Memory Systems
and Memory-Centric Computing Systems
Lecture 6a: Low-Latency Memory II

Prof. Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu

19 June 2019
TU Wien Fast Course 2019

SAFARI ETH zürich Carnegie Mellon

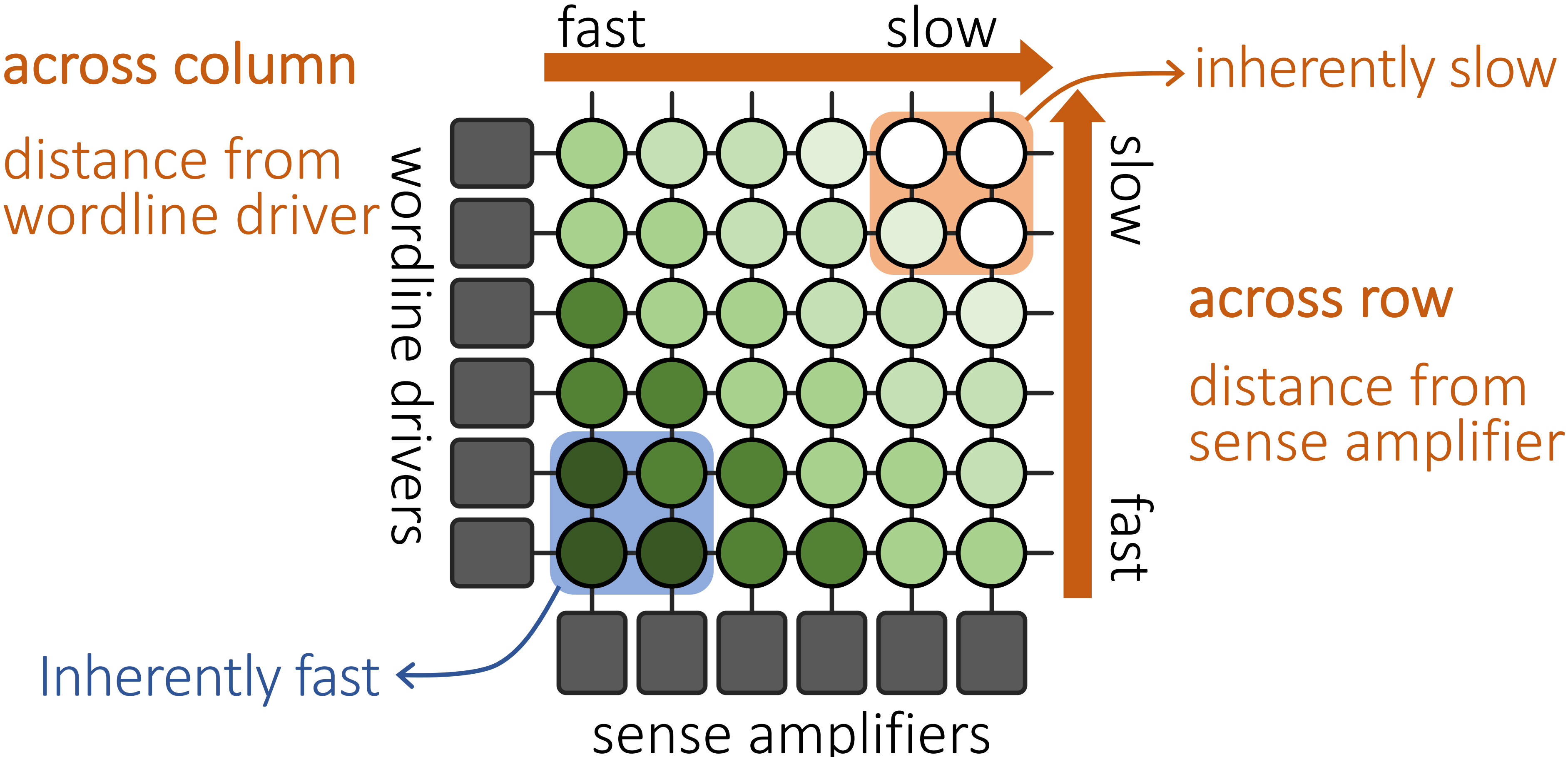
Why the Long Memory Latency?

- Reason 1: Design of DRAM Micro-architecture
  - Goal: Maximize capacity/area, not minimize latency

- Reason 2: "One size fits all" approach to latency specification
  - Same latency parameters for all temperatures
  - Same latency parameters for all DRAM chips
  - Same latency parameters for all parts of a DRAM chip
  - Same latency parameters for all supply voltage levels
  - Same latency parameters for all application data
  - …

Why Is There Spatial Latency Variation Within a Chip?

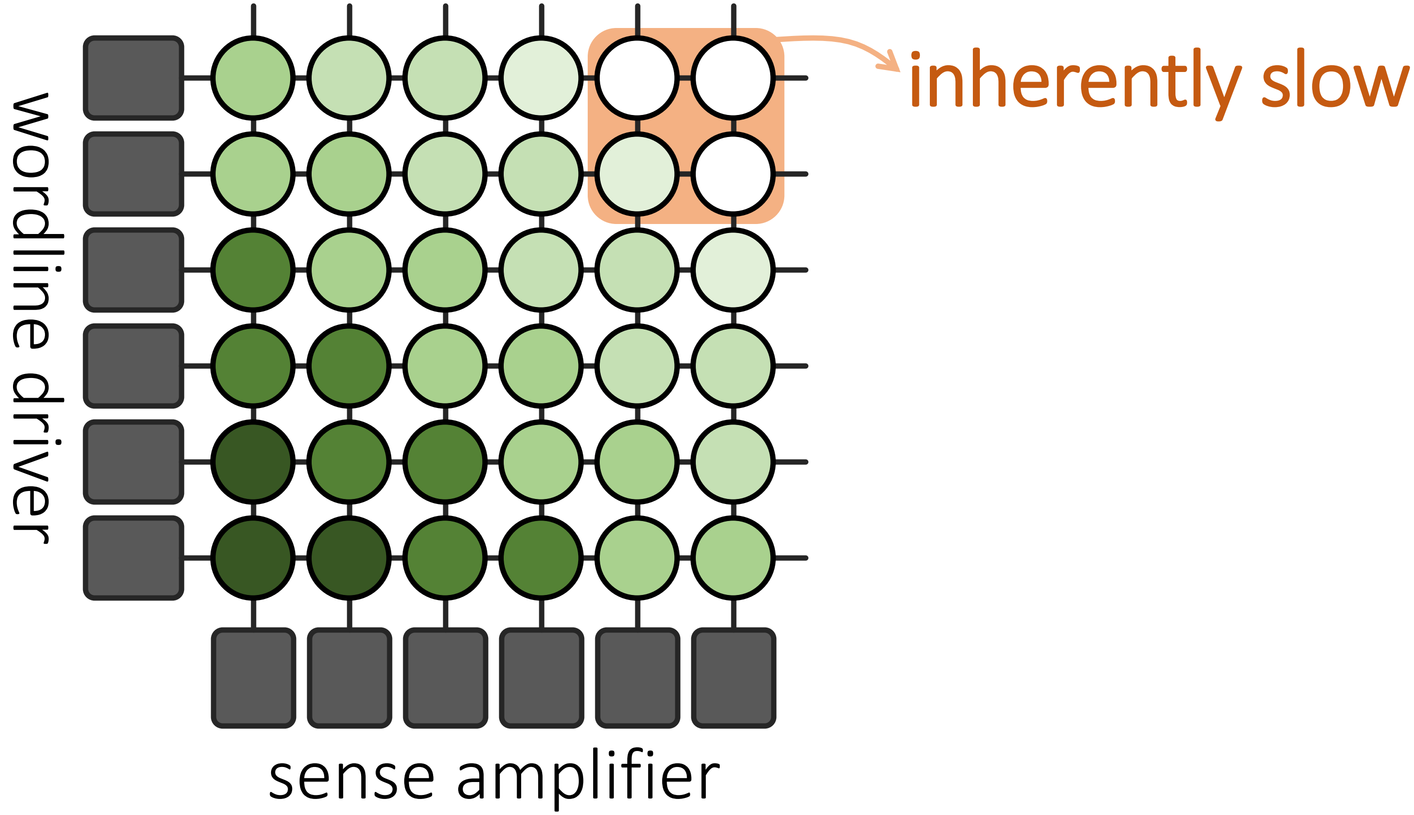
What Is Design-Induced Variation?



Systematic variation in cell access times caused by the ***physical organization*** of DRAM

DIVA Online **Profiling**

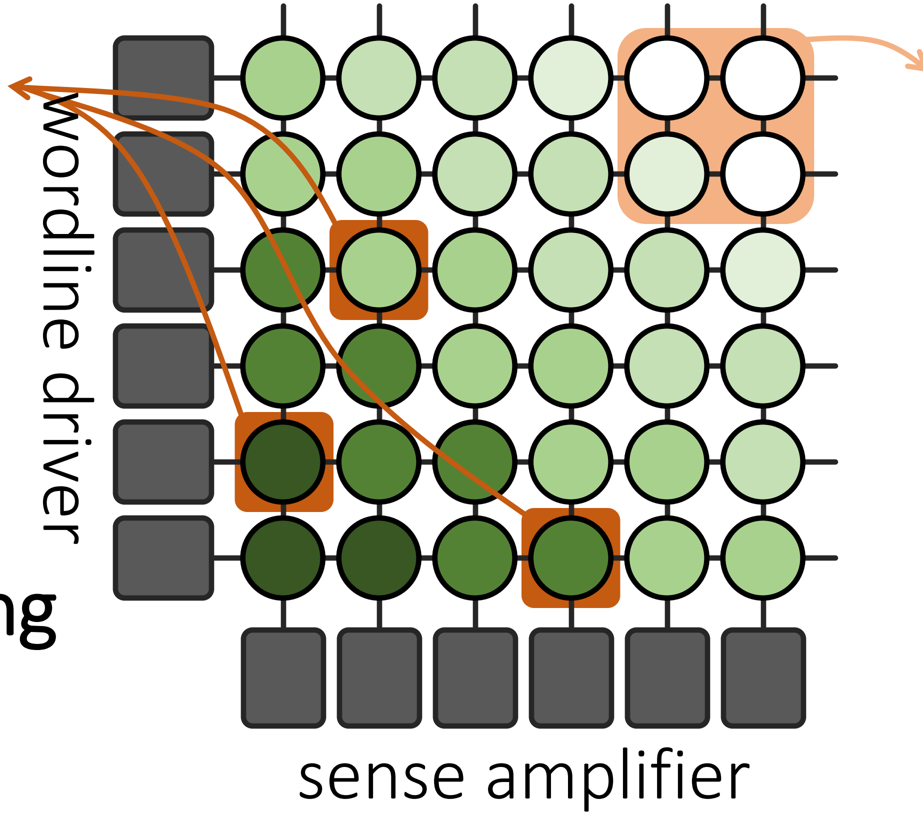
Design-**I**nduced-**V**ariation-**A**ware



Profile *only slow regions* to determine min. latency
→ *Dynamic* & *low cost* latency optimization

DIVA Online Profiling

Design-**I**nduced-**V**ariation-**A**ware



slow cells

process variation

random error

error-correcting code

inherently slow

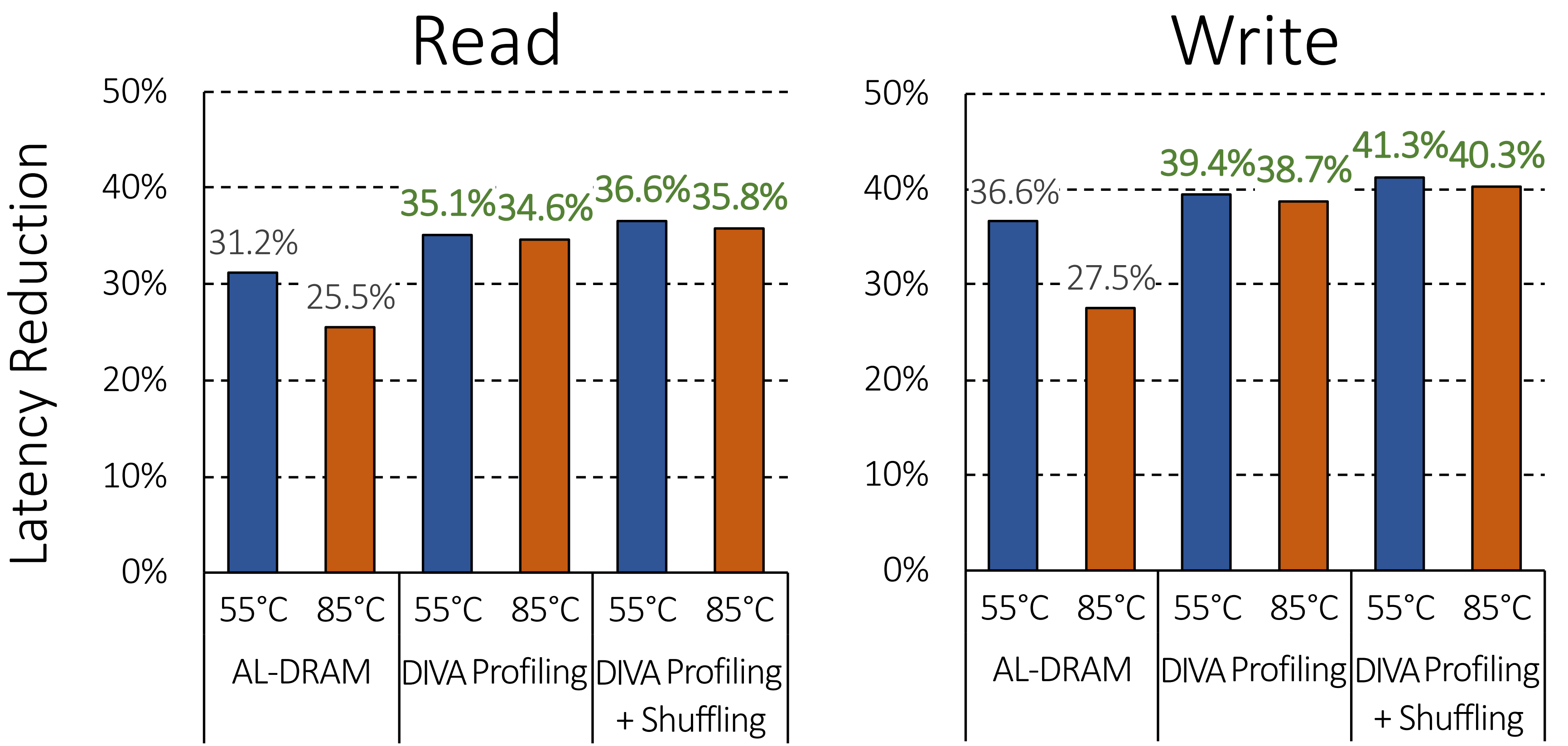
design-induced variation

localized error

online profiling

Combine error-correcting codes & online profiling
→ Reliably reduce DRAM latency

DIVA-DRAM Reduces Latency



DIVA-DRAM ***reduces latency more aggressively***
and uses ECC to correct random slow cells

SAFARI

DIVA-DRAM: Advantages & Disadvantages

- Advantages

++ Automatically finds the lowest reliable operating latency at system runtime (lower production-time testing cost)

+ Reduces latency more than prior methods (w/ ECC)

+ Reduces latency at high temperatures as well

- Disadvantages

- Requires knowledge of inherently-slow regions

- Requires ECC (Error Correcting Codes)

- Imposes overhead during runtime profiling

Design-Induced Latency Variation in DRAM

- Donghyuk Lee, Samira Khan, Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Gennady Pekhimenko, Vivek Seshadri, and Onur Mutlu,
"Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms"
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**)*, Urbana-Champaign, IL, USA, June 2017.

Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms

Donghyuk Lee, NVIDIA and Carnegie Mellon University
Samira Khan, University of Virginia
Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Carnegie Mellon University
Gennady Pekhimenko, Vivek Seshadri, Microsoft Research
Onur Mutlu, ETH Zürich and Carnegie Mellon University

Understanding & Exploiting the Voltage-Latency-Reliability Relationship

High DRAM Power Consumption

- Problem: High DRAM (memory) power in today's systems

>40% in POWER7 (Ware+, HPCA'10)

>40% in GPU (Paul+, ISCA'15)

Low-Voltage Memory

- Existing DRAM designs to help reduce DRAM power by lowering supply voltage conservatively
  - $Power \propto Voltage^2$
- DDR3L (low-voltage) reduces voltage from 1.5V to 1.35V (**-10%**)
- LPDDR4 (low-power) employs low-power I/O interface with 1.2V (lower bandwidth)

Can we reduce DRAM power and energy by *further* reducing supply voltage?

Goals

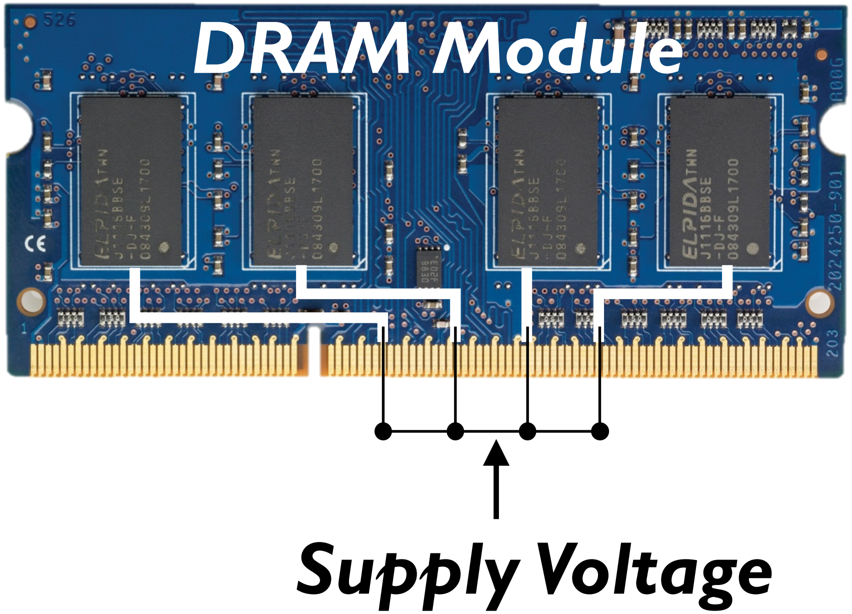
1 Understand and characterize the various characteristics of DRAM under **reduced voltage**

2 Develop a mechanism that reduces DRAM energy by **lowering voltage** while keeping performance loss within a target

Key Questions

- How does reducing voltage affect ***reliability*** (errors)?
- How does reducing voltage affect ***DRAM latency***?
- How do we design a new DRAM energy reduction mechanism?

