Exploring iGPU memory interference response to L2 cache locking

Alfonso Mascareñas González, Jean-Baptiste Chaudron, Régine Leconte, ISAE-SUPAERO
Youcef Bouchebaba, David Doose, ONERA

21st International Workshop on Worst-Case Execution Time Analysis (WCET 2023)

Vienna

11/07/2023

ONERA
THE FRENCH AEROSPACE LAB

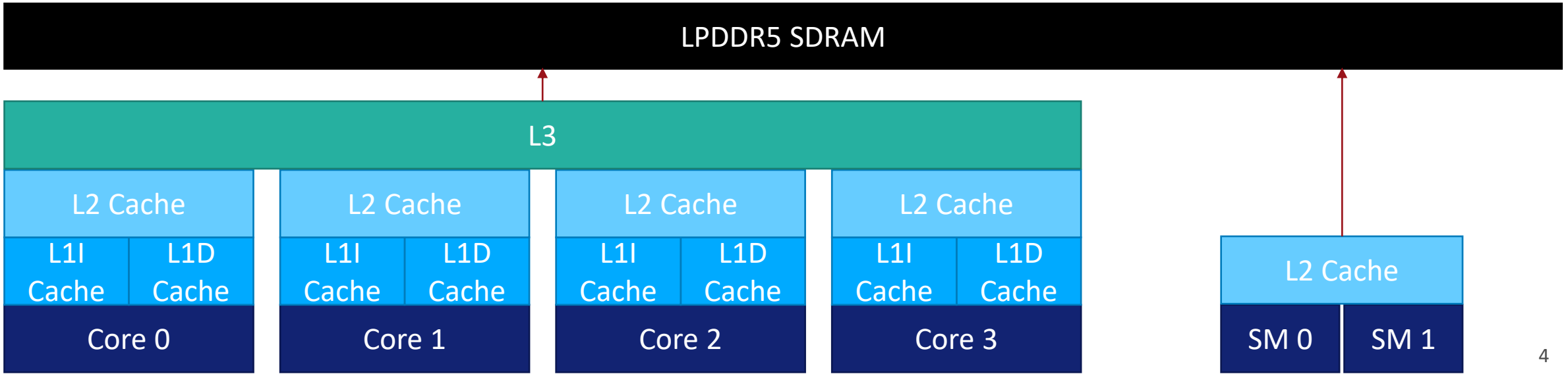
ISAE
Institut Supérieur de l'Aéronautique et de l'Espace
SUPAERO

Content

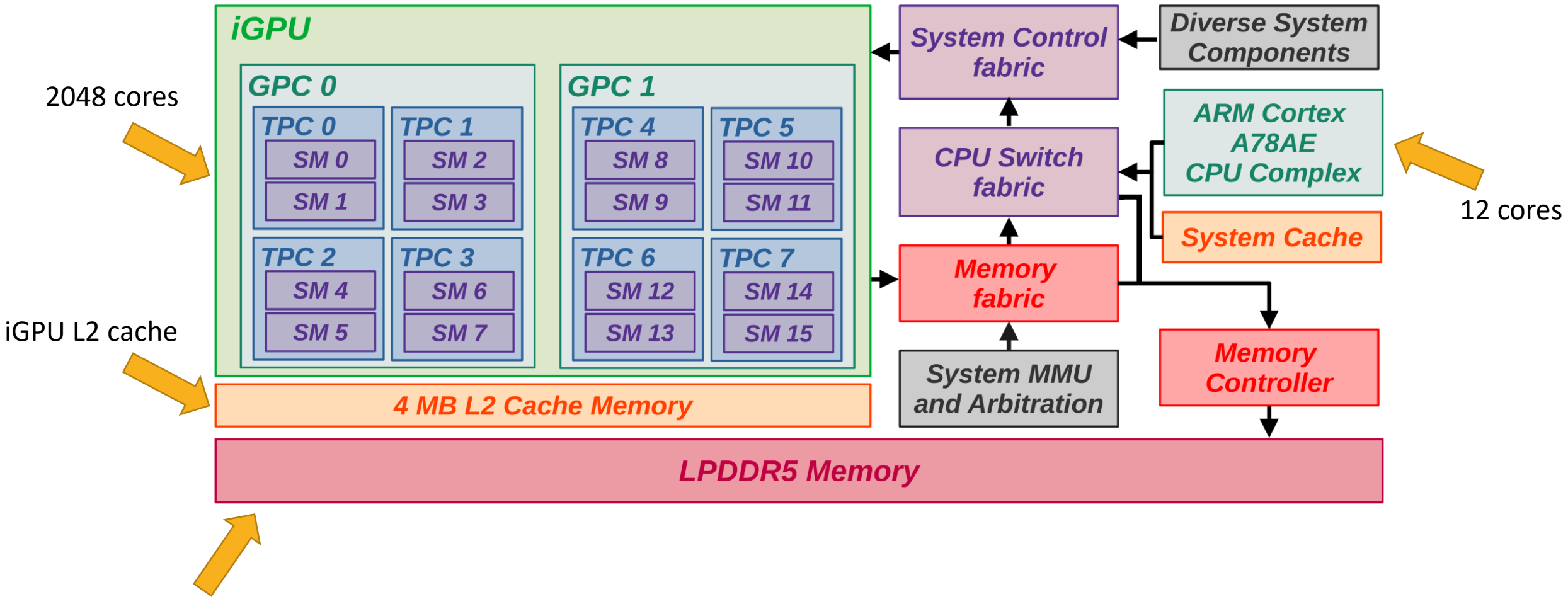
Content

Context: CPU-GPU System on Chips

- Multicore platforms appear in order to improve the SWaP-C properties and throughput.
- More specifically, CPU-GPU architectures are used to meet with the parallelization needs of current applications.
- Inter-core interference appear as a result of resource sharing.
- For instance, memory interference:
  - CPU L1, L2, Ln caches
  - GPU L1, L2, Ln caches
  - DDR SDRAM (GPUs are integrated iGPU)



