

# The Italian research on HPC key technologies across EuroHPC

## Invited Paper

Marco Aldinucci  
DI - University of Torino  
CINI HPC-KTT laboratory  
Torino, Italy  
marco.aldinucci@unito.it

Claudio A. Ardagna  
Università degli Studi di Milano  
CINI HPC-KTT laboratory  
Milano, Italy  
claudio.ardagna@unimi.it

Biagio Cosenza  
University of Salerno  
CINI HPC-KTT laboratory  
Salerno, Italy  
bcosenza@unisa.it

William Fornaciari  
DEIB - Politecnico di Milano  
CINI HPC-KTT laboratory  
Milano, Italy  
william.fornaciari@polimi.it

Raffaele Montella  
DiST - University of Naples  
"Parthenope"  
CINI HPC-KTT laboratory  
Napoli, Italy  
raffaele.montella@uniparthenope.it

Daniele Simoni  
DIME - University of Genova  
CINI HPC-KTT laboratory  
Genova, Italy  
daniele.simoni@unige.it

Giovanni Agosta  
DEIB - Politecnico di Milano  
CINI HPC-KTT laboratory  
Milano, Italy  
agosta@acm.org

Andrea Bartolini  
DEI - Università di Bologna  
CINI HPC-KTT laboratory  
Bologna, Italy  
a.bartolini@unibo.it

Marco Danelutto  
University of Pisa  
CINI HPC-KTT laboratory  
Pisa, Italy  
marco.danelutto@unipi.it

Roberto Giorgi  
DIISM - University of Siena  
CINI HPC-KTT laboratory  
Siena, Italy  
giorgi@unisi.it

Mauro Olivieri  
Sapienza University of Rome  
CINI HPC-KTT laboratory  
Roma, Italy  
mauro.olivieri@uniroma1.it

Massimo Torquati  
University of Pisa  
CINI HPC-KTT laboratory  
Pisa, Italy  
massimo.torquati@unipi.it

Antonio Andreini  
DIEF - University of Florence  
CINI HPC-KTT laboratory  
Firenze, Italy  
antonio.andreini@unifi.it

Alessandro Cilardo  
University of Naples Federico II  
CINI HPC-KTT laboratory  
Napoli, Italy  
acilardo@unina.it

Roberto Esposito  
DI - University of Torino  
CINI HPC-KTT laboratory  
Torino, Italy  
roberto.esposito@unito.it

Davide Lengani  
DIME - University of Genova  
CINI HPC-KTT laboratory  
Genova, Italy  
davide.lengani@unige.it

Sergio Saponara  
DII-University of Pisa  
CINI HPC-KTT laboratory  
Pisa, Italy  
sergio.saponara@unipi.it

## ABSTRACT

High-Performance Computing (HPC) is one of the strategic priorities for research and innovation worldwide due to its relevance for industrial and scientific applications. We envision HPC as composed

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*CF '21, May 11–13, 2021, Virtual Conference, Italy*

© 2021 Association for Computing Machinery.  
ACM ISBN 978-1-4503-8404-9/21/05...\$15.00  
<https://doi.org/10.1145/3457388.3458508>

of three pillars: infrastructures, applications, and key technologies and tools. While infrastructures are by construction centralized in large-scale HPC centers, and applications are generally within the purview of domain-specific organizations, key technologies fall in an intermediate case where coordination is needed, but design and development are often decentralized. A large group of Italian researchers has started a dedicated laboratory within the National Interuniversity Consortium for Informatics (CINI) to address this challenge. The laboratory, albeit young, has managed to succeed in its first attempts to propose a coordinated approach to HPC research within the EuroHPC Joint Undertaking, participating in the calls 2019-20 to five successful proposals for an aggregate total cost of 95M€. In this paper, we outline the working group's scope and

goals and provide an overview of the five funded projects, which become fully operational in March 2021, and cover a selection of key technologies provided by the working group partners, highlighting their usage development within the projects.

## CCS CONCEPTS

• **Software and its engineering** → **Ultra-large-scale systems; Parallel programming languages; Data flow languages; Extra-functional properties**; • **Hardware** → **Hardware accelerators**; • **Applied computing**;

## KEYWORDS

high-performance computing, parallel programming

### ACM Reference Format:

Marco Aldinucci, Giovanni Agosta, Antonio Andreini, Claudio A. Ardagna, Andrea Bartolini, Alessandro Cilardo, Biagio Cosenza, Marco Danelutto, Roberto Esposito, William Fornaciari, Roberto Giorgi, Davide Lengani, Raffaele Montella, Mauro Olivieri, Sergio Saponara, Daniele Simoni, and Massimo Torquati. 2021. The Italian research on HPC key technologies across EuroHPC: Invited Paper. In *Computing Frontiers Conference (CF '21), May 11–13, 2021, Virtual Conference, Italy*. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3457388.3458508>

## 1 A VISION ON HPC EVOLUTION IN EUROPE

The Italian HPC community has been very active in different, specific research areas, ranging from HPC hardware components to programming tools and applications. Notable contributions have been provided in several EU-funded research initiatives in the framework of EuroHPC and European Processor Initiative roadmaps.

