The Italian Pathfinding Strategy for RISC-V-based HPC: Progress within DARE-SGA1 project ⋆

Giovanni Agosta[a,b], Marco Aldinucci[a,c], Marcello Barbirotta[g], Andrea Bartolini[d], Laura Bellentani[a,e], Andrea Biagioni[k], Daniele Cattaneo[a,b], Daniele Cesarini[a,e], Carlotta Chiarini[k], Iacopo Colonnelli[a,c], Francesco Conti[d], Pietro Delugas[a,f], Lev Denisov[a,b], Ottorino Frezza[k], Angelo Garofalo[d], Marco Grangetto[a,c], Juan David Guerrero Balaguera[a,j], Francesca Lo Cicero[k], Alessandro Lonardo[k], Michele Martinelli[k], Andrea Maslov[a,h], Francesco Menichelli[a,g], Mauro Olivieri[a,g], Pierpaolo Perticaroli[k,g], Luca Pontisso[k], Cristian Rossi[k,g], Josie Esteban Rodriguez Condia[a,j], Daniele Rossi[a,i], Davide Rossi[d], Sergio Saponara[a,i], Antonio Sciarappa[a,h], Francesco Simula[k], Matteo Sonza Reorda[a,j], Gabriele Spadaro[a,c], Massimo Torquati[a,i], Piero Vicini[k,*]

[a] *ICSC – Italian Research Center on High Performance Computing, Big Data and Quantum Computing, Casalecchio di Reno (BO), Italy*
[b] *Politecnico di Milano, Milano (MI), Italy [name].[surname]@polimi.it*
[c] *Università degli Studi di Torino, Torino, Italy [name].[surname]@unito.it*
[d] *Alma Mater Università degli Studi di Bologna, Bologna, Italy [name].[surname]@unibo.it*
[e] *CINECA, Casalecchio di Reno (BO), Italy [name].[surname]@unibo.it*
[f] *Scuola Internazionale Superiore di Studi Avanzati (SISSA), Trieste, Italy [name].[surname]@sissa.it*
[g] *Sapienza Università di Roma, Rome, Italy [name].[surname]@uniroma1.it*
[h] *Leonardo, Genova, Italy [name].[surname]@leonardo.com*
[i] *Università degli Studi di Pisa, Pisa, Italy [name].[surname]@unipi.it*
[j] *Politecnico di Torino, Torino, Italy [name].[surname]@polito.it*
[k] *Istituto Nazionale di Fisica Nucleare (INFN) - Sezione di Roma, Roma, Italy [name].[surname]@roma1.infn.it*

Abstract

Driven by the need of technological sovereignty, the European Union has accelerated the development of a sustainable High-Performance Computing (HPC) ecosystem. In this perspective, the DARE SGA1 (Digital Autonomy with RISC-V in Europe) project seeks to decouple European infrastructure from proprietary, non-European hardware by developing high-performance processors and computing accelerators based on the open-source RISC-V architecture. This paper introduces and reports about the current status of the multifaceted contributions of Italian partners to the critical pathfinding activities within the DARE SGA1 framework aiming at closing the gap between current prototypes and future HPC systems as well as establishing a bridge towards a fully independent European supercomputing supply chain.

In particular, the activities focus on i) novel RISC-V-based accelerator designs optimised for energy efficiency and high throughput and scaling-out capability; ii) improvement of RISC-V-based system reliability and resiliency; iii) evolution of the software stack for low level software and compiler optimisations iv) co-design through benchmarking with AI and Quantum Chemistry scalable applications.

Keywords: RISC-V, VPU, HPC, system software, RAS, Quantum Chemistry

1. Introduction

The uncertainty of the value chains and the need to ensure the European Union's competitiveness in the global economy have pushed policymakers to identify High-Performance Computing (HPC), Internet of Things (IoT), Artificial Intelligence (AI), Quantum Computing and advanced (nano-)materials among the Key Enabling Technologies (KET), for which there is a pressing need to reach technological sovereignty [34]. Technological sovereignty in these fields is hindered by dependence on non-European suppliers, lack and/or draining of technical expertise and difficulties in technology transfer to the market.

To address these issues, as well as to improve the overall

⋆The final published version is available at https://doi.org/10.1016/j.micpro.2026.105321

*Corresponding author (piero.vicini@roma1.infn.it)

coordination of European actions towards technological sovereignty, several large-scale initiatives — such as the European Processor Initiative (EPI) [50] and the large-scale pilots [5] — have been set up to augment the HPC capabilities of Europe and to reduce the dependence on non-European chips. These initiatives are managed by the EuroHPC Joint Undertaking (JU), which leverages funding and support from the European Commission and the member states. They are complemented by initiatives of the Chips JU aimed at processors for embedded systems. A key pillar of these initiatives is the reliance on the Open Hardware approach, embodied in the RISC-V instruction set architecture (ISA): an open source, versatile and customizable ISA developed by the RISC-V International Foundation.[1]

To follow up on the above-mentioned initiatives, the EuroHPC JU has set up a long-term initiative, the *Digital Autonomy with RISC-V in Europe* (DARE) Framework Partnership Agreement (FPA)[2], aiming at developing an HPC ecosystem based on RISC-V.[3]

DARE represents a groundbreaking European project with a budget of 240 million euros for its first phase, uniting 38 leading academic and industrial partners under the coordination of the Barcelona Supercomputing Center (BSC). The project builds upon previous foundations of consortia and plans to tape out three specific RISC-V-based chiplets optimized for advanced CMOS technology nodes: i) a Vector Accelerator (*VEC*) tailored for high-precision scientific simulations; ii) an AI Processing Unit (*AIPU*) specialized for low-power machine learning inference; and iii) a General-Purpose Processor (*GPP*) core dedicated to high-performance system orchestration.

The DARE Specific Grant Agreement 1 (SGA1) embodies the first phase of this program, which aims to design, test and evaluate several RISC-V-based solutions for VEC, AIPU and GPP. DARE SGA1 is still led by the Barcelona Supercomputing Center, with European companies OpenChip, Axelera, and Codasip respectively in charge of the three main solutions.[4]

SGA1 framework defines competitive performance and energy targets for its hardware chiplets.
For high-precision scientific computing, the VEC chiplet targets a dense core-level efficiency of 70÷75 DP GFLOPS/W and a peak performance 3÷4× greater than previous baselines, while achieving a sparse matrix efficiency of 2÷4.5%.

Being optimized for machine learning and Large Language Models, the Axelera AIPU chiplet relies on specialized in-memory computing to deliver up to 1,280 INT8 TOPS per chiplet at an extreme efficiency of 6,000 FP8-INT8 GFLOPS/W, scaling up to a dual-chiplet package that provides 2,560 INT8-INT8 TOPS to outperform commercial alternatives like the NVIDIA H100.
Finally, the GPP chiplet unifies processing structures to deliver high server orchestration efficiency, targeting a scalar core performance of 6÷8 SPECint2k6/GHz alongside a vector capability of 10 DP GFLOPS/W per core.

The Italian partnership in DARE. ICSC, the Italian National Research Centre for High-Performance Computing, Big Data and Quantum Computing, plays a pivotal role in coordinating the Italian contribution to DARE SGA1, collecting under its umbrella five Italian universities (Politecnico di Milano, Politecnico di Torino, Università degli Studi di Pisa, Università degli Studi di Roma "La Sapienza" and Università di Torino), as well as the National Institute for Nuclear Physics (INFN) and the graduate school SISSA. Another leading university, Alma Mater Università degli Studi di Bologna, the national supercomputing center CINECA and three companies — Axelera AI, Leonardo and E4 — round out the Italian contribution. The Italian activities in DARE SGA1 lean on the existing building blocks developed in earlier EuroHPC and Chips JU projects [5, 39, 12, 35] and strive to contribute to the pathfinding activities and to scout ahead of the technical developments of DARE SGA1 in sight of its second phase.

Synchronization and cooperation between the Italian consortium and project partners are maintained through a structured Technical Area (TA) framework illustrated in Figure 1 and implemented with periodic working groups and plenary meetings — in order to plan and assess joint activities — and unified hardware and software repositories for IP designs, verification tools, compiler layers and application benchmarks.

Organization of the paper. The rest of this paper is organized as follows. In Section 2, we provide an overview of the pathfinding activities carried out by the Italian partners of DARE SGA1 while additional details, current development status and preliminary achievements are reported in the following sections. In particular, in Section 3, we focus on accelerator designs, enabling technology for low latency scaling out of the accelerators mesh and reliability architecture, while in Section 4, we discuss the pathfinding activities related

[1] riscv.org
[2] DARE project website
[3] Advancing European Sovereignty in HPC with RISC-V
[4] DARE SGA1 announcement

to programming models and system software. Finally, in Section 5, we introduce key applications in Quantum Chemistry and Low-Rank Adaptation while in Section 6 we draw some conclusions and highlight future research directions.

