International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Replication of Quantum Computing towards an Unconventional Environment using QuIDE

E. Subramanian, C. S. Sanoj

Abstract: ***Current quantum computer simulation strategies are inefficient in simulation and their realizations are also failed to minimize those impacts of the exponential complexity for simulated quantum computations. We proposed a Quantum computer simulator model in this paper which is a coordinated Development Environment – QuIDE (Quantum Integrated Development Environment) to support the improvement of algorithm for future quantum computers. The development environment provides the circuit diagram of graphical building and flexibility of source code. Analyze the complexity of algorithms shows the performance results of the simulator and used for simulation as well as result of its deployment during simulation.***

Keywords: ***Development Environment, Quantum Computation, Quantum Algorithms***

I. INTRODUCTION

Simulating global quantum computers in traditional, classical digital systems is a giant task. By escalating the quantum computer qubits by the factor of one doubles the amount of memory required for the digital computer. There are only a few digital computers in the world which performs such calculations but it needs large amount of memory and processors that can be used effectively in today's parallel architecture to do simulations. Usually this type of calculations carried with the help of super computers only.

The fields of quantum computing have been formed on later days only to analyze the outcomes [7] and [8] from claiming test realizations for Shor's Prime factorization calculation [4]. Additionally, the main prototypes of a quantum central processing unit are assembled together. In spite of these limitations, there may be a convincing reason to quantum PC simulators which helps to learn and develop algorithms for future quantum computers. The theory at the back of quantum computing and information has advanced till beyond any physical execution. Many theoretical performance bounds and capabilities of complex algorithms have been established. Hover, scientists have developed simple physical realizations of quantum computers at best. Quantum computer simulation will help to link the disparity between high-level algorithms and quantum computer assembly.

Manuscript published on November 30, 2019.
* Correspondence Author
E. Subramanian*, Department of Information Technology, AMET University, Chennai, India. (Email: esmani84@gmail.com)
C. S. Sanoj, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, India. (Email: sanojcs@svce.ac.in)

In due course, the expectation is to optimize the layout of qubits and other physical components analogous to bits, circuits, and gates in traditional computers. Since simulating quantum information requires exponentially more processing and resources on traditional computers. To execute that efficient algorithms are needed. So the developed algorithms create single qubit gates and controlled qubit gates that are moderately faster than the algorithms that other quantum simulators use. The algorithms ire developed and analyzed in Mat lab.

A machine described by quantum theory is defined as Quantum computer. The basic storage unit of a quantum computer is represented by a two-level system called qubit in uncomplicated form. The status of the qubit is signified by a two dimensional vector $|\Phi\rangle = a\,(0|0\rangle + a\,(1|1\rangle$, Where $0\rangle$ and $1\rangle$ denotes two orthogonal basis vectors of the two dimensional vector space and $a_0 \equiv a\,(0)$ and $a_1 \equiv a\,(1)$ are complex number which is to be normalized such that $|a_0|^2 + |a_1|^2 = 1$. The interior condition of a quantum computer comprises of 'N' qubits which is described by a two dimensional unit vector of complex numbers $|\Phi\rangle = a(0 \ldots 00\rangle + a(0 \ldots 01 \mid 0 \ldots 01\rangle + \ldots +$ $a1 \ldots 101 \ldots 10 + a1 \ldots 11 \;\; 1 \ldots 11$, Where $i=0 2L-1 ai2=1$, this can be achieved by rescaling the complex valued amplitudes a_i , we standardize the vector $|\Phi\rangle$ such that $\langle\Phi|\Phi\rangle = 1$.

