Proposal Document

Hayden Coffey, Kanghee Park, Saurabh Kulkarni

April 22, 2022

1 Problem

We are proposing a tool that can take a quantum assembly program and, given a specific qubit mapping algorithm and the characterization of a target machine, output the estimated error rates and fidelity of the qubits used in the program. This would involve collecting accurate characterization data for the target machine, such as gate error rate, noise, measurement error, gate latency, and topology-based variation in the aforementioned values.

2 Importance

This tool, if modularized, could be used to select a software/hardware stack that is optimal for the written assembly code. It would also give real-time feedback to the programmer when writing code, which would allow the programmer to evaluate whether to approach their problem in a different way. A fast and accurate error rate and fidelity modeling tool would also reduce the amount of machine time used to gather this information, thus reducing project expense.

3 Baseline Implementations

1. Qiskit Noise Simulator: We will be using IBM Qiskit's built-in backend noise simulator. The goal is to match or exceed the accuracy of Qiskit's noise model. We plan to use real hardware benchmarks to verify this.
2. Tket: Open source Qubit mapping framework. Tket advertises itself to make programming on NISQ devices easier by providing tools that mitigate noise, along with a more powerful instruction set and compiler. They claim to do all this while allowing the programmer to remain unaware of the idiosyncrasies of the device their program will run on. While we intend to break some of the abstraction Tket affords the programmer, it seems that at least some of the problems we intend to solve have been tackled by them.

4 Action plan/ Timeline

1. As our project will require characterization and mapping data, our first objective is to read documentation to learn how to collect and parse characterization data for the IBM servers, as well as familiarize ourselves with the Qiskit noise model and a mapping framework such as Tket.
2. Once we have a background understanding of the tools we will be working with, we plan to develop a control program which produces a baseline fidelity calculation for a given quantum circuit using the Qiskit noise model.
3. With a baseline established, we aim to move on to creating our experimental program. Developing a mapping function to parse IBM Characterization data and return an ideal mapping for a given quantum circuit would be the first step in this direction.

(a) Tket could function as a base mapping framework for this and simplify our work. Since it is open source we can explore how it determines mappings and if the algorithm used can be changed.

4. Write a program which, given a quantum circuit, characterization data, and a mapping, predicts circuit fidelity.

5. Evaluate our experimental program by comparing the fidelity prediction from it and the Qiskit noise model baseline to fidelity values on actual hardware. We believe this to be our minimum goal for the project.

6. After getting a sense of our program's performance against the baseline, we can focus on improving its speed and accuracy.

   (a) Optimizing Mapping Function: Using mapping algorithms such as SABRE could provide our program with the speedup necessary to be practical as a real time tool.

   (b) Characterization Data Collection: This application will rely on the characterization data provided by IBM. Finding a way to reduce the cost of collecting this data would be beneficial even outside the scope of the project. One option is to try collecting calibration data for qubits touched by a circuit mapping and accumulate this data to produce a characterization of the system rather than performing an intensive total system calibration.

7. Automate/Implement heuristics for the tool to return the ideal hardware and mapping alongside the predicted fidelity of a quantum circuit given a list of systems' configuration data. Testing different mapping algorithms could also be beneficial for finding the optimal circuit placement, but could cause the problem space to explode. A challenge for this stage will be to find a way to keep the computational cost of this search low and potentially parallelize the work.

8. As the final stretch goal, if we are able to develop a system that can determine optimal circuit placement and mapping across a set of quantum machines, we could work on creating a front end plugin for use with an IDE, enabling developers to view the information as they work.

5 Related Works

1. Pareto-Efficient Quantum Circuit Simulation Using Tensor Contraction Deferral: This paper proposes a methodology to simulate quantum circuits with many qubits. Also, it explains IBM's simulation strategy in detail. [6]

2. Modelling and Simulating the Noisy Behaviour of Near-term Quantum Computers: This paper proposes a method for simulating a quantum program by combining three error groups, which are the main sources of noise in quantum computing, and using a calibrated error rate. It is pretty recent work, and it seems to work well. This will also serve as a reference for simulation algorithm. [1]

3. Modeling quantum noise for efficient testing of fault-tolerant circuits: This paper proposes a method to efficiently test the quantum circuit with noise model though Monte Carlo simulation. It will help us figure out how to simulate the noise model in reasonable time. [4]

4. Stochastic Quantum Circuit Simulation Using Decision Diagrams: This paper proposes a way to reduce resource for quantum circuit simulation, using decision diagrams. Most simulation methods simplify the problem a lot, but this method reflects various errors that can occur in real quantum computing. It might be helpful to implement our simulation. [2]

5. Noise-Adaptive Compiler Mappings for Noisy Intermediate-Scale Quantum Computers: This paper proposes and evaluates various backend compiler algorithms that can be applied to NISQ with diverse hardware characteristics, from LLVM IR to a quantum program that can be executed on an actual IBM machine. In addition, this paper evaluates various methods through the implemented framework. This will help us figure out how to deal with multiple layers in NISQ. [5]

6. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices: This paper proposes a SWAP-based heuristic search algorithm as a solution to the qubit mapping problem. It claims that the method is applicable to NISQ devices with arbitrary connections between qubits. [3]

References

[1] Konstantinos Georgopoulos, Clive Emary, and Paolo Zuliani. Modeling and simulating the noisy behavior of near-term quantum computers. *Physical Review A*, 104(6):062432, 2021.

[2] Thomas Grurl, Richard Kueng, Jürgen Fuß, and Robert Wille. Stochastic quantum circuit simulation using decision diagrams, 2020.

[3] Gushu Li, Yufei Ding, and Yuan Xie. Tackling the qubit mapping problem for nisq-era quantum devices. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1001–1014, 2019.

[4] Easwar Magesan, Daniel Puzzuoli, Christopher E Granade, and David G Cory. Modeling quantum noise for efficient testing of fault-tolerant circuits. *Physical Review A*, 87(1):012324, 2013.

[5] Prakash Murali, Jonathan M Baker, Ali Javadi-Abhari, Frederic T Chong, and Margaret Martonosi. Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1015–1029, 2019.

[6] Edwin Pednault, John A Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, Erik W Draeger, Eric T Holland, and Robert Wisnieff. Pareto-efficient quantum circuit simulation using tensor contraction deferral. *arXiv preprint arXiv:1710.05867*, 2017.