2. Pathfinding Technologies for future RISC-V HPC

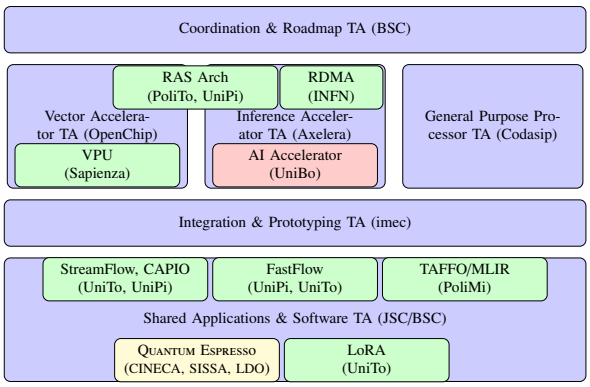


Figure 1: Structure of the DARE SGA1 project by technical areas (TA, in blue); for each TA, the leading partner is reported. The role of the Italian partners is depicted as follows: ICSC partners develop the components in green, and those in yellow are developed in collaboration with other Italian partners. In contrast, components in red are developed by Italian partners who do not operate under the ICSC umbrella.

The Italian consortium partners in DARE SGA1 contribute significantly to the project's technological evolution, focusing on advanced research that extends the capabilities of the underlying RISC-V ecosystem. As illustrated in Figure 1, the project is organized into six Technical Areas (TA).

The hardware foundation is built upon three distinct RISC-V implementations: a vector accelerator (led by OpenChip), an AI inference accelerator (led by Axelera AI) and a general-purpose processor core (GPP led by Codasip).

Coordination and infrastructure support are distributed among Imec (Integration and Prototyping), the Barcelona Supercomputing Center (Roadmap and Coordination) and a joint effort between BSC and the Julich Supercomputing Centre (Software Stack and Applications).

Hardware Development and Scalability Italian research activities are primarily directed towards the development of high-throughput, energy-efficient accelerators for low- and mixed-precision computation, both in the context of vector processing units (VPUs) and AI inference accelerators (AI accelerators), to enhance throughput and energy efficiency.

In addition, to bridge the gap between edge accelerators and High-Performance Computing (HPC) environments, the Italian partners are developing two critical architectural features: the *AI-Direct Remote Direct Memory Access* (RDMA) - a high-performance hardware/software communication primitive designed to enable the scaling of large-scale AI models across distributed nodes - and *Reliability, Availability and Serviceability architecture* (RAS), a comprehensive framework to ensure system-level resilience and fault tolerance in large-scale deployments.

On the software side, affiliates of the ICSC are providing a robust toolchain to support effective programming of future RISC-V based systems. This includes *StreamFlow*, a workflow manager for complex task orchestration, *CAPIO* and *libintercept* for optimization of inter-node I/O programming and *TAFFO*, a compiler-level precision-tuning framework.

Finally, the Italian partners are in charge of porting one of the key applications — Quantum ESPRESSO — and adapting some AI applications to the AI inference accelerator.

3. RISC-V Accelerators

3.1. AI accelerator

A fundamental enabler for next-generation AI systems is the integration of specialized hardware accelerators directly within the compute fabric. In particular, efficient support for matrix and tensor operations has become essential to sustain the computational intensity of modern deep learning models during both inference and training. While inference can often rely on aggressive quantization and low-precision arithmetic, advanced workloads such as fine-tuning, online learning and continual adaptation of Large Language Models (LLMs) still require solid floating-point support to ensure numerical stability and accuracy across heterogeneous datasets. Unlike conventional convolutional workloads, which are typically characterized by fixed weights and regular computation patterns that can be effectively optimized through quantization and pruning, LLM adaptation techniques such as LoRA involve the continuous update of model parameters through large-scale matrix multiplications and accumulation operations. These workloads exhibit a significantly higher sensitivity to numerical errors, since small perturbations introduced during weight updates can accumulate over time and negatively affect convergence and final model quality. As a result, the aggressive compression strategies commonly adopted for inference acceleration are

often ineffective or require costly retraining procedures to recover lost accuracy. In particular, low-bit quantization may substantially degrade gradient estimation and optimization dynamics during fine-tuning, making FP16 computation and higher-precision accumulation essential to preserve training robustness. This observation motivates the requirement for dedicated hardware support capable of efficiently sustaining floating-point matrix operations rather than relying exclusively on reduced-precision inference-oriented accelerators.

In this framework, Axelera AI plans to enhance the Artificial Intelligence Processing Unit (AIPU) architecture with dedicated capabilities targeting LLM fine-tuning and adaptive learning scenarios.

To achieve this objective, UNIBO will define and implement a RISC-V architectural extension specifically designed to accelerate matrix operations, leveraging recent advances in compact floating-point accelerators. The main design goal is to efficiently support FP16 and mixed-precision matrix multiplications while maintaining tight constraints on silicon area and energy consumption, which are crucial for edge and embedded high-performance deployments. The proposed unit will build upon the RedMulE architecture [75], showing how tightly coupled FP16 matrix engines can provide substantial improvements in performance density and energy efficiency within ultra-low-power SoCs. The extension will be integrated into the modular RISC-V ISA ecosystem, following open and extensible design principles to promote portability and broad adoption.

A systematic exploration of the matrix-oriented extensions proposed within the RISC-V landscape will also be conducted. This includes the evaluation of the standard Vector Extension (RVV), emerging tensor and matrix acceleration proposals, and tightly coupled coprocessor-style solutions. The goal is to identify the most effective balance among programmability, compiler support, hardware complexity and performance per watt. Particular attention will be devoted to mapping LLM-related linear algebra kernels onto vector-length-agnostic execution models while preserving compatibility with existing toolchains.

Furthermore, the interaction between the matrix extension and precision tuning strategies will be analyzed to maximize efficiency under constrained power and memory budgets. Mixed-precision schemes combining FP16 operands with higher-precision accumulators will be considered to guarantee convergence robustness during fine-tuning and adaptation.

As a preliminary validation step, a benchmarking activity has been carried out on a representative subset of a LoRA (Low-Rank Adaptation) workload, focusing on its most compute-intensive kernels. The results collected so far refer exclusively to the vector-based implementation exploiting the RISC-V Vector Extension. Compared to a baseline implementation running on a Single Program Multiple Data cluster [63], the vector approach achieves an average 3.6× performance improvement in absolute throughput and a 3.3× gain in energy efficiency across the selected kernels. These measurements provide a solid reference point for the ongoing architectural exploration. The same benchmarking methodology will be applied to the forthcoming matrix extension, enabling a direct and quantitative comparison between pure vector acceleration and dedicated matrix support in terms of performance, energy efficiency, and silicon cost.

3.2. Reduced precision VPU

Vector processing architectures have recently attracted renewed attention across high-performance computing (HPC) [58, 69], AI accelerators [67, 66] and edge computing [19, 20, 18]. In all these domains, performance scaling is increasingly limited by energy efficiency rather than peak computational throughput. As a consequence, reduced-precision floating-point arithmetic has emerged as a key architectural optimization, enabled by the intrinsic numerical tolerance of modern workloads [71].

Within this context, the DARE project "VEC pathfinding" stream of activities, coordinated by the Barcelona Supercomputing Center, is developing a cutting-edge Vector Processing Unit (VPU) that combines aggressive vector parallelism with native support for low-precision floating-point computation. Sapienza University contributes to this effort through the design and verification of a dedicated 8-bit floating-point (FP8) hardware IP, developed at the Digital System Laboratory. The proposed IP enables the VPU to sustain up to 128 FP8 operations per cycle, while ensuring strict compliance with both emerging IEEE standards and widely adopted industry specifications.

The FP8 IP from Sapienza will explicitly support the binary8 formats IEEE P3109 8p3se, IEEE P3109 8p4se and IEEE P3109 8p4ue, all defined as extended binary8 encodings with full support for infinities and NaNs [49]. The first two formats provide signed representations with different exponent widths, while IEEE P3109 8p4ue provides an unsigned extended encoding. In addition, the IP will support the OCP FP8 formats E4M3 and E5M2, which are widely deployed in industrial AI accelerators and adopt, respectively, finite or extended numerical ranges optimized for machine learning workloads [57, 62].

Table 1: Supported FP8 formats and numerical characteristics

Format	Exp. bits	Frac. bits	Bias	Max Finite Value	Min Subnormal
IEEE P3109 8p3se	5	2	16	49152	2^{-17}
IEEE P3109 8p4se	4	3	8	224	2^{-10}
IEEE P3109 8p4ue	5	3	16	53248	2^{-18}
OCP FP8 E4M3	4	3	7	448	2^{-9}
OCP FP8 E5M2	5	2	15	57344	2^{-16}

Table 1 summarizes the FP8 formats supported by the Sapienza IP, using the nomenclature and numerical definitions specified by the corresponding standards. The table reports the exact exponent and mantissa allocation, the associated exponent bias and numerical values for the maximum finite representable number and the minimum subnormal number. For IEEE P3109 formats and OCP E5M2, extended encodings provide support for infinities and NaNs, while OCP E4M3 format rely on finite-only representations.