This paper discusses an assessment result of the most important techniques in the existence as well as their realization. Based upon our knowledge, the illustrated simulators entirely not help to uncover the proficient and suitable way of learning quantum algorithms. Each simulator provided with Application Programming Interface (API), a complex console interface and reduces the Graphical User Interface (GUI) [9]. So, we proposed a new environment to integrate the adaptability of Application Programming Interface (API) to implement the graphical Integrated Development Environment (IDE). This is very helpful to learn, develop and evaluate the quantum circuits. The assessment is done on different methodologies to minimize the effect on the exponential complexity of the simulator. The data structure is investigated which is used to accumulate quantum state and analyzing algorithms. Next, the performance results of the simulator are also illustrated. Consider a problem that is defined in terms of n (unknown) values $f\,(1) \ldots . f\,(n)$. The probabilistic query complexity of a problem is the minimum number of times that an algorithm has to consider the string $f\,(1) \ldots . f\,(n)$ to solve the problem with high probability. A classic example of this setting is the calculation of the OR of n bit values. Here the question arises is whether there is an index i with $f\,(i) = 1$.

The classical query complexity of this n task is$n$, whereas in the quantum setting we only need $O\ \sqrt{n}$ calls to f in order to solve the problem. We therefore say that there is a 'quadratic' separation between the classical and the quantum query complexity of the OR functions. The tasks which allow a quantum reduction [10] in the query complexity is a central one in quantum complexity research. The reason why quantum algorithms sometimes require fewer queries lies in the superposition principle of quantum mechanics. A single call i to the function n establishes the evolution$b\ (i) \rightarrow f\ (i) * b\ (i)$, which results best in classical computation from a f - query. But by the rules of quantum mechanics, we can also consult f in superposition.

II. LITERATURE SURVEY

Scott Aaronson et al. [9] discussed about the problems faced by the architect in the quantum computer. They propose a quantum simulation circuit which consists of CNOT, Hadamard and phase gate to perform simulation effectively and efficiently on traditional computer. It supports for the simulation by removing the noise in the Gaussian elimination at the cost of increasing the factor two in the number of bits needed to represent a situation.

Charles H. Bennett et al. [5] stated the difficulty in the classical cryptosystems which all are made up of full of assumptions and mathematical calculations. In the information theory concepts, the conventional secret key cryptosystems is not fully secure until unless the key which is shared among the systems was used only once. To get rid of this trouble they proposed a protocol called as coin tossing in which the exchange of quantum messages happens with limited number of keys sharing.

Hans De Raedt et al. [5] deals with the memory and processing speed of normal computer to super computer with the quantum calculations. For this they proposed a simulator called Julich universal quantum computer simulator. This follows a quantum gate circuit for quantum computing which can be represented as a sequence of matrix vector process involving sparse matrix. It also requires only a few number of arithmetic calculations used to revise the coefficient of the wave functions.

III. INNOVATIVE QUANTUM COMPUTING SIMULATOR

This paper suggests the Quantum Integrated Development Environment that helps to learning, understanding and examining the quantum algorithms. In addition, its performance should be sufficient to execute the algorithms in a reasonable time on standard Personal Computers. It is applied to set of applications, such as the quantum algorithms like Shor's factoring algorithm [2] and Grover's database search. Basic knowledge of quantum information and computation theory should be understandable of this simulator.

The principle functional requirements of the simulator include providing and dealing the quantum registers and quantum gates and perform the actual computations. The proposed simulator should support classical computations [5] and combining quantum, building custom computation subroutines out of the elementary gates, review the internal state of the simulated quantum system and its execution.

IV. QUIDE ARCHITECTURE

QuIDE comprises of the layers which isolate the application foundation from the interface presented to the client. This is attained by the Model View View - Model (MVVM) compositional configuration design [3], which is represented in Fig. 1. Every layer integrates segments that are in charge of giving particular application capacities. The View layer contains no business foundation. It is in charge of showing the GUI taking into account the information from the View - Model layer and in addition to that it passes the client's activities on to the View - Model layer. The View - Model upgrades the Model by taking into account the client's activities into the View layer. It likewise interprets information from the Model, so it can be shown by the View layer. The Model is the application foundation layer. It is in charge of communicating to the quantum circuits and performing calculations. It equally actualizes the supporting functional elements of the test system, for example, source code from the quantum circuit.

