

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

PROFET: No-stress Performance Prediction

Mariana Carmin

Barcelona Supercomputing Center

It's the Memory, Stupid!

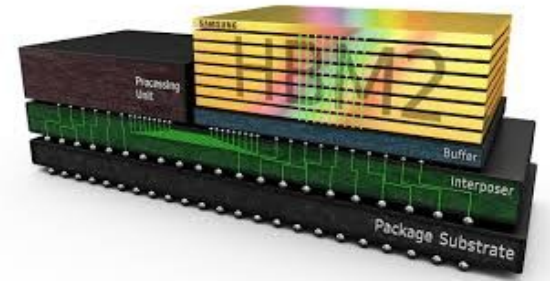
31st of October 2026.

How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options

How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options



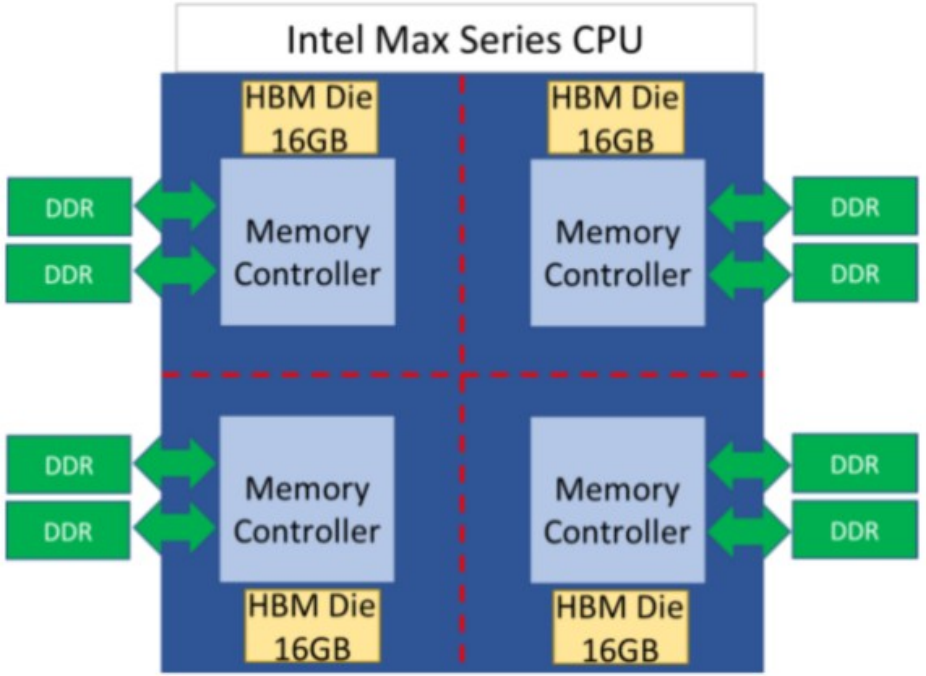


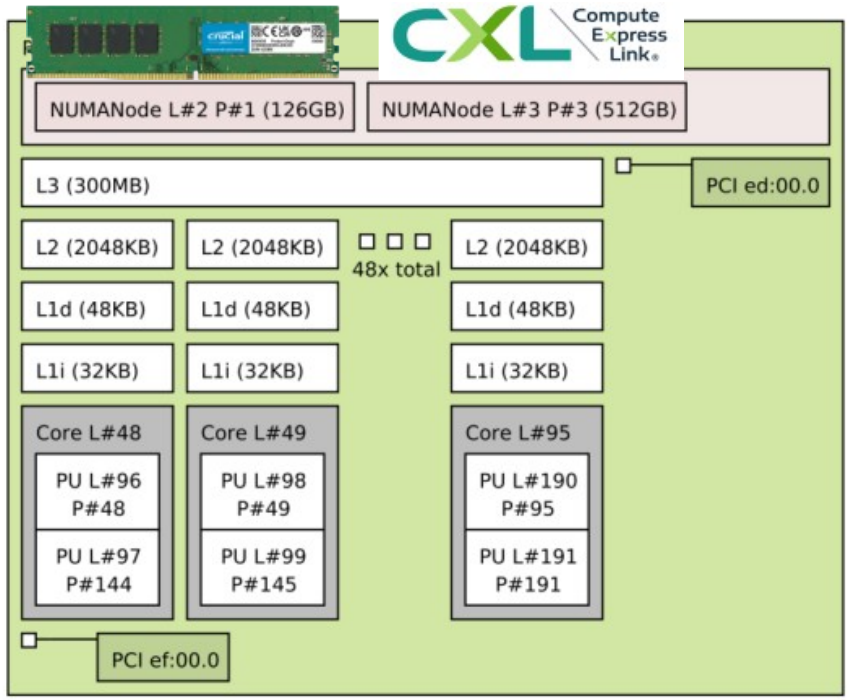
CXL Link®



How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options





How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options

- Conventional approach: Detailed hardware simulation (Zsim + DRAMSim2)

How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options

- Conventional approach: Detailed hardware simulation (Zsim + DRAMSim2)
  - Frustration
    - Detailed hardware simulation
    - Time consuming
    - We had to simulate an existing CPU **only** to drive our memory system simulation

  - Idea: ***There has to be a better way to do this!***

Published in 2019 – Still relevant

- First big tool developed by the BSC memory team
- Used by various industrial partners: **Samsung, Micron, Huawei**
- PROFET passed the test of time
  - Tool/method developed for SandyBridge with DDR3
  - Plug&Play
    - Intel Knights Landing (DDR4, MCDRAM), Cascade Lake (DDR4), Sapphire Rapids (DDR5), Emerald Rapids (DDR5, CXL memory expanders), Granite Rapids (DDR5 RDIMMs and MRDIMMs)
    - Huawei Kunpeng 920 (DDR4)
  - Led to the development of the Mess framework

Milan Radulovic, Rommel Sánchez Verdejo, Paul Carpenter, Petar Radojković, Bruce Jacob, and Eduard Ayguadé. ***PROFET: Modeling System Performance and Energy Without Simulating the CPU***. SIGMETRICS 2019.

PROFET: Memory system design space exploration

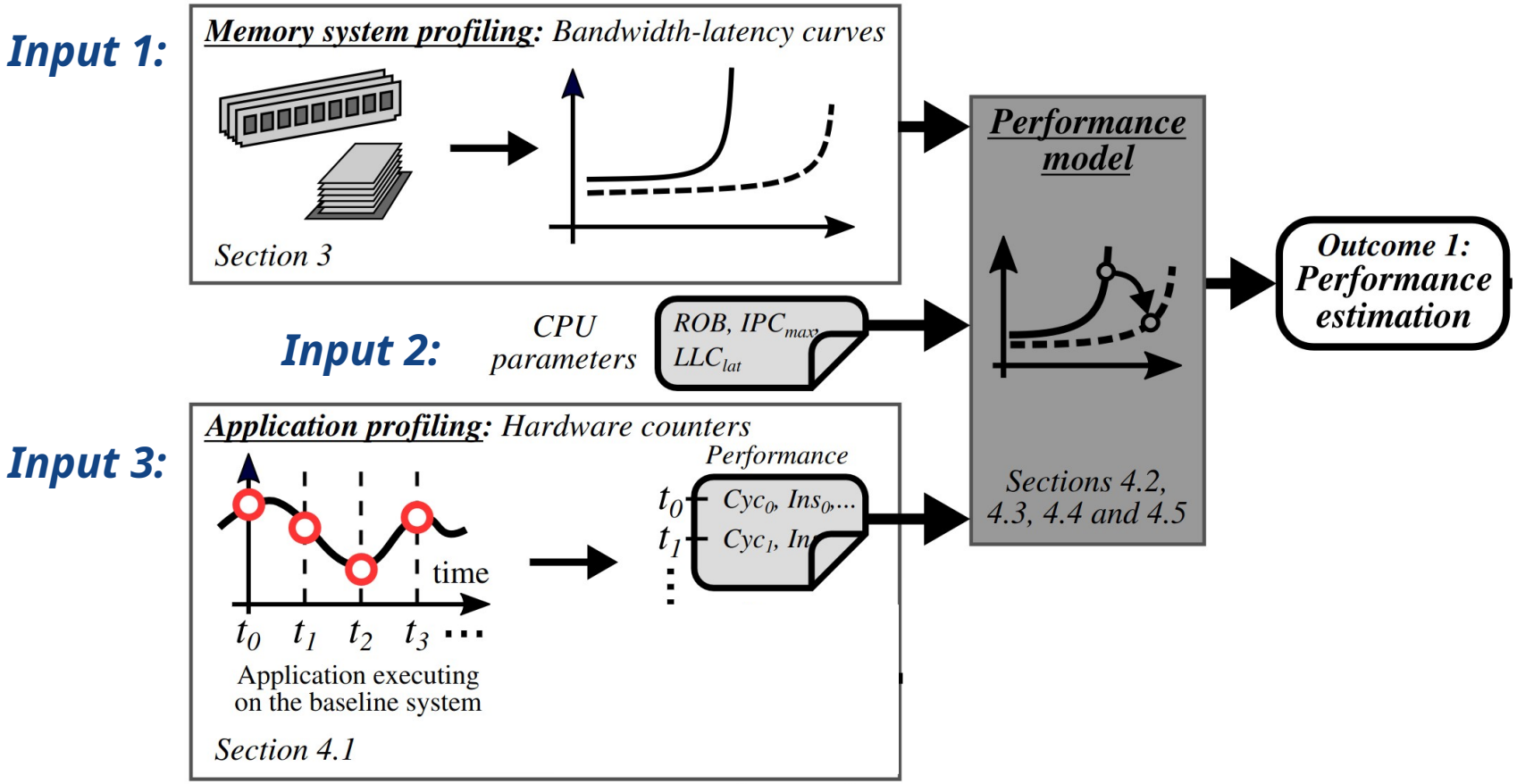
- Analytical model that predicts how your application performance, power and energy change if you change the main memory

- Answering the question:
  "How much will performance, power and energy of my application change if we change the main memory?"