Supply Voltage Control on DRAM



Adjust the *supply voltage* to every chip on the same module

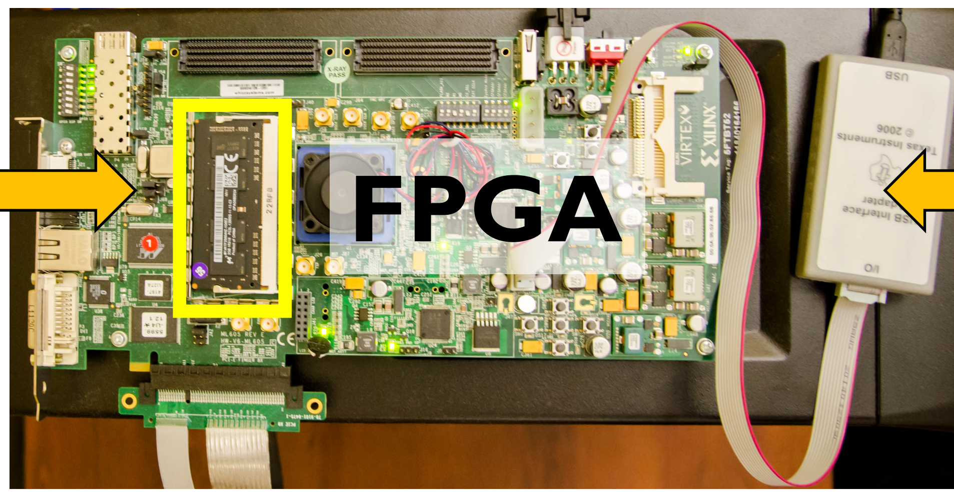
Custom Testing Platform

SoftMC [Hassan+, HPCA'17]: FPGA testing platform to

1) Adjust supply voltage to DRAM modules
2) Schedule DRAM commands to DRAM modules

Existing systems: DRAM commands not exposed to users

DRAM module



Voltage controller

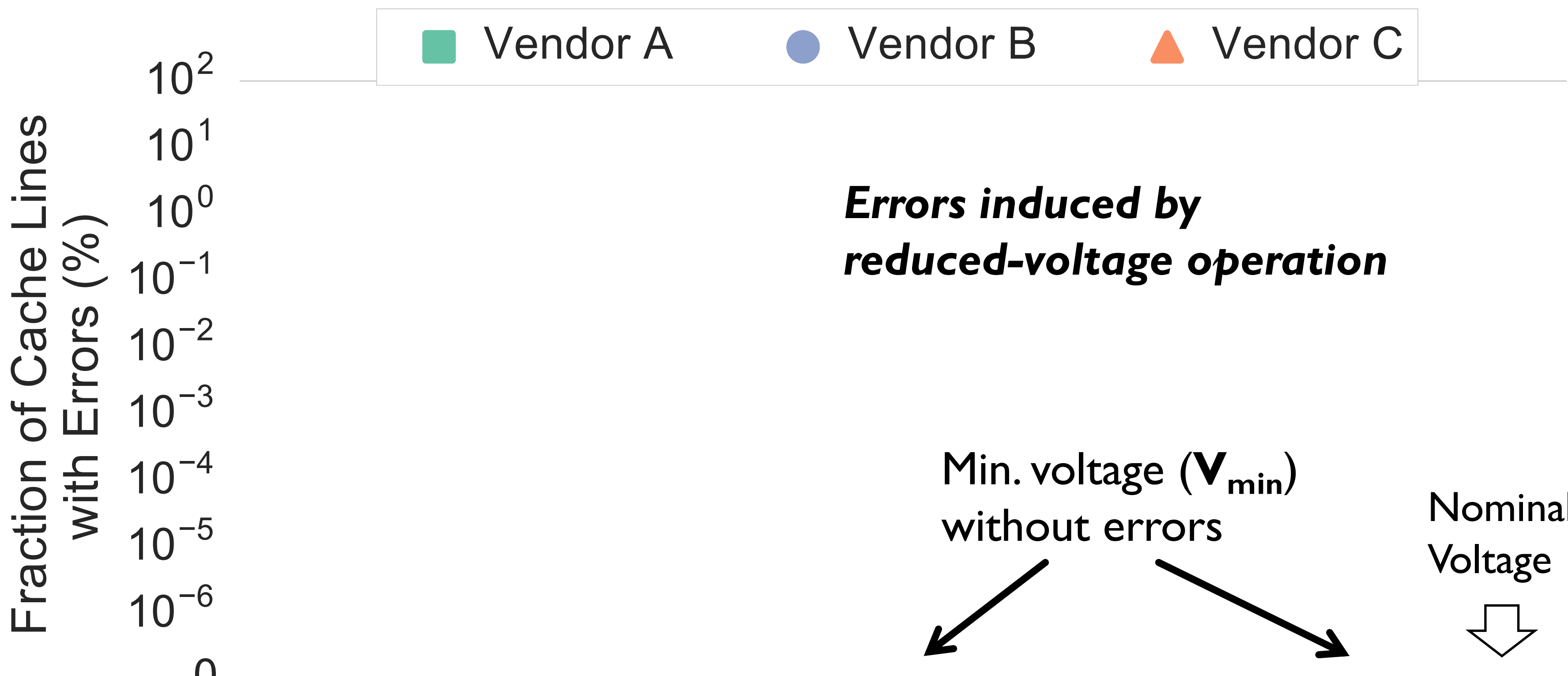
https://github.com/CMU-SAFARI/DRAM-Voltage-Study

Tested DRAM Modules

- **124 DDR3L** (low-voltage) DRAM chips
  - **31 SO-DIMMs**
  - **1.35V** (DDR3 uses 1.5V)
  - Density: 4Gb per chip
  - Three major vendors/manufacturers
  - Manufacturing dates: 2014-2016

- Iteratively read every bit in each 4Gb chip under a wide range of supply voltage levels: 1.35V to 1.0V (**-26%**)

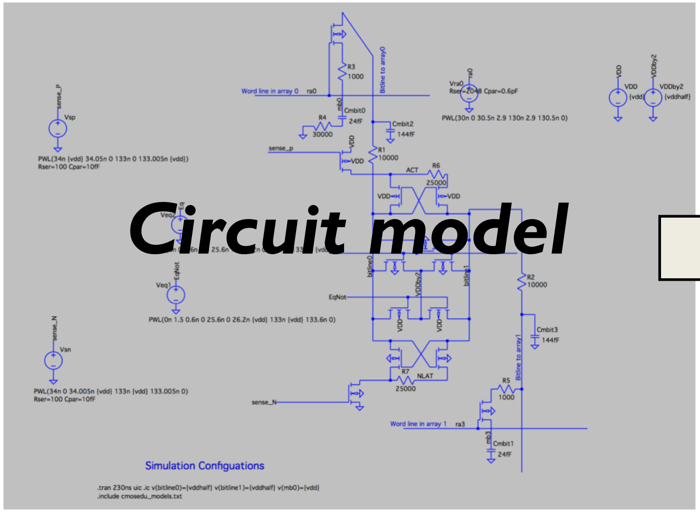
Reliability Worsens with Lower Voltage

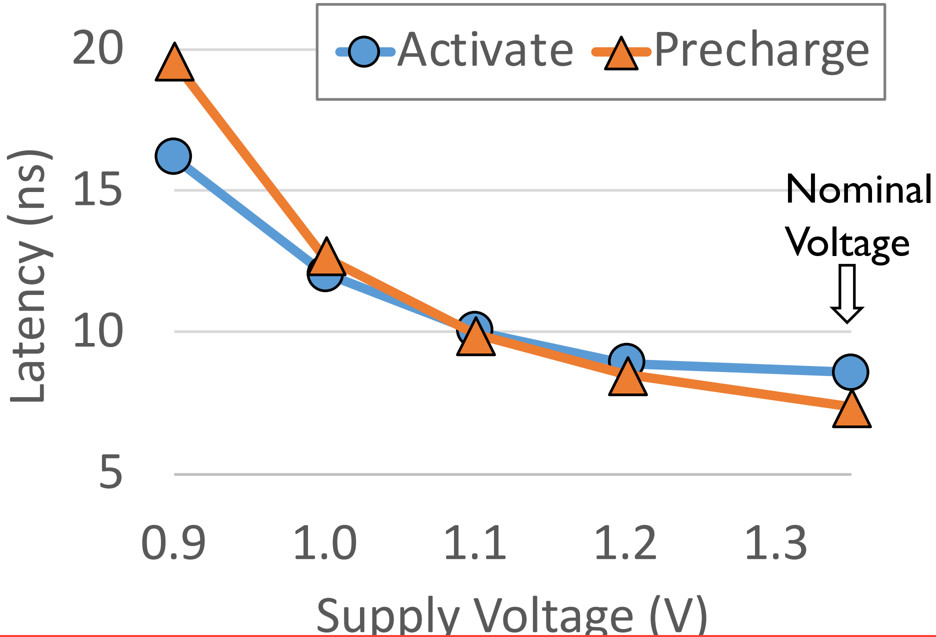


Reducing voltage below V_{min} causes an increasing number of errors

Source of Errors

Detailed circuit simulations (SPICE) of a DRAM cell array to model the behavior of DRAM operations
https://github.com/CMU-SAFARI/DRAM-Voltage-Study

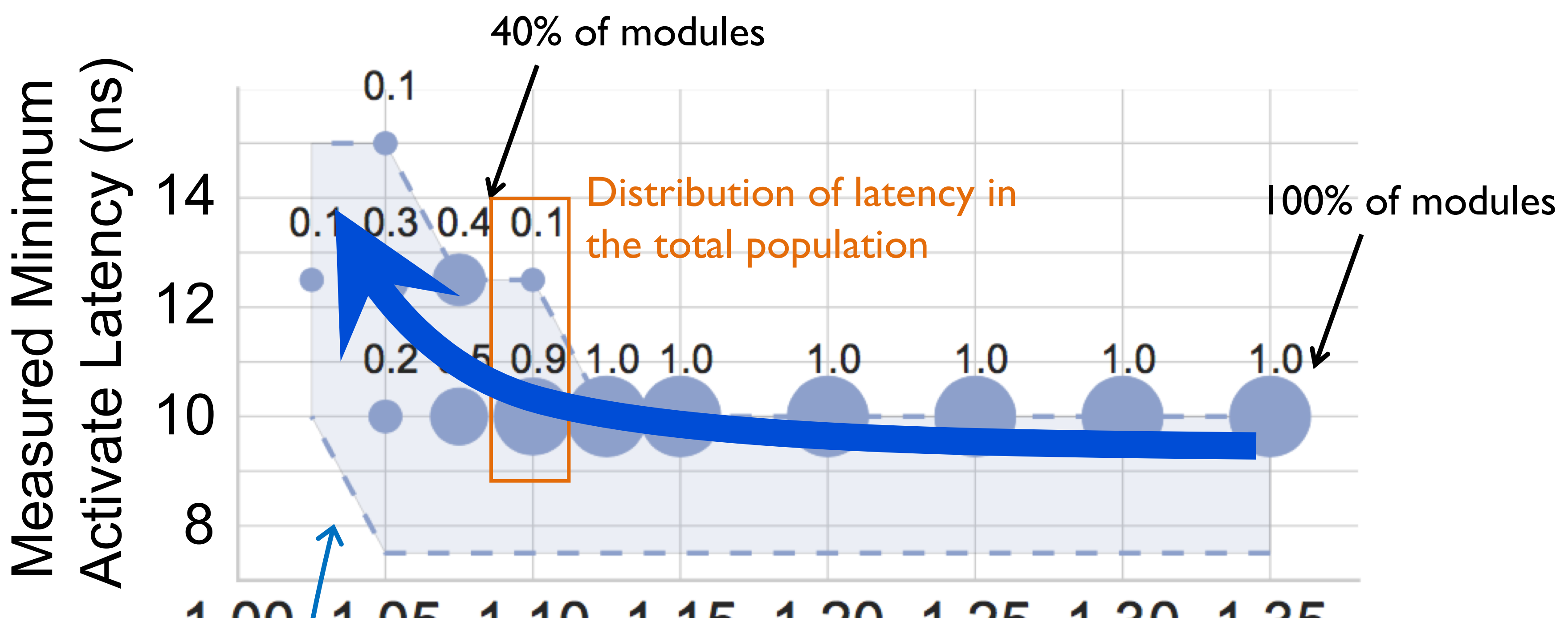




Reliable low-voltage operation requires higher latency

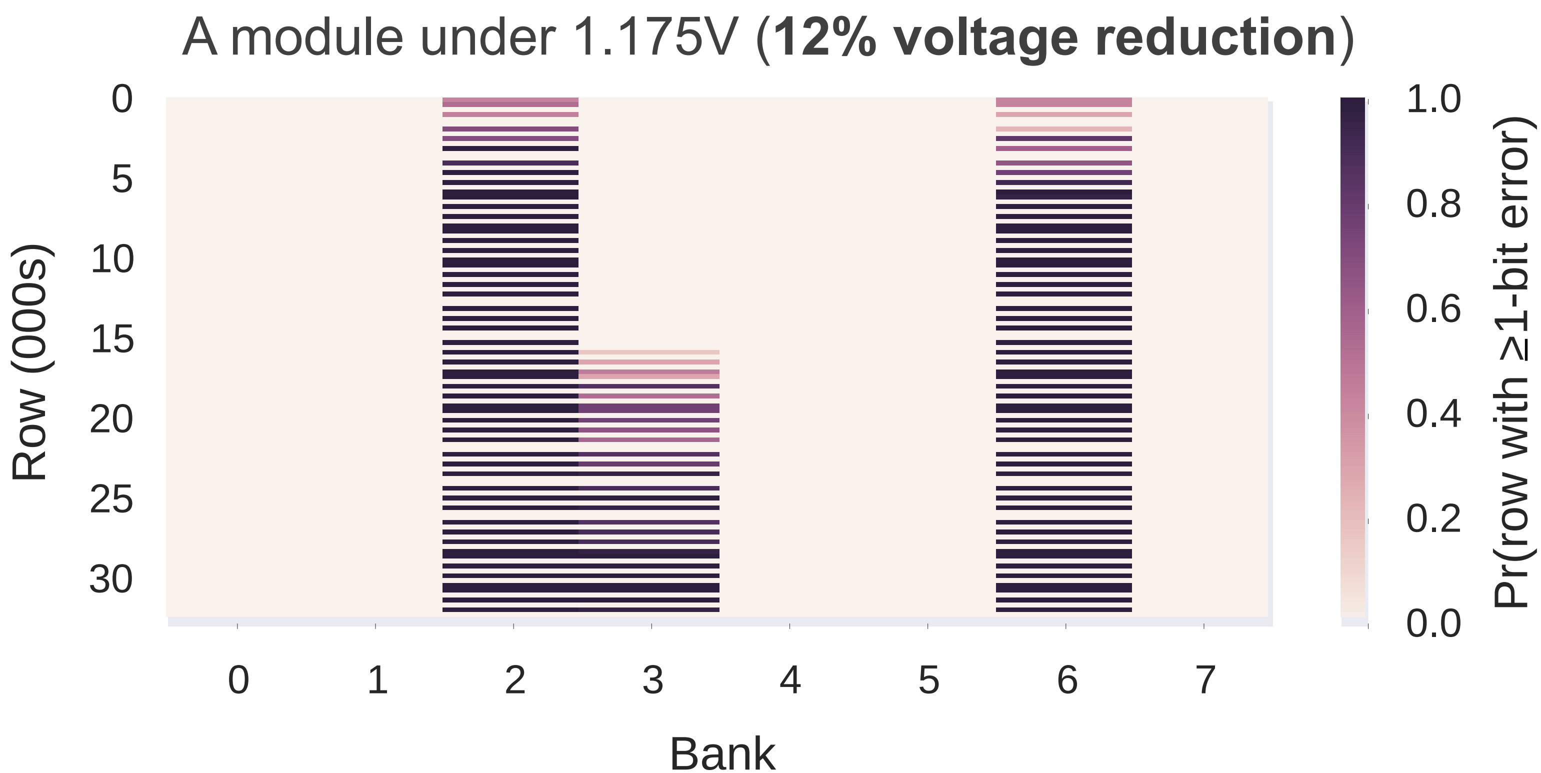
DIMMs Operating at Higher Latency

Measured minimum latency that *does **not** cause errors* in DRAM modules



DRAM requires longer latency to access data **without errors** at lower voltage

Spatial Locality of Errors



Errors concentrate in certain regions

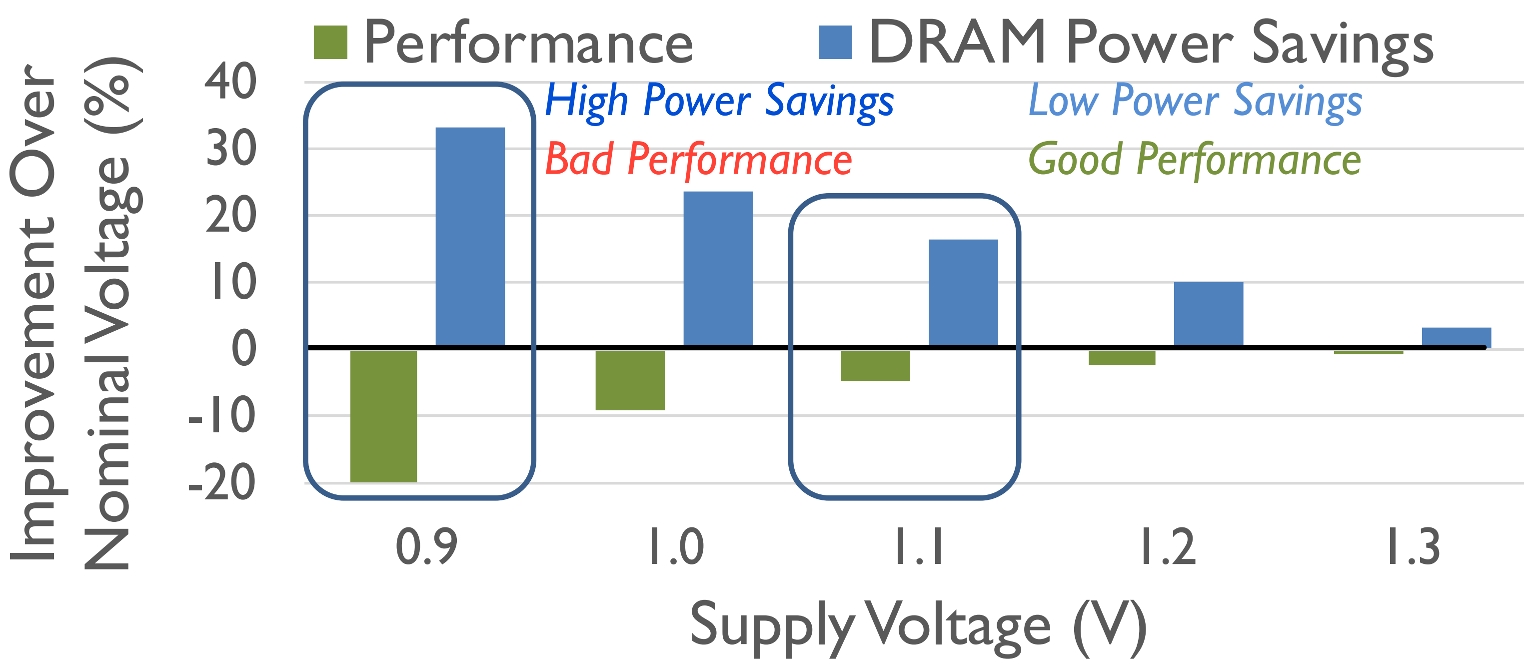
Summary of Key Experimental Observations

- Voltage-induced errors increase as voltage reduces further below V_{min}
- Errors exhibit spatial locality
- Increasing the latency of DRAM operations mitigates voltage-induced errors

DRAM Voltage Adjustment to Reduce Energy

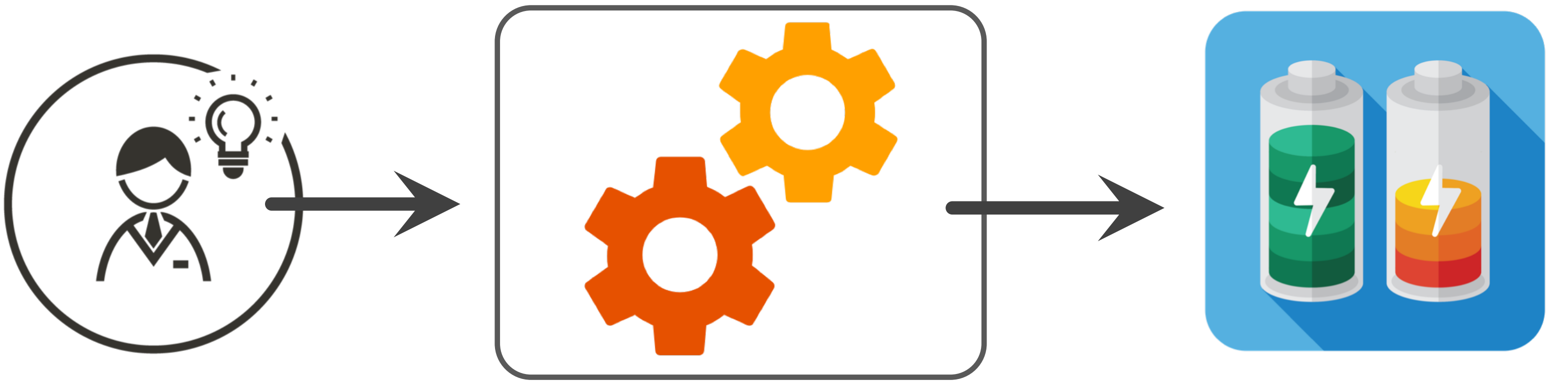
- <u>Goal</u>: Exploit the trade-off between voltage and latency to reduce energy consumption

- <u>Approach</u>: Reduce DRAM voltage **reliably**
  - Performance loss due to increased latency at lower voltage



Voltron Overview

Voltron

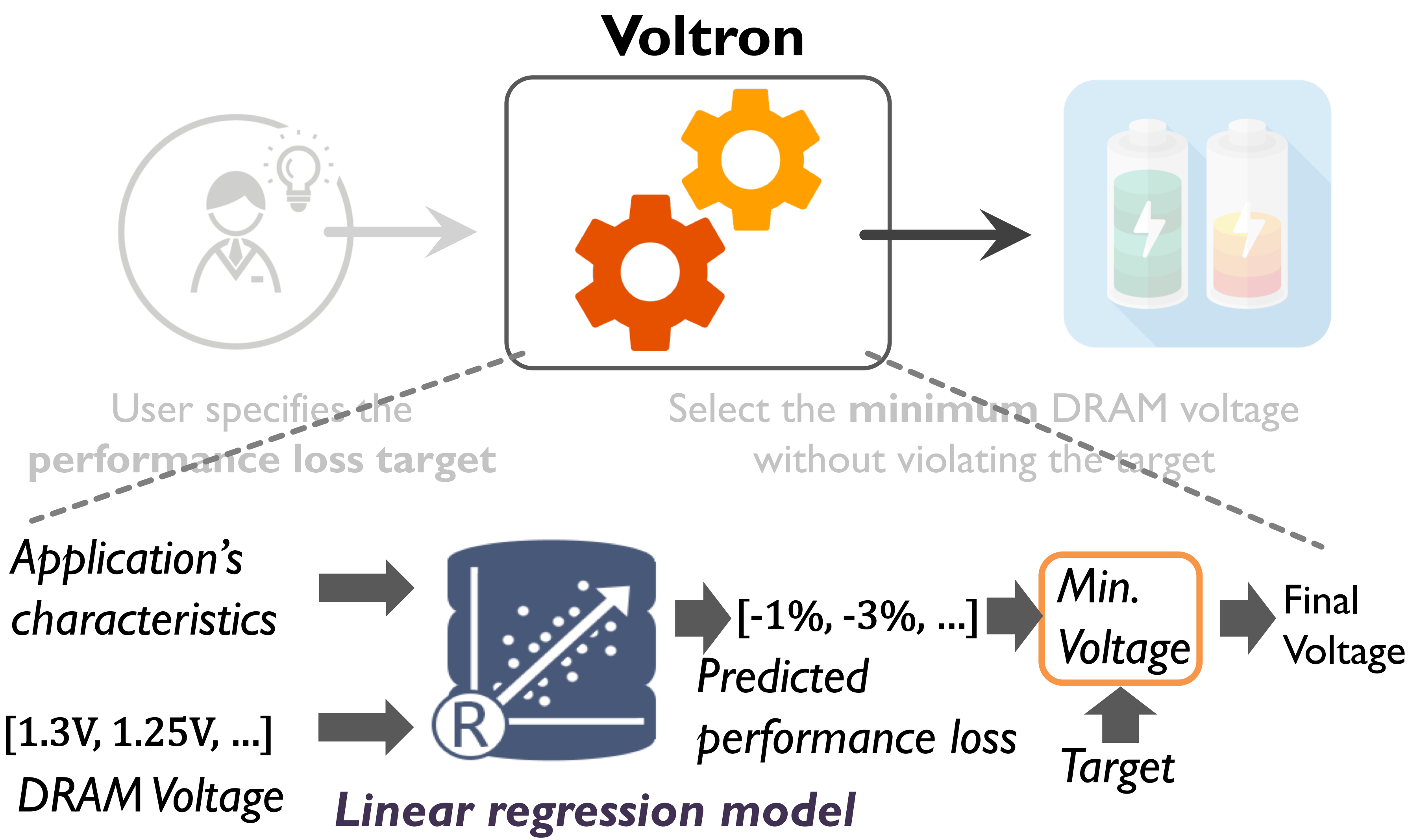


User specifies the **performance loss target**

Select the **minimum** DRAM voltage without violating the target

How do we predict performance loss due to increased latency under low DRAM voltage?

