Explaining the Paper: Quantum Error Correction Below the Surface Code Threshold

Ricard Santiago Raigada García

August 30, 2024

Contents

1 Introducction

Often computational problems are increasingly complex, either due to the computational level required, the type of processing, or because of dealing with intractable dimensionality data. In the last decade, and since the introduction of GPUs for the common user, many of these problems have become easily solvable. Especially in recent years, with the enhancement of machine learning methods. Typically, the complexity of the problems is NP-Hard. This type of problem can be found in complex optimization systems such as finance, logistics, or transportation. Typically, it is believed that quantum computers are somewhere between what is known as P problems and PSPACE. Specifically, BQP-type problems; however, the reality is that the real limits of quantum computing are still unknown, and in any case, conventional computers continue to demonstrate exceptional performance.

What is clear, at least in the coming years, is that both types of computing are not mutually exclusive. This forces quantum computing to define a clear working area where it truly demonstrates what has been called quantum supremacy. However, the main enemy to finding a functional place for quantum computers is the computation time they can endure without errors.

The average coherence time of a qubit is $\mu s \quad (\sim 0.00001 \quad \text{sec})$. This is an approximation because it depends on the construction, whether they are ions, superconductors, neutral atoms, which will also influence the ability to perform operations. The conclusion is that over time, systems lose information or, as physicists like to say, entropy increases. Moving from an ordered and coherent system to work on, to a state of decoherence where any calculation will be erroneous. It is evident that current quantum computers do not seem prepared for the use to which classical computers are currently subjected. However, I don't believe this is the goal of most quantum computing researchers either. Nor should we dramatize; all beginnings were difficult, and until nanofabrication techniques were applied to make the transistors of classical computers, computing wasn't the same either.

Classical computers have ways to detect errors and dynamically correct them. Thus, for example, memories do not become corrupted or enter an intractable state. Similarly, as many have been pointing out for years, it seems that the way forward is to create quantum error correction systems. So they can extend the qubit's lifespan. The purpose of this article is to discuss the publication of Google Quantum AI Acharya et al., 2024 where high-distance surface codes are used to suppress the logical error rate below the critical threshold. For this, I will review some basic concepts to extend the knowledge and end by simply presenting the conclusions of the Google lab research.

2 Fundamentals of Quantum Error Correction

Paraphrasing Google's famous research: Attention is all you need. Error correction is all you need. It is essential due to the fragility of quantum computers to preserve information. You might wonder why not do it like with bits or telecommunications networks that multiplex signals or duplicate bits, thus extending the lifespan of a qubit. Well, a qubit cannot be duplicated. Think of it this way: imagine I give you a box with a canvas and ask you to replicate the painting on a new canvas. It is impossible if you don't know its state beforehand. Basically, this metaphor simply describes what is known as the no-cloning theorem. So, what is done is to distribute quantum information across several physical qubits to form a logical qubit, which is more resistant to errors.

You can consider qubits to be physical and individual elements, for example, a physicist within a group of scientists. The probability of making a mistake is high, especially considering they might not realize the mistake. In contrast, the logical qubit is all the individuals in the scientific group. They cooperate with the goal of correcting any errors made by one of their members to maintain high-quality scientific output.

2.1 The Stabilizer Formalism

The stabilizer formalism is a mathematical tool that allows for the design and analysis of quantum error correction codes. This formalism is based on the Pauli group, which for a single qubit includes the operations I (identity), X (bit-flip), Y (bit-flip followed by a phase-flip), and Z (phase-flip). The Pauli group for n qubits, denoted G_n, is the set of all tensor products of these Pauli matrices. A stabilizer is a set of operators that leaves a subspace of quantum states invariant. If a quantum state is "stabilized" by a set of operators, it means that the state remains unchanged under the action of those operators. Therefore, it is a method used to design and implement quantum error detection and correction.

One of the most basic and well-known methods is the 3-qubit code for correcting bit-flip errors. The stabilizers Z_1Z_2 and Z_2Z_3 can detect if a bit-flip error has occurred by measuring the parity between pairs of qubits. If the parity changes, an error is detected and can be corrected by applying the inverse operation to return the qubit to its state before the error.

2.2 Quantum Error Correction Codes

Among the most well-known and common are:

- 3-QB code for bit-flip errors. Where the parity of adjacent qubits is measured as mentioned earlier.
- 3-QB phase-flip code. The stabilizers X_1X_2 and X_2X_3 are used.
- 5-QB Laflamme code. It is the most concise code to correct an arbitrary Pauli error, that is, bit-flip, phase-flip, or their combination. It is defined as $g_1 = XZZXI$, $g_2 = IXZZX$, $g_3 = XIXZZ$, and $g_4 = ZXIXZ$. For those who find it hard to remember, memorizing g_1 is sufficient since the others shift one position. This code allows detecting and correcting any error in one of the five qubits.
- 7-QB Steane code. With this code, bit-flip and phase-flip errors can be corrected simultaneously.
- 9-QB Shor code. This code is a block construction that protects against bit-flip and phase-flip errors. Stabilizers that are products of the X and Z operators are used.
- 17-QB surface code. This code promises greater error tolerance and scalability.

