National Research Journal of Information Technology & Information Science
Volume No: 13, (January) Year: 2026 (Special Issue)
PP: 723-733
ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

QUANTUM COMPUTING MEETS CLOUD: INTEGRATING AND ENHANCING HIGH-PERFORMANCE COMPUTING APPLICATIONS

Sonali Banyal

Dept. of Computer Science and Engineering, Chandigarh University Mohali, Punjab, India

Rishabh Sharma

Dept. of Computer Science and Engineering, Chandigarh University Mohali, Punjab, India

ABSTRACT

A revolutionary change in the world of computation- ally intensive applications is represented by the mainstreaming of quantum computers into services provided by the cloud. This paper explores the synergy between quantum computing and cloud infrastructures, emphasizing how their convergence can address the computational demands of modern industries. We present an in-depth analysis of existing frameworks, ar- chitectures, and methodologies enabling this fusion, pointing up important issues such data transmission latencies, error correction, and limitations of quantum technology. Furthermore, we provide a brand-new hybrid integration model that max- imizes task distribution among cloud-based conventional and quantum resources, guaranteeing smooth operation and lower overhead. Quantum computing, or QC, has enormous potential to further scientific research in fields including neural networks, optimization, as quantum chemical science. The external noise present in the current noisy intermediate-scale quantum (NISQ) era, however, continues to pose serious problems to QC. The integration of QC as a computational accelerator in traditional high-performance computing (HPC) systems is examined in this work. Using a variety of simulators and quantum hardware, we provide a hardware-neutral framework to use QC to improve the abilities for conventional HPC. Using the Department of Energy's (DOE) lifetime management skills plus leveraging Oak Ridge National Laboratory's (ORNL) HPC expertise, our strategy focuses on strategically integrating QC acceleration into current HPC processes. For the DOE and the Office of Naval Research missions, this entails thorough assessments, benchmarking, and code optimization. The framework creates a unified environment that facilitates research on both quantum and conventional computing by integrating hardware, software, processes, and user interfaces. By bridging QC and HPC, this paper aims to open new computational opportunities, driving scientific innovation and enabling groundbreaking advancements across various research domains.

Index Terms—Quantum-Cloud Integration, Quantum Algo- rithms, Cloud Infrastructure, Hybrid Computing Models, Scal- able Computing.

I. INTRODUCTION

A significant advancement in the development of high- performance computing applications is the incorporation of quantum technology with cloud services. Quantum comput- ing has revolutionary promise in fields like machine learn- ing, cryptography, optimization, and large-scale simulations because it can solve complicated problems tenfold quicker than conventional computers. Meanwhile, cloud computing provides scalable, accessible, and cost-effective infrastructure that has democratized access to computational resources. The convergence of these two technologies holds the promise of creating a new era of computational efficiency, enabling organizations to tackle challenges that were previously deemed intractable.Notwithstanding its enormous promise, quantum computing has some drawbacks, such as a shortage of qubits, error-proneness, and the requirement for extremely special- ized surroundings. [12-13] These challenges have traditionally hindered its adoption at scale. However, the cloud serves as a natural medium to overcome these limitations by offering a distributed platform capable of hosting quantum resources alongside classical systems.This paper explores the framework for integrating quantum computing into cloud services, empha- sizing its implications for HPC applications. We analyze exist- ing technologies, architectures, and methodologies that facil- itate this integration, highlighting how hybrid quantum-cloud systems can optimize computational workflows by intelligently distributing tasks between classical and quantum resources. High-performance computing (HPC) has seen a revolution- ary breakthrough with the introduction of online computing and quantum computing. . [1]As scientific challenges grow increasingly complex, traditional HPC systems face limitations in solving problems related to optimization, cryptography, material science, and machine learning. Quantum computing (QC) introduces a new computational paradigm capable of ad- dressing these limitations through its unique ability to process information in fundamentally different ways.However, because of the early stages of quantum hardware development and its operational limitations, including noise and error rates in the current noisy intermediate-scale quantum (NISQ) era, integrat- ing QC into current HPC processes is still difficult. Many of these issues may be effectively resolved by using cloud com- puting, which offers scalable, on-demand access to quantum resources. With cloud-based quantum services, organizations can bypass the prohibitive costs of building and maintain- ing quantum infrastructure, instead leveraging pay-as-you- go models to access quantum processors and simulators.[3] This

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

democratization of quantum technology, coupled with the high flexibility of cloud platforms, enables researchers and enterprises to explore QC's potential in tandem with classical HPC systems. Cloud-based solutions also facilitate seamless integration by offering hybrid computing environments, where quantum and classical resources can collaborate to enhance computational performance. This paper explores the synergy between quantum computing and cloud technology, focusing on their integration within the HPC landscape. It presents strategies for incorporating quantum accelerators into cloud-enabled HPC systems,[7] including workflow optimization, resource allocation, and software adaptation. By examining real-world applications and use cases, we aim to demonstrate how this convergence can unlock unprecedented capabilities for scientific discovery and innovation. With a framework that emphasizes scalability, accessibility, and efficiency, the union of quantum computing and cloud technology has the potential to redefine the future of high-performance computing.

