Architectural Design and Orchestration of Heterogeneous Quantum-Classical Computing Systems

J.A. Bravo-Montes[1,2][a], Miriam Bastante[1][b] and Cyril Allouche[3]

[1] *HPC & Quantum, Eviden Iberia, Madrid, Spain*

[2] *Faculty of Informatics, Complutense University of Madrid, Madrid, Spain*

[3] *Research Quantum, Eviden, France*

Keywords: Heterogeneous Architecture, HPC, Quantum Computing, Quantum Emulators, Qaptiva.

Abstract: This study presents a comprehensive multilevel software architecture for integrating quantum and classical computing systems in High-Performance Computing environments. The proposed software architecture implements a hierarchical approach that seamlessly connects quantum circuit development with execution in both emulated and physical quantum processors. The system is structured in distinct layers: a Circuit Generator interface, HPC programming environment with specialized APIs and tools, NISQ calibration layer for managing quantum noise, and an integration layer that orchestrates the hybrid workload distribution. We demonstrate the implementation of this architecture in two leading European supercomputing facilities: the TGCC and the JSC. These implementations showcase the capability of the architecture to manage hybrid quantum-classical workflows, incorporating both local and remote QPUs while maintaining security and efficiency. The effectiveness of the architecture was validated through implementation of the VQE algorithm, demonstrating its practical application. The results highlight the architecture's ability to efficiently manage heterogeneous computational resources, provide secure access to quantum hardware, and facilitate the development and execution of hybrid quantum-classical algorithms. This study contributes to the advancement of quantum computing integration in HPC environments, providing a scalable framework for future quantum applications.

1 INTRODUCTION

High-Performance Computing (HPC) systems are nowadays at the forefront of modern computing technology by leveraging the power of parallel processing. This paradigm allows solving complex scientific and engineering problems that are fundamental to the development of new advances in different realms, such as computer science, geoscience, and physics simulation (Park et al., 2012). HPC systems often depend on deeply interconnected nodes that support a large amount of data movement and shared information tasks, allowing multithread processing throughout the integration of conventional Central Processing Units (CPUs) and the more recent Graphics Processing Units (GPUs) (Humble et al., 2021).

However, HPC faces several setbacks that hinder its application to specific problems of exponential nature, which are intractable even for the most advanced HPC systems. To address these problems, the emergence of quantum computing in recent decades offers a solution by harnessing quantum mechanics properties like superposition, entanglement or interference (Forcer et al., 2002). These basic principles have been demonstrated in small-scale systems, and efforts to create large-scale platforms are being pursued globally, reaching $40 billion public investment in 2024 (Insider, 2025). Despite these efforts, the idea of a fully quantum computer replacing classical ones is far from reality and does not really hold up, which is why there is increasing interest in the development of Quantum Processing Units (QPUs) to complement other processing units like CPUs and GPUs.

Historically, a technological revolution has already taken place with the arrival of GPUs in HPC centers. This is why integrating QPUs into the HPC model promises to elevate them to the same level of impact that GPUs had in the past. Companies around the world are in the race to develop and fully leverage this technology, and although QPUs are still in the early stages, various approaches are already under-

[a] https://orcid.org/0000-0002-1935-0997

[b] https://orcid.org/0009-0001-4076-3753

Bravo-Montes, J., Bastante, M. and Allouche, C.
Architectural Design and Orchestration of Heterogeneous Quantum-Classical Computing Systems.
DOI: 10.5220/0013653600004525
In *Proceedings of the 1st International Conference on Quantum Software (IQSOFT 2025)*, pages 154-161
ISBN: 978-989-758-761-0

way to implement them, aiming to mitigate the main challenges arising from their quantum nature, such as decoherence and scalability (Gill et al., 2024).

While fault-tolerant QPUs are still under development, HPC centers are integrating quantum environments that can leverage their HPC infrastructure. These environments enable the simulation of quantum circuit behavior using classical computing resources for various purposes, such as error-correction codes or quantum algorithm development. This approach makes use of HPC cluster infrastructure and allows the execution of larger quantum circuits with reduced execution times, simplifying the implementation of heterogeneous quantum-classical workflows for large-scale industrial problems (Beck et al., 2024). However, there are several challenges and limitations to integrate QPUs into existing HPC architectures. These challenges arise from the agnosticism of current quantum paradigms, the lack of memory resources, and the need for smooth communication between the numerous components of the systems.

