Designing A Cloud-Based Quantum Computing Simulator Approach Utilising Message Passing Interface (MPI) Technology

Saksham Agarwal

Montfort Sr. Sec. School, Ashok Vihar, Delhi

[1]*Received: 10 May 2024; Accepted: 28 July 2024; Published: 03 August 2024*

ABSTRACT

As research into quantum computing progresses, quantum algorithms' computational and time complexity continues to increase. Quantum computing simulators offer an effective platform for developing and verifying these algorithms. However, the running time and memory requirements grow exponentially with the number of qubits, posing a significant challenge for high-speed quantum computing on classical computers. Currently, a single node cannot meet the resource demands of quantum algorithms, necessitating the expansion of memory resources through multi-node computing clusters. The Message Passing Interface (MPI) provides high-performance inter-process communication capabilities, effectively balancing data transmission between hosts. By distributing quantum computing tasks across multiple servers, these tasks can collaboratively address the same problem. This paper proposes an advanced scheme for deploying quantum computing programs on a cloud computing platform, utilizing MPI for inter-server communication to effectively address the resource demands and accelerate quantum computing.

INTRODUCTION

The Message Passing Interface (MPI) is a standardized protocol for exchanging messages between multiple computers running a parallel program across distributed memory systems. This approach is the most prevalent in high-performance computing (HPC) environments. MPI facilitates inter-process communication for various applications, including analogue code and computational algorithms that require significant processing power. Typically, while a single server manages the input data for these applications, the MPI tasks are distributed across thousands of servers, which work together to solve complex computational problems. Applications using MPI are thus parallel and computationally intensive. During execution, the controller node attempts to run all processor cores at total capacity, ensuring that the MPI process is the sole application running on all servers. Although MPI may be interpreted differently across various computing platforms, it is generally analogous to the process concept in operating systems.

The primary purpose of MPI is to act as middleware, enabling upper-level applications to invoke MPI functions for message passing. MPI's significance lies in its ability to facilitate parallel computing tasks across multiple computers or processor cores within a single computer, allowing each node to handle a portion of the overall problem. As a standard interface, MPI synchronizes the actions of each parallel node, manages the entire cluster, and facilitates command execution and data exchange between nodes.

Leveraging MPI's unique computing properties, quantum algorithms can be accelerated significantly. QCMPI is a quantum package designed to evaluate quantum algorithms for many qubits and noise scenarios rapidly. It features the distribution of state-vector amplitudes over processors and qubits communication using the MPI protocol. Another package implementing the quantum trajectories method was presented, focusing on quantum open systems based on the MPI standard. A hybrid MPI/Open MPI parallelization method has been developed for the quantum drift-diffusion model in semiconductors. QMPI, introduced in, extends MPI to distributed quantum computing, enabling the development of portable, high-performance distributed quantum programs. Using the MPI-2 standard, the OPAL simulation architecture as designed for multi-scale, multi-centred simulations with advanced features like dynamic quantum region identification and dynamic process spawning.

Other methods also exist to accelerate quantum algorithms, such as combining GPUs and CPUs to enhance simulation speed and expand total memory space. Approaches using multiple NVIDIA GPUs to simulate an ideal quantum

[1] *How to cite the article:* Agarwal S., (August, 2024); Designing A Cloud-Based Quantum Computing Simulator Approach Utilising Message Passing Interface (MPI) Technology; *International Journal of Advances in Engineering Research*, August 2024, Vol 28, Issue 2, 1-5

computer and GPU Direct P2P transfer for handling high data dependencies between GPUs have been proposed and analysed.

With the continuous improvement of quantum hardware and developments in cloud access to quantum processors and various software packages, quantum computing systems can now be designed more flexibly in different forms. This paper introduces a new method: modifying MPI and Intel's quantum simulator source code for our serial and parallel combination quantum computing framework. Additionally, we have developed a general-purpose interface for leveraging cloud access to quantum processors.

