Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

A Mess of Memory System Benchmarking, Simulation and Application Profiling

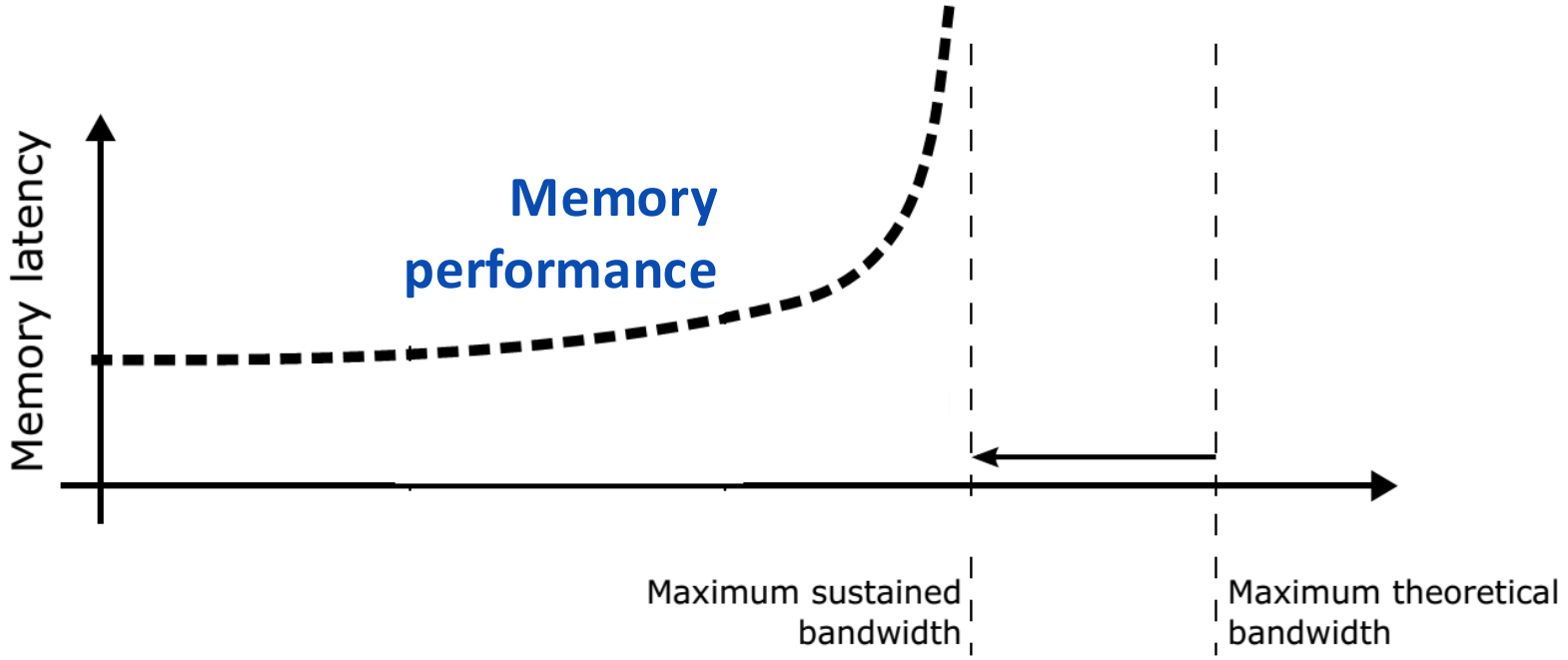
Petar Radojkovic

Barcelona Supercomputing Center

57th IEEE/ACM International Symposium on Microarchitecture
Austin, Texas

4th of November 2024.

Memory latency = func(Used memory bandwidth)

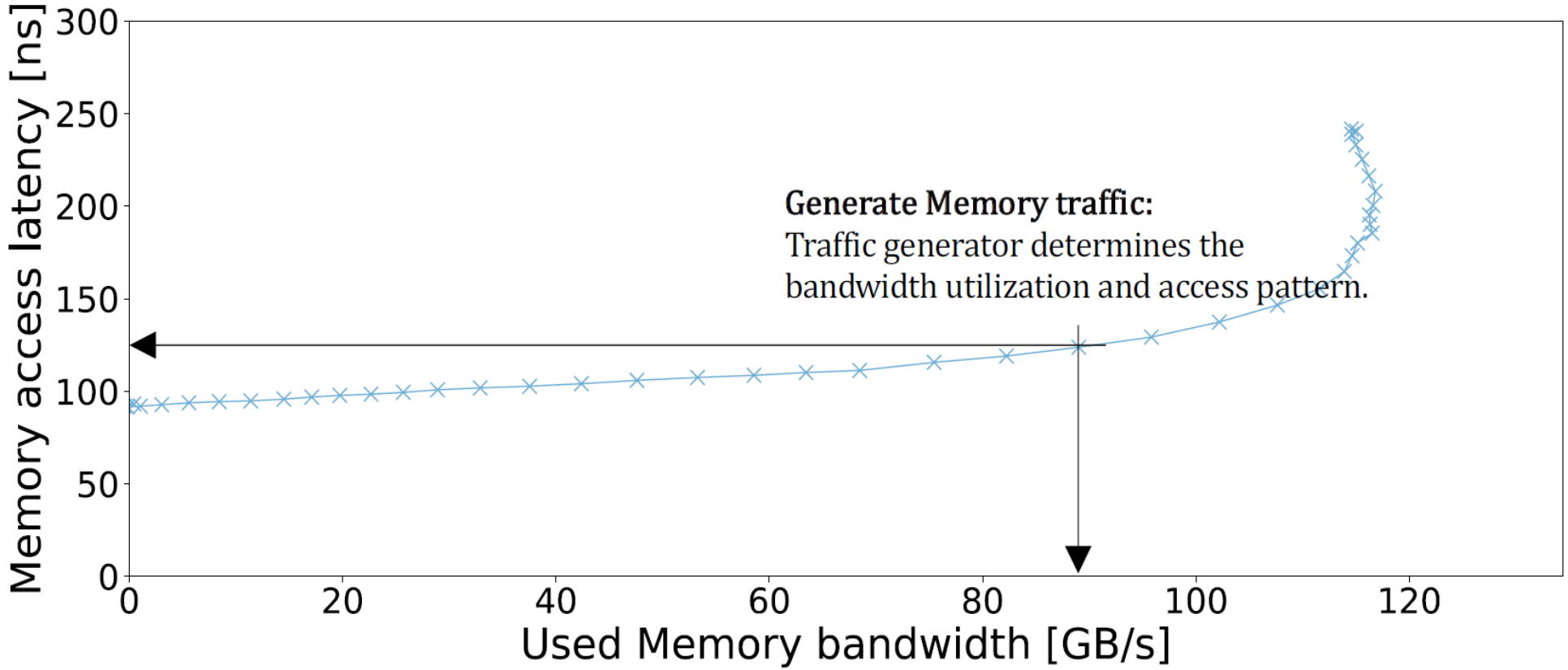


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

B. L. Jacob. The memory system: *You can't avoid it, you can't ignore it, you can't fake it.* Synthesis Lectures on Computer Architecture, 2009.

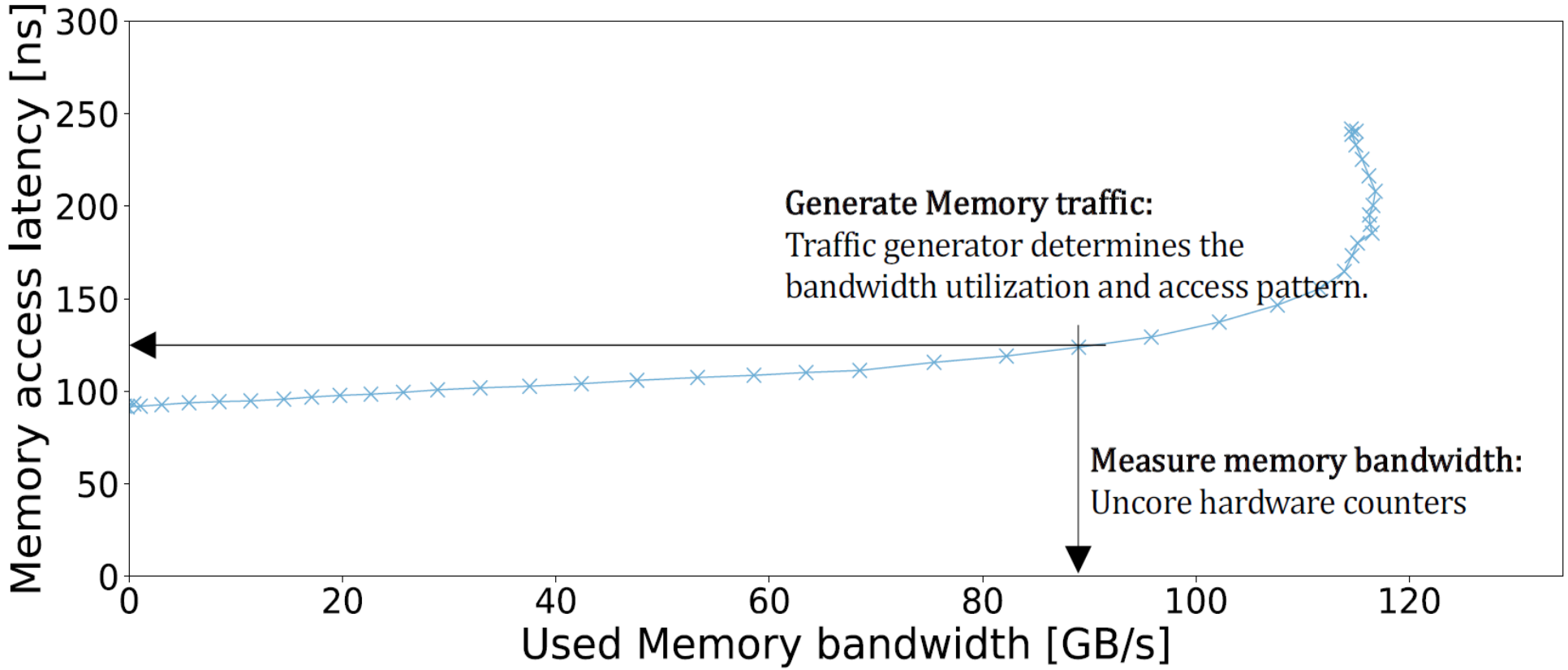
Let’s make a benchmark that will “plot” the curve



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

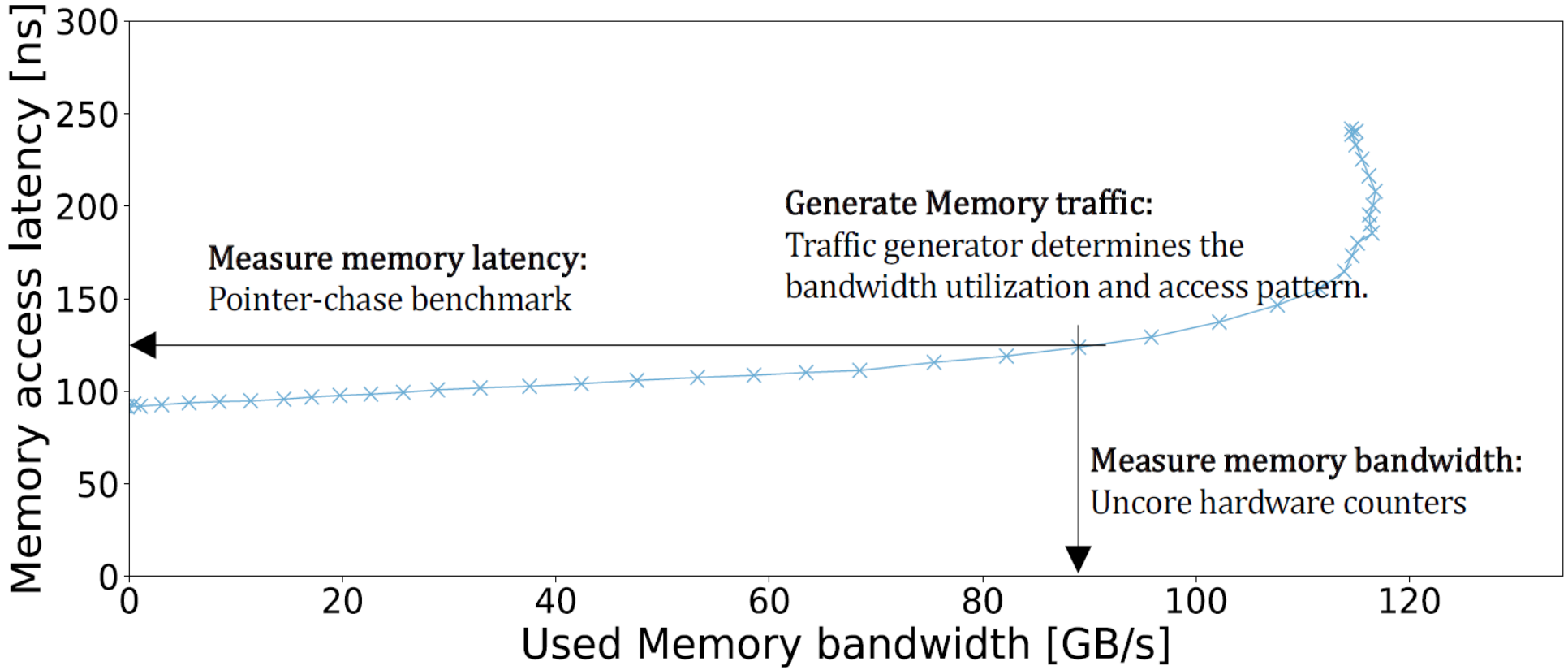
Let’s make a benchmark that will “plot” the curve



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

Let’s make a benchmark that will “plot” the curve

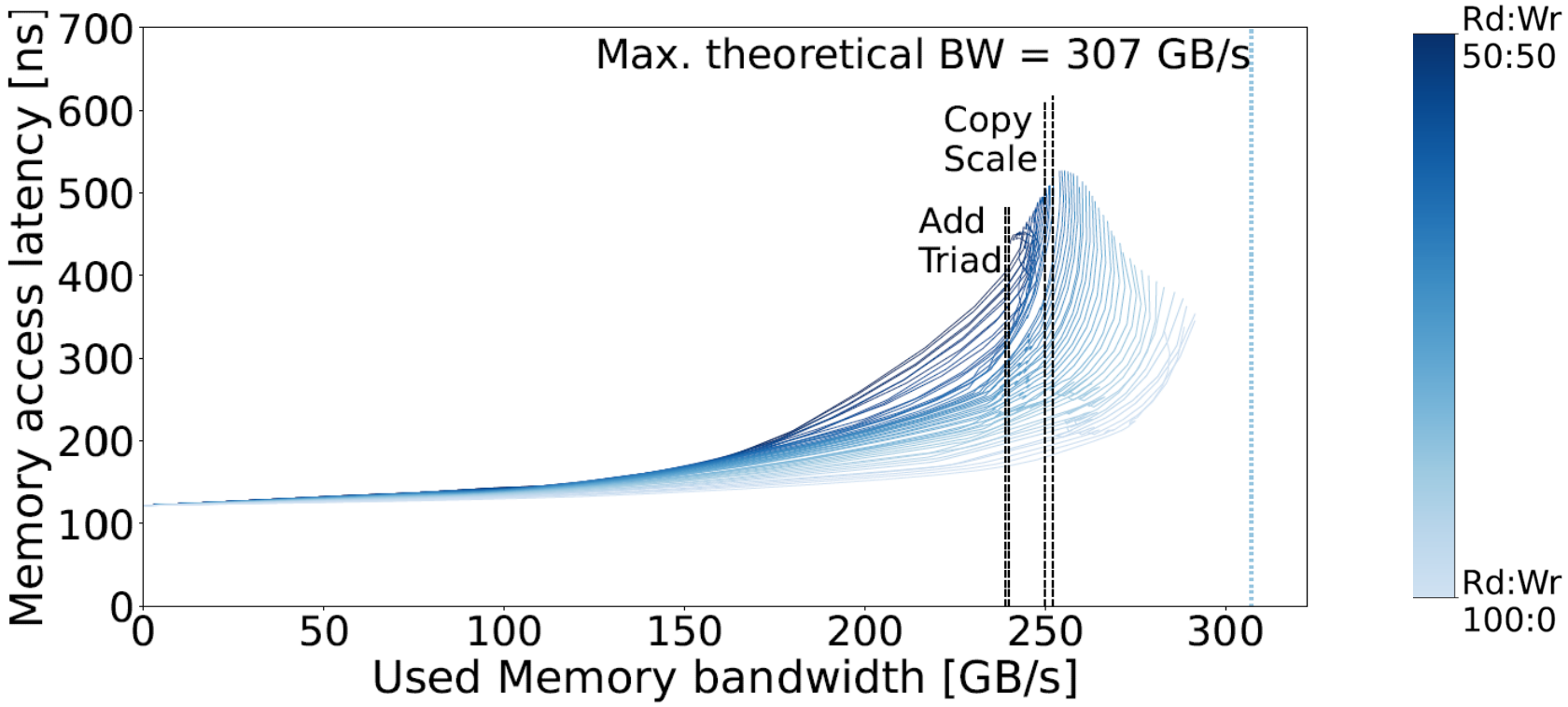


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

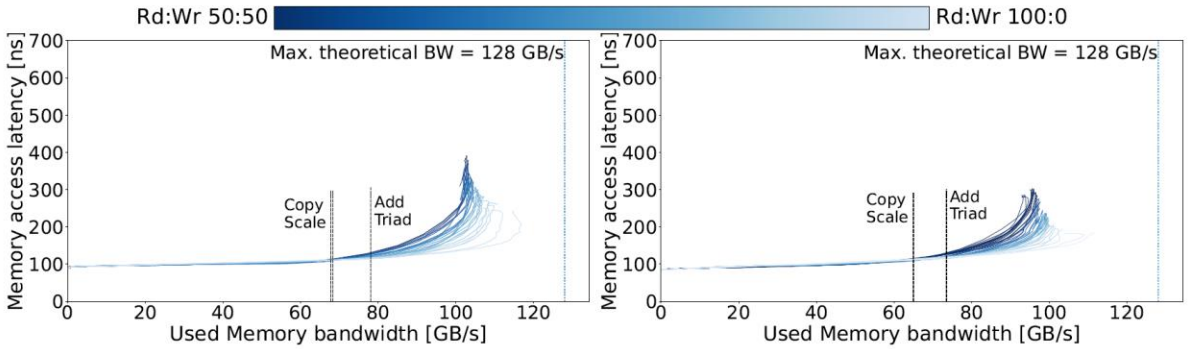
Memory stress (Mess) benchmark

- Connecting the dots



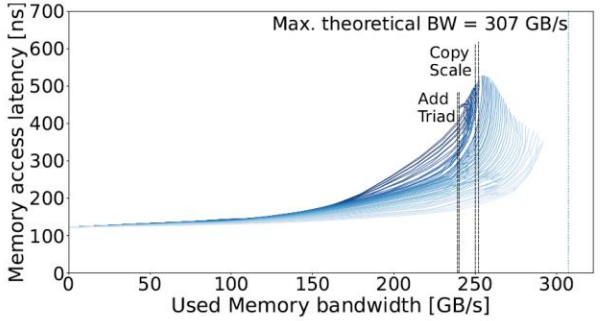
Amazon Graviton3 with 8× DDR5-4800.

Mess benchmark: Actual platforms

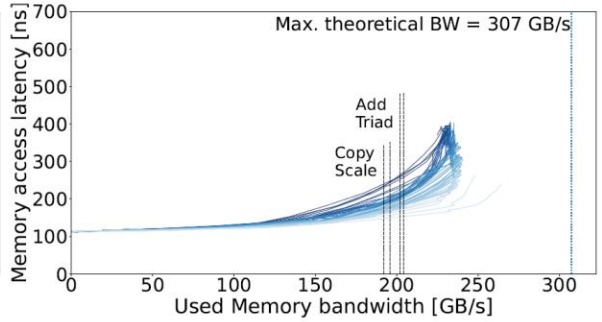


Intel Skylake with 6× DDR4-2666.

