Realising a Processor-Native Distributed Shared Memory Architecture

Kyriakos Paraskevas, Mikel Lujàn, John Goodacre.

Abstract

The application's memory demand has always challenged the available technologies. Ensuring fast memory access to the required capacity is crucial for defining the performance profile of any computing system. It is therefore important to understand that this demand and the consequential limitations in various aspects, led to the appearance of new memory technologies and system designs, but fundamentally, not a single solution has managed to fully solve this memory capacity challenge. With the increasing use of heterogeneous architectures, the computations are offloaded to accelerators, leading to shorter computation times, but at the same time, access to main or storage memory is bottlenecked, since all memory operations are propagated through the host itself. In such a configuration, a centralised shared memory configuration is not desirable due to scalability constraints, but instead, pools of globally accessible distributed memory can be used (Fig 1).

Any elements within a computing system that can initiate read and write transactions to memory directly from their instruction set or functionality are considered "processor-native" devices. An example of a processor native device is a CPU or an accelerator. Questions arise on how to create and map such devices on a Distributed Shared Memory (DSM) system. While such devices can be deployed natively in an embedded system, integration on a desktop or a server is not trivial.

A number of EU funded research projects focus on implementing platforms that provide what a modern HPC system can, but with lower power consumption by using heterogeneous microservers coupled with FPGA. When connecting such heterogeneous platforms together to form a cluster, efficient access to the global memory pool is an important scientific challenge.

Introduction

Computer clusters were the first configurations to eventually provide a Distributed Shared Memory (DSM) system at a linear cost while also being more scalable than the traditional cache coherent Non-Uniform Memory Access (NUMA) systems, however this was achieved by using additional software mechanisms that introduce significant latency when accessing the increased memory capacity. Since the initial software DSM systems, a lot of effort has been invested to create simpler and higher performance solutions including: software libraries, language extensions, high performance interconnects and abstractions via system hypervisors, where each approach allows a more efficient way of memory resource allocation and usage between different nodes in a machine cluster. Despite such efforts, the fundamental problems of maintaining cache coherence across a scaled system with thousands of nodes is not something that any of the current approaches are capable of efficiently providing, and therefore the requirement of delivering a scalable Global Address Space (GAS) still poses a real challenge for system architects. New design concepts and technologies, such as 3D stacked RAM and the UniMem architecture, are promising and can offer a substantial increase in performance and memory capacity. This can be beneficial for various application domains, such as Machine Learning, Deep Learning and Distributed Graph analytics, mainly because of the large number of synchronization messages required as well as the frequent element updates. By leveraging the hardware infrastructure implemented in the EuroEXA project [1,2], the aim is to overcome some of the architectural limitations of current-gen ARM architecture when it comes to memory management and address translation.