- We profile an application on a **baseline system**
  - Baseline system is an **existing server** (with existing memory system), e.g. Intel GNR with RDIMM-6400

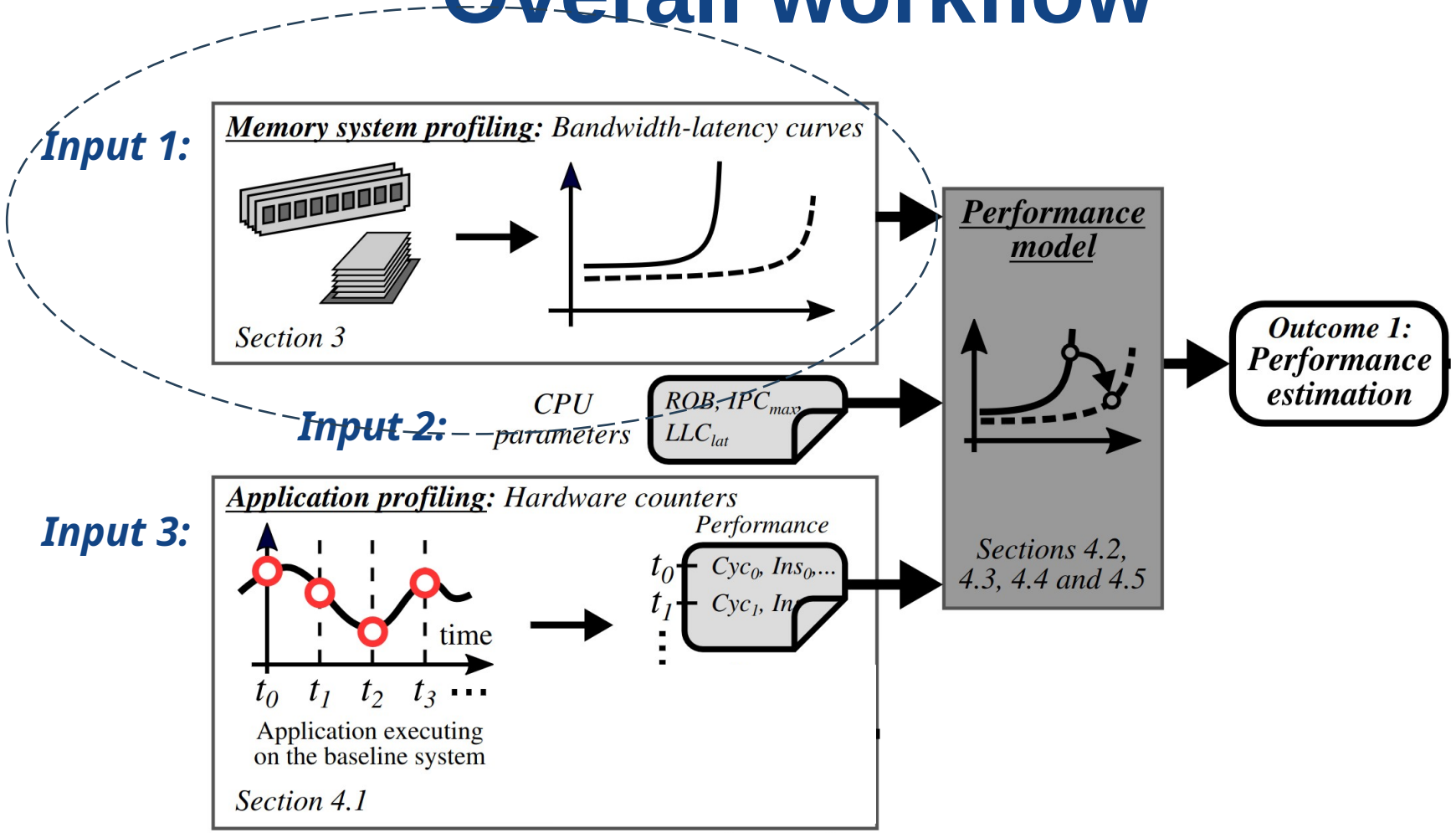
- **Target system**
  - The **CPU remains the same**, but we want to explore the performance, power and energy **implications of a new memory system on my application**:
    - E.g. How much would my application benefit from the RDIMM-6400 → MRDIMM-8800 upgrade of the server

Overall workflow



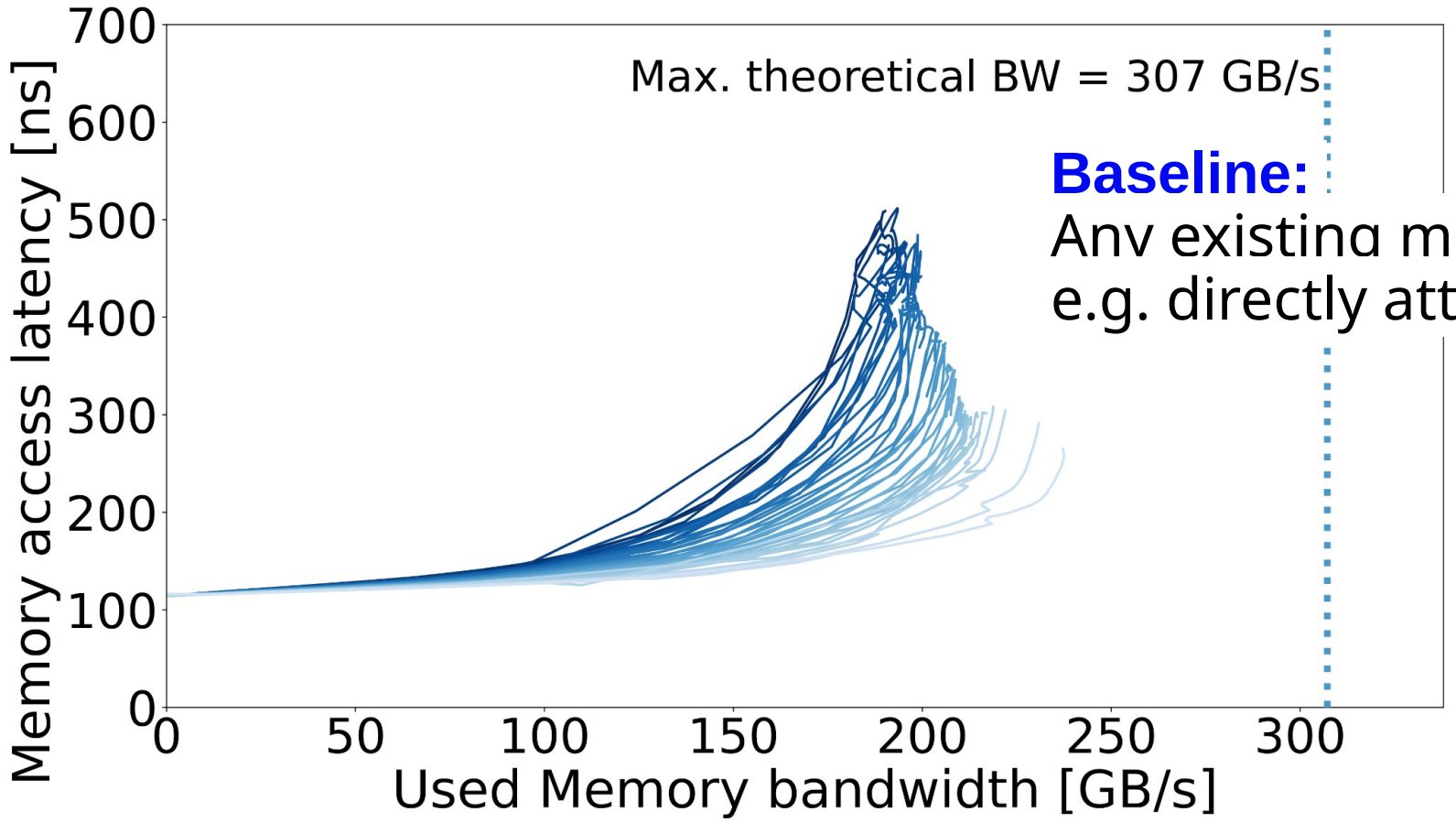
Note: In this presentation we focus on the performance estimate. For power and energy estimate, please refer to the PROFET paper (SIGMETRICS 2019).

Overall workflow



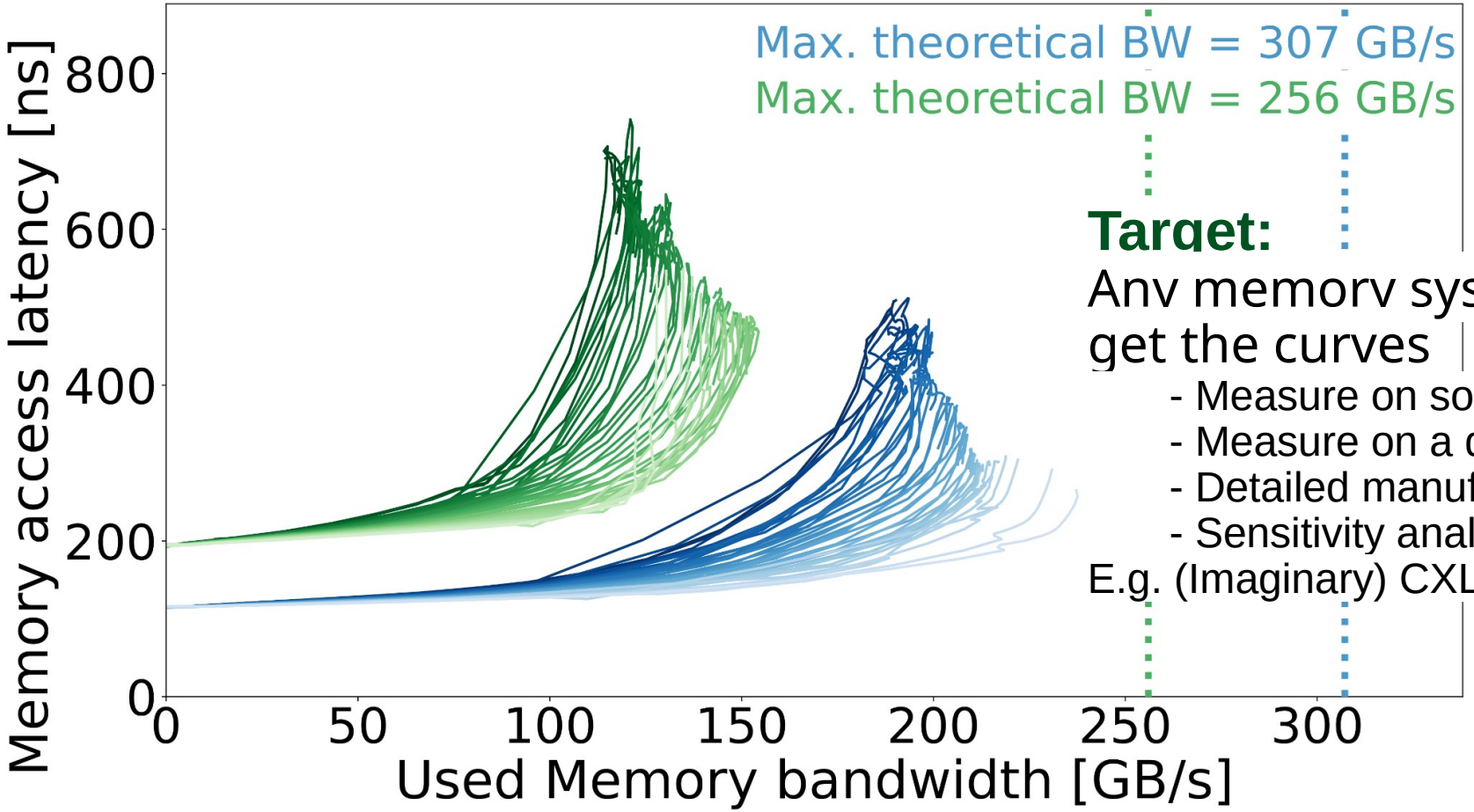
Note: In this presentation we focus on the performance estimate. For power and energy estimate, please refer to the PROFET paper (SIGMETRICS 2019).

