Understanding and Overcoming Challenges of DRAM Refresh

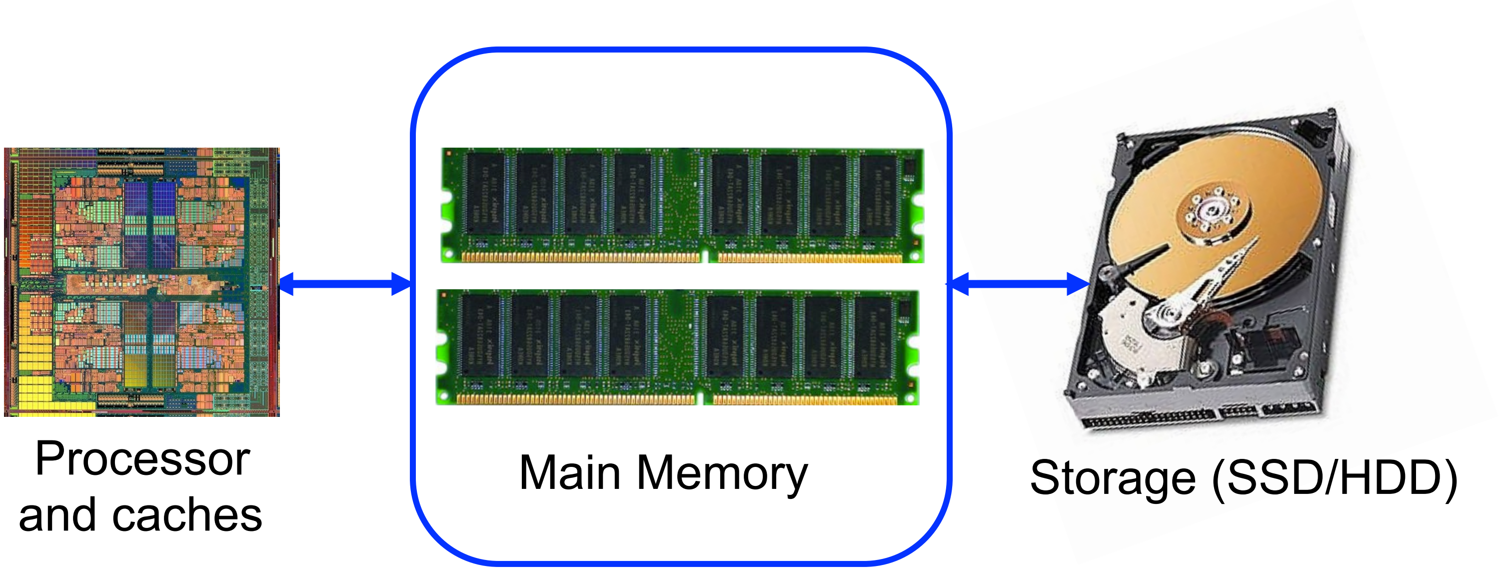
Onur Mutlu
onur@cmu.edu

June 30, 2014
Extreme Scale Scientific Computing Workshop

Carnegie Mellon

SAFARI

The Main Memory System



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

State of the Main Memory System

- Recent technology, architecture, and application trends
  - lead to new requirements
  - exacerbate old requirements

- DRAM and memory controllers, as we know them today, are (will be) unlikely to satisfy all requirements

- Some emerging non-volatile memory technologies (e.g., PCM) enable new opportunities: memory+storage merging

- We need to rethink the main memory system
  - to fix DRAM issues and enable emerging technologies
  - to satisfy all requirements

Agenda

- Major Trends Affecting Main Memory
- The DRAM Scaling Problem
- Refresh as a Limiter of DRAM Scaling
- Solution Directions and Challenges
- Summary

Major Trends Affecting Main Memory (I)

- Need for main memory capacity, bandwidth, QoS increasing

- Main memory energy/power is a key system design concern

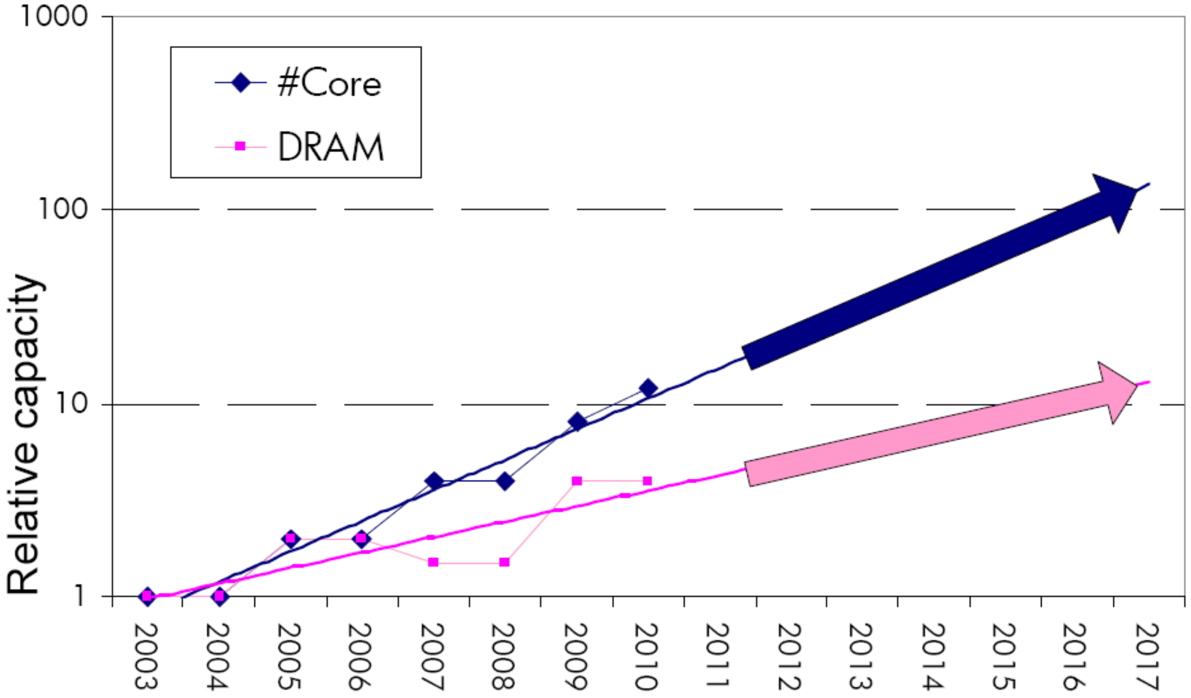
- DRAM technology scaling is ending

Major Trends Affecting Main Memory (II)

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

Example: The Memory Capacity Gap

Core count doubling ~ every 2 years
DRAM DIMM capacity doubling ~ every 3 years



Source: Lim et al., ISCA 2009.

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

Major Trends Affecting Main Memory (III)

- Need for main memory capacity, bandwidth, QoS increasing

- Main memory energy/power is a key system design concern
  - ~40-50% energy spent in off-chip memory hierarchy [Lefurgy, IEEE Computer 2003]
  - DRAM consumes power even when not used (periodic refresh)

- DRAM technology scaling is ending

Major Trends Affecting Main Memory (IV)

- Need for main memory capacity, bandwidth, QoS increasing

- Main memory energy/power is a key system design concern

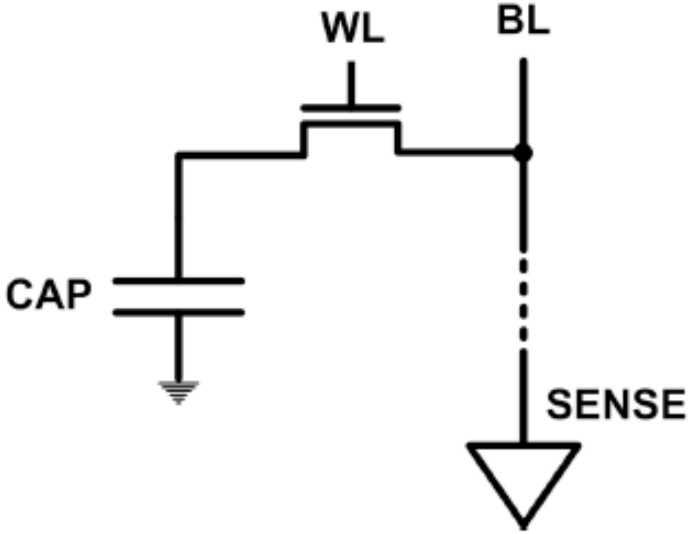
- DRAM technology scaling is ending
  - ITRS projects DRAM will not scale easily below X nm
  - Scaling has provided many benefits:
    - higher capacity (density), lower cost, lower energy

Agenda

- Major Trends Affecting Main Memory
- The DRAM Scaling Problem
- Refresh as a Limiter of DRAM Scaling
- Challenges and Solution Directions
- Summary

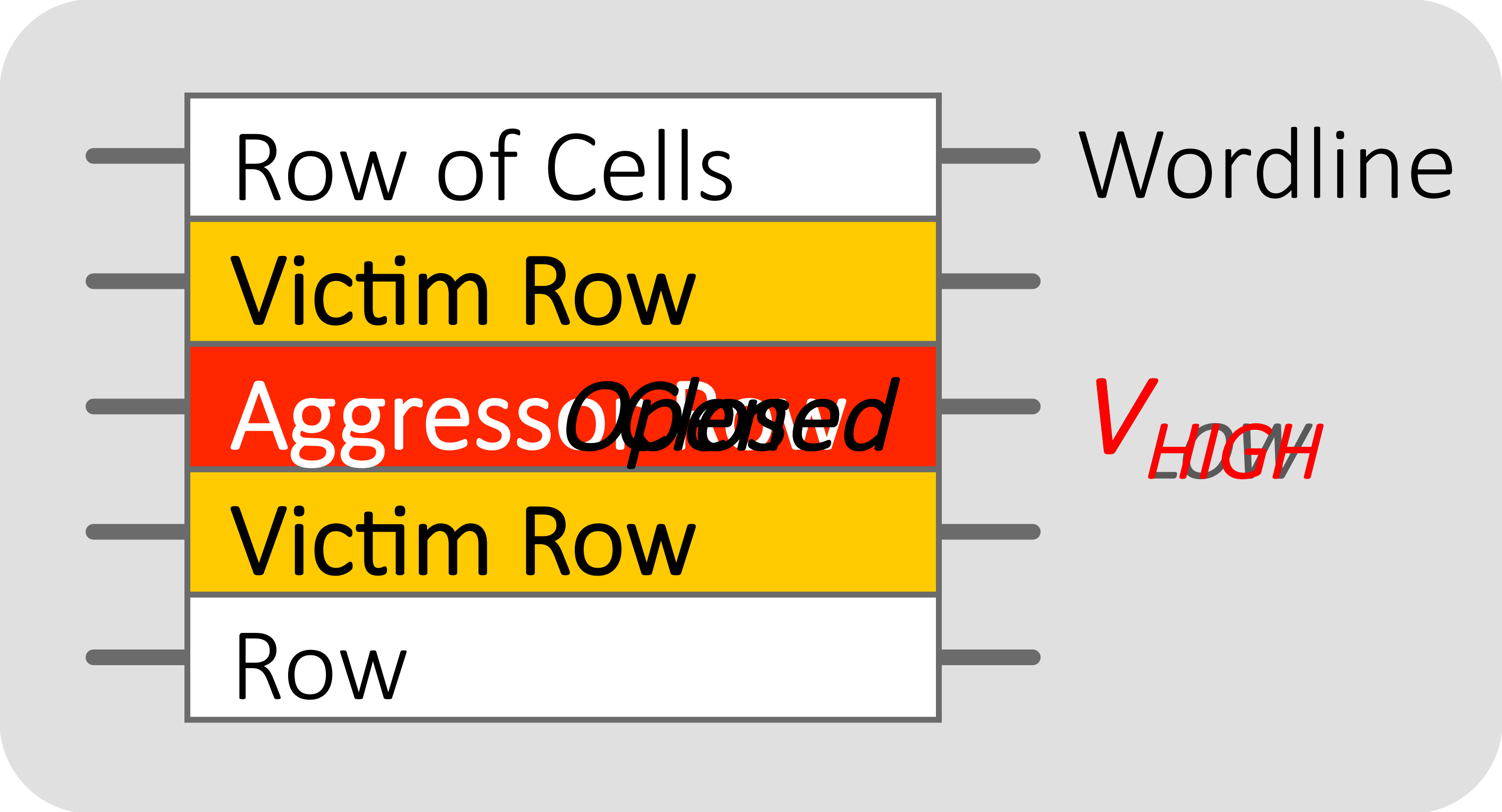
The DRAM Scaling Problem

- 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

An Example of The Scaling Problem



Row of Cells

Victim Row

Aggressor Row

Opened

Closed

Victim Row

Row

Wordline

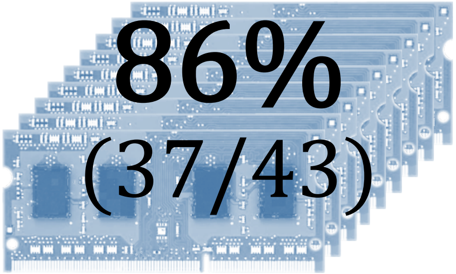
V_{LOW}

V_{HIGH}

*Repeatedly opening and closing a row induces **disturbance errors** in adjacent rows in **most real DRAM chips** [Kim+ ISCA 2014]*

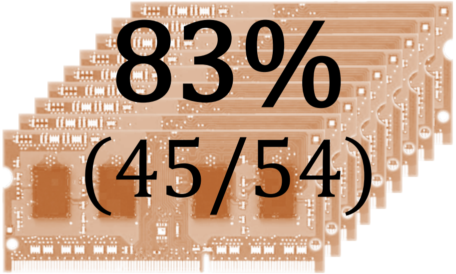
Most DRAM Modules Are at Risk

A company


86%
(37/43)

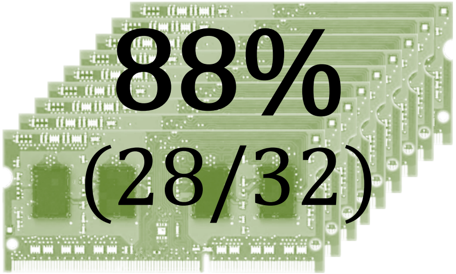
Up to
1.0×10^{7}
errors

B company


83%
(45/54)

Up to
2.7×10^{6}
errors

C company


88%
(28/32)

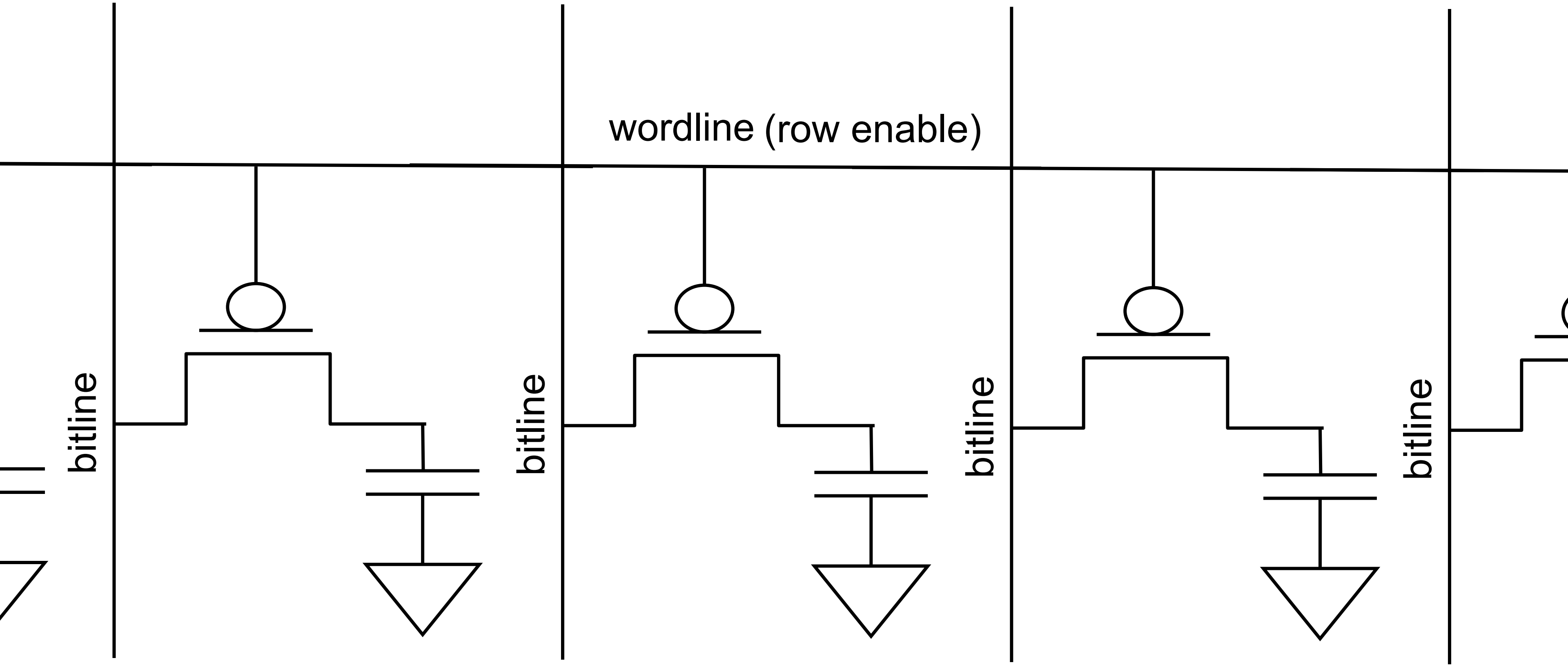
Up to
3.3×10^{5}
errors

Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

Agenda

- Major Trends Affecting Main Memory
- The DRAM Scaling Problem
- Refresh as a Limiter of DRAM Scaling
- Some Solution Directions and Challenges
- Summary

A DRAM Cell



- A DRAM cell consists of a capacitor and an access transistor
- It stores data in terms of charge in the capacitor
- A DRAM chip consists of (10s of 1000s of) rows of such cells

DRAM Refresh

- DRAM capacitor charge leaks over time

- Each DRAM row is periodically refreshed to restore charge
  - Activate each row every N ms
  - Typical N = 64 ms

