilities
 - Very hard to defend against
- Rowhammer is an example
 - Its implications on system security research are tremendous & exciting
- **Good news: We have a lot more to do.**
- **Understand:** Solid methodologies for failure modeling and discovery
 - Modeling based on real device data – small scale and large scale
- **Architect:** Principled co-architecting of system and memory
 - Good partitioning of duties across the stack
- **Design & Test:** Principled electronic design, automation, testing
 - High coverage and good interaction with system reliability methods

Fundamentally Secure, Reliable, Safe Computing Architectures

Main Memory Needs
Intelligent Controllers

Memory Systems

Fundamentals, Recent Research, Challenges, Opportunities

Lecture 2b: Flash Memory Reliability & Security

Prof. Onur Mutlu

omutlu@gmail.com

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

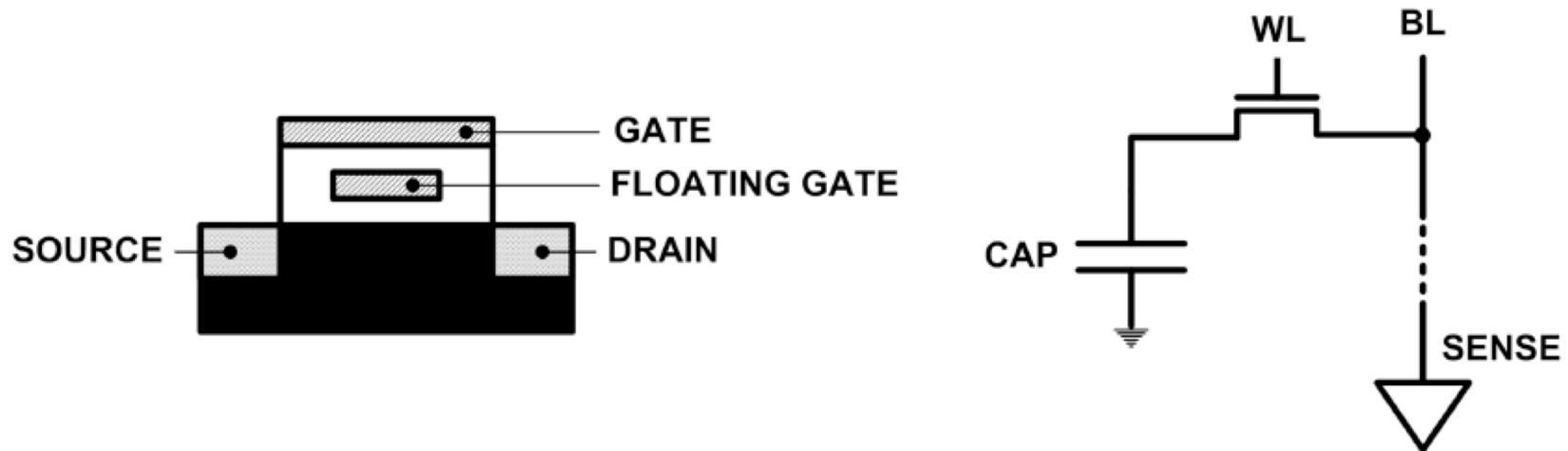
8 October 2018

Technion Fast Course 2018

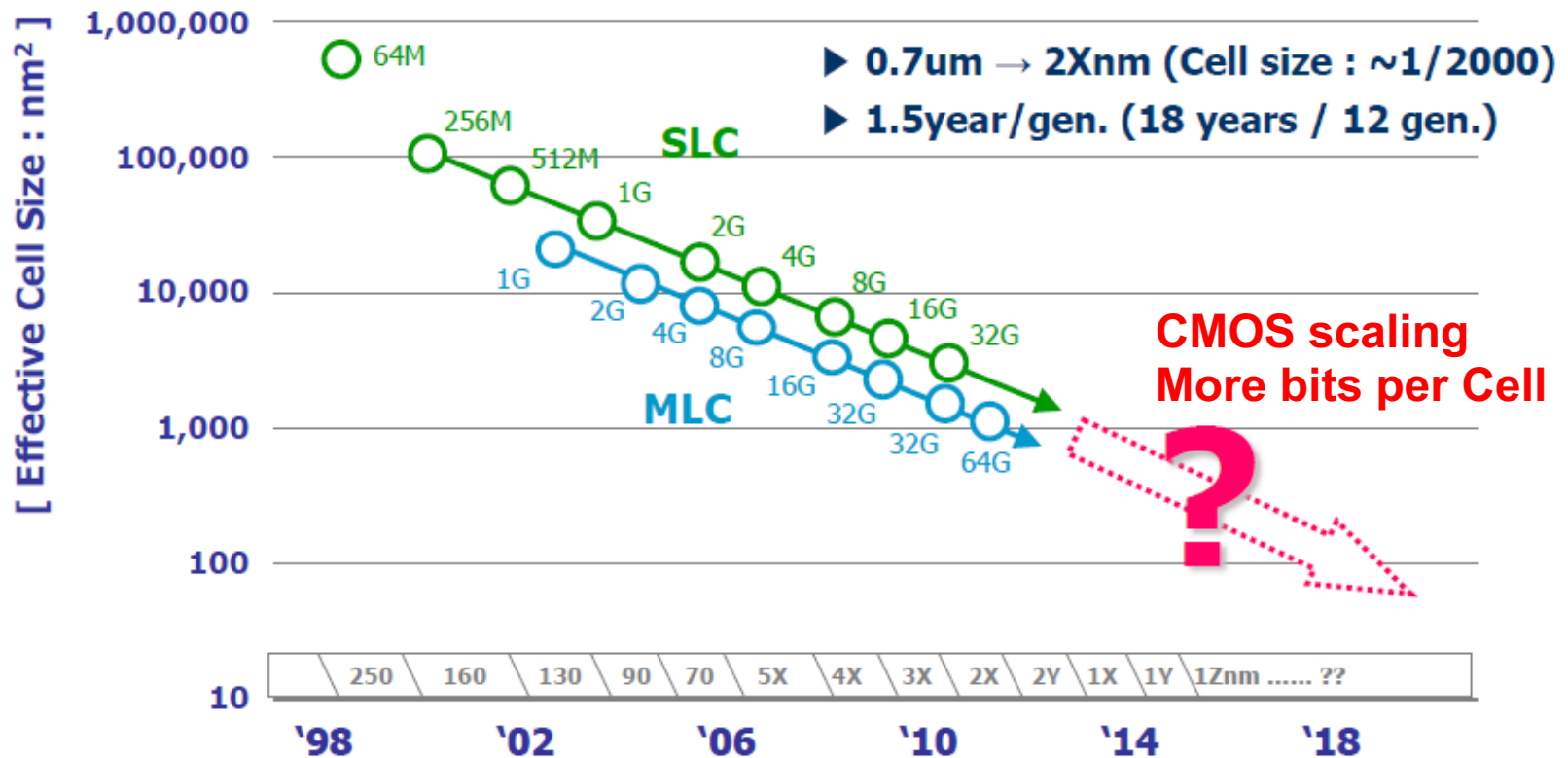
Flash Memory Reliability and Security

Limits of Charge Memory

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



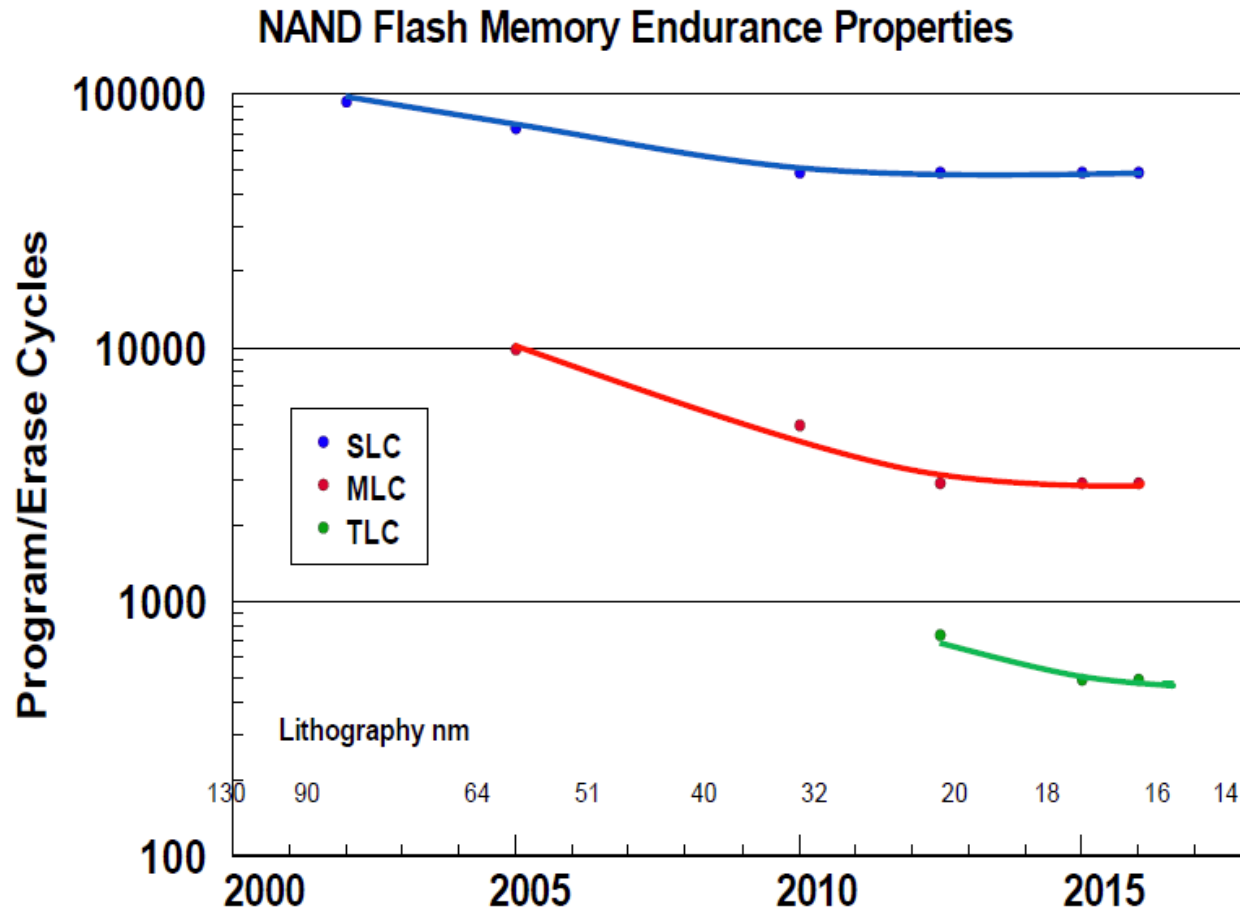
Evolution of NAND Flash Memory



Seaung Suk Lee, "Emerging Challenges in NAND Flash Technology", Flash Summit 2011 (Hynix)

- Flash memory is widening its range of applications
 - Portable consumer devices, laptop PCs and enterprise servers

Flash Challenges: Reliability and Endurance



E. Grochowski et al., "Future technology challenges for NAND flash and HDD products", Flash Memory Summit 2012

- **P/E cycles (provided)**

A few thousand

- **P/E cycles (required)**

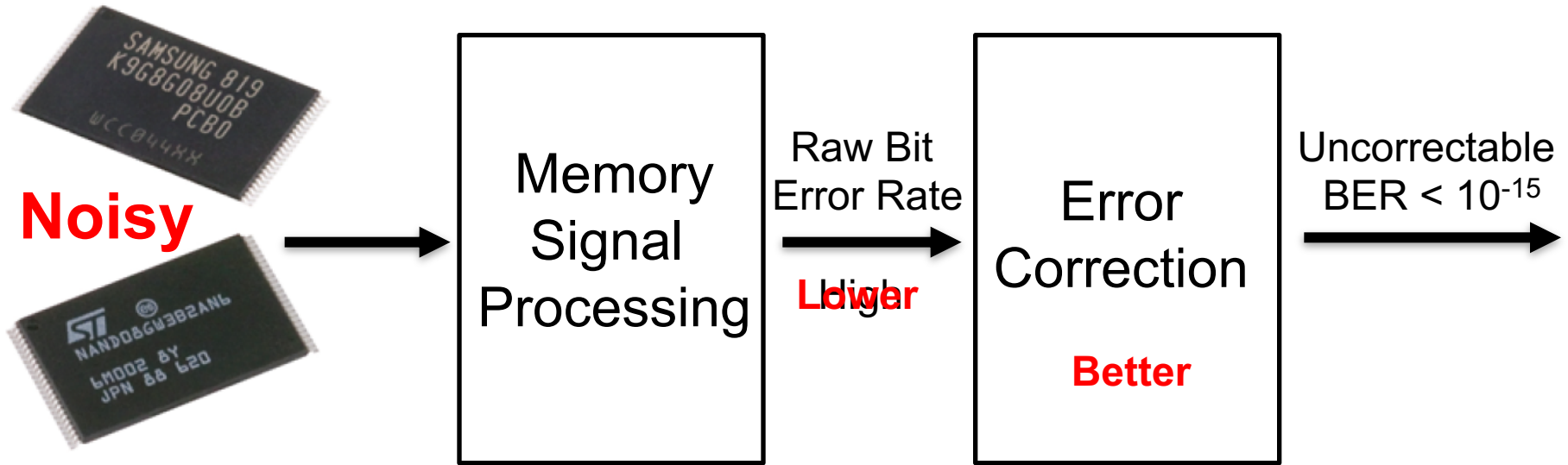
Writing
the full capacity
of the drive
10 times per day
for 5 years
(STEC)

> 50k P/E cycles

NAND Flash Memory is Increasingly Noisy



Future NAND Flash-based Storage Architecture



Our Goals:

Build reliable error models for NAND flash memory

Design efficient reliability mechanisms based on the model

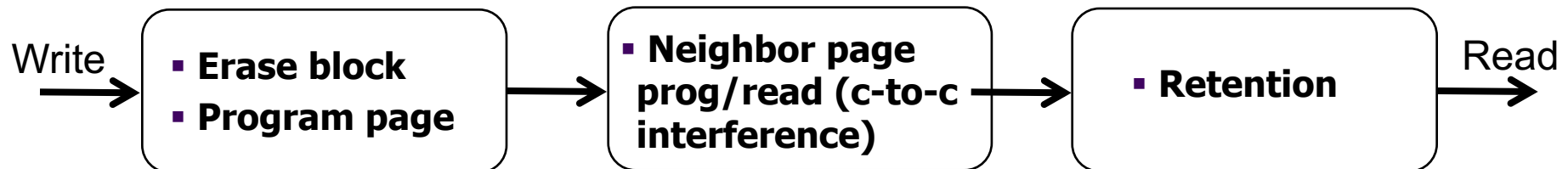
NAND Flash Error Model



Experimentally characterize and model dominant errors

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

Luo et al., "Enabling Accurate and Practical Online Flash Channel Modeling for Modern MLC NAND Flash Memory", **JSAC 2016**



Cai et al., "Threshold voltage distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling", **DATE 2013**

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

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

Cai et al., "Neighbor-Cell Assisted Error Correction in MLC NAND Flash Memories", **SIGMETRICS 2014**

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

Cai et al., "Flash Correct-and-Refresh: Retention-aware error management for increased flash memory lifetime", **ICCD 2012**

Cai et al., "Error Analysis and Retention-Aware Error Management for NAND Flash Memory", **ITJ 2013**

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

Our Goals and Approach

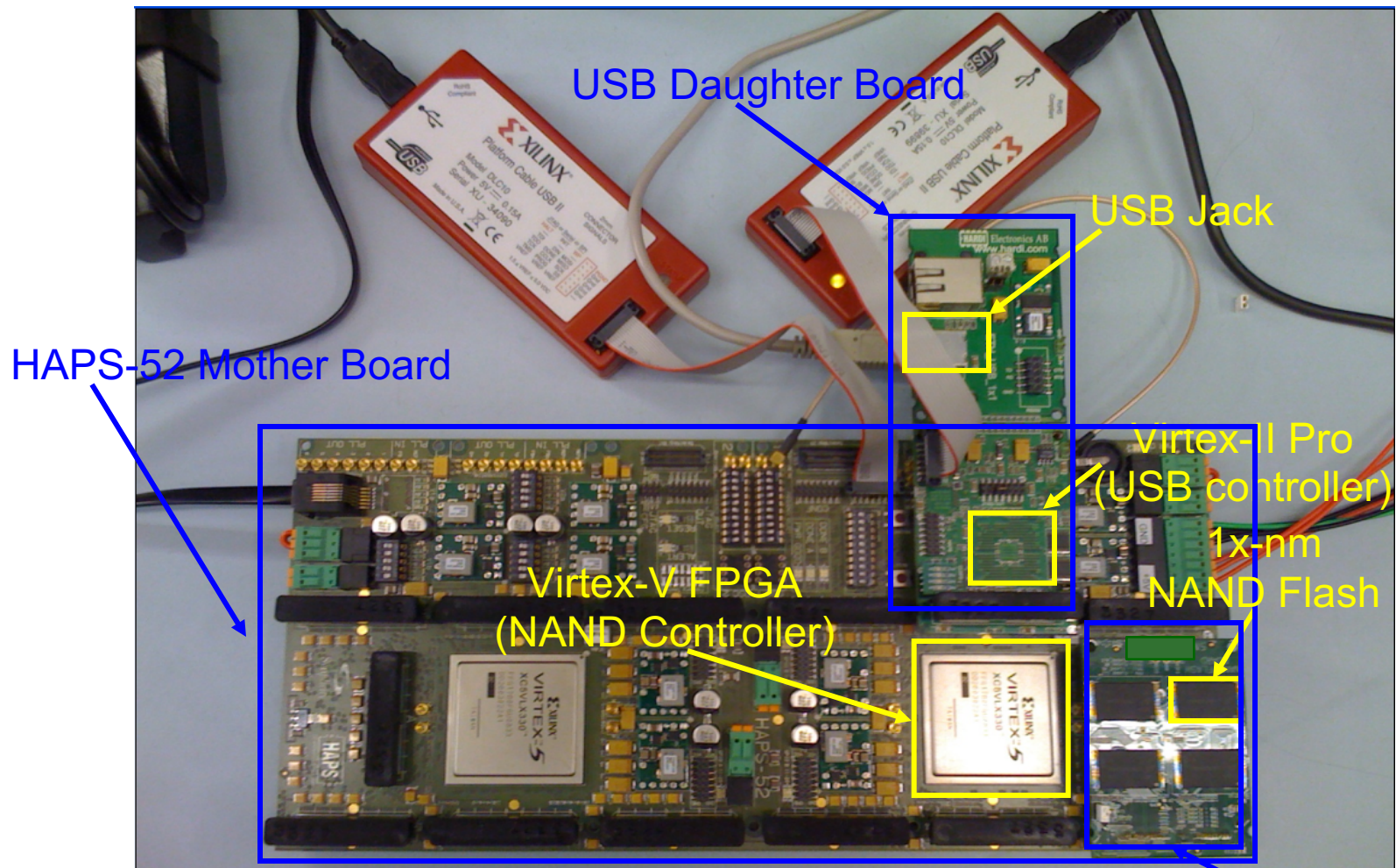
■ Goals:

- Understand error mechanisms and develop reliable predictive models for MLC NAND flash memory errors
- Develop efficient error management techniques to mitigate errors and improve flash reliability and endurance

■ Approach:

- Solid experimental analyses of errors in real MLC NAND flash memory → drive the understanding and models
- Understanding, models, and creativity → drive the new techniques

Experimental Testing Platform



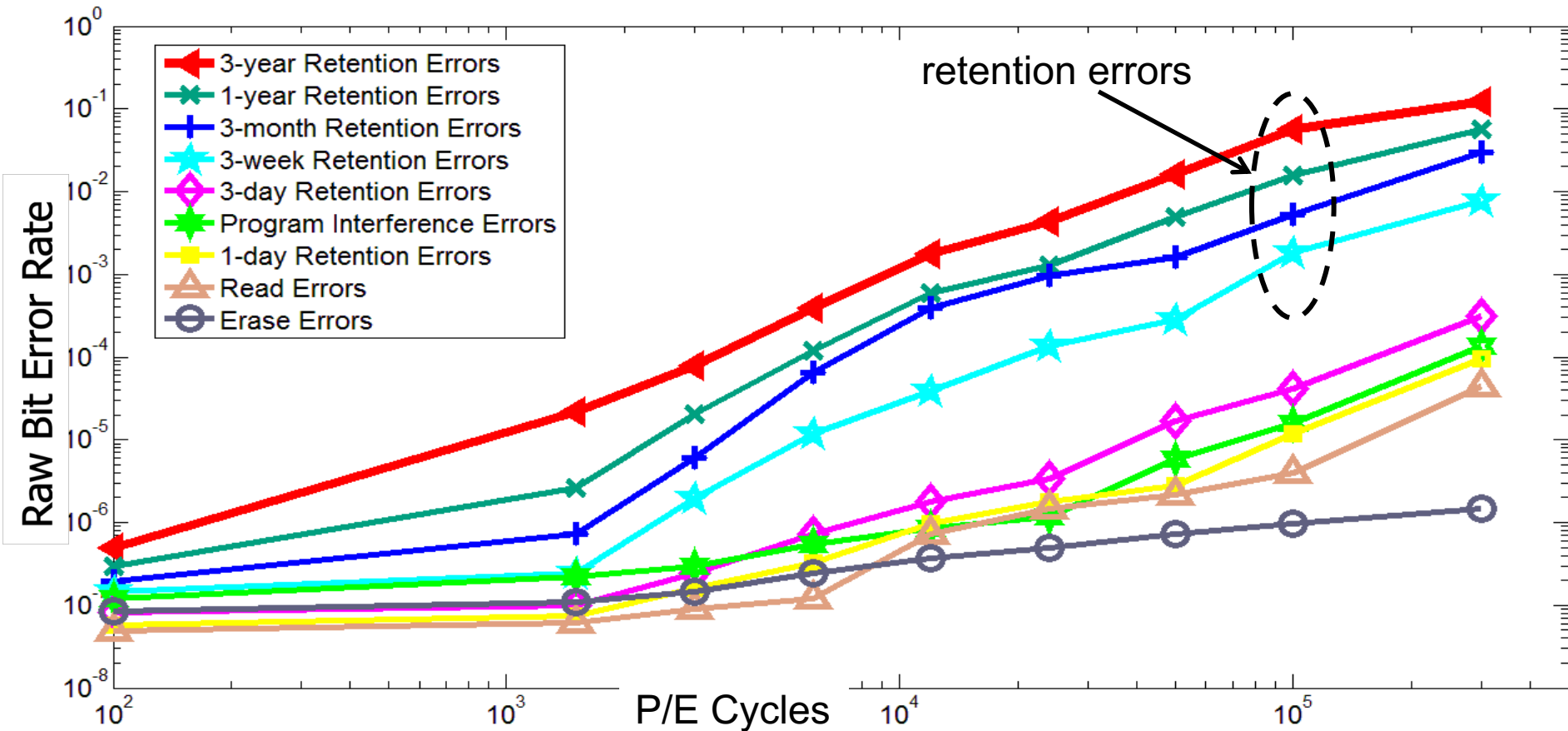
[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018, SIGMETRICS 2018]

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

NAND Flash Error Types

- Four types of errors [Cai+, DATE 2012]
- Caused by common flash operations
 - ❑ Read errors
 - ❑ Erase errors
 - ❑ Program (interference) errors
- Caused by flash cell losing charge over time
 - ❑ Retention errors
 - Whether an error happens depends on required retention time
 - Especially problematic in MLC flash because threshold voltage window to determine stored value is smaller

Observations: Flash Error Analysis



- Raw bit error rate increases exponentially with P/E cycles
- Retention errors are dominant (>99% for 1-year ret. time)
- Retention errors increase with retention time requirement

More on Flash Error Analysis

- Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai, **"Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis"** *Proceedings of the Design, Automation, and Test in Europe Conference (**DATE**), Dresden, Germany, March 2012. Slides (ppt)*

Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis

Yu Cai¹, Erich F. Haratsch², Onur Mutlu¹ and Ken Mai¹

¹Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

²LSI Corporation, 1110 American Parkway NE, Allentown, PA

¹{yucai, onur, kenmai}@andrew.cmu.edu, ²erich.haratsch@lsi.com

Solution to Retention Errors

- Refresh periodically
- Change the period based on P/E cycle wearout
 - Refresh more often at higher P/E cycles
- Use a combination of **in-place** and **remapping-based** refresh
- Cai et al. “**Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime**”, ICCD 2012.

Flash Correct-and-Refresh [ICCD'12]

- Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F. Haratsch, Adrian Cristal, Osman Unsal, and Ken Mai,
"Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime"
Proceedings of the 30th IEEE International Conference on Computer Design (ICCD), Montreal, Quebec, Canada, September 2012. [Slides](#) (ppt)(pdf)

Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime

Yu Cai¹, Gulay Yalcin², Onur Mutlu¹, Erich F. Haratsch³, Adrian Cristal², Osman S. Unsal² and Ken Mai¹

¹DSSC, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

²Barcelona Supercomputing Center, C/Jordi Girona 29, Barcelona, Spain

³LSI Corporation, 1110 American Parkway NE, Allentown, PA

Many Errors and Their Mitigation [PIEEE'17]

Table 3 List of Different Types of Errors Mitigated by NAND Flash Error Mitigation Mechanisms

Mitigation Mechanism	Error Type				
	<i>P/E Cycling</i> [32,33,42] (§IV-A)	<i>Program</i> [40,42,53] (§IV-B)	<i>Cell-to-Cell Interference</i> [32,35,36,55] (§IV-C)	<i>Data Retention</i> [20,32,34,37,39] (§IV-D)	<i>Read Disturb</i> [20,32,38,62] (§IV-E)
Shadow Program Sequencing [35,40] (Section V-A)			X		
Neighbor-Cell Assisted Error Correction [36] (Section V-B)			X		
Refresh [34,39,67,68] (Section V-C)				X	X
Read-Retry [33,72,107] (Section V-D)	X			X	X
Voltage Optimization [37,38,74] (Section V-E)	X			X	X
Hot Data Management [41,63,70] (Section V-F)	X	X	X	X	X
Adaptive Error Mitigation [43,65,77,78,82] (Section V-G)	X	X	X	X	X

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.

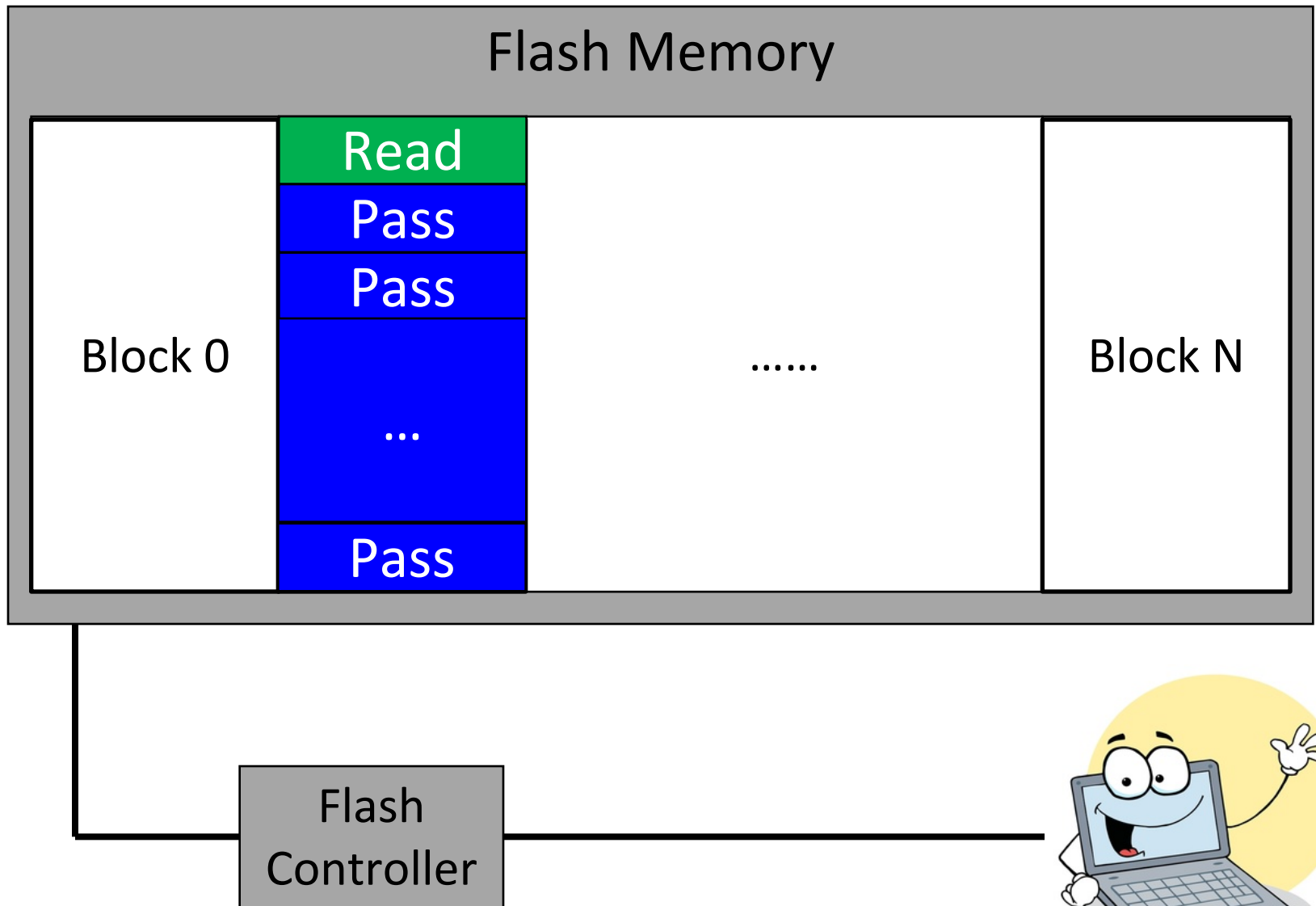
By YU CAI, SAUGATA GHOSE, ERICH F. HARATSCH, YIXIN LUO, AND ONUR MUTLU

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

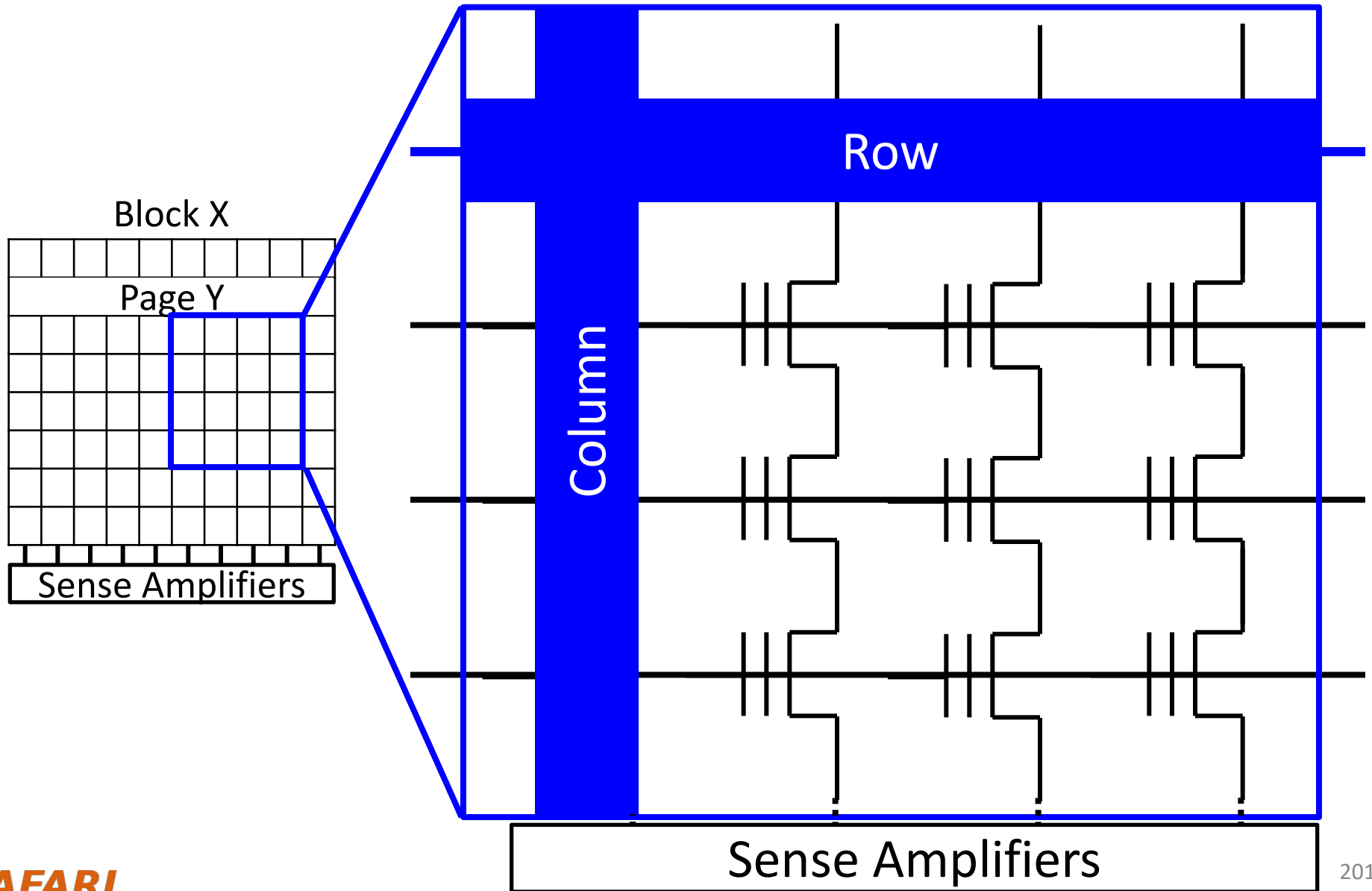
One Issue: Read Disturb in Flash Memory

- All scaled memories are prone to read disturb errors
- DRAM
- SRAM
- Hard Disks: Adjacent Track Interference
- NAND Flash