Input 1:
Memory system profiling: Bandwidth–latency curves



Baseline:
Any existing memory system
e.g. directly attached DDR4/5

Input 1:
Memory system profiling: Bandwidth–latency curves



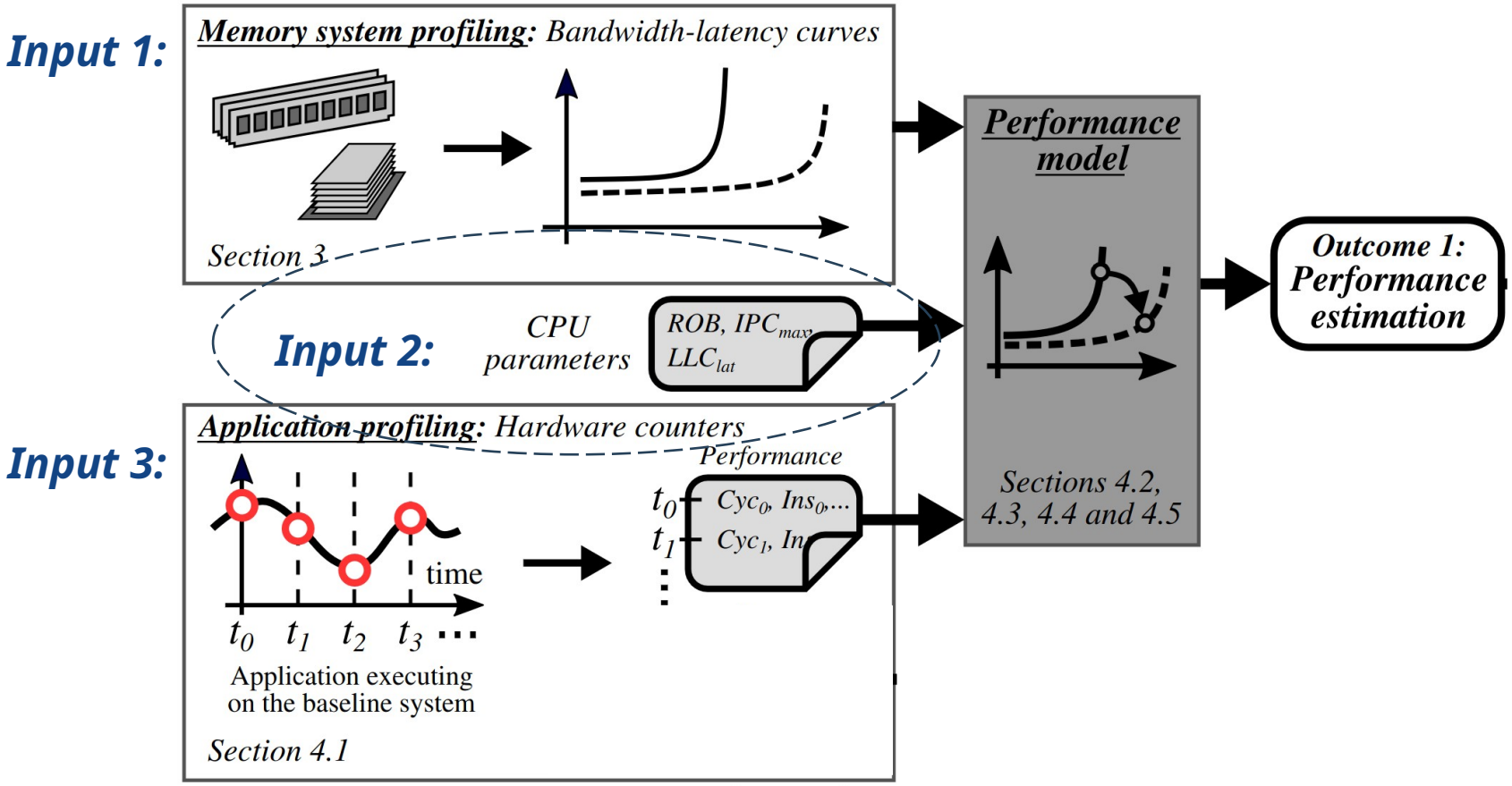
Target:

Any memory system for which you can get the curves

- Measure on some actual system
- Measure on a dev-board or prototype
- Detailed manufacturers simulation/estimate
- Sensitivity analysis

E.g. (Imaginary) CXL memory expander

Overall workflow



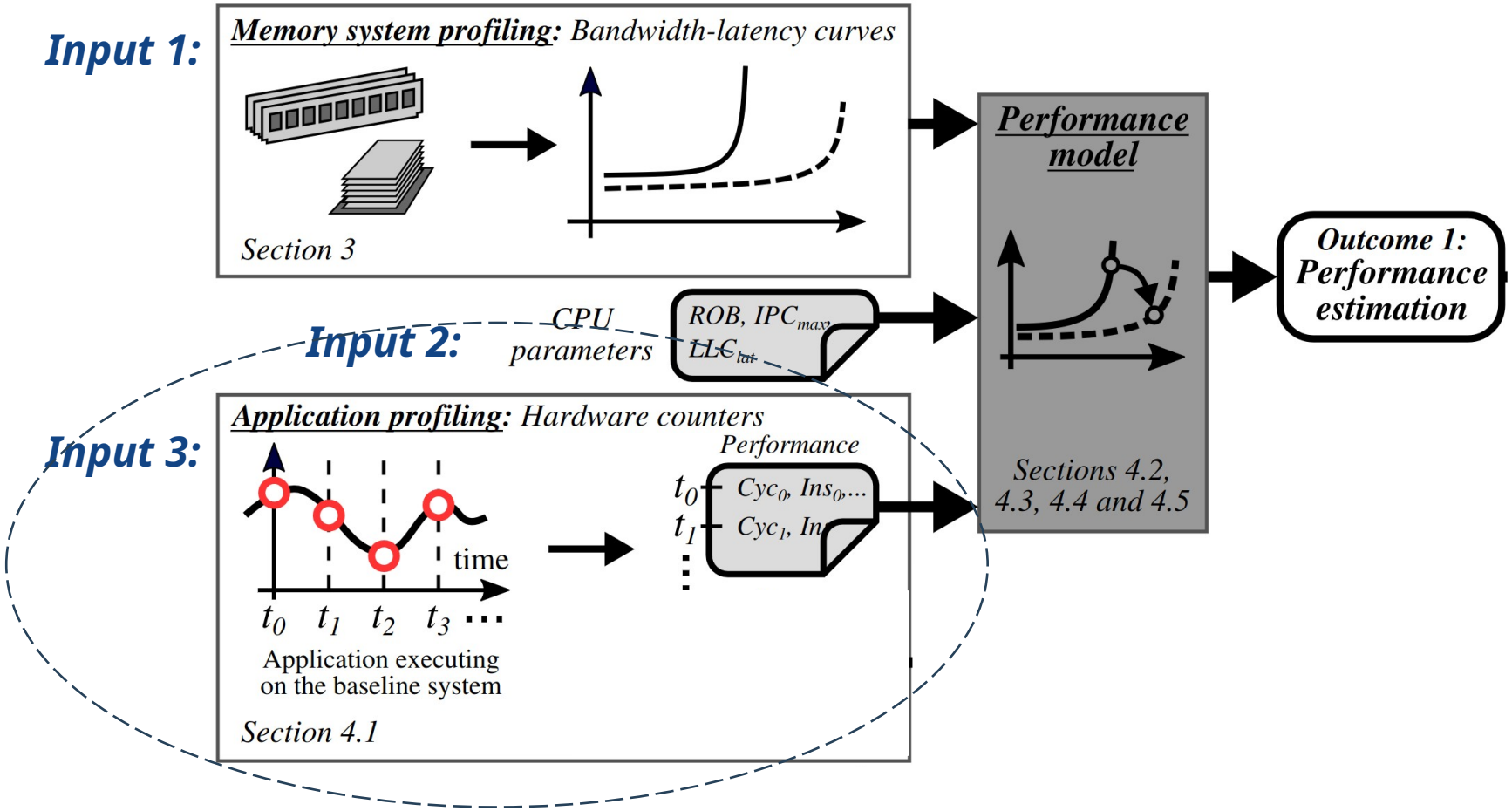
Note: In this presentation we focus on the performance estimate. For power and energy estimate, please refer to the PROFET paper (SIGMETRICS 2019).

Input 2: CPU parameters

- Some (basic) CPU parameters: Typically available in the CPU datasheets
  - Re-order buffer (ROB) capacity
  - Miss information status holding register(MSHR) capacity
  - Minimum theoretical cycles-per-instruction (CPI)
  - Latency of last level cache (LLC)

```
{
    "Ins_ROB": 224,
    "IPC_max": 6,
    "Lat_LLC": 50,
    "MSHR": 10
}
```

Overall workflow



Note: In this presentation we focus on the performance estimate. For power and energy estimate, please refer to the PROFET paper (SIGMETRICS 2019).