- Downsides of refresh

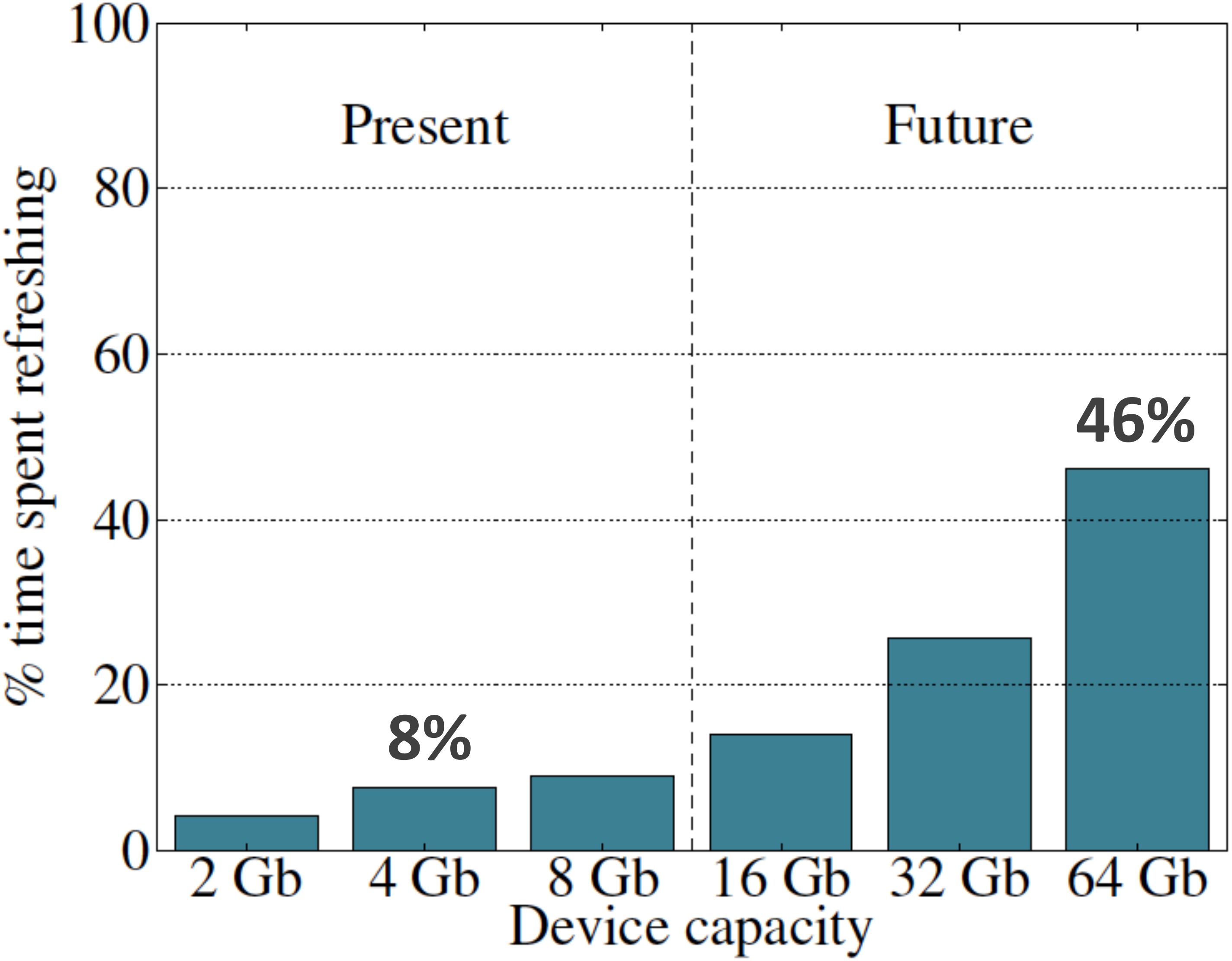
  -- Energy consumption: Each refresh consumes energy

  -- Performance degradation: DRAM rank/bank unavailable while refreshed

  -- QoS/predictability impact: (Long) pause times during refresh

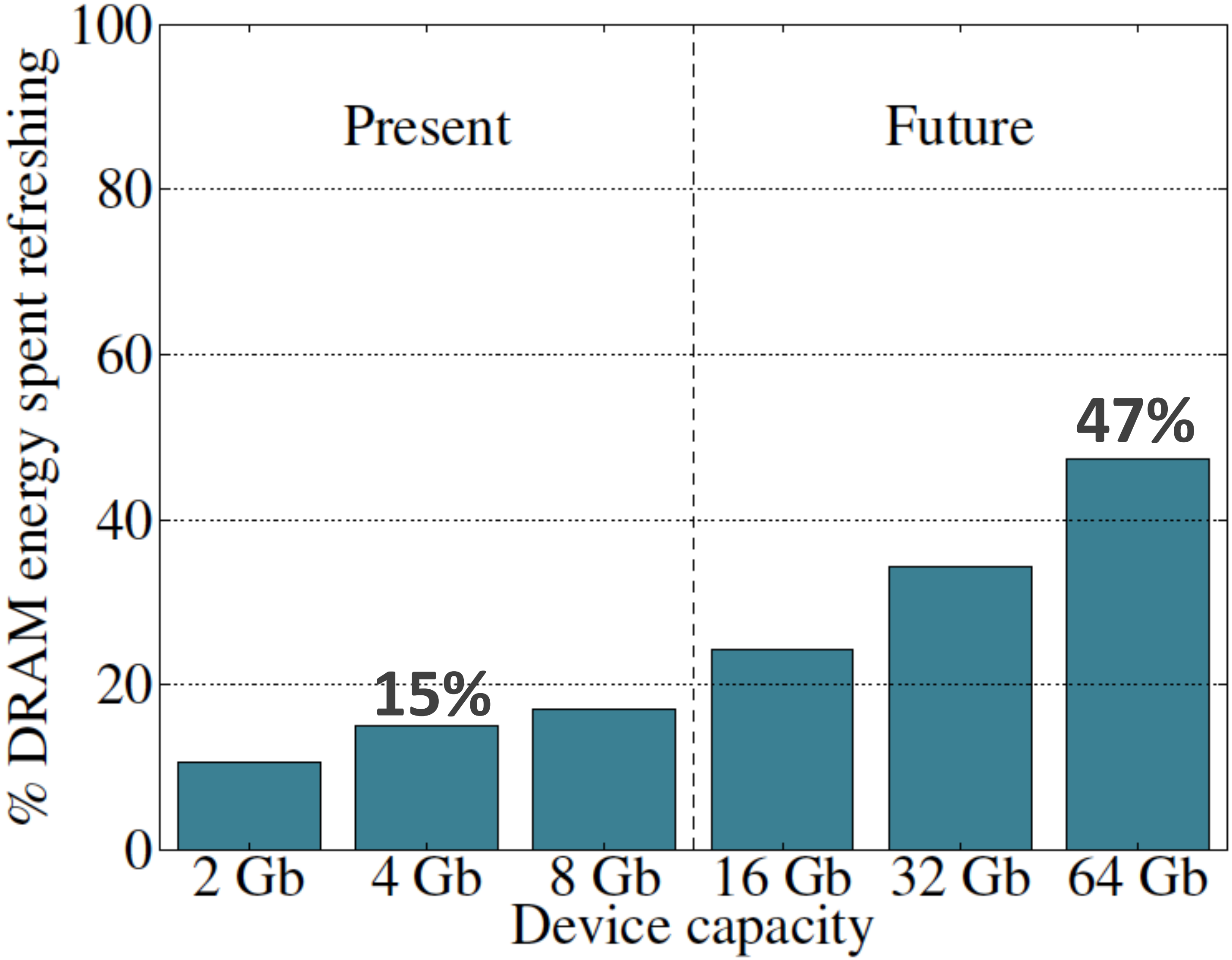
  -- Refresh rate limits DRAM capacity scaling

Refresh Overhead: Performance



Liu et al., "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.

Refresh Overhead: Energy



Liu et al., "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.

Agenda

- Major Trends Affecting Main Memory
- The DRAM Scaling Problem
- Refresh as a Limiter of DRAM Scaling
- Some Solution Directions and Challenges
- Summary

Solutions to the DRAM Scaling Problem

- Two potential solutions
  - Rethink DRAM and refresh (by taking a fresh look at it)
  - Enable emerging non-volatile memory technologies to eliminate/minimize DRAM

- Do both
  - Hybrid memory systems

Solution 1: Rethink DRAM and Refresh

- Overcome DRAM shortcomings with
  - System-DRAM co-design
  - Novel DRAM architectures, interfaces, functions
  - Better waste management (efficient utilization)

- Key issues to tackle
  - Reduce energy
  - Enable reliability at low cost
  - Improve bandwidth, latency, QoS
  - Reduce waste
  - Enable computation close to data

Solution 1: Rethink DRAM and Refresh

- Liu+, "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.
- Kim+, "A Case for Exploiting Subarray-Level Parallelism in DRAM," ISCA 2012.
- Lee+, "Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture," HPCA 2013.
- Liu+, "An Experimental Study of Data Retention Behavior in Modern DRAM Devices," ISCA 2013.
- Seshadri+, "RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.
- Pekhimenko+, "Linearly Compressed Pages: A Main Memory Compression Framework," MICRO 2013.
- Chang+, "Improving DRAM Performance by Parallelizing Refreshes with Accesses," HPCA 2014.
- Khan+, "The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study," SIGMETRICS 2014.
- Luo+, "Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost," DSN 2014.
- Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

Avoid DRAM:

- Seshadri+, "The Evicted-Address Filter: A Unified Mechanism to Address Both Cache Pollution and Thrashing," PACT 2012.
- Pekhimenko+, "Base-Delta-Immediate Compression: Practical Data Compression for On-Chip Caches," PACT 2012.
- Seshadri+, "The Dirty-Block Index," ISCA 2014.

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

Chang+, "Improving DRAM Performance by Parallelizing Refreshes with Accesses," HPCA 2014.

Tackling Refresh: Solutions

- Parallelize refreshes with accesses [Chang+ HPCA'14]

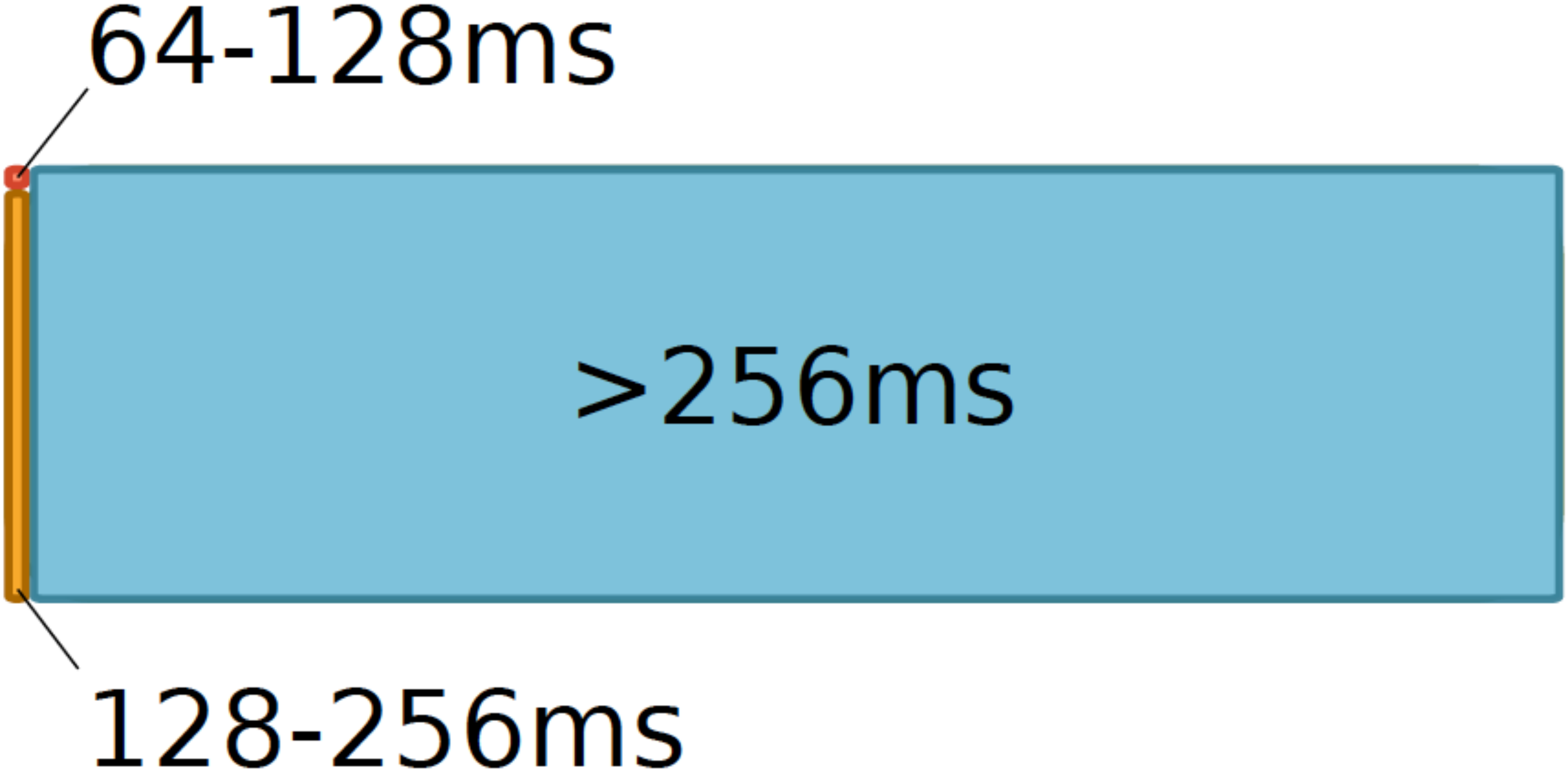
- Eliminate unnecessary refreshes [Liu+ ISCA'12]
  - Exploit device characteristics
  - Exploit data and application characteristics

- 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:



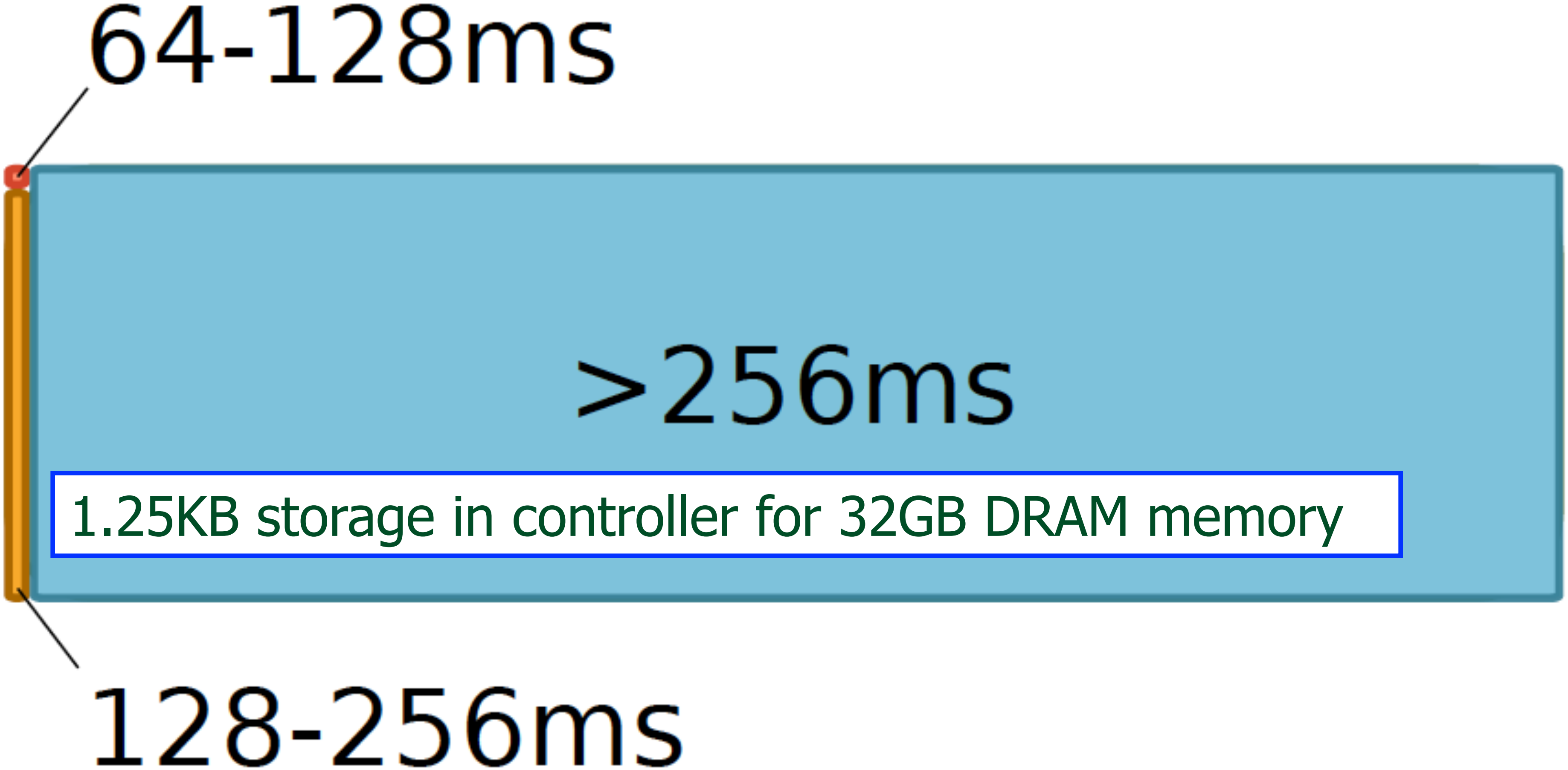
Works on Reducing Refreshes

- Observed significant variation in data retention times of DRAM cells (due to manufacturing process variation)
  - Retention time: maximum time a cell can go without being refreshed while maintaining its stored data

- Proposed methods to take advantage of widely varying retention times among DRAM rows
  - Reduce refresh rate for rows that can retain data for longer than 64 ms, e.g., **[Liu+ ISCA 2012]**
  - Disable rows that have low retention times, e.g., **[Venkatesan+ HPCA 2006]**

- Showed large benefits in energy and performance

An Example: RAIDR [Liu+, ISCA 2012]



64-128ms

>256ms

1.25KB storage in controller for 32GB DRAM memory

128-256ms

Can reduce refreshes by ~75%
→ reduces energy consumption and improves performance

Liu et al., "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.

RAIDR Results

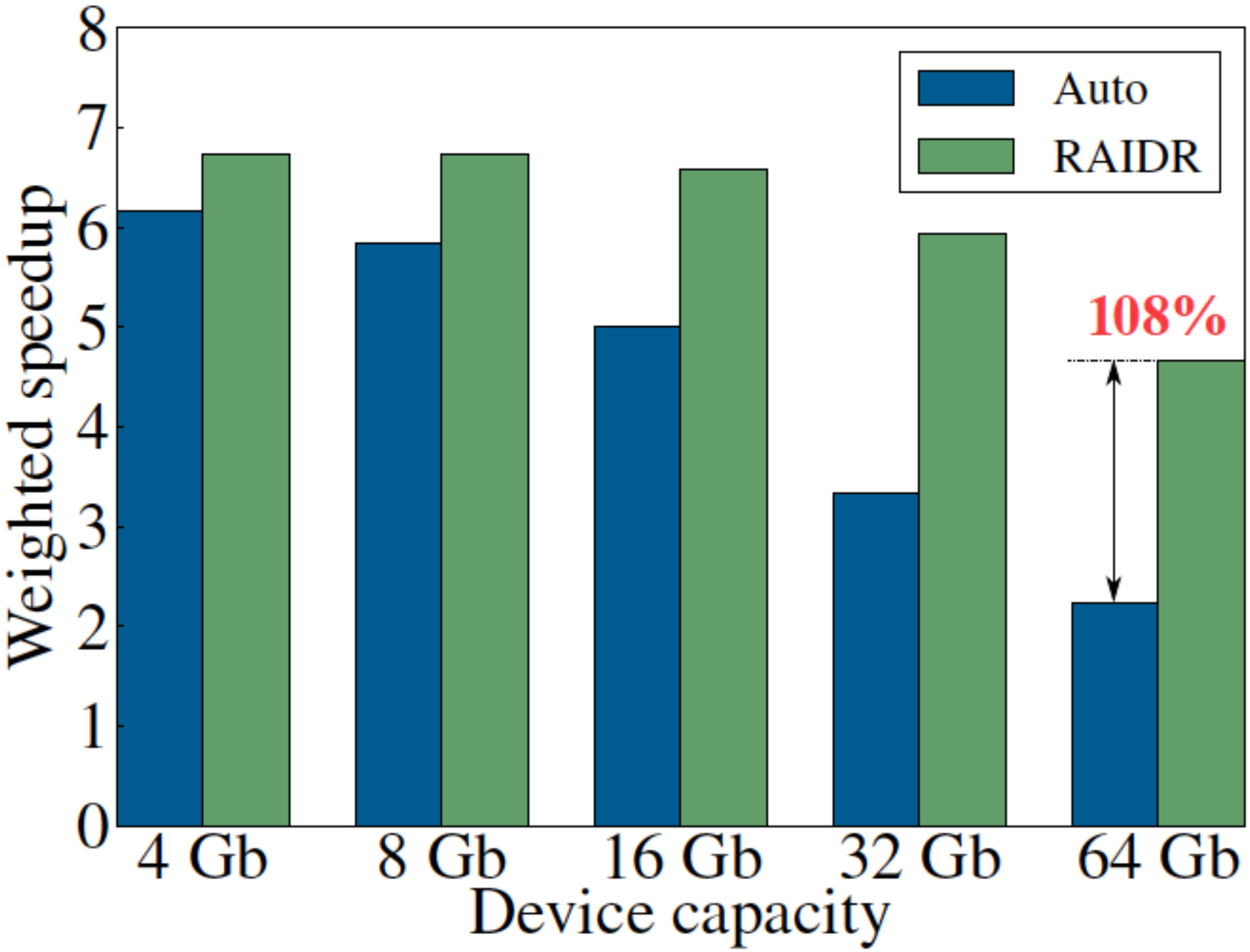
- Baseline:
  - 32 GB DDR3 DRAM system (8 cores, 512KB cache/core)
  - 64ms refresh interval for all rows

- RAIDR:
  - 64–128ms retention range: 256 B Bloom filter, 10 hash functions
  - 128–256ms retention range: 1 KB Bloom filter, 6 hash functions
  - Default refresh interval: 256 ms

- Results on SPEC CPU2006, TPC-C, TPC-H benchmarks
  - 74.6% refresh reduction
  - ~16%/20% DRAM dynamic/idle power reduction
  - ~9% performance improvement

Liu et al., "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.