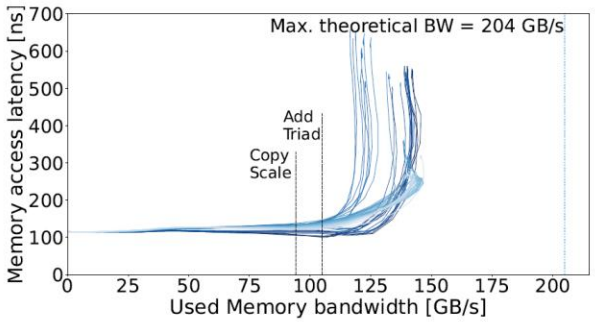
Intel Cascade Lake with 6× DDR4-2666.



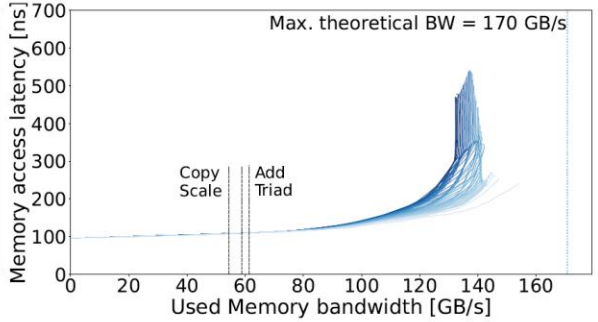
Amazon Graviton3 with 8× DDR5-4800.



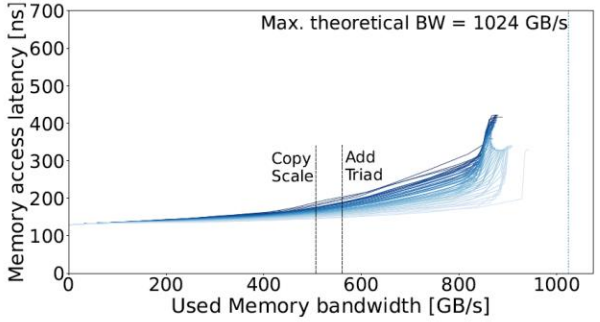
Intel Sapphire Rapids with 8× DDR5-4800.



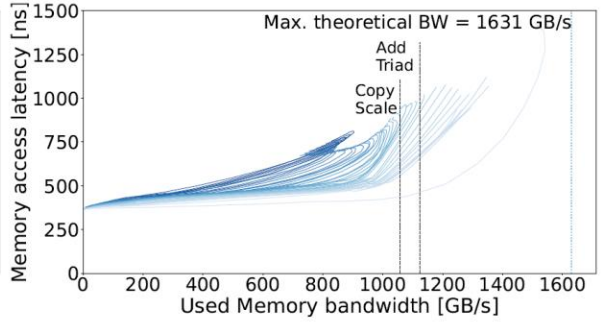
AMD Zen2 with 8× DDR4-3200



IBM Power 9 with 8× DDR4-2666.

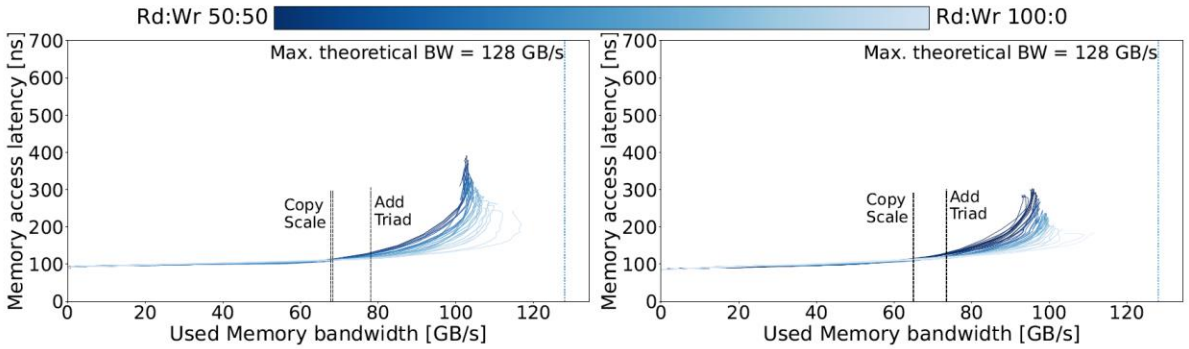


Fujitsu A64FX with 4× HBM2.



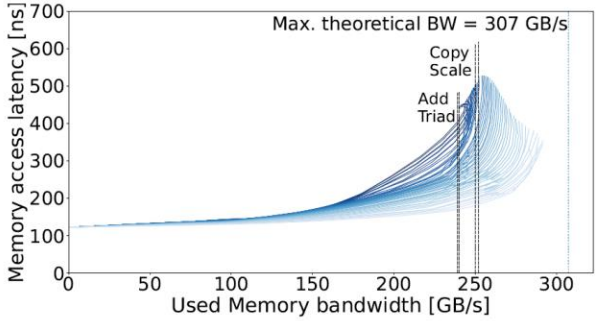
NVIDIA H100 with 4× HBM2E.

Mess benchmark: Actual platforms

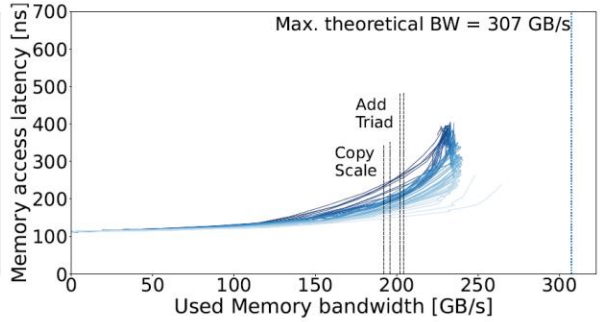


Intel Skylake with 6× DDR4-2666.

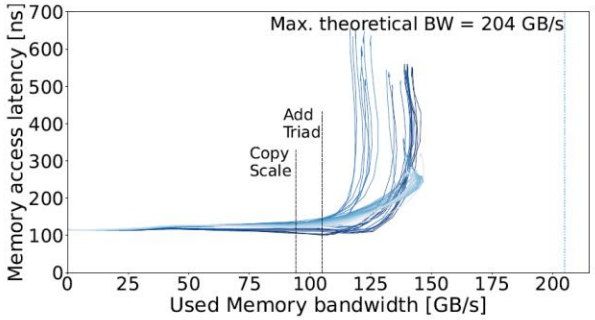
Intel Cascade Lake with 6× DDR4-2666.



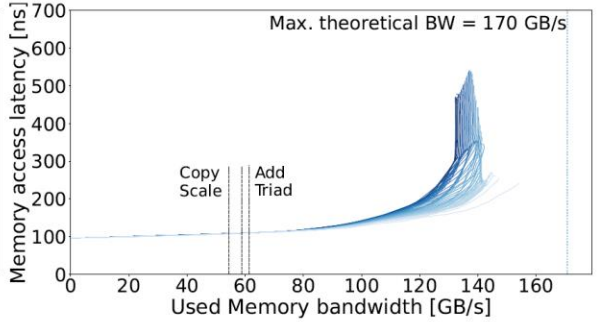
Amazon Graviton3 with 8× DDR5-4800.



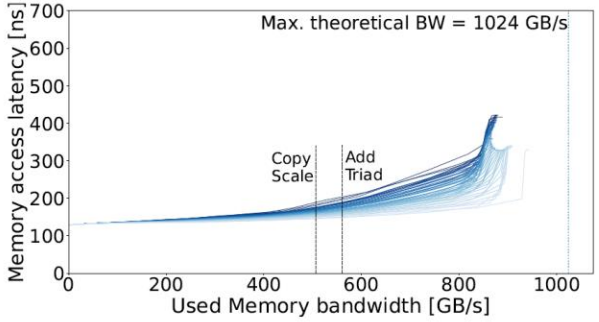
Intel Sapphire Rapids with 8× DDR5-4800.



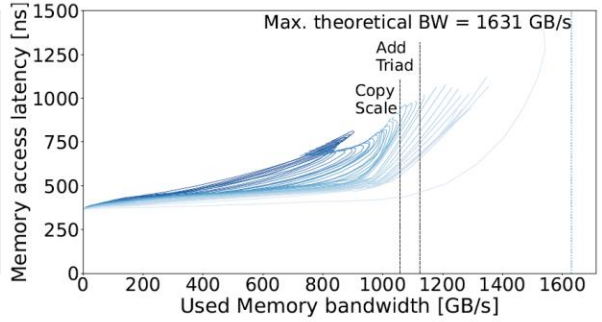
AMD Zen2 with 8× DDR4-3200



IBM Power 9 with 8× DDR4-2666.



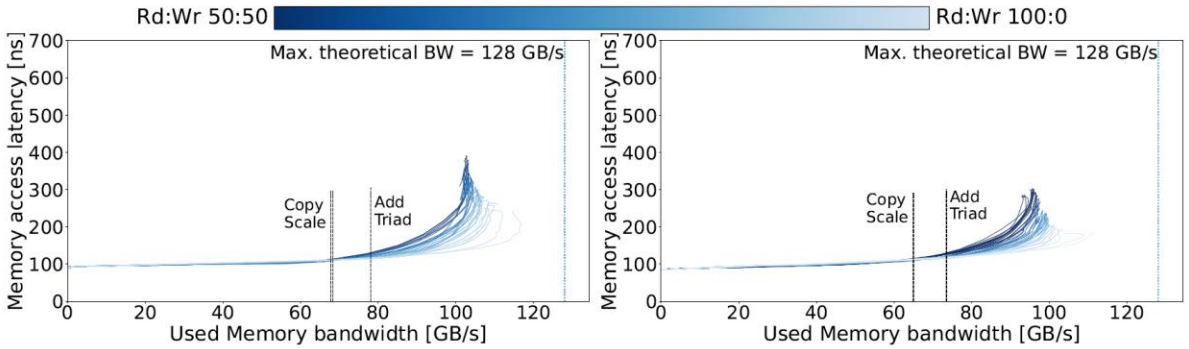
Fujitsu A64FX with 4× HBM2.



NVIDIA H100 with 4× HBM2E.

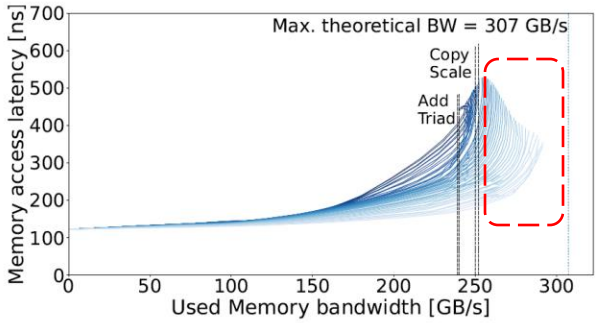
- Memory system performance is very, very different

Mess benchmark: Actual platforms

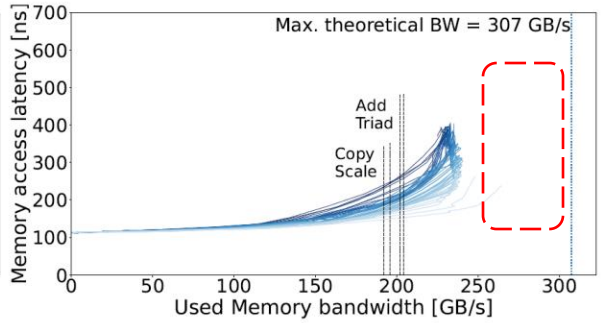


Intel Skylake with 6× DDR4-2666.

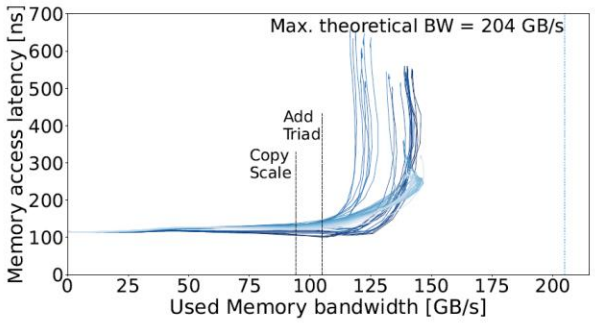
Intel Cascade Lake with 6× DDR4-2666.



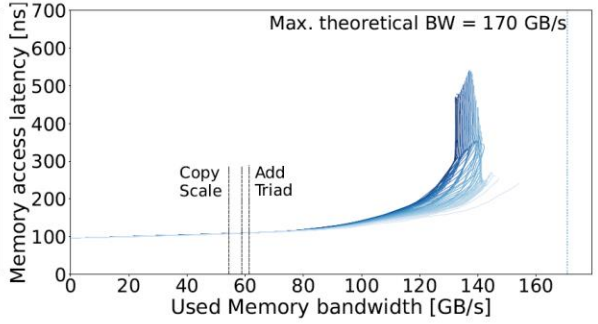
Amazon Graviton3 with 8× DDR5-4800.



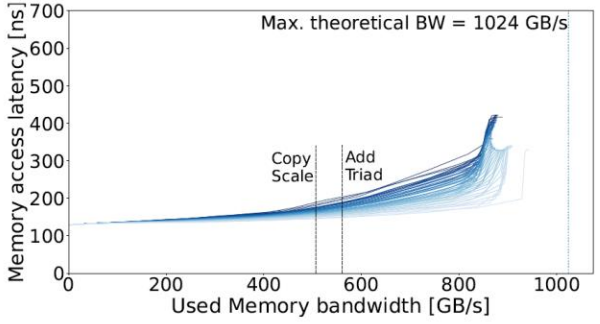
Intel Sapphire Rapids with 8× DDR5-4800.



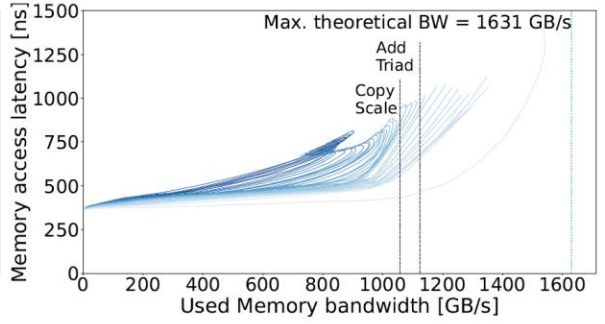
AMD Zen2 with 8× DDR4-3200



IBM Power 9 with 8× DDR4-2666.



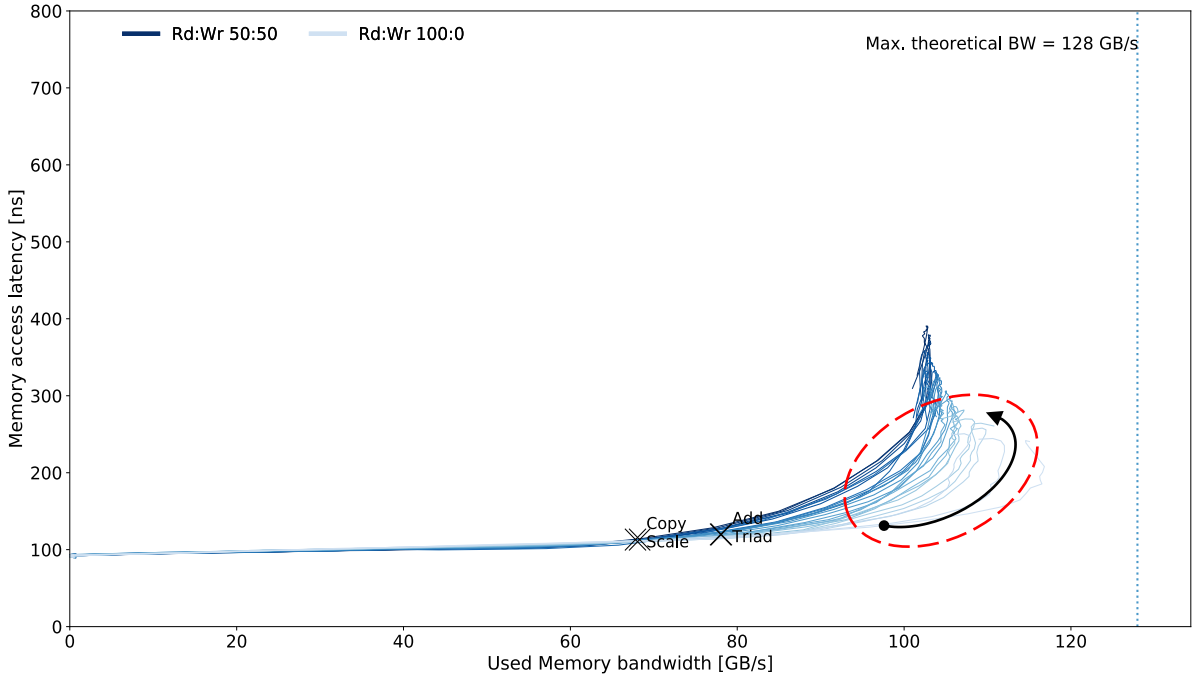
Fujitsu A64FX with 4× HBM2.



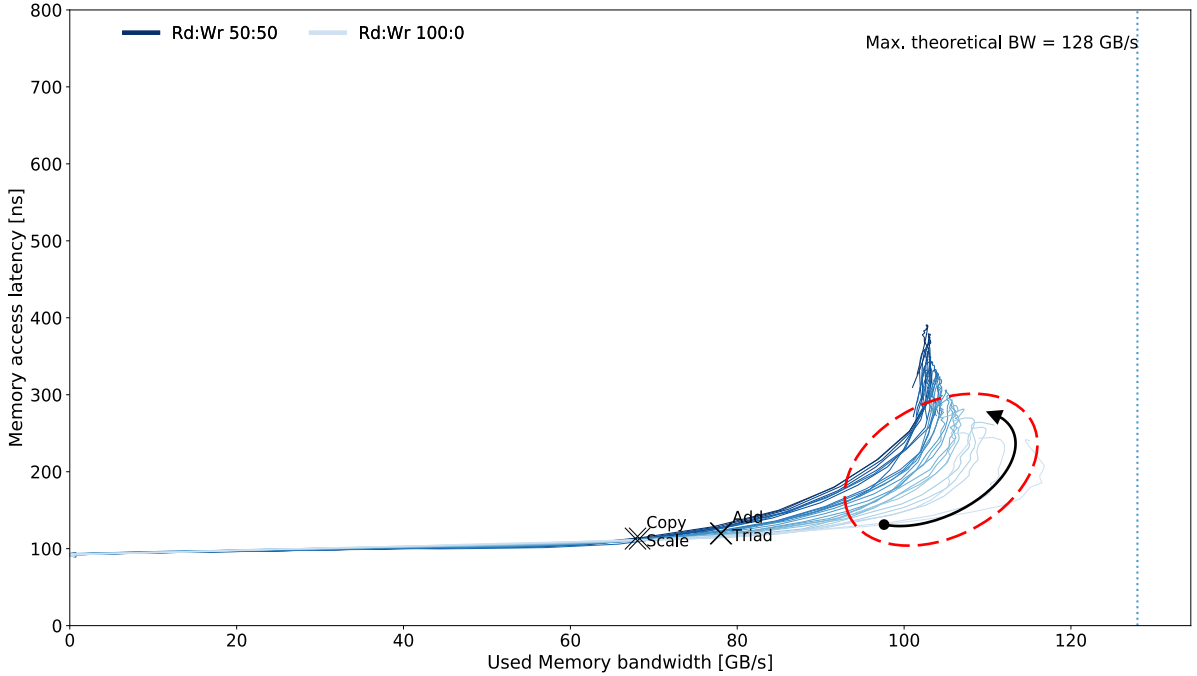
NVIDIA H100 with 4× HBM2E.

