

Tools for Evaluating Memory and Memory-Centric Computing **DRAM, Storage, Virtual Memory**

Onur Mutlu

omutlu@gmail.com

<https://people.inf.ethz.ch/omutlu>

6 July 2026

3rd Ramulator & DRAM Bender Tutorial @ ICS 2026

SAFARI

ETH zürich

Tutorial on DRAM Bender & Ramulator

Tutorial on Ramulator & DRAM Bender: Cutting-Edge Infrastructures for Real and Future Memory System Evaluation

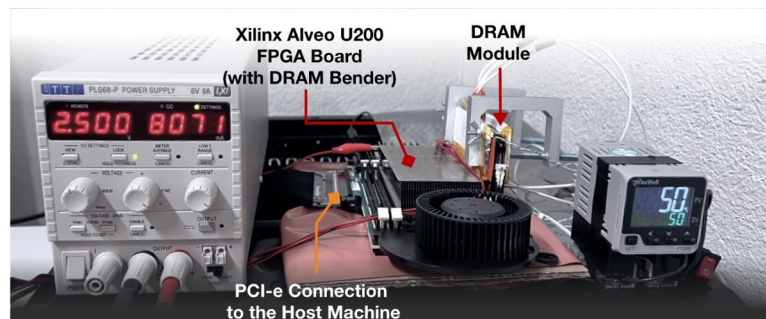


R&DB:

**A half-day tutorial @ ASPLOS 2026, Pittsburgh, USA
22nd March 2026 (Saturday)**

Organizers: Nisa Bostanci, Ataberk Olgun, Ismail Yuksel, Haocong Luo, Prof. Onur Mutlu

- *Evaluating memory performance, emerging memory technologies, and architectural mechanisms*
- *Real DRAM device characterization studies*
- *New features, extensions, and enhancements*
- *Cross-layer research that combines memory system simulation with real-chip DRAM characterization*



**More information &
call for presentation:**

<https://events.safari.ethz.ch/asplos26-ramulator-drambender/>



ISCA 2026: DRAM Bender & Ramulator

2nd Tutorial on Ramulator & DRAM Bender: Cutting-Edge Infrastructures for Real and Future Memory System Evaluation

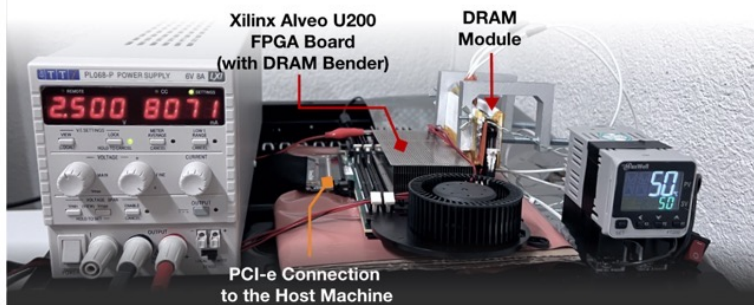


R&DB:

**A full-day tutorial @ ISCA 2026, Raleigh, USA
27 June 2026 (Saturday)**

Organizers: Nisa Bostanci, Ataberk Olgun, Ismail Yuksel, Haocong Luo, Prof. Onur Mutlu

- *Evaluating memory performance, emerging memory technologies, and architectural mechanisms*
- *Real DRAM device characterization studies*
- *New features, extensions, and enhancements*
- *Cross-layer research that combines memory system simulation with real-chip DRAM characterization*



**More information &
call for presentation:**

<https://events.safari.ethz.ch/isca26-ramulator-drambender/>



ICS 2026: DRAM Bender & Ramulator

3rd Tutorial on Ramulator & DRAM Bender: Cutting-Edge Infrastructures for Real and Future Memory System Evaluation

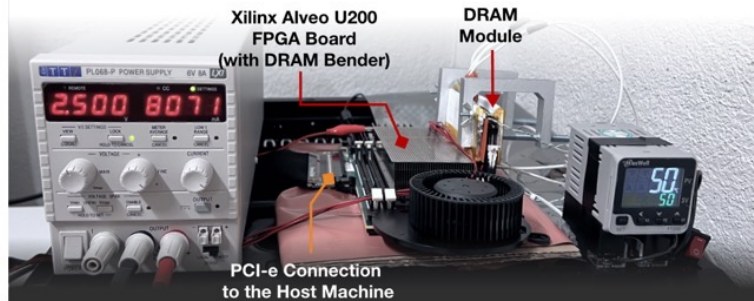


R&DB:

A half-day tutorial @ ICS 2026, Belfast, Northern Ireland, United Kingdom, 6th July 2026 (Monday)

Organizers: Nisa Bostanci, Ataberk Olgun, Ismail Yuksel, Haocong Luo, Prof. Onur Mutlu

- *Evaluating memory performance, emerging memory technologies, and architectural mechanisms*
- *Real DRAM device characterization studies*
- *New features, extensions, and enhancements*
- *Cross-layer research that combines memory system simulation with real-chip DRAM characterization*



***More information &
call for presentation:***

<https://events.safari.ethz.ch/ics26-ramulator-drambender/>



Virtuoso Tutorial & Workshop

VIRTUOSO

✦ 2nd Edition

Infrastructure for Evaluating Novel HW/OS Interfaces

A full-day workshop and tutorial on the principles, methodologies, and practical aspects of hardware/OS co-design for memory management

 Date: Monday, March 23, 2026 (Full Day)

Live in 20 hours
March 23 at 7:45 AM



Notification on

Room Ft. Duquesne, ASPLOS 2026, Pittsburgh, PA

ASPLOS 2026: 2nd Workshop on Virtuoso: Ideas and Infrastructures for Novel HW/OS Interfaces



Onur Mutlu Lectures
60.1K subscribers

Analytics

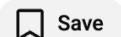
Edit video



3



Share



Save



Evaluating New Ideas for New (Memory) Architectures

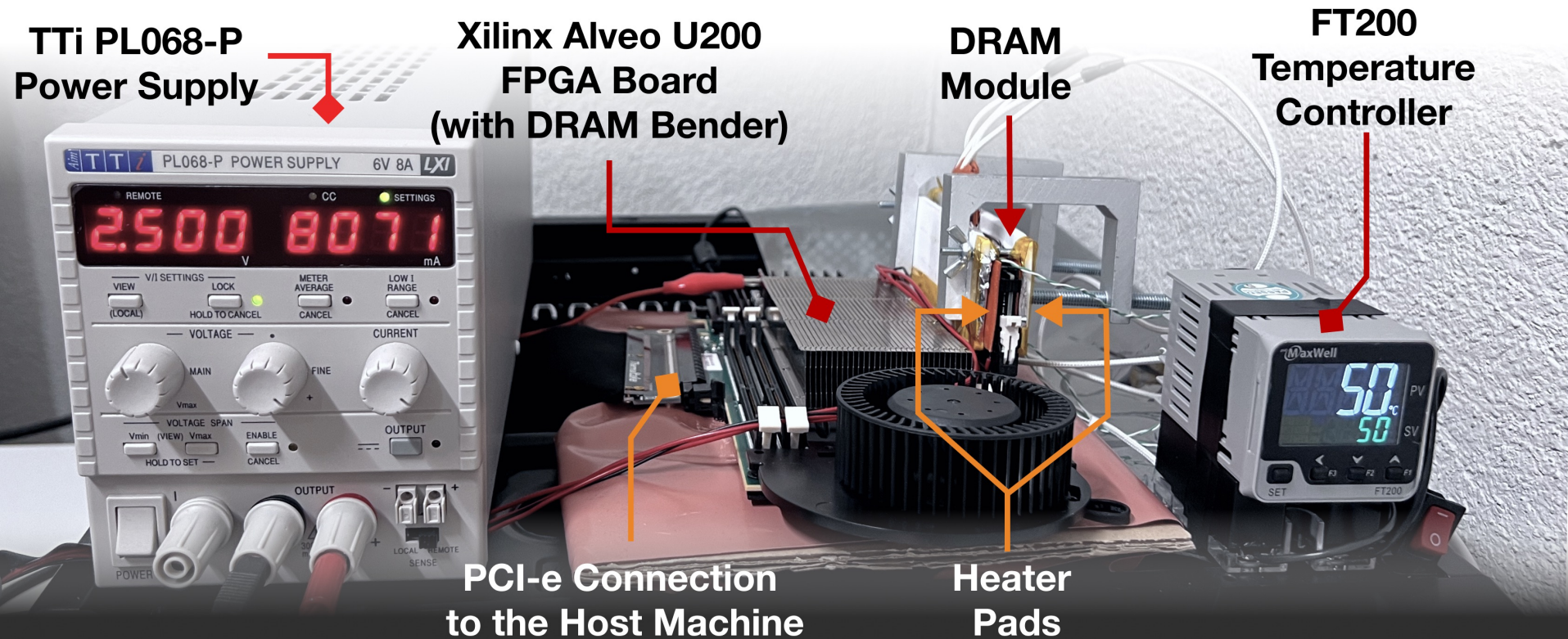
Potential Evaluation Methods

- How do we assess how an idea will affect a target metric X ?
- A variety of evaluation methods are available:
 - Theoretical proof
 - Analytical modeling/estimation
 - Simulation (at varying degrees of abstraction and accuracy)
 - Prototyping with a real system (e.g., FPGAs)
 - Real implementation

An Example Testing Platform

Infrastructures to Understand Memory Issues

DRAM Bender on a Xilinx Virtex UltraScale+ XCU200

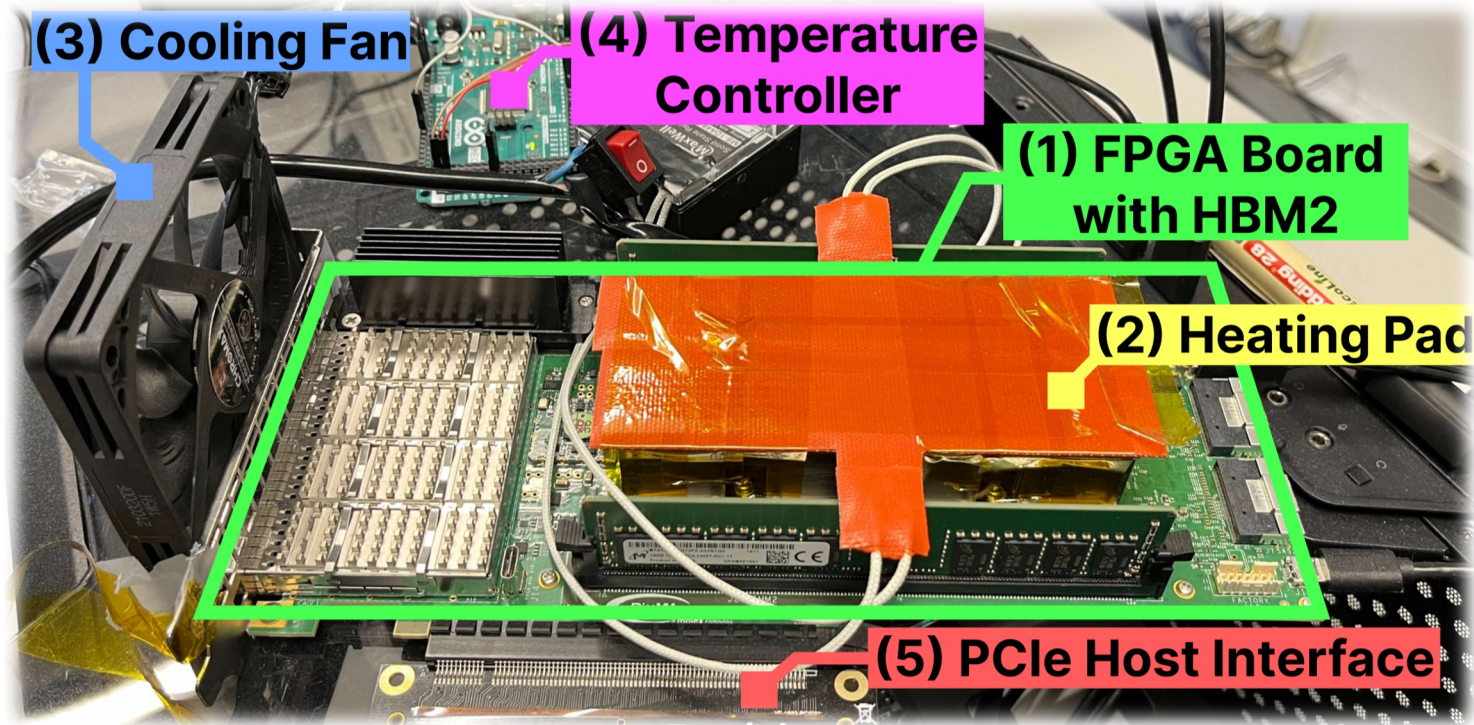


Fine-grained control over **DRAM** commands,
timing parameters ($\pm 1.5\text{ns}$), **temperature ($\pm 0.5^\circ\text{C}$)**,
and **voltage ($\pm 1\text{mV}$)**

*Olgun et al., “[DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips](https://github.com/CMU-SAFARI/DRAM-Bender),” in TCAD, 2023. [GitHub: <https://github.com/CMU-SAFARI/DRAM-Bender>]

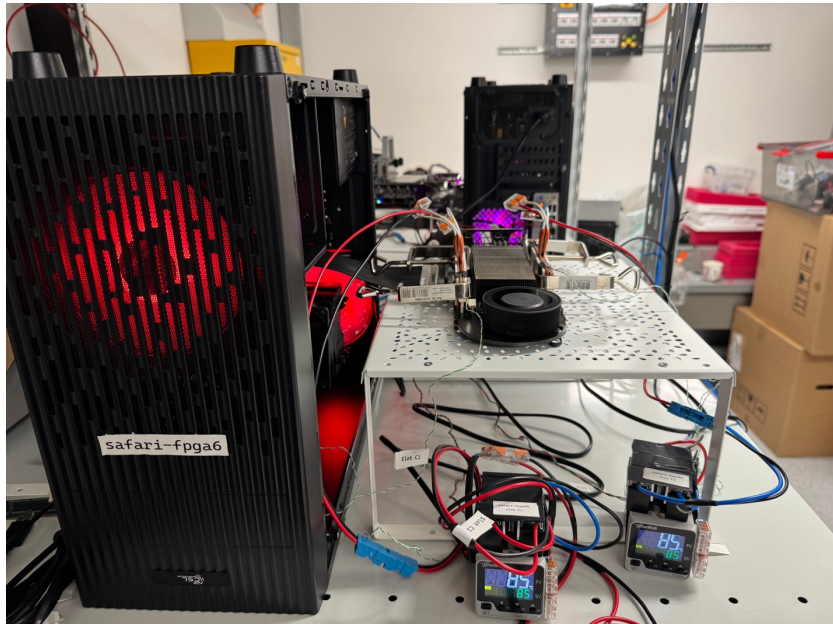
HBM2 DRAM Testing Infrastructure

DRAM Bender on a Bittware XUPV VH



Fine-grained control over DRAM commands,
timing parameters ($\pm 1.67\text{ns}$), and temperature ($\pm 0.5^\circ\text{C}$)

Laboratory for Understanding Memory



Laboratory for Understanding Memory



Laboratory for Understanding Memory

A Modern Large-Scale Memory Characterization Laboratory

Ataberk Olgun[§]

Haocong Luo[§]

İsmail Emir Yüksel[§]

F. Nisa Bostancı[§]

A. Giray Yağlıkcı[§]

Onur Mutlu[§]

[§]SAFARI Research Group

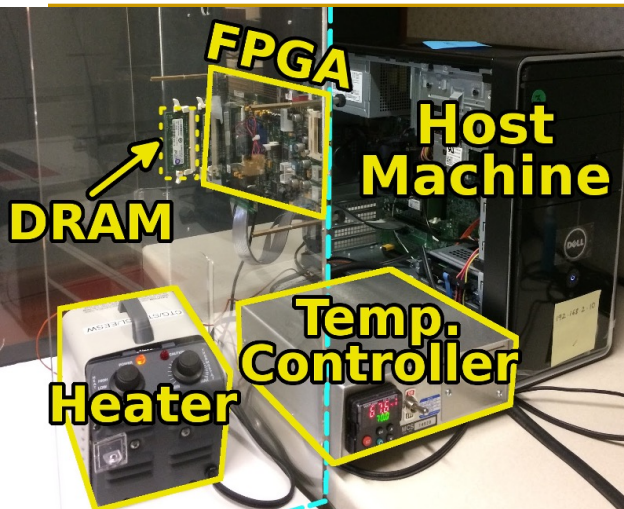
ETH Zürich

Real memory chip characterization yields insights into fundamental operational characteristics of modern memory, enabling new mechanisms that improve memory performance, robustness, security, and energy efficiency. We describe our large-scale DRAM characterization laboratory for understanding DRAM. A key building block of this laboratory is DRAM Bender [1, 2], a versatile and easy-to-use modern DRAM characterization infrastructure. We have updated DRAM Bender to i) introduce support for new types of characterization experiments, ii) expand on its DRAM interface standard support, and iii) make it easier to use at large scale. This paper introduces these updates for the first time. We hope our infrastructure enables the community to discover new problems and solve critical memory scaling issues, enabling the overcoming of the huge memory bottleneck that plagues modern computing systems.

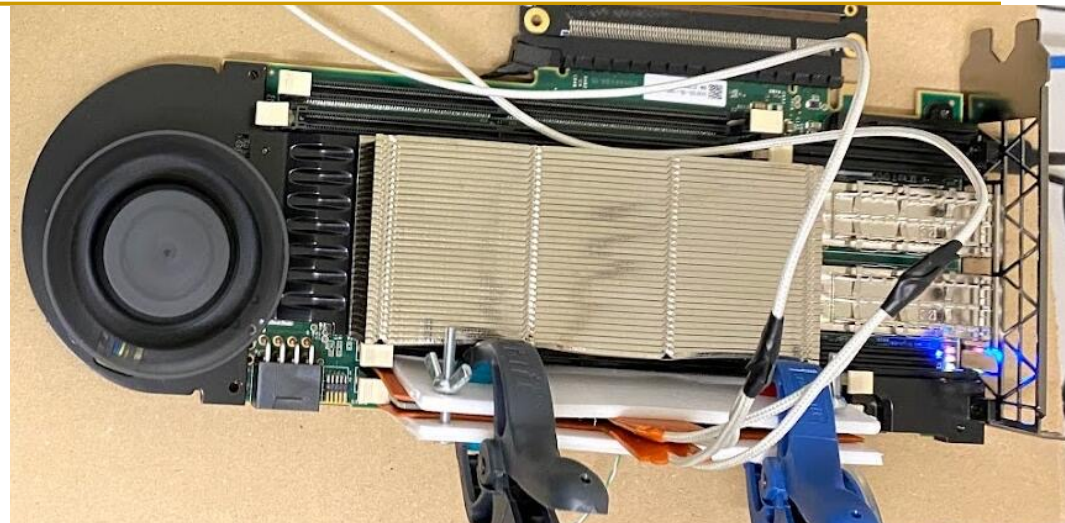
and other major parameters. This infrastructure, which took us more than a year to build and which we later open sourced as SoftMC [13, 14] and DRAM Bender [1, 2], enabled us (and many others) to empirically study and understand many interesting and important characteristics of modern DRAM chips.

Many researchers and practitioners from industry and academia have been using DRAM Bender (and SoftMC) to deeply understand DRAM and solve major memory challenges [11, 15–68]. For example, DRAM Bender has enabled the discovery and study of major DRAM phenomena including RowHammer [28], RowPress [21], Variable Read Disturbance [18], ColumnDisturb [15], the DejaVu effect [57], Variable Retention Time [11, 67, 69, 70], data pattern dependence [11, 27, 71], data retention failures [11, 71], and computational capabilities of real commercial off-the-shelf

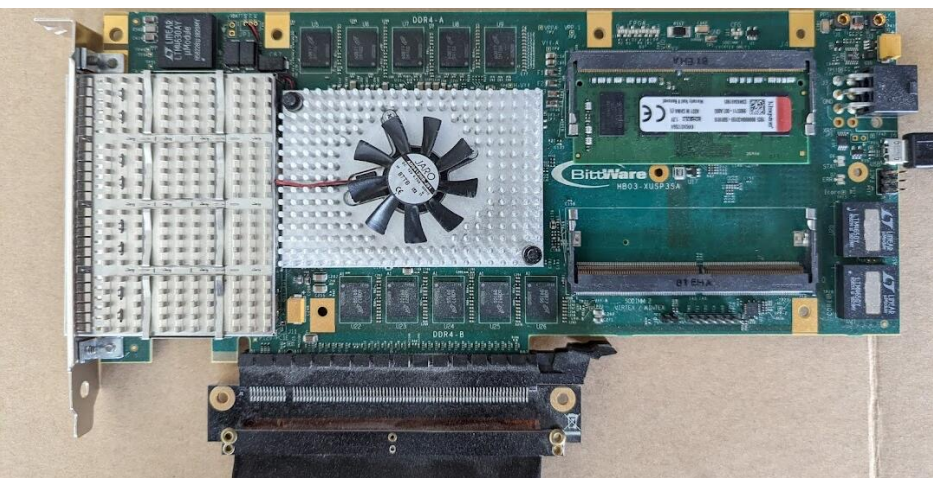
DRAM Testing Infrastructures (I)



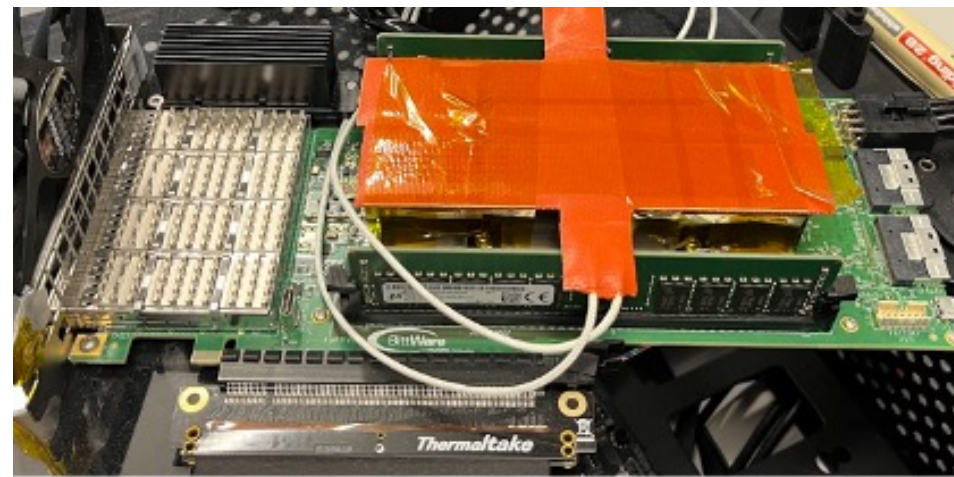
DDR3 DRAM SODIMMs
Xilinx ML605



DDR4 DRAM R/UDIMMs
Xilinx Alveo U200

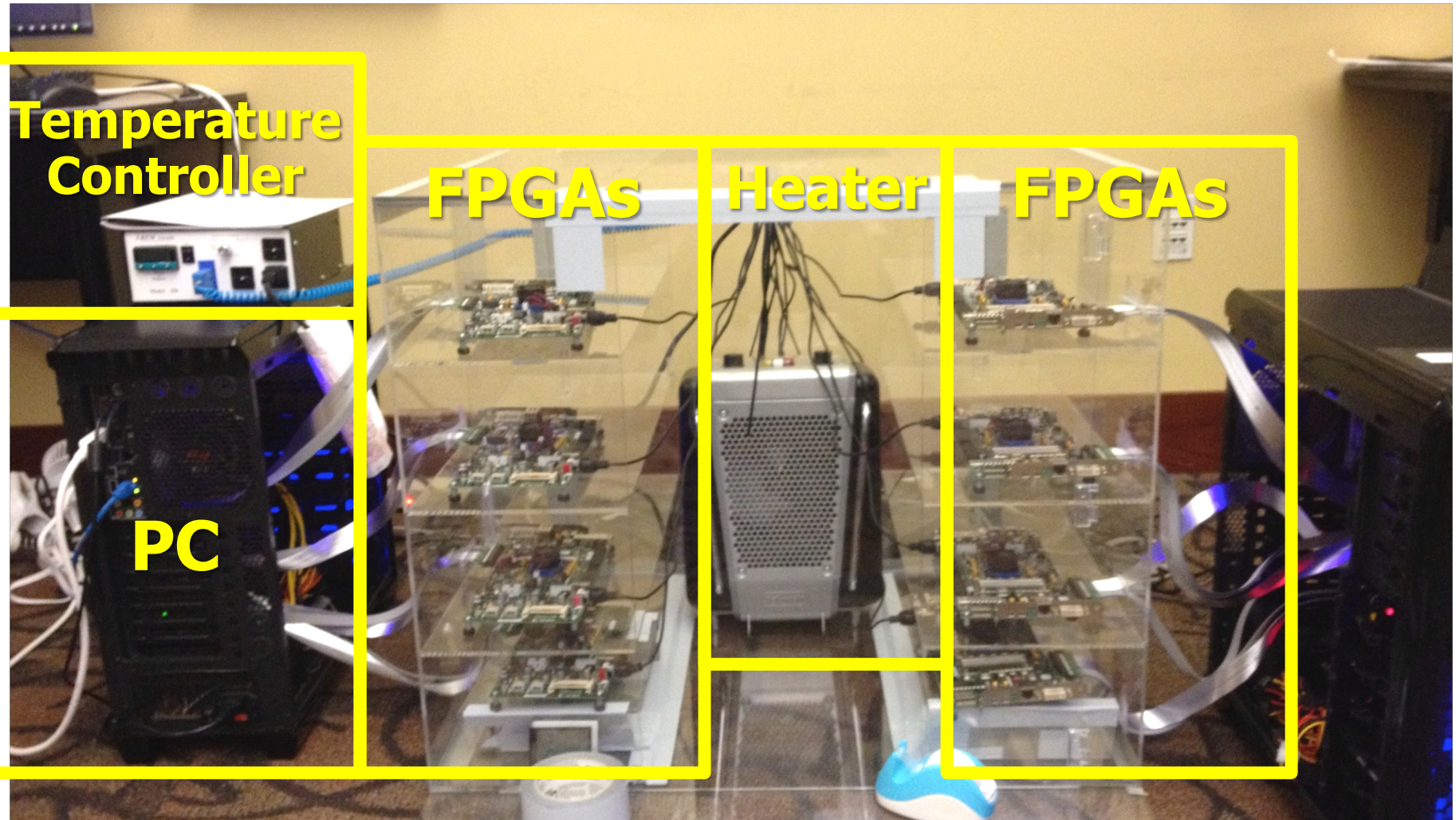


DDR4 DRAM (SODIMM)
Bittware XUSP3S



HBM2 DRAM Chips
Xilinx Alveo U50

DRAM Testing Infrastructures (II)



DRAM Testing Infrastructures (III)



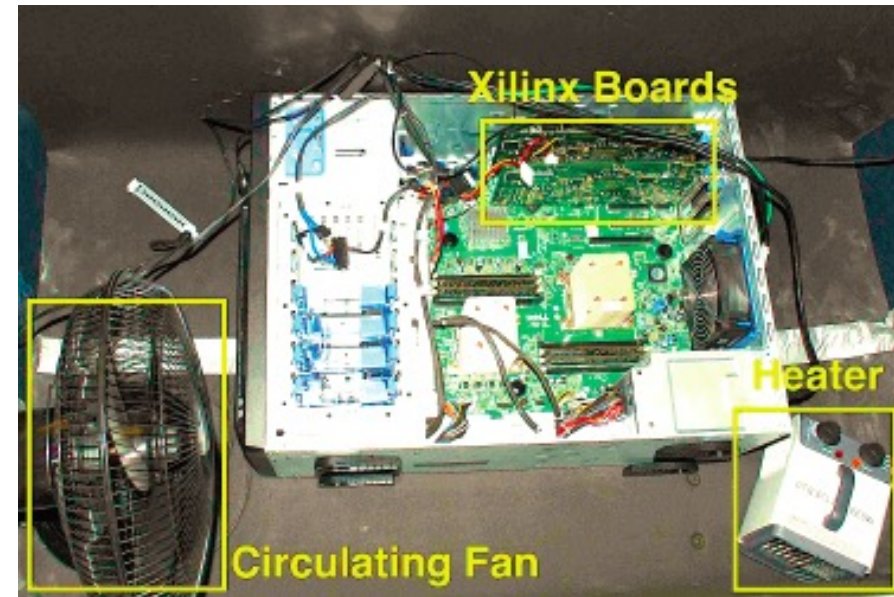
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study (Khan et al., SIGMETRICS 2014)

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

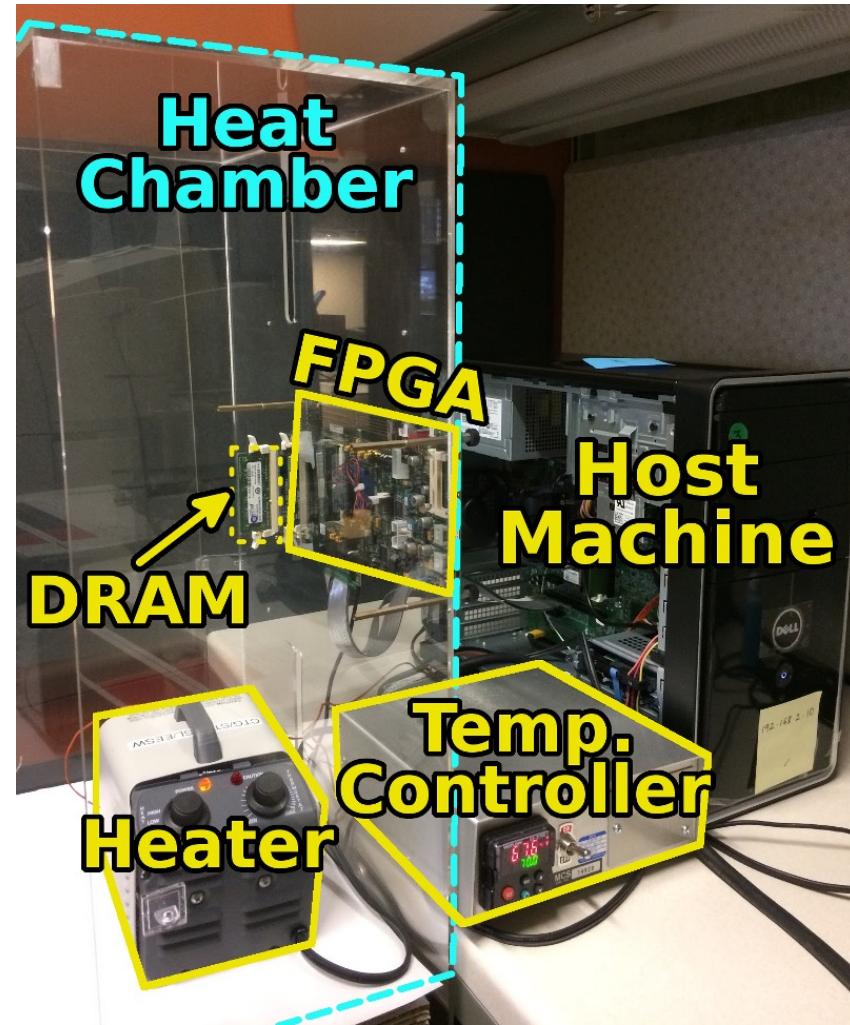
Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case (Lee et al., HPCA 2015)

AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)



SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan et al., “[SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies](#),” HPCA 2017.
- Flexible
- Easy to Use (C++ API)
- Open-source
github.com/CMU-SAFARI/SoftMC



SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan, Nandita Vijaykumar, Samira Khan, Saugata Ghose, Kevin Chang, Gennady Pekhimenko, Donghyuk Lee, Oguz Ergin, and Onur Mutlu,

"SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies"

Proceedings of the 23rd International Symposium on High-Performance Computer Architecture (HPCA), Austin, TX, USA, February 2017.

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]

[Full Talk Lecture (39 minutes)]

[Source Code]

SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan^{1,2,3} Nandita Vijaykumar³ Samira Khan^{4,3} Saugata Ghose³ Kevin Chang³
Gennady Pekhimenko^{5,3} Donghyuk Lee^{6,3} Oguz Ergin² Onur Mutlu^{1,3}

¹ETH Zürich ²TOBB University of Economics & Technology ³Carnegie Mellon University
⁴University of Virginia ⁵Microsoft Research ⁶NVIDIA Research

DRAM Bender

- Ataberk Olgun, Hasan Hassan, A Giray Yağlıkçı, Yahya Can Tuğrul, Lois Orosa, Haocong Luo, Minesh Patel, Oğuz Ergin, and Onur Mutlu,
"DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips"
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 2023.
[[Extended arXiv version](#)]
[[DRAM Bender Source Code](#)]
[[DRAM Bender Tutorial Video](#) (43 minutes)]

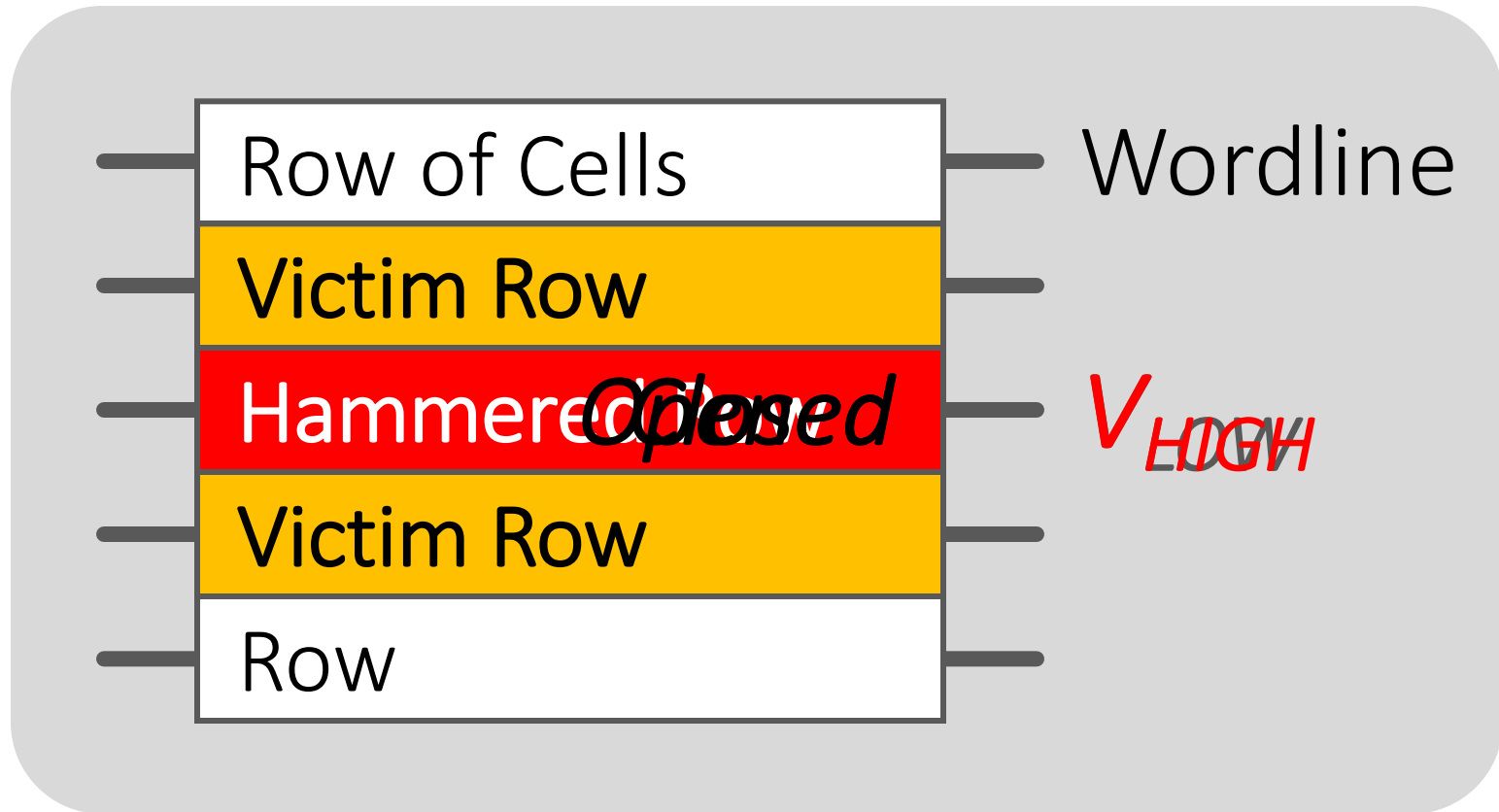
DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips

Ataberk Olgun[§] Hasan Hassan[§] A. Giray Yağlıkçı[§] Yahya Can Tuğrul^{§†}
Lois Orosa^{§⊙} Haocong Luo[§] Minesh Patel[§] Oğuz Ergin[†] Onur Mutlu[§]
 [§]*ETH Zürich* [†]*TOBB ETÜ* [⊙]*Galician Supercomputing Center*



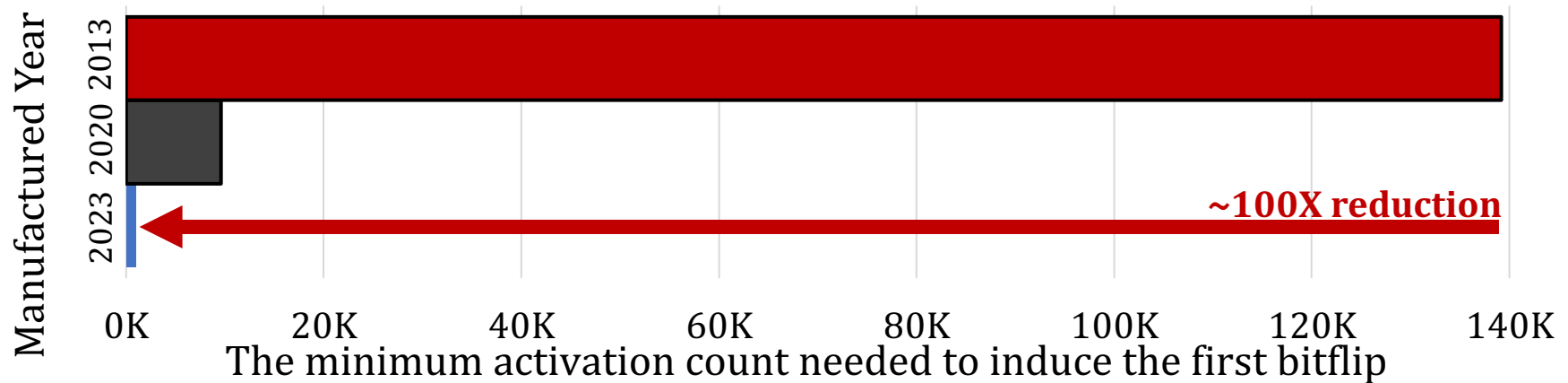
Rowhammer

Modern DRAM is Prone to Disturbance Errors



Repeatedly reading a row enough times (before memory gets refreshed) induces **disturbance errors** in **adjacent rows** in **most real DRAM chips you can buy today**

Read Disturbance Worsens with Scaling



RowHammer [ISCA 2014]

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"
Proceedings of the 41st International Symposium on Computer Architecture (ISCA), Minneapolis, MN, June 2014.
[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Session Slides \(pptx\) \(pdf\)](#)] [[Source Code and Data](#)] [[Lecture Video](#) (1 hr 49 mins), 25 September 2020]
One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD ([link](#)). Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 ([Retrospective \(pdf\)](#) [Full Issue](#)). Winner of the 2024 IFIP Jean-Claude Laprie Award in dependable computing ([link](#)).