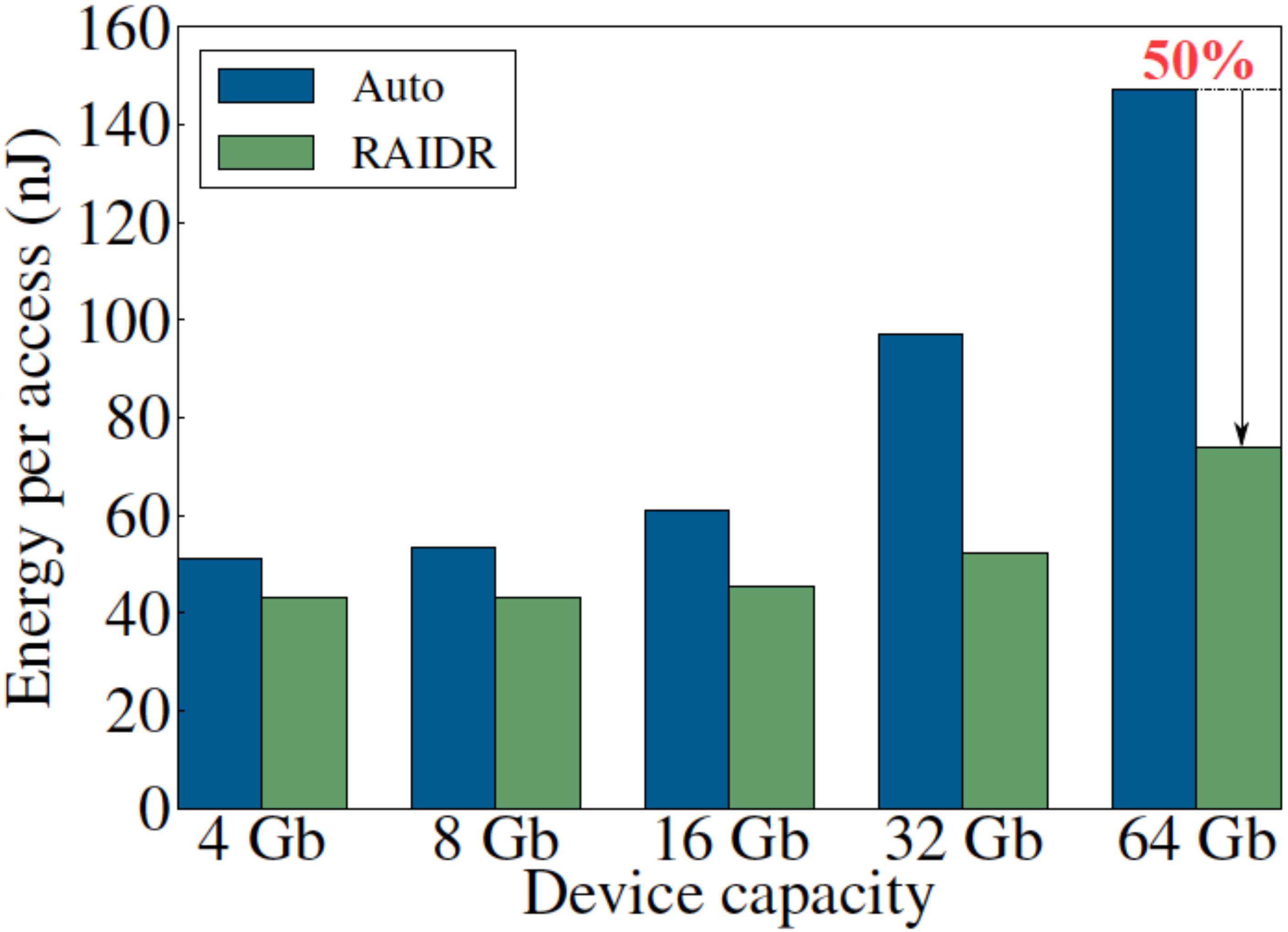
DRAM Device Capacity Scaling: Performance



RAIDR performance benefits increase with DRAM chip capacity

Liu et al., "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.

DRAM Device Capacity Scaling: Energy



RAIDR energy benefits increase with DRAM chip capacity

RAIDR slides

SAFARI Liu et al., "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.

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?

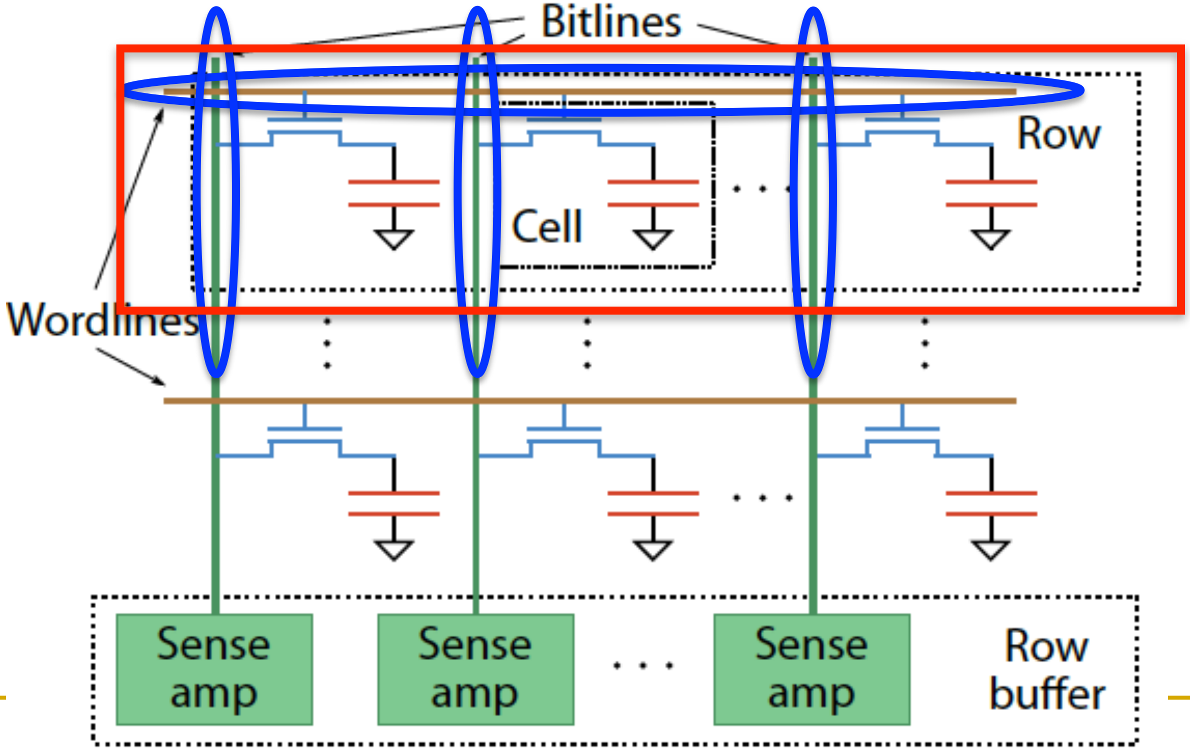
Two Challenges to Retention Time Profiling

- Data Pattern Dependence (DPD) of retention time

- Variable Retention Time (VRT) phenomenon

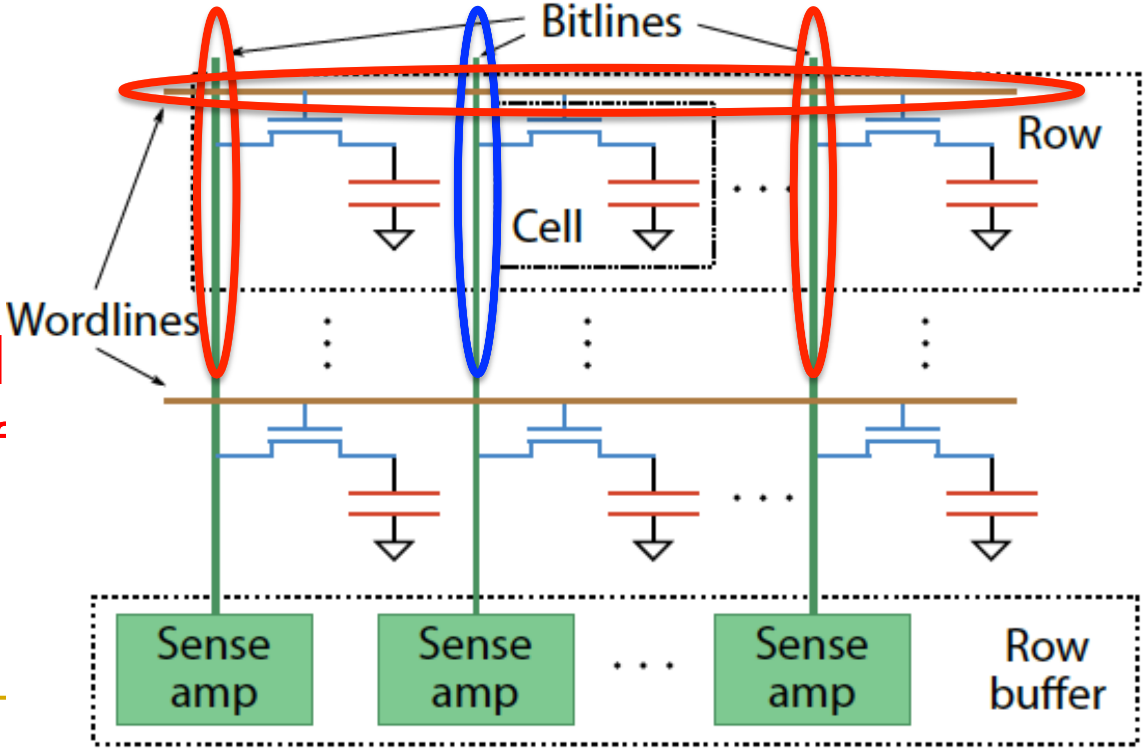
Two Challenges to Retention Time Profiling

- 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

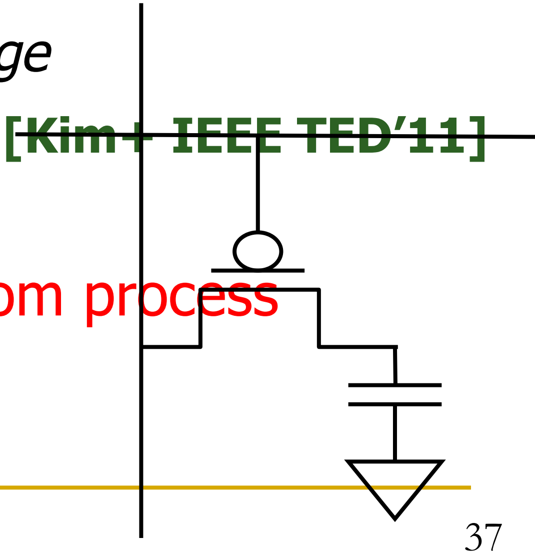
- Electrical noise on the bitline affects reliable sensing of a DRAM cell
- The magnitude of this noise is affected by values of nearby cells via
  - Bitline-bitline coupling → electrical coupling between adjacent bitlines
  - Bitline-wordline coupling → electrical coupling between each bitline and the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



- Retention tored in
  nearby cel
  → need to f etention time

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



Our Goal [Liu+, ISCA 2013]

- Analyze the retention time behavior of DRAM cells in modern commodity DRAM devices
  - to aid the collection of accurate profile information

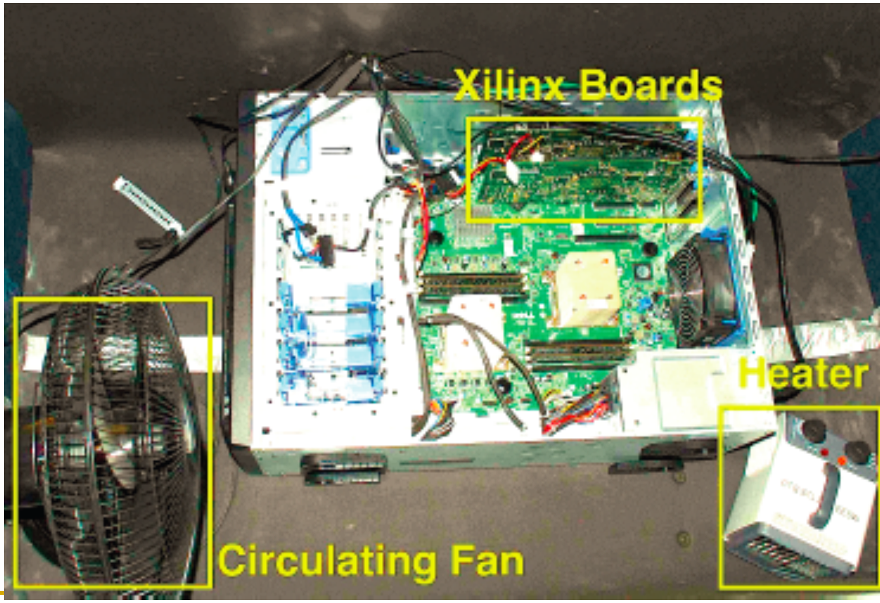
- Provide a comprehensive empirical investigation of two key challenges to retention time profiling
  - Data Pattern Dependence (DPD)
  - Variable Retention Time (VRT)

Liu+, "An Experimental Study of Data Retention Behavior in Modern DRAM Devices," ISCA 2013.

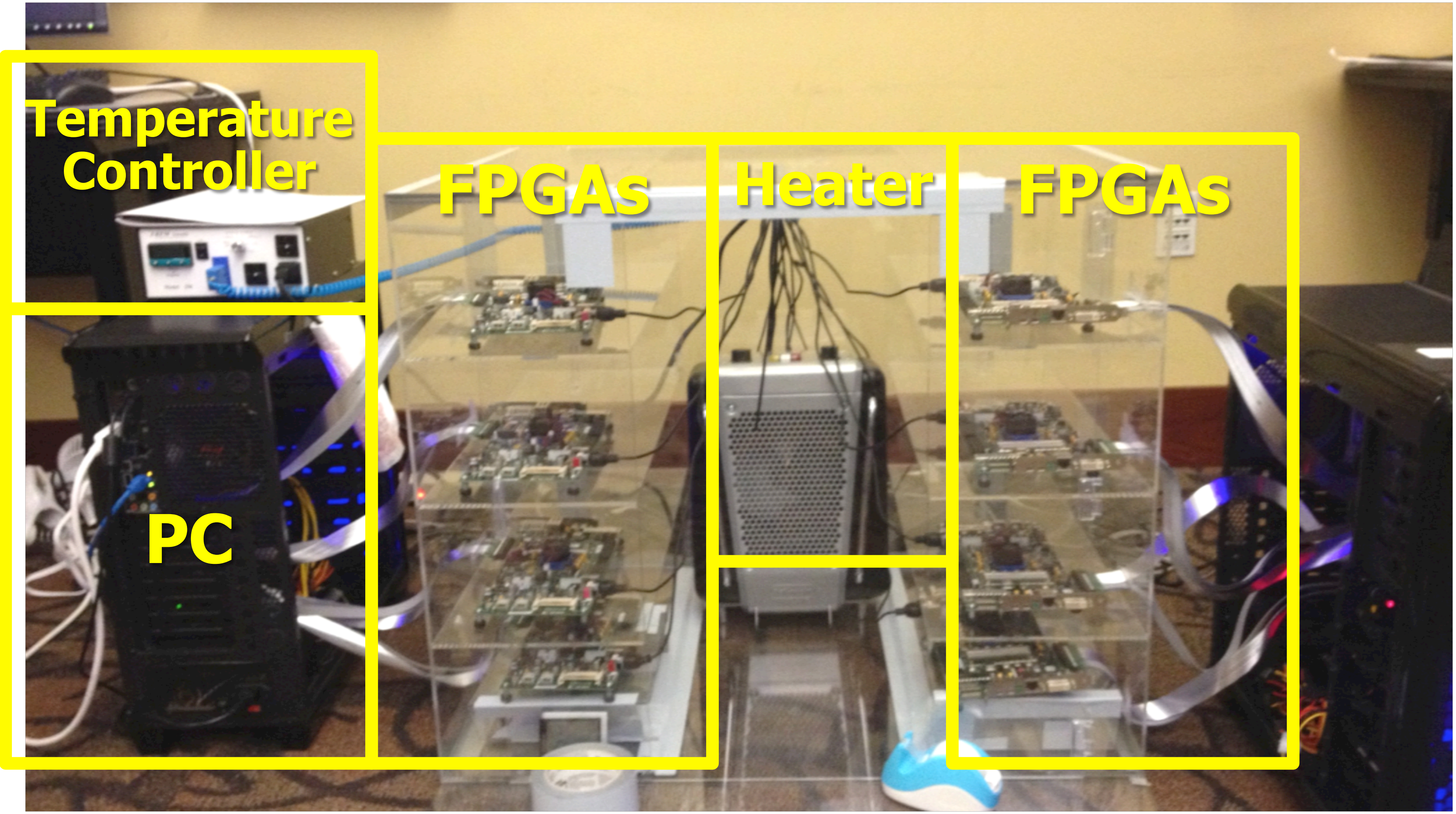
Experimental Infrastructure (DRAM)

Liu+, “An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms”, ISCA 2013.

Khan+, “The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study,” SIGMETRICS 2014.