Context: NVIDIA Jetson AGX Orin 64GB

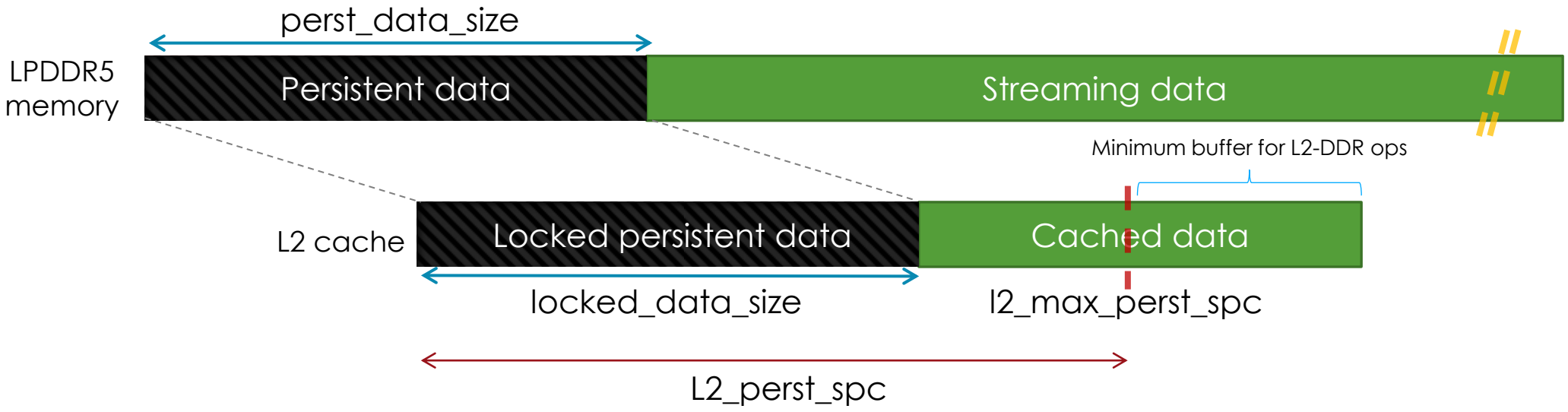


Content

L2 Cache Locking on NVIDIA

Hardware-based cache locking. NVIDIA uses the terms “Persistent data” and “Streaming data” for locked and normal data respectively.

If perst_data_size ≤ l2_perst_spc



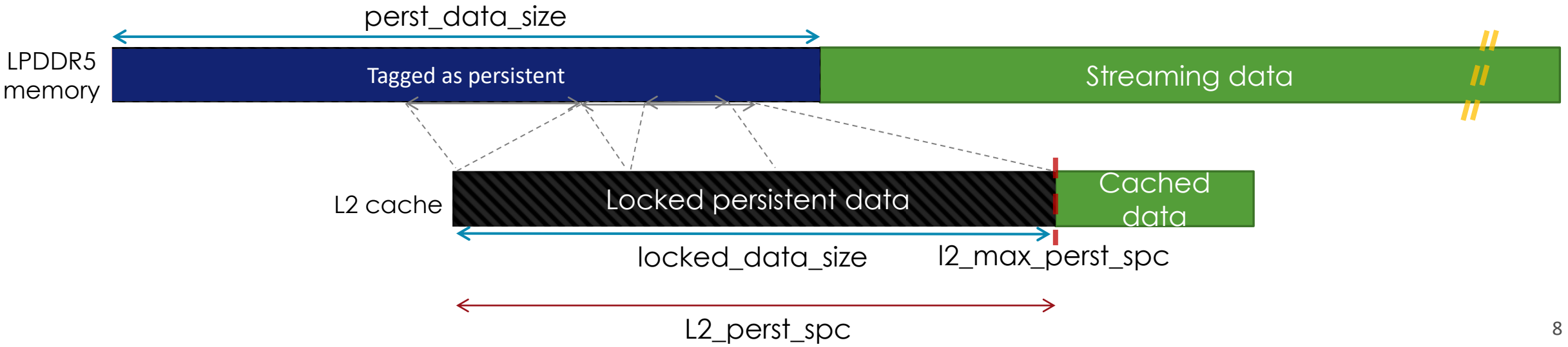
L2 Cache Locking on NVIDIA

Hardware-based cache locking. NVIDIA uses the terms “Persistent data” and “Streaming data” for locked and normal data respectively.

