6-2022

A practical comparison of quantum and classical leaderless consensus

Paul Robert GRIFFIN
Singapore Management University, paulgriffin@smu.edu.sg

Dimple MEVADA
Singapore Management University

A Practical Comparison of Quantum and Classical Leaderless Consensus

Paul Robert Griffin, Dimple Mevada[1]

ABSTRACT

Quantum computing is coming of age and being explored in many business areas for either solving difficult problems or improving business processes. Distributed ledger technology (DLT) is now embedded in many businesses and continues to mature. Consensus, at the heart of DLTs, has practical scaling issues and, as we move into needing bigger datasets, bigger networks and more security, the problem is ever increasing. Consensus agreement is a non-deterministic problem which is a good match to quantum computers due to the probabilistic nature of quantum phenomena. In this paper, we show that quantum nodes entangled in a variety of network topologies perform similarly to classical consensus executed on quantum simulators and real quantum computers with and without noise mitigation. There is no difference in the average time for the network to agree but there is a higher variation in agreement times for quantum compared to classical systems. The implication is that, with continued improvement in quantum technology, the scale and advantages of quantum processing can be exploited to provide for bigger and more sophisticated consensus. Furthermore, exploring the variation in agreement time could potentially lead to shorter consensus times.

Introduction

For distributed information systems, there is frequently a need to have consensus among the nodes in the distributed network. The performance of the consensus method is even more critical in decentralised systems such as

[1]*The authors are with Singapore Management University*

blockchain or distributed ledger technology (DLT). DLTs, and in particular public blockchains, have a fundamental scaling issue in that the size of the data block holding the transactions is limited (for example Bitcoin block size is 1MB) and it is often discussed [1] that it is not possible to simultaneously expand the network size, consensus speed and security. Meaning, for example, as the network size grows with more nodes, the consensus speed and security will decrease. This has been coined as the blockchain trilemma [1]. In this paper we look to quantum technologies to explore how this new technology could help overcome current DLT limitations and in particular for consensus. Quantum networks have already proven to be more secure than classical networks [2] and quantum computers can potentially process enormous amounts of data 2^(number of qubits) [3] and so the first question is whether a quantum consensus protocol can be designed that behaves at least as good as a classical one whilst being able to exploit quantum advantages for further improvements. Results for quantum consensus are shown here for quantum simulators and real quantum processors with and without noise mitigation.

In consensus networks, all nodes must agree on a common value [4] or a set of values at a point in time. These systems share data that is to be kept in consensus by transmitting and receiving over communication networks. To process new data, or any changes in data, consensus is required between the nodes, and if consensus is not reached, the consensus algorithm must be restarted. Consensus can be reached using many different consensus protocols [5] falling into two main types, either to elect a leader temporarily which the other nodes synchronise to (elected leader), or the nodes reach an agreement on the data (leaderless). Most DLTs implement consensus via an elected leader protocol, for example Raft [6] but there are many applications for leaderless consensus protocols such as for the reducing the risk of using external data sources (the oracle problem [7]) and for secure blockchain interoperability [8].

In quantum computation, information is stored in qubits. A qubit is in a probability of being in either of two states i.e. 0 and 1 simultaneously, called superposition. Superposition increases the amount of data that can be processed by a factor of 2^n where n is the number of qubits, potentially increasing the amount of data that can be maintained in consensus. Another unique property of quantum mechanics is the phenomenon of entanglement. An entangled system is said to be non-separable. In other words, if two qubits are entangled and one of the qubits is measured, the state of the other qubit is known without having to be measured. In a distributed system, this property of entangled states can be

used to ensure the integrity of the information being communicated [2]. In a quantum consensus network we assume that the qubits will be transmitted over quantum communication networks. Compared to classical computation, quantum computers have been proved to solve some problems more efficiently in terms of scalability and time performance in a network [9] in particular using a quantum/classical hybrid approach. However, it is important to note that the current state of quantum computing is still technically immature, with small numbers of physical qubits (typically <100) that have a number of noise issues such as decoherence of the quantum state and readout errors to name two.