Experimental Infrastructure (DRAM)



Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

DRAM Testing Platform and Method

- Test platform: Developed a DDR3 DRAM testing platform using the Xilinx ML605 FPGA development board
  - Temperature controlled

- Tested DRAM chips: 248 commodity DRAM chips from five manufacturers (A,B,C,D,E)
- Seven families based on equal capacity per device:
  - A 1Gb, A 2Gb
  - B 2Gb
  - C 2Gb
  - D 1Gb, D 2Gb
  - E 2Gb

Experiment Design

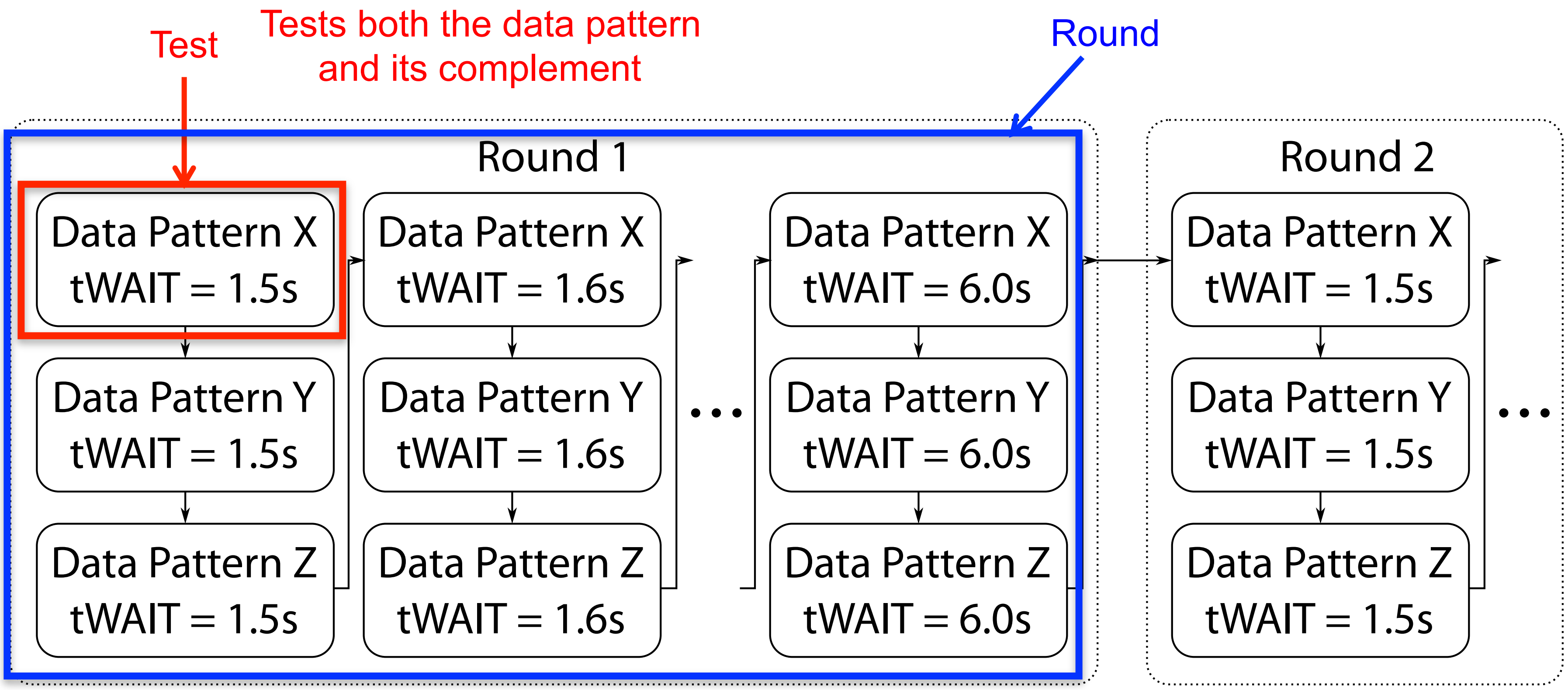
- Each module tested for multiple ***rounds*** of ***tests***.

- Each test searches for the set of cells with a retention time less than a threshold value for a particular data pattern

- High-level structure of a test:
    - Write data pattern to rows in a DRAM bank
    - Prevent refresh for a period of time *tWAIT*, leave DRAM idle
    - Read stored data pattern, compare to written pattern and record corrupt cells as those with retention time < *tWAIT*

- Test details and important issues to pay attention to are discussed in paper

Experiment Structure



Experiment Parameters

- Most tests conducted at 45^{o}C

- No cells observed to have a retention time less than 1.5 second at 45^{o}C

- Tested *tWAIT* in increments of 128ms from 1.5 to 6.1 seconds

Tested Data Patterns

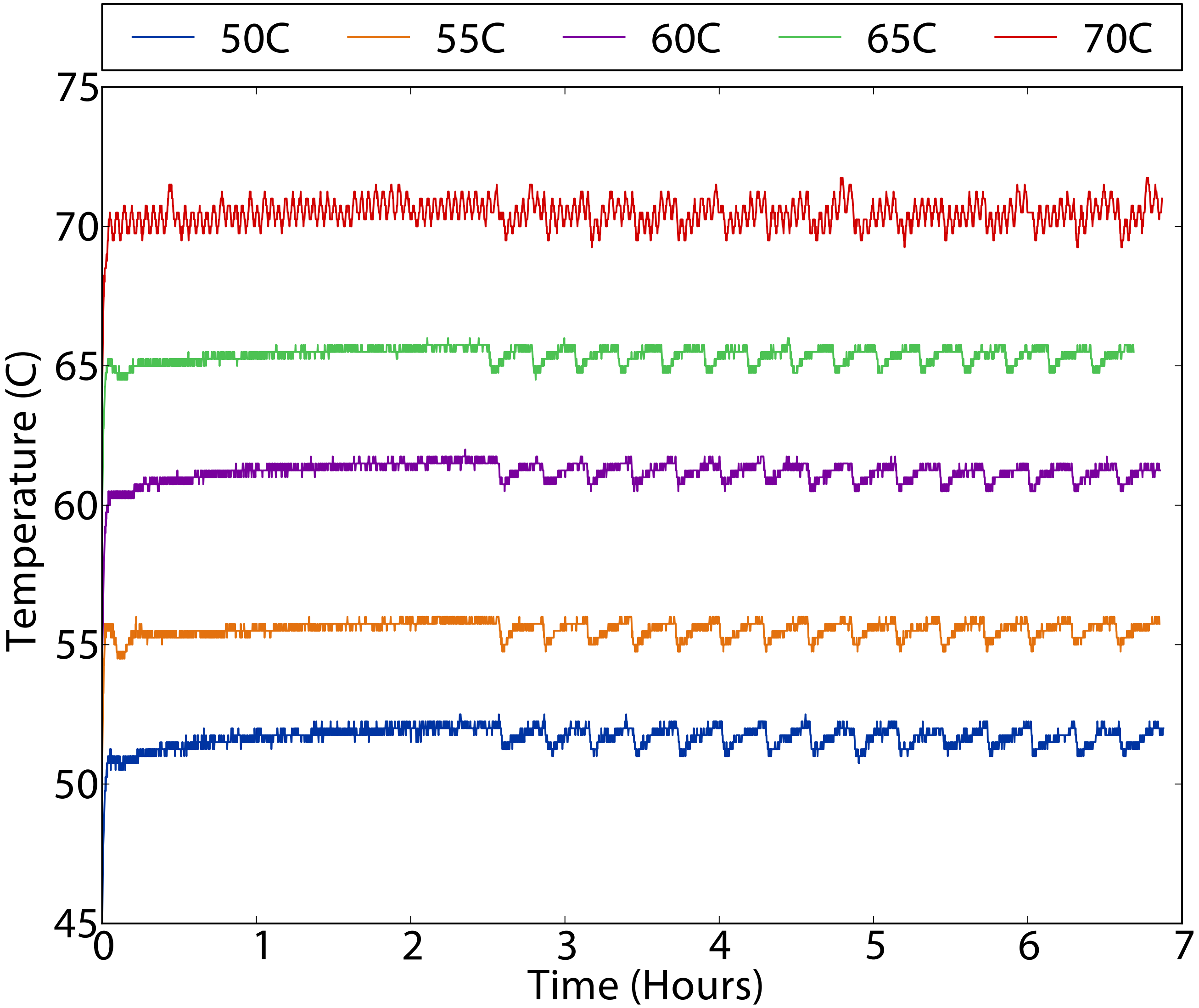
Fixed patterns

- All 0s/1s: Value 0/1 is written to all bits
  - Previous work suggested this is sufficient
- Checkerboard: Consecutive bits alternate between 0 and 1
  - Coupling noise increases with voltage difference between the neighboring bitlines → May induce worst case data pattern (if adjacent bits mapped to adjacent cells)
- Walk: Attempts to ensure a single cell storing 1 is surrounded by cells storing 0
  - This may lead to even worse coupling noise and retention time due to coupling between *nearby* bitlines **[Li+ IEEE TCSI 2011]**
  - Walk pattern is permuted in each round to exercise different cells
- Random: Randomly generated data is written to each row
  - A new set of random data is generated for each round

DRAM Retention Time: Results

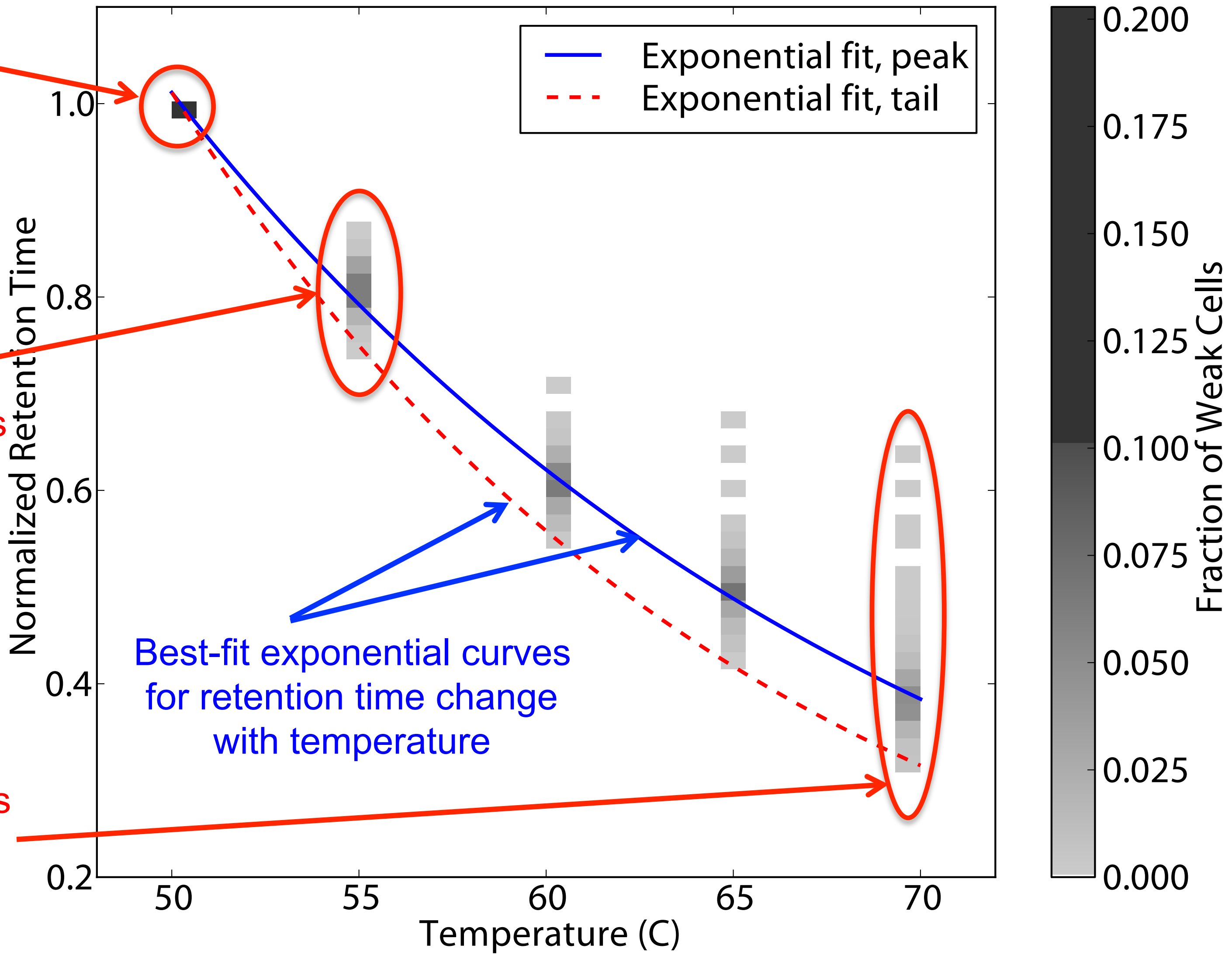
- Foundational Results
    - Temperature Dependence
    - Retention Time Distribution
- Data Pattern Dependence: Analysis and Implications
- Variable Retention Time: Analysis and Implications
- Conclusions

Temperature Stability



Tested chips at five different stable temperatures

Dependence of Retention Time on Temperature



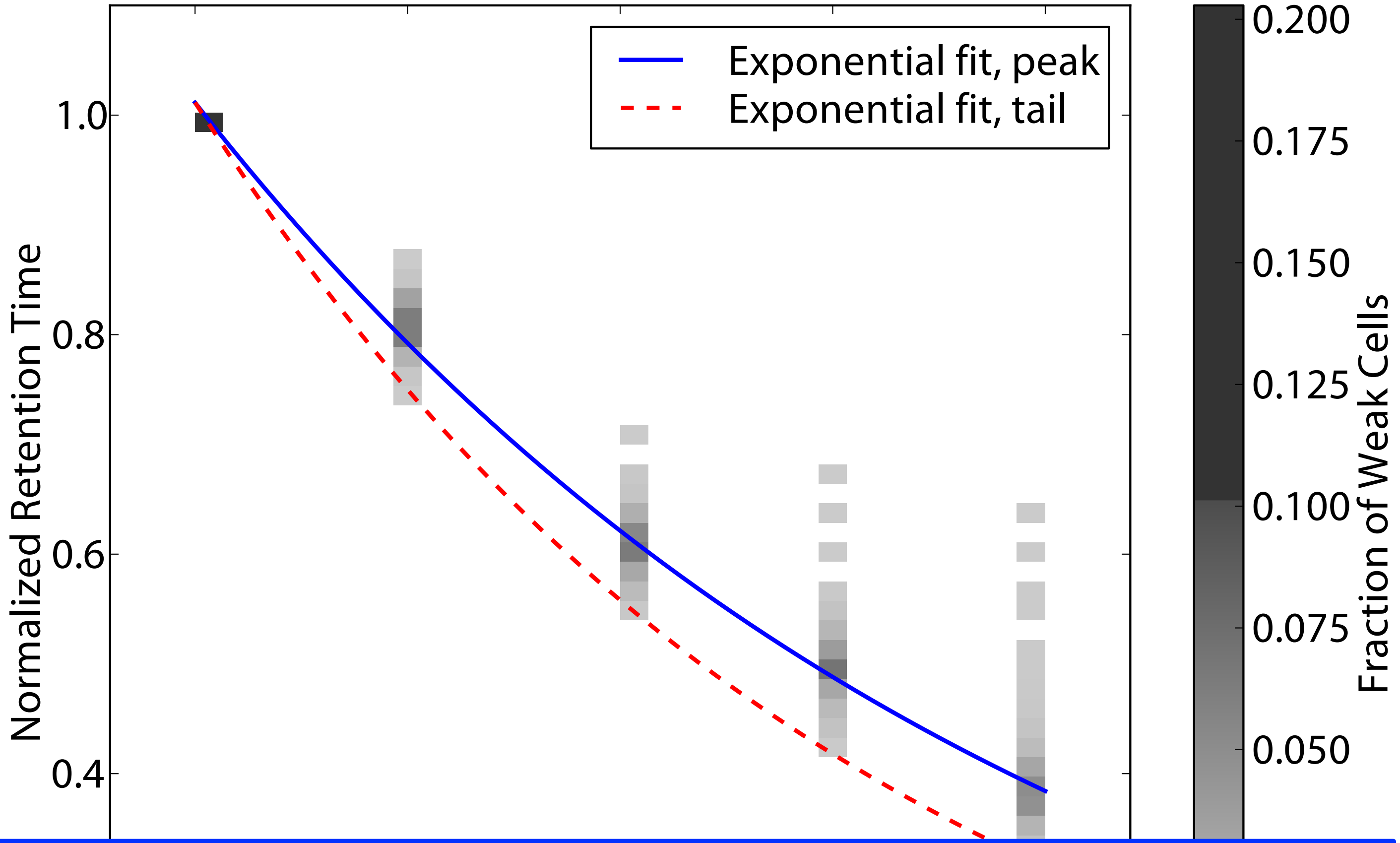
Fraction of cells that exhibited retention time failure at any *tWAIT* for any data pattern at 50°C

Normalized retention times of the same cells at 55°C

Normalized retention times of the same cells At 70°C

Best-fit exponential curves for retention time change with temperature

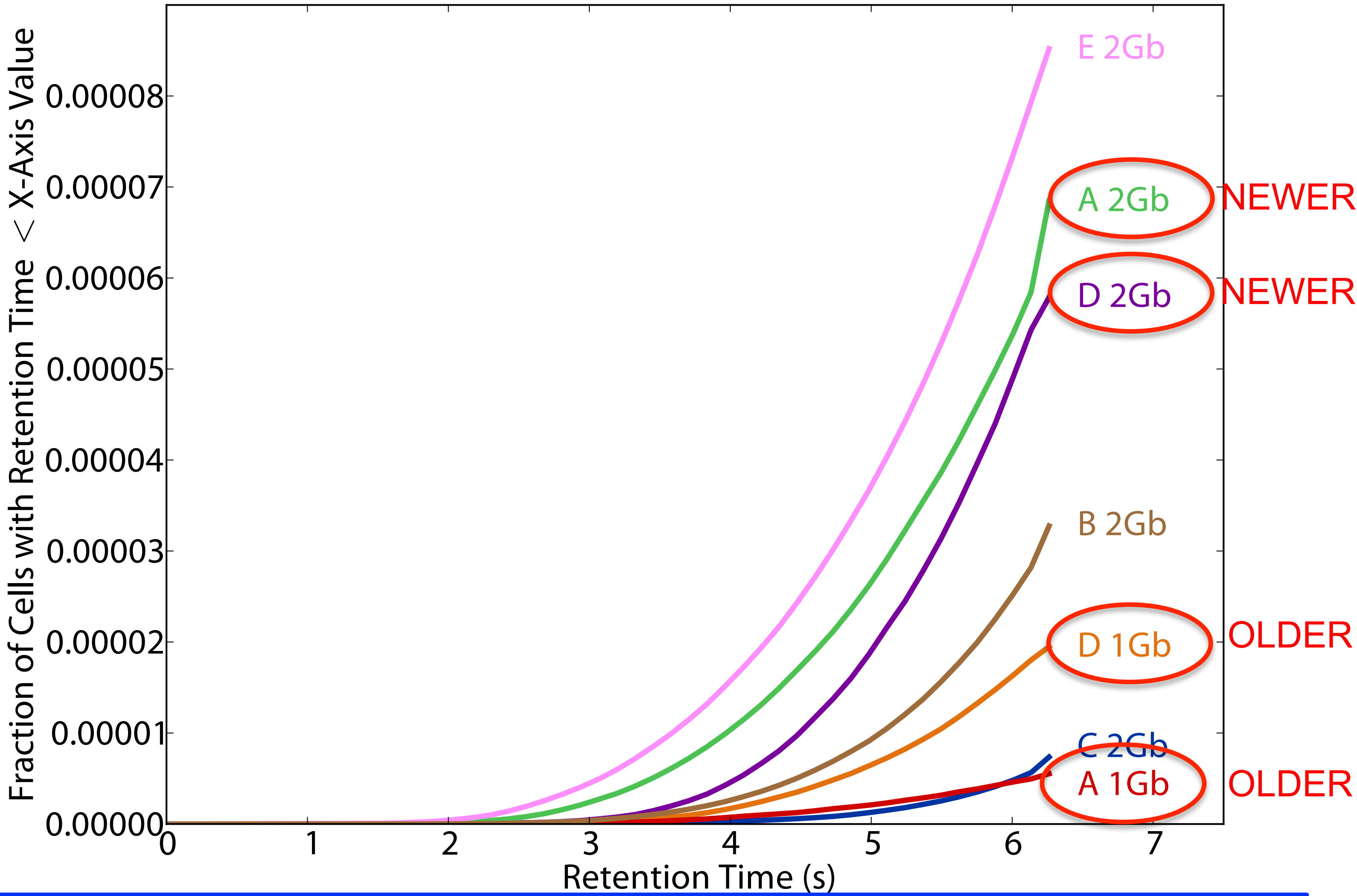
Dependence of Retention Time on Temperature



