How Good are my Qubits?

Benchmarking Quantum Hardware

Brian McDermott
Qiskit Fall Fest 2024

Follow along:

https://github.com/bjmcder/rpi-qiskit-fallfest-benchmark-2024/

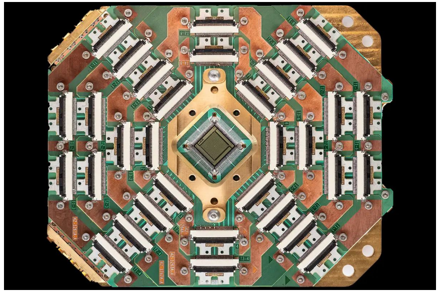
About Me

- Attended RPI 2006-2016
  - Nuclear engineering bachelor’s & PhD
  - Experimental work at LINAC and RCF
  - Greek life, American Nuclear Society
- Currently at NNL/KAPL in Schenectady
  - Future technology group
  - Started quantum work ~2017
  - Projects include quantum computing, sensing and materials
  - Check us out at the career fair!
- Teaching MANE-4960
- Connected with several quantum communities & organizations
  - Industry, academia, government, nonprofit
  - Let me know your interests!

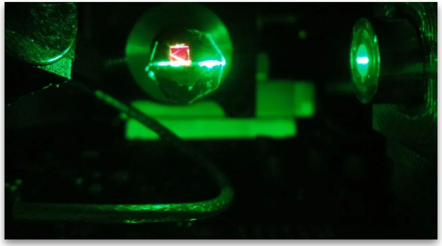
*All opinions expressed are my own and not those of any other organization.

What are Quantum Technologies?

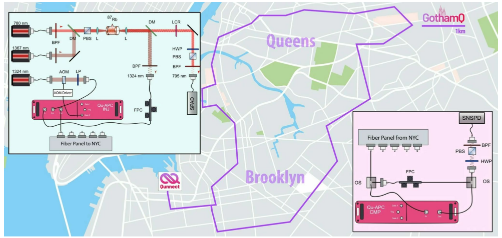
- Any technology that leverages quantum mechanical effects:
  - Superposition
  - Entanglement
  - Tunneling
- We're specifically interested in quantum technologies for measuring, communicating, and processing information.
  - Sensors
  - Networks
  - Computers



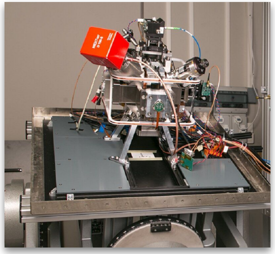
D-Wave Quantum Annealer



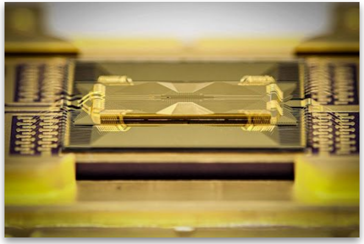
Quantum Diamond Magnetometer (SBQuantum)



Qunnect NYC Quantum Network



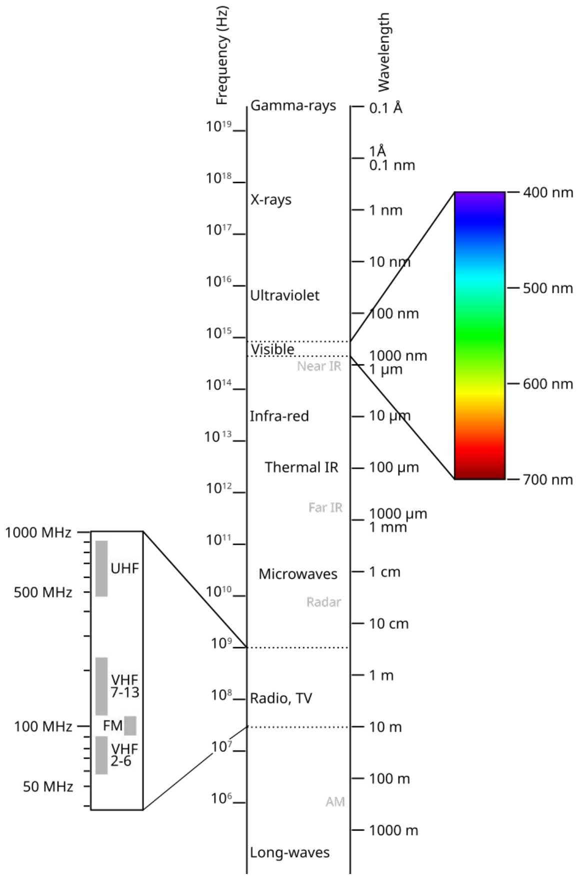
Quantum Navigation System (CNRS)



Ion Trap (IonQ)

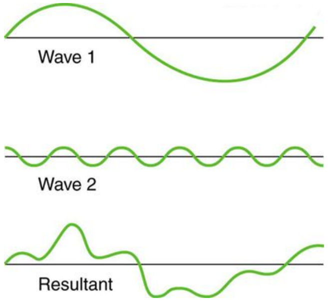
What are they made of?

- Any physical multilevel system:
  - Energy levels in atoms & ions
  - Oscillation modes in superconducting circuits
  - Spins of particles & crystal defects
  - Polarization of photons
- Can be naturally-occuring or engineered
  - Atoms, ions, nuclei, photons
  - CMOS structures, quantum dots, graphene
- Governed by quantum electrodynamics
  - Essentially all manifestations of electromagnetism
  - Microwave to UV ranges

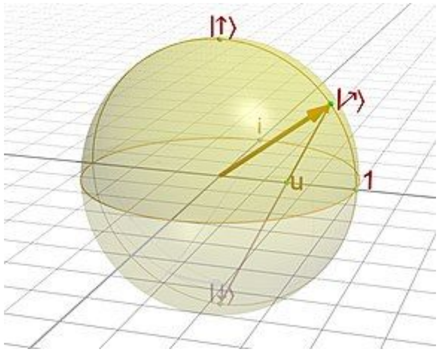


Why Quantum?

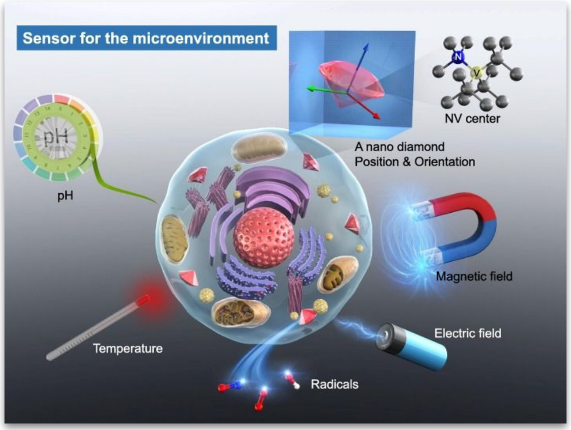
- Compact representations:
  - Superposition lets us combine information compactly.
  - Information is represented as an interacting set of probability amplitudes.
- Large state space:
  - Entanglement lets us grow the space of information we can work with.
  - Amplify correlated probabilities.
  - Exponential amounts of information in a linear number of elements (qubits)
- Useful physical properties:
  - Small size, weight, power consumption, or cost per performance (SWaP-C)
  - Sensitive coupling to environmental parameters (sensors)
  - Immune to copying or eavesdropping (networks)



