Memory management strategies in real-time embedded autonomous systems

Gianluca Brilli, Alessandro Capotondi, Paolo Burgio and Andrea Marongiu

IWES, 2021

<name>.<surname>@unimore.it

HiPeRT Lab
High-Performance Real-Time Lab

Introduction

- Increasing demand for embedded High Performance systems in different real-time safety critical domains.
  - Typically these systems are designed for **average case** performance;
  - Real-time applications need to guarantee also **worst-case** behaviour;
  - Novel systems are compex, several heterogeneity at different level.

Motivation of this work

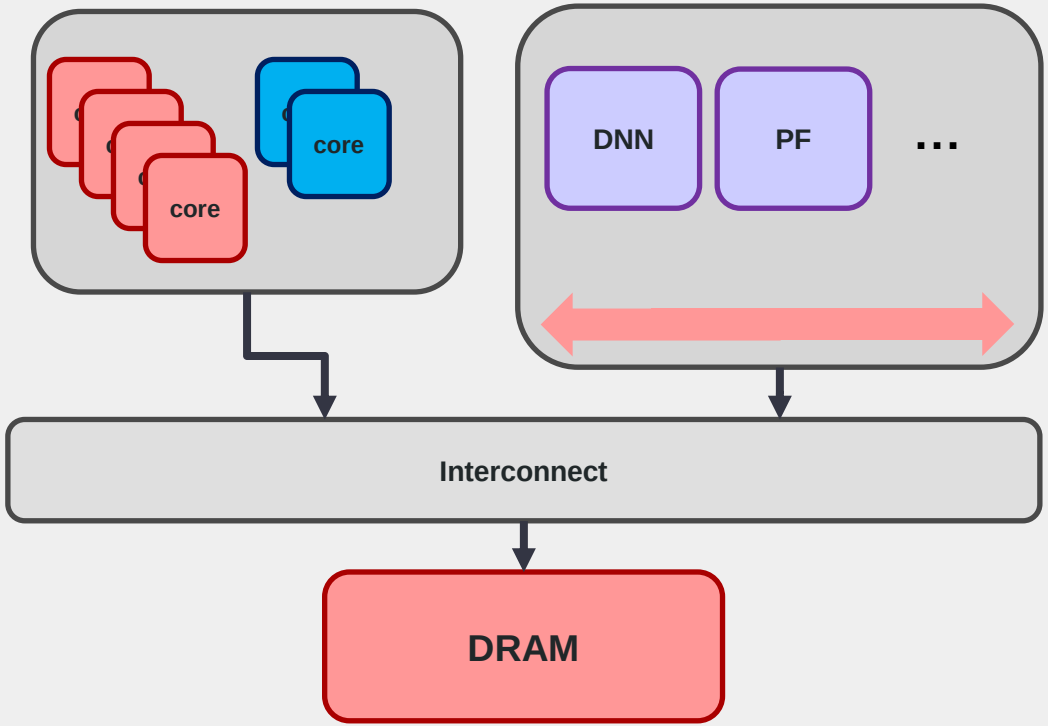
- The main issues that could negatively impact real-time correctness are shared on-chip resources and in particular main memory (DRAM):

  [1] R. Cavicchioli, N.Capodieci, M.Bertogna, ***Memory Interference Characterization between CPU cores and integrated GPUs in Mixed-Criticality Platforms***, ETFA 2017

  [2] A. Bansal, R. Tabish, G. Gracioli, R. Mancuso, R. Pellizzoni and M. Caccamo. **Evaluating the Memory Subsystem of a Configurable Heterogeneous MPSoC**. 14th annual workshop on Operating Systems Platforms for Embedded Real-Time applications (OSPERT 2018).

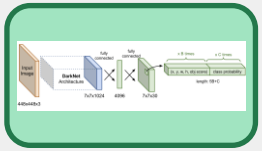
- We aim to **assess the memory contention** on a FPGA-based Heterogeneous SoC (HeSoCs).

Heterogeneous SoC

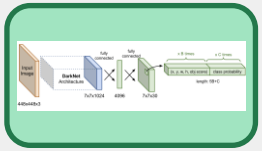


- Some examples:
  - NVIDIA Xavier;
  - **Xilinx Zynq UltraSCALE+;**
  - Xilinx Versal.
- HeSoC is a new emerging trend.

