



Cleaning up the Mess

Re-Evaluating the Real-System Modeling Accuracy of Ramulator 2.0

F. Nisa Bostancı*

Haocong Luo*

Ataberk Olgun*

Maria Makeenkova

Geraldo F. de Oliveira

A. Giray Yağlıkçı

Onur Mutlu

* Equal Contribution

SAFARI

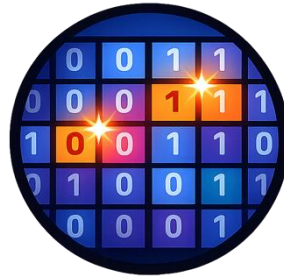
ETH zürich

DRAM and Memory System Simulation

- Cycle-level DRAM simulators provide **accurate** and **flexible models** for DRAM and memory controller operations
- Enable research on **current** and **future memory systems**



**DRAM performance
and latency**



**Solutions for
robustness issues**



**Emerging
architectures**

- Critical for **improving the performance, efficiency, and robustness (security, safety, reliability)** of memory systems

Mess: A MICRO'24 Best Paper Runner-Up

- ***"A Mess of Memory System Benchmarking, Simulation and Application Profiling"***, MICRO'24
 - **A best paper runner-up** with **all 3 artifact badges** (including *"Reproducible"*)
- Proposes **a new benchmark** to evaluate **the real** and **simulated memory system performance**
- Makes **strong negative claims** about two of our simulation infrastructures, Ramulator 2.0 and DAMOV, and shows **unexpected results**
 - Contending that they *"poorly resemble the actual system performance"*

Cleaning up the Mess

We demonstrate that **these claims and results are incorrect** and are due to **configuration** and **simulator usage errors** made in the Mess paper

We describe **four best practices** to aid users and developers of simulation tools

To facilitate transparency and reproducibility, we **open source all our code** and **results**:
<https://github.com/CMU-SAFARI/Cleaning-up-the-Mess>

Cleaning up the Mess

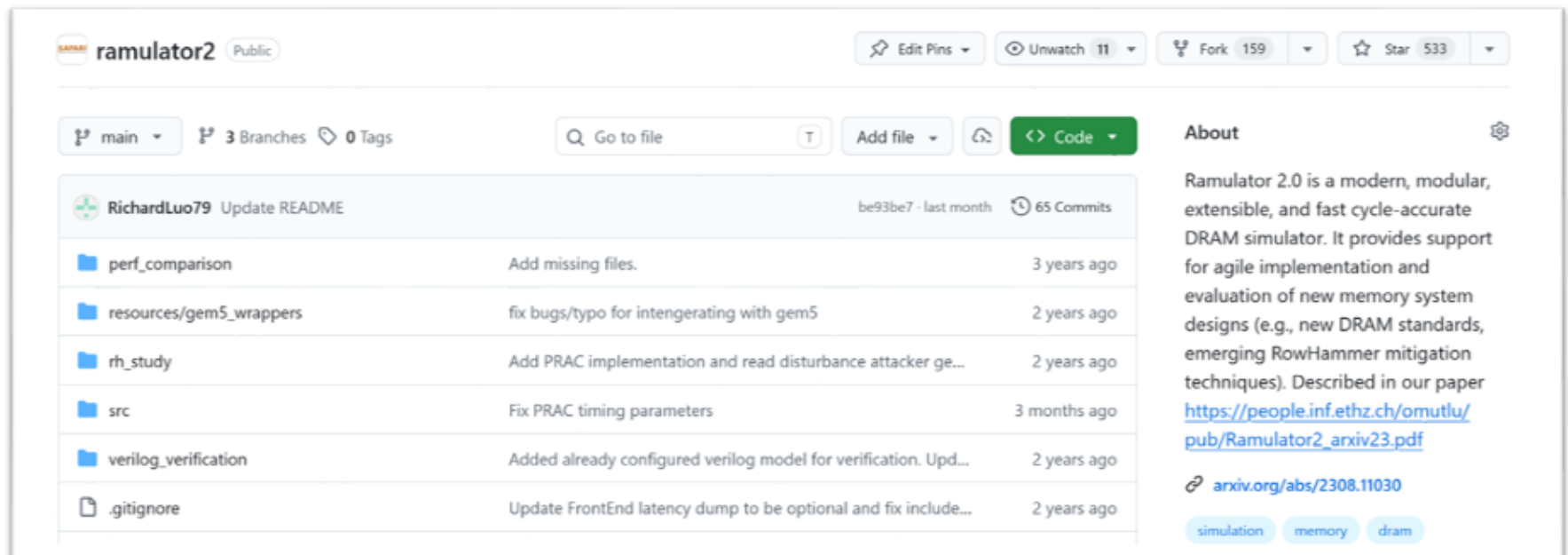
**Analysis of the Evaluation of Ramulator 2.0
in the Mess Paper**

Analysis of the Evaluation of DAMOV
in the Mess Paper

Ramulator 2.0 and DAMOV Best Practices

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

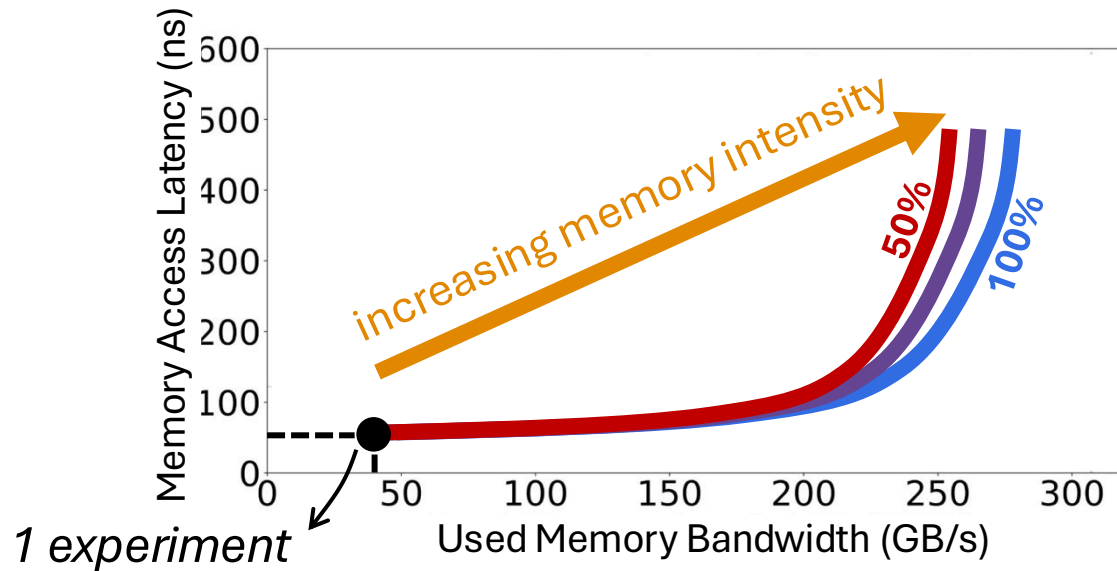
- **Ramulator 2.0** [Luo+, CAL'24] models a wide range of **DRAM standards** and **memory controller functionalities**
 - **Modular and extensible** software architecture
- Enables **fine-grained modeling** of DRAM operation (cycle-level)
- Used in a wide range of research works



High-Level Memory System

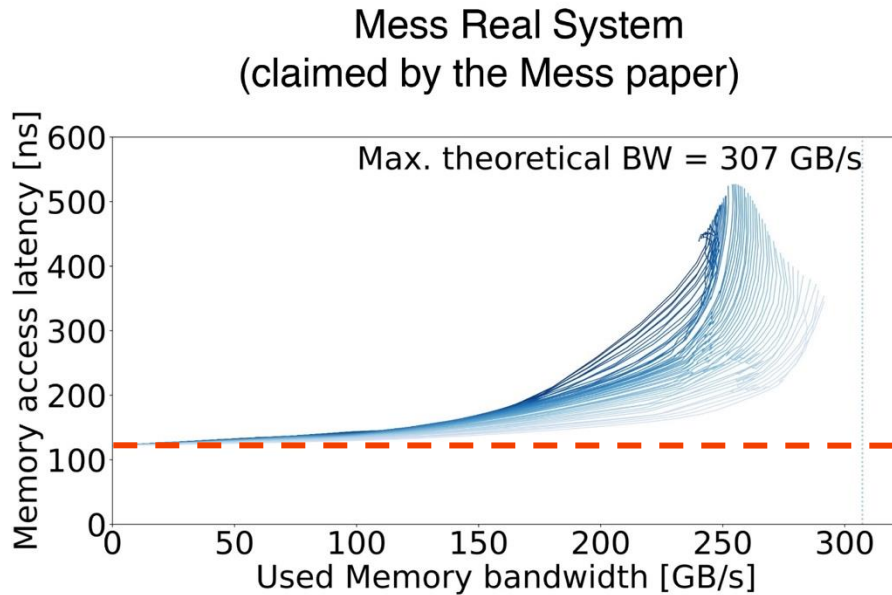
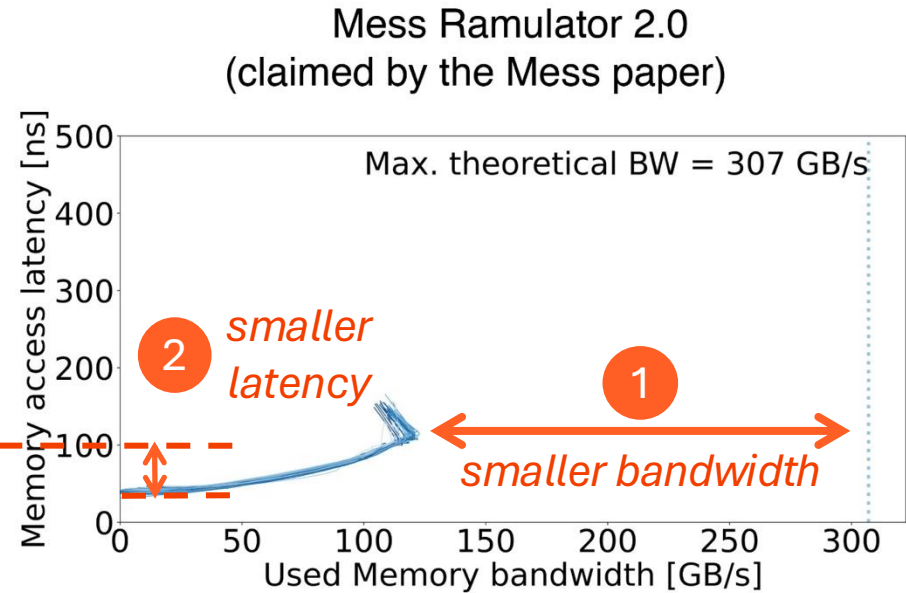
Performance Evaluation & Reasoning

- **The Mess benchmark** aims to characterize a memory system based on a set of *latency-throughput* ("used bandwidth") curves



- Consists of two concurrently running workloads: *Streaming* and *Pointer-Chasing* patterns
 - **Varies the memory intensity** using NOPs between requests
 - Evaluates **a sweep of read/write intensities**
 - (from **50%** to **100%** reads)