The Sapienza FP8 IP will support the RISC-V "V" vector extension (version 1.0), implementing all core floating-point arithmetic operations, including addition, subtraction, multiplication, division, square-root and fused multiply-add (FMA).

The IP will also implement narrowing/widening operations from/to higher-precision 16-bit formats, as well as conversions to and from signed and unsigned 8-bit integer representations. All special cases, including NaNs, infinities where defined, subnormal numbers, and rounding behavior, will be handled in accordance with IEEE floating-point arithmetic rules. All five IEEE 754 rounding modes will be supported.

In order to enable efficient multi-format operation, FP8 operands are converted on-the-fly to a shared internal representation, while results are converted back to the target FP8 encoding. Runtime switching between IEEE P3109 and OCP FP8 formats will be supported with a single-cycle latency. The arithmetic units will be deeply pipelined, with the number of stages configurable at synthesis time, and designed for reuse across other designs.

A further architectural challenge arises from vector sliding and gathering operations. When operating on FP8 data, the increased element density amplifies pressure on the vector register file and associated interconnects, potentially impacting energy efficiency and area. Addressing these effects represents an active area of investigation and may constitute future work at Sapienza.

As with all floating-point architectures, extensive verification is a critical aspect of the design process. A dedicated simulation and validation environment will be developed in close collaboration with the Barcelona Supercomputing Center to ensure correctness, robustness, and full compliance with IEEE P3109 and OCP FP8 specifications when the IP is integrated into the complete VPU.

3.3. AI-Direct RDMA

Deployment of modern large-scale NN models and real-time generative AI requires HPC systems equipped with multiple AI accelerators. Such an envisioned environment, among other things, requires a direct, low-latency and high-bandwidth communication mechanism to efficiently feed the accelerators.

INFN is developing *AI-Direct*, a specialized hardware engine and companion software (Linux device driver, user library) to enable efficient, low-latency communication between RISC-V-based AI accelerators and boost inference performance for large-scale distributed neural network models.

The proposed specialized engine should be capable of Remote Direct Memory Access (RDMA), *i.e.* communication between distributed accelerators across separate network nodes without any host CPU intervention. Traditional Direct Memory Access (DMA) engines operate strictly inside a local memory domain and demand host operating system orchestration to move data into system space before network transit. Conversely, RDMA is a networking technology that enables direct data transfers between the main memories of different networked computers without the intervention of local CPUs, avoiding local buffer copies and achieving ultra-low latency and high-throughput data exchange. In this context, RDMA support integrated in AI-Direct enables peripheral AIPU accelerators to exchange data directly over commodity PCIe links and/or state-of-the-art interconnect technology. The expected performance benefit of this paradigm shift is a drastic reduction in multi-node communication latency during large-scale model synchronization steps. In addition, we also foresee exploring the adoption of CXL [30] standard and/or UALink [76] direct AI accelerators interconnect when available and mature enough.

INFN activities in DARE project involve the incremental development of an FPGA-based testbed designed to integrate and evaluate AI-Direct communication channels over PCIe. The testbed utilizes emulated AIPU traffic to exercise the development system and evaluate architectural solutions in a near-real environment. The first implementation phase provides a simplified FPGA-based demonstrator to measure critical performance metrics, specifically peer-to-peer (P2P) latency. The configuration utilizes a single PCIe domain where two FPGA boards are connected to a shared root complex hub.

The testbed was centered around a SuperMicro

X13SEI-F INTEL-based server featuring PCIe 5.0 lanes. Two AMD Alveo U280 FPGAs were employed as the communication endpoints.
The implementation leveraged the `verilog-pcie`[37] library to create a lightweight PCIe subsystem with Direct Memory Access (DMA) capabilities. Precise temporal measurements were handled by dedicated hardware logic on the FPGAs, while a minimal Linux-based software stack managed memory handles and hardware registers via a custom device driver and user-space IOCTL tools.
The experiment, as depicted in Figure 2 utilized a *Ping-Pong* synchronization mechanism to measure transaction speed. The workflow was structured as follows:

1. Address Exchange: The host system mapped the Base Address Registers (BARs) and facilitated the exchange of BAR4 physical addresses between FPGA0 and FPGA1.
2. Ping Phase: FPGA0 initiated a DMA write transaction targeting the BAR4 memory space of FPGA1.
3. Internal Copy: FPGA1 moved the received data from its BAR4 interface to its local DMA engine.
4. Pong Phase: FPGA1 executed a return DMA write to the BAR4 of FPGA0.

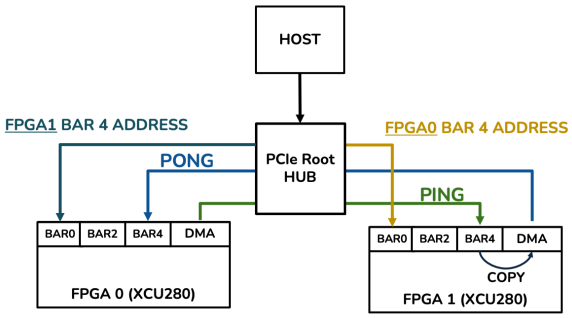


Figure 2: AI-Direct testbed for initial P2P latency measurement

The measurement results for a 32-byte data iteration showed a one-way latency of less than 600 ns.
This benchmark serves as the baseline for upcoming tests on the Alveo V80 platform, which is expected to reduce latency and increase throughput due to its 7 nm process and PCIe Gen5 capabilities.

The final demonstrator will leverage some proprietary Intellectual Properties developed in previous projects: APEnetX [9] (developed during the RED-SEA [12] project), FPGA-based Network Interface Cards (NIC) with PCIe interface addressing the workload generated by scientific applications (Spiking Neural Network simulations and High Energy Physics theoretical computations), and the APEIRON [8] framework (developed during the Textarossa [1, 35] project) aiming at offering hardware and software support for running real-time parallel dataflow applications on a mesh of interconnected FPGAs.

3.4. RAS architecture concept and solutions for error monitoring and management

HPC systems are based on the most advanced semiconductor technologies, which can guarantee the maximum level of performance while keeping power and energy consumption at an acceptable level. However, these technologies are known to be less reliable than the previous ones due to their higher susceptibility to transient and permanent faults [48, 70]. As a result, there is a growing concern, especially among users of large data centers, about the so-called Silent Data Corruption (SDC) effects [32, 46, 80], leading to incorrect results that escape existing fault detection/correction mechanisms. While there may be different root causes for SDCs, there is quite a consensus on the need for new devices for HPC systems to be equipped with a larger set of mechanisms able to monitor their health status, thus contributing to the definition of a consistent and effective RAS (Reliability, Availability, and Serviceability) architecture. Obviously, the health monitoring of SDC effects should complement the adoption for RAS of monitoring solutions also for other fault sources like aging, or transient errors due to electromagnetic interference or hit by ionizing particles. Furthermore, besides the error detection and monitoring phase, it is also important to implement error management solutions, so that even in case of faults, the HPC unit can guarantee availability and serviceability. This way, the DARE HPC units will also be suitable for mission-critical applications. The monitoring mechanisms should be able to address both transient and permanent faults. In the latter case, a significant issue is the lack of effective fault models able to catch the wide set of defects arising in the most advanced semiconductor technologies. Hence, mechanisms based on aging monitors should be complemented with other solutions, e.g., based on the availability of suitable functional tests to be triggered periodically or when required. Given the continuous evolution of underlying semiconductor technologies (and thus the involved defects), it is also crucial to equip future HPC systems with powerful mechanisms to monitor their internal activities and health status. This may allow the systematic collection of data required by possible post-processing resorting to AI-based technologies to identify critical situations early. The innovative contribution by Politecnico di Torino focuses on the error monitoring part, developing effective functional tests for advanced

HPC architectures, building on previous work targeting safety-critical systems (e.g., [45]).

As a result, the extensive experience at Politecnico di Torino in the development of Software-Based Self-Testing (SBST) solutions for Processors and Graphic Processing Units has led to a novel methodology for generating Software Test Libraries (STLs) aiming at the in-field testing of permanent faults that occur during the operation of vector accelerators. This testing process begins by exploring variations in input data from applications that naturally exercise the accelerator, while reusing some existing application code. The method then incorporates specific Ad Hoc routines to detect hard-to-identify faults.

Additionally, the proposed approach utilizes multi-threading to execute different test code segments concurrently. This technique enhances fault-detection capabilities while reducing the overall test execution time. To further address the challenges posed by the highly configurable Vector Processing Unit (VPU), we have created a framework called **PIPPO** (*Parameterizable In-field vector Processor test Program generatOr*). This framework automatically adapts the baseline STL to ensure high fault coverage across various hardware parameters and configurations.

