TEXTAROSSA: Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale

Giovanni Agosta, Daniele Cattaneo, William Fornaciari, Andrea Galimberti, Giuseppe Massari, Federico Reghenzani, Federico Terraneo, Davide Zoni, Carlo Brandolese, Massimo Celino, et al.

TEXTAROSSA: Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale

Giovanni Agosta, Daniele Cattaneo, William Fornaciari, Andrea Galimberti, Giuseppe Massari, Federico Reghenzani, Federico Terraneo, Davide Zoni, Carlo Brandolese
DEIB – Politecnico di Milano, Italy, name.surname@polimi.it

Massimo Celino, Francesco Iannone, Paolo Palazzari, Giuseppe Zummo
ENEA, Italy, name.surname@enea.it

Massimo Bernaschi, Pasqua D'Ambra
Istituto per le Applicazioni del Calcolo (IAC) - CNR, Italy, name.surname@cnr.it

Sergio Saponara, Marco Danelutto, Massimo Torquati
University of Pisa, Italy, name.surname@unipi.it

Marco Aldinucci, Yasir Arfat, Barbara Cantalupo, Iacopo Colonnelli, Roberto Esposito, Alberto R. Martinelli, Gianluca Mittone
University of Torino, Italy, name.surname@unito.it

Olivier Beaumont, Berenger Bramas, Lionel Eyraud-Dubois, Brice Goglin, Abdou Guermouche, Raymond Namyst, Samuel Thibault
Inria - France, name.surname@inria.fr

Antonio Filgueras, Miquel Vidal, Carlos Alvarez, Xavier Martorell
BSC - Spain, name.surname@bsc.es

Ariel Oleksiak, Michal Kulczewski
PSNC, Poland, ariel@man.poznan.pl, kulka@man.poznan.pl

Alessandro Lonardo, Piero Vicini, Francesca Lo Cicero, Francesco Simula, Andrea Biagioni, Paolo Cretaro, Ottorino Frezza, Pier Stanislao Paolucci, Matteo Turisini
INFN Sezione di Roma - Italy, name.surname@roma1.infn.it

Francesco Giacomini
INFN CNAF - Italy, name.surname@cnaf.infn.it

Tommaso Boccali
INFN Sezione di Pisa - Italy, name.surname@pi.infn.it

Simone Montangero
University of Padova and INFN Sezione di Padova - Italy, name.surname@pd.infn.it

Roberto Ammendola
INFN Sezione di Roma Tor Vergata - Italy, name.surname@roma2.infn.it

***Abstract*—To achieve high performance and high energy efficiency on near-future exascale computing systems, three key technology gaps needs to be bridged. These gaps include: energy efficiency and thermal control; extreme computation efficiency via HW acceleration and new arithmetics; methods and tools for seamless integration of reconfigurable accelerators in heterogeneous HPC multi-node platforms. TEXTAROSSA aims at tackling this gap through a co-design approach to heterogeneous HPC solutions, supported by the integration and extension of HW and SW IPs, programming models and tools derived from European research.**

1. Introduction and long-term objectives

High Performance Computing (HPC) is a vital infrastructure for both industry and social actors in any country. In addition to traditional domains, such as oil & gas, finance, or weather forecasting, HPC is increasingly important for emerging domains such as bioinformatics, medicine, security and surveillance, which fall in the spectrum of High Performance Data Analytics (HPDA) and High Per-

978-1-6654-2703-6/21/$31.00 ©2021 IEEE
DOI 10.1109/DSD53832.2021.00051

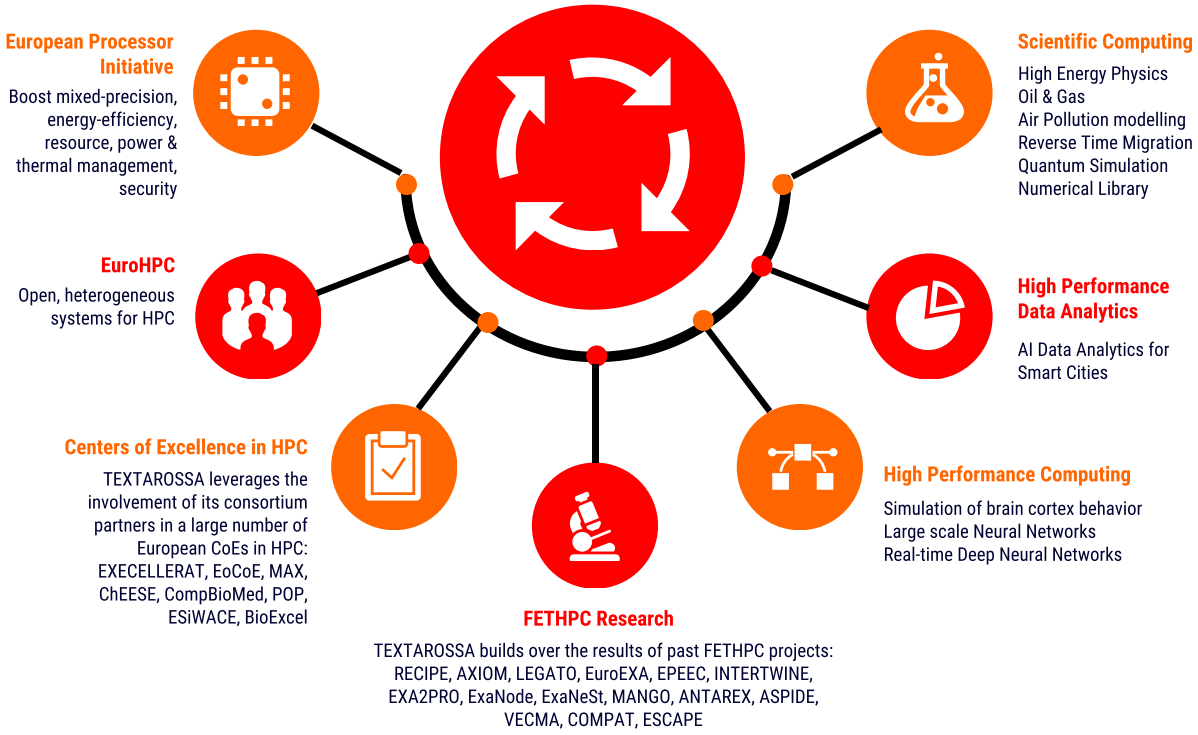

Figure 1. The TEXTAROSSA Impact Strategy within the broader framework of European initiatives on HPC

formance Computing for Artificial Intelligence (HPC-AI). The emergence of heterogeneous HW architectures and the trends towards "Green HPC" have prompted Europe to align its research priorities in HPC along a Strategic Research Agenda (SRA[1]) resulting from wide consultations within the European Technology Platform for HPC (ETP4HPC), the PRACE initiative[2], and the PlanetHPC[3] initiative.

