Paving the way towards a highly energy-efficient and highly integrated compute node for the Exascale revolution: the ExaNoDe approach

Alvise Rigo [1], Christian Pinto [1], Kevin Pouget [1], Daniel Raho [1], Denis Dutoit [2], Pierre-Yves Martinez [2], Chris Doran [3]
Luca Benini [4], Iakovos Mavroidis [5], Manolis Marazakis [5], Valeria Bartsch [6], Guy Lonsdale [7], Antoniu Pop [8]
John Goodacre [8], Annaïk Colliot [9], Paul Carpenter [10], Petar Radojković [10], Dirk Pleiter [11], Dominique Drouin [12]
Benoît Dupont de Dinechin [13]

[1] Virtual Open Systems, France
[2] CEA-LETI, France
[3] ARM, United Kingdom
[4] Eidgenössische Technische Hochschule Zürich, Switzerland
[5] Foundation for Research and Technology Hellas, Greece
[6] Fraunhofer Institute, Germany
[7] scapos AG, Germany
[8] University of Manchester, United Kingdom
[9] Atos-Bull, France
[10] Barcelona Supercomputing Center, Spain
[11] Forschungszentrum Jülich, Germany
[12] Laboratoire Nanotechnologies Nanosystèmes (LN2) - CNRS, Canada
[13] Kalray, France

***Abstract*—Power consumption and high compute density are the key factors to be considered when building a compute node for the upcoming Exascale revolution. Current architectural design and manufacturing technologies are not able to provide the requested level of density and power efficiency to realise an operational Exascale machine. A disruptive change in the hardware design and integration process is needed in order to cope with the requirements of this forthcoming computing target. This paper presents the ExaNoDe H2020 research project aiming to design a highly energy efficient and highly integrated heterogeneous compute node targeting Exascale level computing, mixing low-power processors, heterogeneous co-processors and using advanced hardware integration technologies with the novel UNIMEM Global Address Space memory system.**

I. Introduction

The global race for building ever larger and more powerful supercomputers has over the years led to the creation of a plethora of different architectures, and driven various "revolutions". This race has been recorded over the years in the *Top500 Supercomputers list* [1], where the most powerful supercomputers built so far are ranked based on peak floating point (FP) performance. At the present moment the highest FP performance registered in the list is from yet another new architecture delivering 100 Petaflops (10^{15} FP operations per second). However, even such performance will soon not meet the needs of increasingly complex applications and the research community is already investigating the development of *Exascale computing systems*. The target is to build supercomputers capable to reach a peak-performance in the order of the Exaflop (10^{18} FP operations per second). This trend is a clear priority worldwide, with programs in Europe [2], USA [3], China [4] and Japan [5].

However, the previous considerations are based on results of simple benchmarks, designed to measure the peak theoretical performance of a computing system with effective real-world performance at less than 5% of peak. The real questions are: **why do we actually need Exascale? How can more cost effective performance be realised? What are the concrete benefits of Exascale computing?** There are several answers [6] to support Exascale computing, but it is worth noting that advances in the understanding of some scientific problems are often coupled with advances in computing capabilities. Higher computing capabilities, beyond current state-of-the-art Petascale systems, will help scientists in designing models and running simulations of much more complex problems that could help improve our every-day life (e.g. computational biology, climate control, research on energy, etc.). In addition such huge computing power will have a strong impact on industry as well (e.g., oil exploration and production, aerospace, pharmaceutical, etc).

Despite the strong motivations for Exascale and the commitment of various government agencies, achieving such a goal is not a matter of just adding a number of cabinets to an

existing supercomputer in order to increase its horse-power. To reach the Exascale level, computing systems will need a complete overhaul on how to deal with the challenges in the areas of cost, power, compute efficiency, resiliency and fault tolerance, data movement across the network and, last but not least, programming models. At the heart of this process we can find the compute element which is the fundamental building block of a computing system, and should be considered as the starting point of any research activity towards Exascale.

The European H2020 ExaNoDe [7] research project investigates, develops, integrates and validates the building blocks for a compute element that enables Exascale levels of computation. *The principle of the project is to apply novel 3D integration and hardware design technologies, mixed with virtualization of resources and the UNIMEM memory system to deliver a prototype-level system demonstrating that those technologies are promising candidates towards the definition of a compute node for the Exascale computing.* The ExaNoDe project idea is built around the following design goals:

- **Affordability**: Ensure the solution is commercially viable and competitive in both its performance and its cost of ownership.
- **Design Efficiency**: Using power-efficient compute elements and System design principles to ensure minimum duplication and abstractions within the infrastructure, to avoid unnecessary power consumption and latency overheads.
- **Operational Efficiency**: power consumption proportional to activities making actual progress.
- **Everything Close**: Leverage physical distance and data locality to design for minimum resistance and capacitance, so as to deliver the lowest power overhead associated with the required data connectivity.

