ECOSCALE : A Novel Framework for Utilising Multi-FPGAs in HPC Systems

I Mavroidis[1], P. Malakonakis[1], K. Georgopoulos[1], A. Ioannou[2], I. Papaefstathiou[2]

[1] Telecommunication Systems Institute, Technical University of Crete, Greece

[2] Synelixis Solutions SA, Greece

Abstract. ECOSCALE implements a scalable programming environment and architecture, aiming to substantially reduce energy consumption as well as data traffic and latency. ECOSCALE introduces a novel heterogeneous energy-efficient hierarchical architecture, as well as a hybrid programming environment and runtime system. The ECOSCALE approach is hierarchical and it scales well by partitioning the physical system into multiple independent Workers (i.e. compute nodes). Workers are interconnected in a tree-like fashion and define a contiguous global address space that can be viewed either as a set of partitions in a Partitioned Global Address Space (PGAS), To further increase energy efficiency, as well as to provide resilience, the Workers employ reconfigurable accelerators mapped into the virtual address space utilizing a dual stage System Memory Management Unit with coherent memory access. The implemented UNILOGIC architecture supports shared partitioned reconfigurable resources accessed by any Worker in a PGAS partition, as well as automated hardware synthesis of these resources from an OpenCL-based programming model..

Keywords: HPC, FPGAs, Parallel Systems.

1 Introduction

In order to reach exascale performance current HPC servers need to be improved. Simple scaling is not a feasible solution due to the increasing utility costs and power consumption limitations. Apart from improvements in implementation technology, what is needed is to refine the HPC application development as well as the architecture of the future HPC systems.

ECOSCALE tackles this challenge by employing a scalable programming environment and hardware architecture tailored to the characteristics and trends of current and future HPC applications, reducing significantly the data traffic as well as the energy consumption and delays. In particular ECOSCALE pairs a refined version of the existing UNIMEM (Unified Memory) architecture along with the novel UNILOGIC (Unified Logic) architecture. UNIMEM provides a uniform memory address space across FPGAs. UNILOGIC comes as an extension to this, and introduces the uniform and virtualized access of logic, i.e. of acceleration resources, residing in a multi-

FPGA system. Since a fundamental aim is the graceful scaling to a very large number of nodes (i.e. FPGAs), the proposed architecture is developed to be inherently scalable.

2 UMIMEM

The UNIMEM architecture [1] was first introduced within the EuroServer project [3]. It consists of a powerful set of mechanisms that provide efficient communication among the remote CPU-based nodes of a large computational system. The UNIMEM architecture gives the user the option to move tasks and processes close to data instead of moving data around [2] and thus it reduces significantly the data traffic and the associated energy consumption and communication latency. From the point of view of a processor in a multi-node machine, a memory page can be cacheable at the local coherent node or at a remote coherent node, but not at both. This is the basis of the UNIMEM consistency model, which eliminates global-scope cache coherence protocols providing a scalable solution. Progressive address translation [4] can be further applied on top of UNIMEM in order to provide interprocessor communication.

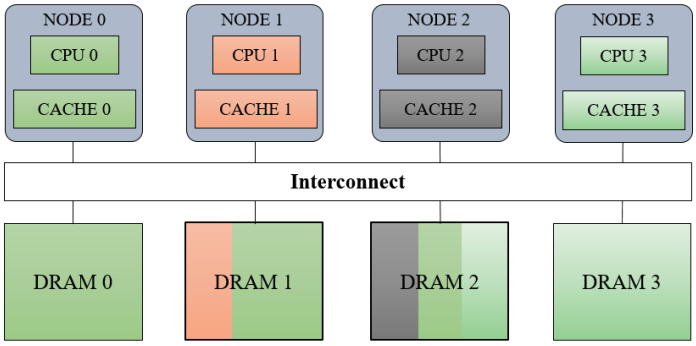

Figure 1: UNIMEM allows pages of DRAM1 cached in CACHE0 of CPU0 -OR- in CACHE1 of CPU1 etc.

An HPC system implementing the UNIMEM architecture consists of a set of computational nodes that are connected through a custom network. UNIMEM enables the nodes to seamlessly access areas of memory located in remote nodes. More specifically, in the UNIMEM architecture, there is a global address space (GAS) that it is accessible to any node. Therefore, a node in the system has the ability to directly access the physical memory of any node through the GAS. UNIMEM allows remote DRAM borrowing and remote load/store instructions, which enable remote-mailbox and remote-interrupt notifications for low-latency protocols. It also allows communication using Remote Direct Memory Access (RDMA) operations, which efficiently deliver data in-place and avoid receiver-side copying. The complexity and costs that the system-level coherence protocols induce [6], are eliminated in the UNIMEM architec-

ture, as it imposes that each page of the physical memory can be cached by at most one node (Figure 1 shows such a use-case).

