ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 07, July 2026

Quantum Application Framework for Software Applications implementation in Quantum Computer

[1] Dr. Pradheep Kumar .K

[1] Department of CSIS, BITS Pilani
Corresponding Author Email: [1] pradjourn@gmail.com

***Abstract*— In this work a quantum application framework has been proposed to design applications on a quantum computer. The framework is scalable and makes software design more flexible and easily debugable. It also helps in providing reliable results and ensures security in applications. Qubits cannot be replicated and it is not possible for information theft using the same. The quantum application framework reduces processing time and memory consumption by 32% and 34% respectively, compared to conventional Machine learning frameworks.**

***Index Terms*— Microservices, saga, qubits, quantum data plane, quantum control plane.**

I. INTRODUCTION

In today's digital world where most of the applications are digital and data sensitive it becomes essential to design high power systems that are capable huge volumes of data. This becomes essential to facilitate problem solving for several real time Artificial Intelligence applications.

The application software needs to be flexible to incorporate features as required based on necessity. If more complex computations are required it would be essential to incorporate additional services. In such cases, a monolithic design approach would not be favourable, As it would involve large amount of time to re-design the software from scratch.

Also based on the data analytics and volume of data processing the computation power tends to vary based on the underlying application. Also the framework design should provide the user with the privilege to customize the application based on the need with different softwares. It should also provide an option to give the software a different layout and design view. Further the application should have a high processing speed and provide reliable inferences.

The paper is organized as follows: Section 2 discusses the literature review. Section 3 proposes the model for the quantum application framework, Section 4 explains the simulation environment and discusses the results in detail. Section 5 concludes the work and Section 6 explains the future directions of research.

The quantum application framework reduces processing time and memory consumption by 32% and 34% respectively, compared to conventional Machine learning frameworks.

II. LITERATURE REVIEW

Ali Khan et al in [1] proposes a full stack iterative model. It includes an Agnostic coding platform which has an integration and development pipeline. It has cloud computing services to handle scalable requests. It acts as decision making junction. It has execution with distinct paths with quantum code with quantum computing services offered via cloud platform. Nils et al in [2] has explained the quantum circuit design module. The use of quantum gates has been explained in detail. The work also discusses the optimization of the quantum circuits so designed. Patel et al in [3] discusses how quantum software has been designed orchestrating quantum cloud services. The obsfucation using quantum logic gates have been explained using X-gates and the circuit optimization for the same has been discussed. The final synthesis of the circuit has also been derived with required optimisations. The associated evaluation, analysis and methodology have been discussed in detail. Manuel et al in [4] explains the road map and challenges associated with quantum software engineering. It also explains quantum state vectors and noise associated with the data generated. Quantum algorithms appropriate for the quantum data under consideration. The programming paradigms based on complexity of circuits, reusable quantum software and abstractions for the software. The architectural decisions and design patterns with empirical evidence have also been discussed. Zhang et al in [5] has proposed a optimized distributed quantum data center explaining the interconnect between the quantum processing units (QPU). The associated quantum communication protocols with the quantum processing units have been illustrated. The photon transmission rate and the propagation speed have been analysed in detail. Azeem et al in [6] proposed a new genre of computing with different software design modules and their associated test cases. It also explains the different modules and their associated features. Tavassoli et al in [7] illustrates quantum computing as a service for a hybrid platform between classical and quantum computing. It also explains the different features of the framework by illustrating the data decoding. It also shows the different services library. Beardow in [8] shows the supporting computing centred intutitions for quantum computing using a game based

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 07, July 2026

approach. It also explains the nature of games to illustrate the quantum software design principles. The associated methods and their limitations and challenges have also been explained. Espositto et al in [9] explains the different quantum programming language patterns and the optimizer module to bring in all the essential features for a quantum language with their associated storage classes. The roadmap for the design with their associated challenges have also been highlighted. Alexeev et al in [10] have also discussed the different software tools for quantum computing. Some include the numerical optimization and Tensor network in quantum circuits at Exascale computation of datasets. Li et al in [11] explains the 2 broad components of hardware and software which explains the mapping created between the software and hardware. It also explains the associated algorithms to carry out the mapping.

III. PROPOSED WORK

In this work a quantum application framework has been proposed to design applications with flexibility as shown in Figure. 1. The following are the component modules of the framework:

A. Quantum Data Plane:

This module contains a large number of qubits which store the data sets. Each qubit has a tag id, phase angle and the data set. The qubits are grouped based on the data set. When a query is raised by an application the same is being decomposed into query components and each query component is assigned with the relevant qubits

B. Quantum Control Plane:

The quantum control plane has the quantum gates which are required to process the qubits. The choice of the gates depend on the nature of the query. An optimal choice of gates are chosen from the quantum control plane.

