Chronus
Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat

Giray Yağlıkçı Geraldo Oliveira Ataberk Olgun

Nisa Bostancı İsmail Emir Yüksel Haocong Luo

Oğuz Ergin Onur Mutlu

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

SAFARI ETH zürich kasırga

Executive Summary

Problem:
- DRAM continues to become more vulnerable to read disturbance
- Latest update (April 2024) to DDR5 standard introduces **Per Row Activation Counting (PRAC)** to mitigate read disturbance
- No prior work investigates **PRAC's security** and **performance**

Goal: Rigorously analyze, characterize, and improve the **security** and **performance** of the DDR5 standard **PRAC** mechanism

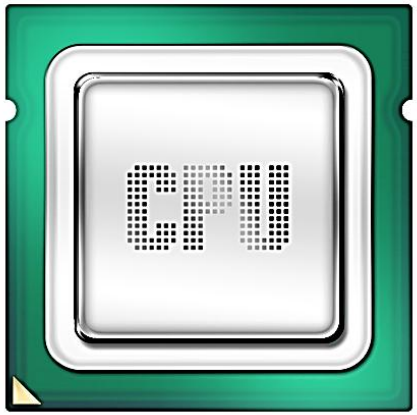
Mathematical analysis & extensive simulations show that: PRAC
- Has significant (10%) performance overhead for modern DRAM chips because **PRAC** requires **additional time to track read disturbance aggressors**
- **Poorly scales** to future DRAM chips that are more vulnerable to read disturbance because **PRAC** is vulnerable to an adversarial access pattern (i.e., the wave attack)

Chronus: Solves **PRAC's** two major weaknesses by
- **Concurrently tracking read disturbance aggressors** while serving accesses
- **Securing PRAC** against a potential wave attack

Key Results: Chronus provides **high performance and low energy** at low hardware complexity overhead and **outperforms** five state-of-the-art solutions

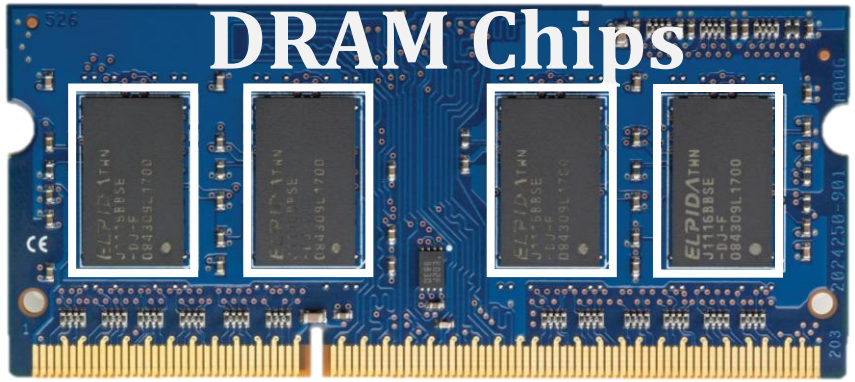
Outline

DRAM Organization



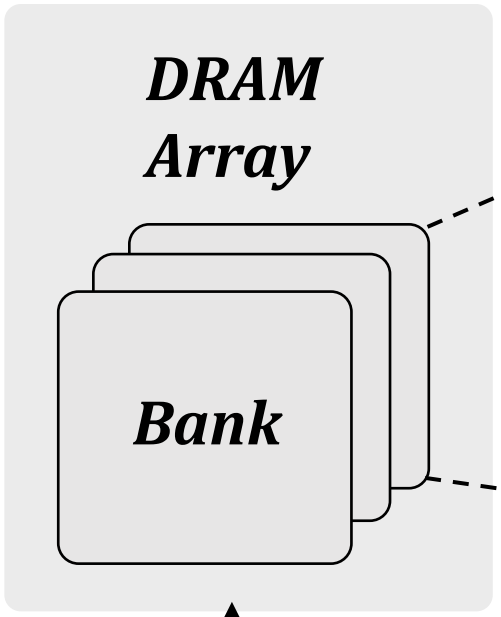
Processor

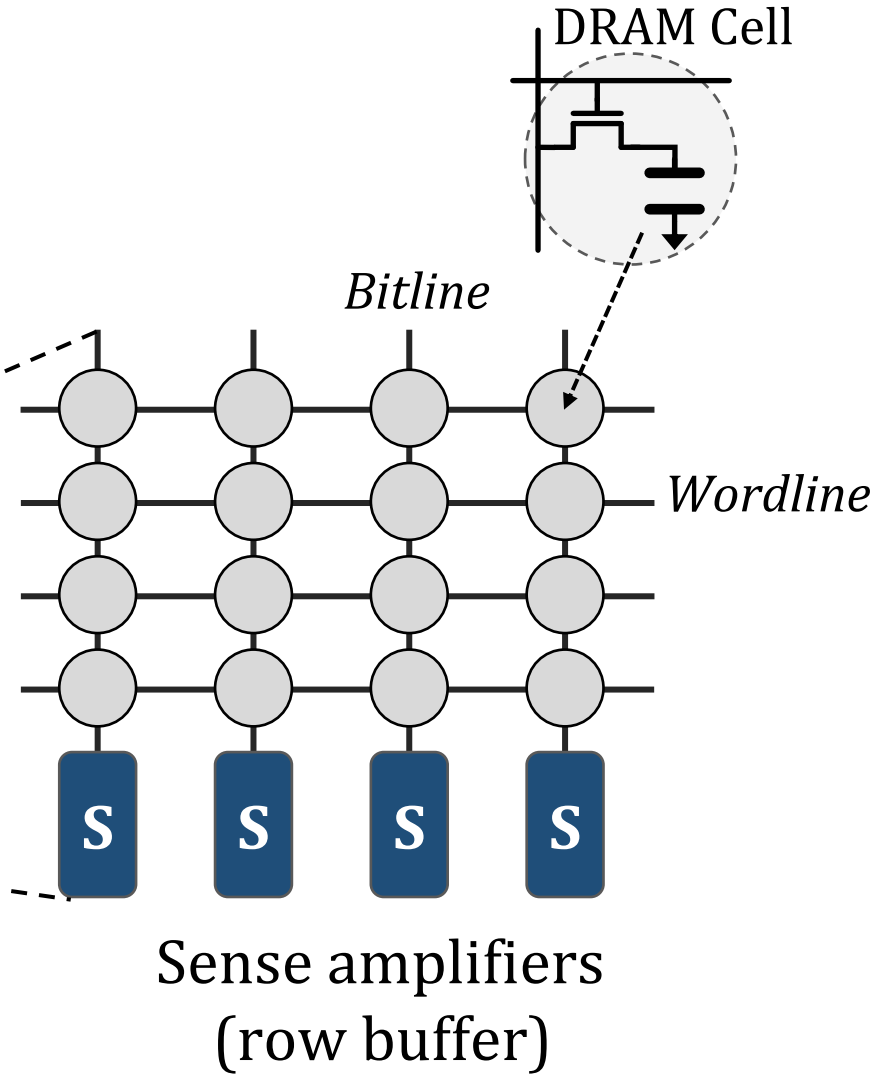
DRAM
Channel



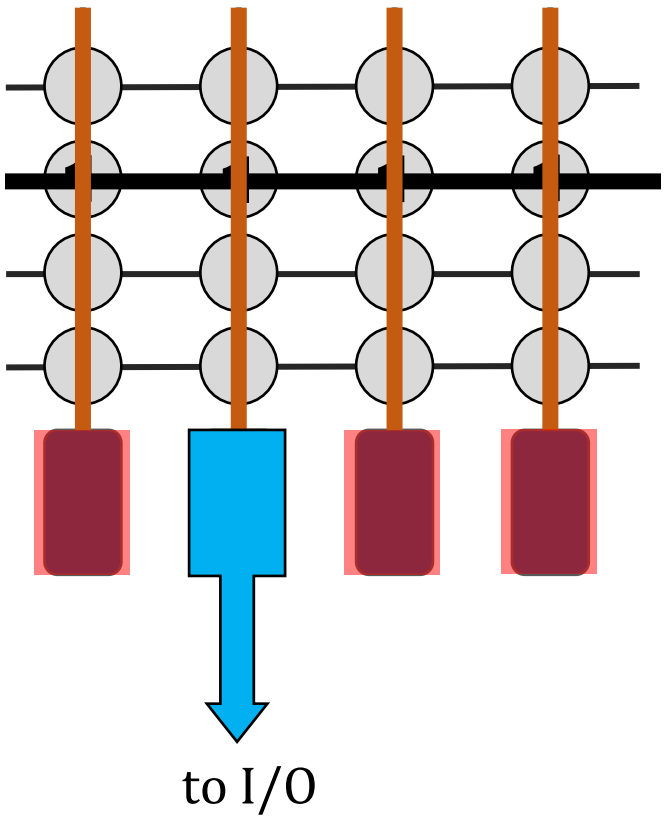
DRAM Module

DRAM Organization





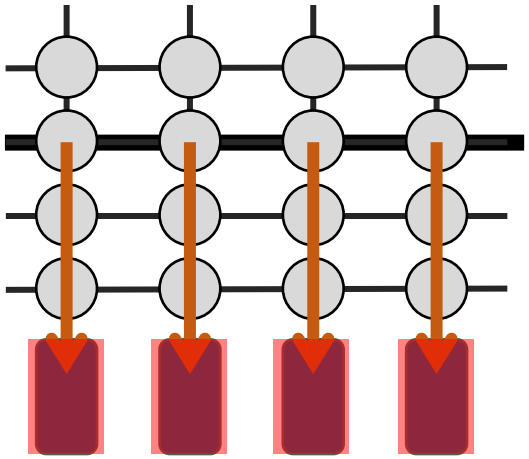
DRAM Operations

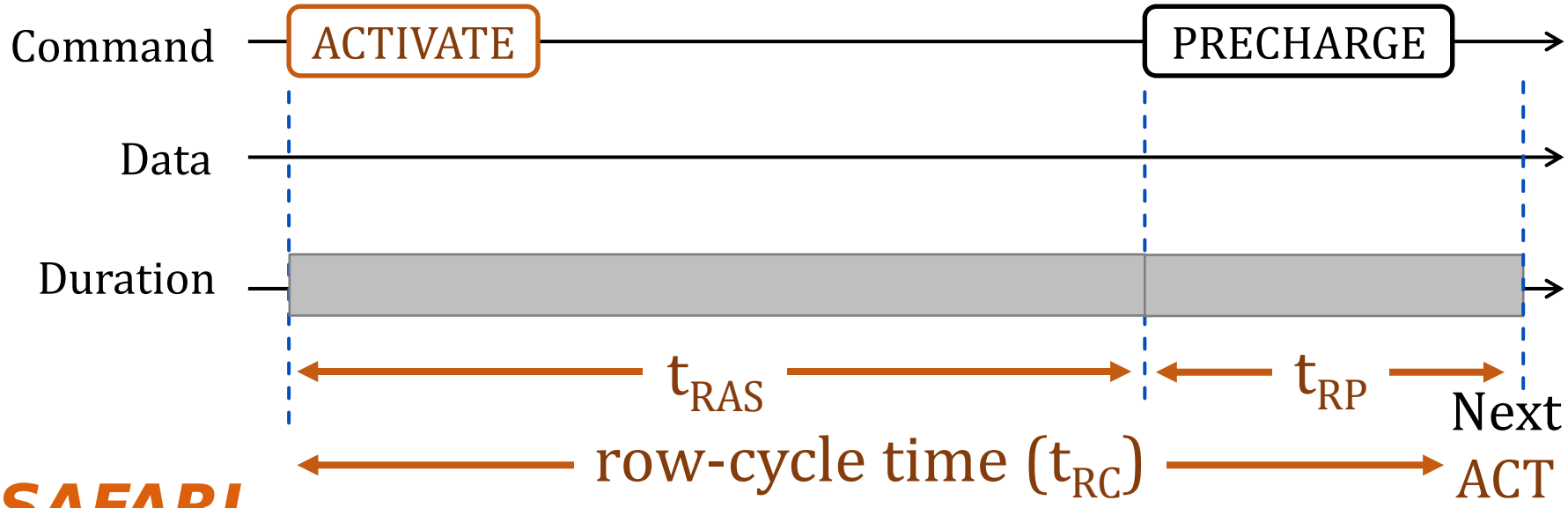


1. **ACTIVATE**: Fetch the row into the **row buffer**
2. **READ/WRITE**: Retrieve or update data
3. **PRECHARGE**: Prepare the bank for a new ACTIVATE

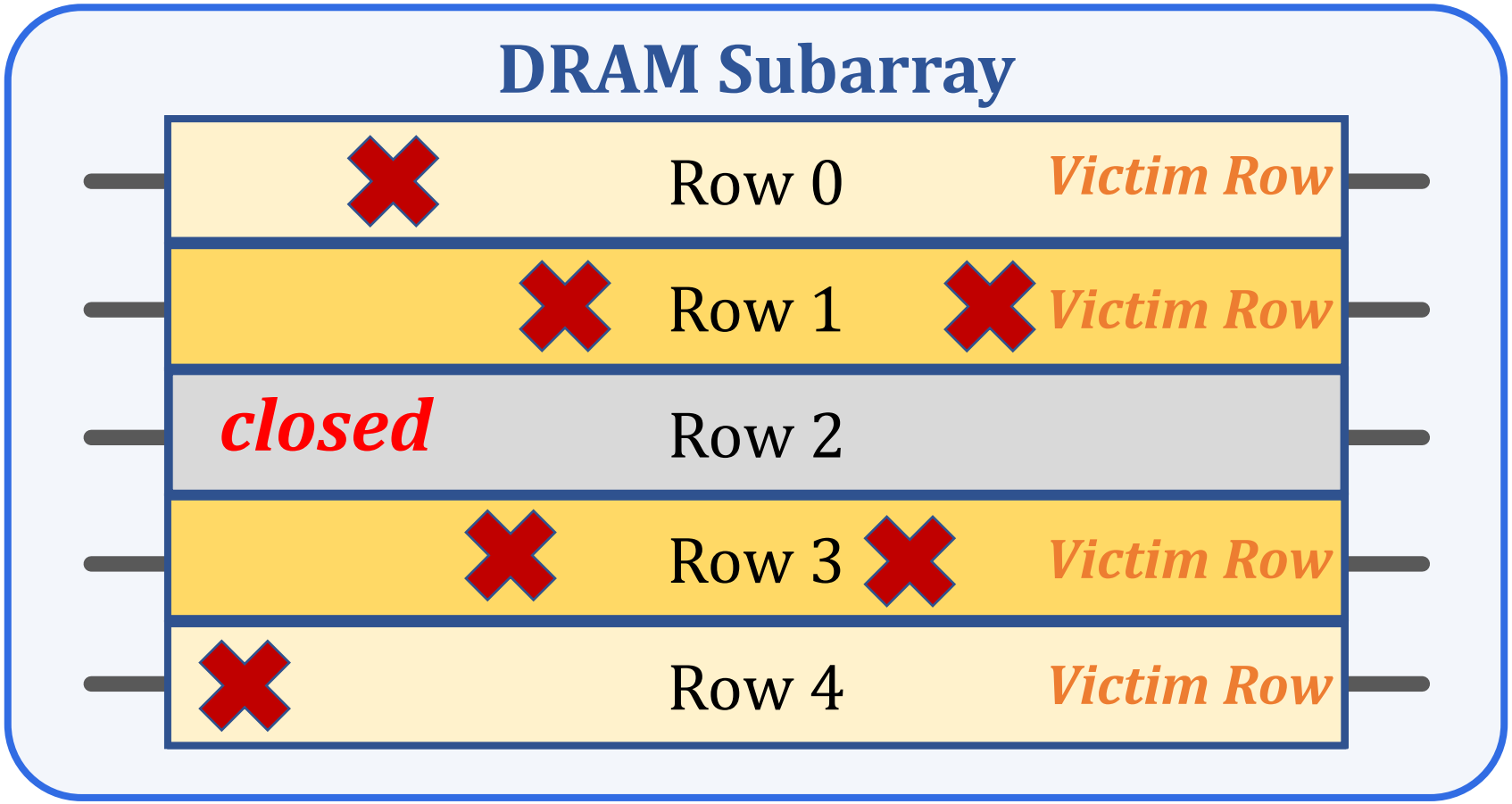
ACTIVATION and **PRECHARGE** are time-consuming operations