Accelerator cluster template

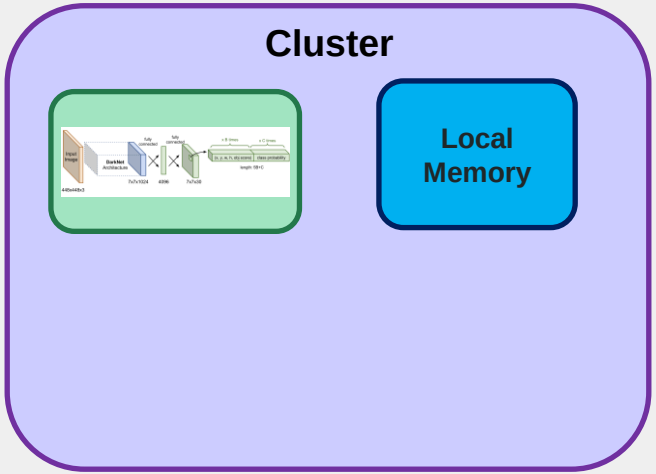


Accelerator cluster template



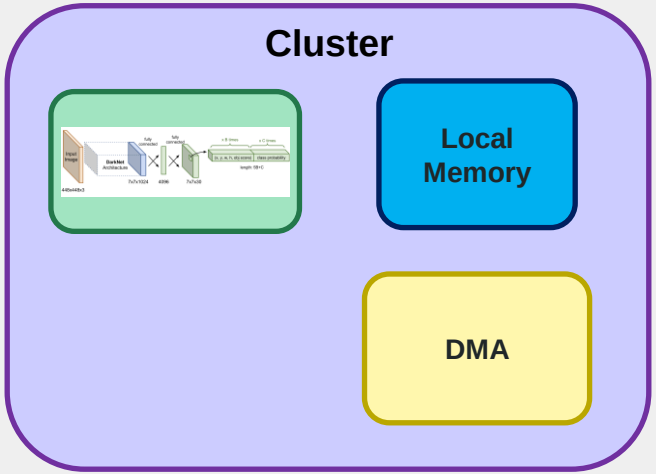
- Some examples:
  - A **DNN engine** to perform object detection;
  - A **localization algorithm;**
  - **...**

Accelerator cluster template



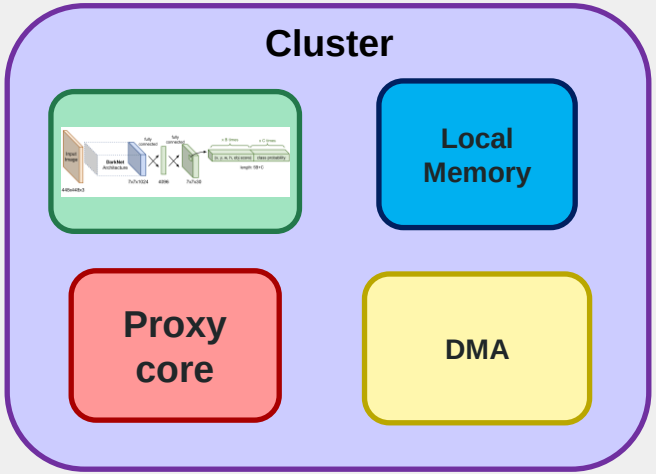
- Some examples:
  - A **DNN engine** to perform object detection;
  - A **localization algorithm;**
  - **...**