MPI Deployment on Cloud Platforms

In the previous section, we discussed how MPI (Message Passing Interface) is primarily used in supercomputers. Intel's quantum computation team notably utilizes MPI supercomputer systems to simulate quantum computing through particle swarm optimization, serving as a classical subroutine. This quantum simulation aims to evaluate and optimize parameters in quantum algorithms and mitigate errors during computations. The latest framework in this domain is the Intel Quantum Simulator (IQS), previously known as qHiPSTER. According to [9], the first generation of IQS leveraged supercomputers by providing resources for quantum problem-solving. It featured two key aspects:

1. Utilizing multiple or multi-core processors to access the same memory concurrently, supported by OpenMP.

2. Handling scenarios where a small memory footprint requires significant computational effort, necessitating efficient communication between different processes supported by MPI.

With the rise of cloud computing technology, the second generation of IQS integrates with cloud computing platforms. It utilizes public cloud infrastructures to distribute computational power for simulating independent quantum states. IQS subdivides computing resources and simulates quantum circuits in parallel. This is achieved by setting up the MPI environment within IQS, which plays a crucial role in enabling parallel simulation of multiple quantum circuits. The computation is divided into groups, storing and updating a single quantum state based on quantum circuit operations.

IQS has proven effective in hypercomputing centres, which are specialized facilities equipped with high-performance computing resources for advanced scientific research. These centres can interface with cloud platforms like Xanadu, IBM-Qiskit, Rigetti-Forest, Google-Cirq, Microsoft Azure Quantum, and Amazon Bracket. Combining IQS, MPI, and cloud computing provides an efficient framework for large-scale quantum computing. This integration leverages the portability, stability, and compatibility of cloud computing to facilitate cross-center computation of complex quantum algorithms.

Inspired by the IQS framework, we deploy MPI across multiple cloud hosts running the BC-Linux operating system, as illustrated in Fig. 1. User applications can invoke predefined C language software interfaces and manage errors, progress, and memory using C++ APIs. Communication strategies include UDP, TCP, shared memory, and IB, all operating on the BC-Cloud platform. This setup enables high-performance communication of quantum state data across multiple nodes using MPI, particularly for short message transmission. It enhances the bandwidth for data transmission between quantum applications and physical hardware and balances computing loads for quantum programs with low resource utilization. Each quantum state program is simulated independently, improving overall efficiency.

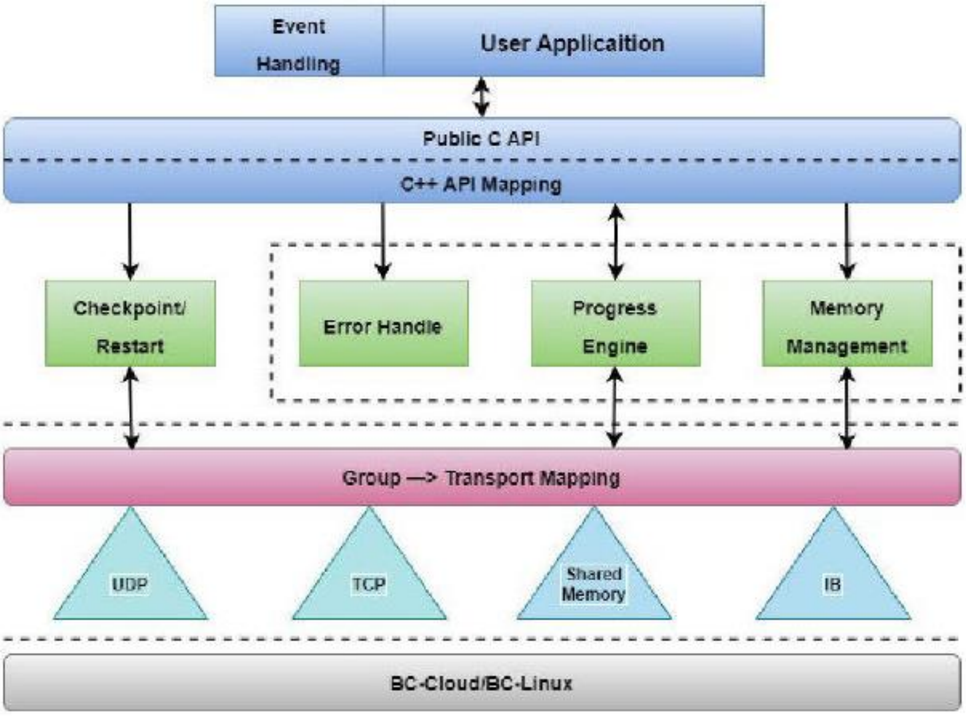

Figure 1. The architecture of user application and MPI deployed on the BCLinux system

Moreover, BC-Cloud, the platform on which we deploy MPI, offers comprehensive support for all MPI plug-ins. This assurance of support for various MPI standards during project development instils confidence in the adaptability of the system. BC-Cloud is a robust and scalable cloud platform that provides the necessary infrastructure for efficient MPI deployment. When components are loaded at runtime, they enforce common APIs while allowing flexibility in implementation. As shown in Fig. 2, the architecture of Open MPI includes multiple types of components within each layer, organized into frameworks. Each component belongs to a single framework, and each framework supports one component.