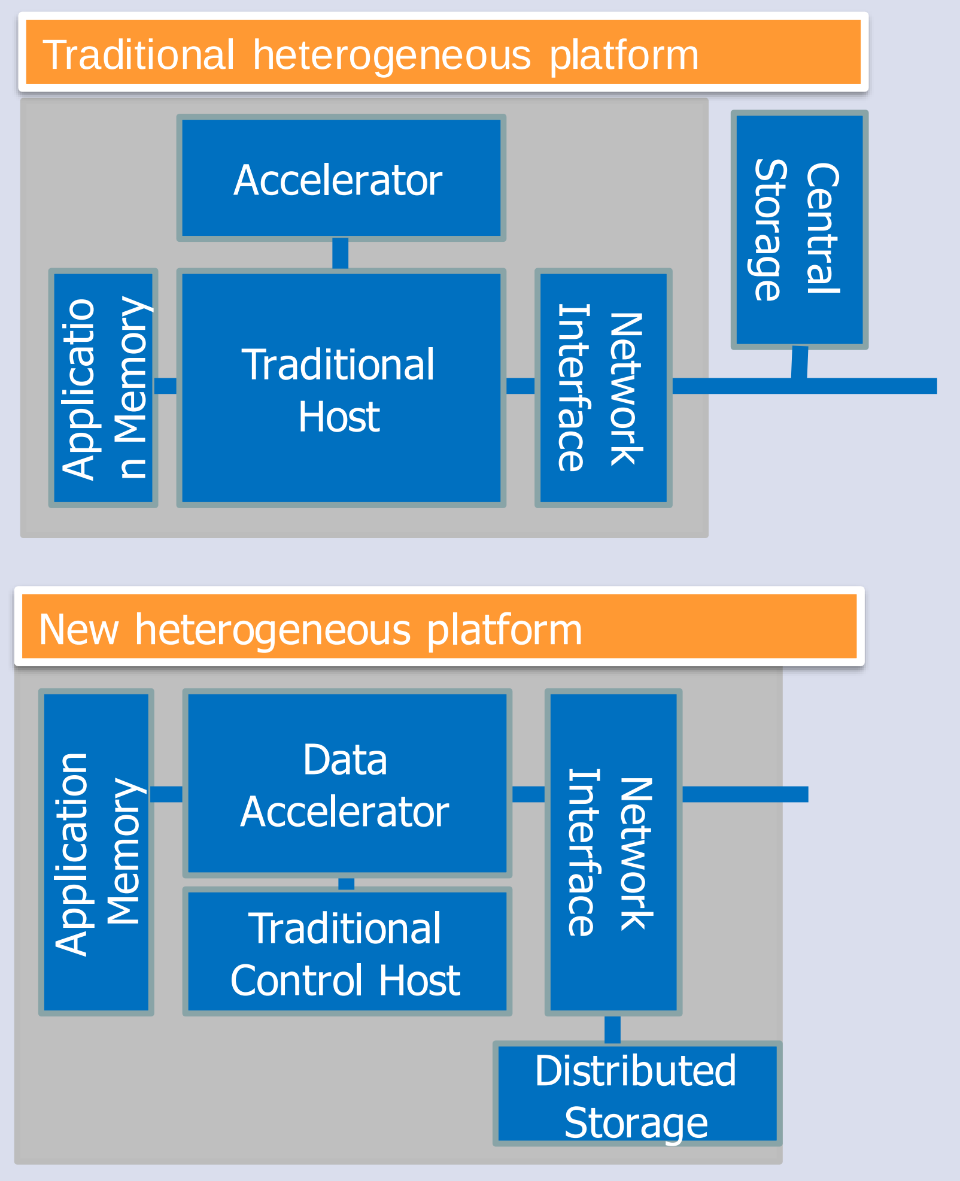

Fig 1: A new new heterogeneous platform approach

Fig 2: The cabinet that will host the liquid-cooled EuroEXA prototype system

Aims

The main research question is the effect of hardware memory management for a processor-native DSM system. Therefore,

Some of the aims of this research are:

1. Efficient address translation of virtual to physical address between processor-native elements across system
2. Tackle the implications of processor-native transactions with respect to the memory model of various buses (eg. PCI)
3. Provide a software model to expose the system capabilities through API.
4. Implement a small scale system where mapped masters and slave devices (eg. memory) are part of a DSM

Architecture

By exploiting the UniMem memory scheme communication mechanisms, such as support for load/store instructions across remote nodes, and the hardware implemented in the context of the EuroEXA project, it is possible to create a native DSM across the system. **Figure 3** provides an overview of the system, depicting how the aforementioned questions can be answered. Additionally, Master-Slave devices are attached to a "global" interconnect that stands between endpoints such as remote memory. The read/write memory transactions of a master device require translation and bridging so that the transactions are propagated to the proper slave interface. The EuroEXA project will provide such hardware.

- ❑ **A scalable memory scheme first described in the Horizon 2020 EUROSERVER project**
  - ▪ Merges the traditional dual memory types: cached and device memory
- ❑ **It enables the low cost implementation of DSM functions in hardware**
  - ▪ By exploiting the existing memory management capabilities of processors
- ❑ **It provides a future-proof 128-bit address space**
- ❑ **RDMA and processor-native load/store support**
- ❑ **Has been evaluated in previous H2020 projects (Euroserver, Exanode, Exanest, Ecoscale) and is being further demonstrated in the EuroEXA project**

Unimem-enabled system

Allowing hardware-native Load/Store to remote memory directly across networking hardware	Flexible bridge reconfiguration for a shared GAS	Provide a DSM-capable system

EuroEXA has been adding the native capability to CPU to access this GAS and this work evaluates that on DSMs

Evaluation

The aim of the evaluation is to show that processor-native is better than software-based only or transport-native approaches. This means that overall, the processor-native system will perform better in terms of memory access latency and bandwidth than a system that is using software network encapsulating protocols or than a system that is using drivers to handle the networking.

The evaluation will take place in two parts:

1. **Performance measurements and evaluation**
   - ▪ Benchmarking application performance and comparing with software-based DSM implementations
2. **Porting and running parallel applications**
   - ▪ To ensure that the system can provide the functionality of a typical cluster configuration

Progress so far

- ❑ **Adaptation of the Virtual Function I/O framework to the implemented hardware design**
  - ▪ In order to bind-unbind and virtualise accelerators to applications
- ❑ **Setting up and configuring the System Memory Management Unit on Xilinx state-of-the-art FPGA boards**
  - ▪ To enable virtual mapping consistency in various scenarios, such as when accelerators share the applications memory space.
- ❑ **Implementation of the software layers required to drive the hardware and provide a GAS.**