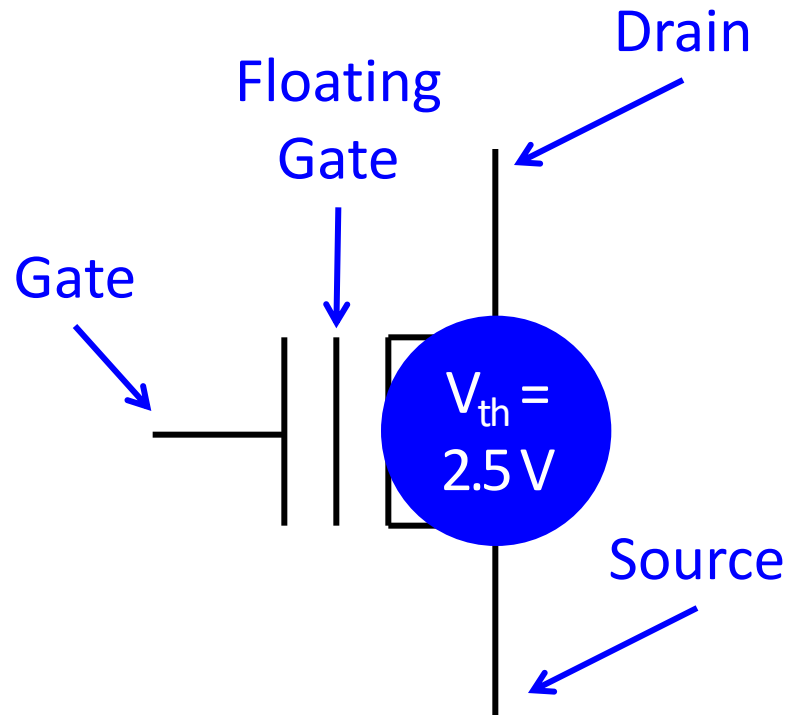
NAND Flash Memory Background



Flash Cell Array

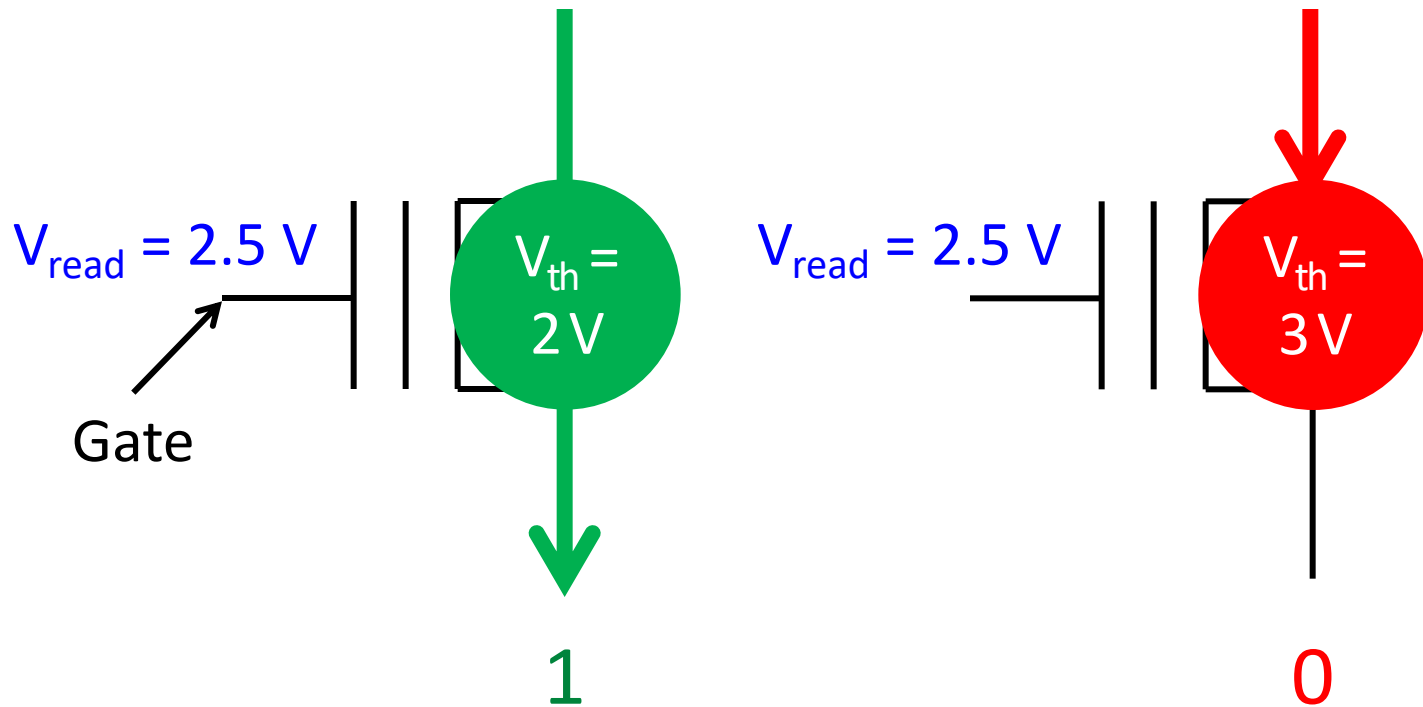


Flash Cell

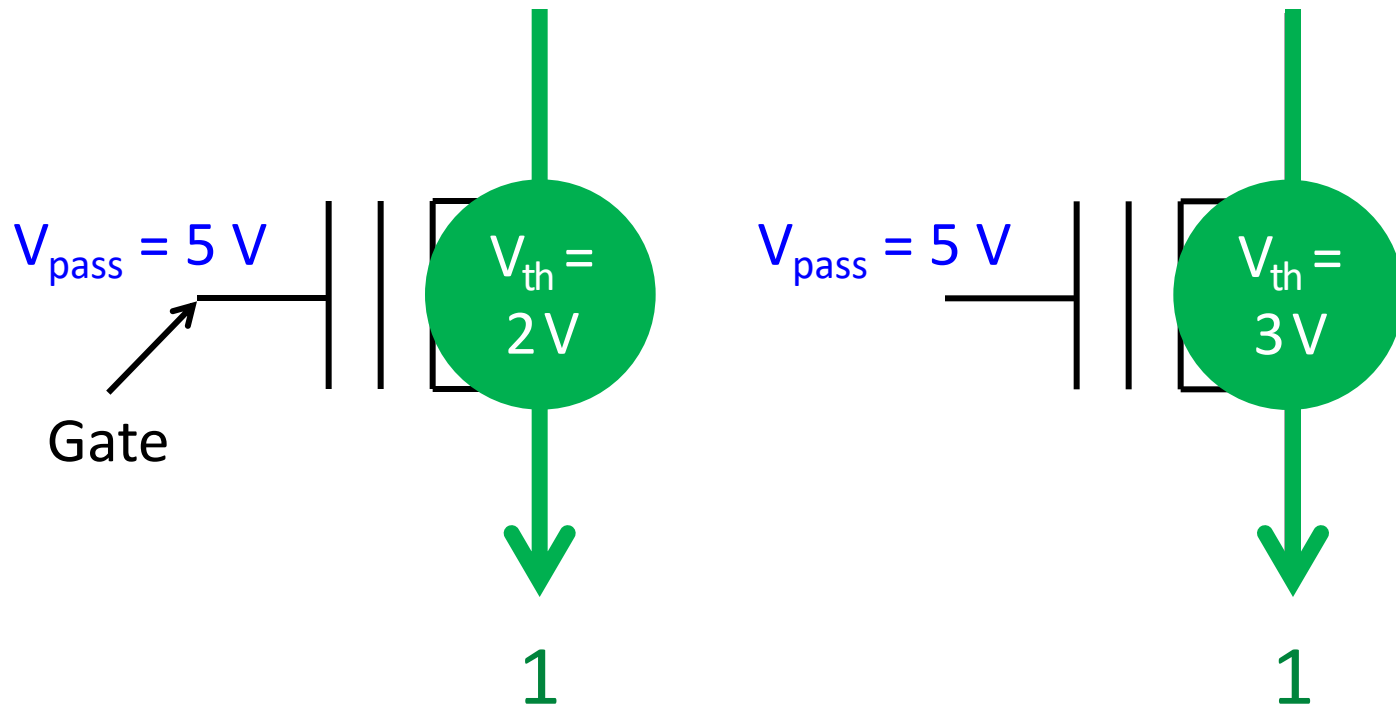


Floating Gate Transistor
(Flash Cell)

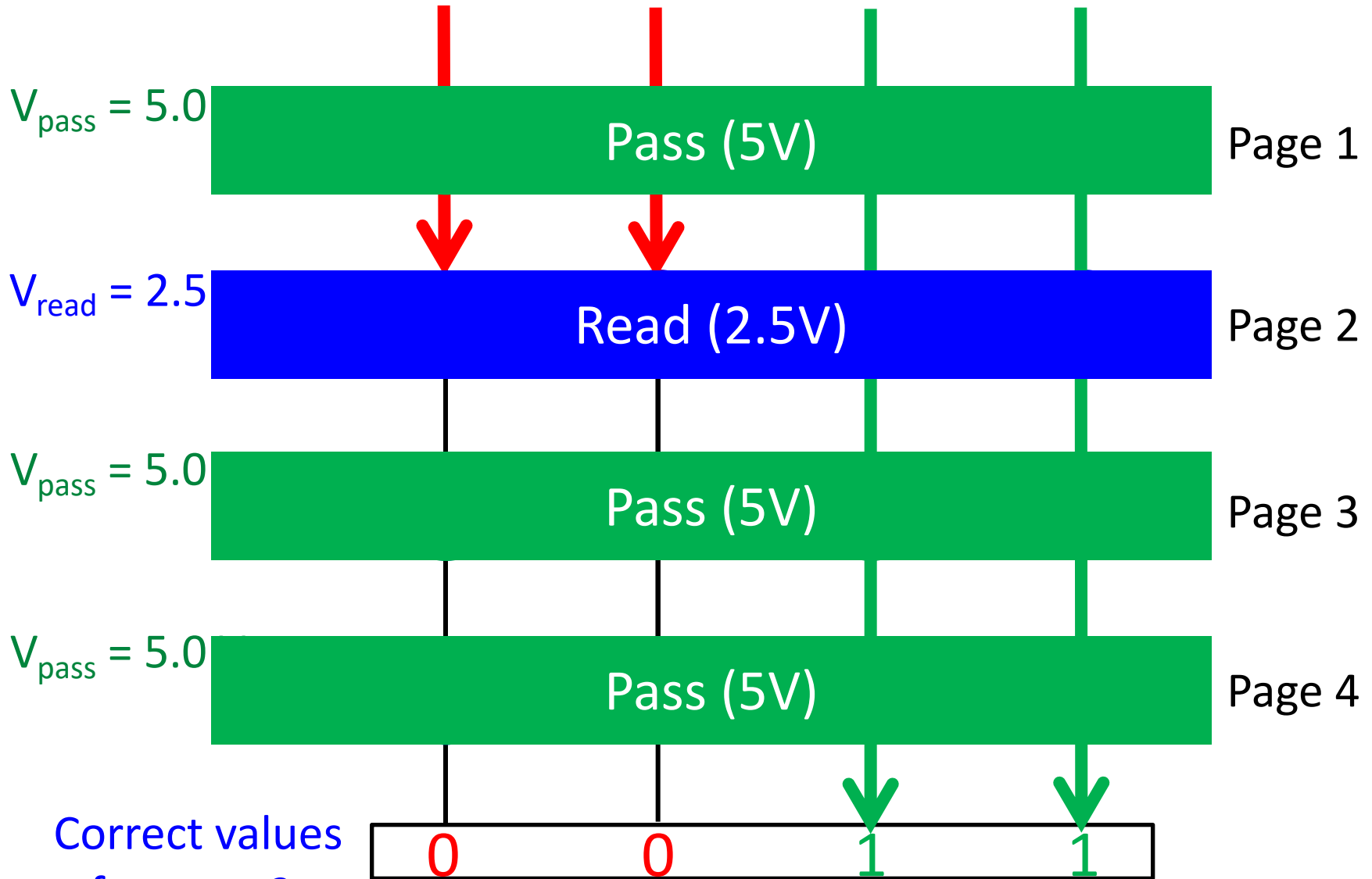
Flash Read



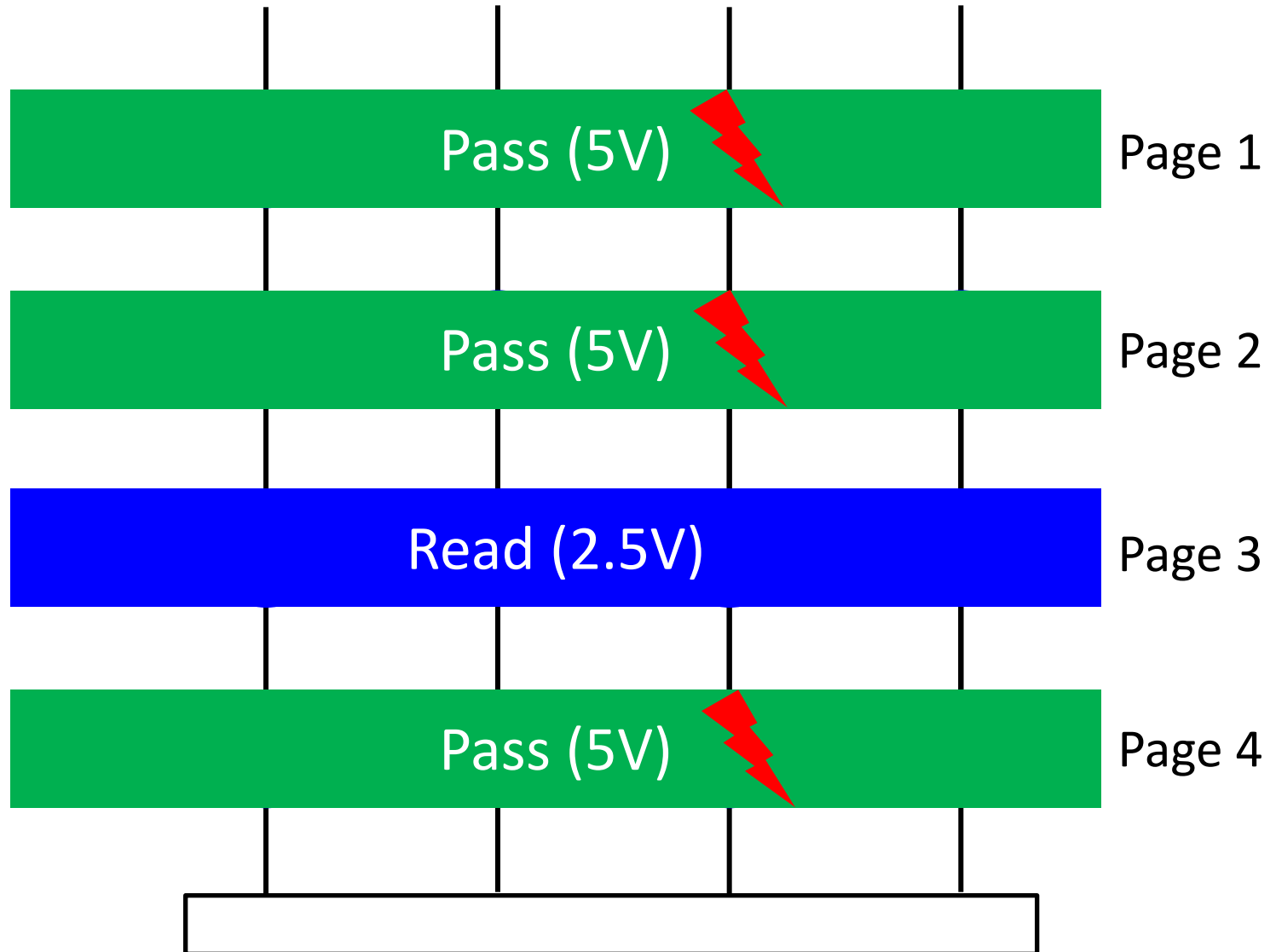
Flash Pass-Through



Read from Flash Cell Array

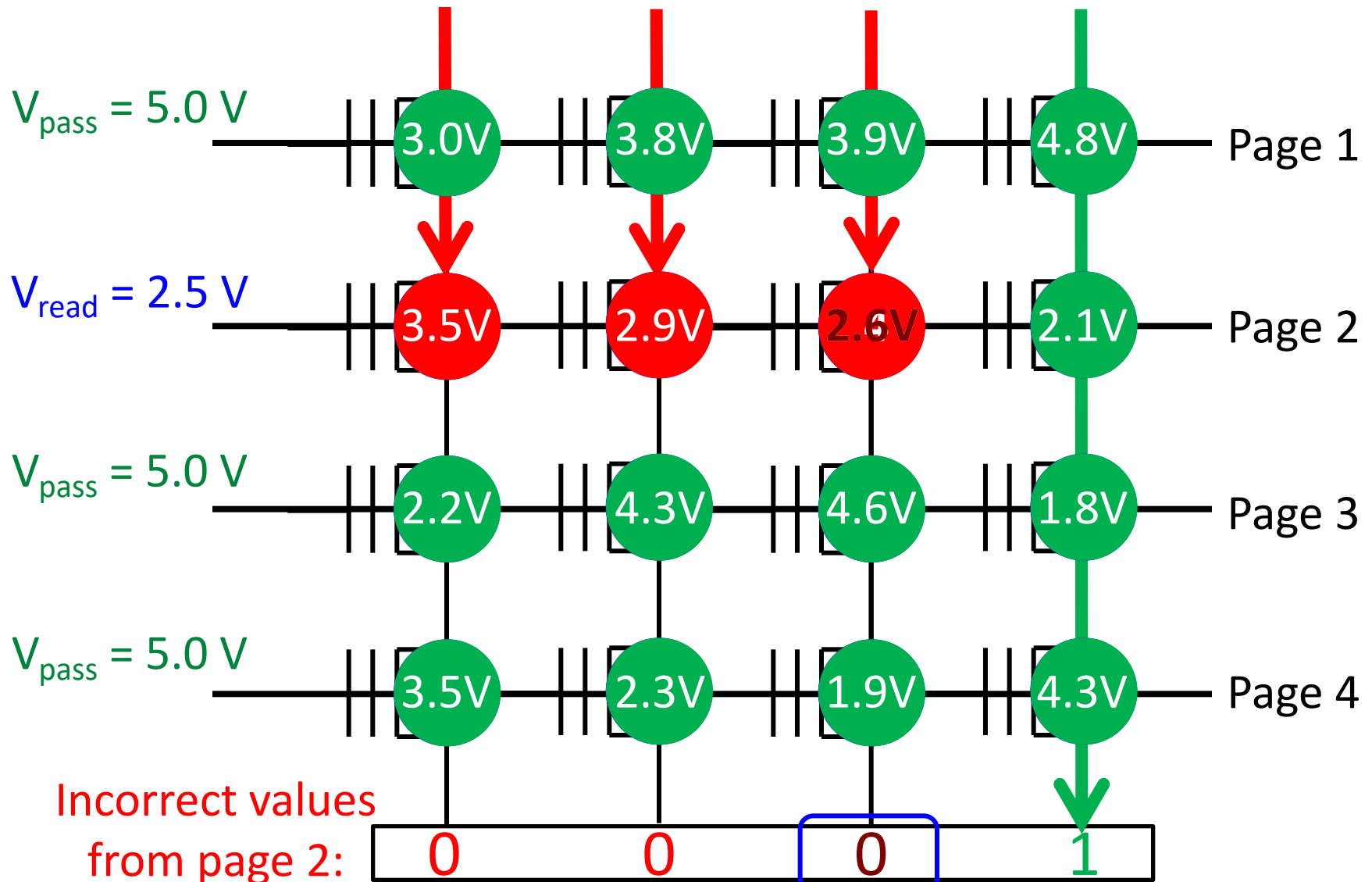


Read Disturb Problem: “Weak Programming” Effect



SAFARI Repeatedly read page 3 (or any page other than page 2)

