ETH zürich

SAFARI

BreakHammer

Enabling Scalable and Low Overhead RowHammer Mitigations via Throttling Preventive Action Triggering Threads

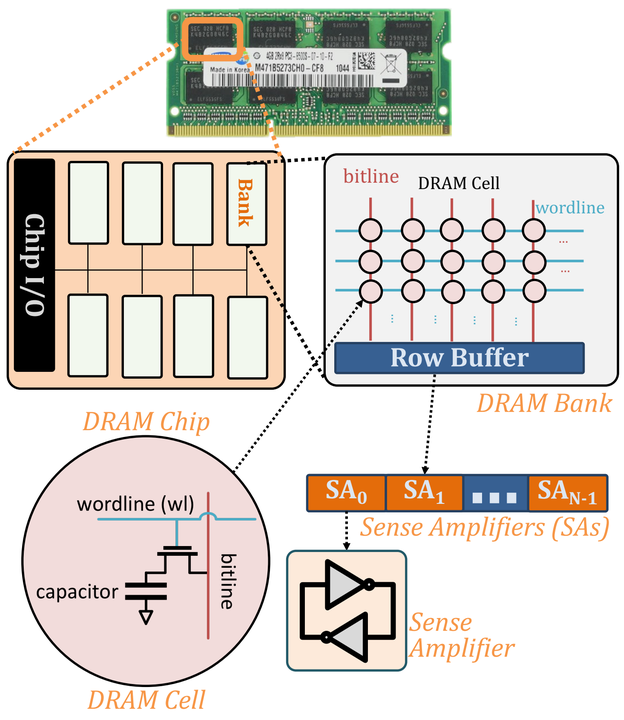
Contact Info

Oğuzhan Canpolat A. Giray Yağlıkçı Ataberk Olgun İsmail Emir Yüksel
Yahya Can Tuğrul Konstantinos Kanellopoulos Oğuz Ergin Onur Mutlu