If perst_data_size > l2_perst_spc

Options:

1. **Tag all data as persistent**
2. Leave NVIDIA randomly choose the portion to lock
3. Lock a portion of data ourselves



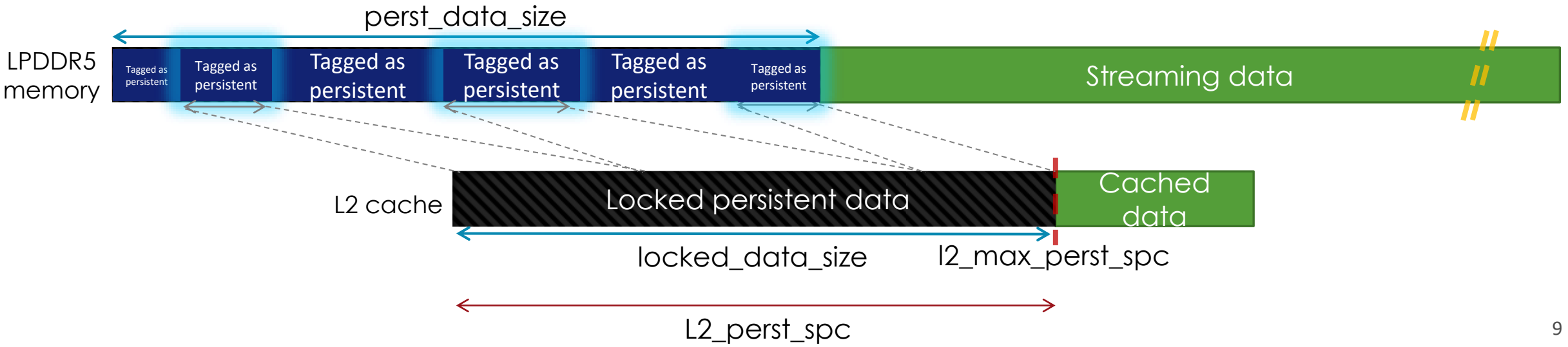
L2 Cache Locking on NVIDIA

Hardware-based cache locking. NVIDIA uses the terms “Persistent data” and “Streaming data” for locked and normal data respectively.

If perst_data_size > l2_perst_spc

Options:

1. Tag all data as persistent
2. **Leave NVIDIA randomly choose the portion to lock**
3. Lock a portion of data ourselves



L2 Cache Locking on NVIDIA

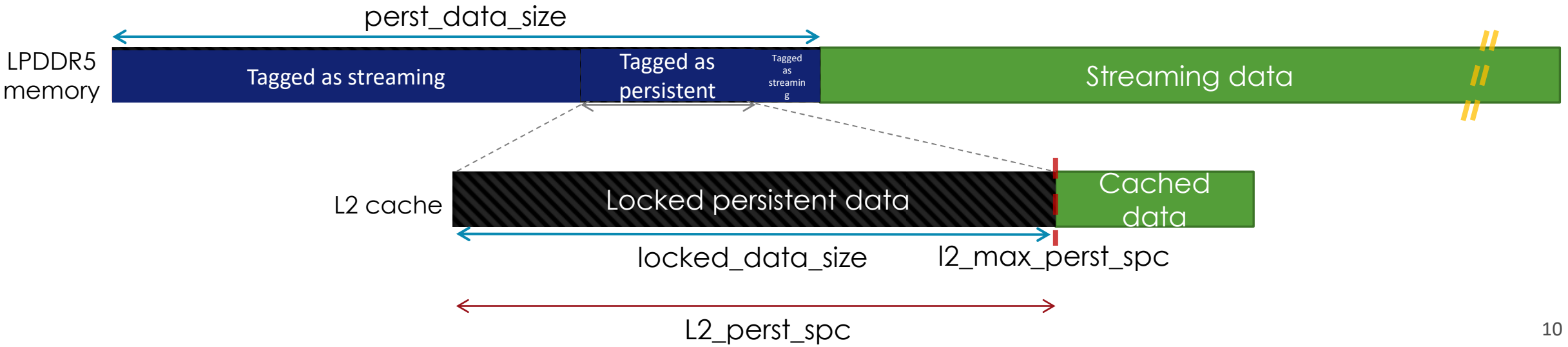
Hardware-based cache locking. NVIDIA uses the terms “Persistent data” and “Streaming data” for locked and normal data respectively.

locked_data_size =
min{persistent_data_size, persistent_cache_spc, 3 MB}

If perst_data_size > l2_perst_spc

Options:

1. Tag all data as persistent
2. Leave NVIDIA randomly choose the portion to lock
3. **Lock a portion of data ourselves**