Superposition of waves



State space of a qubit

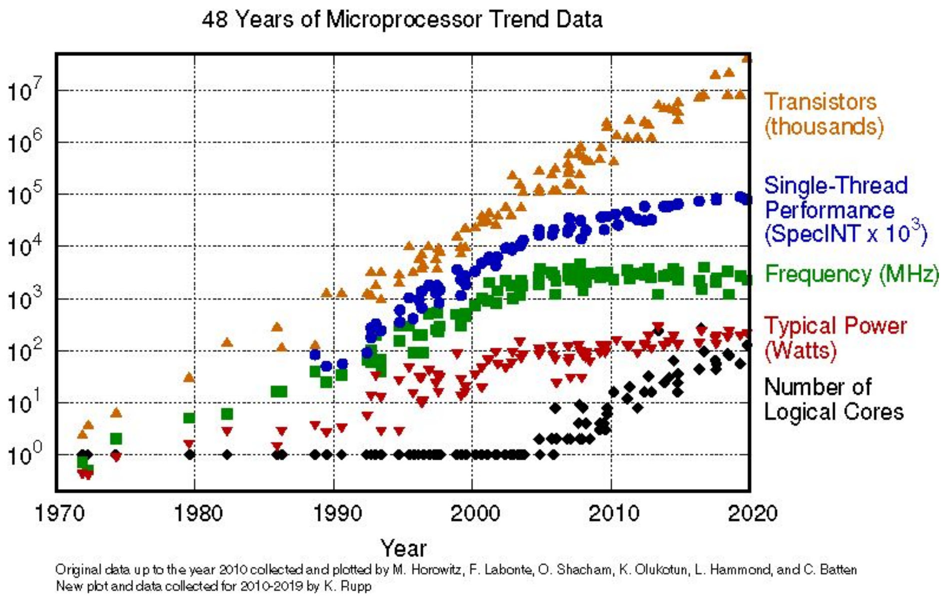


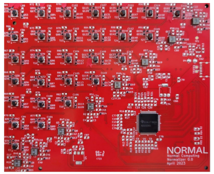
Nanoscale diamond quantum sensors

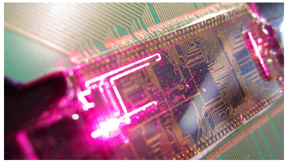
Segawa, Takuya F., and Ryuji Igarashi. "Nanoscale quantum sensing with Nitrogen-Vacancy centers in nanodiamonds–A magnetic resonance perspective." *Progress in Nuclear Magnetic Resonance Spectroscopy* 134 (2023): 20-38.

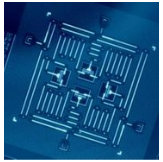
Why Quantum?

- Computing is on a march towards heterogeneity.
  - CPU performance gains are nearing limits
  - Moore’s law not dead, it just went elsewhere
  - The free ride is over
- Specialized accelerators are needed for future performance gains.
  - GPU
  - Quantum
  - Neuromorphic
  - Thermodynamic
  - …
- Quantum is a specific kind of accelerator for “Physics-flavored” problems.



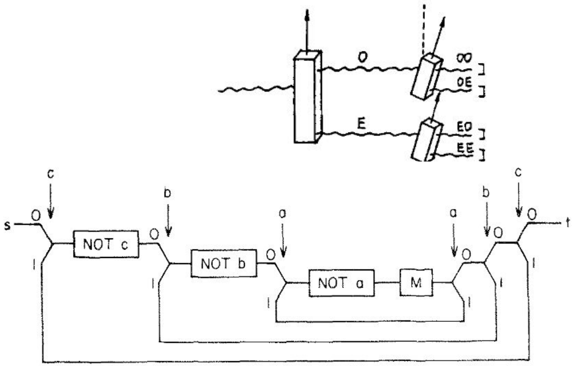




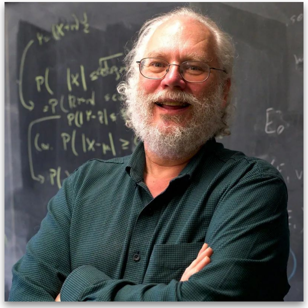


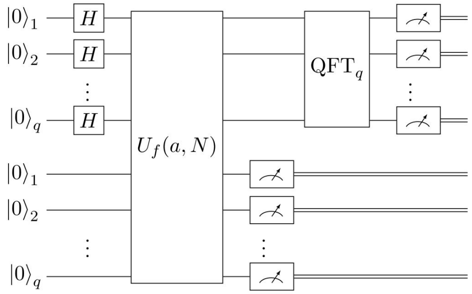
Why now?

- Quantum mechanics has been around a long time.
  - Original theory ~100 years
  - Quantum computing ~40 years
- Shor's Algorithm was the spark that lit the explosion.
  - Proves an efficient way to factor large integers on a quantum computer
  - Critical cybersecurity implications; public-key encryption depends on the presumed difficulty of factoring.
- Initial theories worked out in 90's
  - Error correction & fault tolerance theorems
  - Qubit modalities
- Engineering breakthroughs from the '00s and '10s are yielding rapid growth today.
  - Nanofabrication
  - Photonics
  - Precision Electronics
  - Cloud computing

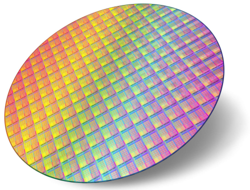


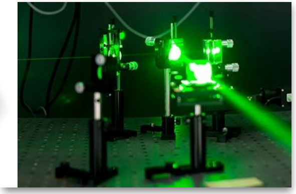
Feynman's original quantum computing concepts [1,2]





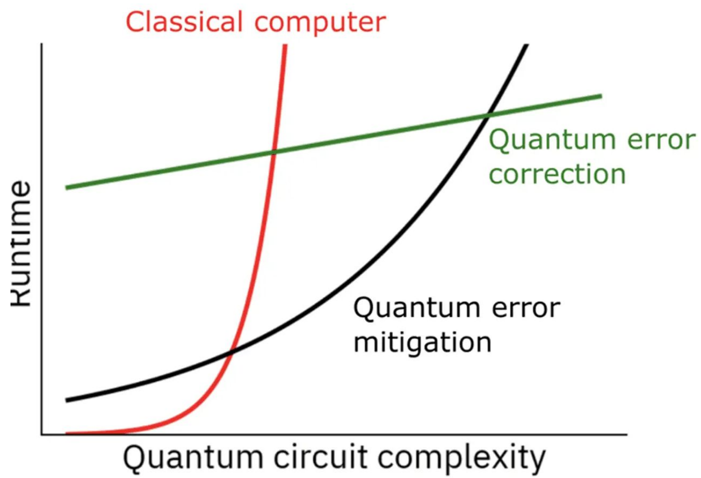
Peter Shor's factoring algorithm [3]

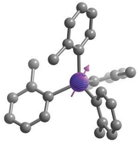


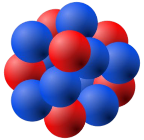


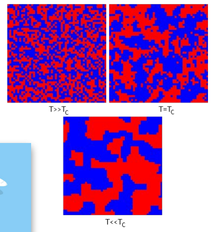
What can it do?

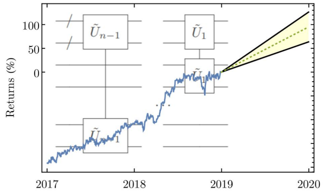
- Quantum computing can improve the <u>scaling behavior</u> of algorithms.
  - Run bigger problems for a given amount of compute resource
  - Run problems that are too big to be tractable.
- Quantum computing is best at specific problems:
  - Strong correlations
  - Probabilistic behaviors
- There are many problems in science, engineering and business that fit this profile.