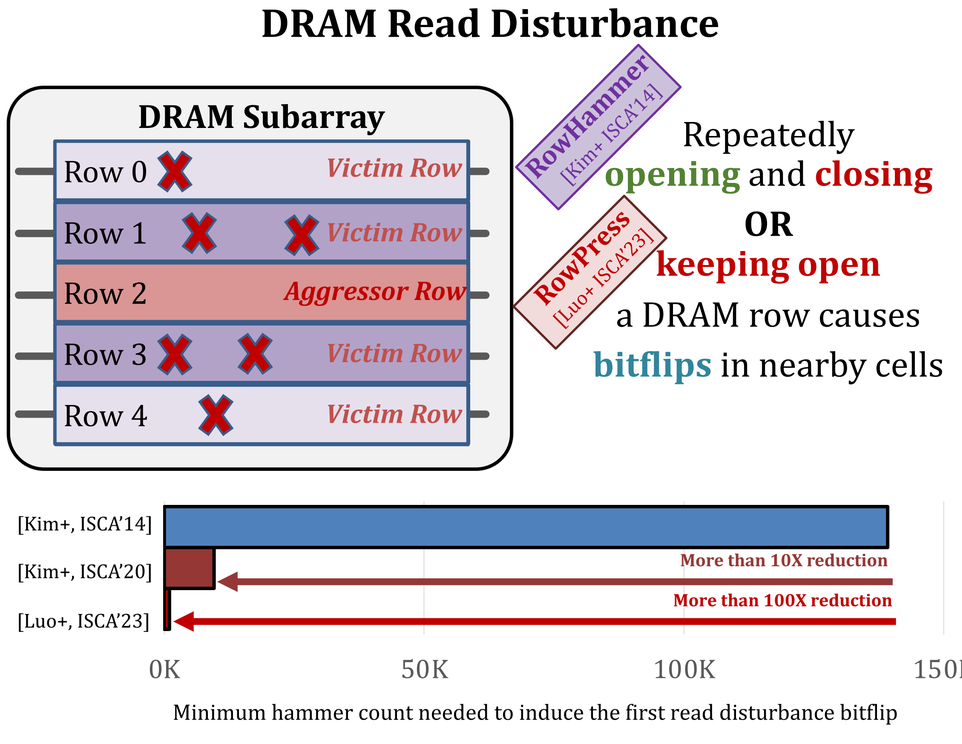
Full Paper

1. DRAM Background

DRAM Organization



DRAM Read Disturbance

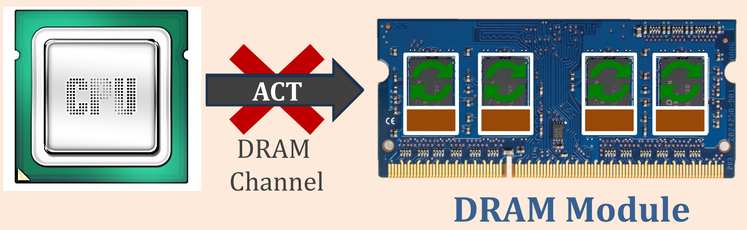


Repeatedly **opening** and **closing** **OR** **keeping open** a DRAM row causes **bitflips** in nearby cells

2. Problem and Motivation

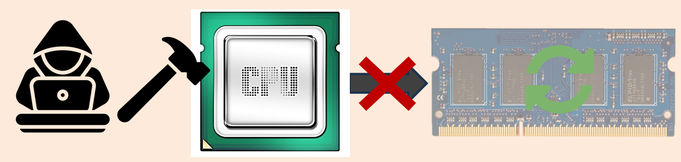
Read Disturbance Mitigation Challenges

Preventive refresh is a **blocking** operation



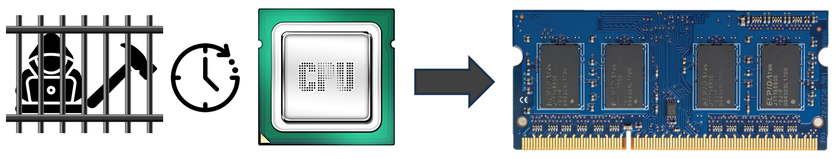
No memory access during a preventive refresh

An attacker can trigger **many** preventive actions to **block** access to main memory



Read disturbance mitigation is a potential **attack vector** to **reduce availability** of the main memory

3. Key Idea



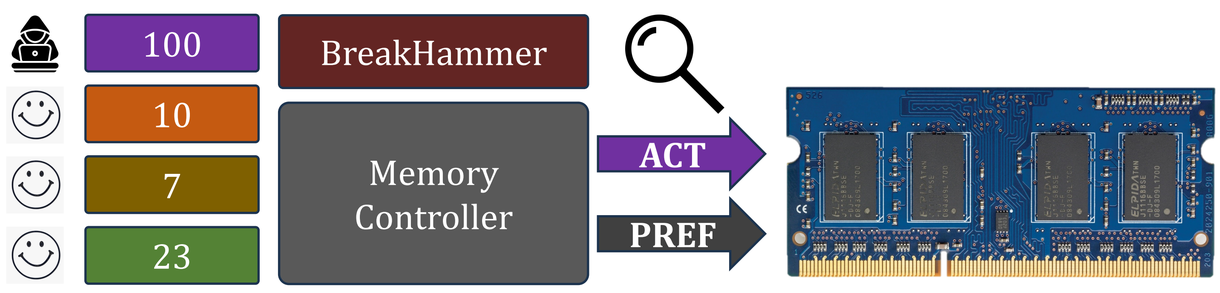
Detect and **slow down** attackers that trigger **many** preventive actions