The SRA highlights significant technology challenges due to the need to achieve high efficiency while remaining within reasonable power and energy bounds. This challenging goal can only be addressed with an holistic approach that takes into account multiple factors across the HPC hardware/software stack, including the use of application-specific, extremely efficient hardware accelerators, efficient software management of resources, data and applications, and efficient cooling systems. Together, these components can provide the desired computational power while keeping under control the power consumption of the supercomputer.

To this end, the TEXTAROSSA project aims at providing key technological advances on all three aspects and validate them on new development platforms representative of future HPC systems, using a wide range of applications from different domains, both within traditional HPC and coming from emerging domains. In particular, TEXTAROSSA aims implementing the above-mentioned approach through the following technical goals.

1. https://www.etp4hpc.eu/sra.html (last accessed July 2021)
2. https://https://prace-ri.eu (last accessed July 2021)
3. https://cordis.europa.eu/project/id/248749 (last accessed July 2021)

Technical goals. 1) *energy efficiency and thermal control* via innovative two-phase cooling technology at node and rack level, fully integrated in an optimized multi-level runtime resource management driven by power, energy, and thermal models fed by on-board sensor data; 2) *sustained application performance* through efficient exploitation of highly concurrent accelerators (GPUs and FPGAs) by focusing on data/stream locality, efficient algorithms and programming models, tuned libraries and innovative IPs; 3) *seamless integration of reconfigurable accelerators* by extending field-proven tools for the design and implementation such as Vitis and OmpSs@FPGA to support new IPs and methodologies such as mixed-precision computing and power monitoring and control. 4) *development of new IPs* for mixed-precision AI computing, data compression, security, power monitoring and control, and scheduling. 5) *Integrated Development Platforms* by developing two architecturally different, heterogeneous Integrated Development Vehicles (IDVs), one as a dedicated testbed for two-phase cooling technology, and one supporting the wider range of project technical goals.

Strategic goals. Figure 1 highlights the strategic goals of TEXTAROSSA, set against the EuroHPC initiative and the broader framework of European research in High Performance Computing. In particular, we highlight the following goals: 1) *alignment with the European Processor Initiative (EPI)* by testing, extending and boosting key technologies applicable to future EPI evolutions; 2) *supporting the objectives of EuroHPC* as reported in ETP4HPC's

Strategic Research Agenda (SRA) for open HW and SW architecture. 3) *building over European expertise* gained through past research projects as well as through the Centers of Excellence in HPC. 4) *opening of new usage domains*, including High Performance Data Analytics (HPDA) and High Performance Artificial Intelligence (HPC-AI) applications, alongside support for traditional HPC domains.

1.1. TEXTAROSSA Consortium

To achieve the above-mentioned goals, TEXTAROSSA, a three-year project co-funded by the European High Performance Computing (EuroHPC) Joint Undertaking, is led by ENEA (Italy) and aggregates 17 institutions and companies, including the linked third parties, located in 5 European countries: CINI, an Italian consortium grouping together three leading universities, Politecnico di Milano, Università degli studi di Torino, and Università di Pisa, Fraunhofer (Germany), INRIA (France), ATOS (France), E4 Computer Engineering (Italy), BSC (Spain), PSNC (Poland), INFN (Italy), CNR (Italy), In Quattro (Italy), Université de Bordeaux (France), CINECA (Italy) and Universitat Politècnica de Catalunya (UPC). The three Italian universities are part of the lab HPC: hpc-key-technologies-and-tools of CINI, created in 2021, that is grouping together the main academic and research entities working in the field of high-performance and exascale computing in Italy. More information on the activities carried out during the execution of TEXTAROSSA can be found in the project website[4].

1.2. Organization of the paper

The rest of this paper is organized as follows. In Section 2, we introduce the TEXTAROSSA co-design approach. In Section 3 we describe the key technological innovations provided by the TEXTAROSSA project, while in Section 4 we provide an overview of the application use cases. In Section 5 we draw some conclusions and highlight future research directions.

2. TEXTAROSSA Co-design approach

From a methodology point of view TEXTAROSSA adopts a co-design process as key strategy for Fast Forward and Exascale computing, considering the entire system stack from underlying technologies to applications. The co-design process concerns five layers covering the whole HPC stack: 1) *User Application*: representing a wide range of scenarios, from mathematical libraries, to miniApps and flagship codes for numerical modelling with massive parallelism in HPC/HPDA/AI applications. 2) *Runtime Services*: ensuring that application requirements are dynamically satisfied and mapped onto system resources, and including execution models with workload handling, fault tolerance and data management. 3) *Programming Models*: underlying the applications, they define the toolchains and SW development

4. https://textarossa.eu (last accessed July 2021)

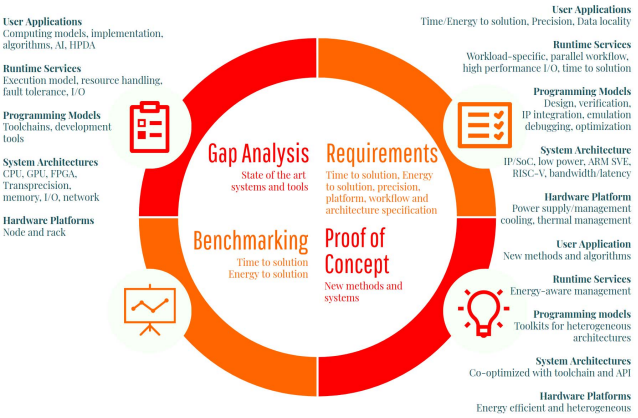

Figure 2. The TEXTAROSSA Co-Design Approach

tools able to implement applications in parallel architectures. 4) *System Architecture*: including the processor core's micro-architecture, the arrangement of cores within a chip, memory hierarchy, system interconnect, and storage subsystems. 5) *HW Platforms*: concerning the HW platform at node and rack level able to achieve performance requirements in terms of computing power and energy consumption.