II. OBJECTIVE OF THE PAPER

Investigating the integration of quantum computing with cloud-enabled high-performance computing (HPC) systems to improve computational capabilities across many scientific and industrial areas is the major goal of this work. Specifically, the paper aims to:

1. ***Analyze Integration Strategies****:* Explore methodologies for incorporating quantum accelerators into existing HPC workflows, leveraging the scalability and flexibility of cloud platforms.
2. ***Optimize Hybrid Workflows****:* Develop and evaluate techniques for seamless collaboration between quantum and classical resources, focusing on efficiency, performance, and resource allocation.
3. ***Assess Real-World Applications****:* Identify and assess use cases where quantum-cloud-HPC integration can provide sig- nificant advantages, such as in optimization, machine learning, and material science.
4. ***Propose a Scalable Framework****:* Design a hardware- agnostic framework that simplifies the adoption of quantum computing within cloud-based HPC environments while ad- dressing challenges such as noise and error correction.

III. RELATED WORK

Quantum computing's integration with cloud-enabled HPC has shown promise for solving optimization problems, enhancing cryptography, and advancing quantum simulations. Platforms like IBM Quantum and AWS Braket provide access to quantum hardware and simulators, while tools like Google Cirq and Pennylane facilitate hybrid quantum-classical workflows.[5-8] Research initiatives at institutions like Oak Ridge National Laboratory (ORNL) focus on combining quantum and classical resources to maximize utility and perfor- mance.Previous research emphasizes issues including program compatibility, hardware constraints, and noise. By putting out a thorough methodology to deal with these problems and further the incorporation of quantum computing into cloud-based HPC systems, this study expands on earlier research.

IV. LITERATURE REVIEW

Recent study has focused upon integrating cloud-enabled supercomputers with quantum computing. The possibility of quantum computing as a revolutionary tool for solving compu- tationally demanding issues has been investigated in a number of research. For example, quantum algorithms like Grover's and Shor's have shown theoretical benefits in uncontrolled information and encrypting analysis searches by **Sharma, H. (2019) [1]**. The potential of quantum systems to supplement traditional HPC in addressing certain problems is highlighted by these developments. Cloud platforms like Google Cloud, IBM Quantum, and Microsoft Azure have played a pivotal role in democratizing access to quantum resources. By offering Infrastructure as a Service (IaaS) for quantum devices, these platforms enable researchers to experiment with quantum algorithms and hybrid workflows without significant upfront investment by **Gathu, S. (2024) [2]** . Academic and industrial collaborations have further advanced the field, focusing on error mitigation techniques and the development of quantum programming environments like Qiskit and Cirq. Additionally, several studies have highlighted the importance of hybrid quantum-classical approaches in optimizing HPC workflows. Researchers have demonstrated that quantum accelerators can improve the efficiency of classical systems in tasks such as material simulation, portfolio optimization, And developing models for machine learning. Notwithstanding these develop- ments, there are still issues, such as the requirement for reliable error correction systems, scalable quantum hardware, and effective integration frameworks by **Padmanaban, H. (2024) [2-3]**,We will prioritize science driver applications that best fit DOE and ORNL's objectives is a component of the mission needs and alternatives study. Energy, earth, materials, and computational sciences, as well as state-of-the-art supercom- puter development research, are some of these drives. We will connect the benefits and drawbacks of each hardware choice to the different domain research applications as a component of the process. The ideal pairings for the direct next procurements may be evaluated with the aid of this mapping. The creation of innovative revolutionary hardware is currently and probably will remain driven by the private sector. This paper's goal is to maximize the academic effect generated by such hardware by devising a plan to ensure its smooth integration to our leading HPC systems. The paper is organized as follows by **Mzukwa, A. (2024) [12-13]**. Following a review of the state-of-the-art in QC/HPC integration, we go over our goals for the proposed work. Overview talks on users, training,

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

applications, hardware, and the software ecosystem come next. The paper's main topic, a comprehensive framework for the integration that describes the integration space, use patterns, and integration models, is presented after the background and motivational talks. A brief review of the integration effort follows. There is discussion of the procedures required to develop a very effective end-to-end integration approach. It offers an initial synopsis of the overall structure of the framework. Through the identification of significant advancements and gaps in the subject, this literature evaluation serves as a basis for the suggested framework. This study attempts to contribute to the continuous development of quantum-cloud-HPC integration by filling up these gaps.

V. QUANTUM COMPUTING

Cloud computing has transformed the delivery of soft- ware and IT infrastructure by offering them as subscription- based services available online using a pay-as-you-go basis. This approach has provided significant advantages, allowing businesses to scale their operations globally with ease while fostering progress in scientific research. This section provides a concise overview of cloud computing models, including the recent rise of the serverless computing paradigm.