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim¹ Ross Daly* Jeremie Kim¹ Chris Fallin* Ji Hye Lee¹
Donghyuk Lee¹ Chris Wilkerson² Konrad Lai Onur Mutlu¹

¹Carnegie Mellon University ²Intel Labs

Many RowHammer Security Exploits

- One can exploit RowHammer to
- Take over a system
- Read data they do not have access to
- Break out of virtual machine sandboxes
- Corrupt important data → render ML inference useless
- Steal secret data (e.g., crypto keys & ML model parameters)



- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tugrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu, **"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"**

Proceedings of the 50th International Symposium on Computer Architecture (ISCA), Orlando, FL, USA, June 2023.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Video](#) (3 minutes)]

[[RowPress Source Code and Datasets \(Officially Artifact Evaluated with All Badges\)](#)]

***Officially artifact evaluated as available, reusable and reproducible.
Best artifact award at ISCA 2023. IEEE Micro Top Pick in 2024.***

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

Haocong Luo Ataberk Olgun A. Giray Yağlıkçı Yahya Can Tuğrul Steve Rhyner
Meryem Banu Cavlak Joël Lindegger Mohammad Sadrosadati Onur Mutlu

ETH Zürich

RowPress vs. RowHammer

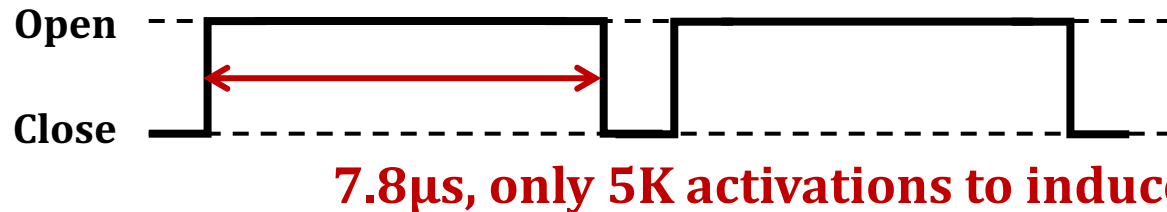
Instead of using a high activation count,

☞ increase the time that the aggressor row stays open

RowHammer
Aggressor Row



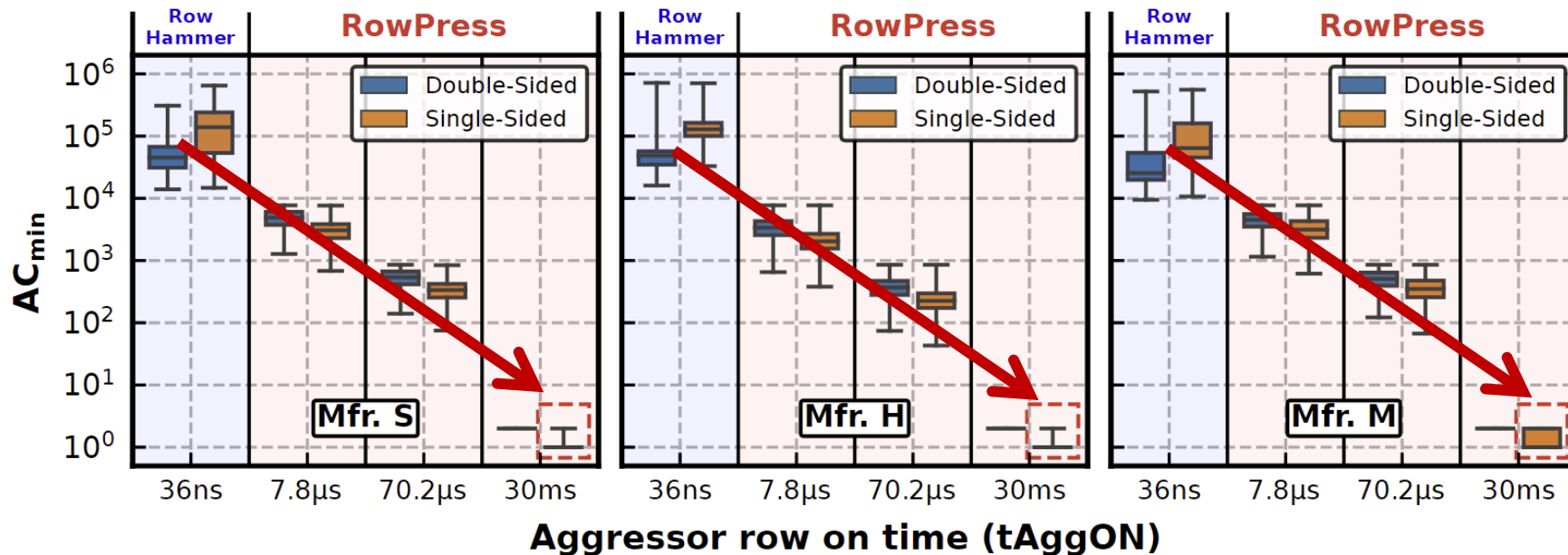
RowPress
Aggressor Row



RowPress reduces the number of activations to induce a bitflip by 1-2 orders of magnitude

Amplifies Read Disturbance in DRAM

- Reduces the minimum number of row activations needed to induce a bitflip (AC_{min}) by **1-2 orders of magnitude**
- In extreme cases, activating a row **only once** induces bitflips



Main Memory Needs Intelligent Controllers

An “Early” Position Paper [IMW 2013]

- Onur Mutlu,
"Memory Scaling: A Systems Architecture Perspective"
*Proceedings of the 5th International Memory Workshop (**IMW**), Monterey, CA, May 2013. Slides
(pptx) (pdf)
EETimes Reprint*

Memory Scaling: A Systems Architecture Perspective

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu
<http://users.ece.cmu.edu/~omutlu/>

Updated Paper 12 Years Later [IMW 2025]

- Onur Mutlu, Ataberk Olgun, and İsmail Emir Yüksel,
**"Memory-Centric Computing: Solving Computing's
Memory Problem"**

*Invited Paper in Proceedings of the 17th IEEE International
Memory Workshop (**IMW**), Monterey, CA, USA, May 2025.*

[Slides (pptx) (pdf)]

Memory-Centric Computing: Solving Computing's Memory Problem

Onur Mutlu Ataberk Olgun İsmail Emir Yüksel

ETH Zürich

<https://www.arxiv.org/pdf/2505.00458>

Industry's Intelligent DRAM Controllers (I)

ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

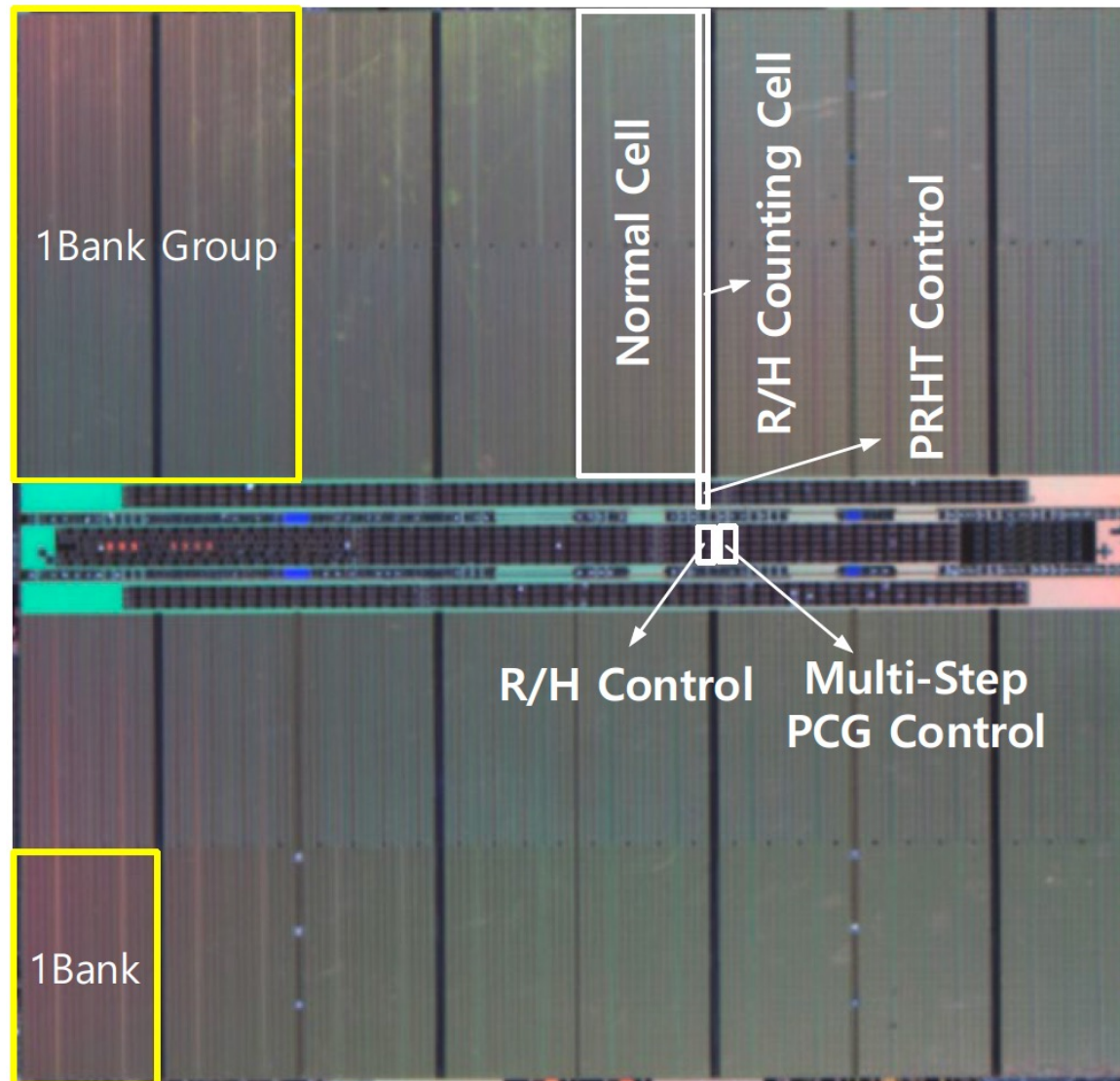


Industry's Intelligent DRAM Controllers (II)

SK hynix Semiconductor, Icheon, Korea

DRAM products have been recently adopted in a wide range of high-performance computing applications: such as in cloud computing, in big data systems, and IoT devices. This demand creates larger memory capacity requirements, thereby requiring aggressive DRAM technology node scaling to reduce the cost per bit [1,2]. However, DRAM manufacturers are facing technology scaling challenges due to row hammer and refresh retention time beyond 1a-nm [2]. Row hammer is a failure mechanism, where repeatedly activating a DRAM row disturbs data in adjacent rows. Scaling down severely threatens reliability since a reduction of DRAM cell size leads to a reduction in the intrinsic row hammer tolerance [2,3]. To improve row hammer tolerance, there is a need to probabilistically activate adjacent rows with carefully sampled active addresses and to improve intrinsic row hammer tolerance [2]. In this paper, row-hammer-protection and refresh-management schemes are presented to guarantee DRAM security and reliability despite the aggressive scaling from 1a-nm to sub 10-nm nodes. The probabilistic-aggressor-tracking scheme with a refresh-management function (RFM) and per-row hammer tracking (PRHT) improve DRAM resilience. A multi-step precharge reinforces intrinsic row-hammer tolerance and a core-bias modulation improves retention time: even in the face of cell-transistor degradation due to technology scaling. This comprehensive scheme leads to a reduced probability of failure, due to row hammer attacks, by 93.1% and an improvement in retention time by 17%.

Industry's Intelligent DRAM Controllers (III)



ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyoung Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

Finally: Serious Changes in Standards (2024)



Global Standards for the Microelectronics Industry

STANDARDS & DOCUMENTS

COMMITTEES

NEWS

EVENTS & MEETINGS

JOIN

DDR5 SDRAM

JESD79-5C

Apr 2024

Release Number: Version 1.30

Version 1.30

This standard defines the DDR5 SDRAM specification, including features, functionalities, AC and DC characteristics, packages, and ball/signal assignments. The purpose of this Standard is to define the minimum set of requirements for JEDEC compliant 8 Gb through 32 Gb for x4, x8, and x16 DDR5 SDRAM devices. This standard was created based on the DDR4 standards (JESD79-4) and some aspects of the DDR, DDR2, DDR3, and LPDDR4 standards (JESD79, JESD79-2, JESD79-3, and JESD209-4).

Committee(s): [JC-42](#), [JC-42.3](#)

Are Solutions Good?



Simulating New Ideas: Ramulator

- Yoongu Kim, Weikun Yang, and Onur Mutlu,
"Ramulator: A Fast and Extensible DRAM Simulator"
IEEE Computer Architecture Letters (**CAL**), March 2015.
[Source Code]
- Source code is released under the liberal MIT License
 - <https://github.com/CMU-SAFARI/ramulator>

Ramulator: A Fast and Extensible DRAM Simulator

Yoongu Kim¹ Weikun Yang^{1,2} Onur Mutlu¹
¹Carnegie Mellon University ²Peking University

Ramulator 2.0: A Much Better Ramulator

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
"Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
*Preprint on **arxiv**, August 2023.*
[[arXiv version](#)]
[[Ramulator 2.0 Source Code](#)]

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

<https://arxiv.org/pdf/2308.11030.pdf>

Ramulator 2.1: A Recent Update

- **Appears at ICS 2026 Workshops Proceedings**

Ramulator 2.1: A Composable Memory System Simulator for Modern DRAM Systems

Haocong Luo F. Nisa Bostancı Ataberk Olgun Maria Makeenkova
Ziad Malik Ipek Akdeniz Onur Mutlu

*SAFARI Research Group
ETH Zürich*

<https://arxiv.org/pdf/2606.13844>

Evaluation of Industry's Recent Solutions

- **Appears at DRAMSec 2024**

Understanding the Security Benefits and Overheads of Emerging Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat^{§†}

A. Giray Yağlıkçı[§]

Geraldo F. Oliveira[§]

Ataberk Olgun[§]

Oğuz Ergin[†]

Onur Mutlu[§]

[§]*ETH Zürich*

[†]*TOBB University of Economics and Technology*

<https://arxiv.org/pdf/2406.19094>

<https://github.com/CMU-SAFARI/ramulator2>

Evaluation of Industry's Recent Solutions

- Oguzhan Canpolat, Abdullah Giray Yaglikci, Geraldo Francisco de Oliveira, Ataberk Olgun, Nisa Bostanci, Ismail Emir Yuksel, Haocong Luo, Oguz Ergin, and Onur Mutlu,
"Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance"

Proceedings of the 31st International Symposium on High-Performance Computer Architecture (HPCA), Las Vegas, NV, USA, March 2025.

[Chronus Source Code (Officially Artifact Evaluated with All Badges)]

Officially artifact evaluated as available, functional, and reproduced.

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*

<https://arxiv.org/pdf/2502.12650>

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

Asking the Right Question

Can We Do Better?

Better Partitioning of DRAM & Controller

- Hasan Hassan, Ataberk Olgun, A. Giray Yaglikci, Haocong Luo, and Onur Mutlu,
"Self-Managing DRAM: A Low-Cost Framework for Enabling Autonomous and Efficient DRAM Maintenance Operations"
Proceedings of the 57th International Symposium on Microarchitecture (MICRO), Austin, TX, USA, November 2024.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[SelfManagingDRAM Source Code](#)]

Self-Managing DRAM: A Low-Cost Framework for Enabling Autonomous and Efficient DRAM Maintenance Operations

Hasan Hassan[†]

Ataberk Olgun[†]

A. Giray Yağlıkçı

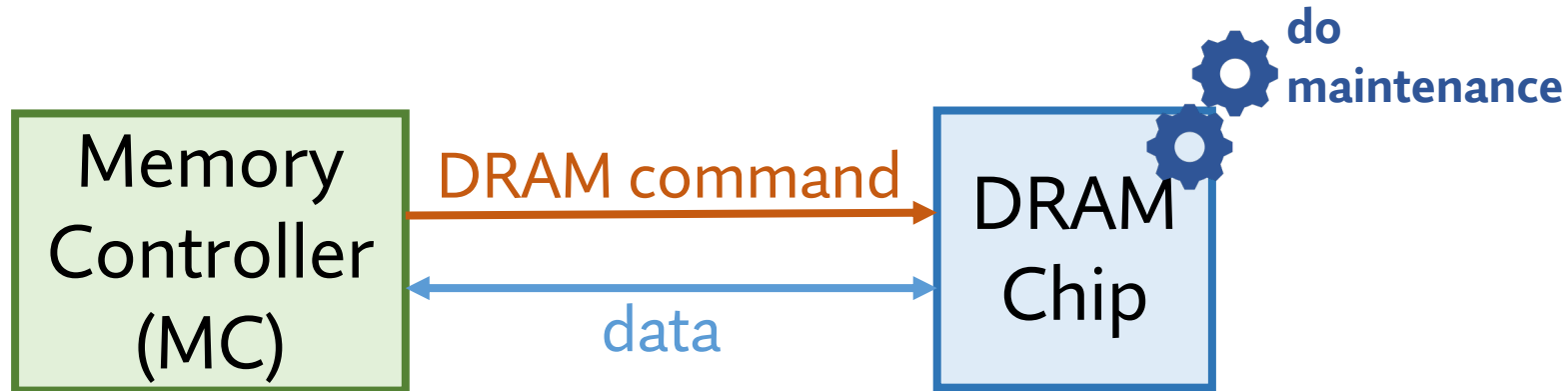
Haocong Luo

Onur Mutlu

ETH Zürich

SMD Key Idea: Autonomous Maintenance

DRAM chip controls in-DRAM maintenance operations

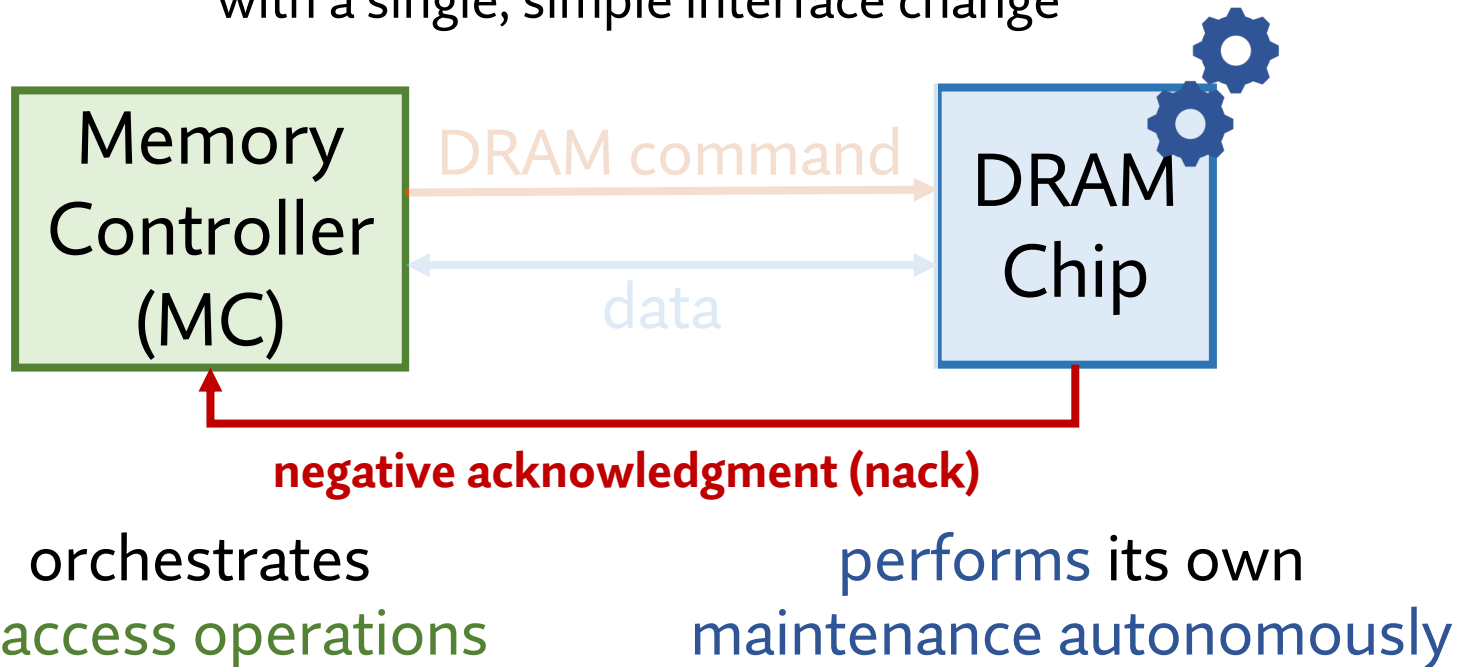


Enable implementing **new maintenance mechanisms** **without** modifying the standard and exposing **DRAM-internal proprietary** information

SMD Key Contribution

DRAM chip controls in-DRAM maintenance operations

with a single, simple interface change



Partition the work nicely between the memory controller and the DRAM chip

SMD-Based Maintenance Mechanisms

DRAM Refresh

Fixed Rate (SMD-FR)

uniformly refreshes **all** DRAM rows with a **fixed** refresh period

Variable Rate (SMD-VR)

skips refreshing rows that can **retain their data for longer** than the default refresh period

RowHammer Protection

Probabilistic (SMD-PRP)

Performs **neighbor** row refresh with a **small probability** on every row activation

Deterministic (SMD-DRP)

keeps track of most **frequently activated** rows and performs **neighbor** row refresh when activation count threshold is exceeded

Memory Scrubbing

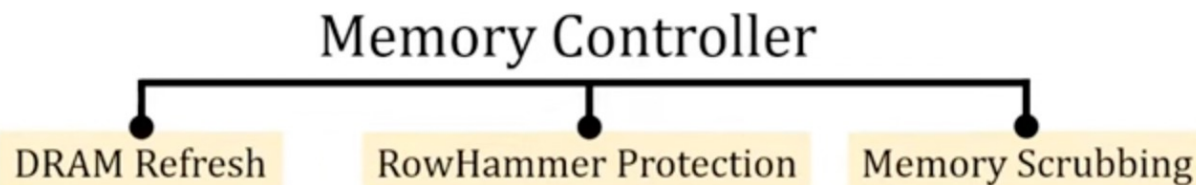
Periodic Scrubbing (SMD-MS)

periodically **scans** the **entire** DRAM for errors and corrects them

Talk on Self-Managing DRAM

Problem: The Rigid DRAM Interface

The **Memory Controller** manages DRAM maintenance operations



Changes to maintenance operations are often reflected to the memory controller design, DRAM interface, and other system components



Implementing new maintenance operations
(or modifying the existing ones) is difficult-to-realize



SAFARI Live Seminars 2022

SAFARI Live Seminar - Improving DRAM Performance, Reliability, and Security by Understanding DRAM

1,039 views • Streamed live on Sep 15, 2022

37 DISLIKE SHARE DOWNLOAD CLIP SAVE ...



Onur Mutlu Lectures
27.6K subscribers

ANALYTICS

EDIT VIDEO

Self-Managing DRAM

- Hasan Hassan, Ataberk Olgun, A. Giray Yaglikci, Haocong Luo, and Onur Mutlu,
"Self-Managing DRAM: A Low-Cost Framework for Enabling Autonomous and Efficient DRAM Maintenance Operations"
Proceedings of the 57th International Symposium on Microarchitecture (MICRO), Austin, TX, USA, November 2024.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[SelfManagingDRAM Source Code](#)]

Self-Managing DRAM: A Low-Cost Framework for Enabling Autonomous and Efficient DRAM Maintenance Operations

Hasan Hassan[†] Ataberk Olgun[†] A. Giray Yağlıkçı Haocong Luo Onur Mutlu
ETH Zürich

More in My MICRO 2025 Keynote Talk



The screenshot shows a YouTube video player with a slide titled "Can We Do Better?". The slide content includes the name "Onur Mutlu", email "omutlu@gmail.com", website "https://people.inf.ethz.ch/omutlu", date "21 October 2025", and event "MICRO 2025 Keynote Talk @ Seoul". Logos for "SAFARI", "ETH zürich", and "Zoom" are visible at the bottom of the slide. A small video inset in the top right corner shows the speaker, Onur Mutlu, on a stage.

Can We Do Better?

Onur Mutlu
omutlu@gmail.com
<https://people.inf.ethz.ch/omutlu>
21 October 2025
MICRO 2025 Keynote Talk @ Seoul

SAFARI ETH zürich Zoom

Can We Do Better? - Keynote Talk at MICRO 2025 - Prof. Onur Mutlu



Onur Mutlu Lectures
58.5K subscribers



2,752 views Streamed live on Oct 21, 2025

Title: Can We Do Better?

Presenter: Prof. Onur Mutlu (<https://people.inf.ethz.ch/omutlu/>)

Date and Time: October 21, 2025, 08:00 AM (KST)

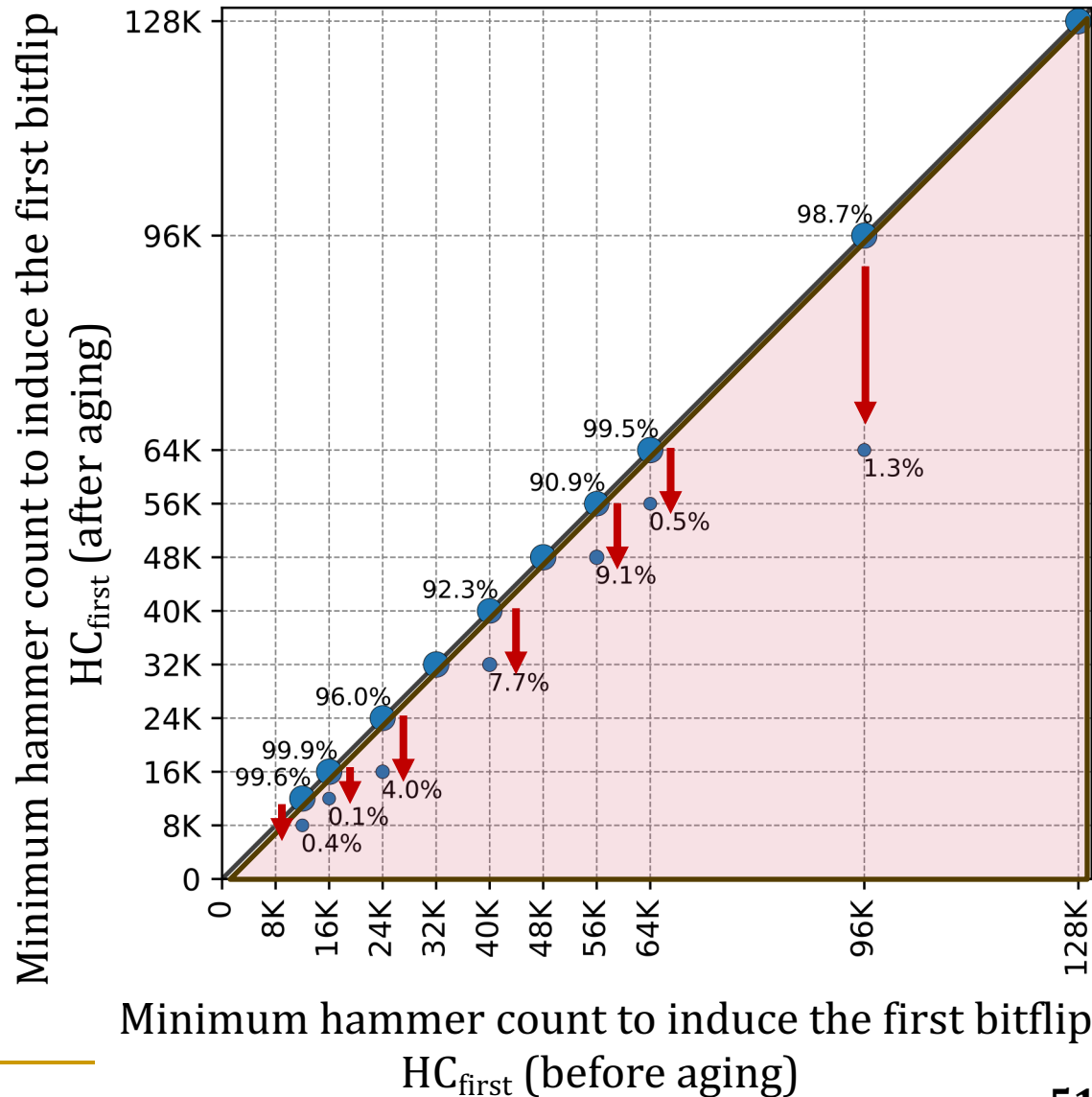
More to Come...

There Is Still
a Whole Lot
We Do Not Know
About DRAM

RowHammer Becomes Worse with Aging

Preliminary data on aging via 68-day of continuous hammering

Aging can lead to read disturbance bitflips at **smaller** hammer counts



RowHammer (Spatial Variation) Analysis (2024)

- **Appears at HPCA 2024**

Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

Abdullah Giray Yağlıkçı Yahya Can Tuğrul Geraldo F. Oliveira
İsmail Emir Yüksel Ataberk Olgun Haocong Luo Onur Mutlu
ETH Zürich

<https://arxiv.org/pdf/2402.18652>

Variable Read Disturbance (2025)

Key Takeaway

The Read Disturbance Threshold (RDT) of a row **changes randomly and unpredictably over time**

Accurately identifying RDT is challenging

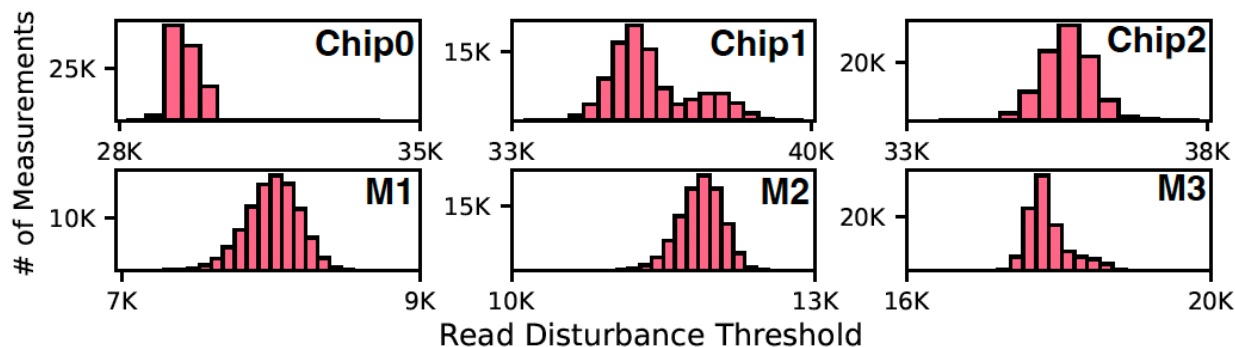


Fig. 2: Read disturbance threshold of a row in each tested HBM2 chip (Chip0-2) and DDR4 module (M1-3) over 100,000 repeated measurements. Adapted from [43].

Variable Read Disturbance (2025)

- **Appears at HPCA 2025**

Variable Read Disturbance:

An Experimental Analysis of Temporal Variation in DRAM Read Disturbance

Ataberk Olgun[†] F. Nisa Bostancı[†] İsmail Emir Yüksel[†] Oğuzhan Canpolat[†] Haocong Luo[†]
Geraldo F. Oliveira[†] A. Giray Yağlıkçı[†] Minesh Patel[‡] Onur Mutlu[†]
ETH Zurich[†] Rutgers University[‡]

Read disturbance threshold varies unpredictably over time

ColumnDisturb (2025)

- **Appears at MICRO 2025**

ColumnDisturb: Understanding Column-based Read Disturbance in Real DRAM Chips and Implications for Future Systems

İsmail Emir Yüksel¹ Ataberk Olgun¹ F. Nisa Bostancı¹
Haocong Luo¹ A. Giray Yağlıkçı^{1,2} Onur Mutlu¹
¹ETH Zürich ²CISPA

**A single row activation flips bits across three DRAM subarrays
(up to 3072 DRAM rows)**

PuDHammer (2025)

- **Appears at ISCA 2025**

PuDHammer: Experimental Analysis of Read Disturbance Effects of Processing-using-DRAM in Real DRAM Chips

İsmail Emir Yüksel Akash Sood Ataberk Olgun Oğuzhan Canpolat Haocong Luo
F. Nisa Bostancı Mohammad Sadrosadati A. Giray Yağlıkçı Onur Mutlu

ETH Zürich

Read disturbance becomes much worse with processing using DRAM

PuDGhost (2026)

■ Appears at ISCA 2026

PuDGhost: Experimental Analysis of Computation Result Corruption in Processing-using-DRAM Operations on Real DRAM Chips and Implications for Future Systems

Daichi Tokuda¹ İsmail Emir Yüksel² Tatsuya Kubo^{1,4} Ataberk Olgun² Haocong Luo² Nisa Bostancı²
Jikun Wang² A. Giray Yağlıkçı³ Shinya Takamaeda-Yamazaki^{1,4} Onur Mutlu²
¹The University of Tokyo ²ETH Zurich ³CISPA ⁴RIKEN

Data in non-activated rows affect PuD computation results

DejaVu (2026)

- **Appears at ISCA 2026**

DejaVu: Why You Should Write to Your DRAM Rows Twice, Carefully

Haocong Luo¹ İsmail Emir Yüksel¹ Ataberk Olgun¹

Nisa Bostanci¹ Orhun Ecemiş² Abdullah Giray Yaglikci³ Onur Mutlu¹

¹*ETH Zurich* ²*TOBB ETÜ* ³*CISPA*

**Data previously written to a row
affects the row's read disturbance vulnerability**

Emerging Memories Also Need Intelligent Controllers

- Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger,
"Architecting Phase Change Memory as a Scalable DRAM Alternative"
Proceedings of the 36th International Symposium on Computer Architecture (ISCA), pages 2-13, Austin, TX, June 2009. [Slides \(pdf\)](#)
One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight. 2022 Persistent Impact Prize.

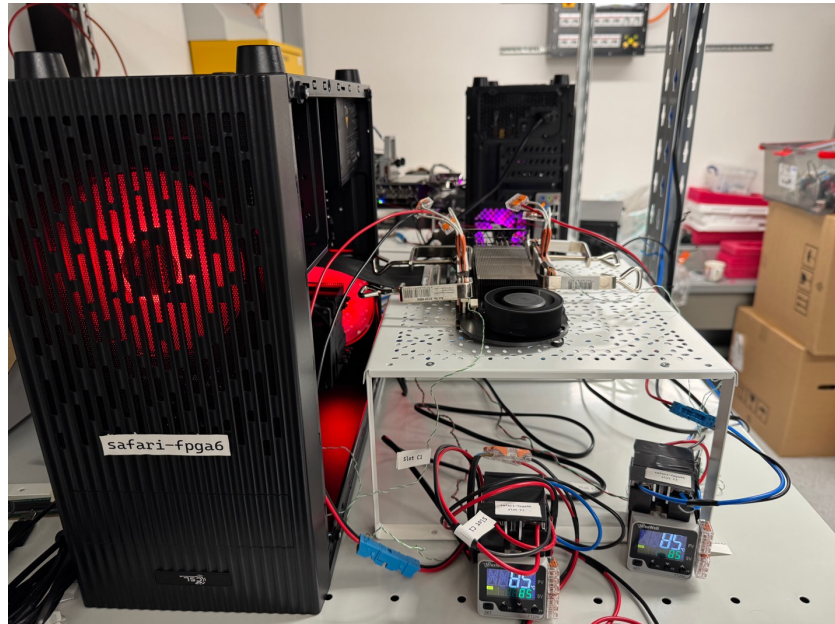
Architecting Phase Change Memory as a Scalable DRAM Alternative

Benjamin C. Lee[†] Engin Ipek[†] Onur Mutlu[‡] Doug Burger[†]

[†]Computer Architecture Group
Microsoft Research
Redmond, WA
{blee, ipek, dburger}@microsoft.com

[‡]Computer Architecture Laboratory
Carnegie Mellon University
Pittsburgh, PA
onur@cmu.edu

Laboratory for Understanding Memory



Laboratory for Understanding Memory

A Modern Large-Scale Memory Characterization Laboratory

Ataberk Olgun[§]

Haocong Luo[§]

İsmail Emir Yüksel[§]

F. Nisa Bostancı[§]

A. Giray Yağlıkcı[§]

Onur Mutlu[§]

[§]SAFARI Research Group

ETH Zürich

Real memory chip characterization yields insights into fundamental operational characteristics of modern memory, enabling new mechanisms that improve memory performance, robustness, security, and energy efficiency. We describe our large-scale DRAM characterization laboratory for understanding DRAM. A key building block of this laboratory is DRAM Bender [1, 2], a versatile and easy-to-use modern DRAM characterization infrastructure. We have updated DRAM Bender to i) introduce support for new types of characterization experiments, ii) expand on its DRAM interface standard support, and iii) make it easier to use at large scale. This paper introduces these updates for the first time. We hope our infrastructure enables the community to discover new problems and solve critical memory scaling issues, enabling the overcoming of the huge memory bottleneck that plagues modern computing systems.

and other major parameters. This infrastructure, which took us more than a year to build and which we later open sourced as SoftMC [13, 14] and DRAM Bender [1, 2], enabled us (and many others) to empirically study and understand many interesting and important characteristics of modern DRAM chips.