DRAM Access Latency





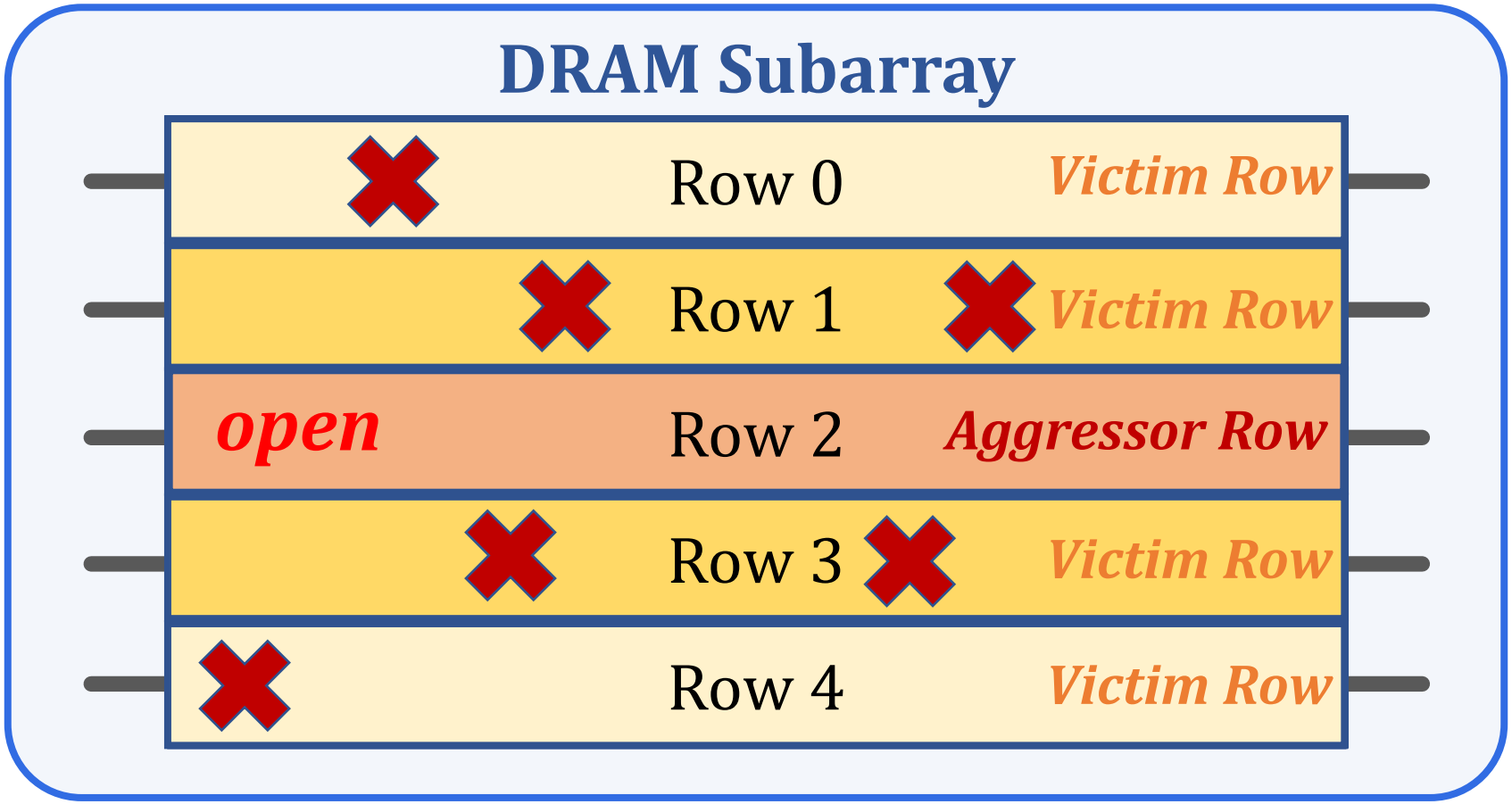
RowHammer: A Prime Example of Read Disturbance



Repeatedly **opening** (activating) and **closing** (precharging) a DRAM row causes **read disturbance bitflips** in nearby cells

[Kim+ ISCA'20]

Read Disturbance Vulnerabilities (I)



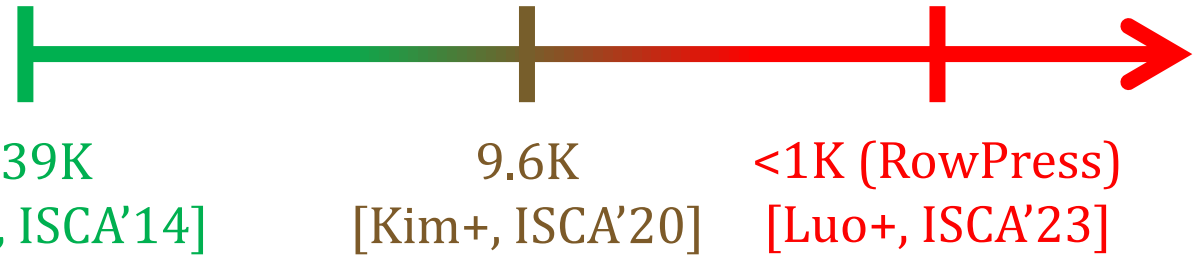
The minimum number of activations that causes a bitflip is called **the RowHammer threshold (N_{RH})**

[Kim+ ISCA'20]

Read Disturbance Vulnerabilities (II)

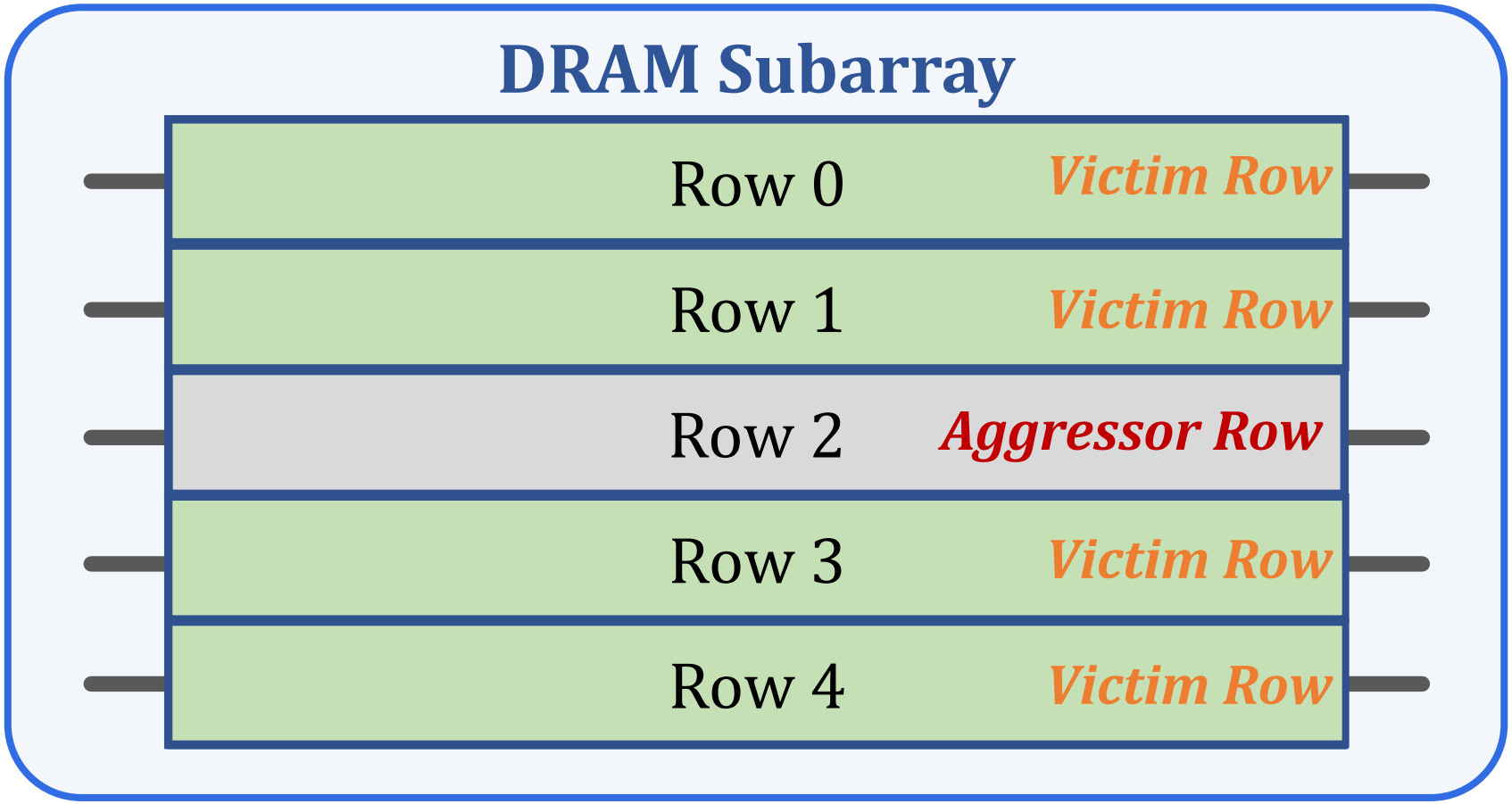
- DRAM chips are more vulnerable to read disturbance today

- Read disturbance bitflips occur at much lower activation counts (more than two orders of magnitude decrease in less than a decade):



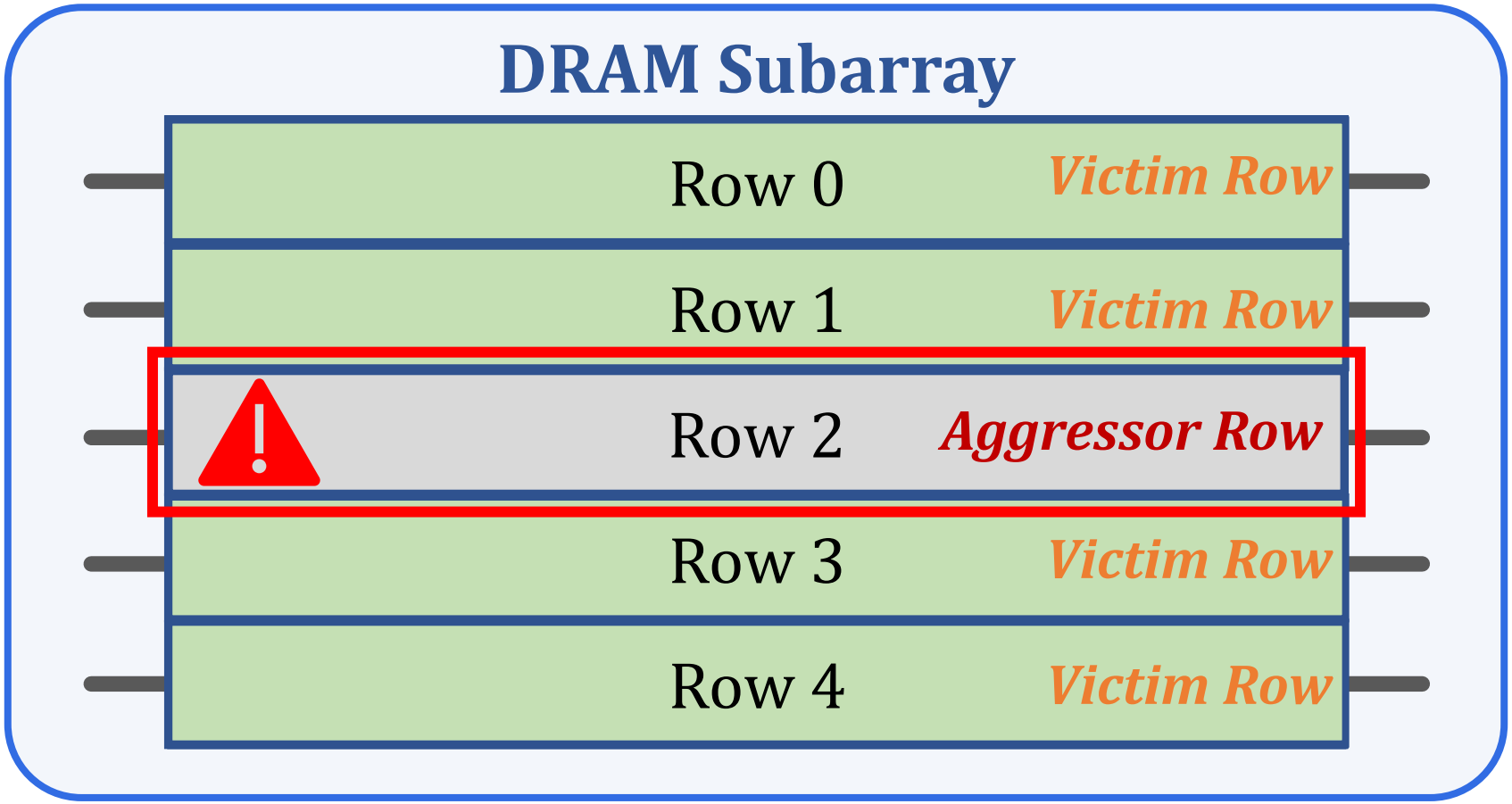
It is **critical** to prevent read disturbance bitflips **effectively** and **efficiently** for highly vulnerable systems

Existing RowHammer Mitigations (I): Preventive Refresh



Refreshing potential victim rows
mitigates RowHammer bitflips

Existing RowHammer Mitigations (II): DRAM Aggressor Row Tracking or Estimation



Necessary to accurately **track** or **estimate** aggressor DRAM row activation counts to preventively refresh potential victim rows

Outline

Background

Industry Solutions to Read Disturbance

Security Analysis of Industry Solutions

Overhead Analysis of Industry Solutions

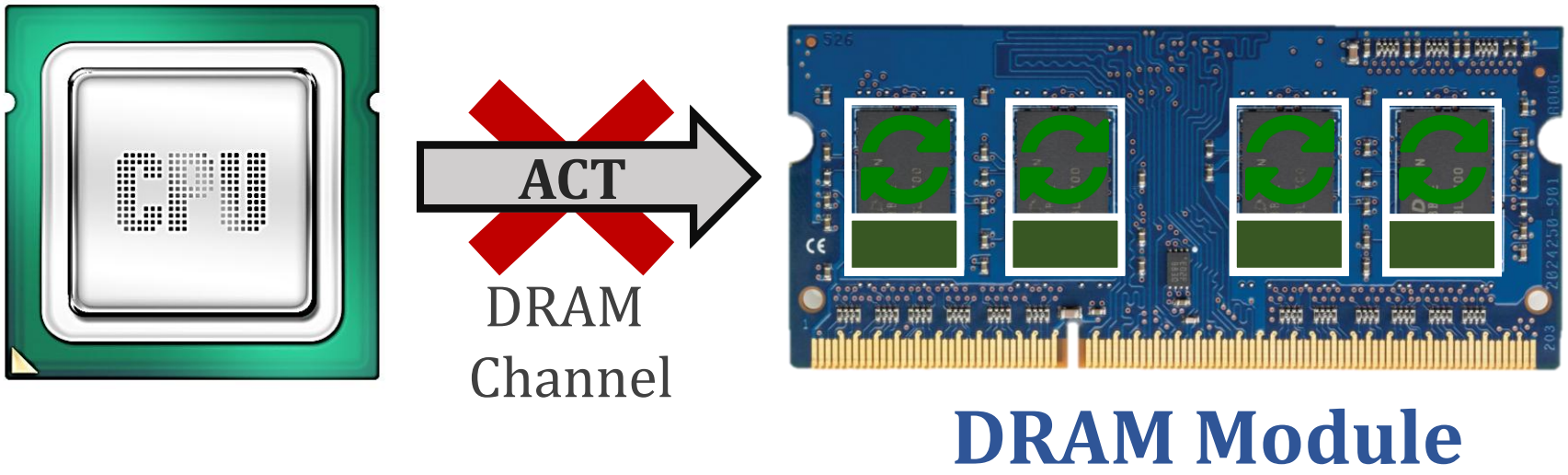
Chronus

Evaluation

Conclusion

Industry Solutions to Read Disturbance: When To Refresh? (I)

Preventive refresh operations are
blocking and time consuming operations

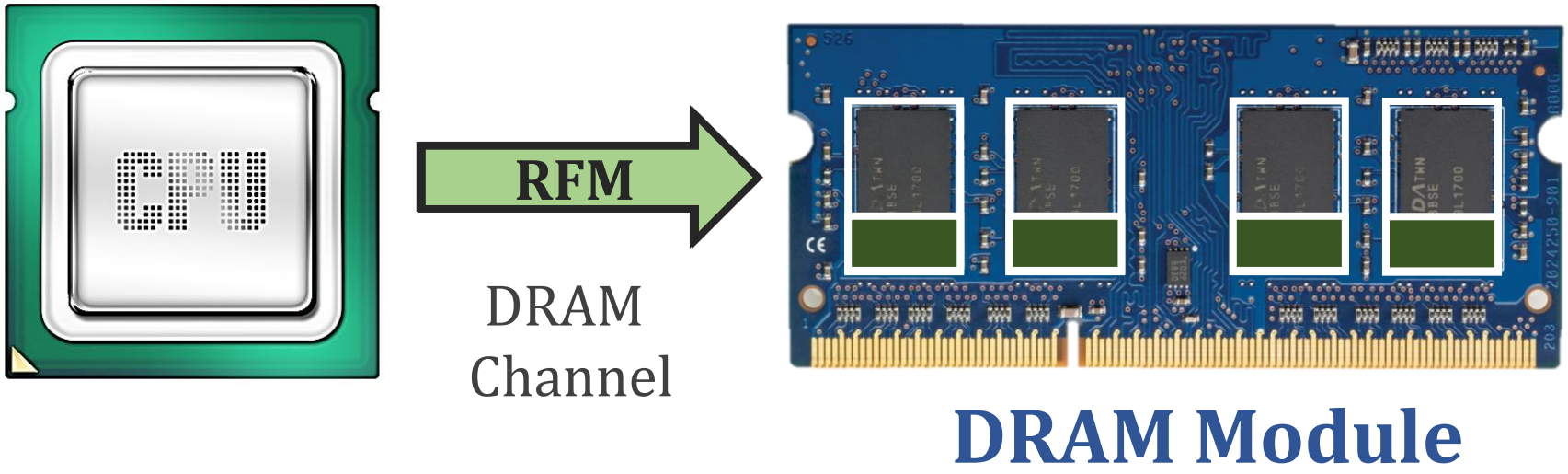


DRAM Module

Memory controller **cannot access** a memory bank undergoing a preventive refresh

Industry Solutions to Read Disturbance: When To Refresh? (II)

Earlier JEDEC DDR5 specifications introduce **Refresh Management (RFM)** commands



Memory controller sends an **RFM command** to allow time for preventive refreshes

Industry Solutions to Read Disturbance

Periodic Refresh Management (PRFM)