Accelerator cluster template



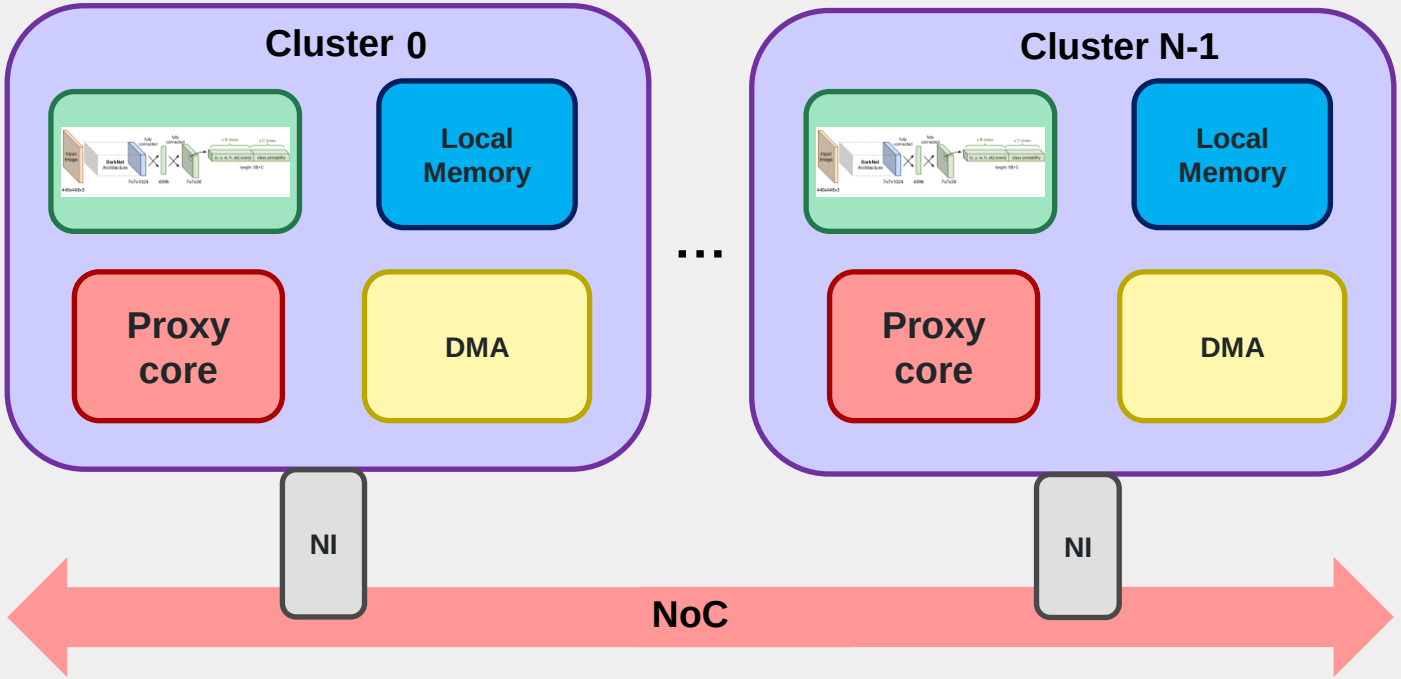
- Some examples:
  - A **DNN engine** to perform object detection;
  - A **localization algorithm;**
  - **...**