Figure 2 provides an overview of the co-design approach adopted in TEXTAROSSA, showing how the five layers of the HPC stack are addressed in each of the four main stages of the co-design process: 1) *Gap Analysis*: to compare the current state-of-art of the technological assets with the objectives of the project in order to identify the gap to be filled by developments or update in co-design process. 2) *Requirements*: to define specifications and requirements of the technological solutions for designing and developing. 3) *Proof of Concept*: to develop HW/SW prototype solutions able to achieve the KPIs (Key Performance Indicatoirs) of the project objectives. 4) *Benchmarking*: to provide performance results of the technological solutions by means of benchmark tools.

3. TEXTAROSSA technologies

TEXTAROSSA develops, starting from the results of previous European research activities mentioned in Figure 1, a set of technologies to deal with each of the four technology layers of the HPC stack, as well as applications, which will be covered in full in Section 4. Figure 3 provides an overview of the primary technology bricks adopted by TEXTAROSSA. In the rest of this section, we provide insights on specific technology bricks developed within the project

3.1. Programming Models & Toolchains

Vitis based HLS flow. Vitis [1] is the HLS flow designed by Xilinx to cover all the steps required to translate an application, described through a C/C++ program, into a

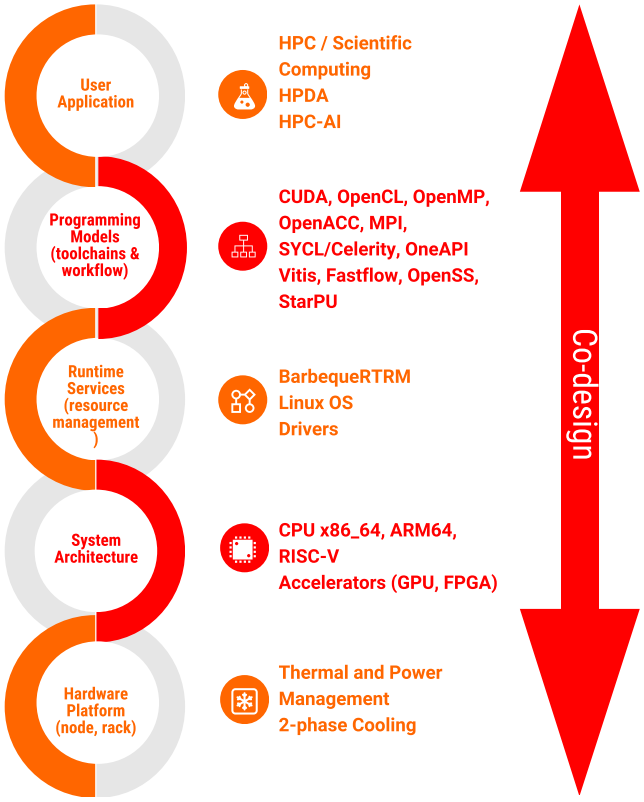

Figure 3. The TEXTAROSSA key technologies

working bitstream running on an FPGA card and communicating with a program running on the host node. The Vitis environment allows emulating the whole system's behavior (host program and FPGA code) by running and debugging the C/C++ code in a standard IDE. Once the functional correctness has been achieved, i.e. the program produces the expected results, the HLS engine translates the pieces of C/C++ code mapped on the FPGA into equivalent, optimized, hardware implementation. The communication with the external world (host node, memory banks) is achieved through a presynthesized layer implementing the interfaces (PCIe, DDR and HBM memory banks) and the FPGA resources are accessed from the host node through the runtime and APIs developed by Xilinx. Vitis is largely customizable thanks to the open-source LLVM-based front-end[5] and many open-source accelerated libraries (e.g., math, video processing, AI, signal processing)[6]. In the TEXTAROSSA project, the Vitis environment will be extended with multi-precision arithmetic, allowing the usage of new data types based on the Posit format, and with a communication library used to perform inter-FPGA direct communications. The inter-node communication layer, namely MPI, will also be updated to transfer data without requiring memory copies between FPGAs and hosts. Furthermore, the APIs to access the FPGA will be used to build the TEXTAROSSA APIs that will be defined to access homogeneously the accelerators. Such extensions will require the integration into the Vitis flow of the new IPs defined in the project, thus we will develop the hardware modules, their C++ functional models and we will encode all the necessary info needed by the HLS to properly manage (i.e. schedule and connect) the new functionalities.

StarPU, OmpSs. Task-based programming models (like OmpSs[7] [2] and StarPU[8] [3]) address the challenges to program on heterogeneous computing nodes while achieving high productivity by providing higher-level abstractions that could help the programmer to generate high-performance code. For example: making the memory allocation and data copies automatic; providing the programmer with facilities to perform blocking from inside the accelerators; automating the code generation of the CPU and FPGA binaries, provided the C/C++ implementation, by transparently running open or vendor tools; allowing the use of parallelism based on tasking (instead of kernel invocations); providing support for data-dependent tasks, and managing the execution based on such data dependencies and/or providing FPGA execution trace generation support; choosing the computation resource that provides the best execution compromise for a given task and situation [4]. This makes the programming environment completely hide the target architectures, providing a clean, high-level, abstract interface to the programmers, and incorporating all the intelligence on management and scheduling into the runtime system. We plan to move part of the runtime work to a fast hardware task scheduler [5] to further enhance the performance of these programming models.