Relationship between retention time and temperature is consistently bounded (predictable) within a device

Every 10^{O}C temperature increase
→ 46.5% reduction in retention time in the worst case

Retention Time Distribution



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

DRAM Retention Time: Results

- Foundational Results
    - Temperature Dependence
    - Retention Time Distribution
- Data Pattern Dependence: Analysis and Implications
- Variable Retention Time: Analysis and Implications
- Conclusions

Some Terminology

- Failure population of cells with Retention Time X: The set of all cells that exhibit retention failure in any test with *any* data pattern at that retention time (*tWAIT*)

- Retention Failure Coverage of a Data Pattern DP: Fraction of cells with retention time X that exhibit retention failure with that *particular* data pattern DP

- If retention times are not dependent on data pattern stored in cells, we would expect
  - Coverage of any data pattern to be 100%
  - In other words, if one data pattern causes a retention failure, any other data pattern also would

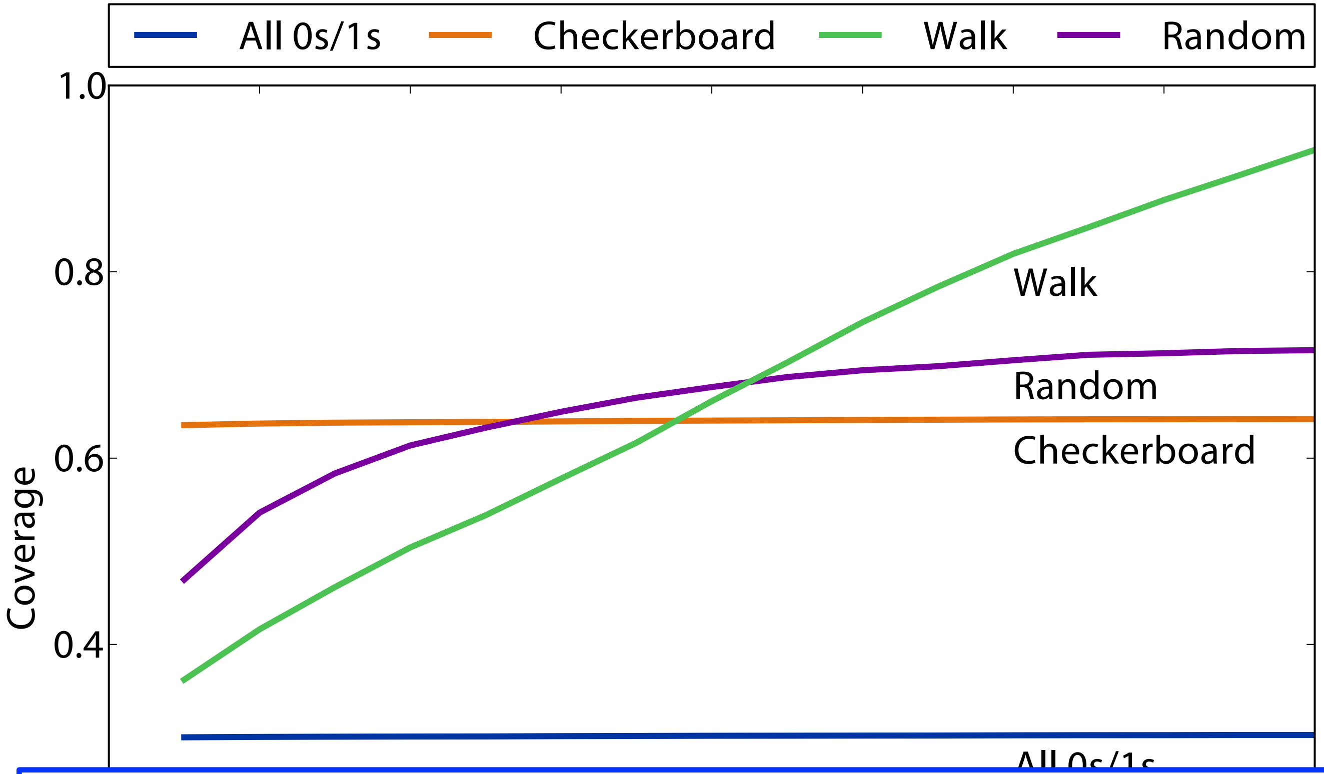
Recall the Tested Data Patterns

- All 0s/1s: Value 0/1 is written to all bits

Fixed patterns

- Checkerboard: Consecutive bits alternate between 0 and 1
- Walk: Attempts to ensure a single cell storing 1 is surrounded by cells storing 0
- Random: Randomly generated data is written to each row

Retention Failure Coverage of Data Patterns



A 2Gb chip family

6.1s retention time

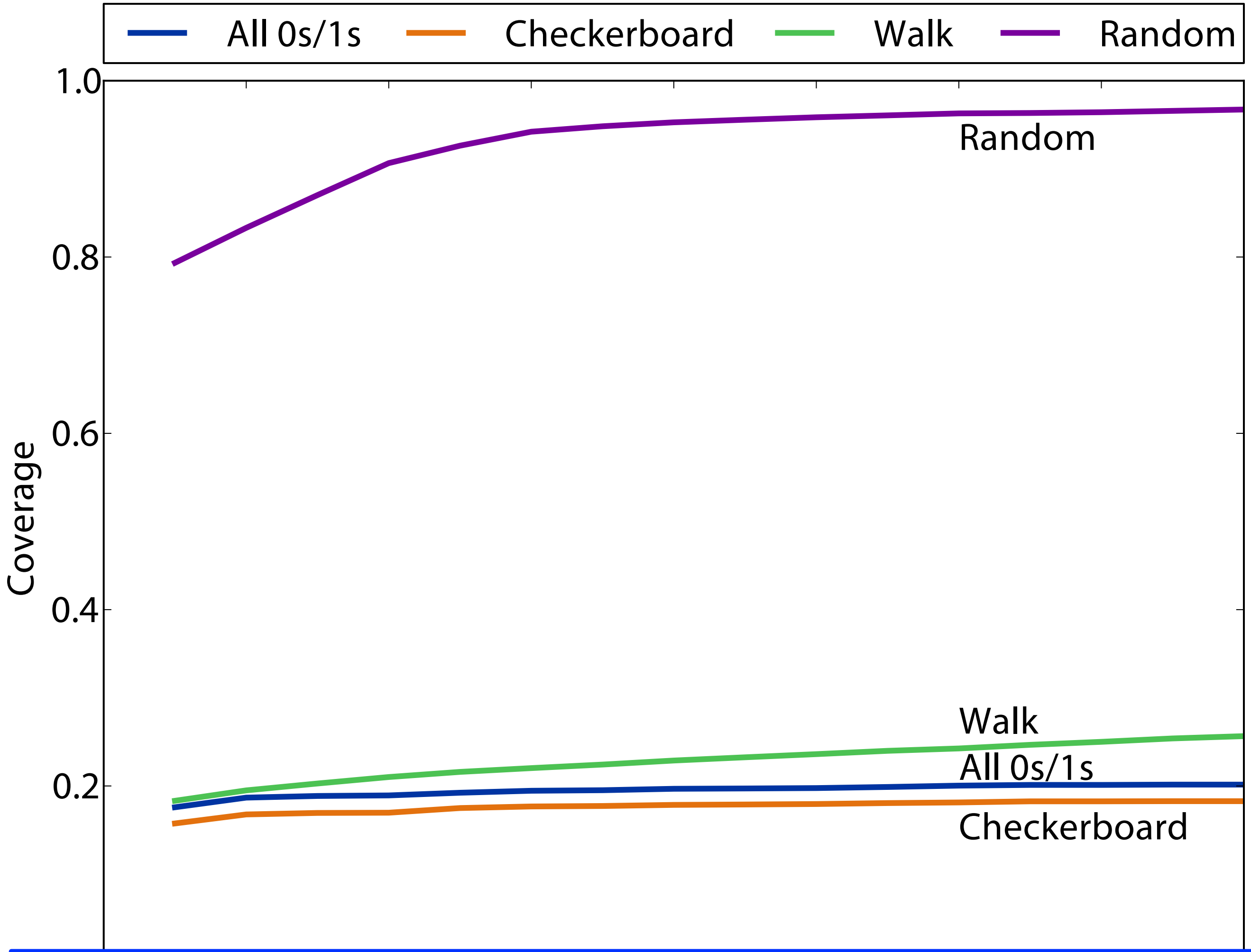
Different data patterns have widely different coverage:
Data pattern dependence exists and is severe

Coverage of fixed patterns is low: ~30% for *All 0s/1s*

***Walk* is the most effective data pattern for this device**

No data pattern achieves 100% coverage

Retention Failure Coverage of Data Patterns



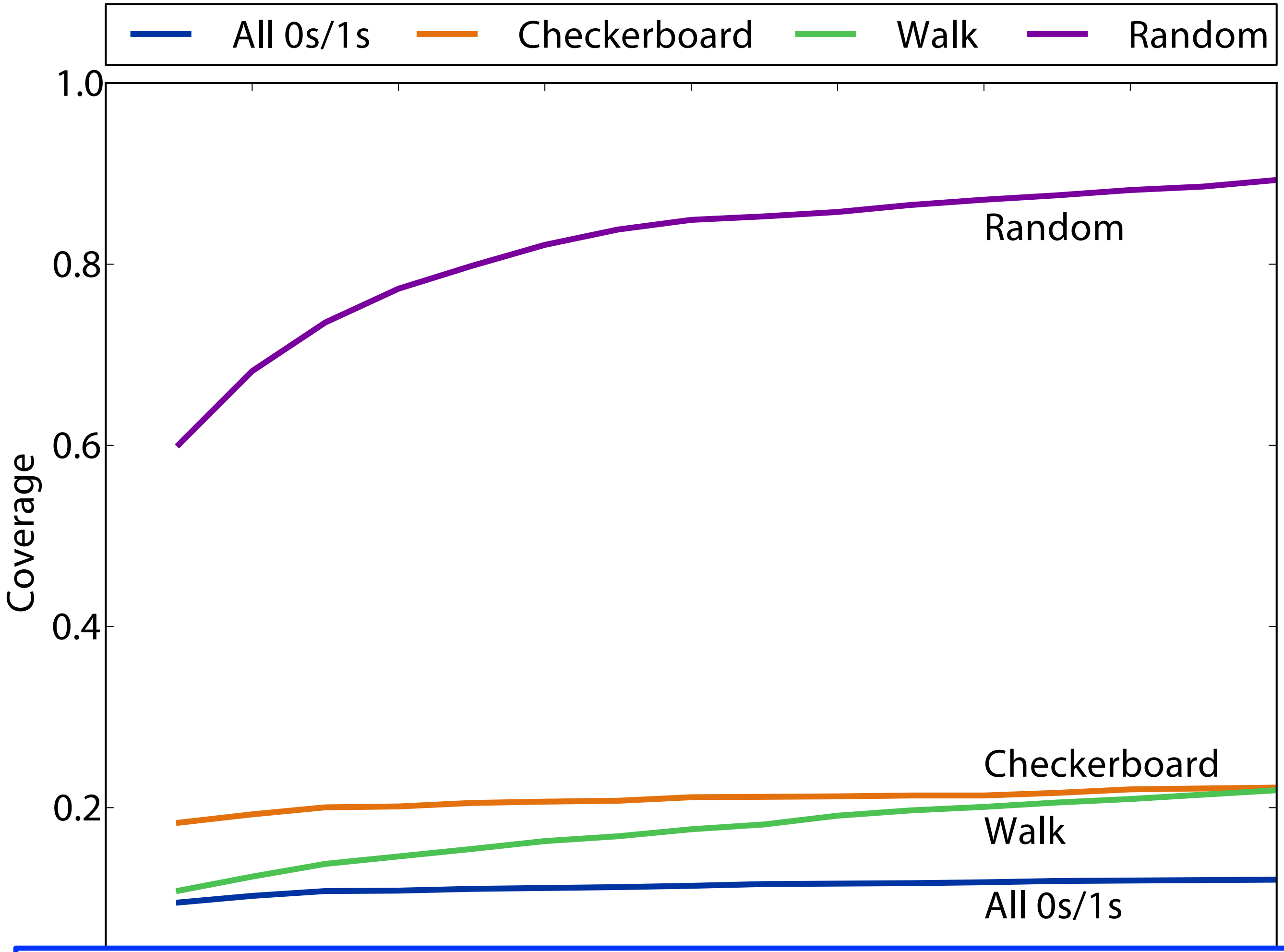
B 2Gb chip family

6.1s retention time

***Random* is the most effective data pattern for this device**

No data pattern achieves 100% coverage

Retention Failure Coverage of Data Patterns



C 2Gb chip family

6.1s retention time

Random **is the most effective data pattern for this device**

No data pattern achieves 100% coverage

Data Pattern Dependence: Observations (I)

- A cell’s retention time is heavily influenced by data pattern stored in other cells
  - Pattern affects the coupling noise, which affects cell leakage

- No tested data pattern exercises the worst case retention time for all cells (no pattern has 100% coverage)
  - No pattern is able to induce the worst-case coupling noise for every cell
  - Problem: Underlying DRAM circuit organization is *not* known to the memory controller → very hard to construct a pattern that exercises the worst-case cell leakage

    → Opaque mapping of addresses to physical DRAM geometry

    → Internal remapping of addresses within DRAM to tolerate faults

    → Second order coupling effects are very hard to determine

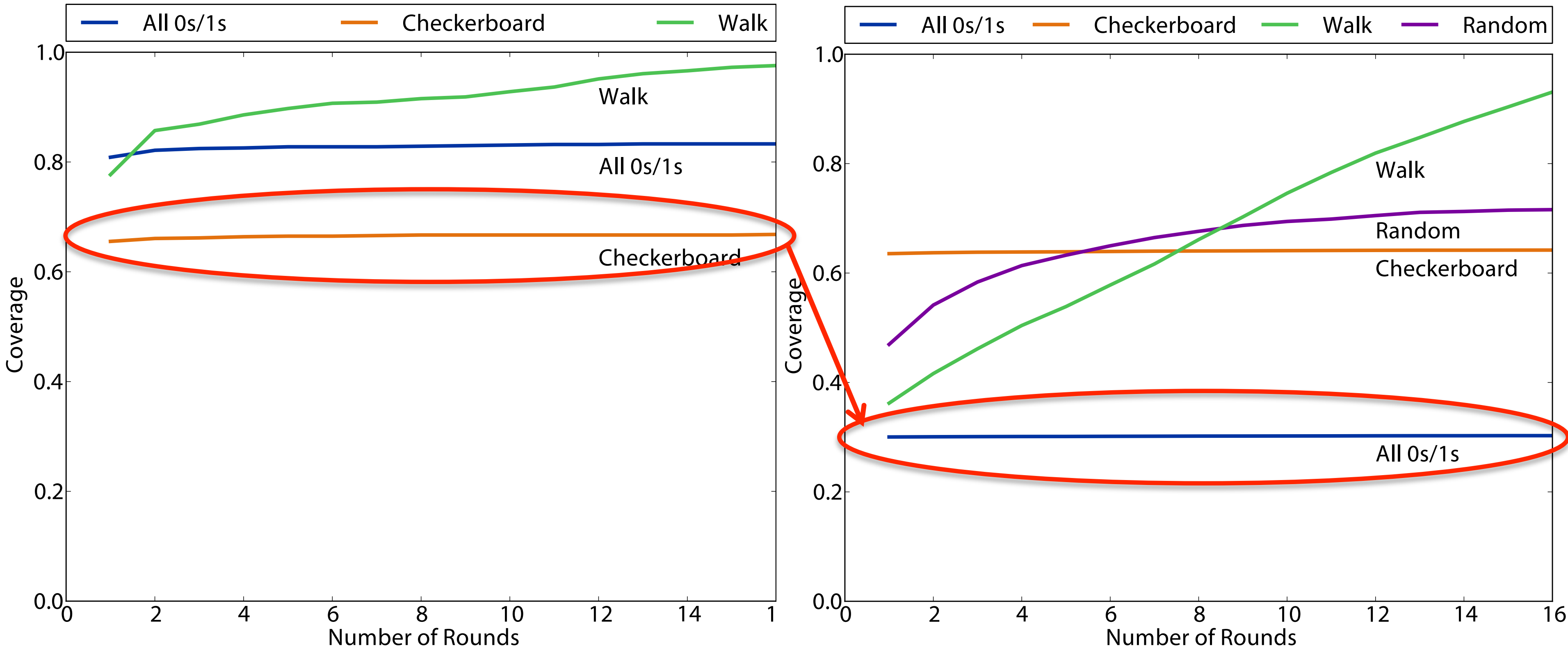
Data Pattern Dependence: Observations (II)

- Fixed, simple data patterns have low coverage
  - They do not exercise the worst-case coupling noise

- The effectiveness of each data pattern varies significantly between DRAM devices (of the same or different vendors)
  - Underlying DRAM circuit organization likely differs between different devices → patterns leading to worst coupling are different in different devices

- Technology scaling appears to increase the impact of data pattern dependence
  - Scaling reduces the physical distance between circuit elements, increasing the magnitude of coupling effects

Effect of Technology Scaling on DPD



A 1Gb chip family

A 2Gb chip family

The lowest-coverage data pattern achieves much lower coverage for the smaller technology node

DPD: Implications on Profiling Mechanisms

- Any retention time profiling mechanism must handle data pattern dependence of retention time
- Intuitive approach: Identify the data pattern that induces the worst-case retention time for a particular cell or device

- Problem 1: Very hard to know at the memory controller which bits actually interfere with each other due to
  - Opaque mapping of addresses to physical DRAM geometry → logically consecutive bits may not be physically consecutive
  - Remapping of faulty bitlines/wordlines to redundant ones internally within DRAM

- Problem 2: Worst-case coupling noise is affected by non-obvious second order bitline coupling effects

DRAM Retention Time: Results

- Foundational Results
  - Temperature Dependence
  - Retention Time Distribution
- Data Pattern Dependence: Analysis and Implications
- Variable Retention Time: Analysis and Implications
- Conclusions

