Not All Qubits Are Created Equal

A Case for Variability-Aware Policies for NISQ-Era Quantum Computers

Swamit Tannu and Moinuddin K. Qureshi
Georgia Institute of Technology

---

1 Introduction and Motivation

Quantum computer (QC) has evolved from a theoretical idea to a realizable system. For example, Google, IBM, and Intel have announced blueprints for QCs with 72, 50 and 49 quantum bits (qubits) respectively [1], [2], [3]. QC can accelerate fundamentally hard problems by leveraging properties of qubits. However, existing QCs are limited as qubit devices are incredibly fickle and tend to lose state due to decoherence. Furthermore, the operations on qubits can also experience errors. Existing and near-term QCs with hundreds of qubits can not utilize quantum error correction to protect the qubit state due to the limited number of qubits. Such machines with 10 to 1000 noisy qubits are termed as *Noisy Intermediate Scale Quantum computers (NISQ)* [4]. Even though NISQ-machines may not have fault-tolerance, they can still provide significant benefits for quantum optimization and quantum chemistry simulations [5], [6], [7].

NISQ applications are run multiple times to improve the accuracy of the solution as the output of single NISQ execution can have errors. Performance of NISQ application depends on the *probability of successful trial (PST)*. When PST is unity, a single run is enough. Whereas for lower PST, the number of runs can increase substantially. PST of an application depends on the error rate of operations in the program. On a NISQ machine with a uniform error rate across qubits, PST is independent of qubit mapping. However, any skew in error rates across qubit devices presents an opportunity to improve the PST. In this paper, we make a case for variation-aware compilation policies that focus on Qubit-Allocation (mapping of program qubits to machine qubits), and Qubit-Movement (routing qubits from one location to another) to improve the overall system reliability by introducing better than worse philosophy for NISQ compilers.

To understand and quantify the variation in the error-rate across qubits, we analyze the publicly-available characterization data for the IBM-Q20 (20 qubits) machine [8]. We observe significant variation in the behavior of different qubits and links – in essence, qubits, and links are not created equal. For example, the links connecting different qubits shows 7x difference in error rate across the weakest and strongest links in the system. These links are essential for performing a two-qubit operation that entangles a pair of qubits. Unfortunately, QC can entangle only the qubits that have a link between them. For example, Figure 1 shows a schematic of IBMQ-20 where circular nodes represent the qubits and edges represent the coupling links between qubits [8]. A pair of qubits can only be entangled if there exists a coupling link between them. Fortunately, quantum computers provide a *SWAP* instruction that can exchange the state of two neighboring qubits. For example, we want to entangle data qubit Q_A and data qubit Q_C which are initially residing at physical qubit-16, and physical qubit-18 respectively as shown in the figure 1. To entangle Q_A and Q_C, a SWAP-operation between qubit-17 and qubit-18 is performed such that Q_A and Q_B interchanges positions. Next, perform a two-qubit operation between qubit-16 (data Q_A) and qubit-17 (data Q_C).

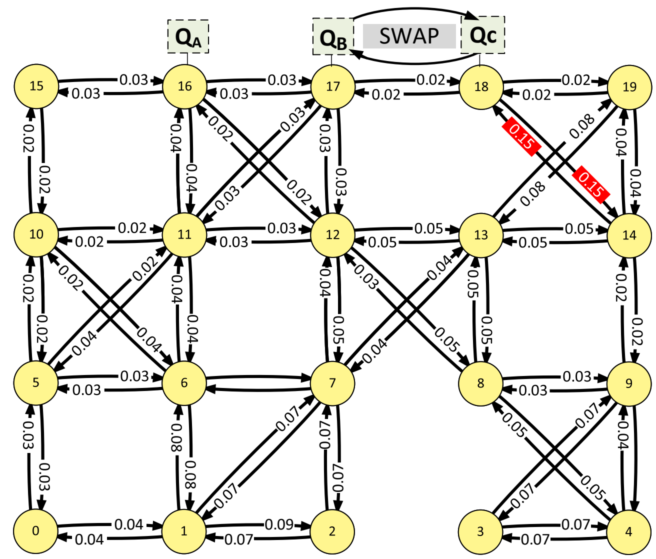

Fig. 1. Layout of IBM's 20 qubit machine, each edge represents a possible 2-qubit operation. The label on the edge represents the probability of failure on that link when an operation is performed. The best link(s) have an error-rate of 0.02, and the worst link has 0.15, so a difference in strength of 7.5x.

