Quantum Computing Fellowship Final Project

Quantum Computer Resource Planner - Emmanuel Omolaja

Technical Guide: How It Works

Quick Summary

This tool gives **rough guesses** for the parts needed to run complex, error-free quantum programs. It uses simple math based on real research.

Accuracy: Good for *planning*, **NOT** for building or buying parts.

1. Simple Ideas

What is a Logical Qubit?

This is the **perfect** quantum bit (or 'qubit') your program uses. It never makes a mistake.

What is a Physical Qubit?

These are the **real, noisy** qubits in the quantum computer. They constantly make errors (about 1 out of every 100 times).

The Challenge

To get just **one reliable Logical Qubit**, you need hundreds or thousands of **Physical Qubits** working together to check and fix errors.

The Fix: Quantum Error Correction (QEC)

QEC is a smart way to store one perfect qubit's information across many noisy qubits, allowing the system to find and correct mistakes without losing the quantum data.

2. Key Numbers You Put In

Logical Qubits

- **What it is:** The number of perfect qubits your program needs.
- **Example:** A code-breaking program needs about 4,000 Logical Qubits.
- **Why it matters:** More perfect qubits means more physical parts are needed.

T Gates

- **What it is:** A specific, very hard type of quantum step (or 'gate').
- **Why it's special:** T Gates are the **most expensive** step in a large quantum computer.
- **The cost:** Each T Gate needs a complex process called "magic state distillation" that uses lots of physical qubits.

- **Counts:** Simple programs use thousands; big programs use **billions**.

Target Error Rate

- **What it is:** How likely you are to accept your whole program failing.
- **Trade-off:** If you want fewer failures, you need more physical qubits to achieve it.
- **Standard:** 0.1% failure rate (1 failure out of 1,000 runs) is common.

Hardware Type

Different machines have different speeds and error levels:

Machine Type	**Step Time**	**Error Level**
Superconducting	Fast (50 ns)	Medium (0.1-1%)
Trapped Ion	Slower (5 μs)	Low (0.01-0.1%)
Neutral Atom	Slower (2 μs)	Medium (0.1-1%)

3. How the Math Works (Simple Steps)

Step 1: Find the Code Strength (Code Distance)

We figure out how strong the error correction needs to be (the **Code Distance**, d). A stronger code (d is higher) fixes more errors but needs more qubits. This is based on your total number of T Gates and your Target Error Rate.

- This is a quick estimate, not a perfect number.

Step 2: Calculate Physical Qubit Count

The total physical qubits is: **Logical Qubits $\times$ Overhead Factor**

The **Overhead Factor** changes based on the error-fixing method (the "Code"):

Fixing Method	**Qubits Needed per Logical Qubit**	**Example (d=15)**
Surface Code	$d \times d \times 2$	450 qubits
Cat Qubits	$d \times 4$	60 qubits

Step 3: Find Runtime (Time Taken)

The total time is based on the number of steps and the speed of the hardware.

- **Warning:** This math **leaves out** the time needed for the T Gate "magic state" creation. In reality, this process can make the total time **10 to 100 times longer** than our estimate.

4. What This Tool Gets Right and Wrong

Right (Good for):

- **Big Picture:** It tells you if you need thousands or millions of qubits.
- **Main Cost:** It correctly shows that T Gates are the biggest problem.
- **Trends:** It shows how using a different fixing method changes the total number of qubits.

Wrong (What it leaves out):

- **The Big Time Cost:** It ignores the massive time needed to make T Gates.
- **Qubit Movement:** It doesn't count the time needed to move qubits around the chip.
- **Real Errors:** It assumes all errors happen one at a time, which is not true for real machines.
- **Classical Computer Time:** It ignores the time the normal computer needs to process the error information.

5. Example Guess

Program: A large search problem.

Inputs: Logical Qubits: 256; T Gates: 500 million; Error Rate: 0.1%.

1. **Code Strength (d):** About 15.
2. **Total Physical Qubits (using Surface Code):** $256 \times 450 = \mathbf{115,200 \text{ qubits}}$.
3. **Estimated Runtime:** About **94 minutes**.

(But remember, the real time is likely much longer due to the missing "magic state" cost!)

6. Final Note

This tool is a **teaching aid** to help understand the huge demands of building an error-free quantum computer.