Ramulator 2

Presenter: Haocong Luo

PhD Student, ETH Zurich

SAFARI Research Group

SAFARI

ETH zürich

Motivation

- **Robustness issues** in DRAM
  - Data retention
  - Read disturbance (RowHammer, RowPress, etc.)
  - …
- **Performance issues** in main memory system
  - Performance overhead analysis of read disturbance mitigation techniques
  - Processing-in-Memory architectures
  - Emerging memory technologies
  - …

DRAM simulation infrastructures is needed to understand, characterize, and evaluate the robustness and performance of DRAM

Executive Summary

Ramulator 2.0: Modern, modular, and extensible DRAM & memory system simulator

- Fine-grained modeling of DRAM operation (cycle-level)
- Unified functional and timing modeling of DRAM based on hierarchical state-machines
- Modular and extensible software architecture
- Models a wide range of DRAM standards and memory controller functionalities
- Used in a wide range of research works

Outline

Outline

Ramulator 2.0

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

Abstract—We present Ramulator 2.0, a highly modular and extensible DRAM simulator that enables rapid and agile implementation and evaluation of design changes in the memory controller and DRAM to meet the increasing research effort in improving the performance, security, and reliability of memory systems. Ramulator 2.0 abstracts and models key components in a DRAM-based memory system and their interactions into shared *interfaces* and independent *implementations*. Doing so enables easy modification and extension of the modeled functions of the memory controller and DRAM in Ramulator 2.0. The DRAM specification syntax of Ramulator 2.0 is concise and human-readable, facilitating easy modifications and extensions. Ramulator 2.0 implements a library of reusable templated lambda functions to model the functionalities of DRAM commands to simplify the implementation of new DRAM standards, including DDR5, LPDDR5, HBM3, and GDDR6. We showcase Ramulator 2.0's modularity and extensibility by implementing and evaluating a wide variety of RowHammer mitigation techniques that require *different* memory controller design changes. These techniques are added modularly as separate implementations *without* changing *any* code in the baseline memory controller implementation. Ramulator 2.0 is rigorously validated and maintains a fast simulation speed compared to existing cycle-accurate DRAM simulators. Ramulator 2.0 is open-sourced under the permissive MIT license at https://github.com/CMU-SAFARI/ramulator2.