Following the traditional organization of the HPC research community in the three distinct pillars of i) infrastructures, ii) applications, and iii) key technologies and tools, a significant number of researchers and research groups actively participating in the research activities of the third HPC pillar, in late 2019, started a working group in the framework of the National Interuniversity Consortium for Informatics (CINI). CINI is the reference consortium for the Italian national academic research in Computer Science, Computer Engineering, and Information Technology and involves 1,300+ professors of Computer Science and Engineering belonging to 39 public universities. CINI currently runs 11 national research labs in different topics, such as AI, Cybersecurity, Smart city.

In March 2021, the working group transformed into the brand new *HPC key technologies and tools* (HPC-KTT) lab that gathers research groups from 27 Italian universities and eight institutes of the National Research Council (CNR). Marco Aldinucci was selected as the first director.

The main activities of the laboratory include: i) *community building*, exposing competencies, fostering cooperation and coordinating research and dissemination activities, organizing *regular meetings and events* aimed at enforcing cooperation on the HPC key infrastructure themes; ii) *coordination of the submission of research project proposals* to national and international calls, with particular emphasis on specific HPC related EU programs and calls; iii) *organization of institutional meetings* with the main stakeholders of the sector, including national research institutions and administrative entities, iv) *driving innovation in the education process* of undergraduate and

master students on the topics related to HPC key technologies. The working group focuses on different research aspects related to the HPC key technologies and tools, which are described in Section 3.

*Programmability and productivity.* The programming of HPC applications, which is being already tricky, will increase in complexity because of the need to combine domain's specific skills (often in highly specialized domains such as computational chemistry or fluid dynamics) with parallel programming skills, which are becoming the real bottleneck for industrial adoption of HPC. Current and future systems require managing heterogeneous and specialized processors, controlling more non-functional properties beyond performance (e.g., robustness, resilience, security), harmonizing different software stacks (such as scientific computing, Big Data, and AI). A considerable evolution of the entire software stack for HPC and the related development methods will be necessary to solve this problem, such that levels gradually closer to the application world provide different abstraction levels to the HPC application programmer.

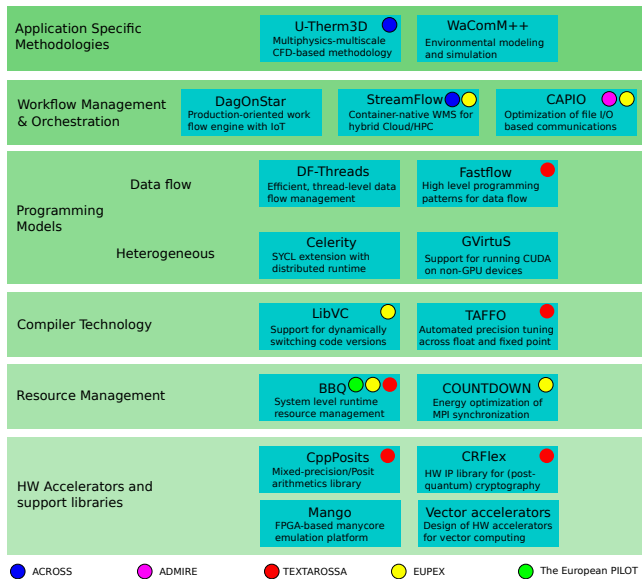
*Energy efficiency.* The power required by an HPC center cannot grow beyond local supply limits, yet the forecast is that post-exascale systems can consume more than 80 MW. To tackle this problem, HPC infrastructures need to evolve in both hardware and software terms. 1) *Hardware*: it is possible to obtain more efficient hardware through the use of specialized heterogeneous accelerators (typically of the reconfigurable type) and/or through "close to memory" calculation. These systems can have a particularly effective impact in the areas of Big Data and Deep Learning. 2) *Software*: the entire software stack should consider the energy consumed so that applications are made aware of the energy consumed at run time. Adaptivity and autotuning will be critical for efficient energy management of long-life applications.

*Programming models.* The programming models should include the possibility of managing energy efficiency and the other non-functional aspects in the same way as all the other more typical issues of the HPC world (e.g., those related to performance). However, they should also allow separating the treatment of functional and extra requirements. Furthermore, it will be essential to ensure the possibility of integrating support for accelerators (even of different types) transparently.

*Compilers.* Compiler technology can address extra-functional properties providing performance and automatic optimization of program properties (e.g., automatic scaling of calculation accuracy).

*High-level synthesis.* It is necessary to provide high-level synthesis systems that can take full advantage of heterogeneous and reconfigurable hardware systems. While dynamic reconfiguration times are not an issue in HPC systems, building custom accelerators requires additional skills that are generally quite difficult to come by. To ensure the possibility of designing custom accelerators, it is necessary to develop a new generation of tools for the automatic generation of components.

*Multi-tenancy and security.* HPC platforms are increasingly exposed to critical data in areas like health, finance, or data analytics. Reasoning on how to enforce privacy and security for critical data on shared infrastructures (as supercomputers) is crucial. Hence,



**Figure 1: CINI Technologies across the EuroHPC projects, classified according to their positioning in the HPC technology stack**

it is of paramount importance to exploit the security features in emerging hardware/software technologies, e.g., Intel SGX and containerization, to support insulated (private, secure, trusted) execution environments, enabling complete segregation in the future multi-tenant HPC.

