Agile Design Methodology for Accelerator-Rich Cluster-based RISC-V SoC

Gianluca Bellocchi[1], Alessandro Capotondi[1], Luca Benini[2] and Andrea Marongiu[1]

[1]University of Modena and Reggio Emilia

[2]University of Bologna - ETH Zürich

Introduction

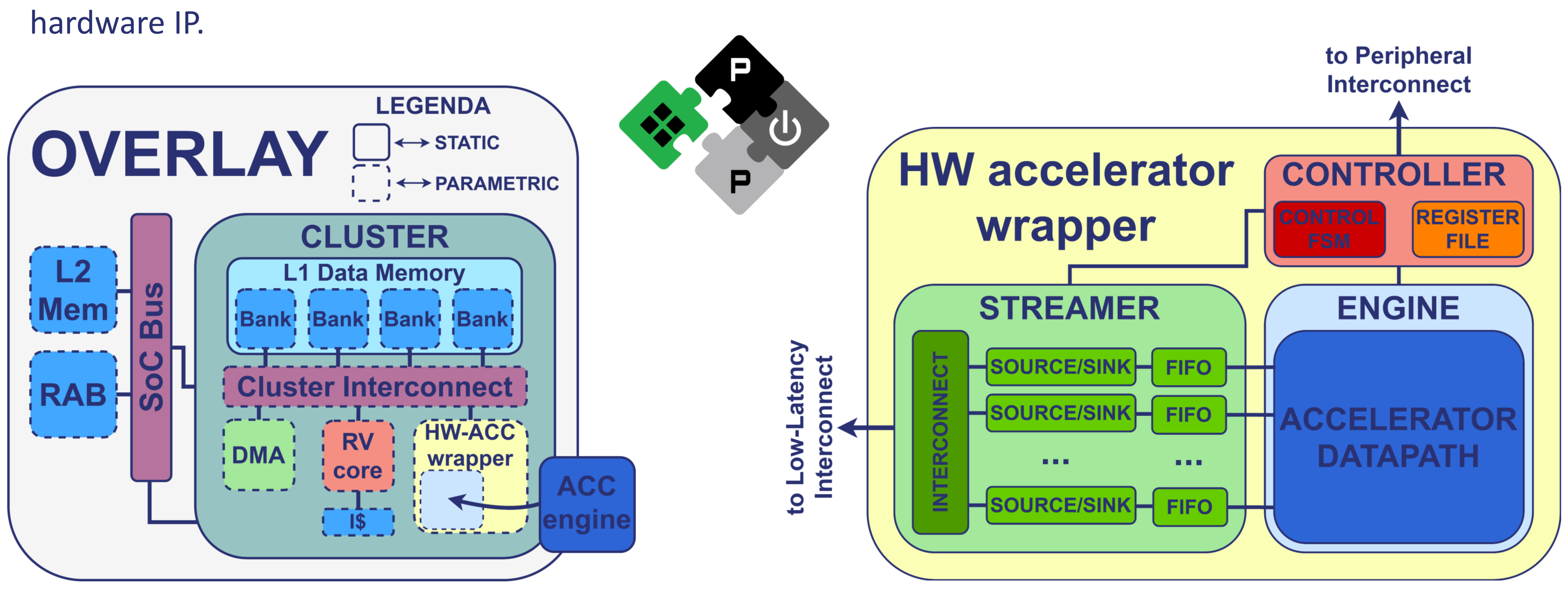
- To enable high performance and energy efficiency, **embedded heterogeneous SoCs** mandate the on-chip integration of general-purpose processors and a plethora of application-specific hardware accelerators in a so-called **accelerator-rich paradigm.**
- The design and testing of the whole system are costly and time-consuming, motivating the need for innovative **automated hardware design flows**.
- Modelling the **accelerator interaction** is a non-straightforward and challenging task.

Motivation

- **System-level design (SLD)** – To streamline the design, integration and evaluation of accelerator-rich systems employing **agile** and **reliable** methodologies.
- **Design space exploration (DSE)** – To explore the design space and find **optimal** HW/SW implementations to meet the wide range of **application-level requirements**.
- **Accelerator design** – Need for more automated **design flows** (e.g. HLS).