Memory controller **periodically** issues RFM commands

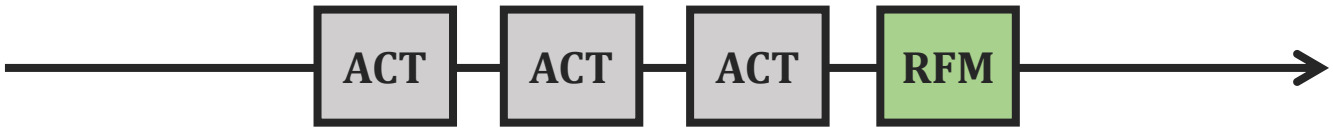
Per Row Activation Counting and Back-Off (PRAC)

DRAM chip **tracks** row activations and **requests** RFMs by sending a **back-off**

Industry Solutions to Read Disturbance: Periodic Refresh Management (PRFM)

Bank Activation Counters			
3	0	0	0

PRFM tracks activations with **low accuracy**, causing a **high number** of preventive refreshes, leading to **large** performance and energy overheads



Industry Solutions to Read Disturbance

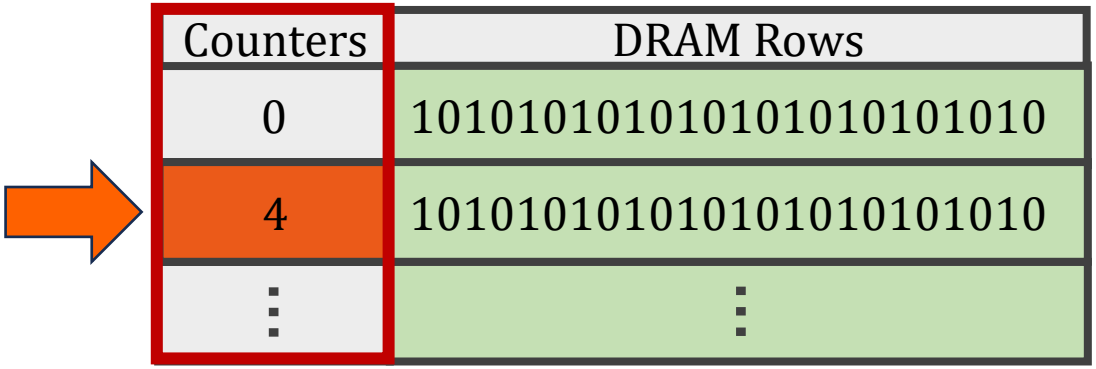
Periodic Refresh Management (PRFM)

Memory controller **periodically** issues RFM commands

Per Row Activation Counting and Back-Off (PRAC)

DRAM chip **tracks** row activations and **requests** RFMs by sending a **back-off**

Industry Solutions to Read Disturbance: Per Row Activation Counting



PRAC allows **accurate tracking** of aggressor row activations

Industry Solutions to Read Disturbance: Per Row Activation Counting DRAM Timings

Counters	DRAM Rows
0	10101010101010101010101010
0	10101010101010101010101010
⋮	⋮

The activation counter of a row is updated while the row is being closed

PRAC increases precharge duration (t_{RP}) by **140%**

[JEDEC 2024]

Industry Solutions to Read Disturbance: Per Row Activation Counting DRAM Timings

Timing parameter changes for DDR5-3200AN speed bin [JEDEC JESD79-5C, April 2024]

t_{RP} : +21ns (+140%)

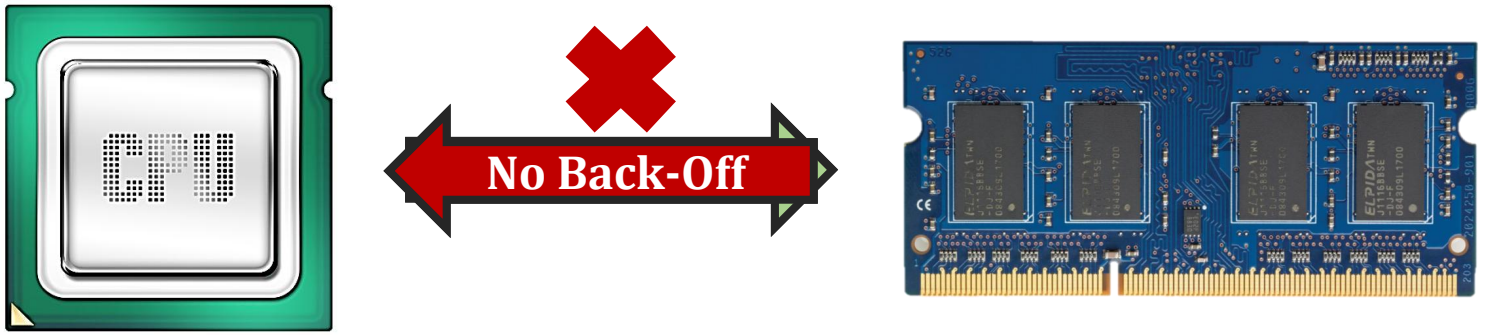
t_{RAS} : -16ns (-50%)

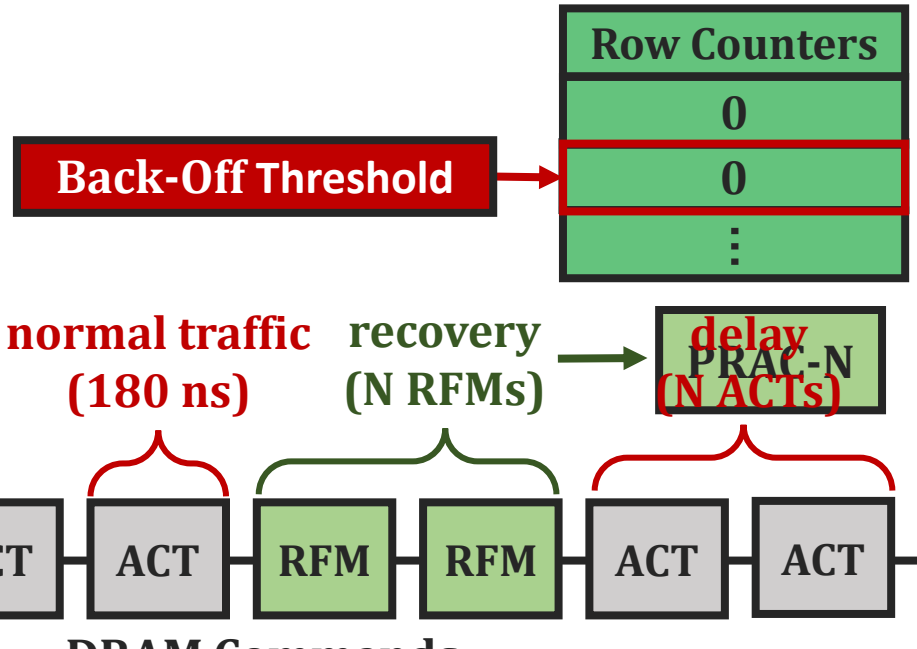
t_{RTP} : -2.5ns (-33%)

t_{WR} : -20ns (-66%)

t_{RC} : +5ns (+10%)

Industry Solutions to Read Disturbance: Per Row Activation Counting (PRAC)





Outline

Background

Industry Solutions to Read Disturbance

Security Analysis of Industry Solutions

Performance Analysis of Industry Solutions

Chronus

Evaluation

Conclusion

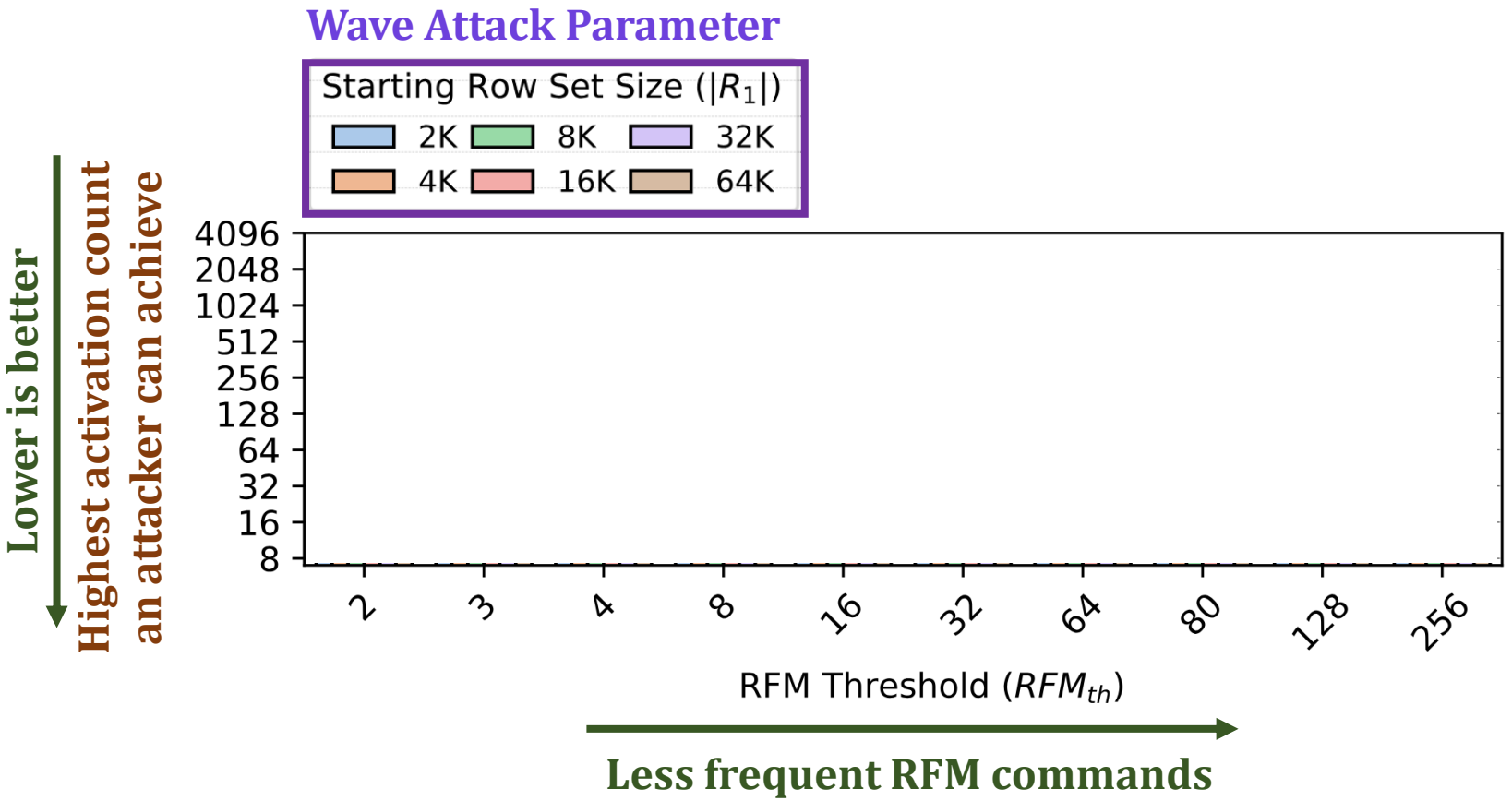
Security Analysis of Industry Solutions: Mathematical Security Analysis Methodology

- Wave attack [Yağlıkçı+, 2021] : **worst-case** access pattern
  - maximizes hammer count by using decoy rows
  - on a system with **PRFM**
  - on a system with **PRAC**

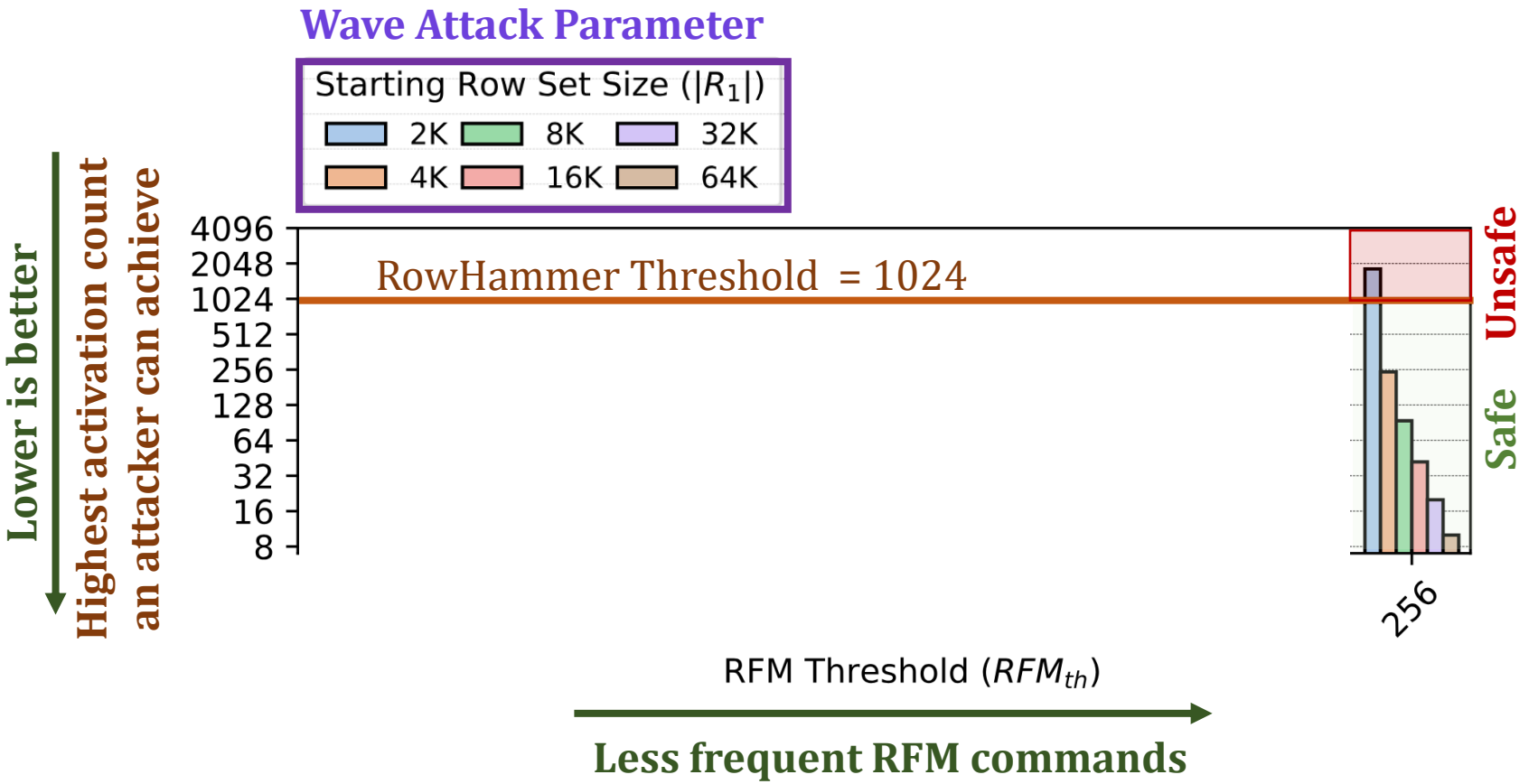
- **Parameters:**
  - **Starting row set size:** # of rows that the wave attack hammers
  - **RFM threshold** (PRFM)
  - **Back-Off threshold** (PRAC)

- **Result:** **Worst possible** (highest) activation count that an attacker can achieve to a row

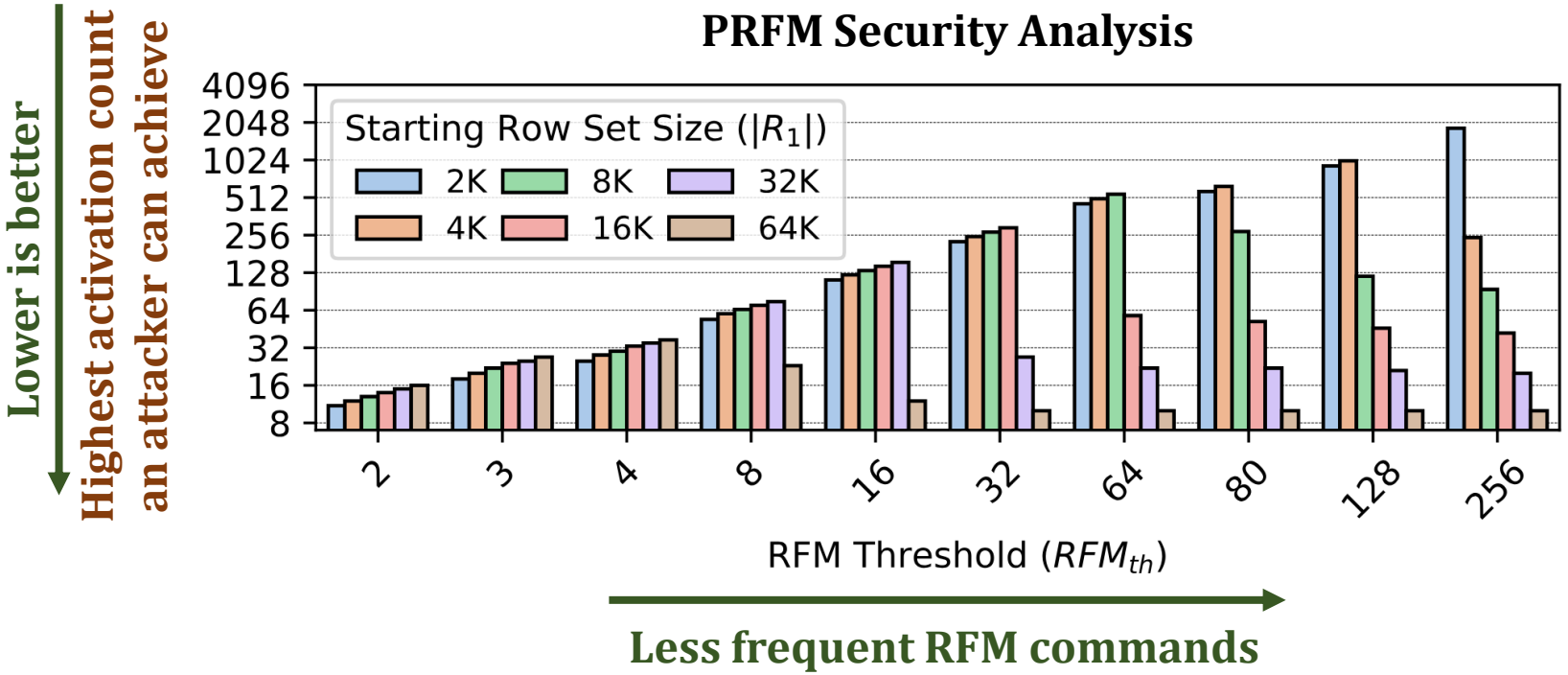
Security Analysis of Industry Solutions: Secure PRFM Configurations