*HPC cloud services.* HPC and cloud are convergent technologies. The migration of HPC applications into services is already ongoing, and we expect to observe an acceleration along the direction of Software-as-a-Service of significant extreme-scale codebases. An enabling technology for this to happen is workflow management systems that can bridge cloud and HPC system software.

*Scientific software algorithms and libraries.* New algorithms and scientific software libraries need to be developed that support computation at different levels of precision, energy efficiency, and heterogeneity. This requires rethinking many computational kernels of HPC applications in search of a compromise between reducing the total energy used in the computation and minimizing the solutions' development time, promoting numerical stability, reproducibility of results scalability.

A taxonomy of the solutions that CINI proposes to address these critical technical goals is reported in Figure 1, along with the developed tools and frameworks.

## 2 THE FUNDED PROJECTS

The working group has been actively participating in several calls in the EuroHPC framework in 2020. Different groups belonging to different universities have participated in different project proposals as members of the CINI HPC-KTT lab figuring as a single Italian research entity with a full range of complementary and synergistic expertise. Eventually, this favored the participation in

several successful project proposals both in H2020-JTI-EuroHPC-2019-1 (Towards Extreme Scale Technologies and Applications) and H2020-JTI-EuroHPC-2020-01 (Advanced pilots towards the European exascale supercomputers) calls. The total cost of the described projects is over 95M€ (for 4 years). This section provides a brief overview of the projects funded with the CINI HPC-KTT working group's participation as a partner.

### 2.1 ACROSS

The *ACROSS project* aims to build an exascale-ready, HPC and data-driven execution platform that supports modern complex workflows mixing HPC, Big Data, and AI high-level tasks. ACROSS will leverage on the next generation of pre-exascale infrastructures, still being ready for exascale systems, and on effective mechanisms to easily describe and manage complex workflows. These resources will be exploited to create value and innovation within the Aviation, Climate and Weather, and Energy industry. The simulation of the flow field within the turbine aero-engine requires several HPC resources and generates big data. ACROSS will develop user-defined in-situ data processing to improve physical understanding and identify and speed-up the most efficient design points leading to the innovation of the aero-engine. The multiphysics process occurring in the combustion chamber of aero-engines (turbulent flows, combustion, convection, radiation) is challenging and demanding for the CFD (Computational Fluid Dynamics) modeling of the component aimed at its optimization in terms of pollutant emissions and durability. In the ACROSS project, a multiscale and multiphysics methodology for the scale resolving CFD analysis of aero-engine combustors will be tested and improved to permit its exploitation at the industrial level on large HPC infrastructures.

### 2.2 ADMIRE

The *ADMIRE project* aims to overcome the flat storage hierarchies' performance limitations in today's HPC infrastructures. Careful control and appropriate API and policies are needed to avoid congestion and balance computation and I/O requirements to maximize storage performance. The ADMIRE project proposes a software-only solution to create an active I/O stack that dynamically adjusts computation and storage requirements through intelligent global coordination, the malleability of computation and I/O, and the scheduling of storage resources along with all levels of the storage hierarchy. The primary objective is to increase individual HPC applications' performance by leveraging fast and power-efficient node-local storage tiers using ad-hoc storage systems and exploiting in-transit/in-situ processing facilities. The ADMIRE I/O software stack will be validated with several use cases from various domains, including climate/weather, life sciences, physics, remote sensing, and deep learning.

### 2.3 TEXTAROSSA

The *TEXTAROSSA project* aims to achieve high performance and high energy efficiency on near-future exascale computing systems by increasing efficiency of computation with extreme efficiency in HW and new arithmetics, as well as to provide methods and tools for seamless integration of reconfigurable accelerators in

heterogeneous HPC multi-node platforms. The main directions for innovation are towards:

- (a) enabling mixed-precision computing, through the definition of IPs, libraries, and compilers supporting novel data types (including Posits), also used to boost the performance of stencil/tensor accelerators;
- (b) implementing new multilevel thermal management and two-phase liquid cooling;
- (c) developing improved data movement and storage tools through efficient lossy compression;
- (d) ensure secure HPC operation through advanced cryptography;
- (e) providing RISC-V based IP for fast task scheduling and IPs for low-latency intra/inter-node communication.

These technologies will be tested on the Integrated Development Vehicles mirroring and extending the European Processor Initiative ARM64-based architecture. To drive the technology development and assess the impact of the proposed innovations, from node to system levels, TEXTAROSSA will use a selected but representative number of HPC, HPDA, and AI demonstrators covering challenging domains such as general-purpose numerical kernels, High Energy Physics (HEP), Oil & Gas, climate modeling, and emerging domains such as HPDA and High-Performance AI.