Methodology

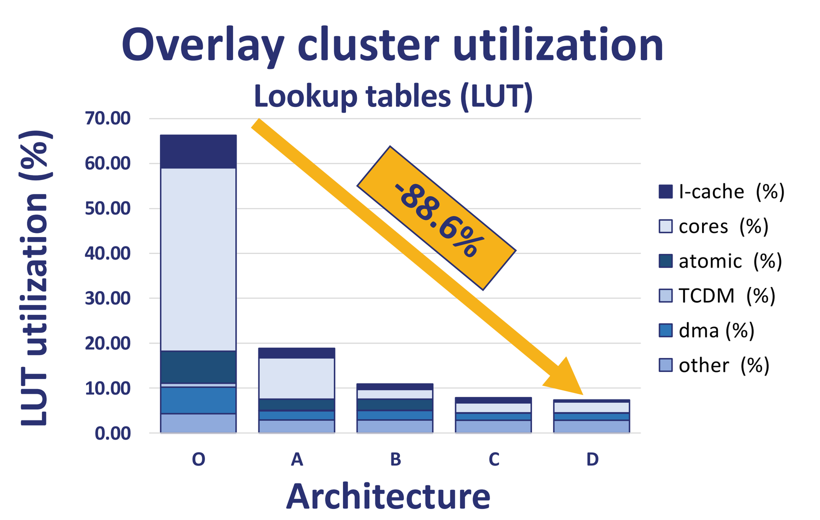
- **FPGA overlay** – Hardware abstraction layer that hides the HW details of the underlying fabric (ASIC, FPGA), thus simplifying system-level design.
- **Accelerator wrapper** – Communication and control protocol for hardware accelerators in the form of a hardware IP.

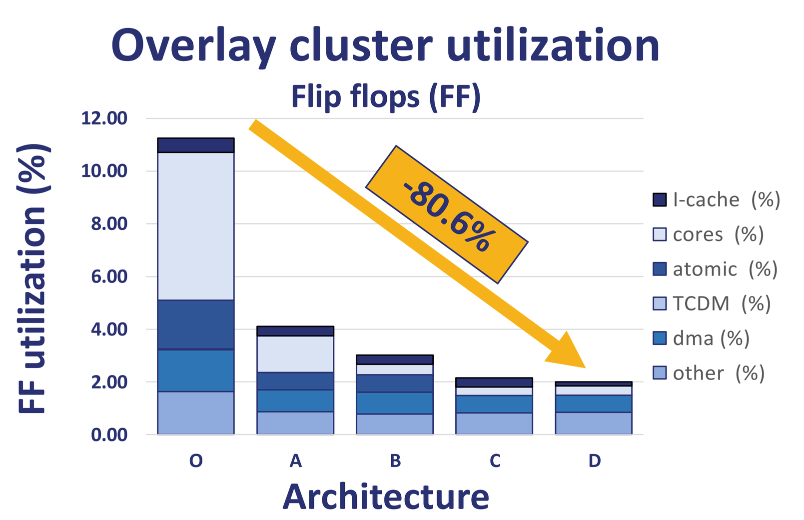


- **Application modelling** – Strong decoupling in designing the wrapper and its accelerator engine (HLS, third-party, etc.).
- **Wrapper generation** – Template-based generation of wrapper resources is fully automated and reliable. Application requirements are specified through a Python interface.
- **System generation** – Generation of a full-custom SoC design to meet the application requirements. Different strategies are to interconnect and orchestrate the accelerator wrappers, increasing the DSE region.
- **Verification and evaluation** – Full support for FPGA deployment and RTL simulation.
- **Prototyping** – Not limited to a specific fabric target: both ASIC and FPGA are good candidates!

