RISC-V-based Platforms for HPC: Analyzing Non-functional Properties for Future HPC and Big-Data Clusters

William Fornaciari[1][0000−0001−8294−730X], Federico Reghenzani[1][0000−0002−1888−9579], Federico Terraneo[1][0000−0001−7475−6167], Davide Baroffio[1][0009−0007−3112−9869], Cecilia Metra[2][0000−0002−1408−5725], Martin Omana[2], Josie E. Rodriguez Condia[3][0000−0001−5957−5624], Matteo Sonza Reorda[3][0000−0002−2872−7071], Robert Birke[4][0000−0003−1144−3707], Iacopo Colonnelli[4][0000−0001−9290−2017], Gianluca Mittone[4][0000−0002−1887−6911], Marco Aldinucci[4][0000−0001−8788−0829], Gabriele Mencagli[7][0000−0002−6263−7723], Francesco Iannone[6][0000−0002−8620−6354], Filippo Palombi[6][0000−0002−3053−3751], Giuseppe Zummo[6], and Daniele Cesarini[5], Federico Tesser[5]

[1] Politecnico di Milano, Milano, Italy
name.surname@polimi.it
https://heaplab.deib.polimi.it

[2] University of Bologna, Bologna, Italy
n.surname@unibo.it
https://www.unibo.it

[3] Politecnico di Torino, Torino, Italy
n.surname@polito.it
https://www.polito.it

[4] University of Turin, Turin, Italy
n.surname@unito.it
https://www.unito.it

[5] CINECA, Bologna, Italy
n.surname@cineca.it
https://www.cineca.it

[6] ENEA, ROMA, Italy
n.surname@enea.it
https://www.enea.it

[7] University of Pisa, Pisa, Italy
n.surname@unipi.it
https://www.unipi.it

Abstract. High-Performance Computing (HPC) have evolved to be used to perform simulations of systems where physical experimentation is prohibitively impractical, expensive, or dangerous. This paper provides a general overview and showcases the analysis of non-functional properties in RISC-V-based platforms for HPCs. In particular, our analyses target the evaluation of power and energy control, thermal management, and reliability assessment of promising systems, structures, and technologies devised for current and future generation of HPC machines. The main set of design methodologies and technologies developed within the activities of the Future and HPC & Big Data spoke of the National Centre of HPC, Big Data and Quantum Computing project are described along with the description of the testbed for experimenting two-phase cooling approaches.

Keywords: High Performance Computing (HPC) · Power Modeling and Control · Reliability · RISC-V-based Platforms

1 Introduction

In the next few years, an unprecedented amount of data are expected to be produced by scientific, industrial, and institutional actors, so we will have to face the challenge of extracting social and economic value from this data explosion. In this context, supercomputing, numerical simulation, Artificial Intelligence, high-performance data analytics and Big Data management will be essential and strategic for understanding and responding to grand societal challenges and in stimulating a people-centered process of sustainable growth and human development, allowing academia, industry and institutions to develop services and discoveries. Current challenges demand effective and extensive computational power resources to perform increasingly accurate and complex simulations within acceptable time frames. Modern HPCs exploit distributed computing strategies, in combination with smart co-design techniques to effectively integrate and correctly operate hardware platforms and software frameworks, prioritizing the operational throughput and performance of the complete system, and aiming to achieve their nominal computational power and acceptable levels of performance efficiently (i.e., in terms of operations per watt).

Based on the above observations, the European Commission and the Italian government recently launched a large project aimed at creating a national HPC infrastructure for research and innovation, and forming a globally attractive ecosystem based on strategic public-private partnerships.

The National Centre on HPC, Big Data and Quantum Computing project is organized in 11 "*spokes*". Spoke 1, named Future HPC & Big Data, aims at developing new HW and SW technologies for future HPC systems. In particular, the spoke activities focus on hardware technologies and systems, on the design of energy-efficient and reliable parallel processors, accelerators, memory, storage hierarchy, and interconnects. Special attention will be devoted to open instruction sets (RISC-V), open architectures, and open hardware for advanced computing. Obviously, the spoke also covers Software Technologies and Tools, such as Programming models for modern HPC applications (shared-memory, message-passing, with-accelerators (e.g., GPU and FPGA), workflow management systems, high-performance I/O, ad-hoc file systems and high-performance streaming, parallel algorithms and libraries for scientific computing, high-performance compilers and run-time support systems, domain-specific languages and tools, benchmarking and software development methods and optimization for HPC-powered innovative applications, middleware for scalable BigData and AI/DL and their convergence with HPC systems, performance modeling, analysis, and simulation for complex parallel systems, heterogeneous computing and resource scheduling; integration of quantum computing kernels into traditional software pipelines, tools and libraries for distributed and Federated Machine Learning. The spoke activities will be organized in 5 workpackages.

This paper focuses on the goals and preliminary results achieved in the frame of the first workpackage, dealing with non-functional properties, allowing design exploration of energy, power and reliability characteristics.