## 2.4 EUPEX

The *EUPEX project* (EUropean Pilot for EXascale) aims to build and validate the first HPC platform integrating European, general-purpose processor technology (EPI), European interconnect technology (BXI), and a European software stack for HPC. EUPEX will co-design a modular exascale-pilot system, integrating European hardware and software technologies to demonstrate the readiness and the scalability of these technologies towards exascale and prepare applications and European users to exploit the future exascale machines efficiently. The hardware platform is sized to be large enough for relevant application preparation and scalability forecast and a proof of concept for a modular architecture relying on European technologies and the European processor technology. The software stack will comprise all necessary software components, combining developments of European-funded R&D projects with best-of-breed open source software where needed. This SW stack will enable the efficient exploitation of the resources available in large-scale modular systems. This way, the co-design applications will be able to demonstrate exascale readiness on the EUPEX platform.

## 2.5 The European PILOT

The *European PILOT* (Pilot using Independent Local & Open Technology) aims to be the first demonstration of two all-European HPC RISC-V-based accelerators, designed, implemented, manufactured, and owned by Europe. The European PILOT combines open source SW and HW design to deliver the first entirely European full-stack software and integrated ecosystem-based on RISC-V accelerators, coupled to any general-purpose CPU via PCIe Gen 6.0 or CXL 3.0. The accelerators will be manufactured in the new GlobalFoundries's 12 nm 3D FinFET transistor technology. This pre-production system can be realized with a combination of existing IPs coming from the ongoing European Processor Initiative project, HW verification,

and emulation based on FPGAs and real silicon IC prototypes that demonstrate the full HW/SW stack feasibility. The implemented applications will span from AI to HPC, while the aggressive chiplet-based IC implementation will be the smallest technology node manufactured in Europe, to be easily adapted for a near-future HPC product fabrication.

## 3 CINI HPC KEY TECHNOLOGIES

In this section, we report on the key technologies that the CINI HPC-KTT and its constituents have developed. Their adoption by the five EuroHPC projects is reported in Figure 1.

*Streamflow.* The StreamFlow framework [14, 15] is a container-native Workflow Management System (WMS) written in Python 3 and based on the Common Workflow Language (CWL) Standard [5]. 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<sup>1</sup>.

*Fastflow.* FastFlow [4] 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.

*CAPIO.* The Cross-Application Programmable I/O (CAPIO) is a research-grade software layer capable of coordinating legacy modules of data-driven large-scale software pipelines that cooperate using the I/O storage as a temporary communication buffer. Through the transparent mapping of standard I/O calls, CAPIO moves data directly between application components, excluding the storage from the critical path. The resulting application-tailored I/O infrastructure reduces the congestion on the back-end layers and increases the opportunity to exploit data locality.

Besides, the emerging inclusion of fast storage tiers in HPC infrastructures enables on-the-fly data processing while data is generated at temporary locations to reduce the final size of the dataset and enable the co-execution of data-intensive analytics tasks together with other computationally intensive tasks. CAPIO provides the suitable entry points to inject in-transit computations

<sup>1</sup><https://streamflow.di.unito.it>

of data at the edge of the cooperating components. CAPIO is an open-source project under the LGPL v3.0 license<sup>2</sup>.

*BBQ*. Barbeque (BBQ) is a run-time resource manager (RTRM) that enables node-level resource management across multiple applications, taking into account the available resources (processing elements, memories, bandwidth, etc.) as well as application needs and priorities expressed through “recipes”, which provide different operating points for the application, using different amounts of resources [3]. The resource management is dynamic, allowing resource reallocation through application feedback, which is provided by collecting heartbeats. BBQ supports three main integration modes: (1) the OpenCL and *mangolibs* programming models for C/C++, which support transparent integration with BBQ for heterogeneous platforms; (2) the Adaptive Execution Model, which is a generic model available in C++, Python and Fortran and allows management of homogeneous platforms; (3) a lightweight control, where no integration is needed, but there is no feedback on application performance.

*Mango*. The MANGO FPGA-based manycore emulation platform has been developed in the framework of an H2020 FETHPC project [17]. The platform provides a large-scale cluster of multi-FPGA boards intended for experimenting with customized many-core systems, at the level of both processor and interconnect/system architecture, along with the supporting software stack. Among other developments, the project has delivered an open-source configurable GPU-like processor and related compilation toolchain. The cluster prototype is currently maintained for demonstration and will be instrumental to the activities of the CINI Lab. As a further development, the RECIPE project [2] is extending the architecture-related investigation of MANGO by developing a multi-FPGA node with additional features of relevance for HPC, particularly checkpoint/restart functions and remote direct accelerator memory access, to be used for example in MPI one-sided communication primitives with heterogeneous workloads.

*Celerity*. Celerity [33] is a programming model that allows programmers to write highly parallel applications that can be run on a cluster of accelerator nodes. It consists of a programming interface that extend SYCL [32], an open-standard high-level C++ language by Khronos Group for programming accelerators, and a distributed runtime system [20], which is responsible for the asynchronous scheduling and distribution of the work. Celerity also provides a compilation framework used for modeling and optimization and a large selection of benchmarks [22].

*DF-Threads*. DF-THREADS[18] is a low-level API which allows an efficient management of thread-level data flow; DF-Threads can outperform by more than one order of magnitude other well-known similar APIs like the Open Community Runtime (OCR). While several parallel programming models like OmpSs, Legion, Dharma are being mapped on OCR, DF-Threads offers an alternative implementation both in software and hardware [19] to achieve better energy efficiency by reducing unnecessary synchronization and data movement and to achieve resiliency for HPC applications due to the idempotency of DF-threads computations [34].