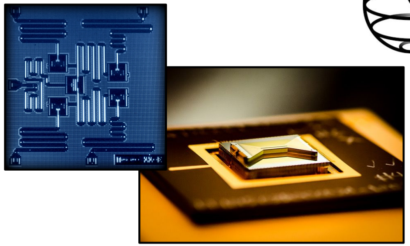


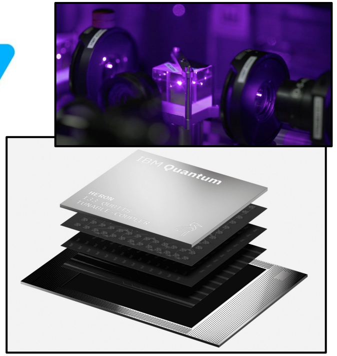


$$E\Psi = \mathbf{H}\Psi$$

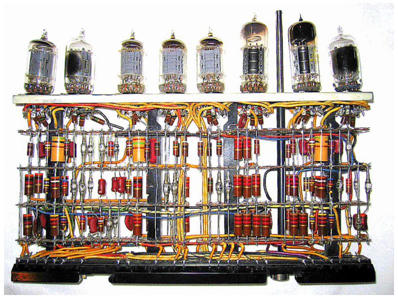
The State of Quantum Computing

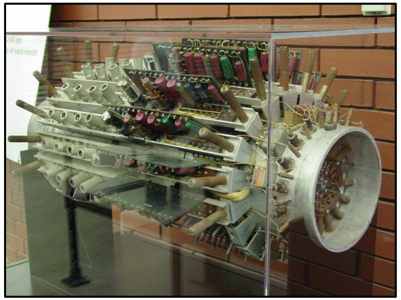
- Quantum computing has made remarkable progress:
  - Useful for research, education and technology tracking.
  - Starting to be useful for Physics research beyond “how to make a quantum computer”
- Still in the “vacuum tube era”
  - No error correction
  - Machines are too small for most industrial problems.





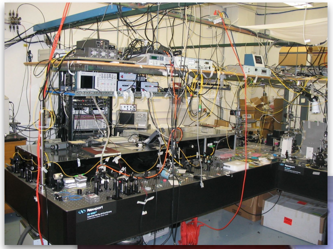
2016 → 2024

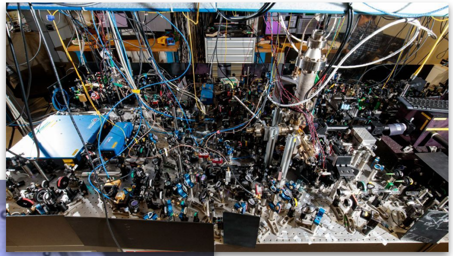


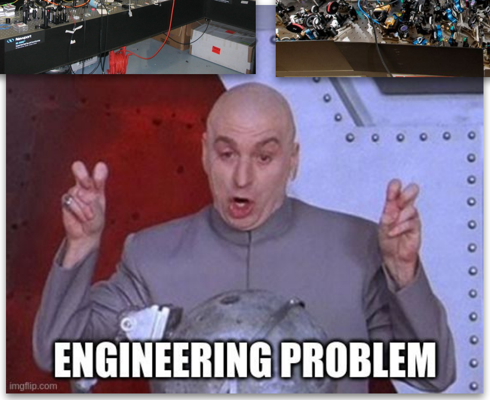


Why Benchmarking?

- Quantum computers need to make the transition from physics experiment to useful appliance.
    - How far are we from that?
    - How will we know when we’re there?
- We want to:
    - …see how well the qubits are performing.
    - …understand sources of error.
    - …track how good the technology is at problems we care about solving.

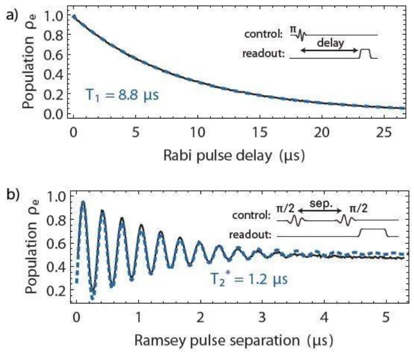


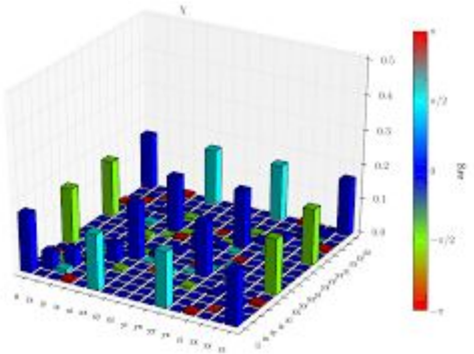


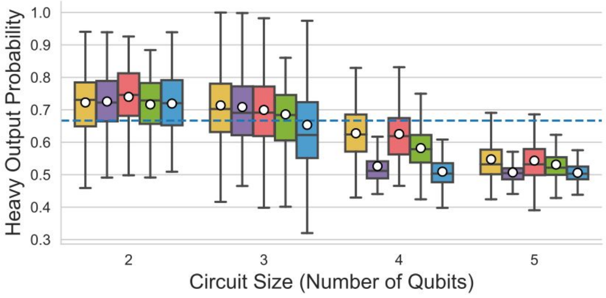


What do we benchmark?

- Qubits
  - Coherence times
  - Operation times
- Operations
  - State preparation and measurement
  - Process fidelity
- Applications
  - Building blocks and Subroutines
  - Mini-apps

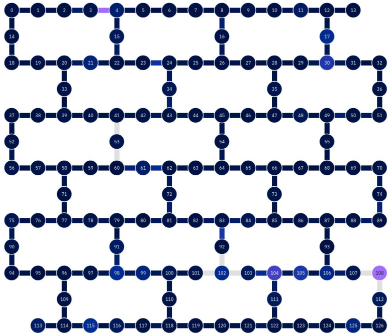




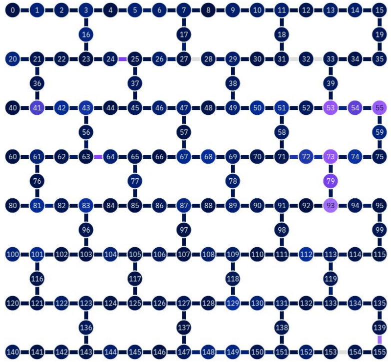


Our Machine - `ibm_rensselaer`

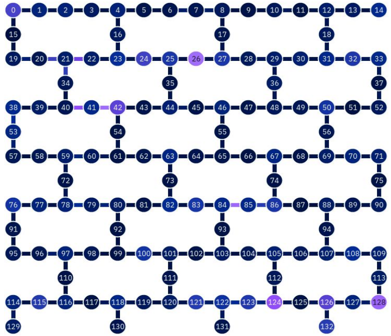
- IBM System One
  - 127-qubit Eagle Architecture
  - “Heavy-hex” connectivity
- Calibrated Every ~6 hours
  - Qubit characteristics
  - Process fidelities
- View latest calibration data:
  - https://quantum.ibm.com/services/resources



IBM Rensselaer (Eagle)



IBM Fez (Heron v2)



IBM Torino (Heron v1)

Calibration Dashboard

ibm_rensselaer
OpenQASM 3

Details

Qubits	2Q Error (best)	2Q Error (layered)	CLOPS
127	3.49e-3	2.10e-2	32K
Status	**Region**	**Processor type** ⓘ:	**Version**
Online	us-east	Eagle r3	1.1.127
Total pending workloads	**Basis gates**	**Your instance usage**	**Median ECR error**
0 jobs	ECR, ID, RZ, SX, X	1120 jobs	8.520e-3
Median SX error	**Median readout error**	**Median T1**	**Median T2**
2.441e-4	8.900e-3	275.1 us	162.96 us