Many researchers and practitioners from industry and academia have been using DRAM Bender (and SoftMC) to deeply understand DRAM and solve major memory challenges [11, 15–68]. For example, DRAM Bender has enabled the discovery and study of major DRAM phenomena including RowHammer [28], RowPress [21], Variable Read Disturbance [18], ColumnDisturb [15], the DejaVu effect [57], Variable Retention Time [11, 67, 69, 70], data pattern dependence [11, 27, 71], data retention failures [11, 71], and computational capabilities of real commercial off-the-shelf

A Very Recent PhD Thesis

- A. Giray Yaglikci, "[Enabling Efficient and Scalable DRAM Read Disturbance Mitigation via New Experimental Insights into Modern DRAM Chips](#)," PhD Thesis, ETH Zürich, 2024.
[[Slides \(pdf\) \(pptx\)](#)]
[[Thesis arXiv \(abs\) \(pdf\)](#)]
[[SAFARI News](#)]

ENABLING EFFICIENT AND SCALABLE DRAM READ DISTURBANCE MITIGATION VIA NEW EXPERIMENTAL INSIGHTS INTO MODERN DRAM CHIPS

ABDULLAH GİRAY YAĞLIKÇI

<https://arxiv.org/pdf/2408.15044.pdf>

Existing DRAM Chips
Are Already Quite Capable

The Capability of COTS DRAM Chips

We demonstrate that COTS DRAM chips:

- 1** Can copy one row into up to 31 other rows with **>99.98%** success rate
- 2** Can perform **NOT operation** with up to **32 output operands**
- 3** Can perform up to **16-input AND, NAND, OR, and NOR** operations

Functionally-Complete DRAM

- Ismail Emir Yüksel, Yahya Can Tuğrul, Ataberk Olgun, F. Nisa Bostancı, A. Giray Yağlıkçı, Geraldo F. Oliveira, Haocong Luo, Juan Gomez-Luna, Mohammad Sadrosadati, and Onur Mutlu,
"Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis"
Proceedings of the 30th International Symposium on High-Performance Computer Architecture (HPCA), April 2024.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[arXiv version](#)]
[[FCDRAM Source Code](#)]

Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis

Ismail Emir Yüksel Yahya Can Tuğrul Ataberk Olgun F. Nisa Bostancı A. Giray Yağlıkçı
Geraldo F. Oliveira Haocong Luo Juan Gómez-Luna Mohammad Sadrosadati Onur Mutlu

ETH Zürich

Simultaneous Row Activation & Multi-Row Copy

- Ismail Emir Yuksel, Yahya Can Tugrul, F. Nisa Bostanci, Geraldo F. Oliveira, A. Giray Yaglikci, Ataberk Olgun, Melina Soysal, Haocong Luo, Juan Gomez-Luna, Mohammad Sadrosadati, and Onur Mutlu,

"Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis"

Proceedings of the 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Brisbane, Australia, June 2024.

[[Slides \(pptx\)](#) ([pdf](#))]

[[arXiv version](#)]

[[SiMRA-DRAM Source Code](#) (Officially Artifact Evaluated with All Badges)]

Officially artifact evaluated as both code and dataset available, reviewed and reproducible.



Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel¹ Yahya Can Tuğrul^{1,2} F. Nisa Bostancı¹ Geraldo F. Oliveira¹
A. Giray Yağlıkçı¹ Ataberk Olgun¹ Melina Soysal¹ Haocong Luo¹
Juan Gómez-Luna¹ Mohammad Sadrosadati¹ Onur Mutlu¹

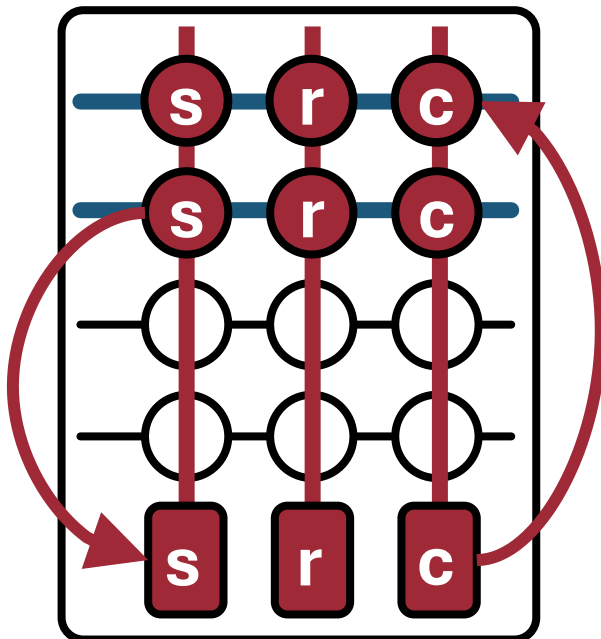
¹ETH Zürich

²TOBB University of Economics and Technology

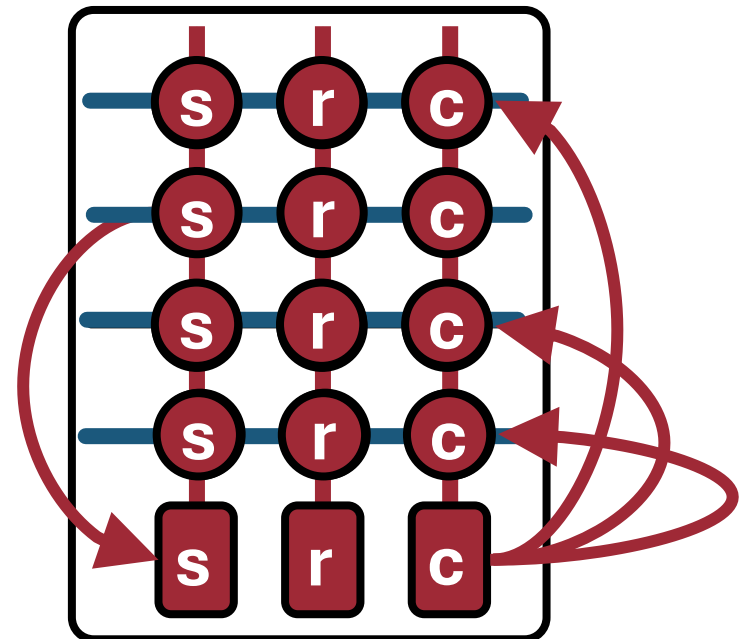
In-DRAM Multiple Row Copy (Multi-RowCopy)

Simultaneously activate many rows to copy **one row's content** to **multiple destination rows**

RowClone

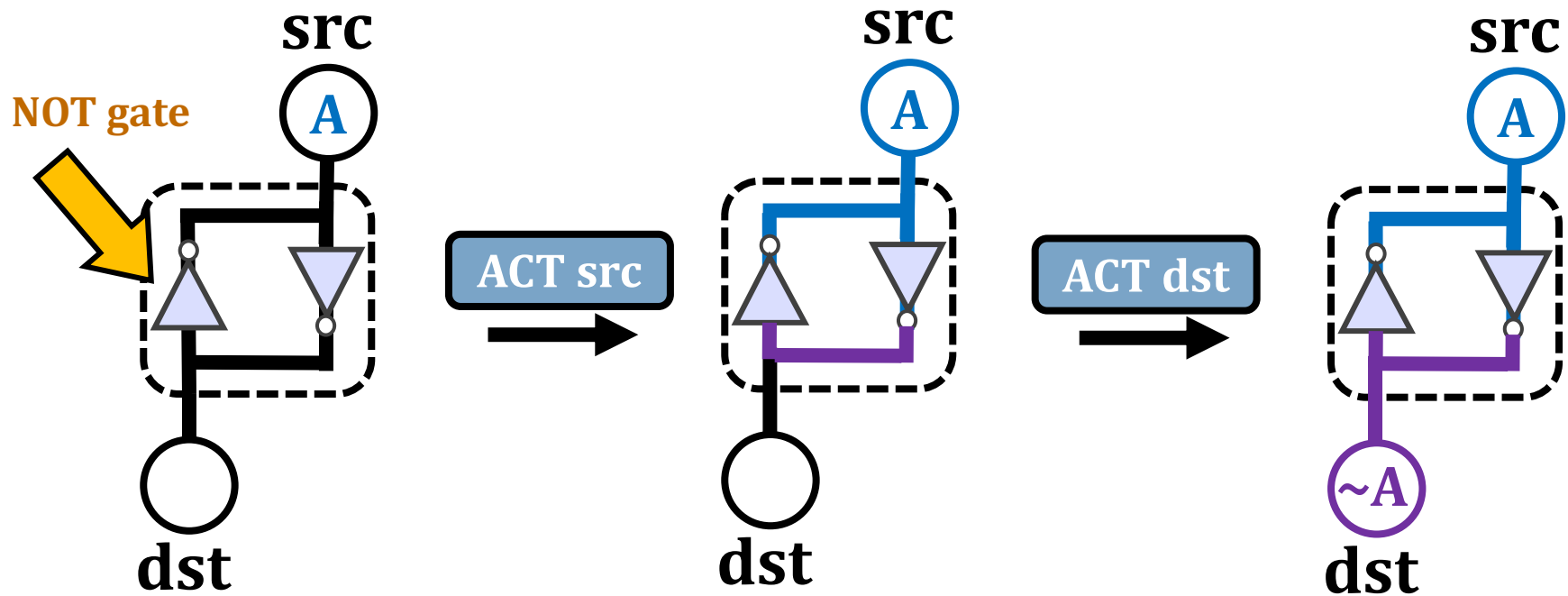


Multi-RowCopy



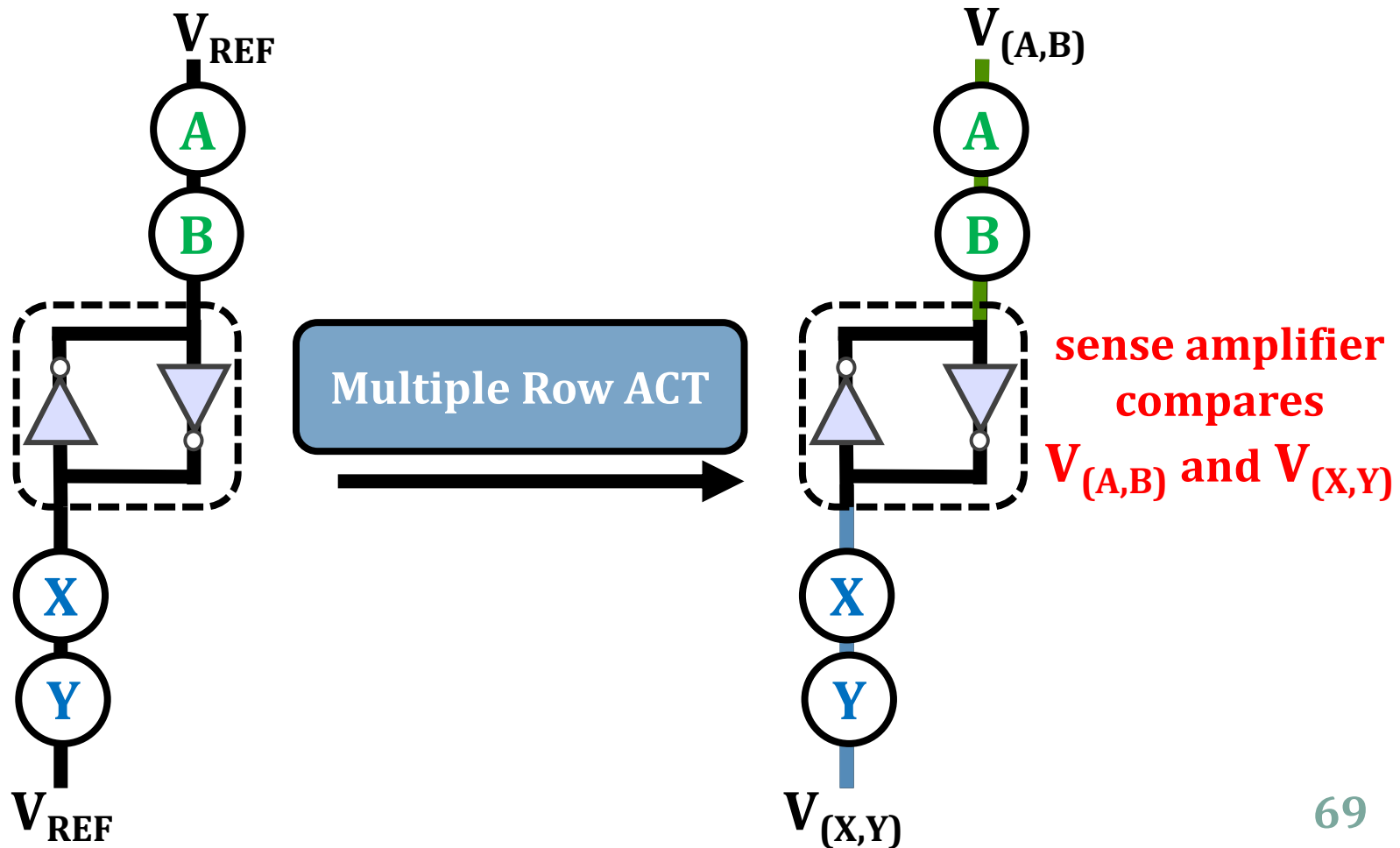
Key Idea: NOT Operation

Connect rows in neighboring subarrays through a **NOT gate** by consecutively activating rows

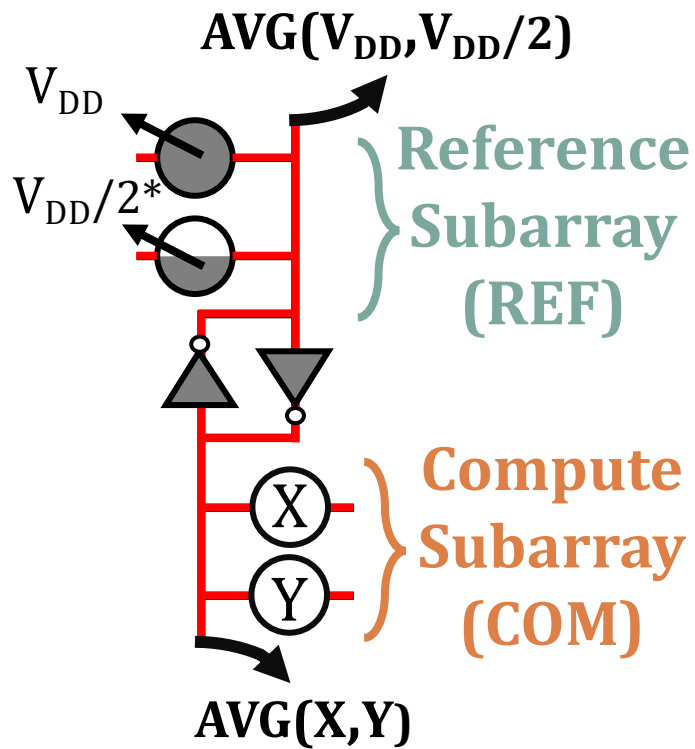


Key Idea: NAND, NOR, AND, OR

Manipulate the bitline voltage to express
a wide variety of functions using
simultaneous multi-row activation in neighboring subarrays



Two-Input AND and NAND Operations



$V_{DD}=1$ & $GND=0$

X	Y	COM	REF
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0
		AND	NAND

Many-Input AND, NAND, OR, and NOR Operations

We can express AND, NAND, OR, and NOR operations by carefully manipulating the reference voltage

Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel Yahya Can Tuğrul Ataberk Olgun F. Nisa Bostancı A. Giray Yağlıkçı
Geraldo F. Oliveira Haocong Luo Juan Gómez-Luna Mohammad Sadrosadati Onur Mutlu

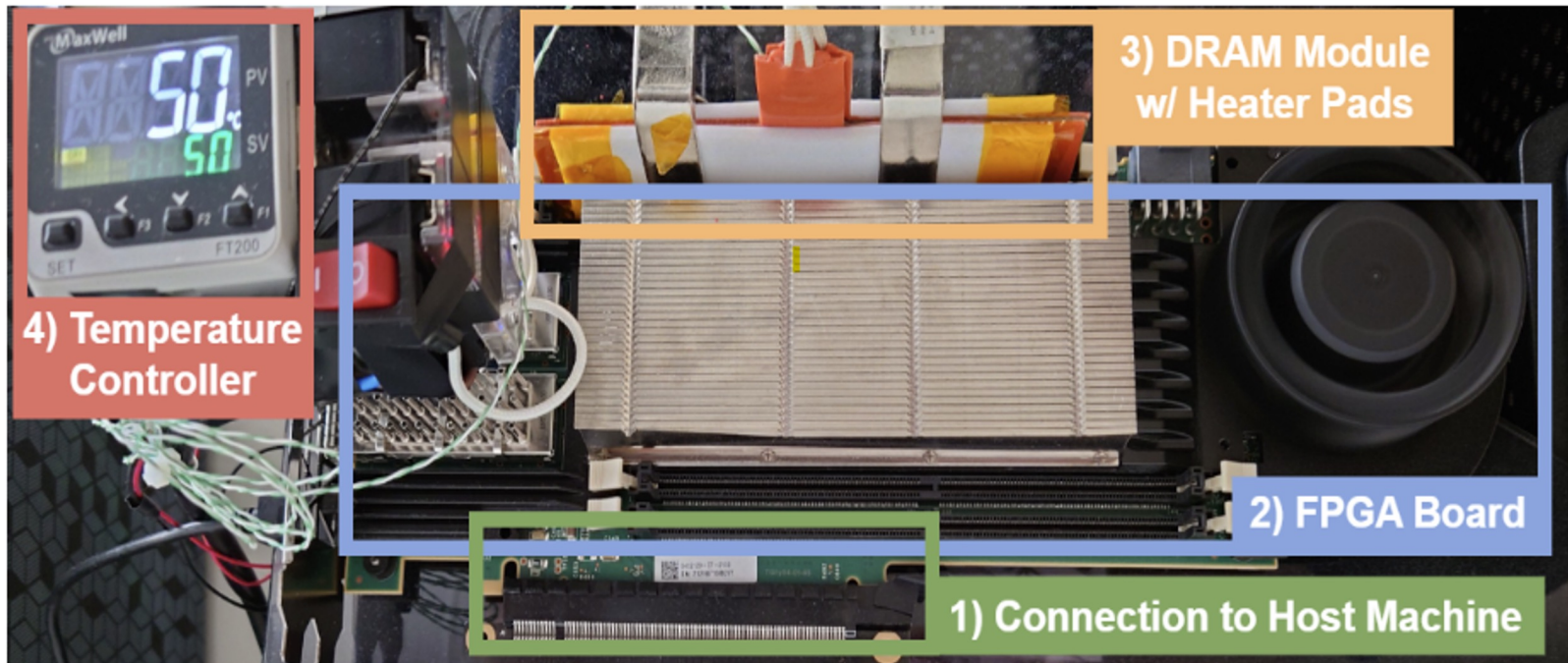
ETH Zürich

(More details in the paper)

<https://arxiv.org/pdf/2402.18736.pdf>

DRAM Testing Infrastructure

- Developed from [DRAM Bender \[Olgun+, TCAD'23\]*](#)
- **Fine-grained control** over DRAM commands, timings, and temperature

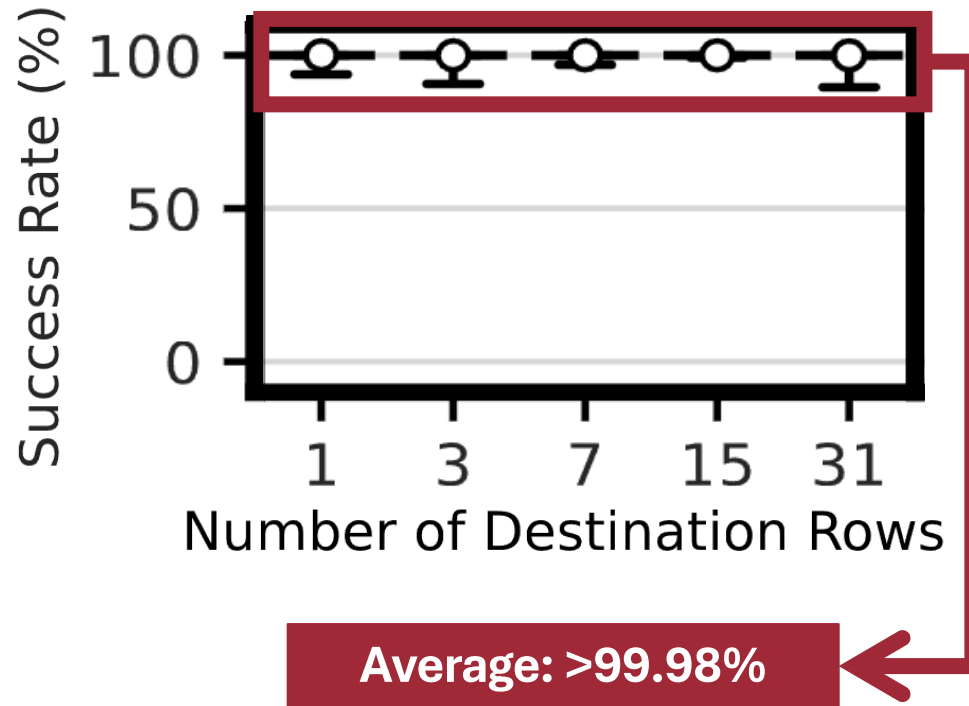


DRAM Chips Tested

- 256 DDR4 chips from two major DRAM manufacturers
- Covers different die revisions and chip densities

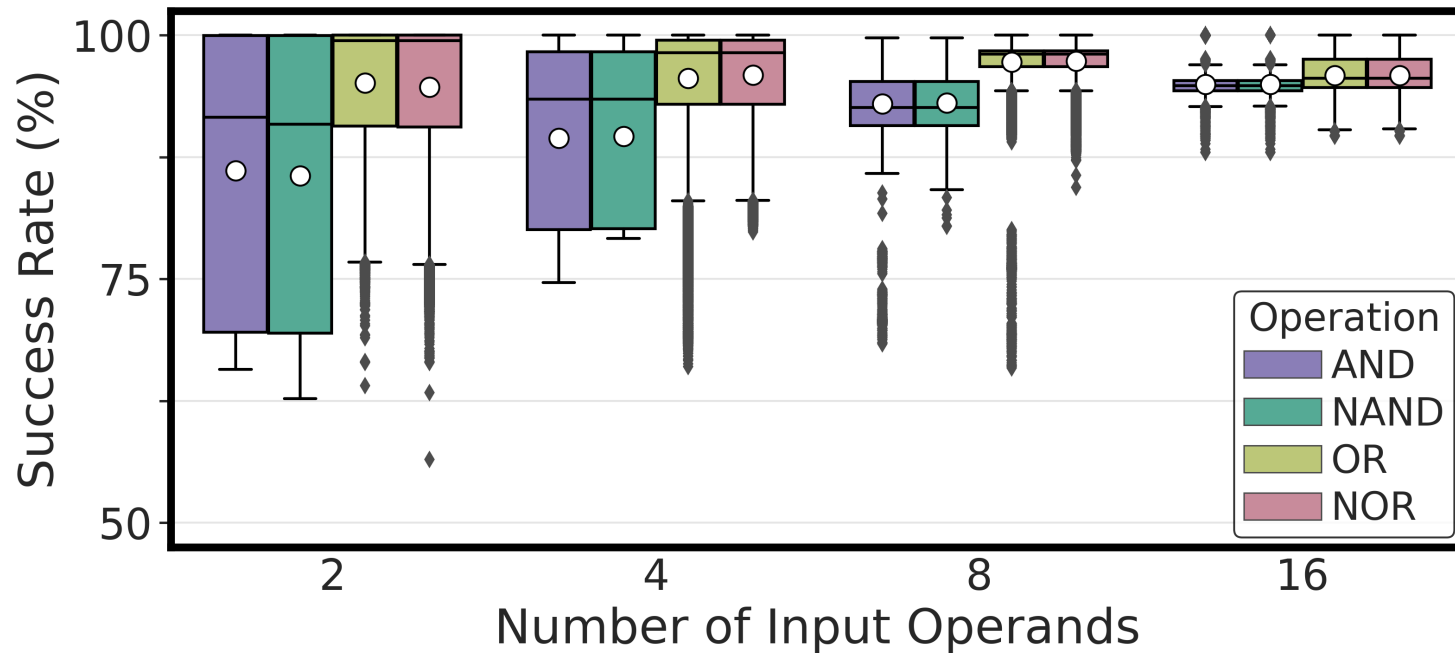
Chip Mfr.	#Modules (#Chips)	Die Rev.	Mfr. Date ^a	Chip Density	Chip Org.	Speed Rate
SK Hynix	9 (72)	M	N/A	4Gb	x8	2666MT/s
	5 (40)	A	N/A	4Gb	x8	2133MT/s
	1 (16)	A	N/A	8Gb	x8	2666MT/s
	1 (32)	A	18-14	4Gb	x4	2400MT/s
	1 (32)	A	16-49	8Gb	x4	2400MT/s
	1 (32)	M	16-22	8Gb	x4	2666MT/s
Samsung	1 (8)	F	21-02	4Gb	x8	2666MT/s
	2 (16)	D	21-10	8Gb	x8	2133MT/s
	1 (8)	A	22-12	8Gb	x8	3200MT/s

Robustness of Multi-RowCopy



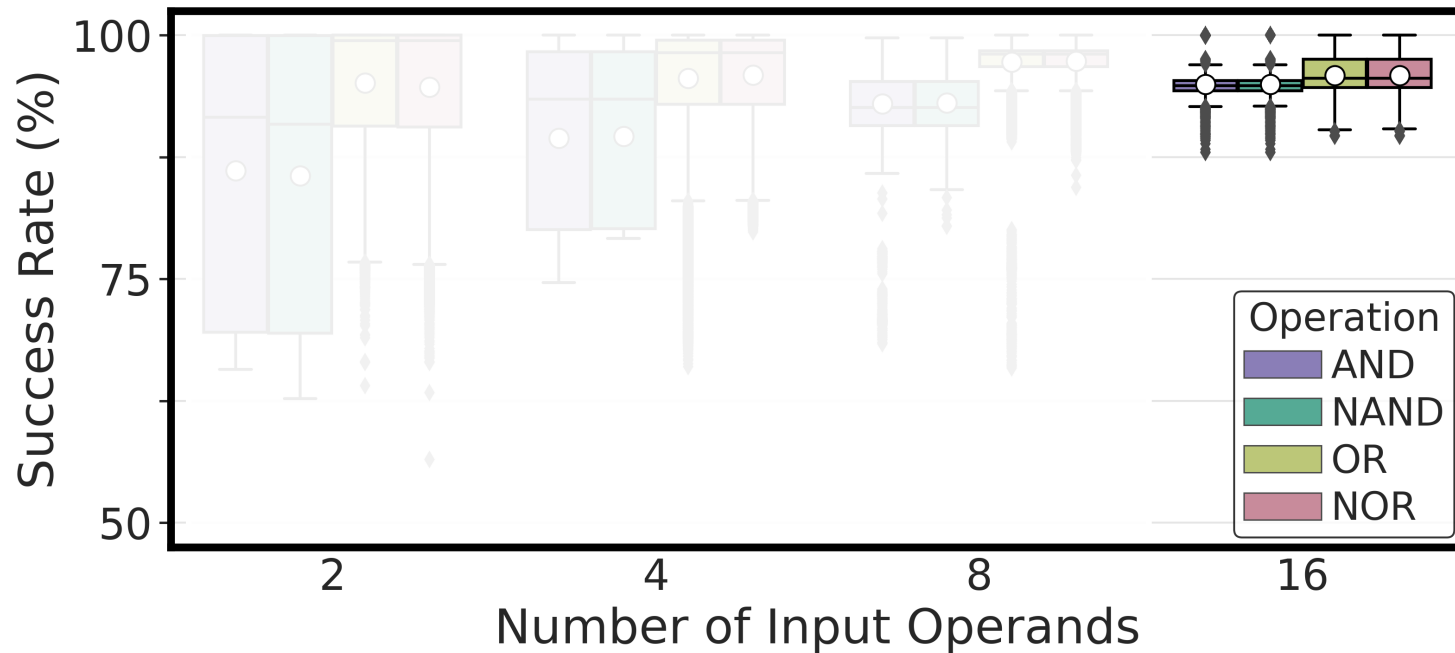
COTS DRAM chips can copy one row's content to up to 31 rows with a very high success rate

Performing AND, NAND, OR, and NOR



COTS DRAM chips can perform {2, 4, 8, 16}-input AND, NAND, OR, and NOR operations

Performing AND, NAND, OR, and NOR



**COTS DRAM chips can perform
16-input AND, NAND, OR, and NOR operations
with very high success rate (>94%)**

More on Multi-Row Copy

- Ismail Emir Yuksel, Yahya Can Tugrul, F. Nisa Bostanci, Geraldo F. Oliveira, A. Giray Yaglikci, Ataberk Olgun, Melina Soysal, Haocong Luo, Juan Gomez-Luna, Mohammad Sadrosadati, and Onur Mutlu,

"Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis"

Proceedings of the 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Brisbane, Australia, June 2024.

[[Slides \(pptx\)](#) ([pdf](#))]

[[arXiv version](#)]

[[SiMRA-DRAM Source Code](#) (Officially Artifact Evaluated with All Badges)]

Officially artifact evaluated as both code and dataset available, reviewed and reproducible.



Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel¹ Yahya Can Tuğrul^{1,2} F. Nisa Bostancı¹ Geraldo F. Oliveira¹
A. Giray Yağlıkçı¹ Ataberk Olgun¹ Melina Soysal¹ Haocong Luo¹
Juan Gómez-Luna¹ Mohammad Sadrosadati¹ Onur Mutlu¹

¹ETH Zürich

²TOBB University of Economics and Technology

More on Functionally-Complete DRAM

- Ismail Emir Yüksel, Yahya Can Tuğrul, Ataberk Olgun, F. Nisa Bostancı, A. Giray Yağlıkçı, Geraldo F. Oliveira, Haocong Luo, Juan Gomez-Luna, Mohammad Sadrosadati, and Onur Mutlu,
"Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis"
Proceedings of the 30th International Symposium on High-Performance Computer Architecture (HPCA), April 2024.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[arXiv version](#)]
[[FCDRAM Source Code](#)]

Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis

Ismail Emir Yüksel Yahya Can Tuğrul Ataberk Olgun F. Nisa Bostancı A. Giray Yağlıkçı
Geraldo F. Oliveira Haocong Luo Juan Gómez-Luna Mohammad Sadrosadati Onur Mutlu

ETH Zürich

An Example Prototyping Platform

Real Processing Using Memory Prototype

- End-to-end RowClone & TRNG using off-the-shelf DRAM chips
- Idea: Violate DRAM timing parameters to mimic RowClone

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun^{§†}

Juan Gómez Luna[§]
Hasan Hassan[§]

Konstantinos Kanellopoulos[§]
Oğuz Ergin[†] Onur Mutlu[§]

Behzad Salami^{§*}

[§]ETH Zürich

[†]TOBB ETÜ

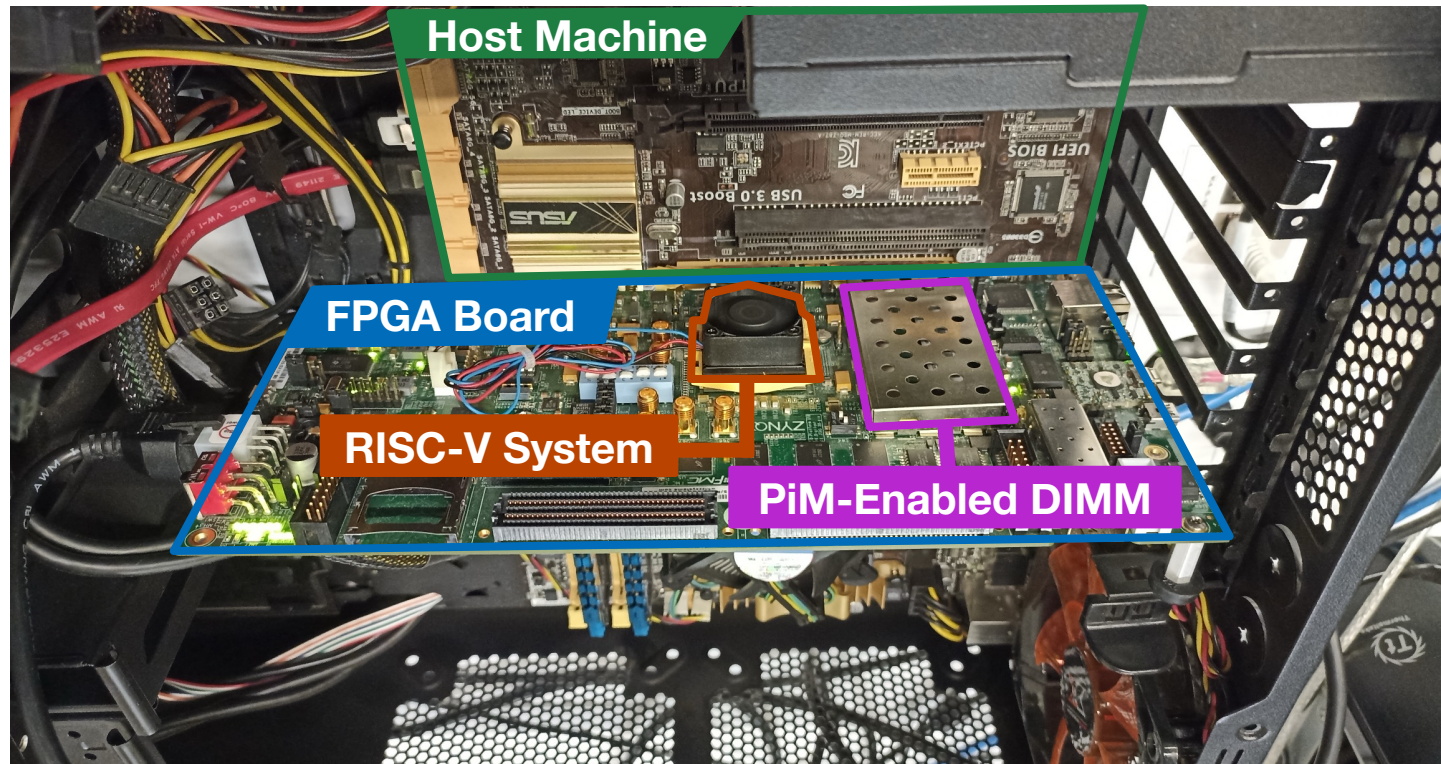
^{*}BSC

<https://arxiv.org/pdf/2111.00082.pdf>

<https://github.com/cmu-safari/pidram>

<https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s>

Real Processing Using Memory Prototype



<https://arxiv.org/pdf/2111.00082.pdf>

<https://github.com/cmu-safari/pidram>

<https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s>

Real Processing Using Memory Prototype

☰ README.md

Building a PiDRAM Prototype

To build PiDRAM's prototype on Xilinx ZC706 boards, developers need to use the two sub-projects in this directory. `fpga-zynq` is a repository branched off of [UCB-BAR's fpga-zynq](#) repository. We use `fpga-zynq` to generate rocket chip designs that support end-to-end DRAM PuM execution. `controller-hardware` is where we keep the main Vivado project and Verilog sources for PiDRAM's memory controller and the top level system design.

Rebuilding Steps

1. Navigate into `fpga-zynq` and read the README file to understand the overall workflow of the repository
 - Follow the readme in `fpga-zynq/rocket-chip/riscv-tools` to install dependencies
2. Create the Verilog source of the rocket chip design using the `ZynqCopyFPGAConfig`
 - Navigate into `zc706`, then run `make rocket CONFIG=ZynqCopyFPGAConfig -j<number of cores>`
3. Copy the generated Verilog file (should be under `zc706/src`) and overwrite the same file in `controller-hardware/source/hdl/impl/rocket-chip`
4. Open the Vivado project in `controller-hardware/Vivado_Project` using Vivado 2016.2
5. Generate a bitstream
6. Copy the bitstream (`system_top.bit`) to `fpga-zynq/zc706`
7. Use the `./build_script.sh` to generate the new `boot.bin` under `fpga-images-zc706`, you can use this file to program the FPGA using the SD-Card
 - For details, follow the relevant instructions in `fpga-zynq/README.md`

You can run programs compiled with the RISC-V Toolchain supplied within the `fpga-zynq` repository. To install the toolchain, follow the instructions under `fpga-zynq/rocket-chip/riscv-tools`.