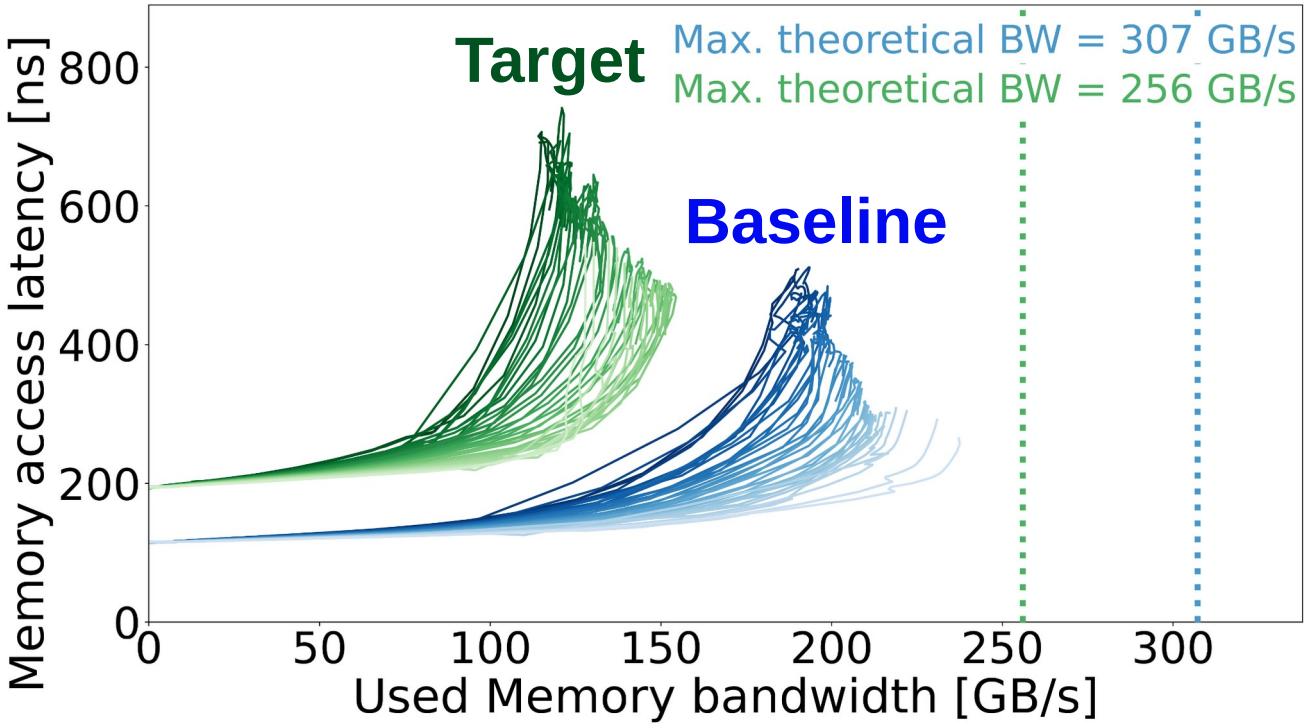
Input 3: Application profiling

- Done on the baseline (actual) memory system, e.g. Intel GRP with RDIMM-6400

- Execute application of the system and read hardware counters
  - Read and write memory bandwidths → The same as the Mess application profiling
  - CPU cycles
  - Number of instructions
  - Number of LLC misses

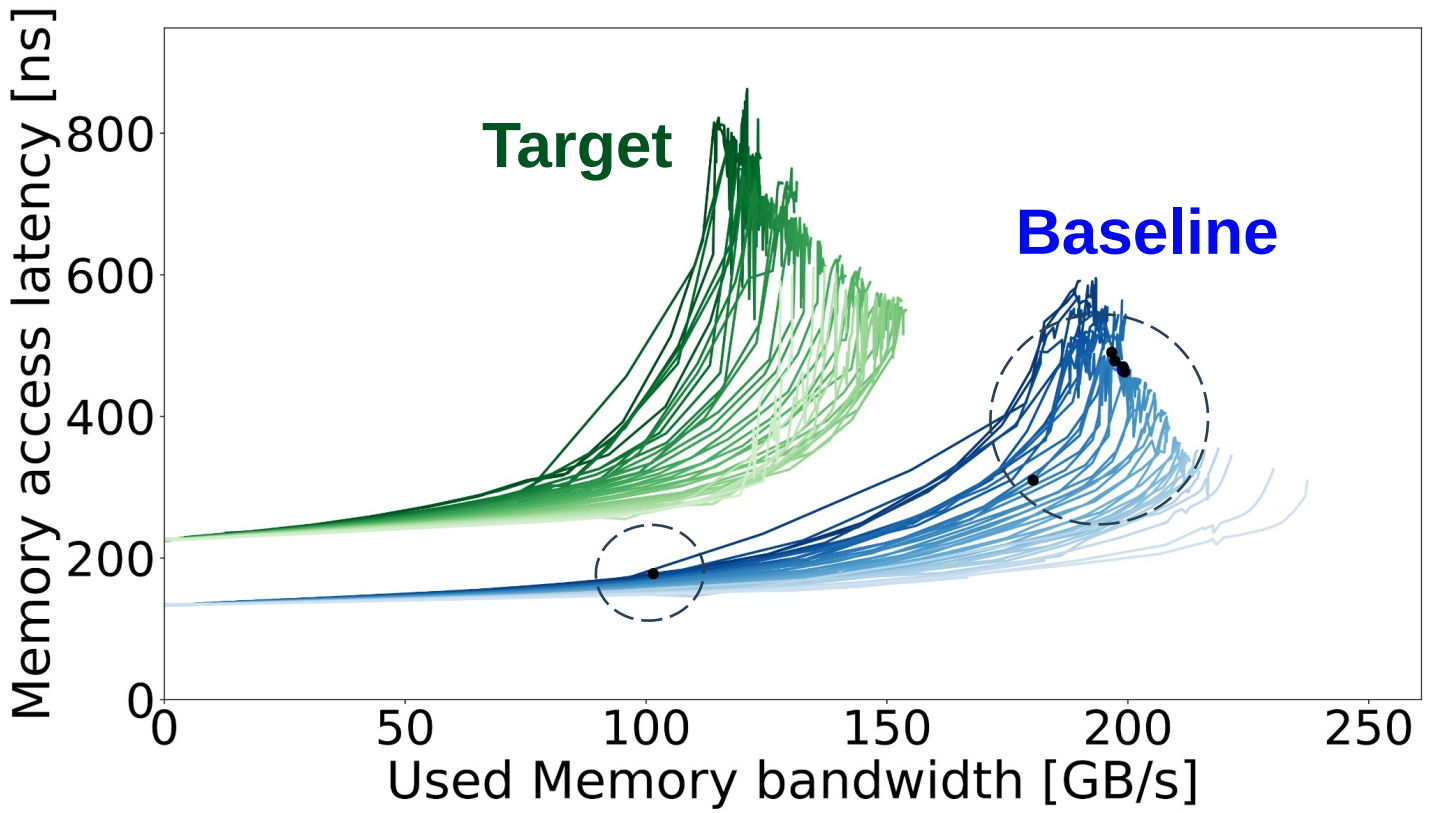
- Sampling in time
  - Reasonable sampling interval: >10ms

Analysis Step 1: Position your app on the curves



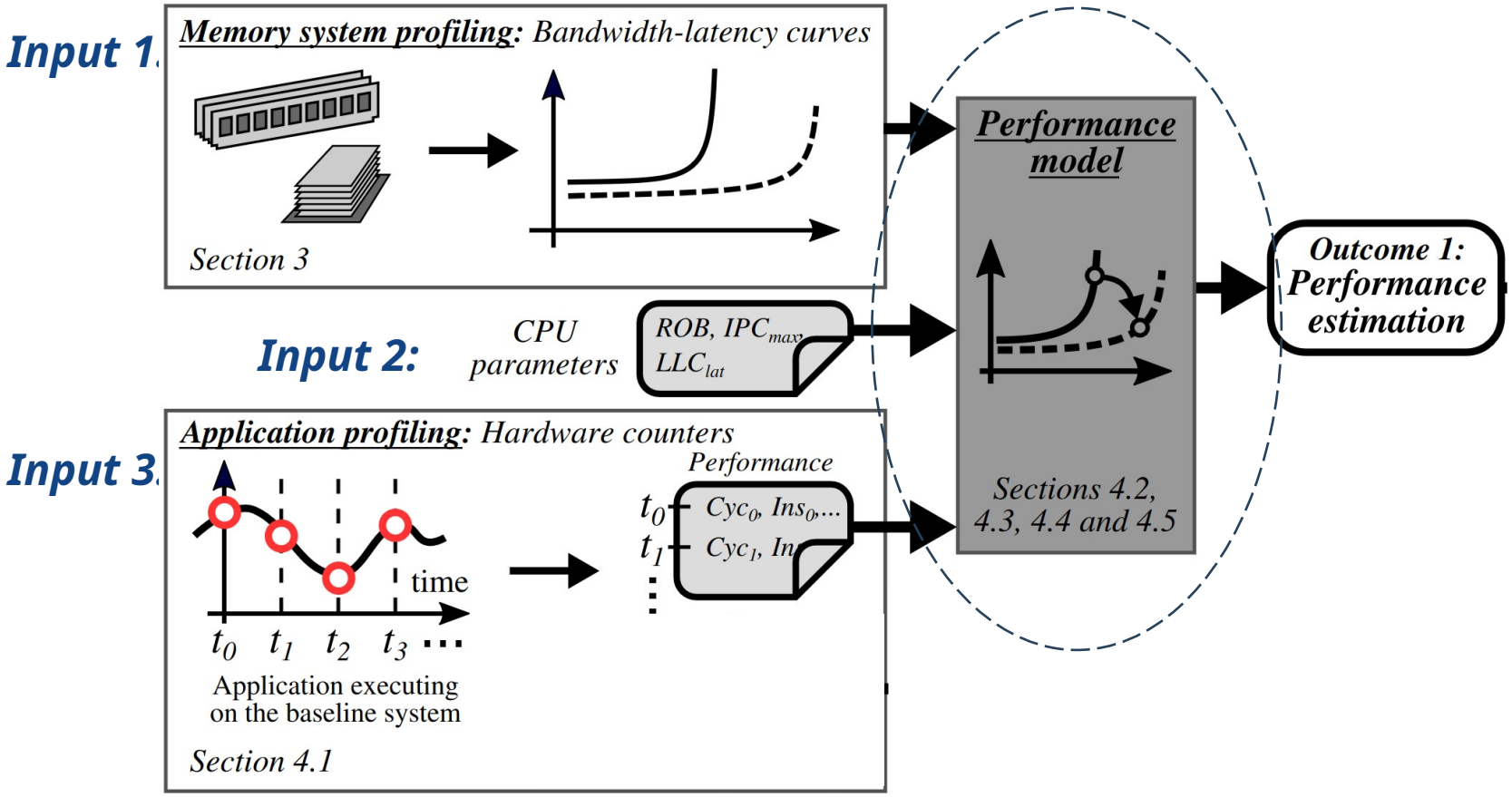
Each point is one application observation

Analysis Step 1: Position your app on the curves



Each point is one application observation

Overall workflow

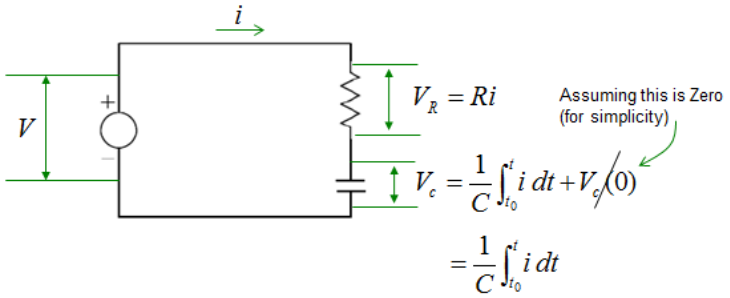


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

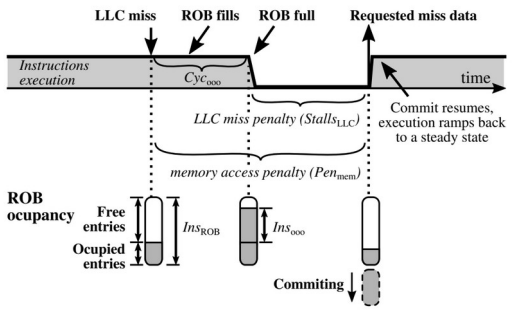
Note: In this presentation we focus on the performance estimate. For power and energy estimate, please refer to the PROFET paper (SIGMETRICS 2019).