- Memory system performance is very, very different

The first one to detect the *“wave form”*?

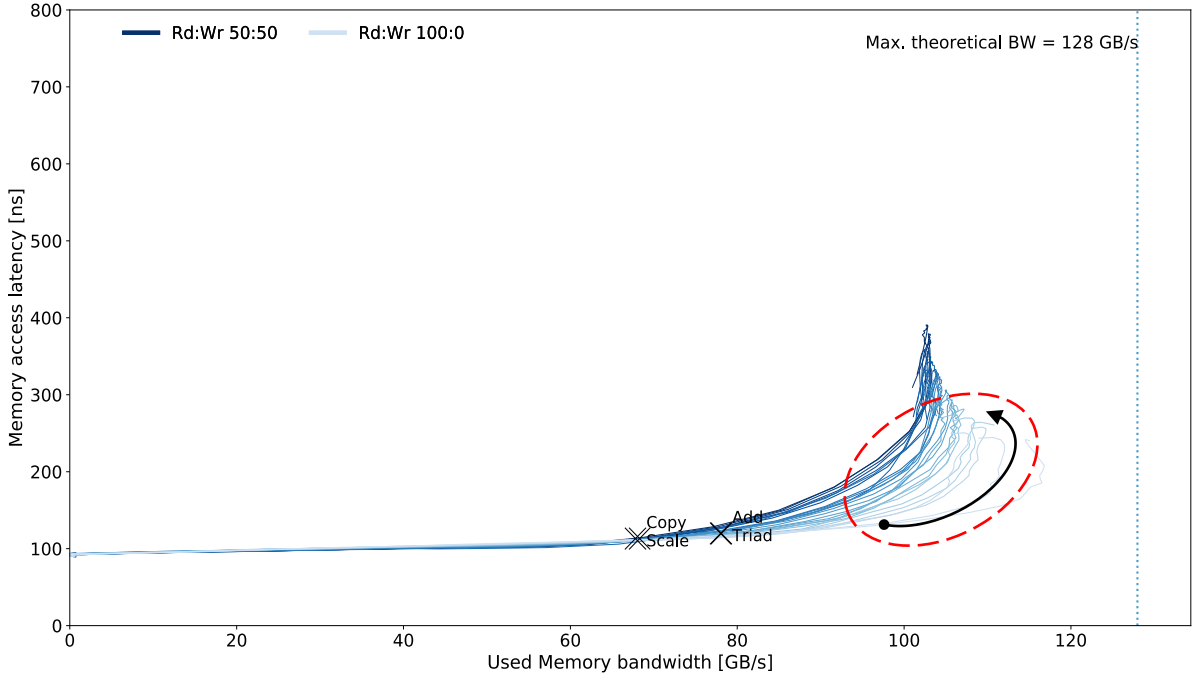


The first one to detect the *"wave form"*?



- ***Why?***
  - Honest answer: **We do not know**

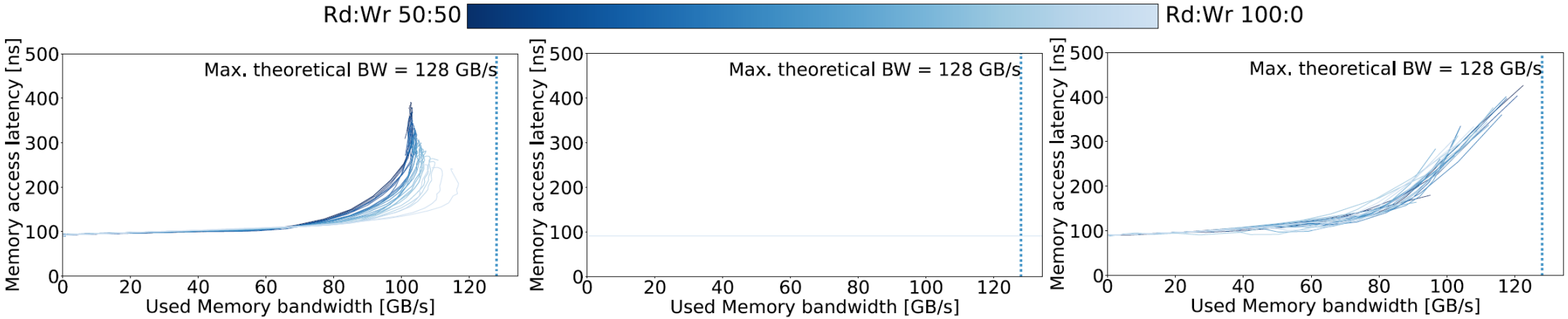
The first one to detect the *"wave form"*?



- ***Why?***
  - Honest answer: **We do not know**
- Preliminary analysis: Row-buffer miss rate increases
- Other hypotheses:
  - Back-pressure, some queues get full
  - Memory device throttling

What about the memory simulators?

ZSim



Intel Skylake with 6×DDR4-2666.

ZSim: Fixed-latency model

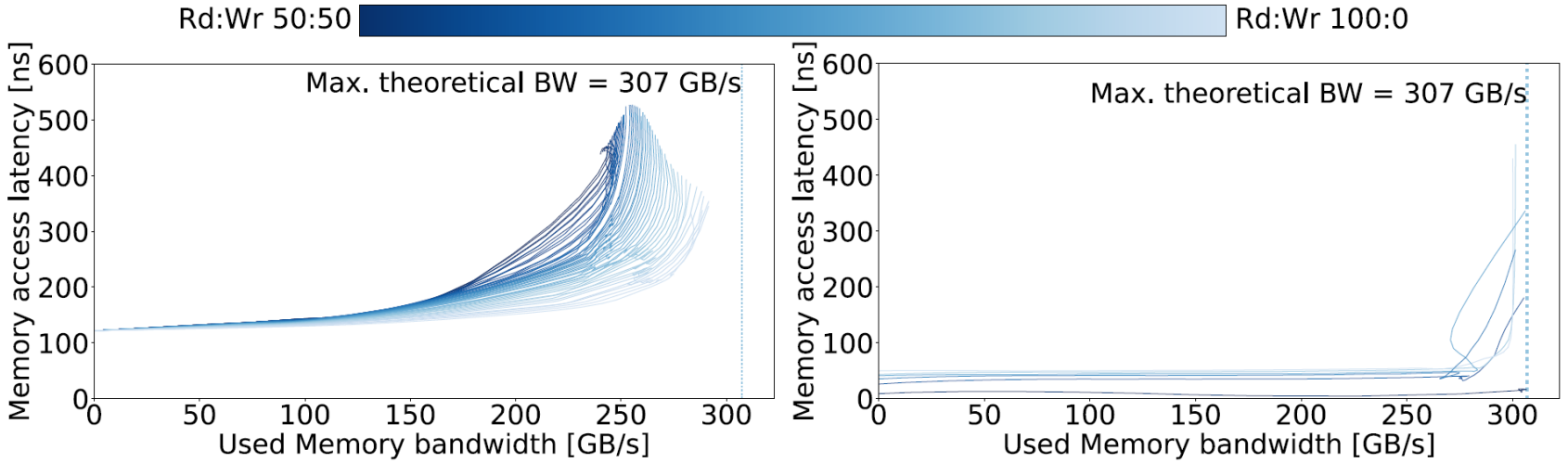
ZSim: M/D/1 Queue model

ZSim: Internal DDR model

ZSim+DRAMsim3 simulator.

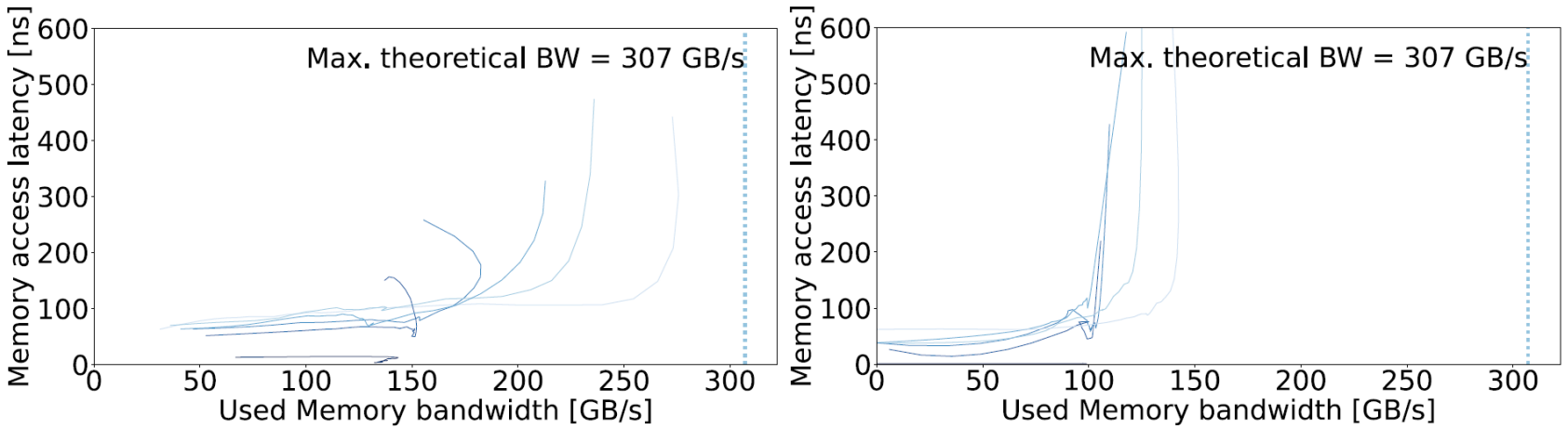
ZSim+Ramulator simulator.

gem5



Amazon Graviton3: 8×DDR5-4800

gem5: Simple memory model



gem5: Internal DDR model

gem5+Ramulator2

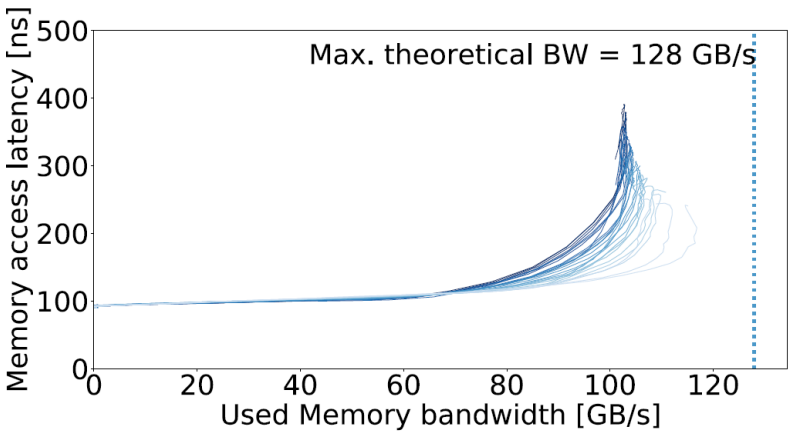
Why so large simulation errors?

- Honest answer: **We do not know**

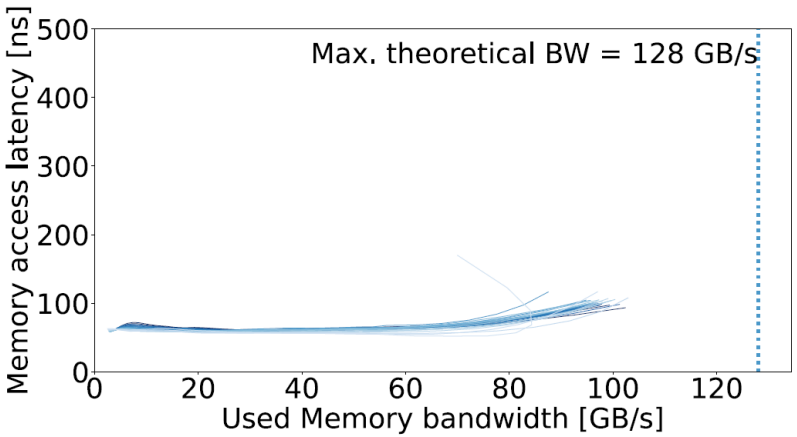
- Currently exploring sources of error:
  - Memory simulators design
    - Row buffer statistics

  - Interfaces: CPU sim → Memory sim

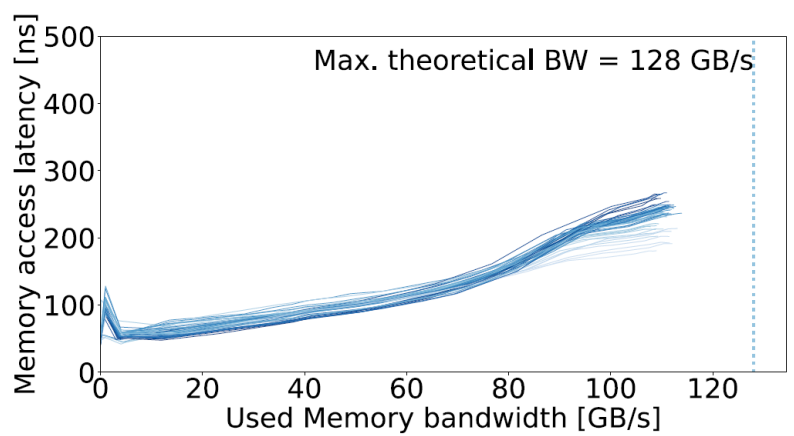
DRAMsim3: ZSim vs. Trace-driven



Intel Skylake with 6×DDR4-2666.

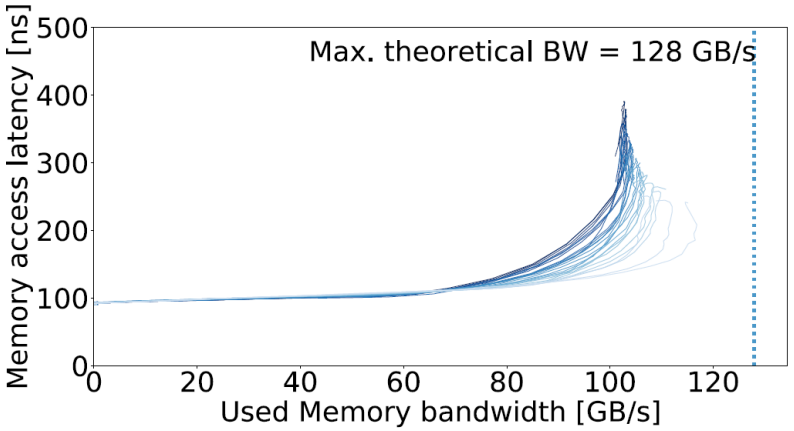


ZSim+DRAMsim3

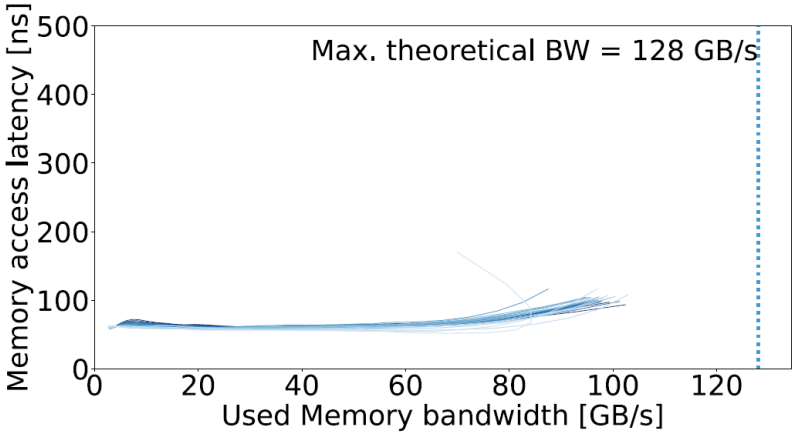


Trace-driven DRAMsim3

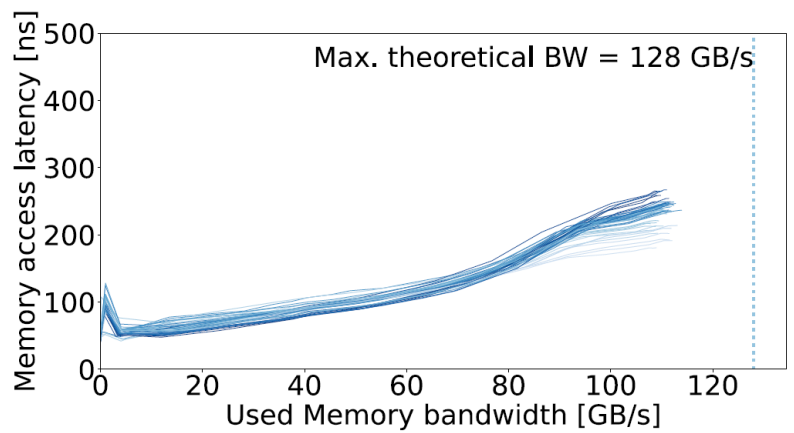
DRAMsim3: ZSim vs. Trace-driven



Intel Skylake with 6×DDR4-2666.