Throughout this paper, we explore the different aspects of integration and orchestration of quantum-classical components in heterogeneous systems, particularly in HPC environments. This study has been structured as follows: section 2 gives an overview of existing processing options for executing jobs and available examples; section 3 details Eviden's proposed heterogeneous ecosystem architecture and other real-world implementations. Finally section 4 concludes the paper by discussing the scalability and flexibility of these architectures, underscoring their role in advancing quantum computing adoption in HPC.

2 BACKGROUND

HPC centers are facilities that house supercomputers and computer clusters designed to process large-scale and complex computational tasks. These centers allow researchers and industries to explore and solve problems in a wide range of applications, including scientific simulations, Artificial Intelligence (AI), and climate modeling (Park et al., 2012).

Although traditional HPC architectures rely on the combination of CPUs and GPUs, the rise in quantum computing has increased interest in integrating QPUs into existing architectures (Beck et al., 2024).

To achieve this heterogeneous system, it is important to orchestrate all its different components so that they can work together. HPC-Quantum environments require specialized middleware to enable interaction between classical and quantum processors. An effective implementation and integration of the building blocks of such architectures will optimize the workflow and lead to more efficient solutions in the industry. However, the agnostic nature of current QPU technologies, as well as frameworks, compilers, and schedulers, makes this task anything but trivial.

The current structure of typical HPC architectures is based on clusters of different types of processing units, including CPUs, GPUs or Parallel Processing Units (PPUs) (Gao and Zhang, 2016). This heterogeneous platform requires several layers before it can be used for real applications. In addition, integrating real (or emulated) QPUs into the system adds complexity to the already sophisticated workflow.

Upon initiation of a computational task by the user, an analysis is performed to determine the most suitable type(s) of processing units required for execution. Subsequently, a scheduler manages the temporal coordination and sequencing of computational processes. When considering computational environments, there are several options depending on the task at hand. HPC clusters provide the power needed for large-scale simulations, while quantum emulators enable the simulation of quantum algorithms on classical hardware. Finally, QPUs introduce a novel computational layer based on a new paradigm.

2.1 HPC Clusters

HPC clusters operate with several processing units capable of working together to perform complex computations at high speeds, leveraging the power of parallel processing and distributing tasks across nodes. With the integration of GPUs into the HPC realm, there has been significant improvement in performance and speed. These systems have applications in a wide range of fields, including AI, DNA sequencing, and stock trading.

The world's fastest supercomputer is *"El Capitan"* (Hewlett Packard Enterprise, 2025) located at Lawrence Livermore National Laboratory (LLNL). This supercomputer can operate at 1.74 EFlops/s and has an optimized processor for nuclear simulations, AI, and scientific research. In Europe, Italy's *"HPC6"* supercomputer is positioned the 5th most powerful supercomputer in the TOP500 list as November 2024 (TOP500 Project, 2024). This cluster has a peak performance of 606 PFloops/s with main applications in optimization of industrial facilities and accuracy of geological and fluid dynamic studies for CO2 storage (Eni S.p.A., 2025). Another remarkable mention in the 9th position is the *"Leonardo"* supercomputer, which allows computations at 306.31 PFlops/s.

Finally, in Spain the *"Marenostrum 5 ACC"*, is the most powerful HPC cluster of the country, with a peak of 249.44 PFlops/s (Barcelona Supercomputing Center, 2025) and is 11th in the TOP500 list. It is at the Barcelona Supercomputing Center and uses an energy-efficient architecture for AI and language model development.

2.2 Quantum Emulators

While some companies work on physical QPUs, another line of research focuses on developing and optimizing a complete quantum software stack ready to take advantage of the power of physical QPUs once they become operational. Quantum computer emulators enable researchers to test and optimize algorithms, diagnose hardware performance, and validate theoretical models without requiring access to physical quantum hardware, the resources of which are usually scarce.

Although emulating a QPU with high hardware resources is computationally expensive, particularly in terms of memory, it allows us to model and control the noise that naturally arises in a quantum computer (LaRose, 2018) (Georgopoulos et al., 2021). By studying different types of noise, we can overcome certain hardware limitations, leading to more powerful QPUs with a larger number of logical qubits.

