ECE 382V: Introduction to Quantum Computing Systems from a Software and Architecture Perspective
Homework 1

Sayam Sethi

September 2023

Contents

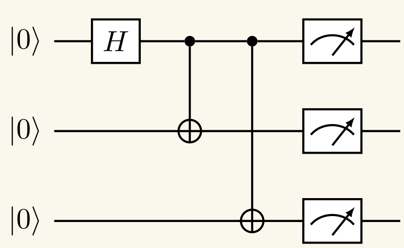
1 Question 1

Error Model

The impact of errors can be analytically modeled by using various parameters accounting for the failure probabilities corresponding to different types of errors. Let us estimate the probability of successfully executing a 3-qubit bell-pair circuit.

Question. *Estimate the error-rate assuming only gate and measurement errors occurs.*

Answer. The circuit for a 3-qubit bell-pair is given by,



Let the probability of a H gate be p_H, the probability of error of a CNOT gate be p_{CNOT} and the probability of error of a measurement be p_M. Then, the probability of successfully executing a 3-qubit bell-pair circuit is given by,

$$P_0 = (1 - p_H) \cdot (1 - p_{CNOT})^2 \cdot (1 - p_M)^3 \tag{1}$$

Note that we have an inequality since the errors in the gates and the measurements might cancel out.

Question. *Now append the error-model to include decoherence. How does the probability of successful circuit execution change?*

Answer. Let the amplitude decoherence time of qubit i be T_{1i}. Also assume that the time it takes for the Hadamard gate is T_H, the time it takes for the CNOT gate is T_{CNOT} and the time it takes to measure the qubit is T_M. Then, the probability of successfully executing a 3-qubit bell-pair circuit is given by,

$$\begin{aligned} P_1 =& P_0 \cdot P_{decoherence,0} \cdot P_{decoherence,1} \cdot P_{decoherence,1} \\ =& P_0 \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{10}}\right)\right) \\ & \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{11}}\right)\right) \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{12}}\right)\right) \\ \implies P_1 =& (1 - p_H) \cdot (1 - p_{CNOT})^2 \cdot (1 - p_M)^3 \\ & \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{10}}\right)\right) \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{11}}\right)\right) \cdot \\ & \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{12}}\right)\right) \end{aligned} \tag{2}$$

Question. *How can you further improve the accuracy of this error model? Estimate the probability of success by including at least one additional parameter.*

Answer. Some of the simplifying assumptions made in the above model are:

1. The errors are assumed to be additive. However, in reality, the errors from different gates and decoherences might cancel out and hence we might have a higher probability of success.
2. We have assumed that the probability of error of both CNOT gates is the same. However, in reality, the probability of error of the CNOT gate might be different for different pairs of qubits.
3. We have also assumed that the total time it takes to measure each qubit and the time it takes to execute each CNOT gate is the same. However, in reality, the times will be different and hence the decoherence probabilities will be different.

For this part, we will get rid of the simplifying assumption 2. The resultant error model becomes,

$$\begin{aligned} P_2 = & (1 - p_H) \cdot (1 - p_{CNOT(0,1)}) \cdot (1 - p_{CNOT(0,2)}) \cdot (1 - p_M)^3 \\ & \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{10}}\right)\right) \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{11}}\right)\right) \cdot \\ & \cdot \left(1 - \exp\left(-\frac{T_H + 2 \cdot T_{CNOT} + T_M}{T_{12}}\right)\right) \end{aligned} \tag{3}$$

Question. *What happens to the complexity of the error-model when you increase the circuit size beyond three qubits?*

Answer. As we increase the circuit size, the number of gates and measurements increase as $O(\mathsf{poly}(n))$, where n is the number of qubits. We can also have exponentially many combinations of multi-qubit gates and measurements, therefore the the error model can have $O(\mathsf{poly}(n))$ different terms. Thus, the probability of success will be negligible since it will be of the order of $O((\max p)^{\mathsf{poly}(n)})$.

2 Question 2

System Calibrations

Quantum systems are frequently calibrated to enable high fidelity quantum gate and measurement operations.

Question. *Why is system calibration non-trivial?*

Answer. System calibration is a non-trivial problem due to a lot of reasons:

1. The quantum systems are very sensitive to the environment and hence the calibration needs to be done frequently.
2. Each qubit is subjected to different operations and measurements and is surrounded by different neighbours. As a result, even if we want to apply the same gate to two different qubits, we will have to calibrate the gate differently for both the qubits.
3. For multi-qubit gates, we need to take into account the parameters for all the qubits involves in the gate. Therefore, the problem of finding the optimal gate parameters becomes exponentially hard as the number of qubits involve increase (since each qubit has multiple parameters involved, such as neighbours, idle gate frequencies, etc).

