An Automated Tool for Mapping Program Variables to Qubits on the IBM Q Topologies

Shin Nishio *, Takahiko Satoh†, and Rodney D. Van Meter ‡

Existing quantum computer processors have **topological limitations on the execution of CNOT gates**. Qubit and connection errors lower fidelity of execution result. Programmers need to generate programs that are high in fidelity and adapted to topology, but this is difficult. It is possible to **apply graph embedding** and solve the **shortest path problem** in which the adjacency matrix containing **the topology and error rate information is taken as the host graph and the CNOT utilization requested by the program to be executed is incorporated as a guest graph**. By choosing the shortest path here, programmers can **create a program with higher fidelity with executable Qubits allocation**. This research reduces waste of a lot of human resources, and programmers can obtain more high fidelity execution results.

Ⅰ. The Circuit Design Limitation Caused by 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, all processors created by IBM are limited in the use of CNOT gates. A CNOT has a control qubit and a target qubit, but in the IBM quantum processors, the choice of qubit for each role is constrained.
This graph of limitation of use of the CNOT gate is referred to as **the processor topology**. For example**, Fig. 1** and **Fig. 2** show the topologies of IBM QX2 and QX4[2]. We have created tools for performing similar mapping [3] in prior work [4]. That work dealt with non-neighboring qubits, but not gate polarity.

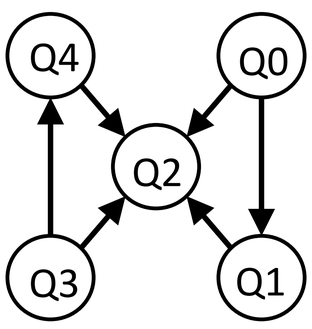

FIG. 1.
The Topology of IBMqx2

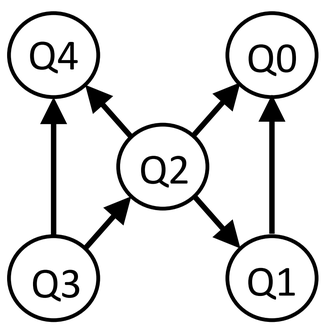

FIG. 2.
The Topology of IBMqx4

IBMQX2 and IBMQX4 are processors with the same number of qubits.
The direction of the arrows indicates the control (tail) and target (head) of CNOT gates.

Suppose that a program is generated so that it can be executed by QX2. To use this program with QX4, you need to renumber the qubits for the QX4 topology shown in **Fig. 3** and **Fig. 4**. In this case, there is no change in the circuit complexity, but **depending on the topology**, the circuit complexity may change due to the occurrence of a bottleneck. Since the error rate increases as the circuit complexity increases, programmers want to reduce the circuit complexity if possible.

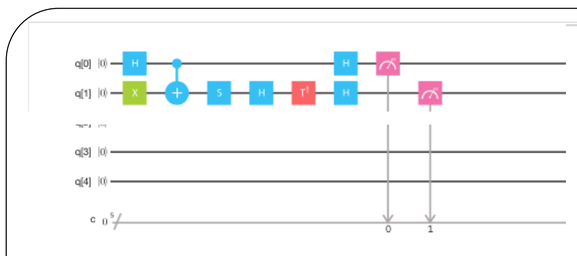

FIG. 3. A Program Made For IBMqx2

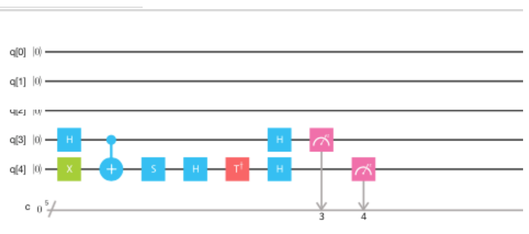

FIG. 4. A Program Made For IBMqx4

Currently, for the programmer to allocate the variable to be used in the program to qubits on the hardware, it is **necessary to refer to the topology**. However, in order to deal with large-scale code and **enhance reusability of various programs** on different hardware, this act should be automated.
In order to obtain the calculation result with the highest fidelity, it is **necessary to consider error rate of each qubit and the topology of the processor**, but this should also be automated. Automation for efficient use of resources like a compiler in a classical computer is required.