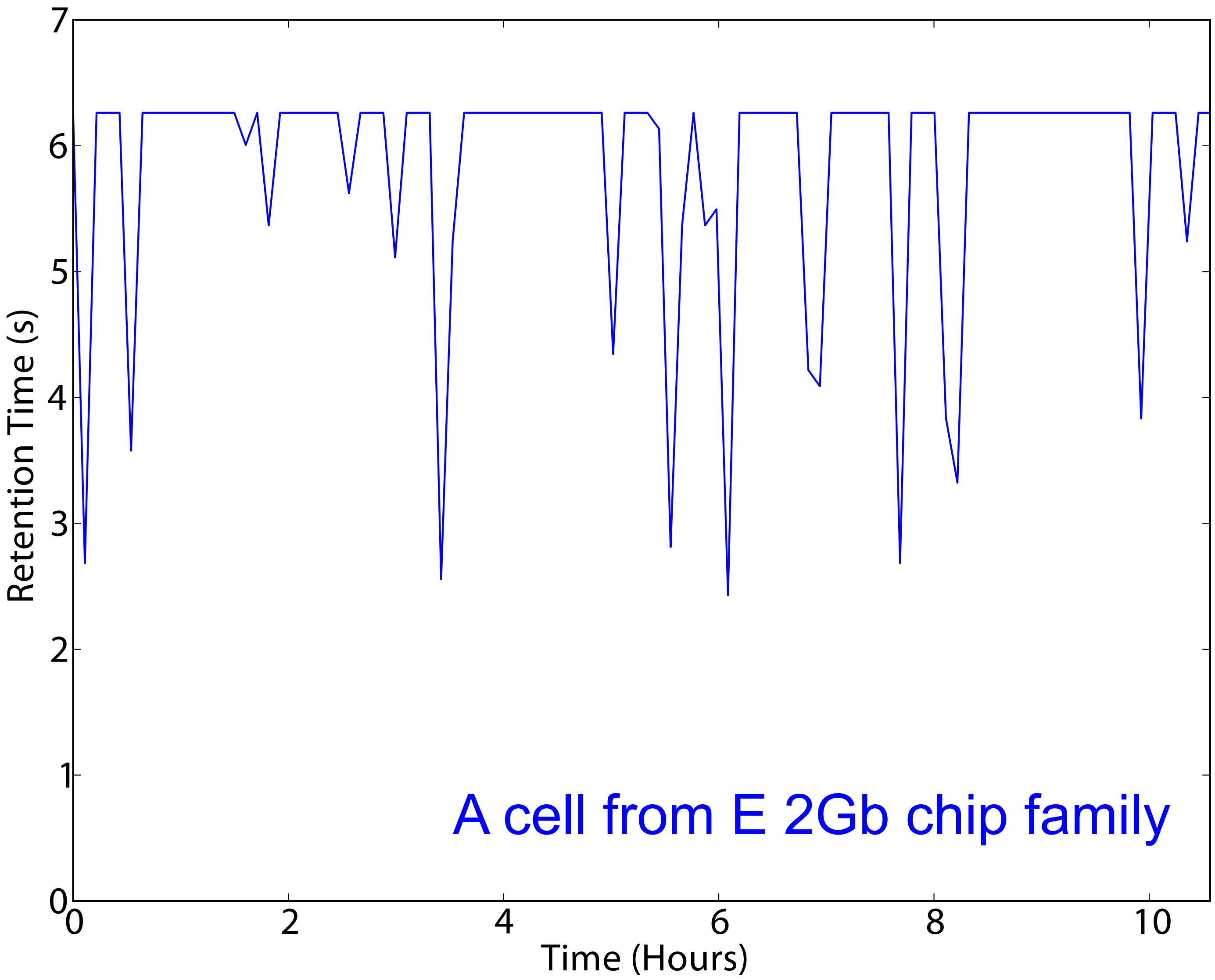
Variable Retention Time

- Retention time of a cell can vary over time

- A cell can randomly switch between multiple leakage current states due to *Trap-Assisted Gate-Induced Drain Leakage,* which appears to be a random process

  [Yaney+ IEDM 1987, Restle+ IEDM 1992]

An Example VRT Cell

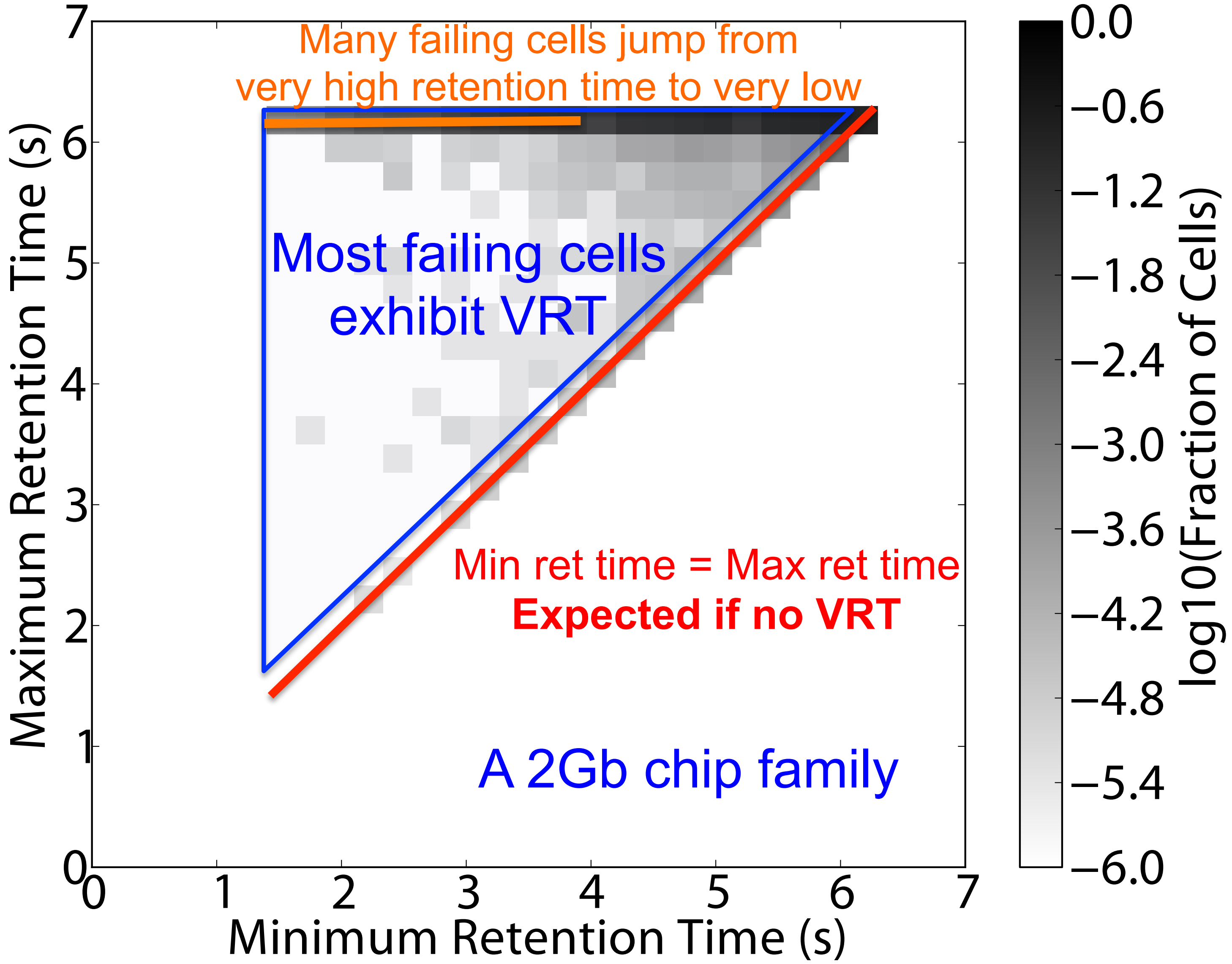


VRT: Questions and Methodology

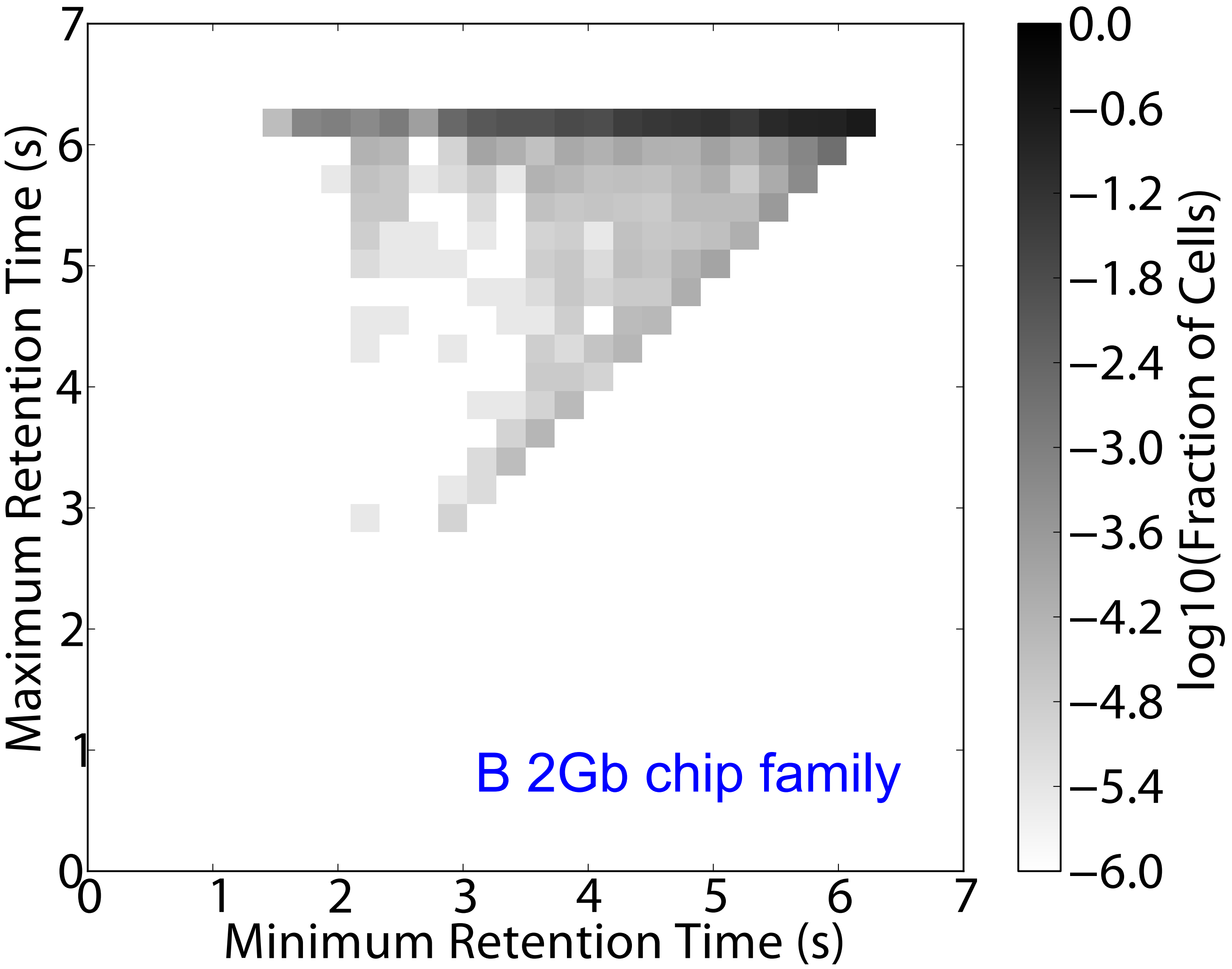
- Key Questions
  - How prevalent is VRT in modern DRAM devices?
  - What is the timescale of observation of the lowest retention time state?
  - What are the implications on retention time profiling?

- Test Methodology
  - Each device was tested for at least 1024 rounds over 24 hours
  - Temperature fixed at 45°C
  - Data pattern used is the most effective data pattern for each device
  - For each cell that fails at any retention time, we record the minimum and the maximum retention time observed

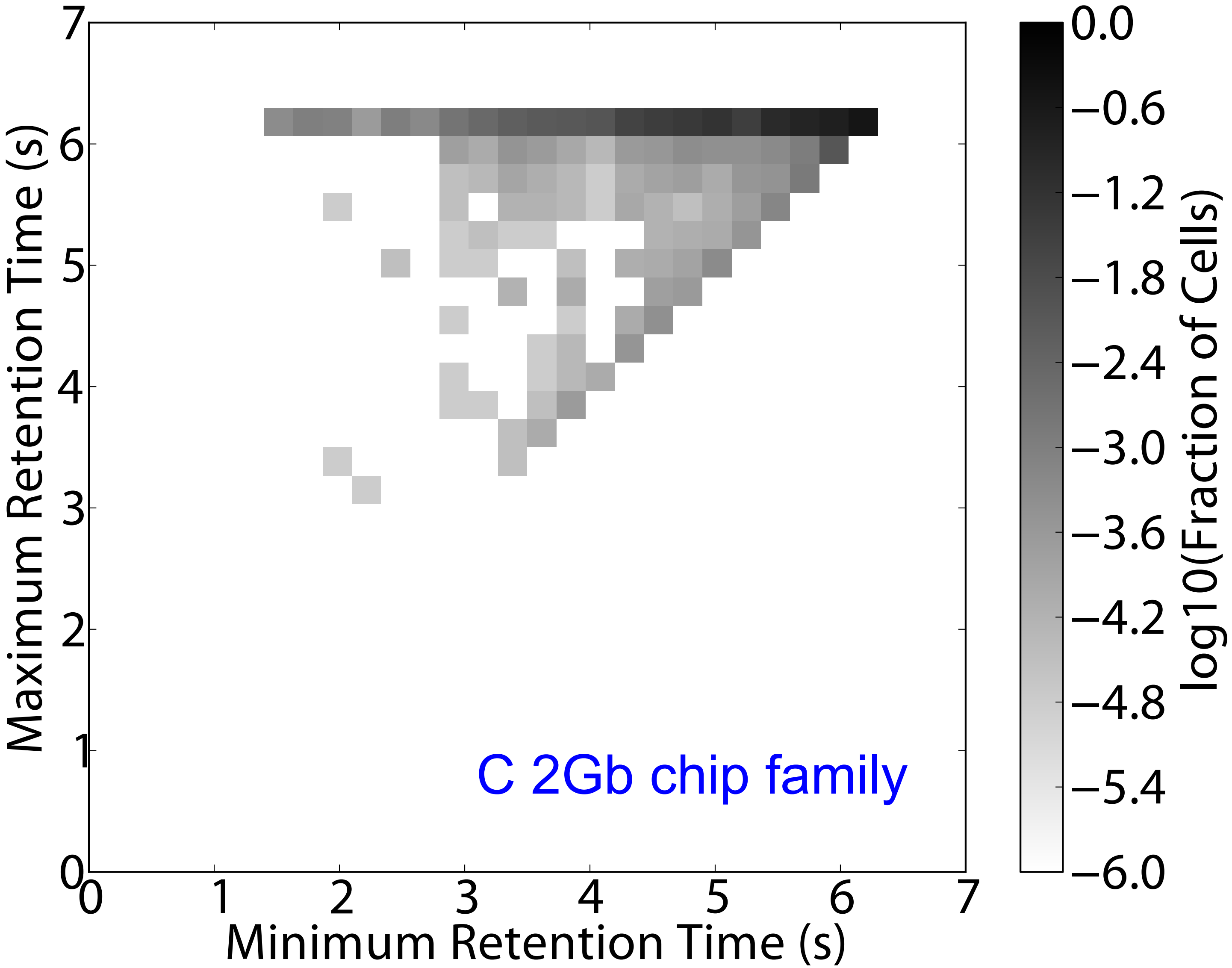
Variable Retention Time



Variable Retention Time



Variable Retention Time



VRT: Observations So Far

- VRT is common among weak cells (i.e., those cells that experience low retention times)

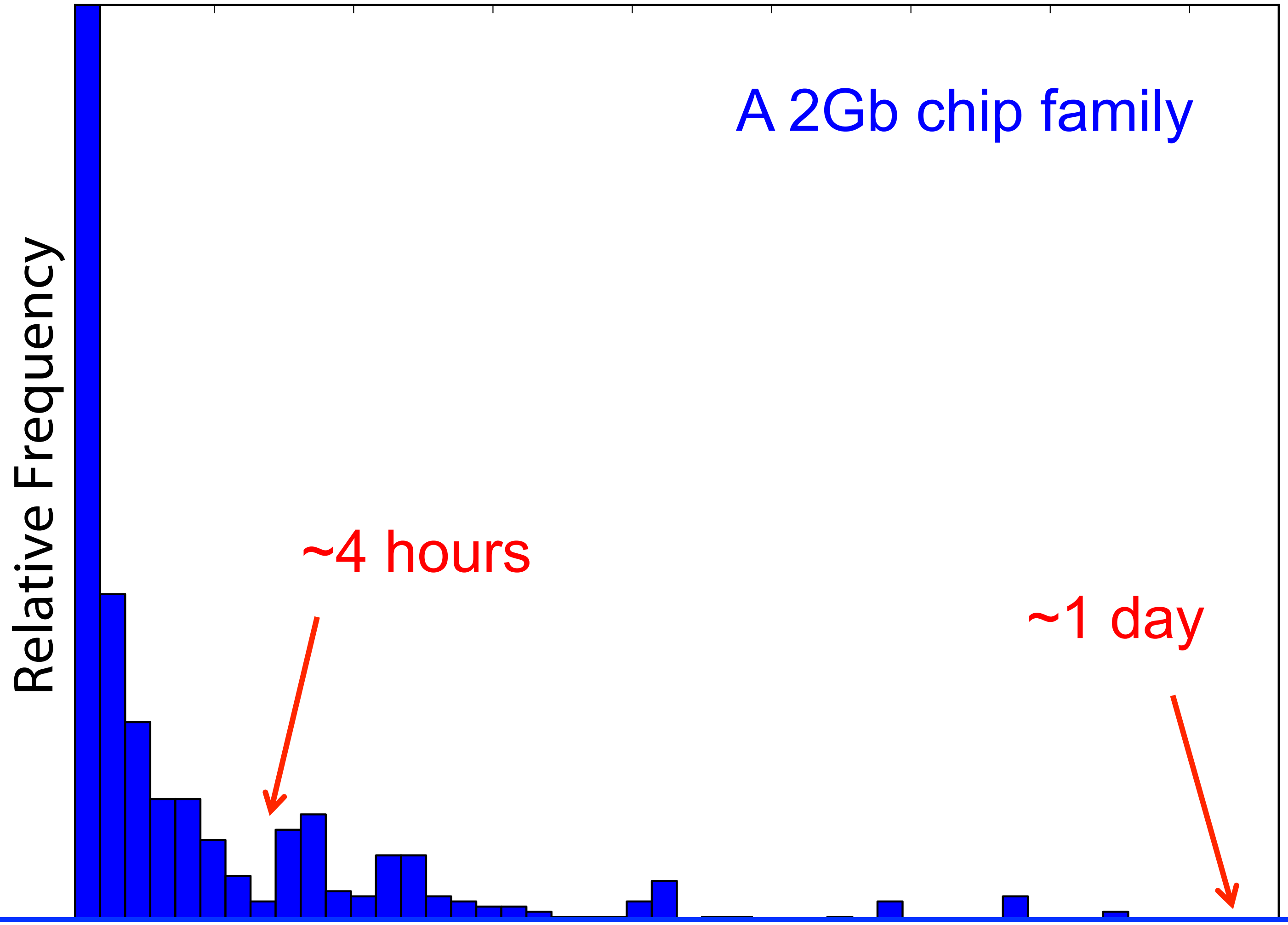
- VRT can result in significant retention time changes
  - Difference between minimum and maximum retention times of a cell can be more than 4x, and may not be bounded
  - Implication: Finding *a* retention time for a cell and using a guardband to ensure minimum retention time is "covered" requires a large guardband or may not work

- Retention time profiling mechanisms must identify lowest retention time in the presence of VRT
  - Question: How long to profile a cell to find its lowest retention time state?