In this paper, we discuss and compare the performance of a leaderless consensus model using the same consensus algorithm implemented on quantum and classical systems. The algorithm performs "rounds" of information exchange, and agreement is found by converging upon an average of the values from the nodes. We measure the number of rounds to reach agreement and the complexity of scaling the number of nodes. The quantum system is implemented on a simulator and on IBM quantum hardware with and without noise mitigation. Quantum entanglement is used to distribute the information among the nodes of the network.

Experimental Setup

In this section we will explain in detail, the architecture, the algorithm used, the coding and the hardware implementations with the noise error mitigations.

For most experiments we consider three autonomous nodes (N=3) where each node comprises three qubits forming the quantum sub-system interfaced to its classical node (Fig. 1). The qubits are arranged so that one qubit is encoded with information to be distributed from its own classical data store (for example q^0_0) and the other qubits (for example q^0_1 and q^0_2) hold the information from the other nodes. For this example, we have a full mesh topology with all nodes connected to each other with bi-directional information flow.

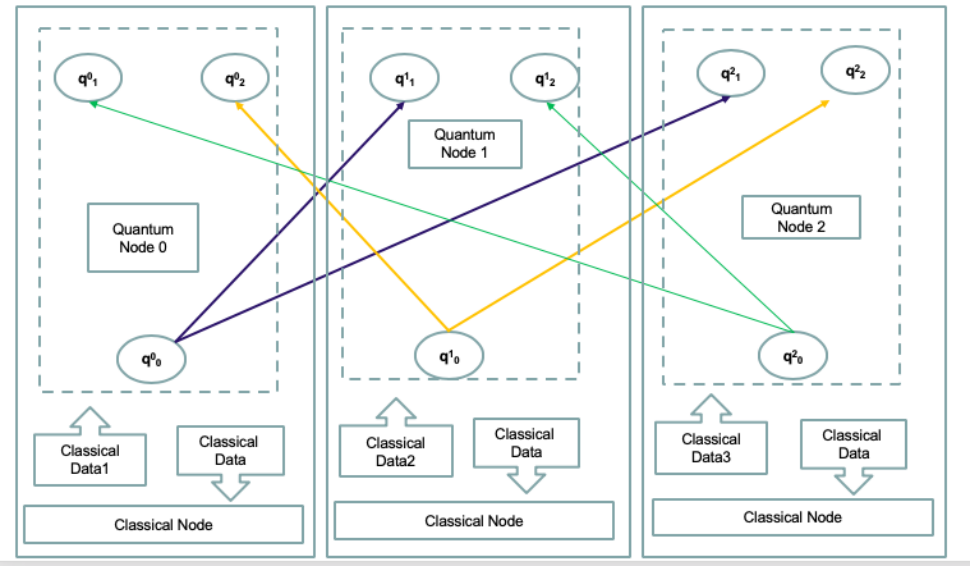


Figure 1: Schematic diagram of three nodes with a three qubit sub-system and interconnections to other nodes via entanglement. Superscripts denotes the quantum computer node and subscripts denotes the qubit number within the node (q^{node}_{qubit}). The thin arrows represent the path of a qubit information between the qubits across the nodes. The thick arrows represent the interface between the quantum computer and the classical computer.

Algorithm Used

The algorithm used in this study is based on the Ben-Or leaderless agreement steps [4]. In the original algorithm, to ensure that there is convergence, a random value is selected. In our version, we removed the random selection and replaced it with a calculation of an average value. This enables the algorithm to be purely deterministic, stabilising the classical behaviour and highlighting non-determinism from quantum computing effects.

Before the algorithm is executed the following properties must be agreed by the participating nodes:

A. The network must agree on a *threshold* value of variance that defines if consensus is sufficiently close enough or not to be in agreement.