ExaNoDe is also closely collaborating with the ExaNeSt [8] and ECOSCALE [9] H2020 projects. ExaNeSt is investigating how storage, interconnections and cooling systems will have to evolve towards Exascale. ECOSCALE, instead, aims to provide a holistic approach for a novel heterogeneous energy-efficient hierarchical architecture, a hybrid MPI+OpenCL programming environment and a runtime system for exascale machines. The combination of these three projects aims at covering the whole picture of an Exascale HPC machine.

The rest of the paper is organized as follows: Section II motivates the need for a disruptive change in the design of compute elements and outlines the ExaNoDe approach. Section III presents the main approach of the project and the high level architecture adopted. Section IV gives an overview of the expected technological results of the ExaNoDe project. Section V concludes the paper summarizing its contribution.

II. Motivations

Among all the essential criteria for the construction of an HPC system aiming at the Exascale level, the ExaNoDe project will address: *power consumption* and *usability of compute resources*. The former, power consumption, is a problem which is immediately apparent considering that today's supercomputer power consumption is already in the order of megawatts (MW). Usability of compute resources is, however, more related to the provision of architectural support enabling a broader range of HPC applications on the same architecture. Both criteria are aligned with the ETP4HPC Strategic Research Agenda [10], outlining the roadmap to achieve Exascale computing capabilities within the European HPC community.

A. Power Consumption

As already stated, supercomputing is an expensive business for research centres and government institutions, reflected in an extremely high investment in terms of management (e.g., cooling, staffing, power provisioning infrastructure, etc.) and power. In 2012 the Oak Ridge national labs introduced their supercomputer named Titan (9MW power consumption, 27 PF theoretical peak performance), impacting the centre's electricity bill to the tune of approximately 9 million $. It is clear that scaling towards the Exascale with the technology available today would lead to peak power consumptions of over 100MW, with an immediate consequence on the cost for management and power. Going in that direction would simply make Exascale not worth the investment, beside the technical difficulties in providing such huge amount of power. The U.S. Department of Energy [11] has set the upper limit power consumption for an Exascale machine to 20 MW, which has also been widely accepted by the whole HPC community.

Speaking of energy efficiency at the system level in terms of Joules per operation (J/Op), the #1 Top500 supercomputer of November 2016 [12] delivers 125PF theoretical peak performance at around 15MW of power that translates into 120 pJ/Op. However, with only a 0.3% HPCG (High Performance Conjugate Gradients [13]) efficiency, it is clear this power efficiency can not be delivered in real-world applications. If we now consider an Exaflop machine (1000PF) with a target power consumption of 20MW, we would need an energy efficiency of $\sim$**20 pJ/Op**. The conclusion is that the key to meet the expected power target is a high improvement in the power efficiency of the overall computing system, driven by a disruptive change in both hardware design and integration process.

As already known and stated in [14], most of the power needed to compute a flop is not due to the actual computation, but rather to data movement between memory and the FPU pipeline. The ExaNoDe project will address increased energy efficiency by the optimisation of data movement using the *Silicon Interposer* (Subsection III-A) technology, to implement the **Everything close** concept at the design level of the compute element. The result will create an unprecedented dense compute system, thanks to 3D integration and nano-technologies, to drastically reduce the energy needed for data movement. The main idea is to reduce the physical distance between components, and make sure that compute elements fully exploit physical locality with memory storage.

B. Usability of Compute Resources

One main problem, that sees no solution today, is to provide applications from different domains with the requested level of performance and flexibility. This is especially true in the supercomputing environment where application programmers are always looking for best performance, but different classes of problems/applications have different needs in terms of I/O capabilities, memory bandwidth, availability of accelerators. As an example there are no processors that provide significantly higher I/O bandwidth, forcing I/O bound applications to be limited by the available hardware interfaces. This happens because supercomputing is to be considered as a niche market, and the extremely high non-recurring engineering (NRE) costs make the design and implementation of specialized processors not commercially viable.

The ExaNoDe project addresses this issue thanks to the package level integration and separation of the building blocks of a compute element (compute, memory, I/O, acceleration) based on nanotechnology, the *Chiplet* approach (Subsection III-A). This new and promising design concept allows multiple combinations of a compute element to be built with a significantly lower NRE cost, tailored to specific application scenarios.

