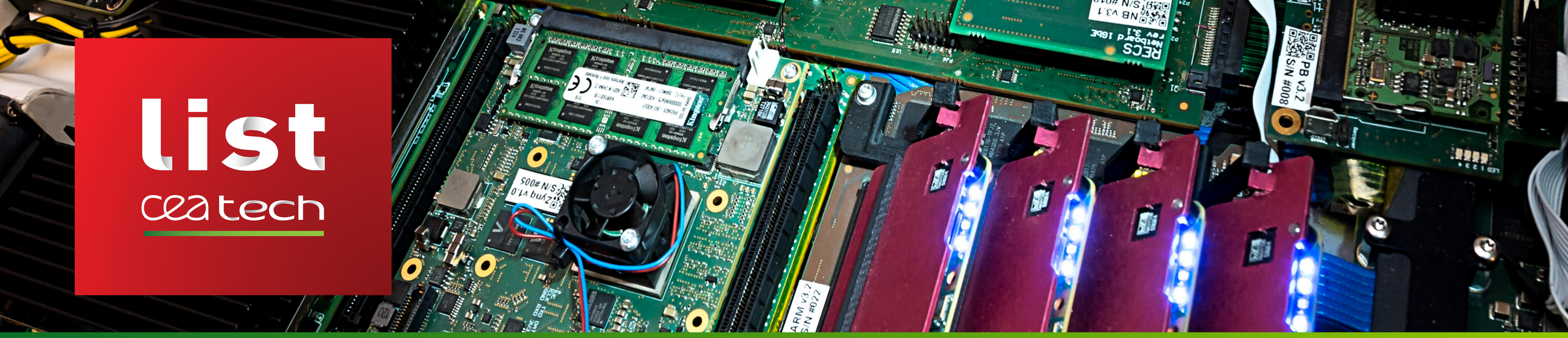
Unifying distributed memories in a heterogeneous system with reconfigurable accelerators

Erwan Lenormand, Loïc Cudennec, Henri-Pierre Charles



UNIFYING DISTRIBUTED MEMORIES IN A HETEROGENEOUS SYSTEM WITH RECONFIGURABLE ACCELERATORS

Erwan Lenormand*†, Loïc Cudennec†, Henri-Pierre Charles‡

*Univ Paris-Saclay | †CEA, List, F-91191, PC 172, Gif-sur-Yvette, France | ‡Univ Grenoble Alpes, CEA, List, F-38000 Grenoble, France

The increase in heterogeneity of computer systems requires a homogenization of the programming models