*TAFFO and LIBVC*. TAFFO [12] is a set of plugins for the LLVM Compiler Framework that enables automated precision tuning. In HPC, often wide float formats are used even when a lower precision could be successfully employed at a better performance/energy point. TAFFO allows fine-tuning the selection of data types, taking into account the cost of casting. It can be coupled with a code versioning library to dynamically perform the tuning, which has been proven a key point in addressing large scale applications where the input data characteristics can change over time [11]. LIBVC [10] is the code versioning library adopted for TAFFO. It has been developed as a tool to support the online exploration of code specialization opportunities in a scenario of continuous optimization in HPC application scenarios, as part of the ANTAREX project [30, 31].

*COUNTDOWN*. COUNTDOWN [7, 8] is a runtime library for performance-neutral energy saving in MPI applications. It saves energy only during MPI synchronization without inducing a time-to-solution increase for applications. Indeed energy-consumption minimization alone can lead to execution time penalties, which reduces capacity and does not always result in a reduction of the total cost of ownership [6]. With COUNTDOWN the energy saving is obtained transparently to the user, without requiring application code modifications or recompilation of the application. The COUNTDOWN runtime will intercept MPI calls in which to reduce the power consumption and the CPU performance, and separate waiting time from copy time and computing time.

*GVirtuS*. It enables the execution of CUDA (and OpenCL) kernels on physical or virtual machines unprovided of general-purpose GPU acceleration [26]. GVirtuS components (the front-end providing the stub libraries, the back-end dealing with the physical accelerated device, and the communicator enabling the remote call invocation) are independent of the hypervisor, the communication technology, and the virtualized/remoted general-purpose GPU [23]. GVirtuS is transparent for developers: no changes are required in the software source code to leverage on virtualization/remoting or any peculiar programming model except the CUDA and OpenCL standards. Within the H2020 RAPID project, GVirtuS has been used in mobile computing acceleration and computation at the edge scenarios [21] where an Android interface has been made available. GVirtuS is an open-source project under the Apache 2.0 license<sup>3</sup>.

*DagOnStar*. Named after the Phoenicians’ god known by ancient Greeks as Triton, it enables the execution of direct acyclic graph (DAG) jobs on anything, ranging from the local machine to virtual HPC clusters hosted on private, public or hybrid clouds [24]. DagOnStar is a production-oriented workflow engine targeting computational scientists focused on operational applications for routinely produced weather and marine forecasts. DagOnStar implements some peculiar features as the workflow:// schema for data-flow implicit dependencies and external application sandbox execution; task-level parallelism based on virtual containers and microservices [28] and the IoT components orchestration [29]. DagOnStar is an open-source project under the MIT license<sup>4</sup>.

<sup>2</sup><https://github.com/alpha-unito/capio>

<sup>3</sup><https://github.com/gvirtus>

<sup>4</sup><https://github.com/dagonstar>

*CppPosits.* CPPPOSITs [13] is a library developed at DII-University of Pisa, supporting mixed-precision and Posits arithmetic, and compliant with LLVM compiler, to increase the efficiency of AI and video computing kernels. CppPosits has been ported on ARM SVE and RISC-V with Vector extension ISA, proving that the same accuracy of FP32 can be achieved by reducing by a factor of 4 the data transfer and storage cost.

*CRFlex.* CRFLEX is a library of HW accelerator IPs developed at DII-University of Pisa, supporting energy-efficient implementation of cryptographic algorithms for symmetric and public key cryptography, hashing, digital signature and verification, secure key generation and management, random number and seed generation. The library includes the support also of new Lattice-based schemes being ready for post-quantum cryptography applications. The IP library has been verified by porting it on reconfigurable HW (FPGAs/FPSOCs) and nanoscale ASICs (<https://www.european-processor-initiative.eu/dissemination-material/crypto-tile-factsheet/>).

*U-Therm3D.* U-THERM3D is a multiphysics-multiscale CFD based methodology developed at DIFE-University of Florence [27] to handle complex phenomena such as the turbulent combustion occurring in gas turbine combustors. Several physical phenomena are involved and strongly interact during the process, requiring dedicated numerical models. The concurring physics (turbulence, chemical kinetics, radiation and heat conduction) are also interested by huge differences in the characteristic time scales which indeed need to be solved in order to properly describe the process. U-THERM3D is based on well established high-fidelity unsteady CFD methods such as LES (Large Eddy Simulation) adopting a time de-synchronization to manage the multiscale nature of the problem, permitting to greatly accelerate the convergence. Different solvers used for the involved physics are managed and integrated in parallel SMP approach suitable for HPC environment.