Time Between Retention Time State Changes

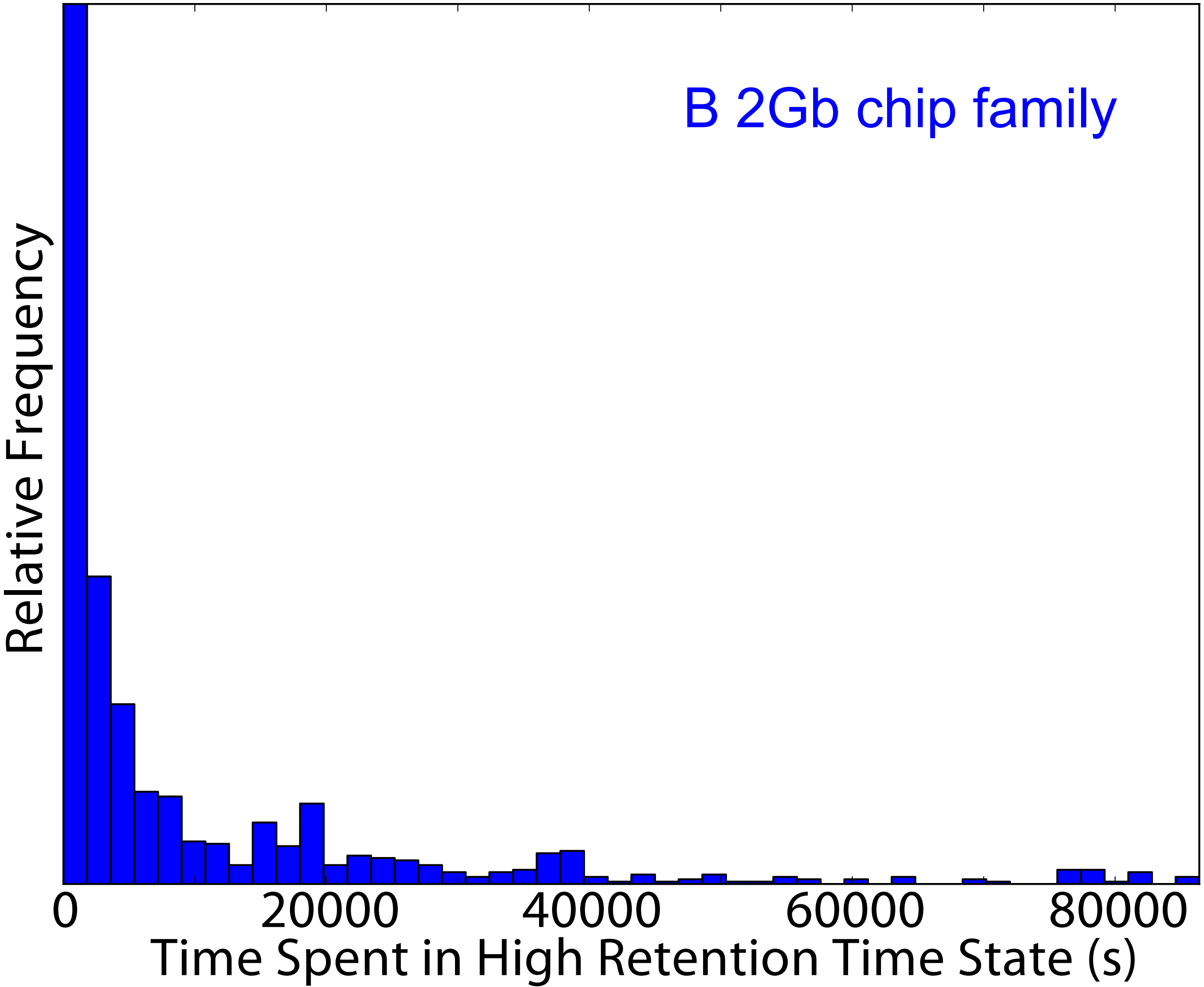
- How much time does a cell spend in a high retention state before switching to the minimum observed retention time state?

Time Spent in High Retention Time State

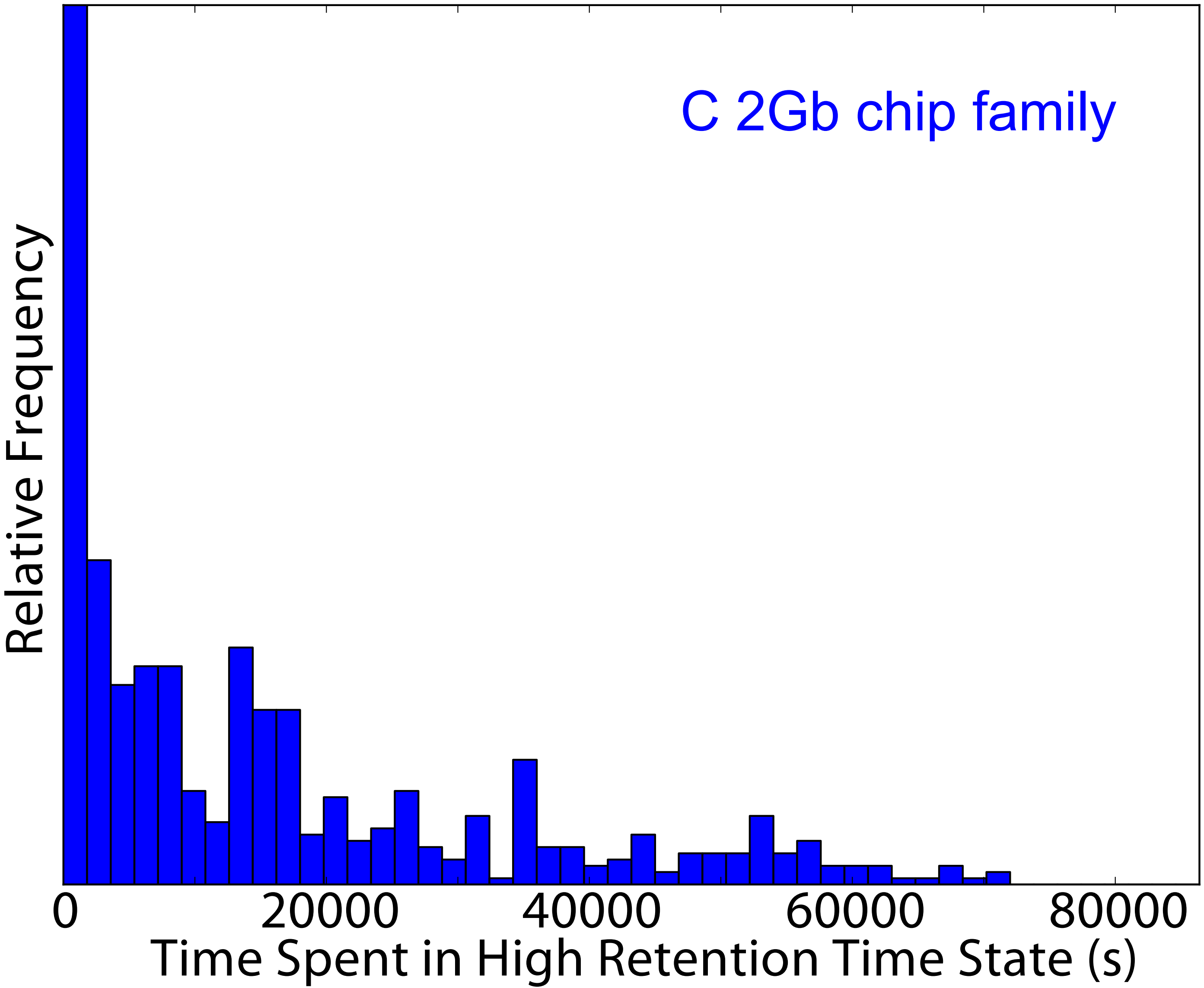


Need to profile for a long time to get to the minimum retention time state

Time Spent in High Retention Time State



Time Spent in High Retention Time State



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

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:

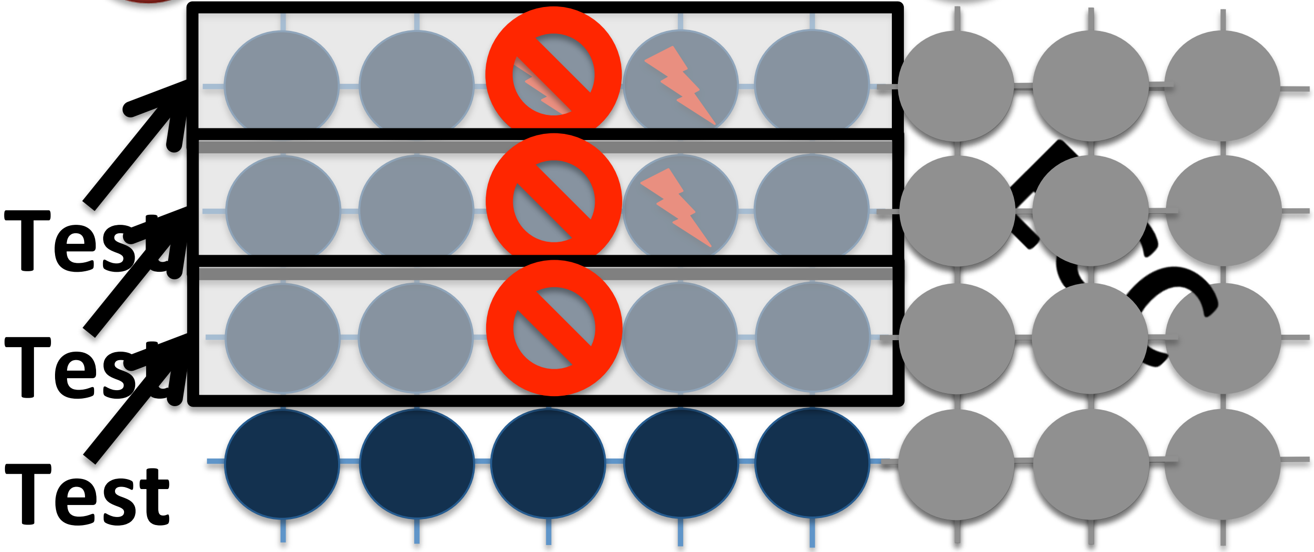
- **Testing alone cannot detect all possible failures**
- **Combination of ECC and other mitigation techniques is much more effective**
  - **But degrades performance**
- **Testing can help to reduce the ECC strength**
  - **Even when starting with a higher strength ECC**

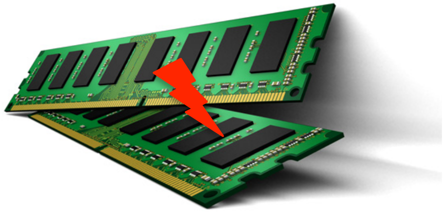
Khan+, "The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study," SIGMETRICS 2014.

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

Agenda

- Major Trends Affecting Main Memory
- The DRAM Scaling Problem
- Refresh as a Limiter of DRAM Scaling
- Some Solution Directions and Challenges
- Summary

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]
- Many avenues for overcoming DRAM refresh challenges
  - Handling DPD/VRT phenomena
  - Enabling online retention time profiling and error mitigation
  - Exploiting application behavior

Referenced Papers

- All are available at http://users.ece.cmu.edu/~omutlu/projects.htm

Related Videos and Course Materials

- Computer Architecture Lecture Videos on Youtube
    - https://www.youtube.com/playlist?list=PL5PHm2jkkXmidJOd59REog9jDnPDTG6IJ
- Computer Architecture Course Materials
    - http://www.ece.cmu.edu/~ece447/s13/doku.php?id=schedule
- Advanced Computer Architecture Course Materials
    - http://www.ece.cmu.edu/~ece740/f13/doku.php?id=schedule
- Advanced Computer Architecture Lecture Videos on Youtube
    - https://www.youtube.com/playlist?list=PL5PHm2jkkXmgDN1PLwOY_tGtUlynnyV6D

Thank you.

Feel free to email me with any questions & feedback

onur@cmu.edu
http://users.ece.cmu.edu/~omutlu/

Understanding and Overcoming Challenges of DRAM Refresh

Onur Mutlu
onur@cmu.edu

June 30, 2014
Extreme Scale Scientific Computing Workshop

Carnegie Mellon

SAFARI

Additional Slides

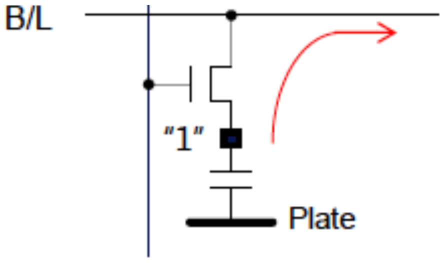
DRAM Process Scaling Challenges

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

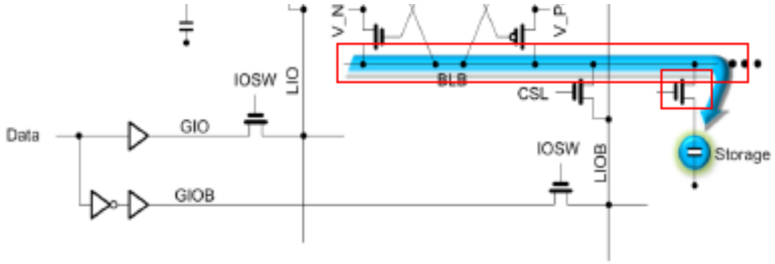
Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

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

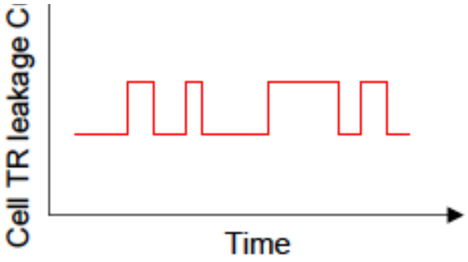
*Samsung Electronics, Hwasung, Korea / *Samsung Electronics, San Jose / **Intel*



Refresh



tWR



VRT

Number of Disturbance Errors

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

- *In a more controlled environment, we can induce as many as ten million disturbance errors*
- *Disturbance errors are a serious reliability issue*

Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

Summary and Conclusions [ISCA'13]

- DRAM refresh is a critical challenge in scaling DRAM technology efficiently to higher capacities and smaller feature sizes
- Understanding the retention time of modern DRAM devices can enable old or new methods to reduce the impact of refresh
  - Many mechanisms require accurate and reliable retention time profiles

- We presented the first work that comprehensively examines data retention behavior in modern commodity DRAM devices
  - Characterized 248 devices from five manufacturers

- Key findings: Retention time of a cell significantly depends on data pattern stored in other cells (data pattern dependence) and changes over time via a random process (variable retention time)
  - Discussed the underlying reasons and provided suggestions
- Future research on retention time profiling should solve the challenges posed by the DPD and VRT phenomena

DPD: Suggestions (for Future Work)

- A mechanism for identifying worst-case data pattern(s) likely requires support from DRAM device
  - DRAM manufacturers might be in a better position to do this
  - But, the ability of the manufacturer to identify and expose the entire retention time profile is limited due to VRT

- An alternative approach: Use random data patterns to increase coverage as much as possible; handle incorrect retention time estimates with ECC
  - Need to keep profiling time in check
  - Need to keep ECC overhead in check

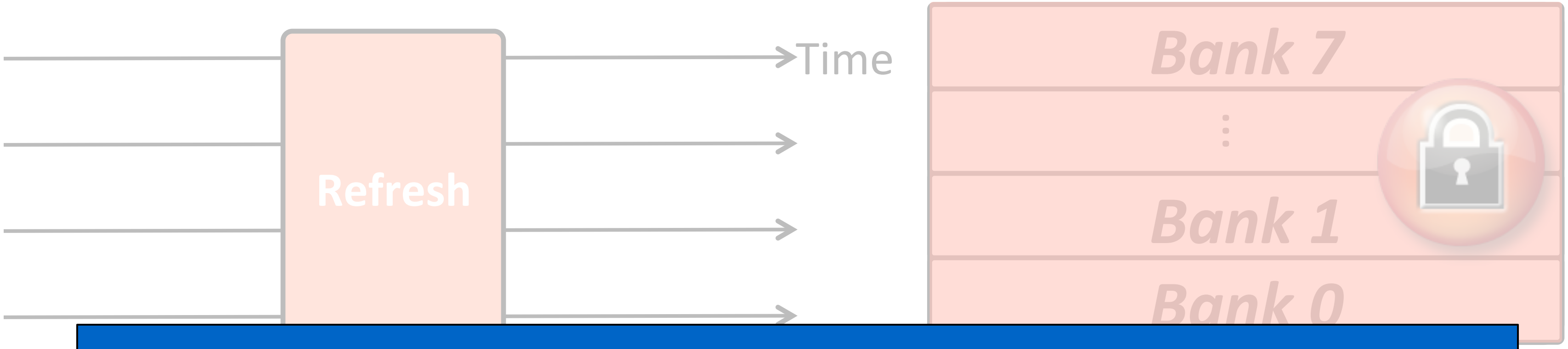
Refresh Penalty

Refresh interferes with memory accesses

Refresh delays requests by 100s of ns

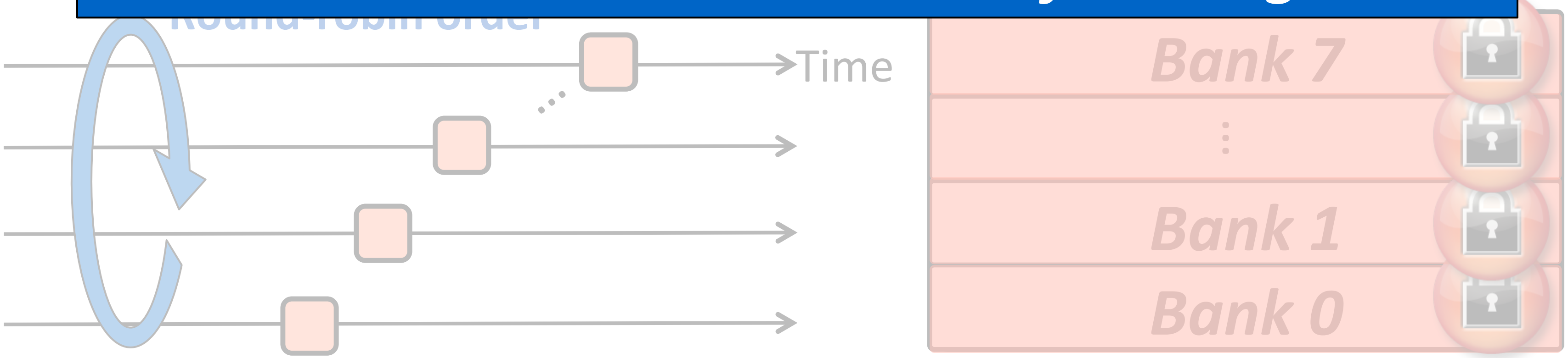
Existing Refresh Modes

All-bank refresh in commodity DRAM (DDRx)