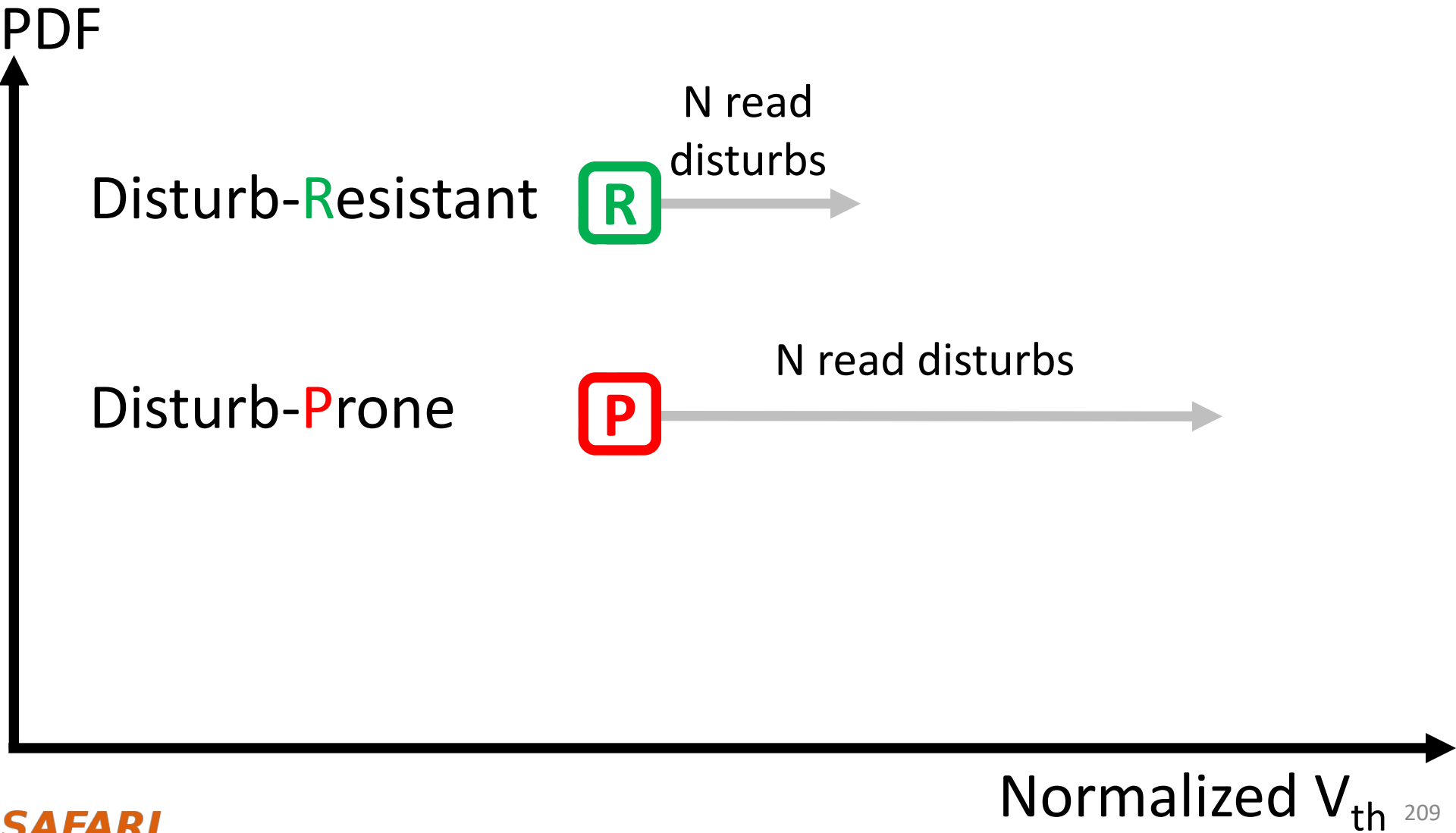
Read Disturb Problem: “Weak Programming” Effect



Executive Summary [DSN'15]

- **Read disturb errors** limit flash memory lifetime today
 - Apply a *high pass-through voltage* (V_{pass}) to multiple pages on a read
 - Repeated application of V_{pass} can alter stored values in unread pages
- We **characterize read disturb** on real NAND flash chips
 - Slightly lowering V_{pass} greatly reduces read disturb errors
 - Some flash cells are more prone to read disturb
- **Technique 1: Mitigate** read disturb errors online
 - V_{pass} **Tuning** dynamically finds and applies a lowered V_{pass} per block
 - Flash memory **lifetime improves by 21%**
- **Technique 2: Recover** after failure to prevent data loss
 - **Read Disturb Oriented Error Recovery** (RDR) selectively corrects cells more susceptible to read disturb errors
 - **Reduces raw bit error rate (RBER) by up to 36%**

Read Disturb Prone vs. Resistant Cells

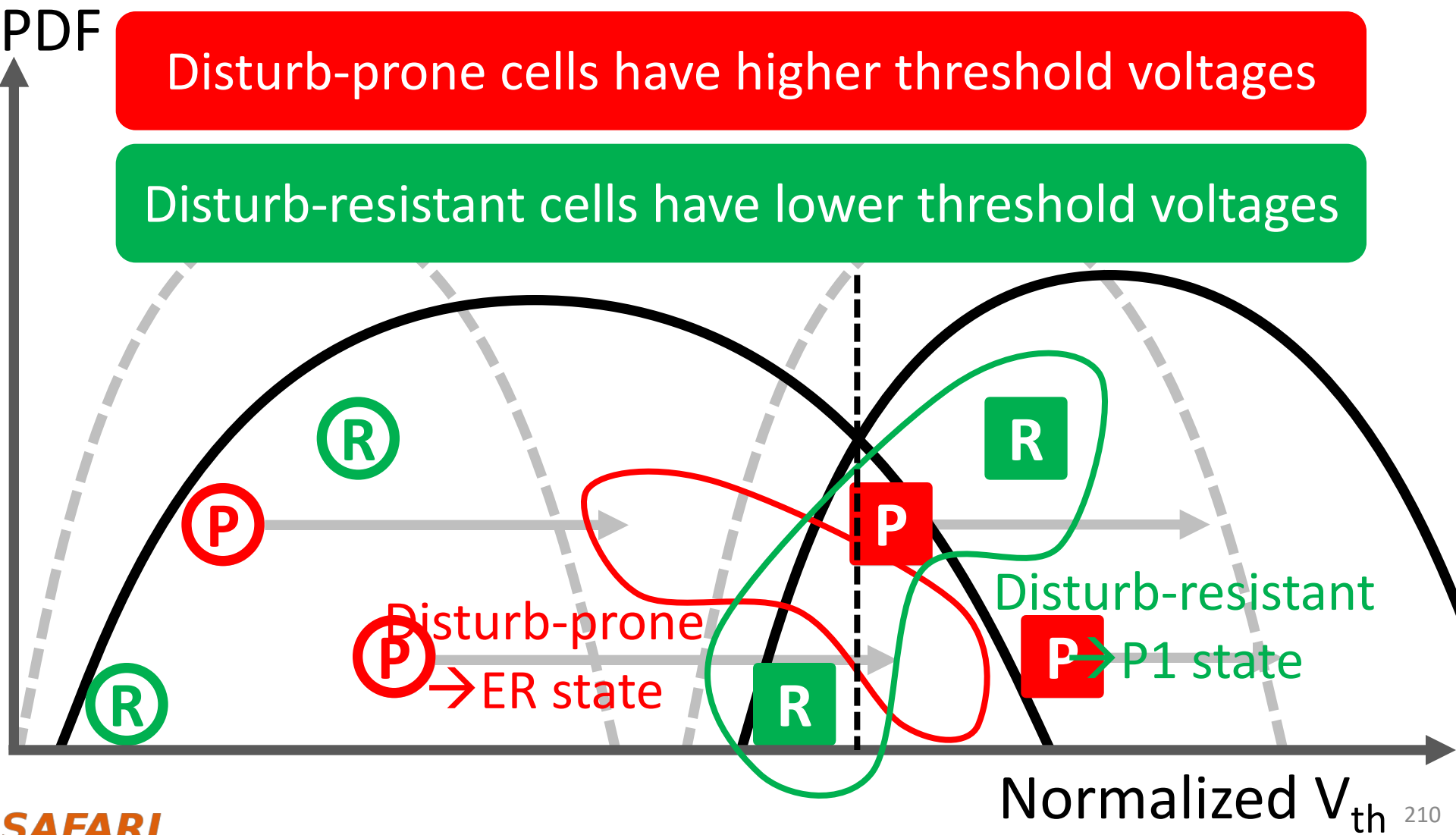


Observation 2: Some Flash Cells Are More Prone to Read Disturb

After 250K read disturbs:

Disturb-prone cells have higher threshold voltages

Disturb-resistant cells have lower threshold voltages



Read Disturb Oriented Error Recovery (RDR)

- Triggered by an uncorrectable flash error
 - Back up all valid data in the faulty block
 - Disturb the faulty page 100K times (more)
 - Compare V_{th} 's before and after read disturb
 - Select cells susceptible to flash errors ($V_{ref}-\sigma < V_{th} < V_{ref}+\sigma$)
 - Predict among these susceptible cells
 - Cells with more V_{th} shifts are disturb-prone → Higher V_{th} state
 - Cells with less V_{th} shifts are disturb-resistant → Lower V_{th} state

Reduces total error count by up to 36% @ 1M read disturbs
ECC can be used to correct the remaining errors

More on Flash Read Disturb Errors [DSN'15]

- Yu Cai, Yixin Luo, Saugata Ghose, Erich F. Haratsch, Ken Mai, and Onur Mutlu,
"Read Disturb Errors in MLC NAND Flash Memory: Characterization and Mitigation"
Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Rio de Janeiro, Brazil, June 2015.

Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery

Yu Cai, Yixin Luo, Saugata Ghose, Erich F. Haratsch*, Ken Mai, Onur Mutlu
Carnegie Mellon University, *Seagate Technology
yucaicai@gmail.com, {[yixinluo](mailto:yixinluo@cmu.edu), [ghose](mailto:ghose@cmu.edu), [kenmai](mailto:kenmai@cmu.edu), [onur](mailto:onur@cmu.edu)}@cmu.edu

Large-Scale SSD Error Analysis [SIGMETRICS'15]

- 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](#)] [[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.
qw@fb.com

Sanjeev Kumar
Facebook, Inc.
skumar@fb.com

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu

Another Lecture: NAND Flash Reliability

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

Proceedings of the IEEE, September 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.

NAND Flash Vulnerabilities [HPCA'17]

HPCA, Feb. 2017

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

Yu Cai[†] Saugata Ghose[†] Yixin Luo^{‡†} Ken Mai[†] Onur Mutlu^{§†} Erich F. Haratsch[‡]
[†]Carnegie Mellon University [‡]Seagate Technology [§]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



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>

Other Works on Flash Memory

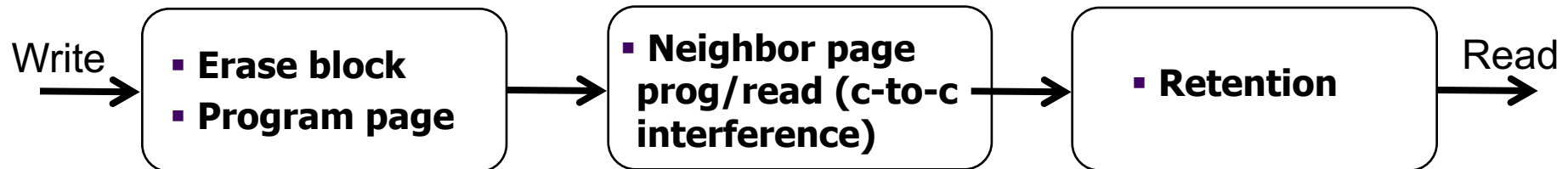
NAND Flash Error Model



Experimentally characterize and model dominant errors

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

Luo et al., "Enabling Accurate and Practical Online Flash Channel Modeling for Modern MLC NAND Flash Memory", **JSAC 2016**



Cai et al., "Threshold voltage distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling", **DATE 2013**

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

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

Cai et al., "Neighbor-Cell Assisted Error Correction in MLC NAND Flash Memories", **SIGMETRICS 2014**

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

Cai et al., "Flash Correct-and-Refresh: Retention-aware error management for increased flash memory lifetime", **ICCD 2012**

Cai et al., "Error Analysis and Retention-Aware Error Management for NAND Flash Memory", **ITJ 2013**

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

Threshold Voltage Distribution

- Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai, **"Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling"** *Proceedings of the Design, Automation, and Test in Europe Conference (**DATE**), Grenoble, France, March 2013. Slides (ppt)*

Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling

Yu Cai¹, Erich F. Haratsch², Onur Mutlu¹ and Ken Mai¹

¹DSSC, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

²LSI Corporation, 1110 American Parkway NE, Allentown, PA

¹{yucai, onur, kenmai}@andrew.cmu.edu, ²erich.haratsch@lsi.com

Program Interference and Vref Prediction

- Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai,
"Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation"
Proceedings of the 31st IEEE International Conference on Computer Design (ICCD), Asheville, NC, October 2013.
[Slides \(pptx\)](#) [\(pdf\)](#) [Lightning Session Slides \(pdf\)](#)

Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation

Yu Cai¹, Onur Mutlu¹, Erich F. Haratsch² and Ken Mai¹

1. Data Storage Systems Center, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

2. LSI Corporation, San Jose, CA

yucaicai@gmail.com, {omutlu, kenmai}@andrew.cmu.edu

Neighbor-Assisted Error Correction

- Yu Cai, Gulay Yalcin, Onur Mutlu, Eric Haratsch, Osman Unsal, Adrian Cristal, and Ken Mai,
"Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories"
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**), Austin, TX, June 2014. [Slides \(ppt\)](#) [\(pdf\)](#)*

Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories

Yu Cai¹, Gulay Yalcin², Onur Mutlu¹, Erich F. Haratsch⁴,
Osman Unsal², Adrian Cristal^{2,3}, and Ken Mai¹

¹Electrical and Computer Engineering Department, Carnegie Mellon University

²Barcelona Supercomputing Center, Spain

³IIIA – CSIC – Spain National Research Council

⁴LSI Corporation

yucaicai@gmail.com, {omutlu, kenmai}@ece.cmu.edu, {gulay.yalcin, adrian.cristal, osman.unsal}@bsc.es

Data Retention

- Yu Cai, Yixin Luo, Erich F. Haratsch, Ken Mai, and Onur Mutlu,
"Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery"
Proceedings of the 21st International Symposium on High-Performance Computer Architecture (HPCA), Bay Area, CA, February 2015.
[[Slides \(pptx\)](#)] [[pdf](#)]

Data Retention in MLC NAND Flash Memory: Characterization, Optimization, and Recovery

Yu Cai, Yixin Luo, Erich F. Haratsch*, Ken Mai, Onur Mutlu
Carnegie Mellon University, *LSI Corporation

yucaicai@gmail.com, yixinluo@cs.cmu.edu, erich.haratsch@lsi.com, {kenmai, omutlu}@ece.cmu.edu

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](#)] [[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.
qw@fb.com

Sanjeev Kumar
Facebook, Inc.
skumar@fb.com

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu

Flash Memory Programming Vulnerabilities

- Yu Cai, Saugata Ghose, Yixin Luo, Ken Mai, Onur Mutlu, and Erich F. Haratsch,
"Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques"
Proceedings of the 23rd International Symposium on High-Performance Computer Architecture (HPCA) Industrial Session, Austin, TX, USA, February 2017.
[[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Session Slides \(pptx\)](#)] [[pdf](#)]

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

Yu Cai[†] Saugata Ghose[†] Yixin Luo^{‡‡} Ken Mai[†] Onur Mutlu^{§†} Erich F. Haratsch[‡]
[†]*Carnegie Mellon University* [‡]*Seagate Technology* [§]*ETH Zürich*

Accurate and Online Channel Modeling

- Yixin Luo, Saugata Ghose, Yu Cai, Erich F. Haratsch, and Onur Mutlu,
**"Enabling Accurate and Practical Online Flash Channel Modeling
for Modern MLC NAND Flash Memory"**
to appear in IEEE Journal on Selected Areas in Communications (JSAC),
2016.

Enabling Accurate and Practical Online Flash Channel Modeling for Modern MLC NAND Flash Memory

Yixin Luo, Saugata Ghose, Yu Cai, Erich F. Haratsch, Onur Mutlu

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[†] Saugata Ghose[†] Yu Cai[‡] Erich F. Haratsch[‡] Onur Mutlu^{§†}
[†]*Carnegie Mellon University* [‡]*Seagate Technology* [§]*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](#)]

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

Yixin Luo[†]

Saugata Ghose[†]

Yu Cai[†]

Erich F. Haratsch[‡]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[‡]Seagate Technology

[§]ETH Zürich

Memory Systems

Fundamentals, Recent Research, Challenges, Opportunities

Lecture 2c: DRAM Refresh

Prof. Onur Mutlu

omutlu@gmail.com

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

8 October 2018

Technion Fast Course 2018

More on DRAM Refresh

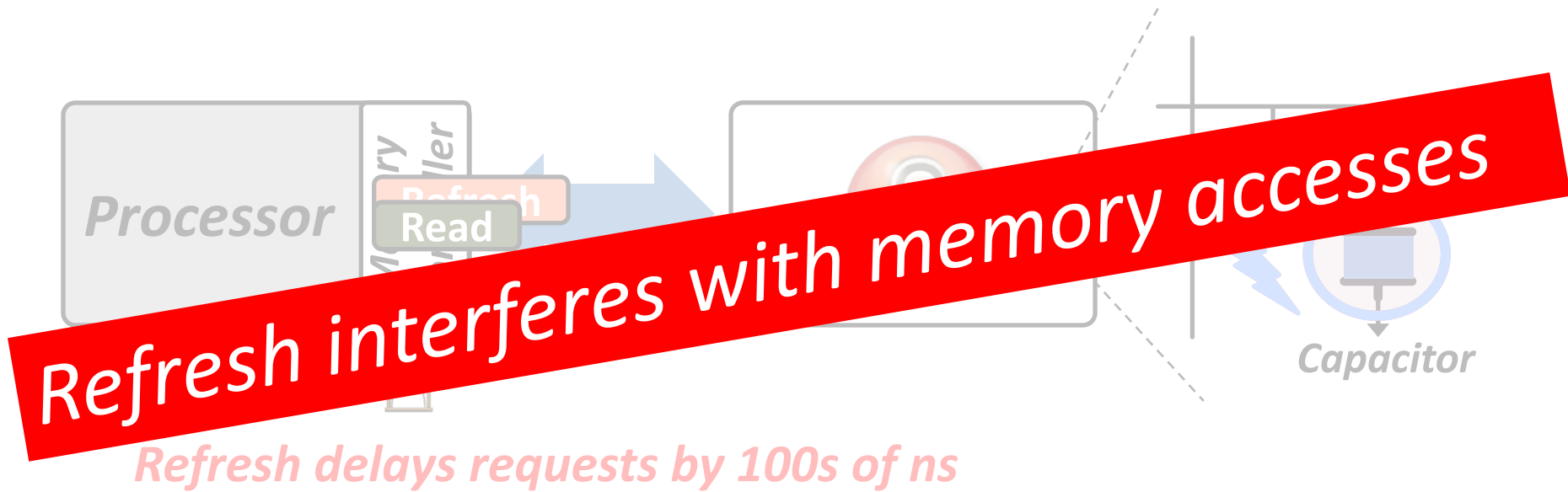
Tackling Refresh: Solutions

- Parallelize refreshes with accesses [Chang+ HPCA'14]
- Eliminate unnecessary refreshes [Liu+ ISCA'12]
 - Exploit device characteristics
 - Exploit data and application characteristics
- Reduce refresh rate and detect+correct errors that occur [Khan+ SIGMETRICS'14]
- Understand retention time behavior in DRAM [Liu+ ISCA'13]