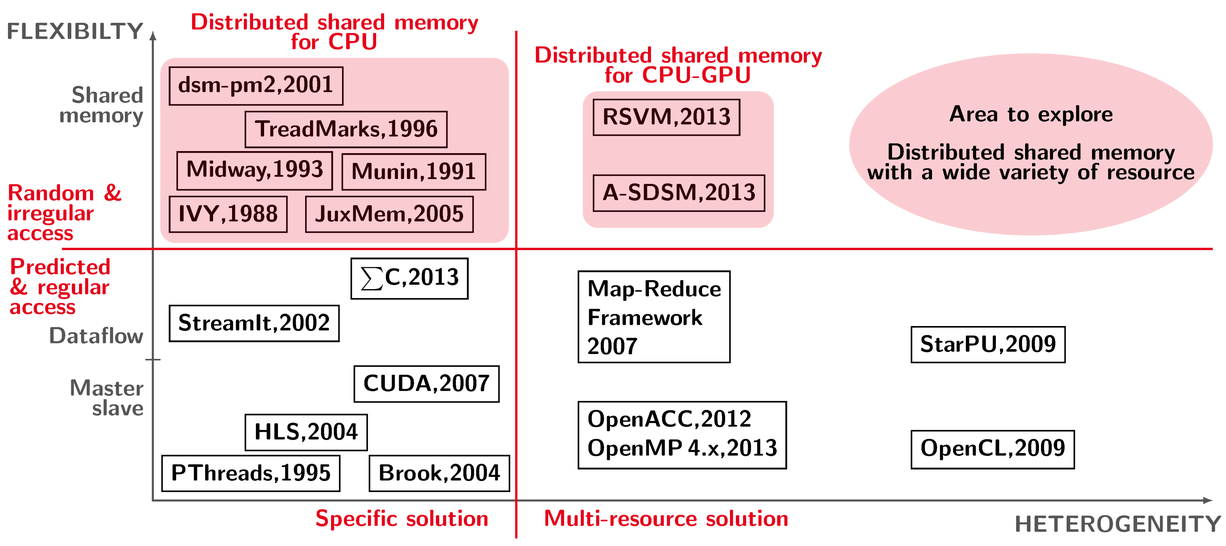
Context, Motivation & Objectives

Performance and energy efficiency: Hardware accelerators integrated in distributed systems, on a node scale, make it possible to increase computing power and to reduce energy consumption

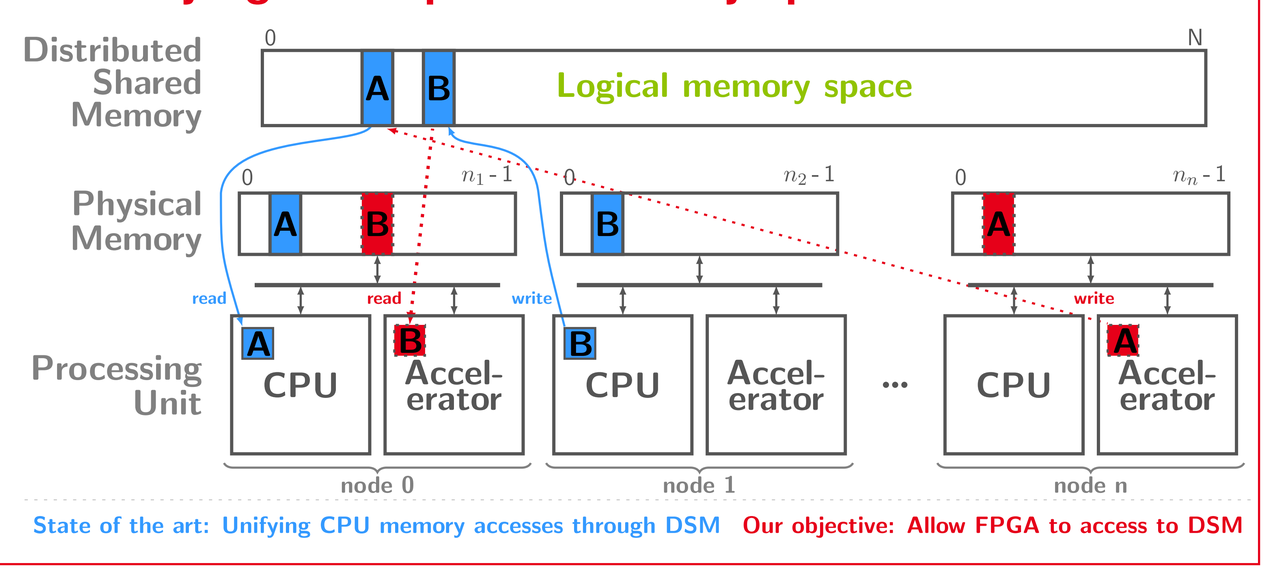
Programming complexity: These platforms are more complex to use. Compute kernels of the application are deported to accelerator functional unit (AFU) developed with accelerator programming languages. Data must be sliced and transferred to accelerators

Objectives: Provide the same task model between CPU and accelerators. Allow accelerator to access irregular data structures whose memory locations are not known at compile time