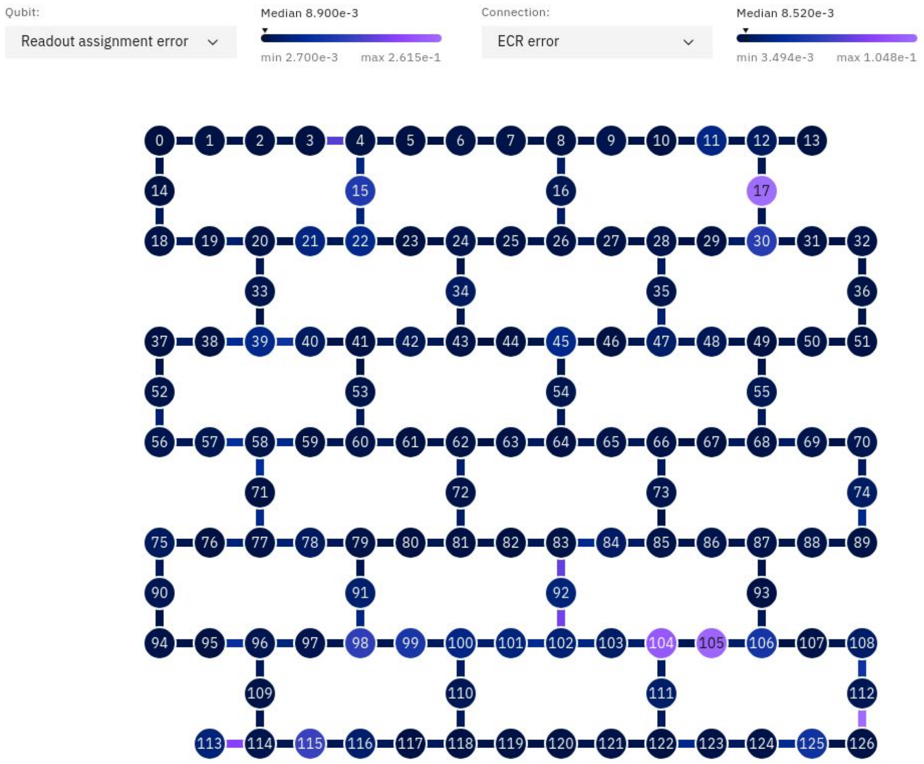
Calibration Dashboard

ibm_rensselaer OpenQASM 3

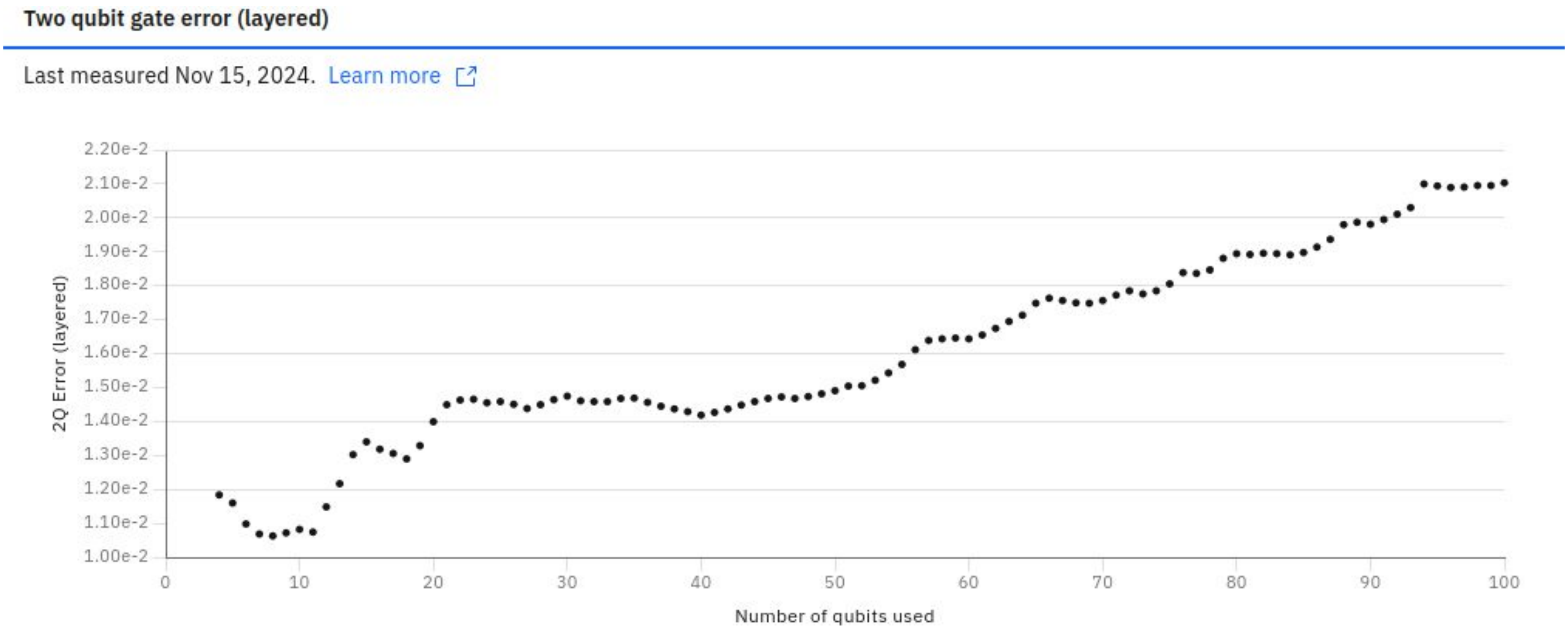
Details

Qubits 127	2Q Error (best) 3.49e-3	2Q Error (layered) 2.10e-2	CLOPS 32K
Status Online	Region us-east	Processor type ⓘ: Eagle r3	Version 1.1.127
Total pending workloads 0 jobs	Basis gates ECR, ID, RZ, SX, X	Your instance usage 1120 jobs	Median ECR error 8.520e-3
Median SX error 2.441e-4	Median readout error 8.900e-3	Median T1 275.1 us	Median T2 162.96 us

Calibration Dashboard



Calibration Dashboard



Where do these numbers come from?

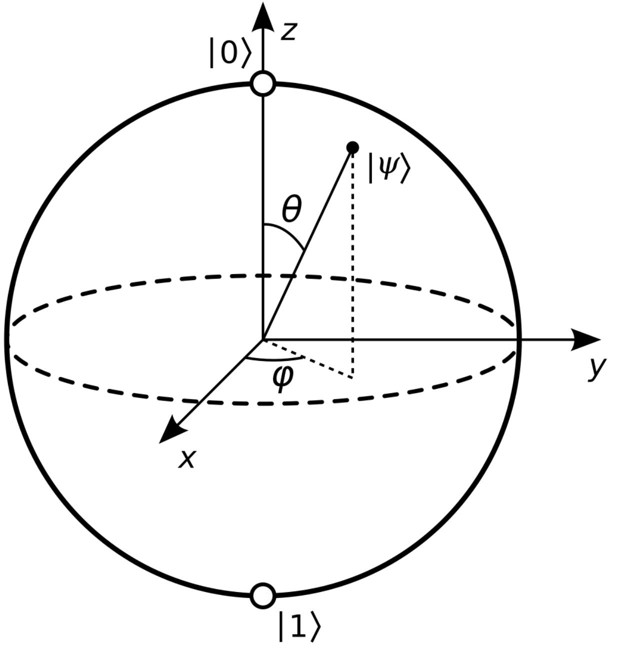
- Every few hours, an automated calibration script runs on the hardware.
  - Queue is paused
  - Takes 20-30 minutes
- You can run these tests yourself → Qiskit Experiments
  - Documentation: https://qiskit-community.github.io/qiskit-experiments/
  - Github: https://github.com/qiskit-community/qiskit-experiments

Some Theory

(hopefully not too much!)