Linear Model to Predict Performance



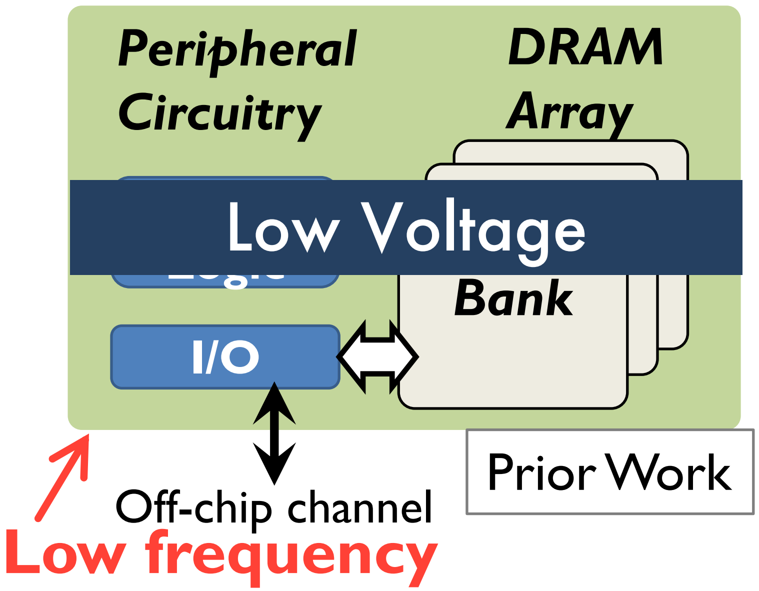
Regression Model to Predict Performance

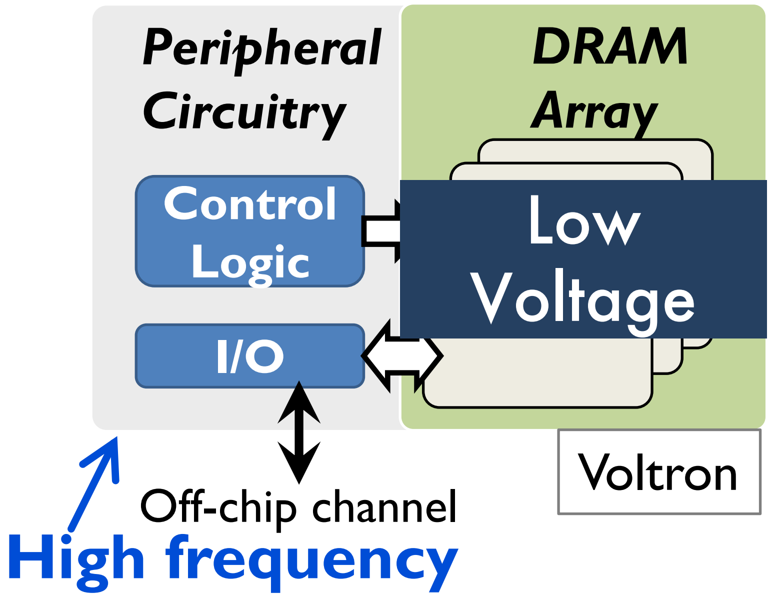
- Application's characteristics for the model:
  - ***Memory intensity***: Frequency of last-level cache misses
  - ***Memory stall time***: Amount of time memory requests stall commit inside CPU

- Handling multiple applications:
  - Predict a performance loss for each application
  - Select the minimum voltage that satisfies the performance target for all applications

Comparison to Prior Work

- Prior work: Dynamically scale *frequency and voltage* of the entire DRAM based on bandwidth demand [David+, ICAC'11]
  - Problem: Lowering voltage on the peripheral circuitry decreases channel frequency (memory data throughput)
- Voltron: Reduce voltage to only DRAM array without changing the voltage to peripheral circuitry



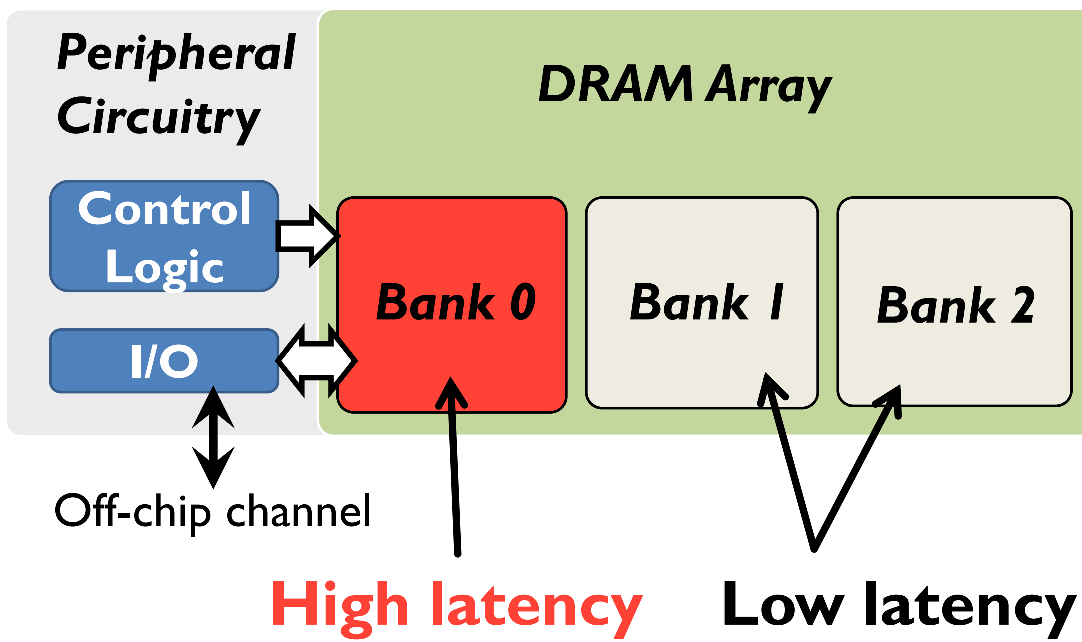


SAFARI

Exploiting Spatial Locality of Errors

<u>Key idea</u>: Increase the latency only for DRAM banks that observe errors under low voltage

- <u>Benefit</u>: Higher performance



Voltron Evaluation Methodology

- **Cycle-level simulator**: Ramulator [CAL'15]
  - **McPAT** and **DRAMPower** for energy measurement

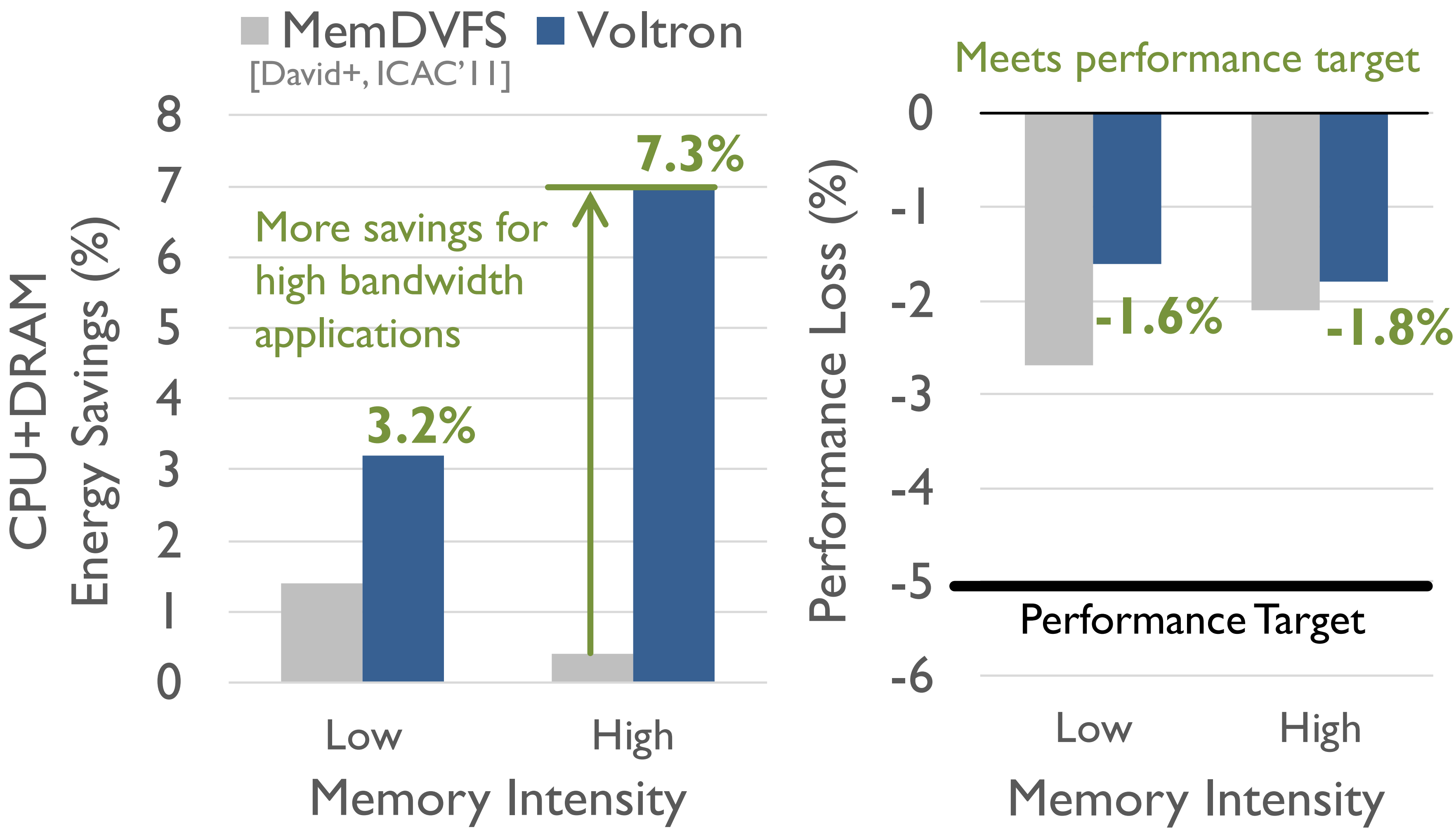
  https://github.com/CMU-SAFARI/ramulator

- **4-core** system with DDR3L memory

- **Benchmarks**: SPEC2006, YCSB

- Comparison to prior work: **MemDVFS** [David+, ICAC'11]
  - Dynamic DRAM frequency and voltage scaling
  - Scaling based on the *memory bandwidth consumption*

Energy Savings with Bounded Performance



SAFARI

Voltron: Advantages & Disadvantages

- Advantages

  + Can trade-off between voltage and latency to improve energy or performance

  + Can exploit the high voltage margin present in DRAM

- Disadvantages

  - Requires finding the reliable operating voltage for each chip → higher testing cost

Analysis of Latency-Voltage in DRAM Chips

- Kevin Chang, A. Giray Yaglikci, Saugata Ghose, Aditya Agrawal, Niladrish Chatterjee, Abhijith Kashyap, Donghyuk Lee, Mike O'Connor, Hasan Hassan, and Onur Mutlu,
"Understanding Reduced-Voltage Operation in Modern DRAM Devices: Experimental Characterization, Analysis, and Mechanisms"
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**)*, Urbana-Champaign, IL, USA, June 2017.

Understanding Reduced-Voltage Operation in Modern DRAM Chips: Characterization, Analysis, and Mechanisms

Kevin K. Chang[†] Abdullah Giray Yağlıkçı[†] Saugata Ghose[†] Aditya Agrawal[¶] Niladrish Chatterjee[¶]
Abhijith Kashyap[†] Donghyuk Lee[¶] Mike O'Connor[¶,‡] Hasan Hassan[§] Onur Mutlu[§,†]

[†]Carnegie Mellon University [¶]NVIDIA [‡]The University of Texas at Austin [§]ETH Zürich

And, What If …

- … we can sacrifice reliability of some data to access it with even lower latency?

Reducing Memory Latency to Support Security Primitives

Using Memory for Security

- Generating True Random Numbers (using DRAM)
  - Kim et al., HPCA 2019
- Evaluating Physically Unclonable Functions (using DRAM)
  - Kim et al., HPCA 2018
- Quickly Destroying In-Memory Data (using DRAM)
  - Orosa et al., arxiv 2019

D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim Minesh Patel
Hasan Hassan Lois Orosa Onur Mutlu

HPCA 2019

SAFARI

ETH Zürich Carnegie Mellon

D-RaNGe Executive Summary

- **Motivation**: High-throughput true random numbers enable system security and various randomized algorithms.
    - Many systems (e.g., IoT, mobile, embedded) do not have dedicated **True Random Number Generator (TRNG)** hardware but have DRAM devices
- **Problem**: Current DRAM-based TRNGs either
    1. do **not** sample a fundamentally non-deterministic entropy source
    2. are **too slow** for continuous high-throughput operation
- **Goal**: A novel and effective TRNG that uses **existing** commodity DRAM to provide random values with 1) **high-throughput,** 2) **low latency** and 3) no adverse effect on concurrently running applications
- **D-RaNGe:** Reduce DRAM access latency **below reliable values** and exploit DRAM cells' failure probabilities to generate random values
- **Evaluation:**
    1. Experimentally characterize **282 real LPDDR4 DRAM devices**
    2. **D-RaNGe (717.4 Mb/s)** has significantly higher throughput (**211x**)
    3. **D-RaNGe (100ns)** has significantly lower latency (**180x**)

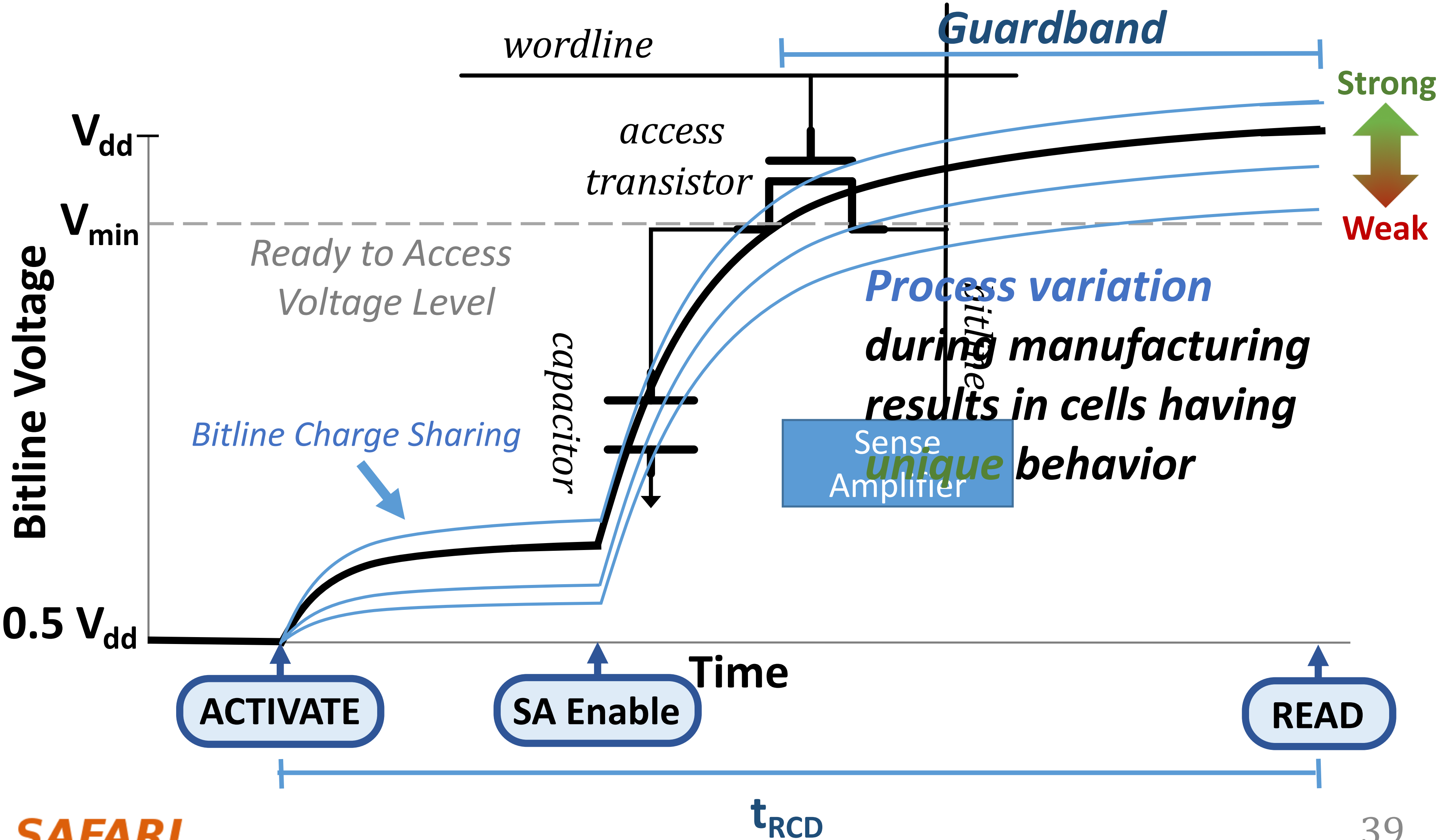
DRAM Latency Characterization of 282 LPDDR4 DRAM Devices

- Latency failures come from accessing DRAM with **reduced** timing parameters.