The Mess Paper's Evaluation of Ramulator 2.0



- Contending that Ramulator 2.0 *"poorly resembles the actual system performance"*

Ramulator 2.0 simulation results in the Mess paper are **incorrect** and, at the time of the publication of the Mess paper, **irreproducible**

Analysis of the Evaluation of Ramulator 2.0 in the Mess Paper

Reproducibility Issues:

1

It was *not possible* to reproduce *any* Ramulator 2.0 result from the Mess paper artifacts available at the time of publication and artifact evaluation of the paper

The missing source codes were uploaded to the Mess GitHub repository 10 months after the artifact evaluation badges

Fundamental Configuration Errors:

2

- 1) The Mess paper effectively compares an **8-channel** DDR5 memory system in Ramulator 2.0 against a **16-channel** real DDR5 memory system
- 2) The Mess paper uses **unrealistically low cache latency configurations** that lead to **low memory access latency**

Scholarship Issues:

3

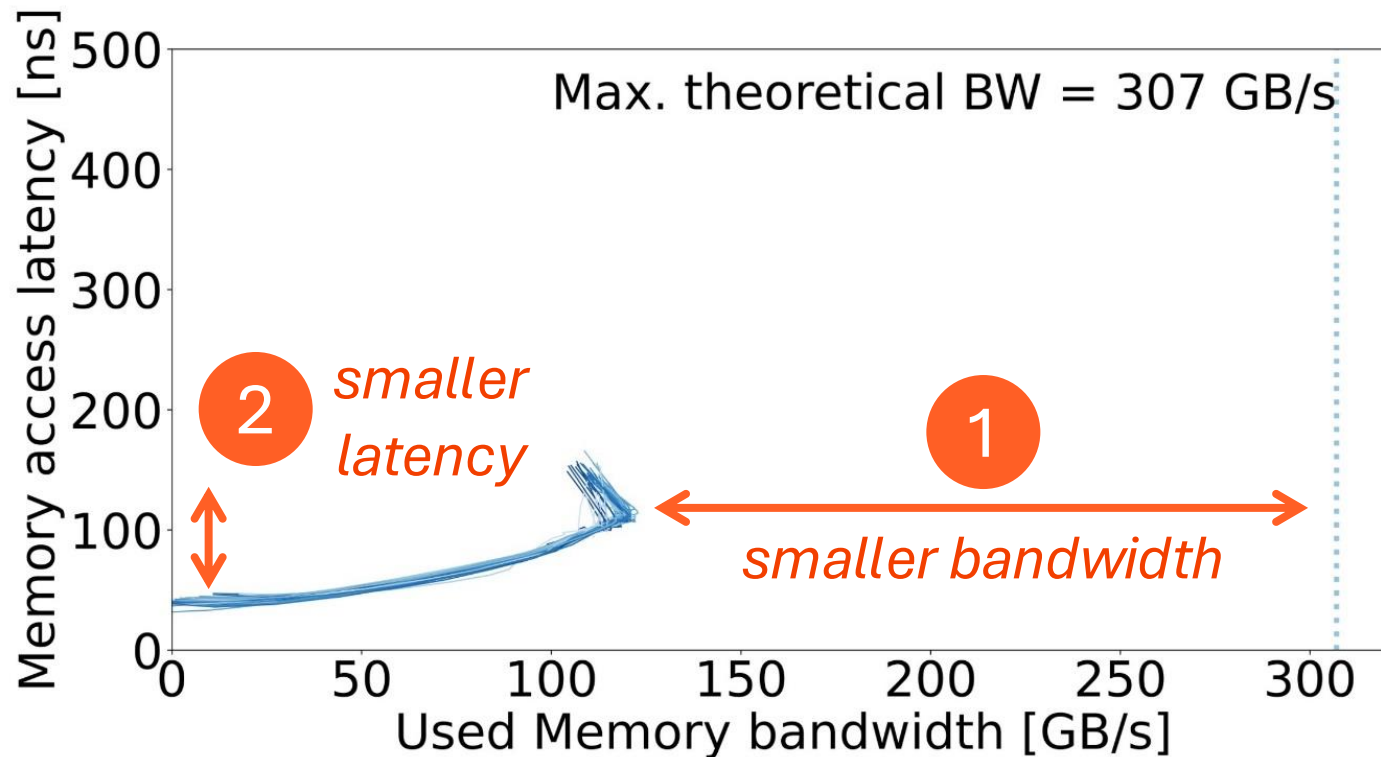
The Mess paper **does not disclose**

- 1) **modifications** to the cache latency configurations, and
- 2) **differences in evaluation methodology** between simulation and real system results (identified via email exchange with Mess paper's authors)

Fundamental Configuration Errors

- **The Mess paper's configuration errors** result in the claimed Ramulator 2.0 inaccuracies

Mess Ramulator 2.0
(claimed by the Mess paper)



Fundamental Configuration Errors: Smaller Used Bandwidth

Mess Paper's
Ramulator 2.0 Configuration

channel: 8



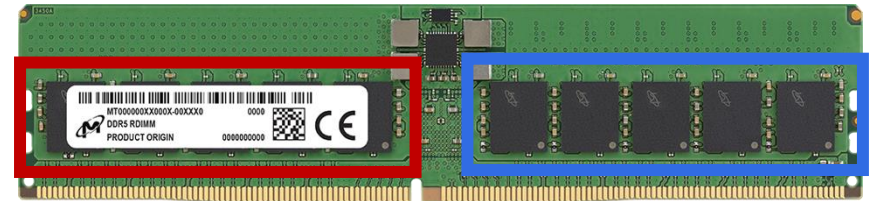
8 DDR5 channels

Mess Paper's
Real System Configuration

“8 x DDR5”

Channel A

Channel B



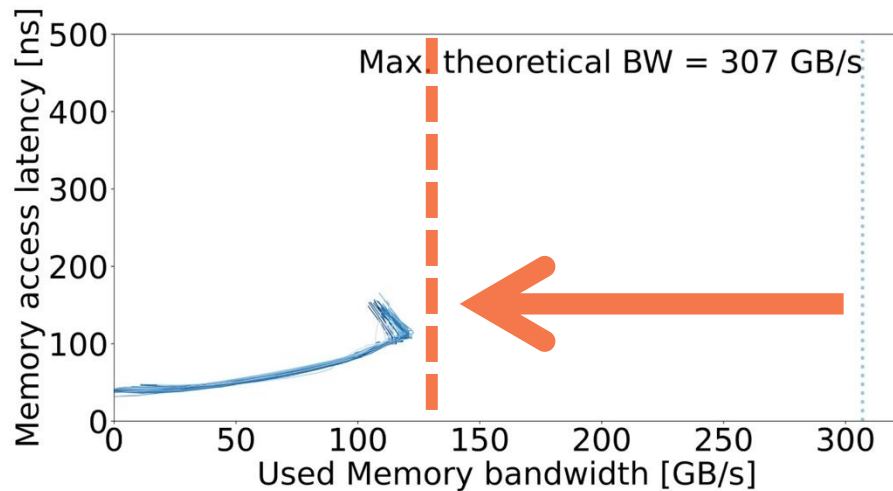
16 DDR5 channels

Fundamental Configuration Errors: Smaller Used Bandwidth

Mess Paper's Ramulator 2.0 Configuration

8 DDR5 channels

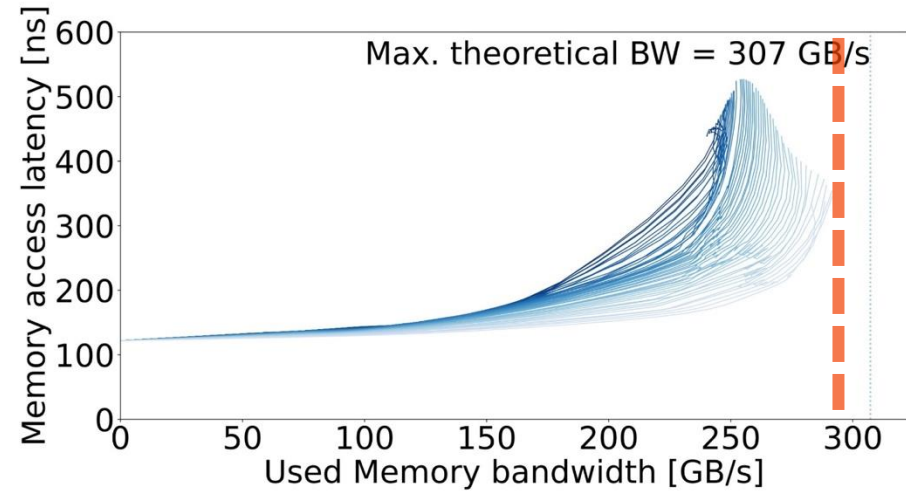
Mess Ramulator 2.0
(claimed by the Mess paper)



Mess Paper's Real System Configuration

16 DDR5 channels

Mess Real System
(claimed by the Mess paper)



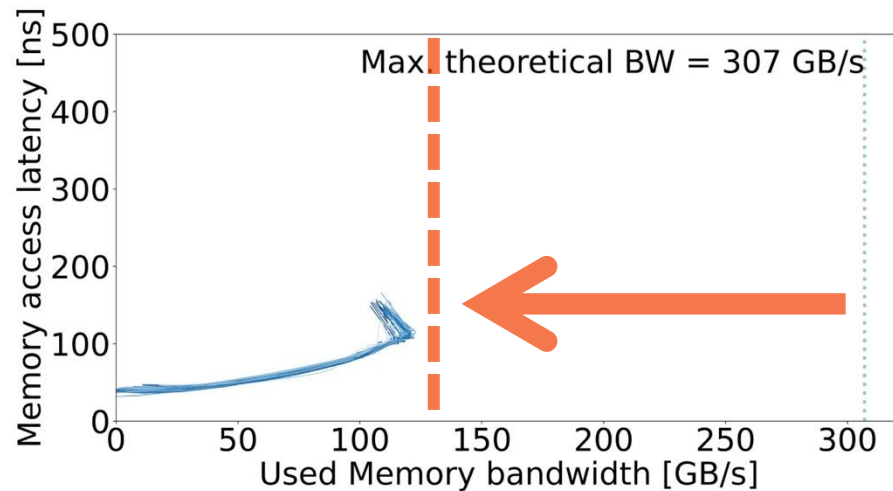
The Mess paper **incorrectly assumes** a DDR5 channel in Ramulator 2.0 is **64-bit**, instead of **the actual width of 32-bit** that is **consistent with JEDEC DDR5 standards** (JESD79-5)