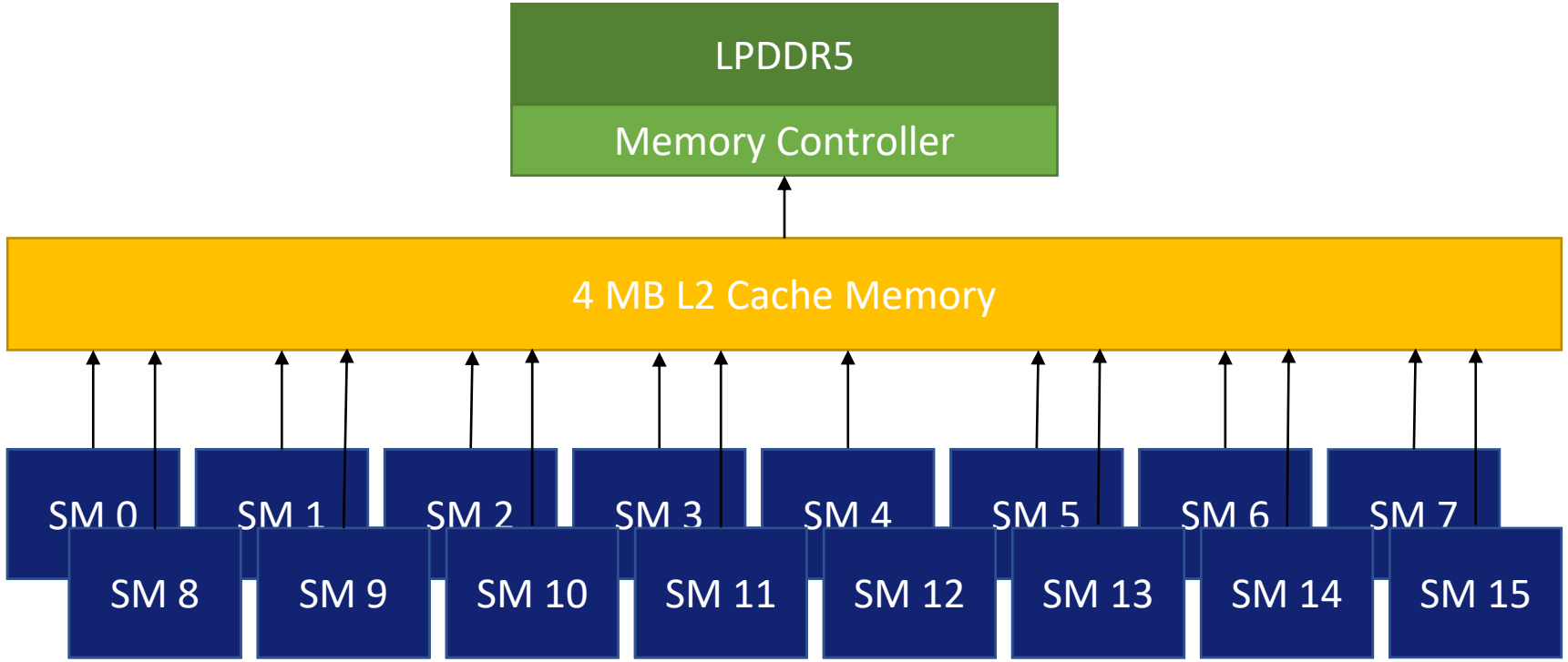


Content

1. Context
2. L2 Cache Locking on NVIDIA
3. **L2 Cache memory interference analysis**
4. Conclusions and future work

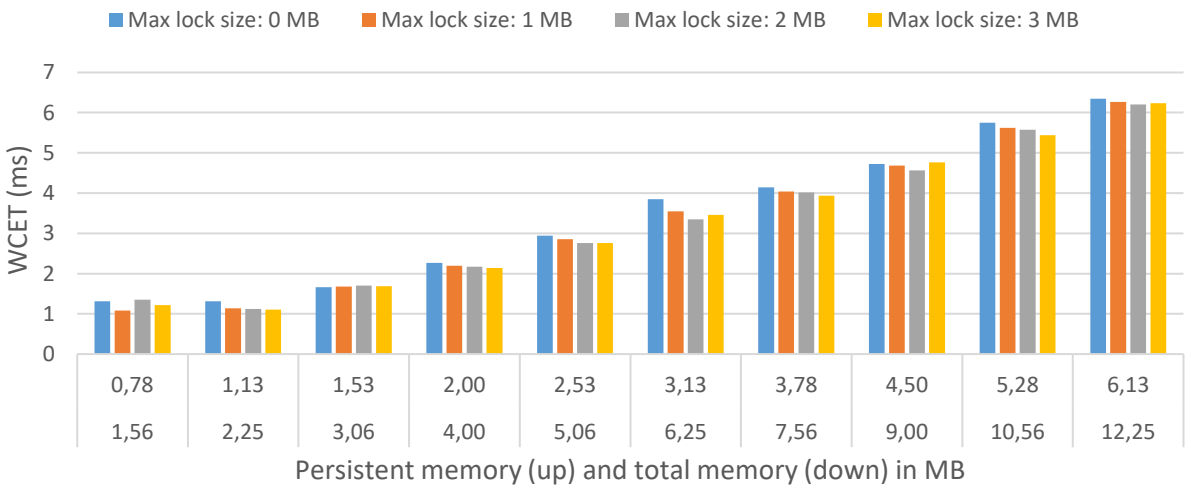
L2 Cache memory interference analysis: Scenario 1

