High Fidelity Qubit Mapping for IBM Q

Shin Nishio[1 2 *] Yulu Pan[2 3 †] Takahiko Satoh [2 ‡] Rodney D. Van Meter[2 4 §]

[1] *Faculty of Policy Management , Keio University, 5322 Endo, Fujisawa City, Kanagawa Japan*
[2] *Keio University Quantum Computing Center, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa Japan*
[3] *Dept. of Information and Computer Science, Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, Japan*
[4] *Faculty of Environment and Information Studies, Keio University, 5322 Endo, Fujisawa City, Kanagawa Japan*

Abstract. Existing quantum computer processors have topological limitations on the execution of CNOT gates. Error rates reduce the fidelity of execution results for the individual qubits and connection via CNOT. The problem of which qubit variable on the program is allocated to which qubit on the processor has a strong influence on the feasibility and fidelity on the current medium-sized and noise-unequal processors. In this research, we are creating software for QISKIT that assigns quantum bit variables to quantum bits of the processor to execute programs on existing IBM quantum processors.

In this study, we evaluate the error of each qubit on the processor used for optimization evaluation function by randomized benchmarking.

Keywords: QISKIT, QASM, IBM Q, Randomized Benchmarking, Qubit Allocation, Qubit Mapping

1 Problems in Adapting Programs to Real Machines

1.1 Processor Topology

The number of qubits in the processor of a quantum computer is increasing day by day [1]. However, it is difficult to generate the entangled state in any physically non-adjacent qubits on the processor. In fact, any processor laid out in 2-D and without a common bus, including all processors created by IBM, are limited in the use of CNOT gates.
This graph of limitation of use of the CNOT gate is referred to as the processor topology. For example, Fig. 1 show the topology of IBM's 20-qubit machine named Austin[2].[1]
Also, as can be seen from this topology, the number of qubits that can use CNOT varies for each qubit. The right side of Fig. 1 shows how many CNOTs can be used for each qubit, corresponding to the node degree.

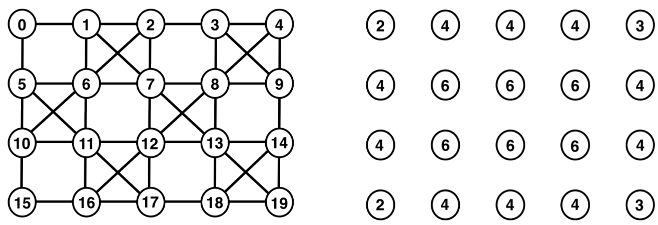

Figure 1: left: IBM Q20 Austin's topology. right: The number of qubits with which each qubit can execute a CNOT corresponding to the node degree.

1.2 Errors and Error Propagation

If we run a quantum circuit in the current IBM processor, the following errors will occur stochastically and the fidelity of our final state will go down.

1. Gate error
   Error occurring when unitary gate is executed for a single qubit.
2. SPAM error (State Preparation And Measurement Error)
   Errors that occur when observing qubits. Often collected with state preparation.
3. Bi-Qubit error
   Errors that occur when using a control gate such as a CNOT gate for 2 qubits.

2 Computational complexity of Quantum Compilation

When evaluating errors accompanying quantum computation with the model as described above, searching a space that is $O(Q! \times 2M)$ is necessary in order for the quantum compiler to calculate the highest fidelity of circuit with M CNOT gates in a processor with Q qubit having variations in fidelity[3]. Since the increase in the number of qubits is assumed to increase the volume of calculation is not practical. In order to perform a heuristic search, there are methods such as beam search and inputting the initial correspondence between logical qubits and physical qubits. In beam search, we discard any search that predicts that the fidelity will fall below a certain level. If the number of branch is B and the number of inputs of correspondence between logical qubits

* parton@sfc.wide.ad.jp
† pandaman@am.ics.keio.ac.jp
‡ satoh@sfc.wide.ad.jp
§ rdv@sfc.wide.ad.jp

[1] This connectivity graph for the current generation of IBM processors is not planar, but the couplers are still limited in distance and number to a very small neighborhood. We will loosely refer to such an arrangement as 2-D in this paper.

and physical qubits is K, the computational complexity for N gates is as follows.