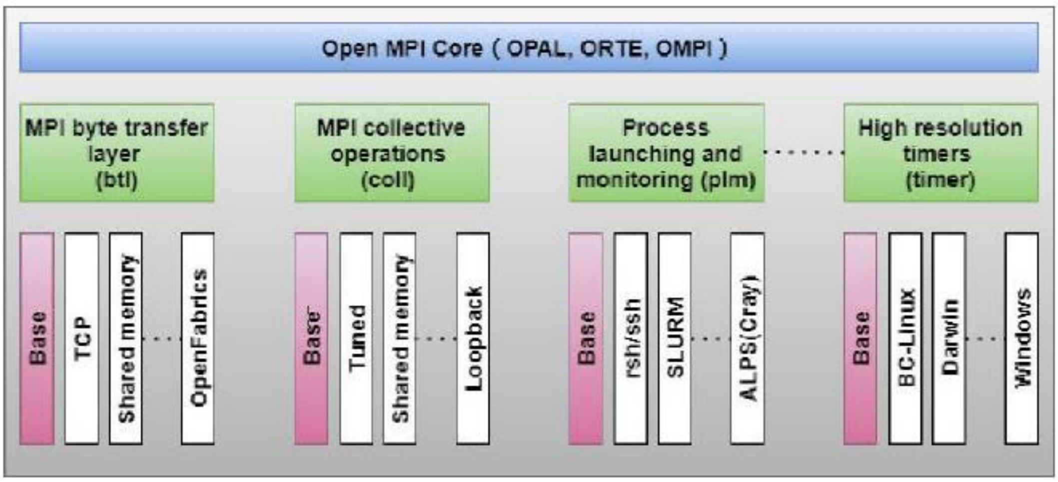

Figure 2. Using Open MPI frameworks and components on BC-Cloud

In summary, the deployment of MPI on cloud platforms, particularly through IQS and BC-Cloud, demonstrates an efficient and scalable approach to quantum computing. This integration, a testament to the practicality of the solution, provides a robust solution for complex quantum algorithm computations, utilizing the strengths of both MPI and cloud infrastructures. This efficient and scalable approach paves the way for exciting possibilities in the field of quantum computing, inspiring further exploration and development.

(IJAER) 2024, Vol. No. 28, Issue No. II, August **e-ISSN: 2231-5152, p-ISSN: 2454-1796**

QUANTUM CLOUD COMPUTING PLATFORM

The Quantum Cloud Computing Platform consists of two main components: the Cloud Host Management Platform and the Quantum Algorithm Simulation Platform. Its primary functions include quantum algorithm verification, cloud host resource management, and MPI parameter management, as illustrated in Figure 3.