State of the art

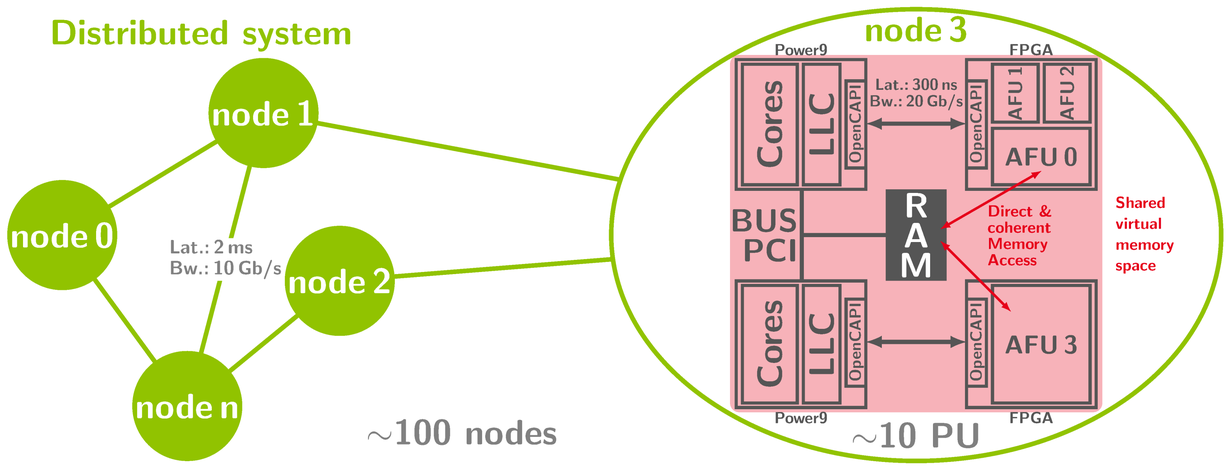


Unifying two separate memory spaces

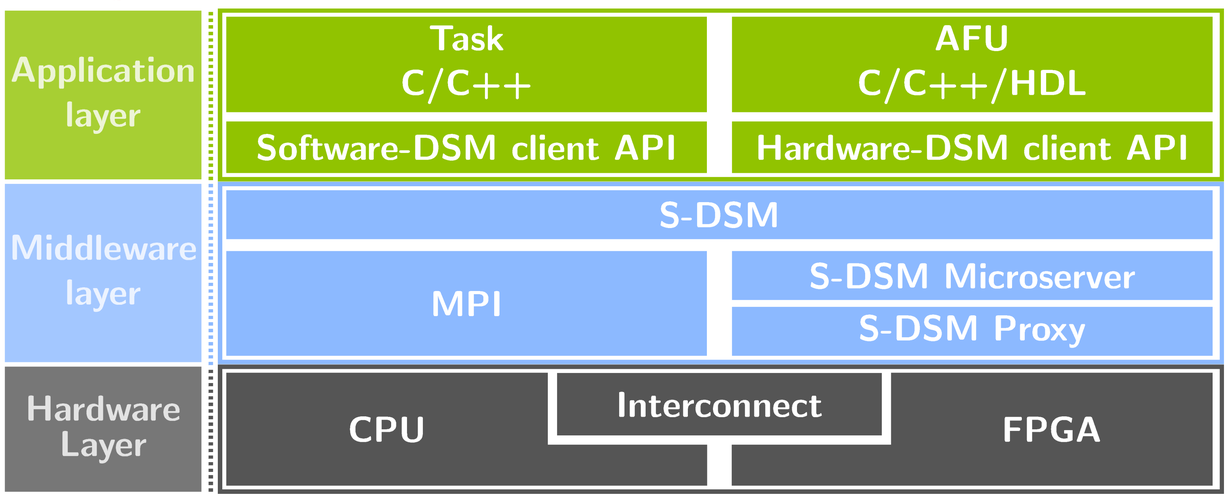


Software-distributed shared memory over heterogeneous system

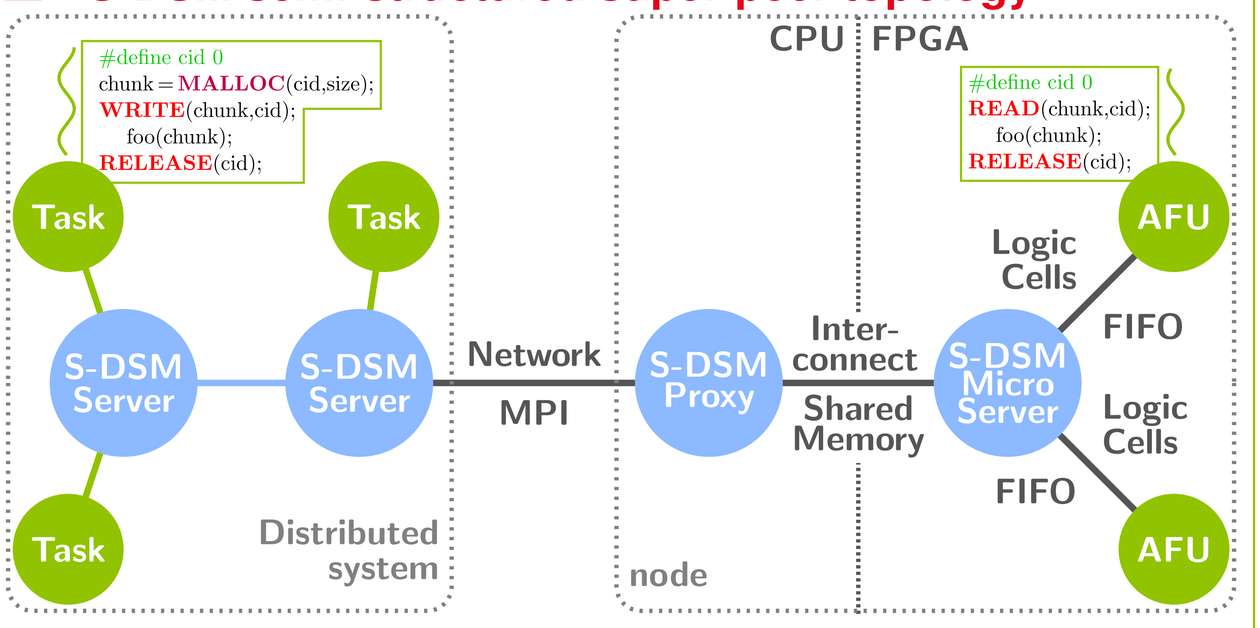
Targeted distributed system overview



System software stack



S-DSM semi-structured super-peer topology



Heterogeneous S-DSM overview

Programming model: Data are chunked in indivisible unit of memory identified by a unique number. The user expresses access to chunks with mutual exclusion primitives. These primitives encapsulate API code making service requests to S-DSM. Tasks and AFUs use the same primitives.

API: The logical organization of the S-DSM follows a client-server model. User tasks correspond to clients. Each client is attached to at least one server. The servers are in charge of data localization and access management. The access requests from FPGA are managed by a microserver. The microserver communicates with servers through a proxy process running on host CPU.

Conclusion & Outlook

Easy to use: The proposed system makes it possible to hide the topology of distributed memories by expressing data access with a common identifier for all the platform. It provides mutual exclusion primitives, used in multi-threaded programming model, to schedule data access at run time.

Improvement: Increase software portability by providing a common programming model for all resources over the system. Avoid copying data between host memory and accelerator memories through shared memory space and reduce the memory footprint.

Outlook: Evaluate this programming model for distributed applications with irregular and non-predictable memory access. Specify and implement the system.

[1] Cudennec L., Software-Distributed Shared Memory over heterogeneous micro-server architecture, In *Euro-Par 2017: Parallel Processing Workshops*, 2017.

[2] Stuecheli J., Starke W. J. et al., Ibm power9 opens up a new era of acceleration enablement: Opencapi, *IBM Journal of Research and Development*, 2018.

CONTACT | erwan.lenormand@cea.fr