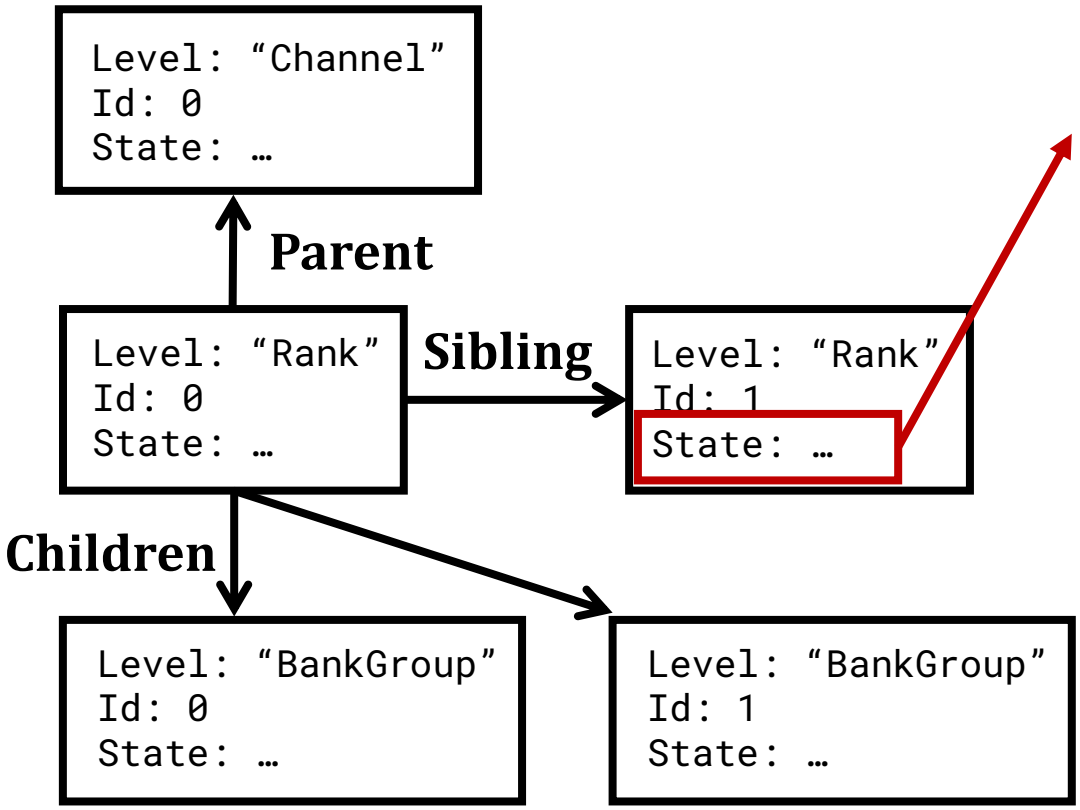
IEEE CAL Paper

Open-source version Github repo:
CMU-SAFARI/ramulator2

Outline

Ramulator 2.0 Design and Features (I)

- **Hierarchical state-machine** based modeling of DRAM



State of a DRAM node:

- **Current state:** open, close, activating, etc.
- **Timing constraints:** Earliest time in the future that each DRAM command is allowed to be issued
- **Energy & power:** Time spent in each state and the number of DRAM commands served (DRAMPower model)
- **Can be extended to include more**

Ramulator 2.0 Design and Features (II)

- DRAM commands implemented as lambda functions that hierarchically traverses and updates the states of the nodes
  1. Checks the current states of the nodes to decode which DRAM command to issue
  2. Programmatically apply state changes
  3. Updates the timing constraints, power metrics, etc.
- Templated library of generalized DRAM command lambda functions allow reuse of command implementations across different DRAM standards

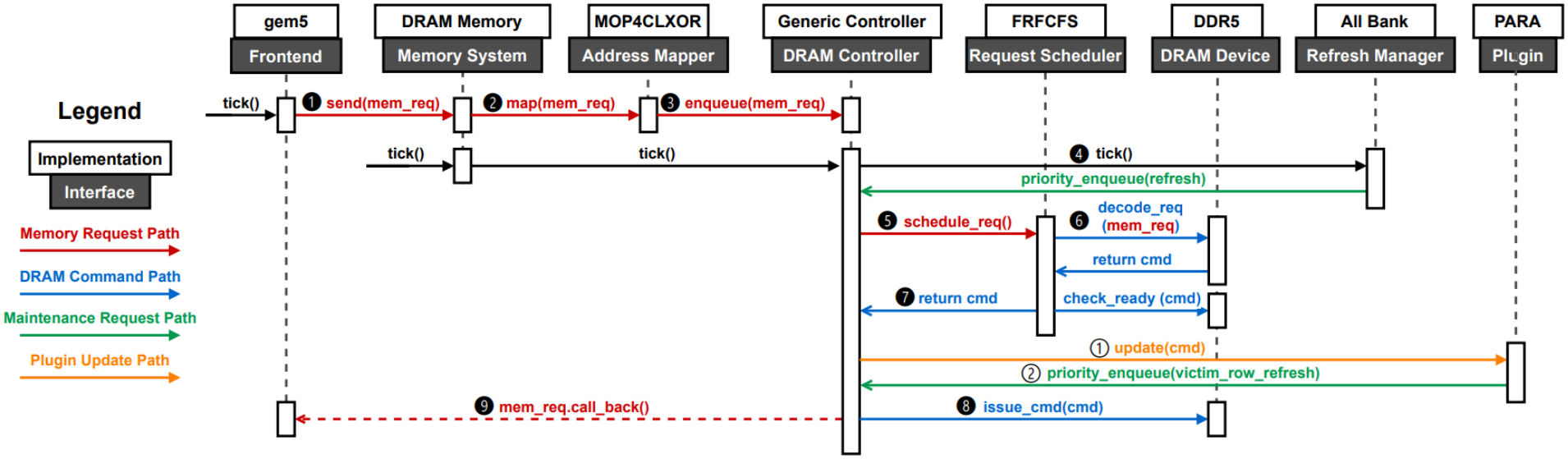
```
template <class DRAM_t>
int RequireAllBanksClosed(typename DRAM_t::Node* node,
     int cmd, int target_id, Clk_t clk) {
  // for all banks {
  // ...
    if (bank->m_state == DRAM_t::m_states["Closed"]) {
      continue;
    } else {
      return T::m_commands["PREab"];
    }
  // }
  return cmd;
};
```