In addition, the Chiplet approach will also help to reach higher compute density, thanks to the integration of multiple compute elements and memories on the same chip. An example of the benefits of this approach is related to compute accelerators (e.g. CPUs, Many-core CPUs, FPGAs), widely used in HPC to boost the performance of some classes of applications. However, such accelerators are usually integrated in a compute node via PCI-e links, thus limited by the memory bandwidth of such interface. A better integration in the compute node is needed in order to provide accelerators faster access to DRAM for on-the-fly data processing, and faster interaction with the main processor. The Chiplet approach combined with the Silicon Interposer technology (used already by some Xilinx products [15]) will be beneficial to the example of accelerators because of the higher level of integration of multiple compute units and memories within the same compute node. This will result in a higher compute density enlarging the application scope of a single compute node, and help in scaling-up the performance of a compute system towards Exascale.

III. The ExaNoDe Approach

The main goal of the ExaNoDe project is to deliver a prototype-level compute element integrating some core technologies that are in line with an HPC system targeting Exascale. The aim is not to create a production level compute element, but instead to pave the way with technology towards Exascale by providing system integrators and software teams a tool for further research, oriented to industrial applicability. However it is not enough to build the core technology of the compute element to actually reach Exascale performance at the applications level. Programming such future massively parallel and heterogeneous systems is not going to be trivial, and programming models and runtime systems will have to be adapted/re-designed accordingly. Another important issue to be considered is resilience, since increasing the number of compute units and their complexity will inevitably lead to an increase of the failure rate. Even though the main focus of the project is to prototype the compute element, partners of the consortium will address programmability and resilience by investigating parallel programming models and virtualization techniques.

The following subsections describe the main technical pillars at the basis of the ExaNoDe project, namely:

A. *Silicon Interposer technology*
B. *Novel memory system*
C. *Software infrastructure*

Each subsection highlights some of the challenges related to the corresponding pillar that will be tackled during the project.

A. Silicon Interposer technology

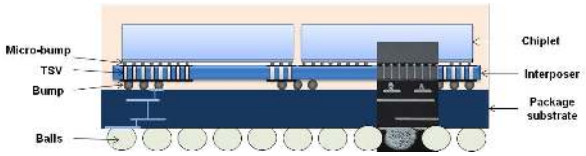

Fig. 1: Silicon Interposer approach

The interposer approach [16] (Fig. 1), by combining the advanced integration technologies with dynamic allocation and management of memory, effectively reduces inter-chip distances, increases compute density while also decreasing transfer energy and latency, increases communication bandwidth [17], and vastly improves system miniaturisation and hence acquisition and ownership costs. It addresses one of the most challenging problems to reach Exaflop level of performance: energy efficiency. In addition, some other important advantages of the interposer approach are: the reduction of costs and time to market for new SoC models, ability to interconnect chiplets manufactured using different technologies, easy adaptation of components to different target markets (e.g. HPC, cloud computing, networking, etc.).

The main idea of the ExaNoDe integration concept is to create a *Modular Compute System* partitioned into a number of chiplets stacked on a Silicon Interposer, several of them being integrated with memory devices and FPGA on a Multi-Chip-Module (MCM), see Fig. 2. This approach improves silicon fabrication yield and hence lowers acquisition and ownership costs. As an example, in case of hardware failures it will be possible to replace only the failed chiplet or memory module, instead of replacing the whole chip. At the same time, by connecting memory next to each compute Chiplet, we bring the memory closer to each computation subsystem reducing the processor to memory communication power of today's DDR memories interconnected with processors on blades.

The partitioning envisioned in the ExaNoDe project prototype is:

- **Compute subsystem partition**: general purpose multicore ARMv8 CPU, plus a set of specialised units such

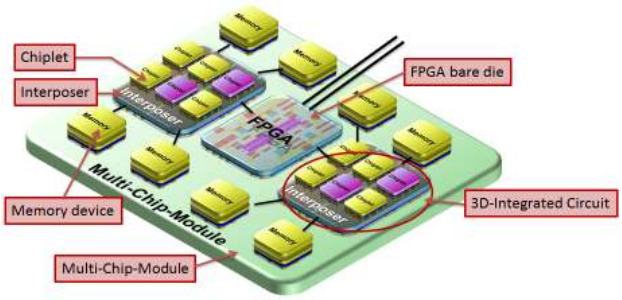

Fig. 2: ExaNoDe integration concept

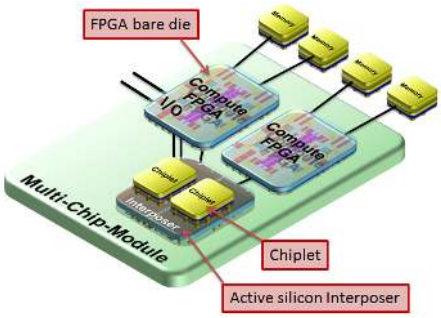

Fig. 3: ExaNoDe Multi-Chip-Module prototype

as graphics processing unit (GPU) and generic heterogeneous accelerators.