A. Cloud computing models

Cloud computing may be broadly divided into three types: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Each of these approaches are designed to fulfill a variety of technical and business needs. With IaaS, organizations can extend their infrastructure on demand and eliminate the need for real hardware by accessing virtual resources like servers, storage, and networking via the internet. PaaS gives programmers a platform to create, test, and launch apps. , [4] offering tools, frameworks, and services without the complexity of managing the infrastructure. SaaS is perfect for end-users since it offers prepared-to-use software products which reside on the cloud, accessible through web browsers, have no need for installation and management. When combined, these models offer scalability, flexibility, and cost-effectiveness, enabling businesses to implement cloud technologies that are customized to meet their unique require- ments. In the field of cloud computing, these vary among service deployment models:

1) ***Service Models****:* Cloud computing operates through sev- eral service models, each designed to cater to different needs and levels of control. The primary service models include Infrastructure as a Service (IaaS), which provides virtualized computing resources like servers and storage; Platform as a Service (PaaS), which offers a framework for developers to build and deploy applications without managing the under- lying infrastructure; and Software as a Service (SaaS), which delivers fully functional applications over the Internet, such as email or office tools.[4] These models enable users to select services that align with their specific requirements, ranging from basic infrastructure to fully managed software solutions.

2) ***Deployment Models****:* Deployment methods in cloud computing explain how cloud resources are managed and made available. While public clouds are shared by many users and offer services via the Internet, private clouds are dedicated to a single organization and provide better security and control; hybrid clouds combine public and private clouds to offer greater flexibility and optimization; and community clouds are shared by organizations with similar needs and concerns, such as security or compliance. The four main deployment models are as follows. [3-4] Each deployment model offers distinct advantages and is selected based on an organization's goals, resources, and regulatory requirements.

B. From Serverful to Serverless computing model

A paradigm change in the use of cloud computing has occurred with the transition form serverful to serverless com- puting. In traditional serverful models, users are responsible for managing servers and computational resources, which involves setup, maintenance, and scaling. While this approach provides control, it also introduces complexity. On the opposite hand, serverless computing allows developers to concentrate only on building and distributing code by abstracting server administration. By charging just for actual consumption and scaling automatically in response to demand, resources are distributed dynamically, minimizing costs.[5] This shift prior- itizes development productivity above managing infrastructure and improving workflows, cutting costs, while promoting effective utilization of resources.

Serverless computing is commonly implemented in two paradigms:

1) ***Function as a Service (FaaS)****:* FaaS makes it possible to run code in reaction to particular triggers or events, usually via APIs. The cloud provider takes care of the supporting infrastructure on its own. Commercial FaaS systems include Microsoft Azure Functions, Google Cloud Functions, and AWS Lambda. Furthermore, a lot of people use open-source FaaS frameworks like Knative, OpenWhisk, and OpenFaaS.

2) ***Backend as a Service (BaaS)****:* Designers may include pre-built backend services from BaaS into their applications, such as handling databases, user authentication, and push alerts. Azure from Microsoft Mobile Apps, AWS Cell Hub, Amazon Amplify, and Firebase are well-known BaaS systems.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

VI. EVOLUTION OF CLOUD COMPUTING FOR HPC

Cloud computing's development for High-Performance Computing is indicative of a gradual shift in the way computationally demanding jobs are carried out. Initially, HPC relied on traditional on-premises supercomputers and dedicated clusters, which required significant capital investment, specialized infrastructure, and extensive maintenance. These systems provided high-speed processing power but lacked flexibility and scalability. With the advent of cloud computing, HPC began to transition towards more dynamic and accessible models.[12-13] Early cloud platforms focused on providing virtualized environments for general-purpose workloads, but as technology advanced, cloud providers began tailoring their services to meet the particular requirements of HPC users.

A. HPC and QC Hardware

High-Performance Computing systems are renowned for their versatility in addressing a wide array of scientific challenges. These systems excel at handling extensive simulations, carrying out sophisticated data analytics, and efficiently carry- ing out challenging mathematical calculations. By using quan- tum algorithms that might outperform their classical coun- terparts in addressing particular kinds of problems, including optimization, cryptography, and quantum chemistry, quantum computers, on the other hand, provide a distinct edge. Before being implemented on real quantum devices, HPC platforms can be crucial in modeling, testing, and verifying quantum algorithms. This helps reduce development costs and speed up the innovation cycle. In this regard, the Oak Ridge Leadership Computing Facility (OLCF) is a shining example of a state- of-the-art setting where cutting-edge quantum and computing technologies meet. This section delves into the computational and quantum capabilities offered by the OLCF, shedding light on their transformative potential for modern scientific and industrial applications.