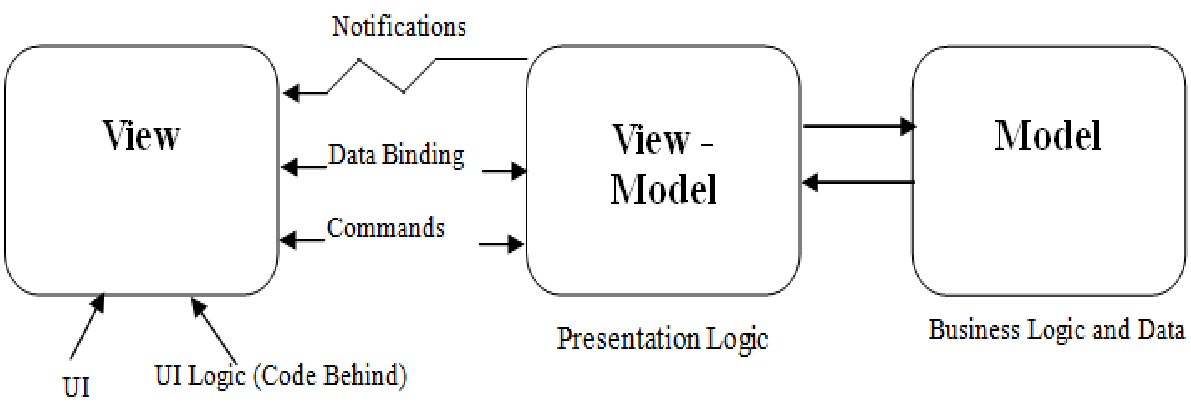

Fig.1. Model View View Model

V. ANALYSIS OF SIMULATION COMPLEXITY

A. *Single Qubit Gates*

To accumulate space at the cost of process time, a single qubit gate S is generated from a 2 x 2 matrix and U corresponds to the process to perform on a particular qubit. Diverse quantum computer simulators construct matrix representations of single qubit gates over the entire n qubit register by using (1) as follows,

$$S = (\otimes\, i-1\; k=1) \otimes U \;\otimes (\otimes\, n\; k=i+1) \qquad (1)$$

Where n the number of qubits is, U is the process applies to the i th qubit, and i is a 2 x 2 identity matrix. At this instant, we call the i th qubit to the target qubit (target = i). This equation interpreted into Mat lab code as follows.

In this code, kron (x, y) calculates the Kronecker tensor product of x and y, and speye (m, n) generates m x n sparse identity matrix.

```
if (target == n)
 gate = kron (speye (2^ (n − 1), 2^ (n − 1) , U);
else if (target == 1)
 gate = kron (U, speye (2^ (n − 1), 2^ (n − 1)));
else
 gate = kron (speye (2^ (target − 1), 2^ (target − 1)) , U);
 gate = kron (gate, speye (2^ (n − target), 2^ (n − target)));
end
```

This algorithm is equal to creating a $2_{i-1} \times 2_{i-1}$ identity matrix, make sure that identity matrix with the 2 x 2 matrix U, and then ensuring the result with a $2_{n-i} \times 2_{n-i}$ identity matrix. Note down the reason to store gates as sparse matrices, which is only to process on non-zero values. In the worst case, that is $n/2$ th qubit, the algorithm executes 2 (n + 1) multiplications and has two function calls. When the target qubit, i is either the 1 st (or) the n th qubit, i also need 2 (n + 1) multiplications but i only executes one function call.

B. *Time Complexity*

The above stated algorithm demonstrates the design about process on the quantum status. Each quantum process in QuIDE is accomplished by using iterative statement “for” and also the condition statement “if”. It iterates to all over the entries in the dictionary to represent state vector. For the decision of swapping the pairs of rows and columns, we simply find the pairs of tuples which have control bits equal to 1, different values for target bits in the same location, and same values for all other bits. The pair of numbers corresponding to the each tuple pair is the rows and columns that require swapping. It takes $\Theta\,(\log n)$. The time taken for an algorithm was stated below in Table - I.