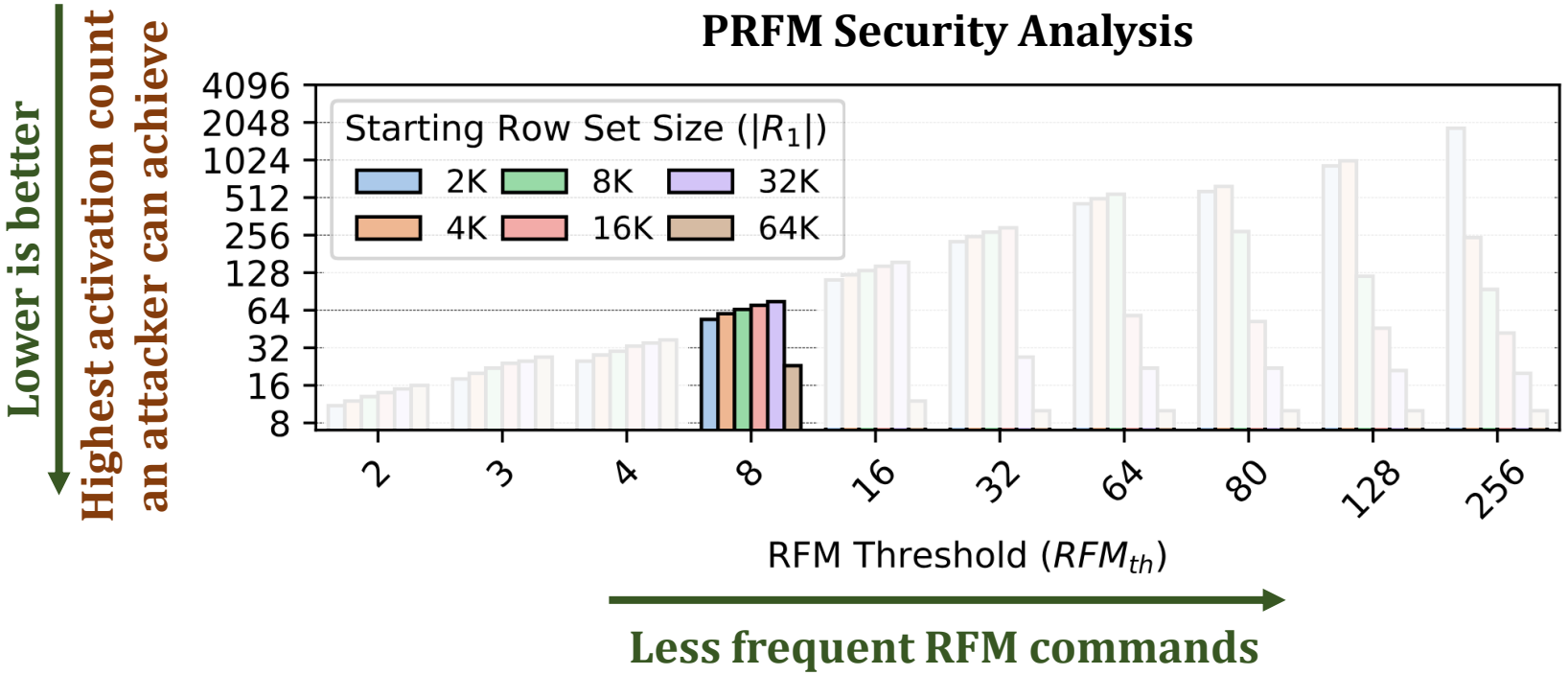
Security Analysis of Industry Solutions: Secure PRFM Configurations



Security Analysis of Industry Solutions: Secure PRFM Configurations

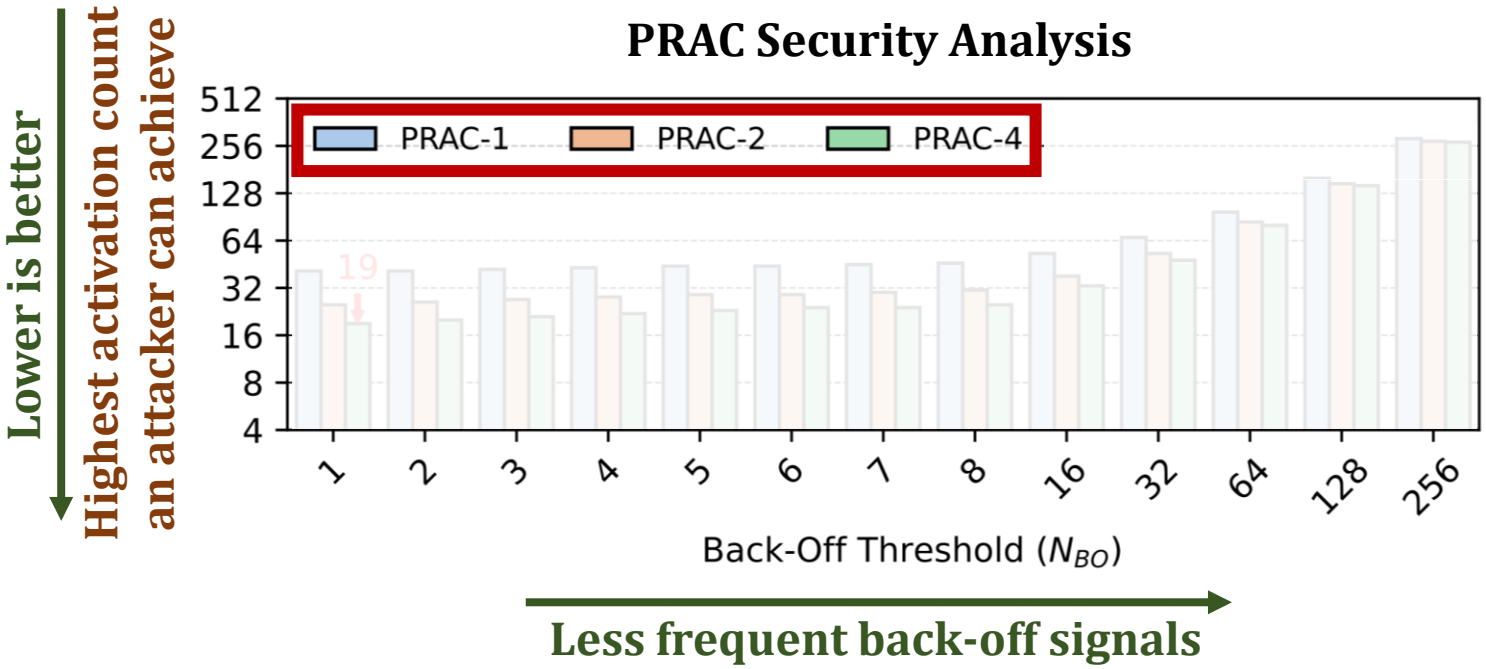


Security Analysis of Industry Solutions: Secure PRFM Configurations

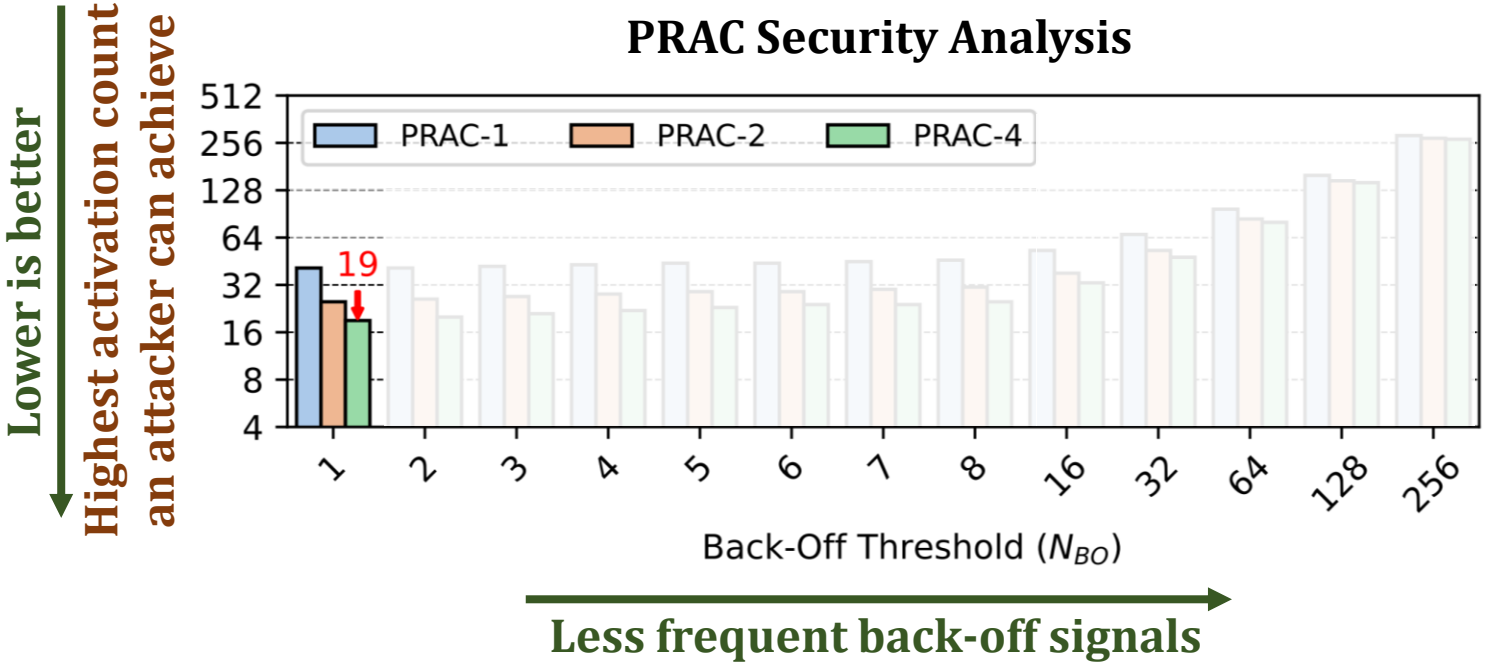


PRFM must send RFM commands **very frequently (every ~8 ACTs)** to prevent bitflips at **low** read disturbance thresholds (below 128)

Security Analysis of Industry Solutions: Secure PRAC Configurations



Security Analysis of Industry Solutions: Secure PRAC Configurations



PRAC can be configured for **secure** operation against RowHammer thresholds **as low as 20**

Outline

Background

Industry Solutions to Read Disturbance

Security Analysis of Industry Solutions

Performance Analysis of Industry Solutions

Chronus

Evaluation

Conclusion

Performance Analysis of Industry Solutions: Evaluation Methodology

- **Performance evaluation:**
  cycle-level simulations using Ramulator 2.0 [Luo+, CAL 2023]

- **System Configuration:**

Processor	4 cores, 4.2GHz clock frequency,
	4-wide issue, 128-entry instruction window
DRAM	DDR5, 1 channel, 2 rank/channel, 8 bank groups,
	4 banks/bank group, 64K rows/bank
Memory Ctrl.	64-entry read and write requests queues,
	Scheduling policy: FR-FCFS with a column cap of 4
	Last-Level Cache 8 MiB (4-core)

- **Workloads:** 60 4-core workload mixes
  - SPEC CPU2006, SPEC CPU2017, TPC, MediaBench, YCSB