Generating DDR3 Controller IP sources

We cannot provide the sources for the Xilinx PHY IP we use in PiDRAM's memory controller due to licensing issues. We describe here how to regenerate them using Vivado 2016.2. First, you need to generate the IP RTL files:

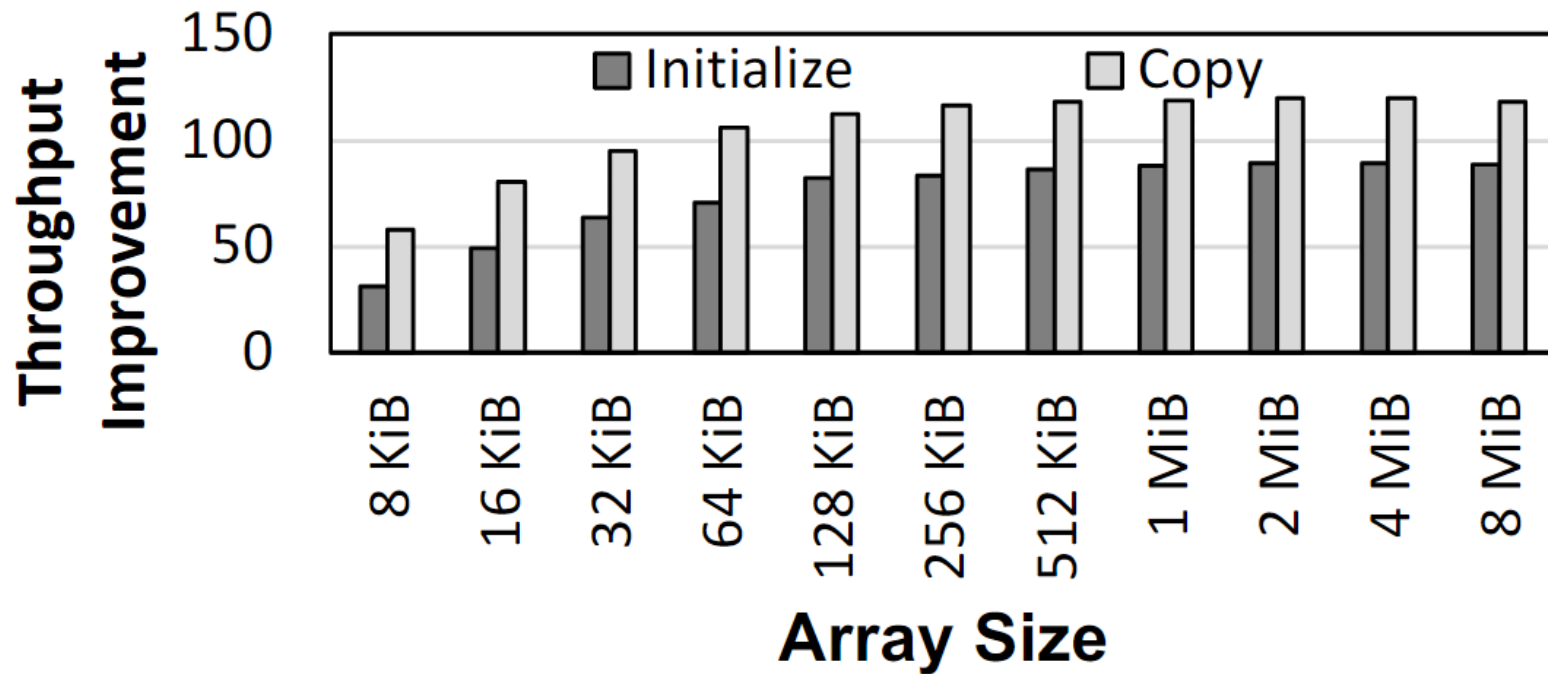
- 1- Open IP Catalog
- 2- Find "Memory Interface Generator (MIG 7 Series)" IP and double click

<https://arxiv.org/pdf/2111.00082.pdf>

<https://github.com/cmu-safari/pidram>

<https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s>

Microbenchmark Copy/Initialization Throughput



**In-DRAM Copy and Initialization
improve throughput by 119x and 89x**

More on PiDRAM

- Ataberk Olgun, Juan Gomez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oguz Ergin, and Onur Mutlu,
"PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM"
ACM Transactions on Architecture and Code Optimization (TACO), March 2023.
[[arXiv version](#)]
Presented at the [18th HiPEAC Conference](#), Toulouse, France, January 2023.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Longer Lecture Slides \(pptx\)](#)] [[pdf](#)]
[[Lecture Video](#) (40 minutes)]
[[PiDRAM Source Code](#)]

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]ETH Zürich

[†]TOBB University of Economics and Technology

Simulation: The Field of Dreams

Dreaming and Reality

- An architect is in part a dreamer, a creator
- Simulation is a key tool of the architect
 - Allows the evaluation & understanding of non-existent systems
- Simulation enables
 - The exploration of many dreams
 - A reality check of the dreams
 - Deciding which dream is better
- Simulation also enables
 - The ability to fool yourself with false dreams

Tradeoffs in Simulation

- Three metrics to evaluate a simulator
 - Speed
 - Flexibility
 - Accuracy
- Speed: How fast the simulator runs (xIPS, xCPS, slowdown)
- Flexibility: How quickly one can modify the simulator to evaluate different algorithms and design choices?
- Accuracy: How accurate the performance (energy) numbers the simulator generates are vs. a real design (Simulation error)
- The relative importance of these metrics varies depending on where you are in the design process (what your goal is)

Trading Off Speed, Flexibility, Accuracy

- Speed & flexibility affect:
 - How quickly you can make design tradeoffs/decisions
- Accuracy affects:
 - How good your design tradeoffs/decisions **might** end up being
 - How fast you can build your simulator (**simulator design time**)
- Flexibility also affects:
 - How much human effort you need to spend modifying the simulator
- You can **trade off between the three to achieve design exploration and decision goals**

Making The Best of Architecture

- A good architect is comfortable at all levels of refinement
 - Including the extremes
- A good architect knows when to use what type of simulation
 - And, more generally, what type of evaluation method
- Recall: A variety of evaluation methods are available:
 - Theoretical proof
 - Analytical modeling
 - Simulation (at varying degrees of abstraction and accuracy)
 - Prototyping with a real system (e.g., FPGAs)
 - Real implementation

More on Simulation: A Lecture

Computer Architecture Lecture 18a: Simulation

Prof. Onur Mutlu

ETH Zürich
Fall 2025
21 November 2025



oom

Comp. Arch. - Lecture 18: Simulation and Prefetching (Fall 2025)



Onur Mutlu Lectures
60.1K subscribers

Analytics

Edit video

👍 36



🔗 Share

📢 Promote

💎 Ask



1,248 views Streamed live on Nov 21, 2025 [Livestream - Computer Architecture \(Fall 2025\)](#)
Computer Architecture, ETH Zürich, Fall 2025 (Course page: <https://safari.ethz.ch/architecture/f...>)

<https://www.youtube.com/watch?v=IJ5UpGvdm4Y>

Ramulator: A **Fast** and **Extensible** DRAM Simulator

[IEEE Comp Arch Letters'15]

Ramulator Motivation

- DRAM and Memory Controller landscape is changing
- Many new and upcoming standards
- Many new controller designs
- A **fast** and **easy-to-extend** simulator is very much needed

<i>Segment</i>	<i>DRAM Standards & Architectures</i>
Commodity	DDR3 (2007) [14]; DDR4 (2012) [18]
Low-Power	LPDDR3 (2012) [17]; LPDDR4 (2014) [20]
Graphics	GDDR5 (2009) [15]
Performance	eDRAM [28], [32]; RLDram3 (2011) [29]
3D-Stacked	WIO (2011) [16]; WIO2 (2014) [21]; MCDRAM (2015) [13]; HBM (2013) [19]; HMC1.0 (2013) [10]; HMC1.1 (2014) [11]
Academic	SBA/SSA (2010) [38]; Staged Reads (2012) [8]; RAIDR (2012) [27]; SALP (2012) [24]; TL-DRAM (2013) [26]; RowClone (2013) [37]; Half-DRAM (2014) [39]; Row-Buffer Decoupling (2014) [33]; SARP (2014) [6]; AL-DRAM (2015) [25]

Table 1. Landscape of DRAM-based memory

Ramulator

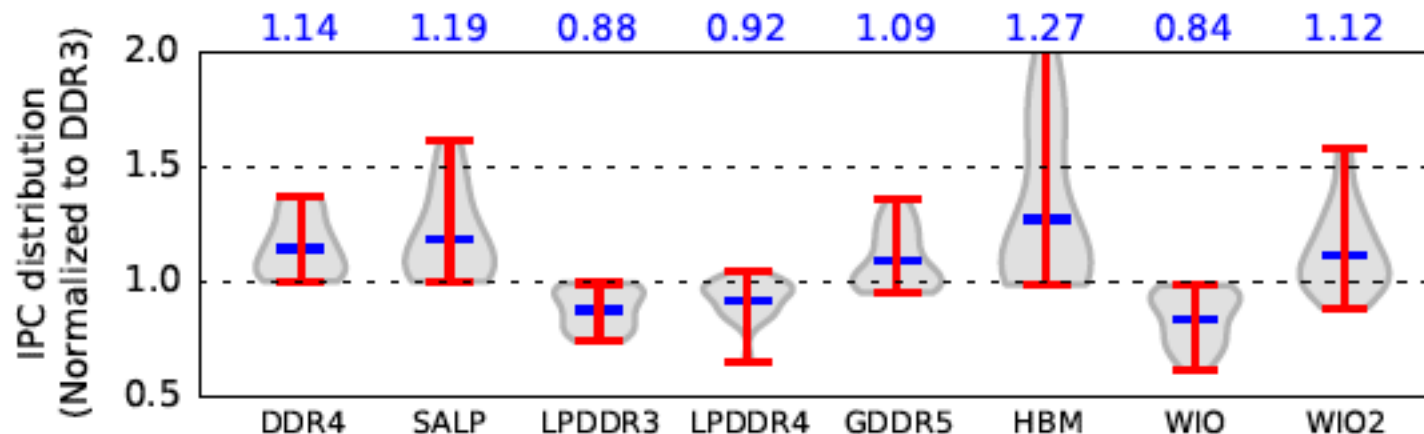
- Provides out-of-the box support for many DRAM standards:
 - DDR3/4, LPDDR3/4, GDDR5, WIO1/2, HBM, plus new proposals (SALP, AL-DRAM, TLDRAM, RowClone, and SARP)
- ~2.5X faster than fastest open-source simulator
- Modular and extensible to different standards

<i>Simulator</i> (clang -O3)	<i>Cycles (10⁶)</i>		<i>Runtime (sec.)</i>		<i>Req/sec (10³)</i>		<i>Memory</i> (MB)
	<i>Random</i>	<i>Stream</i>	<i>Random</i>	<i>Stream</i>	<i>Random</i>	<i>Stream</i>	
Ramulator	652	411	752	249	133	402	2.1
DRAMSim2	645	413	2,030	876	49	114	1.2
USIMM	661	409	1,880	750	53	133	4.5
DrSim	647	406	18,109	12,984	6	8	1.6
NVMain	666	413	6,881	5,023	15	20	4,230.0

Table 3. Comparison of five simulators using two traces

Case Study: Comparison of DRAM Standards

<i>Standard</i>	<i>Rate (MT/s)</i>	<i>Timing (CL-RCD-RP)</i>	<i>Data-Bus (Width×Chan.)</i>	<i>Rank-per-Chan</i>	<i>BW (GB/s)</i>
DDR3	1,600	11-11-11	64-bit × 1	1	11.9
DDR4	2,400	16-16-16	64-bit × 1	1	17.9
SALP [†]	1,600	11-11-11	64-bit × 1	1	11.9
LPDDR3	1,600	12-15-15	64-bit × 1	1	11.9
LPDDR4	2,400	22-22-22	32-bit × 2*	1	17.9
GDDR5 [12]	6,000	18-18-18	64-bit × 1	1	44.7
HBM	1,000	7-7-7	128-bit × 8*	1	119.2
WIO	266	7-7-7	128-bit × 4*	1	15.9
WIO2	1,066	9-10-10	128-bit × 8*	1	127.2



Across 22 workloads, simple CPU model

Figure 2. Performance comparison of DRAM standards

Ramulator Paper and Source Code

- Yoongu Kim, Weikun Yang, and Onur Mutlu,
"Ramulator: A Fast and Extensible DRAM Simulator"
IEEE Computer Architecture Letters (**CAL**), March 2015.
[Source Code]
- Source code is released under the liberal MIT License
 - <https://github.com/CMU-SAFARI/ramulator>

Ramulator: A Fast and Extensible DRAM Simulator

Yoongu Kim¹ Weikun Yang^{1,2} Onur Mutlu¹
¹Carnegie Mellon University ²Peking University

Ramulator: Free & Open Source

☰ README.md

Ramulator: A DRAM Simulator

Ramulator is a fast and cycle-accurate DRAM simulator [1, 2] that supports a wide array of commercial, as well as academic, DRAM standards:

- DDR3 (2007), DDR4 (2012)
- LPDDR3 (2012), LPDDR4 (2014)
- GDDR5 (2009)
- WIO (2011), WIO2 (2014)
- HBM (2013)
- SALP [3]
- TL-DRAM [4]
- RowClone [5]
- DSARP [6]

The initial release of Ramulator is described in the following paper:

Y. Kim, W. Yang, O. Mutlu. "[Ramulator: A Fast and Extensible DRAM Simulator](#)". In *IEEE Computer Architecture Letters*, March 2015.

For information on new features, along with an extensive memory characterization using Ramulator, please read:

S. Ghose, T. Li, N. Hajinazar, D. Senol Cali, O. Mutlu. "[Demystifying Complex Workload-DRAM Interactions: An Experimental Study](#)". In *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2019 ([slides](#)). In *Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS)*, 2019.

[1] Kim et al. *Ramulator: A Fast and Extensible DRAM Simulator*. IEEE CAL 2015.

[2] Ghose et al. *Demystifying Complex Workload-DRAM Interactions: An Experimental Study*. SIGMETRICS 2019.

[3] Kim et al. *A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM*. ISCA 2012.

[4] Lee et al. *Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture*. HPCA 2013.

[5] Seshadri et al. *RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization*. MICRO 2013.

[6] Chang et al. *Improving DRAM Performance by Parallelizing Refreshes with Accesses*. HPCA 2014.

Usage

Ramulator supports three different usage modes.

1. **Memory Trace Driven:** Ramulator directly reads memory traces from a file, and simulates only the DRAM subsystem. Each line in the trace file represents a memory request, with the hexadecimal address followed by 'R' or 'W' for read or write.

- 0x12345680 R
- 0x4cbd56c0 W

• ...

Ramulator: Integration with Other Simulators

☰ README.md

Usage

Ramulator supports three different usage modes.

1. **Memory Trace Driven:** Ramulator directly reads memory traces from a file, and simulates only the DRAM subsystem. Each line in the trace file represents a memory request, with the hexadecimal address followed by 'R' or 'W' for read or write.

- 0x12345680 R
- 0x4cbd56c0 W
- ...

2. **CPU Trace Driven:** Ramulator directly reads instruction traces from a file, and simulates a simplified model of a "core" that generates memory requests to the DRAM subsystem. Each line in the trace file represents a memory request, and can have one of the following two formats.

- `<num-cpuinst> <addr-read>` : For a line with two tokens, the first token represents the number of CPU (i.e., non-memory) instructions before the memory request, and the second token is the decimal address of a *read*.
- `<num-cpuinst> <addr-read> <addr-writeback>` : For a line with three tokens, the third token is the decimal address of the *writeback* request, which is the dirty cache-line eviction caused by the read request before it.

3. **gem5 Driven:** Ramulator runs as part of a full-system simulator (gem5 [7]), from which it receives memory request as they are generated.

For some of the DRAM standards, Ramulator is also capable of reporting power consumption by relying on either VAMPIRE [8] or DRAMPower [9] as the backend.

[7] The gem5 Simulator System.

[8] Ghose et al. *What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study*. SIGMETRICS 2018.

[9] Chandrasekar et al. *DRAMPower: Open-Source DRAM Power & Energy Estimation Tool*. IEEE CAL 2015.

Getting Started

Ramulator requires a C++11 compiler (e.g., `clang++`, `g++-5`).

1. **Memory Trace Driven**

```
$ cd ramulator
$ make -j
$ ./ramulator configs/DDR3-config.cfg --mode=dram dram.trace
Simulation done. Statistics written to DDR3.stats
# NOTE: dram.trace is a very short trace file provided only as an example.
$ ./ramulator configs/DDR3-config.cfg --mode=dram --stats my_output.txt dram.trace
Simulation done. Statistics written to my_output.txt
# NOTE: optional --stats flag changes the statistics output filename
```

2. **CPU Trace Driven**

Ramulator: Reproducibility

Reproducing Results from Paper (Kim et al. [1])

Debugging & Verification (Section 4.1)

For debugging and verification purposes, Ramulator can print the trace of every DRAM command it issues along with their address and timing information. To do so, please turn on the `print_cmd_trace` variable in the configuration file.

Comparison Against Other Simulators (Section 4.2)

For comparing Ramulator against other DRAM simulators, we provide a script that automates the process: `test_ddr3.py`. Before you run this script, however, you must specify the location of their executables and configuration files at designated lines in the script's source code:

- Ramulator
- DRAMSim2 (<https://wiki.umd.edu/DRAMSim2>): `test_ddr3.py` lines 39-40
- USIMM (<http://www.cs.utah.edu/~rajeew/jwac12>): `test_ddr3.py` lines 54-55
- DrSim (<http://ph.ece.utexas.edu/public/Main/DrSim>): `test_ddr3.py` lines 66-67
- NVMain (<http://wiki.nvmain.org>): `test_ddr3.py` lines 78-79

Please refer to their respective websites to download, build, and set-up the other simulators. The simulators must be executed in saturation mode (always filling up the request queues when possible).

All five simulators were configured using the same parameters:

- DDR3-1600K (11-11-11), 1 Channel, 1 Rank, 2Gb x8 chips
- FR-FCFS Scheduling
- Open-Row Policy
- 32/32 Entry Read/Write Queues
- High/Low Watermarks for Write Queue: 28/16

Finally, execute `test_ddr3.py <num-requests>` to start off the simulation. Please make sure that there are no other active processes during simulation to yield accurate measurements of memory usage and CPU time.

Cross-Sectional Study of DRAM Standards (Section 4.3)

Please use the CPU traces (SPEC 2006) provided in the `cpustraces` folder to run CPU trace driven simulations.

Other Tips

Power Estimation

For estimating power consumption, Ramulator can record the trace of every DRAM command it issues to a file in DRAMPower [8] format. To do so, please turn on the `record_cmd_trace` variable in the configuration file. The resulting DRAM command trace (e.g., `cmd-trace-chan-N-rank-M.cmdtrace`) should be fed into a compatible DRAM energy simulator such as VAMPIRE [8] or DRAMPower [9] with the correct configuration (standard/speed/organization) to estimate energy/power usage for a single rank (a current limitation of both VAMPIRE and DRAMPower).

An Example Study using Ramulator (I)

- Saugata Ghose, Tianshi Li, Nastaran Hajinazar, Damla Senol Cali, and Onur Mutlu, **"Demystifying Workload–DRAM Interactions: An Experimental Study"** *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Phoenix, AZ, USA, June 2019.
[[Preliminary arXiv Version](#)]
[[Abstract](#)]
[[Slides \(pptx\)](#) ([pdf](#))]
[[MemBen Benchmark Suite](#)]
[[Source Code for GPGPUSim-Ramulator](#)]

Demystifying Complex Workload–DRAM Interactions: An Experimental Study

Saugata Ghose[†]

Tianshi Li[†]

Nastaran Hajinazar^{‡†}

Damla Senol Cali[†]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[‡]Simon Fraser University

[§]ETH Zürich

1. DRAM latency remains a critical bottleneck for many applications
2. Bank parallelism is not fully utilized by a wide variety of our applications
3. Spatial locality continues to provide significant performance benefits if it is exploited by the memory subsystem
4. For some classes of applications, low-power memory can provide energy savings without sacrificing significant performance

BlockHammer Study in 2021

- A. Giray Yaglikci, Minesh Patel, Jeremie S. Kim, Roknoddin Azizi, Ataberk Olgun, Lois Orosa, Hasan Hassan, Jisung Park, Konstantinos Kanellopoulos, Taha Shahroodi, Saugata Ghose, and Onur Mutlu,

"BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows"

Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Short Talk Slides \(pptx\)](#) ([pdf](#))]

[[Intel Hardware Security Academic Awards Short Talk Slides \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (22 minutes)]

[[Short Talk Video](#) (7 minutes)]

[[Intel Hardware Security Academic Awards Short Talk Video](#) (2 minutes)]

[[BlockHammer Source Code](#)]

Intel Hardware Security Academic Award Finalist (one of 4 finalists out of 34 nominations)

BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows

A. Giray Yağlıkçı¹ Minesh Patel¹ Jeremie S. Kim¹ Roknoddin Azizi¹ Ataberk Olgun¹ Lois Orosa¹
Hasan Hassan¹ Jisung Park¹ Konstantinos Kanellopoulos¹ Taha Shahroodi¹ Saugata Ghose² Onur Mutlu¹

¹ETH Zürich

²University of Illinois at Urbana-Champaign

BlockHammer: Free & Open Source

☰ README.md

BlockHammer

Aggressive memory density scaling causes modern DRAM devices to suffer from [RowHammer](#), a phenomenon where rapidly activating a DRAM row can cause bit-flips in physically-nearby rows. Recent studies [1, 2, 3] demonstrate that modern DRAM chips, including chips previously marketed as RowHammer-safe, are even more vulnerable to RowHammer than older chips. Many works show that attackers can exploit RowHammer bit-flips to reliably mount system-level attacks to escalate privilege and leak private data. Therefore, it is critical to ensure RowHammer-safe operation on all DRAM-based systems.

Unfortunately, state-of-the-art RowHammer mitigation mechanisms face two major challenges. First, they incur increasingly higher performance and/or area overheads when applied to more vulnerable DRAM chips. Second, they require either proprietary information about or modifications to the DRAM chip design. In this paper, we show that it is possible to efficiently and scalably prevent RowHammer bit-flips without knowledge of or modification to DRAM internals.

We introduce BlockHammer, a low-cost, effective, and easy-to-adopt RowHammer mitigation mechanism that overcomes the two key challenges by selectively throttling memory accesses that could otherwise cause RowHammer bit-flips. The key idea of BlockHammer is to (1) track row activation rates using area-efficient Bloom filters and (2) use the tracking data to ensure that no row is ever activated rapidly enough to induce RowHammer bit-flips. By doing so, BlockHammer (1) makes it impossible for a RowHammer bit-flip to occur and (2) greatly reduces a RowHammer attack's impact on the performance of co-running benign applications.

Compared to state-of-the-art RowHammer mitigation mechanisms, BlockHammer provides competitive performance and energy when the system is not under a RowHammer attack and significantly better performance and energy when the system is under attack.

The full paper is published in [HPCA 2021](#) and the pdf is available online: [arXiv:2102.05981 \[cs.CR\]](#)

Citation

Please cite our full HPCA 2021 paper if you find this repository useful.

A. Giray Yaglikci, Minesh Patel, Jeremie S. Kim, Roknoddin Azizi, Ataberk Olgun, Lois Orosa, Hasan Hassan, Jisung Park, Konstantinos Kanellopoulos, Taha Shahroodi, Saugata Ghose, and Onur Mutlu, "[BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows](#)" Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.

Getting Started

This repository has two subdirectories. Please refer to each subdirectory on reproducing results.

- [Ramulator Model](#): This subdirectory includes an extended version of [Ramulator](#) with a RowHammerDefense class, which implements BlockHammer along with six state-of-the-art RowHammer mitigation mechanisms:

Mechanism	Reference
PARA	Y. Kim, et al., "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," in ISCA, 2014.
CBT	S. M. Seyedzadeh et al., "Mitigating Wordline Crosstalk Using Adaptive Trees of Counters," in ISCA, 2018.
ProHIT	M. Son et al., "Making DRAM Stronger Against Row Hammering," in DAC, 2017.
MrLoc	J. M. You et al., "MRLoc: Mitigating Row-Hammering Based on Memory Locality," in DAC, 2019.
TWjCe	E. Lee et al., "TWjCe: Preventing Row-hammering by Exploiting Time Window Counters" in ISCA, 2019.
Graphene	Y. Park et al., "Graphene: Strong yet Lightweight Row Hammer Protection," in MICRO, 2020.

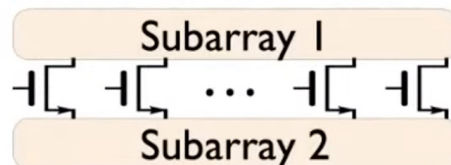
- [RTL Model](#): This subdirectory includes RTL implementation of the counters and buffers used in BlockHammer.

Many Other Ideas Evaluated w/ Ramulator

Key Idea and Applications

- **Low-cost Inter-linked subarrays (LISA)**

- Fast bulk data movement between subarrays
- Wide datapath via isolation transistors: 0.8% DRAM chip area



- LISA is a **versatile substrate** → new applications

Fast bulk data copy: Copy latency 1.363ms → 0.148ms (9.2x)

→ 66% speedup, -55% DRAM energy

In-DRAM caching: Hot data access latency 48.7ns → 21.5ns (2.2x)

→ 5% speedup

Fast precharge: Precharge latency 13.1ns → 5.0ns (2.6x)

→ 8% speedup



zoom

Computer Architecture - Lecture 9: Memory Latency & Memory Controllers (Fall 2022)



Onur Mutlu Lectures
28.8K subscribers

Analytics

Edit video

29



Share



Download



Clip



Save



711 views Streamed live on Oct 27, 2022

Computer Architecture, ETH Zürich, Fall 2022 (<https://safari.ethz.ch/architecture/f...>)

Ramulator Project Course

Exploration of Emerging Memory Systems (Spring/Fall 2022)

Fall 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator

Spring 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator

Youtube Livestream (Spring 2022):

- ❑ https://www.youtube.com/watch?v=aM-lIXRQd3s&list=PL5Q2soXY2Zi_TlmlGw_Z8hBo2925ZApgV

Bachelor's course

- ❑ Elective at ETH Zurich
- ❑ Introduction to memory system simulation
- ❑ Tutorial on using Ramulator
- ❑ C++
- ❑ Potential research exploration

<https://www.youtube.com/onurmutlulectures>

Lecture Video Playlist on YouTube

Lecture Playlist



2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	09.03 Wed.	YouTube Video	M1: Logistics & Intro to Simulating Memory Systems Using Ramulator PDF (PDF) PPT (PPT)		HW0
W2	16.03 Fri.	YouTube Video	M2: Tutorial on Using Ramulator PDF (PDF) PPT (PPT)		
W3	25.02 Fri.	YouTube Video	M3: BlockHammer PDF (PDF) PPT (PPT)		
W4	01.04 Fri.	YouTube Video	M4: CLR-DRAM PDF (PDF) PPT (PPT)		
W5	08.04 Fri.	YouTube Video	M5: SIMDRAM PDF (PDF) PPT (PPT)		
W6	29.04 Fri.	YouTube Video	M6: DAMOV PDF (PDF) PPT (PPT)		
W7	06.05 Fri.	YouTube Video	M7: Synchron PDF (PDF) PPT (PPT)		

Ramulator 2.0: A Much Better Ramulator

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
"Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
*Preprint on **arxiv**, August 2023.*
[[arXiv version](#)]
[[Ramulator 2.0 Source Code](#)]

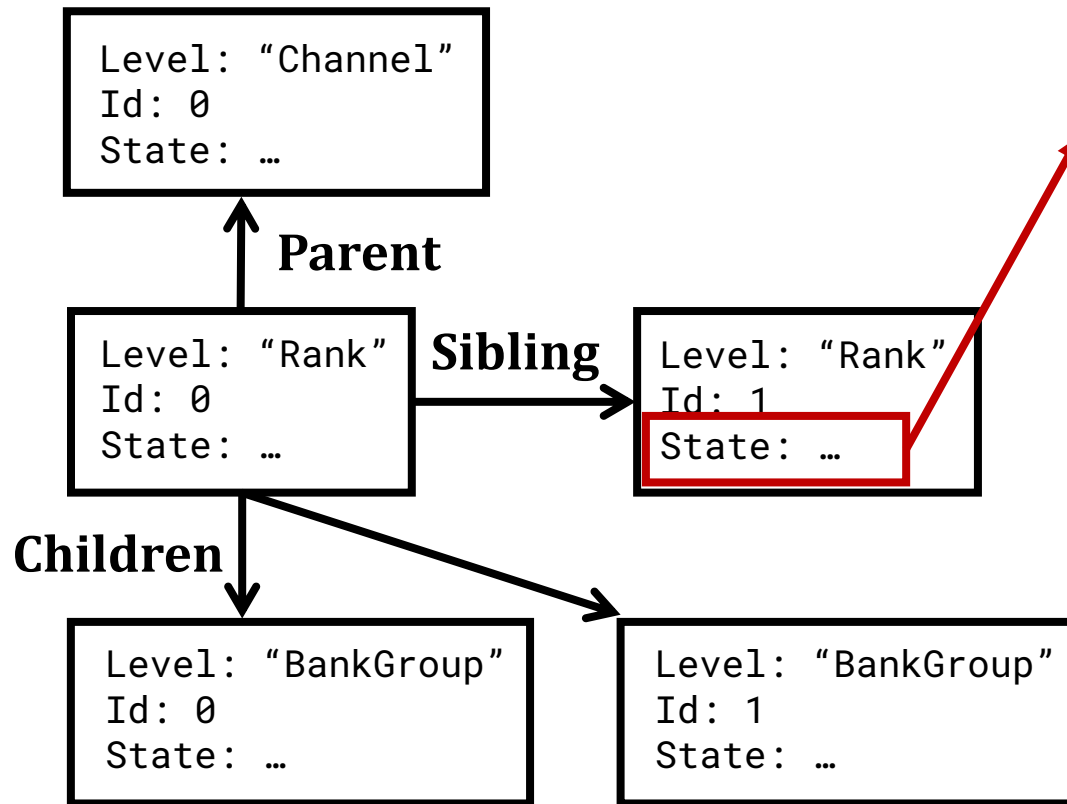
Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

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

<https://arxiv.org/pdf/2308.11030.pdf>

Ramulator 2.0 Design and Features (I)

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

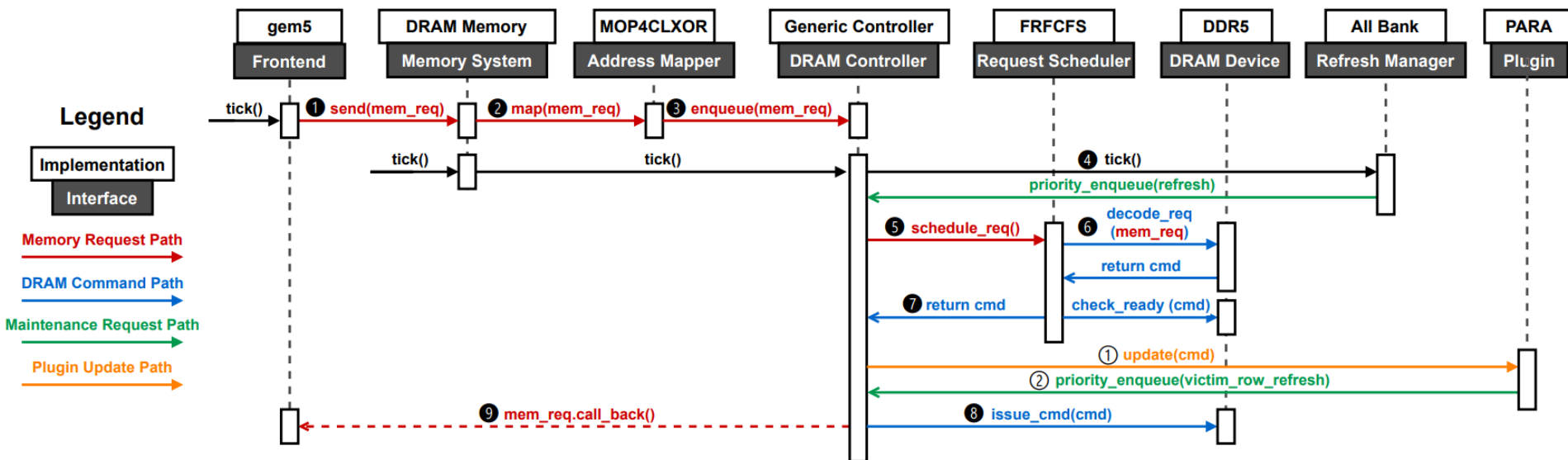


State of a DRAM node:

- **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 (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, ...
 - Each can be **flexibly changed** without hardcoding other parts



Ramulator 2.0 Design and Features (IV)

- 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



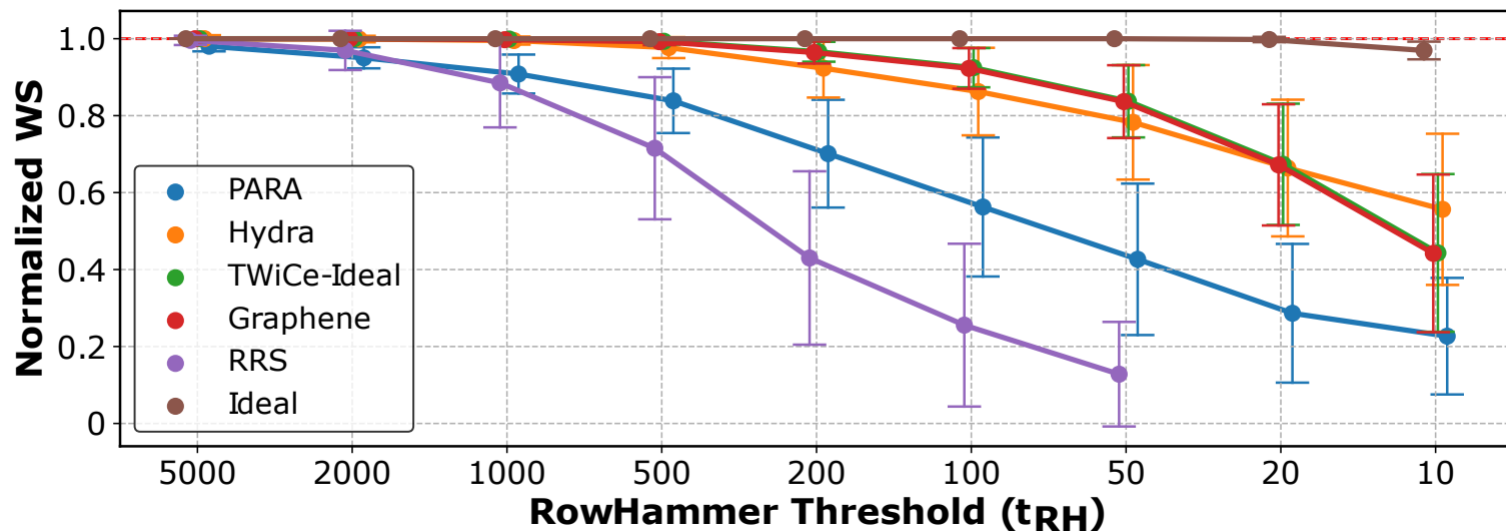
Open-source version Github repo:
CMU-SAFARI/ramulator2

Ramulator 2.0 Case Studies (I)

- **Cross-section performance overhead evaluation of different RowHammer mitigation techniques**

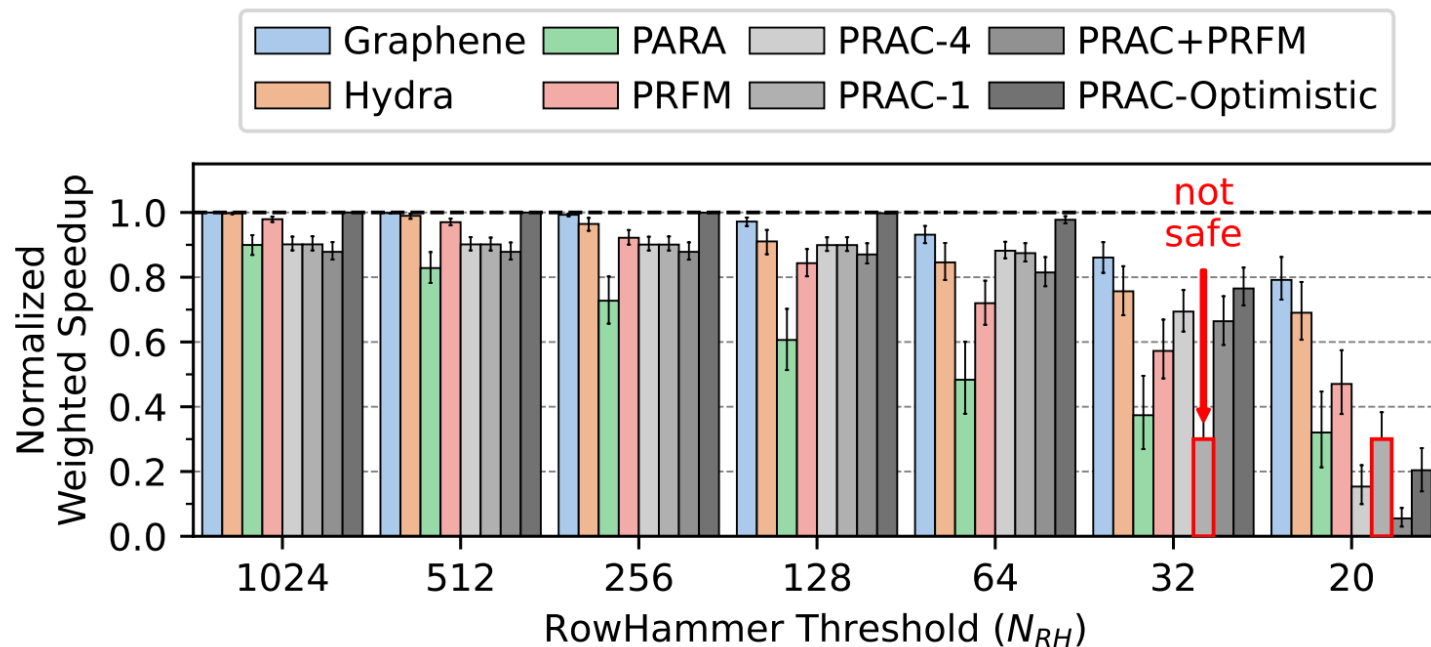
[Luo+, IEEE CAL]

- Six different RowHammer mitigation techniques all implemented **as plugins to the same memory controller implementation**



Ramulator 2.0 Case Studies (II)

- **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**



Evaluation of Industry's Recent Solutions

- Oguzhan Canpolat, Abdullah Giray Yaglikci, Geraldo Francisco de Oliveira, Ataberk Olgun, Nisa Bostanci, Ismail Emir Yuksel, Haocong Luo, Oguz Ergin, and Onur Mutlu,
"Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance"

Proceedings of the 31st International Symposium on High-Performance Computer Architecture (HPCA), Las Vegas, NV, USA, March 2025.