Fundamental Configuration Errors: Smaller Used Bandwidth

Mess Paper's Ramulator 2.0 Configuration

8 DDR5 channels

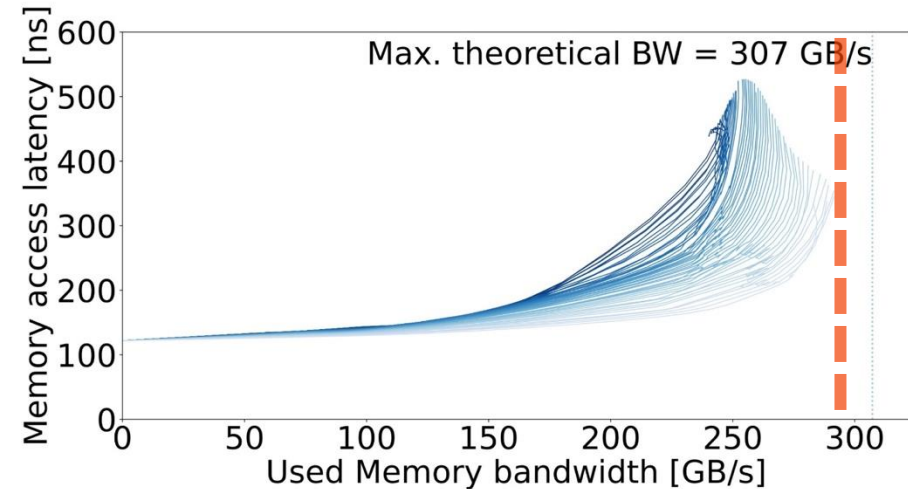
Mess Ramulator 2.0
(claimed by the Mess paper)



Mess Paper's Real System Configuration

16 DDR5 channels

Mess Real System
(claimed by the Mess paper)



The Mess paper configures Ramulator 2.0 with **half the channels** of a real system, and thus, results in **smaller used bandwidth**

Fundamental Configuration Errors: Smaller Memory Access Latency

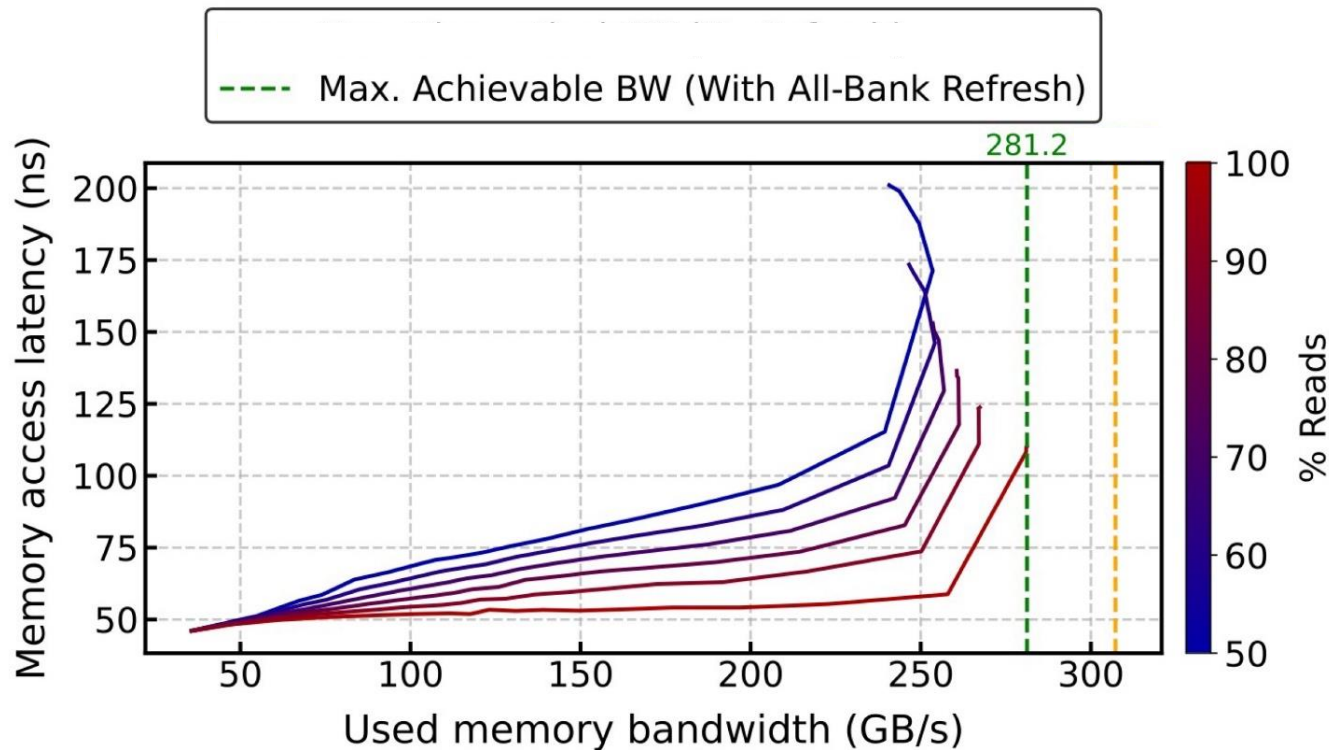
- The Mess paper's artifact hardcodes
 - the cache latency as **0 cycles**
 - the number of miss-status holding registers (MSHRs) per CPU as **1024**
- This leads to significantly smaller DRAM round-trip latency and queueing delays, thus **smaller memory access latency**

The Mess paper uses **unrealistically low cache latency configurations** resulting in the **low memory access latency** that it attributes to Ramulator 2.0

Our Evaluation Methodology

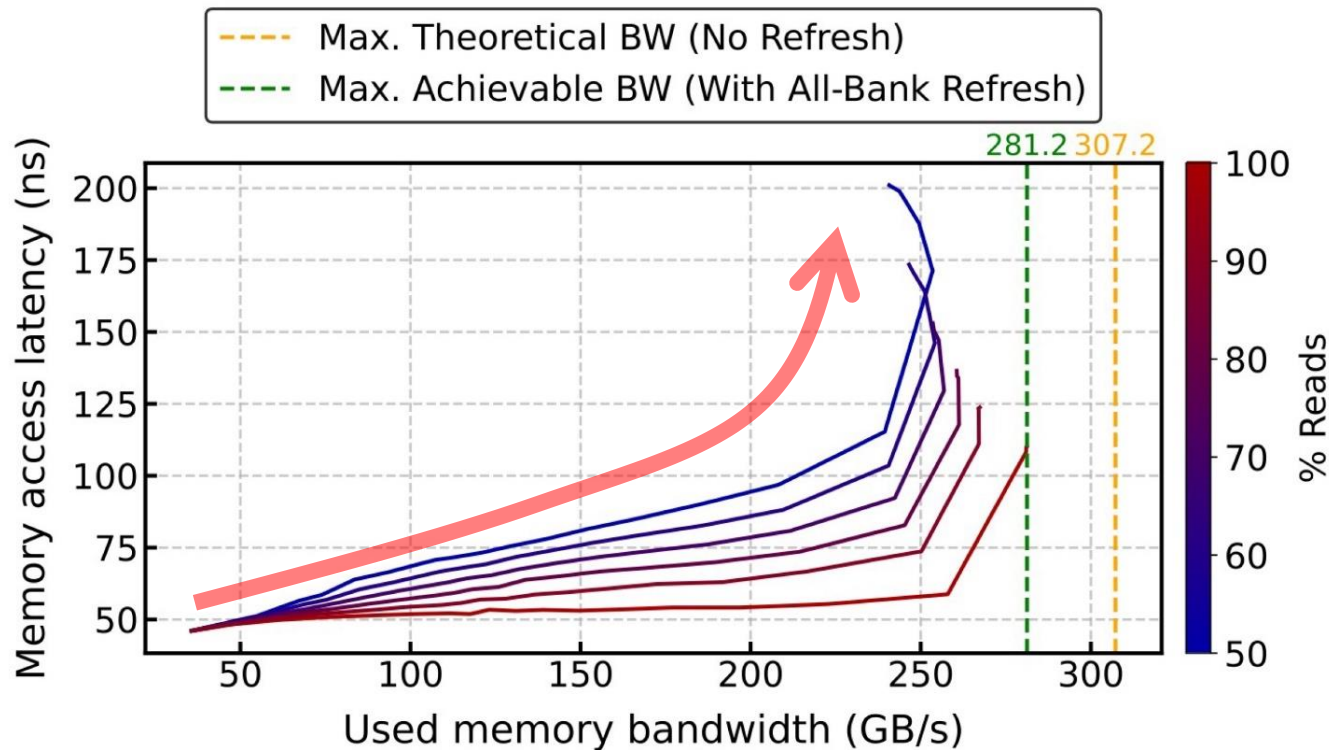
- We **independently evaluate the Mess benchmark** by executing the same access pattern
 - *Streaming* and *Pointer-Chasing* access patterns
 - Sweeping memory intensity
- We **correctly configure Ramulator 2.0**
 - Provide the same number of channels as the real system
 - Evaluate the memory access latency from the memory controller to isolate the DDR5 model's simulation accuracy

Our Key Results



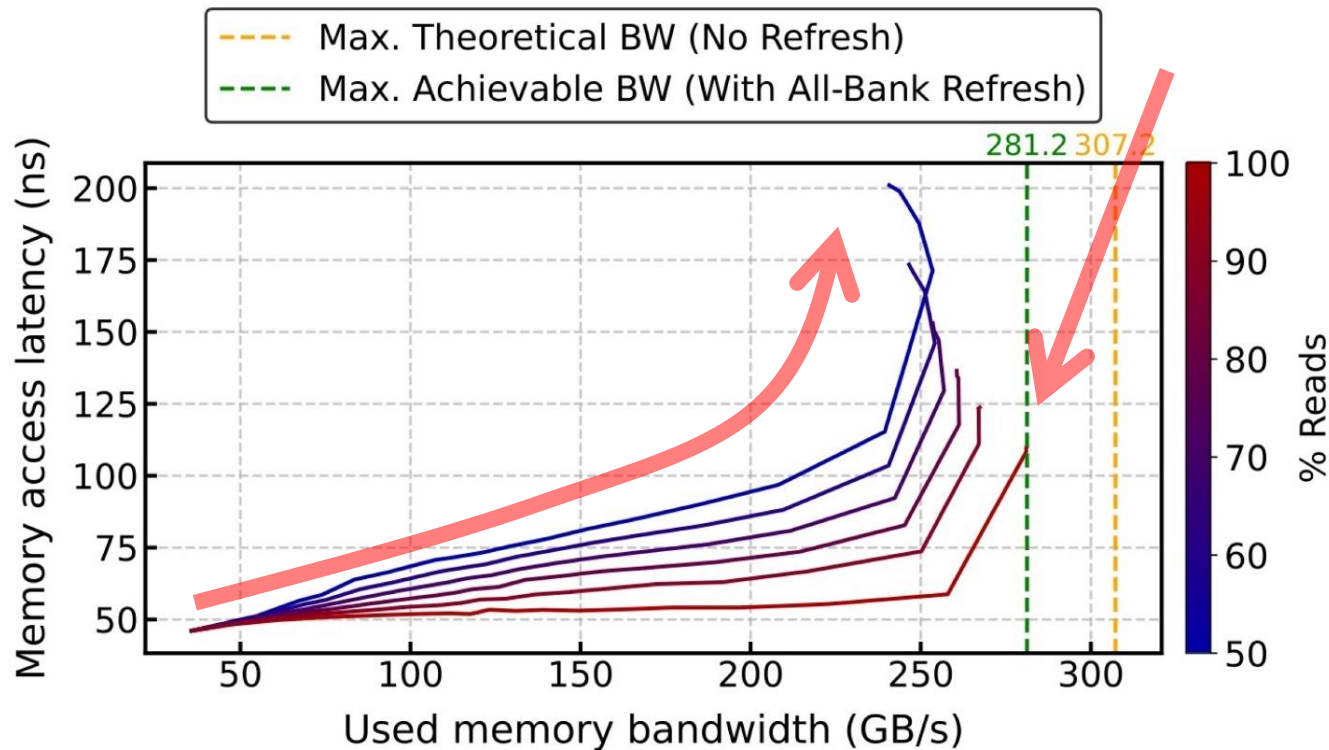
Ramulator 2.0's memory model achieves **high used memory bandwidth**, close to the **achievable bandwidth value**, as opposed to the claims of the Mess paper