Example DRAM Command Decode Function

Applicable to:
DDR3, DDR4, DDR5
LPDDR4, LPDDR5
HBM (1/2/3), GDDR6

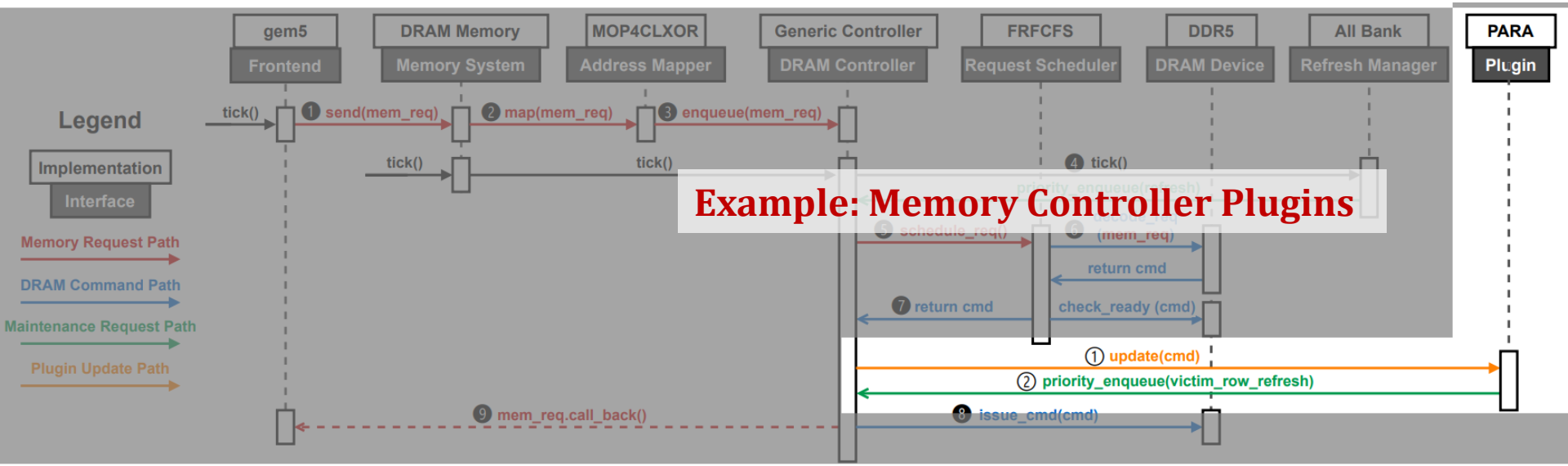
Ramulator 2.0 Design and Features (III)

- Modular and extensible software architecture
  - All components in the memory system modeled with the same interface and different implementations
  - Example: The memory controller include:
    - Address Mapper, Request Scheduler, Refresh Controller, Row Policy, etc.
    - Each can be flexibly changed without hardcoding other parts



Ramulator 2.0 Design and Features (IV)

- Modular and extensible software architecture
  - All components in the memory system modeled with the same interface and different implementations
  - Example: The memory controller include:
    - Address Mapper, Request Scheduler, Refresh Controller, Row Policy, etc.
    - Each can be flexibly changed without hardcoding other parts



Ramulator 2.0 Design and Features (V)

- More in the paper
  - More detailed explanation of modeling methodology
  - Authoring of DRAM specifications (organization, timings, etc.)
  - Memory controller plugin & RowHammer mitigations
  - Performance comparison with other DRAM simulators