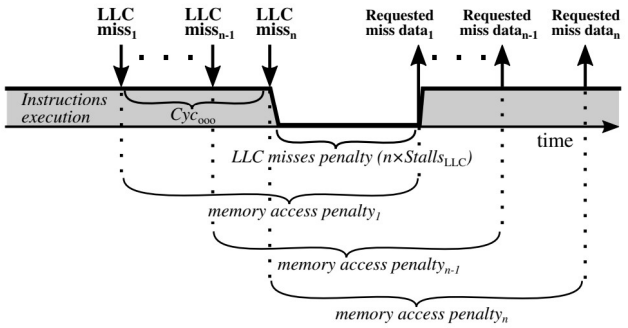
PROFET performance model

- **Old-school electrical engineering mindset**
  - Model a circuit with a system of equations → Solve the system



- **Unconventional approach** to memory design-space exploration





4.4 Performance as a function of latency

This section completes the analysis of out-of-order processor performance as a function of latency. We start by repeating Eq. 4, which gives the predicted CPI in terms of $Stalls_{LLC}$:

$$CPI_{tot}^{(2)} = CPI_{tot}^{(1)} + \frac{Miss_{LLC}}{Ins_{tot}} \times \left(Stalls_{LLC}^{(2)} - Stalls_{LLC}^{(1)}\right) \quad \text{(4 again)}$$

As remarked at the beginning of Section 4.3, in comparison with an in-order processor, an out-of-order processor has a more complex expression for $Stalls_{LLC}$, and this was given in Eq. 12:

$$Stalls_{LLC} = \frac{1}{MLP} \times (Pen_{mem} - CPI_0 \times Ins_{ooo}) \quad \text{(12 again)}$$

Finally we replace the MLP parameter in this equation with the point estimate in Eq. 15:

$$\widehat{MLP}(Ins_{ooo}) = \frac{Miss_{LLC}}{Ins_{tot}} \times Ins_{ooo} + 1 \quad \text{(15 again)}$$

In fact, as explained in Section 4.3.3, this value is restricted to lie between the lower and upper bounds given in that section. For the sake of clarity, we consider the more common case for which it is not necessary.

Combining Eq. 4, Eq.12 and Eq.15, and assuming that Ins_{tot}, $Miss_{LLC}$, CPI_0, Ins_{ooo} and MLP do not change when moving from one memory system configuration to another, then $CPI_{tot}^{(2)}$ can be calculated as:

$$CPI_{tot}^{(2)} = CPI_{tot}^{(1)} + \frac{Pen_{mem}^{(2)} - Pen_{mem}^{(1)}}{Ins_{ooo} + Ins_{tot}/Miss_{LLC}} \quad (16)$$

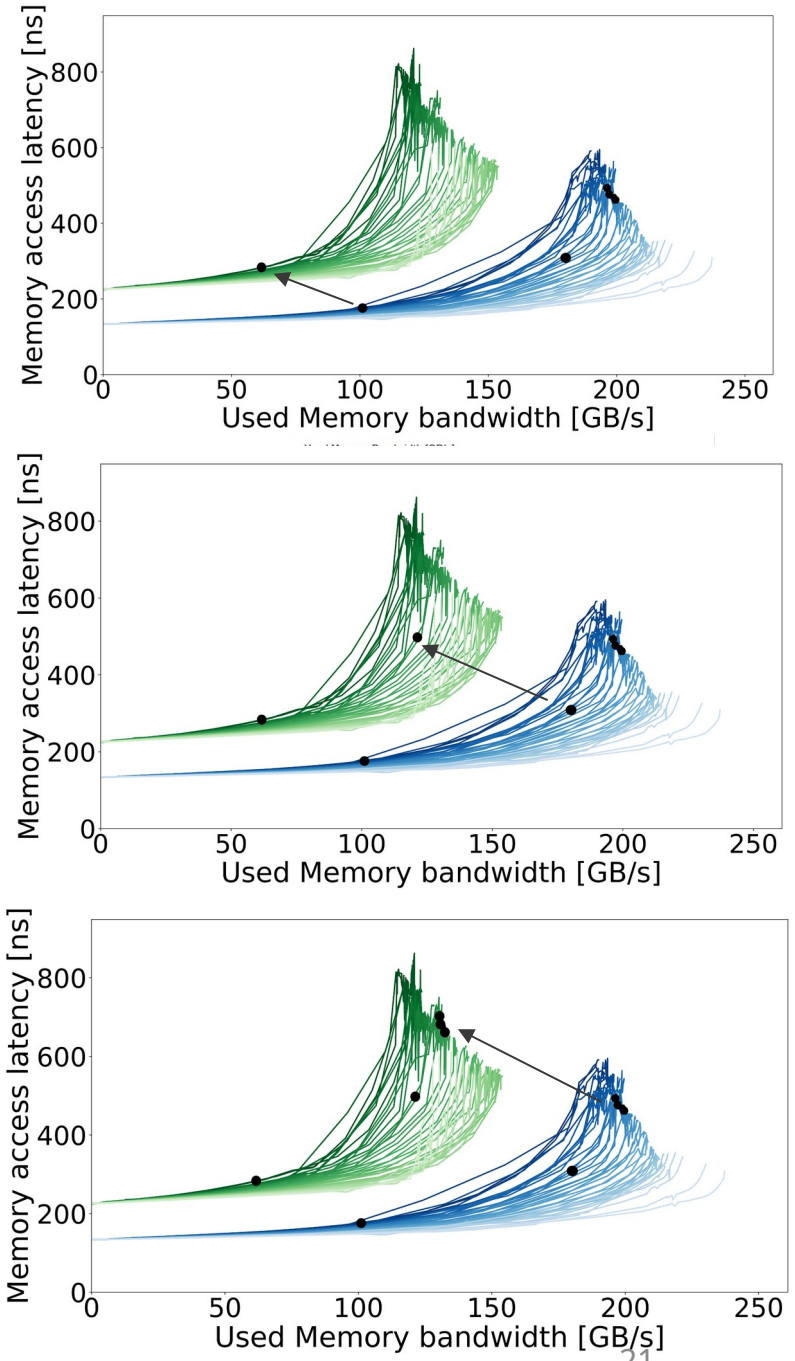
This equation is written in terms of the memory access penalty, Pen_{mem}, but at the system level, outside a detailed analysis of a particular processor's pipeline, only Lat_{mem} is relevant. Recall that Pen_{mem} was defined to be the memory access latency, Lat_{mem} minus the cost of an LLC hit. We note, therefore, that the expression $Pen_{mem}^{(2)} - Pen_{mem}^{(1)}$ is equal to $Lat_{mem}^{(2)} - Lat_{mem}^{(1)}$. Taking account of this and rewriting in terms of the IPC instead of the CPI gives:

$$IPC_{tot}^{(2)} = \frac{IPC_{tot}^{(1)}}{1 + IPC_{tot}^{(1)} \times \frac{Lat_{mem}^{(2)} - Lat_{mem}^{(1)}}{Ins_{ooo} + Ins_{tot}/Miss_{LLC}}} \quad (17)$$

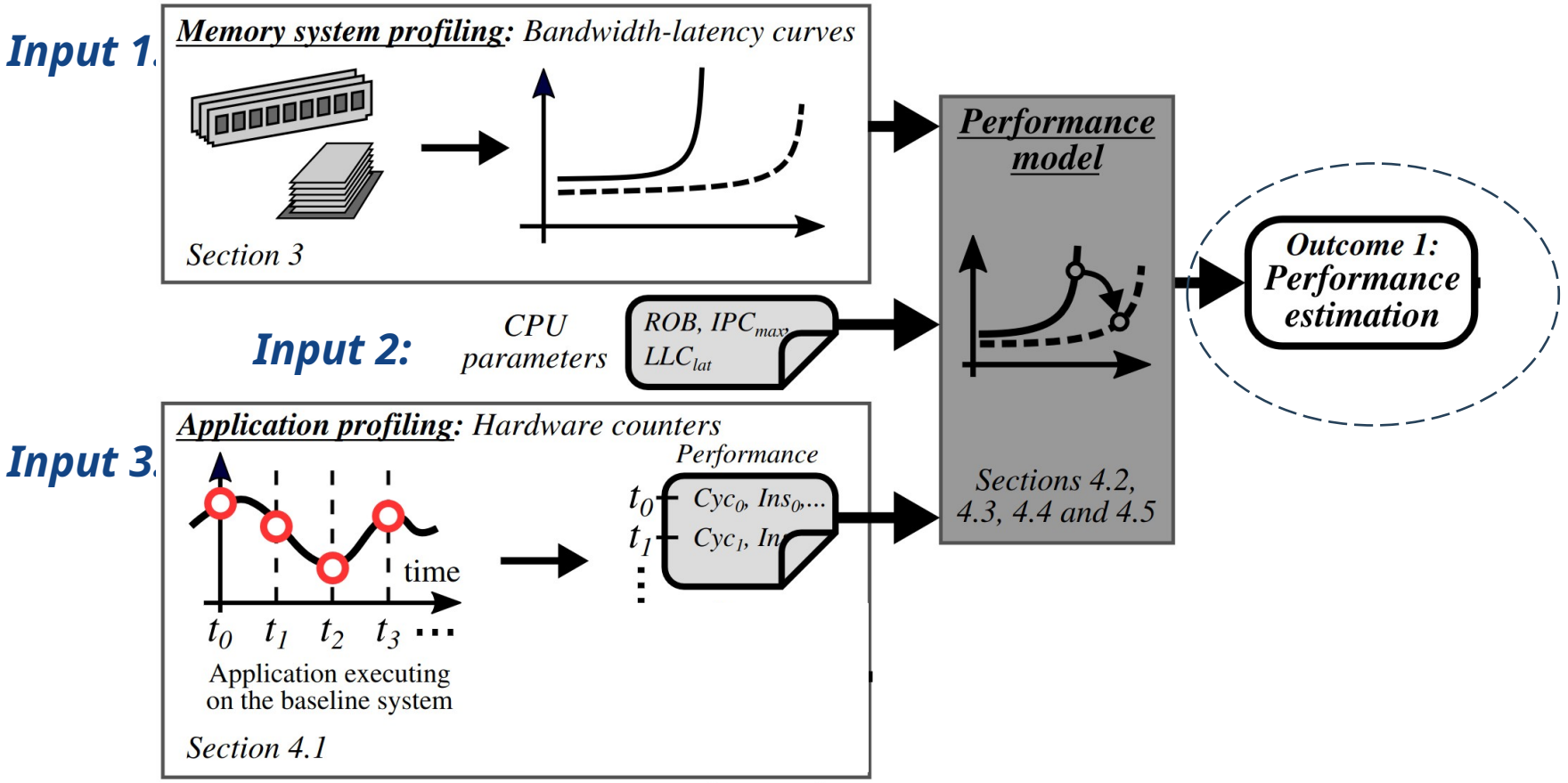
It's the Memory, Stupid!