Our Key Results



The **memory access latency** of Ramulator 2.0's model **increases** as the **used memory bandwidth** and the **write ratio** increases

Our Key Results



When **correctly configured**, Ramulator 2.0's simulated memory system performance **resembles real system characteristics**

Cleaning up the Mess

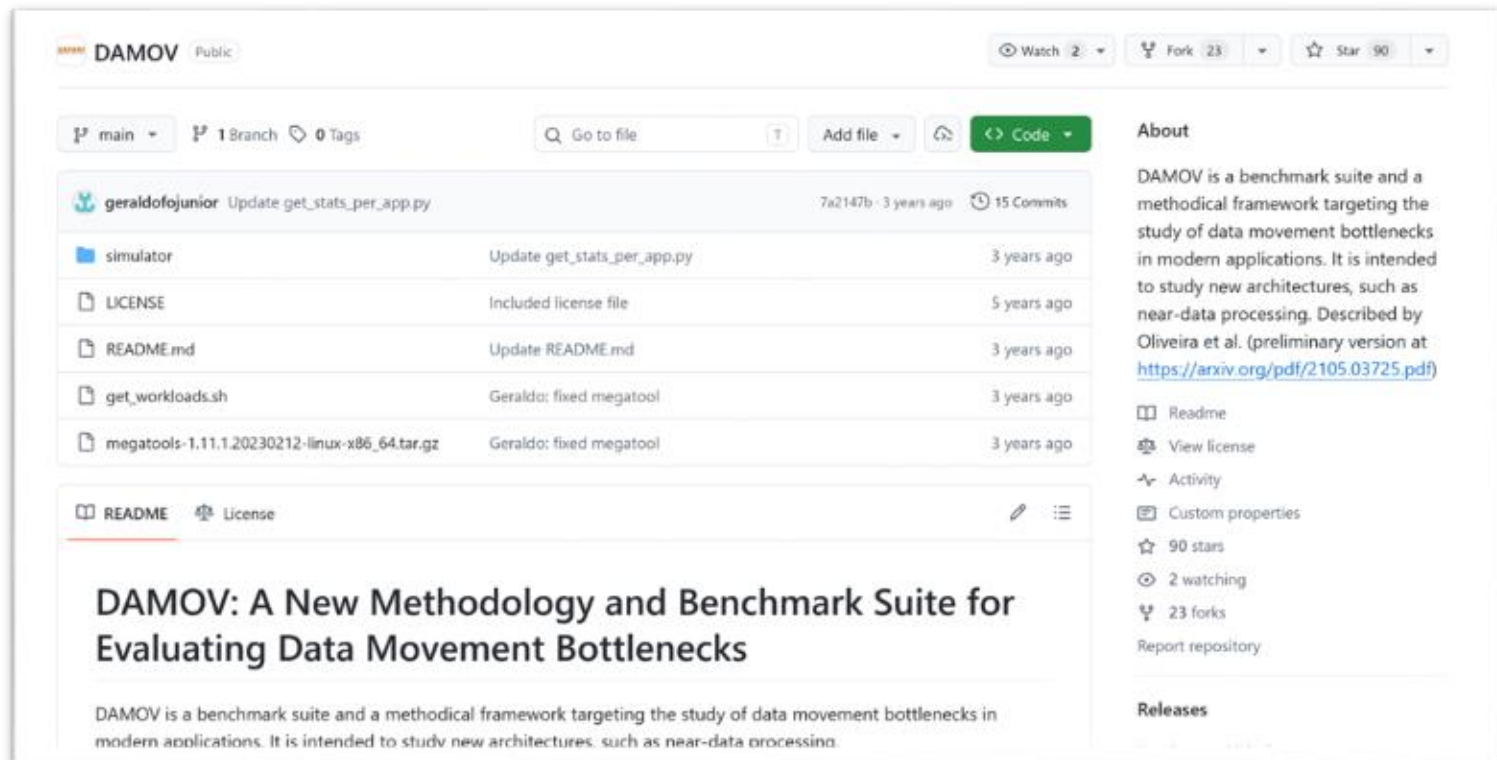
Analysis of the Evaluation of Ramulator 2.0
in the Mess Paper

**Analysis of the Evaluation of DAMOV
in the Mess Paper**

Ramulator 2.0 and DAMOV Best Practices

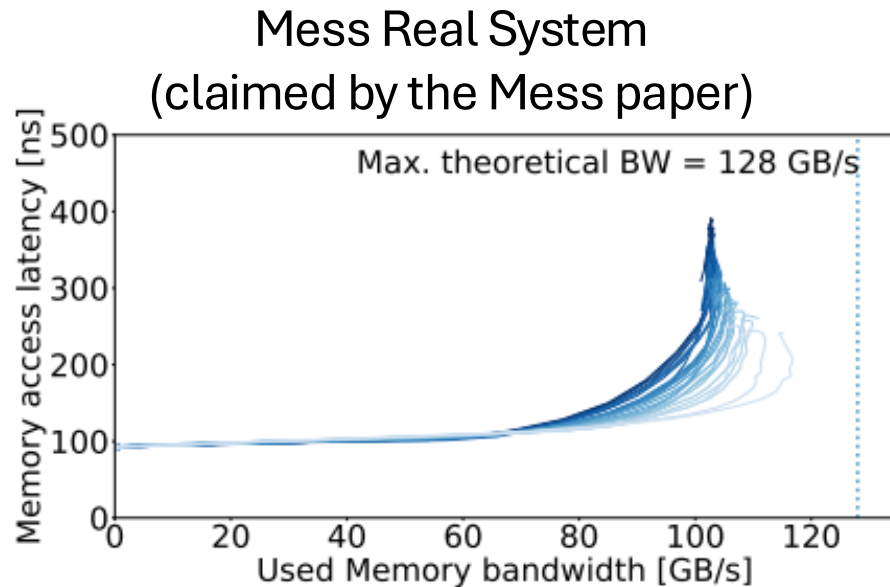
DAMOV: A Simulation Framework for PiM

- DAMOV integrates ZSim [Sanchez+, ISCA'13] with Ramulator [Kim+, CAL'15]
 - Enables evaluating emerging processing-in-memory architectures

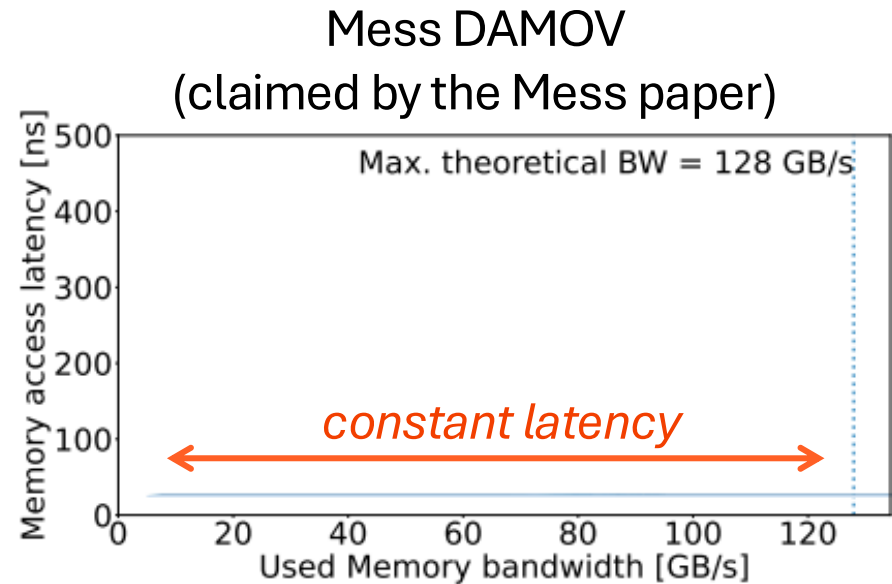


<https://github.com/CMU-SAFARI/DAMOV>

The Mess Paper's Evaluation of DAMOV



(a) Intel Skylake with 6×DDR4-2666.



(f) ZSim+Ramulator simulator.

DAMOV results in the Mess paper **use incorrect simulation statistics** that are unrelated to the simulated DRAM performance

Analysis of the Evaluation of DAMOV in the Mess Paper

Reproducibility Issues:

1

We could not reproduce any DAMOV results in the Mess paper, using the Mess paper's artifact due to:

- a) **Missing source code and configurations** of Ramulator
- b) Rigid **assertion failure** according to the ZSim configurations

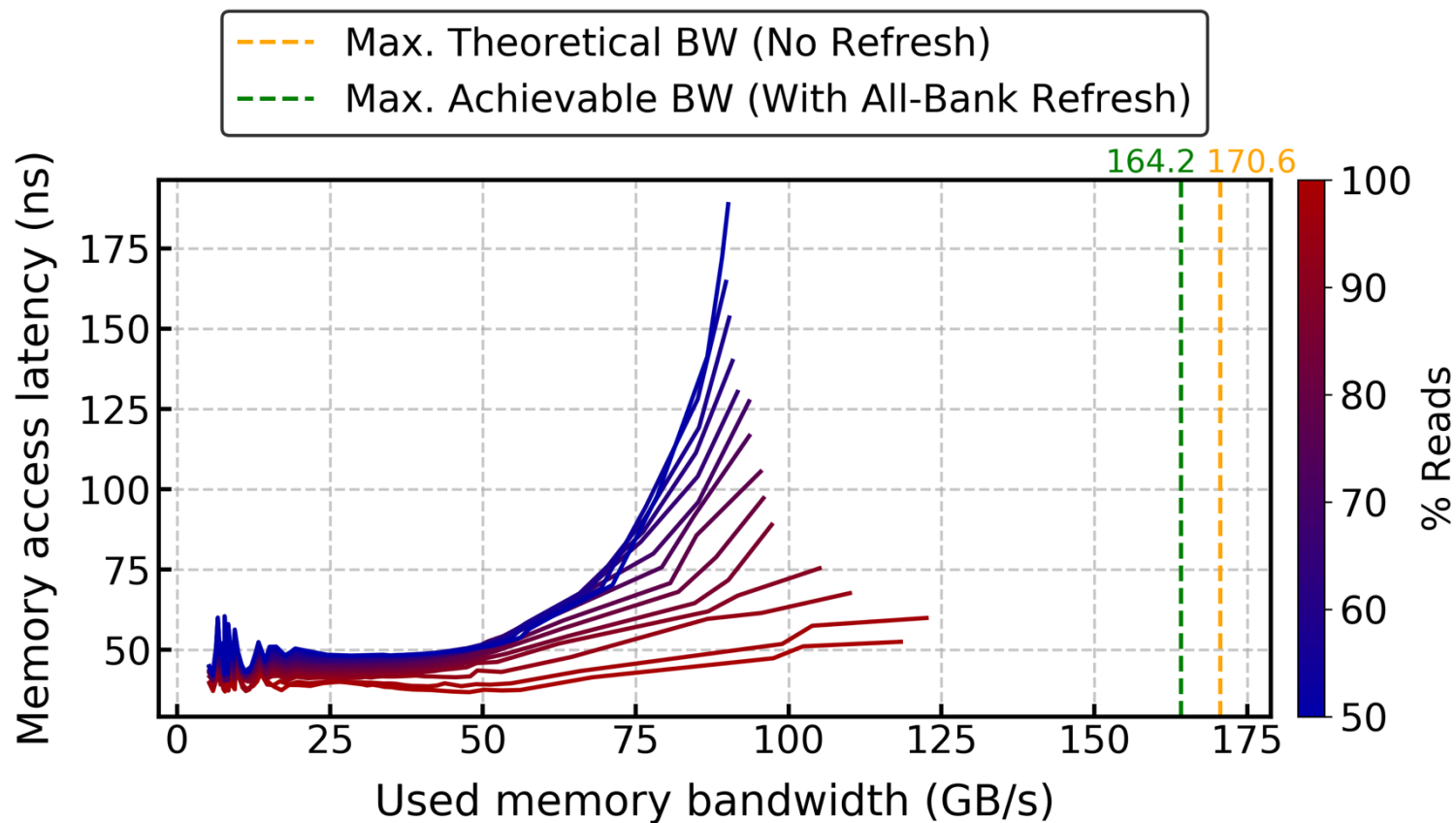
Technical Simulator Usage Errors:

2

- a) Uses two ZSim statistics for DRAM latency: **latGETnl** and **mGETs** from the L1-D cache
- b) These statistics are **not Ramulator statistics** and **does not reflect simulated DRAM latency** from Ramulator

We independently evaluate DAMOV by executing the Mess benchmark and adding missing pieces to the artifact in a best-guess manner and measure the used bandwidth statistics reported by Ramulator

Our Key DAMOV Results



When correct statistics are used, DAMOV's memory access latency is **not constant**, in contrast to the claims of the Mess paper

More in the Paper

Cleaning up the Mess: Re-Evaluating the Real-System Modeling Accuracy of Ramulator 2.0

F. Nisa Bostancı* Haocong Luo* Ataberk Olgun* Maria Makeenkova
Geraldo F. de Oliveira A. Giray Yağlıkçı Onur Mutlu

SAFARI Research Group
ETH Zürich

A MICRO 2024 best paper runner-up publication (the Mess paper [1]) with all three artifact badges awarded (including “Re-producible”) proposes a new benchmark to evaluate real and simulated memory system performance. The publication contends that Ramulator 2.0 [2] and DAMOV [3] (ZSim+Ramulator) (along with other existing memory system simulators) “poorly resemble the actual system performance” and asserts that their simulator is better.

add sanity checks to aid users and developers of simulation tools.

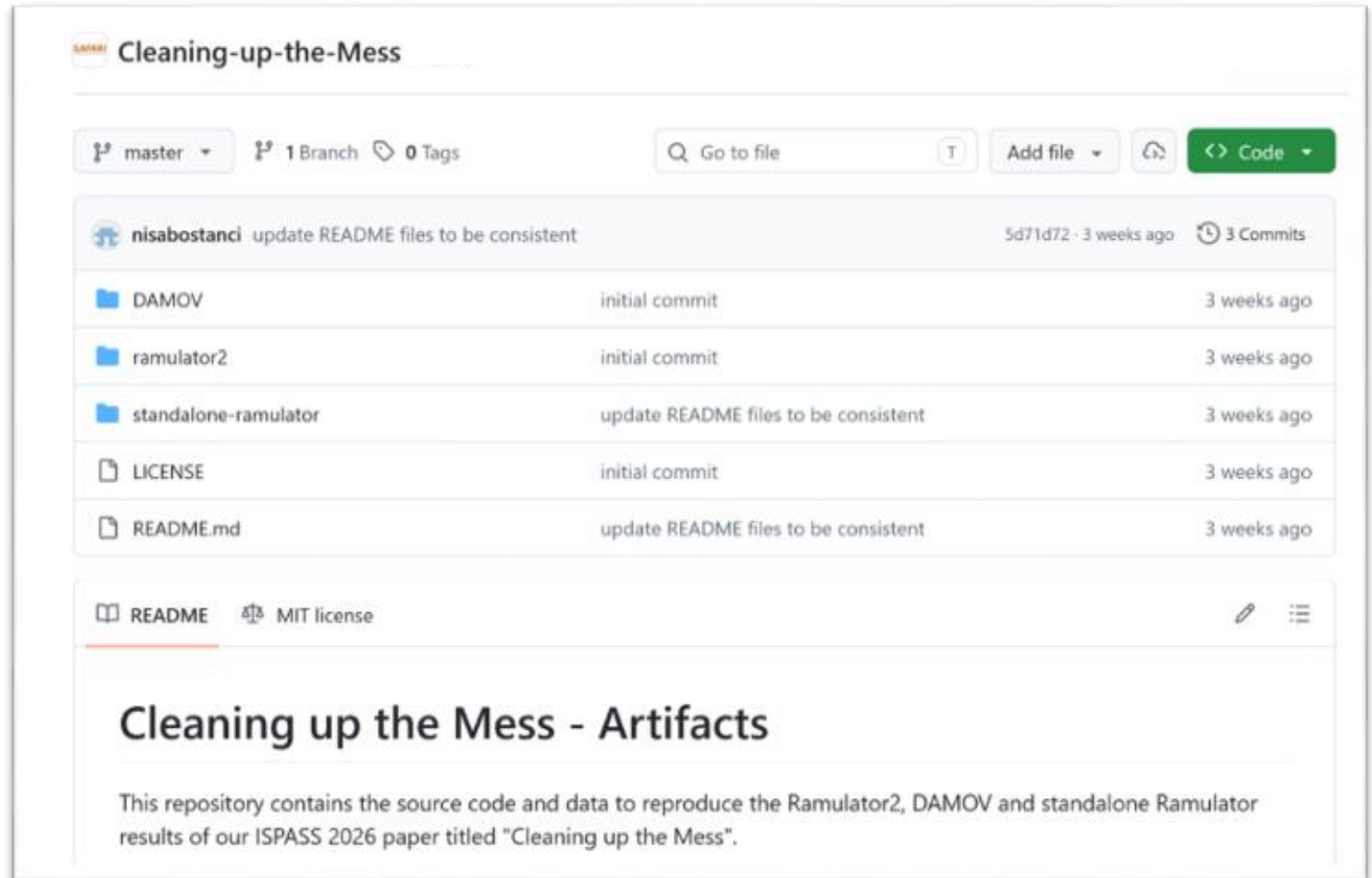
1. Introduction

Cycle-level DRAM simulators enable research focused on improving the performance, efficiency, and robustness (including security, safety, reliability) of DRAM-based memory systems by providing accurate and flexible models for DRAM and memory controller operations. Ramulator 2.0 [2, 6] (which builds

** Equal contribution*

<https://arxiv.org/pdf/2510.15744>

Cleaning up the Mess is Open Source



<https://github.com/CMU-SAFARI/Cleaning-up-the-Mess>

Cleaning up the Mess

Analysis of the Evaluation of Ramulator 2.0
in the Mess Paper

Analysis of the Evaluation of DAMOV
in the Mess Paper

Ramulator 2.0 and DAMOV Best Practices

Ramulator 2.0 and DAMOV Best Practices

1

Be familiarized with JEDEC standard documents

instead of relying on information other than the official standard or the simulator documentation

2

Understand Ramulator 2.0's

memory organization and configurations,

which are documented in the code and in the repository

3

**Understand simulation methodology and
simulator implementations**

sufficiently to use the correct statistics

4

Communicate with the simulator developers

or user community when unexpected results are observed
to ensure correct simulator usage

Conclusion

A recent paper, "**A Mess of Memory System Benchmarking, Simulation and Application Profiling**", MICRO'24 makes **strong negative claims** about Ramulator 2.0 and DAMOV and shows **unexpected results**

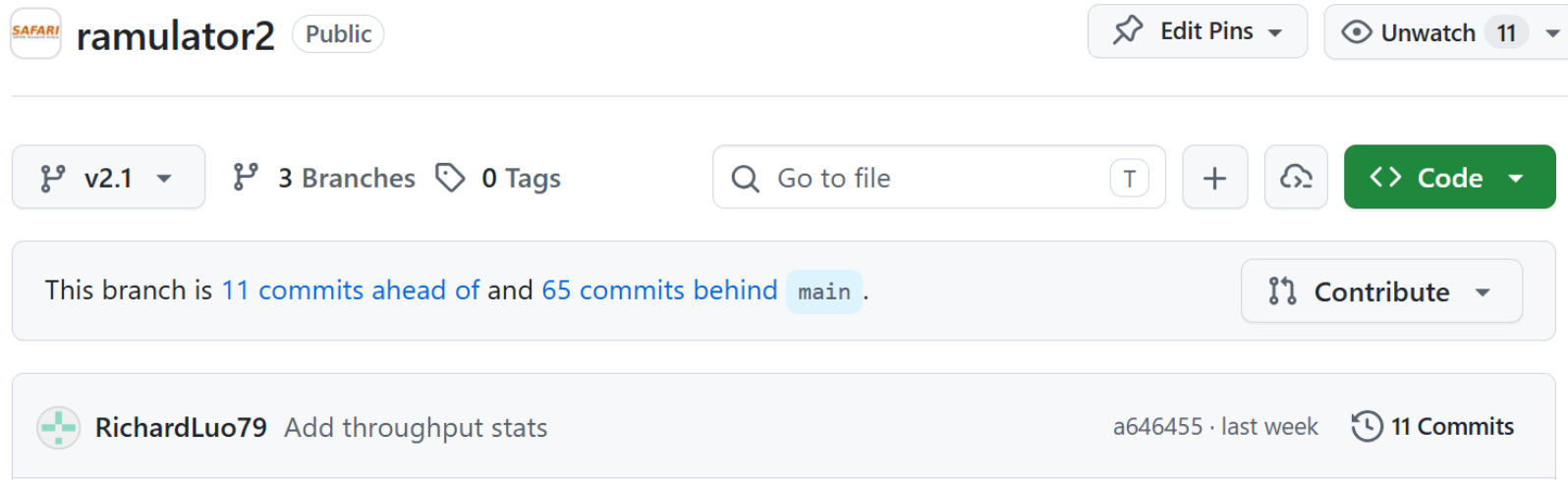
We demonstrate that these claims and results are **incorrect** and are due to **configuration and simulator usage errors** made in the Mess paper

We strongly encourage the computer architecture community to consider our corrections to the Ramulator 2.0 and DAMOV results of the Mess paper to **prevent the propagation of inaccurate and misleading results**

To facilitate **transparency and reproducibility**, we open source all our code and results: <https://github.com/CMU-SAFARI/Cleaning-up-the-Mess>

Ramulator 2.1

- Major overhaul of Ramulator 2.0



- Currently available as a pre-release in the v2.1 branch
<https://github.com/CMU-SAFARI/ramulator2/tree/v2.1>
- Welcomes your use, contribution, and feedback!
 - **We will be making v2.1 the default branch soon**

Join us at ISCA 2026!