Summary: Refresh-Access Parallelization

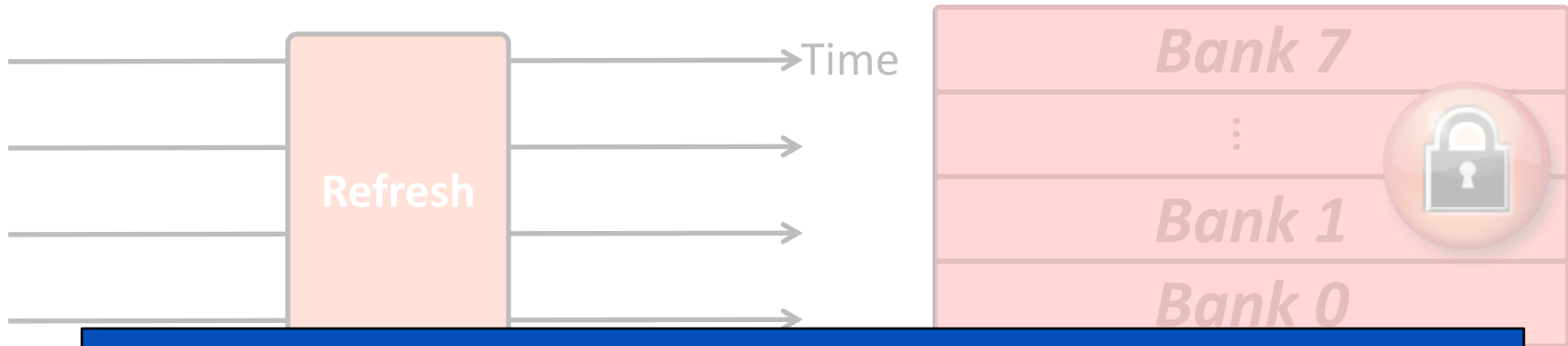
- **DRAM refresh interferes with memory accesses**
 - Degrades system performance and energy efficiency
 - Becomes exacerbated as DRAM density increases
- Goal: Serve memory accesses in parallel with refreshes to reduce refresh interference on demand requests
- Our mechanisms:
 - 1. Enable more parallelization between refreshes and accesses across different banks with **new per-bank refresh scheduling algorithms**
 - 2. Enable serving accesses concurrently with refreshes in the same bank by **exploiting parallelism across DRAM subarrays**
- Improve system performance and energy efficiency for a wide variety of different workloads and DRAM densities
 - 20.2% and 9.0% for 8-core systems using 32Gb DRAM at low cost
 - Very close to the ideal scheme without refreshes

Refresh Penalty



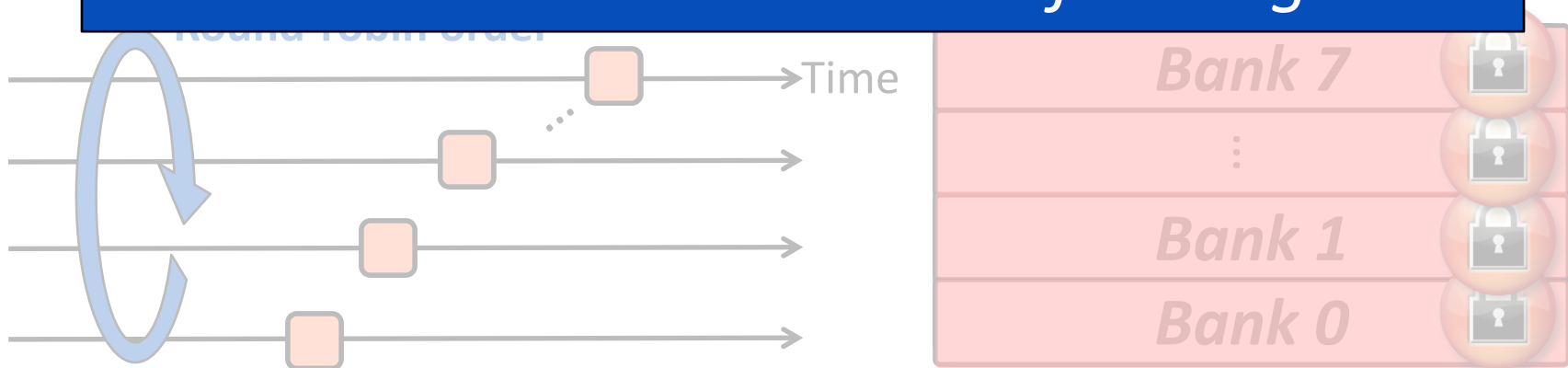
Existing Refresh Modes

All-bank refresh in commodity DRAM (DDR_x)



Per-bank refresh allows accesses to other banks while a bank is refreshing

Per



Shortcomings of Per-Bank Refresh

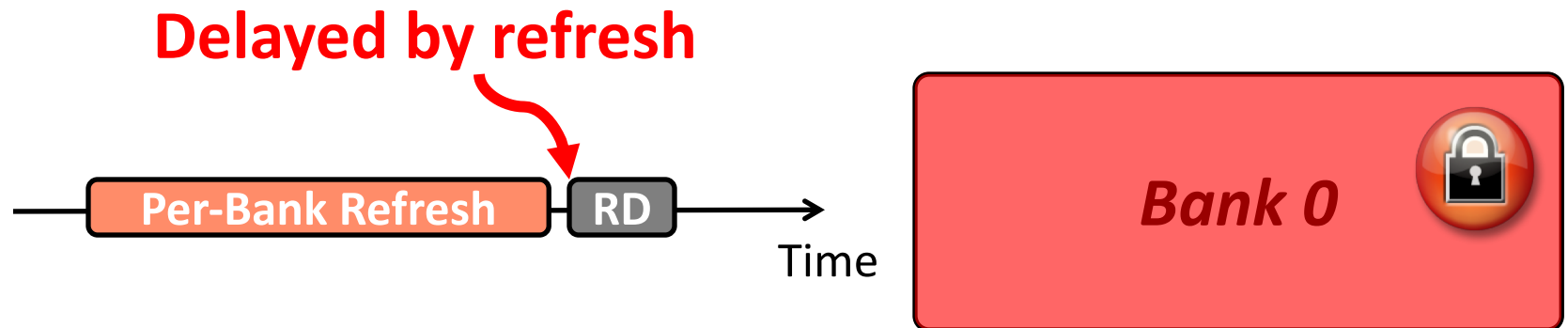
- Problem 1: Refreshes to different banks are scheduled in a **strict round-robin order**
 - The static ordering is hardwired into DRAM chips
 - Refreshes busy banks with many queued requests when other banks are idle
- Key idea: Schedule per-bank refreshes to idle banks opportunistically in a dynamic order

Our First Approach: DARP

- **Dynamic Access-Refresh Parallelization (DARP)**
 - An improved scheduling policy for **per-bank refreshes**
 - Exploits **refresh scheduling flexibility** in DDR DRAM
- Component 1: **Out-of-order per-bank refresh**
 - Avoids poor static scheduling decisions
 - Dynamically issues per-bank refreshes to idle banks
- Component 2: **Write-Refresh Parallelization**
 - Avoids refresh interference on latency-critical reads
 - Parallelizes refreshes with **a batch of writes**

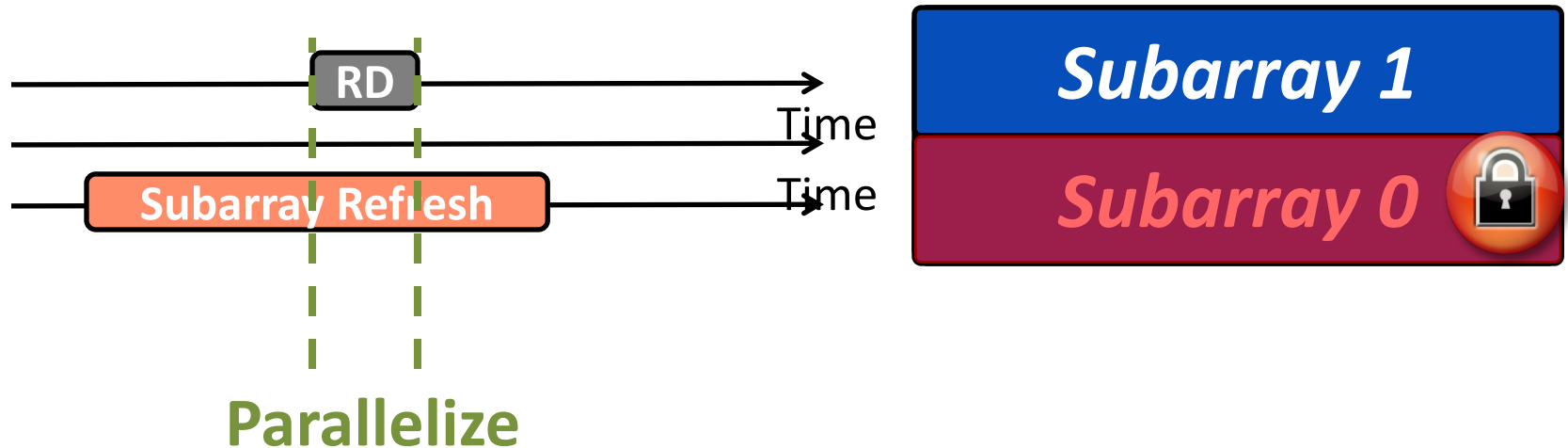
Shortcomings of Per-Bank Refresh

- Problem 2: Banks that are being refreshed cannot concurrently serve memory requests



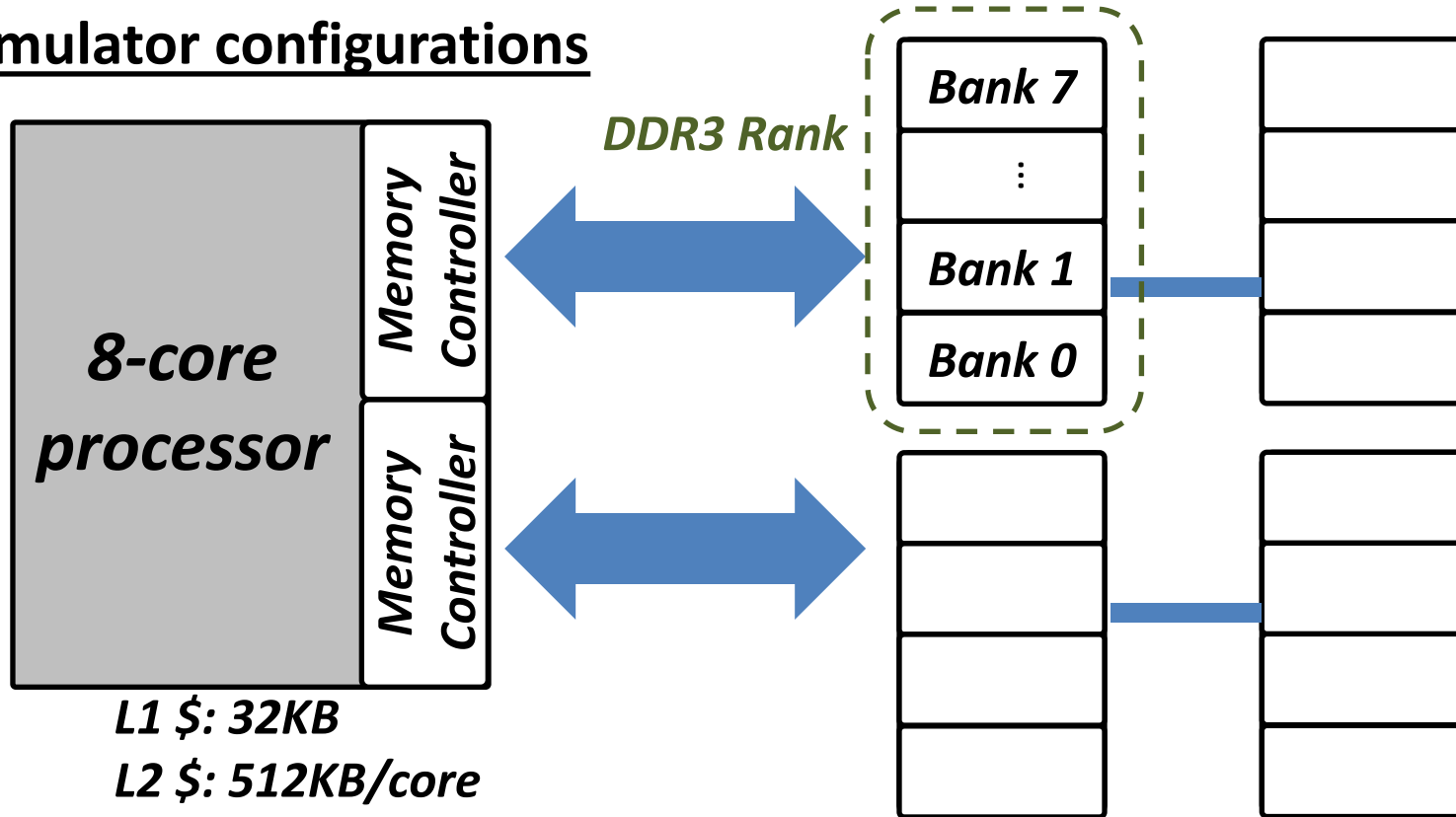
Shortcomings of Per-Bank Refresh

- Problem 2: Refreshing banks cannot concurrently serve memory requests
- Key idea: Exploit **subarrays** within a bank to parallelize refreshes and accesses across **subarrays**



Methodology

Simulator configurations

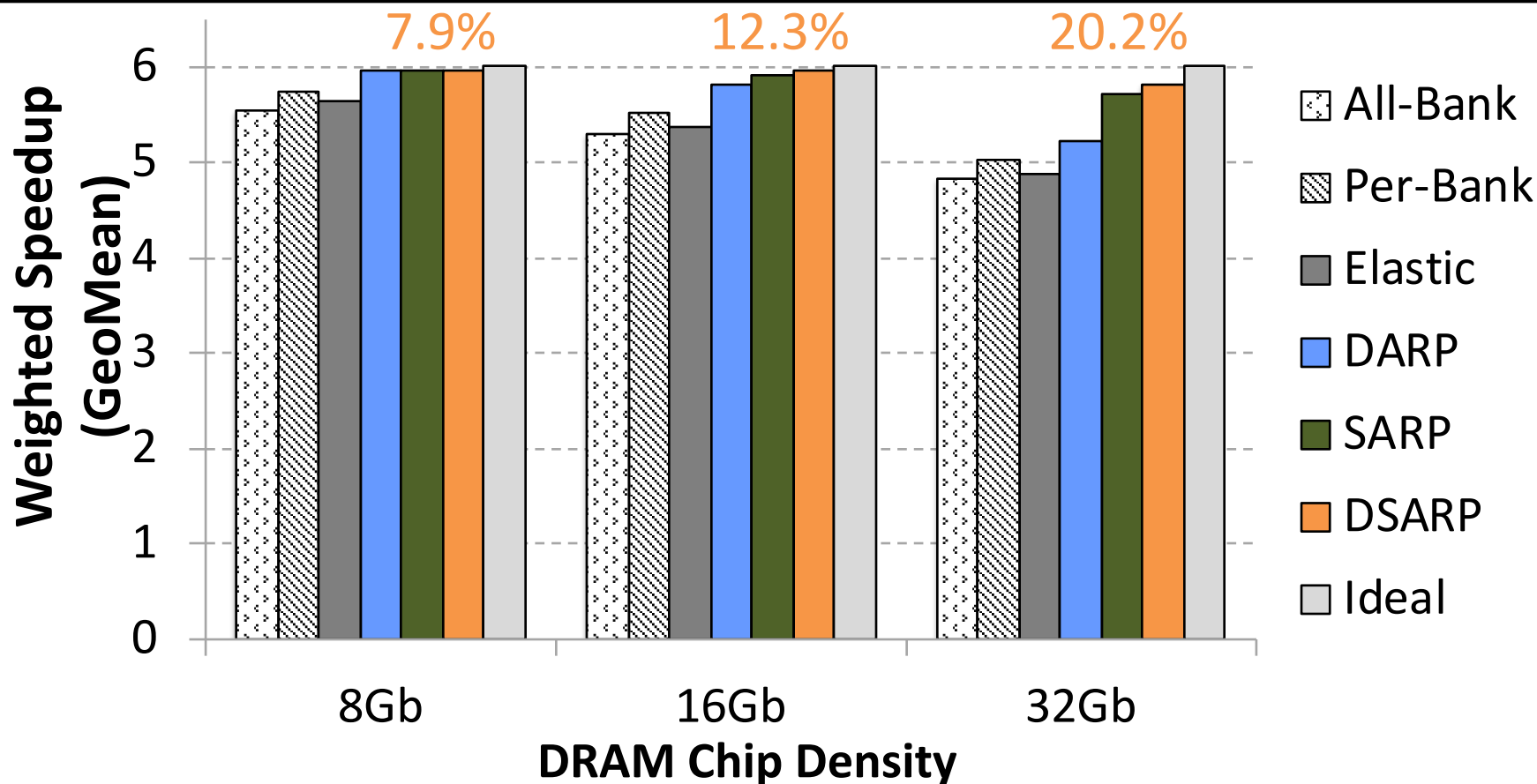


- **100 workloads**: SPEC CPU2006, STREAM, TPC-C/H, random access
- **System performance metric**: *Weighted speedup*