4. Key Mechanism: BreakHammer

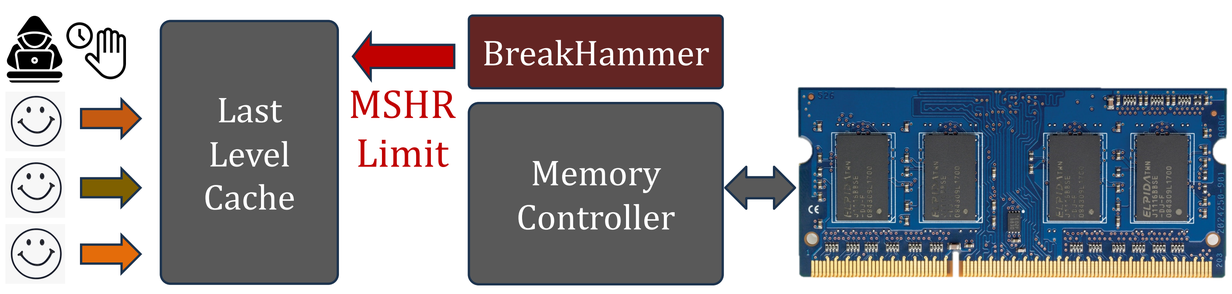
BreakHammer **breaks RowHammer access patterns** to reduce the cost of DRAM maintenance
BreakHammer is **analogous to breakwater**, a structure built in coastal areas to break waves to reduce the cost of dock maintenance

5. BreakHammer Operation

Counts the **number of times** a **hardware thread triggers** a RowHammer-preventive actions (e.g., preventive refreshes)



Slows down suspect threads by reducing the number of miss status holding registers (MSHR) they can allocate in the last level cache (LLC)



6. Performance and Energy Evaluation

Methodology

Cycle-level simulations using Ramulator 2.0 [Luo+, CAL 2023] DRAMPower [Chandrasekar+, DATE 2013]

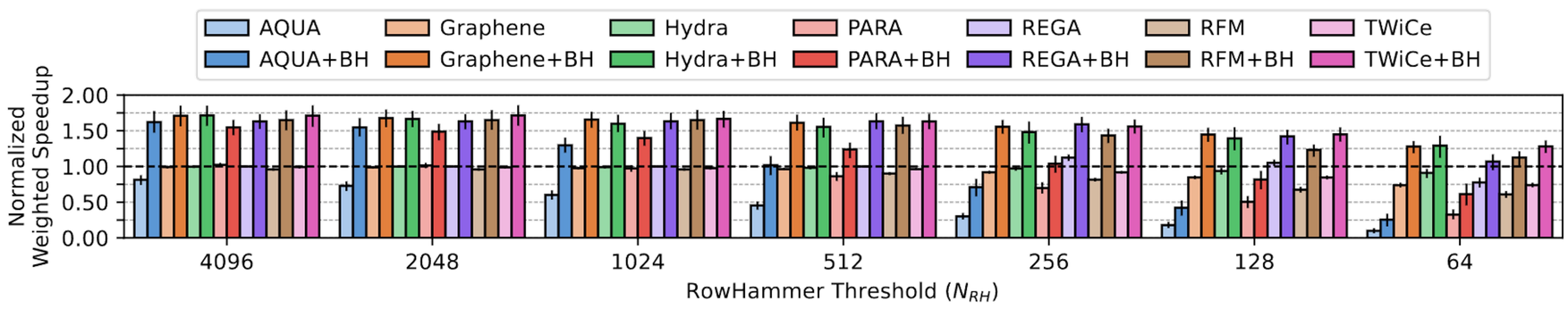
System Configuration:

Processor
4 cores, 4.2GHz clock frequency,
4-wide issue, 128-entry instruction window