Similarly to the physical case, the rapid advancement of research has led to the development of numerous models for simulating quantum computing. On the one hand, we have quantum emulators based on the digital, analog and annealing paradigm. Examples include Cirq (Google) (AI, 2024), Qiskit Aer (IBM Quantum) (Quantum, 2024), and EMU-TN (Pasqal) (Bidzhiev et al., 2023); the latter also uses tensor networks to simulate larger-scale quantum systems. D-Wave (Systems, 2024a) has an emulator for its quantum annealing computer. Moreover, there are also references capable of emulating the three computing paradigms: digital, analog and annealing, such as Qaptiva (Eviden, 2024).

2.3 QPUs

Since any two-level quantum system can be used to create a qubit, researchers are developing various types of qubits, each with characteristics that make them better suited for specific applications. For example, superconducting qubits benefits of already existing methods and processes that have been improved during last years. IBM (Condor) (AbuGhanem, 2024), Google (Willow) (Research, 2023) and Rigetti (Ankaa-3) (Rigetti, 2024) are betting on this technology. Another advanced approach is the trapped-ion qubits, which have longer decoherence times than superconducting qubits (Linke et al., 2017). Companies like Quantinuum (H2) (Quantinuum, 2024), IonQ (Forte) (IonQ, 2022) and Alpine Quantum Technologies (Marmot) ((AQT), 2024) are focusing their efforts on this technology.

Although these are currently the most advanced technologies, many others are in development with very promising characteristics. There is a line of research based on photonic qubits, which could theoretically create a universal quantum computer (Romero and Milburn, 2024). Following these results, Xanadu (X-Series) (Xanadu, 2024), Quantum Computing Inc (Dirac-3) (Diraq, 2024), Alice&Bob (Alice & Bob Quantum Computing, 2024) or PsiQuantum (Omega) (PsiQuantum, 2025) are putting their efforts into this research. Neutral atoms, which uses optically trapped atoms and Rydberg interactions for quantum operations, allowing for scalability and connectivity between qubits. Pasqal (Orion-Beta) (Pasqal, 2025) and QuEra (Aquila) (QuEra, 2022) are the leading companies.

In addition, we can take a look at other technologies such as topoconductors, developed by Microsoft (Majorana1) (Aasen et al., 2025), quantum dots qubits, where we find Quobly (Quobly, 2024) and Diraq (Diraq, 2024), and the more different approach developed by D-Wave with its Advantage2 Prototype (Systems, 2024b).

3 PROPOSAL

The proposed software architecture in Figure 1 implements a multilevel hierarchical approach for efficient integration of quantum and classical systems. In the top layer, the Circuit Generator provides a specialized interface for the generation and manipulation of quantum circuits, establishing an entry point for the development of quantum algorithms. The second layer of the architecture implements three essential components: specialized Application Programming Interfaces (APIs) for systematic interaction, mathematical and scientific libraries that provide fundamental functions for quantum computing, and a robust set of performance and debugging tools.

The Noisy Intermediate-Scale Quantum (NISQ) systems calibration layer is a mandatory component that addresses the challenges inherent in NISQ. This layer implements two essential components: "gate noise" and "qubit noise", facilitating the incorporation of specific noise models that allow accurate calibration of the emulated hardware characteristics with

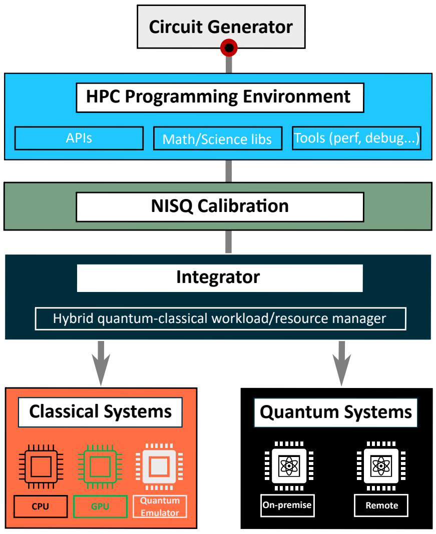

Figure 1: Heterogeneous software architecture for the integration of HPC and Quantum Computing resources.

the target QPU (Bravo-Montes et al., 2024).

The integration layer forms the central component of the system by implementing a dedicated hybridization node that acts as the main orchestrator. This component manages the efficient distribution of tasks between classical and quantum systems, optimizes the use of resources, and ensures optimal performance in hybrid computing environments. The dynamic adaptive capability of this integrator enables the efficient management of available computational resources while implementing robust security protocols that guarantee the integrity and confidentiality of operations throughout the task distribution and execution process.