Accelerator cluster template

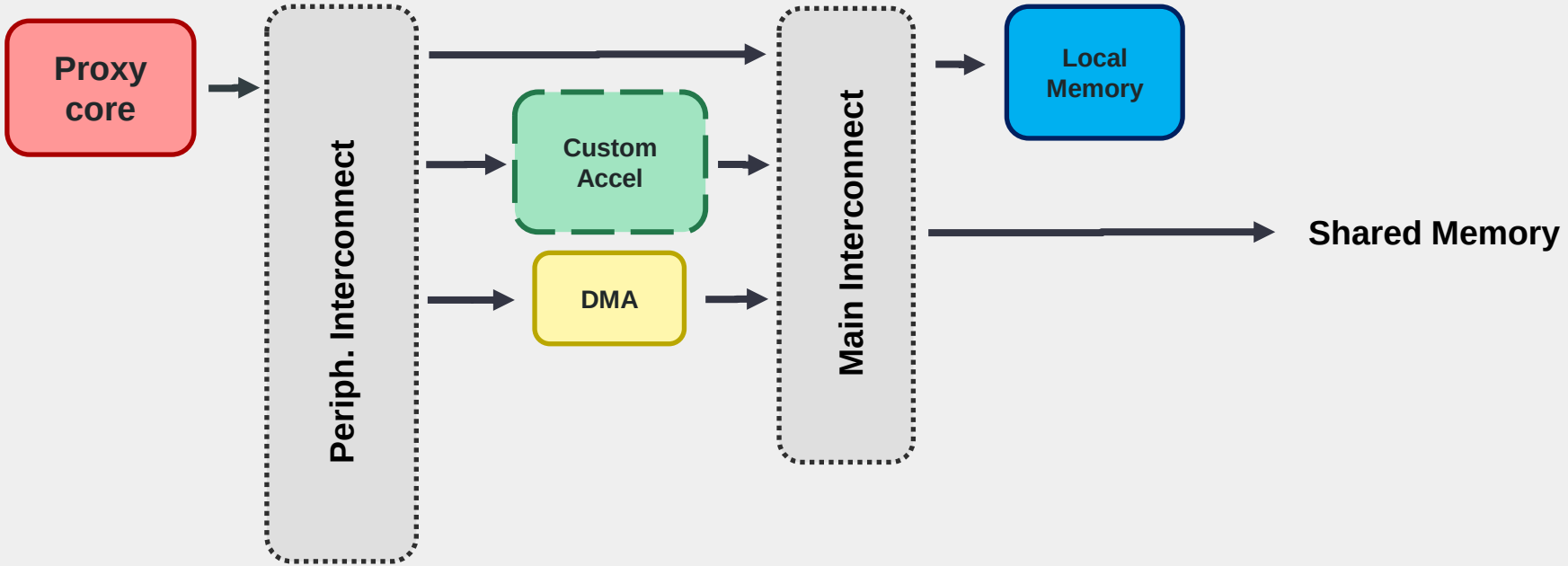


- Some examples:
  - A **DNN engine** to perform object detection;
  - A **localization algorithm;**
  - **...**