1.1 The Italian National Center for High Performance Computing

The National Research Center for High Performance Computing, Big Data, and Quantum Computing (NRHPC) is one of the five National Centers funded by the National Recovery and Resilience Plan (PNRR) and dedicated to strategic sectors for the country's development: simulations, high-performance data computation and analysis, agritech, development of gene therapy and RNA-based drug technologies, sustainable mobility, biodiversity.

The National Supercomputing Center's activities will be divided into two primary areas. Firstly, there will be a strong emphasis on maintaining and improving the Italian HPC and Big Data infrastructure. Secondly, the center will be dedicated to advancing numerical methods, applications, and software tools to seamlessly integrate computation, simulation, data collection, and analysis. These advancements will cater to the needs of research, production, and society as a whole. Furthermore, the center will employ cloud and distributed approaches to achieve this integration.

The NRHPC will actively involve and encourage the collaboration of top interdisciplinary experts in the fields of science and engineering. This will facilitate significant and sustainable innovations across a wide range of domains, including fundamental research, computational and experimental sciences related to climate, environment, and space. Additionally, it will encompass the study of matter, life sciences, medicine, materials technologies, information systems, and devices. Moreover, the NRHPC will provide support for advanced education and play a pivotal role in fostering the development of policies aimed at responsible data management. It will adopt an open data and open science approach, combining elements of regulation, standardization, and compliance to ensure the effective utilization and dissemination of scientific data.

The NRHPC represents a collaboration among universities, public and private research institutions, and businesses throughout the entire country. Its organizational structure follows the Hub and Spoke model, with the Hub overseeing management and coordination, while the Spokes undertake activities to accomplish the objectives.

The Hub assumes responsibility for validating and managing work programs, while the execution of activities is carried out by the Spokes and their associated entities. This includes the provision of open opportunities for research institutions and external companies to participate, irrespective of their affiliation with the ICSC Foundation, which manages the NRHPC.

The NRHPC will consist of a total of 11 Spokes, one dedicated to infrastructure and ten dedicated to specific thematic areas ranging from fundamental research from to HPC and cloud infrastructure. In particular the main focus of Spoke 1, "Future HPC & Big Data," centers around the technological aspect, specifically the development of cutting-edge hardware and software technologies for future supercomputers. The objective of Spoke 1 is to establish new laboratories that form an integral part of a world-class national federated center with expertise in hardware and software co-design. Furthermore, it seeks to enhance Italy's leadership in the EuroHPC Joint Undertaking and the data infrastructure ecosystem serving science and industry. The planned research and development endeavors within Spoke 1 will result in the creation of prototypes and demonstrators showcasing the most promising technologies, thereby facilitating their adoption and fostering industrial advancement. Collaboration with industry will be paramount in defining an innovation strategy that extends beyond supercomputers, exerting a significant impact on high-volume markets like edge servers, IoT gateways, autonomous vehicles, and the cloud. To optimize and assess the socioeconomic impact of the activities, a dedicated research group has been established that cuts across all the spokes.

1.2 Organization of the paper

The organization of the paper is structured as follows: Section 2 provides an overview of the power evaluation and management platforms for HPCs, in particular targeting power monitoring, memory reliability, and thermal management. Section 3 describes the reliability evaluation and management platforms for HPC systems addressing the technology, architecture and system levels. Section 4 introduces the performance monitoring and management platforms approaches for single and multi-node HPC systems. Then, Section 5 describes a case study focusing on thermal management solutions for commodity clusters in HPCs. Finally, Section 6 provides some concluding remarks.

2 Power Evaluation and Management Platforms

2.1 Power Monitoring

High-performance computing systems nowadays face significant challenges correlated to their efficiency. One of the most substantial contest is related to power and energy consumption, which, as a result of the end of validity of the Dennard's scaling, has started to impact the peak performance and cost-effectiveness of supercomputers. Moreover, developing new hardware and software has become challenging for the needs of both security and reliability guarantees. However, these new features add a layer of complexity in the daily management of the systems for the administrators, and they also create a certain amount of sophistications for those users who want to obtain the maximum from their codes (i.e., job performance, power consumption, and anomalies detection - see, e.g., [4]).

For these reasons, data center automation approaches seem to be the right direction for predictive and processes maintenance, creating a monitoring framework able to automatically detect faults and anomalous states and to improve the normal system management, reacting in a proactive way to all the information obtained by a multitude of heterogeneous sensors. Additionally, this approach can be considered entirely realized, if this framework can analyze also the job level, intercepting the intrinsic features of the different applications monitored and being capable to reduce their energy consumption.

Over the past few years, Examon [6] has been developed, as a monitoring framework adaptable and capable of handling GBs of telemetry data per day from the entire datacentre, being also integrable with machine learning and artificial intelligence techniques and tools. In addition, our approach is to integrate it with COUNTDOWN [11] library to monitor application performance and energy efficiency. By linking this information with data from the facility, the idea is to grant to the users access to a visual dashboard, where they can receive run-time details related to energy consumption and performance evaluation of their jobs.