2.3 Error Threshold in Quantum Error Correction

In error correction in the quantum world, we refer to the threshold as the physical error rate below which quantum error correction exponentially suppresses the logical error rate by incorporating more qubits into the code. The relationship between the physical error rate p and the logical error rate ε can be expressed as:

$$\varepsilon \propto \left(\frac{p}{p_{thr}}\right)^{\frac{d+1}{2}} \tag{1}$$

d is the code distance and p_{thr} is the code's error threshold. Thus, it follows that if $p < p_{thr}$, the logical error rate decreases exponentially with d. This means that very precise logical qubits can be built using enough physical qubits. This same expression to describe the behavior is the same as the one found in the introduction of Google's publication.

3 Stabilizer Formalism in Quantum Error Correction

The stabilizer formalism is the way to mathematically describe error detection and correction in quantum systems. Each code is associated with a set of stabilizers that defines its code space. Additionally, errors in the system are identified by measuring syndromes, that is, the result of applying the stabilizers to the quantum system.

3.1 Stabilizers in 3-QB Codes

The 3-qubit code for correcting bit-flip errors uses the stabilizers Z_1Z_2 and Z_2Z_3. These measure the parity between pairs of qubits:

$$Z_1Z_2 |\psi\rangle = \pm 1 |\psi\rangle \quad Z_2Z_3 |\psi\rangle = \pm 1 |\psi\rangle \tag{2}$$

Definitely, when there is a state and a bit-flip error occurs, the parity value changes. Thus, it is known that an error has occurred and which qubit is affected, so it can be corrected. To do this, the inverse operation is applied to return the qubit to its state before the error. For phase-flip errors, the stabilizers X_1X_2 and X_2X_3 are used to detect a phase change between qubits.

3.2 5-QB Laflamme Code

This is a more advanced method for quantum error correction that can correct any Pauli error. Recall the stabilizers described earlier: $g_1 = XZZXI$, $g_2 = IXZZX$, $g_3 = XIXZZ$, and $g_4 = ZXIXZ$. If a bit-flip error occurs in the first qubit, the stabilizer g_4 will anti-commute with the error operation, generating a syndrome that indicates the presence of the error. The logical stabilized states for this code can be expressed as:

$$
\begin{aligned}
|0\rangle_L &= \frac{1}{4}(|00000\rangle + |10010\rangle + |01001\rangle + |10100\rangle + \cdots + |00101\rangle) \\
|1\rangle_L &= \frac{1}{4}(|11111\rangle + |01101\rangle + |10110\rangle + |01011\rangle + \cdots + |11010\rangle)
\end{aligned} \tag{3}
$$

3.3 9-QB Shor Code and 17-QB Surface Code

Examples of these error correction codes are beyond the scope of this article. However, it is worth noting that the Shor code combines multiple 3-qubit codes to correct both bit-flip and phase-flip errors. It has a modular structure, making it easier to implement in quantum hardware. On the other hand, the surface code is interesting due to its potential to scale to large systems. It uses interactions between neighboring qubits, which simplifies implementation in physical quantum architectures.

4 Analysis of the Google Quantum AI Research

The publication describes an advance in quantum error correction using high-distance surface codes, specifically distances-5 and -7, on a superconducting processor (transmon). This is because surface codes are suitable for qubits arranged on a planar architecture. Personally, I believe that this research will be revealed as a major breakthrough over the years. It demonstrates the suppression of the logical error rate below the critical threshold, indicating that it is possible to build fault-tolerant quantum computers.