- Scenario 1: Inter-SM interference with one application

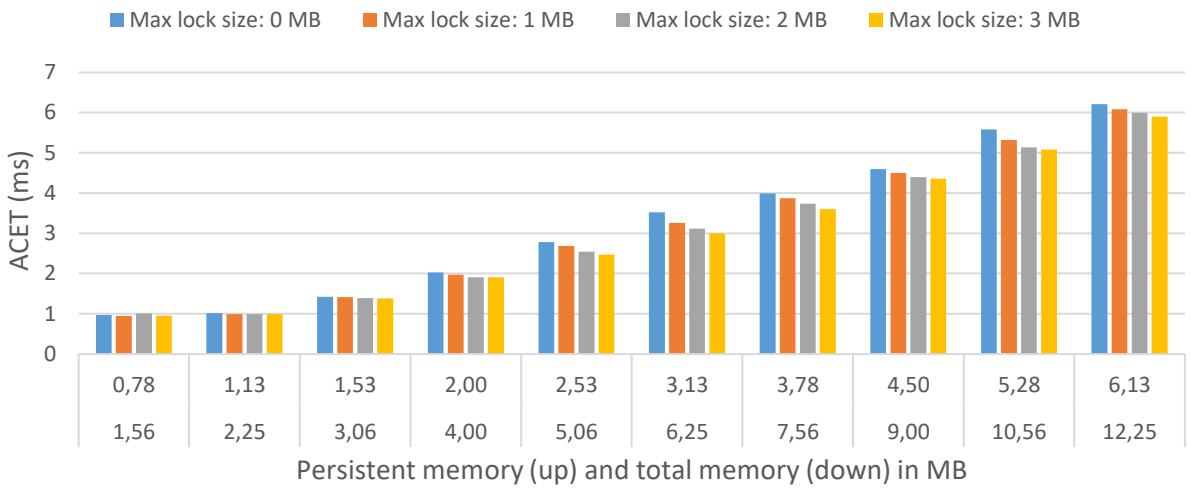


L2 Cache memory interference analysis: Scenario 1

- Scenario 1: Inter-SM interference with one application
- 2D convolution benchmark