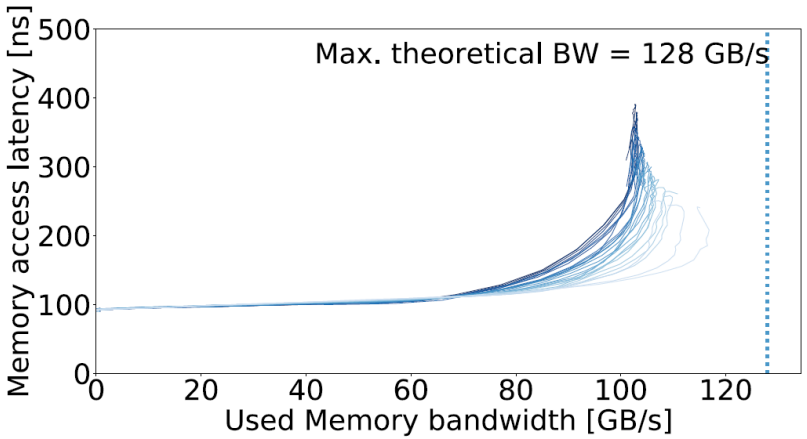


ZSim+DRAMsim3

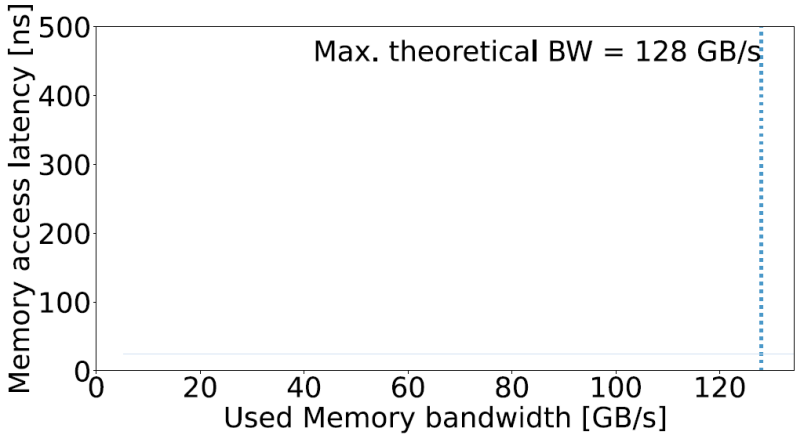


Trace-driven DRAMsim3

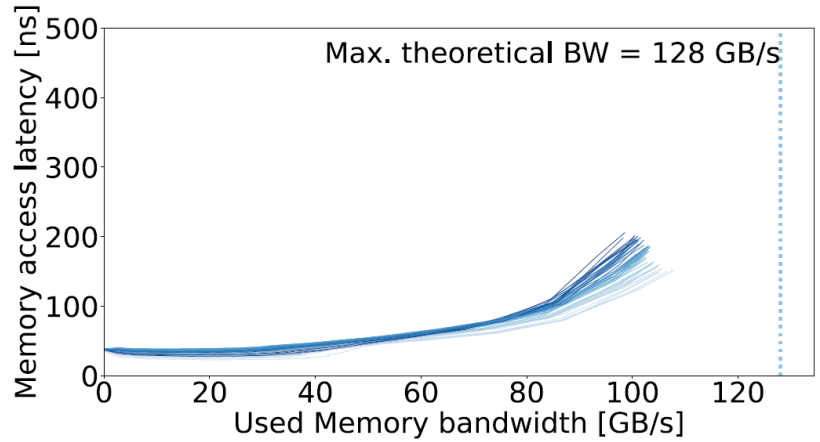
Ramulator: ZSim vs. Trace-driven



Intel Skylake with 6×DDR4-2666



ZSim+Ramulator



Trace-driven Ramulator

“But these simulators are already validated…”

3.1 Validating the Correctness of Memory Simulator X

To make sure Simulator X memory controller and DRAM device model implementation is correct (i.e., the DRAM commands issued by the controller obey both the timing constraints and the state transition rules), we verify the DRAM command trace against Micron’s DDR4 Verilog Model [24] using a similar methodology to prior works [2–4]. To do so, we implement a DRAM command trace recorder as a DRAM controller plugin that can store the issued DRAM commands with the addresses and time stamps using the DDR4 Verilog Model’s format. We collect DRAM command traces from eight streaming-access and eight random-access synthetic memory traces and different intensities (i.e., the number of non-memory instructions between memory instructions). We feed the DRAM command trace to the Verilog Model, configured to use the same DRAM organization and timings as we use in Simulator X We find no timing or state transition violations.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

“But these simulators are already validated…”

3.1 Validating the Correctness of Memory Simulator X

To make sure Simulator X memory controller and DRAM device model implementation is correct (i.e., the DRAM commands issued by the controller obey both the timing constraints and the state transition rules), we verify the DRAM command trace against Micron’s DDR4 Verilog Model [24] using a similar methodology to prior works [2–4]. To do so, we implement a DRAM command trace recorder as a DRAM controller plugin that can store the issued DRAM commands with the addresses and time stamps using the DDR4 Verilog Model’s format. We collect DRAM command traces from eight streaming-access and eight random-access synthetic memory traces and different intensities (i.e., the number of non-memory instructions between memory instructions). We feed the DRAM command trace to the Verilog Model, configured to use the same DRAM organization and timings as we use in Simulator X We find no timing or state transition violations.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

"But these simulators are already validated…"

3.1 Validating the Correctness of Memory Simulator X

To make sure Simulator X memory controller and DRAM device model implementation is correct (i.e., the DRAM commands issued by the controller obey both the timing constraints and the state transition rules), we verify the DRAM command trace against Micron's DDR4 Verilog Model [24] using a similar methodology to prior works [2–4]. To do so, we implement a DRAM command trace recorder as a DRAM controller plugin that can store the issued DRAM commands with the addresses and time stamps using the DDR4 Verilog Model's format. We collect DRAM command traces from eight streaming-access and eight random-access synthetic memory traces and different intensities (i.e., the number of non-memory instructions between memory instructions). We feed the DRAM command trace to the Verilog Model, configured to use the same DRAM organization and timings as we use in Simulator X We find no timing or state transition violations.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

"But these simulators are already validated…"

3.1 Validating the Correctness of Memory Simulator X

To make sure Simulator X memory controller and DRAM device model implementation is correct (i.e., the DRAM commands issued by the controller obey both the timing constraints and the state transition rules), we verify the DRAM command trace against Micron's DDR4 Verilog Model [24] using a similar methodology to prior works [2–4]. To do so, we implement a DRAM command trace recorder as a DRAM controller plugin that can store the issued DRAM commands with the addresses and time stamps using the DDR4 Verilog Model's format. We collect DRAM command traces from eight streaming-access and eight random-access synthetic memory traces and different intensities (i.e., the number of non-memory instructions between memory instructions). We feed the DRAM command trace to the Verilog Model, configured to use the same DRAM organization and timings as we use in Simulator X We find no timing or state transition violations.

- **Implies nothing for**
  DDR5, LPx, GDDRx, or HBMx

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron®

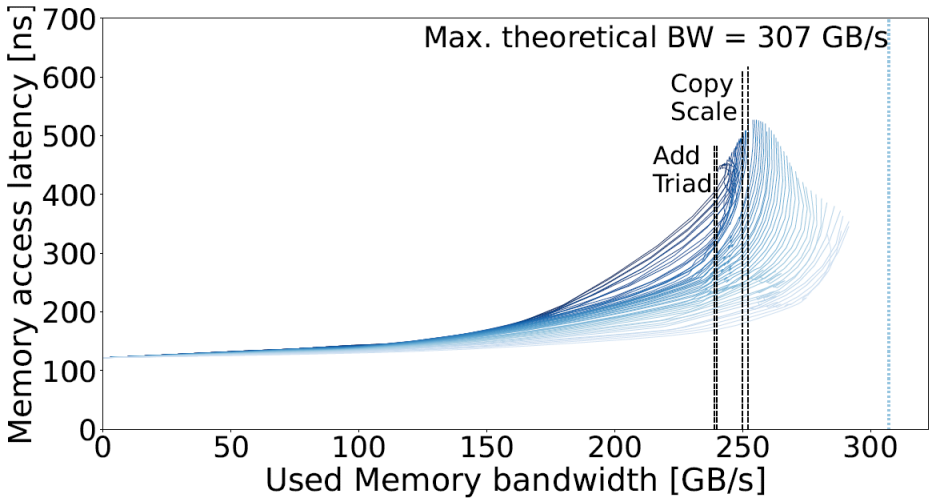
"But these simulators are already validated…"

3.1 Validating the Correctness of Memory Simulator X

To make sure Simulator X memory controller and DRAM device model implementation is correct (i.e., the DRAM commands issued by the controller obey both the timing constraints and the state transition rules), we verify the DRAM command trace against Micron's DDR4 Verilog Model [24] using a similar methodology to prior works [2–4]. To do so, we implement a DRAM command trace recorder as a DRAM controller plugin that can store the issued DRAM commands with the addresses and time stamps using the DDR4 Verilog Model's format. We collect DRAM command traces from eight streaming-access and eight random-access synthetic memory traces and different intensities (i.e., the number of non-memory instructions between memory instructions). We feed the DRAM command trace to the Verilog Model, configured to use the same DRAM organization and timings as we use in Simulator X. We find no timing or state transition violations.

- **Implies nothing for** DDR5, LPx, GDDRx, or HBMx

- **Does not imply** ok performance



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

Large simulation errors: Simulator setup?

- We are (not me!) **skilled users** of the simulators under study
  - If you do not trust me, I have a back-up slide ;-)

- Work presented in the paper: **Typical use-case**
  - Find all info about the simulator installation and setup: papers, git, manuals, blogs, …
  - Unless we had major issues, we asked no support from simulator developers

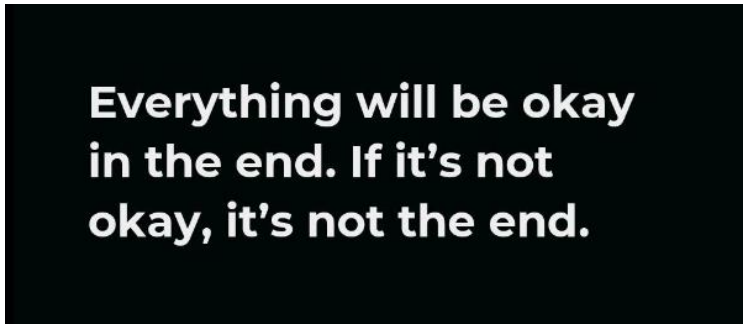
- **We did not get the most** out of these simulators **yet!**

Large simulation errors: Simulator setup?

- **Ongoing work**
  - Contact simulator developers: DRAMSim3, DRAMSys, gem5 (main branch), Ramulator /2
  - Can we figure out together what are the performance issues?
    - Back-up slide: Preliminary nice story

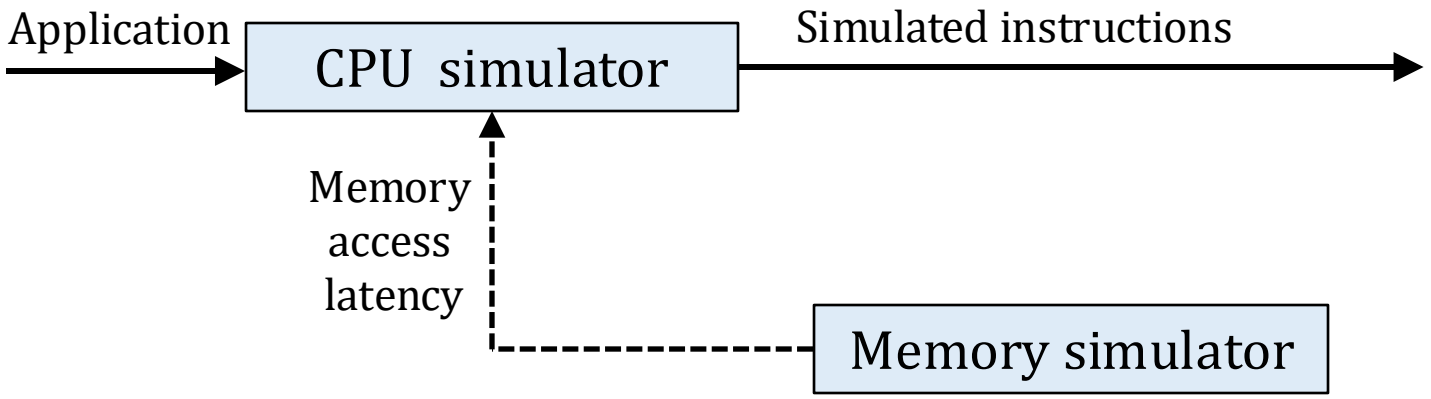
Large simulation errors: Simulator setup?

- **Ongoing work**
  - Contact simulator developers: DRAMSim3, DRAMSys, gem5 (main branch), Ramulator /2
  - Can we figure out together what are the performance issues?
    - Back-up slide: Preliminary nice story
- **My dream**
  - Be here next year (workshop?) to present the results and experiences



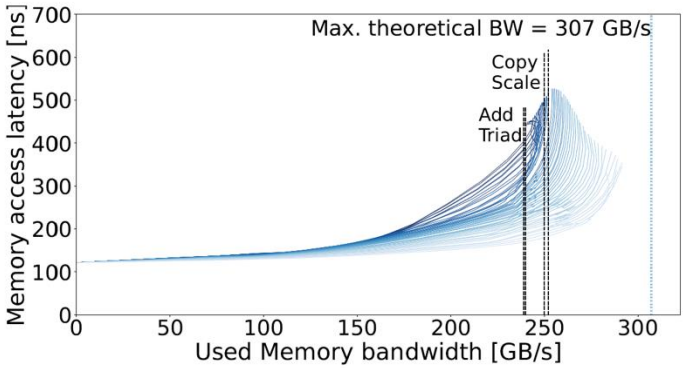
Should we rethink the memory simulation?

- Step back: The main purpose of the memory simulator

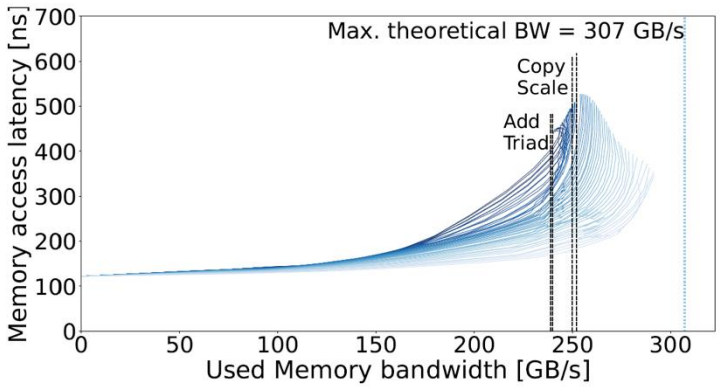


Should we rethink the memory simulation?