*Vector Processing Hardware Acceleration.* Sapienza University developed the RTL implementation of vector processing units in the context of Embedded HPC (Klessydra processor family [9]) and contributed the core RTL design of the Vector Processing Unit in the European Processor Initiative project [1]. The intrinsic parameterization of the vector processing subsystem in Klessydra processors opens the way to configurable vector accelerators in FPGA devices, or to silicon implementations optimally customized for a specific application domain. The binary interface visible to the application programmer is a custom extension of the RISC-V instruction set, and it is integrated in the GCC compiler tool-chain by means of a lightweight set of C language intrinsic functions. Further possibilities include the support of reduced precision arithmetic and the hardware support for resilience to faulty bits, the latter being based on innovative resilience simulation integrated in a Universal Verification Methodology environment.

## 4 OUR APPLICATIONS

While the focus of the HPC-KTT workgroup is on key enabling technologies, members of the workgroup also develop several applications that can serve as a vehicle for demonstrating the key technologies. In this section, we briefly describe two such applications in key industrial and scientific domains.

*Environmental modeling.* Simulation of pollutants transport and dispersion in inshore and offshore marine environments using WaComM++ (Water quality Community Model Plus Plus) [25]. WaComM++ is an MPI, OpenMP, and CUDA hierarchical and heterogeneous parallel model simulating Lagrangian particles as inert tracers. Its parallelization schema enables the user to run the model using advanced High-Performance Computing architectures in any combination of distributed memory processes, shared memory threads, and single or multiple CUDA-enabled GPU devices efficiently and effectively. WaComM++ has a production role in forecasting the contamination of the mussels produced in the farms of the Campania Region, Italy [16]. WaComM++ is an open-source project under the MIT license<sup>5</sup>.

*Greener aeroengines.* ACROSS project will investigate the applicability to the aeronautic sector of breakthrough solutions merging advanced HPC resources with innovative numerical methods based on multiphysics optimization capability and boosted by Artificial Intelligence (AI). The main focus is on the complex workflows typically faced in the design of critical aeroengine components. Two aeronautical pilot cases are considered to improve knowledge and control in a critical area of the engine for what concerns emissions and fuel consumption (efficiency): the combustor, faced by DIFE-University of Florence with U-THERM3D and the low-pressure turbine, studied by DIME-University of Genova. A significant impact in terms of design quality and design-to-market time reduction is expected: this is a fundamental step to improve engineering productivity in New Technology Introduction (NTI) and New Product Introduction (NPI) aeronautical processes.

## 5 FUTURE DIRECTIONS

The HPC-KTT working group turned into an entire National Laboratory within CINI in March 2021. The laboratory will increase the group's national and international visibility and provide a more robust collaboration framework looking forward to the forthcoming Horizon Europe funding programme.

From the technology perspective, the technologies developed by the HPC-KTT working group set it in the ideal position to pursue research and development in the most promising directions for HPC technology, including, among others: (1) the development and exploitation of heterogeneous accelerators; (2) the development of tools to enable the convergence of cloud computing technologies with HPC infrastructures, such as multi-tenancy approaches; (3) the development of effective and scalable tools to control the runtime behavior of ultra-large-scale systems, and the enforcement of extra-functional properties at system and application level; (4) the development of machine learning and artificial intelligence tools that would enhance the effectiveness in designing key industrial components.

## ACKNOWLEDGEMENTS

This work has been partially supported by the EuroHPC JU and the Italian MISE by way of the following projects: ACROSS (G.A. n. 955648), ADMIRE (G.A. n. 956748), TEXTAROSSA (G.A. n. 956831), EUPEX (G.A. 101033975), The European PILOT (G.A. n. 101034126).