PROFET predicts how application “moves” between the curves

- **Idea 1:**
  - Application can be positioned on the memory curves
  - I hope that by now we agree on this
- **Idea 2:**
  - We can predict (calculate) the application’s position
    on the target curves
    - Full system of equations → Paper & Git
    - In this presentation: Trust me that this can be done



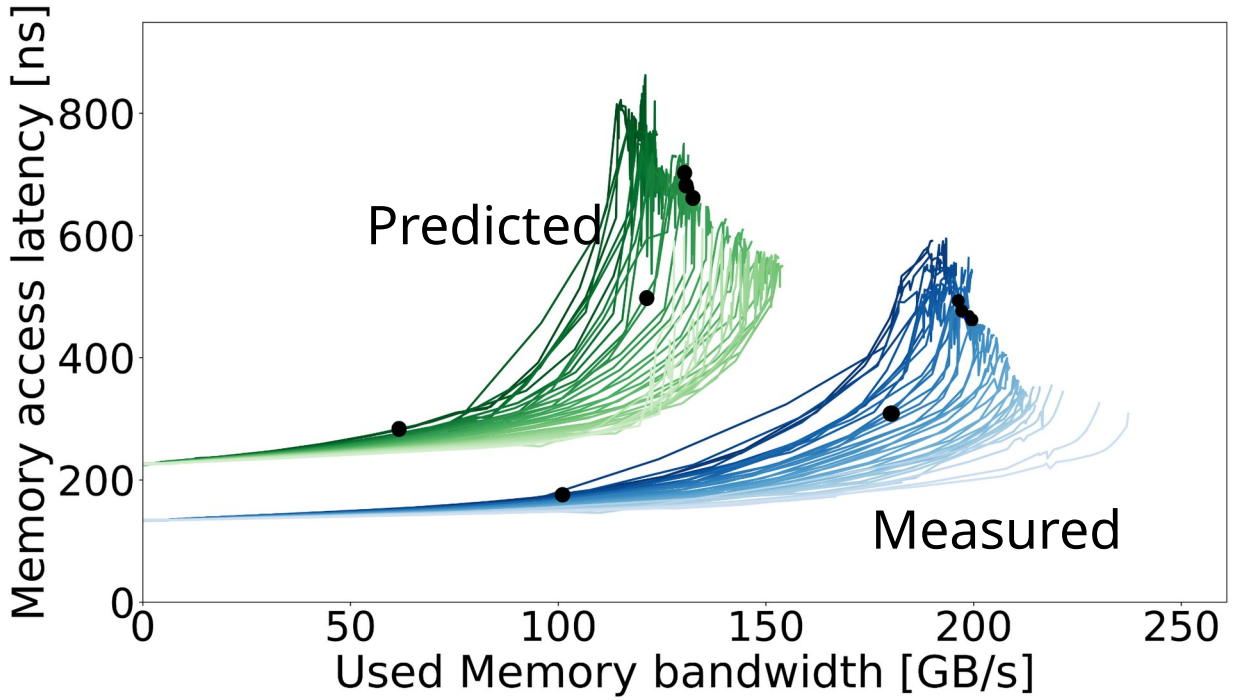
Overall workflow



Note: In this presentation we focus on the performance estimate. For power and energy estimate, please refer to the PROFET paper (SIGMETRICS 2019).

Point by point → Whole application

- **Visual view → Detailed & Intuitive**
  - Not available in current hardware simulators



- **Summary view**
  - Standard performance metrics:
    - IPC/CPI
    - Memory bandwidth, latency
    - …

IPC
CPI
BW
LAT

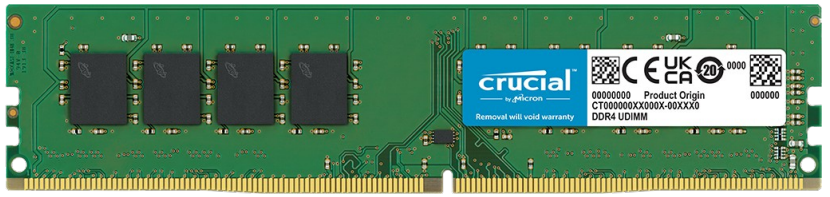
```
IPC base: 0.19
IPC target: 0.246
IPC change avg: 29.493%
IPC change min: 0.066
IPC change max: 0.064
CPI base: 5.269
CPI target: 4.069
CPI change avg: -22.776%
CPI change min: 0.038
CPI change max: 0.039
Lat target: 360.281
BW target: 624.234
BW change avg: 29.338%
BW change min: 29.274
BW change max: 29.398
CPI measured: 4.008
CPI change measured: -23.92
IPC change measured: 31.44
IPC change estimation_error: 1.482
CPI change estimation_error: 1.504
Lost points: 3.62%
```

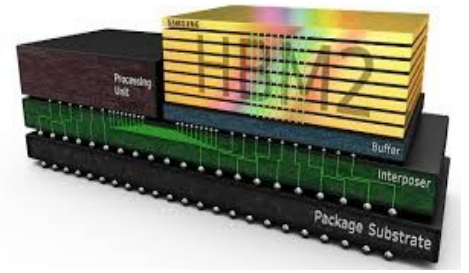
* Accuracy

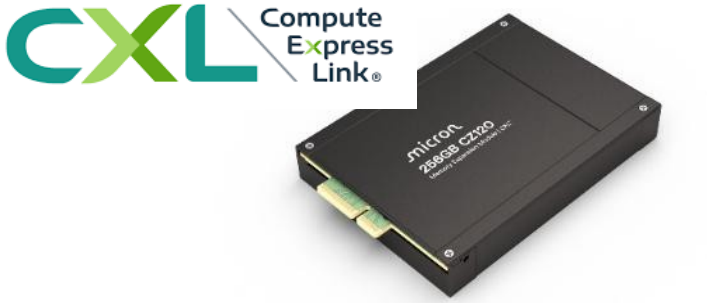
* If target data is measured

Brutal evaluation

- **Evaluated**
  - **Intel Sandy Bridge** *(DDR3:800/1066/1333/1600)*: Error 5%
  - **Intel Knights Landing** *(DDR4-2400/MCDRAM)*: Error 4%
  - **Huawei Kunpeng 920** *(DDR4 1600/1866/2933)*: Error 4.5%
  - **Intel Cascade lake**
    - *DDR4 alone → DDR4+Optane:* Error 0.7%
    - DDR4 2666→ 2133: Error 0.7%
    - DDR4 → Optane: Error 26%
  - **Intel Emerald Rapids** *(DDR5 4800 → 3200)*: Error 3%
  - **Intel Granite Rapids** *(RDIMM-6400 → MRDIMM-8800)*: Error 4%
- **Work in progress**
  - **Intel Max:** *DDR5-4800 + HBM2*
  - **Intel Emerald Rapids:** *DDR5-4800 + CXL Memory Expander*









Benchmarks: SPEC, HPC apps, memory intensive benchmarks

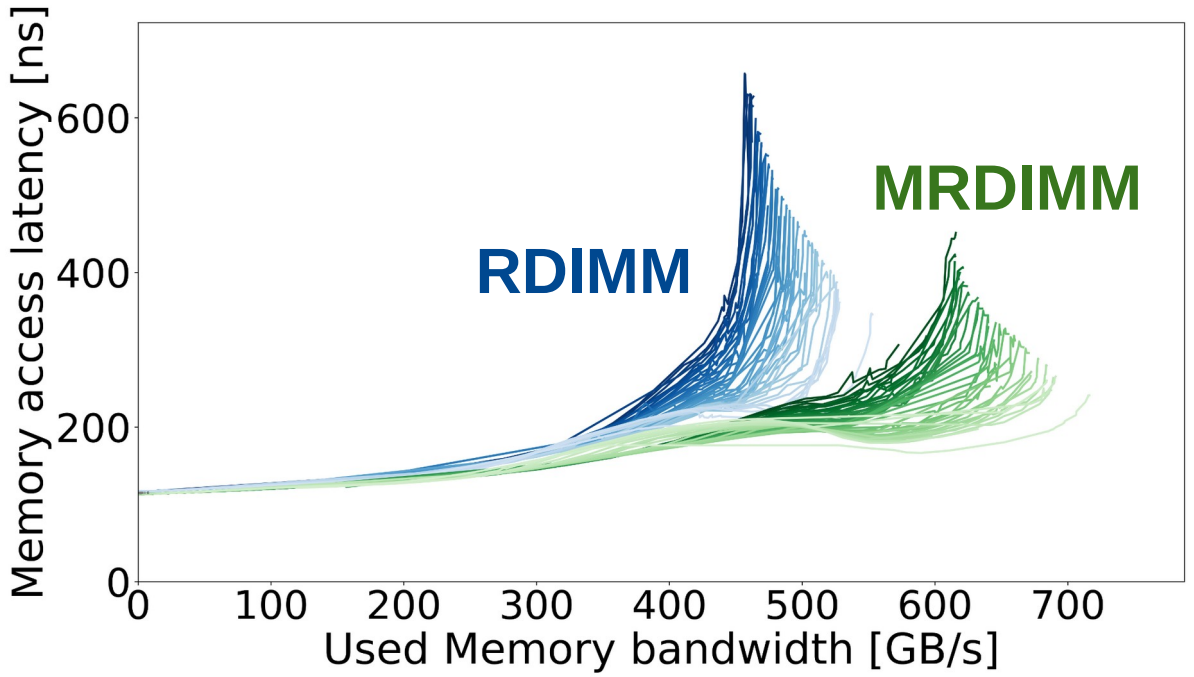
Examples

- We are going to be showing the following apps
  - STREAM Triad
    - https://www.cs.virginia.edu/stream/FTP/Code/
  - Google Multichase
    - https://github.com/google/multichase

- On the following System
  - **Intel Granite Rapids**
    - w/ 12× 64GB **RDIMM–6400**
    - w/ 12× 64GB **MRDIMM–8800**