B. The network must agree on a maximum number of consensus rounds to stop in order to avoid any infinite loops.

The leaderless algorithm steps for each node for each block comprise of:

1: A new state of the classical computer (we will use node 0 as an example but it is the same for all other nodes in the network) is loaded into the primary qubit, q^0_0 (see Fig. 1).

2: Another qubit in the node, q01, is entangled with q00.

3 Qubit q01 is transmitted through a quantum channel to another connected node.

4 Each node measures the quantum states from all the transmitted qubits and calculates a consensus value, c (the average of the values received) and consensus difference d (the variance between the values received).

5 Each node updates its own primary qubit, q00, with the calculated consensus value.

6. If d is > threshold, then another round of communication is performed.

7 If d < threshold for a majority of connected nodes, then:

update the classical state with the value in q00 and go to step 1

else:

go to step 4 if within a maximum number of rounds.

For classical systems, the qubits mentioned in the algorithm are replaced with integer variables representing the value of the classical state. This algorithm can be implemented on one or multiple quantum processors when quantum connections are available.

Code Development

The overall programme flow for is to take the node source value of the initial state of the node, create a quantum circuit, execute the circuit, measure the final qubits states, calculate the average and variance of the node values, decide on the agreement and whether to stop or have another round. The code was written in Python 3 on Jupyter Notebook 6.1.4 using the Qiskit '0.15.1' and executed on laptops with Intel i7 with 16GB of RAM.

A program was written for the algorithm described in the section above and was executed multiple times on classical and quantum computers with different number of nodes and topologies to compare results.

Hybrid Architecture

A classical/quantum hybrid architecture was used. As no quantum network connecting quantum computers is currently available all the qubits are programmed onto one quantum chip (QPU). The classical data from the initial value and from subsequent rounds for each node is embedded into the qubits as a rotation around the y axis in the Bloch sphere (i.e. a single qubit Ry gate rotation gate) [10]. Entanglement gates (CNOT) are then added to the quantum circuit (Fig. 2) according to the topology. For example, with 3 nodes and a unidirectional ring topology each node will have 2 qubits in each node (Fig. 2) requiring three Ry gates and three CNOT gates. Some other rules were required to enable good circuits, such as each node needs to have at least one minimum connection and must be involved in a complete connection path. The total number of qubits required for n nodes is n^n. For example, 3 nodes require 9 qubits, 6 nodes requires 36 qubits.

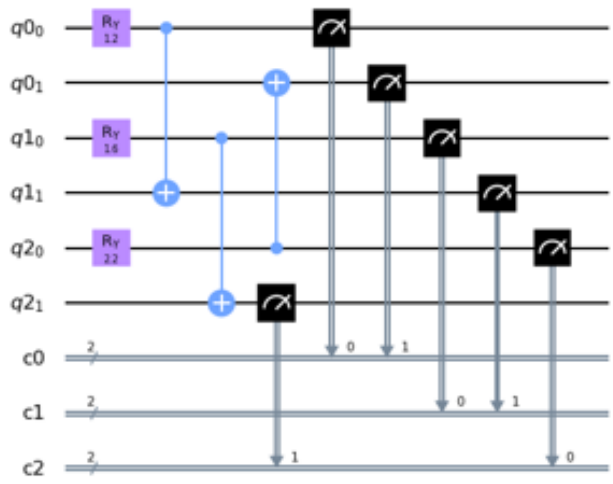


Figure 2. Example quantum circuit diagram

After the circuit is executed for a number of executions (also called “shots” with a default of 1024 in Qiskit), the qubits are measured giving a probability measurement [11] of each qubit of the quantum register being in |1> or |0> states. The probabilities are then used to classically calculate the average value, c, and the variance, the consensus difference, d for each individual node after each round. If the majority of the nodes agree within a defined variance, the algorithm stops, and consensus is declared to be achieved.

For the calculation of the new consensus value to be used for the next round, we scale the probability (from 0 to 1) of the qubit being in the |0> state to angle in radians (0 to pi) to apply to the Ry gate of the primary qubit.