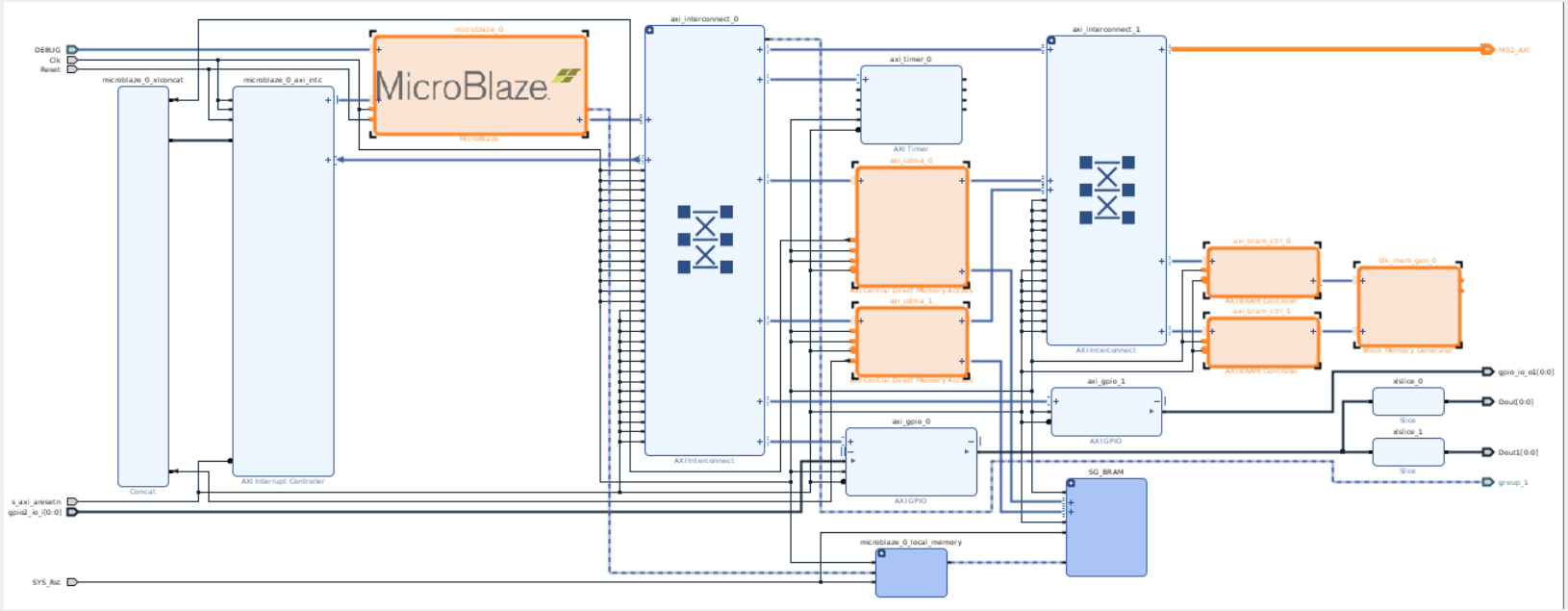
Accelerator cluster template



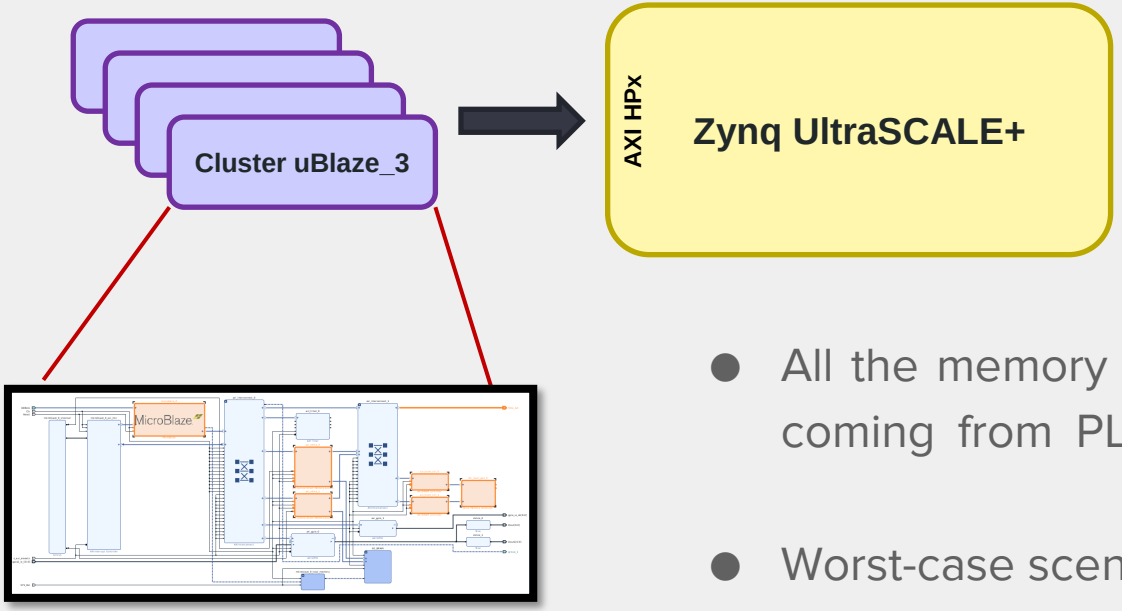
Accelerator cluster template (Cont)



Accelerator cluster template – with Xilinx IPs



Inside a MicroBlaze cluster



- One Cluster uBlaze for each AXI HP.

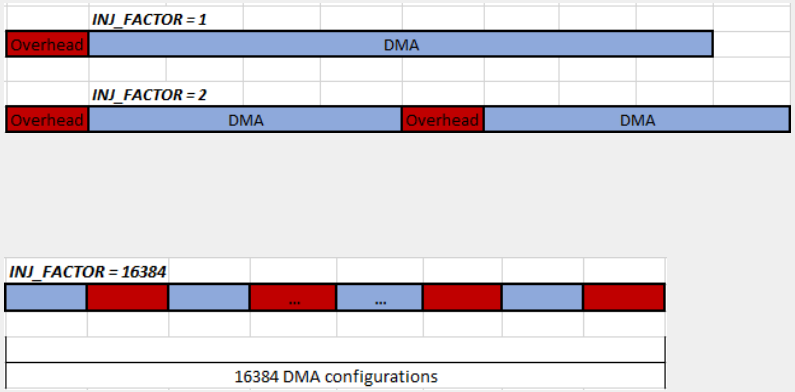
- All the memory controller ports coming from PL are exploited.

- Worst-case scenario!