This software architecture culminates in a lower layer that implements two complementary execution systems that operate synergistically. The Classic Systems integrate CPUs for traditional processing, GPUs for computational acceleration, and a quantum emulator for algorithm emulation and optimization. In parallel, Quantum Systems incorporate both on-premises and remote QPUs, thereby providing flexible access to real quantum hardware.

The proposed software architecture facilitates the adaptive and efficient management of classical-quantum computational resources, allowing the system to dynamically adjust to the specific requirements of each quantum emulation or execution job.

3.1 Heterogeneous Ecosystem Architectures

Eviden (Eviden, 2025a) has developed a heterogeneous software architecture specifically designed to integrate quantum computing into HPC infrastructures. Its design enables the implementation of a complete and transparent workflow, covering all stages from algorithm development to execution on real hardware, whether through emulators or QPUs.

The architecture is structured into interconnected layers, each performing specific functions in managing quantum jobs, including authentication, submission, scheduling, execution, and result retrieval. Regardless of the specific infrastructure utilized, the workflow in Eviden's quantum environments follows a general framework that ensures interoperability and optimization in algorithm execution.

The process begins with user authentication in myQLM (Eviden, 2025b), an open-source framework designed for the development and optimization of quantum algorithms in a flexible environment. During this stage, Qaptiva Access (Eviden, 2024) manages authentication through a robust client-server architecture, enabling access via the myQLM client or through JupyterLab. This system ensures a secure multi-user environment, capable of managing concurrent sessions and integrating external users via authentication mechanisms that preserve the integrity and confidentiality of operations.

Once authenticated, users can submit jobs from myQLM to Qaptiva (Eviden, 2024) through Qaptiva Access. During this stage, the system performs an initial validation of the job to ensure its compatibility with the execution infrastructure and compliance with computational requirements. Depending on the complexity of the algorithm, the job may require optimization and adaptation to the underlying architecture. In this context, Qaptiva Access provides compatibility with external compilers, allowing the compilation process to be adapted according to the characteristics of the target hardware.

The scheduler system, such as Simple Linux Utility for Resource Management (SLURM) (SchedMD, 2025), allocates jobs to available computational resources. This mechanism facilitates efficient workload distribution, optimizing resource utilization across quantum emulators, hybrid infrastructures or physical QPUs.

When a job requires execution on a real quantum processor, the QPU integration layer is activated. This component enables connections to quantum hardware, whether on-premise or remote, through secure communication channels. To connect to third-party

QPUs, a dedicated front-end is necessary to act as a translator between the user software and the physical architecture of the quantum processor. During this process, quantum circuits are optimized to adapt to the hardware topology, adjusting quantum gates to the set of native operations and applying specific calibration parameters if necessary.

The workflow concludes with result collection and analysis. The data generated by the QPU undergoes an initial processing stage by the front-end, which verifies its integrity before sending it to Qaptiva for post-processing. At this stage, advanced statistical analysis techniques and necessary transformations can be applied to facilitate interpretation. Finally, the results are delivered to myQLM, where users can perform detailed interpretations, evaluate performance metrics and explore potential improvements to their algorithms.

The Très Grand Centre de Calcul (TGCC) (CEA TGCC, 2025) and the Jülich Supercomputing Centre (JSC) (Jülich, 2025) are two leading European infrastructures that have adopted a heterogeneous ecosystem similar to the one under discussion for the integration of quantum computing with HPC centers. A detailed description of these infrastructures is provided below.

3.1.1 Très Grand Centre de Calcul (TGCC)

The TGCC is one of France's leading supercomputing infrastructures, capable of hosting petaflop-scale supercomputers. Initially conceived to host Curie, the first petaflop-scale machine in France, TGCC has evolved with the incorporation of advanced systems such as Joliot-Curie (CEA, 2025b), Cobalt, and Topaze, which support scientific and industrial applications through the Centre de Calcul Recherche et Technologie (CCRT) of the Commissariat à l'Énergie Atomique (CEA) (CEA, 2025a).

In the field of quantum computing, TGCC has implemented a preconfigured software ecosystem within the "ccc-quantum container" image. This configuration integrates myQLM, specialized libraries such as Pulser and Perceval, Qaptiva Access, Qaptiva, and a communication front-end for QPUs. Figure 2 illustrates the complete workflow of this architecture, highlighting the interactions between users, CEA, and Pasqal's quantum hardware.

