



SupermarQ: A Scalable Quantum Benchmark Suite



Background

- Till now the benchmarks reigns in compiler and micro-architectures, **shrugs off application part**. On contrary the application-level-benchmarks **out-performs** over hardware-level-benchmark regimes.
- In the era of NISQ devices, the **lack** of experience over originally envisioned QC limits the researchers to flourish in the quantum computing advantages. [gate errors >> number of qubits]
- Such SOTA Benchmarks demonstrate the ability to bridge the forbidden zone of the today Vs tomorrow machine by comparison metric.

Challenges in designing a benchmark for QPU

- Gate Level Measurement :: does not represent performance on applications.
- Circuit Based Benchmarks :: useful for providing insights into the theoretical computational power of a device , are also limited in their scalability and real QC experience.
- Unintended Consequences :: capturing the general performance of a computational system within a single number can be very challenging.
- Application-Level-Benchmarks :: make cross-platform comparisons between different quantum architectures and classical approaches more straightforward.

Motivation

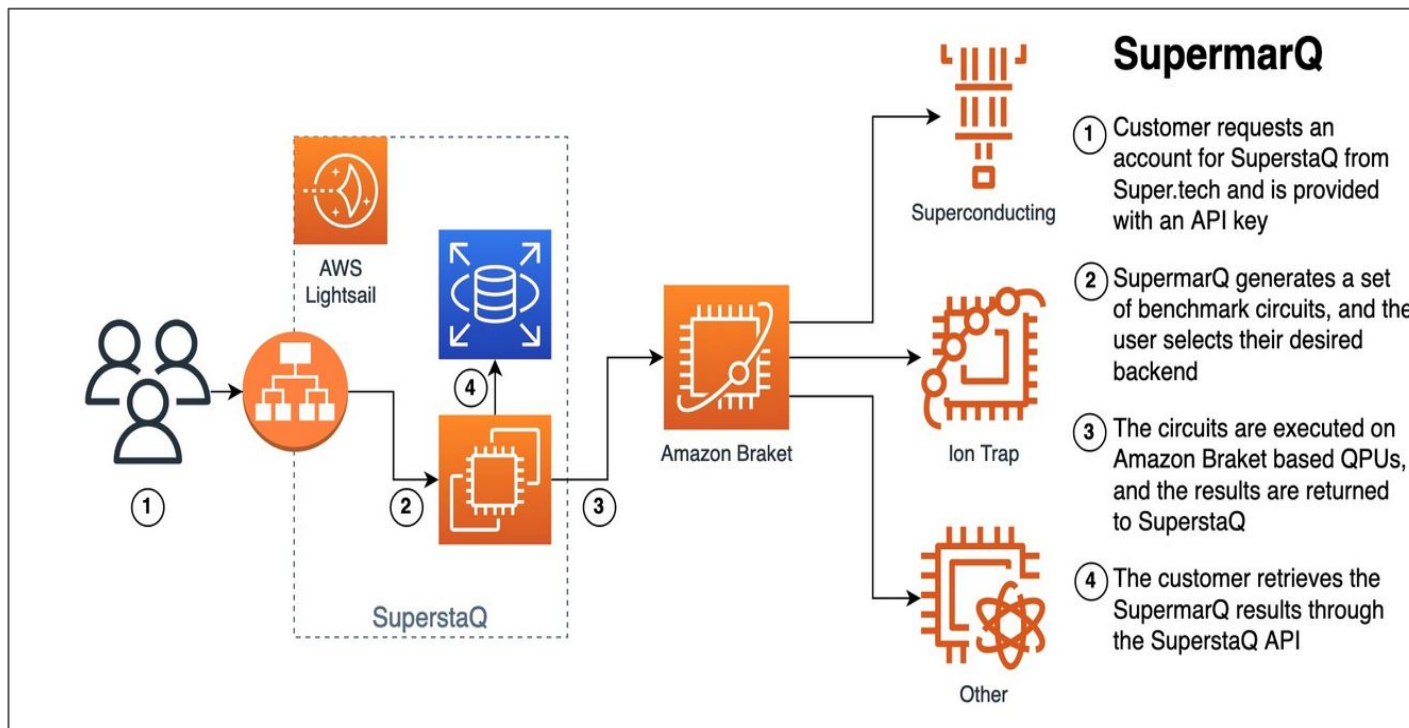
- Owing to the infancy state of quantum computing, the variety of different architecture **questions** the reliable measure and performance comparisons.
- To deal with this, **SupermarQ:: a benchmark suite** which concludes the performance over metrics derived by application level.
- Elected **classical** benchmarking methodology to design this for quantum domain.
- Elucidated some selective set of **feature vectors** for calibration of benchmarking; freedom to select application to **mimic** the real workload parameters, intertwined with IBM , IonQ, AQT.