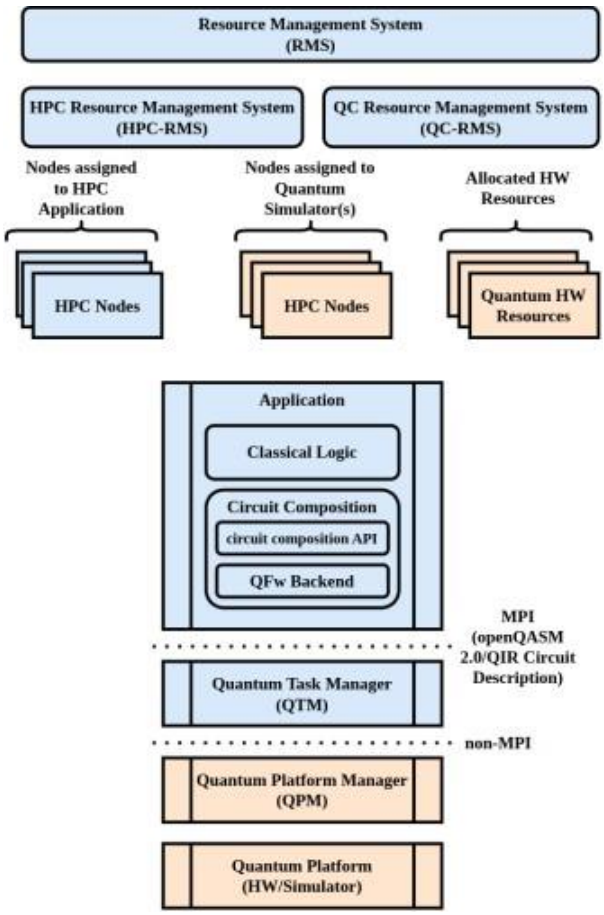
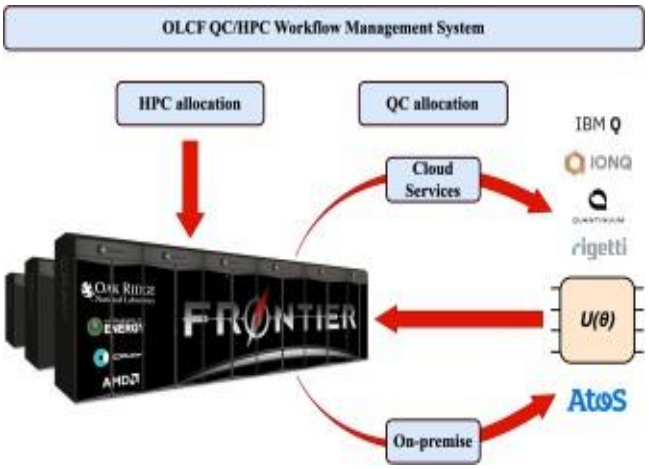


Fig. 1. Integration Framework

1) ***Frontier, Summit, and Advanced Computing Ecosys- tem****:* With an astounding speed of 1.192 exaflops, the Frontier supercomputer from Oak Ridge National Laboratory (ORNL) proudly holds the title of fastest computer in the world and tops the TOP500 list. As the first machine to cross the exascale threshold, Frontier is establishing a new standard for high- performance computing. It offers a tenfold increase in perfor- mance over its predecessor, Summit, with a

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

potential peak of 2 exaflops. 74 HPE Cray EX cabinets, each with more than 9,408 nodes driven by AMD EPYC CPUs and AMD Instinct MI250X GPU accelerators, are part of Frontier's sophisticated architecture. The cutting-edge Slingshot Dragonfly Network, which has a bisection bandwidth of 270 TB/s, connects these parts. [7-10] The cutting-edge Slingshot Dragonfly Network, which has a bisection bandwidth of 270 TB/s, connects these parts. With a peak performance of 200 petaflops, Summit is now placed ninth on the TOP500 list. It has Nvidia V100 GPUs and IBM INFORMATION9 CPUs coupled by a non- blocking fat-tree architecture built on top of Mellanox EDR InfiniBand. Summit has remained a pillar of scientific re- search through its substantial contributions to developments in domains such as materials science, genetics, and artifi- cial intelligence. The Advanced Computing Ecosystem (ACE) testbed, a unique aspect of the OLCF,



complements these supercomputers. This adaptable platform offers a centralized setting for evaluating a range of data and compute workloads across different system architectures.

Fig. 2. Advanced Computing Ecosystem

2) ***Quantum Hardware****:* Although quantum hardware has advanced significantly, it is still in its infancy when compared to traditional computing. Superconducting qubits, trapped ions, silicon spin qubits, photonic qubits, nitrogen vacancy centers, neutral atom qubits, and topological qubits are among the several designs in development. It is difficult to decide which architecture will best meet the DiVincenzo requirements for implementing quantum computing because each one has dis- tinct benefits. The Quantum Computing User Program (QCUP) gives users access to state-of-the-art devices with trapped- ion and superconducting qubits, allowing them to investigate the advantages of these designs for solving challenging com- putational issues. Superconducting qubits, based on electri- cal circuits operating at extremely low temperatures, benefit from similarities to traditional integrated circuit technologies, simplifying design and scalability. Their strong controllability enables two-qubit operations in numerous nanoseconds, even if their coherence durations are just a few hundred microseconds.