More in detail, **PIPPO** implements two main stages: *i) Baseline STL generation*, *ii) STL customization*. The first stage creates the STL for the accelerator with the smallest possible configuration (Baseline) by leveraging common AI primitives applications that naturally exercise the computing units, *i.e.*, Functional Units, or FUs. During this stage, **PIPPO** creates a dataset of functional programs for the VPU that exhibit the highest capabilities for activating and propagating fault effects without requiring changes to their code structure. Such a set of programs is automatically generated by **PIPPO** by running an experimental exploration of the VPU's RTL abstraction of the accelerator using different software parameters (e.g., 32, 16, or 8-bit integers) input values, and sizes, for suitable benchmarks including matrix multiplication (MxM) and 1D/2D Convolution (Conv) applications. The Fault Coverage (FC) obtained from RTL Fault Simulation is used as a metric to assess the fault detection effectiveness of the code exploration stage.

This exploration is followed by a refinement process that utilizes the Gate-Level description of the baseline VPU and the candidate STLs identified in the previous stage. This refinement process begins with an initial fault simulation to identify any remaining undetected faults (UFs). Following this, Ad Hoc routines are introduced to target these undetected faults, which can be categorized as outlined in Table 2. For each category, the rightmost column specifies a strategy to test the corresponding faults, using custom code designed to activate any unused VPU units.

Table 2: Classification of Undetected Faults (UFs) and related test strategies.

Clasification	Cause	Test Strategy
Unused FU unit	The application does not exercise all FUs	Add instructions that explicitly target unused FUs
Unused Instructions	VPU instructions not present in the test code for all the threads using the VPU	Add instructions in the test with pseudo-random operands for all the threads
Unexplored Data Representation	The application uses only one data representation operation, e.g., 32-bits	Generate input vectors with different data-width configurations
Insufficient Data Diversity	Limited operand variation, e.g, fixed-shift amount	Create further datasets to increase operand variability

During STL customization, **PIPPO** selects baseline STLs that meet the desired coverage levels and generates the final STL based on the target hardware parameters and configuration, including the number of parallel cores and FUs or lanes. This process involves modifying the test input data and code, as well as replicating input vectors and instructions to uniformly exercise all functional units and lanes.

PIPPO was successfully applied and validated on the *Klessydra-T* VPU [20] in three hardware configurations (Baseline, or 'STD', Extended SIMD, or 'eSIMD', and Symmetric, or 'SYM'), aiming to assess the fault detection effectiveness under different VPU scenarios. In particular, several experiments were conducted to evaluate the test quality of the generated STLs across six VPU configurations, including varying numbers of FUs per lane and vector lanes. The results showed that PIPPO can generate STLs achieving more than 96% FC when considering stuck-at faults, with moderate testing time, and no area overhead. It is worth noting that the proposed solution fully avoids the adoption of ATPG tools for test pattern generation. Additionally, the proposed strategy and framework are particularly well-suited to resource-constrained systems. In these cases, strict design budgets prevent architectural hardware changes needed for Design for Test (DfT), such as Built-In Self-Test (BIST). More details about PIPPO can be found in [77].

The innovative contribution of University of Pisa to RAS moves the focus from testing and error analysis/detection to the error management part and to the propagation of information about errors from the hardware level to the software level. To this aim, University of Pisa is proposing an error management hardware IP macrocell, integrable with RISC-V scalar and vectorized ISA, able to classify, prioritize and transmit at

the software levels the errors detected both in memories (e.g. the Vector Register File of the vectorized RISC-V processing unit) and in the logic. The interface from hardware level to the software level is compliant with the recent standardized interface of RISC-V called RERI (RAS Error Record Register Interface). In DARE these solutions will be applied with a focus on the VEC (Vectorized RISC-V) computing unit. To this aim, in the project DARE will be extended the work done in the project EPI SGA2 [14].

Connected to the RAS system will be the Error Controller Subsystem module developed by Sapienza University. The Vector Register File (VRF) is a cornerstone of modern Vector Architectures, serving as the high-bandwidth storage for vector operands and holding the critical architectural state of the processor [11]. Given its significant die area and density, the VRF is highly susceptible to transient hardware faults, necessitating a robust protection mechanism to ensure data integrity. To address this, Sapienza proposes integrating a Single Error Correction, Double Error Detection (SECDED) architecture directly into the VRF memory hierarchy at every Lane. As depicted in Figure 3, they will design a memory wrapper that transparently encodes data during writes and checks them during reads, using a sliced data-path organization to support fine-grained, masked write operations without incurring any latency penalties.

Furthermore, to bridge the gap between the hundreds of parallel memory banks and the system's maintenance logic, they will implement a centralized Error Controller. This unit functions as an asynchronous bridge: it captures burst error events in a "pending" register matrix, arbitrates them by severity, prioritizing uncorrectable data corruption over correctable or parity faults, and serializes diagnostic packets into a circular queue. This decoupled architecture ensures that the high-frequency operation of the vector lanes never bottlenecks the RAS logging interface.

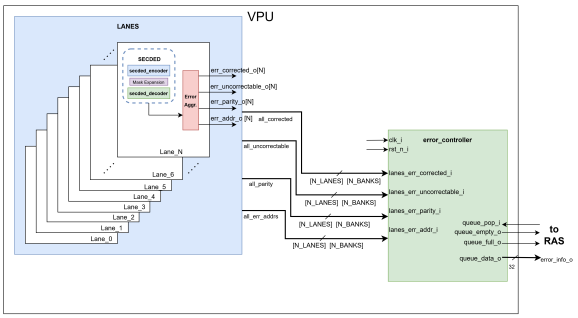


Figure 3: Error Controller block overview

4. Software Stack

4.1. Workflow Management

Workflow models are widely used to design and orchestrate large-scale scientific applications. Several approaches to workflow management exist on the market, exposing different programming interfaces (from low-level task-based computing libraries to high-end graphical user interfaces) and targeting a diverse set of execution architectures (from single machines to high-performance computing facilities to the cloud) [73, 31, 28]. While workflow systems continuously come and go, languages and standards are fated to stay. The Common Workflow Language (CWL) [29] is an open standard for describing command line tools and connecting them to create workflows. The CWL standard is fully declarative: it does not require workflow designers to rework the applications' business code. It counts more than ten independent but interoperable implementations. Among those, StreamFlow [27, 26] leverages the *hybrid workflow* paradigm to orchestrate a CWL workflow on top of heterogeneous execution infrastructures (e.g., cross-facility, multi-cloud, or classical-quantum). The same theoretical approach lies the foundation of Jupyter Workflow [25], a Jupyter extension that transforms a sequential notebook into a distributed workflow graph.

In the DARE SGA1 project, Università di Torino (UniTO) leads a task that extends scientific workflow technologies for the RISC-V standard. Building on prior expertise [3], UniTO will enhance CWL to support RISC-V modularity and update the StreamFlow system accordingly. UniTO and BSC will also integrate Jupyter Workflow with the PyCOMPSs Jupyter interface to enable large-scale task-based programming on heterogeneous RISC-V platforms [25].
As a first result of the DARE project, the StreamFlow ecosystem can now be installed on RISC-V machines, with all native Python dependencies successfully compiled. However, performance remains significantly lower than on more mature architectures such as Intel x86_64.

The performance of a scientific workflow depends primarily on the execution time of the tasks along its critical path, which varies across workflow instances, as well as on the overhead introduced by the workflow system's control plane. In StreamFlow, this overhead is largely determined by I/O operations on intermediate data, including file existence checks, content reading and writing, and checksum computation.

Table 3: Performance of StreamFlow benchmarks on different hardware architectures.

Benchmark	x86_64 (us)	RISC-V (us)
`exists`	11.12 ± 0.27	445 ± 14
`read`	26.67 ± 0.45	927 ± 16
`checksum`	538.19 ± 65.26	7322 ± 328

Table 3 compares the performance of an Intel i7-13700H processor (12 cores, 4.7 GHz) with that of a 64-bit SiFive Essential U74-MC processor (4 cores, 1.2 GHz) across three I/O operations: an `exists` call (which queries file system metadata), a `read` operation on a 100 kB file, and a `checksum` computation on the same file (a compute-bound task). The benchmark refers to StreamFlow 0.2.0.dev14 backed by Python 3.10, and each measurement reports the average of ten rounds of 100 iterations, preceded by one warm-up round. All three operations exhibit the same trend, executing tens of times faster on the Intel processor. The difference in clock frequency alone does not account for such a large performance gap, as evidenced by the fact that the compute-bound `checksum` operation is the least affected. Instead, the gap is mainly attributable to the greater maturity of the Intel hardware and software ecosystem compared to that of RISC-V. Addressing this gap will be a key objective of the remaining DARE project activities.

In DARE, workflows will also benefit from Cross-Application Programmable I/O (CAPIO), *i.e.*, a middleware designed to accelerate scientific workflows by turning their traditional, file-based step-by-step execution into a hybrid model that combines in-memory storage and transparent streaming [55].
It provides developers with a) an in-memory file system for staging intermediate data without touching the parallel/distributed file system and b) transparent streaming under POSIX semantics, so files become multi-producer/multi-consumer data streams. Using its coordination language (CAPIO-CL) [65], developers can annotate which file dependencies should be streamed. At runtime, CAPIO enforces the streaming semantics associated with annotations to automatically optimize end-to-end workflow performance while preserving the correct execution order.