Supermarq : A robust suite of application benchmarks

FOUR cornerstones of SupermarQ:

- I. **Scalability:** A benchmark suite must be composed of applications whose size is parameterizable and performance is easily verifiable by classical machines.
- II. **Meaningful and diverse:** Benchmark applications should reflect workloads that will appear in practice. Incorporating applications from a range of domains – chemistry, finance, machine learning, etc., will allow results to be relevant to the widest range of potential users and use cases.
- III. **Full-system evaluation:** In the NISQ era, many of the unique properties of different quantum implementations are realized at the compiler level when the program is transpiled to a hardware supported gateset. Since the compiler can control program execution, a benchmark suite should be specified at a shared level of abstraction to allow the compiler to play a role in overall system performance.
- IV. **Adaptivity:** Any suite which aims to accurately measure performance must keep pace with the development of new algorithms, compilation optimizations, and hardware.

Pipeline



Key steps to running a SupermarQ benchmark and architecture of SupermarQ running on SuperstaQ, which is built on Amazon Lightsail and uses Amazon Braket to access QPUs.

Feature Vectors: 1

1) Program Communication (PC)

- A qubit's “degree” is the number of other qubits it interacts with via multi-qubit operations.
- The use of the normalized average degree of the program's **interaction graph** to quantify the communication requirements of quantum circuits.
- interaction graph is formed by taking the qubits to be the vertices and inserting an edge between every pair qubits that interact with one another.
- PC computed by taking the **average** degree of the interaction graph **divided by** the average degree of a complete graph with an equivalent number of qubits.

$$C = \frac{\sum_i^N d(q_i)}{N(N-1)}$$

[sparsely connected applications will have values near 0 while denser programs will be close to 1]

For N- qubits, $d(q_i)$: the degree of qubit q_i^{th} qubit.

Feature Vectors: 2

2) Critical-Depth (CD) [Circuits that are heavily serialized will have a CD that's close to 1.]

- Due to **limited Coherence time** of QPU's qubit , is the life of the information in QC.
- Limited lifetime along with potential gate error of the circuit caused LOW FIDELITY.
- The minimum duration for a quantum circuit is determined by the critical path: the longest span of dependent operations from circuit input to output.
- The critical path is a valuable benchmarking metric because quantum hardware performance must reach specific thresholds to accommodate continuously compounding gate errors.
- The CD feature gives context about how many two qubit interactions in a program lie along the critical path and contribute to the overall CD.

$$D = n_{ed} / n_e$$

D(depth) = # of two qubit interaction on longest path (n_{ed}) / all 2-qubit interactions (n_e)

Feature Vectors: 3

3) Entanglement-Ratio (Ent) is given by the proportion of all gates in the circuit which are two-qubit entangling gates.

- Entanglement is a critical property which known as its strength of Quantum Computing.
- While it is in general quite difficult to measure the precise amount of entanglement at every point within a circuit.
- One can roughly capture this feature by computing the proportion of all gate operations (n_g) which are two-qubit interactions (n_e).

$$E = n_e / n_g$$

Ent = 2-qubit interaction / all gate operations

Feature Vectors: 4

4) Parallelism(Par) is comparison ration of qubit (n), gates(n_g) and CD(d).

—The structure of different quantum algorithms allow for varying degrees of parallelization

– Highly parallel applications fit a large number of operations into a relatively small circuit depth and will therefore have a parallelism feature close to 1.

– Parallel operations can also stress the quantum hardware because of correlated noise events known as “cross-talk” that degrade program performance. Cross-talk, often caused by simultaneous gate execution, is a common source of error in NISQ systems.

$$P = \left(\frac{n_g}{d} - 1 \right) \frac{1}{n - 1}$$

Where # of qubits (n), # of gates(n_g) and Critical Depth(d)[feature-Vector#1].