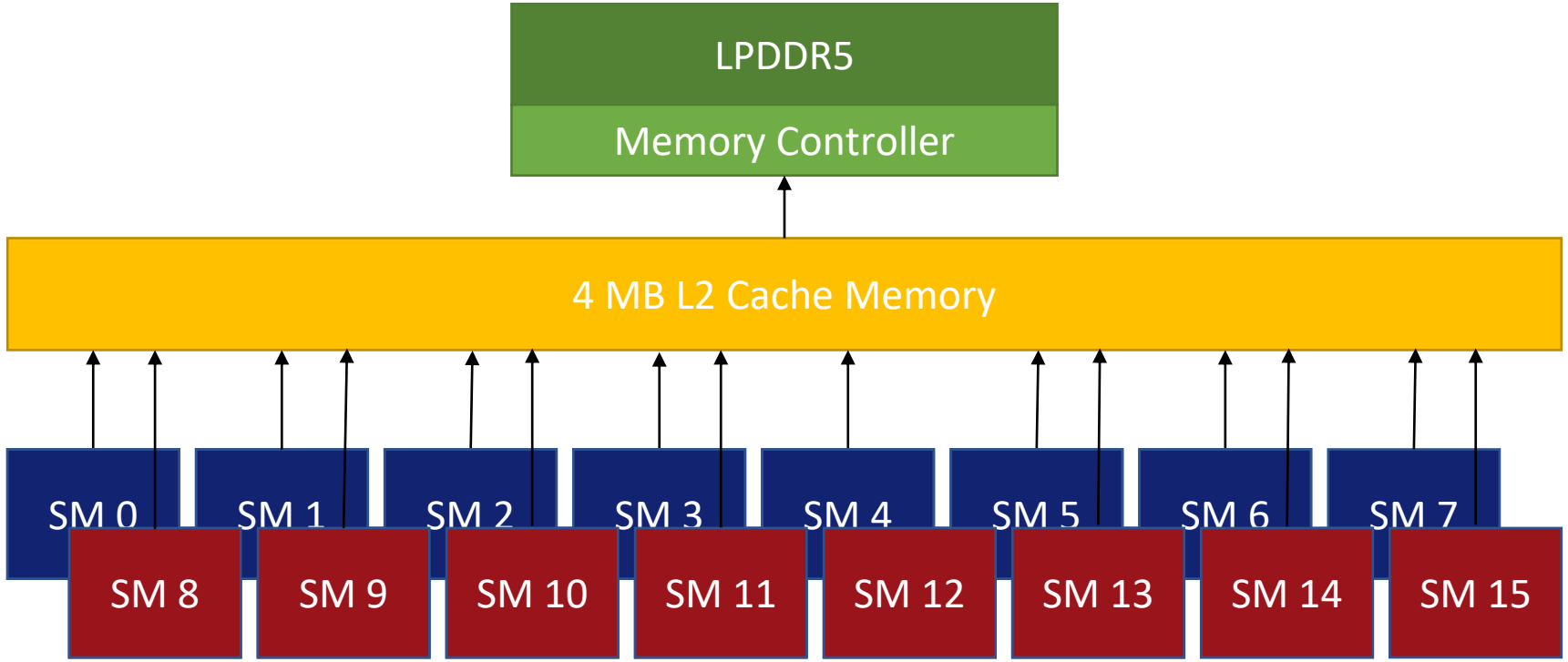
Measured Worst-Case Execution Time



Average Case Execution Time

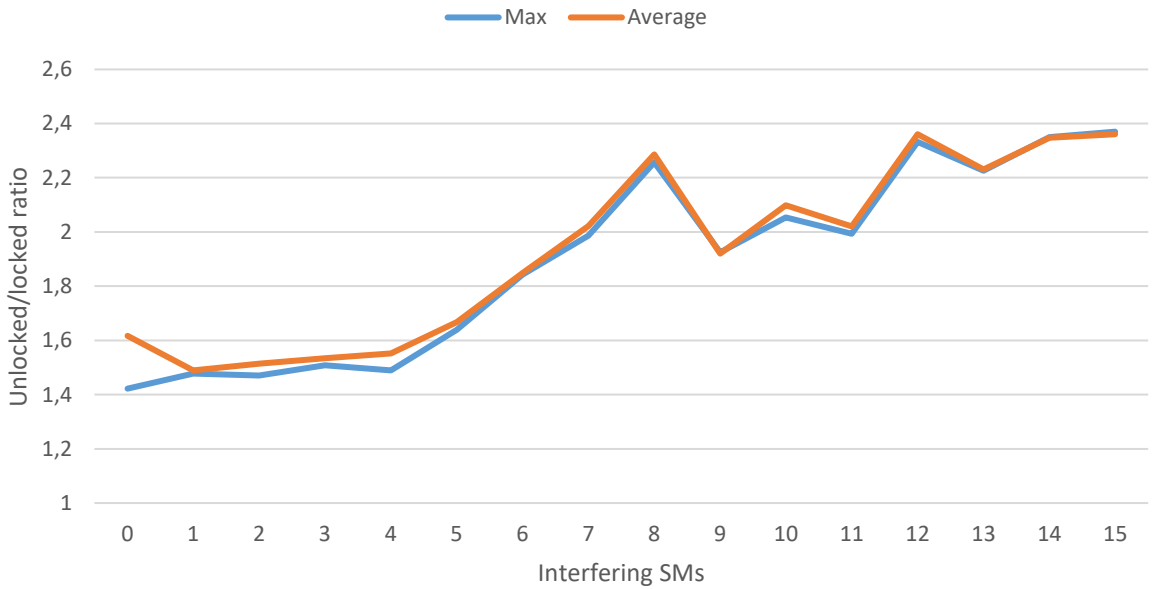
L2 Cache memory interference analysis: Scenario 2

- Scenario 2: Inter-SM interference with several applications

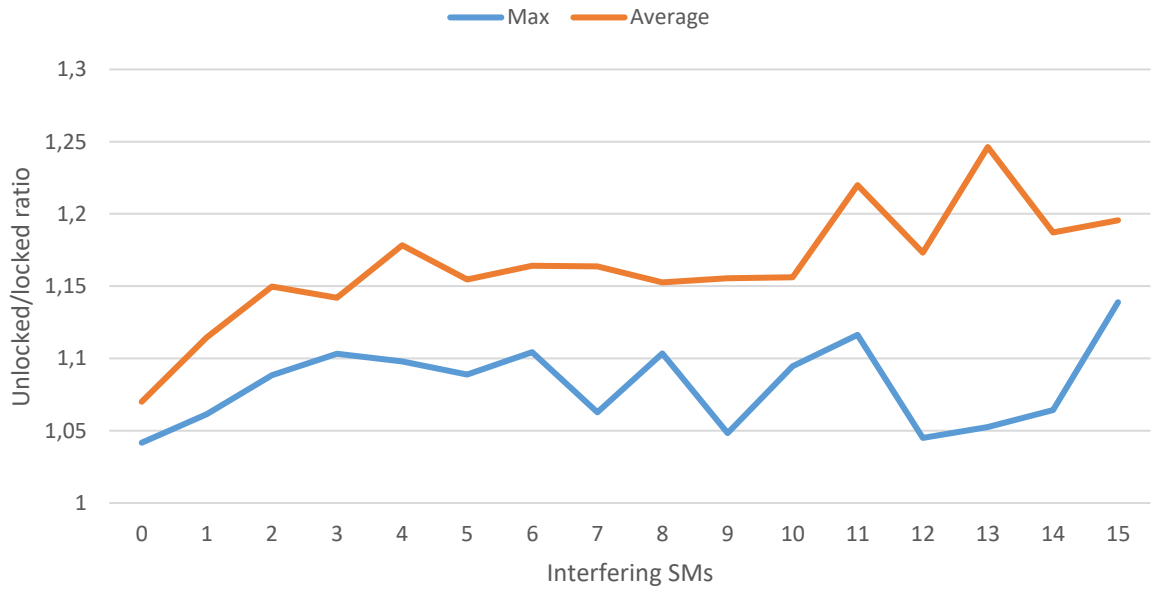


L2 Cache memory interference analysis: Scenario 2

- Scenario 2: Inter-SM interference with several applications
- 4 MB of persistent data and 8.04 MB of total data