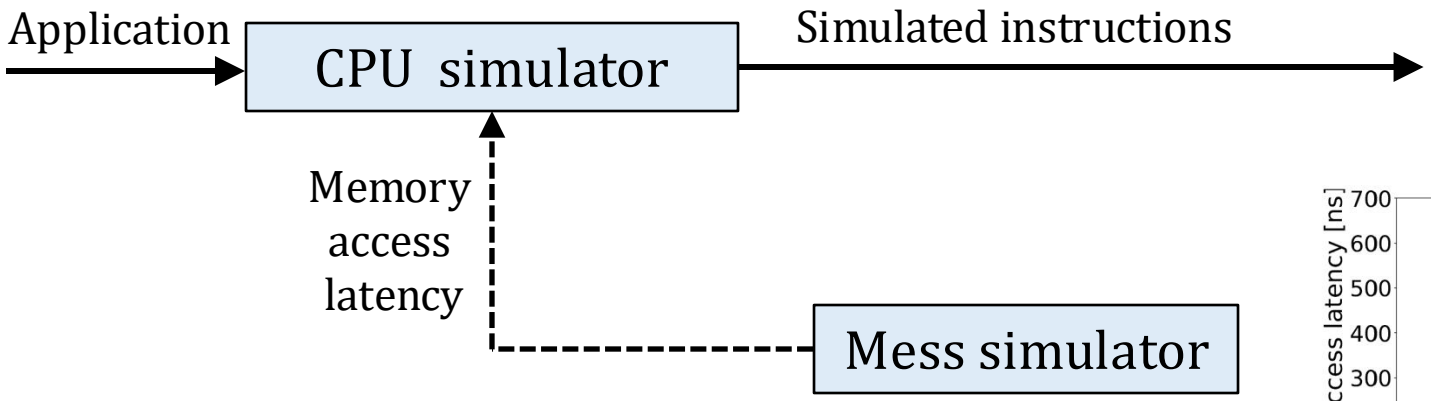
If our objective is to simulate

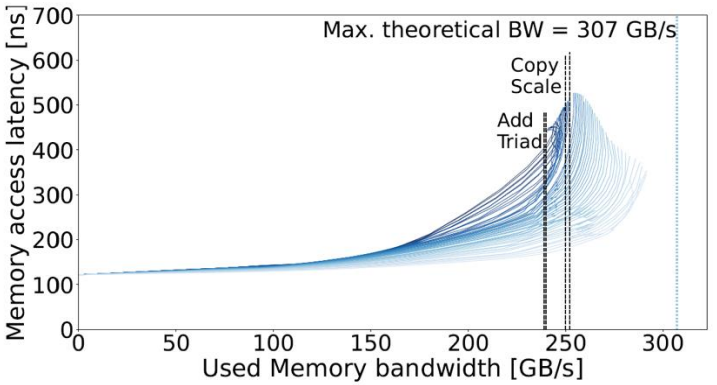


why don’t we use

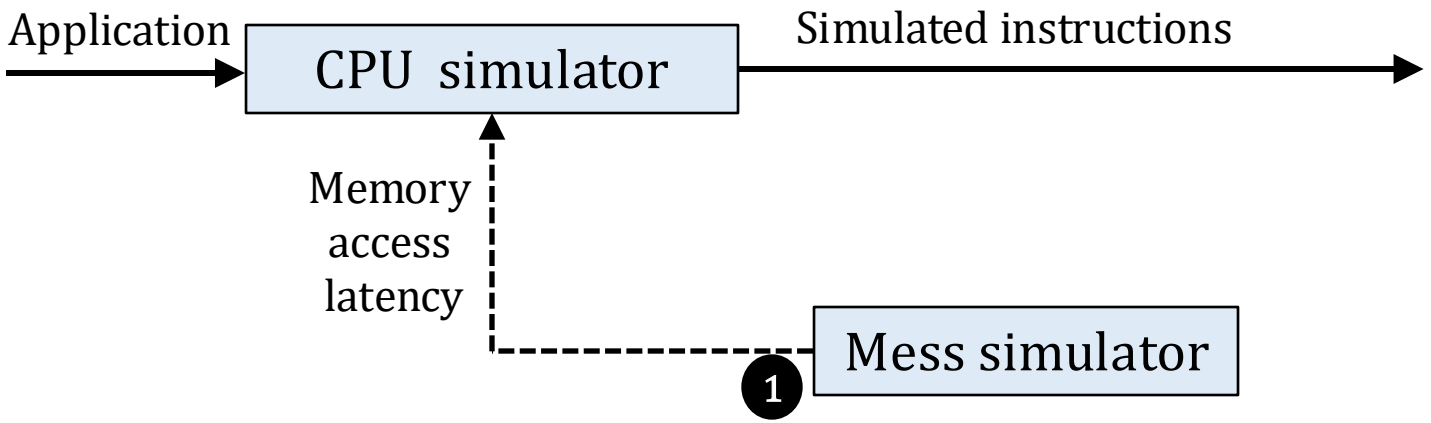


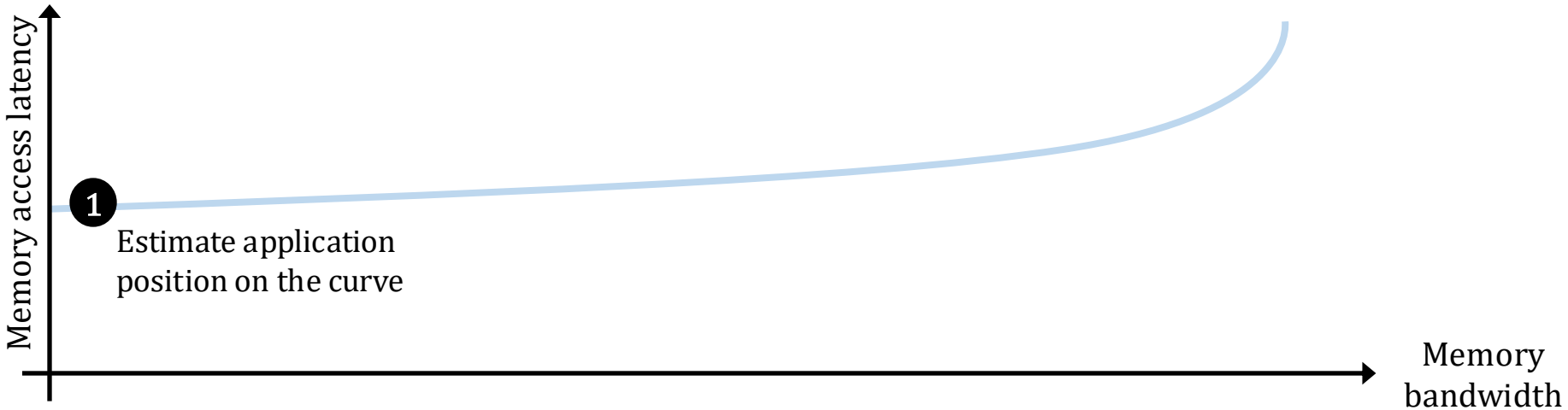
Mess simulator



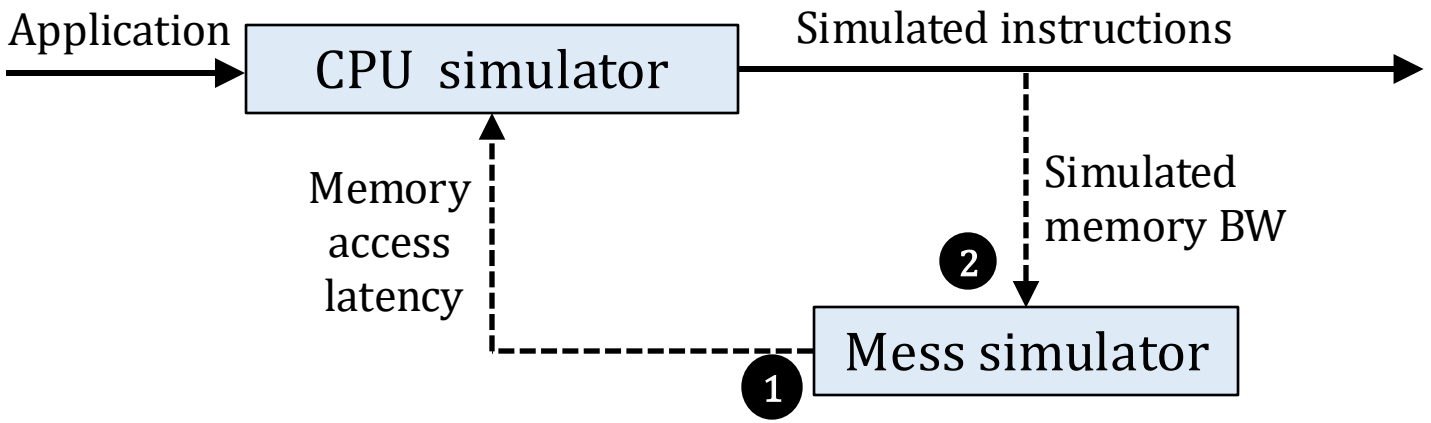


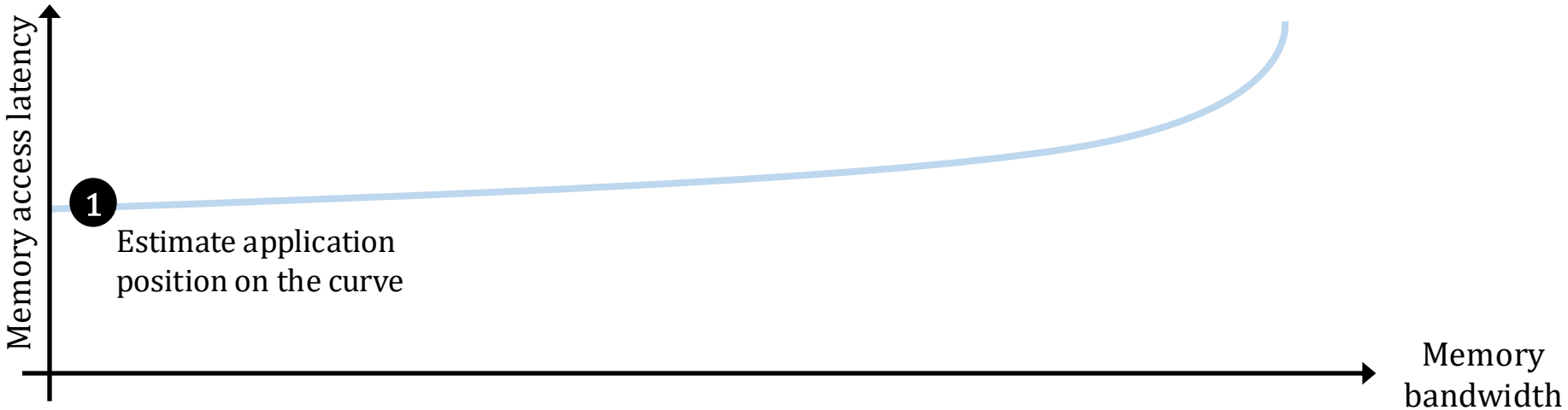
Mess simulator



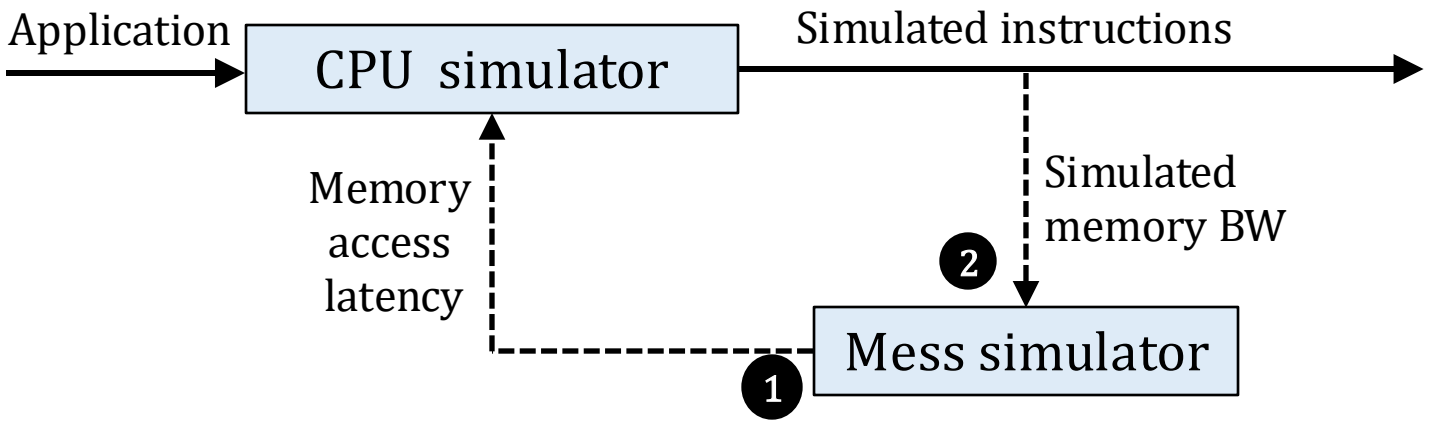


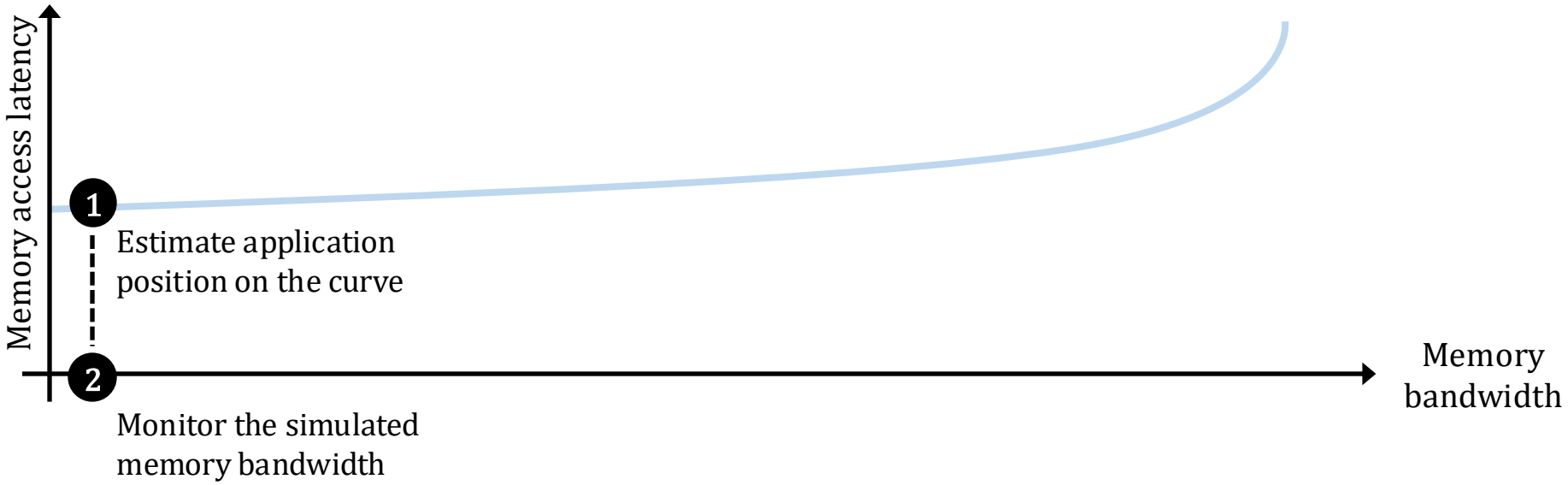
Mess simulator



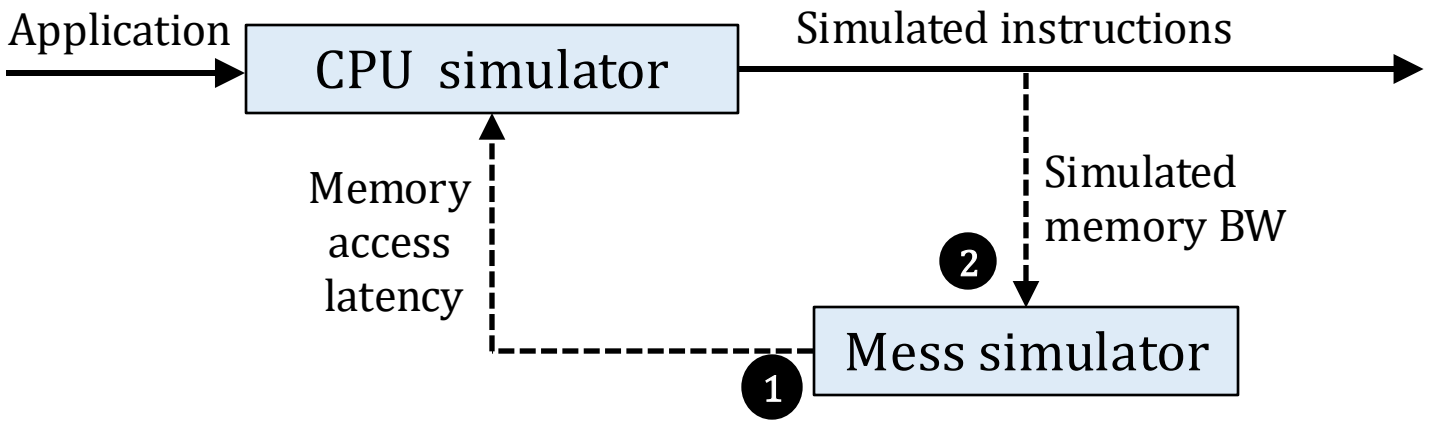


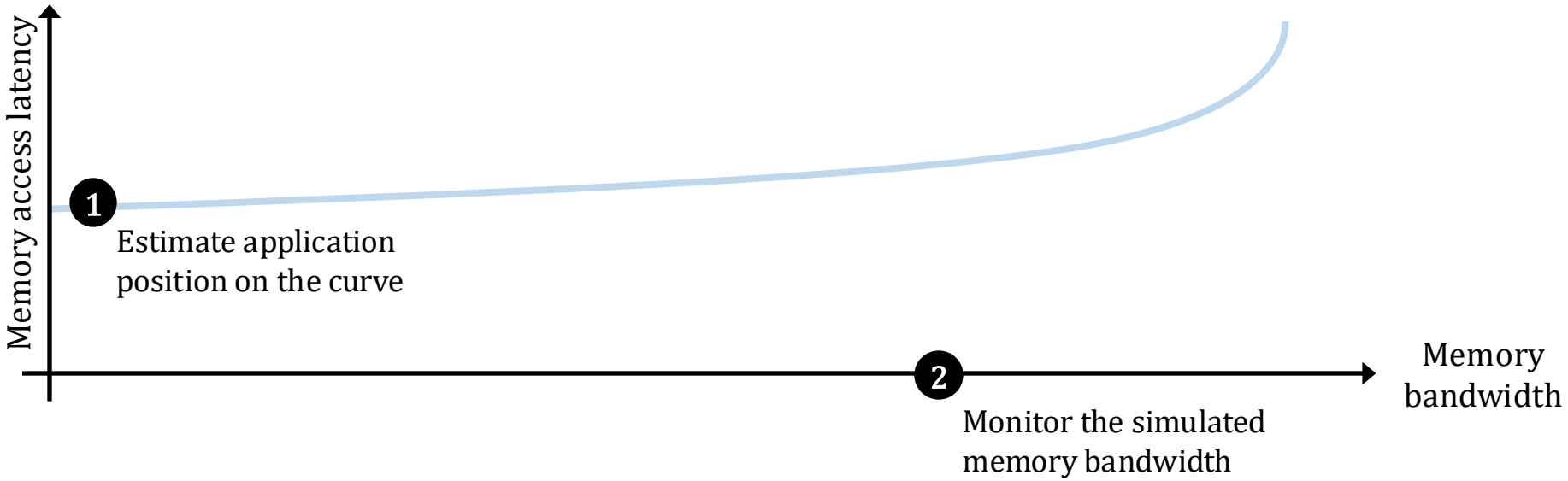
Mess simulator



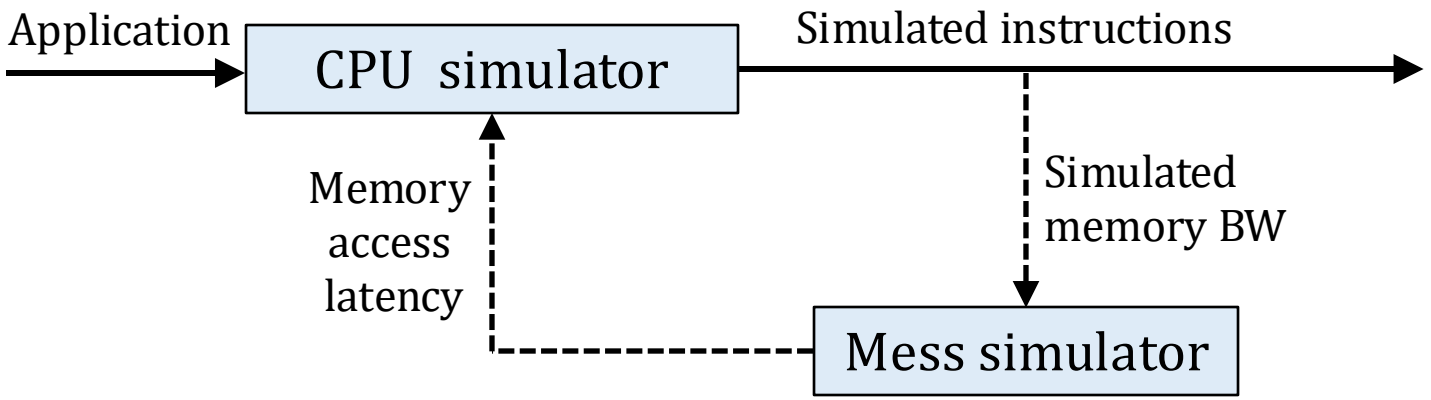


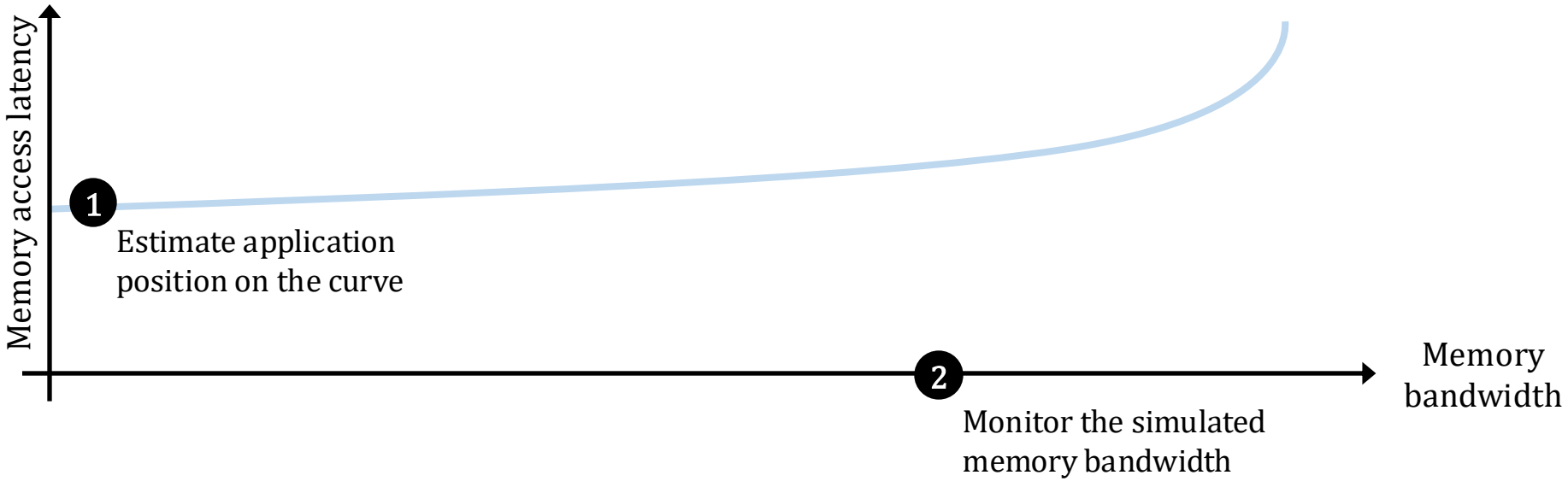
Mess simulator



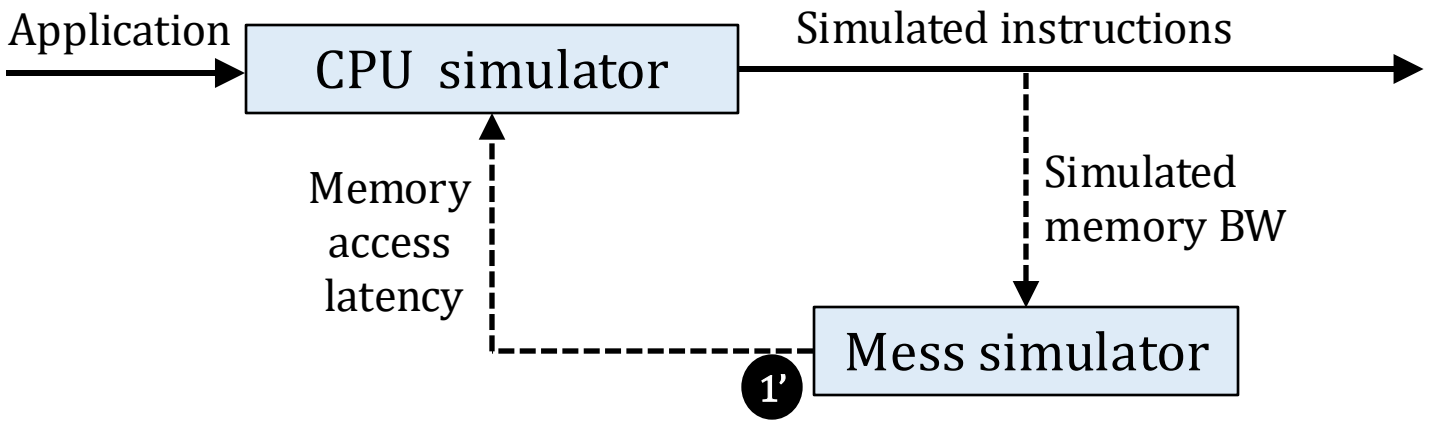


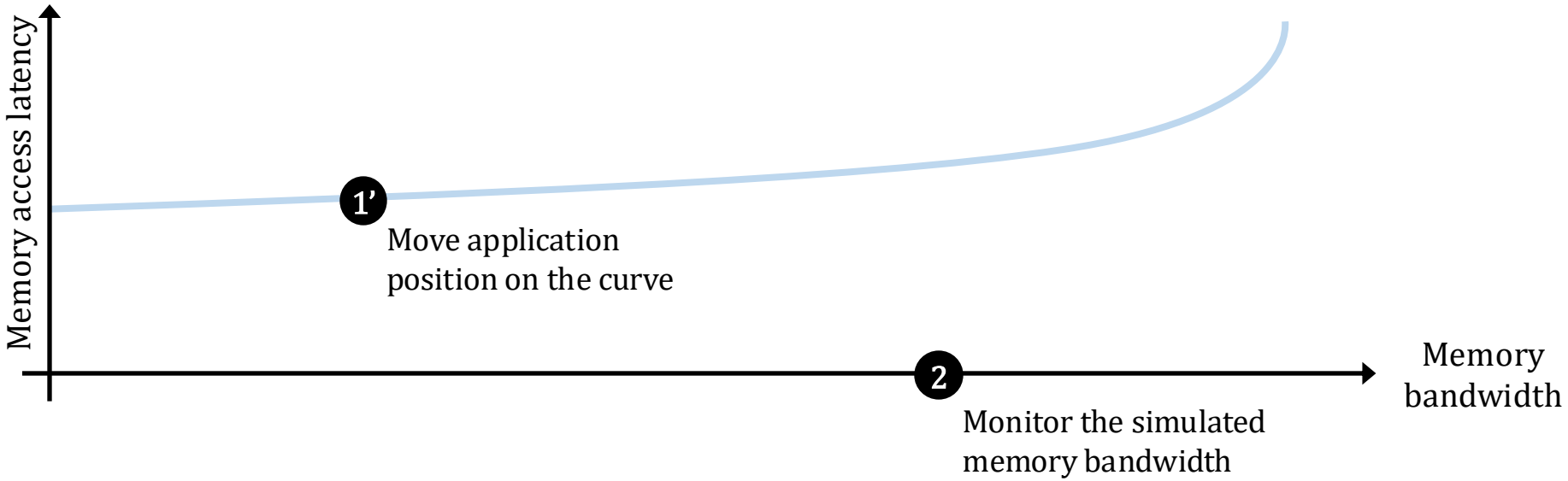