Due to the above unique characteristics, UNIMEM behaves as an evolution of both shared memory and message passing parallel architectures; *Shared Memory*: All memory within the entire GAS is accessible by any node using conventional load and store instructions. Also to avoid the high cost of system-wide hardware cache coherence protocols, a single node at most can cache each page. *Message Passing*: The UNIMEM architecture allows making bulk data transfers directly into the receiver's memory, i.e., zero-copy RDMA. Therefore, system's efficiency is enhanced in terms of both performance and energy.

The UNILOGIC architecture is introduced within the ECOSCALE project for the first time [7], and comes as an extension to the UNIMEM architecture. UNILOGIC adds to UNIMEM the capability to easily access acceleration engines across the entire system and furthermore to easily relocate the hardware acceleration engines and e.g. bring the locally, for instance through dynamic partial reconfiguration [5]. The proposed UNILOGIC+UNIMEM architecture partitions the design into several Worker elements that communicate through a hierarchical communication infrastructure, similar to the one shown in Figure 2. These Worker elements correspond to the partitions of the HPC application. Each Worker node is an entire sub-system including processing units, memory, and reconfigurable logic. Within a Partitioned GAS (PGAS) domain that may consist of several Workers, the proposed architecture offers (1) a shared global address space that can be partitioned for locality and (2) shared reconfigurable resources that can also remotely access cached data via regular load and store instructions, or via initiating block transfers in an RDMA fashion.

3 UNILOGIC

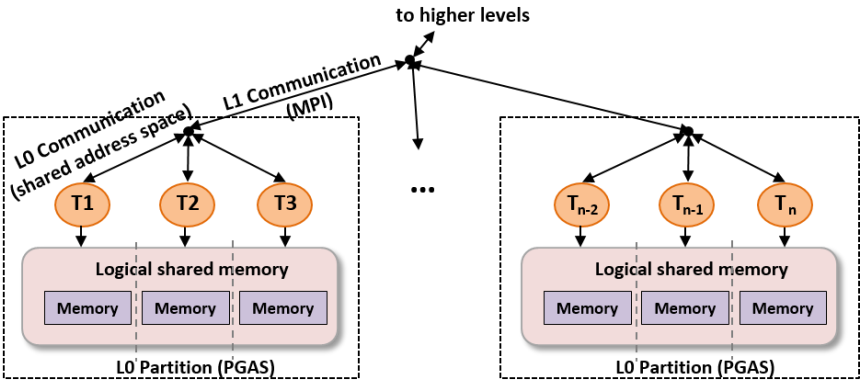

Figure 2: Example hierarchical partitioning (tasks, memory, communication) of an HPC application

This novel system architecture uses CPUs, memory and reconfigurable blocks in a highly parallel manner. Driven by the characteristics and trends of future HPC appli-

cations and following the high-radix partitioning of an HPC application as seen in Figure 2, the proposed UNILOGIC+UNIMEM architecture logically partitions the hardware resources hierarchically (CPUs, reconfigurable logic, memories) into several interconnected Compute Nodes (corresponding to the PGAS partitions of the application) which are further partitioned into several Worker elements, depending on the physical structure of the system. Thus, one or more Compute Nodes create an entire and independent PGAS sub-system including several Worker nodes and offer:

1. UNIMEM: a shared partitioned global address space that allows Worker elements to communicate via regular loads and stores without the drawback of global cache coherence and
2. UNILOGIC: shared partitioned reconfigurable resources that share the UNIMEM space along with the software tasks.