Area results

- **Overlay cluster cost** – The less impacting the overlay, the larger the area for acceleration.
  - Goal - Resource characterization of a set of (empty) cluster configurations on a Xilinx ZU9EG MPSoC.
  - Result - Actual implementation (arch. D) costs: **LUT ≈ 20%, FF ≈ 12%, BRAM ≈ 3.8% and DSP ≈ 0%.**





References

- **G. Bellocchi**, A. Capotondi, F. Conti, A. Marongiu (2021), "A RISC-V-based FPGA Overlay to Simplify Embedded Accelerator Deployment".
- B. Boroujerdian, Y. Jing, A. Kumar, L. Subramanian, L. Yen, V. Venkatesan, A. Jindal, R. Shearer, V. J. Reddi, (2022), "FARSI: Facebook AR System Investigator for Agile Domain-Specific System-on-Chip Exploration".
- D. Giri, K.-L. Chiu, G. Eichler, P. Mantovani, L. P. Carloni (2021), "Accelerator Integration for Open-Source SoC Design".
- S. M. Neuman, B. Plancher, T. Bourgeat, T. Tambe, S. Devadas, V. J. Reddi (2021), "Robomorphic Computing: A Design Methodology for Domain-Specific Accelerators Parameterized by Robot Morphology".
- Q. Huang, C. Yarp, S. Karandikar, N. Pemberton, B. Brock, L. Ma, G. Dai, R. Quitt, K. Asanovic, J. Wawrzynek (2019), "Centrifuge: Evaluating full-system HLS-generated heterogeneous accelerator SoCs using FPGA-Acceleration".
- A. Kurth, A. Capotondi, P. Vogel, L. Benini, A. Marongiu (2018), "HERO: An open-source research platform for HW/SW exploration of heterogeneous manycore systems".
- S. Williams, A. Waterman, D. Patterson (2008), "Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures".

Acknowledgments

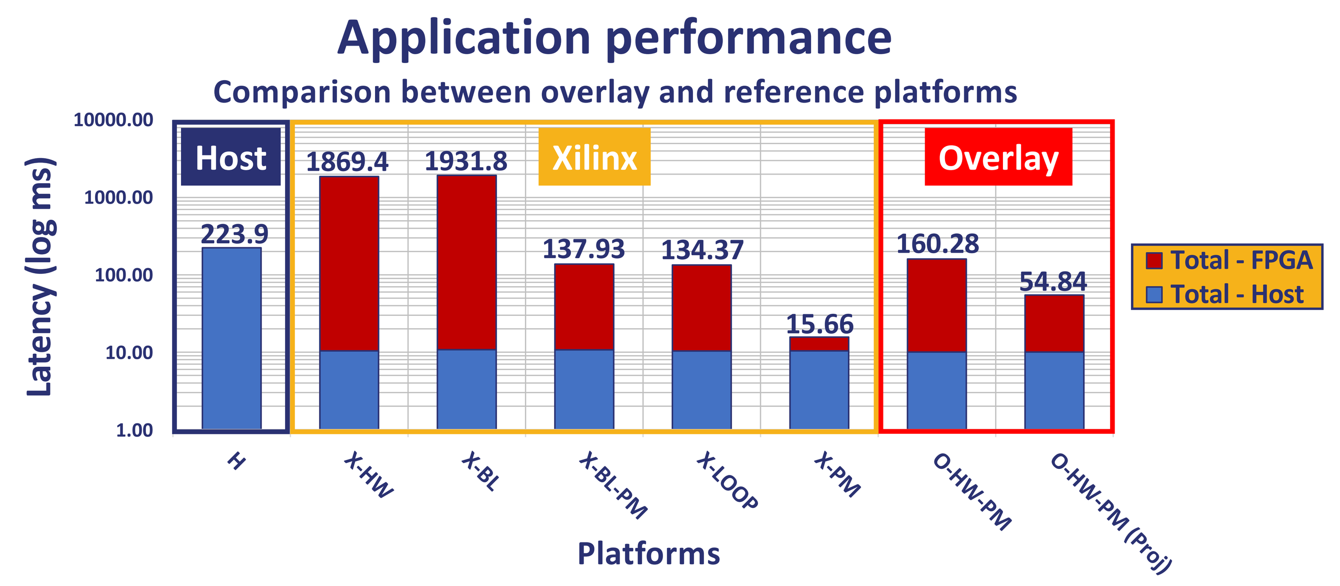
- This work has been supported by the FAR 2020 "Data Management e Data Analytics" fund. The authors would also like to thanks the EU commission for funding the ECSEL-JU COMP4DRONES project (No. 826610).