Hardware

We used two quantum simulators and several real backends with and without error mitigation. The consensus network is implemented with 3 or more nodes all on the same processor. The first quantum backend used was a statevector simulator [12] which is a mathematical model of quantum state evolution. In order to retrieve the probability density information of the subsystem A, from an entangled system AB, we trace over the sub-system B and vice-versa for B using a partial trace, reduced density operator [13]. After running the quantum circuit on a statevector simulator, a QASM simulator was used from the qiskit-Terra package which more closely mimics real quantum hardware than the statevector simulator. The QASM simulator from Qiskit simulates a real backend device but is noise-free. We used the |0> state probability of each node in order to calculate its consensus value in the algorithm.

Next, we ran the algorithm on several real backends with ibmq_boeblingen chosen due to the number of qubits, qubit connectivity (Fig. 3) and noise characteristics such as measurement noise and decoherence. The real backend results are prone to more errors such as gate errors and read-out errors, meaning that if the bit to be read is ‘1’ it will be read out as ‘0’ and vice-versa. Also, for the real backends, we need to select the correct mapping of the logical qubits in the quantum circuit to the available physical qubits of a real backend device. For example, taking the circuit in Fig. 2, the logical qubit q^0_0 should be mapped to the physical qubit 1 and q^1_1 mapped to 6 to minimise the physical distance for the entanglement. Optimising this for larger and more complex topologies gets very complicated.

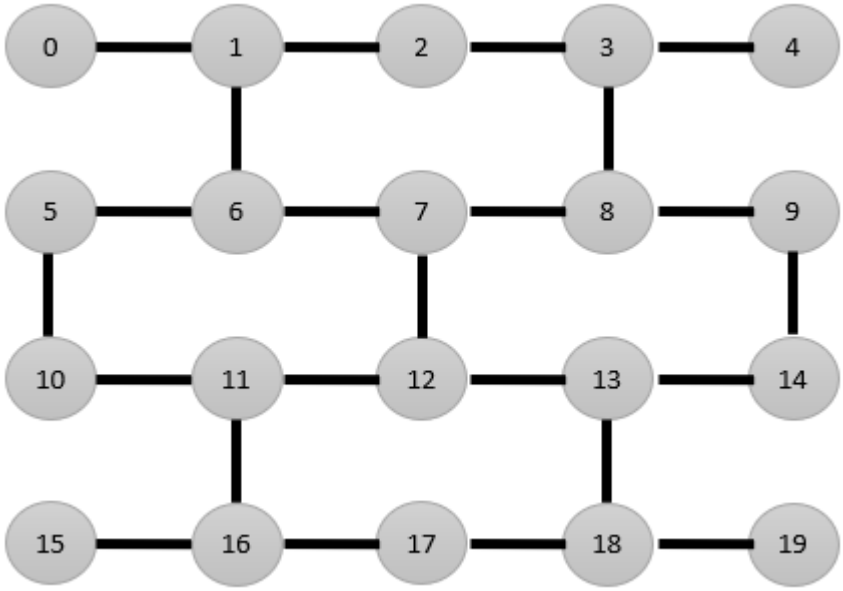


Figure 3. ibmq_boeblingen qubit connectivity.

In order to correct problematic results obtained from using real devices we use a Measurement Error Mitigation [14] methods in the Ignis package in Qiskit for calibrating circuits providing a calibration matrix [14]. The calibration matrix used for error mitigation was generated for the quantum computer ibmq_boeblingen. The functions *complete_meas_cal* and *CompleteMeasFitter* were used to generate the circuits used for the measurement calibration for the entire Hilbert space. For example, if we have n=6 qubits, then we require 2^6=64 calibration circuits. The function *complete_meas_cal* provides a list of calibrated circuits (meas_calibs) with state_lables information taking in the qubit list, quantum register and classical register as its input information. Next, we compute the calibration matrix of 64 X 64 for the backend. The calibration matrix generated has most of the noise information in the diagonal part of the matrix. After computing the calibration matrix, we apply a filter object for the full calibration and the results with the mitigation are obtained. Other error correction methods are possible but demand a large number of physical qubits.