<sup>5</sup><https://github.com/ccmma/wacomplusplus>



## REFERENCES

- [1] 2021. *European Processor Initiative*. <https://www.european-processor-initiative.eu/>
- [2] Giovanni Agosta, William Fornaciari, David Atienza, Ramon Canal, Alessandro Cilardo, José Flich Cardo, Carles Hernandez Luz, Michal Kulczewski, Giuseppe Massari, Rafael Tornero Gavilá, and Marina Zapater. 2020. The RECIPE approach to challenges in deeply heterogeneous high performance systems. *Microprocessors and Microsystems* 77 (2020), 103185. <https://doi.org/10.1016/j.micpro.2020.103185>
- [3] Giovanni Agosta, William Fornaciari, Giuseppe Massari, Anna Pupykina, Federico Reghezani, and Michele Zanella. 2018. Managing heterogeneous resources in hpc systems. In *Proceedings of the 9th Workshop and 7th Workshop on Parallel Programming and RunTime Management Techniques for Manycore Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms (PARMA-DITAM)*, 7–12. <https://doi.org/10.1145/3183767.3183769>
- [4] Marco Aldinucci, Marco Danelutto, Peter Kilpatrick, and Massimo Torquati. 2017. FastFlow: High-Level and Efficient Streaming on multi-core. In *Programming Multi-core and Many-core Computing Systems*, Sabri Pllana and Fatos Xhafa (Eds.). John Wiley & Sons, Inc, Chapter 13. <https://doi.org/10.1002/9781119332015.ch13>
- [5] Peter Amstutz, Michael R Cruseo, Nebojša Tijanić, Brad Chapman, John Chilton, Michael Heuer, Andrey Kartashov, Dan Leehr, Hervé Ménager, Maya Nedeljkovich, et al. 2016. Common workflow language, v1. 0. (2016).
- [6] Andrea Borghesi, Andrea Bartolini, Michela Milano, and Luca Benini. 2019. Pricing schemes for energy-efficient HPC systems: Design and exploration. *The International Journal of High Performance Computing Applications* 33, 4 (2019), 716–734. <https://doi.org/10.1177/1094342018814593>
- [7] Daniele Cesarini, Andrea Bartolini, Piero Bonfà, Carlo Cavazzoni, and Luca Benini. 2020. COUNTDOWN: a Run-time Library for Performance-Neutral Energy Saving in MPI Applications. *IEEE Trans. Comput.* (2020), 1–1. <https://doi.org/10.1109/TC.2020.2995269>
- [8] Daniele Cesarini, Andre Bartolini, Andre Borghesi, Carlo Cavazzoni, Mathieu Luisier, and Luca Benini. 2020. Countdown Slack: A Run-Time Library to Reduce Energy Footprint in Large-Scale MPI Applications. *IEEE Transactions on Parallel and Distributed Systems* 31, 11 (2020), 2696–2709. <https://doi.org/10.1109/TPDS.2020.3000418>
- [9] Abdallah Cheikh, Stefano Sordillo, Antonio Mastrandrea, Francesco Menichelli, Giuseppe Scotti, and Mauro Olivieri. 2021. Klessydra-T: Designing Vector Co-processors for Multi-Threaded Edge-Computing Cores. *IEEE Micro* (2021), 1–1. <https://doi.org/10.1109/MM.2021.3050962>
- [10] Stefano Cherubin and Giovanni Agosta. 2018. libVersioningCompiler: An easy-to-use library for dynamic generation and invocation of multiple code versions. *SoftwareX* 7 (2018), 95 – 100. <https://doi.org/10.1016/j.softx.2018.03.006>
- [11] Stefano Cherubin, Daniele Cattaneo, Michele Chiari, and Giovanni Agosta. 2020. Dynamic precision autotuning with TAFFO. *ACM Transactions on Architecture and Code Optimization (TACO)* 17, 2 (2020), 1–26. <https://doi.org/10.1145/3388785>
- [12] Stefano Cherubin, Daniele Cattaneo, Michele Chiari, Antonio Di Bello, and Giovanni Agosta. 2019. TAFFO: Tuning assistant for floating to fixed point optimization. *IEEE Embedded Systems Letters* 12, 1 (2019), 5–8. <https://doi.org/10.1109/LES.2019.2913774>
- [13] Marco Cococcioni, Federico Rossi, Emanuele Ruffaldi, and Sergio Saponara. 2021. Novel Arithmetics in Deep Neural Networks Signal Processing for Autonomous Driving: Challenges and Opportunities. In *IEEE Signal Processing Magazine vol 38 n 1*. IEEE, 97–110.
- [14] Iacopo Colonnelli, Barbara Cantalupo, Roberto Esposito, Matteo Pennisi, Conetto Spampinato, and Marco Aldinucci. 2021. HPC Application Cloudification: The StreamFlow Toolkit. In *12th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and 10th Workshop on Design Tools and Architectures for Multicore Embedded Computing Platforms (PARMA-DITAM 2021) (OASlcs)*, Vol. 88. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 5:1–5:13. <https://doi.org/10.4230/OASlcs.PARMA-DITAM.2021.5>
- [15] Iacopo Colonnelli, Barbara Cantalupo, Ivan Merelli, and Marco Aldinucci. 2020. StreamFlow: cross-breeding cloud with HPC. *IEEE Transactions on Emerging Topics in Computing* (2020). <https://doi.org/10.1109/TETC.2020.3019202>
- [16] Diana Di Luccio, Angelo Riccio, Ardelio Galletti, Giuliano Laccetti, Marco Lapegna, Livia Marcellino, Sokol Kosta, and Raffaele Montella. 2020. Coastal marine data crowdsourcing using the Internet of Floating Things: Improving the results of a water quality model. *IEEE Access* 8 (2020), 101209–101223.
- [17] José Flich, Giovanni Agosta, Philipp Ampletzer, David Atienza Alonso, Carlo Brandolese, Alessandro Cilardo, William Fornaciari, Ynse Hoornenborg, Mario Kovač, Bruno Maitre, Giuseppe Massari, Hrvoje Mlinarić, Ermis Papastefanakis, Fabrice Roudet, Rafael Tornero, and Davide Zoni. 2016. Enabling HPC for QoS-sensitive applications: The MANGO approach. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, 702–707.