2.2 Memory Reliability

RISC-V based SoCs, like any other high performance SoC, make a massive use of cache memories (up to 80% of the chip area) to eliminate the memory bottleneck problem. As a consequence, soft errors affecting cache memories will be of major concern for RISC-V based SoCs implemented by scaled technologies [18].

Traditionally, Error Correcting Codes (ECCs) are adopted to protect cache memories of high performance SoCs against soft-errors. The adoption of ECCs mandates the addition of encoding/decoding blocks to the memory array. Due to the limited area of these additional blocks compared to the cache array, the occurrence of faults affecting them in the field is typically neglected. Therefore, the encoding/decoding blocks are typically not protected against possible faults affecting themselves. While this risk has been considered acceptable so far, this is no longer the case in the perspective of high performance SoCs to be used in highly autonomous systems (e.g., highly autonomous vehicles, robots, etc.), due to their strong requirements in terms of reliability and functional safety. In fact, it can be expected that faults affecting the encoder/decoder blocks of ECCs may result in a mis-correction, even if the original word read from the cache was error-free. In this case, the decoder will produce an incorrect output word, that will be propagated throughout the system, thus compromising the SoC reliability, with a dramatic impact on system's functional safety.

Some solutions have been presented in the literature to prevent the catastrophic consequences of permanent faults affecting ECC's encoding/decoding blocks [26]. However, they imply a significant impact on performance (which may be over 100%, depending on the considered ECC), and they also require a non-negligible costs in terms of area and power overhead.

In order to fill the gap of the state of the art regarding efficient solutions to prevent the catastrophic consequences of faults affecting ECC's encoding/decoding blocks of modern SoCs, we will first analyze, at the electrical level, the effects of permanent faults possibly affecting the ECCs' encoding/decoding blocks during their operation in the field. We will introduce also metrics to evaluate the risks of the considered faults' effects on functional safety, thus identifying the most critical faults. The performed analyses and metrics will enable to develop low-cost innovative approaches to detect the occurrence of those faults that can compromise system's functional safety, thus enabling the activation of possible recovery mechanisms to re-establish the SoC correct operation.

2.3 Thermal Management

The purpose of the multi-level thermal management policy is to ensure adequate cooling of the computing devices. Compared to standard approaches to thermal management, it takes advantage of the evaporative cooling solution to limit reducing the operating frequency, thereby improving computational performance.

At the same time, the policy adapts to the computational workload to avoid over-provisioning of the available cooling capacity.

Temperature rise in integrated circuits is governed by two timescales, the first induced by the thermal capacitance of the silicon die, which due to its small physical size results in fast temperature transients that in modern HPC chips is in the order of milliseconds to tens of milliseconds. The second timescale is due to the thermal capacitance of the heat dissipation solution, that is significantly bulkier than the silicon chip, resulting in considerably longer timescales in the order of seconds to minutes.

As such, when an increase in power dissipation caused by computational load transients occurs, temperature must first be kept under control using fast actuators such as Dynamic Voltage and Frequency Scaling (DVFS), as simply increasing the coolant flow rate would not be fast enough. However, reducing operational frequencies reduces the dissipated power at the expense of a performance degradation. In the absence of a controllable evaporative cooling solution, this performance degradation will persist as long as the required power consumption of the computational devices exceeds the cooling capacity. This is what happens in commercial thermal policies such as Intel Turbo Boost, where the boost frequency can only be kept for a limited period of time of high CPU activity, after which the frequency is reduced to the base value.

To overcome this limitation, the proposed multi-level thermal management policy is of the hierarchical nature, and adds to the system a second control loop acting on the evaporative coolant flow rate, with the aim of taking advantage of the increasing dissipation heat flux caused by two-phase evaporative cooling to partially restore peak operating frequency and provide sustained high performance operation while keeping the operational temperature under the specified threshold. Experiments will be carried out on the testbed described in Section 5.

3 Reliability Evaluation and Management Platforms

Modern HPC machines progressively scale in size to target exascale performance for demanding applications, which also influence and increase the failure probability in their components (e.g., processors, hardware accelerators, communication links, and circuit sockets). In fact, the resilient operation of HPCs and their underlying hardware and software is crucial to provide services with acceptable quality and accuracy. The HPC dimension and their considerable complexity involve reliability challenges since software and hardware components' fault rates differ during their operative lifetime (from 53% to 64% in hardware components) [32]. Thus, methods and strategies to model, evaluate, and quantify the HPC's state and identify anomalies are mandatory when adapting new processor architectures, such as RISC-V-based SoCs into the HPC domain.

Among the three main guidelines to improve the reliability and resilience of HPCs (overheat management, the identification of fault rate factors, and the development of fault detection and mitigation mechanisms [25]), the reliability evaluation supports the second and third guidelines by providing a method to improve the system's reliability through the characterization of the fault and error effects and how these impact the HPC's hardware and software. These analyses allow the identification of vulnerable structures (or sub-systems) prone to propagate faults and errors. Commonly, the reliability characterization employs one or several fault and error models to represent the impact of physical defects on hardware and corruptions in software.

The reliability assessment in HPCs can be divided in several layers, from the technology level, the structural and modular levels, and the system and application level. The next subsections discuss the main targets for the characterization and reliability evaluation.