Fondo di Ateneo per la Ricerca FAR2020

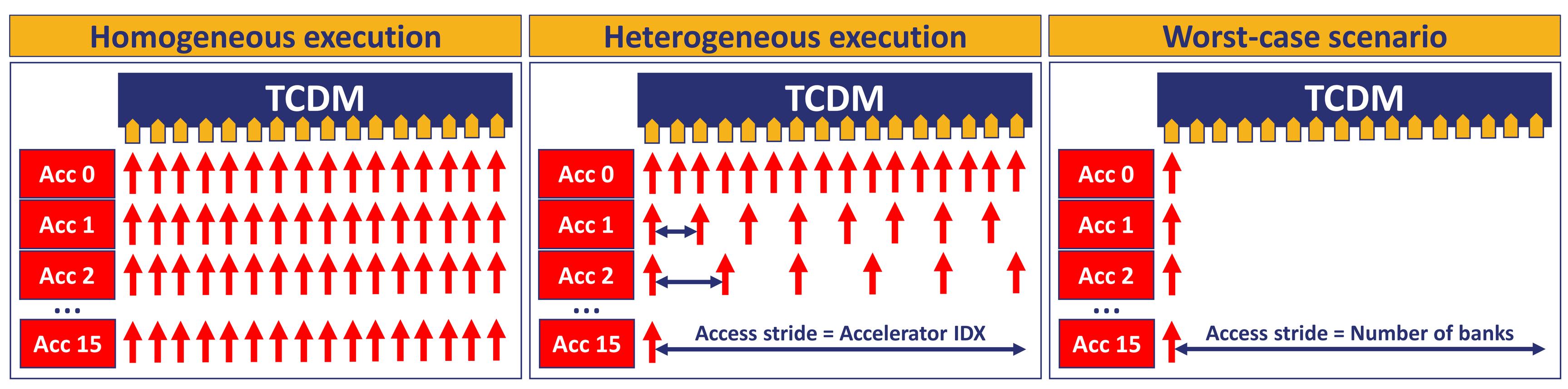
COMP4DRONES

Performance results

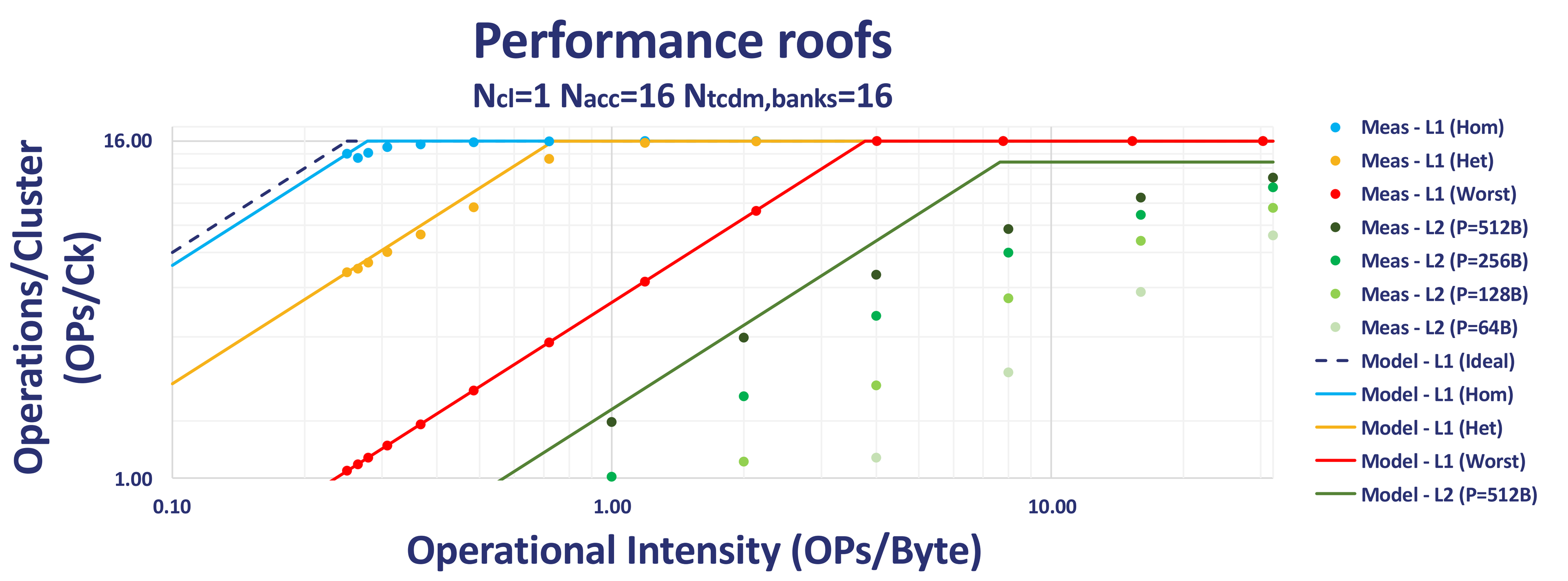
- **Application performance** – How does our solution compare to alternative platforms?
  - Goal - Comparison of overlay-based and platform applications on a Xilinx ZU9EG MPSoC.
  - Algorithm - HW/SW implementations of a **matrix multiplication (AB)** kernel.
  - SW reference - Up to **4.08x** speedup compared to execution on the **host core**.