- [18] Roberto Giorgi and Paolo Faraboschi. 2014. An Introduction to DF-Threads and their Execution Model. In *IEEE MPP*, Paris, France, 60–65. <https://doi.org/10.1109/SBAC-PADW.2014.30>
- [19] Roberto Giorgi and Alberto Scionti. 2015. A scalable thread scheduling co-processor based on data-flow principles. *ELSEVIER Future Generation Computer Systems* 53 (dec 2015), 100–108. <https://doi.org/10.1016/j.future.2014.12.014>
- [20] Ivan Grasso, Simone Pellegrini, Biagio Cosenza, and Thomas Fahringer. 2013. LibWater: heterogeneous distributed computing made easy. In *International Conference on Supercomputing (ICS)*. ACM, 161–172.
- [21] Giuliano Laccetti, Raffaele Montella, Carlo Palmieri, and Valentina Pelliccia. 2013. The high performance internet of things: using GVirtuS to share high-end GPUs with ARM based cluster computing nodes. In *International Conference on Parallel Processing and Applied Mathematics*. Springer, 734–744.
- [22] Sohan Lal, Aksel Alpay, Philip Salzmann, Biagio Cosenza, Alexander Hirsch, Nicolai Stawinoga, Peter Thoman, Thomas Fahringer, and Vincent Heuveline. 2020. SYCL-Bench: A Versatile Cross-Platform Benchmark Suite for Heterogeneous Computing. In *26th International Conference on Parallel and Distributed Computing (Euro-Par) (Lecture Notes in Computer Science)*, Vol. 12247. Springer, 629–644.
- [23] Antonio Mentone, Diana Di Luccio, Luca Landolfi, Sokol Kosta, and Raffaele Montella. 2019. CUDA virtualization and remoting for GPGPU based acceleration offloading at the edge. In *International Conference on Internet and Distributed Computing Systems*. Springer, 414–423.
- [24] Raffaele Montella, Diana Di Luccio, and Sokol Kosta. 2018. DagOn\*: Executing direct acyclic graphs as parallel jobs on anything. In *2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*. IEEE, 64–73.
- [25] Raffaele Montella, Diana Di Luccio, Pasquale Troiano, Angelo Riccio, Alison Brizius, and Ian Foster. 2016. WaComM: A parallel Water quality Community Model for pollutant transport and dispersion operational predictions. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 717–724.
- [26] Raffaele Montella, Giulio Giunta, Giuliano Laccetti, Marco Lapegna, Carlo Palmieri, Carmine Ferraro, Valentina Pelliccia, Cheol-Ho Hong, Ivor Spence, and Dimitrios S Nikolopoulos. 2017. On the virtualization of CUDA based GPU remoting on ARM and X86 machines in the GVirtuS framework. *International Journal of Parallel Programming* 45, 5 (2017), 1142–1163. <https://doi.org/10.1007/s10766-016-0462-1>
- [27] Simone Paccati, Davide Bertini, Lorenzo Mazzei, Stefano Puggelli, and Antonio Andreini. 2020. Large-Eddy Simulation of a Model Aero-Engine Sooting Flame With a Multiphysics Approach. *Flow, Turbulence and Combustion* (Aug. 2020). <https://doi.org/10.1007/s10494-020-00202-5>
- [28] Dante D Sánchez-Gallegos, Diana Di Luccio, JL Gonzalez-Compean, and Raffaele Montella. 2019. A Microservice-Based Building Block Approach for Scientific Workflow Engines: Processing Large Data Volumes with DagOnStar. In *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 368–375.
- [29] Dante D Sánchez-Gallegos, Diana Di Luccio, José Luis Gonzalez-Compean, and Raffaele Montella. 2019. Internet of Things orchestration using DagOn\* workflow engine. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 95–100.
- [30] Cristina Silvano, Giovanni Agosta, Jorge Barbosa, Andrea Bartolini, Andrea R. Beccari, Luca Benini, João Bispo, João M. P. Cardoso, Carlo Cavazzoni, Stefano Cherubin, Radim Cmar, Davide Gadioli, Candida Manelfi, Jan Martinović, Ricardo Nobre, Gianluca Palermo, Martin Palković, Pedro Pinto, Erven Rohou, Nico Sanna, and Katerina Slaninová. 2017. The ANTAREX tool flow for monitoring and autotuning energy efficient HPC systems. In *2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, 308–316. <https://doi.org/10.1109/SAMOS.2017.8344645>
- [31] Cristina Silvano, Giovanni Agosta, Andrea Bartolini, Andrea R Beccari, Luca Benini, João Bispo, Radim Cmar, João MP Cardoso, Carlo Cavazzoni, Jan Martinović, Gianluca Palermo, Martin Palkovic, Pedro Pinto, Erven Rohou, Nico Sanna, and Katerina Slaninová. 2016. AutoTuning and Adaptivity appRoach for Energy efficient eXascale HPC systems: the ANTAREX Approach. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 708–713.
- [32] SYCL Working Group. 2020. *SYCL 2020 Provisional Specification*. Technical Report.
- [33] Peter Thoman, Philip Salzmann, Biagio Cosenza, and Thomas Fahringer. 2019. Celerity: High-Level C++ for Accelerator Clusters. In *25th International Conference on Parallel and Distributed Computing (Euro-Par) (Lecture Notes in Computer Science)*, Vol. 11725. Springer, 291–303. [https://doi.org/10.1007/978-3-030-29400-7\\_21](https://doi.org/10.1007/978-3-030-29400-7_21)
- [34] Sebastian Weis, Arne Garbade, Bernhard Fechner, Avi Mendelson, Roberto Giorgi, and Theo Ungerer. 2016. Architectural Support for Fault Tolerance in a Teradevice Dataflow System. *Springer International Journal of Parallel Programming* 44, 2 (Apr 2016), 208–232. <https://doi.org/10.1007/s10766-014-0312-y>