Comparison Points

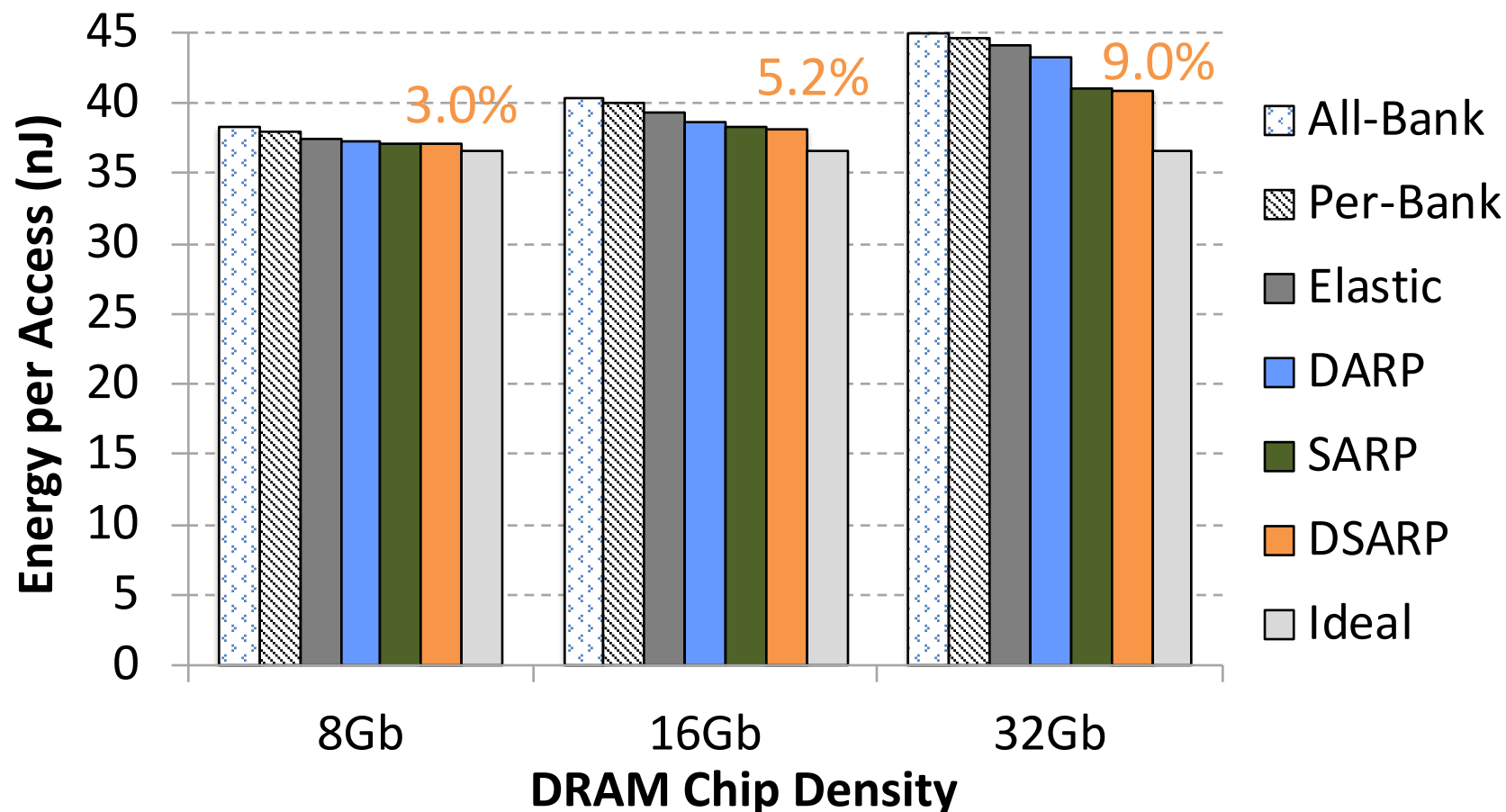
- **All-bank refresh** [DDR3, LPDDR3, ...]
- **Per-bank refresh** [LPDDR3]
- **Elastic refresh** [Stuecheli et al., MICRO '10]:
 - Postpones refreshes by a time delay based on the predicted rank idle time to avoid interference on memory requests
 - Proposed to schedule all-bank refreshes without exploiting per-bank refreshes
 - Cannot parallelize refreshes and accesses within a rank
- **Ideal (no refresh)**

System Performance



2. Consistent system performance improvement across DRAM densities (within 0.9%, 1.2%, and 3.8% of ideal)

Energy Efficiency



Consistent reduction on energy consumption

More Information on Refresh-Access Parallelization

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

Reducing Performance Impact of DRAM Refresh by Parallelizing Refreshes with Accesses

Kevin Kai-Wei Chang Donghyuk Lee Zeshan Chishti†

Alaa R. Alameldeen† Chris Wilkerson† Yoongu Kim Onur Mutlu

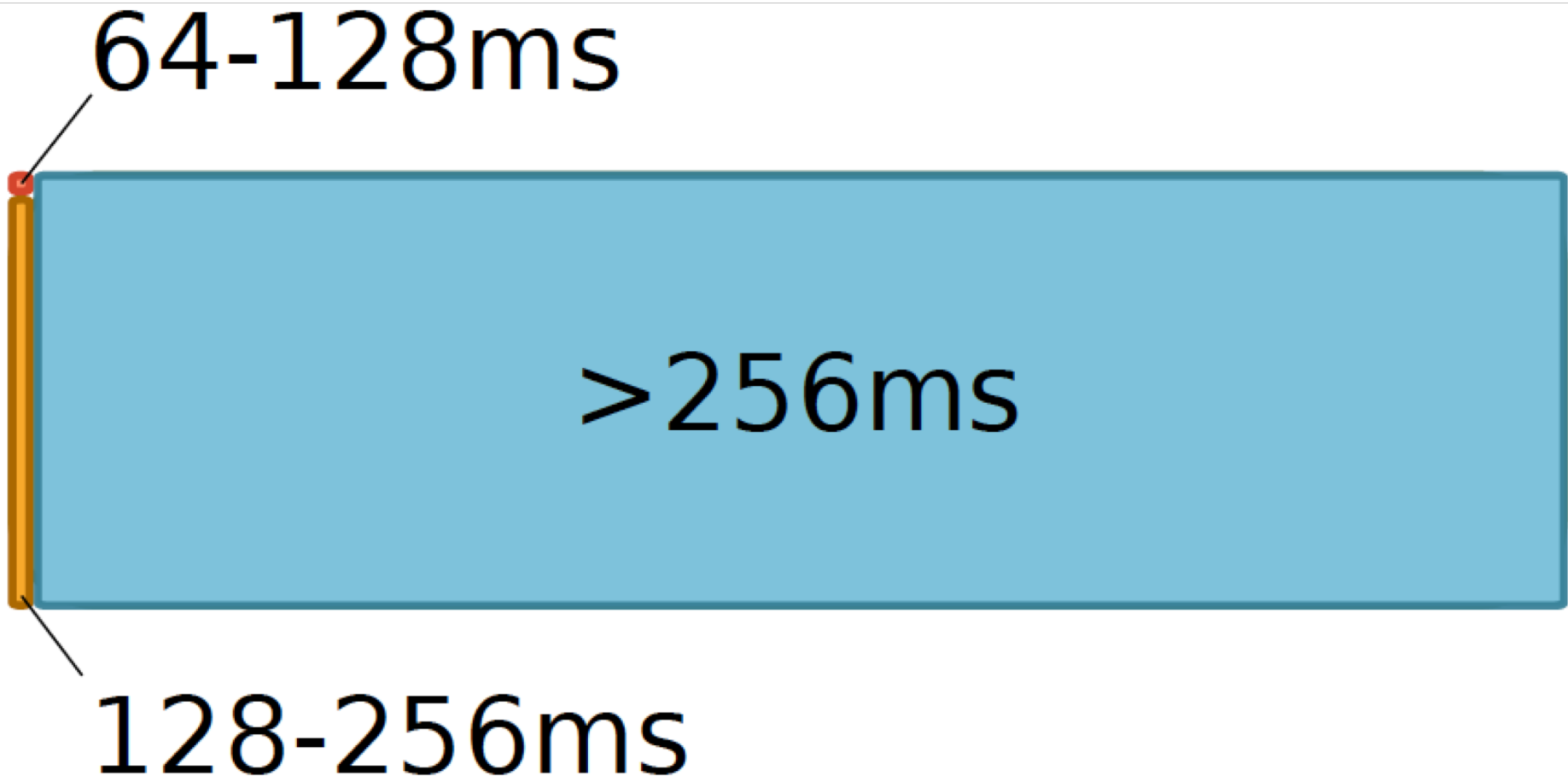
Carnegie Mellon University †Intel Labs

Tackling Refresh: Solutions

- Parallelize refreshes with accesses [Chang+ HPCA'14]
- Eliminate unnecessary refreshes [Liu+ ISCA'12]
 - Exploit device characteristics
 - Exploit data and application characteristics
- Reduce refresh rate and detect+correct errors that occur [Khan+ SIGMETRICS'14]
- Understand retention time behavior in DRAM [Liu+ ISCA'13]

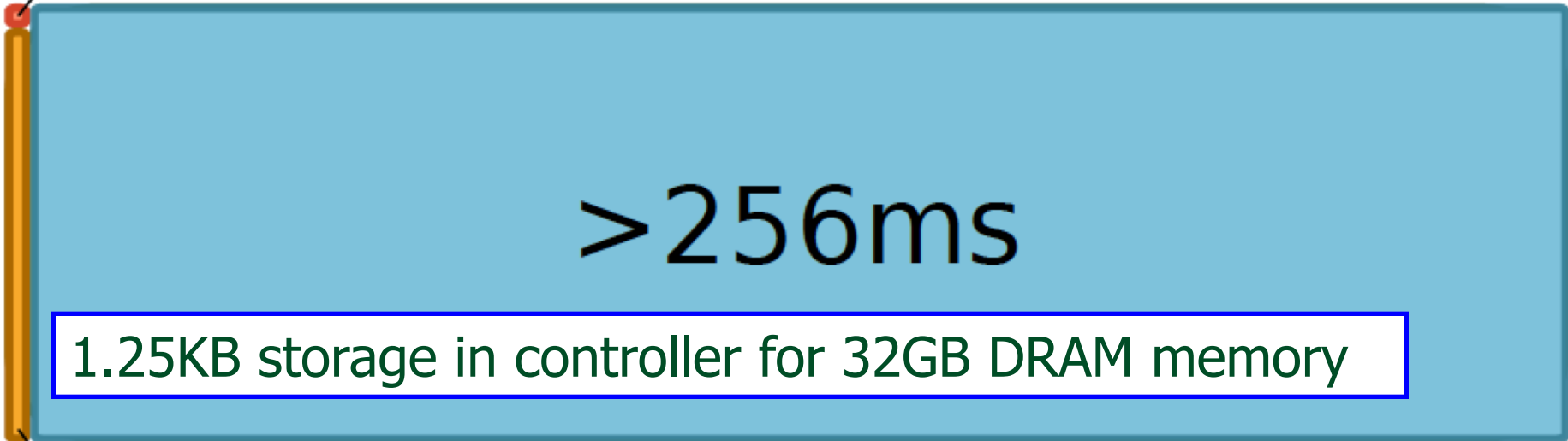
Most Refreshes Are Unnecessary

- Retention Time Profile of DRAM looks like this:



RAIDR: Eliminating Unnecessary Refreshes

64-128ms



>256ms

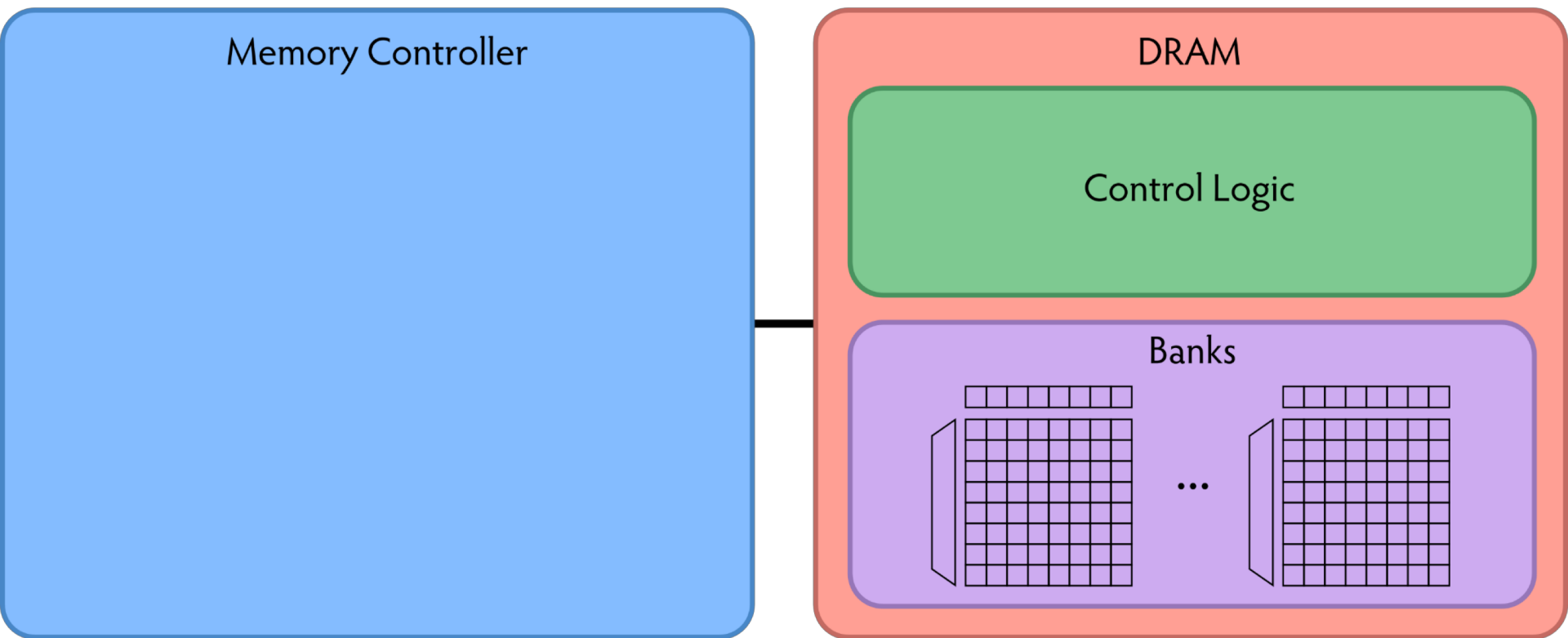
1.25KB storage in controller for 32GB DRAM memory

128-256ms

Can reduce refreshes by $\sim 75\%$

→ reduces energy consumption and improves performance

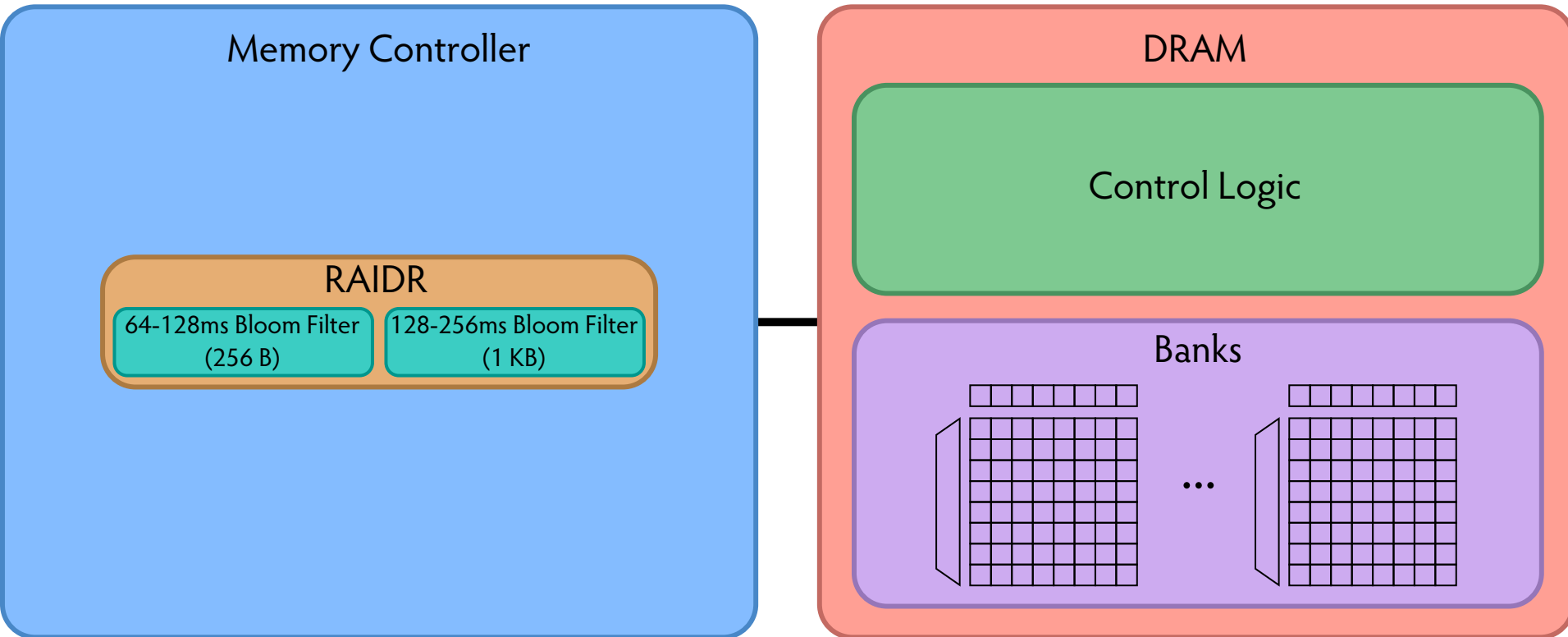
RAIDR: Baseline Design



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

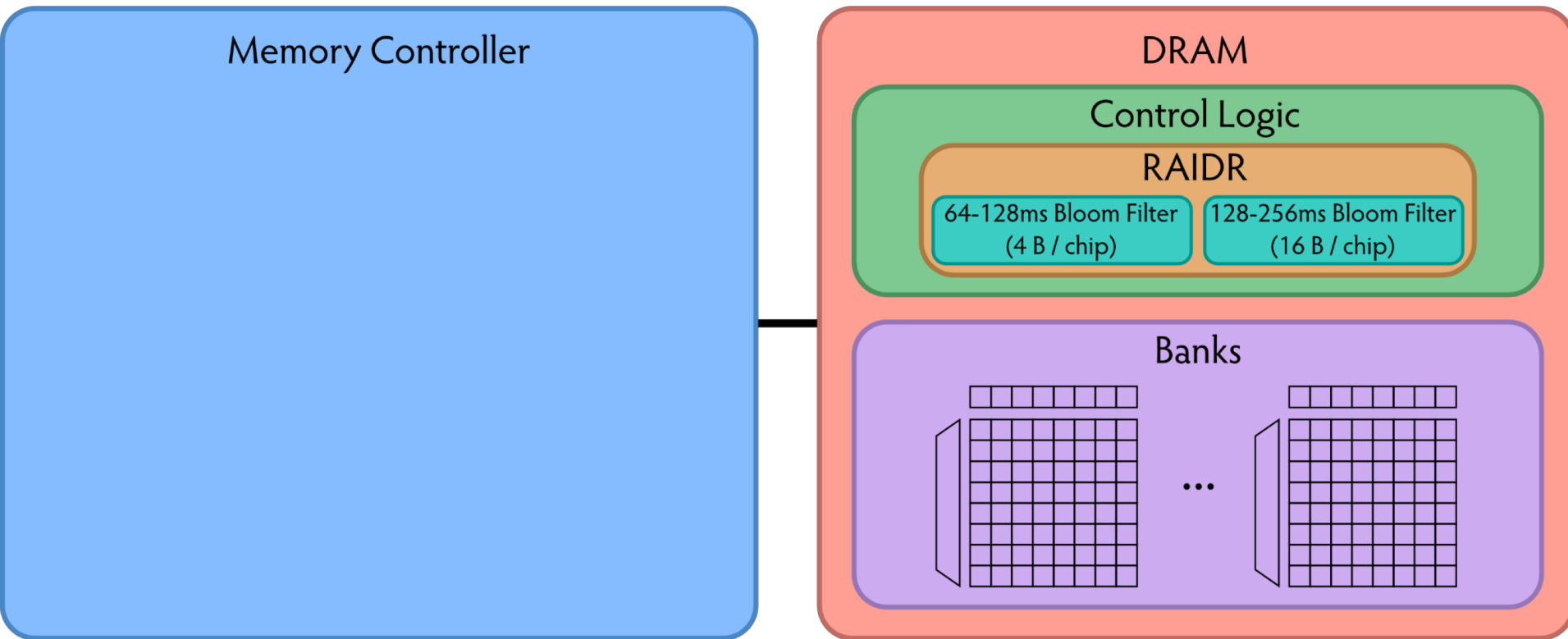
RAIDR can be implemented in either the controller or DRAM