$$O(NKQ^2 + N^2Q^2B\log B)$$

3 Gate error estimation

3.1 Randomized Benchmarking

In section 1.2, we discussed gate errors as a factor we must account for in our compilation. This presumes that we know what those errors are, but in fact determining magnitude of those errors is a complex process involving many tests on the actual processor. Complete characterization of an n-qubit system requires $O(e^n)$ tests, so researchers are investigating more efficient heuristics that will give us useful information. In this paper, we have chosen to implement and test randomized benchmarking (RB)[4]. We are performing RB on the IBM Q Tokyo (20 qubits processor). The random benchmark evaluates the qubits on the processor assuming that the magnitudes of polarization of the multiple Clifford gates are equal. In addition, by changing the number of Clifford gates to be executed and executing it multiple times, it is possible to obtain the value of the gate error as the attenuation factor. At this time, it is an advantage over the process tomography that the SPAM error and the gate error can be evaluated separately. The procedure of the RB performed this time is as follows. It is shown in Fig. 2.

1. Prepare $|0\rangle$ state
2. Select Clifford gates at random.
   Randomly select and arrange m gates from the Clifford gate set. There are 24 1-qubit Clifford gates.
3. Return the state to $|0\rangle$
   Create a gate row in which the reverse operation of the gate selected in Step 2 is arranged from behind.
4. Measurement
   Measurement the state and get qubit's state. If we get $|0\rangle$, the calculation can be regarded as a success. If we get other than $|0\rangle$, the result can be regarded as an error.

These process can be written as follows.

$$C_{m+1} = (\prod_{i=1}^{m} C_i)^{\dagger}$$

$$\prod_{i=1}^{m+1} C_i = I$$

In this research, we are planning to do RB for all qubits in the IBMQ 20Tokyo processor. We are currently working to determine the necessary precision and rate of convergence in order to minimize required execution time.

3.2 Interleaved Randomized Banchmarking

Interleaved Randomized Benchmarking is a way to characterize error of a unitary gate. [5] The procedure is as follows. It is shown in Fig. 3.

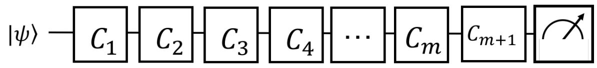

Figure 2: Randomized Benchmarking

1. Prepare $|0\rangle$ state
2. Select Clifford gates at random.
   Randomly select and arrange m gates.
3. Insert the gate which we want to characterize in an alternating fashion.
4. Reverse the state
5. Create a gate sequence in which the reverse operation of the gate selected in Step 2 is arranged.
6. Measurement
   Measure the state and get qubit's state. If we get $|0\rangle$, the calculation can be regarded as a success. If we get $|1\rangle$, the result can be regarded as an error.

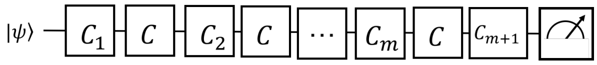

Figure 3: Interleaved Randomized Benchmarking

4 Conclusion

Randomized Benchmarking helps us performe good beam search and build on effective, high performance compilier useful for NISQ era systems with intermediate numbers of qubits and variable fidelity. The key challenge is tradeoff between the fidelity of the benchmarking and the execution time.

Acknowledgements

We would like to thank Shota Nagayama and Takanori Sugiyama for useful discussions.

References

[1] John Preskill, Quantum Computing in the NISQ era and beyond, 27 January 2018, arXiv:1801.00862v2 (2018)

[2] IBM, IBMQ experience Device, `https://quantumexperience.ng.bluemix.net/qx/devices` (accessed 2018 06 07) (2018)

[3] Toshinari Itoko, Rudy Raymond, and Takashi Imamichi , Minimum Cost Mapping of Quantum Circuits to Quantum Computers (2018), JSAI2018(32th), DOI https://doi.org/10.11517/pjsai.JSAI2018.0_1M303 (In Japanese)

[4] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, D. J. Wineland, Randomized Benchmarking of Quantum Gates (2007), arXiv:0707.0963

[5] Easwar Magesan, Jay M. Gambetta, Joseph Emerson, Characterizing Quantum Gates via Randomized Benchmarking (revised 27 Apr 2012), arXiv:1109.6887