[Chronus Source Code (Officially Artifact Evaluated with All Badges)]

Officially artifact evaluated as available, functional, and reproduced.

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*

<https://arxiv.org/pdf/2502.12650>

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

Ramulator 2.1: A Recent Update

- **Appears at ICS 2026 Workshops Proceedings**

Ramulator 2.1: A Composable Memory System Simulator for Modern DRAM Systems

Haocong Luo F. Nisa Bostancı Ataberk Olgun Maria Makeenkova
Ziad Malik Ipek Akdeniz Onur Mutlu

*SAFARI Research Group
ETH Zürich*

<https://arxiv.org/pdf/2606.13844>

More to Come...

Read Disturbance Sessions @ HPCA 2025

HPCA 2025

2025 IEEE International Symposium on High-Performance Computer Architecture,
3/1/2025-3/5/2025, Las Vegas, NV, USA



Session 7A (Acacia A and B): Hammering the Odds – 1

Session Chair: Gururaj Saileshwar (Toronto)

- **Variable Read Disturbance: An Experimental Analysis of Temporal Variation in DRAM Read Disturbance**
Ataberk Olgun (ETH Zürich), Nisa Bostanci (ETH Zürich), Ismail Emir Yuksel (ETH Zürich), Giray Yaglikci (ETH Zürich), Geraldo F. Oliveira (ETH Zürich), Haocong Luo (ETH Zürich), Oguzhan Canpolat (ETH Zürich), Minesh Patel (Rutgers University), Onur Mutlu (ETH Zürich)
- **Understanding RowHammer Under Reduced Refresh Latency: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions**
Yahya Can Tuğrul (TOBB ETÜ & ETH Zürich), Giray Yaglikci (ETH Zürich), Ismail Emir Yuksel (ETH Zürich), Ataberk Olgun (ETH Zürich), Oğuzhan Canpolat (TOBB ETÜ & ETH Zürich), Nisa Bostanci (ETH Zürich), Mohammad Sadrosadati (ETH Zürich), Oguz Ergin (TOBB ETÜ), Onur Mutlu (ETH Zürich)
- **Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance**
Oğuzhan Canpolat (TOBB ETÜ & ETH Zürich), Giray Yaglikci (ETH Zürich), Geraldo Francisco de Oliveira (ETH Zürich), Ataberk Olgun (ETH Zürich), Nisa Bostanci (ETH Zürich), Ismail Emir Yuksel (ETH Zürich), Haocong Luo (ETH Zürich), Oğuz Ergin (TOBB ETÜ), Onur Mutlu (ETH Zürich)

Session 8A (Acacia A and B): Hammering the Odds – 2

Session Chair: Sudhanva Gurumurthi (AMD)

- **AutoRFM: Scaling Low-Cost In-DRAM Trackers to Ultra-Low Rowhammer Thresholds**
Moinuddin Qureshi (Georgia Tech)
- **DAPPER: A Performance-Attack-Resilient Tracker for RowHammer Defense**
Jeonghyun Woo (The University of British Columbia (UBC)), Prashant J. Nair (The University of British Columbia (UBC))
- **QPRAC: Towards Secure and Practical PRAC-based Rowhammer Mitigation using Priority Queues**
Jeonghyun Woo (The University of British Columbia (UBC)), Shaopeng (Chris) Lin (University of Toronto), Prashant J. Nair (The University of British Columbia (UBC)), Aamer Jaleel (NVIDIA), Gururaj Saileshwar (University of Toronto)

Tuesday, March 4th, 11am and 2pm

Read Disturbance Papers @ ASPLOS 2025



Rotterdam, The Netherlands — March 30- April 3, 2025.

Session 4B: Memory & Storage +

LOCATION: VAN OLDENBARNEVELD

Marionette: A RowHammer Attack via Row Coupling

Seungmin Baek (Seoul National University),
Minbok Wi (Seoul National University),
Seonyong Park (Seoul National University),
Hwayong Nam (Seoul National University),
Michael Jaemin Kim (Seoul National University),
Nam Sung Kim (University of Illinois),
Jung Ho Ahn (Seoul National University)

[Paper](#)

MOAT: Securely Mitigating Rowhammer with Per-Row Activation Counters

Moinuddin Qureshi (Georgia Institute of Technology),
Salman Qazi (Google)

[Paper](#)

HyperHammer: Breaking Free from KVM-Enforced Isolation

Wei Chen (Peking University), Zhi Zhang (University of Western Australia), Xin Zhang (Peking University), Qingni Shen (Peking University), Yuval Yarom (Ruhr University Bochum), Daniel Genkin (Georgia Institute of Technology), Chen Yan (Peking University), Zhe Wang (SKLP, Institute of Computing Technology, Chinese Academy of Sciences, Zhongguancun Laboratory)

[Paper](#)

Read Disturbance Session @ ISCA 2025



Session 5A: RowHammer

Location: Okuma Auditorium (Main)

Session Chair: TBA

08:30 AM – 08:50 AM

MoPAC: Efficiently Mitigating Rowhammer with Probabilistic Activation Counting

Suhas Vittal, Salman Qazi, Poulami Das, Moin Qureshi

08:50 AM – 09:10 AM

When Mitigations Backfire: Timing Channel Attacks and Defense for PRAC-Based Rowhammer Mitigations

Jeonghyun Woo, Joyce Qu, Gururaj Saileshwar, Prashant Nair

09:10 AM – 09:30 AM

PuDHammer: Experimental Analysis of Read Disturbance Effects of Processing-using-DRAM in Real DRAM Chips

Ismail Emir Yuksel, Akash Sood, Ataberk Olgun, O?uzhan Canpolat, Haocong Luo, Nisa Bostanci, Mohammad Sadrosadati, Giray Yaglikci, Onur Mutlu

09:30 AM – 09:50 AM

DREAM: Enabling Low-Overhead Rowhammer Mitigation via Directed Refresh Management

Hritvik Taneja, Moin Qureshi

Read Disturbance Papers @ DRAMSec 2025

Accepted papers

Softhammer: Exploiting Rowhammer Bit Flips without Crashing

Finn de Ridder, Patrick Jattke, Kaveh Razavi

Rubber Mallet: A Study of High Frequency Localized Bit Flips and Their Impact on Security

Andrew J. Adiletta, Zane Weissman, Fatemeh Khojasteh Dana, Berk Sunar, Shahin Tajik

CnC-PRAC: Coalesce, not Cache, Per Row Activation Counts for an Efficient in-DRAM Rowhammer Mitigation

Chris S. Lin, Jeonghyun Woo, Prashant J. Nair, Gururaj Saileshwar

A Simulation-based Evaluation Framework for Inter-VM RowHammer Mitigation Techniques

Hidemasa Kawasaki, Soramichi Akiyama

Sudoku: Decomposing DRAM Address Mapping into Component Functions

Minbok Wi, Seungmin Baek, Seonyong Park, Mattan Erez, Jung Ho Ahn

Counterpoint: One-Hot Counting for PRAC-Based RowHammer Mitigation

Shih-Lien Lu, Jeonghyun Woo, Prashant J. Nair

DRFM and the Art of Rowhammer Sampling

Salman Qazi, Moinuddin Qureshi

Keynote

Panel

*Is PRAC a good solution to DRAM read disturbance? Are we missing anything?
Can we (and should we) do much better (and hopefully not worse)?*

Workshop chairs

- Onur Mutlu, ETH Zürich
- Kuljit Bains, NVIDIA

<https://dramsec.ethz.ch/>

<https://www.youtube.com/watch?v=5KmKxFjPopM>

RowHammer @ USENIX Security 2025

Rowhammer-Based Trojan Injection: One Bit Flip Is Sufficient for Backdooring DNNs

Xiang Li, Ying Meng, Junming Chen, Lannan Luo, and Qiang Zeng, *George Mason University*

Long Presentation

AVAILABLE MEDIA  

Posthammer: Pervasive Browser-based Rowhammer Attacks with Postponed Refresh Commands

Finn de Ridder, Patrick Jattke, and Kaveh Razavi, *ETH Zurich*

Short Presentation

AVAILABLE MEDIA  

[Show details](#) ▶

ECC.fail: Mounting Rowhammer Attacks on DDR4 Servers with ECC Memory

Nureddin Kamadan and Walter Wang, *Georgia Tech*; Stephan van Schaik, *University of Michigan*; Christina Garman, *Purdue University*; Daniel Genkin, *Georgia Tech*; Yuval Yarom, *Ruhr University Bochum*

Short Presentation

AVAILABLE MEDIA 

Achilles: A Formal Framework of Leaking Secrets from Signature Schemes via Rowhammer

Junkai Liang, *Peking University*; Zhi Zhang, *The University of Western Australia*; Xin Zhang and Qingni Shen, *Peking University*; Yansong Gao, *The University of Western Australia*; Xingliang Yuan, *The University of Melbourne*; Haiyang Xue and Pengfei Wu, *Singapore Management University*; Zhonghai Wu, *Peking University*

Long Presentation

McSee: Evaluating Advanced Rowhammer Attacks and Defenses via Automated DRAM Traffic Analysis

Patrick Jattke and Michele Marazzi, *ETH Zurich*; Flavien Solt, *UC Berkeley*; Max Wipfli, Stefan Gloor, and Kaveh Razavi, *ETH Zurich*

Long Presentation

AVAILABLE MEDIA  

[Show details](#) ▶

Not so Refreshing: Attacking GPUs using RFM

Rowhammer Mitigation

Ravan Nazaraliyev and Yicheng Zhang, *University of California, Riverside*; Sankha Baran Dutta, *Brookhaven National Laboratory*; Andres Marquez and Kevin Barker, *Pacific Northwest National Laboratory*; Nael Abu-Ghazaleh, *University of California, Riverside*

Short Presentation

AVAILABLE MEDIA   

GPUHammer: Rowhammer Attacks on GPU Memories are Practical

Chris S. Lin, Joyce Qu, and Gururaj Saileshwar, *University of Toronto*

Short Presentation

AVAILABLE MEDIA  

Read Disturbance Papers @ MICRO 2025

Citadel: Rethinking Memory Allocation to Safeguard Against Inter-Domain Rowhammer Exploits

Anish Saxena, Walter Wang, Alexandros Daglis (Georgia Inst. of Technology)

ColumnDisturb: Understanding Column-based Read Disturbance in Real DRAM Chips and Implications for Future Systems

Ismail Emir Yuksel, Ataberk Olgun, Nisa Bostanci, Haocong Luo, Abdullah Giray Yaglikci, Onur Mutlu (ETH Zürich)

Understanding and Mitigating Covert and Side Channel Vulnerabilities Introduced by RowHammer Defenses

Nisa Bostanci (ETH Zürich); Oguzhan Canpolat (TOBB ETÜ and ETH Zurich); Ataberk Olgun, Ismail Emir Yuksel, Konstantinos Kanellopoulos, Mohammad Sadrosadati, Abdullah Giray Yaglikci, Onur Mutlu (ETH Zürich)

pHammer: Reviving RowHammer Attacks on New Architectures via Prefetching

Weijie Chen, Shan Tang, Yulin Tang (Huazhong Univ. of Science and Technology); Xiapu Luo (The Hong Kong Polytechnic Univ.); Yinqian Zhang (Southern Univ. of Science and Technology); Weizhong Qiang (Huazhong Univ. of Science and Technology)

Read Disturbance Papers @ HPCA 2026

11:30 - 12:50 DRAM Security and Reliability at [Collaroy](#)

- | | | | |
|-------|-------------|---|---|
| 11:30 | 20m | ☆ | MIRZA: Efficiently Mitigating Rowhammer with Randomization and ALERT |
| | <i>Talk</i> | | Hritvik Taneja Georgia Tech, Ali Hajiabadi ETH Zurich, Michele Marazzi ABB Research, Kaveh Razavi ETH Zürich, Moinuddin K. Qureshi Georgia Tech |
| 11:50 | 20m | ☆ | SALT: Track-and-Mitigate Subarrays, Not Rows, for Blast-Radius-Free Rowhammer Defense |
| | <i>Talk</i> | | Moinuddin K. Qureshi Georgia Tech |
| 12:10 | 20m | ☆ | ReScue: Reliable and Secure CXL Memory |
| | <i>Talk</i> | | Chihun Song UIUC, Austin Antony Cruz UIUC, Michael Jaemin Kim Meta, Minbok Wi Seoul National University, Gaohan Ye UIUC, Kyungsan Kim Samsung Sung Kim UIUC |

Read Disturbance Papers @ ASPLOS 2026

Session 2D: DRAM Reliability & Security

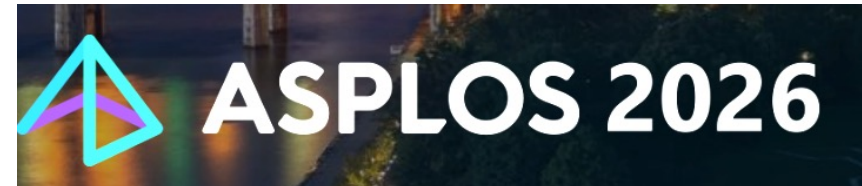
+

RowArmor: Efficient and Comprehensive Protection Against DRAM Disturbance Errors

Minbok Wi (Seoul National University), Yoonyul Yoo (Samsung Electronics), Yoojin Kim (Sungkyunkwan University), Jaeho Shin (Sungkyunkwan University), Jumin Kim (Seoul National University), Yesin Ryu (Sungkyunkwan University, Samsung Electronics), Saeid Gorgin (Sungkyunkwan University), Jung Ho Ahn (Seoul National University), Jungrae Kim (Sungkyunkwan University)

APT: Securing Against DRAM Read Disturbance via Adaptive Probabilistic In-DRAM Trackers

Runjin Wu (School of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan, China), Meng Zhang (School of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan, China), You Zhou (School of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan, China), Changsheng Xie (Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, China), Fei Wu (Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, China)



Read Disturbance Papers @ S&P 2026

MAY 18-21, 2026 AT THE HILTON SAN FRANCISCO UNION SQUARE, SAN FRANCISCO, CA

47th IEEE Symposium on Security and Privacy

Sponsored by the IEEE Computer Society's Technical Community on Security and Privacy in cooperation with the International Association for Cryptologic Research

Phoenix: Rowhammer Attacks on DDR5 with Self-Correcting Synchronization ▼

Diego Meyer¹, Patrick Jattke¹, Michele Marazzi¹, Salman Qazi², Daniel Moghimi², Kaveh Razavi¹

¹: ETH Zurich, ²: Google

SoK: Systematizing a Decade of Architectural RowHammer Defenses Through the Lens of Streaming Algorithms ▼

Michael Jaemin Kim¹, Seungmin Baek², Jumin Kim², Hwayong Nam², Nam Sung Kim³, Jung Ho Ahn²

¹: Meta, ²: Seoul National University, ³: UIUC

GPUBreach: Privilege Escalation Attacks on GPUs using Rowhammer ▼

Chris S. Lin, Yuqin Yan, Joyce Qu, Joseph Zhu, Guozhen Ding, David Lie, Gururaj Saileshwar
University of Toronto

GDDR: Greatly Disturbing DRAM Rows — Cross-Component Rowhammer Attacks from Modern GPUs ▼

Yichang Hu¹, Noah Brown¹, Yuhang Chen¹, Joshua Bakita², Tianlong Chen¹, Daniel Genkin³, Andrew Kwong¹

¹: University of North Carolina at Chapel Hill, ²: Mohamed bin Zayed University of Artificial Intelligence, ³: Georgia Institute

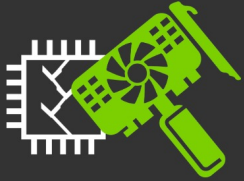
GeForge: Hammering GDDR Memory to Forge GPU Page Tables for Fun and Profit ▼

Junpeng Wan^{1,2}, Yanan Guo³, Zhi Zhang⁴, Zhuo Li⁵, Dave (Jing) Tian¹, Zhenkai Zhang²

¹: Purdue University, ²: Clemson University, ³: University of Rochester, ⁴: University of Western Australia, ⁵: HydroX AI

RowHammer on GPUs

GDDRHammer:

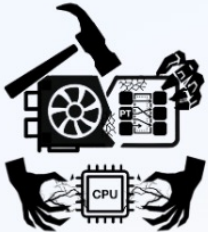


Greatly Disturbing DRAM
Rows -- Cross-
Component
Rowhammer Attacks
from Modern GPUs



GeForge:

Hammering GDDR
Memory to Forge GPU
Page Tables for Fun and
Profit



GPUBreach: Privilege Escalation Attacks on GPUs using Rowhammer



GPUHammer: Rowhammer Attacks on GPU Memories are Practical

Read Disturbance Papers @ DRAMSec 2026

Workshop papers

Hardware Prefetcher-based Rowhammer Attack

Yashwant Kumar Balivada, Maccoy Merrell, Paul Gratz, Stavros Kalafatis

Paper

Temperature-Aware Runtime-Adaptive Thresholding for Rowhammer Mitigation

Aziz Alajmi, Hoeseok Yang

Paper

AI Agents Can Defeat Security by Obscurity for Rowhammer Defenses

Stefan Saroiu, Jay Bosamiya, Alec Wolman, Adam Grenzebach, Paramvir Bahl

Paper

Blindspot: Moving the Needle on Rowhammer-Tolerant DRAM Chips by Understanding Technical and Business Concerns

Moinuddin K. Qureshi

Paper

FlipShield : Self-Improving Software Mitigation Against Rowhammer

Fan Wang, Yuan Xiao, Carter Yagemann

Paper

CNN-Based RowHammer Vulnerability Prediction: Inter-DIMM Generalization and Intra-DIMM Completion from Partial Row Observations

Mohammad Farmani, Vineet Suresh Kumar

Paper

elppiR: Efficiently Protecting PRAC Against Ripple Attacks

Moinuddin K. Qureshi

Paper

AutoPRAC: Automating Attack Discovery for PRAC-Based Rowhammer Defenses using Model Checkers

Joyce Qu, Gururaj Saileshwar

Paper

Keynote

Fantastic Bitflips and Where to Find Them (and How to Tame Them)

Onur Mutlu, ETH Zurich [PPTX] [PDF]

Read Disturbance Papers @ ISCA 2026

Session 6D: Rowhammer and Security —

Location: Room 301

Session Chair: *Tamara Lehman*
(University of Colorado Boulder)

14:15 – 14:35

ColumnKeeper: Efficient Solutions to the ColumnDisturb Vulnerability in DRAM-based Systems

Andreas Kosmas Kakolyris (ETH Zurich), F. Nisa Bostanci (ETH Zurich), Ataberk Olgun (ETH Zurich), Ismail Emir Yuksel (ETH Zurich), Harsh Songara (ETH Zurich), Konstantinos Marios Sgouras (ETH Zurich), Umut Baser (TOBB ETU, ETH Zurich), Konstantinos Kanellopoulos (ETH Zurich), A. Giray Yaglikci (CISPA), Onur Mutlu (ETH Zurich)

14:35 – 14:55

PVAC: A RowHammer Mitigation Architecture Exploiting Per-victim-row Counting

Jumin Kim (Seoul National University), Seungmin Baek (Seoul National University), Hwayong Nam (Seoul National University), Minbok Wi (Samsung Electronics), Nam Sung Kim (University of Illinois, Urbana-Champaign), Jung Ho Ahn (Seoul National University)

14:55 – 15:15

Loaded Dice: Solving the Non-Selection Problem for Scalable Probabilistic RowHammer Defense

Jeonghyun Woo (The University of British Columbia), Junsu Kim (The University of British Columbia), Aamer Jaleel (NVIDIA), Prashant Nair (The University of British Columbia)

15:15 – 15:35

PRowhammer: Propagating Bit-flips from CPU to GPU

Mrityunjay Shukla (Indian Institute of Technology Bombay), Shubham Roy (Indian Institute of Technology Bombay), Sayandeep Saha (Indian Institute of Technology Bombay), Biswabandan Panda (Indian Institute of Technology Bombay)

15:35 – 15:55

DejaVu: Why You Should Write to Your DRAM Rows Twice, Carefully

Haocong Luo (ETH Zurich), Ismail Emir Yuksel (ETH Zurich), Ataberk Olgun (ETH Zurich), Nisa Bostanci (ETH Zurich), Orhun Ecemiş (TOBB ETÜ), Abdullah Giray Yaglikci (CISPA), Onur Mutlu (ETH Zurich)

Session 4B: Industry Track 1 —

Location: Room 302

Session Chair: *Shanmathi Natarajan* (Meta)

17:30 – 17:50

From Lab to Fleet: Building and Deploying a Practical Rowhammer Defense in Cloud SoCs

Stefan Saroiu (Microsoft), Sujay Yadalam (UT Austin), Alec Wolman (Microsoft), Will Remaklus (Microsoft), Daniel Berger (Microsoft), Isaac Hernandez Luna (Microsoft), Ishwar Agarwal (Meta), Jacob R. Lorch (Microsoft)

Tutorial on DRAM Bender & Ramulator

Tutorial on Ramulator & DRAM Bender: Cutting-Edge Infrastructures for Real and Future Memory System Evaluation

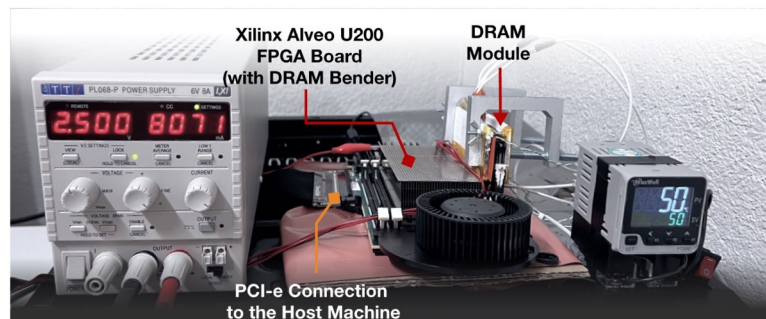


R&DB:

**A half-day tutorial @ ASPLOS 2026, Pittsburgh, USA
22nd March 2026 (Saturday)**

Organizers: Nisa Bostanci, Ataberk Olgun, Ismail Yuksel, Haocong Luo, Prof. Onur Mutlu

- *Evaluating memory performance, emerging memory technologies, and architectural mechanisms*
- *Real DRAM device characterization studies*
- *New features, extensions, and enhancements*
- *Cross-layer research that combines memory system simulation with real-chip DRAM characterization*



**More information &
call for presentation:**

<https://events.safari.ethz.ch/asplos26-ramulator-drambender/>



ISCA 2026: DRAM Bender & Ramulator

2nd Tutorial on Ramulator & DRAM Bender: Cutting-Edge Infrastructures for Real and Future Memory System Evaluation

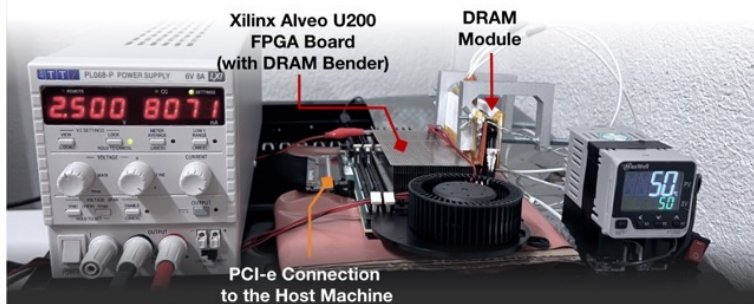


R&DB:

**A full-day tutorial @ ISCA 2026, Raleigh, USA
27 June 2026 (Saturday)**

Organizers: Nisa Bostanci, Ataberk Olgun, Ismail Yuksel, Haocong Luo, Prof. Onur Mutlu

- *Evaluating memory performance, emerging memory technologies, and architectural mechanisms*
- *Real DRAM device characterization studies*
- *New features, extensions, and enhancements*
- *Cross-layer research that combines memory system simulation with real-chip DRAM characterization*



**More information &
call for presentation:**

<https://events.safari.ethz.ch/isca26-ramulator-drambender/>



ICS 2026: DRAM Bender & Ramulator

3rd Tutorial on Ramulator & DRAM Bender: Cutting-Edge Infrastructures for Real and Future Memory System Evaluation

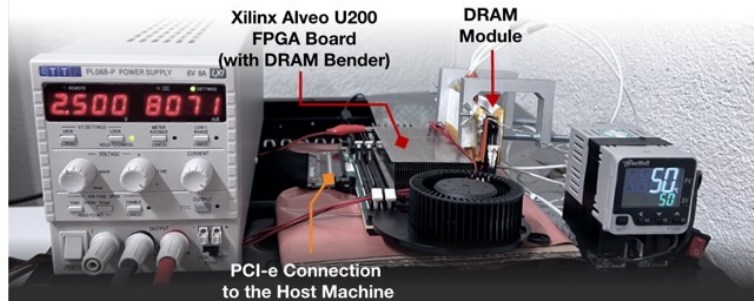


R&DB:

A half-day tutorial @ ICS 2026, Belfast, Northern Ireland, United Kingdom, 6th July 2026 (Monday)

Organizers: Nisa Bostanci, Ataberk Olgun, Ismail Yuksel, Haocong Luo, Prof. Onur Mutlu

- *Evaluating memory performance, emerging memory technologies, and architectural mechanisms*
- *Real DRAM device characterization studies*
- *New features, extensions, and enhancements*
- *Cross-layer research that combines memory system simulation with real-chip DRAM characterization*

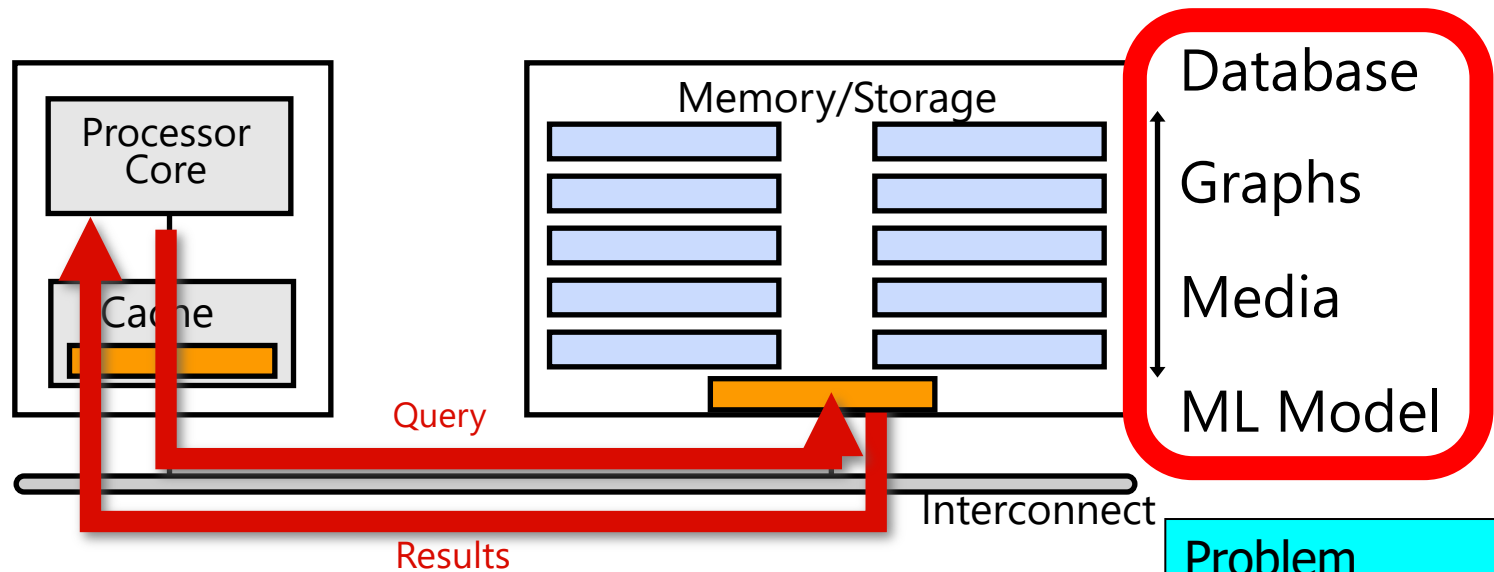


***More information &
call for presentation:***

<https://events.safari.ethz.ch/ics26-ramulator-drambender/>



Goal: Processing Inside Memory/Storage



- Many questions ... How do we design the:
 - ❑ compute-capable memory & controllers?
 - ❑ processors & communication units?
 - ❑ software & hardware interfaces?
 - ❑ system software, compilers, languages?
 - ❑ algorithms & theoretical foundations?

Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Logic
Devices
Electrons

Ramulator for Processing in Memory

Simulation Infrastructure for PIM

- **Ramulator** extended for PIM
 - Flexible and extensible DRAM simulator
 - Can model many different memory standards and proposals
 - Kim+, “**Ramulator: A Flexible and Extensible DRAM Simulator**”, IEEE CAL 2015.
 - <https://github.com/CMU-SAFARI/ramulator-pim>
 - <https://github.com/CMU-SAFARI/ramulator>
 - [\[Source Code for Ramulator-PIM\]](#)

Ramulator: A Fast and Extensible DRAM Simulator

Yoongu Kim¹ Weikun Yang^{1,2} Onur Mutlu¹
¹Carnegie Mellon University ²Peking University

Ramulator for PIM

- Gagandeep Singh, Juan Gomez-Luna, Giovanni Mariani, Geraldo F. Oliveira, Stefano Corda, Sander Stujik, Onur Mutlu, and Henk Corporaal, **"NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning"**
Proceedings of the 56th Design Automation Conference (DAC), Las Vegas, NV, USA, June 2019.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Poster \(pptx\)](#)] [[pdf](#)]
[[Source Code for Ramulator-PIM](#)]

NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning

Gagandeep Singh ^{a,c}	Juan Gómez-Luna ^b	Giovanni Mariani ^c	Geraldo F. Oliveira ^b
Stefano Corda ^{a,c}	Sander Stuijk ^a	Onur Mutlu ^b	Henk Corporaal ^a
^a Eindhoven University of Technology		^b ETH Zürich	^c IBM Research - Zurich

Ramulator Project Course

Exploration of Emerging Memory Systems (Spring/Fall 2022)

Fall 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator

Spring 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator

Youtube Livestream (Spring 2022):

- ❑ https://www.youtube.com/watch?v=aM-lIXRQd3s&list=PL5Q2soXY2Zi_TlmlGw_Z8hBo2925ZApqV

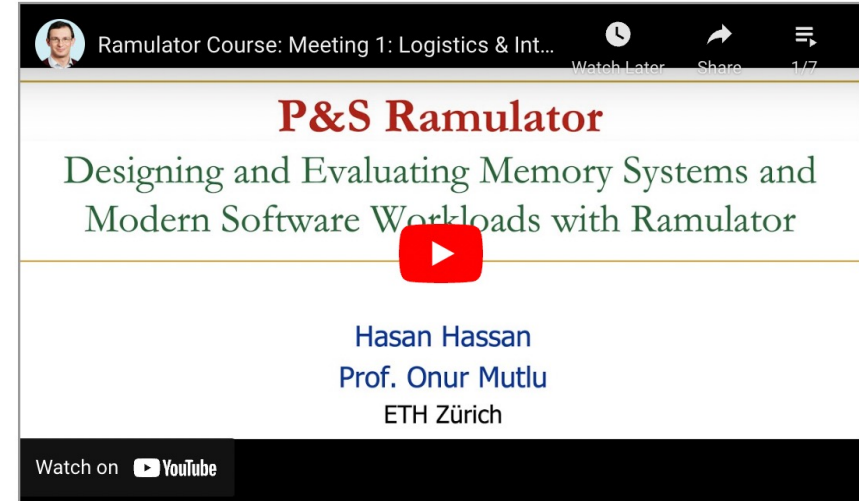
Bachelor's course

- ❑ Elective at ETH Zurich
- ❑ Introduction to memory system simulation
- ❑ Tutorial on using Ramulator
- ❑ C++
- ❑ Potential research exploration

<https://www.youtube.com/onurmutlulectures>

Lecture Video Playlist on YouTube

Lecture Playlist

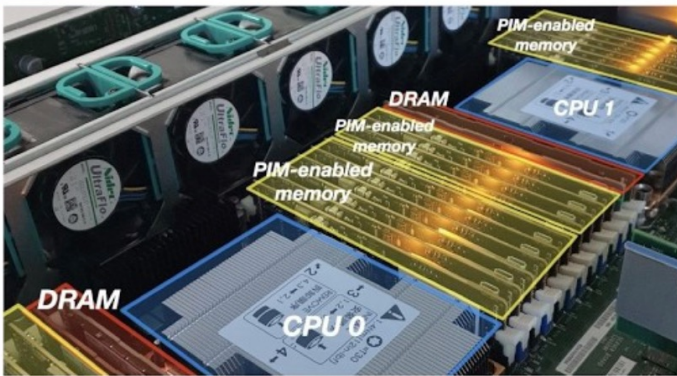


2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	09.03 Wed.	YouTube Video	M1: Logistics & Intro to Simulating Memory Systems Using Ramulator PDF (PDF) PPT (PPT)		HW0
W2	16.03 Fri.	YouTube Video	M2: Tutorial on Using Ramulator PDF (PDF) PPT (PPT)		
W3	25.02 Fri.	YouTube Video	M3: BlockHammer PDF (PDF) PPT (PPT)		
W4	01.04 Fri.	YouTube Video	M4: CLR-DRAM PDF (PDF) PPT (PPT)		
W5	08.04 Fri.	YouTube Video	M5: SIMDRAM PDF (PDF) PPT (PPT)		
W6	29.04 Fri.	YouTube Video	M6: DAMOV PDF (PDF) PPT (PPT)		
W7	06.05 Fri.	YouTube Video	M7: Synchron PDF (PDF) PPT (PPT)		

Memory-Centric Computing Systems Workshop

5th Workshop on Memory-Centric Computing Systems



MCCSys

**A half-day workshop @ ASPLOS 2026, Pittsburgh, USA
23rd March 2026 (Sunday)**

Organizers: Ismail E. Yuksel, F. Nisa Bostanci, Ataberk Olgun, Dr. Zhiheng Yue, Dr. Mohammad Sadrosadati, Dr. Geraldo F. Oliveira, Prof. Onur Mutlu

**More information &
call for presentation:**



Live in 25 hours
March 23 at 1:00 PM



Notification on

- Overview of PIM & PIM Taxonomy
- PIM in Memory & Storage
- Infrastructures for PIM Research
- Real-World PnM Systems
- PuM for Bulk Bitwise Operations
- Programming Techniques & Tools
- Research Challenges & Opportunities

<https://events.safari.ethz.ch/asplos26-MCCSys/>

MCCSys-5: 5th Workshop on Memory-Centric Computing Systems, held with ASPLOS 2026 - 23 March 2026



Onur Mutlu Lectures
60.1K subscribers

Analytics

Edit video



4



Share



Save



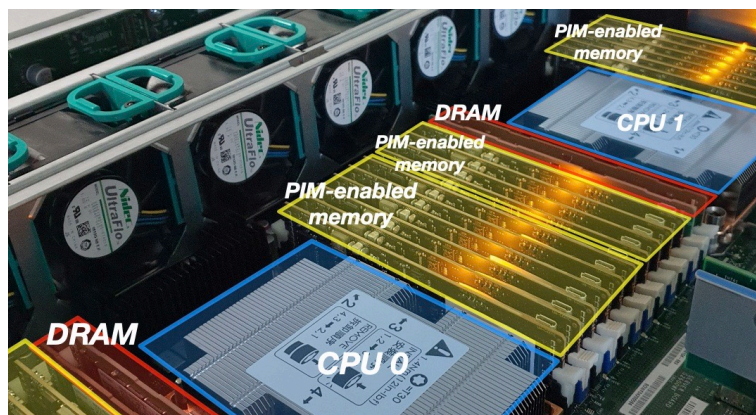
<https://events.safari.ethz.ch/asplos26-MCCSys/>

ISCA 2026



June 27– July 1, 2026

6th Workshop on Memory-Centric Computing Systems



*More information &
call for papers:*



MCCSys

***A full-day workshop @ ISCA 2026
28th June 2026 (Sunday)***

Organizers: Ismail Emir Yuksel, F. Nisa Bostanci,
Ataberk Olgun, Prof. Onur Mutlu

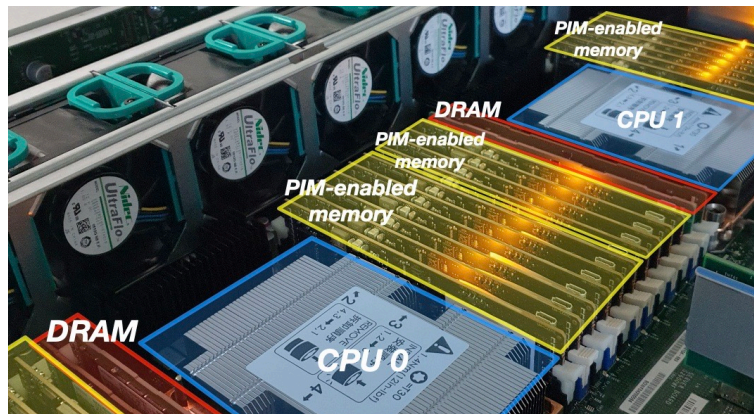
- Overview of PIM & PIM Taxonomy
- PIM in Memory & Storage
- Infrastructures for PIM Research
- Real-World PnM Systems
- PuM for Bulk Bitwise Operations
- Programming Techniques & Tools
- Research Challenges & Opportunities

ACM International Conference on Supercomputing 2026

6-9 July 2026

Belfast, Northern Ireland, United Kingdom

7th Workshop on Memory-Centric Computing Systems



*More information &
call for papers:*



MCCSys

*A full-day workshop @ ICS 2026
6th July 2026 (Monday)*

Organizers: Ismail Emir Yuksel, F. Nisa Bostanci,
Ataberk Olgun, Prof. Onur Mutlu

- Overview of PIM & PIM Taxonomy
- PIM in Memory & Storage
- Infrastructures for PIM Research
- Real-World PnM Systems
- PuM for Bulk Bitwise Operations
- Programming Techniques & Tools
- Research Challenges & Opportunities

Some Other Useful Simulators

Virtuoso Tutorial & Workshop

VIRTUOSO

✦ 2nd Edition

Infrastructure for Evaluating Novel HW/OS Interfaces

A full-day workshop and tutorial on the principles, methodologies, and practical aspects of hardware/OS co-design for memory management

 Date: Monday, March 23, 2026 (Full Day)

Live in 20 hours
March 23 at 7:45 AM



Notification on

Room Ft. Duquesne, ASPLOS 2026, Pittsburgh, PA

ASPLOS 2026: 2nd Workshop on Virtuoso: Ideas and Infrastructures for Novel HW/OS Interfaces



Onur Mutlu Lectures
60.1K subscribers

Analytics

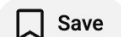
Edit video



3



Share



Save



Many Simulators for Many Purposes

- [Ramulator 2.0](#) for memory simulation
- [gem5](#) full system multi-core simulation
- [MQSim](#) for SSD simulation
- [DiskSim](#) for Hard Disk simulation
- [DAMOV-Sim](#) for Processing-near-Memory simulation
- [Virtuoso](#) for Virtual Memory (HW/SW codesign) simulation
- [Sniper](#) for fast Processor Simulation
- [Scarab](#) for detailed Microarchitectural Simulation
- [Simics](#), [Bochs](#), [QEMU](#) for full-system functional simulation
- ...
- Or, develop your own simulator for your purpose...