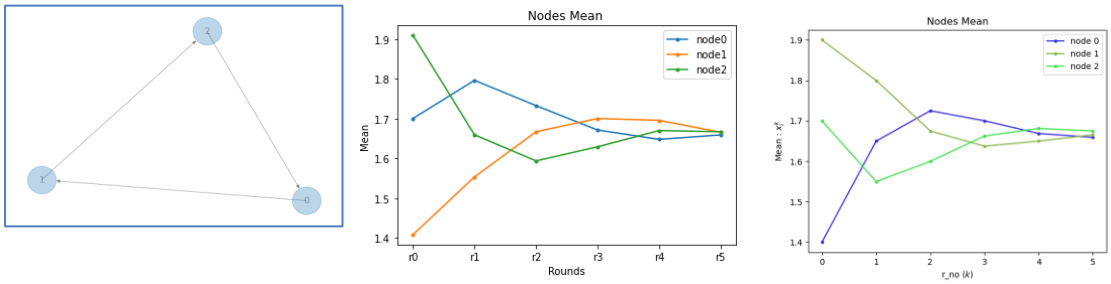
Results

Results are presented for the same consensus algorithm executed on quantum and classical systems for a range of connection topologies with nodes fully and partially connected in one or both directions. We measured the number of rounds required for agreement to reach a consensus value and the computational

complexity as the number of nodes, N, increase. Ideally, quantum communication channels would be used for quantum information distribution but, due to the current lack of availability of suitable hardware, a quantum circuit is generated to link the qubits in the nodes in the same quantum processor.

Number of Rounds

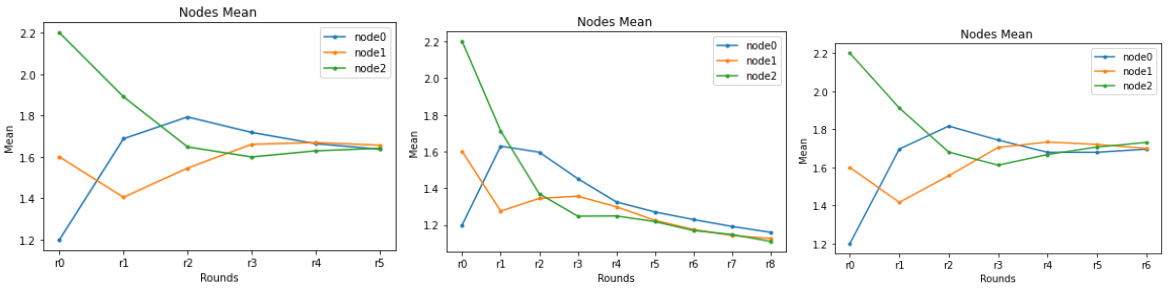
For a fully connected mesh, with the nodes connected bidirectionally, and starting values of 1.41, 1.70 and 1.92, all nodes agreed with the mean of 1.67 within a threshold variance = 0.0001 within one round. For all nodes with unidirectional connections (Fig. 4a) a total of 6 rounds was required executed on a quantum simulator (Fig. 4b) and a classical computer (Fig. 4c).



(a) Topology diagram (b) Quantum (c) Classical

Figure 4: Average distributed consensus for 3 nodes (N=3) connected unidirectionally (a), for a quantum simulator (b) and classical system (c).

With the same topology and connections as in Fig. 4, comparing the quantum (QASM) simulator (Fig. 5a) with a real quantum computer backend (Fig. 5b) we observed that whilst the nodes in the real backend did reach agreement, the agreed value was much lower, around 1.15, than that expected 1.67. This is likely to be due to noise in the quantum system including qubit decoherence, gate noise and measurement noise. The quantum system errors can be mitigated to an extent using the measurement error mitigation module for Qiskit bringing the agreed value much closer to a value of 1.7 (Fig. 5c) to that of 1.67 expected. Unfortunately, repeating the experiments with error mitigation still showed significant variations in the agreed value over time.