- **Key Observations:**
  1. A cell's **latency failure** probability is determined by **random process variation**

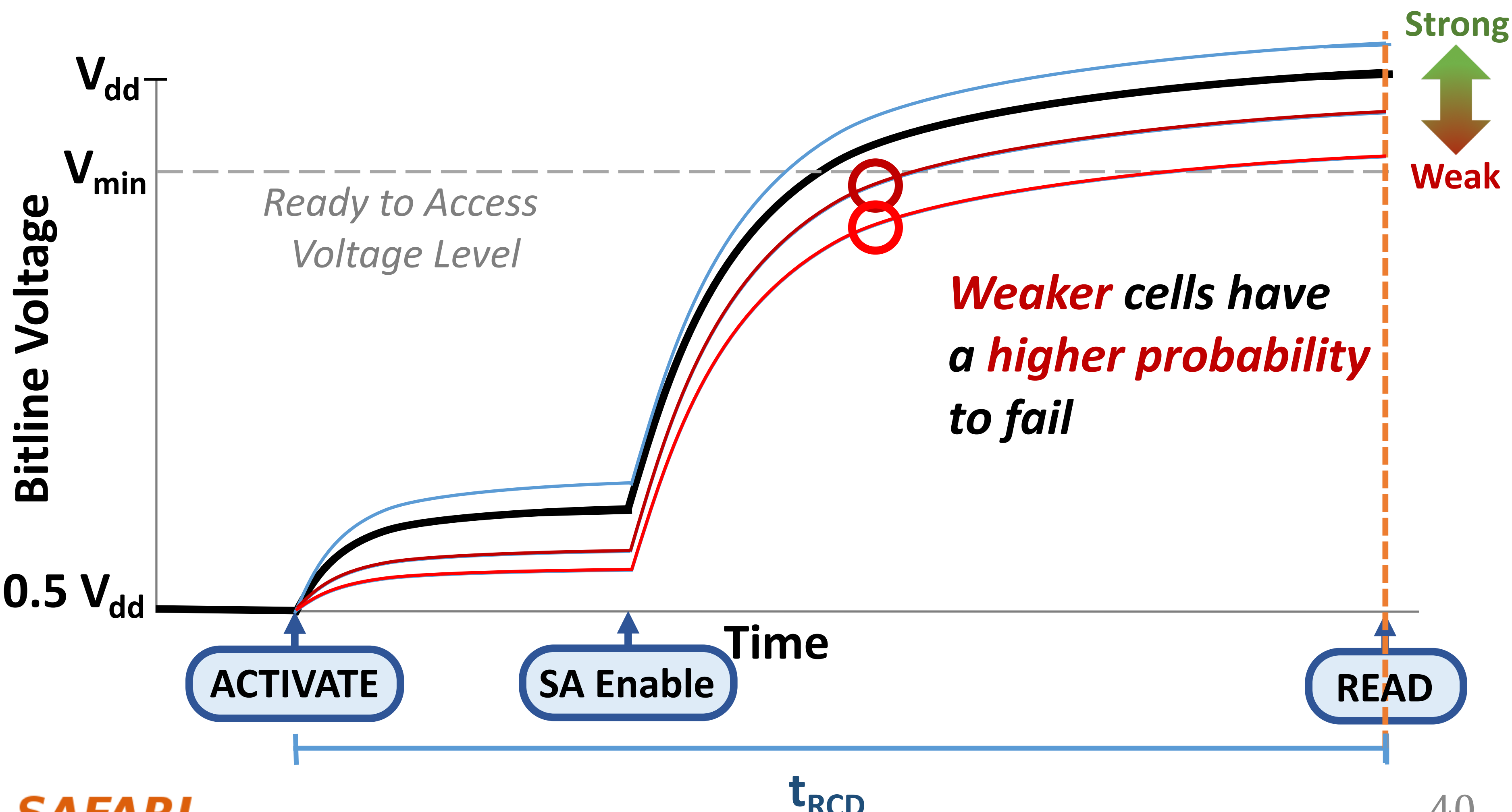
  2. Some cells fail **randomly**

DRAM Accesses and Failures

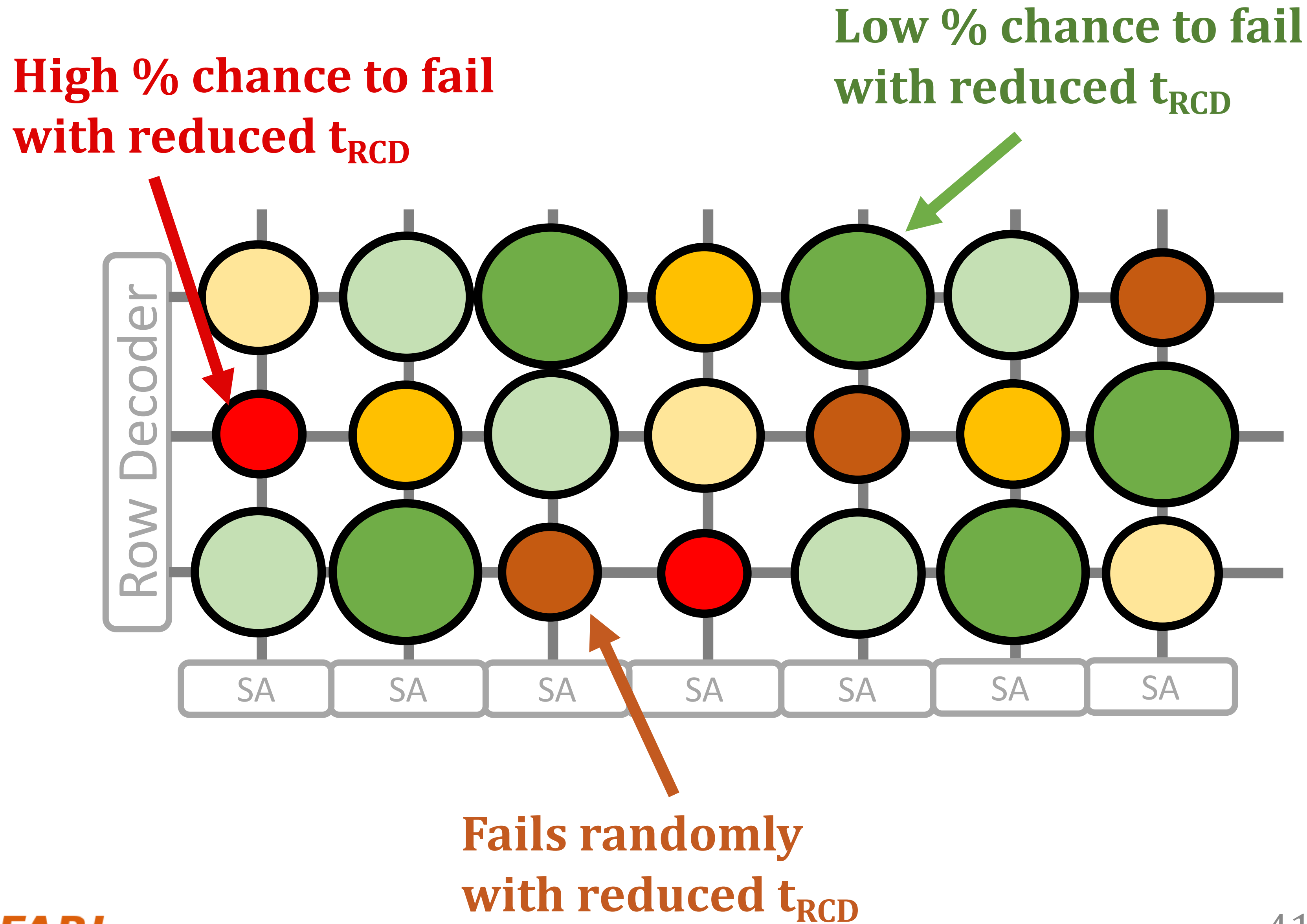


***Process variation* during manufacturing results in cells having *unique* behavior**

DRAM Accesses and Failures



D-RaNGe Key Idea

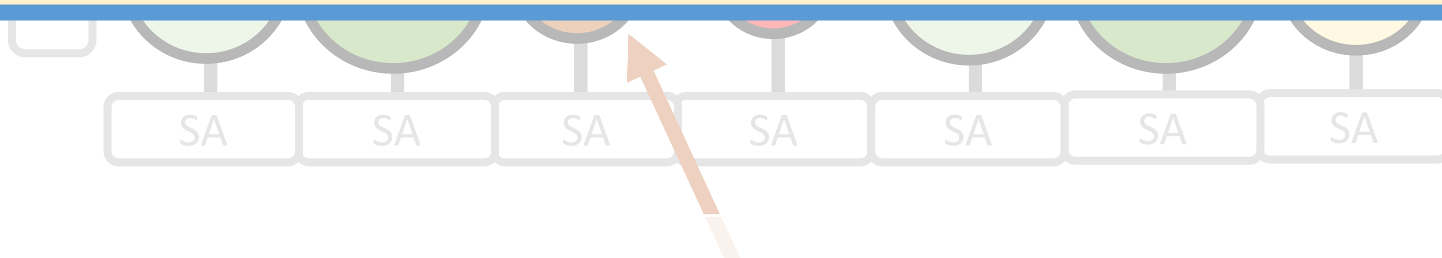


D-RaNGe Key Idea

High % chance to fail with reduced t_{RCD}

Low % chance to fail with reduced t_{RCD}

We refer to cells that fail randomly when accessed with a reduced t_{RCD} as RNG cells



Fails randomly with reduced t_{RCD}

Our D-RaNGe Evaluation

- We generate **random values** by repeatedly accessing **RNG cells** and aggregating the data read

- The random data satisfies the NIST statistical test suite for randomness

- The **D-RaNGE** generates random numbers
  - **Throughput**: 717.4 Mb/s
  - **Latency**: 64 bits in <1us
  - **Power**: 4.4 nJ/bit

D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim **Minesh Patel**
Hasan Hassan **Lois Orosa** **Onur Mutlu**

SAFARI

HPCA 2019

ETH Zürich

Carnegie Mellon

More on D-RaNGe

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, Lois Orosa, and Onur Mutlu, **"D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput"** *Proceedings of the 25th International Symposium on High-Performance Computer Architecture (**HPCA**)*, Washington, DC, USA, February 2019.
  [Slides (pptx) (pdf)]
  [Full Talk Video (21 minutes)]

D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim[‡§] Minesh Patel[§] Hasan Hassan[§] Lois Orosa[§] Onur Mutlu[§‡]
[‡]Carnegie Mellon University [§]ETH Zürich

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**

Systems@ETH Zürich

SAFARI

ETH Zürich

Carnegie Mellon

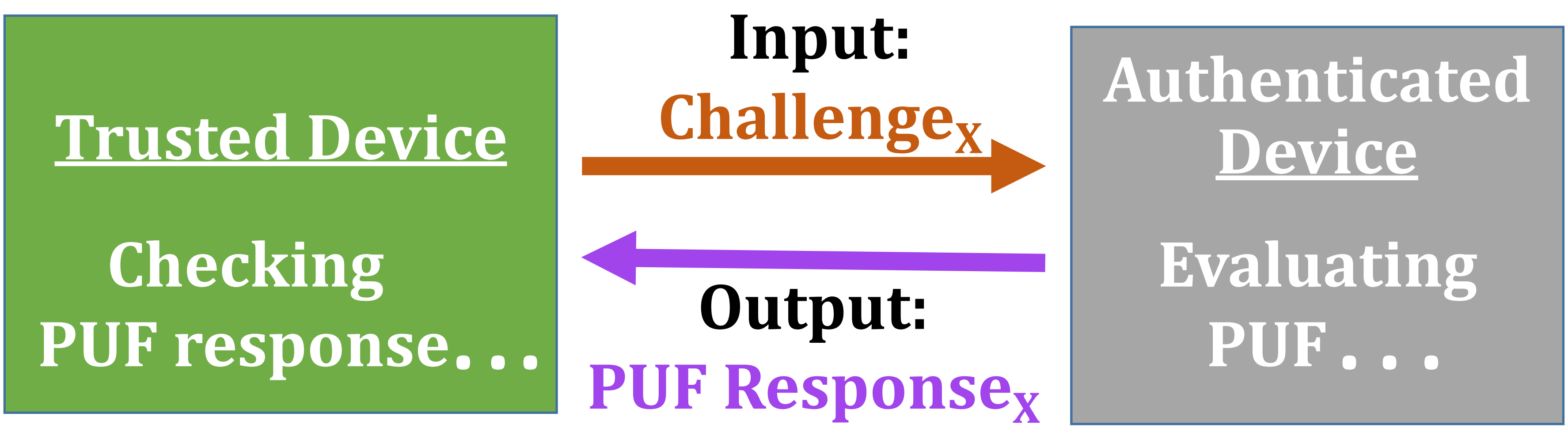
DL-PUF: Executive Summary

- **Motivation**:
  - We can authenticate a system via **unique signatures** if we can evaluate a **Physical Unclonable Function (PUF)** on it
  - Signatures **(PUF response)** reflect inherent properties of a device
  - DRAM is a promising substrate for PUFs because it is **widely** used
- **Problem**: Current DRAM PUFs are 1) very slow, 2) require a DRAM reboot, or 3) require additional custom hardware
- **Goal**: To develop a novel and effective PUF for **existing** commodity DRAM devices with **low-latency evaluation time** and **low system interference** across **all operating temperatures**
- **DRAM Latency PUF:** Reduce DRAM access latency **below reliable values** and exploit the resulting error patterns as **unique identifiers**
- **Evaluation:**
  1. Experimentally characterize **223 real LPDDR4 DRAM devices**
  2. **DRAM latency PUF** (88.2 ms) achieves a speedup of **102x/860x** at 70°C/55°C over prior DRAM PUF evaluation mechanisms

Motivation

We want a way to ensure that a system's components are not **compromised**

- **Physical Unclonable Function (PUF):** a function we **evaluate** on a device to **generate** a **signature unique** to the device
- We refer to the unique signature as a **PUF response**
- Often used in a **Challenge-Response Protocol (CRP)**



Motivation

1. We want a **runtime-accessible** PUF
   - Should be evaluated **quickly** with **minimal** impact on concurrent applications
   - Can protect against **attacks that swap system components with malicious parts**

2. DRAM is a **promising substrate** for evaluating PUFs because it is **ubiquitous** in modern systems
   - Unfortunately, current DRAM PUFs are **slow** and get **exponentially slower** at lower temperatures

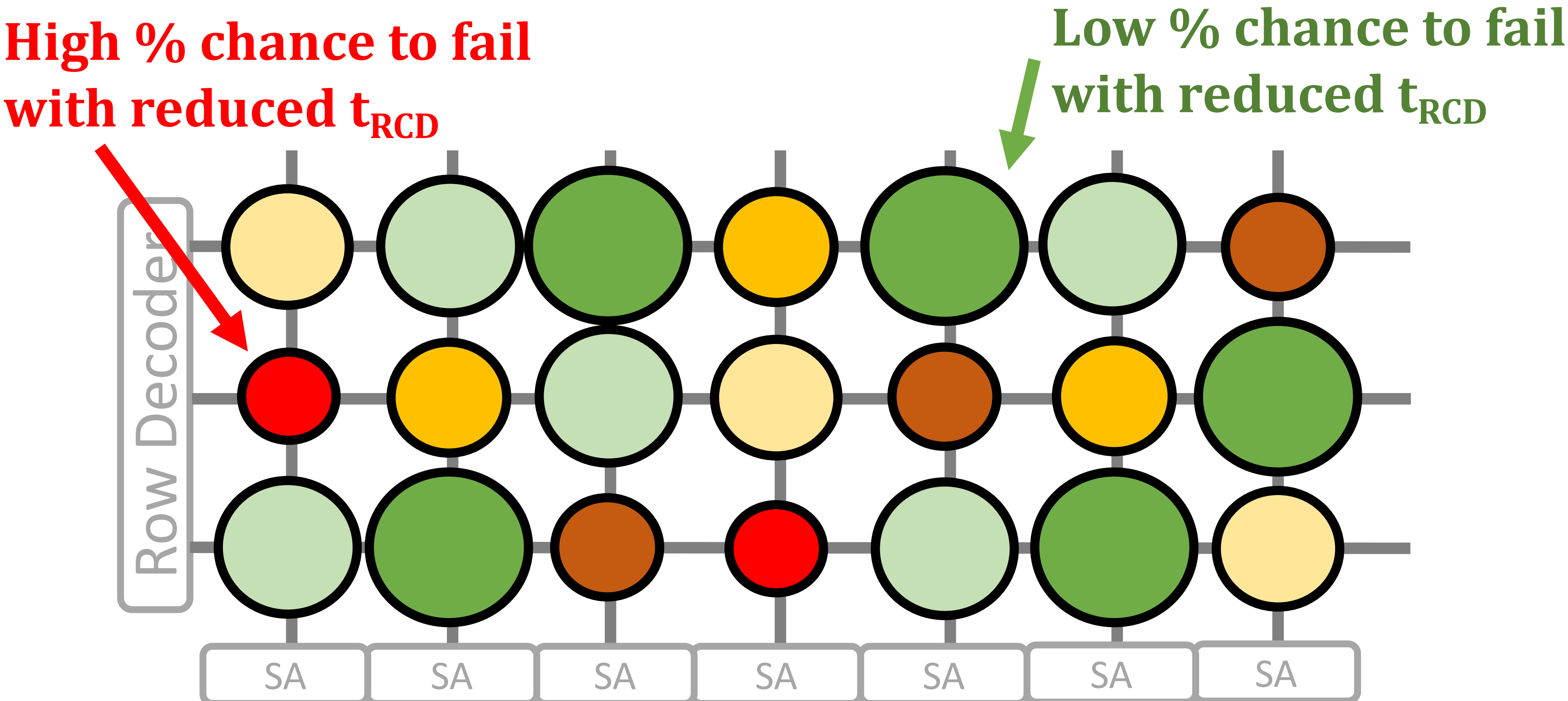
DRAM Latency Characterization of 223 LPDDR4 DRAM Devices

- Latency failures come from accessing DRAM with **reduced** timing parameters.

- **Key Observations:**
  1. A cell's **latency failure** probability is determined by **random process variation**
  2. Latency failure patterns are **repeatable and unique to a device**

DRAM Latency PUF Key Idea

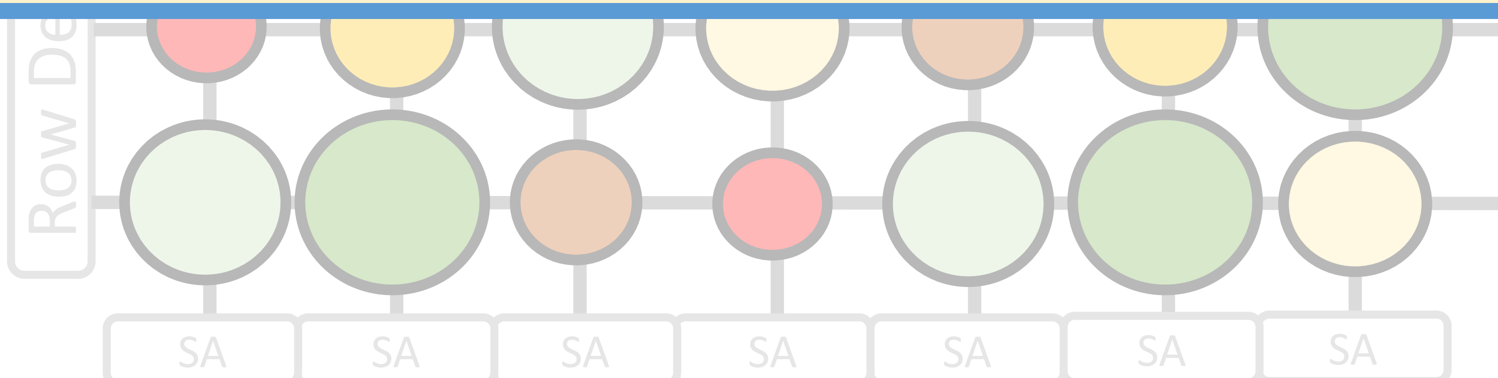
- A cell's latency failure probability is inherently related to **random process variation** from manufacturing
- We can provide **repeatable and unique device signatures** using latency error patterns



DRAM Latency PUF Key Idea

- A cell's latency failure probability is inherently related to **random process variation** from manufacturing
- We can provide **repeatable and unique device**

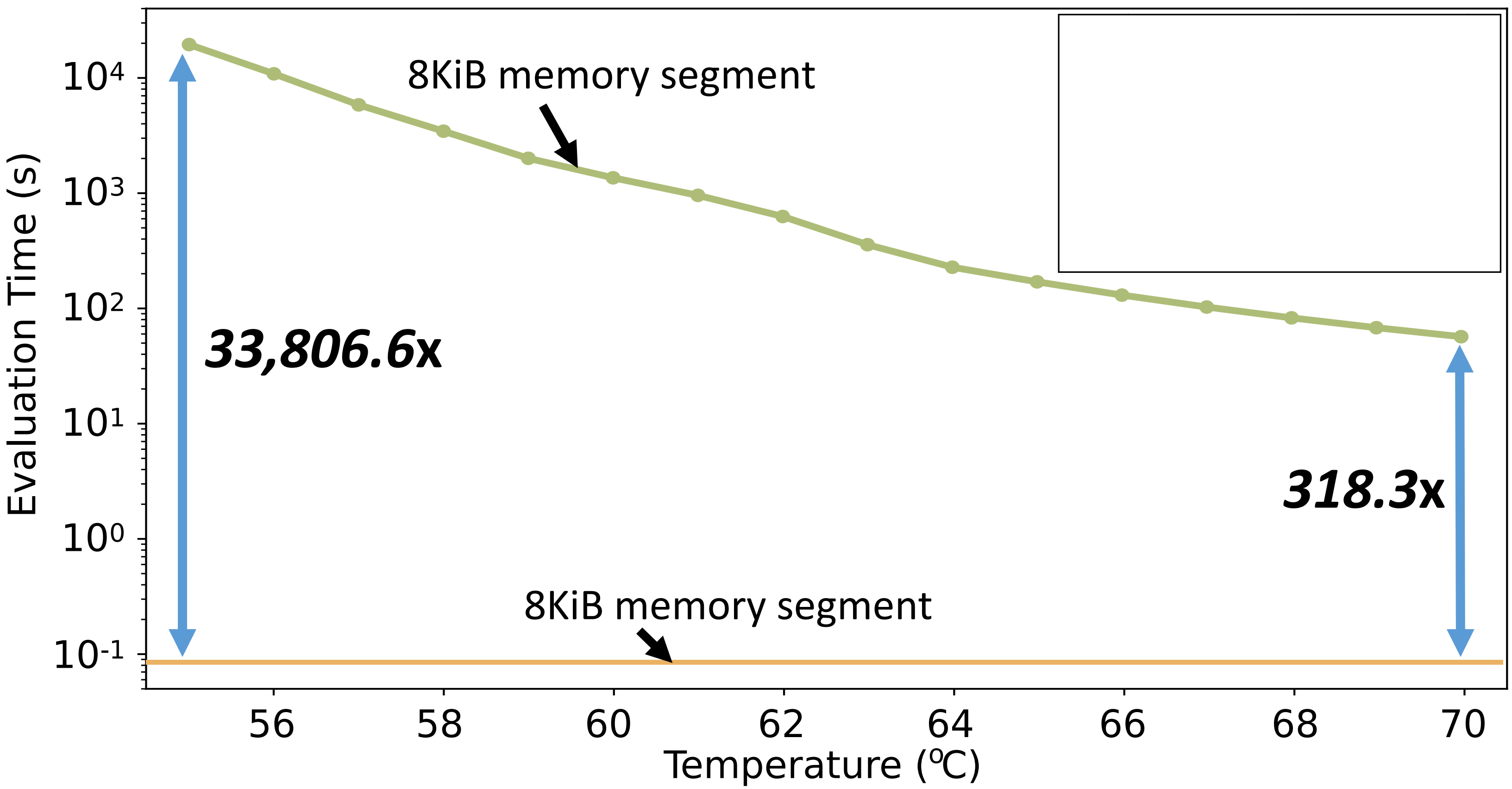
The key idea is to compose a PUF response using the DRAM cells that fail with high probability