There is constant communication between the quantum data plane and quantum control plane.

C. Quantum Scheduler:

The quantum scheduler does a mapping between the qubits and data sets. The qubit utilization is defined by the following expression

$$U(Qb) = \frac{dsval_i}{\sum_{i=1}^{i=n} dsval_i} \quad (1)$$

Where $dsval_i$ is the data set value of a particular data set

The data set value is computed based on the mean square distance from the centroid of the clustered data sets.

Also the sum of the utilizations of qubits should be less than or equal to 1.

$$\sum_{i=1}^{i=n} U(Qb_i) = 1 \quad (2)$$

Where $U(qb_i)$ is the utilization of the qubit

D. Microservices layer:

The microservices layer provides a host of microservices which could be incorporated in your application. Some of them include ML analytics software, Saga a distributed database module, etc

The microservice layer provides the designer with flexibility of choice to choose the required microservice based on the underlying application.

If a particular microservice does not function it raises a notification alert to the user to add an alternative microservice to the application which helps in decoupling and coupling the same to the application. This makes debugging simpler.

E. Architecture layer:

The architecture layer provides interconnection of the microservices. The architecture layer offers 3 basic architecture patterns:

- REST (Representative State Transfer)
- gRPC (Global Remote Procedure Call)
- GraphQL

The REST architecture model is the simplest which connects all modules together based on the states of the event

The gRPC which is used in a client server model where the data resides in a cloud server and data needs to be fetched and stored locally in the system.

The GraphQL model is used when data is being fetched from multiple web sites and then integrated on the main platform.

The architecture layer provides flexibility to the user to choose the appropriate architecture based on the underlying application. Sometimes the application may require 2 different architectures. For example part of the model would use REST and the other part may use Graph QL to fetch the data from several web sites. This flexibility also has been provided by the architecture layer.

F. Orchestration layer:

The Orchestration layer includes the dockers, containers and other encapsulation utilities to package the required services based on the architecture model chosen and automate the process flow of the software.

This layer allows the user to modify the microservices of the software dynamically based on the current requirement and ensure the software caters to the requirements of the user. It serves like a software development kit which uses an integration feature.

G. Application Layer:

The application layer permits the user to customize the layout and theme of the software and provide all appearance features like font size and other animation features to give a unique flavor of choice as desired by the user.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 07, July 2026

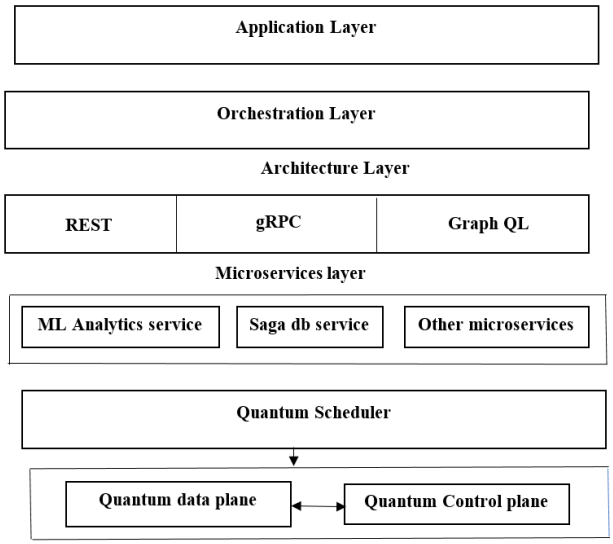

Figure 1. Block diagram for quantum application framework

IV. SIMULATION RESULTS

The enterprise software has been implemented using Qiskit and performance has been assessed for the following 2 metrics

- Processing time
- Memory consumption

The quantum circuit for the application framework has been illustrated in the Figure 2

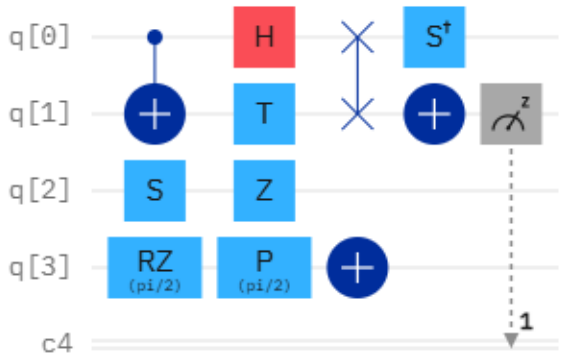

Figure 2. Quantum circuit for the application framework designed

A. Processing Time:

The processing time is defined as the time required for a particular quantum application.