CAPIO transparency is based on system call interception, a technique where a program captures and possibly modifies system calls made by other programs. System calls are how applications request services from the OS (e.g., opening a file or spawning a process). By intercepting these calls, a tool can monitor, modify, or control application behavior: CAPIO transparently substitutes the file system with a node-to-node streaming system. System call interception requires a system call intercept library; CAPIO includes a system call interception library, called `vpoline`, which is the first exhaustive interception library available for RISC-V [60]. The library is already available open source under the Apache 2.0 license,[5] and will be further optimized in the DARE project.

4.2. Programming Models

FastFlow is a high-level parallel programming model and framework implemented in modern C++. It provides developers with essential streaming-based abstractions for multicore systems [4]. With its recent distributed extension [74], a FastFlow streaming graph can be arbitrarily partitioned into independent processes. Inter-process communication is managed by the Multi-Transport Communication Library (MTCL) [36], which supports MPI, UCX, sockets, and other messaging layers.
At its heart, FastFlow offers a pattern-based (a.k.a. skeletons-based) approach to parallelism. Skeletons, such as map, reduce, and pipeline are due to the seminal work of Cole [24]. After long experimenting in research environments [7], the approach became mainstream with Google's MapReduce and Intel TBB at the beginning of the 2000s. All these frameworks allow programmers to compose parallel programs from prebuilt patterns instead of dealing with low-level thread management. Fastflow, differently from other frameworks, targets shared and distributed memory with a uniform approach based on streaming
FastFlow's architecture-agnostic design (pure header-only C++ and portable concurrency primitives) makes it particularly suitable for experimenting with the runtime support of complex AI frameworks on RISC-V. In 2023, Fastflow enabled implementing a Yolo5-based AI application on RISC-V (and the first publicly available porting of the PyTorch ML framework to RISC-V) [59].

In DARE, we will exploit FastFlow to experiment with the TVM (short for Tensor Virtual Machine) deep learning framework. TVM is designed to optimize and compile machine learning models so they can run efficiently on various hardware backends (CPUs, GPUs, specialized accelerators, etc.). TVM is one of the leading AI frameworks of the DARE project. Instead of just running models in a fixed framework (like TensorFlow or PyTorch), TVM compiles them into highly optimized, low-level code customized for the target device.

[5] https://github.com/alpha-unito/vpoline

TVM itself is primarily a single-node compiler. Today, TVM does not directly support distributed training (like Horovod, TensorFlow Distributed, or Ray Serve).

TVM converts a pre-trained model to its Intermediate Representation (Relay IR), which is then "lowered" to target-specific code, e.g., CUDA for NVidia GPUs, LLVM for CPUs, or a C/C++ source code. The latter case can be exploited to generate FastFlow code supporting distributed data and model parallelism.

4.3. Compiler Technology

The usability of computing platforms depends closely on the availability of effective means to deploy applications – thus, compilers are critical. Functional portability on accelerators based on the RISC-V ISA is relatively easy to achieve, as long as standard ISA extensions are employed, but performance portability is limited by the need for microarchitecture- and ISA extension-specific optimizations [72]. Precision tuning [22, 23] can improve the granularity of quantization, enabling fully exploiting low-precision vector units, particularly in AI workloads where operations with different precisions might be needed, such as spiking neural networks [53]. While early effort towards efficient deployment of AI workloads relied on custom tools, the field is rapidly maturing towards infrastructures based on the Multi-Level Intermediate Representation (MLIR) compiler framework [51], which significantly simplifies the development of domain-specific languages and compilers. Currently, several Neural Network deployment frameworks such as PyTorch and Tensorflow have MLIR back-ends enabling optimization and code generation for a variety of platforms [10]. It is worth noting that the most recent Fortran compiler, Flang, is also based on MLIR [13], which enables the reuse of code analysis and transformations also on high performance codes.

Politecnico di Milano contributes to DARE SGA1 by addressing the precision tuning challenge through the design and implementation of an MLIR dialect and its lowering transformations, which aim at fine-grained precision tuning. Based on our previous expertise with the TAFFO framework [16, 17, 38], which automates precision tuning on the lower-level LLVM intermediate representation, our team develops TAFFO-MLIR [40], including a hybrid Value-Range Analysis [6] combining classical interval arithmetics with perturbed affine forms [44]. The current implementation takes as input C code, thanks to a pass that transforms the MLIR representation produced by the Polygeist parser [61] to our TAFFO-specific dialect. Initial micro-benchmarks show major tightening of the conservative approximation of the value range when correlation and cancellation effects arise.

Furthermore, the interaction between precision tuning and other optimizations has been critical for achieving high performance in parallel computing [54]. In DARE, the Politecnico di Milano unit will explore the interactions with vectorization, which promises the most significant advantages in HPC and HPC-AI workloads.

5. HPC Applications

The efforts described above to establish a comprehensive software stack for the DARE architectures are complemented here by a set of representative high-level applications targeting efficient execution on RISC-V-based DARE systems. These applications provide the context in which the components of the software stack are integrated, exercised, and evaluated, and they define the primary usage scenarios considered in the project. In the DARE project, they span a range of scientific and technological domains, including climate and weather modelling (e.g., IFS), geophysics and geosciences (e.g., NEMO, ALYA-pyLOM), molecular dynamics (e.g., GROMACS), materials science (e.g., Quantum ESPRESSO), and data-driven and machine learning workloads (e.g., Content detection and LoRA-based applications). These examples are representative of the broader set of applications considered within DARE and will be refined as the project evolves. Within this framework, applications are ported to DARE platforms and used to assess performance, portability, and scalability on DARE architectures. They also serve as a basis for hardware–software co-design, where insights from application behaviour inform both software optimisation and architectural choices. A central aspect of this activity is the identification and analysis of the dominant computational kernels, which account for a large fraction of execution time. These kernels are extracted, instrumented, and profiled in order to study performance characteristics across different platforms. The overall methodology combines full-application studies with reduced test cases and miniapplications. While complete workflows are used to assess end-to-end behaviour and numerical performance, miniapps enable controlled and fine-grained analyses of specific computational patterns, including memory access, communication, and vectorisation. In the following, we describe in more detail two applications developed and maintained by the Italian DARE partners: Quantum ESPRESSO and Low-Rank Adaptation (LoRA). For

each case, we summarise the application domain, the implementation and porting activities, and the main results obtained so far.

5.1. Quantum ESPRESSO

In the domain of electronic structure and quantum chemistry—major consumers of HPC resources—DARE includes Quantum ESPRESSO (QE) as a representative application. QE is a widely used open-source suite for electronic-structure calculations and materials modelling [41, 42]. Its plane-wave formulation provides systematic convergence and natural parallelisation, making it well suited for modern HPC systems. Its modular structure and extensive user base make it a relevant platform for performance evaluation and code development. Integration of QE into DARE builds upon ongoing efforts to improve modularity and portability [43, 15]. The software adopts a layered architecture (Figure 4), separating high-level simulation components from performance-critical numerical libraries. This structure allows architecture-dependent optimisations to be confined to lower layers, while preserving a consistent interface at the application level.

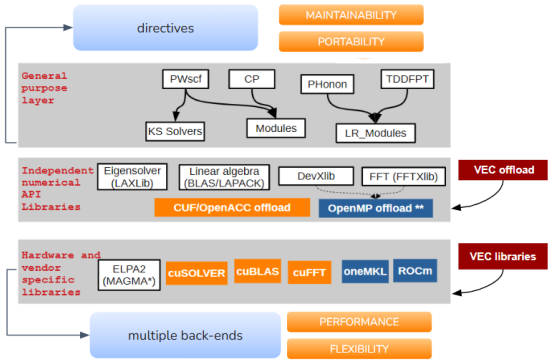


Figure 4: Layered structure of the Quantum ESPRESSO suite. The upper layer comprises simulation engines and property calculators, while the lower layer contains optimised numerical libraries and architecture-specific backends.

QE supports multiple programming models and backend libraries selected at compile time depending on the target platform [64]. Parallelisation is achieved through a combination of MPI and shared-memory threading, enabling scalability across nodes and efficient use of computational resources [56].

5.1.1. Work and results

QE Miniapps on RISCV. To enable detailed performance analysis, a set of QE miniapps has been developed, targeting the most computationally intensive components of the PWSCF engine, in particular FFT distribution and diagonalisation routines implemented in `FFTXlib` and `KS_solvers`. These miniapps reproduce the numerical behaviour of the corresponding parts of the full application while allowing controlled variation of key parameters and execution environments. They have been used to study performance on emulators, FPGA-based systems, and HPC platforms, including DARE prototypes and RISC-V-based configurations supporting RVV1.0, as well as emulation frameworks [79]. The main miniapps include:

- `fftxlib_wfcs`, modelling FFT distribution and communication;
- `cb_miniapp`, targeting iterative diagonalisation.