Pure States

- Most discussions of qubits talk in terms of *statevectors*.
- A statevector represents what is known as a *pure state*
- Each element tells of how much of each basis state we have in the superposition.
- **What happens if qubits aren't perfect?**



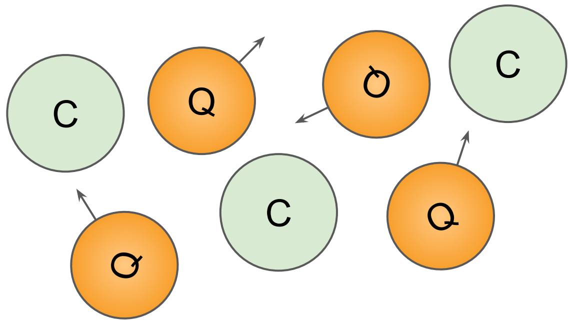
The Bloch sphere represents a pure state

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

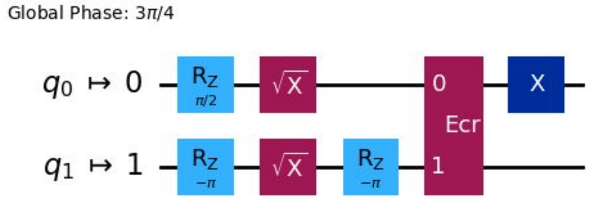
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad |\alpha^2 + \beta^2| = 1$$

Density Matrices

- Consider the case where we have a bunch of qubits in the same system…
  - Clouds of atoms
  - Liquid NMR
  - NV Centers
- …or a single state prepared and measured over and over.
  - Quantum circuits and algorithms
  - Quantum communication protocols
- The density matrix gives us a unified representation of quantum coherence and classical mixtures.



A mixture of quantum and classical states



Running a circuit over and over

Density Matrices

- The outer product of two statevectors gives us a matrix.
  - Projectors, if we're talking about the basis states.
- In general, the outer product of two quantum statevectors gives us a *density matrix*.
- Density matrices represent a statistical mixture of quantum *and* classical states.

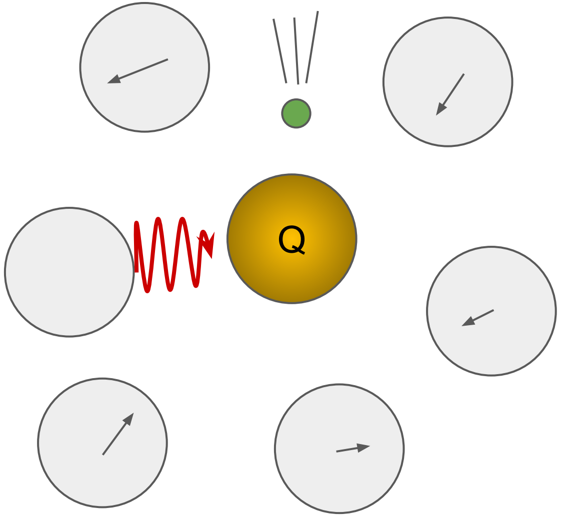
$$P_0 = |0\rangle\langle 0| \quad P_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$P_1 = |1\rangle\langle 1| \quad P_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\rho = \begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{11} & \rho_{11} \end{pmatrix}$$

Physical Decoherence

- Schrodinger's equation assumes a quantum system that is perfectly isolated.
    - Real systems lose energy to the environment
    - Formally, the environment is called a "bath"
- Open quantum systems are governed by the *Lindblad Master Equation*.
    - The real world is non-unitary.
    - Quantum term plus a dissipative term.
    - In the limit of zero dissipation, the Schrodinger equation is recovered.
- Lindblad simulations are used in qubit and quantum device design.

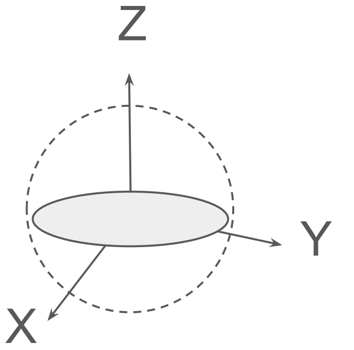


$$\frac{d\rho}{dt} = \underbrace{\boxed{-i[H,\rho]}}_{\text{Coherent Evolution}} + \underbrace{\boxed{\sum_k \left(L_k \rho L_k^\dagger - \frac{1}{2}\left\{ L_k^\dagger L_k, \rho \right\} \right)}}_{\text{Decoherence}}$$

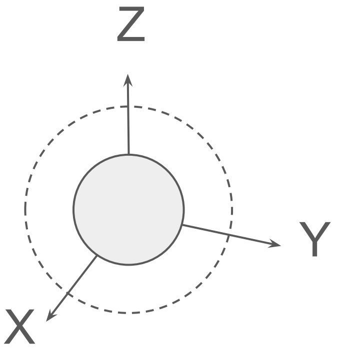
QuTiP - Quantum Toolbox in Python

Decoherence of Qubits

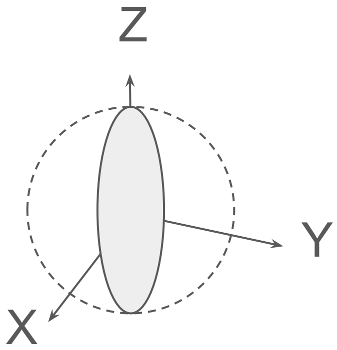
- The quantum state in a qubit (or set of qubits) is very fragile
    - Thermal vibration
    - Stray E/M fields
    - Radiation
- Decoherence is the process that returns quantum states back to classical ones.
    - Measurement is a “controlled” decoherence process.
    - Noise is measurement done by the environment.
- There are several “channels” for decoherence.
    - Amplitude damping
    - Phase damping
    - Depolarization (combination of the first 2)
- Decoherence channels “squish” the reachable parts of the Bloch sphere



Amplitude Damping



Depolarization



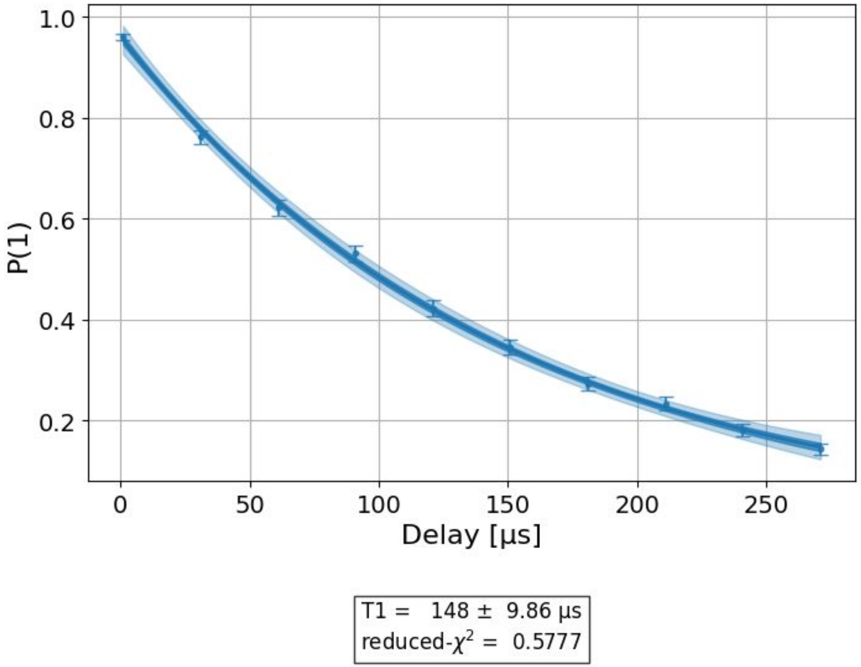
Dephasing

Noise & Performance Benchmarks

- There are so many noise sources, a zoo of benchmarking techniques has been developed to try to characterize them.
- There's a hierarchy of benchmarks:
  - Qubit and gate-level
  - Circuit-level
  - Application-level
- Not all benchmarks are created equal.
  - Physical timings and fidelity of operations are relatively objective.
  - Integrated measures at the circuit and gate level can vary widely.
  - Application-oriented benchmarks
- Pay attention to which benchmarks are preferred by different hardware vendors!