From the user's perspective, interaction with TGCC's quantum environment begins in myQLM or through specialized libraries. Pulser (Pulser Development Team, 2022), developed by Pasqal, allows the design and simulation of laser pulse sequences for controlling neutral-atoms in both analog and digital quantum computing. Perceval (Quandela, 2025),

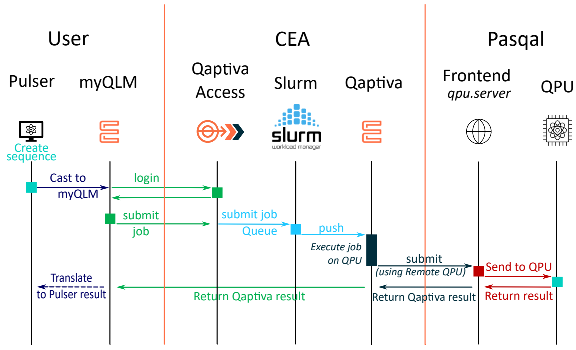

Figure 2: Software architecture deployed at TGCC.

developed by Quandela, provides tools for emulating photonic quantum computing, enabling the modeling of linear optical circuits with components such as beam splitters and phase modulators.

Once the quantum algorithm has been designed, the user can choose between different execution strategies. For small-scale emulations, execution can be performed on a TGCC computing node using myQLM. If greater computational capacity is required, the job is sent via Qaptiva Access to Qaptiva, which allows the emulation of a higher number of qubits and the inclusion of quantum noise models. If the user wishes to execute the algorithm on a real QPU, the job is transferred to the Pasqal front-end, which translates and optimizes the circuit for execution on a neutral-atom QPU.

Finally, the results are processed and analyzed in Qaptiva before being sent to myQLM, where users can assess algorithm performance and iteratively refine their implementations based on the obtained data.

3.1.2 Jülich Supercomputing Centre (JSC)

The JSC has an advanced infrastructure for the integration of quantum and classical computing through the Jülich UNified Infrastructure for Quantum Computing (JUNIQ) (Jülich, 2025). This environment provides access to various quantum technologies via JupyterHub, facilitating the development and experimentation of quantum algorithms within a flexible and efficient framework. JSC hosts multiple quantum systems, including JUQCS (De Raedt et al., 2019), a gate-based quantum computing emulator capable of simulating circuits with up to 43 qubits; Eviden Qaptiva, which enables the emulation of up to 41 qubits and noise modeling; and specialized hardware such as D-Wave Advantage JUPSI (Jülich, 2025) and PASQAL Fresnel. Additionally, projects such as QSolid (QSolid, 2022), focused on the construction of high-fidelity superconducting qubits, and OpenSuperQplus (OpenSuperQPlus, 2023), dedicated to developing a superconducting qubit-based quantum computer, are currently under development.

Figure 3 illustrates the workflow of a hybrid job integrating several of the aforementioned systems within JSC, highlighting the interaction between quantum systems and the classical HPC infrastructure.

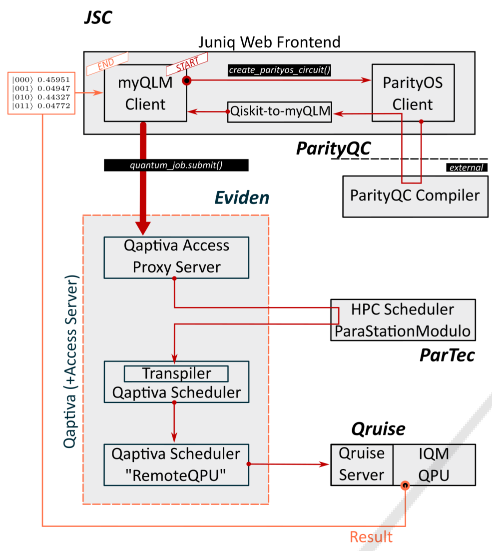

Figure 3: Software architecture deployed at JSC.

The workflow at JSC begins with the development of algorithms in JUNIQ through myQLM. Quantum circuits are optimized using the external compiler ParityQC (ParityQC, 2025), which generates circuits in Qiskit language. These circuits are then converted into myQLM format via an interoperability module included within myQLM. Once transpiled, the circuits are sent to Qaptiva Access, where they are managed by the ParTec Scheduler. This scheduler oversees the transfer of circuits to Qaptiva, optimizing workload distribution across the heterogeneous infrastructure and ensuring that algorithms are executed in the most suitable environment based on their computational characteristics. These circuits may be executed on Qaptiva's emulated QPUs or on a remote QPU through Qruise (QRuise, 2025), which enables the interconnection of Qaptiva with the QPUs provided by IQM.