The DRAM Latency PUF Evaluation

- We generate PUF responses using **latency errors** in a region of DRAM
- The latency error patterns **satisfy PUF requirements**
- The DRAM Latency PUF **generates PUF responses in 88.2ms**

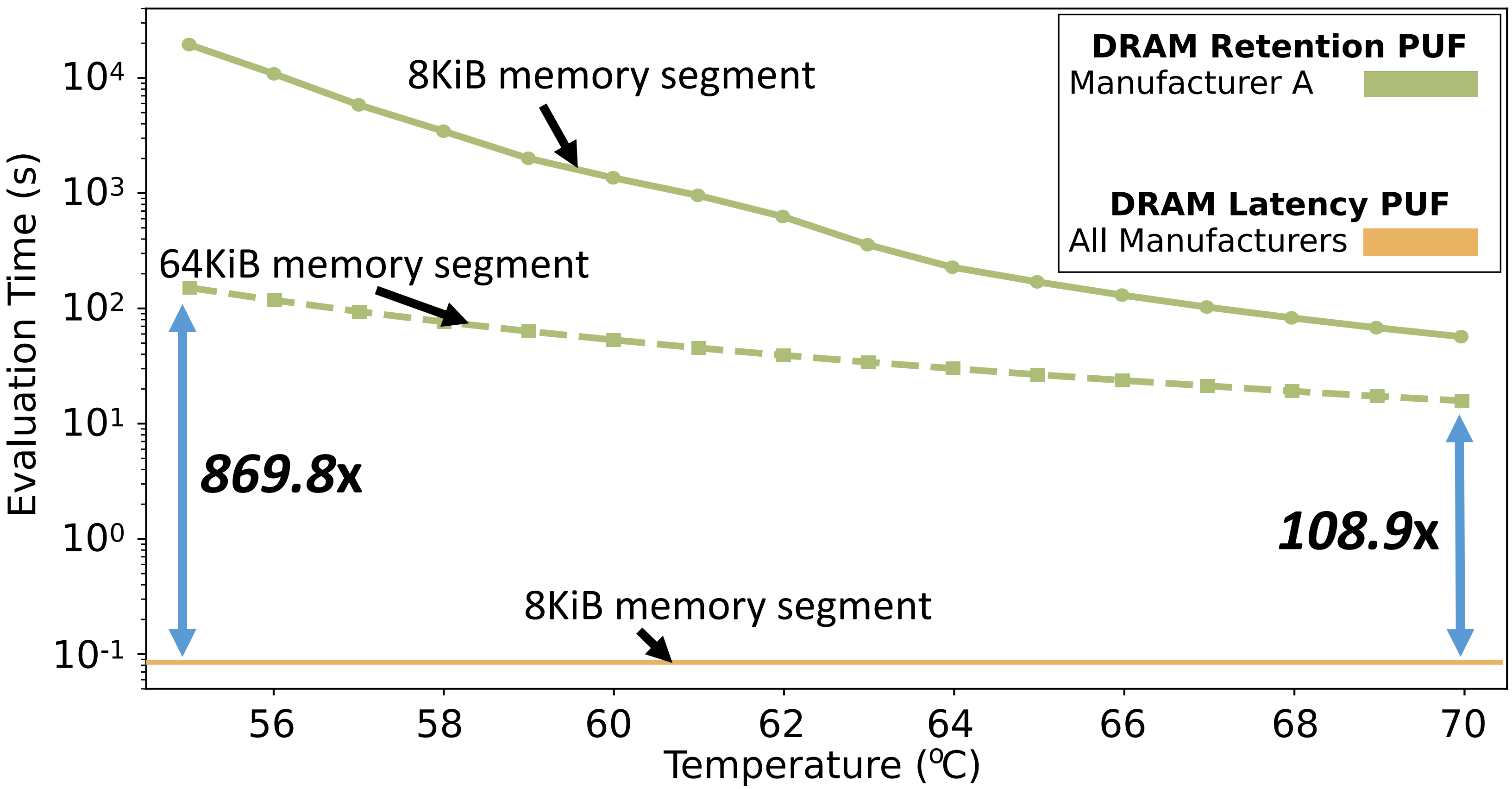
Results – PUF Evaluation Latency



DRAM latency PUF is

1. Fast and constant latency (88.2ms)

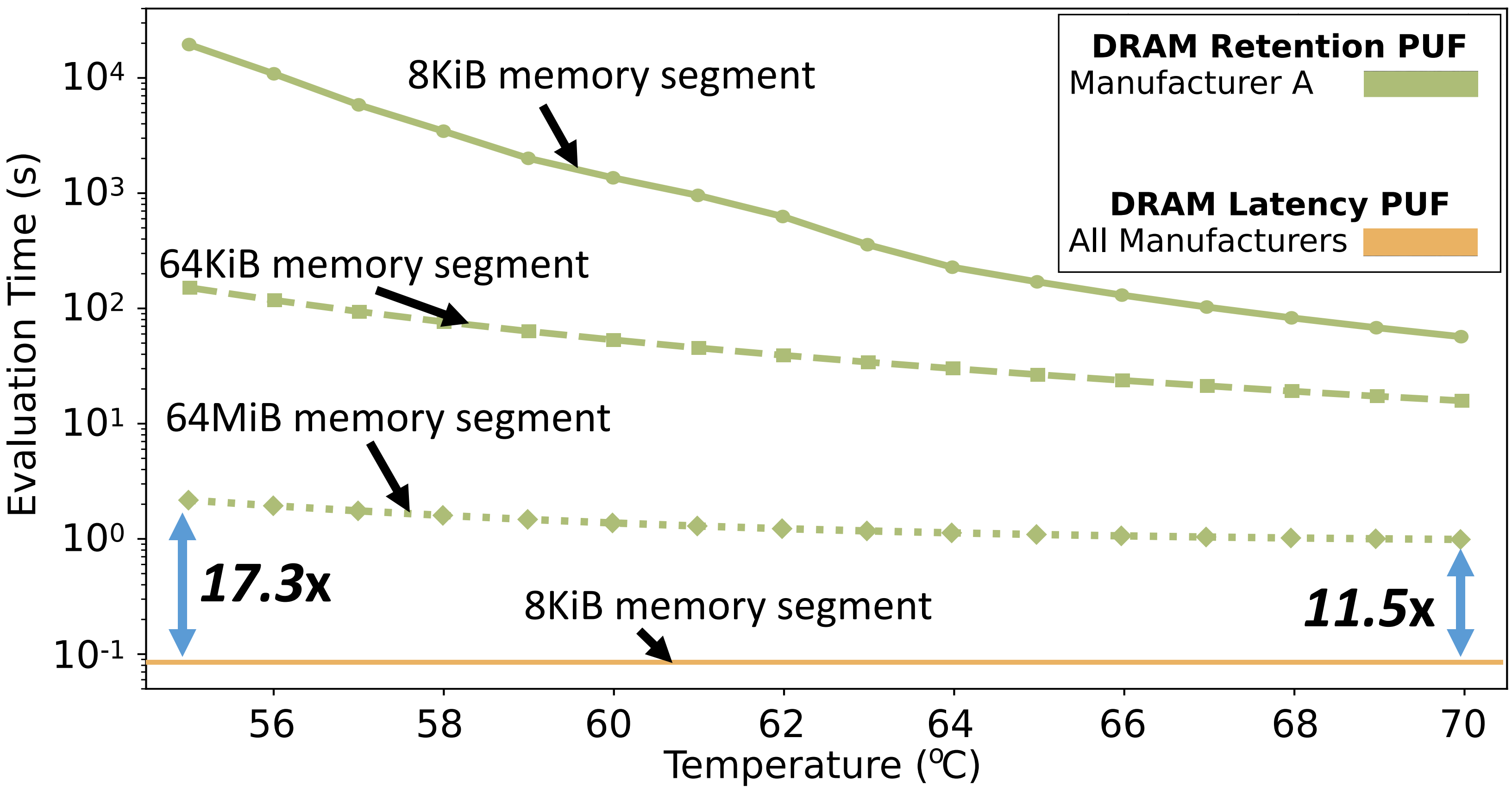
Results – PUF Evaluation Latency



DRAM latency PUF is

1. **Fast and constant latency (88.2ms)**

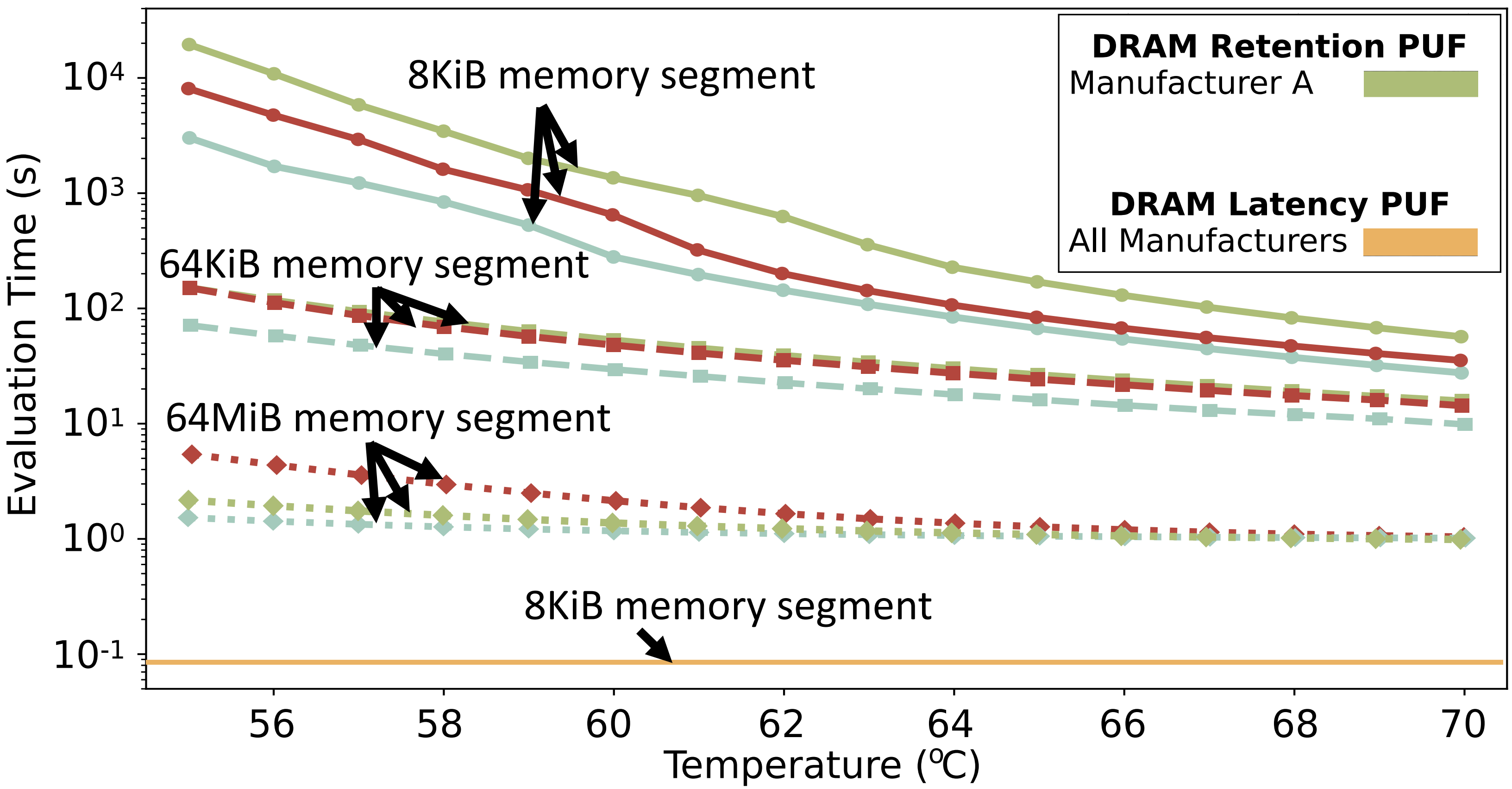
Results – PUF Evaluation Latency



DRAM latency PUF is

1. **Fast and constant latency (88.2ms)**

Results – PUF Evaluation Latency



DRAM latency PUF is

1. Fast and constant latency (88.2ms)
2. On average, 102x/860x faster than the previous DRAM PUF with the same DRAM capacity overhead (64KiB)

Other Results in the Paper

- **How the DRAM latency PUF meets the basic requirements for an effective PUF**
- **A detailed analysis on:**
  - Devices of **the three major DRAM manufacturers**
  - The **evaluation time** of a PUF
- **Further discussion on:**
  - **Optimizing** retention PUFs
  - **System interference** of DRAM retention and latency PUFs
  - Algorithm to **quickly and reliably** evaluate DRAM latency PUF
  - **Design considerations** for a DRAM latency PUF
  - The DRAM Latency PUF overhead analysis

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

Systems@ETH Zürich

ETH Zürich

Carnegie Mellon

DRAM Latency PUFs

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu,
  "The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices"
  *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)]

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[§†]
[†]Carnegie Mellon University [§]ETH Zürich

Reducing Refresh Latency

On Reducing Refresh Latency

- Anup Das, Hasan Hassan, and Onur Mutlu,
"VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency"
Proceedings of the 55th Design Automation Conference (***DAC***), San Francisco, CA, USA, June 2018.
[Slides (pdf)] [Poster (pdf)]

VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency

Anup Das
Drexel University
Philadelphia, PA, USA
anup.das@drexel.edu

Hasan Hassan
ETH Zürich
Zürich, Switzerland
hhasan@ethz.ch

Onur Mutlu
ETH Zürich
Zürich, Switzerland
omutlu@gmail.com

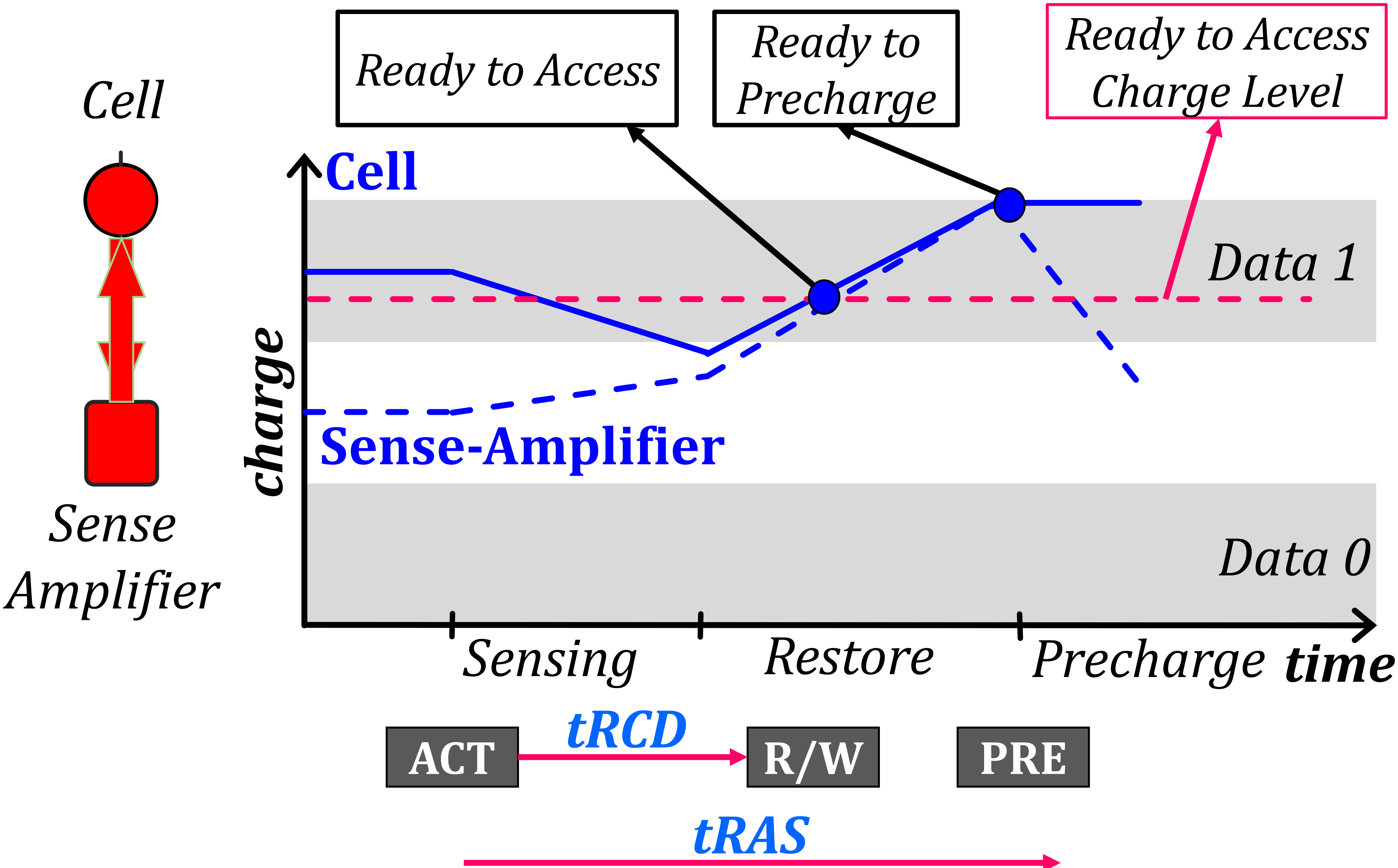
Reducing Memory Latency by Exploiting Memory Access Patterns

ChargeCache: Executive Summary

- **Goal**: Reduce average DRAM access latency with no modification to the existing DRAM chips
- **Observations**:
  1) A highly-charged DRAM row can be accessed with low latency
  2) A row's charge is restored when the row is accessed
  3) A recently-accessed row is likely to be accessed again**:**

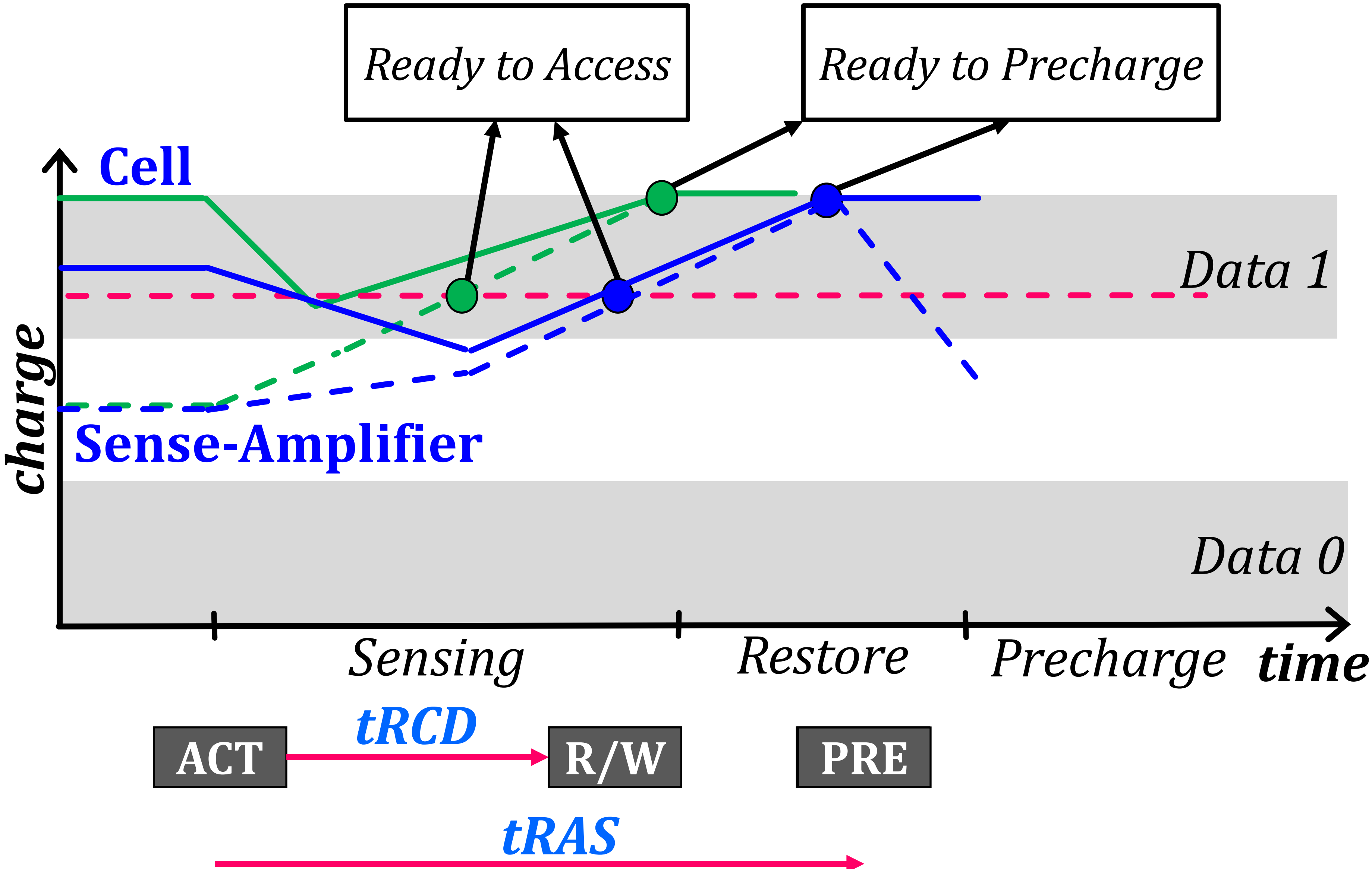
     Row Level Temporal Locality (RLTL)
- **Key Idea**: Track recently-accessed DRAM rows and use lower timing parameters if such rows are accessed again
- **ChargeCache:**
  - Low cost & no modifications to the DRAM
  - Higher performance (**8.6-10.6%** on average for 8-core)
  - Lower DRAM energy (**7.9%** on average)