3) ***Hardware Integration****:* Numerous quantum computing (QC) architectures are being investigated for integration with high-performance computing (HPC), as was previously men- tioned. Which architecture will provide the required scalability

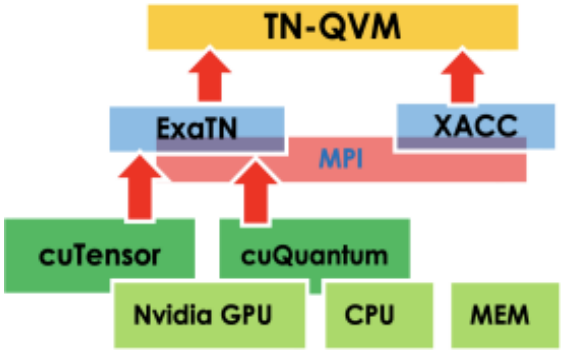


Fig. 3. Integration Flowchat

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

and performance for an on-premise QC/HPC system is still unknown, though. The goal of this project is to find the best solutions by evaluating and comparing several QC/HPC integra- tion prototypes. A number of obstacles must be overcome in order to successfully integrate large-scale superconducting or ion-trap quantum hardware with HPC systems.[10]Achieving high-fidelity qubit management to minimize decoherence, putting in place efficient error correction, and guaranteeing low-latency communication between quantum and classical computers are important concerns. Other challenges include controlling cryogenic temperatures, preserving extremely high vacuum levels, and resolving scaling issues with noise reduc- tion and qubit connectivity.

4) ***Interfaces and Connectivity****:* Quantum processing unit (QPU) integration into high-performance computing (HPC) systems can follow two models: loose and tight integration. Currently, loose integration is the practical approach, where the QPU operates as a separate entity connected to the HPC system via a network. This configuration supports both on-premise and remote QPUs. In contrast, tight integration involves placing the QPU directly on a node alongside CPU and GPU hardware, enabling all components to operate under a unified software system. While the framework in Section 8 accommodates the potential for tight integration, the focus remains on the loose integration model due to its immediate feasibility. In loose integration, communication between HPC hosts and QPU hardware occurs over network interfaces like Infiniband or CXL.

VII. PROPOSED FRAMEWORK FOR QUANTUM-CLOUD INTEGRATION

To achieve seamless integration of quantum computing into cloud platforms, a robust framework must address the technological and architectural challenges while maximizing the benefits of both paradigms. The proposed framework emphasizes modularity, scalability, and interoperability, creating a unified ecosystem for quantum-cloud integration.

Q1: How can cloud and quantum computing be merged effectively?

There is a potent synergy when cloud and quantum comput- ing are combined, leveraging the scalability of cloud platforms alongside the computational potential of quantum systems. To achieve this, several strategies can be implemented. Standardized quantum APIs and SDKs, such as OpenQASM or QIR, can be developed to enable seamless programming of hybrid applications, while middleware can mediate inter- actions between classical and quantum resources, handling tasks like quantum scheduling and error correction. Hybrid cloud models, exemplified by platforms like AWS Braket and Azure Quantum, can integrate quantum processors with classical cloud systems, providing users with flexible access to quantum resources. Virtualized quantum hardware in the cloud can further allow multiple users to share and utilize quantum computing capabilities efficiently. Federated quantum computing frameworks can unify diverse quantum systems, such as superconducting qubits and ion traps, under a single cloud-based infrastructure. Resource scheduling systems can be adapted to co-manage classical and quantum workloads, en- suring optimal utilization of resources. High-speed communi- cation protocols, such as Infiniband or CXL, can be employed to minimize latency, while quantum networks can connect QPUs across data centers for distributed quantum processing. Cloud-based preprocessing pipelines can transform classical data into quantum-compatible formats, enabling seamless data preparation for quantum tasks.

Q2: What possible uses might cloud-integrated quantum computing have?

By enabling applications that take advantage of the spe- cial powers of quantum mechanics and the scalability of cloud platforms, cloud-integrated quantum computing has the potential to completely transform a wide range of sectors. Researchers can create more potent medications by using quantum computers to model molecular interactions at a scale never before possible in drug creation. Quantum algorithms used with cloud platforms can improve fraud detection, risk analysis, and portfolio management in the financial industry. Quantum-enhanced optimization can help logistics and supply chain management by improving scheduling, routing, and resource allocation. Quantum computing has the potential to speed up data processing and machine learning in artificial intelligence, allowing for the quicker training of intricate models and the discovery of patterns in enormous datasets. Cloud-integrated quantum systems can improve cybersecurity and cryptography by facilitating the creation of secure com- munication channels and quantum-safe encryption techniques. Other promising fields where quantum computing can evaluate complex systems and suggest solutions for the distribution of renewable energy and the mitigation of climate change are energy optimization and climate modeling.

A. ***Proposed architecture, requirements and data flow***

Workload management and accessibility are the two main issues that the suggested framework attempts to solve. A serverless FaaS (Functions as a Service) design that pro- vides a language-neutral HTTP interface for data entry makes

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

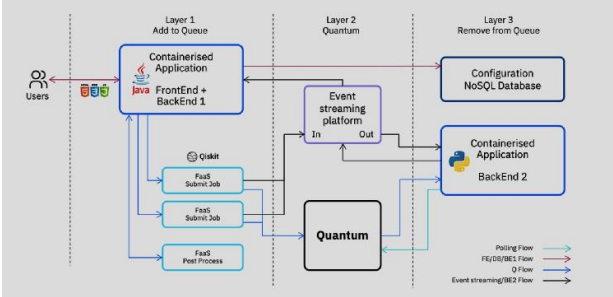


Fig. 4. Architectural overview diagram and flow

the system accessible. This technology offers trigger-based, lightweight computational units that can quickly run brief bursts of code. Among the choices for deployment are single packed executables. (e.g., a **'.jar'** file with a Java runtime) or specialized Docker containers. The framework leverages a Docker container with Python as the foundation image, enhanced with IBM's open-source Qiskit quantum program- ming SDK to improve accessibility. With this configuration, raw data may be converted into quantum circuits via a single HTTP API. Every built quantum method is contained within a specific Docker container, providing a modular, decoupled approach. As a result, deployment and integration are made easier since different teams can separately oversee the FaaS platform and algorithm implementation.