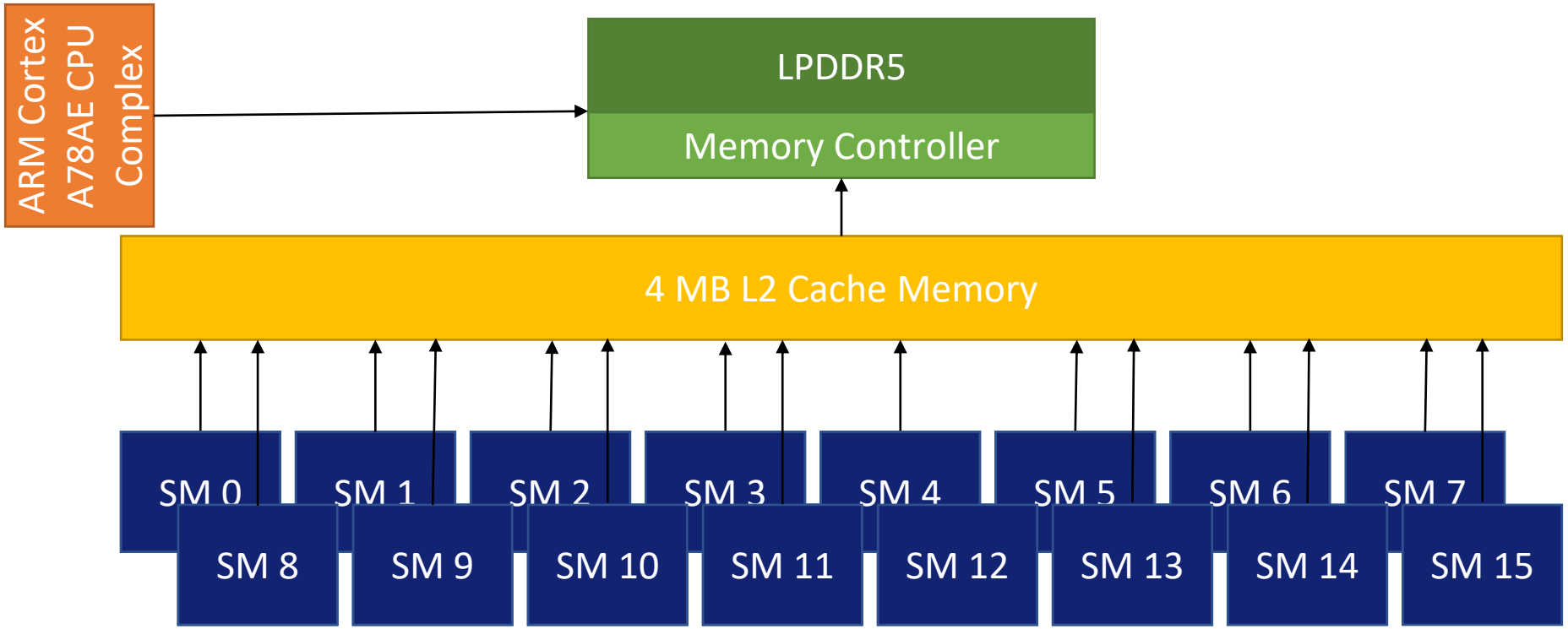
Data reset benchmark



2D convolution benchmark

L2 Cache memory interference analysis: Scenario 3

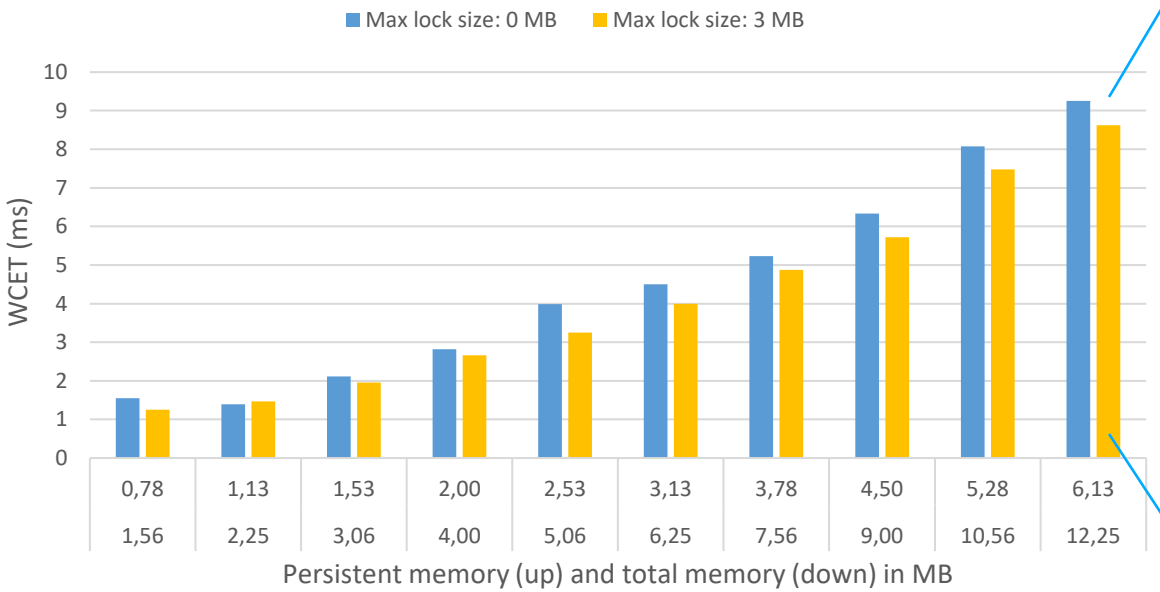
- Scenario 3: Inter-SM interference with one application and DDR SDRAM interference