DRAM Charge over Time



SAFARI

Accessing Highly-charged Rows



Observation 1

A highly-charged DRAM row can be accessed with low latency

- tRCD: **44%**
- tRAS: **37%**

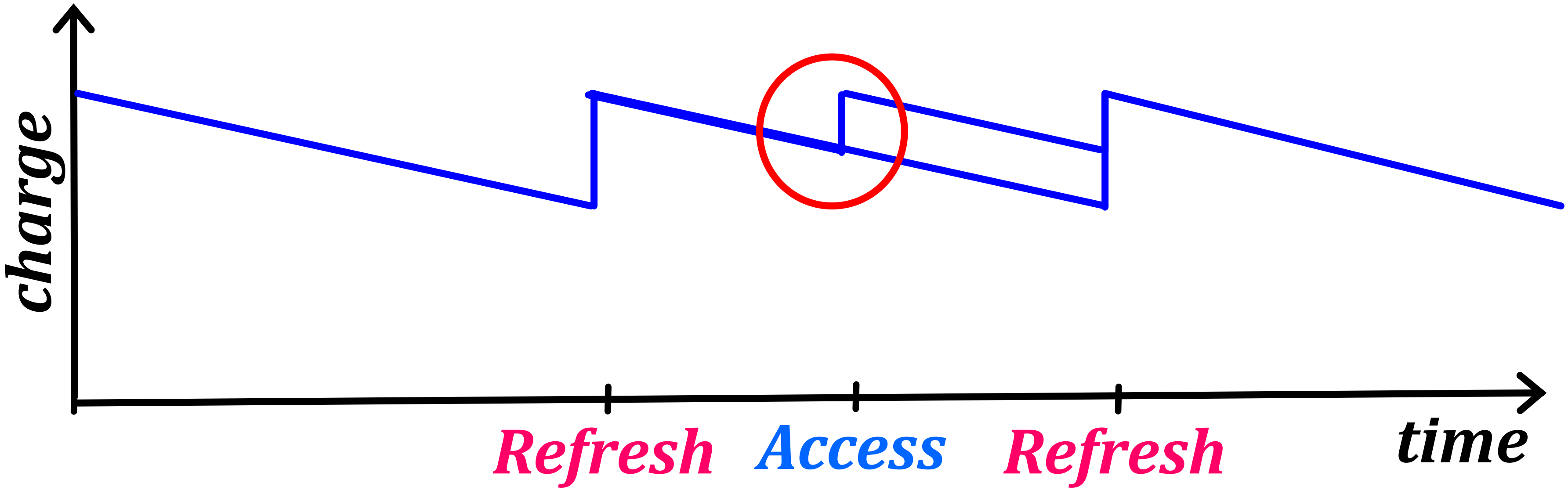
How does a row become highly-charged?

How Does a Row Become Highly-Charged?

DRAM cells **lose charge** over time

Two ways of restoring a row's charge:

- Refresh Operation
- Access



Observation 2

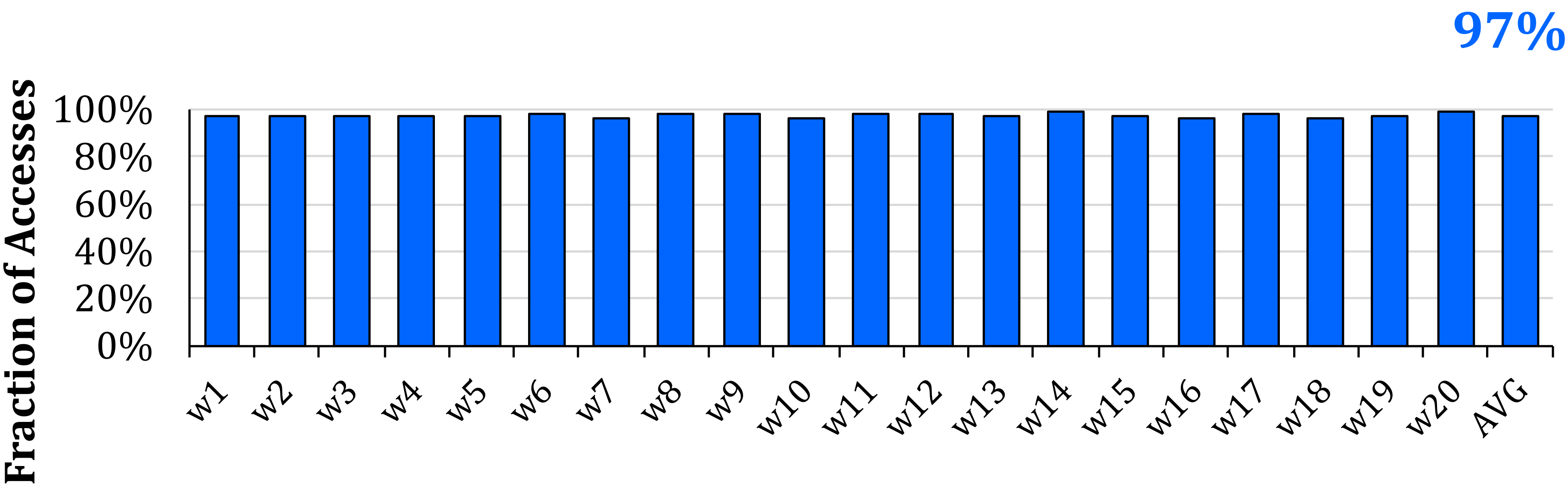
A row's charge is restored when the row is accessed

How likely is a recently-accessed row to be accessed again?

Row Level Temporal Locality (RLTL)

A **recently-accessed** DRAM row is likely to be accessed again.

- ***t***-RLTL: Fraction of rows that are accessed within time ***t*** after their previous access



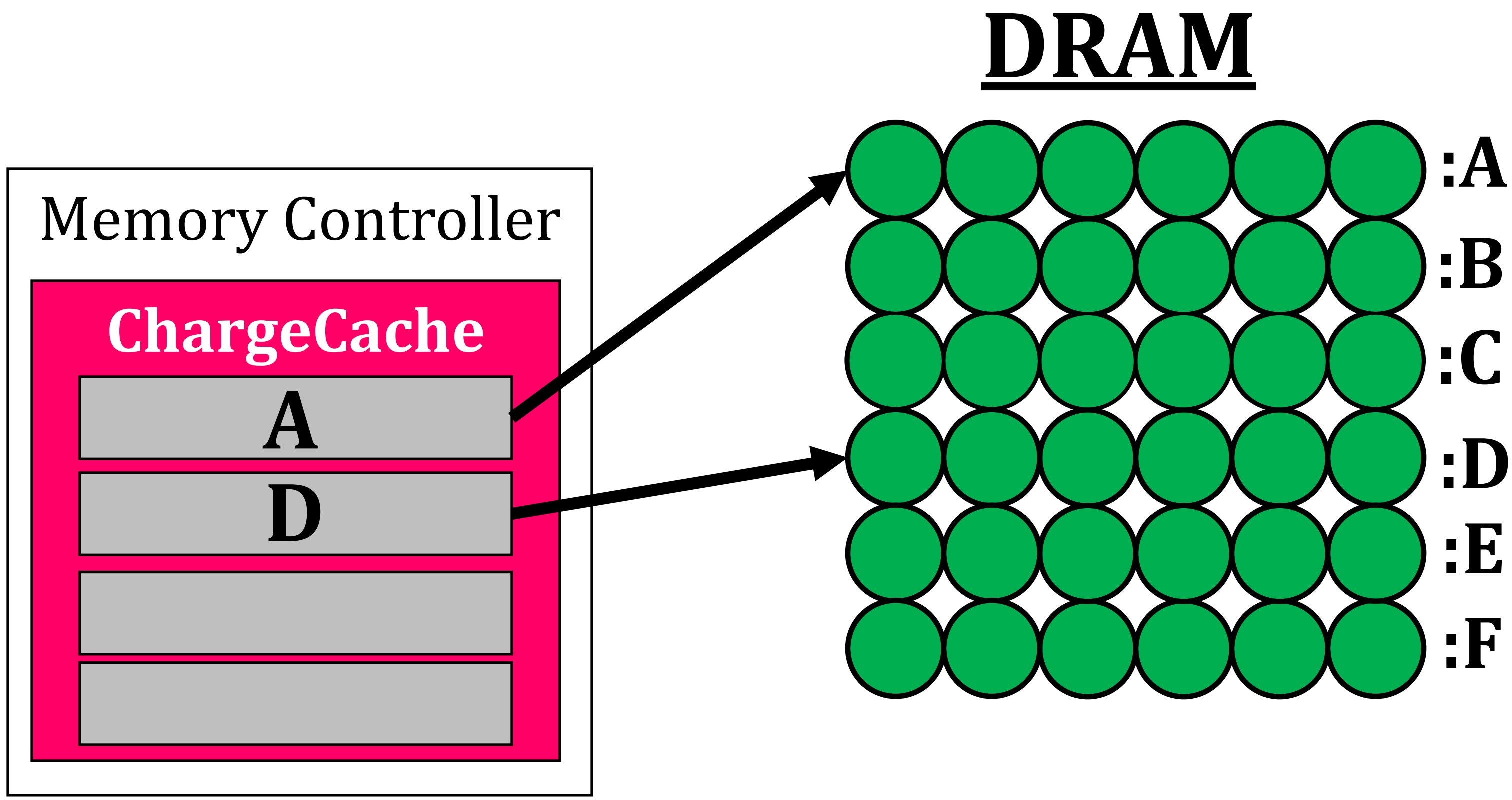
8ms–RLTL for single-core workloads

8ms–RLTL for eight-core workloads

Key Idea

Track **recently-accessed** DRAM rows and use **lower timing parameters** if such rows are accessed again

ChargeCache Overview



Requests: A D A

ChargeCache Miss: Use Default Timings

ChargeCache Hit: Use Lower Timings

Area and Power Overhead

- **Modeled with CACTI**

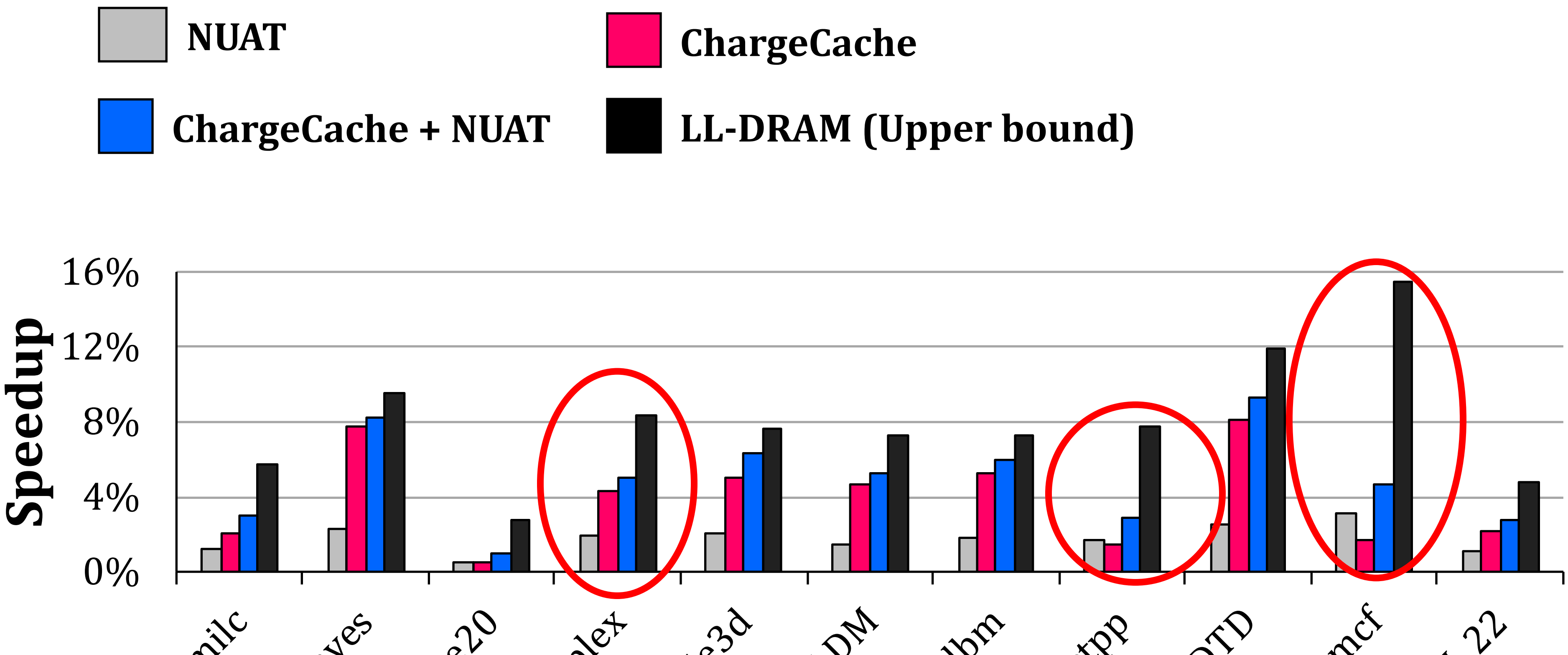
- **Area**
  - ~5KB for 128-entry ChargeCache
  - 0.24% of a 4MB Last Level Cache (LLC) area

- **Power Consumption**
  - 0.15 mW on average (static + dynamic)
  - 0.23% of the 4MB LLC power consumption

Methodology

- **Simulator**
  - DRAM Simulator (Ramulator *[Kim+, CAL'15]*)
    https://github.com/CMU-SAFARI/ramulator
- **Workloads**
  - 22 single-core workloads
    - SPEC CPU2006, TPC, STREAM
  - 20 multi-programmed 8-core workloads
    - By randomly choosing from single-core workloads
  - Execute at least 1 billion representative instructions per core (Pinpoints)
- **System Parameters**
  - 1/8 core system with 4MB LLC
  - Default tRCD/tRAS of 11/28 cycles

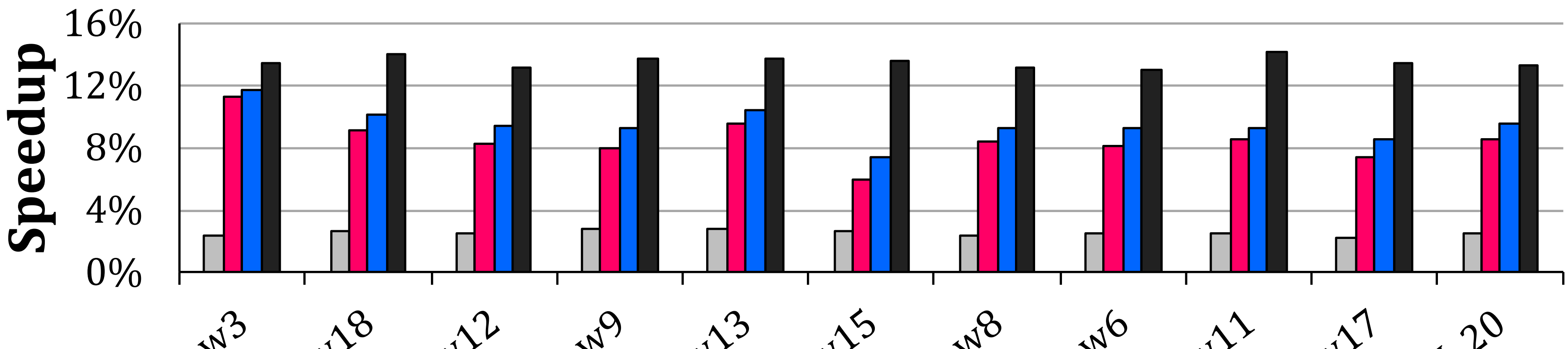
Single-core Performance



ChargeCache improves single-core performance

Eight-core Performance

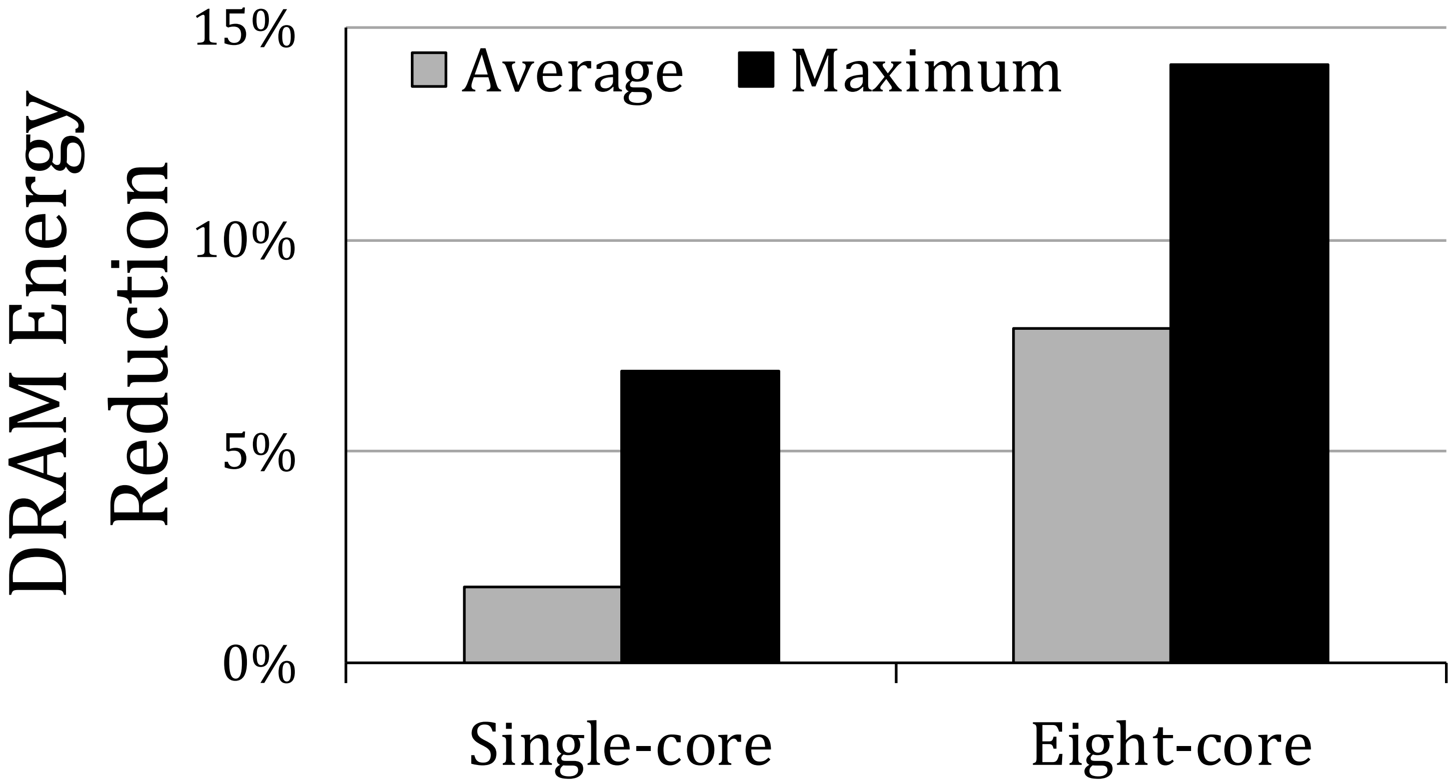
NUAT **2.5%**
ChargeCache **9%**
ChargeCache + NUAT
LL-DRAM (Upperbound) **13%**



ChargeCache significantly improves multi-core performance

SAFARI

DRAM Energy Savings



ChargeCache reduces DRAM energy

More on ChargeCache

- Hasan Hassan, Gennady Pekhimenko, Nandita Vijaykumar, Vivek Seshadri, Donghyuk Lee, Oguz Ergin, and Onur Mutlu,
  "ChargeCache: Reducing DRAM Latency by Exploiting Row Access Locality"
  *Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (**HPCA**)*, Barcelona, Spain, March 2016.
  [Slides (pptx) (pdf)]
  [Source Code]

ChargeCache: Reducing DRAM Latency by Exploiting Row Access Locality

Hasan Hassan[†*], Gennady Pekhimenko[†], Nandita Vijaykumar[†]
Vivek Seshadri[†], Donghyuk Lee[†], Oguz Ergin[*], Onur Mutlu[†]

[†] *Carnegie Mellon University* [*] *TOBB University of Economics & Technology*

A Very Recent Work

- Yaohua Wang, Arash Tavakkol, Lois Orosa, Saugata Ghose, Nika Mansouri Ghiasi, Minesh Patel, Jeremie S. Kim, Hasan Hassan, Mohammad Sadrosadati, and Onur Mutlu,
"Reducing DRAM Latency via Charge-Level-Aware Look-Ahead Partial Restoration"
Proceedings of the 51st International Symposium on Microarchitecture (***MICRO***), Fukuoka, Japan, October 2018.