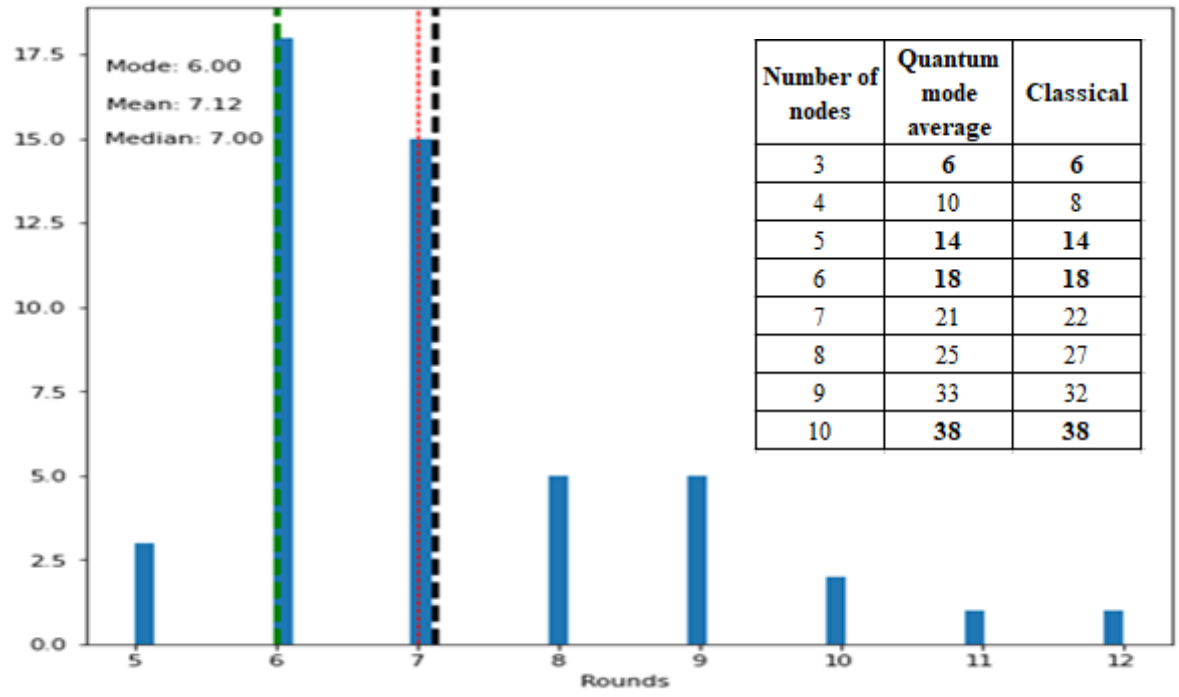


(a) (b) (c)

Figure 5: Convergence for 3 unidirectionally connected nodes for a QASM simulator (a) a real backend (b) and with error mitigation (c).

Complexity with Network Size

To explore the effect of increasing the size of network from 3 to 10 nodes, a ring topology with each node connected unidirectionally to a neighbour was used. The consensus algorithm was executed for 50 iterations. We saw that the average number of rounds required to reach an agreement has approximately linear relationship as the number of nodes increased for both the quantum simulator and classical system as might be expected (Fig. 6 inset).



Number of nodes	Quantum mode average	Classical
3	**6**	**6**
4	10	8
5	**14**	**14**
6	**18**	**18**
7	21	22
8	25	27
9	33	32
10	**38**	**38**

Figure 6: Histogram plot of the counts for the number of rounds to reach agreement for N=3 and a ring topology for the quantum simulator. The mean average (black dashed line), mode average (green dashed line) and median (red dotted line) are shown. The inset table shows the

mode average number of rounds to terminate for the quantum simulator compared to the classical system for numbers of nodes from 3 to 10. Bold numbers are an exact match.