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

IEEE CAL Paper

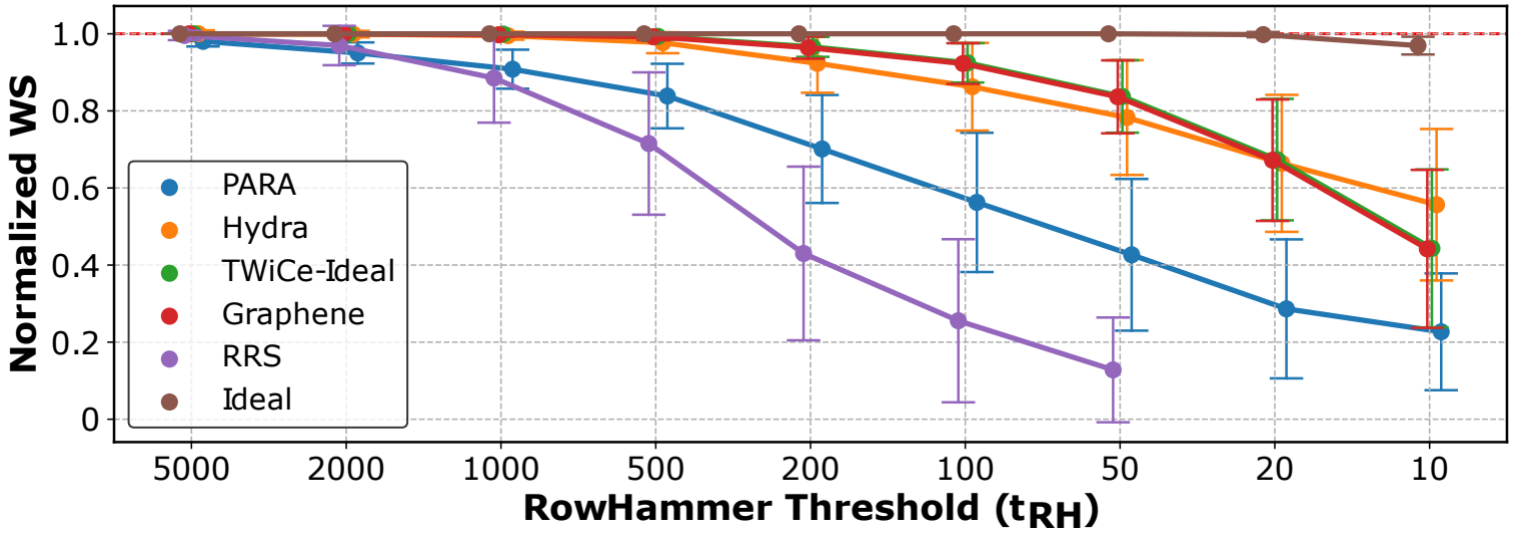
Open-source version Github repo:
CMU-SAFARI/ramulator2

Outline

Outline

Ramulator 2.0 Case Studies (I)

- **Cross-section performance overhead evaluation of different RowHammer mitigation techniques [Luo+, IEEE CAL 2023]**
  - Six different RowHammer mitigation techniques all implemented as plugins to the same memory controller implementation