3.1 Evaluation at the Technology Level

The current technology scaling approaches reduce power-supply voltages (noise margins) and node capacitance that contributes to increase of operating temperature. Thus, the susceptibility to faults (both transient and permanent) in modern SoCs continue increasing, due for example to premature aging phenomena (such as *Bias Temperature Instability*, or BTI) [21][24].

To address the current reliability issues, we target the lack of accurate analysis and modelling approaches to evaluate the effects of latent faults and aging phenomena affecting simultaneously FinFETs transistors of data-paths of modern RISC-V-based SoCs during their in-field operation. Based on the results achieved by the analyses, possible low-cost monitors to detect the presence of latent faults, during SoC operation in the field, might be then derived.

We plan to evaluate, at the electrical level, the effects of likely FinFETs faults (e.g., at 7nm technology using six and eight fins) occurring individually (i.e., not combined with aging phenomena). The goal of this evaluation targets the identification of the subset of FinFET faults that may not be detected during manufacturing testing, thus becoming "latent" faults that could combine with aging phenomena during the SoC operation in the field.

3.2 Evaluation at Architecture and System Levels

We employ architectural and low-level microarchitectural descriptions of the hardware to perform focused reliability evaluations on individual units (e.g., processor cores, such as **RI5CY**[8] or **Hero RISC-V**[9], and accelerators, such as in-chip GPUs[12] or **NVDLA** [10]) that interact with the HPC system, or based on complete systems running equivalent workloads that must consider effects of in-field operation and representative applications.

We target the reliability evaluation and characterization resorting to simulation-based fault injection campaigns in combination with deployment in real platforms (i.e., using efficient co-simulation and cross-layer strategies[13][31]) for the reliability evaluation of the architectural features and the system operation of individual commodity clusters, as well as more elaborated HPC machines.

Since HPC workloads are massive in size and the reliability evaluation must determine the incidence of the faults in the system, both factors (workload size and fault universe) influence the evaluation times. Thus, in this case, the use of efficient and effective evaluation strategies involves cross-layer operations to evaluate and propagate faults effects, so aiming at identifying vulnerable structures in the architecture of a component or sub-system under feasible evaluation times. The main outcomes can be used to address and propose hardware-based hardening solutions for the execution cores (processors and hardware accelerators) by exploring and adapting mitigation strategies, such as flexible Built-In Self-Repair mechanisms[14], and re-configurable mechanisms.

An outstanding opportunity to increase the reliability of RISC-V-based platforms relies on the proposal of effective and accurate functional tests solutions, considering the underlying hardware. Unfortunately, until now, most solutions are based on high-level software approaches that focus on verifying the software layers and the complete system state. However, hardware testing (focused on the underlying architecture of the commodity clusters, such as processors and hardware accelerators) is barely deployed during the production stages of the HPC by restrictions on their execution time or the availability of effective hardware tests due to the lack of hardware details. Interestingly, both restrictions can be solved in open-hardware environments, such as those based on RISC-V platforms for HPCs to improve the effectiveness of functional testing mechanisms for HPCs, allowing the merging of performance and functional test goals (typical of HPC system tests) with hardware testing goals. The availability of the hardware architecture in combination with the adaption of functional testing strategies for hardware, such as the Software-Based Self-Test (SBST) [15][20], might contribute to designing more effective testing routines considering the architectural

[8] https://github.com/embecosm/ri5cy

[9] https://pulp-platform.org/hero.html

[10] http://nvdla.org/

features of all hardware elements composing the commodity clusters (processors, accelerators, and intra-node interconnect infrastructures).

3.3 System-level Fault Tolerance for Real-time Applications

Fault resilience is traditionally implemented with hardware solutions. A different and more flexible approach consists of implementing the different aspects of fault tolerance at the software level and, in particular, at the operating system level, which obviously benefits development costs and maintainability. These techniques are under the umbrella term *Software-Implemented Hardware Fault Tolerance (SIHFT)* [19], including both fault detection and fault recovery strategies.

Regarding fault detection, recent tools implemented into compilers, especially LLVM, can be used to automatically implement SIHFT techniques transparently to the developer [8,5]. These tools are still experimental and may require extensions and further research, especially with the integration with HPC libraries.

Implementing SIHFT and workload migration (at any level) present numerous challenges when the applications must satisfy real-time constraints. Time-critical applications need to satisfy the time constraint even in a case of fault: the recovery process from a fault must still satisfy the timing constraints. Scheduling policies must be aware of the failure requirements and the presence of SIHFT mechanisms. Recently, novel models that integrate real-time requirements with the aforementioned failure requirements have been developed [27,29]. Further developing these models and their implementation in the HPC context is a key enabler to allow real-time and fault-tolerant applications in the domain. Another important issue is to determine the WCET via proper tools. Indeed SIHFT approaches would often require re-execution or running in replica-mode multiple tasks. Therefore, a tight WCET estimation is very important, to avoid duplicating, or even more, the over-approximations of existing tools. In this context, we can exploit the probabilistic estimations, such as the `chronovise` tool, to obtain a tight estimation. The whole picture of the existing works/tools can be get from a recent survey [30].