- **Memory subsystem partition**: DRAM and non-volatile memory (NVM) modules.
- **Interconnect and I/O partition**: traffic routing and scheduling, accesses to remote memory and storage, legacy I/O interfaces (e.g., PCI-e).

Fig. 3 depicts the ExaNoDe prototype compute element MCM. The FPGA components in the figure are used to implement the I/O interfaces and the actual Compute subsystems. ExaNoDe plans to use off-the-shelf ARMv8 based SoCs integrating on the same die an FPGA and a multi-core ARMv8 CPU, since the design of the compute Chiplet is out of the scope of the project. The selected target for the project is the Zynq UltraScale+ MPSoC [18], embedding a quad ARM Cortex-A53 processor, a Mali-400 MP2 GPU and an FPGA. Both the ARMv8 CPU and the FPGA will be used as compute units, in order to achieve the requested level of performance. The Cortex-A53 CPUs are not to be considered as a performance target, but rather an obligated choice since part of the Xilinx Zynq UltraScale+ MPSoC. In the future ARM CPUs will be equipped with the Scalable Vector Extensions (SVE) [19] that are a clear sign of ARM CPUs going in the direction of HPC computing. We expect production level exascale machines to adopt such enhanced ARM CPUs. The Xilinx SoC has been selected mainly for two reasons: 1) the high level of integration between heterogeneous compute units (i.e., CPU, GPU, FPGA), that goes in the direction of a higher compute density; and 2) the flexibility they provide for integration in a prototype demonstrator (i.e., FPGAs used not only as compute units but also to host glue logic between components). Multiple chiplets will lie on the Silicon Interposer which provides Chiplet-to-Chiplet, Chiplet-to-IO and Chiplet-to-Memory communication.

ExaNoDe will address Processor-to-Memory and Processor-to-Processor bandwidth versus energy efficiency by implementing new memory schemes in order to realise the benefits of the 3D integration technologies and determining the optimal trade-offs for an interposer-based 3D implementation. Even though the ExaNoDe partners have the right skills and expertise to deliver a compute element based on this technology, some other challenges will need to be tackled. There are a number of important aspects to note related to industrial applicability: dedicated design tools, manufacturing technologies, along with the necessary ecosystem of partners and vendors to fuel such drastic change in the traditional semiconductor manufacturing process. Within the ExaNoDe project those problems will be evaluated to propose industrially viable solutions for Interposer and Chiplet manufacturing and assembly.

B. Novel memory system

Message passing is to be considered as the de-facto standard for software communication in the HPC domain, mostly for scalability needs that make other approaches not viable. However message passing involves copies of data (between network interfaces and main memory), and the interaction between application and various other software layers before data can actually leave a node (e.g., message passing API, kernel modules, application buffers, etc.). There are various initiatives that attempt to hide such overheads, by defining APIs that mimic a shared memory abstraction such as the Partitioned Global Address Space (PGAS). Even in this case the shared memory abstraction is created by moving data between partitions, using techniques such as Remote Direct Memory Access (RDMA) since it is impossible for today's systems to use the fundamental efficiency of the processor's read/write instruction to manipulate global memory shared between processing nodes.

The ExaNoDe memory model instead will enable a real global address space (GAS), where data are actually shared between multiple partitions without the need to perform ad-hoc memory transfers. This removes the dependency from any software API to build the shared memory abstraction. Such a global shared address space leverages the capabilities of the modern ARMv8-A processor to create a global shared memory up to 48 bits in size. In addition, each SMP node within the larger GAS can have its own private 48-bit local memory partition, in which it can simply map pages from the shared GAS into its local partition and make this visible to applications through their specific virtual address space.

Fig. 4 shows an example of the ExaNoDe memory scheme, with two compute nodes each of them with their own local memory partition divided between DRAM, peripherals and an unused part. If an application in one of the two nodes wants