Question. *What are the trade-offs involved in too frequent system calibrations and infrequent system calibrations?*

Answer.
Too frequent system calibrations: If we calibrate the system too frequently, then we will not be able to perform any useful computation on the system. This is because the system will be in the calibration mode for most of the time and hence we will not be able to perform any useful computation on the system.
Infrequent system calibrations: If we do not calibrate the system frequently, then the system errors will continue to accumulate and hence the results will not be accurate. This would make the computation useless.

Question. *Most device providers opt for localized recalibrations as opposed to full-system calibrations. Why? How does this impact the "performance" of the system?*

Answer. A full-system recalibration will be computationally much heavier than a localized recalibration. Additionally, we will have to perform a full-system recalibration at the required frequency of the most error-prone regions. This would lead to useless computation for those qubits that do not require calibrations that frequently. Performing a calibration taking all qubits (and their neighbours) into account would also lead to a more complex calibration process since it would consider a larger set of parameters. Instead, a localized calibration process will be computationally more efficient even if we consider the computaiton done per qubit (since we assume that the parameters of the qubits outside of the neighbourhood are fixed).

Question. *In the class, we discussed the snake optimizer routine for performing large-scale system calibrations. What are the potential drawbacks of this technique? How do you think this can be improved?*

Answer. The snake optimizer routine is a greedy algorithm and hence it might get stuck in a local minima. This can be improved by using a more sophisticated optimization algorithm such as gradient descent. Additionally, we can also use a more sophisticated cost function that takes into account the errors of the neighbouring qubits as well. This would lead to a more accurate calibration process.

3 Question 3

Compilation: Qubit Mapping and Routing

Compilers translate a sequence of high-level instructions (program) into a functionally equivalent sequence of low-level native gates (assembly). Qubit mapping and routing are two fundamental steps involved in this process.

Question. *How does the quality of initial qubit mapping or allocation impact the performance of the quantum computer for a given application?*

Answer. The initial qubit mapping determines the relative positions of the qubits. This is crucial to the routing decisions since qubits have limited connectivity, usually in a grid-like layout. Thus, if a pair of qubits that have a lot of 2-qubit gates between then are located far away, we would require a larger number of swap operations than if they were located close by.

Question. *How does the routing policy impact the quality of the solutions for a given application?*

Answer. A bad routing policy would require more swap operations for each two-qubit (or multi-qubit) gate to place the associated qubits next to each other. On the other hand, a good routing policy would reduce the number of swap operations required significantly thus improving the quantum program run time. This would also reduce the decoherence error since a reduced execution time between measurements would reduce the effects of decoherence. Additionally, lesser gates also implies lesser accumulated errors thus increasing the program accuracy.

Question. *There exists numerous routing policies in the compilation space. Why? Which would you like to opt for in order to run any given program of your choice?*

Answer.

Question. *What are the trade-offs involved in routing overheads and device topology?*

Answer.

4 Question 4

Pulse Compilation

Most quantum algorithms can be described with circuit operations alone. When we need more control over the low-level implementation of our program, we can use pulse gates. Pulse gates remove the constraint of executing circuits with basis gates only, and also allow you to override the default implementation of any basis gate. Programmers can leverage this property to directly compile their programs into a series of pulses. Often these pulses are more compact compared to the pulses generated using only the basis gate set. For example, a SWAP pulse is typically shorter than the pulse generated by concatenating the CNOT pulse thrice.

Question. *What are the potential advantages of this approach?*

Answer. The advantage to this approach is that we can reduce the program execution time significantly which will also lead to reduced errors. This also increases the scope of optimization since we have a larger set of pulses to choose from than we would have if we used the basis set.

Question. *What are the key limitations in this approach?*

Answer. The drawback of this approach is that the compiler code will become more complex and it will take more time to execute. Additionally, if we have more pulses that we can use, this can lead to difficulties in the calibration of the system since we will have to choose more pulses from the same frequency band and thus the problem of choosing wide enough pulses becomes harder.

Question. *How do you think a programmer can best exploit this trade-off to their advantage?*

Answer. A programmer can inherit the flexibility of the pulses while not increasing the complexity of the compiler and the calibration algorithm by choosing different sets of pulses for different qubits. Not all qubits need to be implemented using pulse compilation. For example, the programmer can only choose the qubits that have a lot of swap operations to be implemented using pulse compilation and the other qubits can be implemented using the basis set. This will reduce the complexity of the compiler and the calibration algorithm while still giving the programmer the flexibility of the pulses.

Question. *Can you transfer the pulse schedule from one machine to another?*

Answer.