DAMOV Simulator, Methods & Benchmarks

- Geraldo F. Oliveira, Juan Gomez-Luna, Lois Orosa, Saugata Ghose, Nandita Vijaykumar, Ivan fernandez, Mohammad Sadrosadati, and Onur Mutlu,
"DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks"
IEEE Access, 8 September 2021.
Preprint in arXiv, 8 May 2021.
[[arXiv preprint](#)]
[[IEEE Access version](#)]
[[DAMOV Suite and Simulator Source Code](#)]
[[SAFARI Live Seminar Video](#) (2 hrs 40 mins)]
[[Short Talk Video](#) (21 minutes)]

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

MOHAMMAD SADROSADATI, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

DAMOV is Open Source

- We open-source our **benchmark suite** and our **toolchain**

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About



DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing. Described by Oliveira et al. (preliminary version at <https://arxiv.org/pdf/2105.03725.pdf>)

Readme

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages



omutlu Update README.md

ce1b4ea 17 days ago 5 commits

simulator	Cleaning	19 days ago
README.md	Update README.md	17 days ago
get_workloads.sh	DAMOV -- first commit	19 days ago

README.md

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including [BWA](#), [Chai](#), [Darknet](#), [GASE](#), [Hardware Effects](#), [Hashjoin](#), [HPCC](#), [HPCG](#), [Ligra](#), [PARSEC](#), [Parboil](#), [PolyBench](#), [Phoenix](#), [Rodinia](#), [SPLASH-2](#), [STREAM](#).

DAMOV-SIM

DAMOV
Benchmarks

SAFARI

DAMOV is Open Source

- We open-source our [benchmark suite](#) and our [toolchain](#)

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About

DAMOV is a benchmark suite and a

Get DAMOV at:

<https://github.com/CMU-SAFARI/DAMOV>

README.md

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including [BWA](#), [Chai](#), [Darknet](#), [GASE](#), [Hardware Effects](#), [Hashjoin](#), [HPCC](#), [HPCG](#), [Ligra](#), [PARSEC](#), [Parboil](#), [PolyBench](#), [Phoenix](#), [Rodinia](#), [SPLASH-2](#), [STREAM](#).

Readme

Releases

No releases published
[Create a new release](#)

Packages

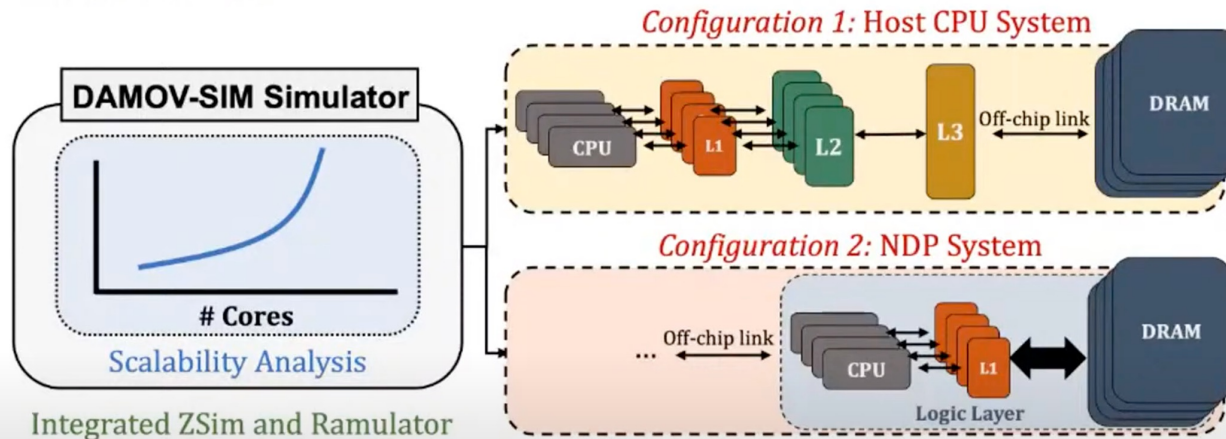
No packages published
[Publish your first package](#)

Languages

More on DAMOV Analysis Methodology & Workloads

Step 3: Memory Bottleneck Classification (2/2)

- **Goal:** identify the specific sources of data movement bottlenecks



- **Scalability Analysis:**
 - 1, 4, 16, 64, and 256 out-of-order/in-order host and NDP CPU cores
 - 3D-stacked memory as main memory

SAFARI DAMOV-SIM: <https://github.com/CMU-SAFARI/DAMOV> 30

SAFARI Live Seminar: DAMOV: A New Methodology & Benchmark Suite for Data Movement Bottlenecks

352 views • Streamed live on Jul 22, 2021

18 0 SHARE SAVE ...



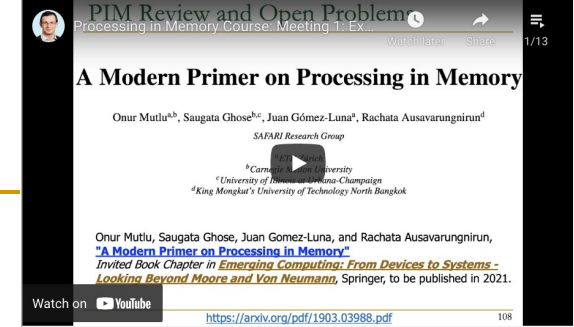
Onur Mutlu Lectures
17.7K subscribers

ANALYTICS

EDIT VIDEO

https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=3

PIM Course (Spring 2022)



Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	10.03 Thu.	YouTube Live	M1: P&S PIM Course Presentation (PDF) (PPT)	Required Materials Recommended Materials	HW 0 Out
W2	15.03 Tue. 17.03 Thu.	YouTube Premiere	M2: Real-world PIM: UPMEM PIM (PDF) (PPT)		
W3	24.03 Thu.	YouTube Live	M3: Real-world PIM: Microbenchmarking of UPMEM PIM (PDF) (PPT)		
W4	31.03 Thu.	YouTube Live	M4: Real-world PIM: Samsung HBM-PIM (PDF) (PPT)		
W5	07.04 Thu.	YouTube Live	M5: How to Evaluate Data Movement Bottlenecks (PDF) (PPT)		
W6	14.04 Thu.	YouTube Live	M6: Real-world PIM: SK Hynix AIM (PDF) (PPT)		
W7	21.04 Thu.	YouTube Premiere	M7: Programming PIM Architectures (PDF) (PPT)		
W8	28.04 Thu.	YouTube Premiere	M8: Benchmarking and Workload Suitability on PIM (PDF) (PPT)		
W9	05.05 Thu.	YouTube Premiere	M9: Real-world PIM: Samsung AxoDIMM (PDF) (PPT)		
W10	12.05 Thu.	YouTube Premiere	M10: Real-world PIM: Alibaba HB-PNM (PDF) (PPT)		
W11	19.05 Thu.	YouTube Live	M11: SpMV on a Real PIM Architecture (PDF) (PPT)		
W12	26.05 Thu.	YouTube Live	M12: End-to-End Framework for Processing-using-Memory (PDF) (PPT)		
W13	02.06 Thu.	YouTube Live	M13: Bit-Serial SIMD Processing using DRAM (PDF) (PPT)		
W14	09.06 Thu.	YouTube Live	M14: Analyzing and Mitigating ML Inference Bottlenecks (PDF) (PPT)		
W15	15.06 Thu.	YouTube Live	M15: In-Memory HTAP Databases with HW/SW Co-design (PDF) (PPT)		
W16	23.06 Thu.	YouTube Live	M16: In-Storage Processing for Genome Analysis (PDF) (PPT)		
W17	18.07 Mon.	YouTube Premiere	M17: How to Enable the Adoption of PIM? (PDF) (PPT)		
W18	09.08 Tue.	YouTube Premiere	SS1: ISVLSI 2022 Special Session on PIM (PDF & PPT)		

Spring 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory

Youtube Livestream:

- <https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX>

Project course

- Taken by Bachelor's/Master's students
- Processing-in-Memory lectures
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>

MQSim for Modern SSD Simulation

- Arash Tavakkol, Juan Gomez-Luna, Mohammad Sadrosadati, Saugata Ghose, and Onur Mutlu,
"MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices"
Proceedings of the 16th USENIX Conference on File and Storage Technologies (FAST), Oakland, CA, USA, February 2018.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Source Code](#)]

MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices

Arash Tavakkol[†], Juan Gómez-Luna[†], Mohammad Sadrosadati[†], Saugata Ghose[‡], Onur Mutlu^{†‡}
[†]*ETH Zürich* [‡]*Carnegie Mellon University*

Solid-State Drives Course (Spring 2022)

■ Spring 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=modern_sds

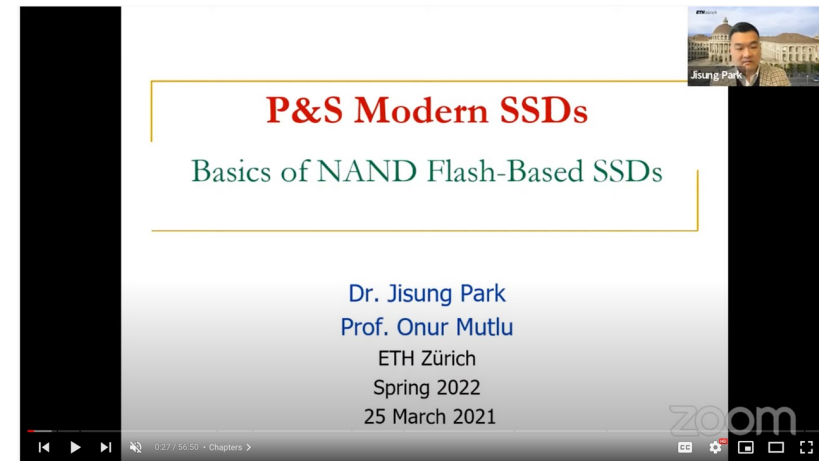
■ Youtube Livestream:

- ❑ <https://www.youtube.com/watch?v=q4rm71DsY4&list=PL5Q2soXY2Zi8vabcse1kL22DEcgMI2RAq>

■ Project course

- ❑ Taken by Bachelor's/Master's students
- ❑ SSD Basics and Advanced Topics
- ❑ Hands-on research exploration
- ❑ Many research readings

<https://www.youtube.com/onurmutlulectures>



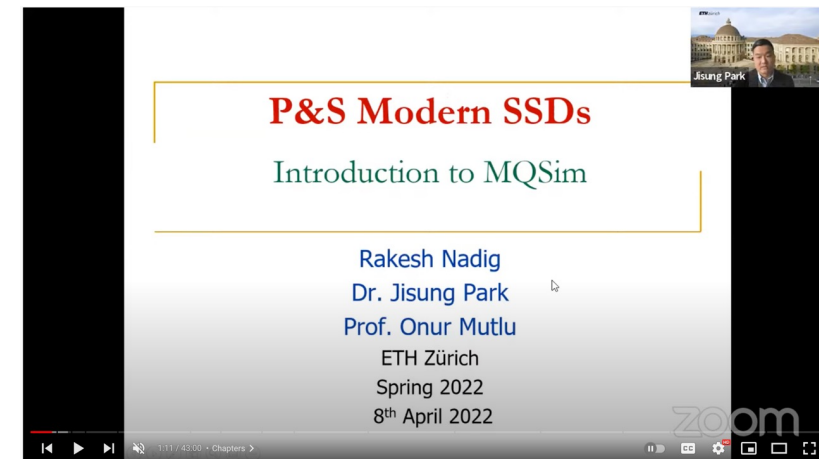
Modern Solid-State Drives (SSDs) Course - Meeting 2: Basics of NAND Flash-Based SSDs (Spring 2022)

807 views • Streamed live on Mar 25, 2022

16 DISLIKE SHARE DOWNLOAD CLIP SAVE

Onur Mutlu Lectures
25K subscribers

ANALYTICS EDIT VIDEO



Modern Solid-State Drives (SSDs) Course - Meeting 4: Introduction to MQSim (Spring 2022)

310 views • Streamed live on Apr 8, 2022

17 DISLIKE SHARE DOWNLOAD CLIP SAVE

Onur Mutlu Lectures
25K subscribers

ANALYTICS EDIT VIDEO

Many More Simulators ...



SAFARI Research Group at ETH Zurich and Carnegie Mellon University

Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.

👤 440 followers

📍 ETH Zurich and Carnegie Mellon U...

🔗 <https://safari.ethz.ch/>

✉ omutlu@gmail.com

🏠 Overview

📁 Repositories 80

📁 Projects

📁 Packages

👤 People 13

📁 ramulator Public

A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...

● C++ ☆ 591 🍴 209

📁 prim-benchmarks Public

PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publ...

● C ☆ 140 🍴 49

📁 MQSim Public

MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

● C++ ☆ 280 🍴 148

📁 rowhammer Public

Source code for testing the Row Hammer error mechanism in DRAM devices. Described in the ISCA 2014 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf.

● C ☆ 221 🍴 42

📁 SoftMC Public

SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

● Verilog ☆ 128 🍴 27

📁 Pythia Public

A customizable hardware prefetching framework using online reinforcement learning as described in the MICRO 2021 paper by Bera et al. (<https://arxiv.org/pdf/2109.12021.pdf>).

● C++ ☆ 118 🍴 37

<https://github.com/CMU-SAFARI/>

Virtuoso

- Konstantinos Kanellopoulos, Konstantinos Sgouras, Nisa Bostanci, Andreas Kosmas Kakolyris, Berkin Kerim Konar, Rahul Bera, Rakesh Kumar, Mohammad Sadrosadati, Nandita Vijaykumar, and Onur Mutlu,
"Virtuoso: Enabling Fast and Accurate Virtual Memory Research via an Imitation-based Operating System Simulation Methodology,"
Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Rotterdam, Netherlands, April 2025.
[[Slides \(pptx\) \(pdf\)](#)]
[[Virtuoso Source Code](#)]

Virtuoso: Enabling Fast and Accurate Virtual Memory Research via an Imitation-based Operating System Simulation Methodology

Konstantinos Kanellopoulos¹ Konstantinos Sgouras¹ F. Nisa Bostanci¹
Andreas Kosmas Kakolyris¹ Berkin Kerim Konar¹ Rahul Bera¹
Mohammad Sadrosadati¹ Rakesh Kumar² Nandita Vijaykumar³ Onur Mutlu¹

¹ETH Zürich ²Norwegian University of Science and Technology ³University of Toronto

<https://arxiv.org/pdf/2403.04635>

Virtuoso Tutorial & Workshop

VIRTUOSO

✦ 2nd Edition

Infrastructure for Evaluating Novel HW/OS Interfaces

A full-day workshop and tutorial on the principles, methodologies, and practical aspects of hardware/OS co-design for memory management

 Date: Monday, March 23, 2026 (Full Day)

Live in 20 hours
March 23 at 7:45 AM



Notification on

Room Ft. Duquesne, ASPLOS 2026, Pittsburgh, PA

ASPLOS 2026: 2nd Workshop on Virtuoso: Ideas and Infrastructures for Novel HW/OS Interfaces



Onur Mutlu Lectures
60.1K subscribers

Analytics

Edit video



3

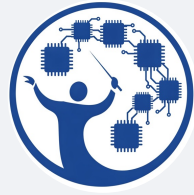


Share



Save





Virtuoso

Enabling Fast and Accurate Virtual Memory Research with an Imitation-based Operating System Simulation Methodology

Konstantinos Kanellopoulos Konstantinos Sgouras F. Nisa Bostanci

Andreas Kosmas Kakolyris Berkin Kerim Konar Rahul Bera

Mohammad Sadrosadati Rakesh Kumar Nandita Vijaykumar Onur Mutlu

<https://github.com/CMU-SAFARI/Virtuoso>

SAFARI
SAFARI Research Group
safari.ethz.ch

ETH zürich

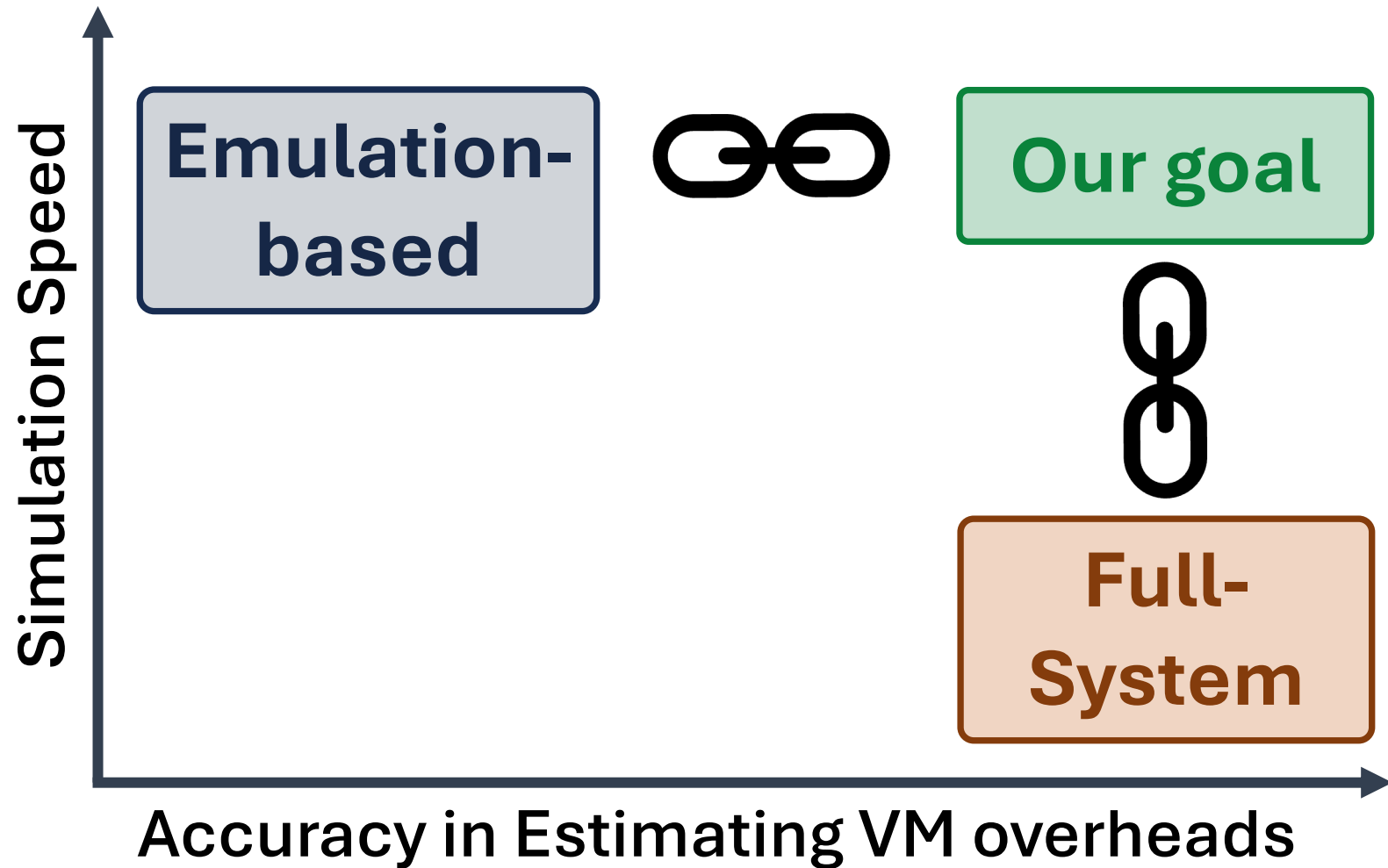


UNIVERSITY OF
TORONTO

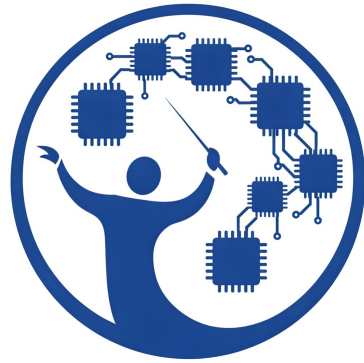


NTNU

Existing Architectural Simulators



Enabling Fast and Accurate VM Research



Virtuoso

New simulation framework that enables **fast and accurate** prototyping and evaluation of the ***software and hardware*** components of VM

VirTool: State-of-the-art VM techniques



Toolset encompassing the HW/SW components of multiple state-of-the-art VM techniques

4 page table designs

Software-Managed TLB

Page-Size Prediction

Nested MMU Support

2 Hash-based address mapping schemes

2 Memory Tagging Schemes

2 THP Policies

2 Contiguity-aware Schemes

2 Intermediate Address Space Schemes

Speculative Translation

TLB Prefetching

Integrated Virtuoso with 5 Diverse Architectural Simulators

gem5-SE

Ramulator

Sniper

ChampSim

**Focus on
storage devices**

[Tavakkol+ FAST 2018]

MQSim

<https://github.com/CMU-SAFARI/MQSim>

More Details in the Virtuoso Paper

Virtuoso: Enabling Fast and Accurate Virtual Memory Research via an Imitation-based Operating System Simulation Methodology

Konstantinos Kanellopoulos¹ Konstantinos Sgouras¹ F. Nisa Bostanci¹
Andreas Kosmas Kakolyris¹ Berkin Kerim Konar¹ Rahul Bera¹
Mohammad Sadrosadati¹ Rakesh Kumar² Nandita Vijaykumar³ Onur Mutlu¹

¹ETH Zürich ²Norwegian University of Science and Technology ³University of Toronto

Abstract

The unprecedented growth in data demand from emerging applications has turned virtual memory (VM) into a major performance bottleneck. VM's overheads are expected to persist as memory requirements continue to increase. Researchers explore new hardware/OS co-designs to optimize VM across diverse applications and systems. To evaluate such designs, researchers rely on various simulation methodologies to model VM components. Unfortunately, current simulation tools (i) either lack the desired accuracy in modeling VM's software components or (ii) are too slow and complex to prototype and evaluate schemes that span across the hardware/software boundary.

We introduce Virtuoso, a new simulation framework that enables quick and accurate prototyping and evaluation of the *software and hardware components* of the VM subsystem. The key idea of Virtuoso is to employ a *lightweight userspace OS kernel*, called MimicOS, that (i) accelerates simulation time by imitating *only* the desired kernel functionalities, (ii) facilitates the development of new OS routines that imitate

simulation time overhead of only 20%, on top of four baseline architectural simulators. The source code of Virtuoso is freely available at <https://github.com/CMU-SAFARI/Virtuoso>.

1 Introduction

Virtual memory (VM) [1–23] is a cornerstone of modern computing systems, enabling application-transparent physical memory management, isolation and data sharing. Contemporary applications (e.g., [24–45]) exhibit different characteristics that stress the VM subsystem. We classify these workloads into two broad categories: (i) long-running workloads (i.e., execution time larger than 100s of seconds) [24, 28–31, 33–35] with large data footprints and irregular memory access patterns, that exhibit high address translation overheads, and (ii) short-running workloads (i.e., execution time often lower than 1 second) [36–45] whose execution time does not amortize the overheads of system software operations (e.g., physical memory allocation). Multiple prior works and industrial studies [46–57] have shown that address trans-

<https://arxiv.org/pdf/2403.04635>

Tools for Evaluating Memory and Memory-Centric Computing **DRAM, Storage, Virtual Memory**

Onur Mutlu

omutlu@gmail.com

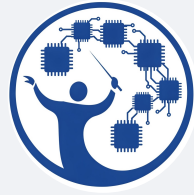
<https://people.inf.ethz.ch/omutlu>

6 July 2026

3rd Ramulator & DRAM Bender Tutorial @ ICS 2026

SAFARI

ETH zürich



Virtuoso

Enabling Fast and Accurate Virtual Memory Research with an Imitation-based Operating System Simulation Methodology

Konstantinos Kanellopoulos

Konstantinos Sgouras

F. Nisa Bostanci

Andreas Kosmas Kakolyris

Berkin Kerim Konar

Rahul Bera

Mohammad Sadrosadati

Rakesh Kumar

Nandita Vijaykumar

Onur Mutlu

<https://github.com/CMU-SAFARI/Virtuoso>

SAFARI
SAFARI Research Group
safari.ethz.ch

ETH zürich

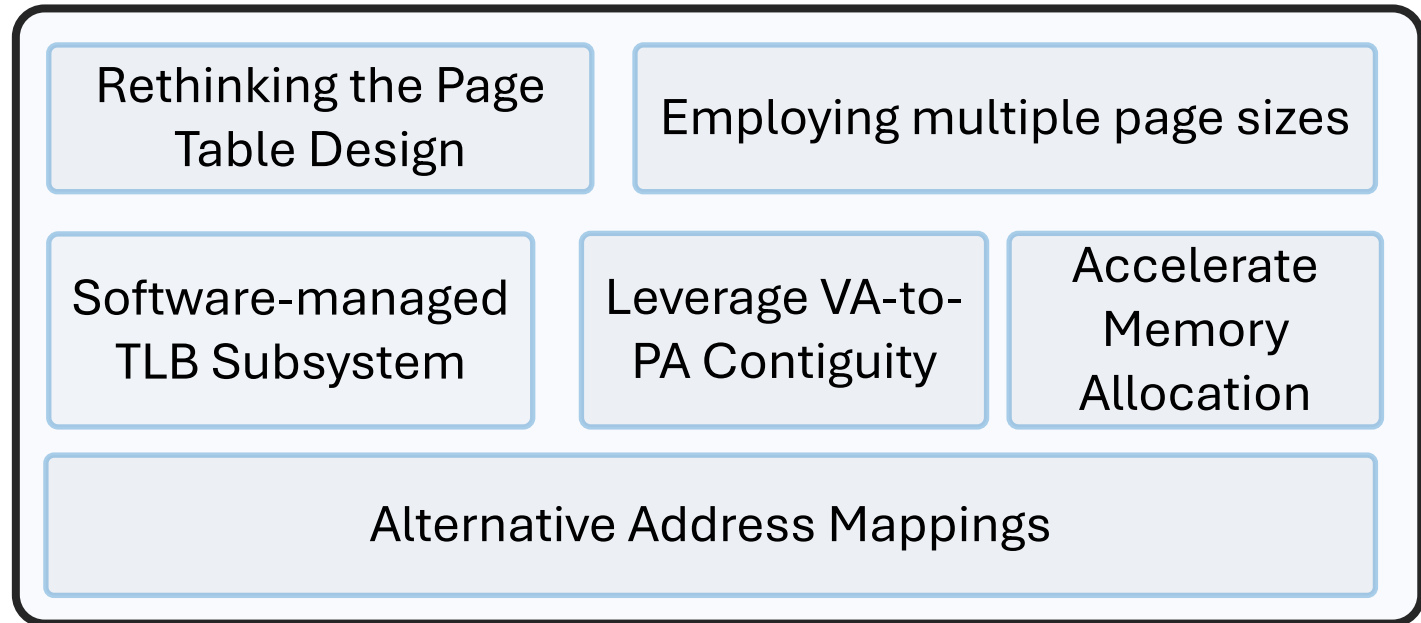


UNIVERSITY OF
TORONTO



NTNU

Co-design the hardware and the OS to reduce VM overheads

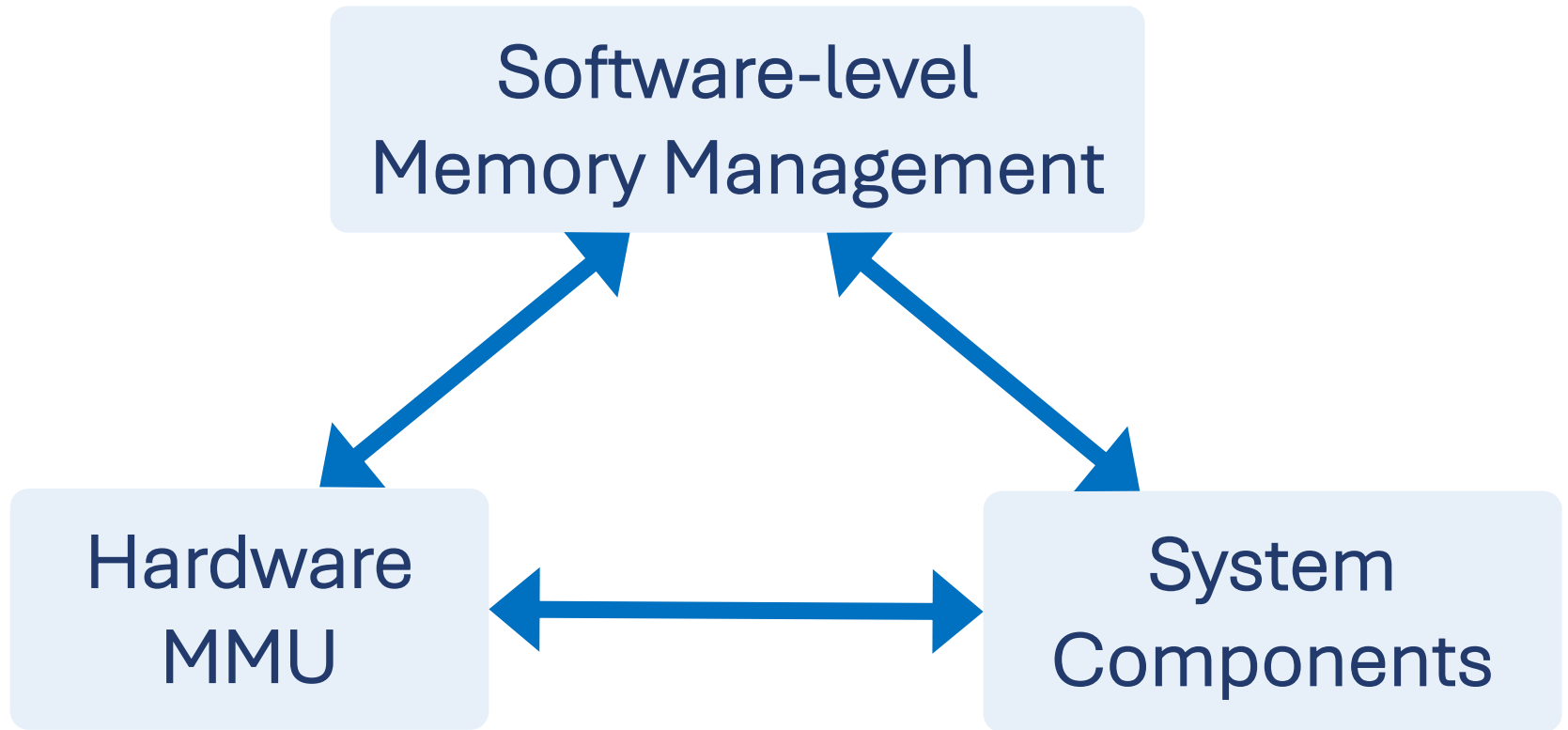


Wide range of hardware-OS co-design techniques that aim to improve virtual memory

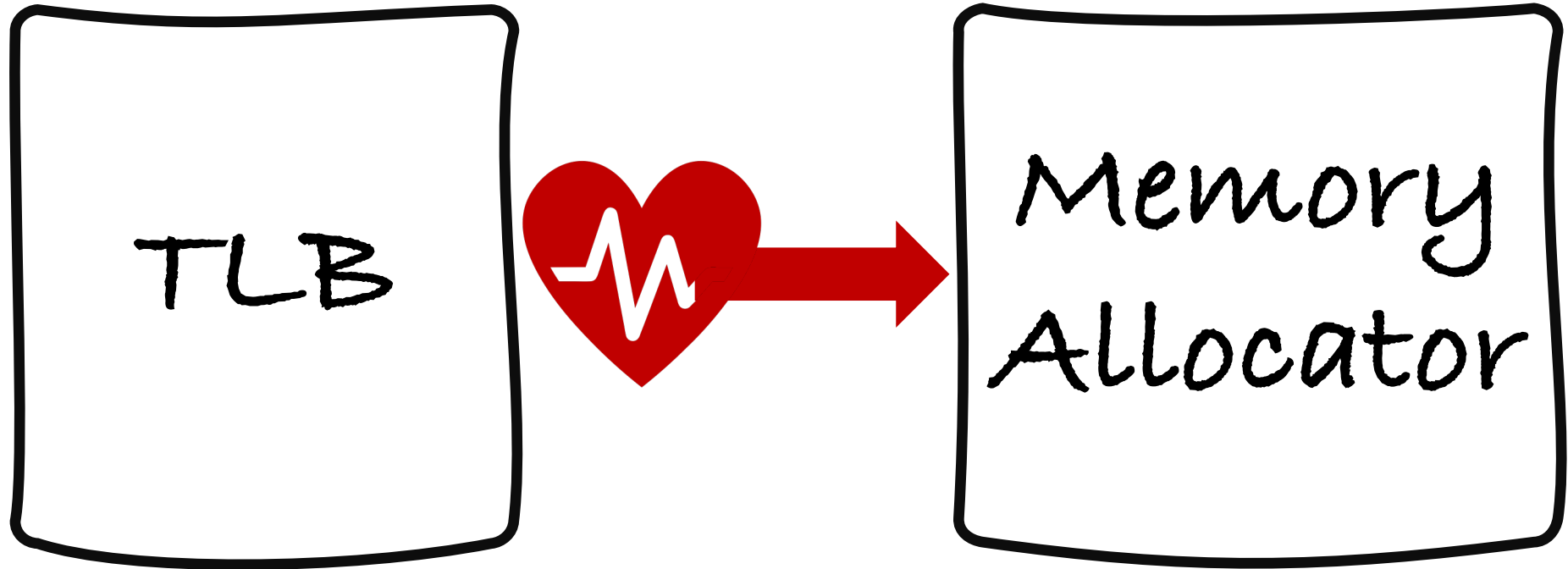
**Effectively evaluating VM techniques
is crucial for progress in the domain**

Challenges in Evaluation

Interplay between system components



Example: TLB and Memory Allocator



TLB performance **heavily depends** on
the memory allocator (e.g., # of 2MB pages)

Challenges in Evaluation

Software-level

Evaluating the interplay between components is critical when assessing current and new VM techniques