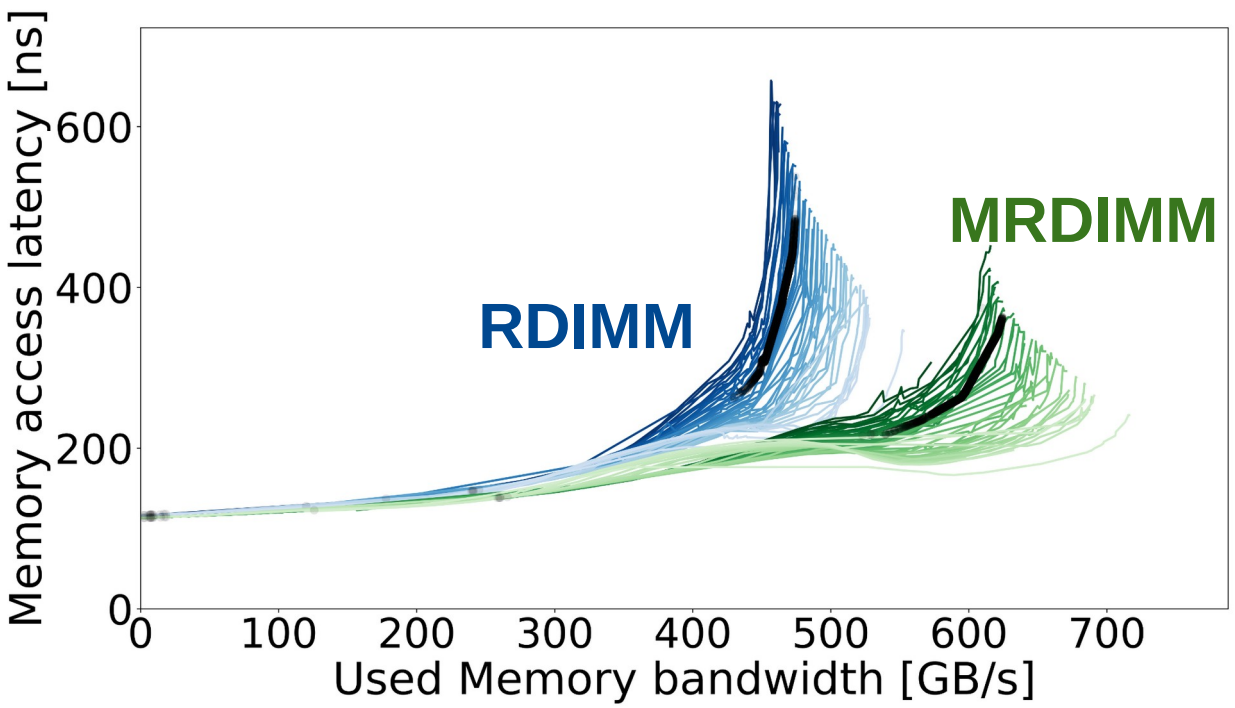
Intel GNR RDIMM-6400 → MRDIMM-8800

- **Unprecedented memory system performance**
  - *+75% bandwidth utilization: Sustained/Max theoretical → Comparable to or better than servers with DDR4/5 RDIMMs*
  - *Similar latency to DDR4/5 RDIMMs*



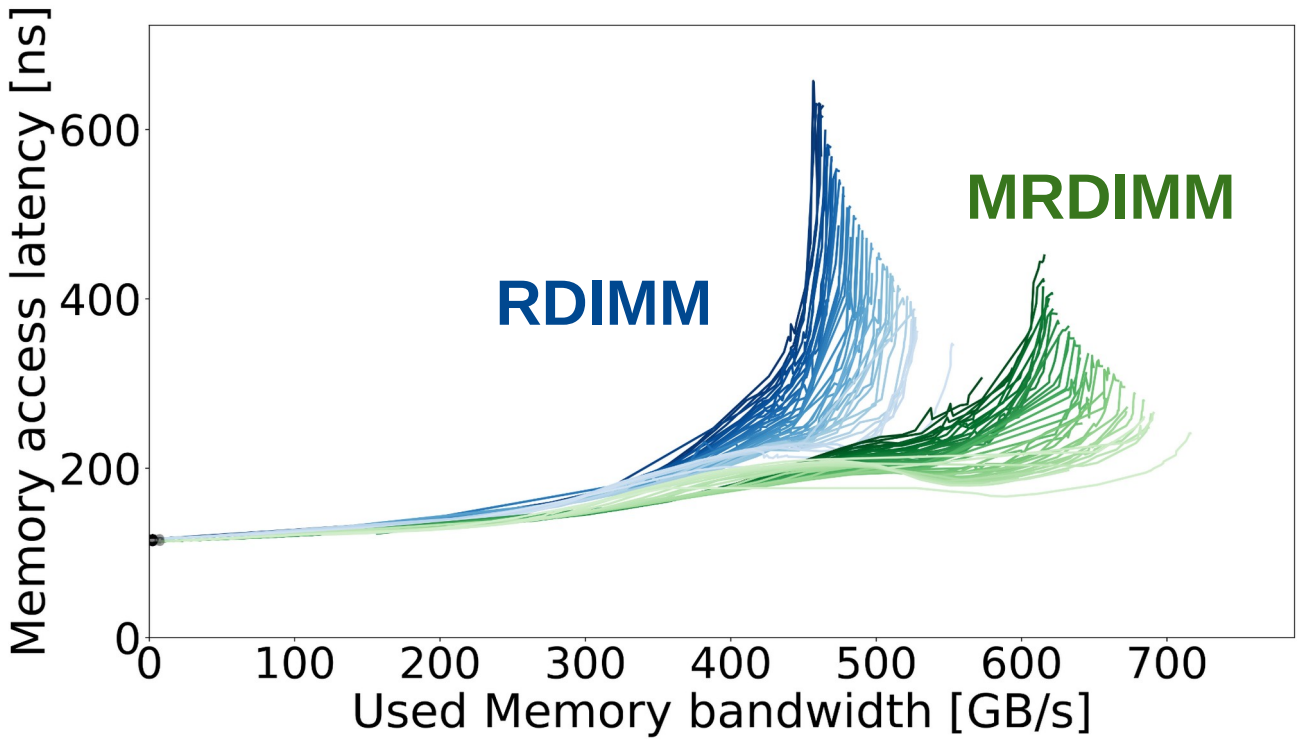
Intel GNR RDIMM-6400 → MRDIMM-8800

- **STREAM TRIAD**
  - *IPC change estimated: 30%*
  - *CPI change estimated: -23%*
  - *Bandwidth change estimated: 30%*
  - *Estimation error: 2%*



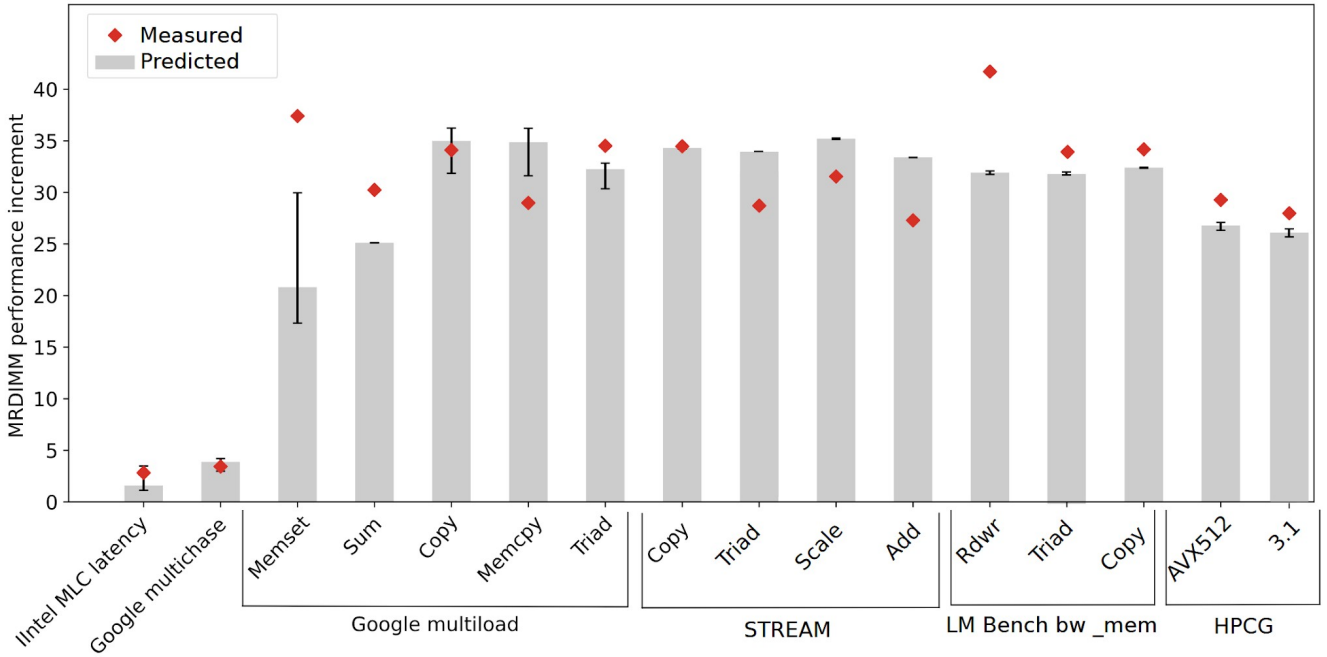
Intel GNR RDIMM-6400 → MRDIMM-8800

- **Google multichase**
  - IPC change estimated: 2%
  - *CPI change estimated: -2%*
  - *Bandwidth change: 2%*
  - *Error: 4%*

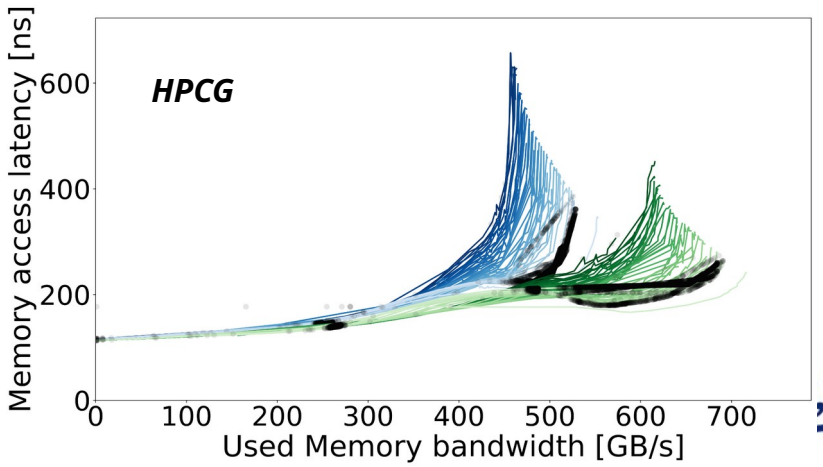


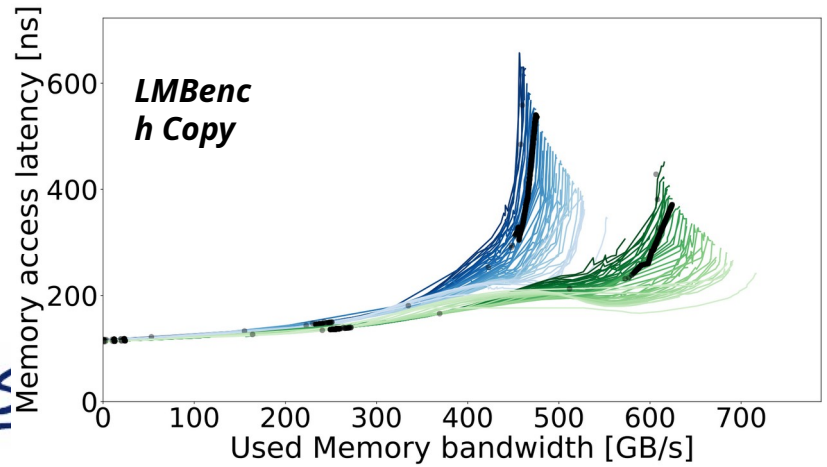
Intel GNR RDIMM-6400 → MRDIMM-8800

- **Bar chart (right)**
  - Estimation error bars:
    Based on boundaries for MLP and
  - OOO-execution
    (we cannot measure the exact values)