Training Deep Neural Network (DNN) is a memory-intensive operation. Indeed, the training algorithms of most DNNs require to store both the model weights and the forward activations in order to perform back-propagation. In practice, training is performed automatically and transparently to the user through autograd tools for back-propagation. Unfortunately, the memory limitation of current HW often prevents data scientists from considering larger models, larger image sizes or larger batch sizes, especially in recent NLP models [6]. Our goal is to extend StarPU to enable inference and learning, taking advantage of both the heterogeneity of the architecture to place layers and the memory architecture to minimize transfers. In particular, we will rely on the RoToR framework [9] [7] which allows to control the memory consumption and to minimize the energy consumed by the data exchanges.

Several problems should be solved to use FPGAs more efficiently and control energy consumption. In the TEXTAROSSA project, we will extend these two task-based runtime systems, OmpSs and StarPU, which use different approaches to support FPGA. This will allow the integrated development vehicle to benefit from their existing features,

5. https://github.com/Xilinx/HLS (last accessed July 2021)
6. https://xilinx.github.io/Vitis_Libraries (last accessed July 2021)
7. https://github.com/bsc-pm-ompss-at-fpga (last accessed July 2021)
8. https://starpu.gitlabpages.inria.fr (last accessed July 2021)
9. https://gitlab.inria.fr/hiepacs/rotor (last accessed July 2021)

but also to study their complementarity while validating the robustness of the new HW against different runtime systems.

FastFlow. Stream processing is gaining increasing industrial attention for real-time data analytics and data-driven applications [8]. FastFlow [9] is a C++ programming library targeting multi/many-cores. It offers both a set of high-level ready-to-use parallel pattern implementations and a set of mechanisms and composable components (called building blocks) to support low-latency and high-throughput data-flow streaming networks. FastFlow simplifies the development of parallel applications modeled as a structured, directed graph of processing nodes. The graph of concurrent nodes is constructed by the assembly of sequential and parallel building blocks and higher-level components (i.e., parallel patterns) modeling recurrent schemas of parallel computations (e.g., pipeline, task-farm, parallel-for, etc.). FastFlow efficiency stems from the optimized implementation of the base communication and synchronization mechanisms and its layered software design. Besides, stream processing is the natural paradigm for event-driven distributed applications that need to communicate with each other via message passing. Finally, some data streaming paradigms are naturally suited for implementation on reconfigurable platforms [10], e.g. the dataflow/actor paradigm. In TEXTAROSSA, we aim at exploiting reconfigurable platforms to accelerate HPDA tasks leveraging the FastFlow framework [11].

Streamflow. The StreamFlow framework [12], [13] is a container-native Workflow Management System (WMS) written in Python 3 and based on the Common Workflow Language (CWL) Standard [14]. StreamFlow has been designed around two main principles: 1) Allowing the execution of tasks in multi-container environments, in order to support concurrent execution of multiple communicating tasks in a multi-agent ecosystem; 2) Relaxing the requirement of a single shared data space, in order to allow for hybrid workflow executions on top of multi-cloud or hybrid cloud/HPC infrastructures. StreamFlow source code is available on GitHub under the LGPLv3 license. A Python package is downloadable from PyPI and Docker containers can be found on Docker Hub. More details about the tool and its applications can be found in the StreamFlow website[10].

Compiler Technology for Mixed-Precision Support. Error-tolerating applications are increasingly common in HPC. Proposals have been made at the HW level to take advantage of inherent perceptual limitations, redundant data, or reduced precision input [15], as well as to reduce system costs or improve power efficiency [16]. At the same time, works on floating-point to fixed-point conversion tools [17], [18] allow us to trade-off the algorithm exactness for a more efficient implementation. Finally, new data types are emerging including BFloat16 [19] and Posit [20]. BFloat16 (Brain Floating Point) is used in upcoming Intel AI processors (NERVANA), XEON processors, Google Cloud TPU and ARMv8.6-A, as well as in RISC-V extensions [21]. Posit are a new compressed floating-point data format for which University of Pisa has developed a SW library called CppPosit [22], [23]. From the first results of applying the CppPosit library to AI/DNN problems, Posit can lead to the same processing accuracy of float but with a data compression from a factor 2 to 4 [22], [23]. This means that applying Posit to the application cases (HPC, HPDA and AI/CNN) has the potential to reduce data storage issues and allows for fast data movement. In TEXTAROSSA, we aim at exploiting and extending the tools for precision tuning developed as part of the H2020 FETHPC ANTAREX project [24] to cover a wider range of target platforms, targeting FPGAs through integration with the TEXTAROSSA High Level Synthesis (HLS) toolchain. These tools, collected in the TAFFO framework [25], [26] are implemented as a set of plugins for the LLVM compiler, and, based on programmer hints expressed as attributes, perform value range analysis, data type and code conversion, and static estimation of the performance impact. We aim at improving the performance estimation by exploiting recent analysis techniques [27] as well as deeper understanding of the target processor pipeline, by expanding the use of the tools to heterogeneous systems with reconfigurable components, and by considering emerging data types such as Posit and Bfloat16.

10. https://streamflow.di.unito.it (last accessed July 2021)

3.2. Runtime Services: Energy/Power Management

Power consumption represents one of the most remarkable cost items in the balance of the overall costs of an HPC center. With exascale computing platforms a linear increment of power consumption with computing power (due to the end of Dennard's scaling) becomes unsustainable. An energy-efficient exascale HPC infrastructure, much like current supercomputers, will rely on heterogeneous computing resources. Energy efficiency can then be guaranteed only if, on the SW side, we can rely on a suitable (hierarchical) resource management framework. Although the state-of-the-art already includes some solutions, recent projects, like MANGO [28] and RECIPE [29], show that optimized solutions need to take into account the platform-specific characteristics and control knobs to profile the applications at design-time and monitor them at runtime [30], [31], enabling more accurate resource mappings [32], [33]. Furthermore, an integration of the resource manager with the programming model allows dynamically tuning the numerical accuracy (precision) of the tasks, with respect to the actual application requirements and power/energy constraints. Given the reference HW platform and the application use cases, the TEXTAROSSA project would represent an extremely interesting testbed for exploring novel power and energy management solutions, at the HW but also at the SW level. At the HW level, a specific support will be introduced to automatically instrument the computing platform with ad-hoc power monitors [34] and controllers [30], [31], in order to reduce the response time of the power management dramatically, while increasing the effectiveness of the thermal management. On the software side, starting from an already existing resource management framework [35], we aim to extend it with the support for the new HW and, of course, new resource management policies, along with

the integration of the precision tuning tool. Overall, this would allow us to explore all the possibilities offered by the platform, and the application-side integration, to increase the FLOPS-per-Watt ratio, with respect to state- of-the-art solutions in HPC.