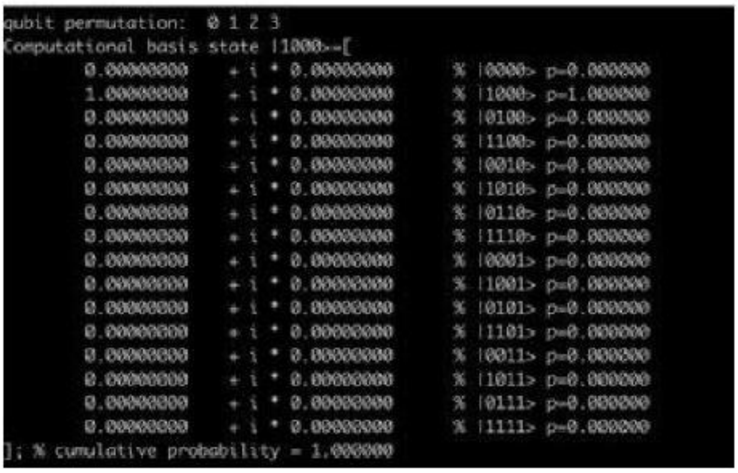

Figure 3. An instantiation: Quantum algorithm simulation platform

The platform allows for flexible scaling of the number of cloud hosts based on the complexity of quantum algorithms. For instance, consider a setup with four cloud hosts, with IP addresses ranging from 192.168.0.1 to 192.168.0.4. The configuration details for each cloud host are provided in Table I. One host is designated the controller node, which acts as the administrator and controls all other nodes. File sharing between cloud hosts on the same network is managed via the path `/parameter/root/mpi_share 192.168.0.x` with settings for read-write, synchronization, no root squash, and no subtree check.

TABLE I THE CONFIGURATION PARAMETERS OF CLOUD HOST.

Function	Configuration
CPU	16 Core
Memory	64G
network bandwidth	1.2Gbit/s
hard disc capacity	20G
Number of network cards	1
operating system	Ubuntu18.04 64 bit

The quantum algorithm is executed on the configured cloud hosts using the MPI computing mode. As illustrated in Figure 4, four cloud hosts operate simultaneously to expedite the calculation process of the quantum algorithm. For example, when running a quantum neural network algorithm, this parallel processing approach significantly reduces the training time.

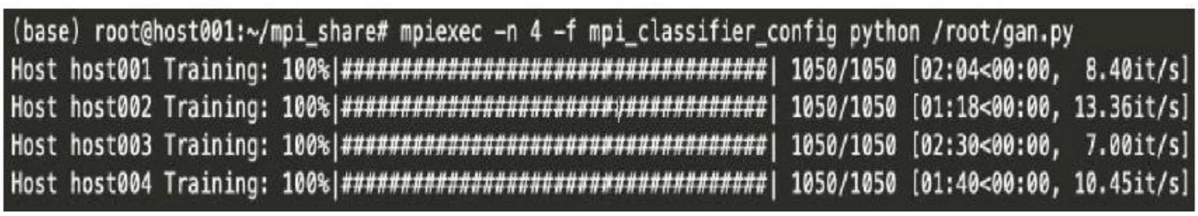

Figure 4. Quantum Generative Adversarial Network Training Diagram

CONCLUSIONS

This paper primarily discusses the development of a distributed quantum simulation platform, a quantum computing resource consumption model, and a quantum neural network algorithm. As a crucial innovation platform, the quantum computing cloud serves as the primary means for verifying the correctness of quantum programs. It enables enterprises and researchers to rapidly advance and innovate quantum methods, facilitating the swift resolution of practical problems.

REFERENCES

[1] Segall R S, Cook J S. Overview of Big-Data-Intensive Storage and Its Technologies[M]//Handbook of Research on Big Data Storage and Visualization Techniques. IGI Global, 2018: 33-74. 754

[2] Tabakin F, Julia-Diaz B. QCMPI: A parallel environment for quantum computing[J]. Computer Physics Communications, 2009, 180(6): 948- 964.

[3].Sawerwain M, Wisniewska J. QTM: Computational package using MPI protocol for Quantum Trajectories Method[J]. Plos one, 2018, 13(12): c0208263.

[4]. Sho S, Odanaka S. A hybrid MPI/OpenMP parallelization method for a quantum drift-diffusion model[C]//2017 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD). IEEE, 2017: 33-36.

[5]. Haner T, Steiger D S, Hoefler T, et al. Distributed Quantum Computing with QMPI[J]. arXiv preprint arXiv:2105.01109, 2021.

[6]. Cao C, Chen Y, Wu Y, et al. OPAL: A multiscale multicenter simulation package based on MPI-2 protocol[J]. International Journal of Quantum Chemistry, 2011, 111(15): 4020-4029.

[7]. Doi J, Takahashi H, Raymond R, et al. Quantum computing simulator on a heterogenous hpc system[C]//Proceedings of the 16th ACM International Conference on Computing Frontiers. 2019: 85-93.

[8]. Zhang P, Yuan J, Lu X. Quantum computer simulation on multi-GPU incorporating data locality[C]//International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, 2015: 241-256.

[9]. Guerreschi G, Hogaboam J, Baruffa F, et al. Intel Quantum Simulator: A cloud ready high performance simulator of quantum circuits[J]. 2020.