Table – I Execution Time of an algorithm

Number of qubits	Target = first qubit (sec)	Target = last qubit n (sec)	Target = middle qubit n/2 (sec)
14	2.97	2.76	1.77
15	6.01	5.72	3.93
16	12.3	11	7.69
17	25.3	24	14.9
18	62.5	50	30.0
19	138	107	75.7
20	285	213	151
21	505	464	310
22	1110	1040	637
23	2230	2100	1280

C. *Space Complexity*

The quantum system’s status vector is stored in the database and used to signify the quantum calculations only for the data structure. It also reveals the present situation that in which those amplitudes are non-zero would put away in the word reference. In n th qubit, quantum framework is of the pure basis and the dictionary size of one is sufficient to store the data and its takes $\Theta\,(n \log n)$.

D. *Implementation*

Consider n persons, each of whom at first knows one secret, with no two people knowing the same secret. Each day, two people selected unvaryingly at arbitrary meet and swap over all the secrets they know. Now the experiment starts on what is the likely number of days until everyone knows everyone else’s secrets? By instinct, the answer is $\Theta\,(n \log n)$, since any given person has to wait $\Theta\,(n)$ days between meetings, and at each meeting, the number of secrets he knows approximately doubles (or) near to the end, the number of secrets he doesn’t know is roughly cut in two. Replacing persons by qubits and meetings by OR gates, we see a ‘segment transition’ from a thin to a thick representation, occurs after $\Theta\,(n \log n)$ arbitrary gates are applied. On the other hand, this argument does not hold down the proportionality constant, so that is we varied throughout in the experiment.

The results of the experiment are shown in Fig. 2. When $\beta = 0.6$, the time per measurement appears to grow approximately in a linear form n, whereas when $\beta = 1.2$ then the time per measurement appears to grow approximately in a quadratic form, so that executing the simulation took four hours of computing time. Thus, Fig. 2 clearly explains the ‘segment transition’ in simulation time, as increasing the number of gates by only a stable factors, alters the status from a system of simple states that are easy to quantify, to a system of complex states that are tough to quantify.

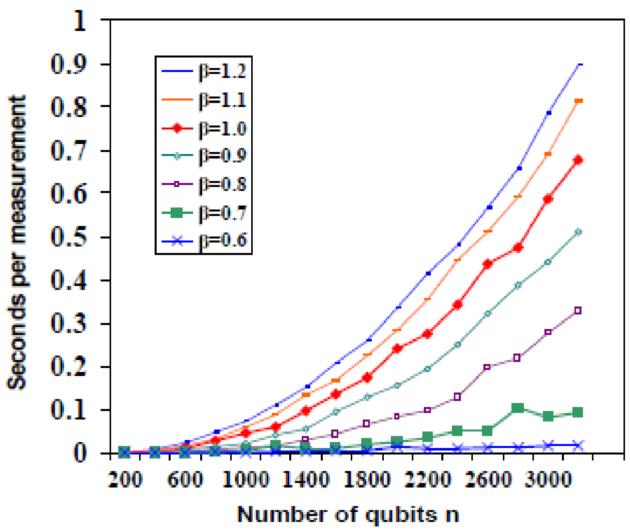

Fig. 2. Average time needed to simulate a measurement after applying $\beta n \log_2 n$ gates to n qubits

VI. CONCLUSION

This paper discusses with the simulation atmosphere to support well-organized and convenient learning and building the quantum computer algorithms. This methodology assessed existing quantum computer techniques and their prominent usage. The basic features of the prompt atmosphere concerning the convenient learning are: Graphically with a alteration between both approaches of algorithm, Helps to build quantum algorithms via source code and Systematic execution with the step back option to review the actual quantum state. In regards to that effectiveness criterion, the assessment of the execution done and compare with the existing simulators. QuIDE exhibits the creation of a usable tool, which have been effectively utilized to educational purposes. The one of the best currently available tools of this type is the performance and functionality.