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

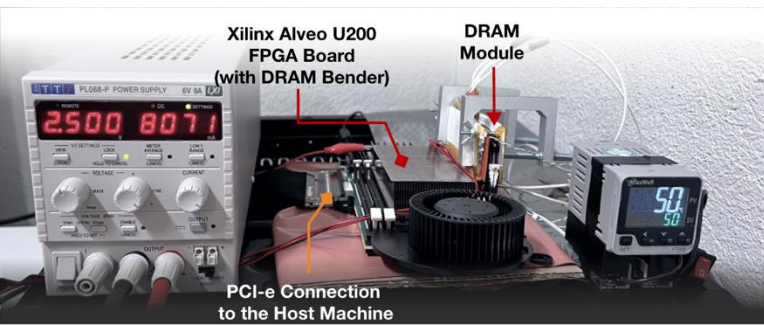


R&DB:

**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/>



Join us at ICS 2026!

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

Included in
Proceedings!

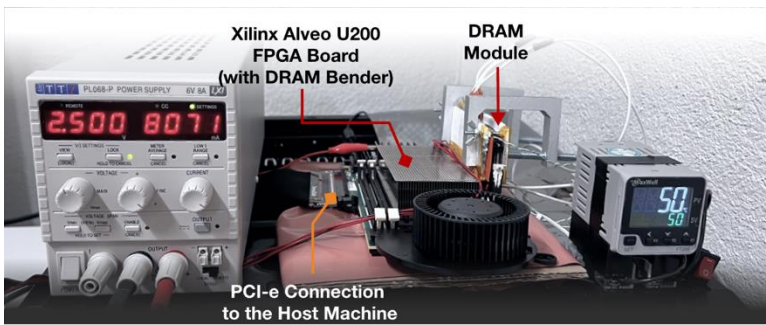
ICS 2026

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/>





Cleaning up the Mess

Re-Evaluating the Real-System Modeling Accuracy of Ramulator 2.0

F. Nisa Bostancı*

Haocong Luo*

Ataberk Olgun*

Maria Makeenkova

Geraldo F. de Oliveira

A. Giray Yağlıkçı

Onur Mutlu

* Equal Contribution

SAFARI

ETH zürich



Paper



GitHub



Cleaning up the Mess

Re-Evaluating the Real-System Modeling Accuracy of Ramulator 2.0

F. Nisa Bostancı*

Haocong Luo*

Ataberk Olgun*

Maria Makeenkova

Geraldo F. de Oliveira

A. Giray Yağlıkçı

Onur Mutlu

* Equal Contribution

SAFARI

ETH zürich



Cleaning up the Mess

Re-Evaluating the Real-System Modeling Accuracy of Ramulator 2.0

Backup Slides

F. Nisa Bostancı*

Haocong Luo*

Ataberk Olgun*

Maria Makeenkova

Geraldo F. de Oliveira

A. Giray Yağlıkçı

Onur Mutlu

* Equal Contribution

SAFARI

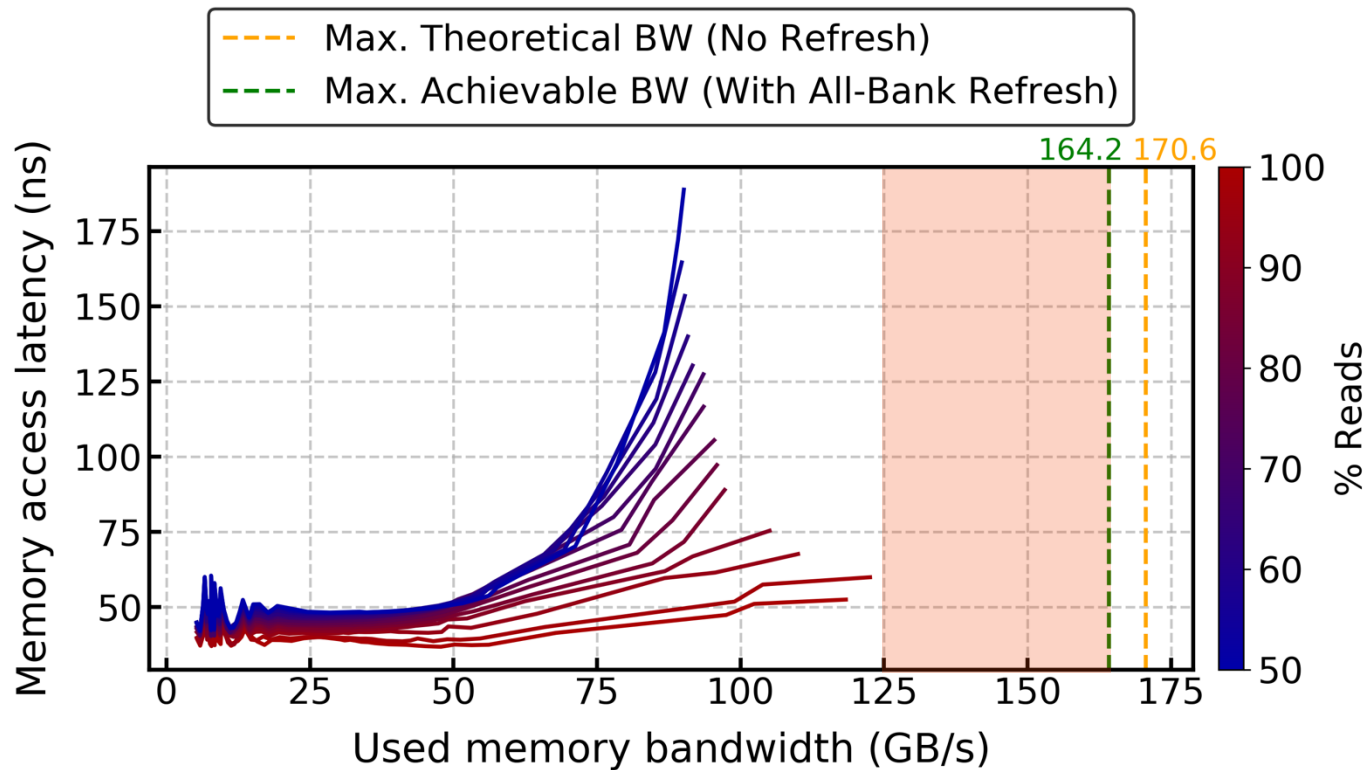
ETH zürich

Maximum Achievable Bandwidth

$$BW_{achievable} = BW_{theoretical} \cdot \left(1 - \frac{REF_{penalty}}{tREFI} \right)$$

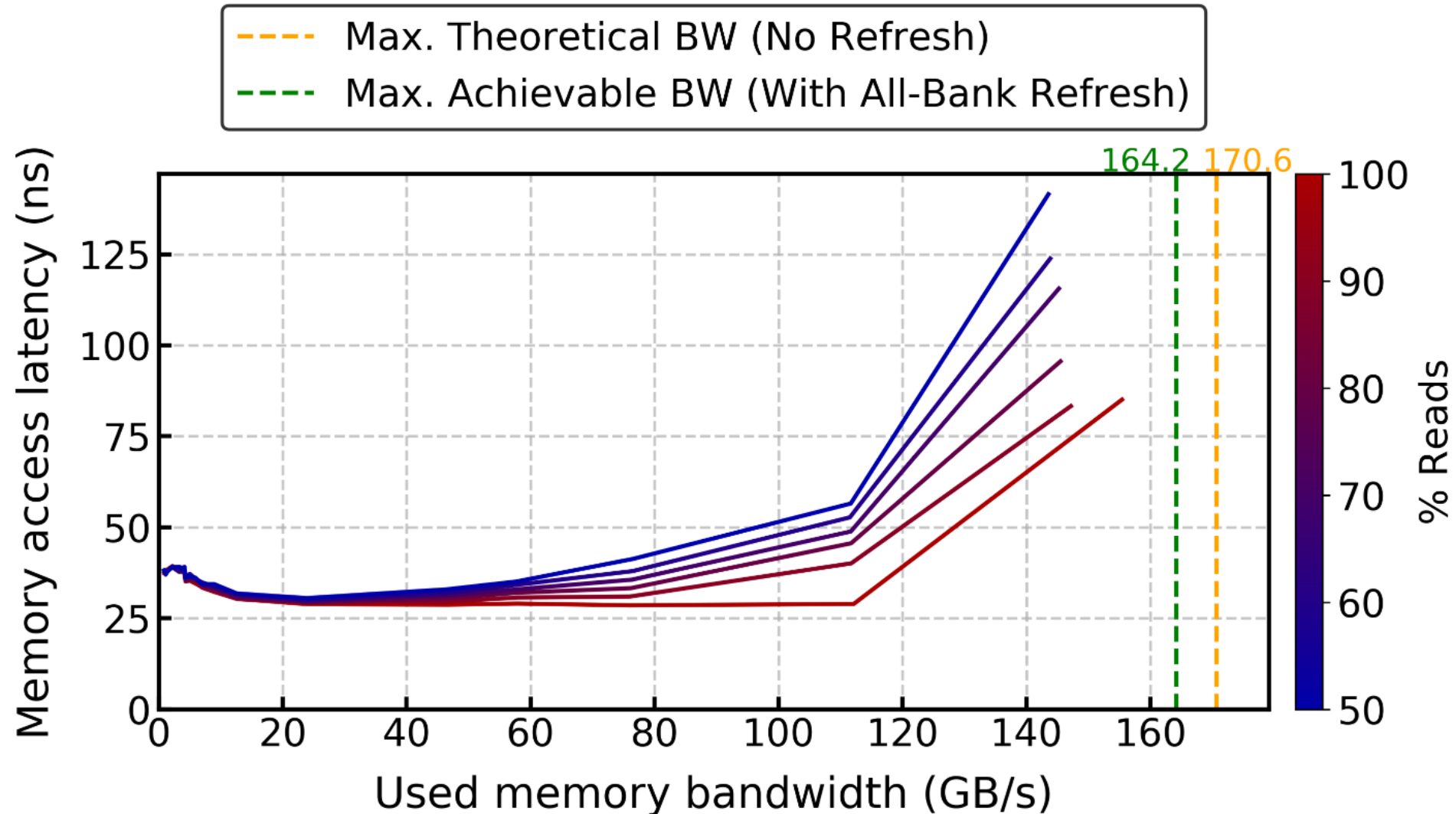
$$REF_{penalty} = tRTP + tRP + tRFC + tRCD$$