Feature Vectors: 5

5) Liveness (Liv):

- During program execution, a qubit will either be involved in computation or it will be idle; waiting for its next instruction.
- In reality, unwanted environmental interactions such as amplitude damping, dephasing, and correlated noise cause decoherence, unlike in an ideal environment, the qubit's state would stay coherent while idling.
- Liv captures aspects of an application's qubit status during its lifetime.
- A is the liveness matrix defined by taking a quantum circuit and forming a matrix with n-rows equal to the number of qubits and a number of columns equal to the circuit depth(d).

$$L = \frac{\sum_{ij} A_{ij}}{nd}$$

A qubit may either be involved in an operation or idle, corresponding to entries

in the liveness matrix $A[n \times d]$: n= number of qubit and d =CD

1-L = inactivity of q_ckt

Feature Vectors: 6

6) Measurement (Mea) : NISQ devices suffer from non-trivial amounts of measurement error.

- Qubit-specific measurement is a critical part of quantum computing.
- In fault-tolerant quantum computing, error correcting codes use measurement to extract entropy from a noisy quantum system.
- It focuses specifically on the mid-circuit measurement and reset operations within a quantum program.











$$M = l_{mcm} / d$$

Where L = # of layers, d =Circuit Depth (Feature Vector #1).

d= sequential layers of gate operations

l_{mcm} is the number of layers which contain these measurement and reset operations.

SupermarQ's Benchmark Suite & their Applicable Domains:

 QAQA Finance  Logistics	 VQE Chemistry  Battery Development	 Hamiltonian Simulation Pharma  Energy (nuclear, photosynthesis)
 GHZ Quantum Networks  Sensing Physical Signals	 Mermin-Bell Test of "Quantum-ness"	 Error Correction Immunity to Noise

Feature maps and sample circuits for each benchmark

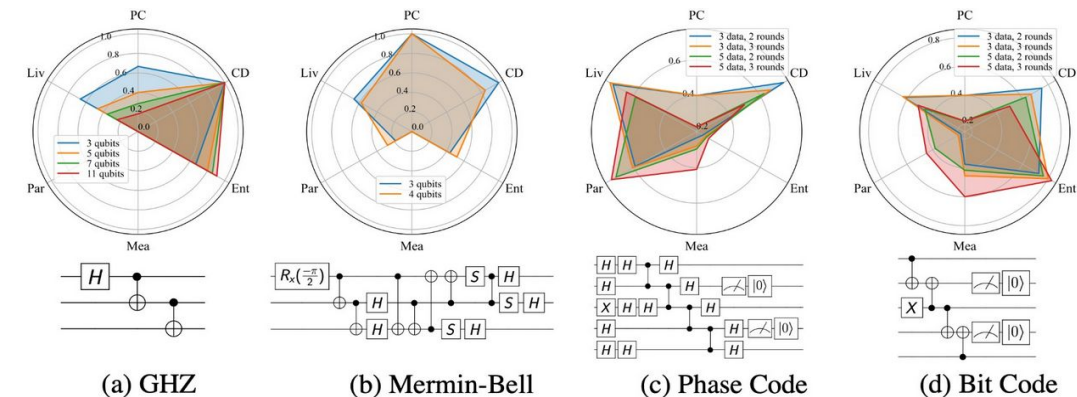
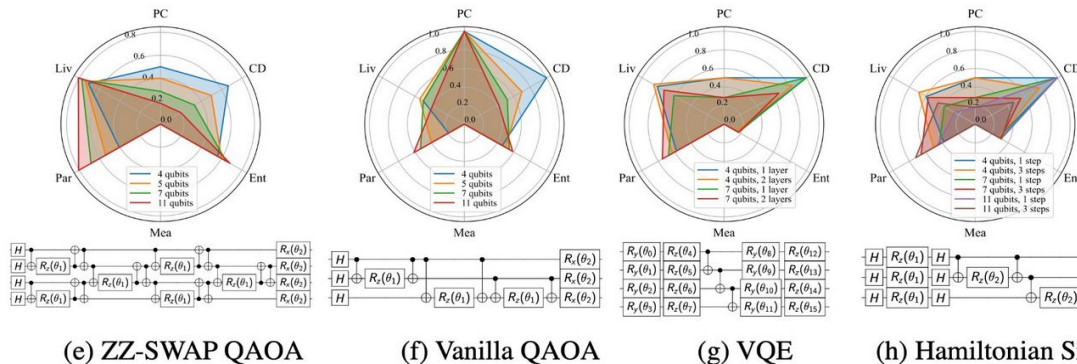
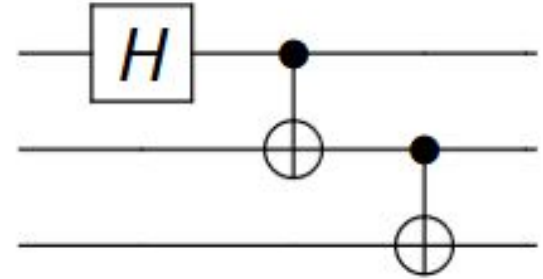