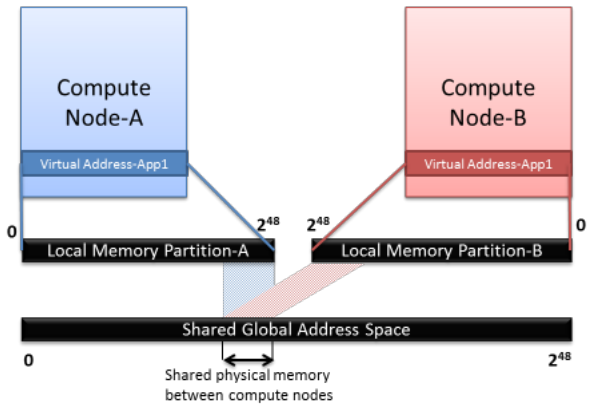

Fig. 4: ExaNoDe memory scheme

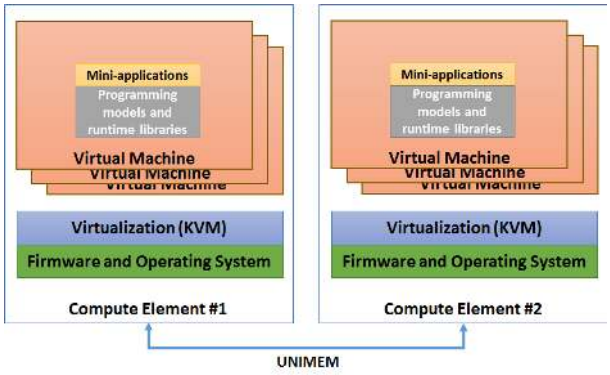

Fig. 5: ExaNoDe software architecture

to make part of the local memory visible to other nodes, it will have to request the system to map some pages from the global address space into its own local partition. The other node will then perform a similar procedure, mapping one of its local pages to a page in the global address space. From now on both nodes are able to write to the same physical memory page. This concept is implemented by the UNIMEM memory system, developed in the context of the FP7 EUROSERVER project [20].

UNIMEM is an advanced form of Global Address Space (GAS) architecture. It provides both RDMA based and zero-copy communication (message passing), essential to save energy by eliminating the redundant data copies between NIC and main memory, between kernel and user space, and/or between the receive buffer and application memory. In addition, it provides a coherent partitioned global address space model, without requiring a global cache coherence mechanism, by specifying that each page in the GAS is only cacheable in a single coherence island (which may or may not be the same as the island on whose DRAM the page is currently mapped). Since there is no need for a global cache coherence hardware mechanism, this model is scalable in the same way that the message passing model is. A UNIMEM enabled platform needs support both by hardware and software. The routing in the UNIMEM global address space will be implemented in the FPGAs and the Chiplets (see previous section), while the software support will be integrated in the firmware and operating system of the node.

C. Software infrastructure

As already anticipated, besides the hardware prototype design and implementation, the ExaNoDe project partners will devote significant effort on software research and development as medium used for testing the compute element with real-world applications, to extend programming models and runtimes towards Exascale and to support the unique characteristics of the ExaNoDe platform, including UNIMEM. Fig. 5 shows the four layers of the software architecture developed to run on each compute element, spanning the bare metal/OS level up-to the programming and application level.

Firmware and Operating system: ExaNoDe firmware will enable UNIMEM data movement mechanisms and integration with peripheral devices, leveraging the memory hierarchy to provide to the OS an interface for light RDMA operations. Within the node, the firmware will accelerate interactions between different modules, bypassing heavy legacy protocols and implementing more direct interfaces to NVM and accelerators. The firmware will also provide support for out-of-node communication by implementing shared (virtualized) DMA engine(s) and Ethernet NIC(s), as well as switching and routing for the next-level interconnect. The ExaNoDe operating system will be based on NUMA-aware Linux that supports dynamic distributed memory management and user-level socket-based communication across servers using RDMA to offer existing TCP/IP communicating applications the benefits of UNIMEM system architecture.

Virtualization: One-to-one correspondence between applications and hardware can no longer be applied to HPC systems, especially if looking towards Exascale. This is mainly due to resilience needs, and to management of a computing system that has to adapt to the changing conditions and needs of applications. Increased needs for resilience and improved manageability make virtualization the turnkey solution to achieve at the same time maximum performance and high flexibility. However, virtualization of compute accelerators and efficient check-pointing of virtual machines are still active research topics. In ExaNoDe, virtualization will be at the basis of each compute element with a resident KVM hypervisor [21], and applications wrapped within virtual machines. KVM has been chosen as target hypervisor because of its ease of use and high availability, being it part of any upstream Linux kernel. This feature enables the application programmer to test applications/virtual-machines on his own programming environment (i.e., regular workstation), and later flawlessly migrate to the HPC cluster.

The main improvements beyond the state-of-the-art will be the design and implementation of a complete solution for virtual machines check-pointing that minimizes the impact

on the running system (i.e., virtual machine stall time), so as to enable a finer grain control of the state of running applications. In addition, virtualization of accelerators will also be implemented, to enable multiple virtual machines on the same node to offload computation to the hardware accelerators available. The main acceleration hardware chosen in ExaNoDe are the FPGAs embedded in the Xilinx UltraScale+ MPSoC, programmed using the runtime system developed within the H2020 ECOSCALE project. According to the availability of an IOMMU on the target prototype, both full-hardware and software based virtualization techniques will be explored.