  - HW reference - Comparison to **Xilinx HLS flow** demonstrates **comparable performance** of accelerated applications.



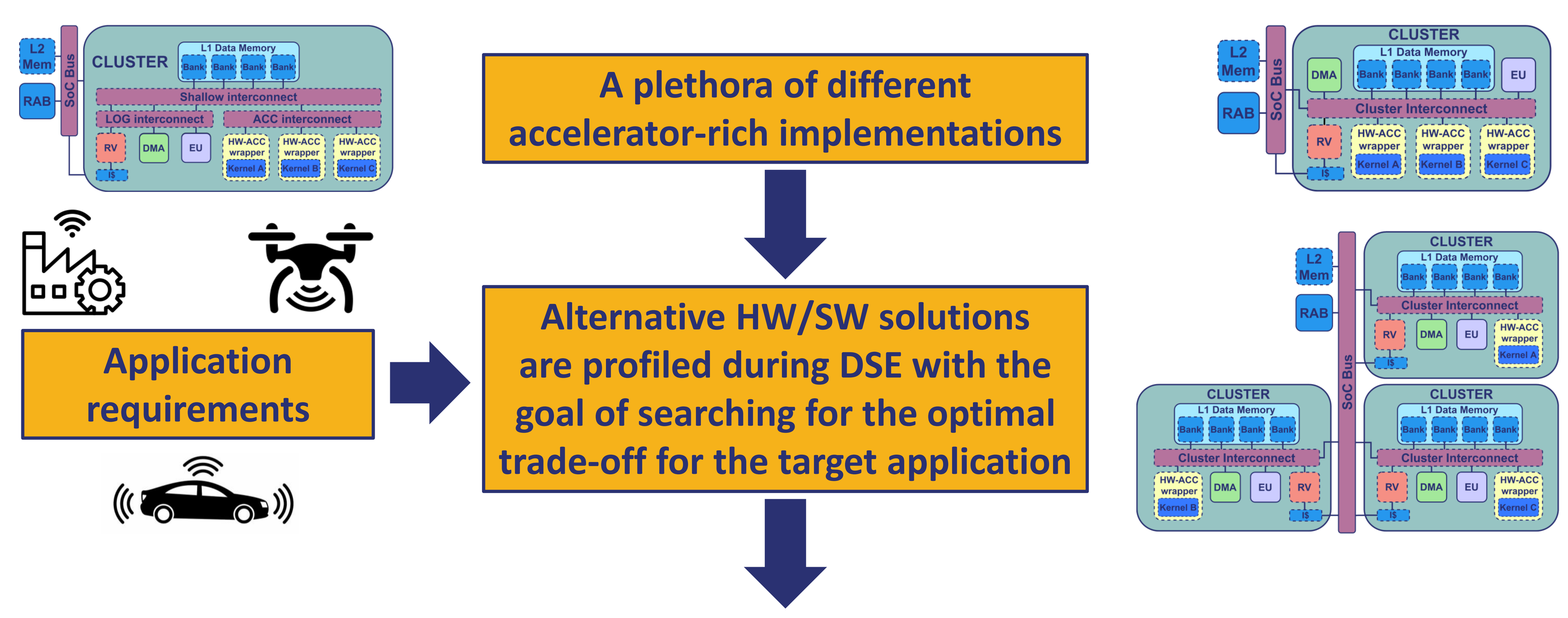
- **Accelerator-rich system evaluation** – Shared memory bandwidth and heterogeneous access patterns are constraining resources in the scalability of highly heterogeneous and dense accelerator-rich systems.
  - Goal – Performance characterization using multiple synthetic accelerators, or traffic generators (TGs).
  - Implemented system – Instantiation of a system with a single cluster that is enriched with 16 TGs.