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



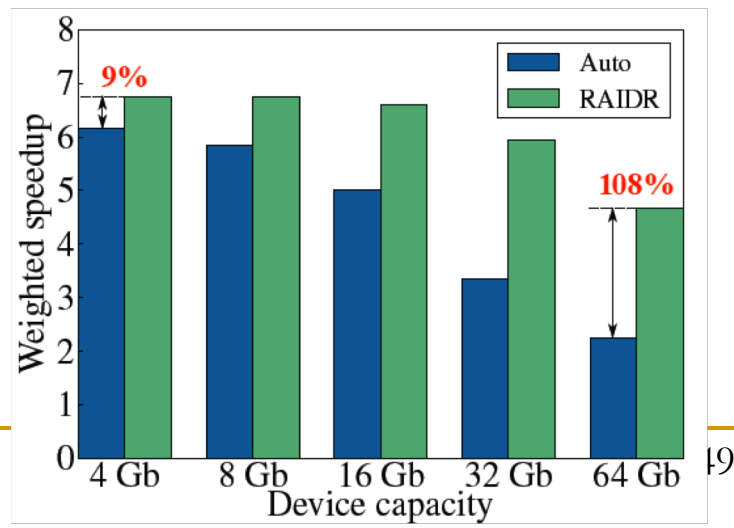
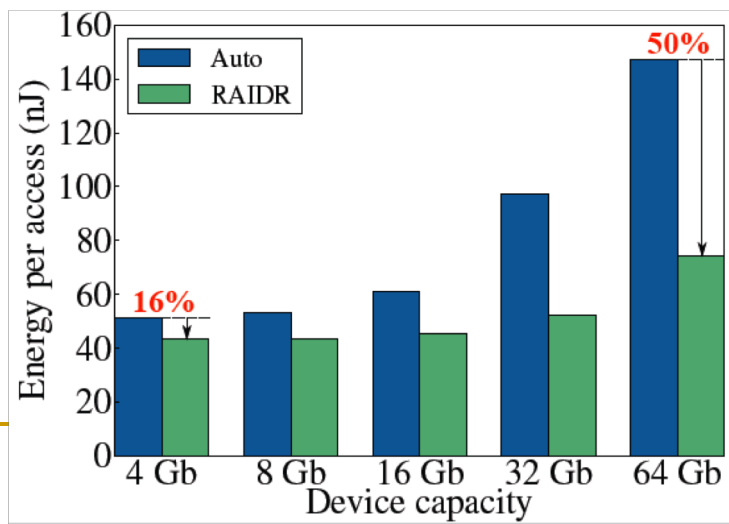
Overhead of RAIDR in DRAM chip:

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

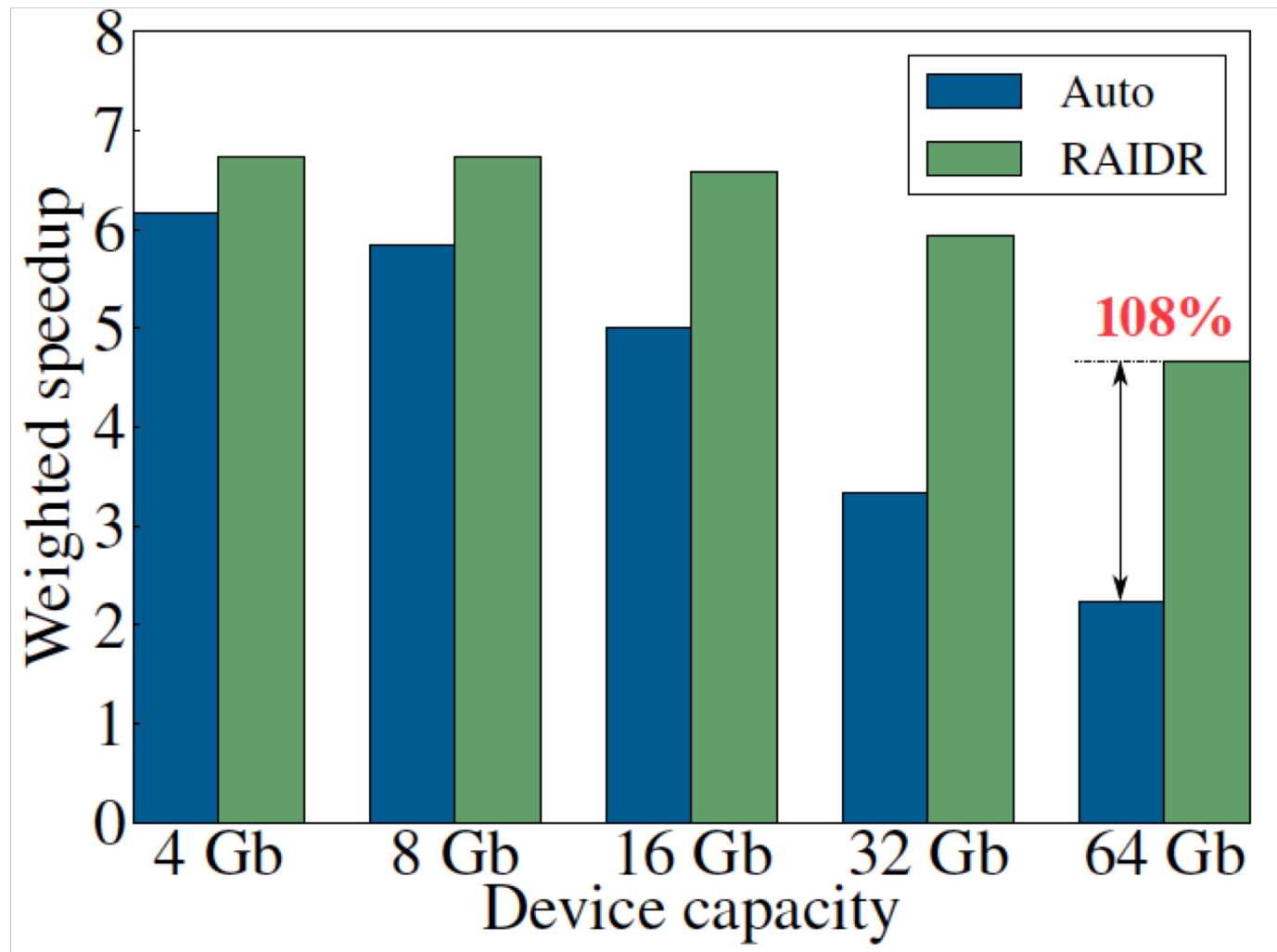
Total overhead: 1.25KB Bloom Filters, 64 counters (32 GB DRAM)

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

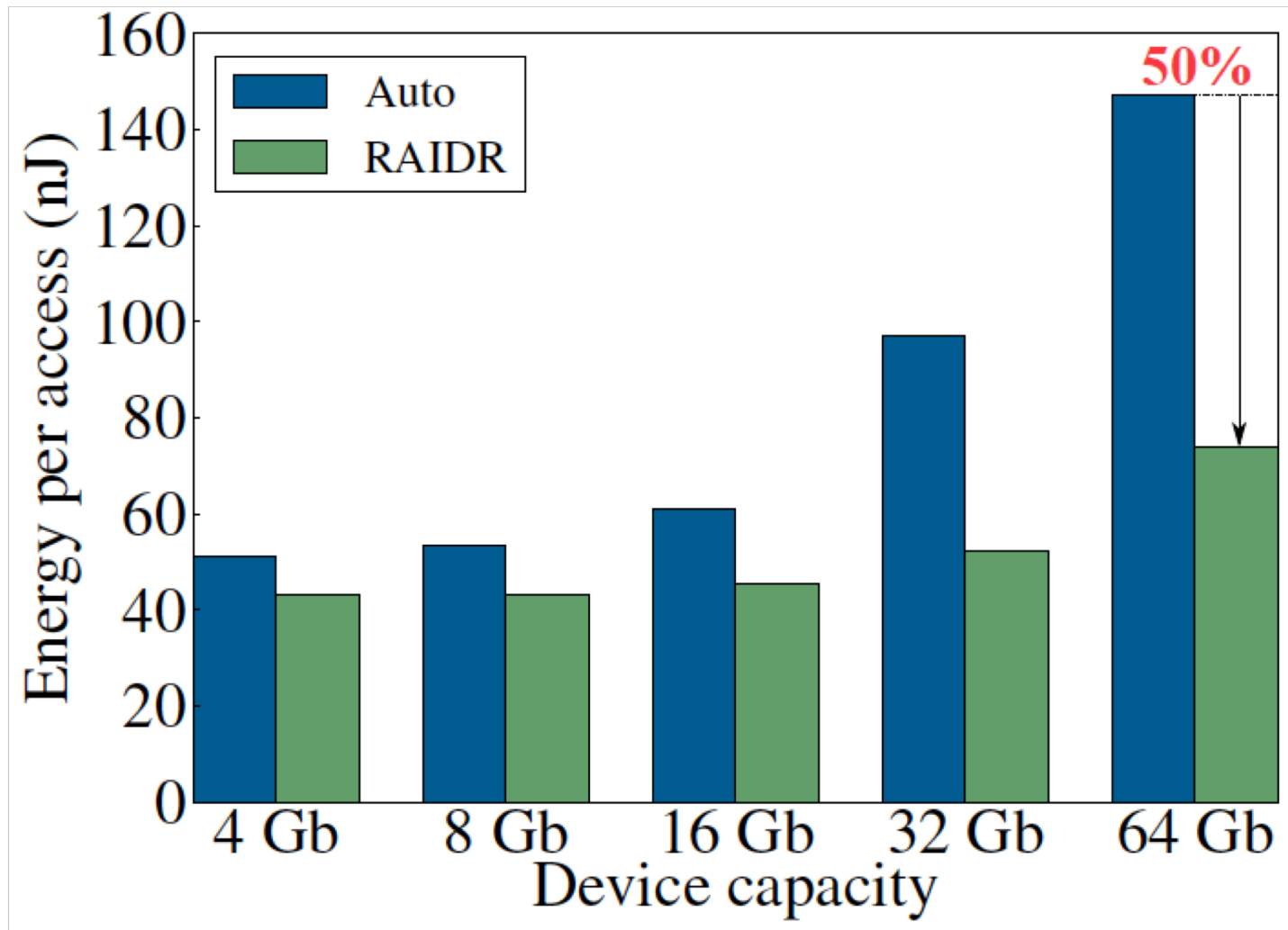


DRAM Device Capacity Scaling: Performance



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

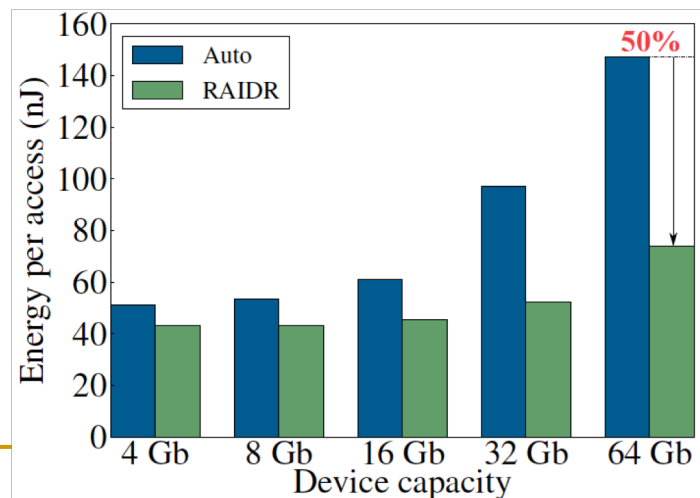
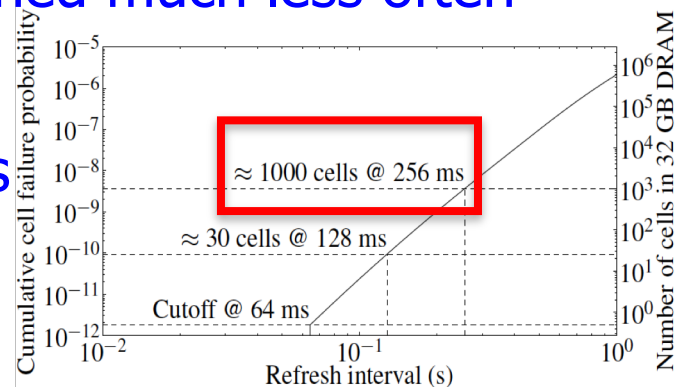
2. **Binning:** Store rows into bins by retention time in memory controller

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

- 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

Tackling Refresh: Solutions

- Parallelize refreshes with accesses [Chang+ HPCA'14]
- Eliminate unnecessary refreshes [Liu+ ISCA'12]
 - Exploit device characteristics
 - Exploit data and application characteristics
- Reduce refresh rate and detect+correct errors that occur [Khan+ SIGMETRICS'14]
- Understand retention time behavior in DRAM [Liu+ ISCA'13]

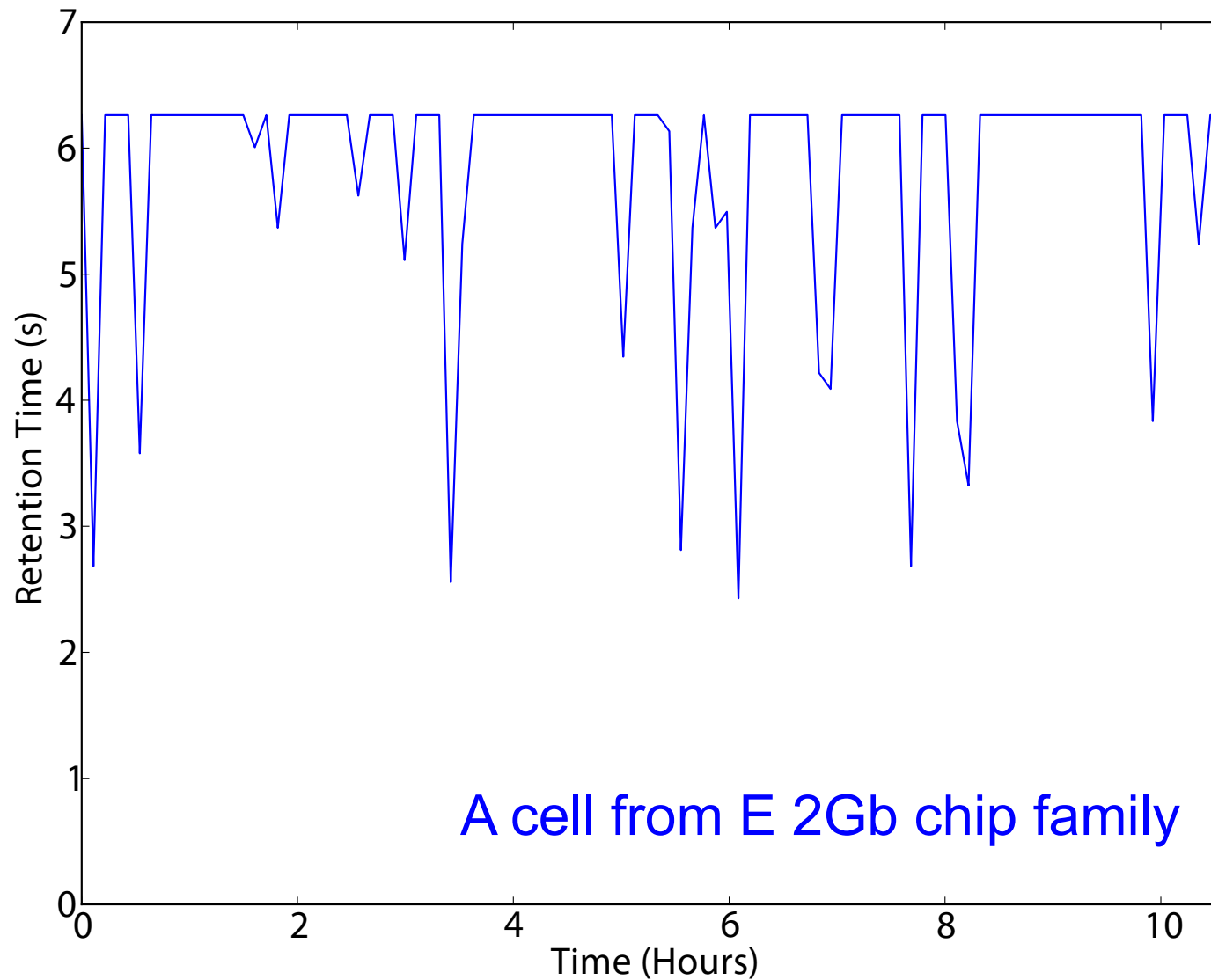
Motivation: Understanding Retention

- Past works require **accurate and reliable measurement of retention time of each DRAM row**
 - To maintain data integrity while reducing refreshes
- Assumption: **worst-case retention time of each row can be determined and stays the same at a given temperature**
 - Some works propose writing all 1's and 0's to a row, and measuring the time before data corruption
- Question:
 - Can we reliably and accurately determine retention times of all DRAM rows?