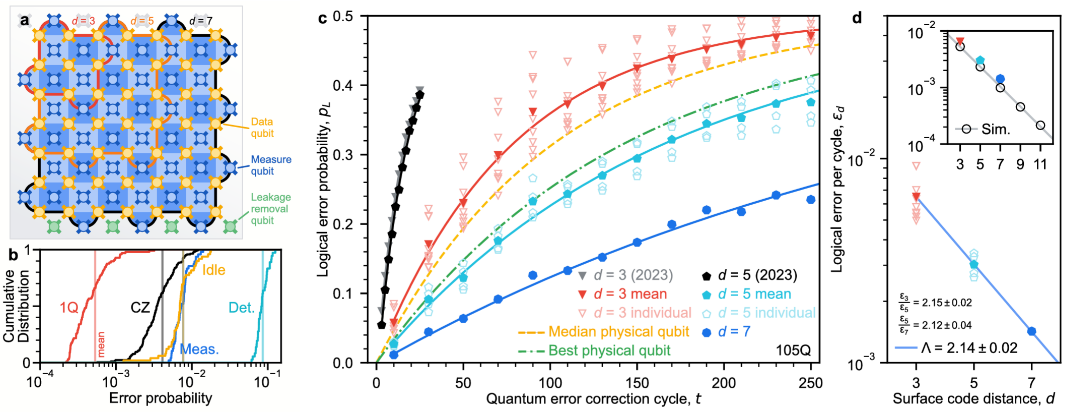

Figure 1: FIG.1. Surface code performance

Figure 1a shows the schematic of the distance-7 surface code with 105 qubits. Each measurement qubit is represented in blue and is associated with a stabilizer, represented by each blue tile in the grid. The data qubits are arranged in a lattice structure and are represented in gold. Each qubit is connected to adjacent measurement qubits, forming a network that facilitates error detection and correction. The distance-7 correction codes have the capability to correct up to 3 arbitrary errors without confusing them with other errors.

The red lines delineate a subarea that configures one of the nine distance-3 surface codes measured for comparison (arranged in 3×3 arrays). The orange lines delineate the distance-5 code, and the black line indicates the edge, representing the complete distance-7 code. The lower distance codes are used to compare how the code distance affects the logical error rate.

The figure $1b$ represents the cumulative distributions of error probabilities measured on the processor. The curves represent different types of errors in quantum gates and qubits. The red line, labeled $1Q$, represents the distribution of Pauli errors for single-qubit gates. The black line, labeled CZ, represents the distribution of Pauli errors on control-Z gates (which are essential for quantum entanglement between qubits). The blue line, labeled Meas., indicates the average identification error during measurements. The Idle line represents errors for qubits that are inactive during the error correction cycle. The green line, labeled Det., corresponds to the probability of error detection during the correction cycle.

The figure $1c$ represents the logical error probability, p_L, as a function of the number of quantum error correction cycles t for different code distances. The distance-3 codes (2023) show an increase in logical error probability as the number of cycles increases, indicating less error suppression. The distance-5 and distance-7 codes have lower logical error probability as the number of cycles increases. The dotted lines, orange and green, are exponential fits of the logical error probability p_L after averaging the results over the logical bases X_L and Z_L. Each point in the graph corresponds to experimental data obtained from the 105 qubits, and the decoding is performed using neural networks that average over these logical bases.

Figure $1d$ shows the logical error rate per cycle, which decreases with increasing surface code distance d. Therefore, it fits the expected relationship from equation 1. The individual data points represent the uncertainty of each measurement, which is less than 5×10^{-5}. The line fitted to the data points results in a factor $\Lambda = 2.14 \pm 0.02$, demonstrating the effectiveness of the error correction. The inset graph is a simulation that extends up to $d = 11$. Theoretically, the trend is linear, and at greater distances, there is greater capacity to detect and correct quantum errors.

4.1 Experiments with Surface Code

The Google Quantum AI lab has implemented a distance-7 surface code with 105 physical qubits, achieving a logical error rate per cycle of $\varepsilon_7 = 1.43 \times 10^{-3}$. This represents an improvement factor of 2.4. The logical error rate was reduced by a factor of $\Lambda = 2.14 \pm 0.02$ compared to the lower-distance code. Additionally, it was also demonstrated that if the code distance is increased, in this case from 5 to 7, the error suppression follows the mathematical relationship described earlier, ε_d.

4.2 Implementation of Real-Time Decoder

This is also very innovative, as Google has implemented a real-time decoder capable of processing syndrome information with an average latency of 64 microseconds with a distance-5 code. Why is this important? It is important because, in practical applications, a very fast error correction cycle is required.

5 Relationship with Error Threshold Theory in Quantum Computing

The error threshold is a theoretical barrier that defines whether a quantum system can scale without errors accumulating uncontrollably. The results demonstrate that it is possible to operate effectively below this threshold, suggesting that fault-tolerant quantum computing is possible in practice.

Although this is a giant step, there are many research directions and many approaches to small parts of the process of creating a quantum computer. For example, the reduction of correlated errors. If you have reached this point and think the conclusion is to add more physical qubits to achieve it, let me tell you that brute force does not work, and there are many barriers that prevent this vision. Although, from my standpoint, this approach is not always bad and allows us to learn a lot. I'll leave this topic for another article.

Bibliography

Acharya, R., Aghababaie-Beni, L., Aleiner, I., Andersen, T. I., Ansmann, M., Arute, F., Arya, K., Asfaw, A., Astrakhantsev, N., Atalaya, J., Babbush, R., Bacon, D., Ballard, B., Bardin, J. C., Bausch, J., Bengtsson, A., Bilmes, A., Blackwell, S., Boixo, S., … Zobrist, N. (2024). Quantum error correction below the surface code threshold. https://arxiv.org/abs/2408.13687v1