Performance Analysis of Industry Solutions: Industry Solution Variants

1 **PRFM**

Memory controller **periodically** issues RFM

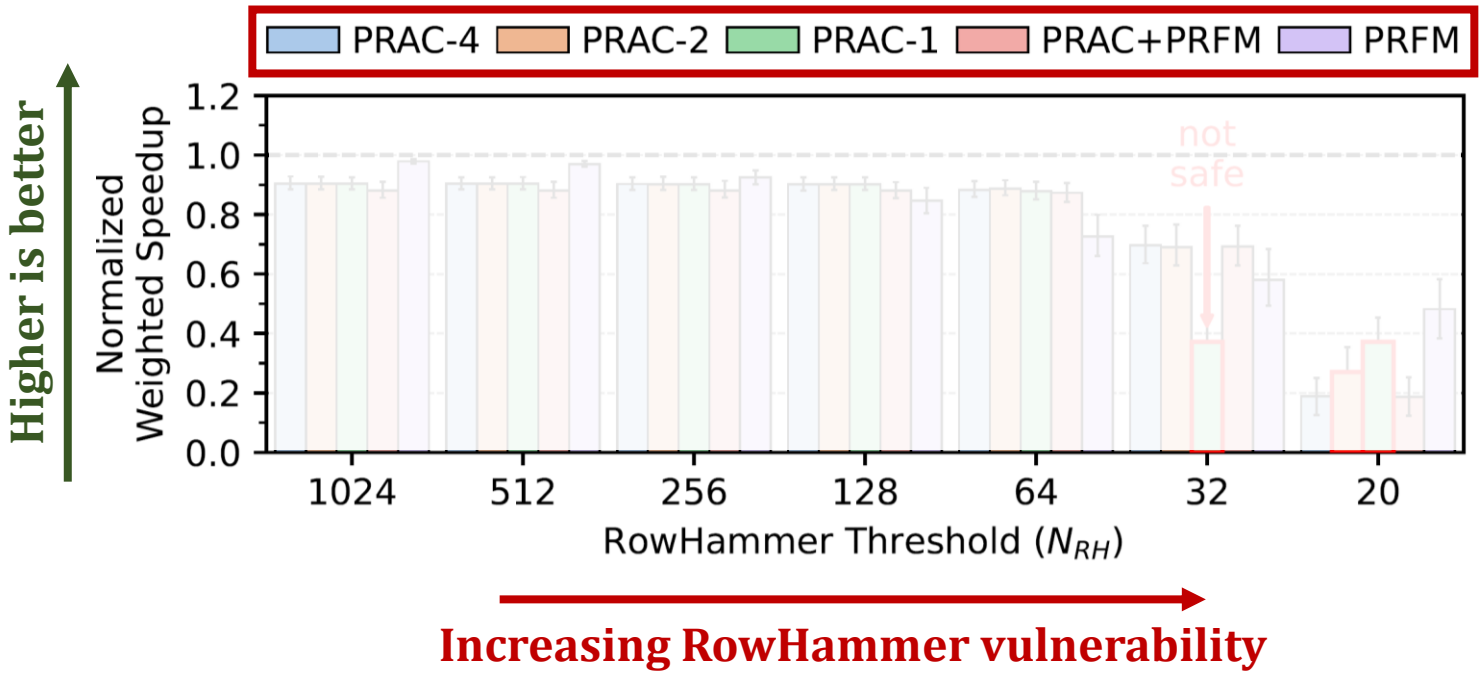
2 **PRAC-N**

Memory controller issues **N** RFMs each with **back-off**

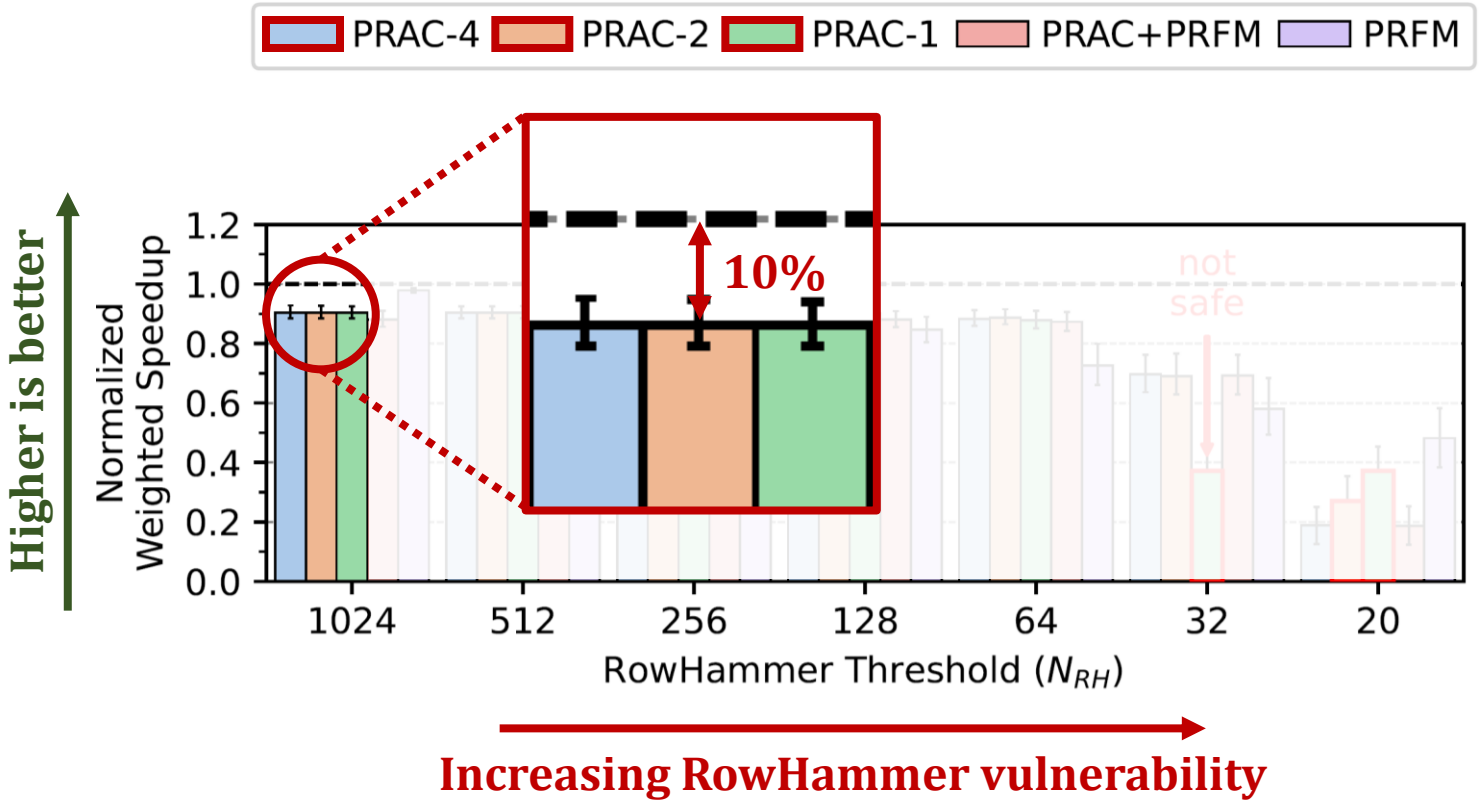
3 **PRAC+PRFM**

Memory controller issues RFM **periodically** and with **back-offs**

Experimental Results: Performance Overhead and Its Scaling

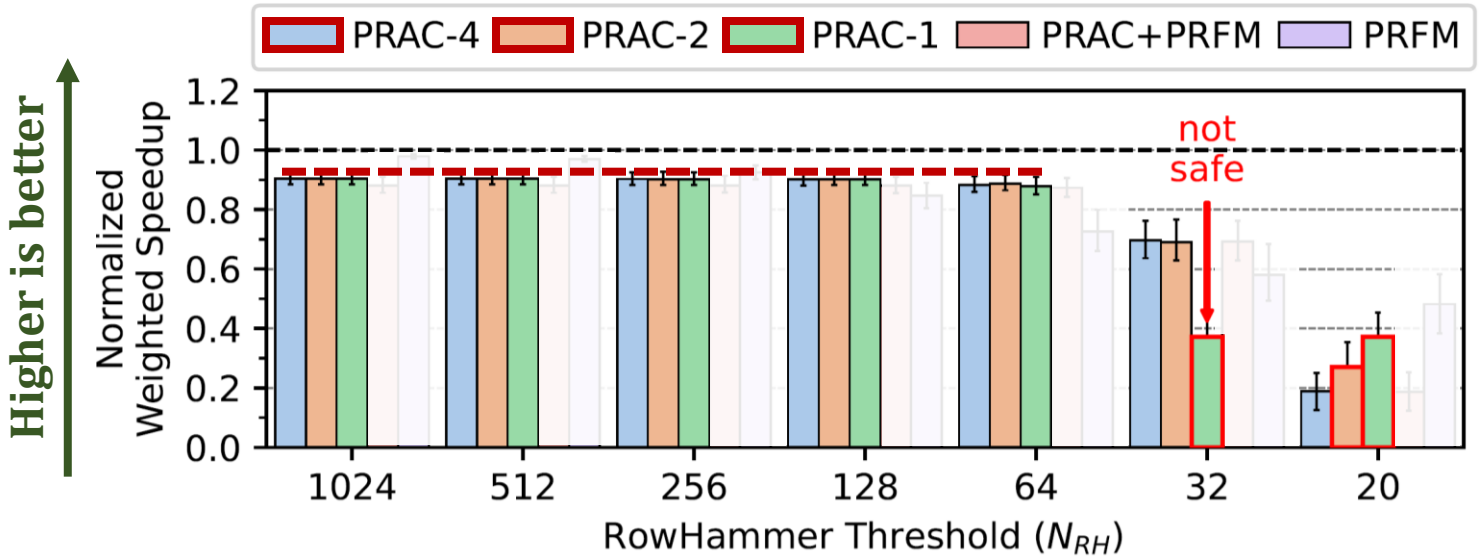


Experimental Results: Performance Overhead and Its Scaling



At high N_{RH} values, **PRAC** has **non-negligible** (**10%**) performance overhead due to **increased** DRAM access latency

Experimental Results: Performance Overhead and Its Scaling



Above N_{RH} of 64, **PRAC** overhead only **slightly** increases due to **timely** preventive refreshes

Below N_{RH} of 64, **PRAC** overhead **significantly** increases due to conservative configuration against a potential **wave attack**

PRAC's Two Major Outstanding Problems

1

Increased critical DRAM **timing parameters**

Large performance overhead even at high read disturbance thresholds

2

Wave attack vulnerability requires **aggressive configuration**

Poor scaling to low read disturbance thresholds

Outline

Background

Industry Solutions to Read Disturbance

Security Analysis of Industry Solutions

Performance Analysis of Industry Solutions

Chronus

Evaluation

Conclusion

Chronus: Key Ideas

1

Concurrently update counters while serving DRAM accesses

No increase in critical DRAM timing parameters

2

Prevent wave attacks by providing DRAM chip more control over back-offs and preventive refreshes

Better scaling to low read disturbance thresholds

Chronus: Overview

Concurrent Counter Update

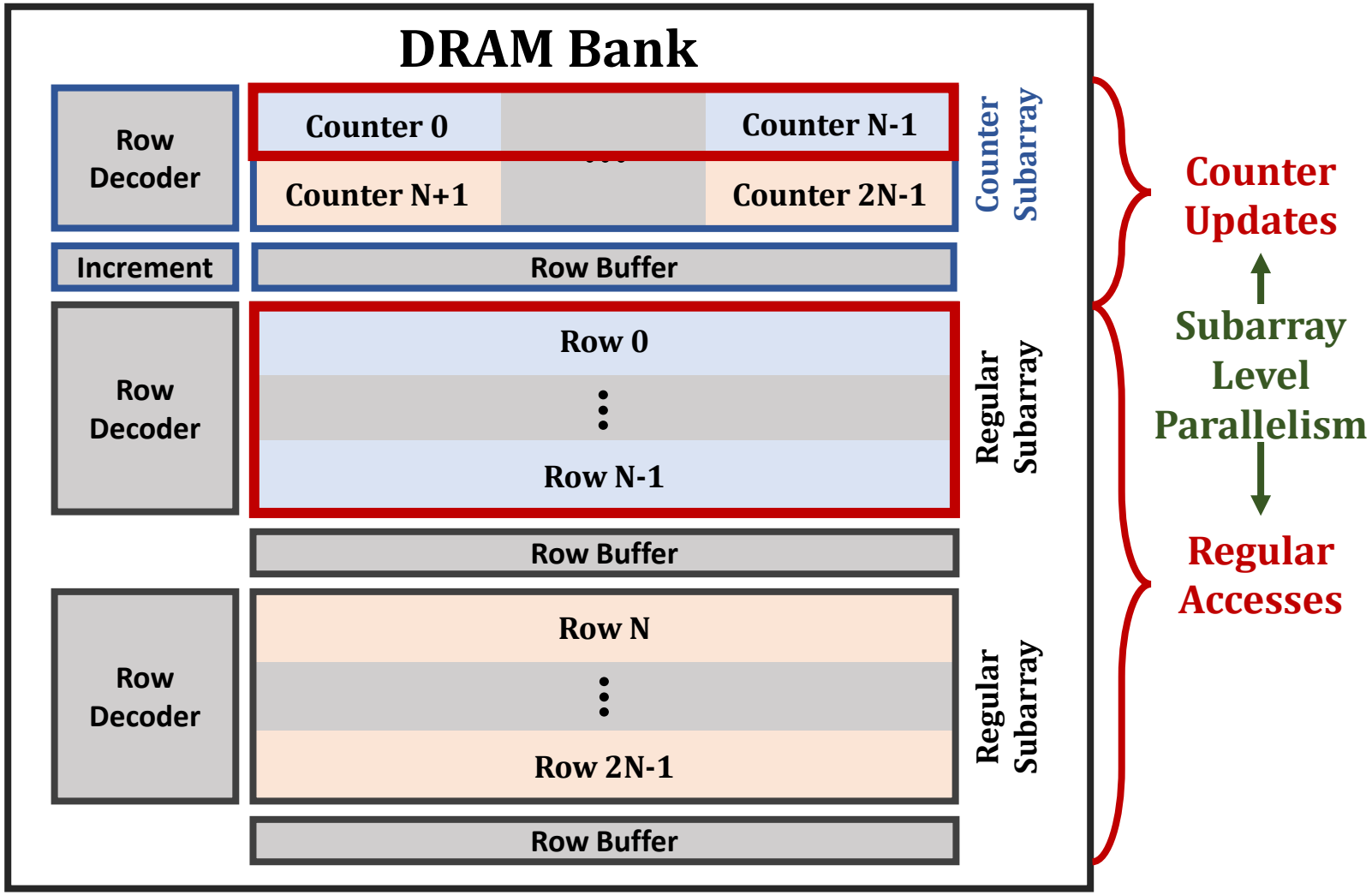
Concurrently update counters while serving DRAM accesses

Chronus Back-Off

Dynamically control the number of refreshes as needed

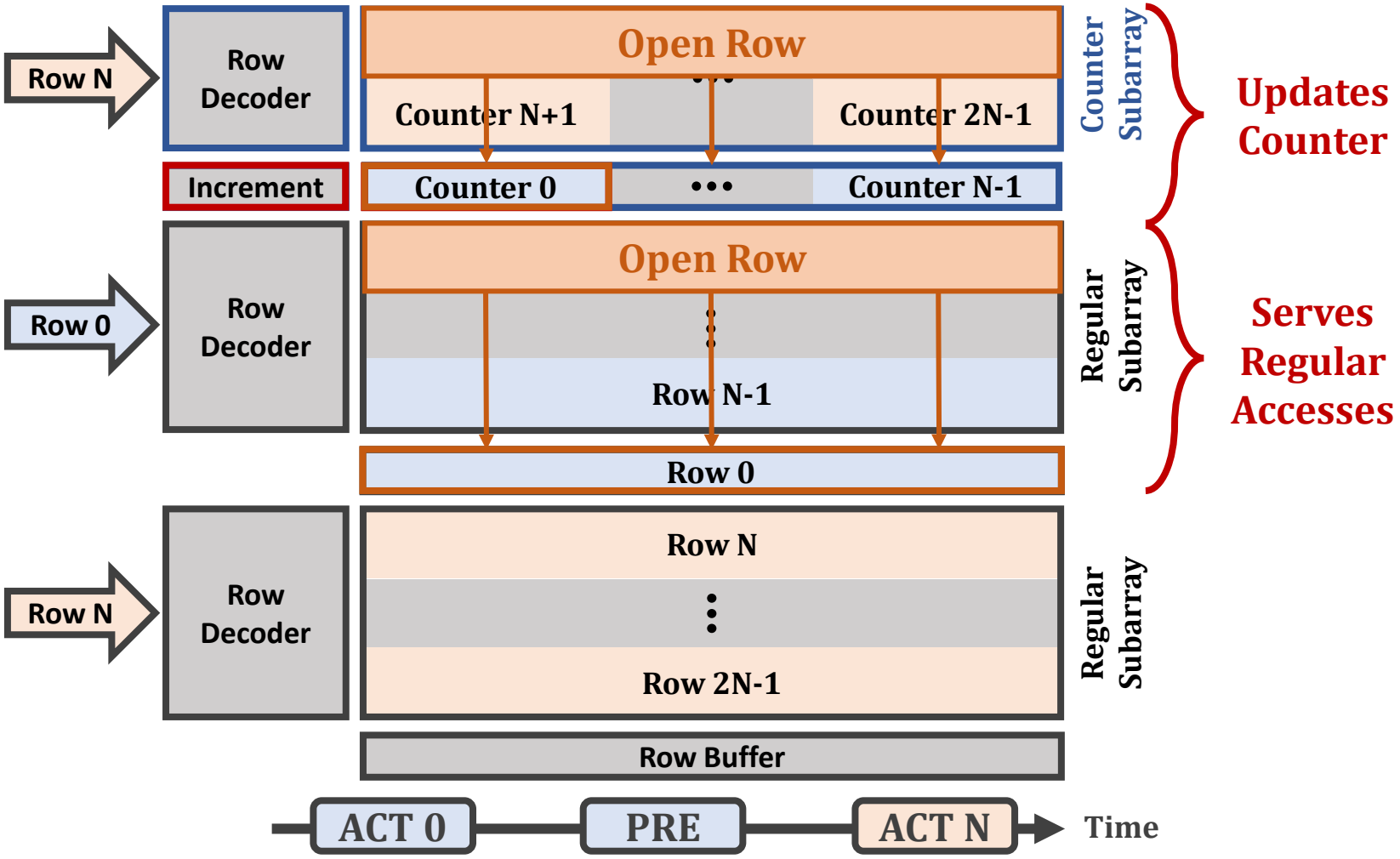
Chronus: Concurrent Counter Update

Chronus concurrently updates counters while serving accesses



Chronus: Concurrent Counter Update

Chronus concurrently updates counters while serving accesses



Chronus: Concurrent Counter Update (More in the paper)

https://arxiv.org/abs/2502.12650

2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA)

Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat[§†] A. Giray Yağlıkçı[§] Geraldo F. Oliveira[§] Ataberk Olgun[§]
Nisa Bostancı[§] Ismail Emir Yuksel[§] Haocong Luo[§] Oğuz Ergin[‡†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology* [‡]*University of Sharjah*

Read disturbance in modern DRAM is an important robustness (security, safety, and reliability) problem, where repeatedly accessing (hammering) a row of DRAM cells (DRAM row) in-

ory address should *not* cause unintended side-effects on data stored in other addresses [1]. Unfortunately, with aggressive technology scaling, DRAM [2], the prevalent main memory

Counter Subarray
incurs 0.5% area overhead per bank

Counter Subarray
induces 19% energy overhead to opening and closing a row

Chronus: Overview

Concurrent Counter Update

Concurrently update counters while serving DRAM accesses

Chronus Back-Off

Dynamically control the number of **refreshes as needed**

Chronus: Chronus Back-Off

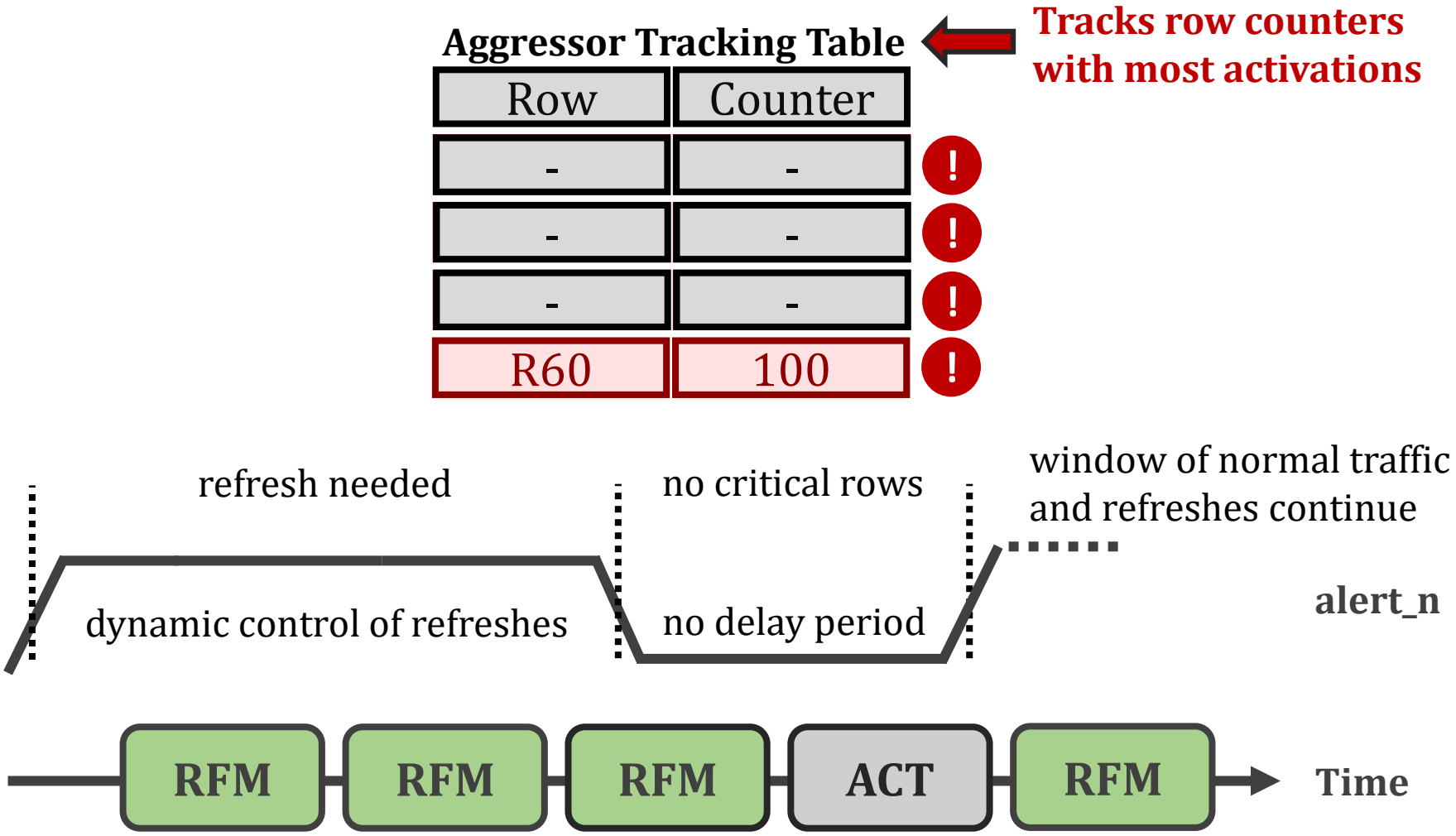
Chronus Back-Off prevents a potential **wave attack** with two simple changes to PRAC Back-Off