Amplitude Damping

- A qubit placed in an excited state only stays there temporarily before losing energy to the environment.
  - De-excitation
  - Spin relaxation
- This decoherence process is called *amplitude damping*.
- The “decay constant” that represents its de-excitation back to the 0 state is denoted $\mathbf{T_1}$.



For superconducting qubits, T_1 values of ~150 μs are typical

Amplitude Damping

- We can measure T_1 using individual qubits, or many in parallel.
- Experiment Algorithm:
  - Start with a qubit in the 0 state.
  - Apply an X gate to put it in the 1 state.
  - Wait for a predefined delay time, t_d.
  - Measure the qubit.
  - Tally the expectation value with the (I-Z)/2 operator.
  - Repeat for N_d delay times.
- Fitting an exponential function to the data points gives us T_1.

Phase Damping

- When we place a qubit in superposition, it has a well-defined phase.
  - Consider the + and - Hadamard states.
- As it undergoes interactions with the environment, the phase drifts.
- Coherent phase information leaks into the classical elements of the density matrix.
- The time constant for this phase coherence loss is called $\mathbf{T_2}$ or $\mathbf{T_2}^*$, depending on how it is measured.

Phase Damping

- T_2 is measured using a *Hahn Echo* sequence
  - Apply a Hadamard gate
  - Wait for a delay time
  - Apply an X gate
  - Wait another delay time
  - Apply a Hadamard
  - Measure
- During the delay the qubit precesses around the Z-axis due to its natural frequency.
  - Applying an X gate flips it 180 degrees.
  - Applying the last Hadamard should bring us back to 0, if the coherent phase is well defined.

Phase Damping

- T_2^* is an alternate measurement that can also provide an estimate of the qubit's intrinsic frequency.
- Measurement protocol:
  - Apply a Hadamard gate
  - Wait a delay time
  - Apply an R_z rotation around the Z-axis.
  - Apply another Hadamard gate
  - Measure
- If the R_z gate and the natural precession offset each other perfectly, then the qubit will return to zero.
- Offsets from zero occur as the phase coherence decays.

Demo

Multi-Qubit Fidelity

- Single-qubit coherence times and fidelity measures are only part of the story.
- Multi-qubit operations are also a big (if not bigger) source of error.
- We need a way to measure errors that occur when we entangle qubits.
- We need a way to quantify how errors compound to determine the maximum size of circuits we can run.

Clifford Gates

- A Clifford gate is any gate that can be constructed with a combination of:
  - Hadamard (H)
  - Phase (S)
  - CNOT (CX)
- Clifford gates + T-gates are universal.
  - You can make any other gates from these.
  - T-gates are needed to get irrational rotation values.
  - Only Clifford + T-gates lead to quantum advantages.

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

William Kingdon Clifford

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

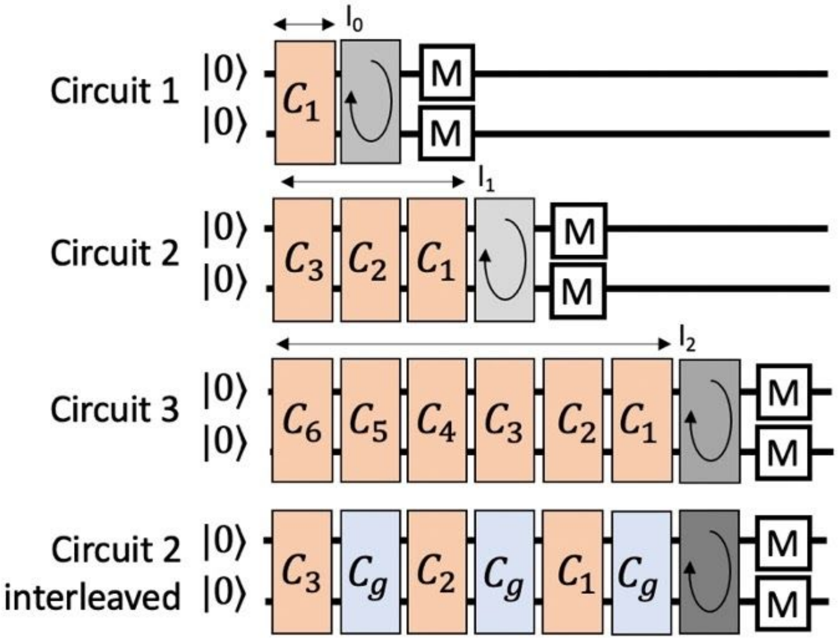
$$\boxed{T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}}$$

Not a Clifford gate!

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Randomized Benchmarking

- A common fidelity test is to run a random string of Clifford gates, then run its inverse. **Randomized Benchmarking (RB)**
- In a perfect machine, this will give an identity.
- The deviation from the starting state is the loss of fidelity.
- Longer strings (e.g. deeper circuits) will result in larger loss of fidelity.