Further Evaluation of DAMOV Results

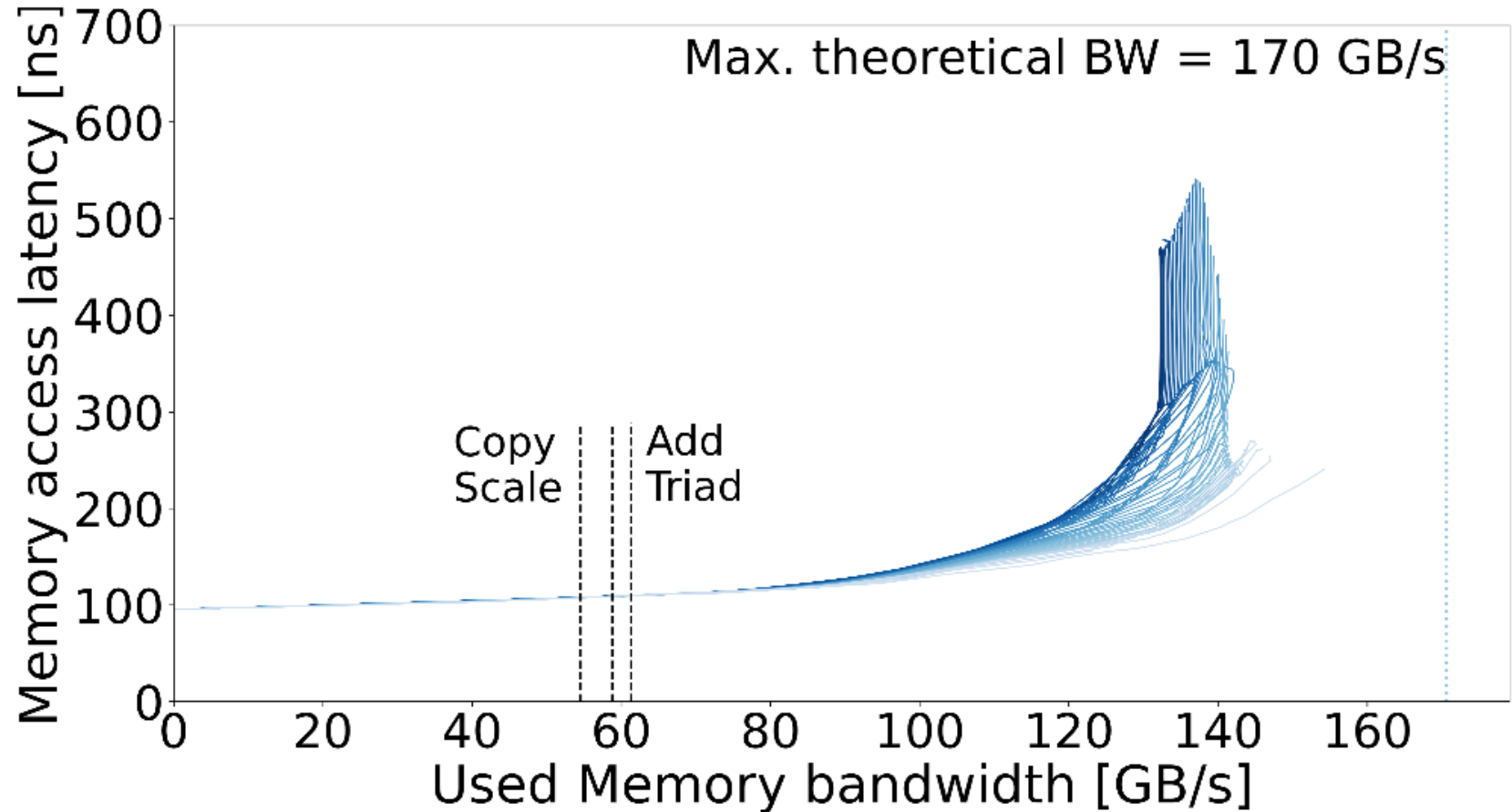


- The bandwidth gap can be caused by a combination of workload characteristics, system configuration, and the integration of the core and memory models
 - E.g., DRAM address mapping, core model optimizations

Ramulator (v1) – Mess Benchmark

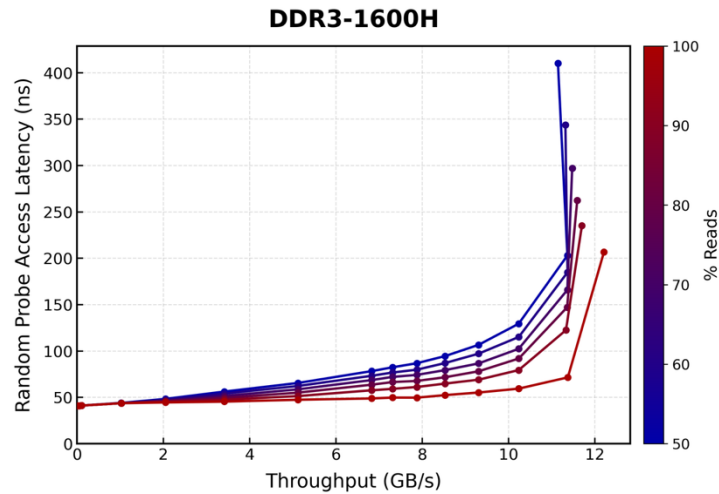
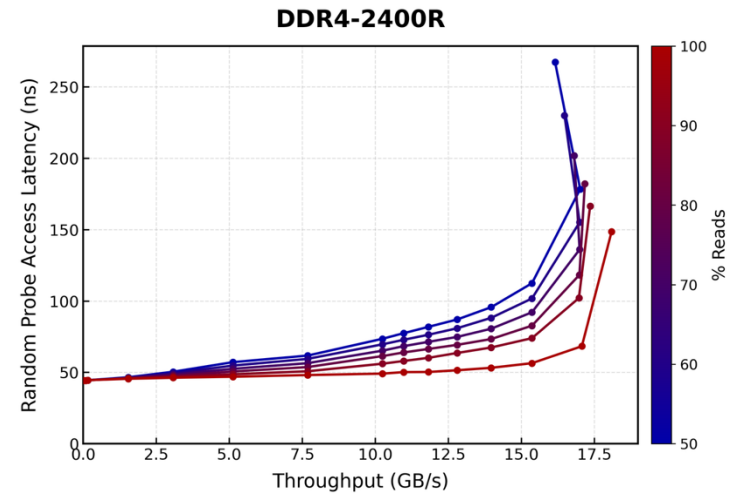
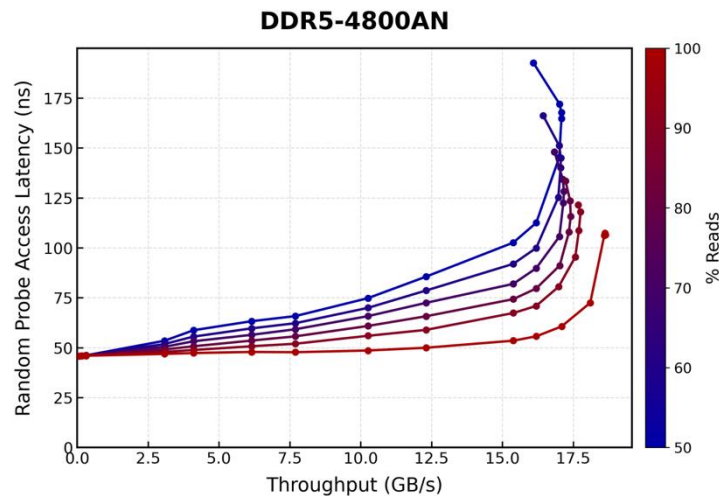
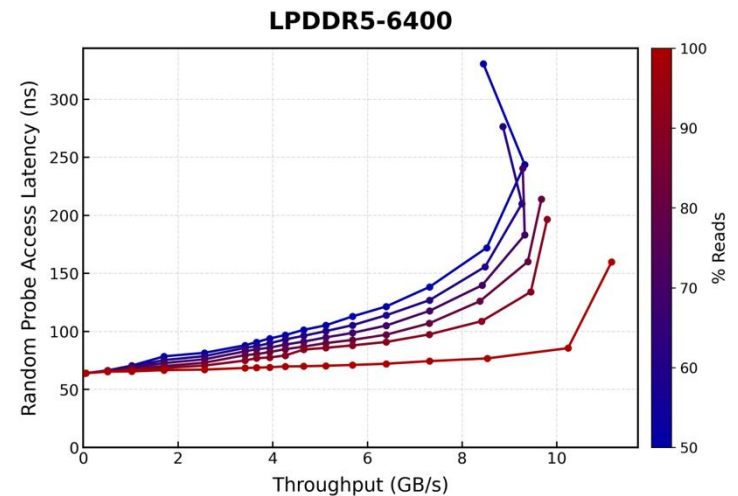


Another DDR4 Real System Result from the Mess Paper



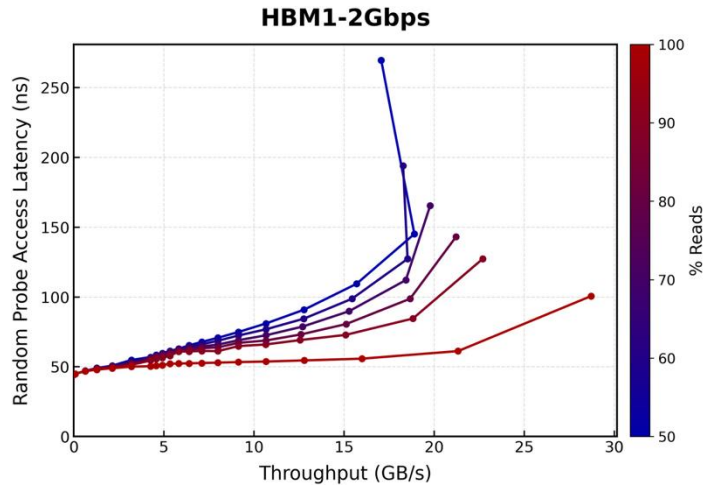
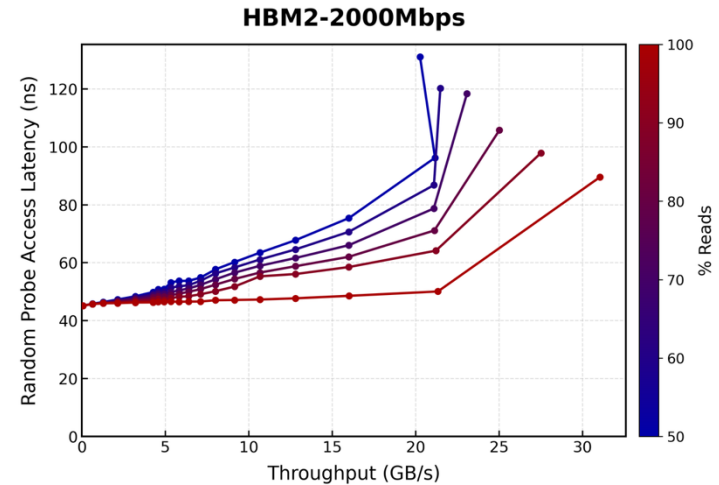
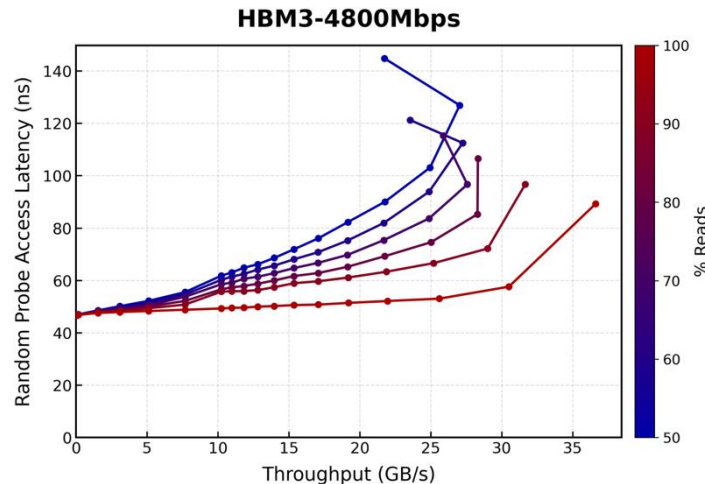
(d) IBM Power 9 with $8 \times$ DDR4-2666.