L2 Cache memory interference analysis: Scenario 3

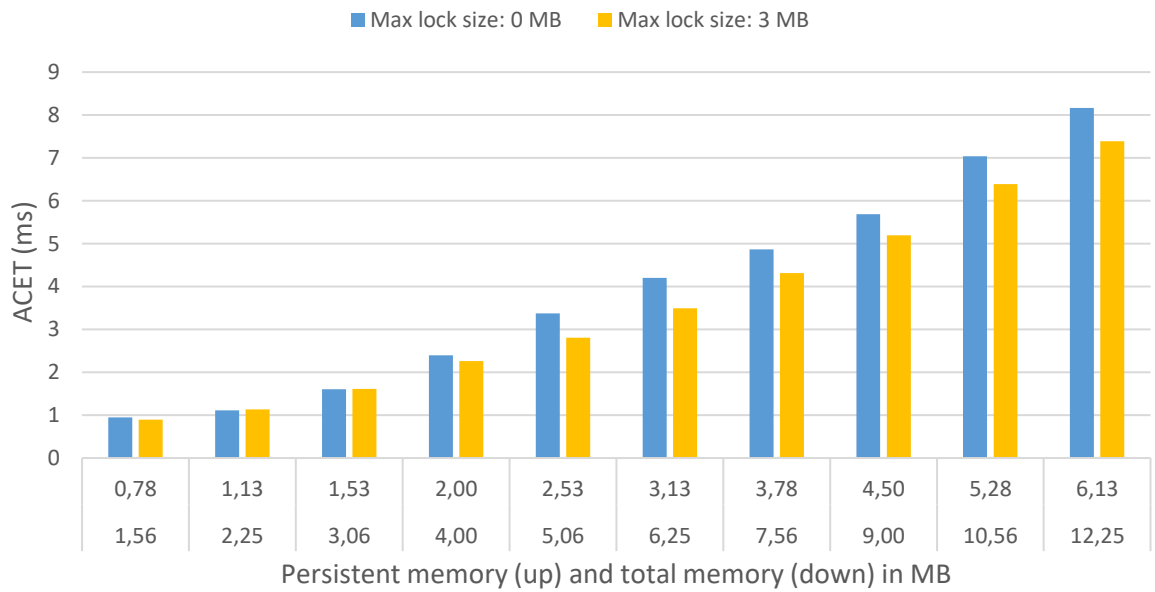
- Scenario 3: Inter-SM interference with one application and DDR SDRAM interference
- 2D convolution

Δ = 0.629ms



Measured Worst-Case Execution Time

Scenario 1:
6.347 ms (no lock)
6.235 ms (lock)
Δ = 0.112 ms



Average Case Execution Time

Content

1. Context
2. L2 Cache Locking on NVIDIA
3. L2 Cache memory interference analysis
4. **Conclusions and future work**

Conclusions

The effectiveness of the L2 cache locking depends on the:

- Application properties (e.g., L2 memory access frequency)
- iGPU resources (e.g., SMs)
- iGPU L1 and L2 cache capacity
- Locked data
- Total data

Results indicate that for a single kernel application:

- mWCET and ACET improve when recurrent data is locked
- Less pressure is put onto the DDR SDRAM
- Data is protected when SM partitioning is performed

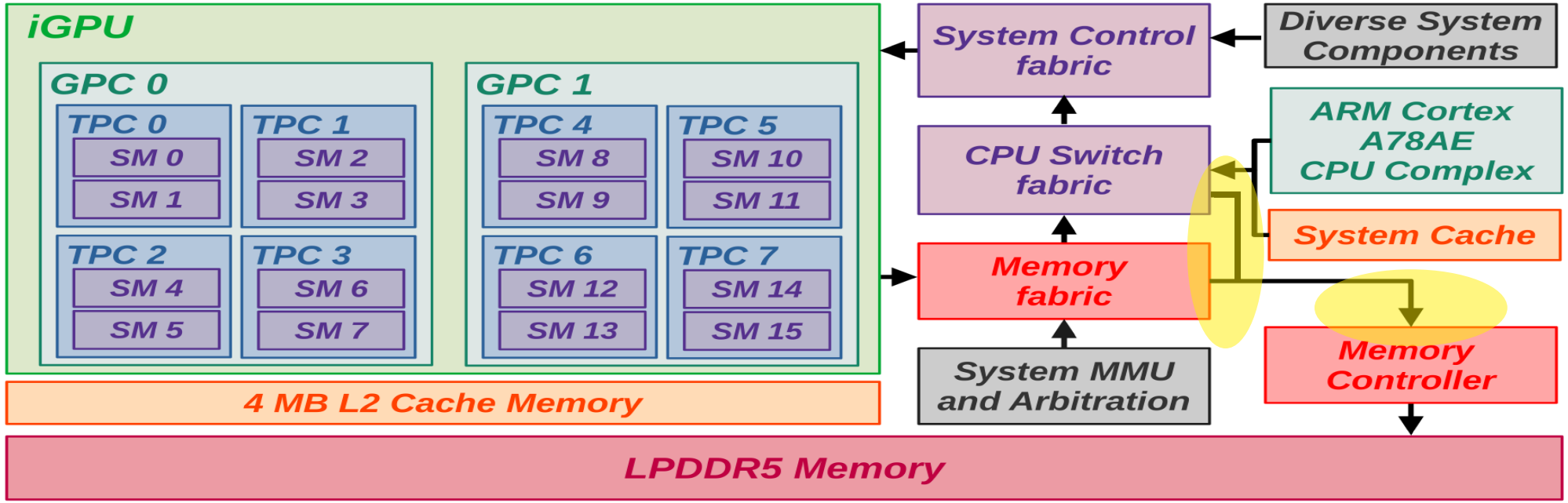
Future Work

- L2 cache locking on applications consisting of several kernels, e.g., Segnet:

WCET	ACET	BCET
5.91%	2.92%	2.77%

→ Inter-SM L2 cache interference reduction. Dynamic locking.

- Additional analysis of the DDR SDRAM interference reduction through iGPU L2 cache locking



Thank you for your attention