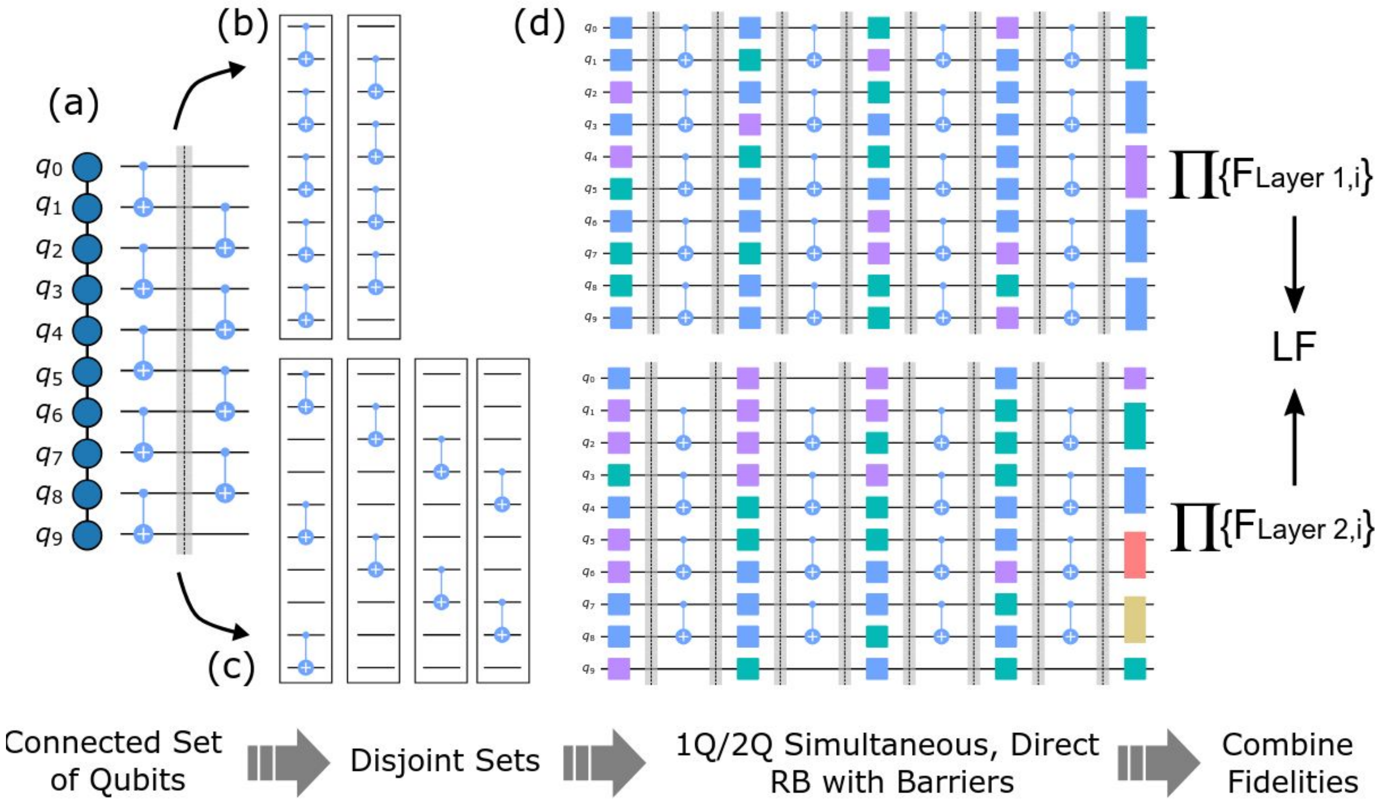


Layer Fidelity Benchmarking

- Randomized benchmarking estimates the per-gate error rates on single and small groups of isolated qubits.
- On real devices, there are also global errors and *crosstalk* between qubits.
- *Layer fidelity* benchmarking attempts to extend RB to capture these other effects.

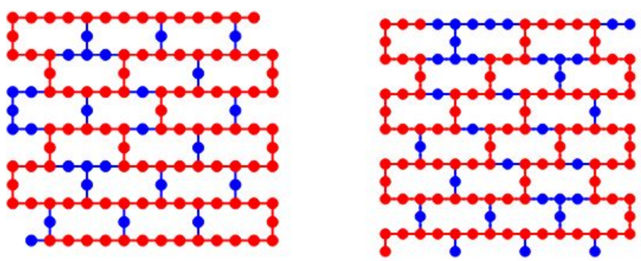
Layer Fidelity Benchmarking

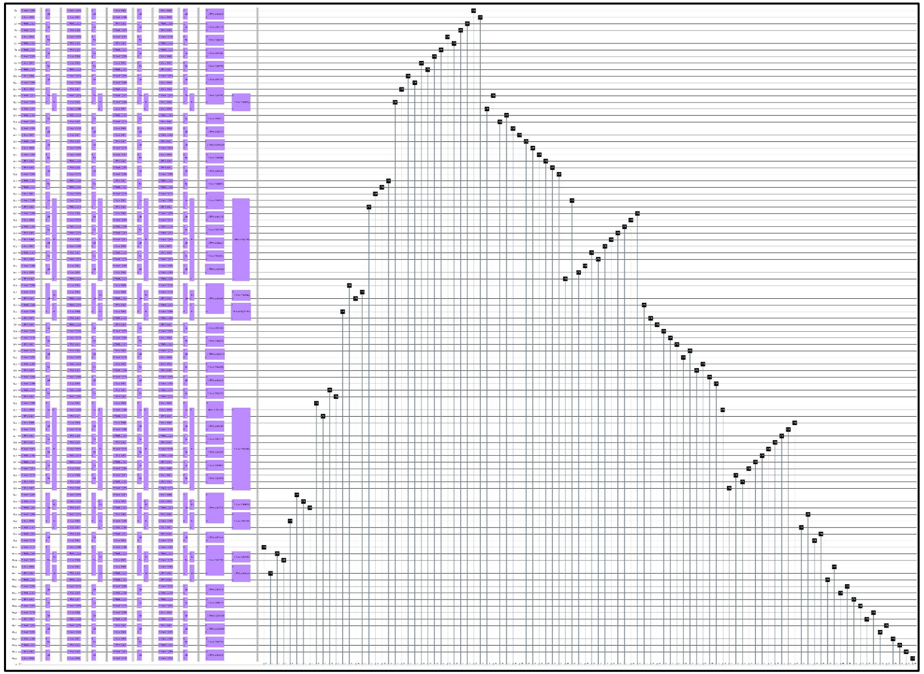
D. C. McKay, et. al., *Benchmarking Quantum Processor Performance at Scale*, arXiv:2311.05933.



Layer Fidelity

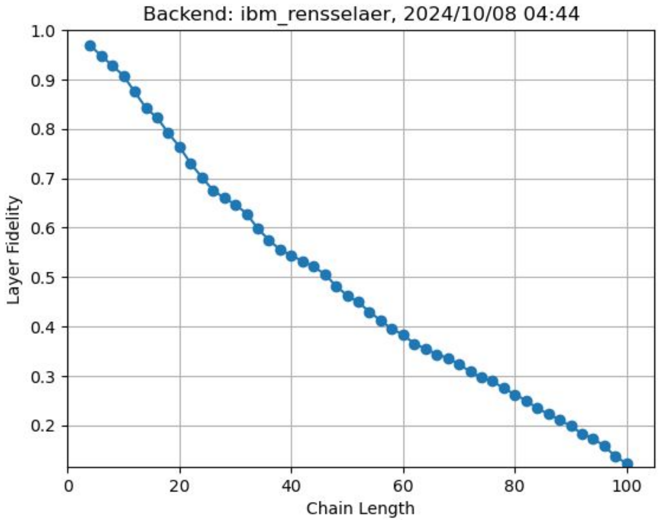
- Grab a long chain of qubits
  - Thousands of possibilities
  - Estimate the best chain
- Split circuit into disjoint sets
  - Need to apply gates onto qubits in a non-overlapping way
- Generate randomized benchmark circuits.
- Plot the fidelity as number of layers increases.
- “Error Per Layered Gates” (ELPG)

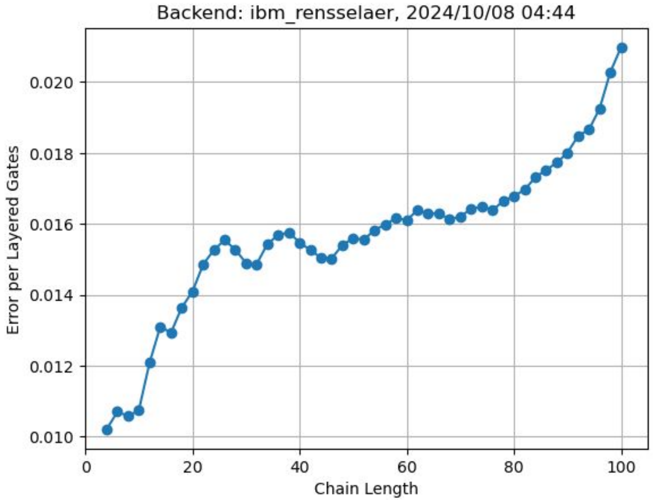




Layer Fidelity

- Error accumulation is a function of both qubit count and circuit depth.
- Layer fidelity gives an estimate of how much error we can expect to accumulate as we stack up operations.
  - Layer fidelity of 0.8 → 80% probability that a layer executed with no errors.
- ELPG estimates the errors normalized per gate in the layer.