1) **Chronus Back-Off dynamically controls** the number of refreshes as needed

2) **Chronus Back-Off** does **not** have a delay period

Chronus: Chronus Back-Off

Chronus refreshes all rows that exceed the back-off threshold



Chronus: Chronus Back-Off (More in the paper)

https://arxiv.org/abs/2502.12650

2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA)

Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat[§†] A. Giray Yağlıkçı[§] Geraldo F. Oliveira[§] Ataberk Olgun[§]
Nisa Bostancı[§] Ismail Emir Yuksel[§] Haocong Luo[§] Oğuz Ergin[‡†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology* [‡]*University of Sharjah*

Read disturbance in modern DRAM is an important robustness (security, safety, and reliability) problem, where repeatedly accessing (hammering) a row of DRAM cells (DRAM row) in-

ory address should *not* cause unintended side-effects on data stored in other addresses [1]. Unfortunately, with aggressive technology scaling, DRAM [2], the prevalent main memory

Aggressor Tracking Table
requires only 4 entries

Chronus Back-Off
security analysis

Outline

Background

Industry Solutions to Read Disturbance

Security Analysis of Industry Solutions

Performance Analysis of Industry Solutions

Chronus

Evaluation

Conclusion

Evaluation Methodology

- **Performance and energy consumption evaluation:** cycle-level simulations using Ramulator 2.0 [Luo+, CAL 2023] and DRAMPower [Chandrasekar+, DATE 2013]

- **System Configuration:**

Processor	4 cores, 4.2GHz clock frequency, 4-wide issue, 128-entry instruction window
DRAM	DDR5, 1 channel, 2 rank/channel, 8 bank groups, 4 banks/bank group, 64K rows/bank
Memory Ctrl.	64-entry read and write requests queues, Scheduling policy: FR-FCFS with a column cap of 4 Last-Level Cache 8 MiB (4-core)

- **Workloads:** 60 4-core workload mixes
  - SPEC CPU2006, SPEC CPU2017, TPC, MediaBench, YCSB

Evaluation Methodology: Chronus Variants

1 **Chronus**

Concurrently updates counters while serving accesses
and uses Chronus Back-Off

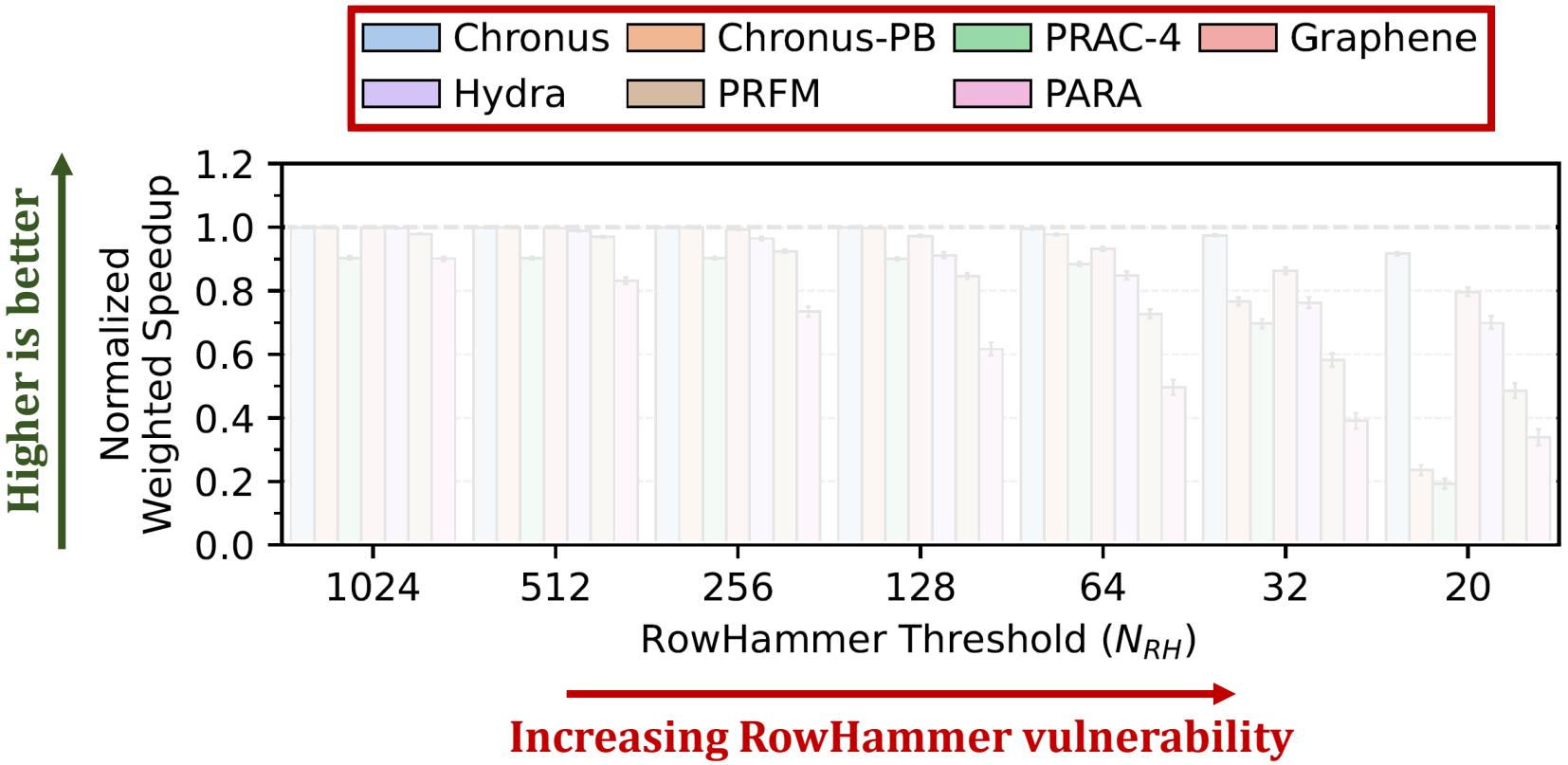
2 **Chronus-PB**

Concurrently updates counters while serving accesses
but uses PRAC Back-Off

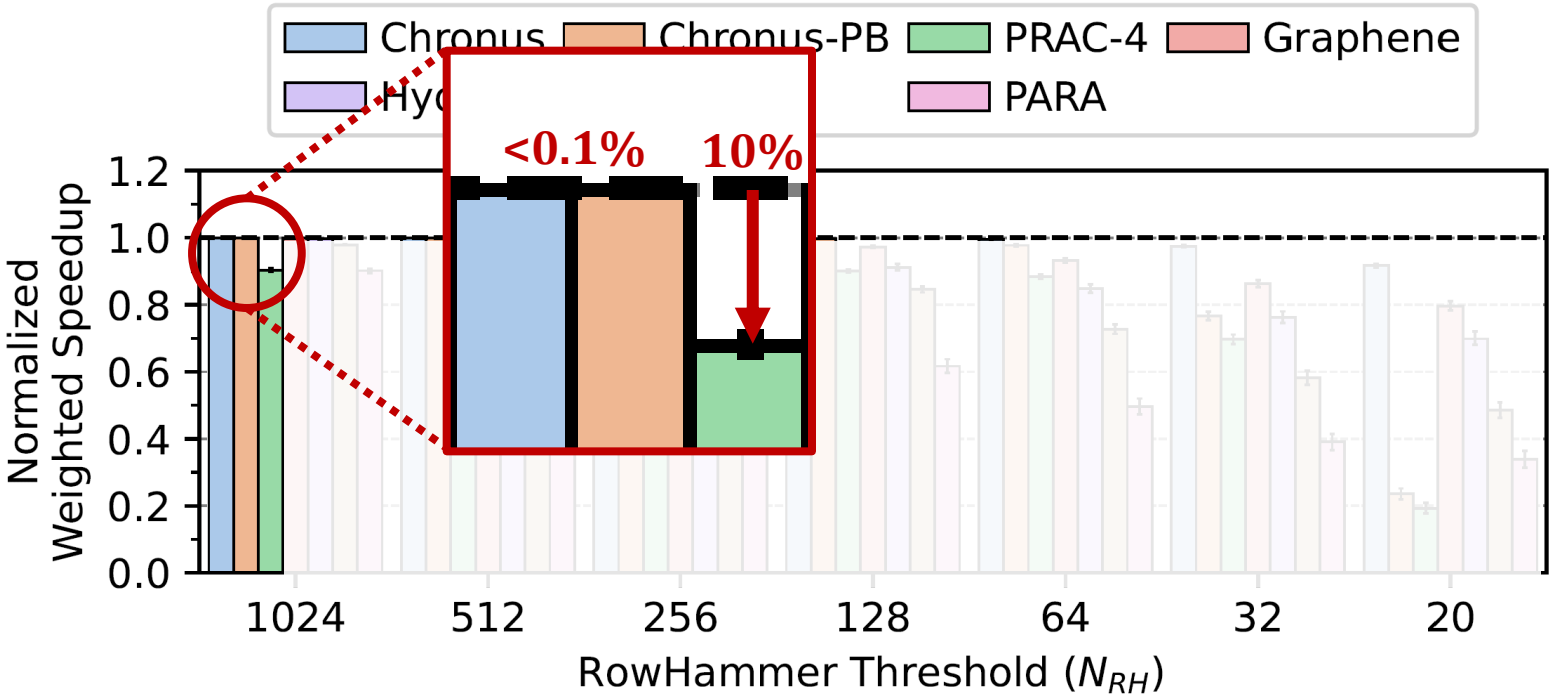
Evaluation Methodology

- **Comparison Points:** Two Chronus variants are compared against 5 state-of-the-art DRAM read disturbance mitigation mechanisms:
  - **PARA** [Kim+, ISCA 2014]
  - **Graphene** [Park+, MICRO 2020]
  - **Hydra** [Qureshi+, ISCA 2022]
  - **PRAC** [JEDEC 2024]
  - **PRFM** [JEDEC 2020]

Evaluation: System Performance and Its Scaling

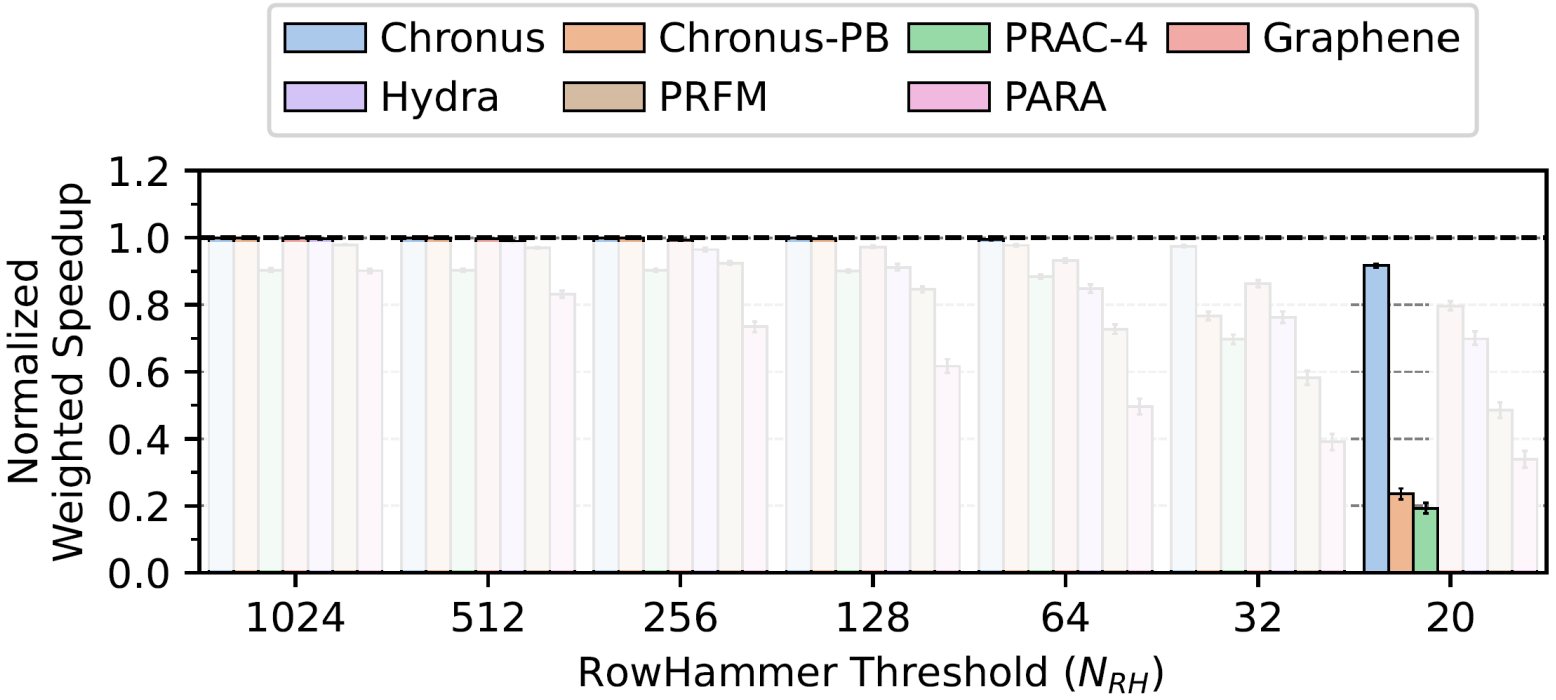


Evaluation: System Performance and Its Scaling



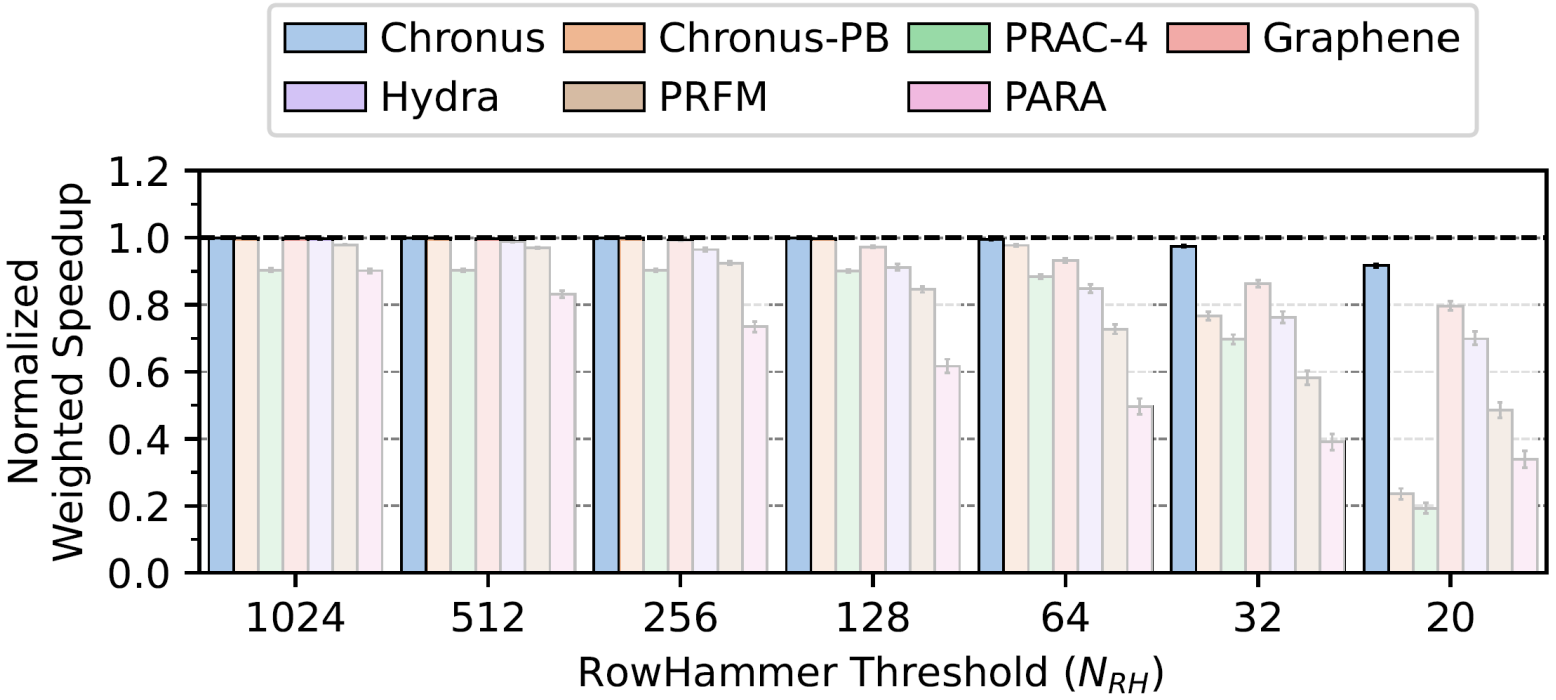
At high N_{RH} values, **Chronus** and **Chronus-PB**'s concurrent counter update mechanism prevents the high performance overhead of **PRAC**