Reducing DRAM Latency
via Charge-Level-Aware Look-Ahead Partial Restoration

Yaohua Wang[†§] Arash Tavakkol[†] Lois Orosa[†*] Saugata Ghose[‡] Nika Mansouri Ghiasi[†]
Minesh Patel[†] Jeremie S. Kim[‡†] Hasan Hassan[†] Mohammad Sadrosadati[†] Onur Mutlu[†‡]

[†]*ETH Zürich* [§]*National University of Defense Technology*
[‡]*Carnegie Mellon University* [*]*University of Campinas*

Summary: Low-Latency Memory

Summary: Tackling Long Memory Latency

- Reason 1: Design of DRAM Micro-architecture
  - Goal: Maximize capacity/area, not minimize latency

- Reason 2: "One size fits all" approach to latency specification
  - Same latency parameters for all temperatures
  - Same latency parameters for all DRAM chips (e.g., rows)
  - Same latency parameters for all parts of a DRAM chip
  - Same latency parameters for all supply voltage levels
  - Same latency parameters for all application data
  - …

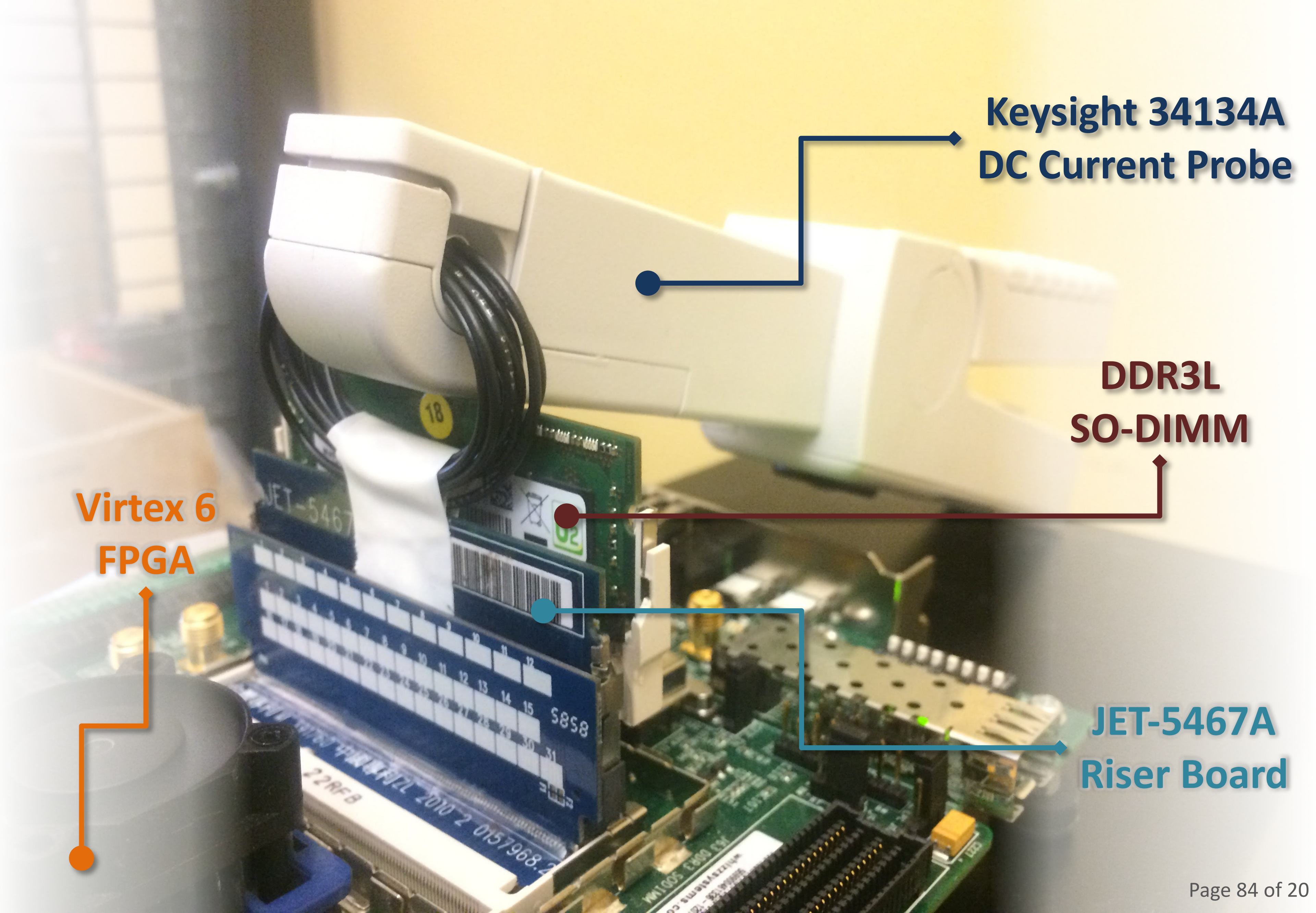
Challenge and Opportunity for Future

Fundamentally
Low Latency
Computing Architectures

On DRAM Power Consumption

Power Measurement Platform

SAFARI



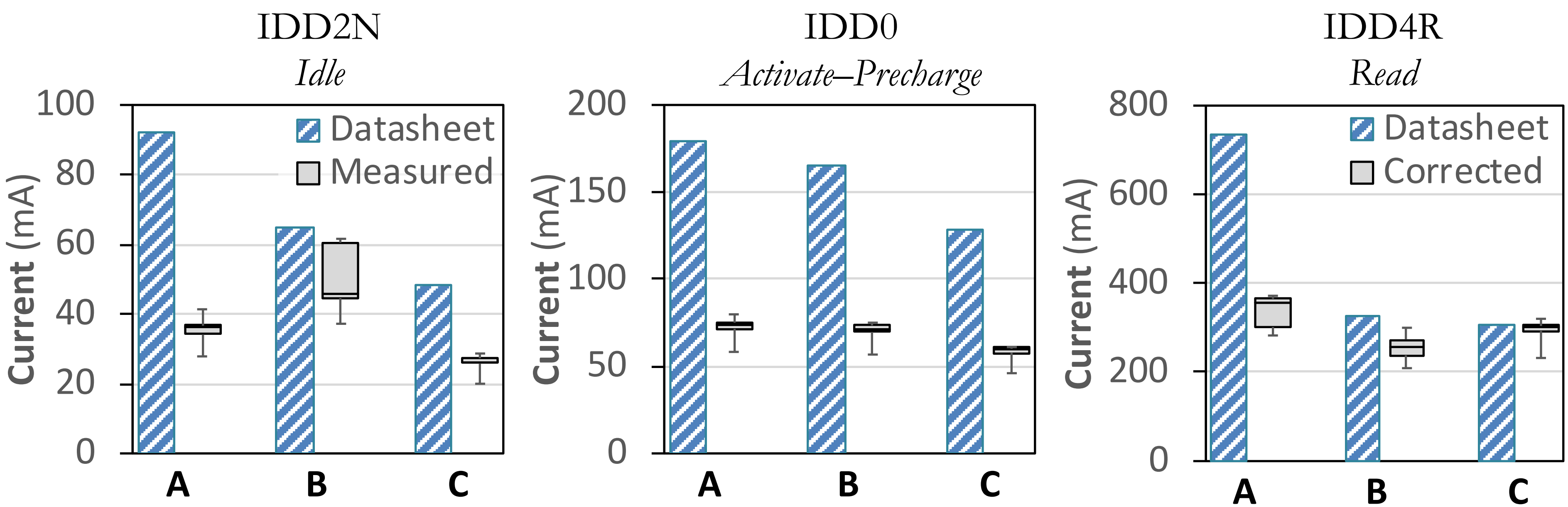
Power Measurement Methodology

SAFARI

- **SoftMC: an FPGA-based memory controller** [Hassan+ HPCA '17]
  - Modified to repeatedly loop commands
  - Open-source: https://github.com/CMU-SAFARI/SoftMC
- **Measure current consumed by a module during a SoftMC test**
- **Tested 50 DDR3L DRAM modules** (200 DRAM chips)
  - Supply voltage: 1.35 V
  - **Three major vendors: A, B, C**
  - Manufactured between 2014 and 2016
- **For each experimental test that we perform**
  - 10 runs of each test per module
  - At least 10 current samples per run

1. Real DRAM Power Varies Widely from IDD Values

SAFARI

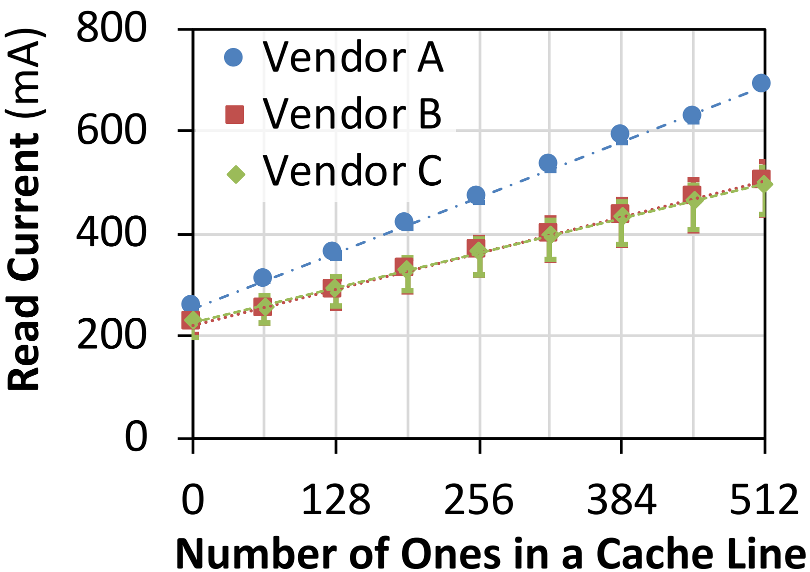


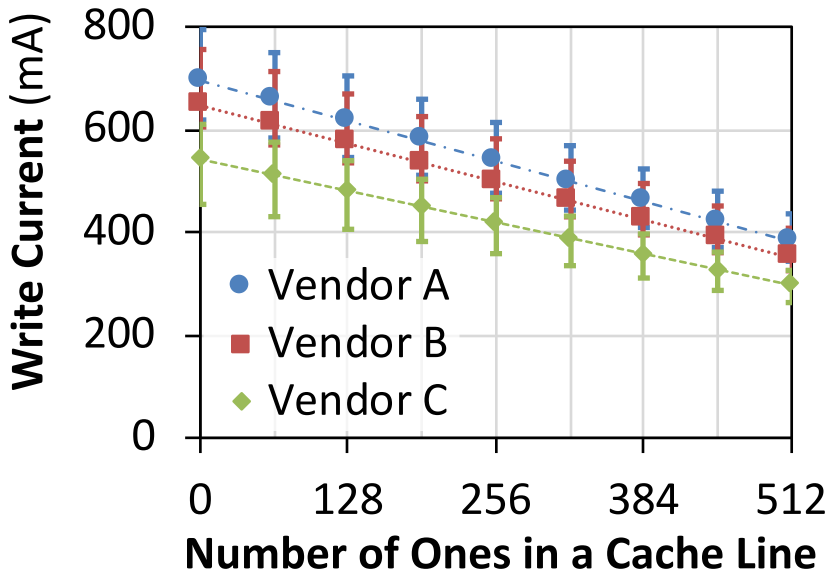
- **Different vendors have very different margins (i.e., *guardbands*)**
- **Low variance among different modules from same vendor**

Current consumed by real DRAM modules varies significantly for all IDD values that we measure

2. DRAM Power is Dependent on Data Values

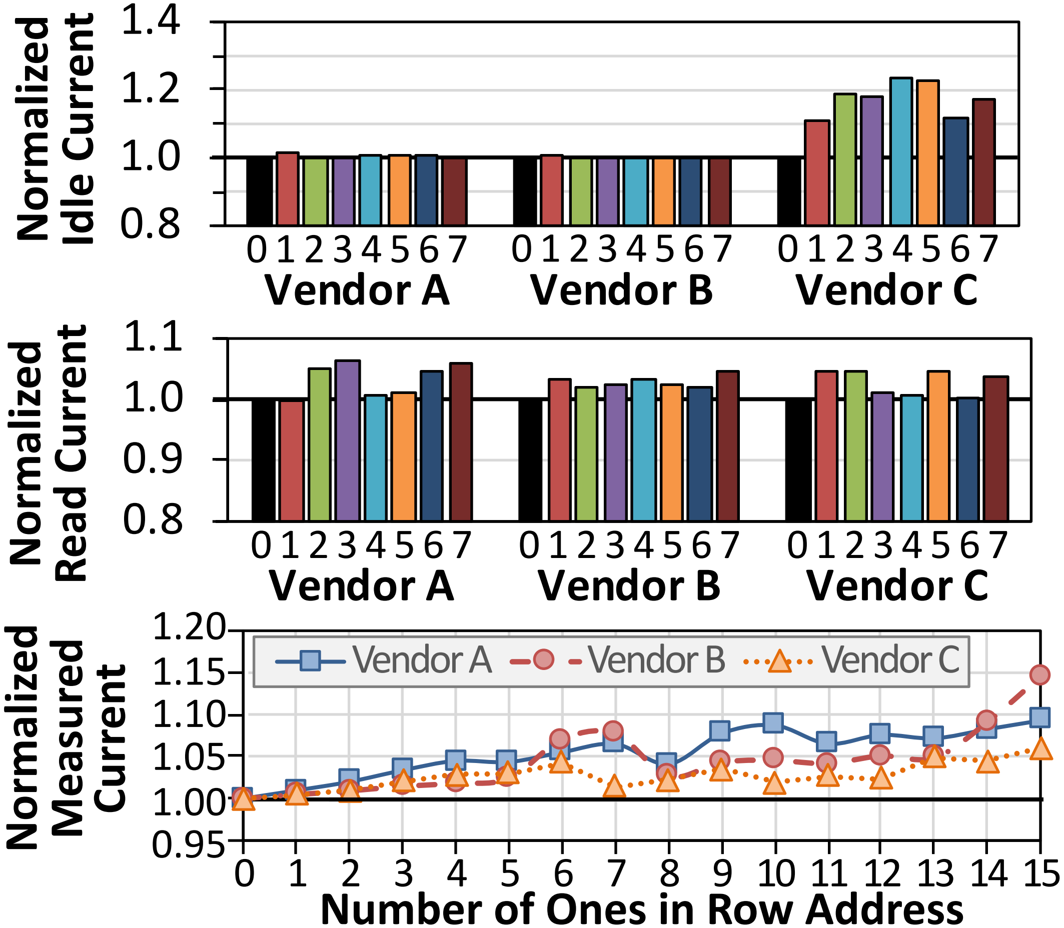
SAFARI





- Some variation due to infrastructure – can be subtracted
- Without infrastructure variation: up to 230 mA of change
- Toggle affects power consumption, but < 0.15 mA per bit

DRAM power consumption depends *strongly* on the data value

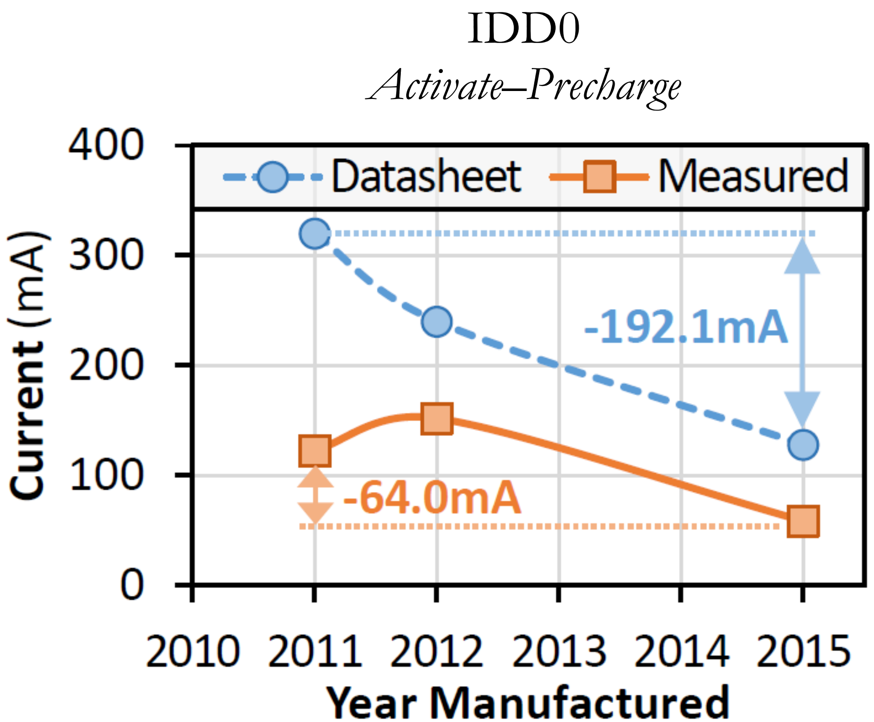


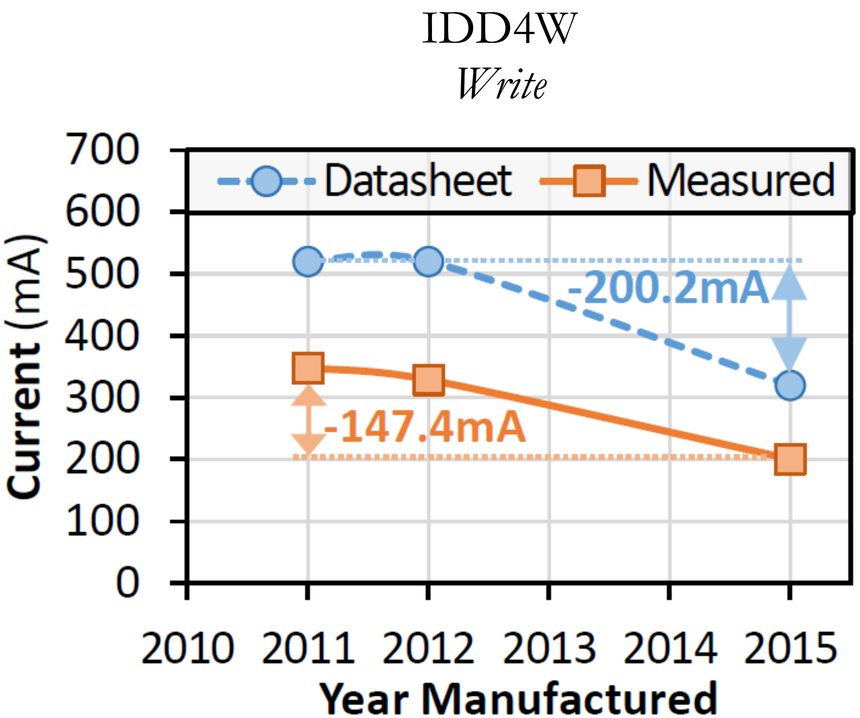
- Vendor C: variation in idle current across banks
- All vendors: variation in read current across banks
- All vendors: variation in activation based on

Significant structural variation:
DRAM power varies systematically by bank and row

4. Generational Savings Are Smaller Than Expected

SAFARI





- Similar trends for idle and read currents

Actual power savings of newer DRAM is *much lower* than the savings indicated in the datasheets