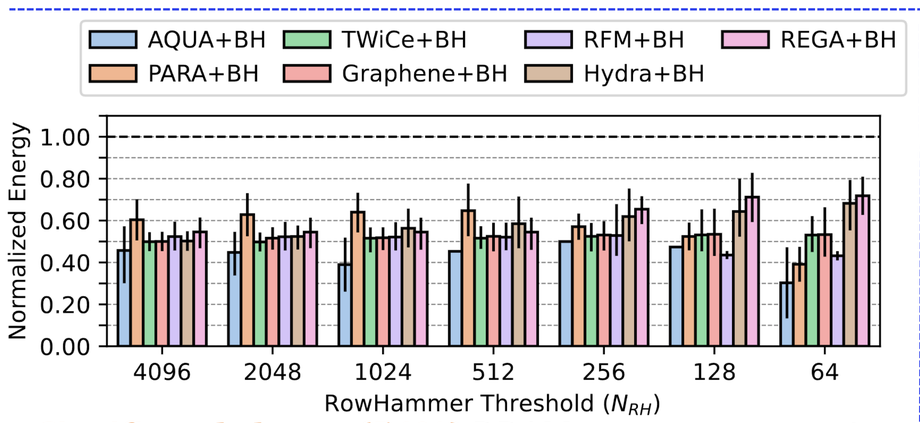
DRAM
DDR5, 1 channel, 2 rank/channel,
8 bank groups x 4 banks x 64K rows

Memory Ctrl.
64-entry read and write requests queues, FR-FCFS scheduling with a column cap of 4
Last-Level Cache 8 MiB (4-core)

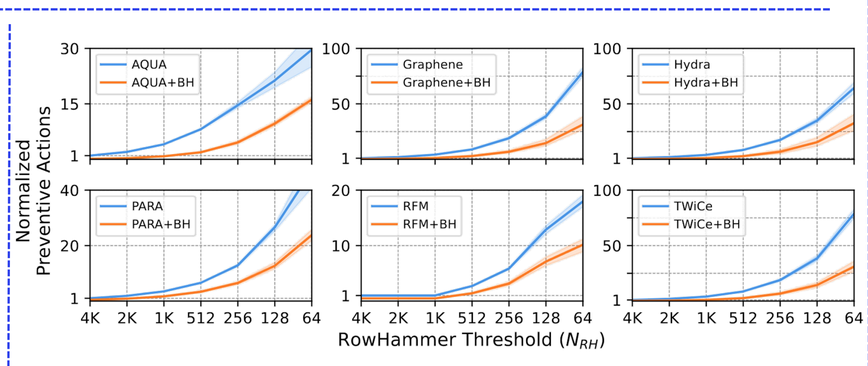
Workloads: From SPEC CPU2006, SPEC CPU2017, TPC, MediaBench, YCSB: 1) 60 mixes all benign and 2) 60 mixes with 3 benign + 1 attack



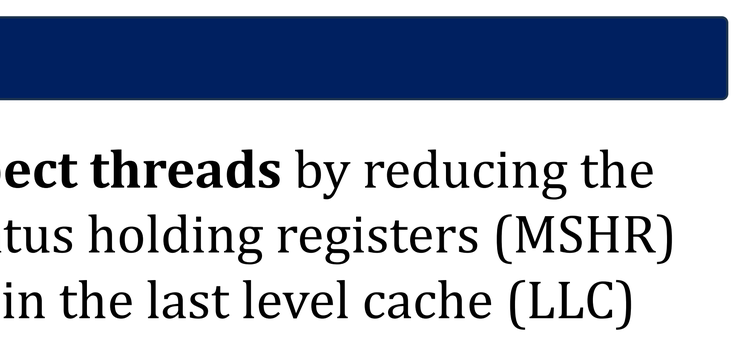
Significantly increases (by 40%) system performance, exceeding baseline with no mitigation



Significantly lower (54%) DRAM energy consumption



Significantly reduces (by 66%) the number of preventive refreshes performed



Smaller memory latency for **benign workloads**

7. Conclusion

Problem:
- DRAM continues to become more vulnerable to RowHammer
- RowHammer-preventive actions block memory accesses
  - Significant performance overheads
  - A new performance attack vector

Goal:
Reduce the performance overhead of mitigation mechanisms by reducing the number of preventive operations without compromising their security guarantees

Mechanism:
Slows down the threads that frequently trigger RowHammer-preventive refreshes

Key Results:
- Fewer preventive actions performed across all mitigations
- Significantly higher system performance
- Significantly lower DRAM energy consumption
- No performance/energy overhead for benign applications
- Near zero hardware complexity