Evaluation: System Performance and Its Scaling



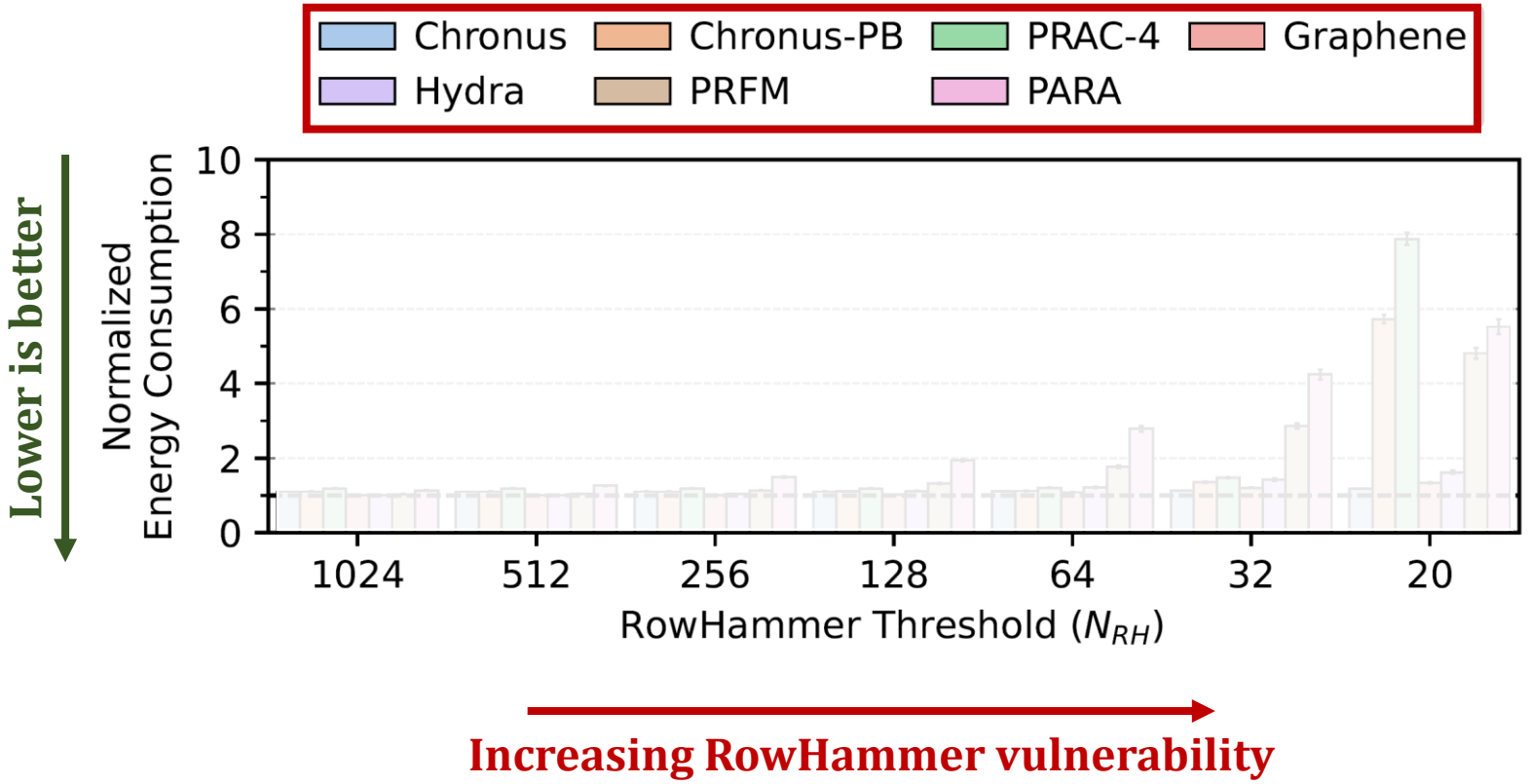
At low N_{RH} values, **Chronus Back-Off** prevents the high performance overhead due to aggressive **PRAC** configuration

Evaluation: System Performance and Its Scaling

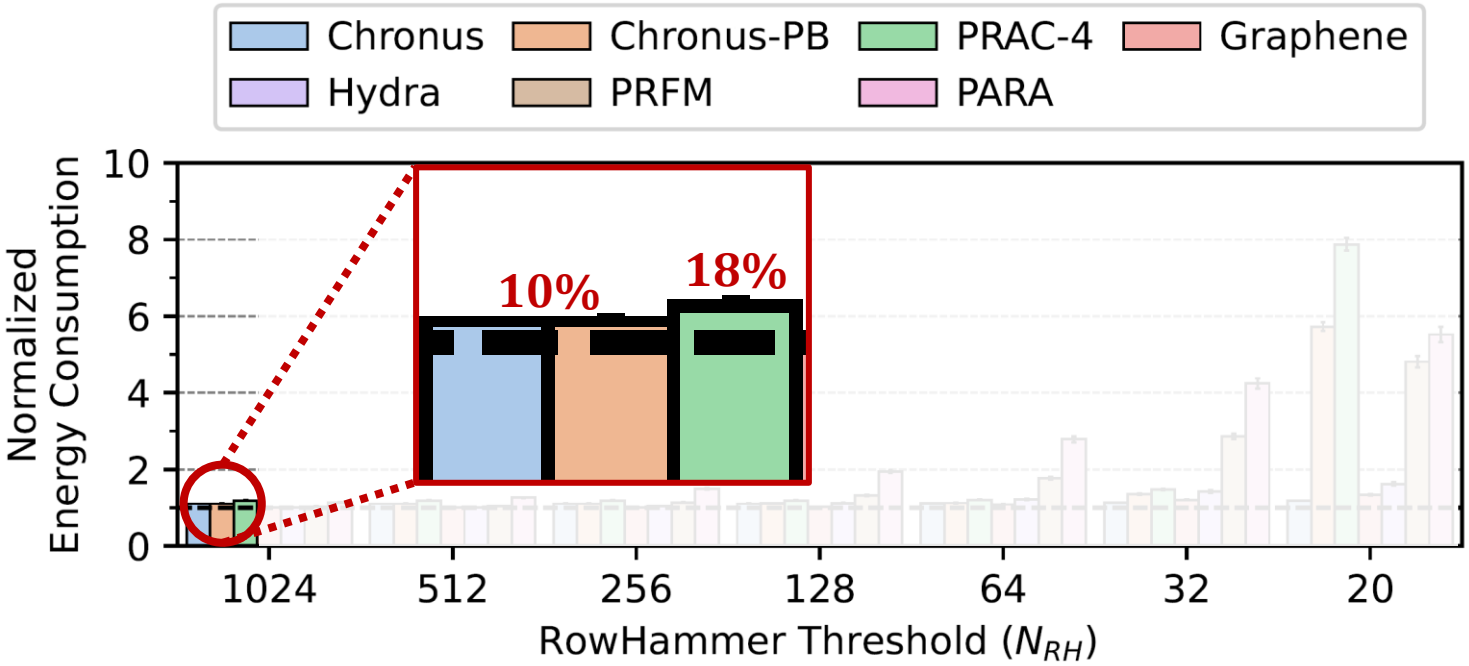


Chronus outperforms all evaluated mitigation mechanisms at all evaluated RowHammer thresholds

Evaluation: DRAM Energy and Its Scaling

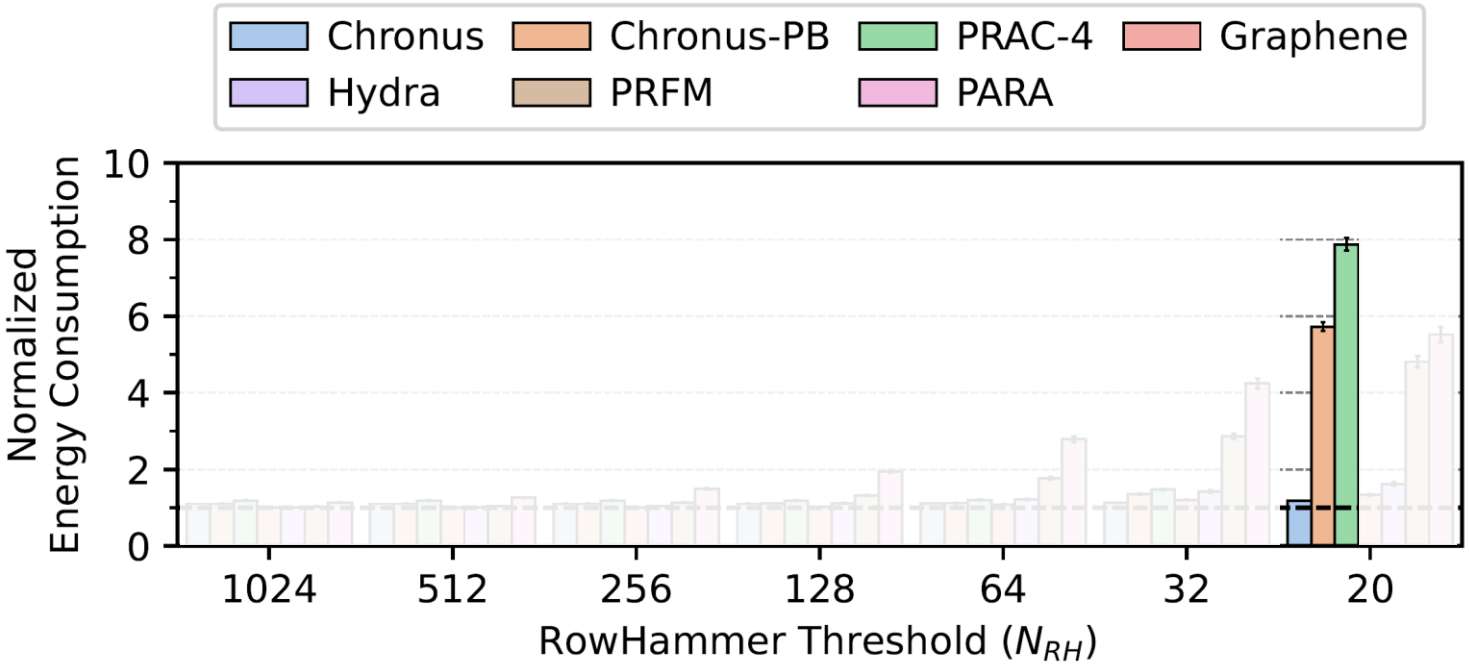


Evaluation: DRAM Energy and Its Scaling



At high N_{RH} values, **Chronus** reduces **PRAC** energy overhead by 44%

Evaluation: DRAM Energy and Its Scaling



At low N_{RH} values, **Chronus Back-Off** prevents the high energy overhead due to aggressive **PRAC** configuration

Evaluation Summary

Chronus significantly **reduces**
the **negative performance and energy overheads** of **PRAC**

1) **Chronus concurrently updates counters**
while serving DRAM accesses

2) **Chronus dynamically controls**
the number of refreshes as needed

More in the Paper

- **Detailed Background on PRAC**
  - More information on PRAC and RFM
  - Determining which rows to refresh during a back-off
- **Security Analysis of PRAC**
  - Threat Model
  - Secure Configurations
- **Details on Chronus**
  - Preventing bitflips in the counter subarray
  - Security proof
  - Hardware complexity
- **Evaluation**
  - Chronus outperforms all mechanisms in single-core workloads
  - Effect of workload memory intensity on system performance
- **System Performance Adversarial Workloads**
  - Chronus reduces PRAC's system performance overhead under a system performance adversarial workload by 66%

https://arxiv.org/abs/2502.12650

The Paper

2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA)

Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat[§†] A. Giray Yağlıkçı[§] Geraldo F. Oliveira[§] Ataberk Olgun[§]
Nisa Bostancı[§] Ismail Emir Yuksel[§] Haocong Luo[§] Oğuz Ergin[‡†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology* [‡]*University of Sharjah*

Read disturbance in modern DRAM is an important robustness (security, safety, and reliability) problem, where repeatedly accessing (hammering) a row of DRAM cells (DRAM row) in-

ory address should *not* cause unintended side-effects on data stored in other addresses [1]. Unfortunately, with aggressive technology scaling, DRAM [2], the prevalent main memory

https://arxiv.org/abs/2502.12650

Outline

Background

Industry Solutions to Read Disturbance

Security Analysis of Industry Solutions

Performance Analysis of Industry Solutions

Chronus

Evaluation

Conclusion

Conclusion

We rigorously analyzed and characterized the security and performance implications of recently introduced industry solutions to DRAM read disturbance

Mathematical analysis & extensive simulations show that: PRAC

- Has significant (10%) performance overhead for modern DRAM chips because **PRAC** requires **additional time to update row activation counters**
- **Poorly scales** to future DRAM chips that are more vulnerable to read disturbance because **PRAC** is vulnerable to an adversarial access pattern (i.e., the wave attack)

Chronus: Solves PRAC's two major weaknesses by

- **Concurrently updating counters** while serving accesses
- **Securing PRAC** against a potential wave attack

Key Results: Chronus

- Significantly reduces the negative performance and energy overheads of **PRAC** for both modern and future DRAM chips that are more vulnerable to read disturbance
- Outperforms five state-of-the-art academic and industry solutions in terms of system performance and energy

Open Source and Artifact Evaluated

CMU-SAFARI / Chronus

Type / to search

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

Chronus Public

Edit Pins | Watch 3 | Fork 0 | Star 1

master | 1 Branch | 0 Tags | Go to file t | Add file | Code

kirbyydoge Add and update camera ready plotting scripts — ffdb2fb · last month — 8 Commits

ae_results	Add and update camera ready plotting scripts	last month
mixes	Add plotting scripts and update CPU trace zenodo link	2 months ago
plotting_scripts	Add and update camera ready plotting scripts	last month
scripts	Add missing figures, update configurations and pre-finished...	2 months ago
src	Initial commit	3 months ago
.gitattributes	Initial commit	3 months ago

About

No description, website, or topics provided.

Readme
Activity
Custom properties
1 star
3 watching
0 forks
Report repository

Releases

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

Chronus
Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat

Giray Yağlıkçı Geraldo Oliveira Ataberk Olgun

Nisa Bostancı İsmail Emir Yüksel Haocong Luo

Oğuz Ergin Onur Mutlu

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

SAFARI ETH zürich kasırga

Chronus
Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance

BACKUP SLIDES

Oğuzhan Canpolat

Giray Yağlıkçı Geraldo Oliveira Ataberk Olgun

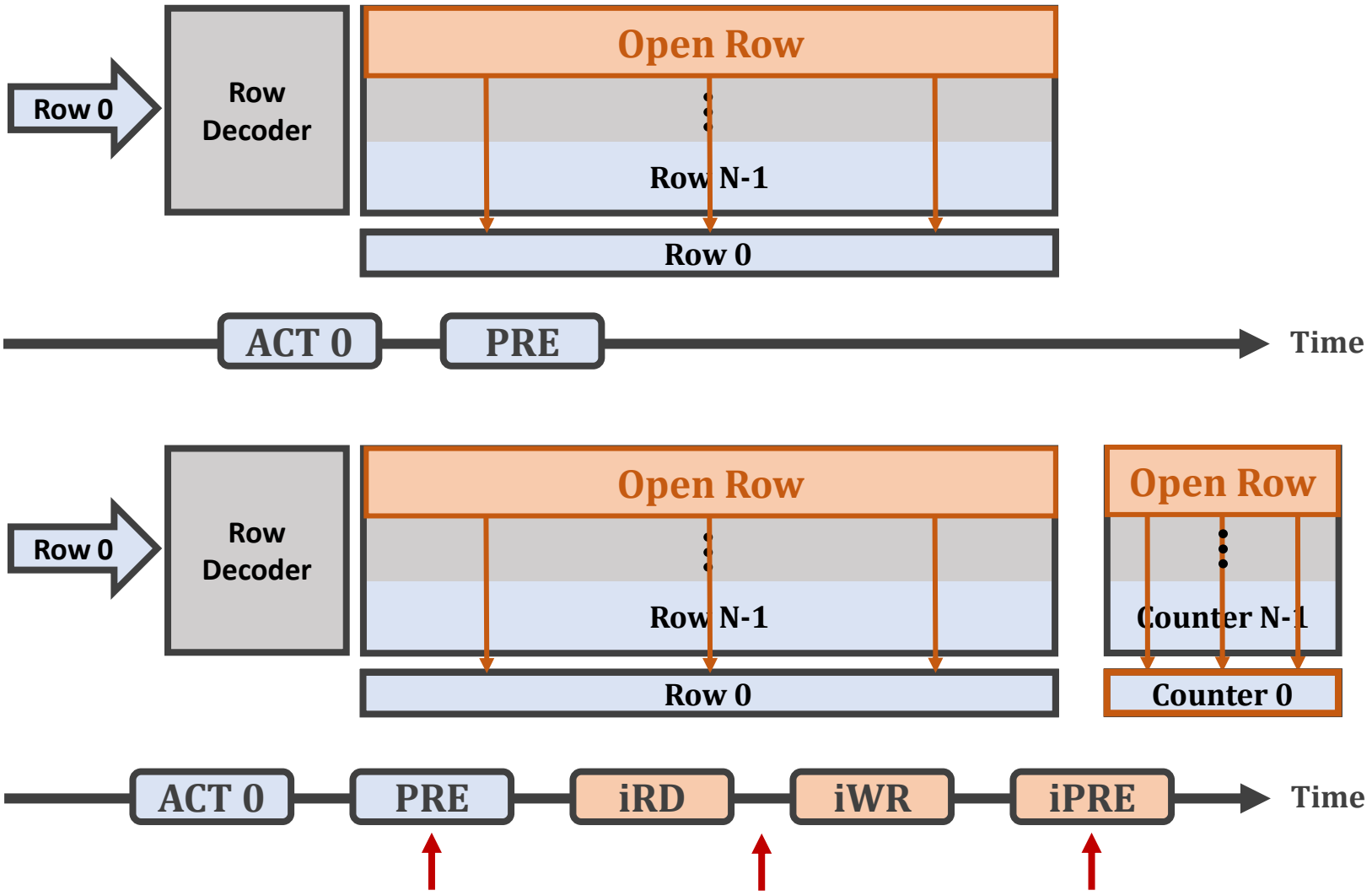
Nisa Bostancı İsmail Emir Yüksel Haocong Luo

