ECE 382V: Introduction to Quantum Computing Systems from a Software and Architecture Perspective
Homework 2

Sayam Sethi

September 2023

Contents

1 Question 1

Gate Error Mitigation

Question. *What are the benefits of instruction reordering in quantum programs?*

Answer. Instruction reordering can lead to reduced number of swaps and hence an overall reduced circuit depth in the final circuit that is obtained after the entire compilation process. This also allows for increased possibilities of gate cancellation. Additionally, the reordered circuit might allow for more parallel computation that the original circuit which reduces the total error and the execution time of the entire circuit.

Question. *Compilers perform gate cancellations in two passes, before and after routing. Why?*

Answer. The `SWAP` gates introduced during the routing process introduce multiple possible gate cancellation opportunities. However, we still need to perform gate cancellation before routing since this reduces the circuit depth on which routing will be performed thus reducing the time taken by the routing algorithm (since it has to operate on a smaller input size). This can allow the routing algorithm to make more intelligent decisions which might have been impossible if the number of gates in the input were higher.

Question. *Gate nativization is a crucial step in program compilation. What are the trade-offs in nativization on systems that offer multiple native gates to decompose a high-level instruction? What are the drawbacks of the gate selection method proposed in the paper "The Imitation Game"?*

Answer.

Advantages: Different qubits might have different error rates for the possible native gates and hence it might be better to use a certain native set for one qubit (or a pair of qubits) and another for a different qubit (or a pair of qubits). Different native gates might also have different execution times, and sometimes it might be possible to squeeze in a native gate only from the subset of the possible native gates between two other gate operations since they take lesser time than the other native gates.

Disadvantages: Choosing the best native gate for each operation will require us to search from exponentially many possibilities which is computationally impossible for the compiler to do in a small amount of time. Making wrong or bad decisions can potentially hurt the performance more than just choosing from the default native gate set.

Drawbacks in "The Imitation Game":

1. The algorithms considers a single native gate for each link and doesn't account for using different native gates for each gate in the circuit on the same link.
2. Since ANGEL is a greedy algorithm, it might get stuck in a local minimum and not be able to find the optimal solution.
3. Additionally, it doesn't scale well with the circuit size since simulating the circuit becomes harder. The limit of 20 non-Clifford gates is suitable for small circuit sizes since the circuit can be simulated and will lead to a low-entropy output distribution. However, for larger circuits, the output distribution will still have a high entropy since 20 non-Clifford circuits won't be enough to reduce the entropy and increasing the limit will make it harder to simulate the circuit.
4. Device drifts during ANGEL's execution can lead to incorrect results.

Question. *Circuit cutting is an emerging error mitigation technique in NISQ-era. What are the overheads of circuit cutting? How can these overheads be reduced?*

Answer. Circuit cutting can lead to large number of sub-circuits which does not scale well with the circuit size. The knitting algorithm also becomes more involved and increases exponentially with the number of cut circuits. The problem of deciding how, where and when to cut the circuit is also an open research problem. The circuit execution overheads can be reduced by executing the smaller circuits parallelly on the same machine. Additionally, to reduce the classical overheads, application-specific knitting can be performed to make the knitting algorithm more efficient. Also, in the NISQ-era, approximate knitting suffices since the final output distribution would be noisy anyway. It is also possible to communicate the classical information between the runs of different sub-circuits to reduce the classical overheads.

2 Question 2

Measurement Error Mitigation

Question. *Why are measurement errors hard to eliminate at the device-level?*

Answer. Measurements are very sensitive to noise at all the different stages of the measurement process. At the lower levels, the noise is due to device imperfections (since maintaining low temperatures and prevent interference from the environment is hard). Being able to efficiently distinguish between the two states at the software level is a very sensitive process because of the accumulated noise and thus the classifier needs to be very accurate which is hard to achieve.

Question. *What are the drawbacks of the matrix-based measurement error mitigation techniques?*

Answer. The matrix-based measurement error mitigation techniques require a matrix whose size increases exponentially with the number of qubits since we require a noise matrix which has dimensions $2^n \times 2^n$ (where n is the number of qubits). Even for small input sizes, the problem of deciding on how to compute the noise matrix is an open problem. One needs to decide on how to characterize the noise and should be able to capture the system drifts at the right time.

Question. *What are the challenges in state-transformation-based measurement error mitigation? How to overcome these challenges?*

Answer. Deciding on when to insert `X` gates (search space) scales exponentially with the input size. Since simulating the circuit is not computationally feasible for larger inputs, it is also impossible to compare the performance of different circuit choices in the search space. One way of solving these issues is by constructing two circuits – the original circuit and another which has an `X` gate on all qubits. The output distributions can then be classically processed to obtain the resultant distribution.

Question. *JigSaw is a measurement error mitigation technique. It uses a recompilation step to measure the program qubits on physical qubits with the lowest measurement errors. How can you reduce this recompilation overhead?*

Answer. The recompilation overhead can be reduced by adding a threshold on the measurement error rate, i.e., the recompilation only happens if the measurement error rate for the qubits on which measurement is to be performed exceeds the threshold. Otherwise, we proceed with the same compiled circuit. This will significantly reduce the overhead since we can also intelligently compile the circuit giving preference to the qubits that have a lower readout error. Alternatively, we can compile a single circuit and define a set of qubits as the 'measure' qubits. We can then introduce `SWAP` gates at the very end to bring the qubits to be measured to these 'measure' qubits.