3.3. Posit Hardware Accelerators

Within EPI, the Stencil and Tensor Accelerator (STX), based on RISC-V core, enables energy-efficiency for applications where the main computational kernels work on discretized grids and perform a series of operations like convolutions. It implements several floating point units able to work on fp32, fp16, fp8 IEEE and bfloat16 floating-point formats. To increase performance, one could try to reduce the number of bits in the floating point representation. But when reducing the precision of the used arithmetic, iterating over many timesteps, the derivations may become unstable and hence affect the final result of the simulations. The novel Posit binary arithmetic format can offer higher precision while using less bits than standard IEEE floating-point numbers. Recent literature shows [22], [23] that 16bit Posit can leverage comparable results like fp32 and Posit with 8bit precision outperform in terms of accuracy fp16 (for CNN 8bit Posit can leverage comparable results like fp32). Calculations can be done even with simple bit manipulations on the Posit format without extraction, further decreasing the complexity of the operations. It is thus possible to enhance memory bandwidth, and lower level cache utilization, power footprint, and throughput of the arithmetic units.

Within TEXTAROSSA, a RISC-V unit will be extended with support for alternative data representations, including fine-grained reduced precision floating point [21] and Posit arithmetic. To complement the hardware IP developments, the LLVM compiler, also adopted in EPI, will be extended for Posit and data compression support and real-world HPC applications and CNN kernels will be ported to leverage the IP. Fast software co-design will be enabled through software implementation of Posit in the CppPosit library[11]. The novel IP will be ported to FPGAs for benchmarking. Both techniques have not been implemented on top of a completely co-designed accelerator so far and will provide a huge benefit for the European IP portfolio.

3.4. Hardware Platform Optimization

HPC systems have historically always been limited by thermal considerations and computing architecture needs both optimized heat dissipation solutions and run-time thermal control policies to operate reliably and efficiently. In-Quattro has developed an innovative thermal management solution (patent pending) based on two-phase mechanically pumped loops, which uses a flow boiling heat transfer for cooling electronics in a more efficient way. This would allow the use of the latent heat of vaporization so that flow rates would be significantly reduced, temperature gradients would be small, and heat transfer coefficients could be large compared to liquid cooling and air cooling systems. Two-phase cooling systems using evaporation and condensation are known to be the best way to meet demanding cooling requirements in terms of compactness, weight and energy-consumption. A possible configuration of the two-phase cooling system for a single node is shown in Figure 4.

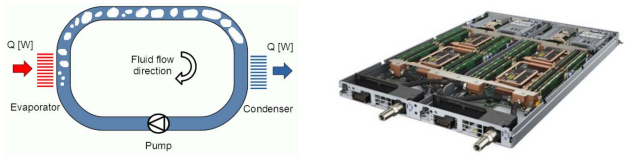

Figure 4. Schematic of the two-phase cooling system (left) and its embedding in a server (right).

The two-phase cooling system will be adapted to node(s) provided by E4 and ATOS and developed with the objective to serve an entire rack. In TEXTAROSSA, the innovative cooling system will be customized to fit the requirements of node and rack levels for exascale applications. It is foreseen to develop two solutions of the two-phase cooling system that could be patented during the project on the principal components of the cooling system (evaporator, condenser). This innovative technology is expected to improve the cooling efficiency up to 70% compared to traditional air cooling, and up to 30% compared to existing liquid cooling and will be tested on both ATOS and E4 infrastructures.

Both the design of cooling solutions and thermal control policies critically relies on thermal models, which in turn rely on experimental validation data. Thermal simulators for CPUs/MPSoCs have been proposed, but they can only represent a limited range of heat dissipation solutions, and are not easy to extend towards two-phase liquid cooling solutions or to encompass the simulation of an entire rack. In TEXTAROSSA, we will design and validate thermal models taking advantage of equation-based object-oriented modeling languages to increase the abstraction level of thermal models [36], thus overcoming the inflexibility of current simulators, and simplifying the design of nonlinear thermal models to account for two phase liquid cooling. For model validation, we will use a thermal test chip platform [37] to capture accurate thermal maps of chips connected to the proposed heat dissipation solution, considering spatial and temporal temperature variations and hot spots. The collected experimental data will be used to validate the thermal models and for the design of the policies. Through a multilevel thermal control strategy we aim to overcome the complexity of controlling an HPC platform from node to system level with minimal overhead. As the fastest temperature gradients occur at the silicon active layer, we will use fast event-based control loops [38] acting on DVFS to limit the maximum operating temperature of compute elements. These inner control loop will in turn interact with higher level control loops operating the two phase cooling infrastructure of the node, which is comparatively slower and has higher overheads but has the capability to increase

11. https://github.com/eruffaldi/cppPosit (last accessed July 2021)

the heat transfer coefficient on-demand, thus allowing to relieve the need to reduce frequency using DVFS, in turn improving performance. A further supervisory control layer will allow to set the desired temperatures at the rack level based on reliability metrics. Multilevel control allows thus to partition the system level control problem into multiple interacting control loops, each optimized for the specific thermal dynamics to control.