4 Performance Monitoring and Management

4.1 Performance and Power monitoring at the distributed level

A distributed system in an HPC reality consists of a collection of multiple computing systems linked to one another through a high-bandwidth and low-latency network, which presents some advantages like being efficient, scalable and highly available. Of course, the computational entities that are part of the distributed system must be able to coordinate among themselves, in order to share all the resources of each component in their totality, and to give to the users the perception of using a single computing unity.

In this context, it https://it.overleaf.com/project/64526b0222776c44b02a83f4has been introduced the Message Passing Interface (MPI), which is a standardized and portable message-passing system specific for distributed and parallel computing, which lets different processes to exchange explicit messages by abstracting the underlying network level.

But when the scale of the application increases, the time spent in the MPI library becomes not negligible and sophistication arises, impacting the overall power consumption and the analysis of performances and possible bottlenecks. That's why is important, in this cases, being able to analyze the behaviour of your own applications, without considerably increasing the original time to solution (TTS). Moreover, extracting workload traces of the underlying distributed applications can become an hard task, taking into account a lot of architectural features (super-scalarity, out-of-order execution, complex instructions, multi-threads/cores/sockets/caches, NUMA domains, ...), different performance events (on-core and off-core) and microarchitectures to analyze, and the fact of merging together all the informations from multiple computing systems.

The COUNTDOWN [10] runtime library frees the users by all these low level intricacies: it automatically reduces the power consumption of the computing elements during MPI communication and synchronization, and can extract workload traces using a user-defined time-based approach. Everything is done transparently to the user, with a negligible overhead. Future works on COUNTDOWM will take into consideration the Roofline Model [34], to give to the user an estimation of how well performed the monitored application, without asking them to define specific measurement events or to analyze their associated traces.

4.2 Performance monitoring of parallel applications

Monitoring the non-functional behavior of parallel applications is a critical activity. An effective monitoring approach should be less intrusive as possible, so exhibiting low run-time overheads. The approaches studied over the years are based on *profiling* and *tracing* techniques [1]. Profiling-based approaches gather online statistics from the running application and provide coarse-grained information aimed at identifying performance bottlenecks. Tracing-based approaches are instead more compelling, since they capture the whole time-series of both software-related and hardware-related events. They allow a more sophisticated *ex-post* analysis able to identify the root cause of bottlenecks. However their adoption at runtime, to identify and removing bottlenecks through runtime reconfigurations, is very challenging since they require a large computational and storage overheads. An interesting research perspective is the one provided by the so-called *Structured Parallel Programming* methodology where profiling/tracing techniques can be enhanced with *model-driven approaches* where the knowledge about the application structure can be profitably used to build effective performance prediction models, e.g., based on Queueing Networks [22]. This idea is currently under development in the FastFlow parallel programming framework [2], which has been recently ported to RISC-V platforms in addition to the full support already existing for commodity multi-core architectures based on Intel/AMD/Power CPUs.

4.3 Estimation of the probabilistic-WCET

Estimating the Worst-Case Execution Time (WCET) of tasks in HPC centers is extremely difficult due to the intrinsic temporal non-determinism of modern hardware architectures and the high complexity of such distributed systems. Indeed, traditional static techniques fail in determining a safe and tight WCET in such systems. A possible solution is to use measurement-based WCET analyses, that infer it by observing the execution time rather than performing a static analysis of the software and hardware. The use of probabilistic techniques to obtain the probabilistic-WCET (pWCET) in embedded systems dates back to 2001 [17] and two surveys [9,16] recap all recent works in the field. A preliminary study on the use of pWCET for HPC has been published in 2020 [28]. How to design the computing platform and HPC clusters as a whole is still an open problem and it will be addressed during the project timeframe.

4.4 Performance Comparison of RISC-V ML software

The RISC-V platform is experiencing a double-fold developmental stress: on the one side, researchers are pushing it toward the performance and scalability properties needed for building HPC infrastructures while, on the other side, its low power consumption makes it a desirable candidate for IoT applications. One example of HPC-oriented employment of RISC-V processors is Monte Cimone [7], the first prototype of a RISC-V-based HPC cluster. In addition, many researchers are currently spending their effort on developing RISC-V-based accelerators and ISA extensions to support better modern workloads, such as ML-based ones. In this context, we started to develop an experimental software, FastFederatedLearning [11] (FFL) [23].

FFL is fully implemented with C/C++ code to retain high execution performance and not spoil the RISC-V's limited computational power. Our RISC-V porting of PyTorch [12] backs up the ML computations, while the high-performance C/C++ header-only FastFlow [2,33] programming framework provides the distributed communication infrastructure. We selected two use cases for our experiments: training a simple Multi-Layer Perceptron (MLP) on the MNIST dataset and running inference with a large-scale Deep Neural Network called YOLO-v5n on a short 30-seconds video. We evaluate the results obtained from the perspective of execution time and power consumption, comparing them to the more advanced x86-64 and ARM-v8 platforms. This information will help us understand the current maturity level of the RISC-V platform and which are the development steps to be taken further.