Table 1: Processing time for the quantum applications

	Number of QML applications	Processing time		Reduction (%)
		ML Framework	QML Framework	
	10	14.56	9.48	34.89
	15	17.65	11.46	35.07
	20	19.56	13.45	31.24
	25	21.45	14.34	33.15
	30	23.45	15.45	34.12
	35	26.54	17.65	33.50
	40	28.78	19.54	32.11
	45	31.45	21.56	31.45
	50	35.56	24.65	30.68
	55	37.65	25.76	31.58
	60	39.56	27.56	30.33
	65	41.56	28.65	31.06
	70	43.45	29.65	31.76
	75	45.65	31.45	31.11
	80	47.56	32.65	31.35
	85	49.76	33.56	32.56
	90	51.65	34.56	33.09
	95	53.56	35.76	33.23
	100	54.65	36.76	32.74

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 07, July 2026

	Number of QML applications	Processing time		Reduction (%)
		ML Framework	QML Framework	
Average	55	36.00	24.42	32.37

It could be observed from table 1, Figure 3 and Figure 4 that for an average of 55 quantum applications the quantum framework consumes 36 seconds compared to the conventional machine learning framework which consumes 24 secs, yielding a reduction of 33%.

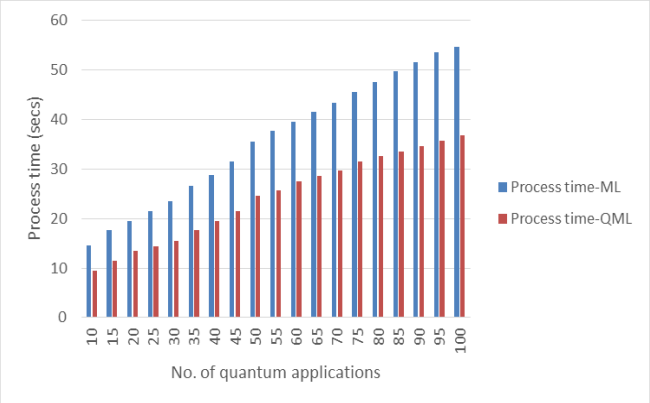

Figure 3. Plot comparing the processing times (ML Application Vs Quantum ML applications)

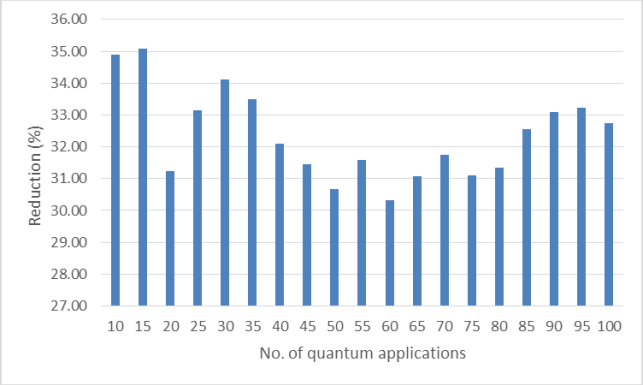

Figure 4. Plot showing reduction in processing time for Quantum ML applications

B. Memory Consumption:

The memory consumption is the amount of memory consumed to run a particular quantum computing application

	Number of QML applications	Memory consumption		Reduction (%)
		ML Framework	QML Framework	
	10	18.65	12.45	33.24
	15	21.56	14.56	32.47
	20	23.45	15.65	33.26
	25	25.76	16.56	35.71
	30	27.56	18.65	32.33
	35	29.56	19.56	33.83
	40	31.34	20.56	34.40
	45	33.56	21.65	35.49
	50	36.65	22.56	38.44
	55	37.56	23.45	37.57
	60	38.56	24.56	36.31
	65	39.65	26.56	33.01
	70	41.45	27.56	33.51
	75	43.56	28.65	34.23
	80	45.65	29.54	35.29
	85	47.65	31.45	34.00
	90	49.67	33.45	32.66
	95	51.56	34.56	32.97
	100	53.56	35.65	33.44
Average	55	36.68	24.09	34.32

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 07, July 2026

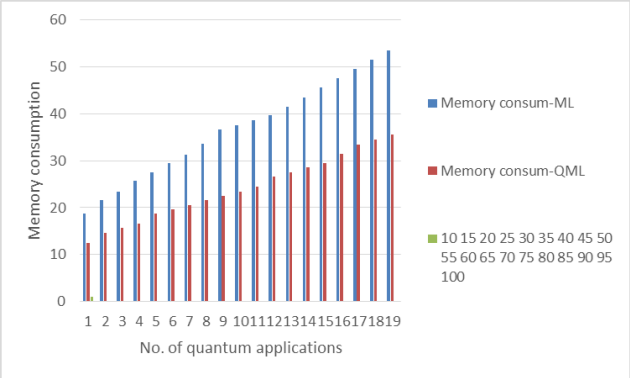

Figure 5. Plot comparing the Memory consumption (ML Application Vs Quantum ML applications)