Mess simulator



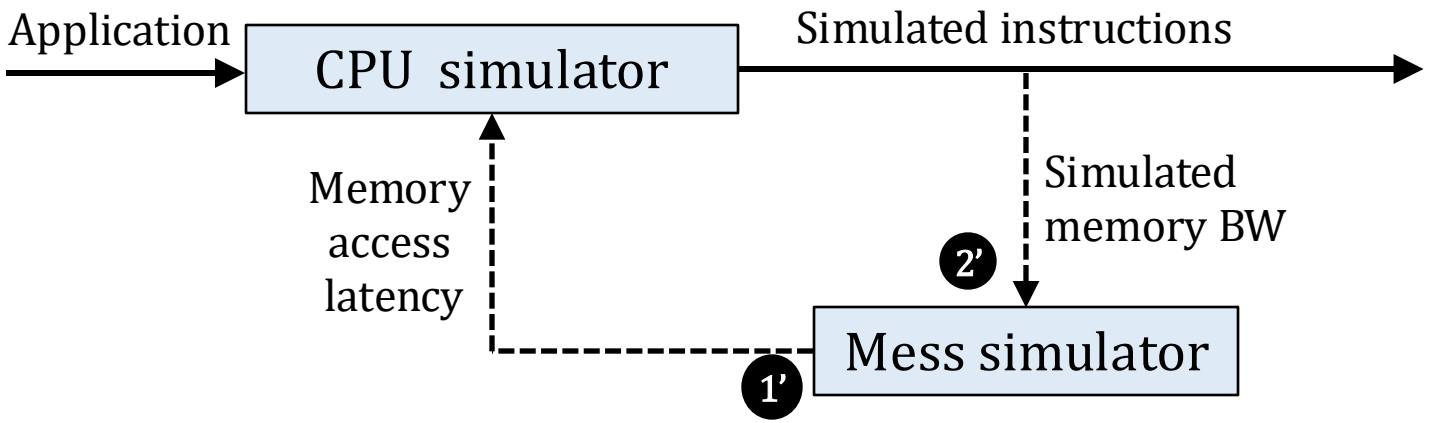


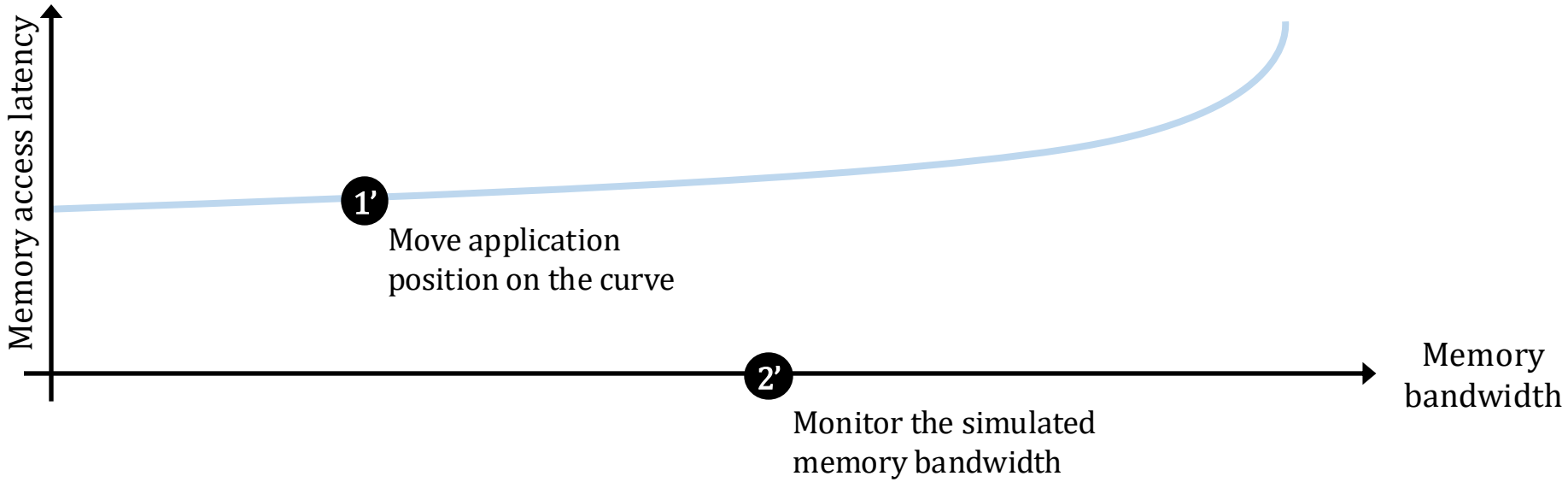
Mess simulator



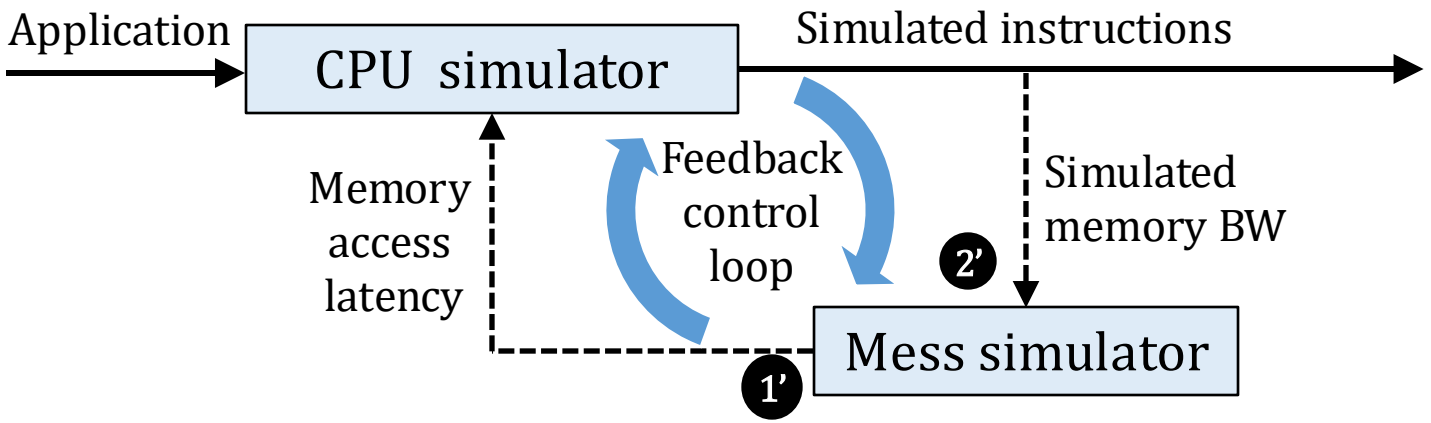


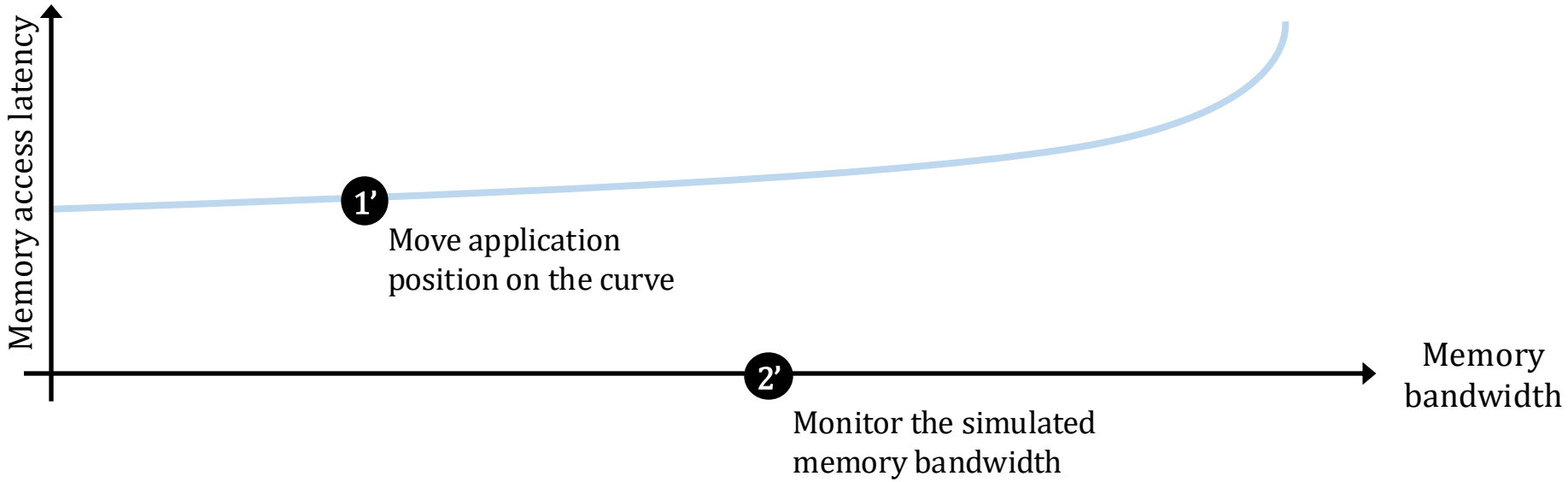
Mess simulator



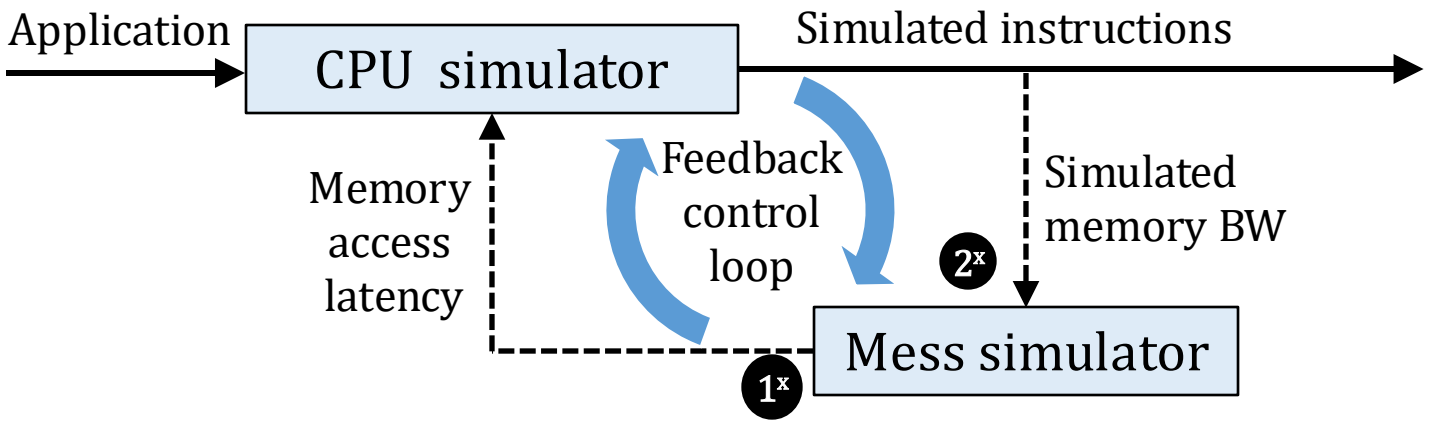


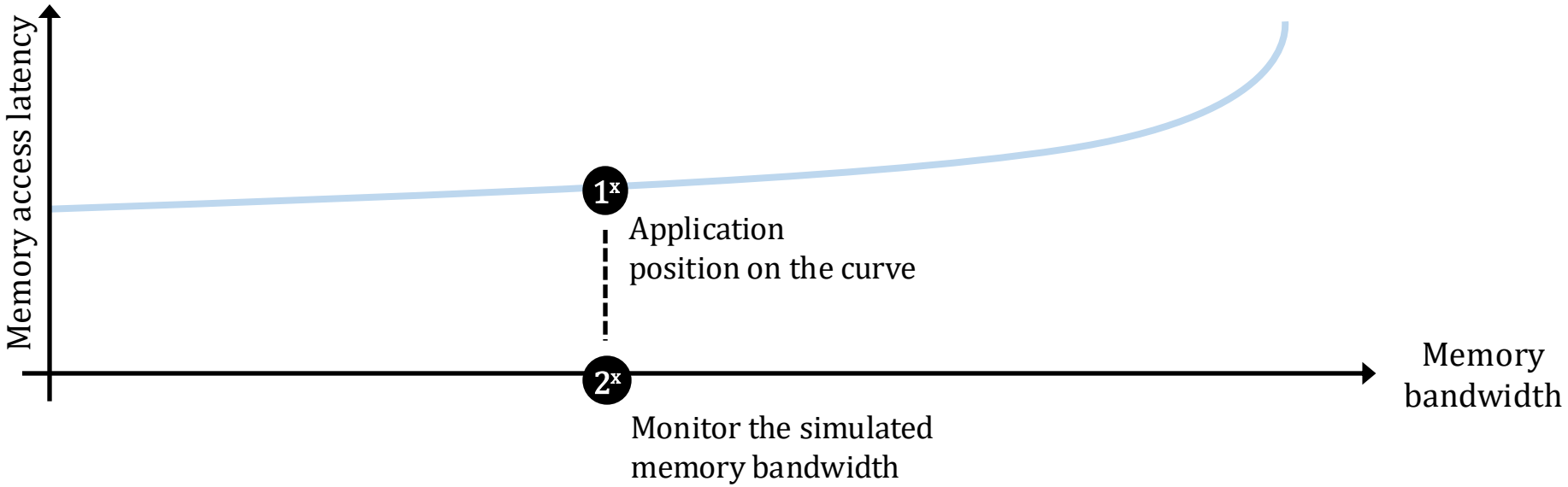
Mess simulator





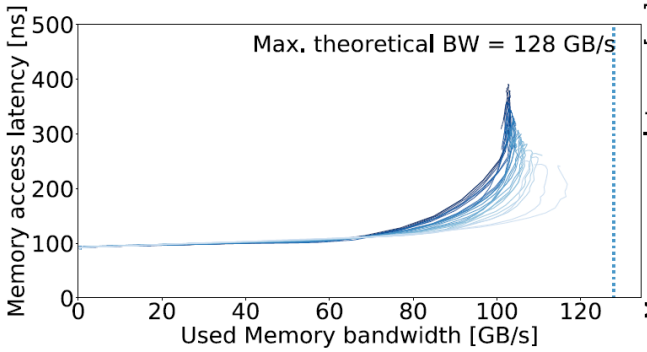
Mess simulator



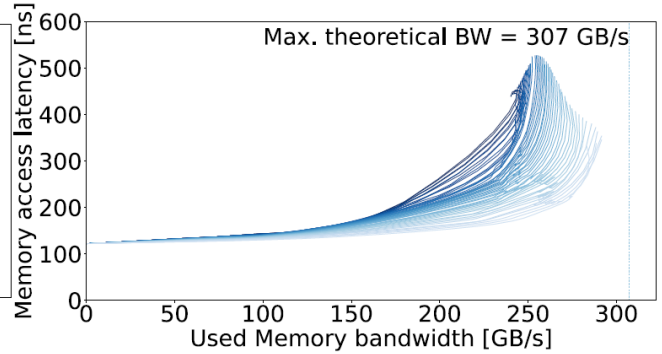


Mess simulator evaluation: ZSim

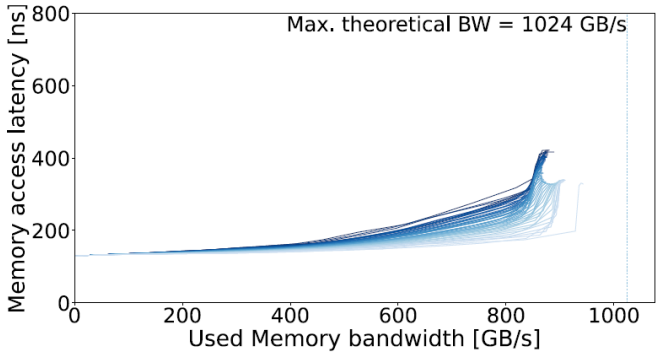
Actual (memory) system



Intel Skylake with 6×DDR4-2666.

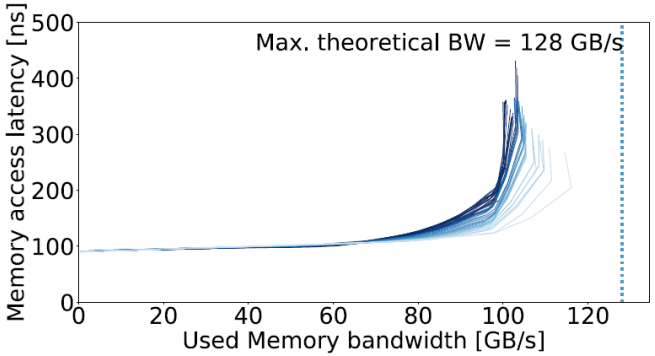


Amazon Graviton3: 8×DDR5-4800

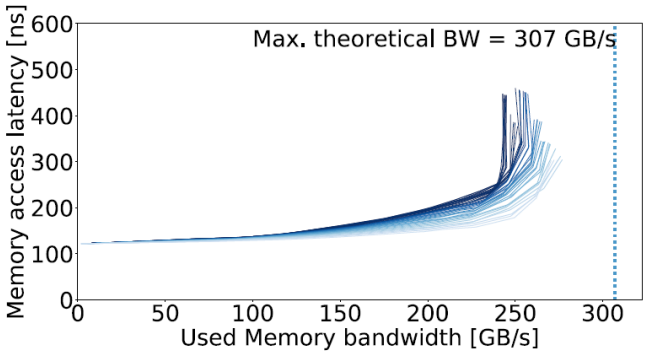


Fujitsu A64FX with 4× HBM2.