1) ***Scalability Through FaaS and Kafka Integration****:* Scalability is another core requirement. The framework fa- cilitates high scalability by leveraging the FaaS architecture to handle numerous external system requests without perfor- mance degradation. This approach is analogous to job schedul- ing in HPC (High-Performance Computing) architectures. To enhance scalability further, especially in result retrieval, the framework integrates a Kafka-based queue. The queue handles job submission reports and facilitates result retrieval through specific topics. It enables seamless communication between the FaaS component (submission phase) and the results collec- tor component (retrieval phase). High throughput is a critical prerequisite for the queue to manage message traffic efficiently. The backend components, implemented in Java (Backend 1) and Python (Backend 2), work in tandem to process and deliver results. These components are containerized for porta- bility and deployed in Cloud Foundry, leveraging its auto- configuration and user-friendly interfaces for reduced setup effort.

2) ***Application Flow and Architecture****:* The framework's architecture reflects a hybrid quantum-classical approach across three layers:

- **Layer 1 (User Interface):** Users interact via a web- based UI, submitting input (e.g., two emoticons) for quantum processing. A submission triggers job creation and sets the UI as a listener for outcomes.
- **Layer 2 (Quantum Processing):** The IBM Quantum computer and Event Streams are used by backend systems to process the work. This layer ensures efficient job handling andscheduling.
- **Layer 3 (Result Processing):** Once the quantum compu- tation completes, the results are processed by Cloud Foundry backends and relayed to the UI.

3) ***Demonstration: Emoticon Superposition****:* As a demon- stration, users input two emoticons through the web UI, which leverages the quantum superposition property to visualize them. Upon submission, the system queues the job in Layer 1, processes it through the quantum computer in Layer 2, and displays the results in Layer 3. This streamlined process showcases the framework's potential to integrate classical systems with quantum computing seamlessly.

B. FrontEnd

The User Interface (UI) of the suggested framework gathers inputs and displays outputs using a structured data model, asynchronous Ajax calls, and JavaScript functions. It combines JavaScript, CSS, and HTML elements and may be customized using frameworks like React or Bootstrap. With its mobile- first design system, it guarantees cross-browser and cross- device compatibility. The front-end collects data, sends it to BackEnd 1 via WebSocket, and subscribes to retrieve results asynchronously. Results are stored in browser cache using a JSON-based data model, ensuring seamless communication. Customizable structure and graphics enhance flexibility, while core JavaScript functions remain unchanged for integration tasks.

C. BackEnd 1

One Java application running on IBM Cloud Foundry makes up BackEnd 1, which communicates with Cloudant to retrieve configuration settings needed to invoke Cloud Functions. The data model consists of a single collection, config, which stores parameters enabling BackEnd 1 to call Cloud Functions. For instance, BackEnd 1 retrieves data from the config collection and translates it into RESTful API calls, specifying the method (POST), endpoint, authentication, and headers.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

Adding new functionalities involves simply adding another record to the config collection, ensuring extensibility and reusability.

{

"_id":"smile_super_position", "functionHttpMethod": "POST", "functionBackendUrl": "URL", "functionParams": {

"body":"incomingRequestBody", "headers": {

"Authorization": "IAMBearerToken", "Content-Type":"application/json", "Accept": "application/json"

}

}

}

A special Event Streams topic appears when a job is submitted to the quantum computer using Qiskit's quantum

API (e.g., topic-1234) is assigned for user segregation, allowing each client to listen to a specific topic. The BackEnd 1 is developed using Java Spring Boot, providing a production- ready environment with automated configurations and robust third-party library management, ensuring seamless scalability and integration for future expansions.

D. Cloud Functions

Serverless architectures are perfect for integrating quantum computing because they let developers concentrate on business logic rather than runtime setup, deployment, or infrastruc- ture maintenance. The proposed framework uses IBM Cloud Functions to build quantum algorithms and execute circuits on IBM Quantum systems, allowing selection between quantum hardware or simulators.BackEnd 1 uses this serverless "ac- tion" to handle user requests from the FrontEnd. The action, which is written in Python using Qiskit, establishes IBM Quantum jobs, generates quantum circuits depending on input parameters, and then dispatches them for execution. Upon job submission, the action retrieves the job ID, passing it to BackEnd 2 via an Event Streams queue. Parameters include algorithm inputs, backend type (real hardware or simulator), and client/job IDs. Real quantum devices are selected based on qubit requirements, while the "qasm simulator" handles sim- ulations. Errors during job submission or queue transmission return corresponding error messages for resolution.