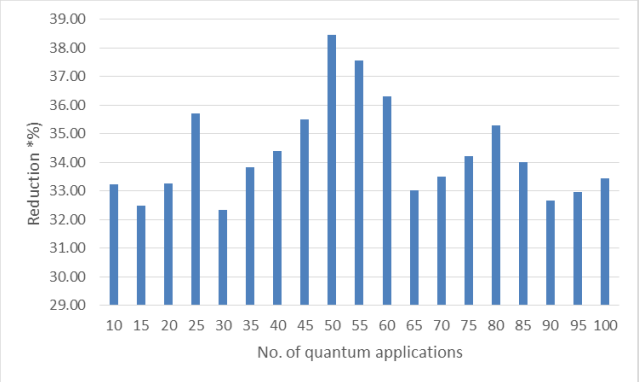

Figure 6. Plot showing reduction in memory consumption for Quantum ML applications

It could be observed from table 2, Figure 5 and Figure 6 that for an average of 55 quantum applications the quantum framework consumes 37 units of memory compared to the conventional machine learning framework which consumes 24 units yielding a reduction of 34%.

V. CONCLUSION

In this paper a quantum application framework has been proposed that reduces processing time and memory consumption significantly. The quantum application framework serves as a platform to design several quantum machine learning applications on a quantum computer and also makes the quantum computer more effective in solving real world machine learning applications.

VI. FUTURE WORK

The quantum application framework could be made light weighted with essential services and made compatible to several IoT platforms like mobile phones and other gadgets which could use the application with lesser computation power.

REFERENCES

[1] Arif Ali Khan, Davide Taibi, Cecile M. Perrault, Muhammed Azeem Akbar. “Advancing Quantum Software Engineering: A vision of hybrid full-stack iterative model”, SAC’25, pp 1444-1448, 2025

[2] Nils quetschlich, Lukas Burgholzer, Robert Wille. “MQT Predictor: Automatic Device Selection with Device-Specific Circuit Compilation for Quantum Computing” pp 10:1-10:26, Vol 6, No.1 Article 10, ACM, 2025.

[3] Tirthak Patel, Aditya Ranjan, Daniel Silver, Harshitta Gandhi, William Cutler, Devesh Tiwari. “Opaque: Program Output Obfuscation for Quantum Software Circuits in Quantum Clouds”, ICS’25, pp 1079-1091, 2025.

[4] Juan Manuel, Murillo, Jose Garcia, et al. “Quantum Software Engineering: Roadmap and Challenges Ahead”, pp 154:1-154:48, ACM, 2025.

[5] Hezi Zhang, Yiran Xu, Haotian Hu, Keyi Yin, Hassan Shapourian, Jiapeng Zhao, et al. “SwitchQNet: Optimising Distributed Quantum Computing for Quantum Data Centers with Switch Networks”, pp 1449-1463, ISCA’25, 2025.

[6] Muhammad Azeem Akbar, Arif Ali Khan, Sajjad Mahmood, Saima Rafi. “Quantum Software Engineering: A New Genre of Computing” pp 1-6, QSE-NE’24, 2024.

[7] Maryam Tavassoli Sabzevari, Matteo Esposito, Davide Taibi and Arif Ali Khan. “QCSHQD: Quantum Computing as a Service for Hybrid-Classical -Quantum Software Development: A Vision”, pp 7-10, QSE-NE’24, 2024.

[8] Caiseal Beardow. “Supporting Computing-Centred Intuitions for Quantum Computing through play”, pp 404-408, CHI PLAY Companion’24, 2024.

[9] Matteo Esposito, Maryam Tavassoli Sabzevari Boshuai Ye, et al. “Classi|Q>: Towards a Translational Framework to Bridge the Classical-Quantum Programming Gap”pp 11-14, QSE-NE’24, 2024.

[10] Y. Alexeev et al. “Introduction to the Special Issue on Software Tools for Quantum Computing: Part 2”, pp 1:1-1:3, ACM, 2023.

[11] Gushu Li, Anbang Wu, Yunong Shi, et al. “On the Co-Design of Quantum Software and Hardware”pp 1-6, NANCOM’21, ACM, 2021.