Weekly Report 07/10/2018

Yonggan Fu

1、 Reading List

- 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.
- A. Zulehner, A. Paler, and R. Wille, "Efficient mapping of quantum circuits to the ibm qx architectures," arXiv preprint arXiv:1712.04722, 2017.
- Swamit S. Tannu and Moinuddin K.Qureshi, "A Case for Variability-Aware Policies for NISQ-Era Quantum Computers", arXiv:1805.10224, 2018.
- Damian S. Steiger, Thomas Häner and Matthias Troyer, "ProjectQ: an open source software framework for quantum computing", arXiv:1612.08091v2, 2018.
- Jeff Heckey, Shruti Patil, Ali JavadiAbhari, Adam Holmes, Daniel Kudrow, Kenneth R. Brown, Diana Franklin, Frederic T. Chong, and Margaret Martonosi, "Compiler Management of Communication and Parallelism for Quantum Computation", SIGPLAN Not. 50, 4 (March 2015), 445-456, 2015.

2、 Ideas

- **Take reversal and bridge transformation into consideration.**

All the papers analyze the mapping procedure based on swap numbers but incorporating reversal and bridge transformation can reduce the total cost further. (Cost is defined to be the gate number to perform CNOT operation.)

It's obvious that the cost of reversal is less than swap. We need to analyze the cost of bridge. I discover that the edge direction between physical qubits influence the bridge cost greatly. To be more specific, we can assume there's 3 physical qubits A, B, C in a line, A connecting with B and B connecting with C. The next operation is CNOT(A, C). There are 4 possible situations of the combination of the two edges' direction between A, B and B, C:

(1) A -> B -> C (2) A <- B -> C (3) A -> B <- C (4) A <- B <- C

Based on the property of Hadamard gate that HH=I (identity matrix), under certain circumstances H gate can offset each other. After drawing the circuits of the 4 situations above, we can compute the cost of different transformations to perform CNOT(A, C) in different situations:

(1) bridge_cost = 4 , swap_cost = 7+1 = 8 ;
(2) bridge_cost = 10 , swap_cost = 8 ;
(3) the same as (2)
(4) bridge_cost = 10 , swap_cost = 2*7+1 = 15 , swap+reversal_cost= 7+4+1 = 12

It shows that under situation (1) and (4), using bridge transformation has better effects than the original strategy.

Limit:

After talking with Gushu Li on the phone, I realize that in IBM qx2 (5 qubits) and IBM qx3 (16 qubits), the direction of each edge is indeed a problem. However, after IBM qx4 (20 qubits), all the edges will become bidirectional, so the edge direction won't be important any more. But the analysis above can be practical for unidirectionally connected hardware and bridge transformation can reduce the cost on IBM qx4 too.

What's more, the error rate of bridge transformation needs to be considered.

- **Fix the most frequent pseudo qubits on the strongest physical qubits.**

The heuristic solution provided by the papers introduce an initial mapping method. The initial mapping is just based on the count of appearance in Φ (dependency sequence) and map the control qubits to the physical qubits with similar out-degree.

There exists a big problem when extending the initial mapping to handle Φ. For instance, A is the pseudo qubit which appears to be the most frequent control qubit in Φ. In the original initial mapping, A tends to be mapped to the strongest physical qubit with largest out-degree. But before A controls other qubits, (B, A) appears in Φ and B is a rare qubit with low frequency. Then A is moved away from its original position through swap.

Similar problems may occur when swapping B to its destination with A staying in B's moving path. The most important A may be moved by other unrelated swaps.

To solve the problem, I think that we can fix the most frequent pseudo qubits on the specific physical qubits (the strongest ones). With bridge transformation, they can cover a large area. In this way the most important qubits can never be moved from their initial position which can perpetuate their fan-out capabilities.

There are several requests for the fixed mapping qubits. First, the fixed qubits can operate with each other through reversal or bridge transformation. This limits the number of fixed qubits. What's more, the fixed part cannot break the connectivity of the whole architecture or it will obstruct the swap transformation of the unfixed part. The architecture of IBM qx3 and qx4 offer several alternative positions of fixed qubits.

- **Analyze the sequence of Φ more deeply.**

The heuristic solution in the paper did not analyze the character of the dependency sequence Φ deeply. Similar to the Longest Path First Schedule algorithm proposed by the Muti-SIMD(k, d) architecture, we can also analyze the sequence of appearance in Φ from this aspect. For instance, we can define pseudo qubit A's path to be the number of dependency pairs (A, X) before the first (Y, A) shows in Φ. We can find the k longest path and map the k pseudo qubits to the k strongest physical qubits.

Other patterns of Φ can also be analyzed to find a better mapping solution. All the efforts can help solve the problems mentioned above.

3、 <u>Deliverables</u>

- Builded the environment of QISKit, a Python package for quantum algorithms and circuit design offered by IBM.
- Ran the simplest quantum circuits on IBM qx2 hardware.

（But I cannot run the program on the local qasm simulator and I'm still trying to figure out the reason.）

4、 <u>Plan</u>

- Verify my initial ideas on qubit allocation by discussing with prof. Ding and doctors in the group and dig it further.
- Find papers about analyzing the error rate of quantum gates, both one-qubit gates and two-qubit gates. Try to build a better model to depict the error of each gate in the hardware architecture.