References

[1] J. Q. You and Franco Nori, Superconducting Circuits and Quantum Information, Phys. Today 58 (11), 42 (2005) .
[2] IBM, IBM Q experience Device, https://quantumexperience.ng.bluemix.net/qx/devices (accessed 2018 01 15) (2017)
[3] Kaori Ishizaki, An algorithm for optimizing movement of quantum variables on arbitrary physical qubit structures, Bachelor's thesis, Keio University (2011)
[4] Choi, Byung-Soo and Van Meter, Rodney, On the Effect of Quantum Interaction Distance on Quantum Addition Circuits, J. Emerg. Technol. Comput. Syst, August 2011, 7, 3, 1550-4832, 11-17 (2011)

Ⅱ. Mapping Program Variables to Qubits

This algorithm adapts the variables of the program to the topology to be high fidelity. The system for mapping program variables to quantum bits has the following procedure.
1). Graph the topology necessary for the program to be executed. This is a **guest graph**.
2). Convert the topology of the processor that wishes to execute the algorithm and its error rate to an adjacency matrix. This is a **host graph**. As a guide for programmers, IBM published the error rate of each qubit on each processor. Fidelity prediction values of the whole program can be derived by using the circuit complexity and these numerical values. At this time, the following three error rates exist which showed in **Fig. 5.**
a). Error accompanying operation of single qubit (**Gate error**)
b). Error accompanying the observation of a single qubit (**Measurement error**)
c). Error accompanying operation on multiple qubits (**Bi-Qubit gate error**)

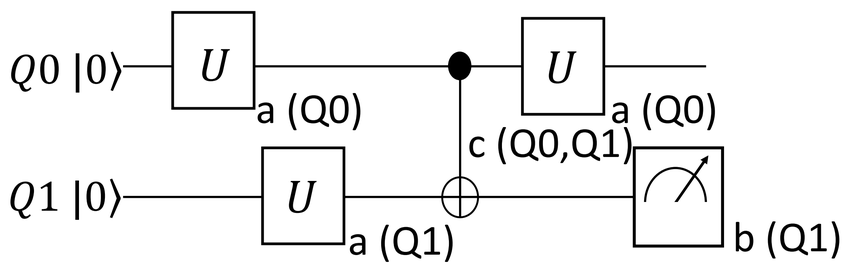

FIG. 5. Gate error(a), Measurement error(b), Multi-Qubit gate error

In our host graph adjacency matrix, Gate error (G) and Bi-Qubit error (B) can be expressed as weighting on the matrix diagonal. Also, Multi-Qubit gate errors can be represented as a weighting between two qubits. **Fig. 6** is the adjacency matrix of IBMqx2.

$$
\begin{array}{l}
\quad\quad Q0 \quad Q1 \quad Q2 \quad Q3 \quad Q4 \\
\text{adj} = \{[G_0 \ , B_{01} \ , B_{02}, 0 \ \ , 0 \ \ ], \quad \#Q0 \\
\quad\quad [0 \ \ , G_1 \ \ , B_{12}, 0 \ \ , 0 \ \ ], \quad \#Q1 \\
\quad\quad [0 \ \ , 0 \ \ \ , G_2 \ , 0 \ \ , 0 \ \ ], \quad \#Q2 \\
\quad\quad [0 \ \ , 0 \ \ \ , B_{32} \ , G_3 \ , 0 \ \ ], \quad \#Q3 \\
\quad\quad [0 \ \ , 0 \ \ \ , B_{42} \ , 0 \ \ \ , G_4 \ \ ] \quad \#Q4 \ \}
\end{array}
$$

FIG. 6.
The adjacency matrix of IBMqx2

3). Generate an embedded graph containing the guest graph in the host graph. At this time, the algorithm of shortest path problem can be used.
4). Output a program in which quantum variables are re-allocated based on the embedded graph.
5). Output **the estimate of the fidelity** of the qubit observed by that program.

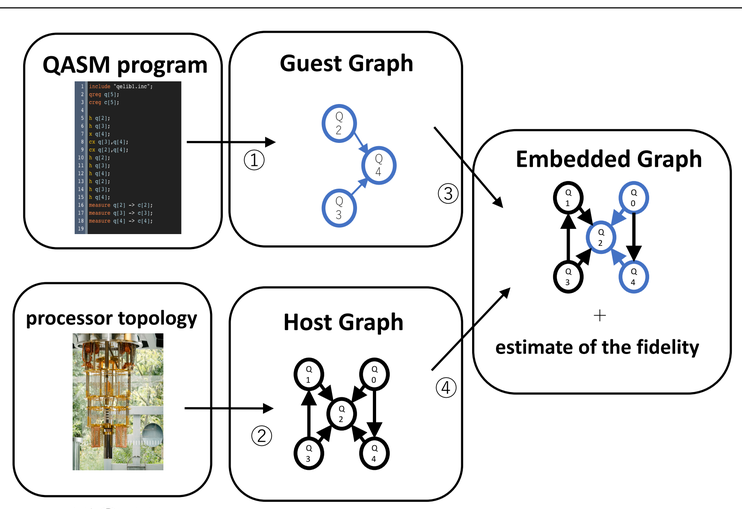

FIG. 7.
Mapping Program Variables to Qubits

International Conference on Challenges in Quantum Information Science, April 9-11, 2018

*Keio University Faculty of Policy Management
†Keio University Graduate School of Media and Governance
‡ Keio University Faculty of Environment and Information Studies

Keio University 1858 CALAMVS GLADIO FORTIOR

AdvancingQuantumArchitecture

http://aqua.sfc.wide.ad.jp