Quantum Variations

Running the same consensus algorithm for the same topology on a quantum simulator produces variations in the number of rounds needed to agree even though the consensus algorithm used is deterministic. For 3 nodes fully connected unidirectionally in a ring topology and executed for 50 iterations we observed variations in the number of rounds to reach to an agreement (Fig. 6) and that the mean, mode and median averages are different.

The classical system produces the same number of rounds each iteration as expected. The mode average (repeated most often) in the quantum system for the number of rounds required to reach an agreement is the closest to the classical system (Fig. 6 inset).

Topology Variations

We have explored a range of topologies providing different routes for the nodes to reach an agreement. Mesh topology is a network where all the nodes are connected to each other with bi-directional information flow. A partial mesh network is where all the nodes are indirectly connected to each other and better mimics the real world where the nodes connect to some other nodes more distantly. We obtained results for 5 nodes where each node is bidirectionally connected to 2 to 4 other nodes and observed again that the number of rounds required to reach similar agreements for quantum at 6 and classical at 7 rounds.

Discussion

Consensus for leaderless protocols is achieved by transferring information among the nodes of a network and reaching an agreement. In these experiments we used a deterministic algorithm to ensure the classical comparison is consistent and better highlight differences between quantum and classical systems. Firstly, we observed that the number of rounds required to reach an agreement is similar in quantum systems to classical systems which is not surprising as the information being transferred is the same just encoded into quantum states for the quantum system. As far as computational complexity as the network grows, again we observed similar results of a linear relationship pattern for a ring topology. This

implies that, with potentially large information storage of quantum registers, we should be able to process more data with quantum systems. The current problem is to load classical efficiently but if the data was quantum, for example photonic qubits incident on a photonic quantum computer, the loading problem is less problematic.

It is not surprising that there is a variation in number of rounds for the quantum system as the qubits are probabilistic. What is interesting is that the median and mean averages are larger than the mode indicating the distribution is not Gaussian (Fig. 6 inset) and so the variation is not random. This is seen on a simulator as well as the real backends. If the mechanism behind the distribution is understood and can be engineered then the number of rounds for agreement could be reduced overall.

Using the IBM real quantum computer hardware with and without error mitigation compared to the QASM simulator highlights the problems with current quantum hardware. These NISQ machines need to be improved for errors and for stability. We can see the errors build up with each round of agreement showing a drooping of the consensus value (Fig. 5b). Even though error mitigation helped (Fig. 5c) it was found to be unstable as the machines were recalibrated often.

In conclusion, we have introduced the probabilistic consensus model using quantum mechanics to potentially solve the blockchain trilemma. In this paper, we have investigated the quantum aspects of the agreement in the distributed systems for a partially connected network and found good agreement with classical with a number of topologies. This bodes well for future post-NISQ machines that can handle larger datasets and could process quantum data.

As well as the DLT consensus trilemma issue there are also issues around oracles [7] and blockchain interoperability [8]. Oracles are off-chain data sources and can potentially put the integrity of blockchains at risk due to having to be trusted. Interoperability between blockchains is an issue due to data privacy concerns and protocol differences. Leaderless consensus can help for these two issues. The algorithm here for quantum consensus is independent of the implementation and can be implemented in one quantum system or in a hybrid system with both classical and quantum components.

There are many ways to extend this work such as increasing the size of the encoded data by arbitrary initialization provided the normalization of the qubits. Also, introducing a random selection of variables, full quantum circuits and, of course, deploy onto the quantum internet when available. Quantum teleportation [15] protocols could be introduced for state transfer across the nodes and secured with quantum cryptography techniques. Mitigation methods can be explored further to enhance the accuracy on real backends.

References:

[1] Vitalik Buterin, "The Limits to Blockchain Scalability," May 23, 2021. https://vitalik.ca/general/2021/05/23/scaling.html (accessed Aug. 31, 2021).

[2] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Rev. Mod. Phys.*, vol. 81, no. 3, pp. 1301–1350, Sep. 2009, doi: 10.1103/RevModPhys.81.1301.