3.5. Low-latency Communication between FPGAs

The usage of FPGAs as accelerators is getting so widespread that even big cloud providers are now installing reconfigurable devices in their instances (e.g. on Microsoft Azure and Amazon EC2). Interaction of hundreds to thousands of FPGAs require a scalable approach to hold them together, allowing a low latency connection among them, but a definitive approach has to be found to let users make the most of their flexibility and in the meanwhile easing the usage for software developers. As an example, the latest version of the Microsoft Catapult fabric, puts a Stratix 10 device between each NIC on the X86 servers and the ToR switch, enabling a fast path for accelerators to communicate among themselves with a few microseconds latency. The Brainwave project [39] leverages this architecture to provide a deep learning platform for real-time AI inference on the cloud. While this framework offers a very friendly interface for users to deploy their models on top of this architecture, it loses the flexibility of delivering the cores as black boxes, and providing an implementation of only a few pre-trained models. Our approach, on the other hand, let users full control of the platform, allowing the implementation of custom processing tasks on FPGAs, still maintaining ease of usage by supplying a set of interfaces to integrate with the HLS tools developed in the project. Thus will allow developers to define a scalable application using a streaming programming model (Kahn Process Network [40]) that can be efficiently deployed on a multi-FPGAs system. TEXTAROSSA will develop a communication IP and its SW stack, providing the implementation of a direct network that allows low-latency communication between processing tasks deployed on FPGAs, even hosted in different computing nodes. The communication IP will be based on the ExaNet IPs (switch, router, high-speed channels [41]) developed in ExaNeSt H2020 project [42] and EuroEXA H2020 [43]. The direct communication between tasks deployed on FPGAs will avoid the involvement of the CPUs and system bus resources in the data transfers, improving the platform's energy efficiency and reducing communication latency.

4. TEXTAROSSA applications

In this section, we briefly provide an overview of the use case applications adopted in TEXTAROSSA. To address the variety of application domains of future Exascale systems, TEXTAROSSA applications include basic mathematical building blocks (*MathLib*), traditional HPC applications (*UrbanAir*, *TNM*, *HEP*, and *RTM*), and applications from emerging domains (*RAIDER*, *DPSNN*, *Danger Detection*).

4.1. MathLib

One of the basic guidelines in energy efficient computing is the optimization and the acceleration of algorithms and SW libraries that provide a reduction of the elapsed time of HPC applications and thus a significant cut in energy consumption. The new power-to-solution metrics requires a rethinking of many computational kernels of HPC applications looking for a trade-off between the reduction of the total energy and the minimization of the time-to-solution, promoting scalability. Within this context, extensions and improvements of high-performance algorithms and SW libraries for kernels in numerical linear algebra [44], [45] and graph computation, such as iterative [46], [47], [48], [49] and direct linear solvers, edge weighted graph matching, and fast multipole methods [50] will be deployed.

4.2. HPDA and HPC-AI Applications

Real-time AI-based Data analytics on hetERogeneous distributed systems (RAIDER). Leveraging on the experience gained on the design of the GPU-RICH system for the NA62 experiment at CERN [51], a proof-of concept of a real-time AI-based data analytics on heterogeneous distributed systems shall be designed. The application setup deploys a set of data streams from sensors as input to a Deep Neural Network, implemented over a pool of heterogeneous processing layers, that is in charge of performing the data analysis. Data from different streams are recombined through the processing layers through the network infrastructure.

Brain Simulation (DPSNN). The Distributed and Plastic Spiking Neural Network (DPSNN) application was developed by INFN to model brain cortex behaviour [52], [53] and more recently to study sleep-related learning activities [54]. It is a scalable neural network simulation C++/MPI code for HPC platforms at extreme scales. It simulates the spiking dynamics of the brain cortex by slicing it into a grid of cortical columns populated with neurons and their interconnecting synapses.

Smart Cities (Danger Detection). In the context of smart cities the distributed video surveillance systems provide a huge amount of data for processing with AI techniques. A design activity shall be carried out for a real-time danger alarm system (e.g. smoke/fire detection in smart transportation or smart cities context). The system is composed of a network of smart cameras where an AI algorithm is implemented on an EDGE server implementing a preprocessing stage plus a CNN.

4.3. Traditional HPC Applications

Air Pollution (UrbanAir). The UrbanAir concerns the modelling and forecasting of the concentration and

dispersion of pollutants. It is a 3D multiscale model that combines a numerical weather prediction (NWP) model, running at larger scale (e.g. mesoscale), with a city-scale geophysical flow solver (EULAG) for accurate prediction of contaminant (e.g. NO2, PM2.5, PM10) transportation through the street corridors, over buildings and obstacles. A design activity shall be carried out to use mixed-precision computing and energy-efficient accelerators for faster response while preserving results accuracy.

Quantum Simulation (TNM). The Tensor Network Method (TNM) is a class of powerful numerical methods developed to study the equilibrium and out-of-equilibrium properties of strongly correlated many- body quantum systems [55]. TNM are complementary to Monte Carlo methods as they do not suffer from the sign problem. A design activity shall be performed to overcome current TNM limitations studying how pushing the simulation boundaries towards high-dimensional systems with the support of HPC infrastructure. We aim to perform the first scalable 2D and 3D simulations of LGT in and out of equilibrium.

High Energy Physics (HEP). HEP community has built a collection of high-level SW frameworks largely used for simulation and data analysis of the LHC experiments. They include widespread libraries like Geant4 and Fluka for particle-matter simulation, the use of high-level analysis tools like those in ROOT, and simulation packages of high energy collisions. A design activity shall be focused to optimize these SW frameworks for the realization of code bases able to execute on multiple architectures, including the next generations of pre- and exascale EU HPCs.

Biomedical Application (HPC-Drugs). Ligand binding affinity predictions carried out with Molecular Dynamics simulation is one of the main research focus in computational chemistry today due its potential impact in industrial drug discovery. A design activity shall be carried out for HPC-backed pharmaceutical applications based on n-body kernel functions running in specialized cores of GPU and FPGA, relying on recently discovered non-equilibrium thermodynamics theorems and capable of delivering absolute binding free energies of drug-size molecules in a predictable wall-clock time with a credible confidence interval, hence bypassing the limitations of the traditional equilibrium-based Free Energy Perturbation (FEP) alchemical approaches [56].