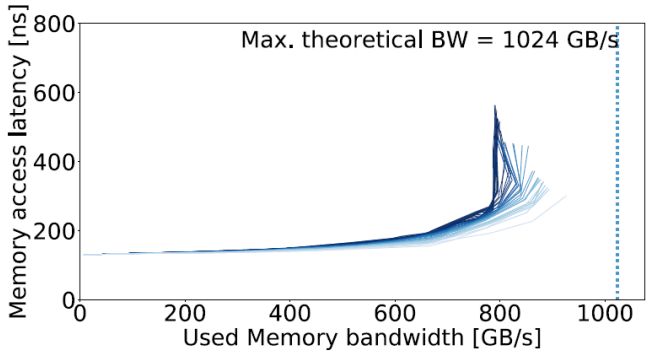
Zsim+Mess simulation



ZSim: 24 cores, 6× DDR4-2666



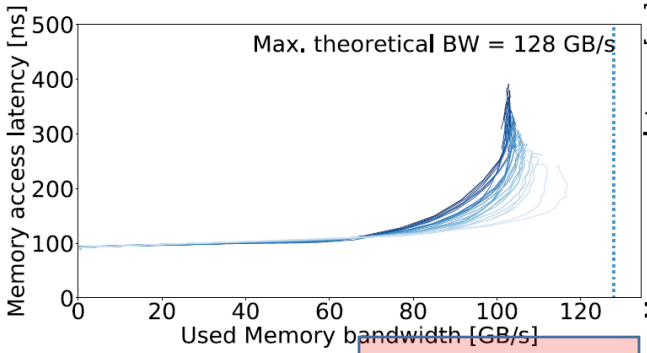
ZSim: 58 cores, 8× DDR5-4800



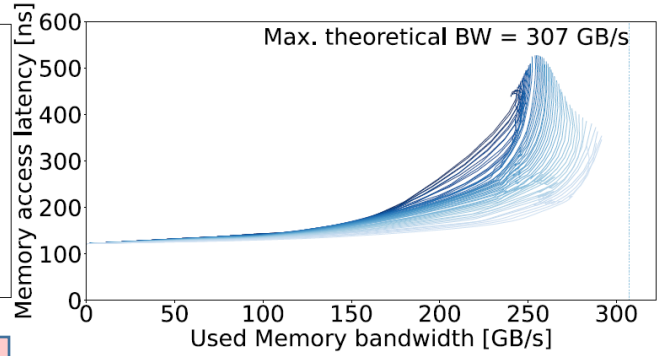
ZSim: 192 cores, 32× HBM2 channels

Mess simulator evaluation: ZSim

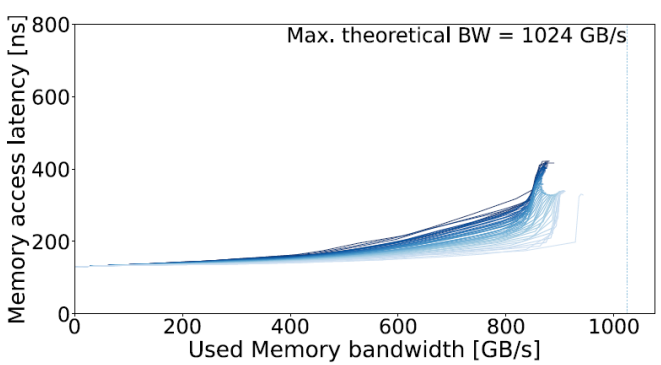
Actual (memory) system



Intel Skylake with 6×DDR4-2666.

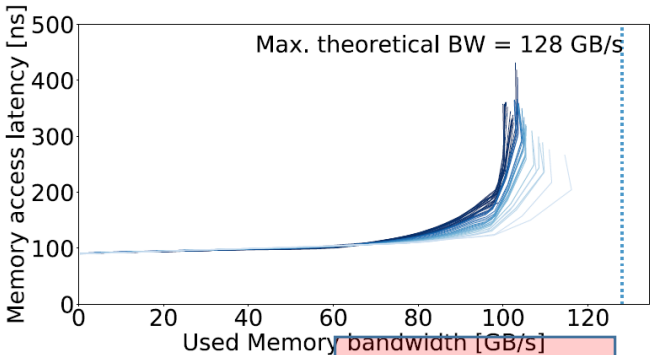


Amazon Graviton3: 8×DDR5-4800

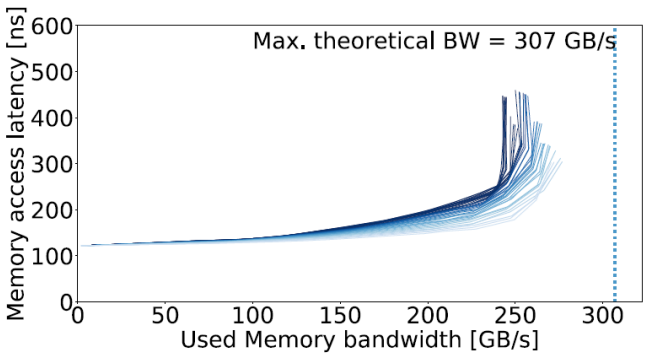


Fujitsu A64FX with 4× HBM2.

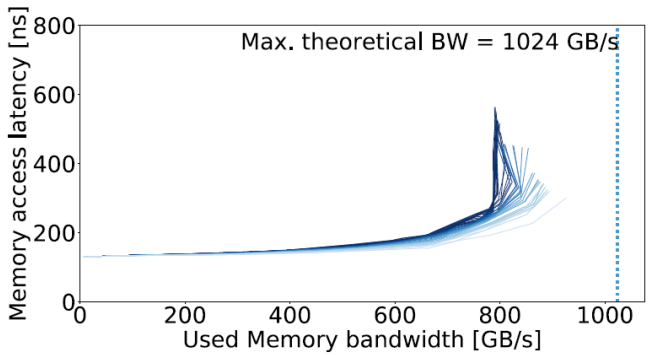
Zsim+Mess simulation



ZSim: 24 cores, 6× DDR4-2666



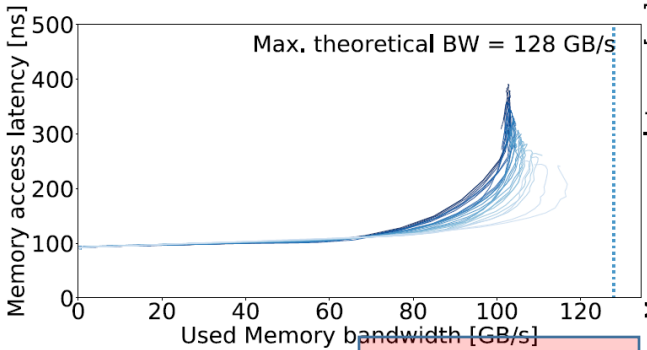
ZSim: 58 cores, 8× DDR5-4800



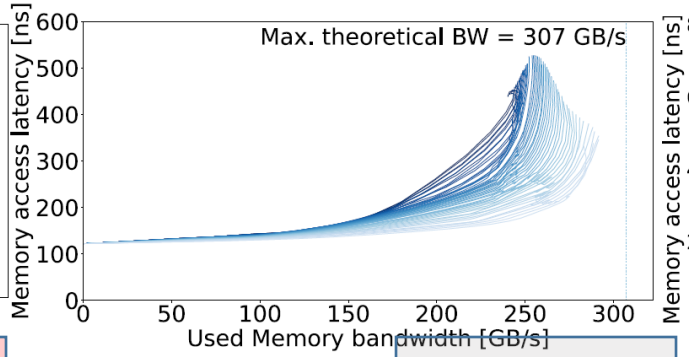
ZSim: 192 cores, 32× HBM2 channels

Mess simulator evaluation: ZSim

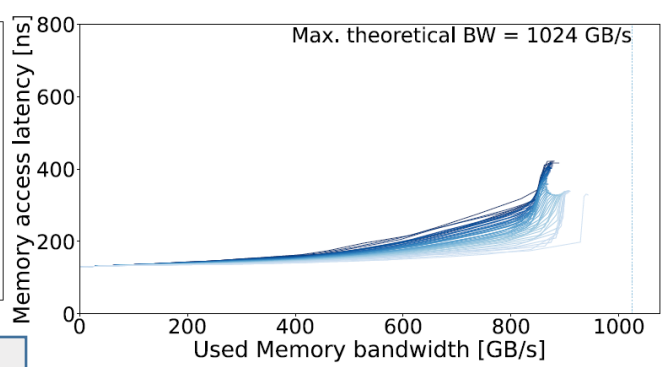
Actual (memory) system



Intel Skylake with 6×DDR4-2666.

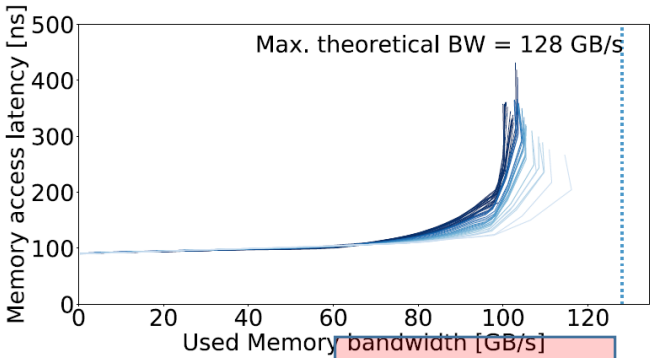


Amazon Graviton3: 8×DDR5-4800

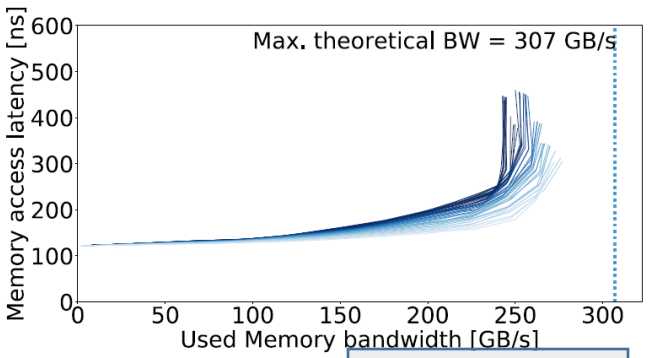


Fujitsu A64FX with 4× HBM2.

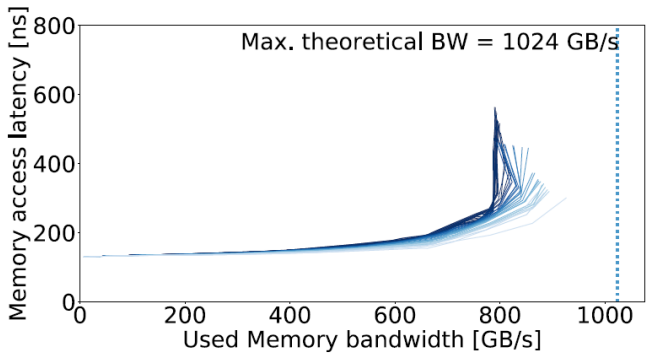
Zsim+Mess simulation



ZSim: 24 cores, 6× DDR4-2666



ZSim: 58 cores, 8× DDR5-4800

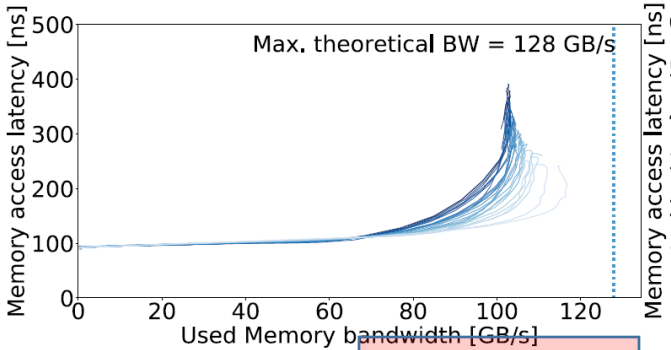


ZSim: 192 cores, 32× HBM2 channels

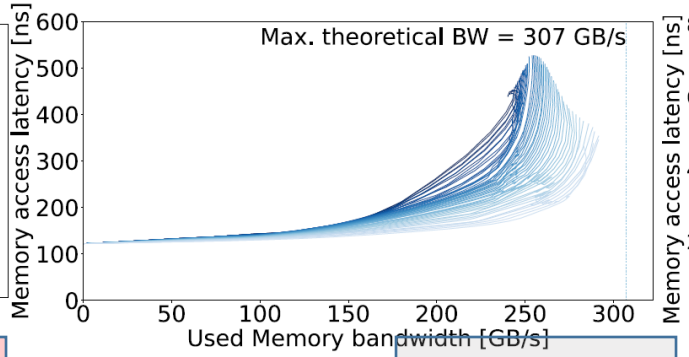
Mess simulator evaluation: ZSim

Actual (memory) system

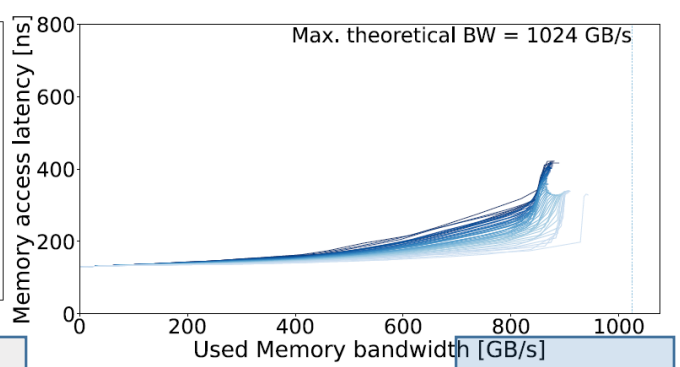
Zsim+Mess simulation



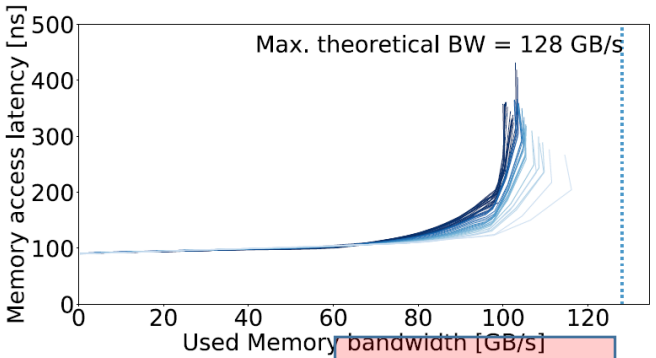
Intel Skylake with 6×DDR4-2666.



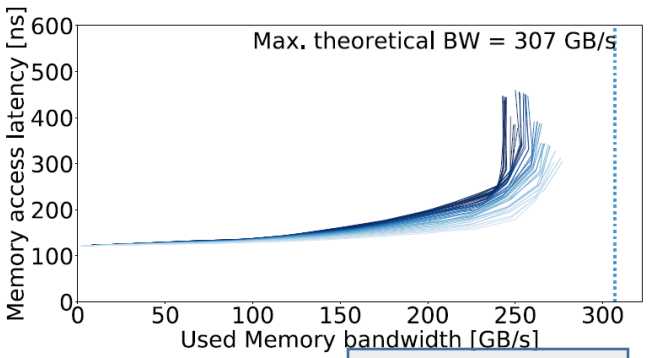
Amazon Graviton3: 8×DDR5-4800



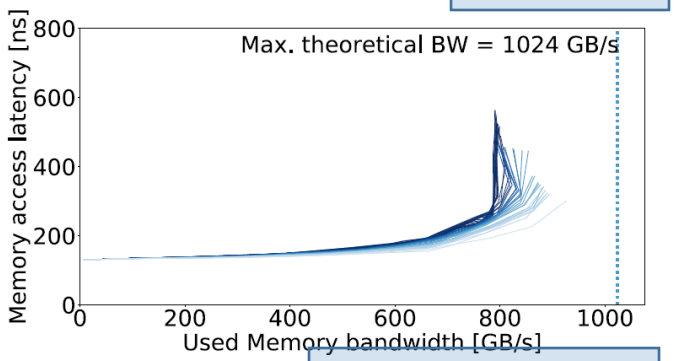
Fujitsu A64FX with 4× HBM2.



ZSim: 24 cores, 6× DDR4-2666



ZSim: 58 cores, 8× DDR5-4800



ZSim: 192 cores, 32× HBM2 channels

Mess simulator evaluation

- **Zsim *vs.* Actual Intel Skylake.** 24 cores with 6x DDR4-2666.

- **Zsim memory models**
  - Fixed latency
  - M/D/1 queue
  - Internal DDR
  - DRAMsim3
  - Ramulator
  - Mess

- **Memory-intensive benchmarks**
  - STREAM: copy, scale, add, triad
  - LMbench
  - Google multichase

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

Mess simulator evaluation

- **Zsim *vs.* Actual Intel Skylake.** 24 cores with 6x DDR4-2666.

- **Zsim memory models**
  - Fixed latency
  - M/D/1 queue
  - Internal DDR
  - DRAMsim3
  - Ramulator
  - Mess  1.3% error

- **Memory-intensive benchmarks**
  - STREAM: copy, scale, add, triad
  - LMbench
  - Google multichase

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

Mess simulator evaluation

- **Zsim *vs.* Actual Intel Skylake.** 24 cores with 6x DDR4-2666.

- **Zsim memory models**
  - Fixed latency
  - M/D/1 queue
  - Internal DDR
  - DRAMsim3
  - Ramulator
  - Mess  1.3% error

- **Gem5 evaluation:** Mess  3% error

- **Memory-intensive benchmarks**
  - STREAM: copy, scale, add, triad
  - LMbench
  - Google multichase

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

Mess Simulator: CXL Memory Expander

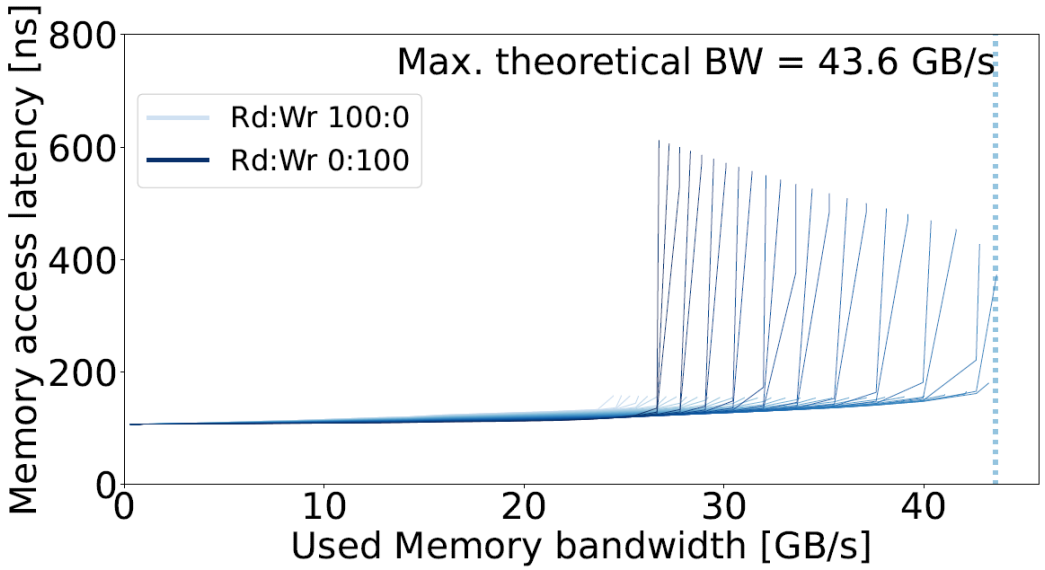
- Mess simulation possible **as soon as the bandwidth-latency curves are available**

Mess Simulator: CXL Memory Expander

- Mess simulation possible **as soon as the bandwidth-latency curves are available**
  - Measured on the actual hardware: Amazon cloud, Alibaba cloud, …

Mess Simulator: CXL Memory Expander

- Mess simulation possible **as soon as the bandwidth-latency curves are available**
  - Measured on the actual hardware: Amazon cloud, Alibaba cloud, …
  - Provided by device manufacturers
    - Bandwidth-latency curves give us **everything we need**; expose **nothing delicate or confidential**
    - Example: Micron's SystemC model of CXL memory expander