After execution, the results are processed and visualized in myQLM, allowing for subsequent detailed analysis and iterative optimization of the algorithms.

3.2 Application of Heterogeneous Architecture for the VQE Algorithm

The Variational Quantum Eigensolver (VQE) algorithm is a hybrid quantum-classical approach designed to solve eigenvalue problems in quantum mechanics. Its primary application lies in material simulations and quantum chemistry, where it approximates the ground states of complex quantum systems. The algorithm operates through a parameterized quantum circuit, whose parameters are iteratively optimized using a classical algorithm to minimize the expected value of a given Hamiltonian, thereby estimating the system's energy efficiently.

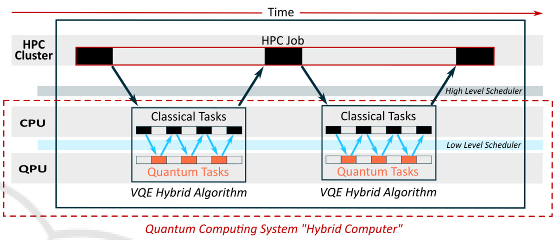

Figure 4: Example for Heterogeneous Quantum-Classical VQE Algorithm.

For efficient execution, it is essential to distribute the computational workload effectively, particularly in heterogeneous environments. Figure 4 illustrates the workflow of a hybrid quantum-classical job within an HPC system, demonstrating how classical and quantum tasks seamlessly interleave to achieve an optimal balance of computational resources. This scalable architecture facilitates the execution of large-scale hybrid jobs, driving advancements in the simulation of complex quantum systems with high computational demands. The process follows three main stages:

1. Connection to the HPC Cluster: The process begins with this step, establishing a link between computational resources and hybrid job management tools. This connection is crucial for coordinating the distributed execution of tasks and ensuring efficient communication across different computational levels. Resource allocation within the cluster allows both classical and quantum computation to run in parallel, optimizing the overall system performance.

2. Work Distribution Between CPU and QPU: Once the connection is established, the workload is distributed between the CPU and the QPU. The CPU handles parameter optimization using numerical algorithms, while the QPU performs quantum calculations by evaluating the expected value of the Hamiltonian for the current parameters. Depending on the available infrastructure, the QPU may be a quantum emulator or a remote quantum device, introducing challenges in latency management and response times within the hybrid system.

3. Execution of the VQE Algorithm: The process starts with an initial set of parameters, which are iteratively adjusted in the CPU based on measurements obtained from the QPU. In each iteration, the QPU uses the updated parameters to compute the expected energy of the system. This feedback cycle continues until convergence is reached, meaning the variation in expected energy falls below a defined threshold. Convergence ensures that a good approximation of the quantum system's ground state has been found.

To enhance efficiency, execution planning occurs at two levels. A high-level scheduler oversees job execution within the HPC cluster, managing resource access and workload distribution. Meanwhile, a low-level scheduler optimizes communication between the CPU and QPU, reducing data transfer latency and improving computational efficiency.

4 CONCLUSIONS

The integration and orchestration of quantum-classical components in heterogeneous systems represent a significant advancement in the development of quantum computing, particularly in HPC environments. As analyzed throughout this article, numerous companies and institutions have recognized the potential of heterogeneous ecosystems, where CPUs, GPUs, and QPUs work together to tackle complex problems in optimization, material simulation, drug development, cryptography, and more.

In this context, we have presented two software architectures implemented in leading European supercomputing centers, TGCC and JSC, where quantum-classical integration is based on the principles outlined in this study. The versatility of this architecture allows compatibility with a wide range of technologies and the incorporation of various QPUs through the creation of specific front-ends for their access. Additionally, its hardware-agnostic approach enables interoperability with different compilers and alternative schedulers to SLURM, optimizing computational efficiency based on the specific requirements of each application.

The infrastructure implemented at TGCC and JSC demonstrates that quantum-classical computing is not only viable but also constitutes a scalable solution adaptable to technological advancements. Through tools like Qaptiva Access, the integration of quantum computing into classical supercomputing environments is facilitated, enabling the efficient and flexible development of hybrid algorithms. This convergence between classical and quantum computing not only accelerates the exploration of new applications but also contributes to the advancement of applied quantum computing, laying the foundation for broader adoption of these systems in the future.