Question. *What are the trade-offs in designing measurement classifiers for quantum systems?*

Answer. Most measurement classifiers for quantum systems use Machine Learning. This leads to either a huge hardware overhead if we want the results of the classifier with low latency (IBM's classifier) or a huge latency and an increased software overhead if we want to reduce the hardware overhead (full neural network).

3 Question 3

Idle/Crosstalk Error Mitigation

Question. *Why are idle errors important to eliminate at the application-level?*

Answer. Idle errors can be a major source of error in the circuit since they increase exponentially with the idle time. Therefore, it is crucial to minimize the idle time of the qubits and this can be somewhat easily done by Dynamical Decoupling.

Question. *What are the trade-offs involved in concurrent CNOT scheduling?*

Answer. Executing `CNOT` gates concurrently can reduce the circuit depth and hence reduce the total decoherence error and the execution time of the circuit. However, this leads to increased crosstalk errors which is also a significant source of error in NISQ-era quantum computers.

Question. *What are the trade-offs in incorporating device-level techniques for reducing idle errors naively at the application-level?*

Answer. Performing Dynamical Decoupling cannot be done naively for all the qubits since different qubits have different error rates. It is only beneficial to perform DD on those qubits which actually reduce the total circuit error. This decision also depends on the device and the system drift and hence cannot be globally applied at the application-level and transported to any device. Additionally, performing DD can also lead to increased circuit depth and can potentially increase the crosstalk error (but not by much since DD involves single-qubit gates).

Question. *Is characterization the right approach to reduce crosstalk/idle errors? Why or why not?*

Answer. Since crosstalk and idle errors are device-specific, we need a way to be able to characterize the device and then decide on how to reduce the errors. The drawback of this approach is that the error mitigation techniques cannot be generalized at the application level and need to be associated with each device for each application. We would also need to re-characterize the optimized circuit if we want to run the application at a later time in the future since device drifts would change the characterization. Therefore, characterization is required, but the current characterization methods might not be the best ones as quantum computers scale in size and usage.

4 Question 4

Variational Quantum Algorithms

Question. *What are the potential benefits of variational quantum algorithms? What are some of their key drawbacks?*

Answer. Variational Quantum Algorithms (VQAs) offer heuristic-based speed-ups over the classical approach and thus have the potential to solve NP-hard problems faster than the classical computers. However, in the current state of quantum computers, these algorithms are very noisy to run and even on the best computers currently available commercially, they can be run only for inputs of very small sizes. Additionally, finding the optimal solvers for the quantum algorithms is an ongoing research problem which would need to be improved or perfect before we can use VQAs to solve real-world problems. However, VQAs are still the best path forward to prove quantum 'supremacy' on noisy system since they can handle small amounts of noise and do not require the exact output distribution to be obtained. Another concern is that there is no theoretical guarantee that these algorithms would be faster on a quantum computer since it is possible that a classical algorithm is designed in the future that beats the quantum advantage obtained via VQAs.

Question. *How can you use application specific properties to reduce the impact of hardware errors in variation quantum algorithms?*

Answer. Since the order of the `CPHASE` gates does not change the optimal result, we have increased opportunities for instruction reordering which would lead to increased parallel execution and reduced circuit depths. We can also have a low-overhead circuit cutting by 'freezing' (classically solving) the 'hotspot' nodes since these nodes would have increased error rates which would reduce the quality of the solution.

Question. *Google uses three types of graphs to study the performance of QAOA on real systems. What is the specific reason to choose the exact kinds of graphs chosen for this study?*

Answer. The three different kinds of graphs used are as follows:

1. Hardware Grid: This graph models the hardware connectivity in the graph. This leads to the best performance on physical systems since the only multi-qubit gates are between adjacent nodes and hence there is no requirement of `SWAP` gate insertion. Google was able to run QAOA on 23 qubits on the Sycamore processor using this graph.

2. 3-regular Graphs: These graphs have a theoretical guarantee on the quality of solution. Thus, we can easily compare the result obtained from the noisy algorithm with the optimal solution obtained from simulating the QAOA algorithm to see how noise affects the performance. Google was able to run QAOA on 14 qubits on the Sycamore processor using this graph.

3. Full Connected Graphs: These graphs represent the worst-case scenario for QAOA since the number of edges is maximum and hence the number of `SWAP` gates required to map the graph to the hardware connectivity is maximum. Google was able to run QAOA on 11 qubits on the Sycamore processor using this graph.

Question. *IBM recently introduced the Qiskit Runtime environment to run variational quantum algorithms. What are the benefits of this approach? What are the drawbacks?*

Answer. The integrated runtime environment allows for faster interaction between the quantum solver and the classical optimizer which makes the overall process faster. This also allows for application-aware optimization possibilities on both the device side and the application size. Additionally, based on the device and the application, the optimizer can make more intelligent decisions which would improve the results of the solver. The major drawback of this approach is that the flexibility from the programmer is reduced since the programmer has to send in the circuit and the solver at once and cannot make any changes to the circuit based on the results of the solver. This also makes it harder to debug the circuit since the programmer cannot see the intermediate results of the solver and hence cannot make any changes to the circuit based on the intermediate results. Additionally, there are security concerns with the optimizer running on the cloud rather than locally on the user's machine.