We assessed that the RISC-V platform is an order of magnitude slower than the x86-64 and ARM architectures in doing the same amount of computation while consuming a comparable or even greater quantity of energy. This fact indicates a significant lack of efficiency in the RISC-V platform that should be ad-

[11] https://github.com/alpha-unito/FastFederatedLearning

[12] https://github.com/pytorch/cpuinfo

dressed: despite its low Thermal Design Power of only 5W (x86-64: 125W, ARM-v8: 250W), we assessed an energy-per-FLOP ratio of 15.9nJ, which is the highest in the comparison (x86-64: 12.8nJ, ARM-v8: 3.2nJ). While this fact is due to the novelty of the RISC-V platform, it should be considered when designing future RISC-V software. In particular, we highlight how running general-purpose commercial off-the-shelf code can be suboptimal, thus making it preferable to look more at an HPC-oriented software stack for this platform.

We start our software investigation by comparing how a single Monte Cimone node performs on the MNIST benchmark from PyTorch's official repository with the two available APIs. Python requires 442.8s, while C++ only 314.5s (mean of 5 runs). Since PyTorch's Python API is only a wrapper of the underlying C++ code, we compare our full-stack C/C++ FL software with a standard, Python-based FL one called OpenFL to investigate how deeply the Python code impacts execution performance on both the RISC-V and x86-64 platforms. On the RISC-V, with FFL we measure a mean of 673.70 seconds for training a simple MLP on the MNIST dataset for 100 epochs against 2,486.52 seconds with OpenFL. On the x86-64 platform, we measure 23.56 seconds for FFL and 59.15 seconds for OpenFL for the exact computation as before. While there is a decent speedup in both cases (2.5 for x86-64 and 3.6 for RISC-V), it should be noted that RISC-V suffers more from the execution of Python code than x86-64.

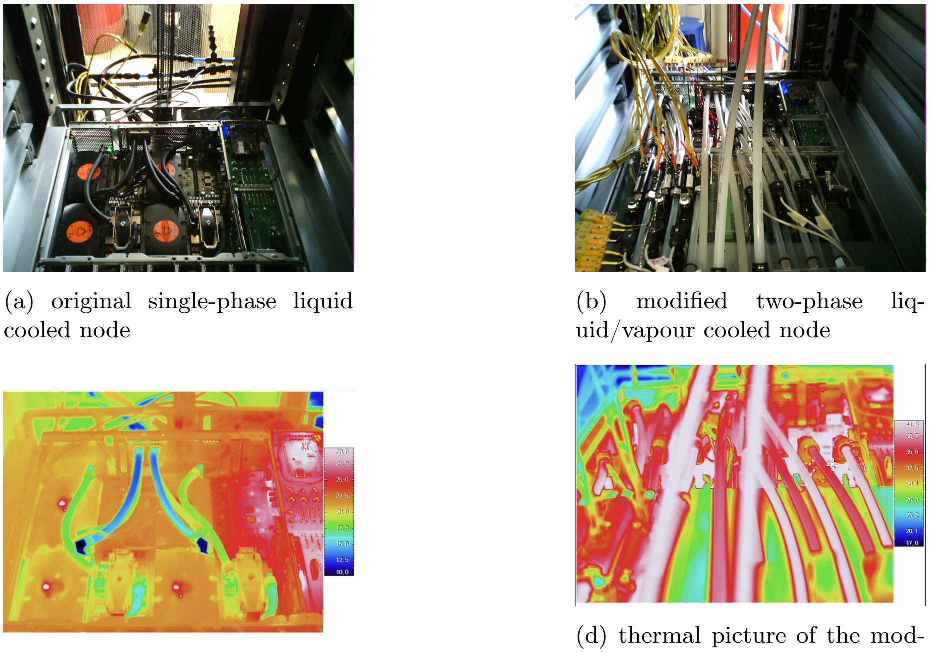
Given these results, we advocate the need for further development of the RISC-V software stack to both improve the performance and compatibility of existing commercial code and to produce new, native software capable of taking full advantage of the RISC-V open ISA and overcoming the low efficiency of the current hardware implementation.

5 Experimental Testbed for Two-Phase Cooling

Using a closed-loop liquid circuit to cool electronic components is not a novel technology. This method was initially used in mainframes or HPC systems. Nowadays, cost-effective variations of fluid-based cooling have been created and made accessible to PC users aiming at optimizing their computer's performance.

Direct liquid cooling (DLC) can be single- or two-phase. In a two-phase system, both latent and sensible heat are used. The cooled fluid flows from a cold heat exchanger/condenser to a heatsink (CPU or GPU). Here, the fluid heats and evaporates. Vapour circulates back to the condenser, where heat dissipates in the outside environment, vapour condenses to the liquid phase and the loop starts again. An expansion tank (reservoir) regulates saturation conditions.

To assess the improvement in energy saving of the two-phase cooling technology compared to the traditional single-phase one, we have been developing an experimental test setup in which commercial computing nodes based on single-phase liquid cooling of high-heat generating components (CPUs and GPUs) have been modified so as to implement two-phase liquid/vapour cooling, as shown in Figs. 1a–1d.