Reverse Time Migration (RTM). The Reverse Time Migration application and mini-kernels are used within EPI to co-design the STX Accelerator and have been ported to FPGAs within the EuroEXA project. The respective kernels will be analysed to which extent they can leverage the new, energy-efficient, capabilities like Posit arithmetic and lossy compression to enhance performance and energy efficiency. The RTM kernels are stencil-based kernels. Hence, they provide conclusions on many stencil based applications. Reverse Time Migration by FHG for HPC applications to Oil & Gas and Geo-Services.

5. Conclusions

The TEXTAROSSA project aims to achieve a broad impact on the HPC field both in pre-exascale and exascale scenarios. The TEXTAROSSA consortium will develop new IPs, algorithms, methods and software components for HPC-AI, HPC and HPDA applications, mostly Open Source and able to be adopted as standalone building blocks or to interoperate with other Exascale-ready components. Through the participation of three supercomputing centers in the consortium, the proposed technologies will be tested by and known to the HPC community.

Acknowledgements

This work is supported by the TEXTAROSSA project G.A. n.956831, as part of the EuroHPC initiative.

References

[1] V. Kathail, "Xilinx vitis unified software platform," in *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 173–174, 2020.

[2] A. Duran *et al.*, "Ompss: a proposal for programming heterogeneous multi-core architectures," *Parallel processing letters*, vol. 21, no. 02, pp. 173–193, 2011.

[3] C. Augonnet *et al.*, "Starpu: a unified platform for task scheduling on heterogeneous multicore architectures," *Concurrency and Computation: Practice and Experience*, vol. 23, no. 2, pp. 187–198, 2011.

[4] J. Bosch *et al.*, "Application acceleration on fpgas with ompss@fpga," in *FPT 2018, Naha, Okinawa, Japan, December 10-14, 2018*, pp. 70–77, 2018.

[5] X. Tan *et al.*, "A hardware runtime for task-based programming models," *IEEE Trans. Par. Distributed Syst.*, vol. 30, no. 9, pp. 1932–1946, 2019.

[6] J. Devlin *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] O. Beaumont *et al.*, "Optimal checkpointing for heterogeneous chains: how to train deep neural networks with limited memory," Report RR-9302, Inria Bordeaux Sud-Ouest, Nov. 2019.

[8] M. Aldinucci, S. Ruggieri, and M. Torquati, "Porting decision tree algorithms to multicore using FastFlow," in *Conference in Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, vol. 6321 of *LNCS*, pp. 7–23, Sept. 2010.

[9] M. Aldinucci *et al.*, "Fastflow: high-level and efficient streaming on multi-core," in *Programming Multi-core and Many-core Computing Systems*, Parallel and Distributed Computing, ch. 13, 2017.

[10] S. Neuendorffer and K. Vissers, "Streaming systems in fpgas," in *Embedded Computer Systems: Architectures, Modeling, and Simulation*, pp. 147–156, 2008.

[11] M. Aldinucci *et al.*, "Design patterns percolating to parallel programming framework implementation," *International Journal of Parallel Programming*, vol. 42, no. 6, pp. 1012–1031, 2014.

[12] I. Colonnelli *et al.*, "Streamflow: cross-breeding cloud with HPC," *IEEE Transactions on Emerging Topics in Computing*, 2020.

[13] I. Colonnelli *et al.*, "HPC Application Cloudification: The StreamFlow Toolkit," in *PARMA-DITAM HiPEAC workshop, 2021)*, vol. 88 of *OASIcs*, (Dagstuhl, Germany), pp. 5:1–5:13, 2021.

[14] P. Amstutz *et al.*, "Common workflow language, v1. 0," 2016.

[15] P. Stanley-Marbell *et al.*, "Exploiting errors for efficiency: A survey from circuits to applications," *ACM Comp Surveys*, vol. 53, no. 3, 2020.

[16] S. Venkataramani *et al.*, "Approximate computing and the quest for computing efficiency," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, 2015.

[17] S. Cherubin and G. Agosta, "Tools for reduced precision computation: a survey," *ACM Computing Surveys*, vol. 53, Apr 2020.

[18] D. Cattaneo *et al.*, "Embedded operating system optimization through floating to fixed point compiler transformation," in *21st Euromicro Conf on Digital System Design*, pp. 172–176, 2018.

[19] N. Burgess *et al.*, "Bfloat16 processing for neural networks," in *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pp. 88–91, IEEE, 2019.

[20] J. L. Gustafson and I. T. Yonemoto, "Beating floating point at its own game: Posit arithmetic," *Supercomputing Frontiers and Innovations*, vol. 4, no. 2, pp. 71–86, 2017.

[21] D. Zoni, A. Galimberti, and W. Fornaciari, "An fpu design template to optimize the accuracy-efficiency-area trade-off," *Sustainable Computing: Informatics and Systems*, vol. 29, p. 100450, 2021.

[22] M. Cococcioni *et al.*, "Vectorizing posit operations on risc-v for faster deep neural networks: experiments and comparison with arm sve," *Neural Computing and Applications*, pp. 1–11, 2021.

[23] M. Cococcioni *et al.*, "A fast approximation of the hyperbolic tangent when using posit numbers and its application to deep neural networks," in *International Conference on Applications in Electronics Pervading Industry, Environment and Society*, pp. 213–221, 2019.

[24] C. Silvano *et al.*, "The antarex domain specific language for high performance computing," *Microprocessors and Microsystems*, vol. 68, pp. 58–73, 2019.

[25] S. Cherubin, D. Cattaneo, M. Chiari, A. Di Bello, and G. Agosta, "TAFFO: Tuning assistant for floating to fixed point optimization," *IEEE Embedded Systems Letters*, 2019.

[26] S. Cherubin *et al.*, "Dynamic precision autotuning with taffo," *ACM Trans. Archit. Code Optim.*, vol. 17, May 2020.