Parallel Programming Models and Runtime Libraries: Parallel programming models are essential to improve the programmability of parallel platforms, presenting programmers with a more uniform, simplified view of the system. Runtime systems dynamically manage the interaction between applications and the execution platform, enabling applications to portably utilize parallel platforms. Runtime optimizations can significantly enhance data locality [22], improving performance and reducing the energy used in data transfers. For each new hardware architecture, programming model runtime systems need to be customized – communication primitives need to be mapped on the data movement interface provided by the hardware, synchronization primitives need to be optimized and parallelization primitives must be tuned according to the primitive functionalities of the new machine. The following programming models are considered in the ExaNoDe project: MPI [23], GPI [24], OmpSs [25] and OpenStream [26]. OmpSs and OpenStream are extensions of OpenMP with new directives for offloading tasks. OmpSs uses directionality clauses on tasks and address-based tracking of data dependencies in the runtime system, and it supports heterogeneous devices such as GPUs and FPGAs. OpenStream has explicit dependencies in the source program marked using streams. Together, OmpSs and OpenStream explore two different trade-offs relating to performance and overheads vs. ease of programming. GPI is an open-source communication library that implements the GASPI PGAS API. It provides a portable and lightweight API that leverages remote completion and one-sided RDMA-driven communication, both being efficiently supported by the UNIMEM architecture. As such, GPI is an appropriate communication library to benefit from and evaluate the UNIMEM architecture. In addition, a power and thermal management runtime for dynamic adaptation of the operating conditions according to application demand and silicon thermal evolution, will be developed for the target prototype.

Mini-applications: Mini-applications are small self-contained proxies for large-scale production HPC applications. They capture most important performance features of HPC applications. Due to their simplicity they can be used for performance estimation and to stress specific functionalities of the underlying hardware (e.g., memory bandwidth, I/O bandwidth, computing performance, etc.). These applications will be used throughout the project to clarify system design trade-offs, and to evaluate the ExaNoDe Prototype with respect to usability and performance. The mini-applications will be extracted from large-scale production HPC applications in the following domains: High energy physics, Material sciences and Life sciences. Since the mini-applications will be used to stress the ExaNoDe architecture as if it were part of a large Exascale HPC system, their selection will be based on the performance characteristics and requirements that Exascale HPC systems should support.

IV. Expected results

ExaNoDe touches many research areas where promising technological results are expected. These all together will contribute to solve some limitations of the current available technologies related to HPC, advancing the state of the art in some specific domains and research topics that find applications also outside the HPC domain. The major results expected by the project will be now listed.

From the integration point of view, ExaNoDe is acting at both physical device level and software programmatic level. In this context, FPGA has been chosen not only for its widely-explored acceleration possibilities, but also as a mechanism that can enable the integration of the UNIMEM technology introduced in Subsection III-B. ExaNoDe will focus its effort in innovating new technologies regarding the integration of FPGAs, providing as well the reference software API to fully exploits the tight integration between the FPGA and the CPU. The power efficiency of the compute node will be achieved by an increased system's compute density, for which nano-technologies will be assessed and further developed by ExaNoDe. Specifically, the project has identified enhancements for a network on chip (NoC) technology which has to be properly extended to use the UNIMEM memory system architecture; in addition, still related to nano-technologies enhancements, potential improvements have been identified in the transceiver used above the silicon pad to further reduce the energy consumption of high bandwidth communication. The research results related to this topic will be brought to an operation prototype level that go beyond the 2.5D technology, which uses leading organic substrate for its device-level integration. The project has concrete objectives related to the prototype of the compute element where the 2.5D results will be deployed. This will be realized as a printed-circuit board (PCB) which is designed to be aligned and inter-operable with the PCB compute element being developed within the associated HPC ExaNeSt project, which in turn is inter-operable with the new form factor being used commercially by some industrial players.

Regarding the software innovations, as introduced in Subsection III-C, virtualization extensions play a key role to drastically improve resiliency and manageability of the computing system. Open source solutions for virtualization, such as KVM, offer already a solid infrastructure, very extensible and based on available open source code, which however

has not been designed specifically for HPC use-cases. In this context, the project aims at enhancing open source solutions to improve snapshot and check-pointing support, reducing at a minimum the down-time of the virtual machine and, as a consequence, improving the overall efficiency of the system. The virtualization solution will be extended accordingly to reduce the gap between the virtual machines and the underlying hardware as, for instance, the FPGA. This will require not only to extend device-assignment solutions, but also to design and implement the needed infrastructure to expose the ExaNoDe hardware to the guest systems.