Existing Architectural Simulators

Two main classes of simulators

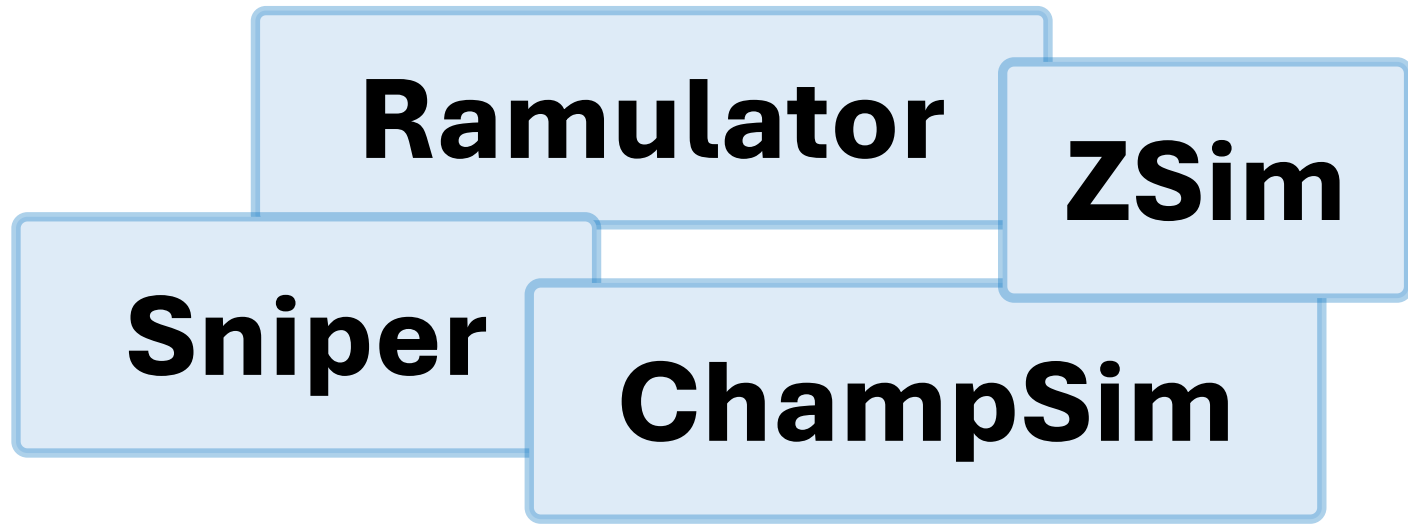
1

Emulation-based

2

Full-system

Emulation-based Simulators



High simulation speed with a focus on modeling microarchitectural features

Limited support for OS primitives

Analytically estimate VM Overheads

Full-System Simulators

gem5-FS

QFlex

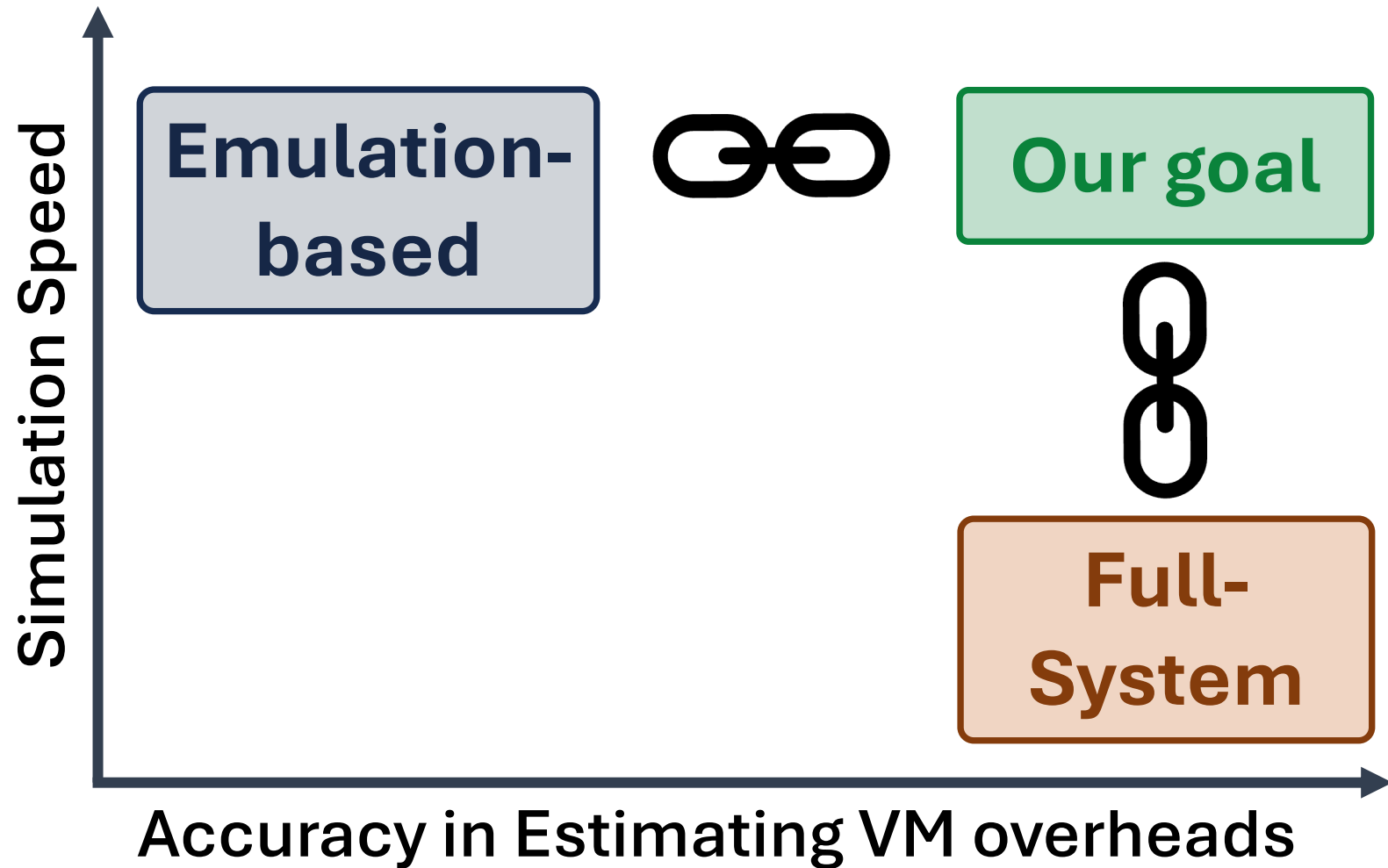
PTLsim

Enable the execution of a **full-blown OS**
on top of a hardware simulator

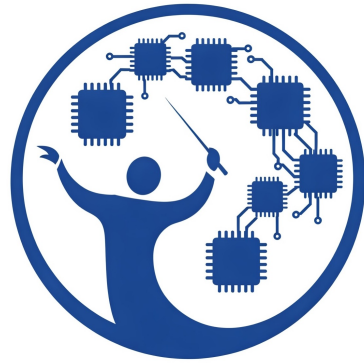
**Low simulation
speed**

Hard to develop

Existing Architectural Simulators



Enabling Fast and Accurate VM Research



Virtuoso

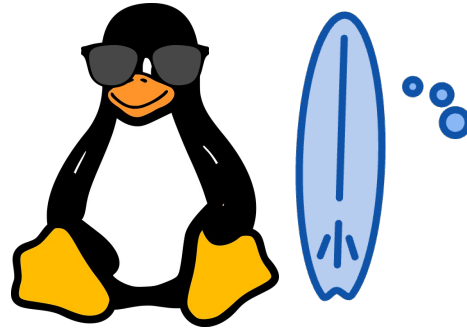
New simulation framework that enables **fast and accurate** prototyping and evaluation of the ***software and hardware*** components of VM

Imitation-based Simulation Methodology

Lightweight Userspace Kernel

written in a high-level programming language

*“I will try to imitate
the full-blown OS”*



**Choose OS modules only related
to the desired research**

Imitation-based Simulation Methodology

Monitoring

Full-blown Kernel

Interrupts

Thread Scheduler

Managing NUMA

Data Sharing
Support

Transparent
Huge Pages

Swapping

Virtual Memory
Area Handling

Cgroups

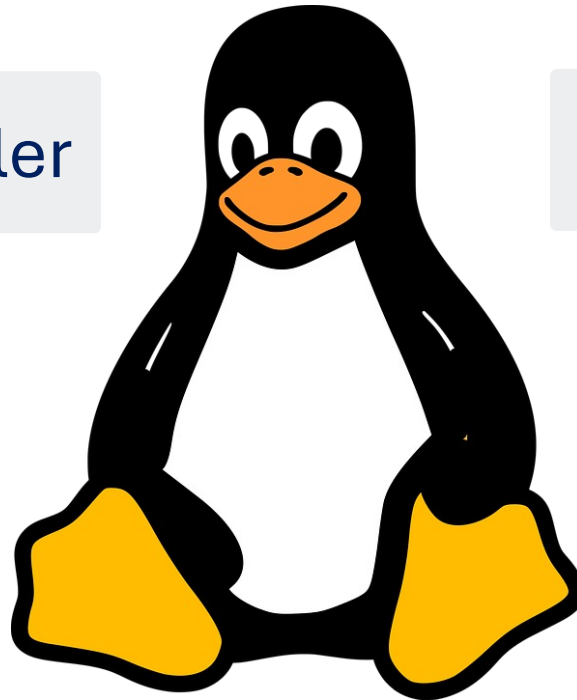
Boot process

Protection

Block device
driver

SLAB Allocator

hugetlbfs



Imitation-based Simulation Methodology

1

Rapid development and prototyping

2

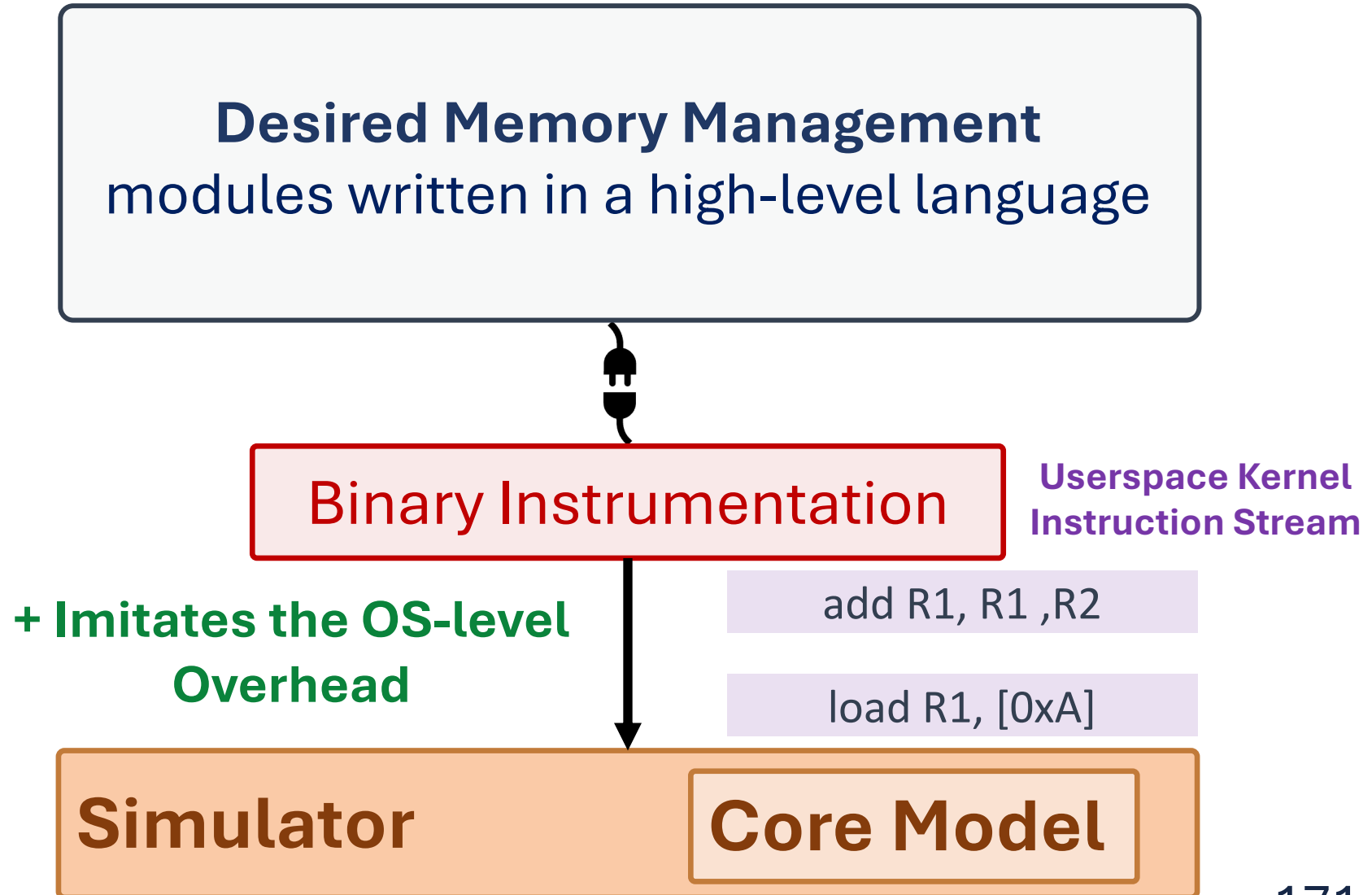
High simulation speed by executing only the desired OS kernel functionality

3

Accurate evaluation by integrating Virtuoso with an architectural simulator

Imitation-based Simulation Methodology

Lightweight Userspace Kernel



Imitation-based Simulation Methodology

Lightweight Userspace Kernel

Desired Memory Management
modules written in a high-level language



Binary Instrumentation

Userspace Kernel
Instruction Stream

+ Enables emulation
of kernel modules

add R1, R1, R2

load R1, [0xA]

**Functional
Outcome**

Simulator

Core Model

MimicOS: Imitating the Linux Kernel

MimicOS modules written in C++

Buddy
Allocator

Radix-based
page table

hugetlbfs

Transparent
Huge Pages

Swap Space

Page Cache

More details in the paper

<https://arxiv.org/pdf/2403.04635>

VirTool: State-of-the-art VM techniques



Toolset encompassing the HW/SW components of multiple state-of-the-art VM techniques

4 page table designs

Software-Managed TLB

Page-Size Prediction

Nested MMU Support

2 Hash-based address mapping schemes

2 Memory Tagging Schemes

2 THP Policies

2 Contiguity-aware Schemes

2 Intermediate Address Space Schemes

Speculative Translation

TLB Prefetching

VirTool: State-of-the-art VM techniques



Toolset encompassing multiple
state-of-the-art VM techniques

4 page table

Software-

Page-Size Prediction

**Enables researchers to evaluate
and prototype current and
brand new VM techniques**

2 THP
Policies

2 Intermediate Address Space Schemes

VirTool: State-of-the-art VM techniques



Toolset encompassing multiple
state-of-the-art VM techniques

4 page table

Software-

Page-Size Prediction

**Virtuoso is a highly versatile
simulation framework**

2 IHP
Policies

2 Intermediate Address Space Schemes

Integrated Virtuoso with 5 Diverse Architectural Simulators

gem5-SE

System-call emulation
mode of gem5

[Lowe-Power+ arXiv 2020]

[Binkert+ SIGARCH News 2011]

<http://gem5.org/>

Integrated Virtuoso with 5 Diverse Architectural Simulators

gem5-SE

Ramulator

**Focus on main
memory subsystem**

[Luo+ CAL 2023]

<https://github.com/CMU-SAFARI/ramulator2>

Integrated Virtuoso with 5 Diverse Architectural Simulators

gem5-SE

Ramulator

Sniper

**Focus on
multicore systems**

[Carlson+ TACO 2012]

<https://github.com/snipersim/snipersim>

Integrated Virtuoso with 5 Diverse Architectural Simulators

gem5-SE

Ramulator

Sniper

ChampSim

**Focus on
microarchitecture**

[Gober+ arXiv 2022]

<https://github.com/ChampSim/ChampSim>

Integrated Virtuoso with 5 Diverse Architectural Simulators

gem5-SE

Ramulator

Sniper

ChampSim

**Focus on
storage devices**

[Tavakkol+ FAST 2018]

MQSim

<https://github.com/CMU-SAFARI/MQSim>

Virtuoso's Versatility

We showcase Virtuoso's versatility by evaluating multiple different use cases

Demonstrate new insights on state-of-the-art VM techniques

5 use cases in the paper

<https://arxiv.org/pdf/2403.04635>

Virtuoso Example Use Cases

Evaluating Different
Page Table Designs

**Evaluating Physical Memory
Allocation Policies**

Evaluation Methodology

Buddy: Baseline allocator with the buddy system

Utopia:¹ Part of memory is organized using a hash-based mapping

Workloads: **Short-input Short-output LLM inference**



Mistral-7B



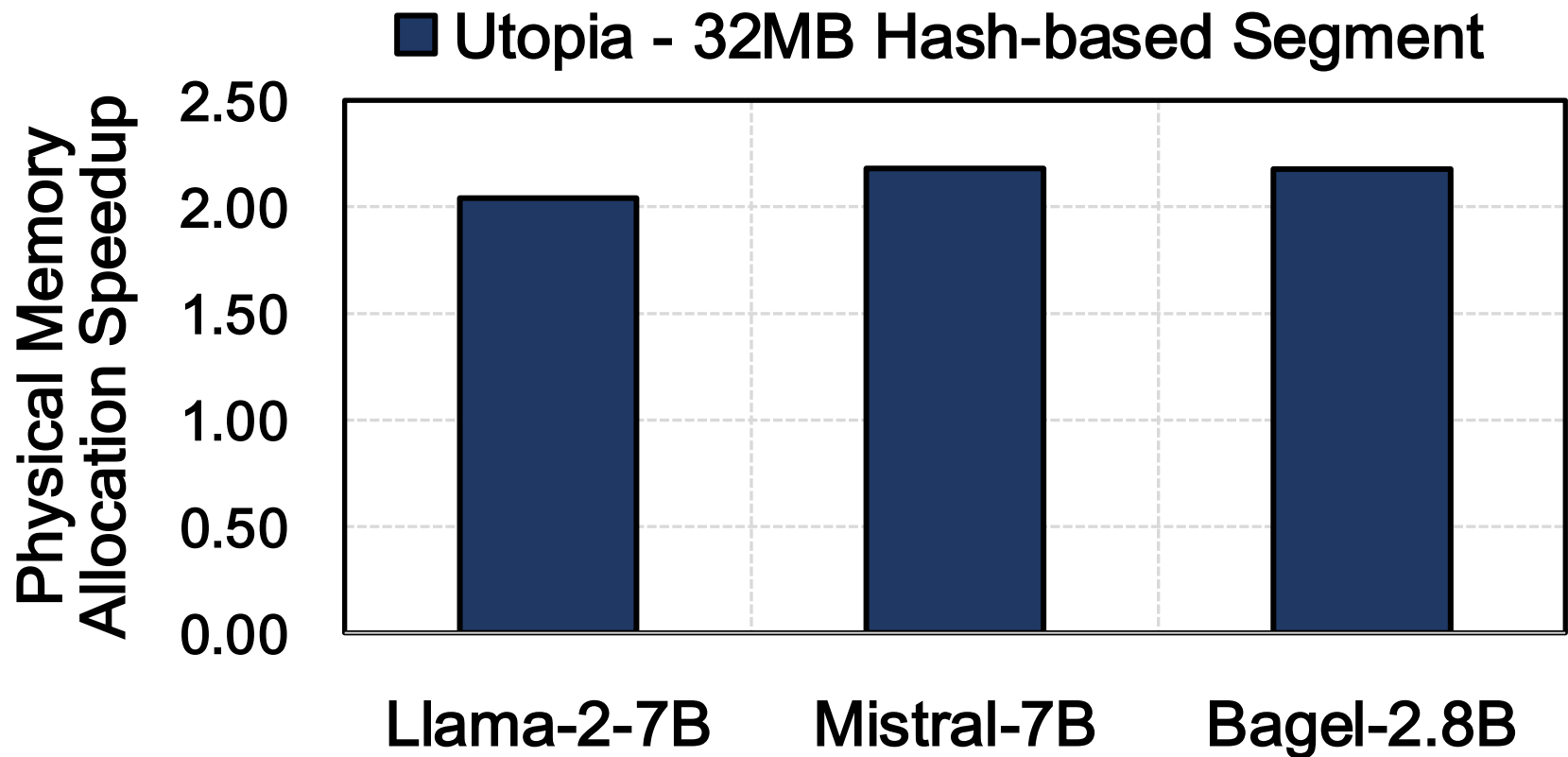
Llama-2-7B

BAGEL

Bagel-2.8B

[1] Kanellopoulos et al. “Utopia: Fast and Efficient Address Translation via Hybrid Restrictive & Flexible Virtual-to-Physical Address Mappings” MICRO 2023

Accelerating Memory Allocation



Restricting the virtual-to-physical mapping speeds up memory allocation by up to 2.17x

More Details in the Paper

- Detailed description of communication primitives
- Detailed description of MimicOS modules
- Integration methodology with different simulators
- Integration with heterogeneous system simulation
- Evaluation of intermediate address space schemes
- Evaluation of swapping activity and translation latency in hash-based address mapping schemes
- Evaluation of two additional hash-based page tables

and more to come ...

More Details in the Paper

Virtuoso: Enabling Fast and Accurate Virtual Memory Research via an Imitation-based Operating System Simulation Methodology

Konstantinos Kanellopoulos¹ Konstantinos Sgouras¹ F. Nisa Bostanci¹
Andreas Kosmas Kakolyris¹ Berkin Kerim Konar¹ Rahul Bera¹
Mohammad Sadrosadati¹ Rakesh Kumar² Nandita Vijaykumar³ Onur Mutlu¹

¹ETH Zürich ²Norwegian University of Science and Technology ³University of Toronto

Abstract

The unprecedented growth in data demand from emerging applications has turned virtual memory (VM) into a major performance bottleneck. VM's overheads are expected to persist as memory requirements continue to increase. Researchers explore new hardware/OS co-designs to optimize VM across diverse applications and systems. To evaluate such designs, researchers rely on various simulation methodologies to model VM components. Unfortunately, current simulation tools (i) either lack the desired accuracy in modeling VM's software components or (ii) are too slow and complex to prototype and evaluate schemes that span across the hardware/software boundary.

We introduce Virtuoso, a new simulation framework that enables quick and accurate prototyping and evaluation of the *software and hardware components* of the VM subsystem. The key idea of Virtuoso is to employ a *lightweight userspace OS kernel*, called MimicOS, that (i) accelerates simulation time by imitating *only* the desired kernel functionalities, (ii) facilitates the development of new OS routines that imitate

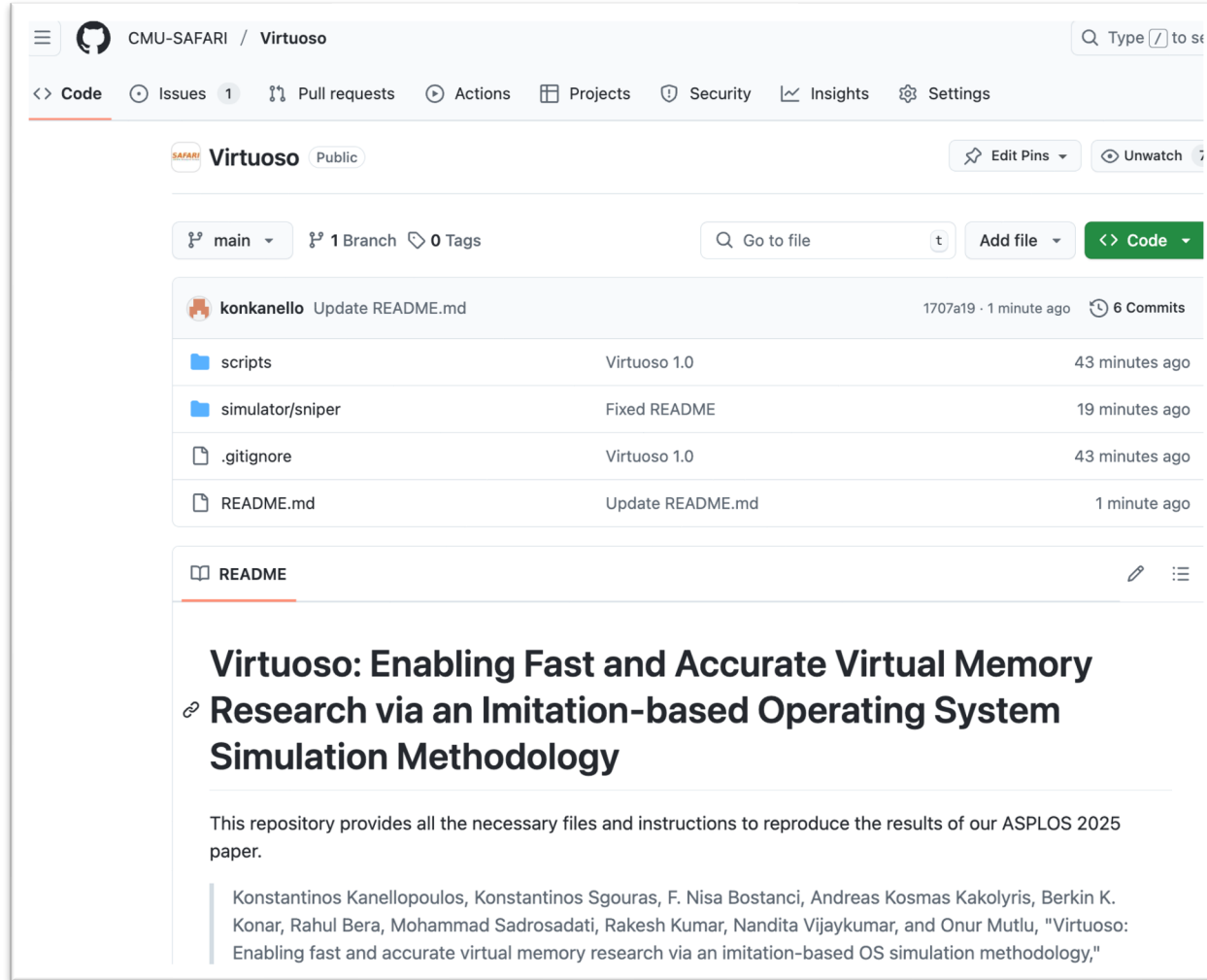
simulation time overhead of only 20%, on top of four baseline architectural simulators. The source code of Virtuoso is freely available at <https://github.com/CMU-SAFARI/Virtuoso>.

1 Introduction

Virtual memory (VM) [1–23] is a cornerstone of modern computing systems, enabling application-transparent physical memory management, isolation and data sharing. Contemporary applications (e.g., [24–45]) exhibit different characteristics that stress the VM subsystem. We classify these workloads into two broad categories: (i) long-running workloads (i.e., execution time larger than 100s of seconds) [24, 28–31, 33–35] with large data footprints and irregular memory access patterns, that exhibit high address translation overheads, and (ii) short-running workloads (i.e., execution time often lower than 1 second) [36–45] whose execution time does not amortize the overheads of system software operations (e.g., physical memory allocation). Multiple prior works and industrial studies [46–57] have shown that address trans-

<https://arxiv.org/pdf/2403.04635>

Virtuoso is Open-Source



CMU-SAFARI / Virtuoso

Code Issues 1 Pull requests Actions Projects Security Insights Settings

Virtuoso Public

main 1 Branch 0 Tags

konkanello Update README.md 1707a19 · 1 minute ago 6 Commits

scripts	Virtuoso 1.0	43 minutes ago
simulator/sniper	Fixed README	19 minutes ago
.gitignore	Virtuoso 1.0	43 minutes ago
README.md	Update README.md	1 minute ago

README

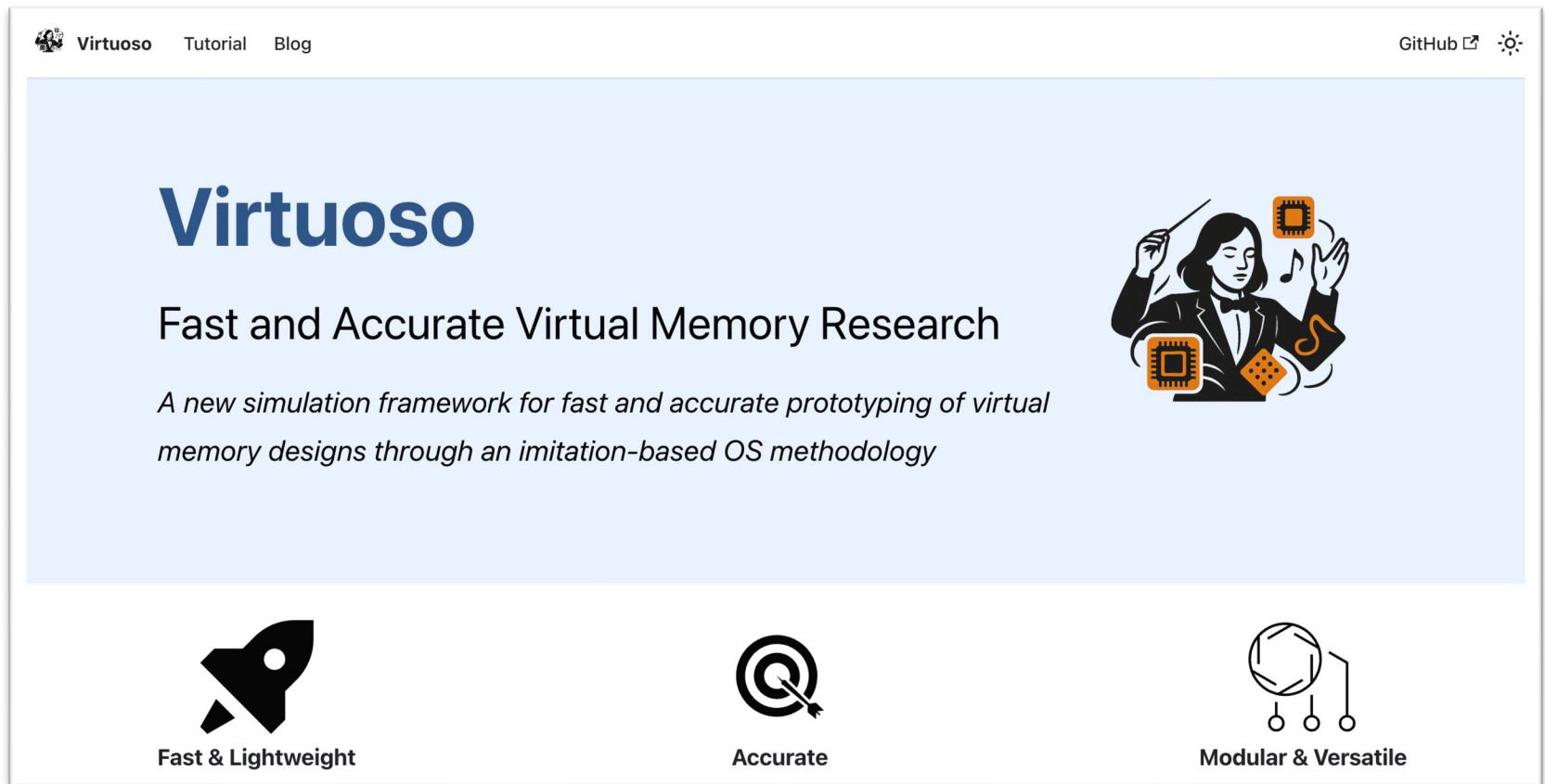
Virtuoso: Enabling Fast and Accurate Virtual Memory Research via an Imitation-based Operating System Simulation Methodology

This repository provides all the necessary files and instructions to reproduce the results of our ASPLOS 2025 paper.

Konstantinos Kanellopoulos, Konstantinos Sgouras, F. Nisa Bostanci, Andreas Kosmas Kakolyris, Berkin K. Konar, Rahul Bera, Mohammad Sadrosadati, Rakesh Kumar, Nandita Vijaykumar, and Onur Mutlu, "Virtuoso: Enabling fast and accurate virtual memory research via an imitation-based OS simulation methodology,"

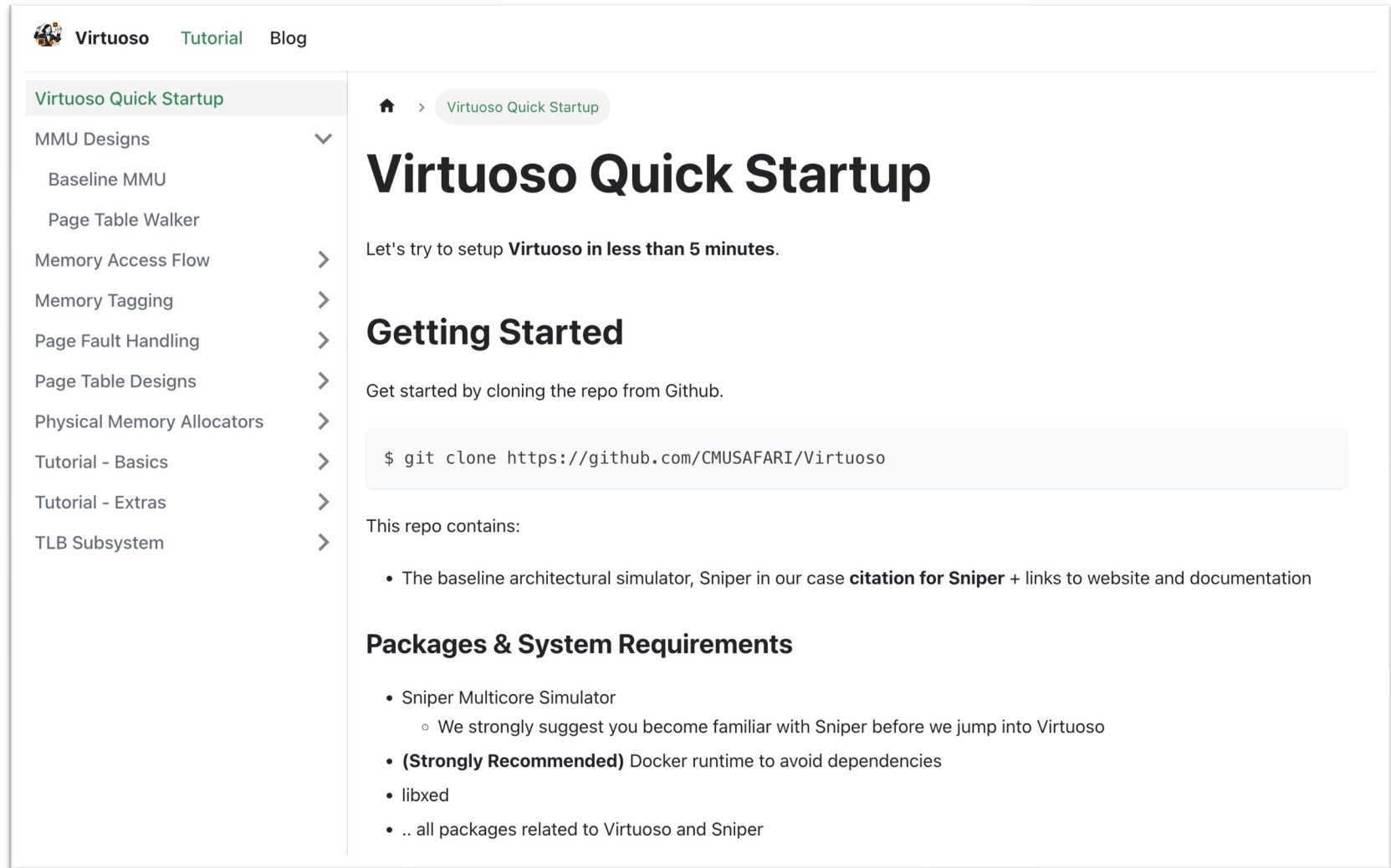
<https://github.com/CMU-SAFARI/Virtuoso>

Virtuoso Website – Getting Started



<https://github.com/CMU-SAFARI/Virtuoso/website>

Virtuoso Website - Documentation



The screenshot shows the 'Virtuoso Quick Startup' page on the Virtuoso website. The page has a navigation bar with 'Virtuoso', 'Tutorial', and 'Blog' links. A sidebar on the left lists various topics, with 'Virtuoso Quick Startup' highlighted. The main content area features a breadcrumb trail 'Virtuoso Quick Startup', a large title 'Virtuoso Quick Startup', and a sub-header 'Getting Started'. The text 'Let's try to setup Virtuoso in less than 5 minutes.' is followed by a code block containing the command to clone the repository. Below this, a list of contents is provided, including the baseline architectural simulator and system requirements like Docker, libx, and related packages.

Virtuoso Tutorial Blog

Virtuoso Quick Startup

MMU Designs

- Baseline MMU
- Page Table Walker

Memory Access Flow

Memory Tagging

Page Fault Handling

Page Table Designs

Physical Memory Allocators

Tutorial - Basics

Tutorial - Extras

TLB Subsystem

Home > Virtuoso Quick Startup

Virtuoso Quick Startup

Let's try to setup **Virtuoso** in less than 5 minutes.