[27] E. Darulova and V. Kuncak, "Towards a compiler for reals," *ACM Trans. Program. Lang. Syst.*, vol. 39, pp. 8:1–8:28, Mar. 2017.

[28] J. Flich *et al.*, "Exploring manycore architectures for next-generation HPC systems through the MANGO approach," *Microprocessors and Microsystems*, vol. 61, pp. 154 – 170, 2018.

[29] G. Agosta *et al.*, "The RECIPE Approach to Challenges in Deeply Heterogeneous High Performance Systems," *Microprocessors & Microsystems*, 2020.

[30] D. Zoni, L. Cremona, and W. Fornaciari, "All-digital control-theoretic scheme to optimize energy budget and allocation in multi-cores," *IEEE Transactions on Computers*, vol. 69, no. 5, pp. 706–721, 2020.

[31] D. Zoni, L. Cremona, and W. Fornaciari, "All-digital energy-constrained controller for general-purpose accelerators and cpus," *IEEE Embedded Systems Letters*, vol. 12, no. 1, pp. 17–20, 2020.

[32] G. Massari *et al.*, "Predictive resource management for next-generation high-performance computing heterogeneous platforms," in *SAMOS'19*, Jul 2019.

[33] C. Brandolese, S. Corbetta, and W. Fornaciari, "Software energy estimation based on statistical characterization of intermediate compilation code," in *IEEE/ACM International Symposium on Low Power Electronics and Design*, pp. 333–338, 2011.

[34] D. Zoni *et al.*, "Powertap: All-digital power meter modeling for run-time power monitoring," *Microprocessors and Microsystems*, vol. 63, pp. 128–139, 2018.

[35] P. Bellasi, G. Massari, and W. Fornaciari, "Effective runtime resource management using linux control groups with the barbequertrm framework," *ACM Trans. Embed. Comput. Syst.*, vol. 14, pp. 39:1–39:17, Mar. 2015.

[36] F. Terraneo *et al.*, "3D-ICE 3.0: efficient nonlinear MPSoC thermal simulation with pluggable heat sink models," *IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2021.

[37] F. Terraneo *et al.*, "An Open-Hardware Platform for MPSoC Thermal Modeling," in *SAMOS'19*, pp. 184–196, 2019.

[38] A. Leva *et al.*, "Event-based power/performance-aware thermal management for high-density microprocessors," *IEEE Trans on Control Systems Technology*, vol. 26, no. 2, pp. 535–550, 2018.

[39] E. Chung and Alii, "Serving dnns in real time at datacenter scale with project brainwave," *IEEE Micro*, vol. 38, no. 2, pp. 8–20, 2018.

[40] G. Kahn, "The semantics of a simple language for parallel programming," in *Information Processing, 6th IFIP Congress 1974*, pp. 471–475, 1974.

[41] R. Ammendola *et al.*, "Large scale low power computing system: Status of network design in exanest and euroexa projects," *Advances in Parallel Computing*, vol. 32, pp. 750–759, 2018.

[42] M. Katevenis, R. Ammendola, *et al.*, "Next generation of exascale-class systems: Exanest project and the status of its interconnect and storage development," *Microprocessors and Microsystems*, vol. 61, pp. 58 – 71, 2018.

[43] Biagioni, Andrea *et al.*, "Euroexa custom switch: an innovative fpga-based system for extreme scale computing in europe," *EPJ Web Conf.*, vol. 245, p. 09004, 2020.

[44] E. Agullo *et al.*, "Achieving high performance on supercomputers with a sequential task-based programming model," *IEEE TPDS*, 2017.

[45] T. Cojean *et al.*, "Resource aggregation for task-based cholesky factorization on top of modern architectures," *Parallel Computing*, vol. 83, pp. 73–92, 2019.

[46] M. Bernaschi *et al.*, "A factored sparse approximate inverse preconditioned conjugate gradient solver on graphics processing units," *SIAM Journal on Scientific Computing*, vol. 38, no. 1, pp. C53–C72, 2016.

[47] M. Bernaschi, P. D'Ambra, and D. Pasquini, "AMG based on compatible weighted matching for GPUs," *Parallel Computing*, vol. 92, p. 102599, 2020.

[48] M. Bernaschi, P. D'Ambra, and D. Pasquini, "BootCMatchG: An adaptive algebraic multigrid linear solver for GPUs," *Software Impacts*, vol. 6, p. 100041, 2020.

[49] P. D'Ambra and S. Filippone, "A parallel generalized relaxation method for high-performance image segmentation on GPUs," *J. of Computational and Applied Mathematics*, vol. 293, pp. 35–44, 2016.

[50] E. Agullo *et al.*, "Task-based fmm for heterogeneous architectures," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 9, pp. 2608–2629, 2016.

[51] R. Ammendola *et al.*, "NaNet: a flexible and configurable low-latency NIC for real-time trigger systems based on GPUs," *Journal of Instrumentation*, vol. 9, no. 02, p. C02023, 2014.

[52] E. Pastorelli *et al.*, "Gaussian and exponential lateral connectivity on distributed spiking neural network simulation," in *26th Euromicro PDP*, pp. 658–665, IEEE, 2018.

[53] R. Ammendola *et al.*, "The brain on low power architectures-efficient simulation of cortical slow waves and asynchronous states," *Advances in Parallel Computing*, vol. 32, p. 760=769, 2018.

[54] C. Capone *et al.*, "Sleep-like slow oscillations improve visual classification through synaptic homeostasis and memory association in a thalamo-cortical model," *Scientific reports*, vol. 9, no. 1, pp. 1–11, 2019.

[55] A. Omran *et al.*, "Generation and manipulation of schrödinger cat states in rydberg atom arrays," *Science*, vol. 365, no. 6453, pp. 570–574, 2019.

[56] M. Macchiagodena *et al.*, "Virtual double-system single-box: A nonequilibrium alchemical technique for absolute binding free energy calculations: Application to ligands of the sars-cov-2 main protease," *Journal of Chemical Theory and Computation*, vol. 16, no. 11, 2020.