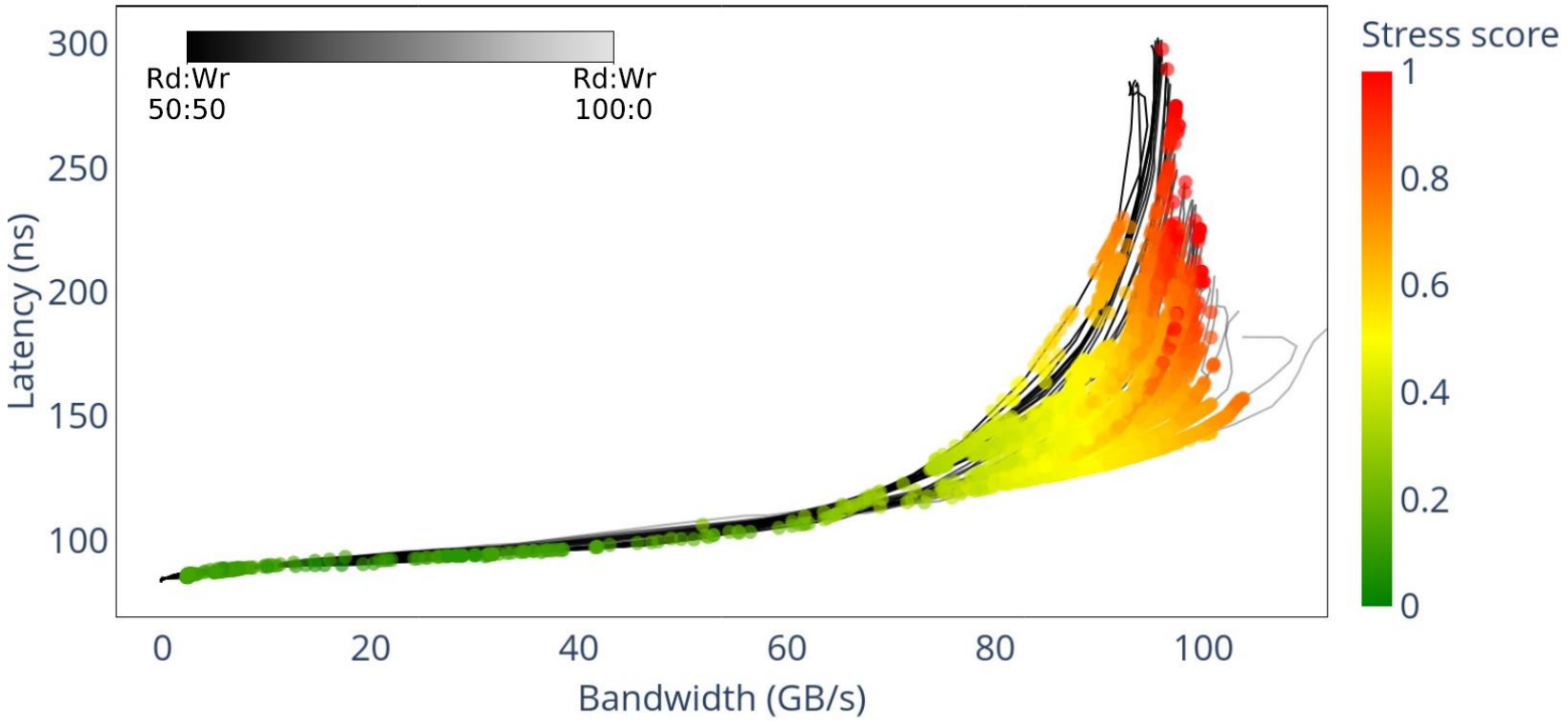
Manufacturer's SystemC model

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

Application memory-related profiling

- → Paper & Poster



24-core Intel Skylake CPU with 6x DDR4-2666.
HPCG. Sampling frequency: 10 ms.

Impact

- Mess framework is **publicly-released** and ready to be used by the community
  - Keep adding material: Actual CXL memory expanders

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron®

Impact

- Mess framework is **publicly-released** and ready to be used by the community
  - Keep adding material: Actual CXL memory expanders

- Mess simulation possible **as soon as the bandwidth-latency curves are available**

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

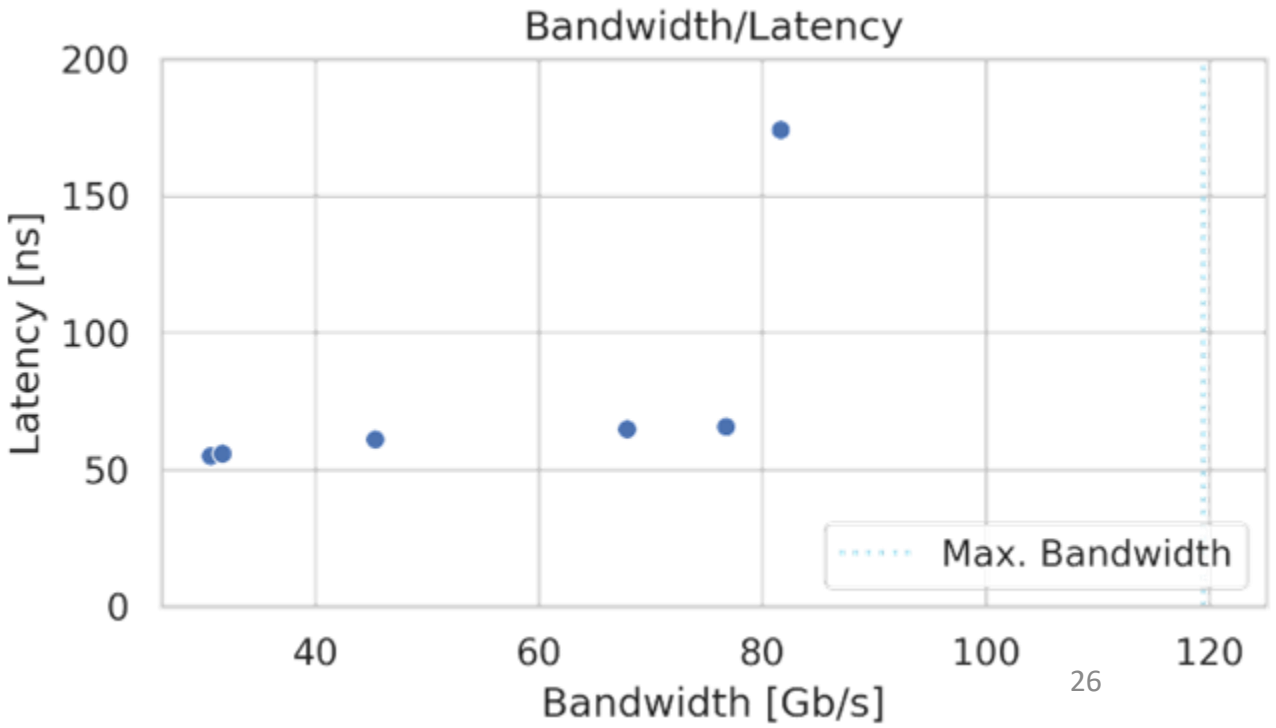
micron®

Impact

- Mess framework is **publicly-released** and ready to be used by the community
  - Keep adding material: Actual CXL memory expanders

- Mess simulation possible **as soon as the bandwidth-latency curves are available**

- **Significant uptake by the community**
  - Discussing benchmarking and integration
    - DRAMsim3, **DRAMSys**
    - gem5, gems, ChampSim
    - (Cache-aware) Roofline

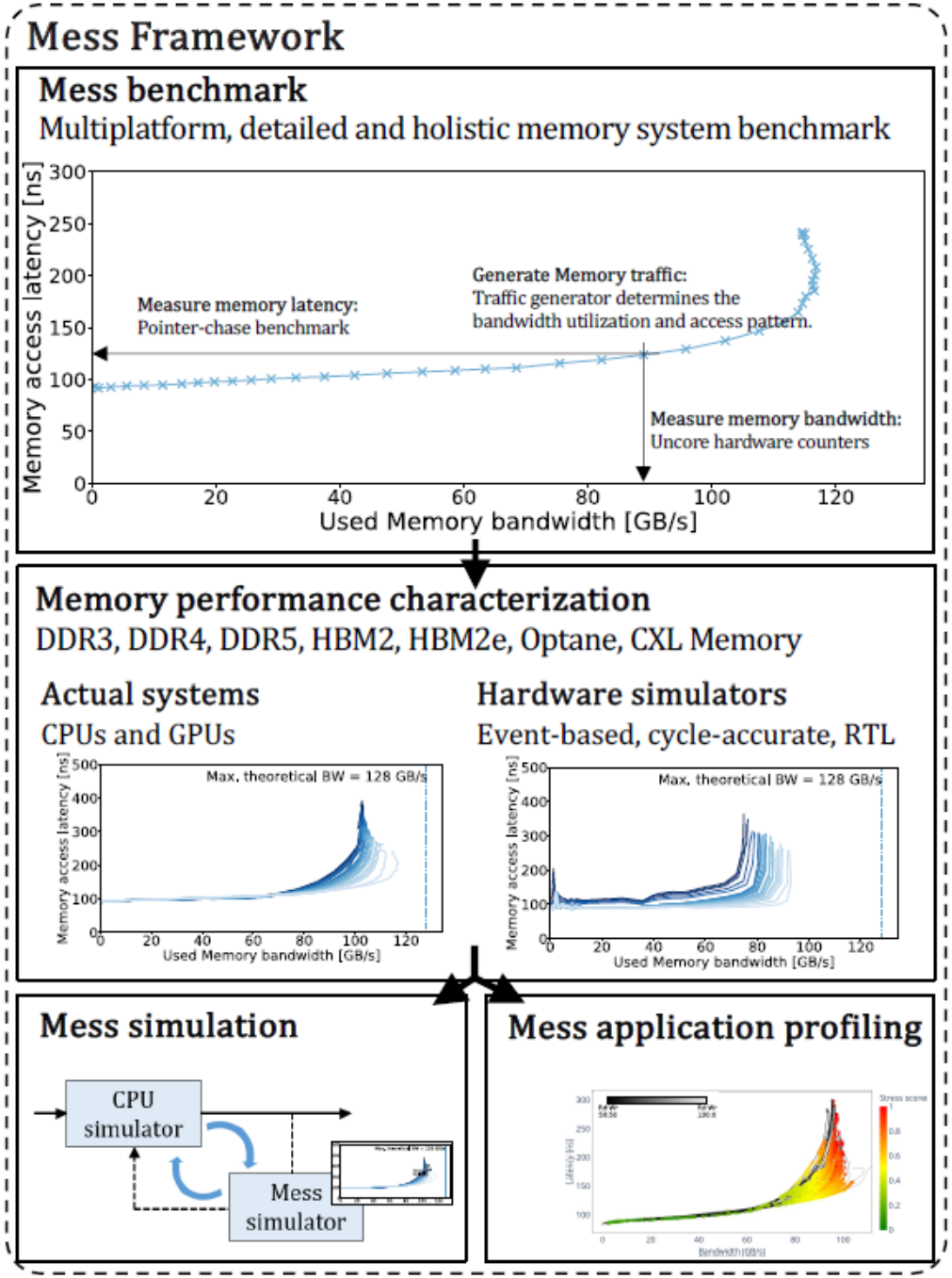


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

Let's talk!

- *Email:*
  - *pouya.esmaili@bsc.es*
  - *petar.radojkovic@bsc.es*
- *@MICRO-57*
  - *Anytime*
  - *Poster session:* ***Tue, 11-12h***



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Thank you!

petar.radojkovic@bsc.es

Large simulation errors: User setup?

Large simulation errors: User setup?

- Skilled users:
  - Zsim@BSC: 7-year experience. Zsim+DRAMSim2/3 interfaces with U. Maryland.

Rethinking Cycle Accurate DRAM Simulation

Shang Li
shangli@umd.edu
University of Maryland, College Park

Rommel Sánchez Verdejo
rommel.sanchez@bsc.es
Barcelona Supercomputing Center (BSC)
Universitat Politécnica de Catalunya (UPC)
Spain

Petar Radojković
Barcelona Supercomputing Center (BSC)
Barcelona, Spain

Bruce Jacob
blj@umd.edu
University of Maryland, College Park

Large simulation errors: User setup?

- Skilled users:
  - Zsim@BSC: 7-year experience. Zsim+DRAMSim2/3 interfaces with U. Maryland.
  - gem5 @ BSC: Arm CoE, European Processor Initiative

Rethinking Cycle Accurate DRAM Simulation

Shang Li
shangli@umd.edu
University of Maryland, College Park

Rommel Sánchez Verdejo
rommel.sanchez@bsc.es
Barcelona Supercomputing Center (BSC)
Universitat Politécnica de Catalunya (UPC)
Spain

Petar Radojković
Barcelona Supercomputing Center (BSC)
Barcelona, Spain

Bruce Jacob
blj@umd.edu
University of Maryland, College Park

The gem5 Simulator: Version 20.0+*

A new era for the open-source computer architecture simulator

Jason Lowe-Power, Abdul Mutaal Ahmad, Ayaz Akram, Mohammad Alian, Rico Amslinger, Matteo Andreozzi, Adrià Armejach, Nils Asmussen, Brad Beckmann, Srikant Bharadwaj, Gabe Black, Gedare Bloom, Bobby R. Bruce, Daniel Rodrigues Carvalho, Jeronimo Castrillon, Lizhong Chen, Nicolas Derumigny, Stephan Diestelhorst, Wendy Elsasser, Carlos Escuin, Marjan Fariborz, Amin Farmahini-Farahani, Pouya Fotouhi, Ryan Gambord, Jayneel Gandhi, Dibakar Gope, Thomas Grass, Anthony Gutierrez, Bagus Hanindhito, Andreas Hansson, Swapnil Haria, Austin Harris, Timothy Hayes, Adrian Herrera, Matthew Horsnell, Syed Ali Raza Jafri, Radhika Jagtap, Hanhwi Jang, Reiley Jeyapaul, Timothy M. Jones, Matthias Jung, Subash Kannoth, Hamidreza Khaleghzadeh, Yuetsu Kodama, Tushar Krishna, Tommaso Marinelli, Christian Menard, Andrea Mondelli, Miquel Moreto, Tiago Mück, Omar Naji, Krishnendra Nathella, Hoa Nguyen, Nikos Nikoleris, Lena E. Olson, Marc Orr, Binh Pham, Pablo Prieto, Trivikram Reddy, Alec Roelke, Mahyar Samani, Andreas Sandberg, Javier Setoain, Boris Shingarov, Matthew D. Sinclair, Tuan Ta, Rahul Thakur, Giacomo Travaglini, Michael Upton, Nilay Vaish, Ilias Vougioukas, William Wang, Zhengrong Wang, Norbert Wehn, Christian Weis, David A. Wood, Hongil Yoon, Éder F. Zulian †

Large simulation errors: User setup?

- Skilled users:
  - Zsim@BSC: 7-year experience. Zsim+DRAMSim2/3 interfaces with U. Maryland.
  - gem5 @ BSC: Arm CoE, European Processor Initiative

Rethinking Cycle Accurate DRAM Simulation

Shang Li
shangli@umd.edu
University of Maryland, College Park

Rommel Sánchez Verdejo
rommel.sanchez@bsc.es
Barcelona Supercomputing Center (BSC)
Universitat Politécnica de Catalunya (UPC)
Spain

Petar Radojković
Barcelona Supercomputing Center (BSC)
Barcelona, Spain

Bruce Jacob
blj@umd.edu
University of Maryland, College Park

The gem5 Simulator: Version 20.0+*

A new era for the open-source computer architecture simulator

Jason Lowe-Power, Abdul Mutaal Ahmad, Ayaz Akram, Mohammad Alian, Rico Amslinger, Matteo Andreozzi, Adrià Armejach, Nils Asmussen, Brad Beckmann, Srikant Bharadwaj, Gabe Black, Gedare Bloom, Bobby R. Bruce, Daniel Rodrigues Carvalho, Jeronimo Castrillon, Lizhong Chen, Nicolas Derumigny, Stephan Diestelhorst, Wendy Elsasser, Carlos Escuin, Marjan Fariborz, Amin Farmahini-Farahani, Pouya Fotouhi, Ryan Gambord, Jayneel Gandhi, Dibakar Gope, Thomas Grass, Anthony Gutierrez, Bagus Hanindhito, Andreas Hansson, Swapnil Haria, Austin Harris, Timothy Hayes, Adrian Herrera, Matthew Horsnell, Syed Ali Raza Jafri, Radhika Jagtap, Hanhwi Jang, Reiley Jeyapaul, Timothy M. Jones, Matthias Jung, Subash Kannoth, Hamidreza Khaleghzadeh, Yuetsu Kodama, Tushar Krishna, Tommaso Marinelli, Christian Menard, Andrea Mondelli, Miquel Moreto, Tiago Mück, Omar Naji, Krishnendra Nathella, Hoa Nguyen, Nikos Nikoleris, Lena E. Olson, Marc Orr, Binh Pham, Pablo Prieto, Trivikram Reddy, Alec Roelke, Mahyar Samani, Andreas Sandberg, Javier Setoain, Boris Shingarov, Matthew D. Sinclair, Tuan Ta, Rahul Thakur, Giacomo Travaglini, Michael Upton, Nilay Vaish, Ilias Vougioukas, William Wang, Zhengrong Wang, Norbert Wehn, Christian Weis, David A. Wood, Hongil Yoon, Éder F. Zulian †

- We knew that one day we will present these charts → Very, very tedious setup

What happens when we talk

- Panel discussion at MEMSYS 2024

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron®

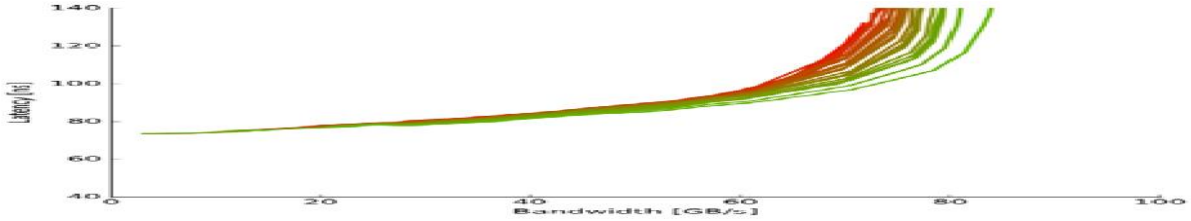
What happens when we talk

- Panel discussion at MEMSYS 2024
  - Bruce Jacob Re:DRAMsim3
    “Did you try DRAMsim2?
    It should be much better.”

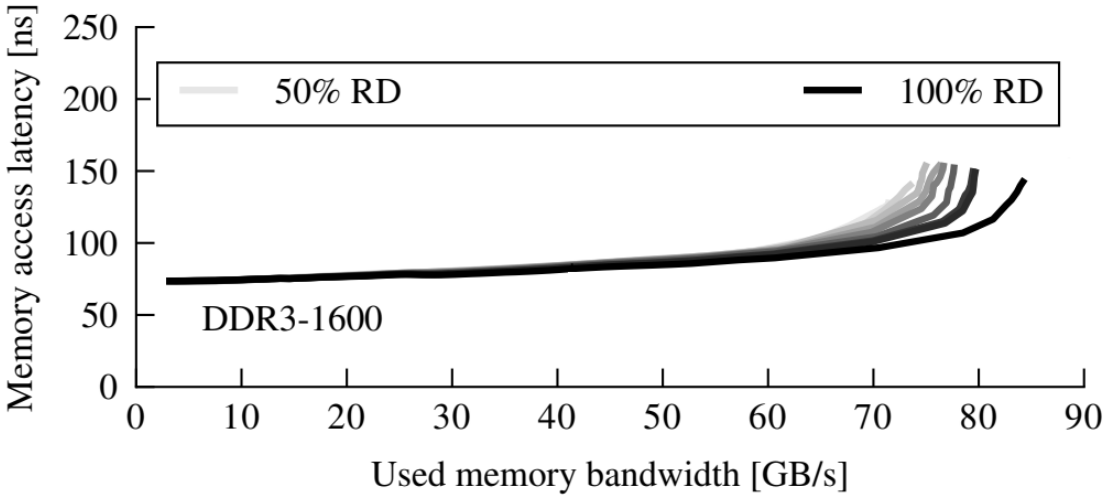
What happens when we talk

- Panel discussion at MEMSYS 2024
  - Bruce Jacob Re:DRAMsim3
    “Did you try DRAMsim2?
    It should be much better.”

DRAMSim2. Trace-driven. 6xDDR3-1600



Actual Intel Sandy Bridge with 6xDDR3-1600



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

micron