E. BackEnd 2

Regardless of the IBM Quantum backend (quantum devices or simulators) that is chosen, the BackEnd 2 containerized application is responsible for obtaining quantum job results as soon as they become available. It combines two main elements in its operation:

Kafka Client: enables input and output communication be- tween Cloud Functions, BackEnd 1, and BackEnd 2. **Polling:** uses Event Streams to transmit results back to BackEnd 1 and Qiskit libraries to get task results.

Fig. 5. Example of input JSON

Python's multi-process module is used in BackEnd 2, which is developed in Python and installed on an IBM Cloud Foundry instance, to handle multiple requests at once. Its single-process main function retrieves events from IBM Event Streams, placing them in a process-safe manager pool queue shared among running processes. Workers are triggered by events (JSON messages) from Cloud Functions, containing job details. Each worker monitors assigned jobs until results are ready. This ensures efficient, scalable job result management.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

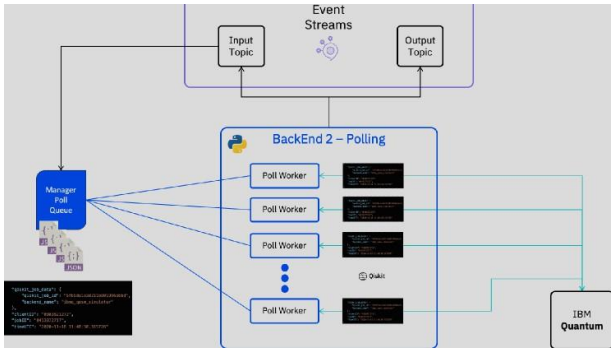


Fig. 6. Polling mechanism overview

F. Kafka Client Component

By enabling asynchronous communication between the quantum and classical systems, the Kafka-based Event Streams component shortens the time it takes to connect to the IBM Quantum online platform. This component comprises two Python classes that manage input and output queues, operating on distinct topics within the same Event Streams instance. The input component processes JSON messages containing call execution and job IDs submitted to IBM Quantum for execution. It can act as a producer to re-insert unprocessed task IDs into the queue if processing time beyond a predetermined threshold, or as a consumer to gather incoming messages and initiate polling through the Polling component. This optimization minimizes polling delays. The output component allows BackEnd 2 to write results or error messages to the queue, making them accessible to BackEnd 1 for post- processing and subsequent forwarding to the front end. This decoupling mechanism ensures efficient communication, data management, and scalability between components.

G. Polling Component

The Polling component incorporates Qiskit dependencies to monitor and manage the progress of quantum jobs. Initialized as poll workers, each worker independently processes events retrieved from a shared manager pool queue. It comprises three modules: PollingIBMQWorker, which handles individual polling tasks; PollingIBMQ, which contains the logic for re-trieving and managing job results, including final state checks (DONE, CANCELED, or ERROR); and QuantumUtils, which manages IBM Quantum cloud authentication and backend connections. For real quantum devices, the Polling component estimates job completion times to prioritize processing. If the estimated time is within a predefined threshold, the worker waits for job completion; otherwise, the event is resubmitted to the input topic. For simulators, jobs are processed immediately due to shorter waiting times. Completed jobs or errors are sent to the output topic, allowing BackEnd 1 to process results efficiently. This strategy improves response times for frontend users while managing multiple requests seamlessly.

Fig. 7. Example of final JSON

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

VIII. **CONCLUSIONS**

The proposed framework delivers a groundbreaking ap- proach to integrating quantum and classical computing, lever- aging modern cloud-based architectures to create a highly scalable, efficient, and user-friendly system. By employing serverless technologies such as IBM Cloud Functions and Kafka-based Event Streams, it decouples the complexities of quantum circuit execution from the classical backend, enabling seamless communication and asynchronous process- ing. This approach significantly reduces latency, enhances resource utilization, and simplifies the integration of additional functionalities in the future. The integration of components like the Kafka Client and Polling modules ensures efficient job management, from submission to result retrieval. These components prioritize tasks based on readiness and reduce waiting times, providing an optimized experience for end- users. The adoption of Qiskit as the quantum programming library aligns the framework with cutting-edge quantum tech- nologies, enabling compatibility with various IBM Quantum backends, including real quantum hardware and simulators. Furthermore, the framework emphasizes error handling and results prioritization strategies to ensure reliable and consistent operations. Its modular design allows the addition of new algorithms by updating configurations, ensuring flexibility for evolving user needs and advancements in quantum comput- ing. Overall, the framework stands as a robust solution that bridges the gap between quantum and classical systems. It not only addresses the challenges of accessibility, scalability, and extensibility but also sets the stage for future innovations in quantum computing applications, empowering organizations to harness the full potential of this transformative technology.