Ramulator 2.1 DRAM Standards Lat-Tp Plots



Ramulator 2.1 DRAM Standards Lat-Tp Plots

* HBM timings are guestimates



Ramulator 2.1

- **Major overhaul of Ramulator 2.0** that includes
 - Aggregated bug fixes and code quality improvements
 - Better support of newer DRAM and controller features, e.g.,
 - Row-Col command dual issue controller for HBM2+
 - WCK2CK sync and expiry tracking and ACT-1/ACT-2 split with tAAD deadline aware scheduling for LPDDR5
 - Significantly improved ease-of-use, configuration, and extension
 - More comprehensive test suite
 - More comprehensive documentation
- Currently available as a pre-release in the v2.1 branch
<https://github.com/CMU-SAFARI/ramulator2/tree/v2.1>
- Welcomes your use, contribution, and feedback!
 - **We will be making v2.1 the default branch soon**

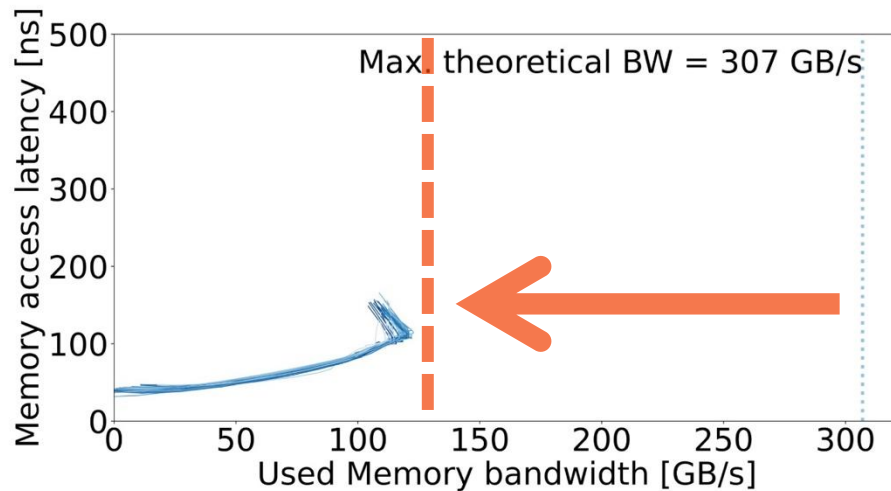
Fundamental Configuration Errors: Smaller Used Bandwidth

The Mess paper's evaluation:

Ramulator 2.0

8 DDR5 channels

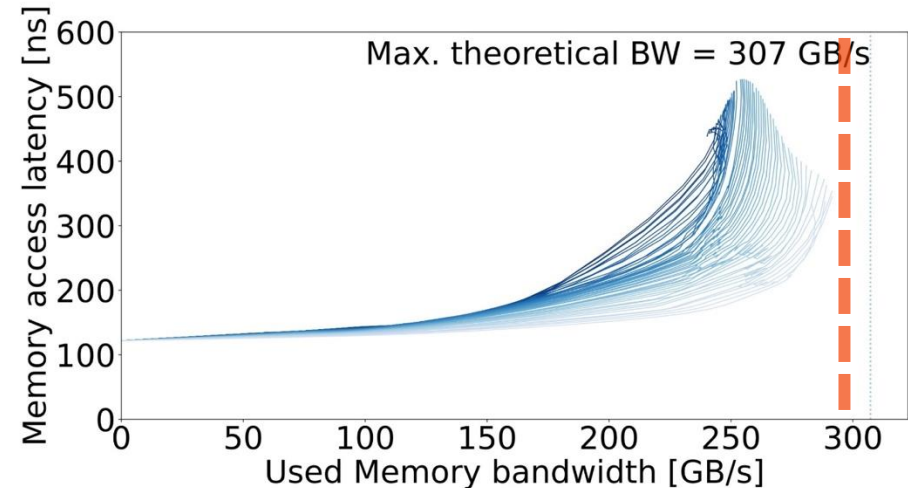
Mess Ramulator 2.0
(claimed by the Mess paper)



Real System

16 DDR5 channels

Mess Real System
(claimed by the Mess paper)



The Mess paper **incorrectly assumes** a DDR5 channel in Ramulator 2.0 is **64-bit**, instead of **the actual width of 32-bit** that is **consistent with JEDEC DDR5 standards** (JESD79-5)

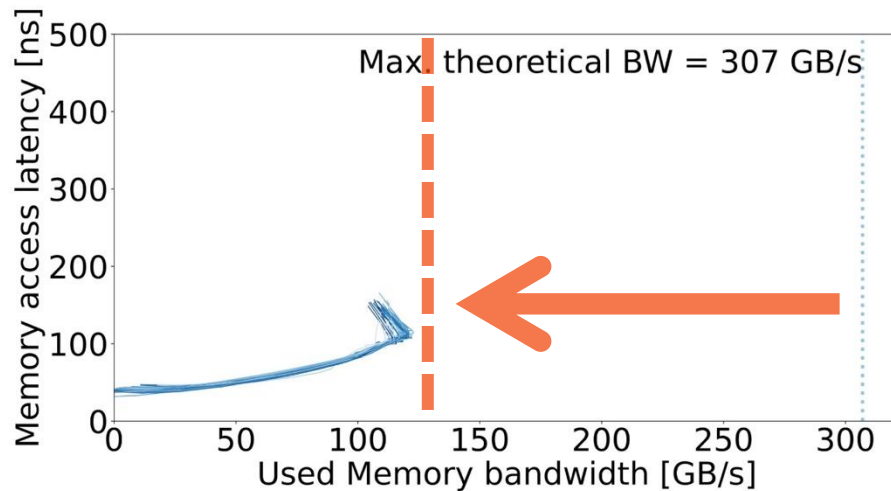
Fundamental Configuration Errors: Smaller Used Bandwidth

The Mess paper's evaluation:

Ramulator 2.0

8 DDR5 channels

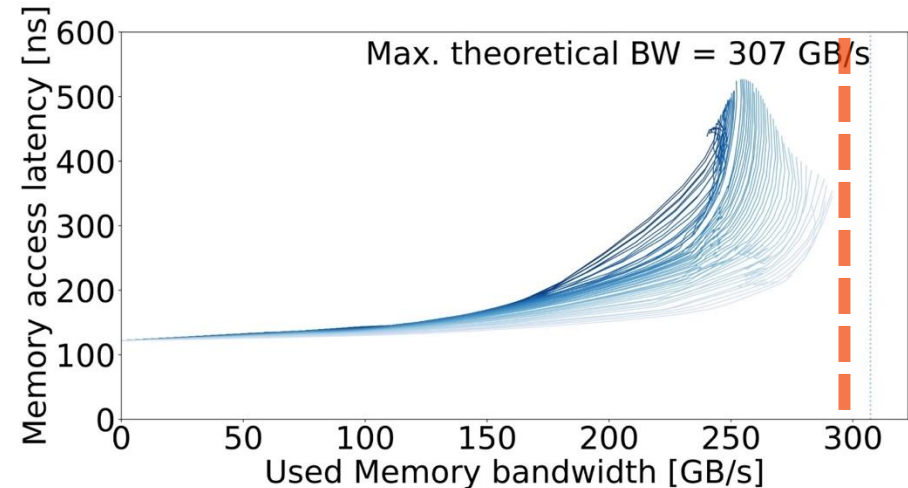
Mess Ramulator 2.0
(claimed by the Mess paper)



Real System

16 DDR5 channels

Mess Real System
(claimed by the Mess paper)



The Mess paper configures Ramulator 2.0 with **half the channels** of a real system, and thus, results in **smaller used bandwidth**

DDR5 Channel Terminology

JEDEC Standard No. 79-5B_v1.30
Page 459

14 DDR5 Module Rank and Channel Timings

14.1 Module Rank and Channel Limitations for DDR5 DIMMs

To achieve efficient module power supply design for JEDEC-standard DDR5 DIMMs, minimum timings as well as limitations in the number of DRAMs are provided for Refresh, and Write operations occurring on a single module. As well, since these modules are organized as two independent 36-bit or 40-bit channels (32 bits for non-ECC DIMMs), additional restrictions apply in order to limit localized power delivery noise on the module. To provide best performance, the different channels may initiate commands on the same cycle provided the rank to rank timings are met, the maximum number of DRAMs in a given activity is not exceeded, and the applicable component timings shown elsewhere in this specification are met. The timing and operational relationships for DDR5 DIMMs are shown in Table 340.

[ramulator2](#) / [src](#) / [dram](#) / [impl](#) / [DDR5.cpp](#)

Code

Blame

814 lines (687 loc) · 39.3 KB

```
6      class DDR5 : public IDRAM, public Implementation {  
  
314          private:  
315  ✓      void set_organization() {  
316          // Channel width  
317          m_channel_width = param_group("org").param<int>("channel_width").default_val(32);  
318
```

Ramulator 2.0 Simulation Configurations

Table 1: Simulated System Configuration

Frontend:	Mess Request Generator for Ramulator 2.0 [2]
DRAM:	DDR5-4800AN [13]; 16 channels (32-bit each), 1 rank, 8 bank groups, 4 banks
Timing Parameters:	t_{RCD} : 14.166 ns, t_{RAS} : 32.000 ns, t_{RP} : 14.166 ns, t_{RTP} : 7.5 ns, t_{CCD_S} : 8 nCK, t_{RFC} : 295 ns, t_{REFI} : 3.9 μ s
Memory System:	FR-FCFS request scheduling policy [18, 19], All-bank refresh, Read queue size: 32, Write queue size: 32

DAMOV Simulation Configurations

Table 2: Simulated System Configuration in DAMOV

Processor:	24 cores @ 2.4 GHz
Cache Hierarchy:	L1i and L1d: 8-way 64 KB, L2: 16-way 1 MB, LLC: 11-way 33 MBs.
DRAM:	DDR4-2666T [22]; 8 channels (64-bit each), 1 rank, 4 bank groups, 4 banks
Timing Parameters:	t_{RCD} : 12.75 ns, t_{RAS} : 32.25 ns, t_{RP} : 12.75 ns, t_{RTP} : 7.5 ns, t_{CCD_S} : 4 nCK, t_{RFC} : 261 ns, t_{REFI} : 7.8 μ s
Memory System:	FR-FCFS request scheduling policy [18, 19], All-bank refresh, Read queue size: 32, Write queue size: 32