Demo

What do I do with this information?

- Qubit-level benchmarks give you a raw measure of the hardware quality.
  - Used by hardware developers to improve designs and manufacturing methods.
  - Used to determine the fault-tolerance threshold.
- Process-level benchmarks give you an estimate of how much work you can do.
- Both benchmarks let you develop *noise models*.

Noise Models

- Quantum simulators can model the noise behavior of quantum computers.
  - Useful when you want to check if something is even worth running.
  - Essential when you don't have hardware access.
- Most quantum computing research is done with noise models.
- "All models are wrong, some are useful!"

Noise Models

- A noisy simulator models the evolution of the density matrix with each gate application.
- The effects of decoherence are applied using *Kraus operators*.
- Qiskit Aer lets you define your own Kraus operators, or pick from a library of common error channels.
- Once we have T_1 and T_2, we can define Kraus operators for these processes.

$$\rho' = \sum_k \boxed{E_k} \rho E_k^\dagger$$

The new density matrix after the application of *k* different noise channels with Kraus operators

$$\gamma = 1 - e^{t/T_1}$$
$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}$$
$$E_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}$$

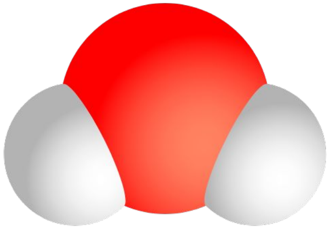
Amplitude Damping

$$\lambda = 1 - e^{t/T_2}$$
$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}$$
$$E_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}$$

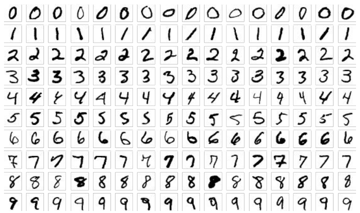
Phase Damping

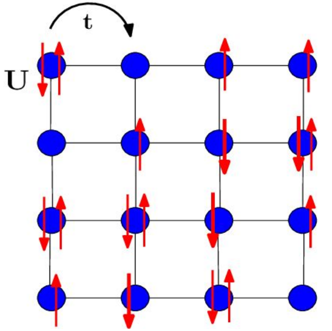
Using T_1 and T_2, we define *damping parameters* that we use to make the Kraus operators

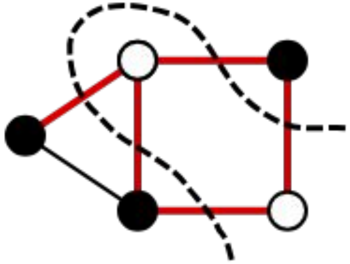
Benchmarking Real(-ish) Problems

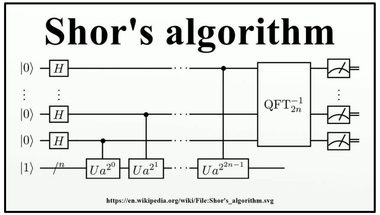


- *Application-oriented* benchmarks replicate an algorithm (or part of an algorithm) in a full application.
- These give an integrated view of how quantum hardware performs on actual problems people care about.
    - Simulations
    - Machine Learning
    - Optimization
    - Crypto & Shor's Algorithm
- They have a known answer and performance profile.
    - Analytically solvable, or experimentally known answers.
    - Solved many times with different classical methods



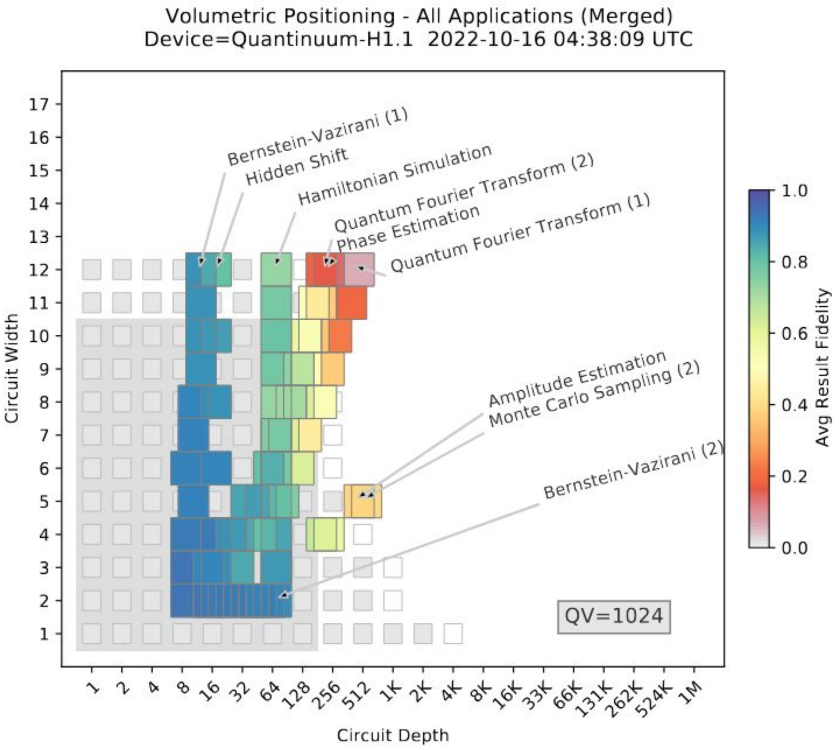






QED-C Benchmark Suite

- The Quantum Economic Development Consortium (QED-C) publishes a set of benchmark problems.
- ~16 application areas and growing
  - QFT, Grover, Shor
  - Simulation
  - Optimization, QML
- Some need updating (Qiskit changes)
- Open source: https://github.com/SRI-International/QC-App-Oriented-Benchmarks.git



Making sense of benchmarks

- Pay careful attention to the baseline of comparison!
    - Equivalent classical brute-force methods
    - Best-available classical methods
    - Industry-standard classical methods
    - “Good-enough” classical approximations
- Watch out for “one number” benchmarks.
    - e.g. Quantum Volume, Algorithmic Qubits
    - These can be misleading and potentially gamed.
    - Usually show preference for a particular vendor or hardware type.

Application-Oriented Benchmarking

- Helpful for bottom-line decisions.
  - "Can a quantum computer run my problem?"
  - When do I need to plan to buy one?
- Many more are possible.
  - Find a well-known problem with a well-known answer.
  - Write a quantum version of it.
  - Compare it rigorously to classical methods.
- My opinion: This is a good near-term research area!
  - End-users don't know (or even care) about coherence times and gate fidelities.
  - People want to know if/when quantum computing will have an advantage for their problem.

Open Source and Research Opportunities

- QED-C benchmarks
  - Keep up to date with Qiskit and other quantum SDKs
  - Code refactoring and maintenance
  - New benchmarks
- Qiskit Benchpress
  - Benchmarks Qiskit itself and some quantum algorithms
  - Improve timing and accuracy performance
- New benchmarks
  - Process-level benchmarks
  - Error-mitigation strategies
  - New application-oriented benchmarks

Resources

- Qiskit Experiments: https://qiskit-community.github.io/qiskit-experiments/
- Qiskit Benchpress: https://github.com/Qiskit/benchpress
- QED-C Benchmarks: https://github.com/SRI-International/QC-App-Oriented-Benchmarks
- Quantum Algorithm Implementations for Beginners
  - Paper: https://dl.acm.org/doi/10.1145/3517340
  - Github: https://github.com/lanl/quantum_algorithms

Thank You!