INTRODUCTION

What is Benchmarking?

Benchmarking is how the performance & capabilities of a computing system is determined. One must choose the appropriate benchmark and metrics to extract meaningful results. Different benchmarks test the system in different ways and each individual metric may or may not be of interest. Some vital characteristics for Benchmarks:

- **Relevance:** Benchmarks should measure important features.
- **Representativeness**: Benchmark performance metrices should be broadly accepted by industry and academia.
- **Equity:** All systems should be fairly compared.
- **Repeatability:** Benchmark results should be verifiable.
- **Cost-effectiveness:** Benchmark tests should be economical.
- **Scalability:** Benchmark tests should measure from single server to multiple servers.
- **Transparency:** Benchmark metrices should be readily understandable.

Performance Metrices in Classical Computing:

- **Raw Performance of Individual Hardware Components**
- **Time to Solution-** Getting the job done as soon as possible
- **Throughput-** Number of jobs that the computer can satisfy simultaneously
- **Instruction Set Architecture-** Compatibility

What are we trying to Benchmark in Quantum Computers?

Benchmarking in quantum computing is more complicated than in other fields, where there is a wider range of hardware (from laser-controlled arrays of ions to microwave-controlled spinning electrons), fewer established guidelines, and additional complicating factors. The capacity of a quantum computer to perform meaningful computations is not just determined by the number of qubits, but by the quality of those qubits. Harnessing the power of quantum computing requires overcoming errors and other imperfections that are inevitable in all quantum computing hardware. Quantum Computer components fail in many different ways:

- Unintended bit-flips etc
- Systematic errors that add up coherently and unpredictably
- Drift in performance over (Operations drifting out of calibration)
- Complex & poorly understood “crosstalk” and integration failures in larger devices

In the near and medium-term, benchmarking quantum computers is all about quantifying all sources of error and understanding how this errors impact applications

Complexity of Error Problem

- Complexity of error model scales faster than the exponential gain of the quantum computer

For an n-qubit quantum computer:

- Quantum computing power **grows exponentially**: 2^n
- Quantum error modes **grow even faster***: 2^{4n}

*This is assuming only Markovian errors on the time-scale of gates, otherwise the complexity is even worse

Performance Metrices in Quantum Computing:

Measurements Protocols- Some Single Qubit Metrices

- **Preparation fidelity:** ability to prepare a qubit in a reference state
- **Measurement fidelity:** ability to distinguish $|0\rangle$ and in $|1\rangle$ measurement.
- **Coherence time T1:** Usually time for $|1\rangle$ to decay into $|0\rangle$
- **Coherence time T2:** Time for superposition $|0\rangle + |1\rangle$ to lose phase coherence.
- **Logic gate fidelities.**
- **Logic gate times.**
- **Number of logic gates / decoherence time.**
- **How many operations you can do before things decohere.**

Measurements Protocols- Some Few-Qubit Metrices

- **Average gate error (for each qubits)**
- **Connectivity (which qubits connect to which qubits)**
- **Crosstalk (unwanted interaction between qubits)**
- **Time dependence AKA “non-Markovianity” (gates get worse later in calculation)**
  - Drift of starting conditions, such as temperature, power of losers, microwave pulse control parameters
- **Measurement / feed-forward time**
  - Relevant when the measurement of a qubit is used to decide what to do next ; an essential part of error correction
- **Circuit depth (Number of logic gates in series)**
  - Limited by coherence times, time for a gate to function, temperature change, etc.
  - Very demanding metric

Measurement Protocols- Some System-Level Metrices

- **Number of qubits**
  - Popular but necessarily meaningful
- **Quantum Volume**
- **Exemplar algorithms**
  - Shor’s, Grover’s, etc.
- **Time to solve**
- **Fidelity / quality of solution**
- **Comparison to “best” Classical computer**
  - Quantum Supremacy