Instrumentation with tools such as PAPI and RAVE APIs enables detailed profiling of computational kernels. Results highlight the importance of optimised numerical libraries for achieving efficient vectorisation and overall performance, particularly for emerging RISC-V vector architectures [78].

Industrial Workflow Demonstrator. An application workflow based on QE is also considered to evaluate performance in a more realistic setting. The workflow focuses on adsorption energy calculations on metallic surfaces using plane-wave DFT with PAW datasets. The complexity of the system is progressively increased, allowing systematic studies with respect to system size, basis-set parameters, and sampling. The workflow exercises key aspects of the application, including iterative solvers, hybrid parallelism, and I/O. These experiments complement kernel-level analyses by providing insight into application-level behaviour and the interaction between different computational components.

5.2. Low-Rank Adaptation (LoRA)

LoRA is a technique originally proposed to fine-tune large LLM transformers efficiently and cheaply by only adjusting a small number of parameters instead of the whole model [47]. In transformer models, weight matrices (like the ones in attention layers) are very large. Instead of fine-tuning the full matrices, the LoRA approach freezes them and adds small trainable low-rank matrices that "nudge" the output. These small matrices have far fewer parameters and can be trained much faster, using less memory. The approach has become a de-facto standard for fine-tuning huge multimodal transformer architectures [21]. DARE will aim at advancing the state of the art in the exploitation of LoRA in the context of experimental/small RISC-V

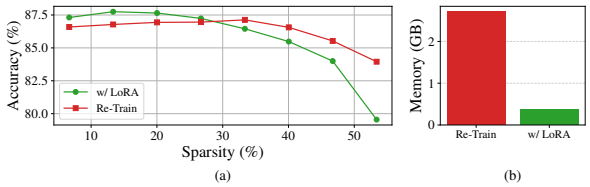


Figure 5: (a) Accuracy under increasing sparsity levels for different training strategies with a ViT-Tiny model trained on Tiny-ImageNet. Training mask-specific LoRA adapters achieves performance comparable to fully re-training at different pruning levels, (b) while significantly reducing memory requirements.

processors, by combining quantization and pruning techniques to get lightweight fine-tuned models. The LoRA module will be part of the compiler toolchain of Axelera AI.

5.2.1. Preliminary Results

Besides allowing for a more efficient fine-tuning of large models, the use of these simple adapter modules has several indirect yet highly significant advantages. By simply swapping adapters, the behavior of a single pre-trained model can be changed on-the-fly to accommodate, for instance, different tasks. This not only substantially increases model flexibility but also allows for a considerable reduction in memory usage [68]. Indeed, only a single base model, along with a collection of lightweight adapters, is needed to handle multiple tasks.

Together, these characteristics make adapters a suitable solution for a dynamic and efficient deployment of large models on edge devices or resource-constrained environments. In particular, a single backbone model can switch between multiple pruning masks, thus allowing dynamic control of model complexity during inference. Adapters here compensate for the performance degradation induced by pruning without requiring full retraining of the underlying model. As a result, the backbone remains shared across all sparsity levels, while mask-specific adapters restore task performance. This strategy allows the model to easily switch between computational budgets during inference.

Figure 5 shows the results of this experiment on a ViT-Tiny [33] model trained on Tiny-ImageNet [52]. The proposed approach achieves performance comparable to independently fine-tuning a separate model for each pruning mask, while requiring significantly less memory.

6. Conclusions

In this paper, we introduced the contribution of the DARE Italian partners, mainly under the umbrella of the ICSC Foundation, to the pathfinding activities of the DARE SGA1 project. Following up on previous European multi-partner and pilot deployments aimed at future Exascale computing architectures, such as EUPilot, EUPEX, TEXTAROSSA, ACROSS, ADMIRE and RED SEA [5, 2, 35, 12], the Italian partners developed activities towards scale-up of RISC-V-based processors and accelerators for HPC and HPC-AI applications.

Considerable pathfinding progress has been achieved across all targeted layers of the hardware, system software, and application co-design stacks.
In particular, regarding the hardware and communication testbeds, the consortium validated new FP16, mixed-precision matrix acceleration processor and and FP8 IP extensions that significantly improve computing speed and energy efficiency and designed an FPGA testbeds to evaluate AI-Direct solution for low-latency communication between distributed multi-AIPU inference system.
Furthermore, the PIPPO framework was deployed to generate test libraries for the VPU, while SECDED wrappers were integrated into the VRF and coupled with the RERI interface to safeguard against faults.
On the software stack and optimizations front, StreamFlow was natively compiled on the architecture, utilizing CAPIO, FastFlow, and TAFFO-MLIR to optimize system performance and close the maturity gap.
Finally, for application integration, the design loop was closed by extracting heavy computing routines from QUANTUM ESPRESSO into standalone miniapps like fftxlib_wfcs and cb_miniapp, allowing for streamlined profiling across the ecosystem.

Future work will focus on finalizing activities and planning the integration of our contributions in DARE architectures to contribute to a sustainable, architecture-agnostic baseline,toward a fully independent and digitally autonomous European supercomputing supply

Acknowledgement

The DARE SGA1 project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101202459. The JU receives support from the European Union's Horizon Europe research and innovation programme and Spain, Germany, Czechia, Italy, Netherlands, Belgium, Finland, Greece, Croatia, Portugal, Poland,

Sweden, France, and Austria. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or of the granting authority. Neither the European Union nor the granting authority can be held responsible for them.
Pierpaolo Perticaroli is a PhD student enrolled in the National PhD program in Artificial Intelligence, XXXIX cycle, course on Health and life sciences, organized by Università Campus Bio-Medico di Roma.

During the preparation of this work, the authors used Google Gemini Pro to revise and improve language. The authors reviewed and edited the output as needed and take full responsibility for the content of the published article.

References

[1] G. Agosta, M. Aldinucci, C. Alvarez, R. Ammendola, Y. Arfat, O. Beaumont, M. Bernaschi, A. Biagioni, T. Boccali, B. Bramas, C. Brandolese, B. Cantalupo, M. Carrozzo, D. Cattaneo, A. Celestini, M. Celino, I. Colonnelli, P. Cretaro, P. D'Ambra, M. Danelutto, R. Esposito, L. Eyraud-Dubois, A. Filgueras, W. Fornaciari, O. Frezza, A. Galimberti, F. Giacomini, B. Goglin, D. Gregori, A. Guermouche, F. Iannone, M. Kulczewski, F. Lo Cicero, A. Lonardo, A. R. Martinelli, M. Martinelli, X. Martorell, G. Massari, S. Montangero, G. Mittone, R. Namyst, A. Oleksiak, P. Palazzari, P. S. Paolucci, F. Reghenzani, C. Rossi, S. Saponara, F. Simula, F. Terraneo, S. Thibault, M. Torquati, M. Turisini, P. Vicini, M. Vidal, D. Zoni, and G. Zummo. Towards extreme scale technologies and accelerators for eurohpc hw/sw supercomputing applications for exascale: The TEXTAROSSA approach. *Microprocessors and Microsystems*, 95:104679, 2022. [Online].

[2] G. Agosta et al. Textarossa: Towards extreme scale technologies and accelerators for EuroHPC hw/sw supercomputing applications for exascale. In *2021 24th Euromicro Conference on Digital System Design (DSD)*, pages 286–294. IEEE, 2021. [Online].

[3] M. Aldinucci, E. M. Baralis, V. Cardellini, I. Colonnelli, M. Danelutto, S. Decherchi, G. D. Modica, L. Ferrucci, M. Gribaudo, F. Iannone, M. Lapegna, D. Medic, G. Muscianisi, F. Righetti, E. Sciacca, N. Tonellotto, M. Tortonesi, P. Trunfio, and T. Vardanega. A systematic mapping study of Italian research on workflows. In *Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis, SC-W 2023, Denver, CO, USA, November 12-17, 2023*, pages 2065–2076. ACM, 2023.

[4] M. Aldinucci, M. Danelutto, P. Kilpatrick, and M. Torquati. *Fastflow: High-Level and Efficient Streaming on Multicore*, chapter 13, pages 261–280. John Wiley & Sons, Ltd, 2017.

[5] M. Aldinucci et al. The italian research on HPC key technologies across EuroHPC. In *18th ACM Int'l Conf on Computing Frontiers*, page 178–184, 2021. [Online].

[6] M. Allahakbari. Bringing perturbed affine arithmetic to MLIR. Master's thesis, Politecnico di Milano, 2025. [Online].

[7] V. Amaral, B. Norberto, M. Goulão, M. Aldinucci, S. Benkner, A. Bracciali, P. Carreira, E. Celms, L. Correia, C. Grelck, H. Karatza, C. Kessler, P. Kilpatrick, H. Martiniano, I. Mavridis, S. Pllana, A. Respício, J. Simão, L. Veiga, and A. Visa. Programming languages for data-intensive hpc applications: A systematic mapping study. *Parallel Computing*, page 102584, 2020.