2 Variation Aware Compiler Policies

In NISQ programs, a significant number of SWAP instructions are inserted to move data to enable arbitrary two-qubit operation. The insertion of SWAP instructions is done statically by a compiler. Therefore the information about link usage is available and deterministic, and routing can be done without deadlocks. Unfortunately, SWAP instructions have

3x higher latency and significantly less fidelity. To improve reliability, existing quantum compilers attempt to reduce the number of SWAPs by using intelligent Qubit-Movement and Qubit Allocation policies [9], [10]. The Qubit-Movement policy deals with the problem of selecting a route to move the data of one qubit to another. Existing compiler policies assume uniform cost of performing SWAP operations. However, in reality, we expect variation in the behavior of different qubits and links, and optimizing for a uniform behavior may not result in the best policy when device variation is taken into account. We propose *Variation-Aware Qubit Movement (VQM)* policy that routes the qubit from source to destination based on minimizing the probability of failure. Baseline design always selects the shortest path that requires a minimum number of SWAPs. Whereas, VQM selects a most reliable path (sometimes that uses a slightly longer path to avoid weak links).

We also propose *Variation-Aware Qubit Allocation (VQA)* that performs the mapping of program-qubit to physical-qubit to improve overall system reliability. A conventional mapping policy can choose any of the listed mapping possibilities as they all would have similar cost in terms of SWAP operations. However, VQA uses the mapping that uses the most reliable links and would improve the overall system reliability. We extend prior proposals for Qubit-Allocation with VQA and show that VQA improves system reliability significantly. We evaluate VQM and VQA on 5-qubit IBM Quantum Computer and 20 qubit simulator. Our evaluations show with combined VQA and VQM policies system reliability improves by 1.9x on IBM quantum computer and 1.8x improvement for simulations.

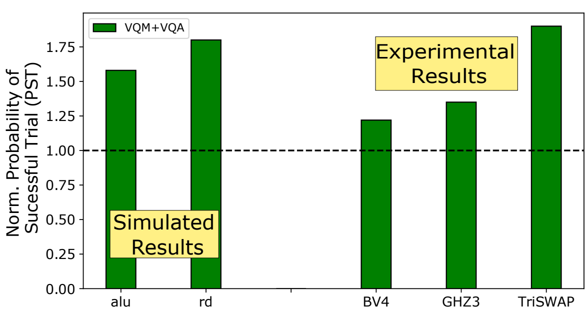

Fig. 2. Improvement in PST over variation unaware baseline by using Variation-aware qubit move (VQM) and variation aware qubit allocation(VQA) policy on a simulator and real IBM Quantum Computer

References

[1] W. Knight, "IBM Raises the Bar with a 50-Qubit Quantum Computer," https://www.technologyreview.com/s/609451/ibm-raises-the-bar-with-a-50-qubit-quantum-computer/, 2017, [Online; accessed 3-April-2018].

[2] J. Hsu, "CES:Intel's 49-Qubit Chip Shoots for Quantum Supermacy," https://spectrum.ieee.org/tech-talk/computing/hardware/intels-49qubit-chip-aims-for-quantum-supremacy, 2017, [Online; accessed 3-April-2018].

[3] J. Kelly, "A Preview of Bristlecone, Googles New Quantum Processor," https://research.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html, 2018, [Online; accessed 3-April-2018].

[4] J. Preskill, "Quantum computing in the nisq era and beyond," *arXiv preprint arXiv:1801.00862*, 2018.

[5] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, 2017.

[6] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn *et al.*, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology*, vol. 3, 2018.

[7] A. Morello and D. Reilly, "What would you do with 1000 qubits?" *Quantum Science and Technology*, vol. 3, 2018.

[8] I. B. M. Corporation, "Universal Quantum Computer Development at IBM:," http://research.ibm.com/ibm-q/research/, 2017, [Online; accessed 3-April-2017].

[9] A. Zulehner, A. Paler, and R. Wille, "Efficient mapping of quantum circuits to the ibm qx architectures," *arXiv preprint arXiv:1712.04722*, 2017.

[10] M. Siraichi, V. F. Dos Santos, S. Collange, and F. M. Q. Pereira, "Qubit allocation," in *CGO 2018-IEEE/ACM International Symposium on Code Generation and Optimization*, 2018.