All the novelty regarding hardware components and software infrastructure is coupled with programming models (GPI-2, MPI, OmpSs and OpenStream) that will be extended to use UNIMEM. In this context, ExaNoDe will ease the integration of the UNIMEM concept to existing HPC applications by extending the programming models mentioned in the previous Section. This expected result will have a huge impact on how the ExaNoDe concepts will be received by the HPC software developers: providing a variety of programming models to test and exploit the ExaNoDe technology will ease the spread and dissemination of the ExaNoDe results. More specifically, the MPICH [27] model which has been chosen as target implementation of the MPI standard, will be extended with a network module plugin that extends the base runtime with UNIMEM based communication optimizations, allowing unmodified HPC application to exploit this new communication back-end. OpenStream will be also target of the ExaNoDe research activities as it will be enhanced to operate over both the traditional SMP shared memory model, as well as the UNIMEM memory-scalability model adopted by this project. This programming environment, together with OmpSs, will be extended to leverage the UNIMEM architecture, with specific optimizations in the compiler (OpenStream) and runtime system (OmpSs and OpenStream). UNIMEM, on the other hand, introduced some challenges in regard to remote atomic operations over the global address space that the ExaNoDe partners were asked to solve. Firstly, an *atomic service* has been envisioned to handle atomic requests that target the global address space. Furthermore, an additional challenge came from the way the interconnection works, since the atomic operations issued by different CPUs might not reach the destination node in the correct order. The consortium expects to resolve this challenge as well, developing specific IPs for the final prototype.

Besides the ExaNoDe commitment to lower the power consumption drawn by the cores thanks to the improved compute density, additional results are expected oriented towards a self-adaptive management of the CPU power consumption. This will be achieved by designing and deploying a power and thermal management runtime that harmonize the power supply among the cores according to the software requirements. This component will be developed as an open source library that will be employed by system integrators as well as super computing centres and datacentres to respect the power consumption boundaries while ensuring the higher priority cores to still benefit from higher frequencies.

V. Conclusions

This paper presented the H2020 ExaNoDe project, a European funded project aimed at paving the way towards the definition of a compute element that supports the UNIMEM system architecture for an Exascale level computing system. In ExaNoDe the main focus is on the compute element, as the heart of a computing system. Thanks to 3D integration and Silicon Interposer technologies the project will create a disruption in how to design highly modular, yet power efficient computing elements. Such novelty will not only make it possible to reach the energy efficiency requested to make an Exascale system affordable, but will also open the door to actual industrial exploitation thanks to reduced time to market and easier integration of hardware components. ExaNoDe will build a compute element prototype based on ARMv8 processors, FPGAs to accelerate software and a memory hierarchy and locality architecture integrated using the aforementioned 3D technologies. In addition, the prototype will be complemented by a full software stack including: firmware and operating system support, virtualization for resilience and sharing of resources, and parallel programming models to actually span computation over the available computing resources.

Acknowledgement

This work was supported by the *ExaNoDe* project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 671578. The work presented in this paper reflects only authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

References

[1] "TOP500 Supercomputers List," https://www.top500.org/, last access: 2017-03-02.

[2] "Europe towards exascale," https://ec.europa.eu/digital-single-market/en/news/europe-towards-Exascale, last access: 2017-06-29.

[3] "Exascale Computing Project (ECP) Update," http://science.energy.gov/~/media/ascr/ascac/pdf/meetings/201604/ECP_ASCAC_Overview_final.pdf.

[4] "China's Exascale supercomputer operational by 2020," http://english.gov.cn/news/top_news/2016/06/16/content_281475373200996.htm, last access: 2017-03-02.

[5] "RIKEN selected to develop Exascale supercomputer," http://www.riken.jp/en/pr/topics/2013/20131226_1/, last access: 2017-03-02.

[6] "U.S. Department of Energy: The Opportunities and Challenges of Exascale Computing," http://science.energy.gov/~/media/ascr/ascac/pdf/reports/Exascale_subcommittee_report.pdf, last access: 2017-03-02.

[7] "H2020 ExaNoDe web-site," http://exanode.eu.

[8] M. Katevenis, N. Chrysos, M. Marazakis, I. Mavroidis, F. Chaix, N. Kallimanis, J. Navaridas, J. Goodacre, P. Vicini, A. Biagioni, P. S. Paolucci, A. Lonardo, E. Pastorelli, F. L. Cicero, R. Ammendola, P. Hopton, P. Coates, G. Taffoni, S. Cozzini, M. Kersten, Y. Zhang, J. Sahuquillo, S. Lechago, C. Pinto, B. Lietzow, D. Everett, and G. Perna, "The exanest project: Interconnects, storage, and packaging for exascale systems," in *2016 Euromicro Conference on Digital System Design (DSD)*, Aug 2016, pp. 60–67.

[9] I. Mavroidis, I. Papaefstathiou, L. Lavagno, D. S. Nikolopoulos, D. Koch, J. Goodacre, I. Sourdis, V. Papaefstathiou, M. Coppola, and M. Palomino, "Ecoscale: Reconfigurable computing and runtime system for future exascale systems," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016*. IEEE, 2016, pp. 696–701.