[8] R. Ammendola, A. Biagioni, C. Chiarini, A. Ciardiello, P. Cretaro, O. Frezza, F. Lo Cicero, A. Lonardo, M. Martinelli, P. S. Paolucci, L. Pontisso, F. Simula, C. Rossi, M. Turisini, and P. Vicini. Apeiron: A framework for high level programming of dataflow applications on multi-fpga systems. *EPJ Web of Conf.*, 295:11002, 2024.

[9] R. Ammendola, A. Biagioni, C. Chiarini, P. Cretaro, O. Frezza, F. L. Cicero, A. Lonardo, M. Martinelli, P. S. Paolucci, E. Pastorelli, L. Pontisso, C. Rossit, F. Simula, and P. Vicini. APEnetX: A Custom NIC for Cluster Interconnects. In *2024 27th Euromicro Conference on Digital System Design (DSD)*, pages 90–97, 2024.

[10] G. Bao et al. UFront: Toward A Unified MLIR Frontend for Deep Learning. In *39th IEEE/ACM Int'l Conf on Automated Software Engineering*, pages 255–267, 2024.

[11] M. Barbirotta, F. Minervini, C. R. Morales, A. Cristal, O. Unsal, and M. Olivieri. Enhancing

Fault Tolerance in High-Performance Computing: A Real Hardware Case Study on a RISC-V Vector Processing Unit. *IEEE Open Journal of the Computer Society*, 5:553–565, 2024.

[12] A. Biagioni et al. Red-sea: Network solution for exascale architectures. In *2022 25th Euromicro Conference on Digital System Design (DSD)*, pages 712–719, 2022.

[13] N. Brown. Fully integrating the flang fortran compiler with standard mlir. In *SC24-W: Workshops of the Int'l Conf for HPC, Networking, Storage and Analysis*, pages 939–949. IEEE, 2024.

[14] N. Canino, S. DiMatteo, D. Rossi, and S. Saponara. Hw-sw interface design and implementation for error logging and reporting for ras improvement. *IEEE ACCESS*, 12:1–21, Dec. 2024.

[15] I. Carnimeo, F. Affinito, S. Baroni, O. Baseggio, L. Bellentani, R. Bertossa, P. D. Delugas, F. F. Ruffino, S. Orlandini, F. Spiga, and P. Giannozzi. Quantum ESPRESSO: One Further Step toward the Exascale. *Journal of Chemical Theory and Computation*, 19(20):6992–7006, 2023. PMID: 37523670.

[16] D. Cattaneo et al. Architecture-aware precision tuning with multiple number representation systems. In *58th ACM/IEEE Design Automation Conference (DAC)*, pages 673–678. IEEE, 2021. [Online].

[17] D. Cattaneo et al. TAFFO: The compiler-based precision tuner. *SoftwareX*, 20:101238, 2022. [Online].

[18] M. Cavalcante, F. Schuiki, F. Zaruba, M. Schaffner, and L. Benini. Ara: A 1-GHz+ Scalable and Energy-Efficient RISC-V Vector Processor With Multiprecision Floating-Point Support in 22-nm FD-SOI. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(2):530–543, 2020.

[19] V. N. Chander and K. Varghese. A soft RISC-V Vector processor for Edge-AI. In *2022 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID)*, pages 263–268, 2022.

[20] A. Cheikh, S. Sordillo, A. Mastrandrea, F. Menichelli, G. Scotti, and M. Olivieri. Klessydra-T: Designing vector coprocessors for multithreaded edge-computing cores. *IEEE Micro*, 41(2):64–71, 2021.

[21] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[22] S. Cherubin and G. Agosta. Tools for reduced precision computation: a survey. *ACM Computing Surveys (CSUR)*, 53(2):1–35, 2020. [Online].

[23] S. Cherubin et al. Implications of reduced-precision computations in hpc: Performance, energy and error. In *Parallel computing is everywhere (PARCO)*, pages 297–306. IOS Press, 2018. [Online].

[24] M. Cole. *Algorithmic Skeletons: Structured Management of Parallel Computations*. Research Monographs in Par. and Distrib. Computing. Pitman, 1989.

[25] I. Colonnelli, M. Aldinucci, B. Cantalupo, L. Padovani, S. Rabellino, C. Spampinato, R. Morelli, R. Di Carlo, N. Magini, and C. Cavazzoni. Distributed workflows with Jupyter. *Future Generation Computer Systems*, 128:282–298, 2022.

[26] I. Colonnelli, B. Cantalupo, R. Esposito, M. Pennisi, C. Spampinato, and M. Aldinucci. 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, January 19, Budapest, Hungary*, volume 88 of *OASIcs*, pages 5:1–5:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[27] I. Colonnelli, B. Cantalupo, I. Merelli, and M. Aldinucci. StreamFlow: cross-breeding cloud with HPC. *IEEE Trans. Emerg. Top. Comput.*, 9(4):1723–1737, 2021.

[28] T. G. Community. The Galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update. *Nucleic Acids Research*, 52(W1):W83–W94, 05 2024.

[29] M. R. Crusoe, S. Abeln, A. Iosup, P. Amstutz, J. Chilton, et al. Methods included: Standardizing computational reuse and portability with the Common Workflow Language. *Communication of the ACM*, 2022.

[30] D. Das Sharma, R. Blankenship, and D. Berger. An introduction to the compute express link (cxl) interconnect. *ACM Computing Surveys*, 56(11):1–37, 2024.

[31] E. Deelman, K. Vahi, M. Rynge, R. Mayani, R. F. da Silva, G. Papadimitriou, and M. Livny. The evolution of the Pegasus workflow management software. *Comput. Sci. Eng.*, 21(4):22–36, 2019.

[32] H. D. Dixit, S. Pendharkar, M. Beadon, C. Mason, T. Chakravarthy, B. Muthiah, and S. Sankar. Silent data corruptions at scale. *CoRR*, dblp computer science bibliography abs/2102.11245, 2021.

[33] A. Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[34] European Parliament et al. *Key enabling technologies for Europe's technological sovereignty*. European Parliament, 2021. [Online].

[35] A. Filgueras, G. Agosta, M. Aldinucci, C. Alvarez, P. D'Ambra, M. Bernaschi, A. Biagioni, D. Cattaneo, A. Celestini, M. Celino, C. Chiarini, F. Lo Cicero, P. Cretaro, W. Fornaciari, O. Frezza, A. Galimberti, F. Giacomini, J. M. de Haro Ruiz, F. Iannone, D. Jaschke, D. JimÃ©nez-GonzÃ¡lez, M. Kulczewski, A. Leva, A. Lonardo, M. Martinelli, X. Martorell, S. Montangero, L. Morais, A. Oleksiak, P. Palazzari, L. Pontisso, F. Reghenzani, C. Rossi, S. Saponara, C. S. Lodi, F. Simula, F. Terraneo, P. Vicini, M. Vidal, D. Zoni, and G. Zummo. The TEXTAROSSA project: Cool all the way down to the hardware. *Microprocessors and Microsystems*, 122:105254, 2026. [Online].

[36] F. Finocchio, N. Tonci, and M. Torquati. MTCL: A multi-transport communication library. In e. a. Zeinalipour, editor, *Euro-Par 2023: Parallel Processing Workshops*, pages 55–67, Cham, 2024. Springer Nature Switzerland.

[37] A. Forencich, A. C. Snoeren, G. Porter, and G. Papen. Corundum: An open-source 100-Gbps NIC. In *28th IEEE International Symposium on Field-Programmable Custom Computing Machines*, 2020. `https://github.com/corundum/verilog-pcie`.

[38] W. Fornaciari et al. Hardware and software support for mixed precision computing: a roadmap for embedded and HPC systems. In *2023 Design, Automation & Test in Europe (DATE)*, pages 1–6. IEEE, 2023. [Online].

[39] W. Fornaciari et al. RISC-V processor technologies for aerospace applications in the ISOLDE project. In *Int'l Conf on Embedded Computer Systems*, pages 363–378. Springer, 2023. [Online].

[40] P. Galli. MLIR dialect for precision tuning. Master's thesis, Politecnico di Milano, 2024. [Online].

[41] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. Buongiorno Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. Dal Corso, S. de Gironcoli, P. Delugas, R. A. DiStasio, A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Küçükbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N. L. Nguyen, H.-V. Nguyen, A. Otero-de-la Roza, L. Paulatto, S. Poncé, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A. P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, and S. Baroni. Advanced capabilities for materials modelling with quantum espresso. *Journal of Physics: Condensed Matter*, 29(46):465901, Oct. 2017.

[42] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch. Quantum espresso: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter*, 21(39):395502, Sept. 2009.

[43] P. Giannozzi, O. Baseggio, P. Bonfà, D. Brunato, R. Car, I. Carnimeo, C. Cavazzoni, S. de Gironcoli, P. Delugas, F. Ferrari Ruffino, A. Ferretti, N. Marzari, I. Timrov, A. Urru, and S. Baroni. Quantum espresso toward the exascale. *The*

Journal of Chemical Physics, 152(15):154105, 04 2020.