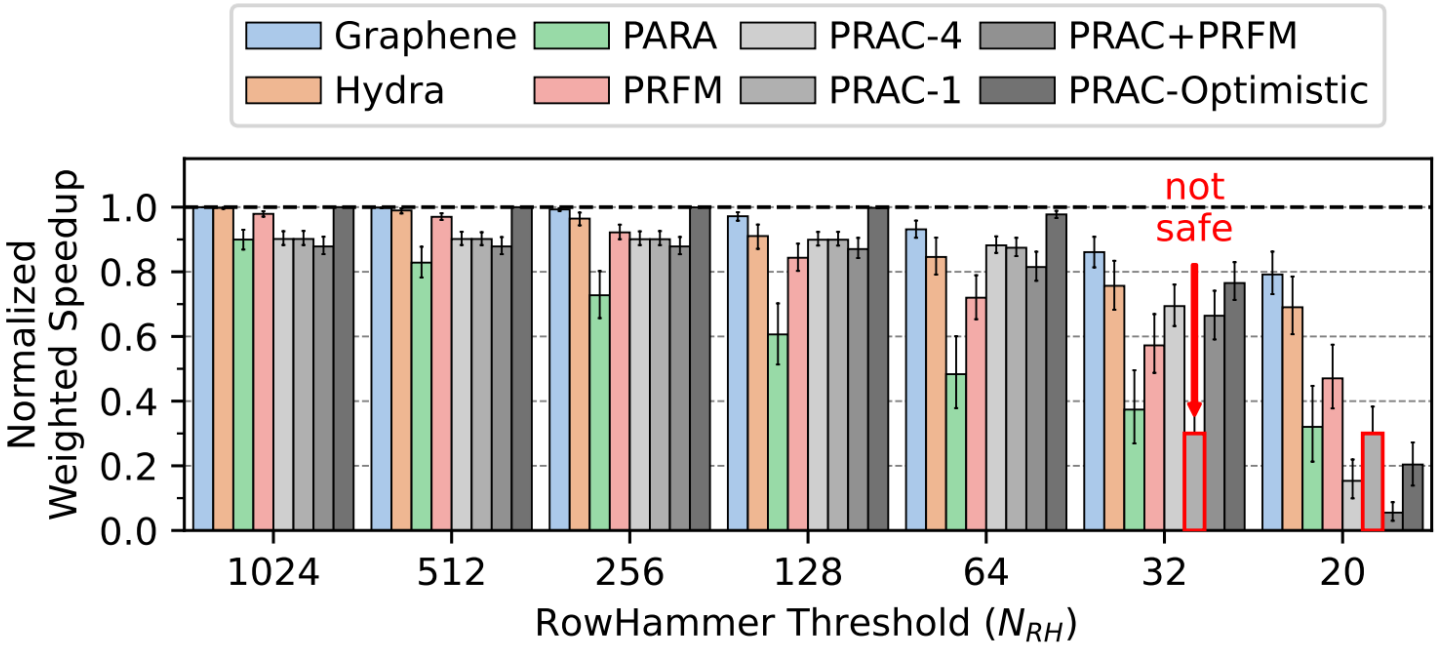
Outline

Ramulator 2.0 Case Studies (II)

- **New Read Disturbance Mitigation Features in DDR5**
  - **RFM Command:** New DRAM command that gives the memory controller a longer time window so that the DRAM can refresh potential victim rows
  - **Per Row Activation Counting (PRAC):** In-DRAM *per-row* activation counter (implemented as extra columns of cells)
- **PRAC Workflow**
  - Row activations increment the PRAC counters
  - If the counter value reaches a critical threshold, the DRAM send a *back-off signal* to the memory controller
  - Upon receiving the back-off signal, the memory controller send RFM commands to refresh the potential victim rows

Ramulator 2.0 Case Studies (III)

- **Performance evaluation of DDR5 Per Row Activation Counting (PRAC) [Canpolat+, DRAMsec'24]**
  - Memory controller implementation extended with support for per-row activation count tracking and back-off signal