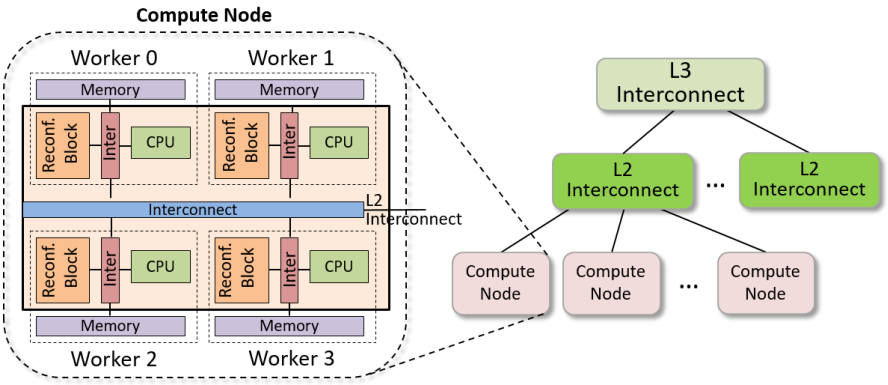

Figure 3: The ECOSCALE Hardware Architecture

The proposed HPC architecture, where (1) each Compute Node is a PGAS subsystem providing a shared address space and reconfigurable acceleration logic, and (2) MPI is used for communication between Compute Nodes via CPU-based routers following the application topology, is shown in Figure 3. It consists of several Worker elements communicating through a multi-layer interconnection. The actual number of Workers inside a Compute Node depends on the integration capabilities of current and future technologies. Each Worker is an independent computing unit that can execute, fork and join tasks or threads of an HPC application in parallel with other Workers. It includes a CPU, reconfigurable logic and off-chip DRAM memory. The communication and synchronization between the Workers and through the multi-layer interconnection, allows load and store commands, DMA operations, interrupts, and synchronization between the Workers of a Compute Node. The Compute Node's PGAS sub-systems correspond to the application's PGAS-based partitions shown in Figure 2. Matching the application logical topology of Figure 2, the Compute Nodes are interconnected through an MPI-based multi-layer interconnection.

In an HPC application, virtualization and context switching enables multiple tasks or threads to share a single CPU in order to maximize the utilization of the CPU resources. Similarly, our architecture supports fine-grain sharing of the FPGA re-

sources, where a function implemented in hardware can be “called” by different tasks or threads of an HPC application in parallel, through a custom hardware Virtualization block. This scheduling block and the High Level Synthesis (HLS) tool provide a mechanism to execute multiple function calls in a fully pipelined fashion. Moreover, the UNILOGIC architecture supports coarse-grain time-sharing of the reconfigurable resources through partial runtime reconfiguration. Partial reconfiguration is supported either locally or remotely (i.e. partially reconfiguring remote FPGA logic) and has been already integrated in our platform.

4 Conclusions

ECOSCALE provides a novel methodology and architecture to automatically execute HPC applications onto an HPC platform that supports thousands or millions of reconfigurable hardware blocks, while taking into account the projected trends and characteristics of HPC applications. Within this context, the project links and extends various disconnected existing FPGA-based acceleration approaches and adapt them so as to work in an HPC environment. In order to efficiently do so, ECOSCALE follows a holistic approach providing solutions for all the aspects of an HPC environment based on two innovative architectures, the UNILOGIC and the UNIMEM.

5 Acknowledgements

This research project is supported by the European Commission under the H2020 Programme and the ECOSCALE project (grant agreement 671632).

6 References

1. M. Marazakis et al. 2016. EUROSERVER: Share-anything Scale-out Micro-server Design. In Proceedings of the 2016 Conference on Design, Automation & Test in Europe (DATE ’16) . 678–683.
2. Y. Durand et al. 2014. EUROSERVER: Energy Efficient Node for European Micro-Servers. In 17th Euromicro Conference on Digital System Design, DSD 2014, Verona, Italy, August 27-29, 2014 . 206–213.
3. EU. 2013-2017. The Euroserver Project. http://www.euroserver-project.eu.
4. M. Katevenis. 2007. Interprocessor Communication seen as Load-Store Instruction Generalization. In The Future of Computing, essays in memory of Stamatis Vassiliadis . K. Bertels e.a. Editors, Delft, The Netherlands, 28 Sep. 2007, 55–68.
5. D. Koch. 2012. Partial Reconfiguration on FPGAs–Architectures, Tools and Applications .
6. J. Laudon and D. Lenoski. 1997. The SGI Origin: A ccNUMA Highly Scalable Server. In Proceedings of the 24th International Symposium on Computer Architecture, Denver, Colorado, USA, June 2-4, 1997 . 241–251.
7. I. Mavroidis, et al, “ECOSCALE: Reconfigurable computing and runtime system for future exascale systems”. In 2016 Design, Automation & Test in Europe Conference & Exhibition, DATE 2016, Dresden, Germany, March 14-18, 2016 . 696–701.