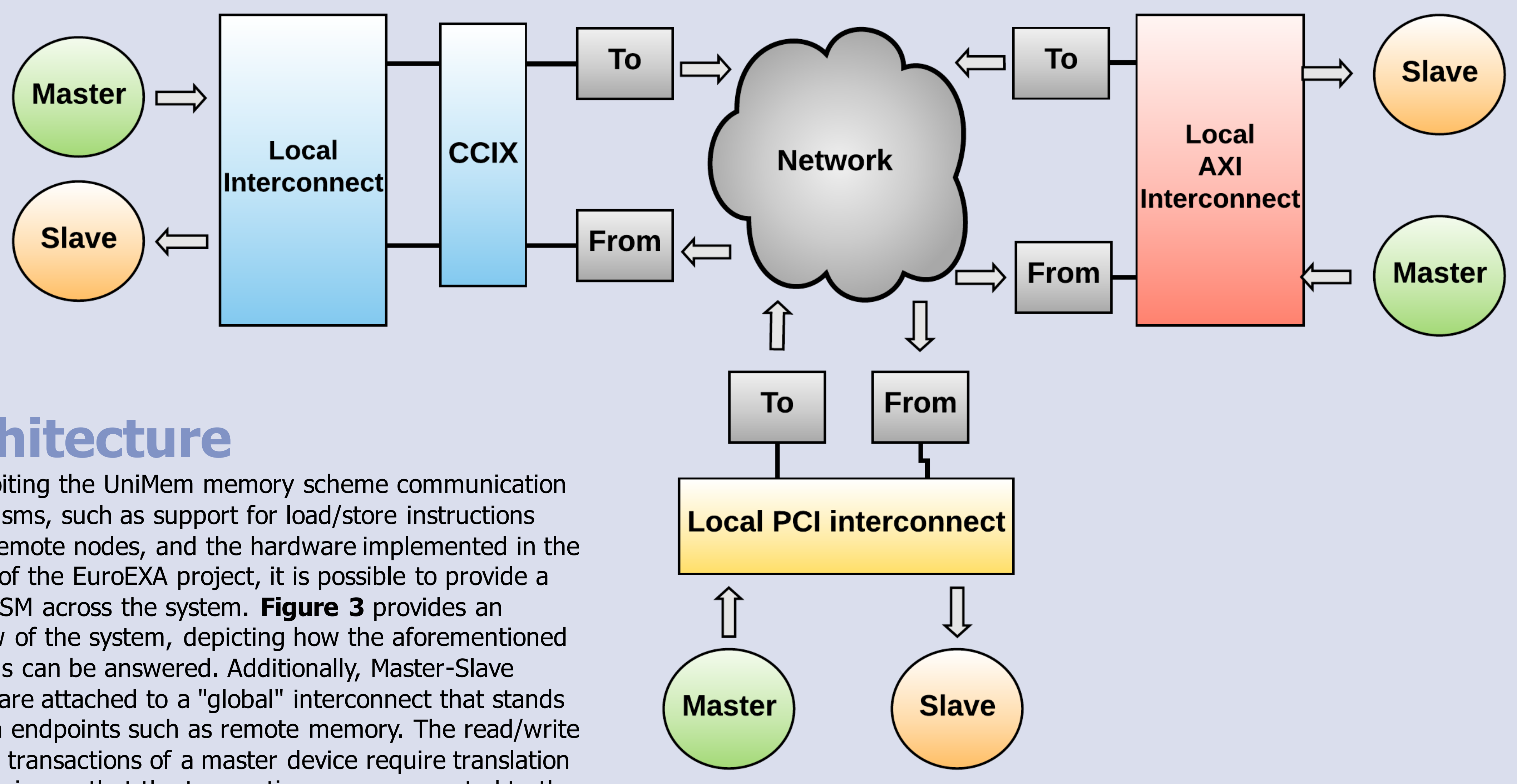

Fig 3: Evaluation framework overview. Devices on different buses are accessible with processor-native transactions. Dedicated hardware is implemented to enable such transactions across the system.

Future work

- ❑ **Testing and integration of the implemented software drivers on state-of-the-art MPSoC devices.**
- ❑ **Re-implementing the network stack of GAS-enabling libraries to make use of EuroEXA infrastructure**
- ❑ **Creation of a small-scale processor-native enabled system with a small number of system nodes connected to a central switch**
- ❑ **Performance benchmarking**

References

[1] Co-designed Innovation and System for Resilient Exascale Computing in Europe: From Applications to Silicon. http://cordis.europa.eu/project/rcn/210095{_}en.html

[2] Roberto Ammendola, Andrea Biagioni, Fabrizio Capuani, Paolo Cretaro, Giulia De Bonis, Francesca Lo Cicero, Alessandro Lonardo, Michele Martinelli, Pier Stanislao Paolucci, Elena Pastorelli, et al. 2018. Large Scale Low Power Computing System-Status of Network Design in ExaNeSt and EuroExa Projects.

[3] Kyriakos Paraskevas et al, Scaling the capacity of memory systems; evolution and key approaches