DMA programming costs

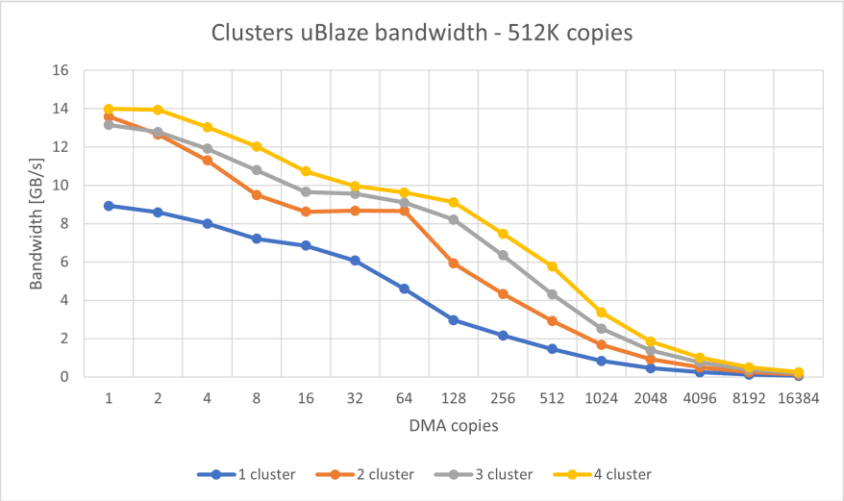
- Memory traffic is controlled on the softcore;
- By splitting the copy into sub-copies.

```
for(size_t j = 0; j < INJ_FACTOR ++j) {
        cdma_memcpy(&cdma_0,
            (UINTPTR)(DST_1 + j*(SIZE/COPIES)),
            (UINTPTR)(SRC_1 + j*(SIZE/COPIES)),
            (SIZE/COPIES)
        );
        cdma_memcpy(&cdma_1,
            (UINTPTR)(DST_2 + j*(SIZE/COPIES)),
            (UINTPTR)(SRC_2 + j*(SIZE/COPIES)),
            (SIZE/COPIES)
        );
        cdma_wait(&cdma_0);
        cdma_wait(&cdma_1);
}
```



Experimental setup

- Memory traffic varying the **number of active clusters** and **copy size**.



Copy size	Bandwidth [GB/s]
32K	7.43245218
64K	8.051900569
128K	8.006368012
256K	8.597385762
512K	**8.938787908**
1M	9.112728932
2M	9.199301772
3M	9.23046379

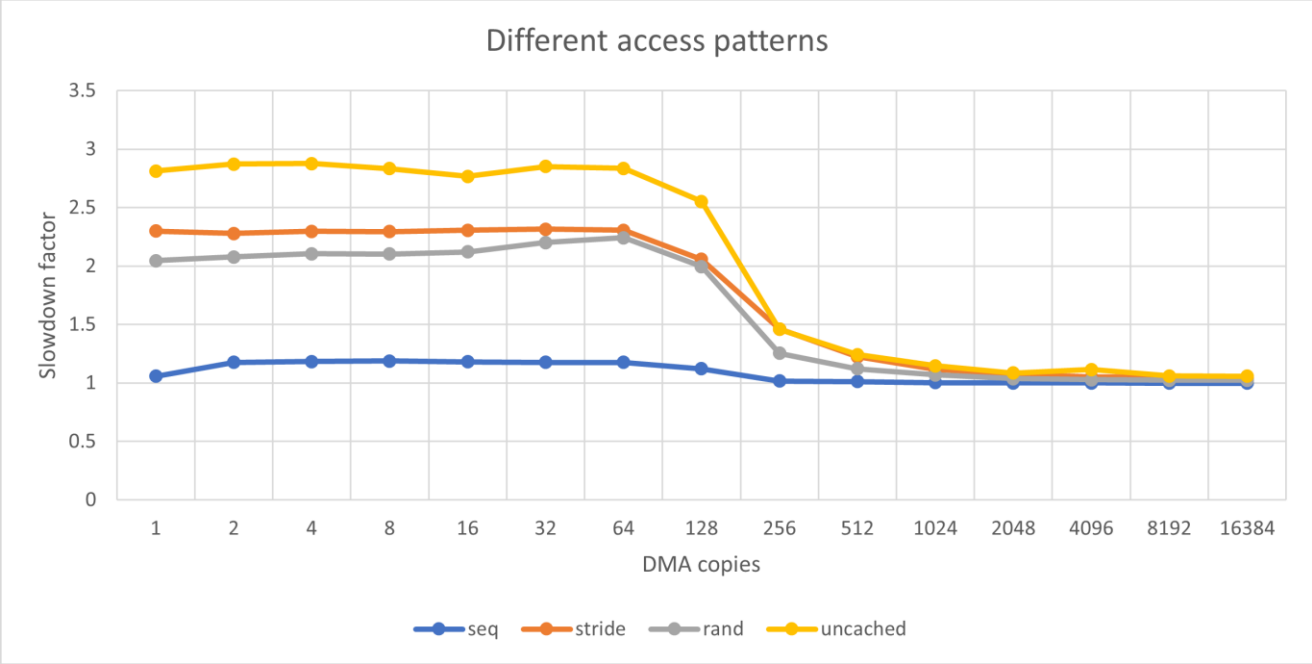
- Bandwidth injected by **one cluster**, varying the **WSS**.