REFERENCES

1. Hans De Raedt, Fengping Jin, Dennis Willsch, Naoki Yoshioka, Nobuyasu Ito, Shengjun Yuan, “Massively parallel quantum computer simulator”, *Computer Physics Communications*, vol. 237, pp. 47 - 61, 2019
2. Shanthi H.J., “A study on Urban planning using Big Data based on IOT”, *International Journal of Scientific Research and Review*, vol. 7(9), pp.122 - 130, 2018
3. Archenaa J., Anita E.A.M., “Interactive Big Data Management in Healthcare Using Spark”, *In: Vijayakumar V., Neelanarayanan V. (eds) Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC – 16’). Smart Innovation, Systems and Technologies*, *Springer,* vol. 49., Cham, 2016
4. Kirubakaramoorthi R., Arivazhagan D., “Analysis of Cloud Computing Technology”, *Indian Journal of Science and Technology*, vol. 8(21), pp.1 - 3, 2015
5. Charles H. Bennett, Gilles Brassard, “Quantum cryptography: Public key distribution and coin tossing”, *Theoretical Computer Science*, vol. 560, pp. 7 - 11, 2014
6. Kottalanka Srikanth, Arivazhagan D., “Genetic scheduling to optimize Resource utilization for Hospitals”, *International Journal of Computer Engineering and Technology*, 5(4), pp.126 - 137, 2014
7. Ganeshkumar K., Arivazhagan D., “Implecation of Cyber Security, Cyber Forensics and Online Privacy in Organizations”, *International Journal of Applied Engineering Research*, vol. 9(15), pp.2981 - 2990, 2014
8. Enrique Martin - Lopez, Anthony Laing, Lawson T, Alvarez R, Zhou X. Q, O’Brien J, “Experimental realization of Shor’s quantum factoring algorithm using qubit recycling”, *International Conference on Quantum Electronics*, vol. 6, pp. 773 - 776, 2012
9. Scott Aaronson, Daniel Gottesman, “Improved simulation of stabilizer circuits”, *Phys. Rev. A*, vol. 70, pp. 1 - 15, 2004.
10. Yamakami T, “A Foundation of Programming a Multi-tape Quantum Turing Machine”, *Mathematical Foundations of Computer Science*, *Springer*, vol. 1672, pp.430 - 441, 1999
11. Harry Buhrman , Richard Cleve , Avi Wigderson, “Quantum vs. classical communication and computation”, *Computer Physics Communications*, pp.63 - 68, 1998
12. Peter W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring”, *SIAM Journal on Computing*, vol. 26 (5), pp. 124 –134, 1997
13. Ethan Bernstein, Umesh Vazirani, “Quantum Complexity Theory”, *SIAM Journal on Computing*, vol. 26(5), pp. 1411 – 1473, 1997
14. Adriano Barenco., Charles H. Bennett, Richard Cleve, David P. Vincenzo, Norman Margolus, Peter Shor, “Elementary gates for quantum computation”, *Physical Review Letters*, vol. 52 (5), pp. 3457 – 3467, 1995
15. Charles H. Bennett, Gilles Brassard, Richard Jozsa, “Teleporting an unknown quantum state via dual classical and Einstein - Podolsky - Rosen channels”, *Physical Review Letters*, vol. 70 (13), pp. 1895 – 1899, 1993

AUTHORS PROFILE

E. Subramanian is an Assistant Professor in the Department of Information Technology in AMET University, Chennai, Tamil Nadu, India. He received his Master Degree in Computer Science and Engineering from College of Engineering, Guindy campus, Anna University, Chennai. He is the recipient of the Best Teacher Award of ESN 2019. His current research interest includes Big Data and Query optimization.

C.S. Sanoj is an Assistant Professor in the Department of Computer Science and Engineering at Sri Venkateshwara College of Engineering, Sriperumbudur, Tamil Nadu, India. He is a Life member of Indian Society of Technical Education. His educational background in Computer Science and Information Technology has given him a broad base from which to approach many topics. His current research work includes Human-Machine Interaction and Big Data Analytics.