[44] E. Goubault and S. Putot. Static analysis of numerical algorithms. In *Static Analysis*, pages 18–34, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[45] J.-D. Guerrero-Balaguera, J. E. R. Condia, and M. Sonza Reorda. STLs for GPUs: Using High-Level Language Approaches. *IEEE Design & Test*, 40(4):51–60, 2023.

[46] P. H. Hochschild, P. Turner, J. C. Mogul, R. Govindaraju, P. Ranganathan, D. E. Culler, and A. Vahdat. Cores that don't count. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, page 9–16, 2021.

[47] E. J. Hu et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. [Online].

[48] IEEE. International roadmap for devices and systems (irds™) 2022 edition. In *Institute of Electrical and Electronics Engineers (IEEE)*, 2022.

[49] IEEE Standards Association. IEEE P3109: Standard for Floating-Point Arithmetic for Machine Learning. [Online], 2023. Draft Standard.

[50] M. Kovač et al. How europe is preparing its core solution for exascale machines and a global, sovereign, advanced computing platform. *Mathematical and computational applications*, 25(3):46, 2020.

[51] C. Lattner, M. Amini, U. Bondhugula, A. Cohen, A. Davis, J. Pienaar, R. Riddle, T. Shpeisman, N. Vasilache, and O. Zinenko. Mlir: A compiler infrastructure for the end of moore's law. *arXiv preprint arXiv:2002.11054*, 2020.

[52] Y. Le and X. S. Yang. Tiny imagenet visual recognition challenge, 2015.

[53] H. W. Lui and E. Neftci. Hessian aware quantization of spiking neural networks. In *International Conference on Neuromorphic Systems 2021*, ICONS 2021, New York, NY, USA, 2021. Association for Computing Machinery.

[54] G. Magnani et al. SeTHet-Sending Tuned numbers over DMA onto Heterogeneous clusters: an automated precision tuning story. In *Proceedings of the 21st ACM International Conference on Computing Frontiers*, pages 258–266, 2024. [Online].

[55] A. R. Martinelli, M. Torquati, M. Aldinucci, I. Colonnelli, and B. Cantalupo. CAPIO: a middleware for transparent i/o streaming in data-intensive workflows. In *2023 IEEE 30th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, Goa, India, 2023. IEEE.

[56] Benchmark repository of MaX. [Online].

[57] P. Micikevicius, D. Stosic, N. Burgess, M. Cornea, P. Dubey, R. Grisenthwaite, S. Ha, A. Heinecke, P. Judd, J. Kamalu, N. Mellempudi, S. Oberman, M. Shoeybi, M. Siu, and H. Wu. FP8 formats for deep learning, 2022.

[58] F. Minervini, O. Palomar, O. Unsal, E. Reggiani, J. Quiroga, J. Marimon, C. Rojas, R. Figueras, A. Ruiz, A. Gonzalez, J. Mendoza, I. Vargas, C. Hernandez, J. Cabre, L. Khoirunisya, M. Bouhali, J. Pavon, F. Moll, M. Olivieri, M. Kovac, M. Kovac, L. Dragic, M. Valero, and A. Cristal. Vitruvius+: An area-efficient RISC-V decoupled vector coprocessor for high performance computing applications. *ACM Trans. Archit. Code Optim.*, 20(2), Mar. 2023.

[59] G. Mittone, N. Tonci, R. Birke, I. Colonnelli, D. Medić, A. Bartolini, R. Esposito, E. Parisi, F. Beneventi, M. Polato, M. Torquati, L. Benini, and M. Aldinucci. Experimenting with emerging RISC-V systems for decentralised machine learning. In *20th ACM International Conference on Computing Frontiers (CF '23)*, Bologna, Italy, 2023. ACM. [Online].

[60] O. Monticelli, M. E. Santimaria, M. Aldinucci, and I. Colonnelli. Heuristic-free system call interception on RISC-V. In *RISC-V Summit Europe 2026*, Bologna, Italy, June 2026.

[61] W. S. Moses, L. Chelini, R. Zhao, and O. Zinenko. Polygeist: Raising c to polyhedral mlir. In *Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques*, PACT '21, page 45–59. IEEE Press, 2024.

[62] Open Compute Project. OCP 8-bit floating point specification (OFP8). [Online], 2023.

[63] D. Rossi, F. Conti, M. Eggiman, A. D. Mauro, G. Tagliavini, S. Mach, M. Guermandi, A. Pullini, I. Loi, J. Chen, E. Flamand, and L. Benini. Vega: A ten-core soc for iot endnodes with dnn acceleration and cognitive wake-up from mram-based

state-retentive sleep mode. *IEEE Journal of Solid-State Circuits*, 57(1):127–139, 2022.

[64] F. F. Ruffino, L. Bellentani, G. Rossi, F. Affinito, S. Baroni, O. Baseggio, P. Delugas, P. Giannozzi, J. Kurzak, Y. Luo, O. O'Reilly, S. Orlandini, and I. Carnimeo. Quantum espresso towards performance portability: Gpu offload with openmp. *Procedia Computer Science*, 240:52–60, 2024. Proceedings of the First EuroHPC user day.

[65] M. E. Santimaria, A. R. Martinelli, I. Colonnelli, B. Cantalupo, M. Torquati, and M. Aldinucci. CAPIO-CL: The CAPIO coordination language. *International Journal of Parallel Programming*, 53(2):10, Feb. 2025.

[66] SemiconductorEngineering. Speeding up AI with vector instructions, 2020. [Online] (Accessed: 2026-06-23).

[67] SiFive. Introduction to SiFive vector processors, 2022. [Online] (Accessed: 2026-06-23).

[68] G. Spadaro, M. S. Ali, A. Presta, G. Pilo, S.-H. Bae, J. H. Giraldo, A. Fiandrotti, M. Grangetto, and E. Tartaglione. ALICE: Adapt your Learnable Image Compression modEl for variable bitrates. In *2024 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2024.

[69] N. Stephens, S. Biles, M. Boettcher, J. Eapen, M. Eyole, G. Gabrielli, M. Horsnell, G. Magklis, A. Martinez, N. Premillieu, A. Reid, A. Rico, and P. Walker. The ARM scalable vector extension. *IEEE Micro*, 37(2):26–39, 2017.

[70] A. J. Strojwas, K. Doong, and D. Ciplickas. Yield and reliability challenges at 7nm and below. In *2019 Electron Devices Technology and Manufacturing Conference (EDTM)*, pages 179–181, 2019.

[71] X. Sun, J. Choi, C.-Y. Chen, N. Wang, S. Venkataramani, V. Srinivasan, X. Cui, W. Zhang, and K. Gopalakrishnan. Hybrid 8-bit floating point (HFP8) training and inference for deep neural networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[72] J. Tang et al. The compilation performance tests and analysis of RISC-V compilation toolchain on the spec cpu® 2017. In *9th Int'l Symp on Next Generation Electronics (ISNE)*, pages 1–4. IEEE, 2021.

[73] E. Tejedor, Y. Becerra, G. Alomar, A. Queralt, R. M. Badia, J. Torres, T. Cortes, and J. Labarta. Pycompss: Parallel computational workflows in python. *Int. J. High Perform. Comput. Appl.*, 31(1):66–82, 2017.

[74] N. Tonci, M. Torquati, G. Mencagli, and M. Danelutto. Distributed-memory fastflow building blocks. *International Journal of Parallel Programming*, 51(1):1–21, Dec. 2022.

[75] Y. Tortorella, L. Bertaccini, D. Rossi, L. Benini, and F. Conti. RedMulE: A Compact FP16 Matrix-Multiplication Accelerator for Adaptive Deep Learning on RISC-V-Based Ultra-Low-Power SoCs. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1099–1102, 2022.

[76] UALink Consortium. About UAlink, 2025. [Online].

[77] G. Vilar De Farias, J. E. Rodriguez Condia, and M. Sonza Reorda. A flexible framework for vector accelerators in-field testing. *Proceedings of the 44th IEEE VLSI Test Symposium*, 2026.

[78] P. Vizcaino, F. Mantovani, R. Ferrer, and J. Labarta. Acceleration with long vector architectures: Implementation and evaluation of the FFT kernel on NEC SX-Aurora and RISC-V vector extension. *Concurrency and Computation: Practice and Experience*, 35(20):e7424, 2023.

[79] P. Vizcaino, F. Mantovani, J. Labarta, and R. Ferrer. Rave: Risc-v analyzer of vector executions, a qemu tracing plugin. In *Parallel Processing and Applied Mathematics: 15th International Conference, PPAM 2024, Ostrava, Czech Republic, September 8–11, 2024, Revised Selected Papers, Part II*, page 3–16, Berlin, Heidelberg, 2025. Springer-Verlag.

[80] S. Wang, G. Zhang, J. Wei, Y. Wang, J. Wu, and Q. Luo. Understanding silent data corruptions in a large production cpu population. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 216–230, New York, NY, USA, 2023. Association for Computing Machinery.