Two Challenges to Retention Time Profiling

- Data Pattern Dependence (DPD) of retention time
- Variable Retention Time (VRT) phenomenon

An Example VRT Cell



VRT: Implications on Profiling Mechanisms

- Problem 1: There does not seem to be a way of determining if a cell exhibits VRT without actually observing a cell exhibiting VRT
 - VRT is a memoryless random process [Kim+ JJAP 2010]
- Problem 2: VRT complicates retention time profiling by DRAM manufacturers
 - Exposure to very high temperatures can induce VRT in cells that were not previously susceptible
 - can happen during soldering of DRAM chips
 - manufacturer's retention time profile may not be accurate
- One option for future work: Use ECC to continuously profile DRAM online while aggressively reducing refresh rate
 - Need to keep ECC overhead in check

More on DRAM Retention Analysis

- 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

Tackling Refresh: Solutions

- Parallelize refreshes with accesses [Chang+ HPCA'14]
- Eliminate unnecessary refreshes [Liu+ ISCA'12]
 - Exploit device characteristics
 - Exploit data and application characteristics
- Reduce refresh rate and detect+correct errors that occur [Khan+ SIGMETRICS'14]
- Understand retention time behavior in DRAM [Liu+ ISCA'13]

Towards an Online Profiling System

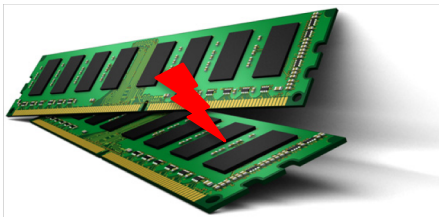
Key Observations:

- **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

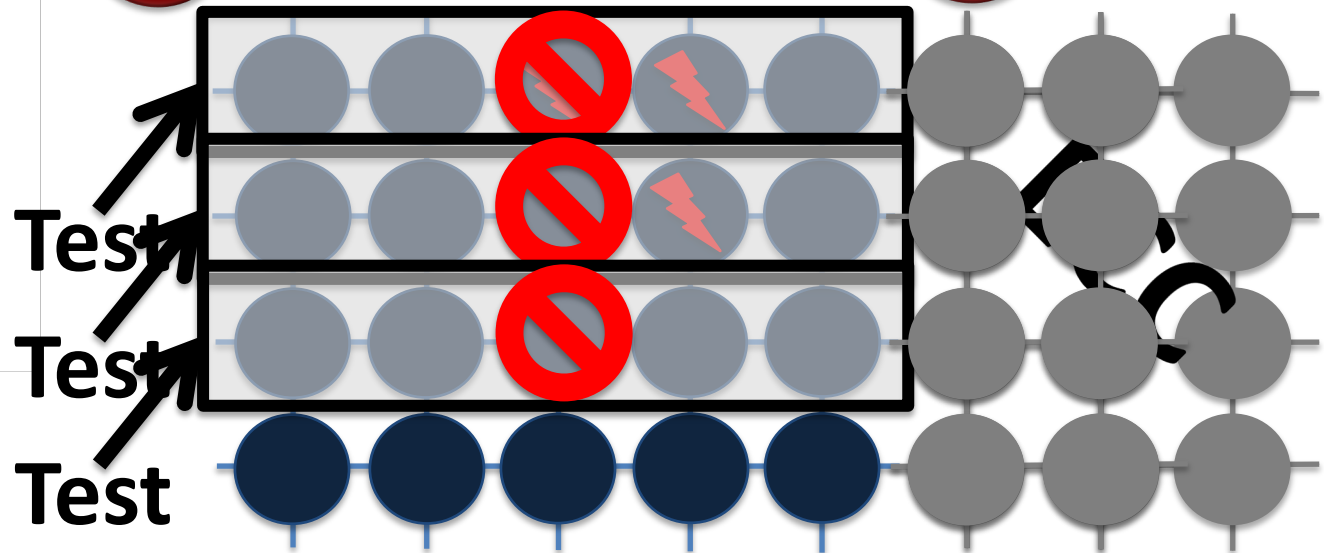
Initially Protect DRAM
with Strong ECC

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

More on Online Profiling of DRAM

- 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^{†*}
samirakhan@cmu.edu

Donghyuk Lee[†]
donghyuk1@cmu.edu

Yoongu Kim[†]
yoongukim@cmu.edu

Alaa R. Alameldeen^{*}
alaa.r.alameldeen@intel.com

Chris Wilkerson^{*}
chris.wilkerson@intel.com

Onur Mutlu[†]
onur@cmu.edu

[†]Carnegie Mellon University

^{*}Intel Labs

How Do We Make RAIDR Work in the Presence of the VRT Phenomenon?

Making RAIDR Work w/ Online Profiling & ECC

- 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](#)]

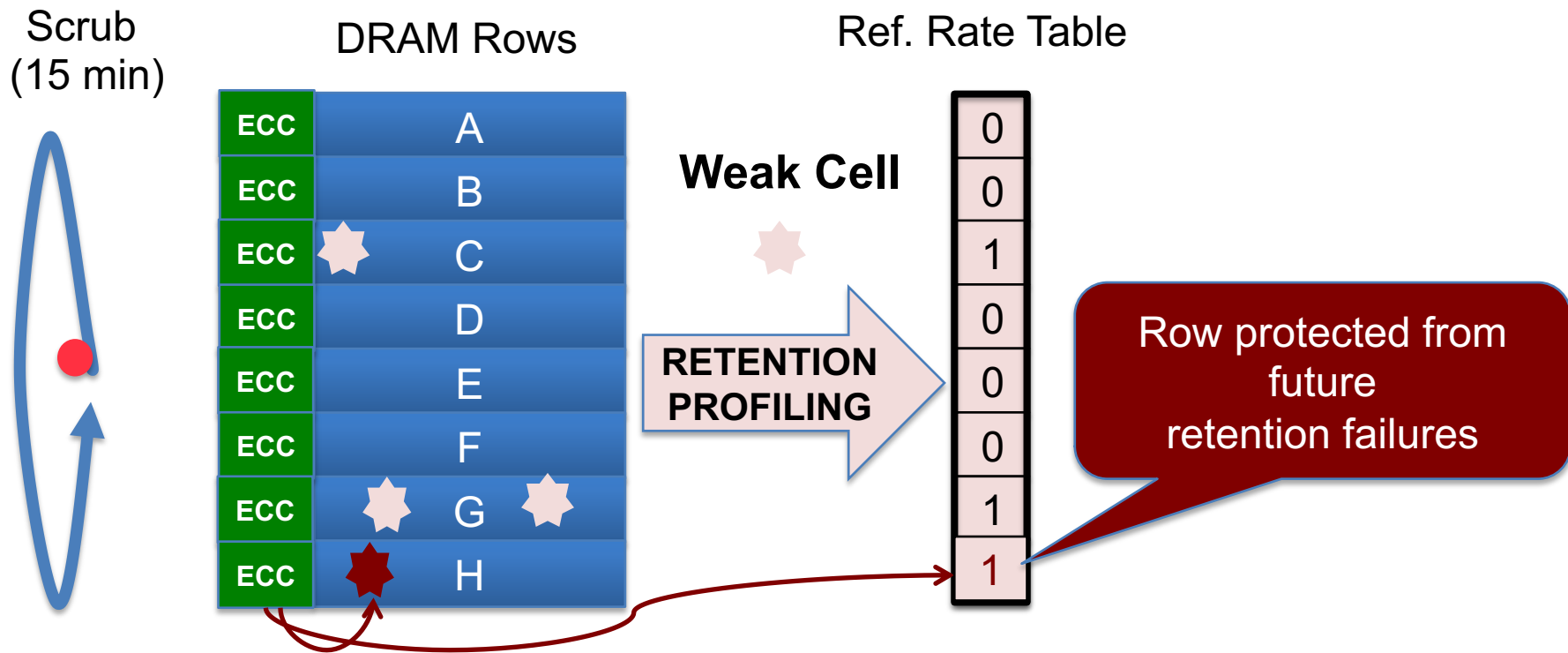
AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems

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

AVATAR

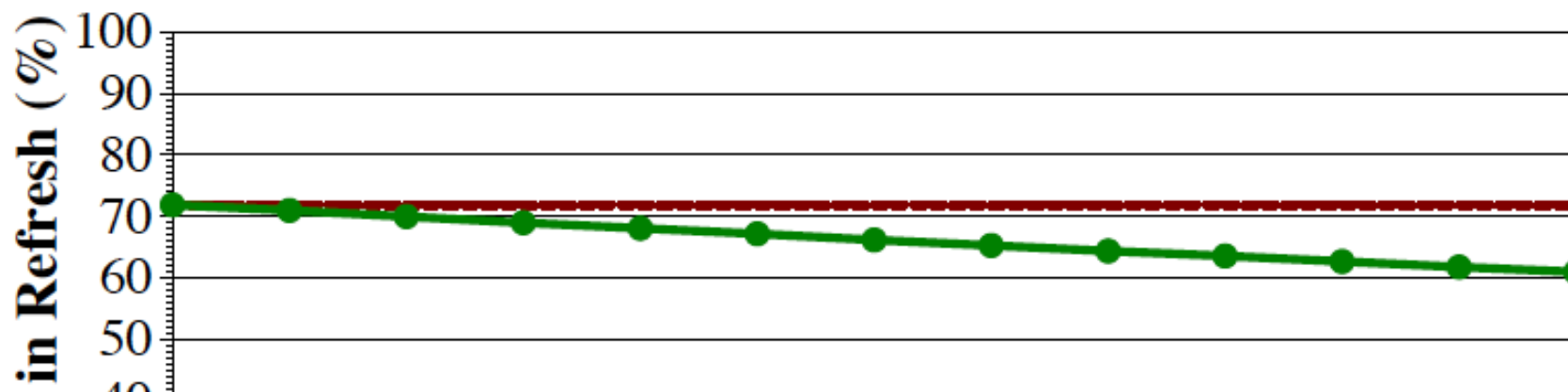
Insight: Avoid retention failures → Upgrade row on ECC error

Observation: Rate of VRT >> Rate of soft error (50x-2500x)

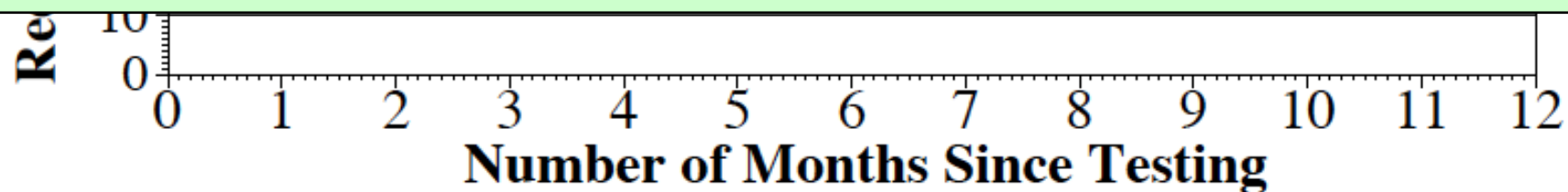


AVATAR mitigates VRT by increasing refresh rate on error

RESULTS: REFRESH SAVINGS

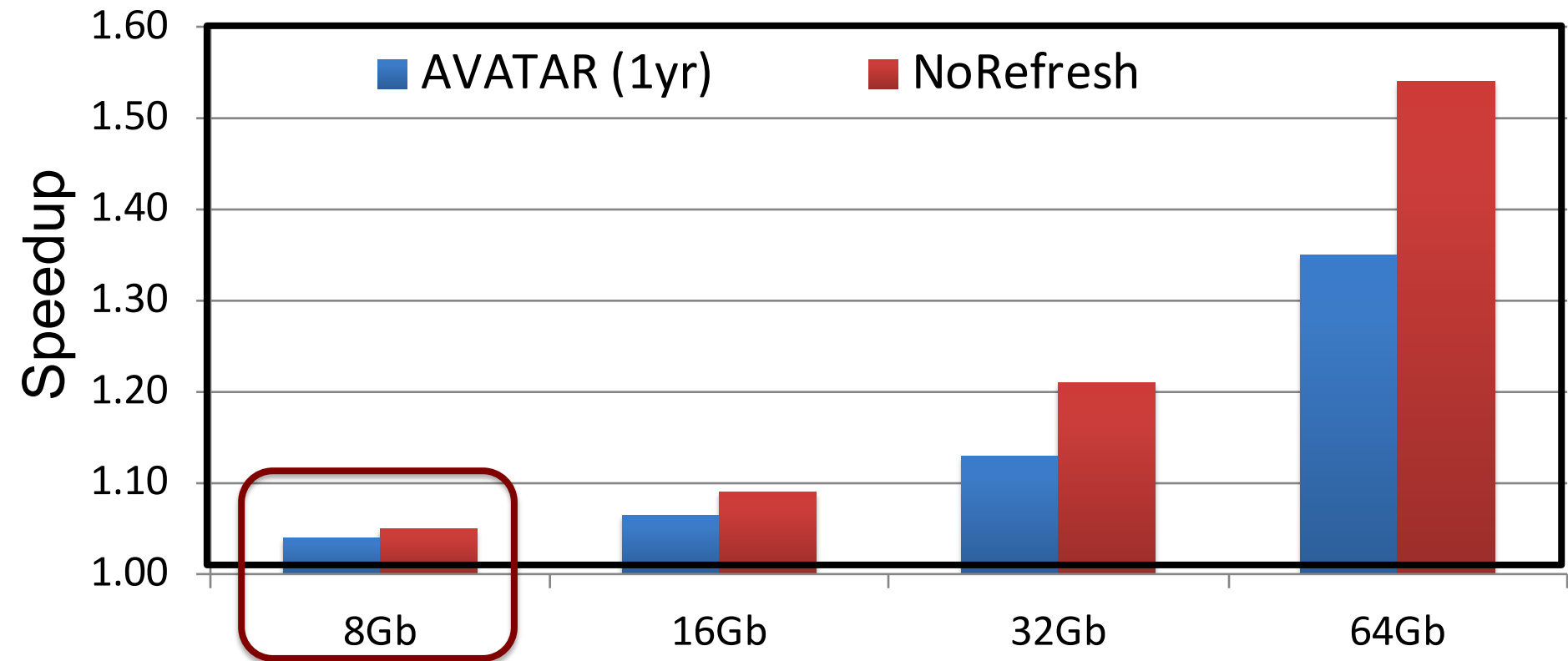


Retention Testing Once a Year can revert refresh saving from 60% to 70%



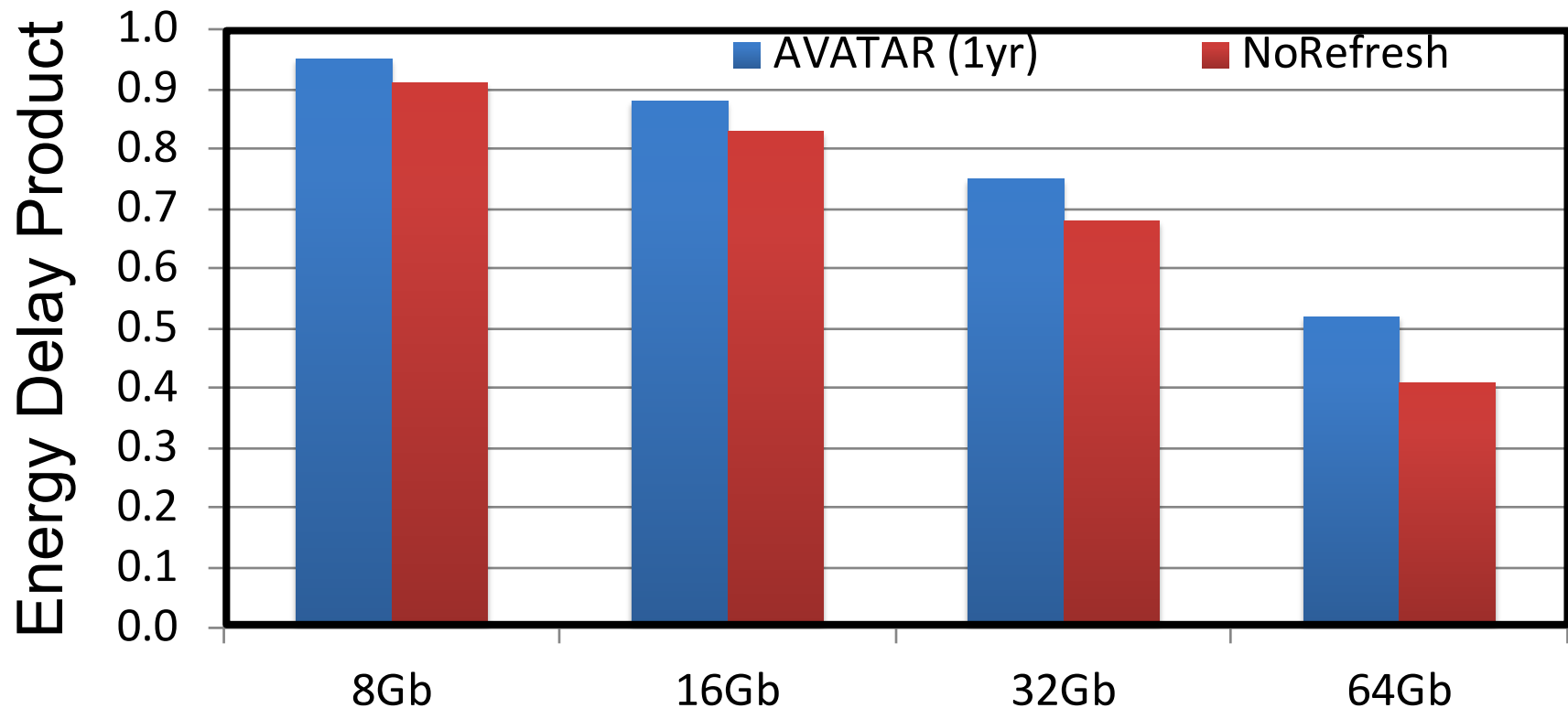
AVATAR reduces refresh by 60%-70%, similar to multi rate refresh but with VRT tolerance

SPEEDUP



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

ENERGY DELAY PRODUCT



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

Making RAIDR Work w/ Online Profiling & ECC

- 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 [†]	Dae-Hyun Kim [†]	Samira Khan [‡]	Prashant J. Nair [†]	Onur Mutlu [‡]
[†] Georgia Institute of Technology		[‡] Carnegie Mellon University		
{moin, dhkim, pnair6}@ece.gatech.edu		{samirakhan, onur}@cmu.edu		

DRAM Refresh: Summary and Conclusions

- **DRAM refresh is a critical challenge**
 - in scaling DRAM technology efficiently to higher capacities
- **Discussed several promising solution directions**
 - Parallelize refreshes with accesses [Chang+ HPCA'14]
 - Eliminate unnecessary refreshes [Liu+ ISCA'12]
 - Reduce refresh rate and detect+correct errors that occur [Khan+ SIGMETRICS'14]
- **Examined properties of retention time behavior** [Liu+ ISCA'13]
 - Enable realistic VRT-Aware refresh techniques [Qureshi+ DSN'15]
- **Many avenues for overcoming DRAM refresh challenges**
 - Handling DPD/VRT phenomena
 - Enabling online retention time profiling and error mitigation
 - Exploiting application behavior

Memory Systems

Fundamentals, Recent Research, Challenges, Opportunities

Lecture 2c: DRAM Refresh

Prof. Onur Mutlu

omutlu@gmail.com

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

8 October 2018

Technion Fast Course 2018