[10] "ETP4HPC Strategic Research Agenda," http://www.etp4hpc.eu/en/sra.html, last access: 2017-06-29.

[11] U.S. Department of Energy, "Top Ten Exascale Research Challenges," http://science.energy.gov/~/media/ascr/ascac/pdf/meetings/20140210/Top10reportFEB14.pdf, last access: 2017-03-02.

[12] "TOP500 - TOP 10 Sites for November 2016," https://www.top500.org/lists/2016/11/, last access: 2017-03-02.

[13] "High Performance Conjugate Gradients (HPCG) Benchmark project," http://www.hpcg-benchmark.org/, last access: 2017-03-06.

[14] J. Shalf, S. Dosanjh, and J. Morrison, "Exascale computing technology challenges," in *International Conference on High Performance Computing for Computational Science*. Springer, 2010, pp. 1–25.

[15] Kirk Saban, "Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency," https://www.xilinx.com/support/documentation/white_papers/wp380_Stacked_Silicon_Interconnect_Technology.pdf, last access: 2017-03-02.

[16] P. Vivet, C. Bernard, F. Clermidy, D. Dutoit, E. Guthmuller, I.-M. Panadès, G. Pillonnet, Y. Thonnart, A. Garnier, D. Lattard *et al.*, "3D advanced integration technology for heterogeneous systems," in *3D Systems Integration Conference (3DIC), 2015 International*. IEEE, 2015, pp. FS6–1.

[17] P. Vivet, Y. Thonnart, R. Lemaire, C. Santos, E. Beign, C. Bernard, F. Darve, D. Lattard, I. Miro-Panads, D. Dutoit, F. Clermidy, S. Cheramy, A. Sheibanyrad, F. Ptrot, E. Flamand, J. Michailos, A. Arriordaz, L. Wang, and J. Schloeffel, "A 4 x 4 x 2 Homogeneous Scalable 3D Network-on-Chip Circuit With 326 MFlit/s 0.66 pJ/b Robust and Fault Tolerant Asynchronous 3D Links," *IEEE Journal of Solid-State Circuits*, vol. PP, no. 99, pp. 1–17, 2016.

[18] "Unleash the Unparalleled Power and Flexibility of Zynq UltraScale+ MPSoCs," https://www.xilinx.com/support/documentation/white_papers/wp470-ultrascale-plus-power-flexibility.pdf, last access: 2017-06-29.

[19] N. Stephens, "ARMv8-A Next-Generation Vector Architecture for HPC," https://community.arm.com/cfs-file/__key/telligent-evolution-components-attachments/01-2142-00-00-00-01-20-49/ARMv8_2D00_A-SVE-technology-Hot-Chips-v12.pdf, last access: 2017-03-02.

[20] Y. Durand, P. M. Carpenter, S. Adami, A. Bilas, D. Dutoit, A. Farcy, G. Gaydadjiev, J. Goodacre, M. Katevenis, M. Marazakis *et al.*, "EUROSERVER: Energy efficient node for European micro-servers," in *Digital System Design (DSD), 2014 17th Euromicro Conference on*. IEEE, 2014, pp. 206–213.

[21] C. Dall and J. Nieh, "KVM/ARM: the design and implementation of the linux ARM hypervisor," in *ACM SIGPLAN Notices*, vol. 49, no. 4. ACM, 2014, pp. 333–348.

[22] A. Drebes, A. Pop, K. Heydemann, A. Cohen, and N. Drach, "Scalable Task Parallelism for NUMA: A Uniform Abstraction for Coordinated Scheduling and Memory Management," in *Proceedings of the 2016 International Conference on Parallel Architectures and Compilation*. ACM, 2016, pp. 125–137.

[23] M. P. Forum, "MPI: A Message-Passing Interface Standard," Knoxville, TN, USA, Tech. Rep., 1994.

[24] Fraunhofer Institut für Technound Wirtschaftsmathematik ITWM, "The building blocks for HPC: GPI and MCTP," http://www.gpi-site.com/cms/sites/default/files/GPI_Whitepaper.pdf, last access: 2017-06-29.

[25] A. Duran, E. Ayguadé, R. M. Badia, J. Labarta, L. Martinell, X. Martorell, and J. Planas, "Ompss: a proposal for programming heterogeneous multi-core architectures," *Parallel Processing Letters*, vol. 21, no. 02, pp. 173–193, 2011.

[26] A. Pop and A. Cohen, "OpenStream: Expressiveness and data-flow compilation of OpenMP streaming programs," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 9, no. 4, p. 53, 2013.

[27] "High performance and widely portable implementation of the message passing interface (mpi)," https://www.mpich.org, last access: 2017-04-10.