(a) original single-phase liquid cooled node

(b) modified two-phase liquid/vapour cooled node

(c) thermal picture of the original single-phase liquid cooled node

(d) thermal picture of the modified two-phase liquid/vapour cooled node

Fig. 1: Pictures of the testbed

A schematic layout of the testbed is shown in Fig. 2. We summarize below its main features:

- Two identical computing nodes in two distinct racks, one being cooled by single-phase liquid direct cooling, the other being cooled by two-phase liquid/vapour direct cooling;
- Direct cooling is applied to all CPU/GPU components. Each node has an Intelligent Platform Management Interface (IPMI) to provide and record front panel inlet and outlet fluid temperatures and CPU/GPU temperatures, as well as a data acquisition system for electric energy consumption;
- Asetek RackCDU technology on the node with single-phase liquid direct cooling. It consists of a rack-mounted CDU providing cooling water distribution to the computing node, as well as cooling devices placed inside and outside the node;
- IN4 CDU technology on the node with two-phase liquid/vapour direct cooling. It allows variable inlet fluid temperatures and flow rates. Temperature set points are manually tuned and the fluid flow rate is manually controlled between the building chilled water system and the IN4 CDU as requested for temperature stability and to adjust the inlet fluid temperature;

- An external microcontroller to acquire the electric power of each computing node with a precision of at least 1% and a sampling rate of at least 1sec. The acquired data are collected via JSON objects accessible through web services;
- several sensors on the Asetek RackCDU and IN4 CDU to measure flow rates, inlet and outlet fluid temperatures;
- Workload software based on Quantum Fourier-Transform to stress CPUs/GPUs. Run configuration is varied to control the power and the heat generated by each node.

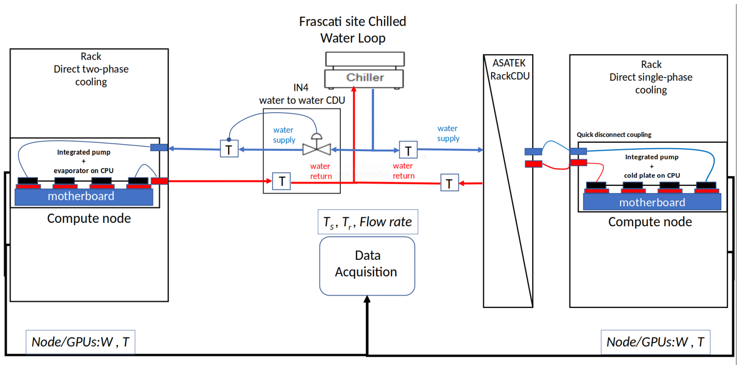


Fig. 2: Schematic layout of the testbed

Once our experimental testbed is ready for clean measurements, we can vary experimental parameters, including inlet fluid temperature, inlet fluid flow rate, and computing power of the nodes. Specifically, we can analyze the cooling performance obtained with different outlet fluid temperatures to investigate the possibility of heat reuse. At the end of the project, we expect to have a proof-of-concept of two-phase vapour/liquid cooling on an HPC system with quantitative estimates of the achievable energy savings.

6 Concluding remarks

This paper presented the preliminary achievements and plans for the activities that will be carried out in the first workpackage of the Future and HPC & Big Data spoke of the National Centre of HPC, Big Data and Quantum Computing project, whose focus is on developing reliable HPC platforms. The testbed used for experimenting innovative two-phase cooling solutions has been also described. As part of the future plans, the HPC4AI open access lab of the University of Turin [3] is acquiring a computing platform to be used also for commercial purposes, exploiting the two-phase cooling strategies developed during this project.

Acknowledgements

This work has received funding by Spoke "*Future HPC & Big Data*" of the National Resilience and Recovery Plan (PNRR) through the National Center for HPC, Big Data and Quantum Computing (ICSC), funded by European Union — NextGenerationEU.