The Current State of Benchmarking Quantum Computers

Systems benchmark ⟷ Application benchmark

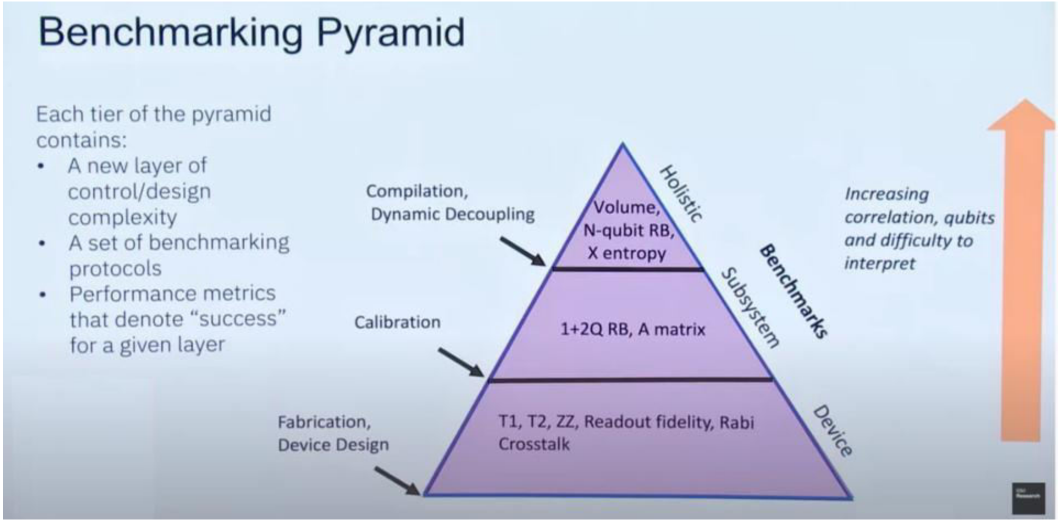
Origin	IBM		Sandia National Laboratories	UC Berkeley / Berkeley Lab	Atos	QED-C	Super.tech
Benchmark	Quantum Volume	Circuit Layer Ops / Second (CLOPS)	Mirror Circuits	Quantum LINPACK	Q-Score	App-Oriented Suite	SupermarQ Suite
Basis	Maximum size of square quantum circuits that can be implemented	Speed in executing layers of a parameterized model circuit	Executing quantum circuits forward and backward	Performance in a prerequisite task for linear algebra	Performance in solving a standard optimization problem	Executing common quantum algorithms / programs	Executing common quantum algorithms / programs plus error correction
Pros	✓ Inclusive measure of performance ✓ Practical measure of noise ✓ Cannot be "gamed" with classical improvements	✓ Evaluates speed of whole-machine operations ✓ Covers quantum-classical latency	✓ Captures significant error sources outside of gate error rates ✓ The forward-backward "mirror" execution makes the benchmarks easily verifiable	✓ Predicts efficacy in scientific computing applications	✓ Predicts efficacy in optimization applications	✓ Targeted toward practical applications ✓ Evaluates whole-machine operations	✓ Targeted toward practical applications ✓ Evaluates whole-machine operations ✓ Scalable for post-supremacy testing
Cons	× Non-square circuits can be predicted only directionally × Applies only to near-term quantum computers	× May not measure speed in all applications × No distinction between classical and quantum improvements	× Indirect measure of application performance	× Less useful for comparing computers that "pass" the test × Restricted to linear algebra	× Restricted to optimization	× Applies only to near-term quantum computers × No distinction between open and closed systems × No distinction between classical and quantum improvements	× No distinction between classical and quantum improvements × No distinction between open and closed systems

Source: IBM; Sandia National Laboratories; UC Berkeley; American Physical Society; Atos; Quantum Economic Development Consortium; Super.tech; BCG analysis.

Type of Benchmarks

System Benchmark: Hardware Performance with such matrices

- **Scale (number of qubits):** Amount of information that can be encoded in quantum system
- **Quality:** Quality of circuits implemented in hardware with low operation errors
- **Speed:** The number of circuits that can run on hardware in given time



Application Benchmark: How well a specific quantum device work at specific task

- **LINPACK-** perform a key prerequisite computational task for linear algebra
- **Q-Score-** performance on a full optimization problem by assessing the maximum number of variables a quantum processor can optimize
- **SupermarQ-** applies techniques from classical benchmarking methodologies to the quantum domain.

We have a lot of low-level benchmarks for charactering few-qubit devices, or a few qubits at a time in a large device (This is how standard data is obtained)

- **Quantum State Tomography**
- **Randomized Benchmarking**

But most current methods do not scale too many qubits! There is a bunch of recent methods that have better scalability

- **Quantum Volume**
- **Cross Entropy Benchmarking**

There is some application-oriented benchmarks that test the performance of quantum computers on practically relevant tasks

- **Algorithms Based Benchmarking**