REFERENCES

Aasen, D., Aghaee, M., Alam, Z., Andrzejczuk, M., Antipov, A., Astafev, M., and et al. (2025). Roadmap to fault tolerant quantum computation using topological qubit arrays. *arXiv*.

AbuGhanem, M. (2024). Ibm quantum computers: Evolution, performance, and future directions. https://arxiv.org/abs/2410.00916.

AI, G. Q. (2024). Cirq: Quantum circuit simulation. https://quantumai.google/cirq/simulate/simulation. Accessed: Apr. 01, 2025.

Alice & Bob Quantum Computing (2024). Alice & bob: Quantum computing technology. https://alice-bob.com/quantum-computing/technology/. Accessed: Apr. 01, 2025.

(AQT), A. Q. T. (2024). Marmot - the first commercial 19-inch rack-mounted quantum computer. https://www.aqt.eu/products/marmot/. Accessed: Apr. 01, 2025.

Barcelona Supercomputing Center (2025). Marenostrum supercomputer. https://www.bsc.es/es/marenostrum/marenostrum. Accessed: Apr. 01, 2025.

Beck, T., Baroni, A., Bennink, R., Buchs, G., Pérez, E. A. C., Eisenbach, M., and et al. (2024). Integrating quantum computing resources into scientific hpc ecosystems. *Future Generation Computer Systems*, 161:11–25.

Bidzhiev, K., Wennersteen, A., Beji, M., Dagrada, M., D'Arcangelo, M., Grijalva, S., and et al. (2023). Cloud on-demand emulation of quantum dynamics with tensor networks. *arXiv*.

Bravo-Montes, J., Bastante, M., Botella, G., del Barrio, A., and García-Herrero, F. (2024). A methodology to select and adjust quantum noise models through emulators: benchmarking against real backends. *EPJ Quantum Technology*, 11(1):71.

CEA (2025a). Centre de calcul recherche et technologie. https://www-ccrt.cea.fr/. Accessed: Apr. 03, 2025.

CEA (2025b). Joliot-curie supercomputer. https://www-hpc.cea.fr/en/Joliot-Curie.html. Accessed: Apr. 03, 2025.

CEA TGCC (2025). Très grand centre de calcul (tgcc). https://hpc.cea.fr/tgcc-public/en/html/tgcc-public.html. Accessed: Apr. 02, 2025.

De Raedt, H., Jin, F., Willsch, D., Willsch, M., Yoshioka, N., Ito, N., and et al. (2019). Massively parallel quantum computer simulator, eleven years later. *Computer Physics Communications*, 237:47–61.

Diraq (2024). Diraq. https://diraq.com/. Accessed: Apr. 01, 2025.

Eni S.p.A. (2025). Supercomputing and artificial intelligence at eni. https://www.eni.com/en-IT/actions/energy-transition-technologies/supercomputing-artificial-intelligence/supercomputer.html. Accessed: Apr. 01, 2025.

Eviden (2024). Qaptiva 800 range. https://eviden.com/wp-content/uploads/2024/09/Eviden-Qaptiva-800-Quantum-Computing-brochure-en.pdf. Accessed Apr. 03, 2025.

Eviden (2025a). Eviden website. https://eviden.com/. Accessed Apr. 03, 2025.

Eviden (2025b). myqlm documentation. https://myqlm.github.io/index.html. Accessed: Apr. 02, 2025.

Forcer, T., Hey, T., Ross, D., and Smith, P. (2002). Superposition, entanglement and quantum computation. *Quantum Information & Computation*, 2.

Gao, Y. and Zhang, P. (2016). A survey of homogeneous and heterogeneous system architectures in high performance computing. In *Proceedings of the IEEE International Conference on Smart Cloud (SmartCloud)*, pages 170–175.

Georgopoulos, K., Emary, C., and Zuliani, P. (2021). Modeling and simulating the noisy behavior of near-term quantum computers. *Physical Review A*, 104(6).

Gill, S. S., Cetinkaya, O., Marrone, S., Claudino, D., Haunschild, D., Schlote, L., and et al. (2024). Quantum computing: Vision and challenges. *arXiv*.

Hewlett Packard Enterprise (2025). Lawrence livermore national laboratory. https://www.hpe.com/es/es/compute/hpc/lawrence-livermore-national-laboratory.html. Accessed: Apr. 01, 2025.