Getting Started

Get started by cloning the repo from Github.

```
$ git clone https://github.com/CMUSAFARI/Virtuoso
```

This repo contains:

- The baseline architectural simulator, Sniper in our case **citation for Sniper** + links to website and documentation

Packages & System Requirements

- Sniper Multicore Simulator
 - We strongly suggest you become familiar with Sniper before we jump into Virtuoso
- (Strongly Recommended)** Docker runtime to avoid dependencies
- libx
- .. all packages related to Virtuoso and Sniper

<https://safari.ethz.ch/virtuoso>

Backup Slides

Tradeoffs in Simulation

- Three metrics to evaluate a simulator
 - Speed
 - Flexibility
 - Accuracy
- Speed: How fast the simulator runs (xIPS, xCPS, slowdown)
- Flexibility: How quickly one can modify the simulator to evaluate different algorithms and design choices?
- Accuracy: How accurate the performance (energy) numbers the simulator generates are vs. a real design (Simulation error)
- The relative importance of these metrics varies depending on where you are in the design process (what your goal is)

Trading Off Speed, Flexibility, Accuracy

- Speed & flexibility affect:
 - How quickly you can make design tradeoffs/decisions
- Accuracy affects:
 - How good your design tradeoffs/decisions **might** end up being
 - How fast you can build your simulator (**simulator design time**)
- Flexibility also affects:
 - How much human effort you need to spend modifying the simulator
- You can **trade off between the three to achieve design exploration and decision goals**

Making The Best of Architecture

- A good architect is comfortable at all levels of refinement
 - Including the extremes
- A good architect knows when to use what type of simulation
 - And, more generally, what type of evaluation method
- Recall: A variety of evaluation methods are available:
 - Theoretical proof
 - Analytical modeling
 - Simulation (at varying degrees of abstraction and accuracy)
 - Prototyping with a real system (e.g., FPGAs)
 - Real implementation

An Example Study using Ramulator (I)

- Saugata Ghose, Tianshi Li, Nastaran Hajinazar, Damla Senol Cali, and Onur Mutlu, **["Demystifying Workload–DRAM Interactions: An Experimental Study"](#)** *Proceedings of the [ACM International Conference on Measurement and Modeling of Computer Systems](#) (**SIGMETRICS**)*, Phoenix, AZ, USA, June 2019.
[[Preliminary arXiv Version](#)]
[[Abstract](#)]
[[Slides \(pptx\)](#) ([pdf](#))]
[[MemBen Benchmark Suite](#)]
[[Source Code for GPGPUSim-Ramulator](#)]

Demystifying Complex Workload–DRAM Interactions: An Experimental Study

Saugata Ghose[†]

Tianshi Li[†]

Nastaran Hajinazar^{‡†}

Damla Senol Cali[†]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[‡]Simon Fraser University

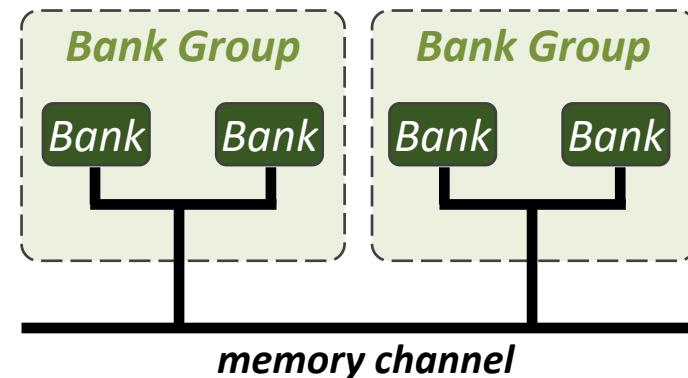
[§]ETH Zürich

- Manufacturers are developing many new types of DRAM
 - **DRAM limits performance, energy improvements:**
new types may overcome some limitations
 - Memory systems now serve a **very diverse set of applications:**
can no longer take a one-size-fits-all approach
- **So which DRAM type works best with which application?**
 - Difficult to understand intuitively due to the complexity of the interaction
 - Can't be tested methodically on real systems: new type needs a new CPU
- We perform a **wide-ranging experimental study to uncover the combined behavior** of workloads and DRAM types
 - **115 prevalent/emerging applications and multiprogrammed workloads**
 - **9 modern DRAM types:** DDR3, DDR4, GDDR5, HBM, HMC, LPDDR3, LPDDR4, Wide I/O, Wide I/O 2

Modern DRAM Types: Comparison to DDR3

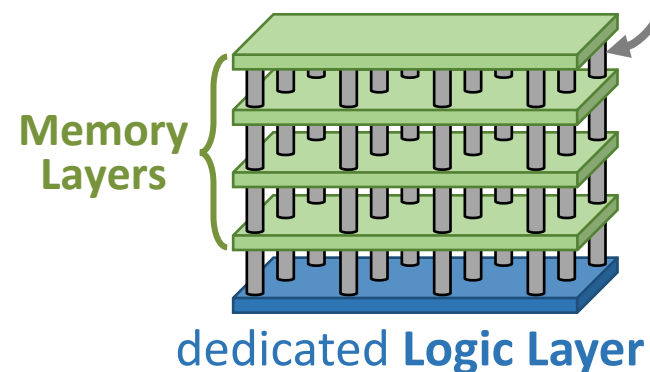
DRAM Type	Banks per Rank	Bank Groups	3D-Stacked	Low-Power
DDR3	8			
DDR4	16	✓	increased latency	
GDDR5	16	✓	increased area/power	
HBM High-Bandwidth Memory	16		✓	
HMC Hybrid Memory Cube	256	narrower rows, higher latency	✓	
Wide I/O	4		✓	✓
Wide I/O 2	8		✓	✓
LPDDR3	8			✓
LPDDR4	16			✓

■ Bank groups



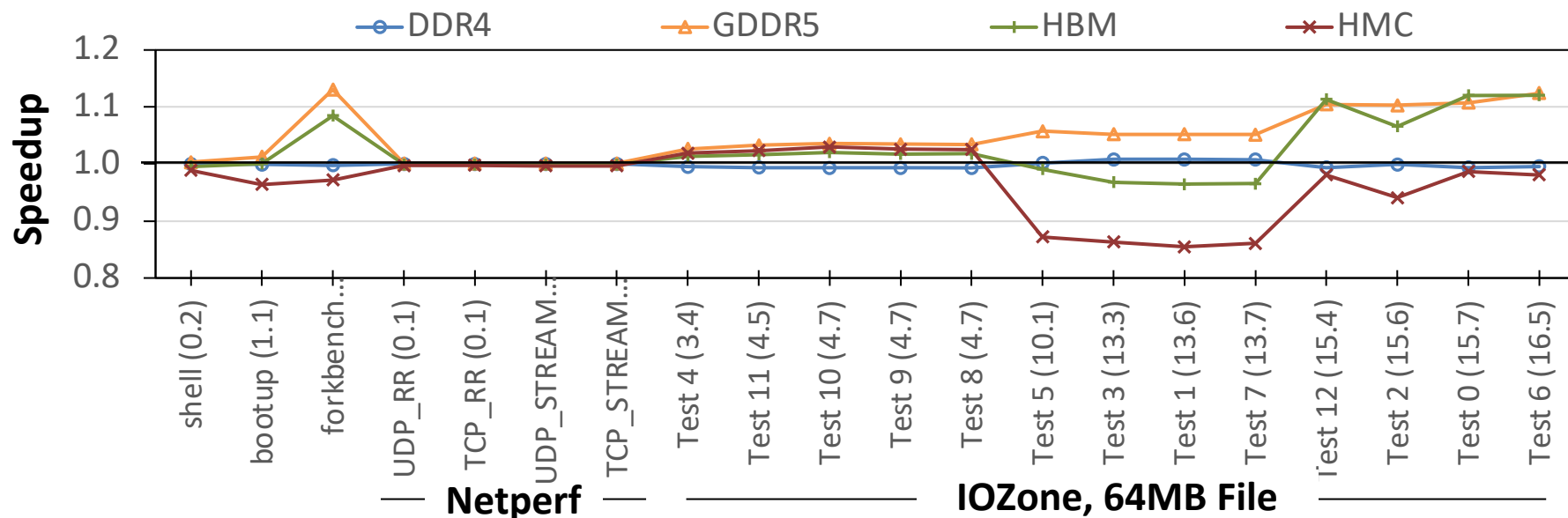
■ 3D-stacked DRAM

high bandwidth with
Through-Silicon Vias (TSVs)



4. Need for Lower Access Latency: Performance

- New DRAM types often increase access latency in order to provide more banks, higher throughput
- Many applications can't make up for the increased latency
 - Especially true of common OS routines (e.g., file I/O, process forking)



- A variety of desktop/scientific, server/cloud, GPGPU applications

Several applications don't benefit from more parallelism

1. DRAM latency remains a critical bottleneck for many applications
2. Bank parallelism is not fully utilized by a wide variety of our applications
3. Spatial locality continues to provide significant performance benefits if it is exploited by the memory subsystem
4. For some classes of applications, low-power memory can provide energy savings without sacrificing significant performance

- Manufacturers are developing many new types of DRAM
 - **DRAM limits performance, energy improvements:**
new types may overcome some limitations
 - Memory systems now serve a **very diverse set of applications:**
can no longer take a one-size-fits-all approach
 - Difficult to intuitively determine which DRAM–workload pair works best
- We perform a **wide-ranging experimental study to uncover the combined behavior** of workloads, DRAM types
 - 115 prevalent/emerging applications and multiprogrammed workloads
 - 9 modern DRAM types
- 12 key observations on DRAM–workload behavior

Open-source tools: <https://github.com/CMU-SAFARI/ramulator>

Full paper: <https://arxiv.org/pdf/1902.07609>

For More Information...

- Saugata Ghose, Tianshi Li, Nastaran Hajinazar, Damla Senol Cali, and Onur Mutlu, **"Demystifying Workload–DRAM Interactions: An Experimental Study"**
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**)*, Phoenix, AZ, USA, June 2019.
[[Preliminary arXiv Version](#)]
[[Abstract](#)]
[[Slides \(pptx\)](#) ([pdf](#))]
[[MemBen Benchmark Suite](#)]
[[Source Code for GPGPUSim-Ramulator](#)]

Demystifying Complex Workload–DRAM Interactions: An Experimental Study

Saugata Ghose[†]

Tianshi Li[†]

Nastaran Hajinazar^{‡†}

Damla Senol Cali[†]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[‡]Simon Fraser University

[§]ETH Zürich

BlockHammer Study in 2021

- A. Giray Yaglikci, Minesh Patel, Jeremie S. Kim, Roknoddin Azizi, Ataberk Olgun, Lois Orosa, Hasan Hassan, Jisung Park, Konstantinos Kanellopoulos, Taha Shahroodi, Saugata Ghose, and Onur Mutlu,

"BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows"

Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Short Talk Slides \(pptx\)](#) ([pdf](#))]

[[Intel Hardware Security Academic Awards Short Talk Slides \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (22 minutes)]

[[Short Talk Video](#) (7 minutes)]

[[Intel Hardware Security Academic Awards Short Talk Video](#) (2 minutes)]

[[BlockHammer Source Code](#)]

Intel Hardware Security Academic Award Finalist (one of 4 finalists out of 34 nominations)

BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows

A. Giray Yağlıkçı¹ Minesh Patel¹ Jeremie S. Kim¹ Roknoddin Azizi¹ Ataberk Olgun¹ Lois Orosa¹
Hasan Hassan¹ Jisung Park¹ Konstantinos Kanellopoulos¹ Taha Shahroodi¹ Saugata Ghose² Onur Mutlu¹

¹ETH Zürich

²University of Illinois at Urbana-Champaign

Summary: BlockHammer

- BlockHammer is **the first work to practically enable throttling-based RowHammer mitigation**
- BlockHammer is implemented in **the memory controller** (*no proprietary information of / no modifications to DRAM chips*)
- BlockHammer is *both* **scalable with worsening RowHammer** and **compatible with commodity DRAM chips**
- BlockHammer is **open-source** along with **six state-of-the-art mechanisms**: <https://github.com/CMU-SAFARI/BlockHammer>



BlockHammer: Free & Open Source

☰ README.md

BlockHammer

Aggressive memory density scaling causes modern DRAM devices to suffer from [RowHammer](#), a phenomenon where rapidly activating a DRAM row can cause bit-flips in physically-nearby rows. Recent studies [1, 2, 3] demonstrate that modern DRAM chips, including chips previously marketed as RowHammer-safe, are even more vulnerable to RowHammer than older chips. Many works show that attackers can exploit RowHammer bit-flips to reliably mount system-level attacks to escalate privilege and leak private data. Therefore, it is critical to ensure RowHammer-safe operation on all DRAM-based systems.

Unfortunately, state-of-the-art RowHammer mitigation mechanisms face two major challenges. First, they incur increasingly higher performance and/or area overheads when applied to more vulnerable DRAM chips. Second, they require either proprietary information about or modifications to the DRAM chip design. In this paper, we show that it is possible to efficiently and scalably prevent RowHammer bit-flips without knowledge of or modification to DRAM internals.

We introduce BlockHammer, a low-cost, effective, and easy-to-adopt RowHammer mitigation mechanism that overcomes the two key challenges by selectively throttling memory accesses that could otherwise cause RowHammer bit-flips. The key idea of BlockHammer is to (1) track row activation rates using area-efficient Bloom filters and (2) use the tracking data to ensure that no row is ever activated rapidly enough to induce RowHammer bit-flips. By doing so, BlockHammer (1) makes it impossible for a RowHammer bit-flip to occur and (2) greatly reduces a RowHammer attack's impact on the performance of co-running benign applications.

Compared to state-of-the-art RowHammer mitigation mechanisms, BlockHammer provides competitive performance and energy when the system is not under a RowHammer attack and significantly better performance and energy when the system is under attack.

The full paper is published in [HPCA 2021](#) and the pdf is available online: [arXiv:2102.05981 \[cs.CR\]](#)

Citation

Please cite our full HPCA 2021 paper if you find this repository useful.

A. Giray Yaglikci, Minesh Patel, Jeremie S. Kim, Roknoddin Azizi, Ataberk Olgun, Lois Orosa, Hasan Hassan, Jisung Park, Konstantinos Kanellopoulos, Taha Shahroodi, Saugata Ghose, and Onur Mutlu, "[BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows](#)" Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.

Getting Started

This repository has two subdirectories. Please refer to each subdirectory on reproducing results.

- [Ramulator Model](#): This subdirectory includes an extended version of [Ramulator](#) with a RowHammerDefense class, which implements BlockHammer along with six state-of-the-art RowHammer mitigation mechanisms:

Mechanism	Reference
PARA	Y. Kim, et al., "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," in ISCA, 2014.
CBT	S. M. Seyedzadeh et al., "Mitigating Wordline Crosstalk Using Adaptive Trees of Counters," in ISCA, 2018.
ProHIT	M. Son et al., "Making DRAM Stronger Against Row Hammering," in DAC, 2017.
MrLoc	J. M. You et al., "MRLoc: Mitigating Row-Hammering Based on Memory Locality," in DAC, 2019.
TWjCe	E. Lee et al., "TWjCe: Preventing Row-hammering by Exploiting Time Window Counters" in ISCA, 2019.
Graphene	Y. Park et al., "Graphene: Strong yet Lightweight Row Hammer Protection," in MICRO, 2020.

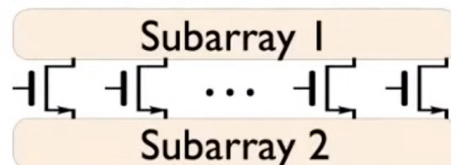
- [RTL Model](#): This subdirectory includes RTL implementation of the counters and buffers used in BlockHammer.

Many Other Ideas Evaluated w/ Ramulator

Key Idea and Applications

- **Low-cost Inter-linked subarrays (LISA)**

- Fast bulk data movement between subarrays
- Wide datapath via isolation transistors: 0.8% DRAM chip area



- LISA is a **versatile substrate** → new applications

Fast bulk data copy: Copy latency 1.363ms → 0.148ms (9.2x)

→ 66% speedup, -55% DRAM energy

In-DRAM caching: Hot data access latency 48.7ns → 21.5ns (2.2x)

→ 5% speedup

Fast precharge: Precharge latency 13.1ns → 5.0ns (2.6x)

→ 8% speedup



zoom

Computer Architecture - Lecture 9: Memory Latency & Memory Controllers (Fall 2022)



Onur Mutlu Lectures
28.8K subscribers

Analytics

Edit video

29



Share

Download

Clip

Save



711 views Streamed live on Oct 27, 2022

Computer Architecture, ETH Zürich, Fall 2022 (<https://safari.ethz.ch/architecture/f...>)

Ramulator for Processing in Memory

Simulation Infrastructure for PIM

- **Ramulator** extended for PIM
 - Flexible and extensible DRAM simulator
 - Can model many different memory standards and proposals
 - Kim+, “**Ramulator: A Flexible and Extensible DRAM Simulator**”, IEEE CAL 2015.
 - <https://github.com/CMU-SAFARI/ramulator-pim>
 - <https://github.com/CMU-SAFARI/ramulator>
 - [[Source Code for Ramulator-PIM](#)]

Ramulator: A Fast and Extensible DRAM Simulator

Yoongu Kim¹ Weikun Yang^{1,2} Onur Mutlu¹
¹Carnegie Mellon University ²Peking University

Ramulator for PIM

- Gagandeep Singh, Juan Gomez-Luna, Giovanni Mariani, Geraldo F. Oliveira, Stefano Corda, Sander Stujik, Onur Mutlu, and Henk Corporaal, **"NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning"**
Proceedings of the 56th Design Automation Conference (DAC), Las Vegas, NV, USA, June 2019.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Poster \(pptx\)](#)] [[pdf](#)]
[[Source Code for Ramulator-PIM](#)]

NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning

Gagandeep Singh ^{a,c}	Juan Gómez-Luna ^b	Giovanni Mariani ^c	Geraldo F. Oliveira ^b
Stefano Corda ^{a,c}	Sander Stuijk ^a	Onur Mutlu ^b	Henk Corporaal ^a
^a Eindhoven University of Technology		^b ETH Zürich	^c IBM Research - Zurich

Ramulator Project Course

Exploration of Emerging Memory Systems (Spring/Fall 2022)

Fall 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator

Spring 2022 Edition:

- ❑ https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator

Youtube Livestream (Spring 2022):

- ❑ https://www.youtube.com/watch?v=aM-lIXRQd3s&list=PL5Q2soXY2Zi_TlmlGw_Z8hBo2925ZApqV

Bachelor's course

- ❑ Elective at ETH Zurich
- ❑ Introduction to memory system simulation
- ❑ Tutorial on using Ramulator
- ❑ C++
- ❑ Potential research exploration

<https://www.youtube.com/onurmutlulectures>

Lecture Video Playlist on YouTube

Lecture Playlist



2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	09.03 Wed.	YouTube Video	M1: Logistics & Intro to Simulating Memory Systems Using Ramulator PDF (PDF) PPT (PPT)		HW0
W2	16.03 Fri.	YouTube Video	M2: Tutorial on Using Ramulator PDF (PDF) PPT (PPT)		
W3	25.02 Fri.	YouTube Video	M3: BlockHammer PDF (PDF) PPT (PPT)		
W4	01.04 Fri.	YouTube Video	M4: CLR-DRAM PDF (PDF) PPT (PPT)		
W5	08.04 Fri.	YouTube Video	M5: SIMDRAM PDF (PDF) PPT (PPT)		
W6	29.04 Fri.	YouTube Video	M6: DAMOV PDF (PDF) PPT (PPT)		
W7	06.05 Fri.	YouTube Video	M7: Synchron PDF (PDF) PPT (PPT)		

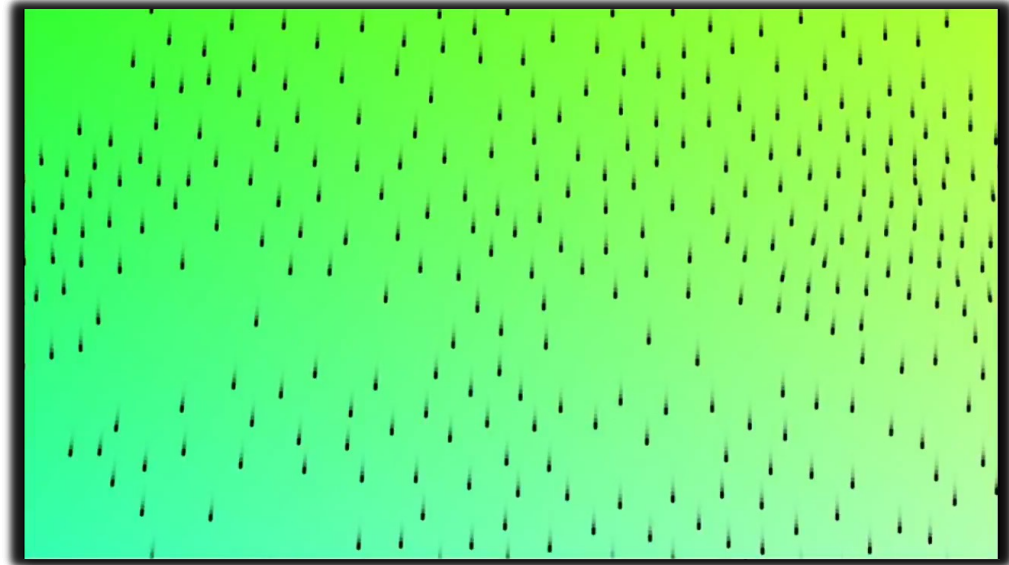
What We Discussed Is Applicable to
Simulation in Other Domains

Agent-Based Simulation

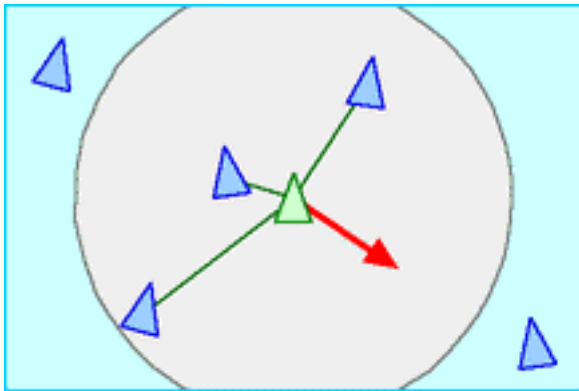
Introduction to Agent-Based Modeling: Modeling a Flock of Birds

- Agent is a bird with the following attributes:

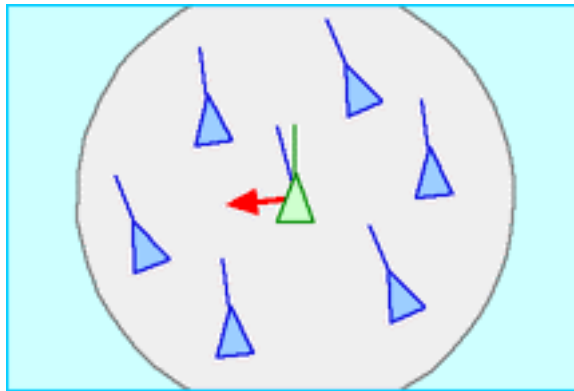
- position
- velocity



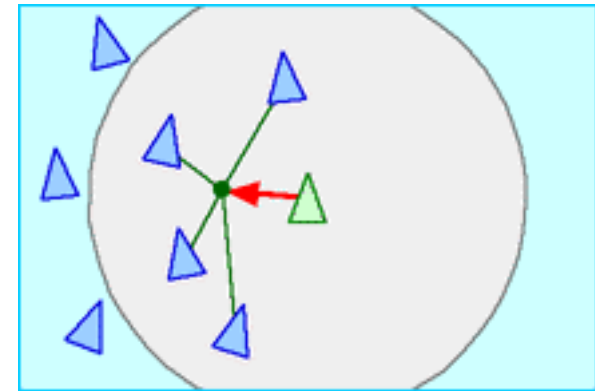
- And three behaviors:



Separation

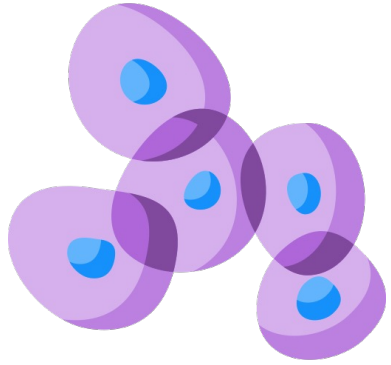


Alignment

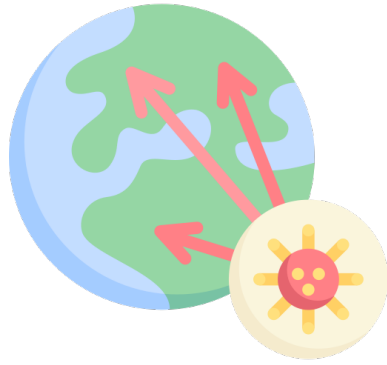


Cohesion

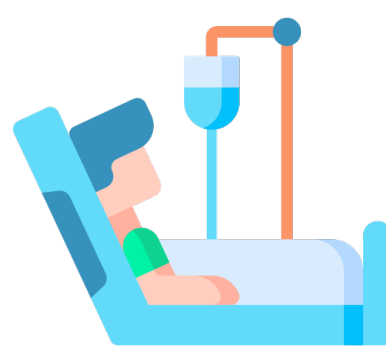
Agent-Based Simulation is Versatile



Biology



Epidemiology



Medicine



Social Sciences



Finance



Transport



Ecology

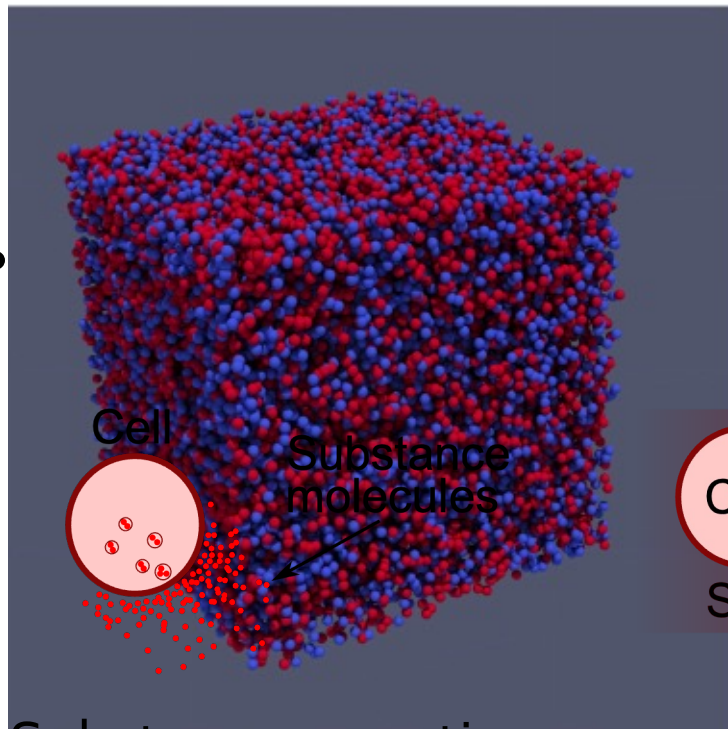


Marketing

Cell Sorting in BioDynaMo

- Agent

Initial state

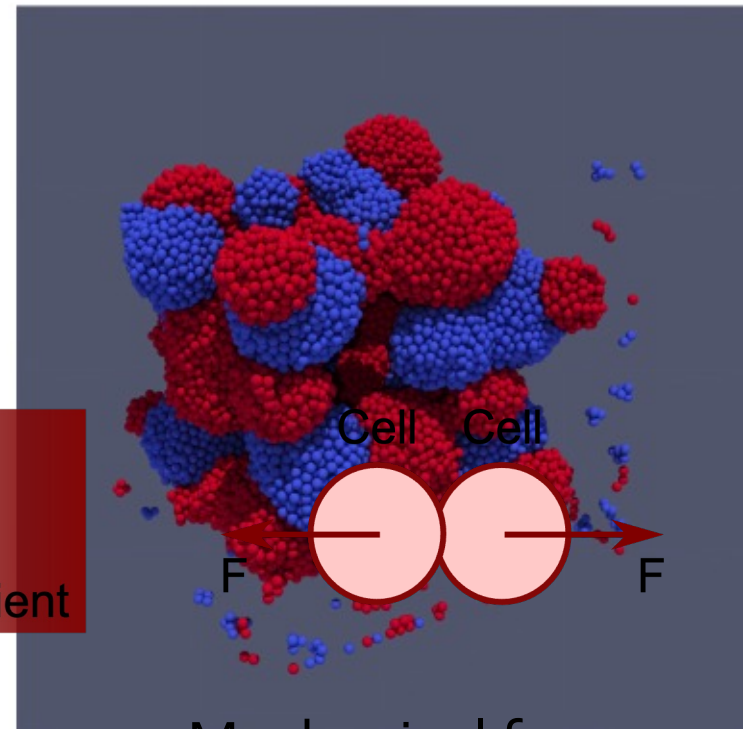


Substance secretion



Chemotaxis

Final state



Mechanical forces

BioDynamo Modeling Infrastructure

- Lukas Breitwieser, Ahmad Hesam, Fons Rademakers, Juan Gómez Luna, and Onur Mutlu, **"High-Performance and Scalable Agent-Based Simulation with BioDynaMo"**
Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming (PPoPP), Montreal, QC, Canada, February 2023.
[[arXiv version](#)]
[[Slides \(pdf\)](#) ([web](#))]
[[BioDynamo Source Code \(Officially Artifact Evaluated with All Badges\)](#)]
Officially artifact evaluated as available, reusable and reproducible.
Best artifact award at PPoPP 2023.

High-Performance and Scalable Agent-Based Simulation with BioDynaMo

Lukas Breitwieser*

CERN, Switzerland
ETH Zurich, Switzerland

Ahmad Hesam

Delft University of Technology,
The Netherlands

Fons Rademakers

CERN, Switzerland

Juan Gómez Luna

ETH Zurich, Switzerland

Onur Mutlu[†]

ETH Zurich, Switzerland

BioDynamo Modeling Infrastructure

- Lukas Breitwieser, Ahmad Hesam, Jean de Montigny, Vasileios Vavourakis, Alexandros Iosif, Jack Jennings, Marcus Kaiser, Marco Manca, Alberto Di Meglio, Zaid Al-Ars, Fons Rademakers, Onur Mutlu, and Roman Bauer,
"BioDynaMo: A General Platform for Scalable Agent-based Simulation"
Bioinformatics, 16 September 2021.
[[Online link at Bioinformatics Journal](#)]
[Preprint in [arXiv](#) and [bioRxiv](#), 5 February 2021]
[[bioRxiv preprint](#)]
[[arXiv preprint](#)]
[[BioDynaMo Source Code and Data](#)]

Systems biology

BioDynaMo: a general platform for scalable agent-based simulation

Lukas Breitwieser^{1,2,*}, Ahmad Hesam^{1,3,*}, Jean de Montigny¹, Vasileios Vavourakis^{4,5}, Alexandros Iosif⁴, Jack Jennings⁶, Marcus Kaiser^{6,7,8}, Marco Manca⁹, Alberto Di Meglio¹, Zaid Al-Ars³, Fons Rademakers¹, Onur Mutlu^{2,*}, Roman Bauer^{10,*}

Case Study: COVID-19 Spread Modeling and Prediction



COVIDHunter: COVID-19 Pandemic Wave Prediction and Mitigation via Seasonality Aware Modeling

Mohammed Alser*, Jeremie S. Kim, Nour Almadhoun Alser, Stefan W. Tell and Onur Mutlu

Department of Information Technology and Electrical Engineering (D-ITET), ETH Zurich, Zurich, Switzerland

OPEN ACCESS

Edited by:

Zisis Kozlakidis,
International Agency For Research On
Cancer (IARC), France

Reviewed by:

Shudi Li,
University of Texas Health Science
Center at Houston, United States
Vivek Bora,
Nirma University, India

*Correspondence:

Mohammed Alser
alserm@ethz.ch

Specialty section:

This article was submitted to
Infectious Diseases – Surveillance,
Prevention and Treatment,
a section of the journal
Frontiers in Public Health

Received: 16 February 2022

Accepted: 20 May 2022

Published: 17 June 2022

Citation:

Alser M, Kim JS, Almadhoun Alser N,
Tell SW and Mutlu O (2022)
COVIDHunter: COVID-19 Pandemic
Wave Prediction and Mitigation via
Seasonality Aware Modeling.
Front. Public Health 10:877621.
doi: 10.3389/fpubh.2022.877621

Early detection and isolation of COVID-19 patients are essential for successful implementation of mitigation strategies and eventually curbing the disease spread. With a limited number of daily COVID-19 tests performed in every country, simulating the COVID-19 spread along with the potential effect of each mitigation strategy currently remains one of the most effective ways in managing the healthcare system and guiding policy-makers. We introduce COVIDHunter, a flexible and accurate COVID-19 outbreak simulation model that evaluates the current mitigation measures that are applied to a region, predicts COVID-19 statistics (the daily number of cases, hospitalizations, and deaths), and provides suggestions on what strength the upcoming mitigation measure should be. The key idea of COVIDHunter is to quantify the spread of COVID-19 in a geographical region by simulating the average number of new infections caused by an infected person considering the effect of external factors, such as environmental conditions (e.g., climate, temperature, humidity), different variants of concern, vaccination rate, and mitigation measures. Using Switzerland as a case study, COVIDHunter estimates that we are experiencing a deadly new wave that will peak on 26 January 2022, which is very similar in numbers to the wave we had in February 2020. The policy-makers have only one choice that is to increase the strength of the currently applied mitigation measures for 30 days. Unlike existing models, the COVIDHunter model accurately monitors and predicts the daily number of cases, hospitalizations, and deaths due to COVID-19. Our model is flexible to configure and simple to modify for modeling different scenarios under different environmental conditions and mitigation measures. We release the source code of the COVIDHunter implementation at <https://github.com/CMU-SAFARI/COVIDHunter> and show how to flexibly configure our model for any scenario and easily extend it for different measures and conditions than we account for.

Keywords: epidemiological modeling, COVID-19 outbreak simulation, seasonal epidemic, outbreak prevention and control, vaccination

<https://arxiv.org/pdf/2102.03667.pdf>

INTRODUCTION

Coronavirus disease 2019 (COVID-19) is caused by SARS-CoV-2 virus, which has rapidly spread to nearly every corner of the globe and has been declared a pandemic in March 2020 by the World Health Organization (WHO) (1). As of November 2021, only about 40% of the entire world population is fully vaccinated and their protection wanes after a few months (2). Until an effective drug or vaccination is made widely available to everyone, early detection and isolation of

COVID-19 Measures: Evaluation Methods

- How do we assess how an idea will affect a target metric X?
- A variety of evaluation methods are available:
 - Theoretical proof
 - Analytical modeling/estimation
 - Simulation (at varying degrees of abstraction and accuracy)
 - Prototyping with a real system (e.g., FPGAs)
 - Real implementation

Simulating & Predicting COVID-19 Spread

- An architect is in part a dreamer, a creator
- Simulation is a key tool of the architect
 - Allows the evaluation & understanding of non-existent systems
- Simulation enables
 - The exploration of many dreams
 - A reality check of the dreams
 - Deciding which dream is better
- Simulation also enables
 - The ability to fool yourself with false dreams

Simulating & Predicting COVID-19 Spread

To our knowledge, there is currently no model capable of accurately monitoring the current epidemiological situation and predicting future scenarios while considering a reasonably low number of parameters and accounting for the effects of environmental conditions (**Table 1**).

Our **goal** in this work is to develop and validate such a COVID-19 outbreak simulation model. To this end, we introduce COVIDHunter, a simulation model that evaluates the current mitigation measures (i.e., non-pharmaceutical intervention or NPI) that are applied to a region and provides insight into what strength the upcoming mitigation measure should be and for how long it should be applied, while considering the potential effect of environmental conditions. Our model accurately forecasts the

TABLE 1 | Comparison to other models used to inform government policymakers, as of January 2021.

Model	Open source	Well documented [#]	Accounting for seasonality	Low number of parameters	Reported COVID-19 statistics
COVIDHunter (this work)	✓	✓	✓	✓	✓ (<i>R</i> , cases, hospitalizations, and deaths)
IBZ (11)	✓	✗	✗	✓	✗ (only <i>R</i>)
LSHTM (7)	✓	✗	✗	✓	✗ (only cases)
ICL (9)	✓	✓	✗	✗	✓ (<i>R</i> , cases, hospitalizations, and deaths)
IHME (10)	✓*	✗	✗	✗	✗ (cases, hospitalizations, and deaths)

*The available packages are configured only for the IHME infrastructure. # Based on the documentation available on each model's GitHub page (all models are available on GitHub).

Simulating & Predicting COVID-19 Spread

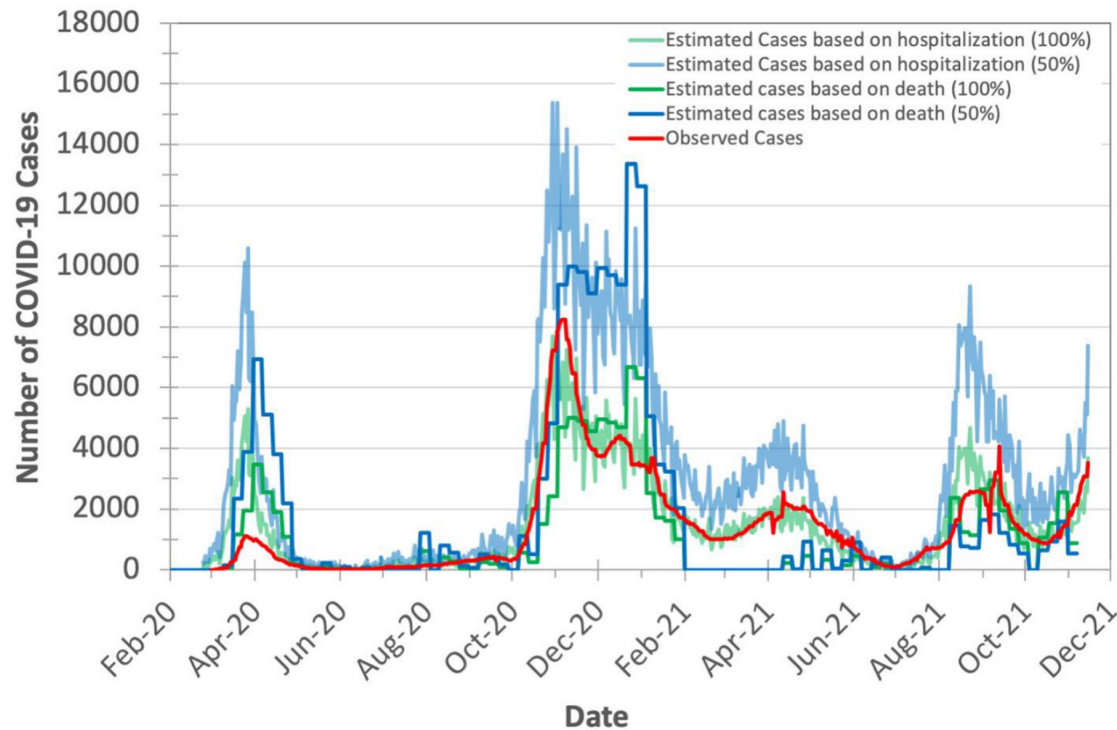


FIGURE 4 | Observed (officially reported) and expected number of COVID-19 cases in Switzerland during the years 2020 and 2021. We calculate the expected number of cases based on both the hospitalizations-to-cases and deaths-to-cases ratios for the second wave. We assume two certainty rate levels of 50 and 100%.

Predicting Effectiveness of Measures

Epidemiological Situation in Switzerland Using COVIDHunter

We use Switzerland as a use-case for all the experiments. However, our model is not limited to any specific region as the parameters of COVIDHunter are completely configurable. Using COVIDHunter on **20 November 2021**, we make four key observations:

- Prediction
- The spread of COVID-19 in Switzerland is **still active** as the reproduction number (R) is still greater than 1.0. The R number value will remain greater than 1 until at least February 2022.
 - COVIDHunter estimates that we are experiencing a deadly new wave that will peak on the last week of January 2022, which is very similar in numbers to the wave we had in February 2020.
 - The policy-makers have only one choice that is increasing the strength of the currently applied mitigation measures for 30 days. Relaxing the mitigation measures should not be an option before at least February 2022 as it would increase exponentially the number of cases, hospitalizations, and deaths by 5.5x.
 - COVIDHunter forecasts the effect of **relaxing the current mitigation measures on November 20, 2021** on the daily maximum number of COVID-19 cases, hospitalizations, and deaths as follows:

Strengths of the mitigation measures during November-December 2021	0.2	0.3	0.40	0.50	0.60	0.70
Mitigation measures with similar strength were applied on	29 February - 11 March 2020	1 October - 20 October 2020	28 February - 15 March 2020	24 June - 20 August 2020	12-29 October 2020	28 November - 22 December 2020
Predicted daily number of cases	7'822-43'518	5'706-22'413	3'401-10'714	542-1'310	70-410	3-104
Predicted daily number of hospitalizations	124-693	90-357	54-170	9-21	1-7	1-2
Predicted daily number of deaths	38-216	28-112	16-53	3-7	1-3	1-2