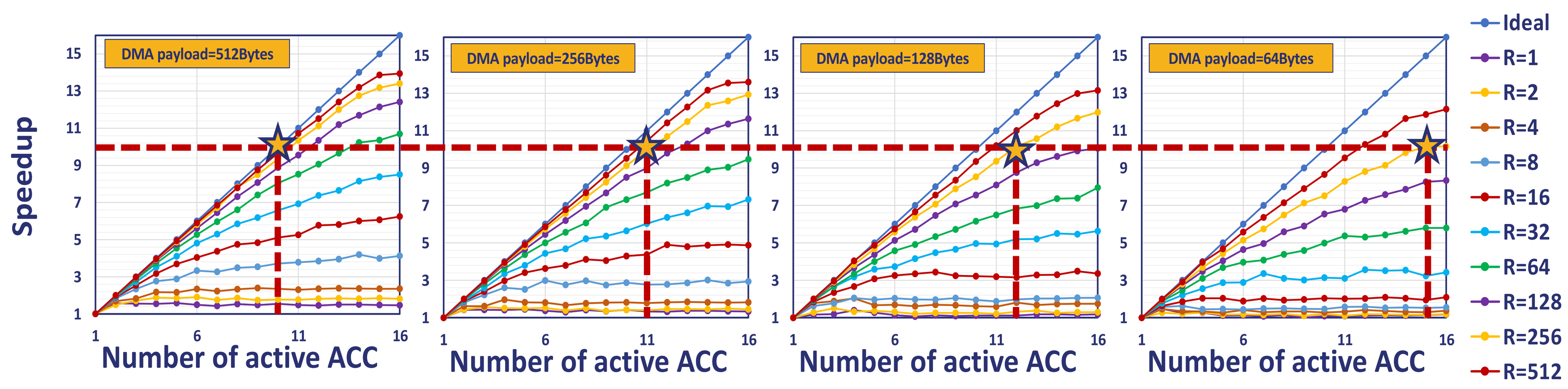


  - Interaction with L1 memory – Cluster bandwidth reduces by **2.6x** running **heterogeneous loads**, with a **5.2x** improvement over the worst-case scenario.
  - Interaction with L2 memory – Another constraint comes from **sequential DMA transfers**. According to the transferred data payload, system bandwidth further reduces from **32.7x** up to **192x** even with homogeneous access patterns.



  - Multi-cluster scaling – To solve the bottleneck is necessary to **scale the number of clusters** and distribute the accelerators according to the application requirements.
  - Design space exploration – **Automated** search of the optimal working points that fulfill **application requirements** is a goal of our methodology.





Full-stack generation of custom accelerator-rich SoC tailored to the application needs