Humble, T. S., McCaskey, A., Lyakh, D. I., Gowrishankar, M., Frisch, A., and Monz, T. (2021). Quantum computers for high-performance computing. *IEEE Micro*, 41(5):15–23.

Insider, T. Q. (2025). Public annual report 2025. *The Quantum Insider*. Accessed: May 26, 2025.

IonQ (2022). Ionq forte. https://ionq.com/quantum-systems/forte. Accessed: Apr. 01, 2025.

Jülich, F. (2025). Jülich supercomputing centre (jsc). https://www.fz-juelich.de/en/ias/jsc. Accessed: Apr. 03, 2025.

LaRose, R. (2018). Distributed memory techniques for classical simulation of quantum circuits. https://arxiv.org/abs/1801.01037. Accessed: Apr. 01, 2025.

Linke, N. M., Maslov, D., Roetteler, M., Debnath, S., Figgatt, C., Landsman, K. A., and et al. (2017). Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences*, 114(13):3305–3310.

OpenSuperQPlus (2023). Opensuperqplus. https://opensuperqplus.eu/. Accessed: Apr. 04, 2025.

ParityQC (2025). Parityos. https://parityqc.com/parityos. Accessed: Apr. 02, 2025.

Park, B.-H., Kim, Y., Kim, B.-D., Hong, T., Kim, S., and Lee, J. (2012). High performance computing: Infrastructure, application, and operation. *Journal of Computing Science and Engineering*, 6:280–286.

Pasqal (2025). Pasqal quantum computing processor brochure. https://www.pasqal.com/wp-content/uploads/2025/03/2503_Pasqal_Quantum-Computing-Processor_Borchure.pdf. Accessed: Apr. 02, 2025.

PsiQuantum (2025). Psiquantum announces omega, a manufacturable chipset for photonic quantum computing. https://www.psiquantum.com/featured-news/omega. Accessed: Apr. 01, 2025.

Pulser Development Team (2022). Pulser documentation. https://pulser.readthedocs.io/en/stable/index.html. Accessed: Apr. 02, 2025.

QRuise (2025). Qruise os. https://www.qruise.com/products/qruise-os. Accessed: Apr. 02, 2025.

QSolid (2022). Qsolid. https://www.q-solid.de/. Accessed: Apr. 04, 2025.

Quandela (2025). Perceval documentation. https://perceval.quandela.net/docs/v0.13/index.html. Accessed: Apr. 02, 2025.

Quantinuum (2024). Quantinuum system model h2 product data sheet. https://docs.quantinuum.com/systems/data_sheets/Quantinuum%20H2%20Product%20Data%20Sheet.pdf. Accessed: Apr. 02, 2025.

Quantum, I. (2024). Qiskit aersimulator. https://docs.quantum.ibm.com/api/qiskit/0.30/qiskit.providers.aer.AerSimulator. Accessed: Apr. 01, 2025.

QuEra (2022). Aquila, our 256-qubit analog quantum computer. https://www.quera.com/aquila. Accessed: Apr. 01, 2025.

Quobly (2024). Vlsi: Our path to scaling up silicon spin qubits. https://quobly.io/vlsi-our-path-to-scaling-up-silicon-spin-qubits/. Accessed: Apr. 01, 2025.

Research, G. (2023). Google unveils willow quantum chip. https://blog.google/technology/research/google-willow-quantum-chip/. Accessed: Apr. 01, 2025.

Rigetti (2024). Quantum computing. solved. https://www.rigetti.com/. Accessed: Apr. 01, 2025.

Romero, J. and Milburn, G. (2024). Photonic quantum computing. *arXiv*.

SchedMD (2025). Slurm workload manager documentation. https://slurm.schedmd.com/documentation.html. Accessed: Apr. 04, 2025.

Systems, D.-W. (2024a). D-wave quantum: Quantum computing solutions. https://www.dwavequantum.com/. Accessed: Apr. 01, 2025.

Systems, D.-W. (2024b). Per-qpu solver properties and schedules: Advantage2_prototype2.6. https://docs.dwavequantum.com/en/latest/quantum_research/solver_properties_specific.html#advantage2-prototype2-6. Accessed: Apr. 02, 2025.

TOP500 Project (2024). Top500 list - november 2024. https://top500.org/lists/top500/2024/11/. Accessed: Apr. 01, 2025.

Xanadu (2024). X-series quantum computers. https://www.xanadu.ai/products/x-series. Accessed: Apr. 01, 2025.