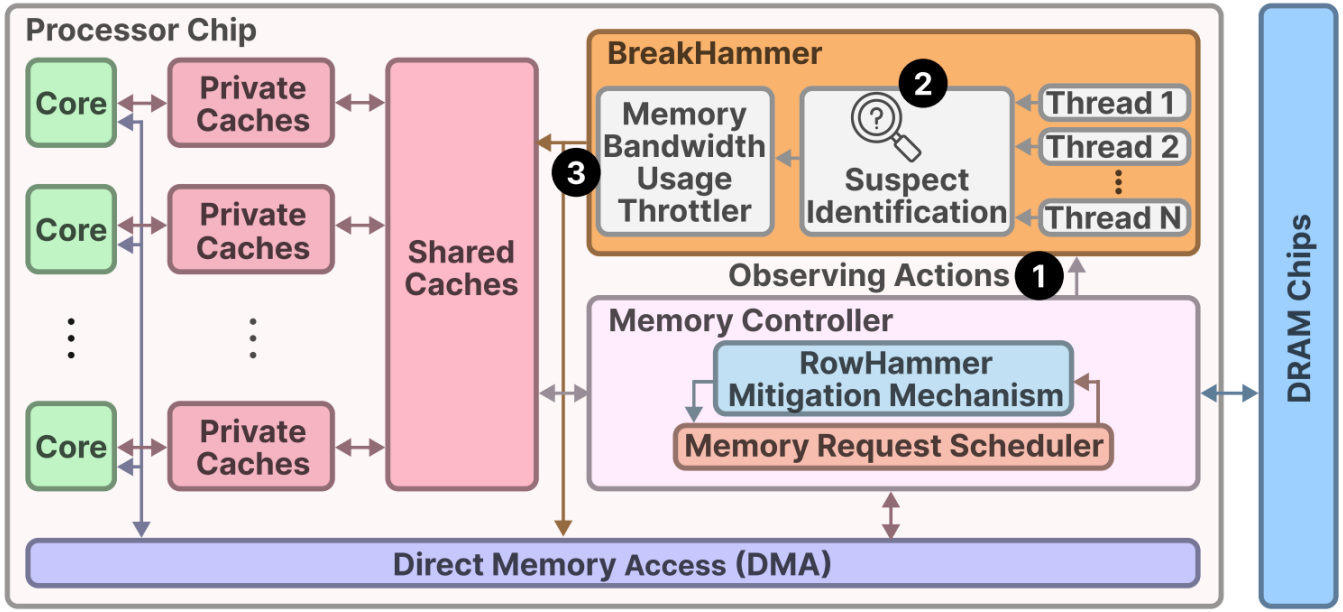
Outline

Ramulator 2.0 Case Studies (IV)

- **BreakHammer: Enhancing RowHammer Mitigations by Carefully Throttling Suspect Threads [Canpolat+, MICRO'24]**
  - **Goal:** Reduce the performance overhead of RowHammer mitigation mechanisms by carefully reducing the number of performed RowHammer-preventive actions without compromising system robustness
  - **Key Idea:** Limit the dynamic memory request count of a hardware thread based on how frequently the thread triggers RowHammer-preventive actions

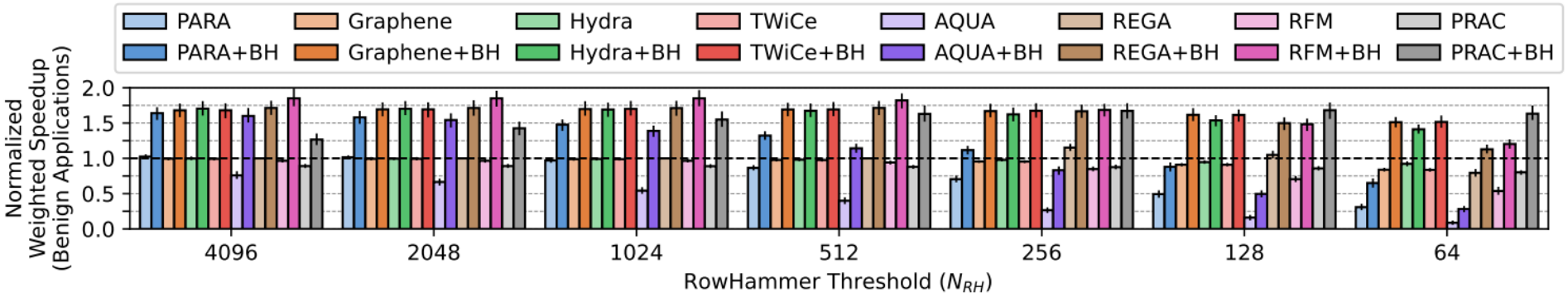
Ramulator 2.0 Case Studies (V)

- **BreakHammer: Enhancing RowHammer Mitigations by Carefully Throttling Suspect Threads [Canpolat+, MICRO'24]**
  - **Key Mechanism:** 1) Observe the triggered RowHammer preventive actions, 2) identify suspect threads, and 3) reduce the request count of the suspect threads



Ramulator 2.0 Case Studies (VI)

- **BreakHammer: Enhancing RowHammer Mitigations by Carefully Throttling Suspect Threads [Canpolat+, MICRO'24]**
  - BreakHammer's performance scaling for existin RowHammer mitigation mechanisms with an attacker present



Outline

Conclusion & Future Work

Ramulator 2.0: Modern, modular, and extensible DRAM & memory system simulator

- ❑Fine-grained modeling of DRAM operation (cycle-level)
- ❑Models a wide range of DRAM standards and memory controller functionalities
- ❑Used in a wide range of research works

Ongoing & Future Works

- ❑Unit & regression test coverage
- ❑More DRAM standards and emerging technologies
- ❑More detailed memory controller modeling (i.e., pipelined scheduler and gear ratio)
- ❑Generalizable modeling for PuM/PnM architectures
- ❑…

DRAM Simulation and Testing Infrastructures

Presenter: Haocong Luo

Ramulator 2 Paper

Github repo: CMU-SAFARI/ramulator2

SAFARI

ETH zürich