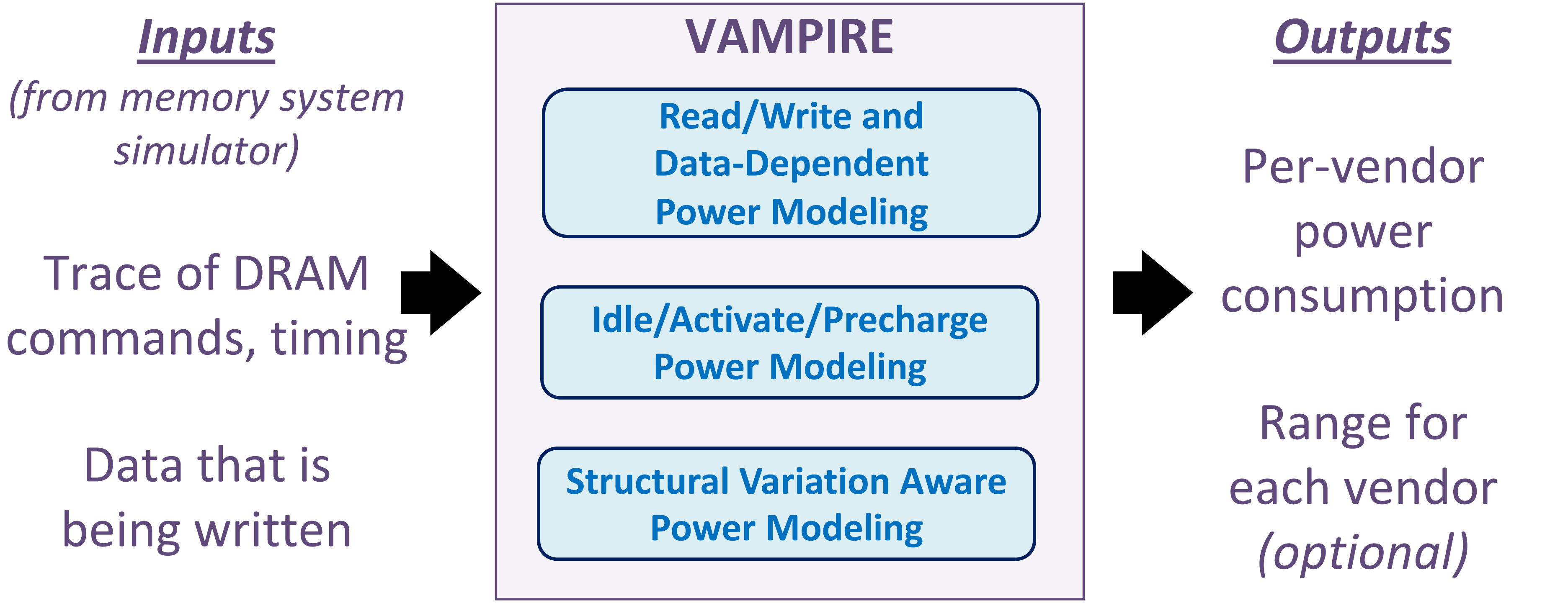
Summary of New Observations on DRAM Power

SAFARI

1. Real DRAM modules often consume less power than vendor-provided IDD values state
2. DRAM power consumption is dependent on the data value that is read/written
3. Across banks and rows, structural variation affects power consumption of DRAM
4. Newer DRAM modules save less power than indicated in datasheets by vendors

Detailed observations and analyses in the paper

- VAMPIRE: Variation-Aware model of Memory Power Informed by Real Experiments

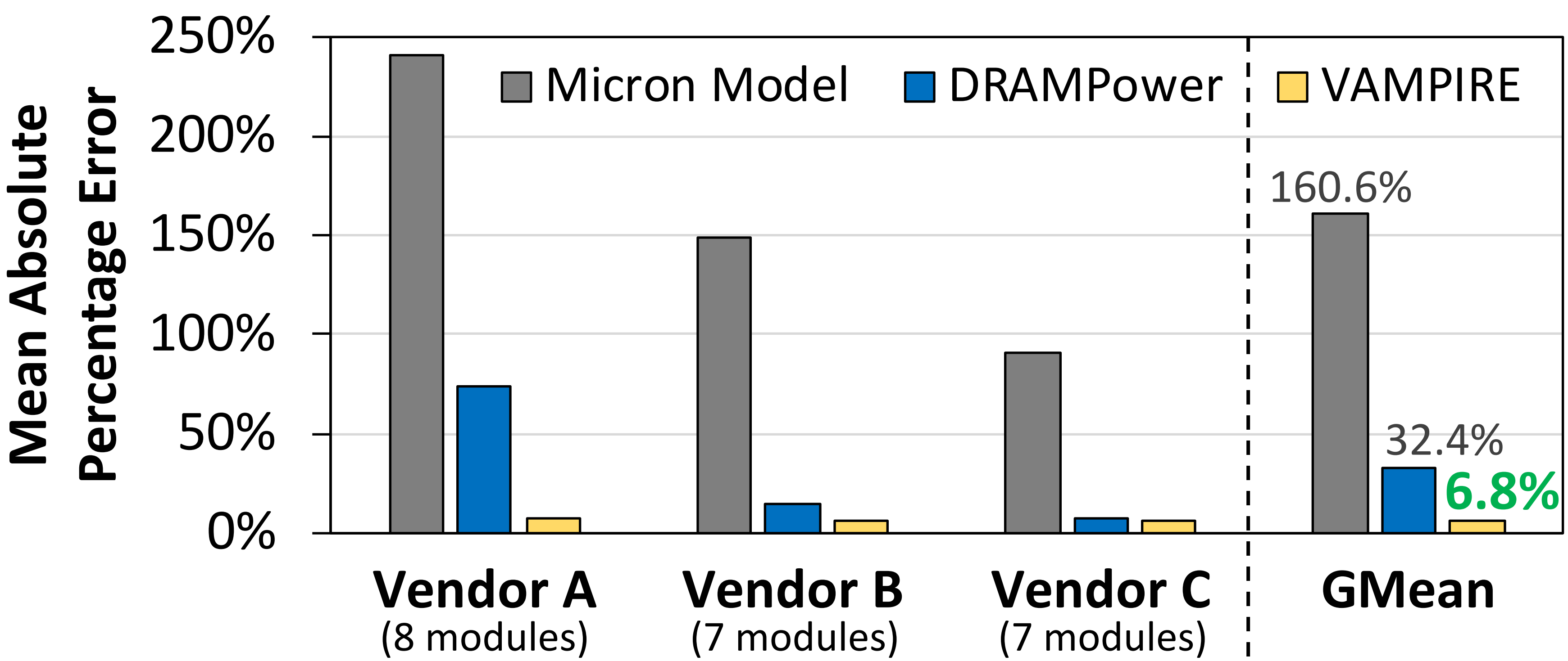


- VAMPIRE and raw characterization data are open-source: https://github.com/CMU-SAFARI/VAMPIRE

VAMPIRE Has Lower Error Than Existing Models

SAFARI

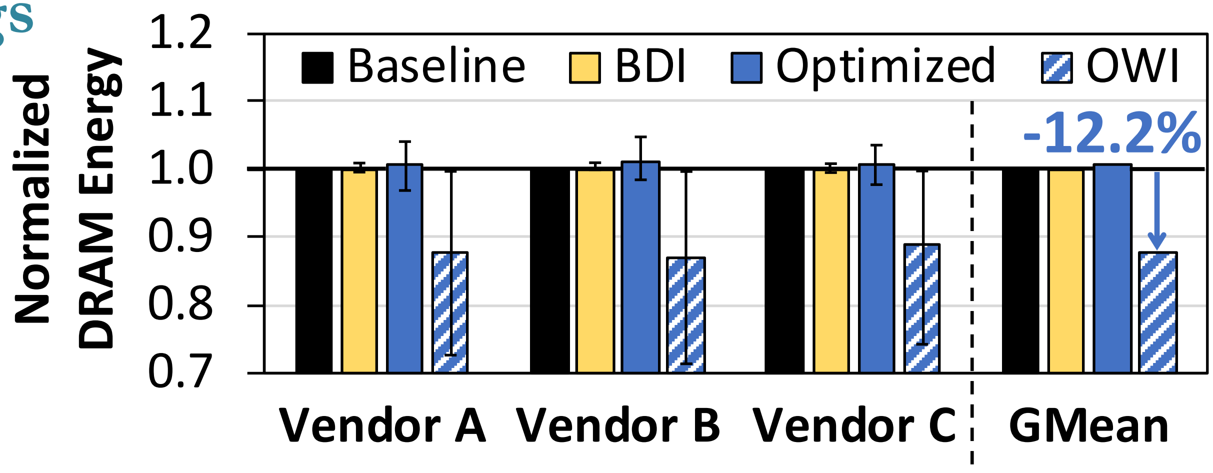
- Validated using new power measurements: details in the



VAMPIRE has very low error for *all* vendors: 6.8%
Much more accurate than prior models

VAMPIRE Enables Several New Studies

- Taking advantage of structural variation to perform variation-aware physical page allocation to reduce power
- Smarter DRAM power-down scheduling
- Reducing DRAM energy with data-dependency-aware cache line encodings
  - 23 applications from the SPEC 2006 benchmark suite
  - Traces collected using Pin and Ramulator



- We expect there to be many other new studies in the future

VAMPIRE DRAM Power Model

- Saugata Ghose, A. Giray Yaglikci, Raghav Gupta, Donghyuk Lee, Kais Kudrolli, William X. Liu, Hasan Hassan, Kevin K. Chang, Niladrish Chatterjee, Aditya Agrawal, Mike O'Connor, and Onur Mutlu,
  "What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study"
  *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)]
  [VAMPIRE DRAM Power Model]

What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study

Saugata Ghose[†] Abdullah Giray Yağlıkçı[‡†] Raghav Gupta[†] Donghyuk Lee[§]
Kais Kudrolli[†] William X. Liu[†] Hasan Hassan[‡] Kevin K. Chang[†]
Niladrish Chatterjee[§] Aditya Agrawal[§] Mike O'Connor[§¶] Onur Mutlu[‡†]

[†]Carnegie Mellon University [‡]ETH Zürich [§]NVIDIA [¶]University of Texas at Austin

Conclusion

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

Some Solution Principles (So Far)

- Data-centric system design & intelligence spread around
  - Do not center everything around traditional computation units

- Better cooperation across layers of the system
  - Careful co-design of components and layers: system/arch/device
  - Better, richer, more expressive and flexible interfaces

- Better-than-worst-case design
  - Do not optimize for the worst case
  - Worst case should not determine the common case

- Heterogeneity in design (specialization, asymmetry)
  - Enables a more efficient design (No one size fits all)

Some Solution Principles (More Compact)

- Data-centric design
- All components intelligent
- Better cross-layer communication, better interfaces
- Better-than-worst-case design
- Heterogeneity
- Flexibility, adaptability

Open minds

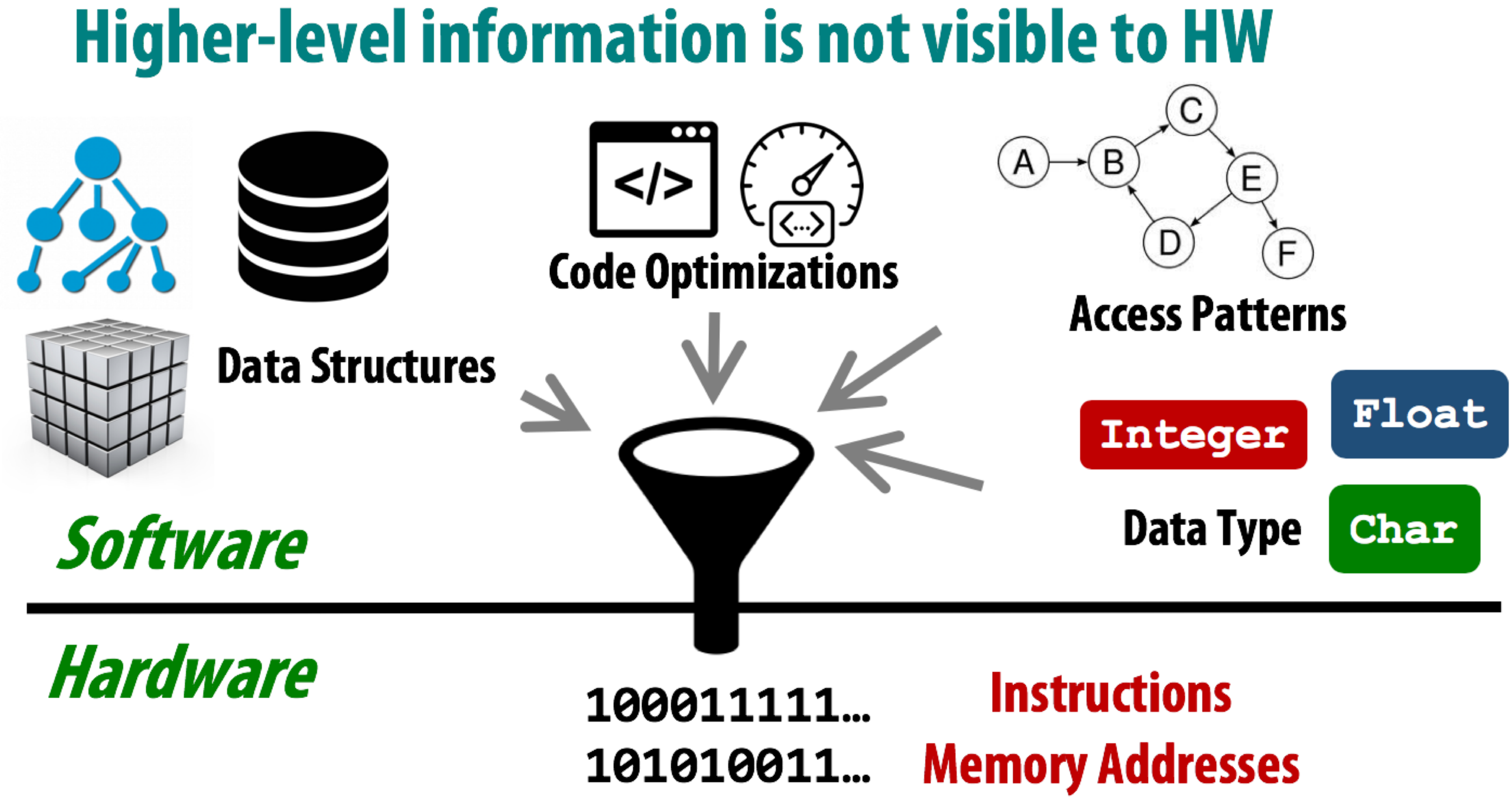
Data-Aware Architectures

Data-Aware Architectures

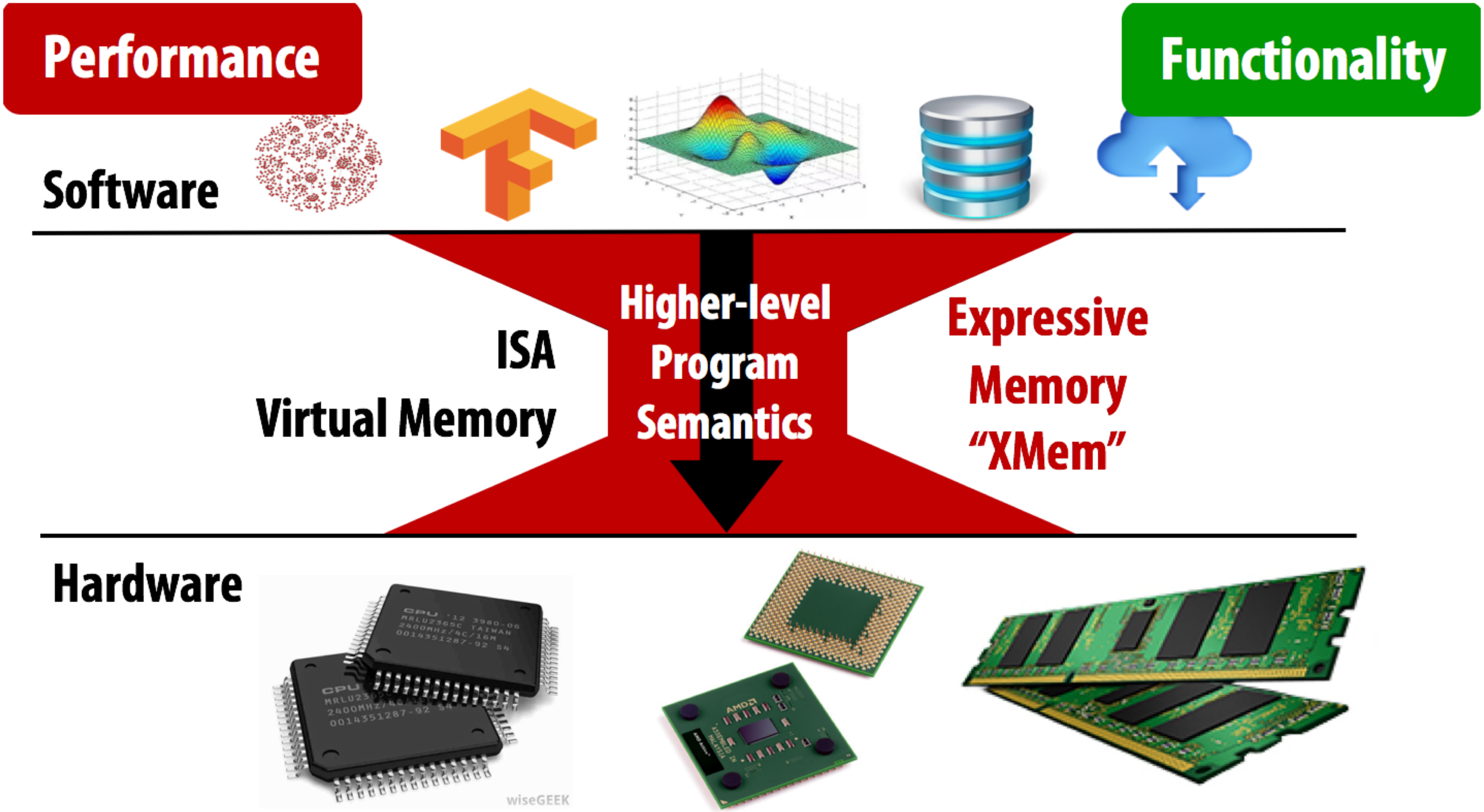
- A data-aware architecture understands what it can do with and to each piece of data

- It makes use of different properties of data to improve performance, efficiency and other metrics
  - Compressibility
  - Approximability
  - Locality
  - Sparsity
  - Criticality for Computation X
  - Access Semantics
  - …

One Problem: Limited Interfaces



A Solution: More Expressive Interfaces



Expressive (Memory) Interfaces

- Nandita Vijaykumar, Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko, Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons and Onur Mutlu, **"A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory"**
  Proceedings of the 45th International Symposium on Computer Architecture (***ISCA***), Los Angeles, CA, USA, June 2018.
  [Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video]

**A Case for Richer Cross-layer Abstractions:
Bridging the Semantic Gap with Expressive Memory**

Nandita Vijaykumar[†§] Abhilasha Jain[†] Diptesh Majumdar[†] Kevin Hsieh[†] Gennady Pekhimenko[‡]
Eiman Ebrahimi[ℵ] Nastaran Hajinazar[+] Phillip B. Gibbons[†] Onur Mutlu[§†]

[†]**Carnegie Mellon University** [‡]**University of Toronto** [ℵ]**NVIDIA**
[+]**Simon Fraser University** [§]**ETH Zürich**

Expressive (Memory) Interfaces for GPUs

- Nandita Vijaykumar, Eiman Ebrahimi, Kevin Hsieh, Phillip B. Gibbons and Onur Mutlu, **"The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs"** *Proceedings of the 45th International Symposium on Computer Architecture* (***ISCA***), Los Angeles, CA, USA, June 2018.
  [Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video]

The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs

Nandita Vijaykumar[†§] Eiman Ebrahimi[‡] Kevin Hsieh[†]
Phillip B. Gibbons[†] Onur Mutlu[§†]

[†]**Carnegie Mellon University** [‡]**NVIDIA** [§]**ETH Zürich**

Data-centric

Data-driven

Data-aware

Source: http://spectrum.ieee.org/image/MjYzMzAyMg.jpeg

It Is Time to …

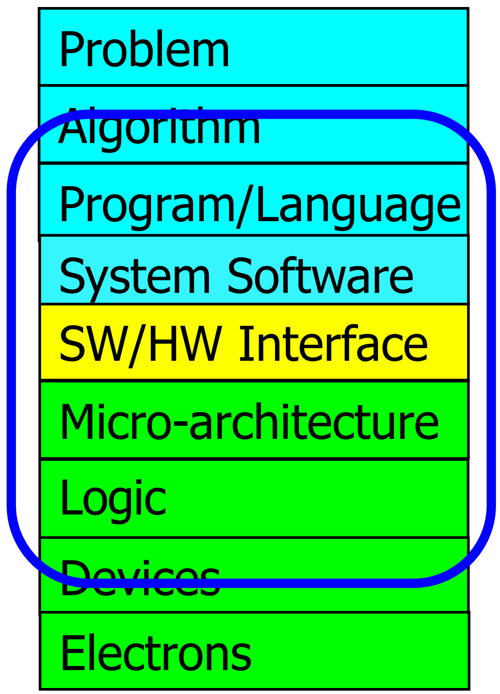
- … design **principled system architectures** to solve the **memory problem**

- … design complete systems to be balanced, high-performance, and energy-efficient, i.e., data-centric (or memory-centric)

- … **make memory a key priority** in system design and optimize it & integrate it better into the system

- This can
  - Lead to **orders-of-magnitude** improvements
  - **Enable new applications & computing platforms**
  - **Enable better understanding of nature**
  - **…**

We Need to Revisit the Entire Stack



We can get there step by step

Memory Systems
and Memory-Centric Computing Systems
Lecture 6a: Low-Latency Memory II

Prof. Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu

19 June 2019

TU Wien Fast Course 2019

SAFARI ETH zürich Carnegie Mellon