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

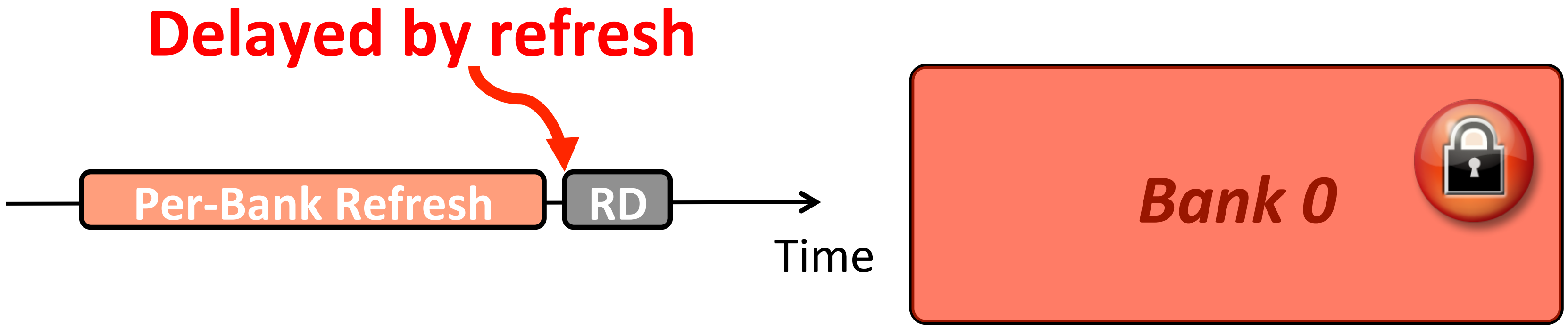
- **Dynamic Access-Refresh Parallelization (DARP)**
  - An improved scheduling policy for per-bank refreshes
  - Exploits refresh scheduling flexibility in DDR DRAM

- 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**

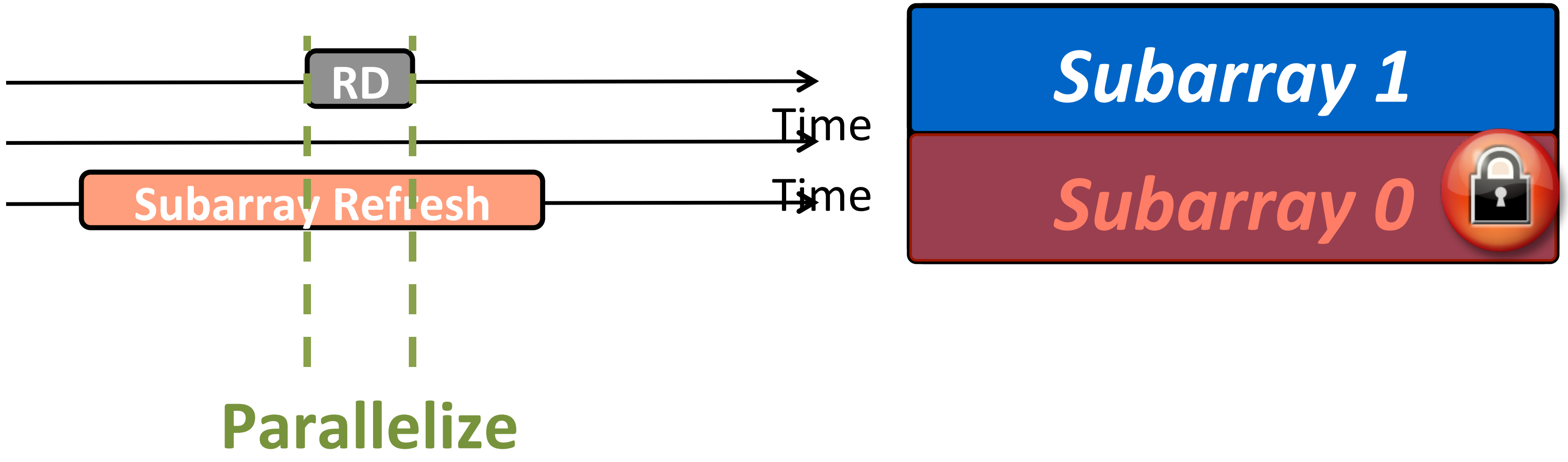
Shortcomings of Per-Bank Refresh

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



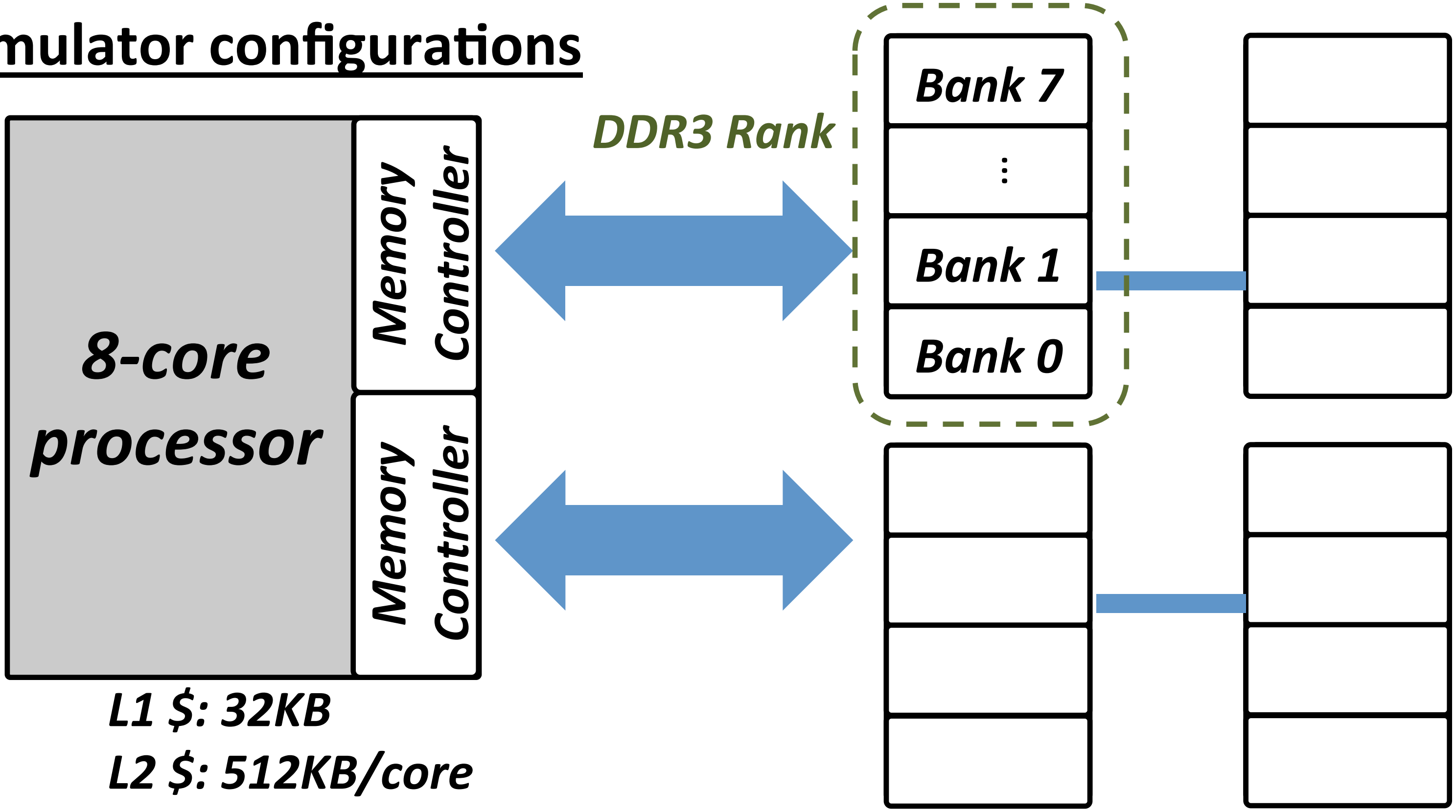
Shortcomings of Per-Bank Refresh

- 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

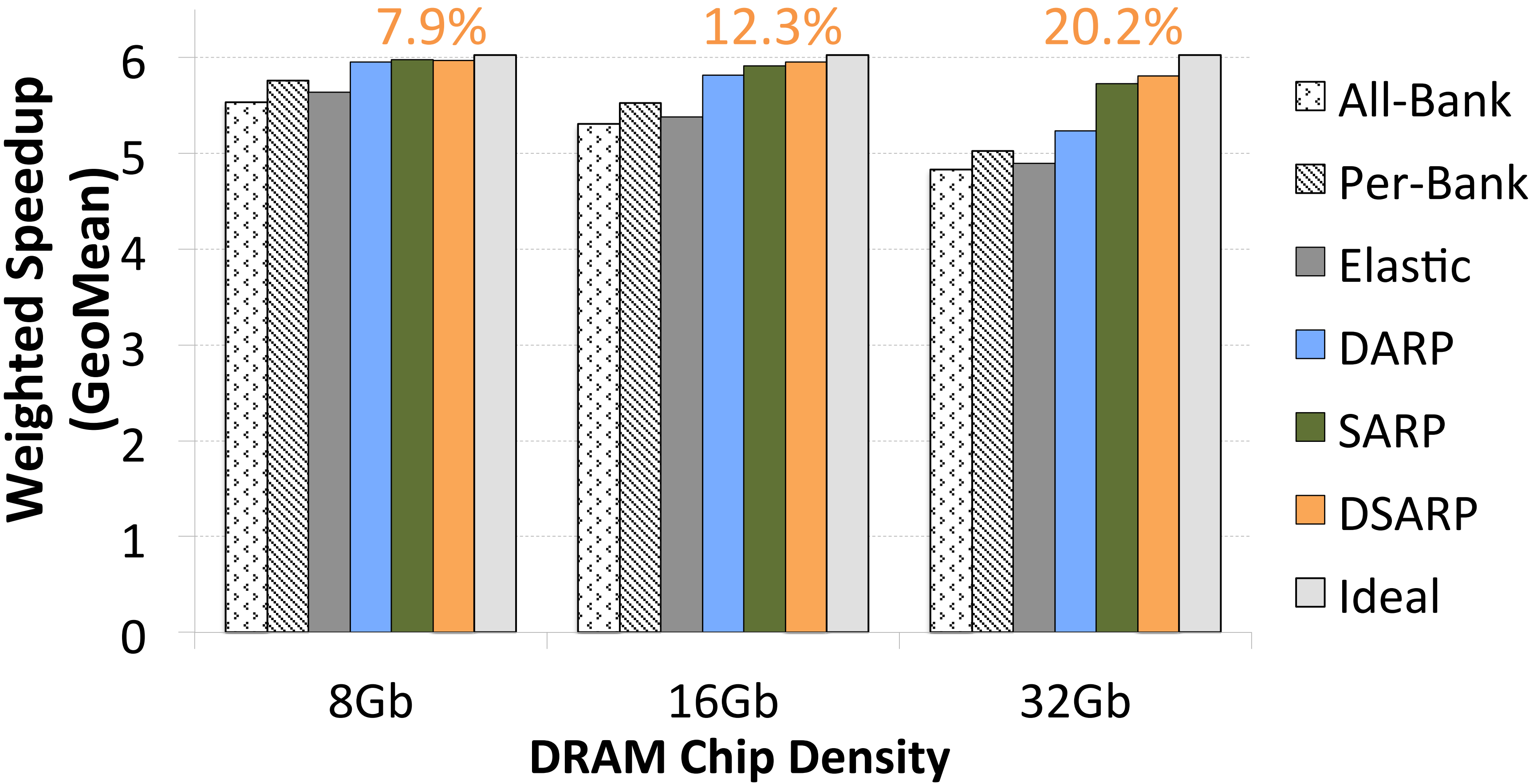
- **All-bank refresh** [DDR3, LPDDR3, …]

- **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

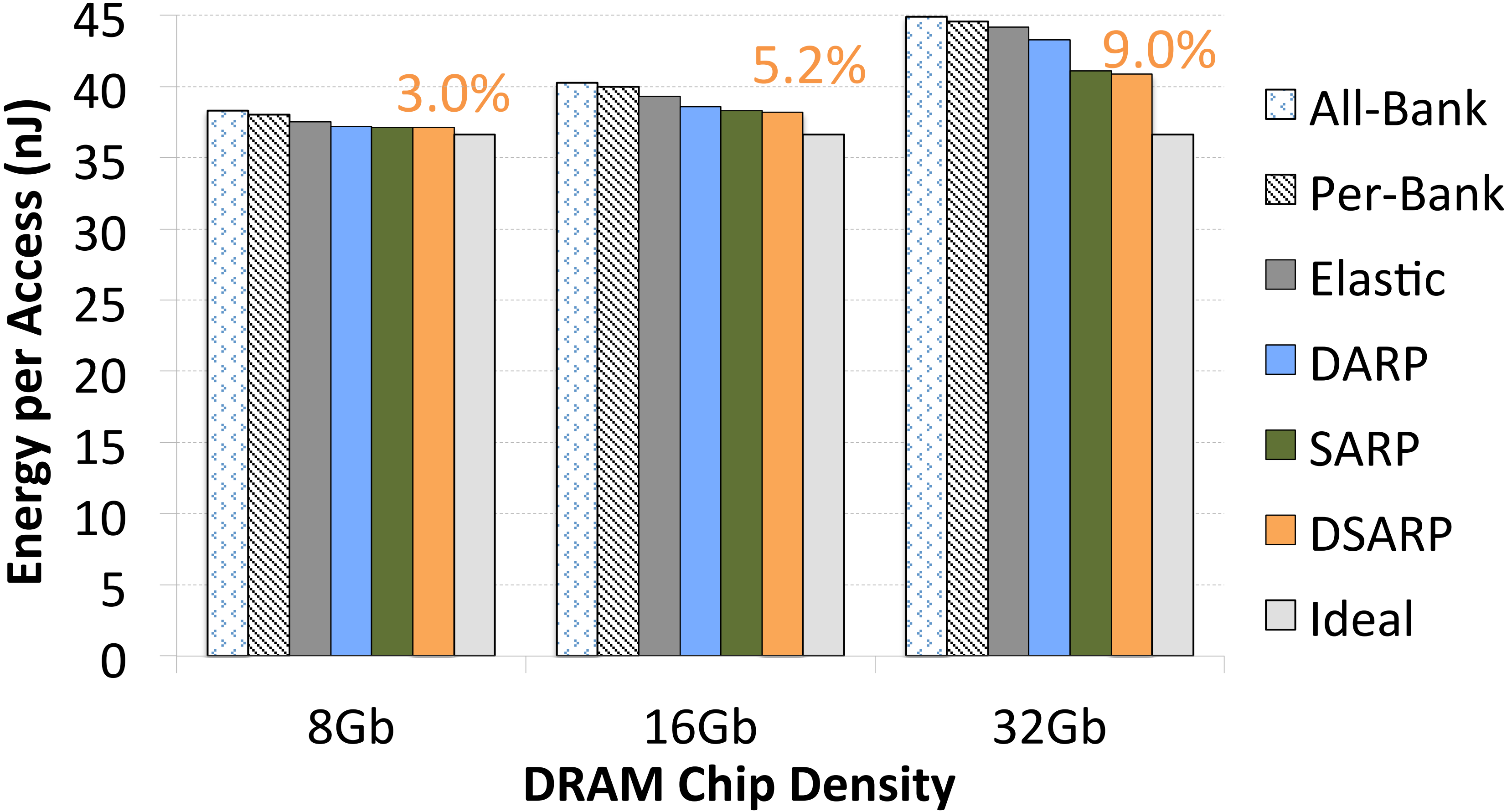
- **Ideal (no refresh)**

System Performance



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

Energy Efficiency

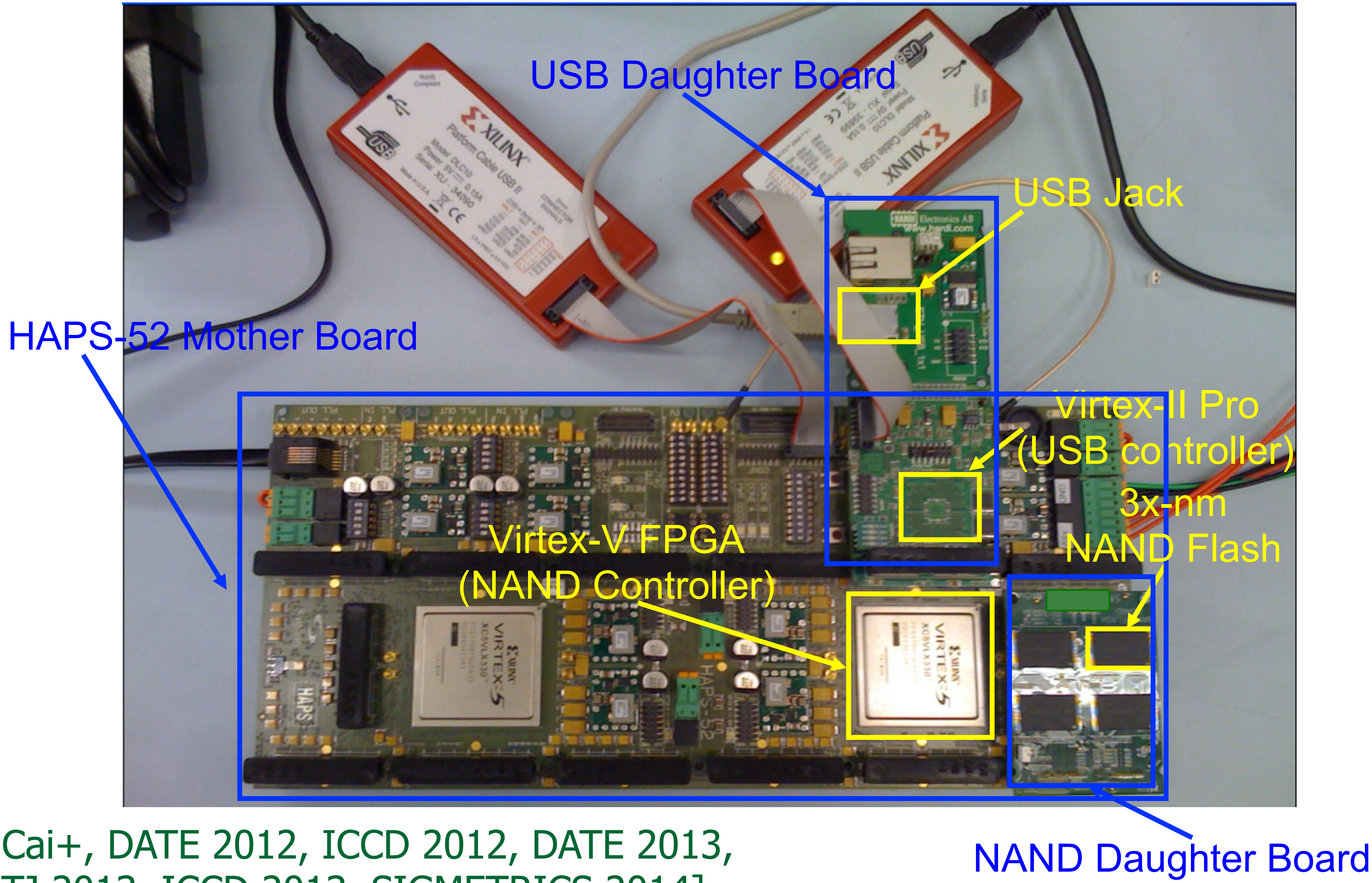


Consistent reduction on energy consumption

Another Talk: NAND Flash Scaling Challenges

- 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 Tech 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.

Experimental Infrastructure (Flash)



[Cai+, DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014]