Experimental setup (Cont)

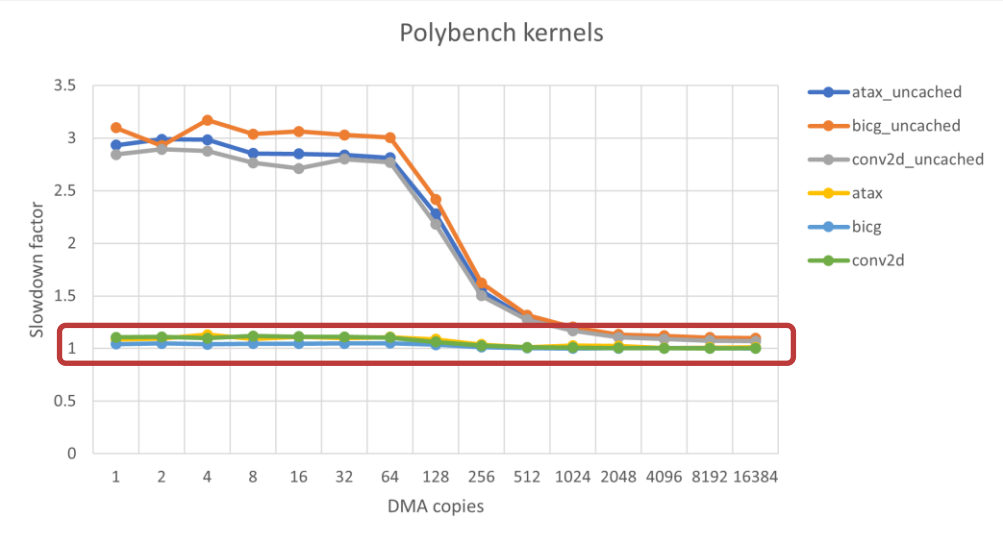
- Arm A53 cores are the **tasks under test (UT)**.
- The tests are a collection of benchmarks extracted from the Polybench suite.
- The benchmarks are executed on top of a custom Petalinux-based system.

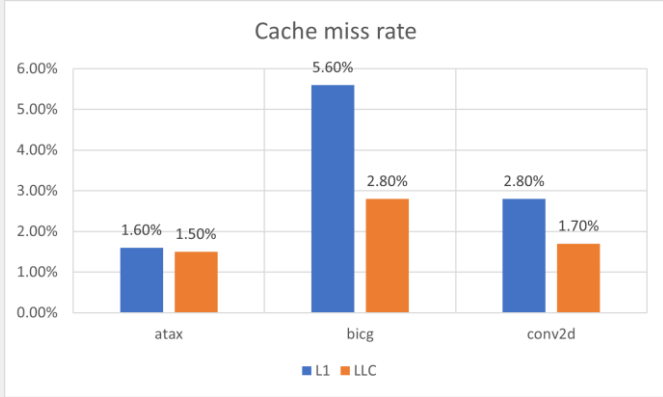
	Bandwidth [GB/s]
Core 0	1.755
Core 1	1.774
Core 2	1.772
Core 3	1.761
Total	7.064

Slowdown of different access patterns



Polybench kernels





Conclusion and Future Works

- We presented a preliminary study regarding the interference generated by the PL and experimented by A53 cores;
- Implements a PL-based mechanism to automatically regulate PL-generated BW;
- Evaluate memory contention on Xilinx Versal.

Thank you!

Gianluca Brilli