References

1. Adhianto, L., et al.: Hpctoolkit: Tools for performance analysis of optimized parallel programs. Concurr. Comput.: Pract. Exper. **22**(6), 685–701 (apr 2010)
2. Aldinucci, M., et al.: Fastflow: High-Level and Efficient Streaming on Multicore, chap. 13, pp. 261–280. John Wiley & Sons, Ltd (2017)
3. Aldinucci, M., et al.: HPC4AI, an AI-on-demand federated platform endeavour. In: 15th ACM International Conference on Computing Frontiers (CF '18) (May 2018)
4. Barcelo, N., et al.: On the complexity of speed scaling. In: Mathematical Foundations of Computer Science (MFCS) 2015 - 40th International Symposium. pp. 75–89 (2015)
5. Baroffio, D., et al.: Compiler-injected sihft for embedded operating systems. In: 20th ACM International Conference on Computing Frontiers (CF'23). pp. 1–7. ACM (2023)
6. Bartolini, A., et al.: Paving the way toward energy-aware and automated datacentre. in: Proceedings of the 48th International Conference on Parallel Processing (2019)
7. Bartolini, A., et al.: Monte cimone: Paving the road for the first generation of risc-v high-performance computers. In: 2022 IEEE 35th International System-on-Chip Conference (SOCC). pp. 1–6. IEEE, Belfast, United Kingdom (September 2022)
8. Bohman, M., et al.: Microcontroller compiler-assisted software fault tolerance. IEEE Transactions on Nuclear Science **66**(1), 223–232 (2019)
9. Cazorla, F.J., et al.: Probabilistic worst-case timing analysis: Taxonomy and comprehensive survey. ACM Comput. Surv. **52**(1) (2 2019)
10. Cesarini, D., et al.: Countdown slack: A run-time library to reduce energy footprint in large-scale mpi applications. IEEE Transactions on Parallel and Distributed Systems (2020)
11. Cesarini, D., et al.: Countdown: A run-time library for performance-neutral energy saving in mpi applications. IEEE Transactions on Computers (2021)
12. Condia, J.E.R., et al.: Flexgripplus: An improved gpgpu model to support reliability analysis. Microelectronics Reliability **109**, 113660 (2020)
13. Condia, J.E.R., et al.: Combining architectural simulation and software fault injection for a fast and accurate cnns reliability evaluation on gpus. In: 2021 IEEE 39th VLSI Test Symposium (VTS). pp. 1–7 (2021)
14. Condia, J.E.R., et al.: Dyre: a dynamic reconfigurable solution to increase gpgpu's reliability. The Journal of Supercomputing **77**, 11625–11642 (2021)
15. Condia, J.E.R., et al.: Using stls for effective in-field test of gpus. IEEE Design & Test **40**(2), 109–117 (2023)
16. Davis, R.I., Cucu-Grosjean, L.: A survey of probabilistic schedulability analysis techniques for real-time systems. Leibniz Transactions on Embedded Systems **6**(1), 04:1–04:53 (5 2019)

17. Edgar, S., Burns, A.: Statistical analysis of wcet for scheduling. In: Proceedings 22nd IEEE Real-Time Systems Symposium (RTSS 2001) (Cat. No.01PR1420). pp. 215–224 (2001)
18. Gava, J., et. Al.: Soft error assessment of cnn inference models running on a risc-v processor. In: 2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS). pp. 1–4 (2022)
19. Goloubeva, O., et al.: Software-implemented hardware fault tolerance. Springer Science & Business Media (2006)
20. Guerrero-Balaguera, J.D., et al.: Stls for gpus: Using high-level language approaches. IEEE Design & Test pp. 1–7 (2023)
21. Lodéa, N., et al.: Early soft error reliability analysis on risc-v. IEEE Latin America Transactions **20**(9), 2139–2145 (2022)
22. Mencagli, G., et al.: Spinstreams: A static optimization tool for data stream processing applications. In: Proceedings of the 19th International Middleware Conference. p. 66–79. Middleware '18 (2018)
23. Mittone, G., et al.: Experimenting with emerging risc-v systems for decentralised machine learning. In: 20th ACM International Conference on Computing Frontiers (CF '23) (May 2023)
24. Omaña, M., et al.: Low-cost strategy to mitigate the impact of aging on latches' robustness. IEEE Transactions on Emerging Topics in Computing **6**(4), 488–497 (2018)
25. Radojkovic, P., et al.: Towards resilient eu hpc systems: A blueprint. European HPC resilience initiative (2020)
26. Redinbo, G.R.: Fault-tolerant decoders for cyclic error-correcting codes. IEEE Transactions on Computers **C-36**(1), 47–63 (1987)
27. Reghenzani, F., Fornaciari, W.: Mixed-criticality with integer multiple wcets and dropping relations: New scheduling challenges. In: Proceedings of the 28th Asia and South Pacific Design Automation Conference. p. 320–325. ASPDAC '23, Association for Computing Machinery (2023)
28. Reghenzani, F., et al.: Timing predictability in high-performance computing with probabilistic real-time. IEEE Access **8**, 208566–208582 (2020)
29. Reghenzani, F., et al.: A mixed-criticality approach to fault tolerance: Integrating schedulability and failure requirements. In: 2022 IEEE 28th Real-Time and Embedded Technology and Applications Symposium (RTAS). pp. 27–39 (2022)
30. Reghenzani, F., et al.: Software fault tolerance in real-time systems: Identifying the future research questions. ACM Comput. Surv. (mar 2023), just Accepted
31. Santos, F.F.d., et al.: Revealing gpus vulnerabilities by combining register-transfer and software-level fault injection. In: 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 292–304 (2021)
32. Schroeder, B., Gibson, G.A.: A large-scale study of failures in high-performance computing systems. IEEE Transactions on Dependable and Secure Computing **7**(4), 337–350 (2010)
33. Tonci, N., et al.: Distributed-memory fastflow building blocks. International of Parallel Programming **51**, 1–21 (December 2023)
34. Williams, S., et al.: Roofline: an insightful visual performance model for multicore architectures. Communications of the ACM (2009)