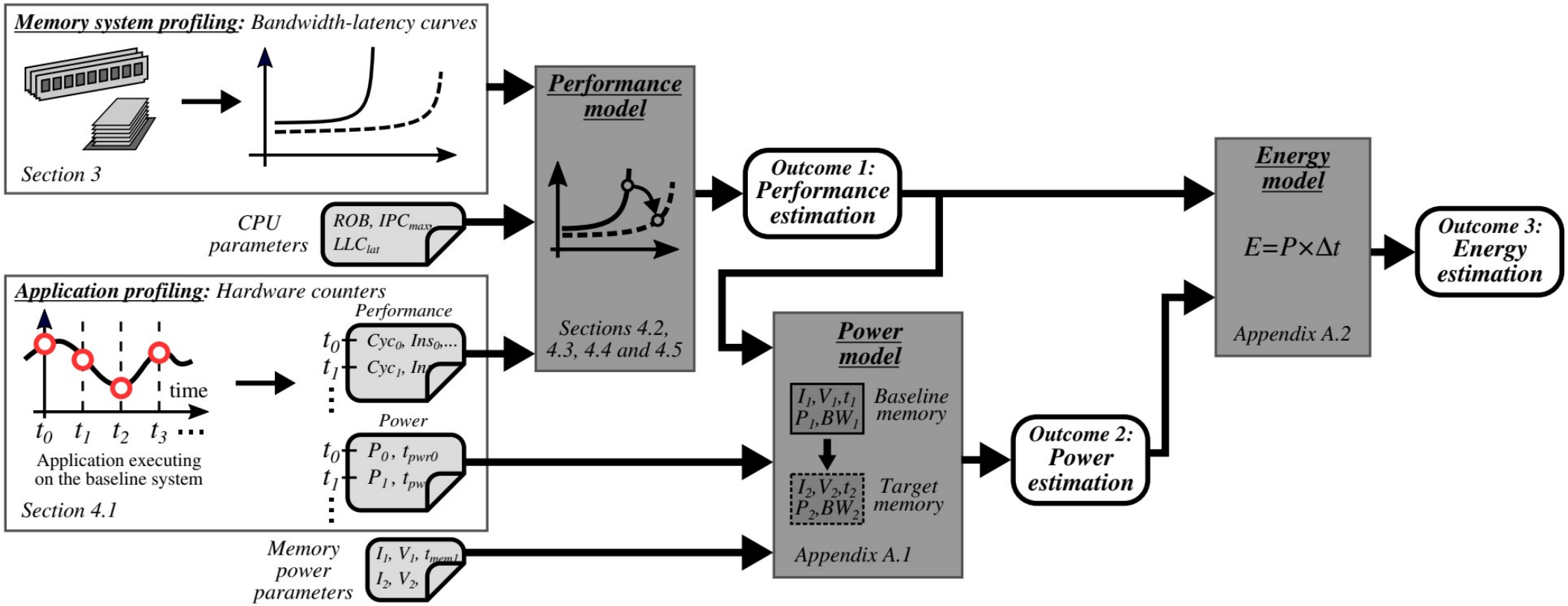
- **Application segments (examples)**





Barcelona Supercomputing Center
Centro Nacional de Supercomputación

PROFET predicts: Performance, power and energy



- This presentation focused on **performance**
- **Power energy**: paper, git

Why PROFET is cool

- **Disruptive approach to memory design space exploration** (hardware simulation)

- Instead of simulating your application on a baseline hardware → **Run & Profile**
  - Simpler, faster, scalable, more correct

- **Memory bandwidth-latency curves**
  - Not trivial (especially in 2019)
  - But still, much simpler, faster, scalable and more correct than memory systems simulation

- Designed for SandyBridge with DDR3 → **Plug&Play "everywhere"**
  - See the evaluation slide → The PROFET did not change from SandyBridge
  - The PROFET model is
    - Generic: Requires standard CPU parameters (datasheets) and profiling (hardware counters)
    - Formal: No "magic numbers" which vary from one system to another

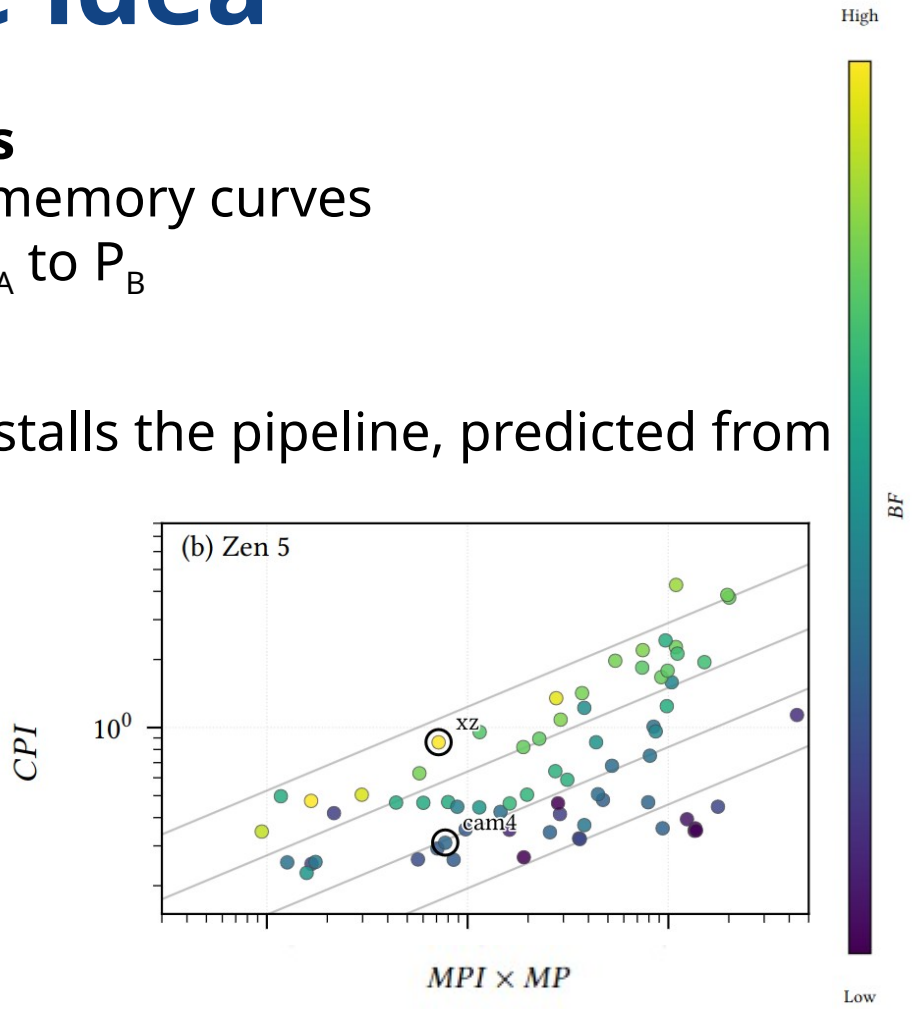
- **Brutally evaluated against actual hardware**

Aneto: The core idea

- **Analytical model** built on **bandwidth–latency curves**
  - Performance = **intersection** of the workload and memory curves
  - Memory A → B: The operating point moves from P_A to P_B

- **Blocking factor:** fraction of the memory penalty that stalls the pipeline, predicted from one baseline run with three inputs:
  - Cycles per instruction (**CPI**)
  - Last level cache misses per instructions (**LLC MPI**)
  - Memory penalty (**MP**).

- **Cross-workload regression**: learned once per machine from other workloads, not per application



R. Taranco, R. Mueller, and M. Giardino, “Aneto: Predicting system performance by exploiting cross-workload regularity,” in *Proc. 59th IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Athens, Greece, 2026, to appear. [Online]. Available: arXiv:2608.07179

Aneto: platforms and memory configurations

Platforms

- Real hardware: AMD Zen 2 (DDR4-2133), Zen 3 (DDR4-3200), Zen 4 (DDR5-4800), Zen 5 (DDR5-4800), Intel Comet Lake (DDR4-2933), Arm server (DDR-5600, HBM)
- Simulator: ChampSim-pf, Champsim-nopf and Sniper.

Benchmarks: SPEC CPU2017, PARSEC, GAP, Ligra, GMS, DaCapo, AI-ML and DPC-4 traces.

Accuracy:

- Across eight platforms at 8x memory penalty: **median CPI error 14.7%**
- ARM Server DDR5-5600 → HBM (near and far): **CPI error 12.7%**

Paper presentation: Session 7B: Performance & Power modeling for DSE, **November 4th, 10:40 AM - 12:25 PM**.

Contact: Raul Taranco (raul.taranco@h-partners.com), Rene Mueller (rene.mueller@huawei.com) and Michael Giardino (michael.giardino@huawei.com).

R. Taranco, R. Mueller, and M. Giardino, “Aneto: Predicting system performance by exploiting cross-workload regularity,” in *Proc. 59th IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Athens, Greece, 2026, to appear. [Online]. Available: arXiv:2608.07179

Explore more

PROFET paper: Milan Radulovic, Rommel Sánchez Verdejo, Paul Carpenter, Petar Radojković, Bruce Jacob, and Eduard Ayguadé. ***PROFET: Modeling System Performance and Energy Without Simulating the CPU***. SIGMETRICS 2019.

PROFET github: https://github.com/bsc-mem/PROFET

PROFET sensitivity analysis: https://esecretary.salle.url.edu/zona_autentica/memoriesTFC/25299_GI_2023.pdf

Material page: https://memory.bsc.es/training/its-the-memory-stupid/materials



mariana.carmin@bsc.es



mariana.carmin@bsc.es

Maybe an interesting story to share: PROFET (2019) → Mess (2024)

- **Frustration: Some users (PhD students) were struggling with PROFET**
  - ***Equations?***
  - ***Complex overall workflow?***
    - Bandwidth—latency curves
    - CPU parameters
    - Application profiling
    - Equations

- ***Can we simplify PROFET?***

- ***What is the simplest stand-alone PROFET component?***
  - Bandwidth-latency curves for memory system profiling
  - This idea is the starting point of Mess

It's the Memory, Stupid!