REFERENCES

1. Sharma, H. (2019). HIGH PERFORMANCE COMPUTING IN CLOUD
2. ENVIRONMENT. International Journal of Computer Engineering and Technology, 10(5), 183-210.
3. Gathu, S. (2024). High-Performance Computing and Big Data: Emerging Trends in Advanced Computing Systems for Data-Intensive Applica- tions. Journal of Advanced Computing Systems, 4(8), 22-35.
4. Padmanaban, H. (2024). Quantum Computing and AI in the Cloud. Jour- nal of Computational Intelligence and Robotics, 4(1), 14-32.
5. Humble, T. S., McCaskey, A., Lyakh, D. I., Gowrishankar, M., Frisch, A., Monz, T. (2021). Quantum computers for high-performance com- puting. IEEE Micro, 41(5), 15-23.
6. Garcia-Buendia, N., Mun˜oz-Montoro, A. J., Cortina, R., Maqueira- Mar´ın, J. M., Moyano-Fuentes, J. (2024). Mapping the landscape of quantum computing and high performance computing research over the last decade. IEEE Access.
7. Beck, T., Baroni, A., Bennink, R., Buchs, G., Pe´rez, E. A. C., Eisenbach, M., ... Zimmer, C. (2024). Integrating quantum computing resources into scientific HPC ecosystems. Future Generation Computer Systems, 161, 11-25.
8. Elsharkawy, A., To, X. T. M., Seitz, P., Chen, Y., Stade, Y., Geiger, M., ... Schulz, M. (2023). Integration of quantum accelerators with high performance computing–a review of quantum programming tools. arXiv preprint arXiv:2309.06167.
9. Visalaxi, G., Muthukumaravel, A. (2024). Towards Quantum Computing-Inspired Evolutionary Algorithm for Optimized Cloud Re- source Management. International Journal of Intelligent Engineering Systems, 17(3).
10. Navaux, P. O. A., Lorenzon, A. F., da Silva Serpa, M. (2023). Challenges in high-performance computing. Journal of the Brazilian Computer Society, 29(1), 51-62.
11. Nguyen, H. T., Krishnan, P., Krishnaswamy, D., Usman, M., Buyya, R. (2024). Quantum cloud computing: a review, open problems, and future directions. arXiv preprint arXiv:2404.11420.
12. Dou, M., Zou, T., Fang, Y., Wang, J., Zhao, D., Yu, L., ... Guo, G.
13. (2022). QPanda: high-performance quantum computing framework for multiple application scenarios. arXiv preprint arXiv:2212.14201.
14. Shehata, A., Naughton, T., Suh, I. S. (2024, September). A framework for integrating quantum simulation and high performance computing. In 2024 IEEE International Conference on Quantum Computing and Engineering (QCE) (Vol. 2, pp. 300-305). IEEE.
15. Mzukwa, A. (2024). Exploring Next-Generation Architectures for Ad- vanced Computing Systems: Challenges and Opportunities. Journal of Advanced Computing Systems, 4(6), 9-18.
16. Mzukwa, A. (2024). Exploring Next-Generation Architectures for Ad- vanced Computing Systems: Challenges and Opportunities. Journal of Advanced Computing Systems, 4(6), 9-18.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

17. Marosi, A. C., Farkas, A., Ma´ray, T., Lovas, R. (2023). Toward a Quantum-Science Gateway: A Hybrid Reference Architecture Facili- tating Quantum Computing Capabilities for Cloud Utilization. IEEE Access, 11, 143913-143924.

18. Perri, D., Simonetti, M., Gervasi, O., Tasso, S. (2022). High- performance computing and computational intelligence applications with a multi-chaos perspective. In Multi-Chaos, Fractal and Multi-Fractional Artificial Intelligence of Different Complex Systems (pp. 55-76). Aca- demic Press.

19. Nguyen, H. T., Usman, M., Buyya, R. (2024). iQuantum: A toolkit for modeling and simulation of quantum computing environments. Software: Practice and Experience, 54(6), 1141-1171.

20. Haldorai, A. (2024). Advancements and Applications of Quantum Com- puting in Robotics. Journal of Computing and Natural Science, 053-063.

21. Wang, L., Wang, H. (2024). Big Data in Genomics: Overcoming Chal- lenges Through High-Performance Computing. Computational Molecu- lar Biology, 14.

22. How, M. L., Cheah, S. M. (2023). Business Renaissance: Opportunities and challenges at the dawn of the Quantum Computing Era. Busi- nesses, 3(4), 585-605.

23. Jackson, K. R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., ... Wright, N. J. (2010, November). Performance analysis of high performance computing applications on the amazon web services cloud. In 2010 IEEE second international conference on cloud comput- ing technology and science (pp. 159-168). IEEE.

24. Dornala, R. R., Ponnapalli, S., Sai, K. T., Koteru, S. R. K. R., Koteru,

25. R. R., Koteru, B. (2023, December). Quantum based Fault-Tolerant Load Balancing in Cloud Computing with Quantum Computing. In 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 1153-1160). IEEE.

26. Ramouthar, R., Seker, H. (2023). Hybrid Quantum-Classical Computing-A Fusion of Classical And Quantum Computational Sub- strates.

27. Rallis, K., Liliopoulos, I., Tsipas, E., Varsamis, G. D., Melissourgos, N., Karafyllidis, I. G., ... Dimitrakis, P. (2025). Hardware-level Inter- faces for Hybrid Quantum-Classical Computing Systems. arXiv preprint arXiv:2503.18868.

28. Sudha, S. K., Aji, S. (2024). Exploring the advancements in high- performance computing paradigm for remote sensing big data analyt- ics. Cloud Computing and Data Science, 50-61.