[3] J. Preskill, "Quantum computing 40 years later," *ArXiv210610522v2 Quant-Ph*, Jun. 2021, Accessed: Aug. 31, 2021. [Online]. Available: http://arxiv.org/abs/2106.10522

[4] M. Ben-Or, "Another advantage of free choice (Extended Abstract): Completely asynchronous agreement protocols," in *Proceedings of the second annual ACM symposium on Principles of distributed computing*, New York, NY, USA, Aug. 1983, pp. 27–30. doi: 10.1145/800221.806707.

[5] R. Wattenhofer, *The Science of the Blockchain*, 1st ed. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2016.

[6] "Raft Consensus Algorithm," n.d. https://raft.github.io/ (accessed Aug. 31, 2021).

[7] G. Caldarelli and J. Ellul, "The Blockchain Oracle Problem in Decentralized Finance - A Multivocal Approach," Jul. 2021, doi: 10.20944/preprints202107.0231.v1.

[8] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A Survey on Blockchain Interoperability: Past, Present, and Future Trends," *ArXiv200514282 Cs*, Mar. 2021, Accessed: Aug. 31, 2021. [Online]. Available: http://arxiv.org/abs/2005.14282

[9] B. S. Chlebus, D. R. Kowalski, and M. Strojnowski, "Scalable quantum consensus for crash failures," in *Proceedings of the 24th international conference on Distributed computing*, Berlin, Heidelberg, Sep. 2010, pp. 236–250.

[10] IBM, “Quantum Computing | IBM,” n.d. https://www.ibm.com/quantum-computing/ (accessed Aug. 31, 2021).
[11] IBM, “Lab 2. Quantum Measurement.” https://community.qiskit.org/textbook/ch-labs/Lab02_QuantumMeasurement.html (accessed Feb. 28, 2022).
[12] “IBM Quantum simulators,” *IBM Quantum*. https://quantum-computing.ibm.com/services/docs/services/manage/simulator/ (accessed Aug. 31, 2021).
[13] L. Mazzarella, A. Sarlette, and F. Ticozzi, “Consensus for Quantum Networks: From Symmetry to Gossip Iterations,” *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 158–172, Jan. 2015, doi: 10.1109/TAC.2014.2336351.
[14] IBM, “Measurement Error Mitigation — Qiskit 0.29.0 documentation,” n.d. https://qiskit.org/documentation/tutorials/noise/3_measurement_error_mitigation.html (accessed Aug. 31, 2021).
[15] M. Vogel, “Quantum Computation and Quantum Information, by M.A. Nielsen and I.L. Chuang,” *Contemp. Phys.*, vol. 52, no. 6, pp. 604–605, Nov. 2011, doi: 10.1080/00107514.2011.587535.

Biographies

PAUL ROBERT GRIFFIN (paulgriffin@smu.edu.sg) is currently in Singapore Management University teaching postgraduate and undergraduate students in IT and FinTech as an Associate Professor of Information Systems. He received a Ph.D in quantum well photovoltaics at Imperial College London in 1997 and is now researching disruptive technologies applications and impact specializing in blockchain and quantum computing. Prior to SMU he was leading application development on global IT projects in banking for over 15 years in the UK and Asia in the financial industry.

DIMPLE MEVADA (dimple.bav@gmail.com) received her B.Sc. degree in Physics and M.Sc. degree in Physics from Mithibai College, University of Mumbai, India in 2004 and 2006 respectively and a P.G.D.I.T. degree in Information Technology from University of Mumbai , Mumbai , India in 2008. Her research interests include embedded systems, quantum systems communication and applications.

Acknowledgments:

The authors would like to acknowledge One Connect Financial Technology for funding the work, Centre of Quantum Technology (NUS) for access to quantum computers and IBM Quantum Experience and Qiskit developers for their platform and advice. In particular, the authors would like to thank Corey Mason Manders and Kim Eun Taek (ET) for insightful and enjoyable discussion during and after the project.

Keywords

DLT, blockchain, distributed, ledger, decentralised, quantum, computing, consensus