Oğuz Ergin Onur Mutlu

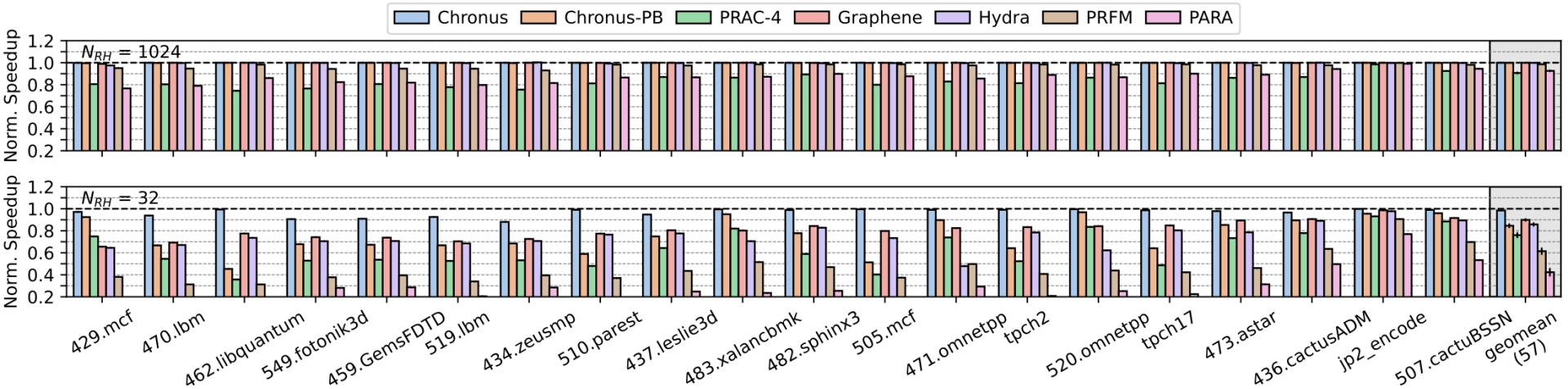
https://github.com/CMU-SAFARI/Chronus

SAFARI ETH zürich kasırga

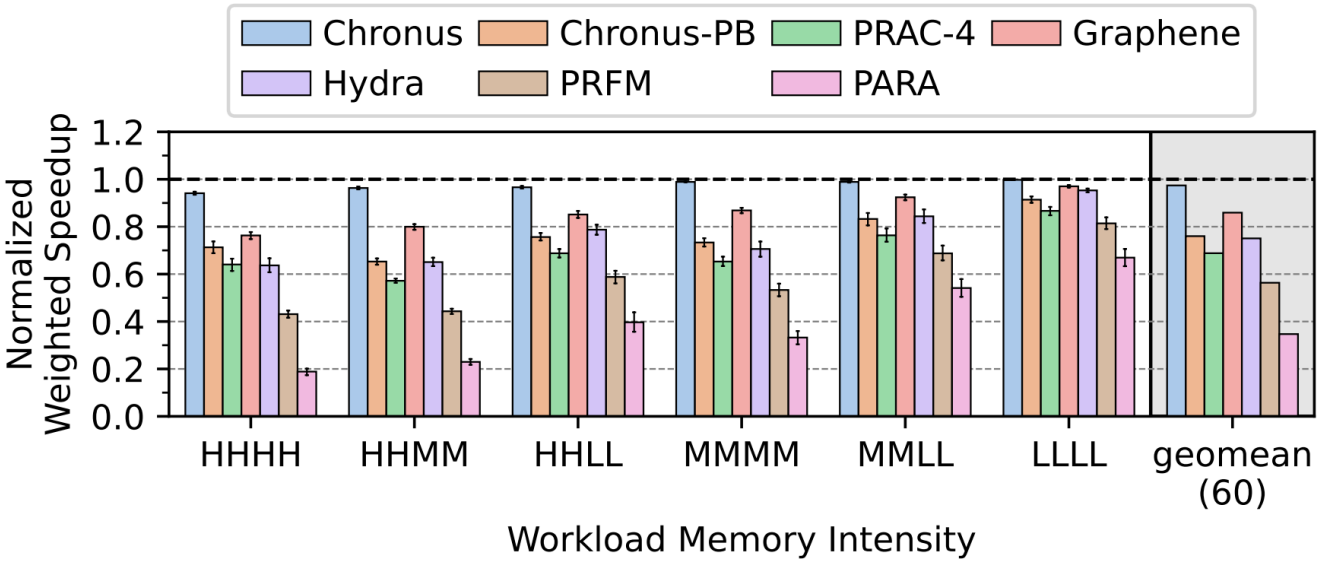
PRAC Counter Update



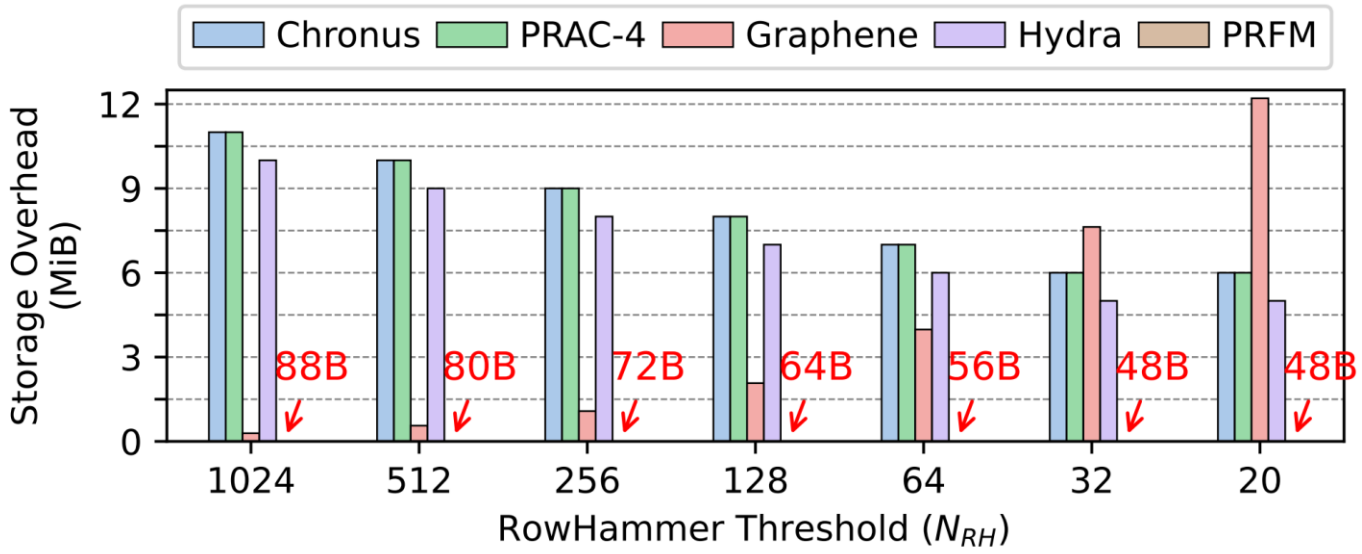
Evaluation: Singe-core System Performance



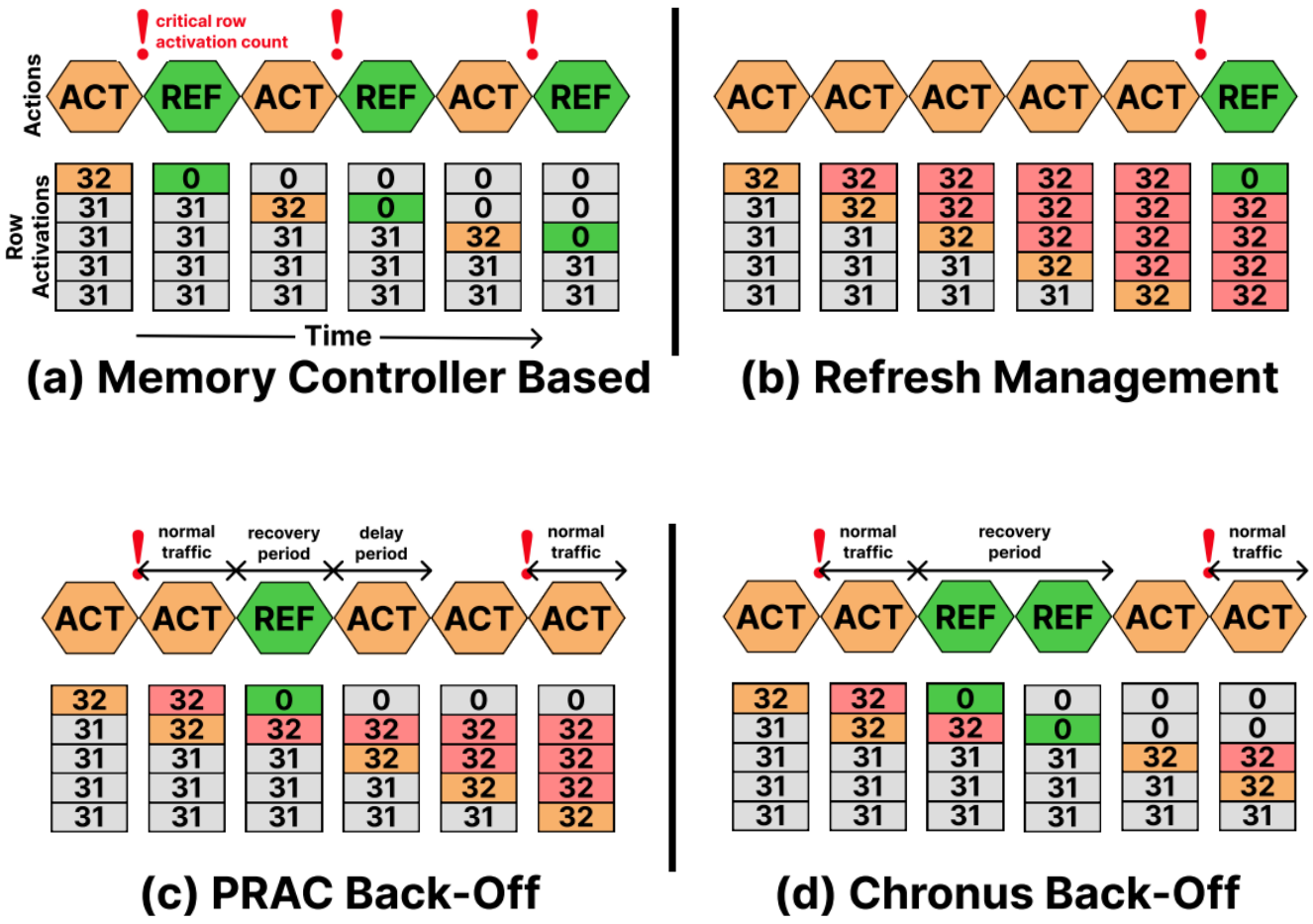
Evaluation: Effect of Workload Memory Intensity (N_{RH}=32)



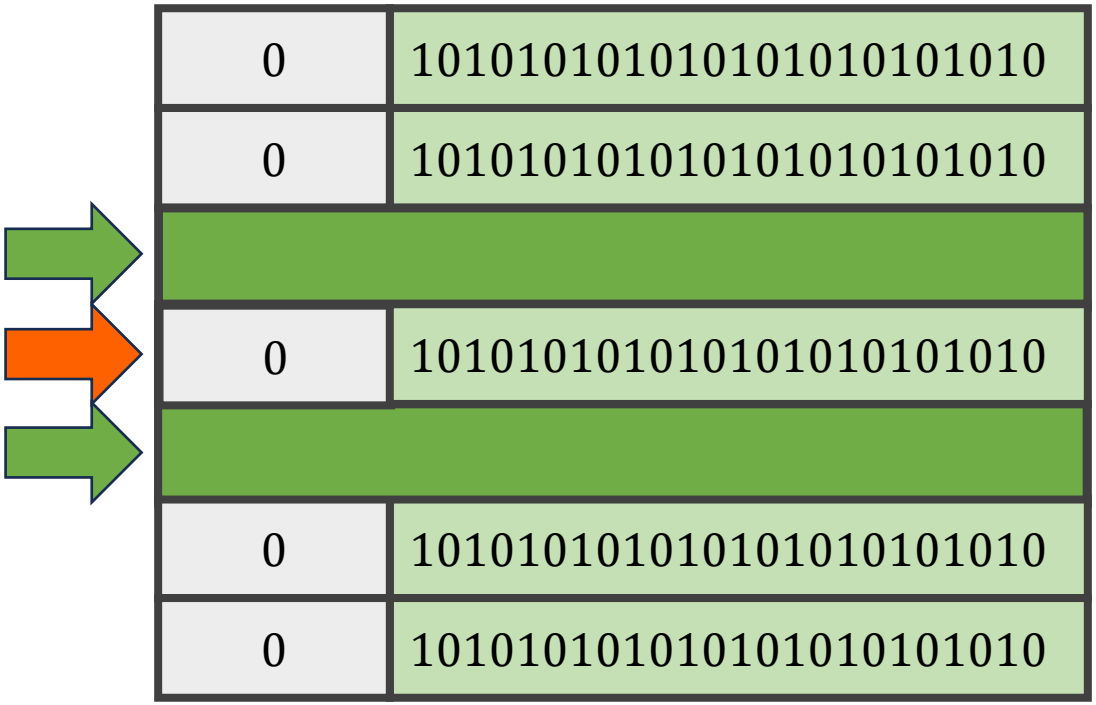
Evaluation: Storage Overhead



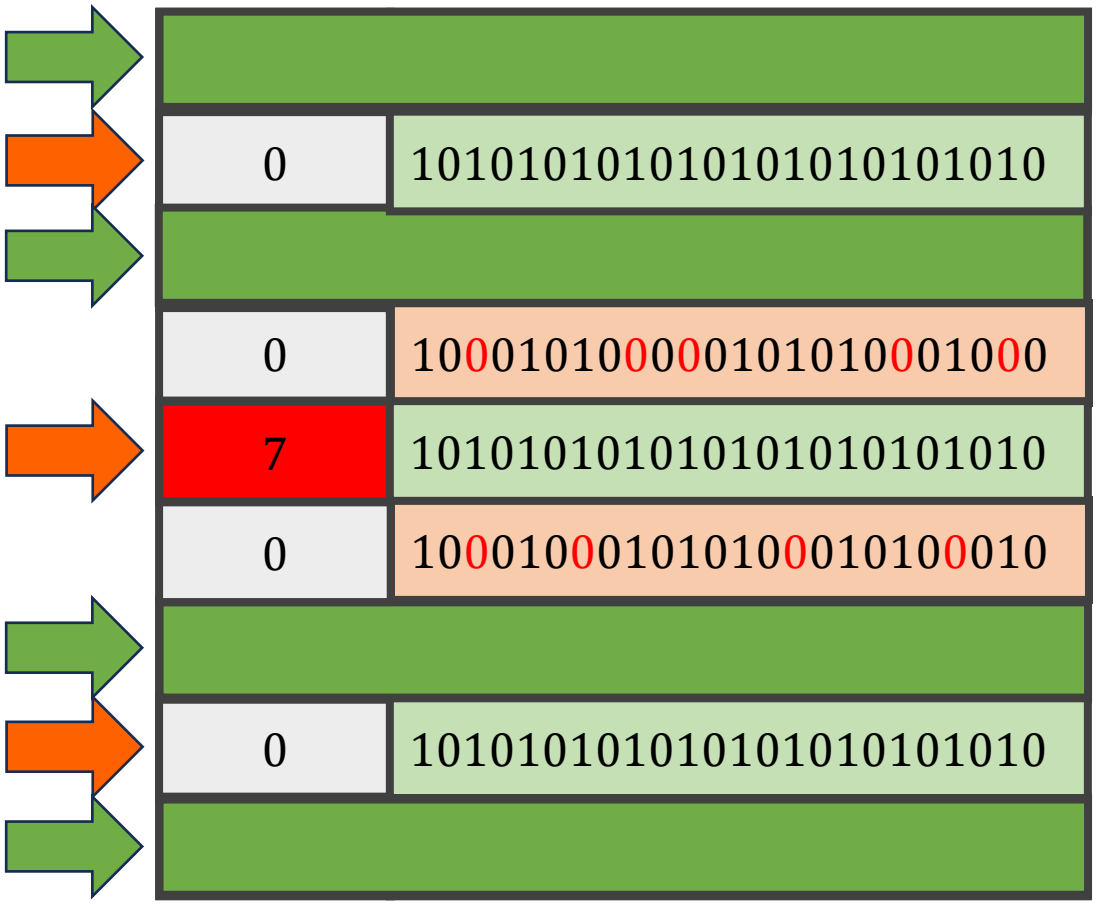
Wave Attack Visualization



Security Analysis: Wave Attack (I)



Security Analysis: Wave Attack (II)



Secure configuration is **not trivial**