Figure: Feature maps and sample circuits for each of the benchmarks evaluated



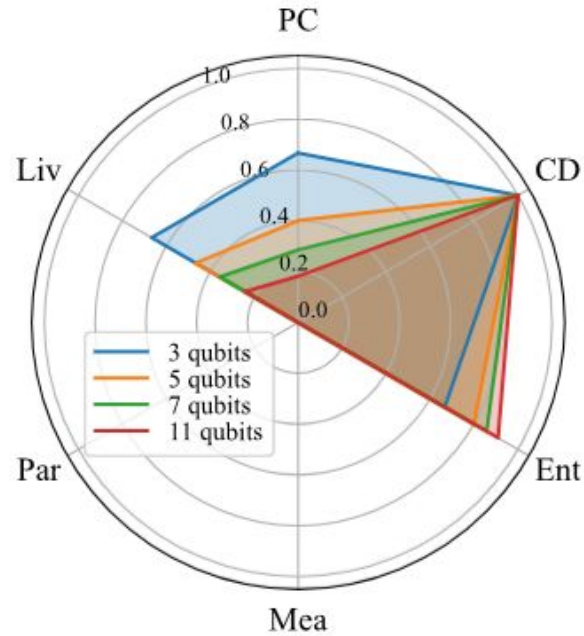
Benchmark 1 : GHZ(Greenberger-Horne-Zeilinger):

- The generation of entanglement between qubits is one of the most important tasks in quantum computing, sensing, and networking.
- The GHZ benchmark consists of a Hadamard gate followed by a ladder of CNOTs to produce the entangled state: $(|00\dots 0\rangle + |11\dots 1\rangle) / \sqrt{2}$
- The performance metric is the [Hellinger fidelity](#) between the experimentally observed probability distribution and the ideal distribution (50% $|00\dots 0\rangle$ and 50% $|11\dots 1\rangle$)



The fidelity is defined as $(1-H^2)^2$, where H is the Hellinger distance. This value is bounded in the range $[0, 1]$

Benchmark 1 : GHZ(Greenberger-Horne-Zeilinger):



Benchmark 2 : Mermin-Bell: An example of bell-inequality test.

- In this benchmark, a GHZ state, $|\phi\rangle = (1/\sqrt{2})(|00 \dots 0\rangle + i |11 \dots 1\rangle)$, is first prepared before measuring the expectation value of the Mermin operator,

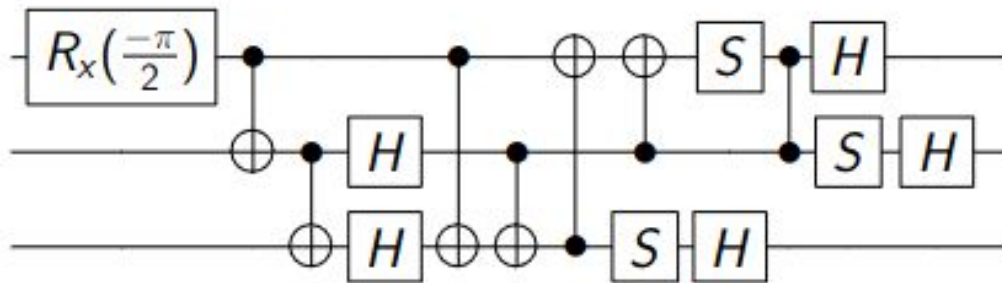
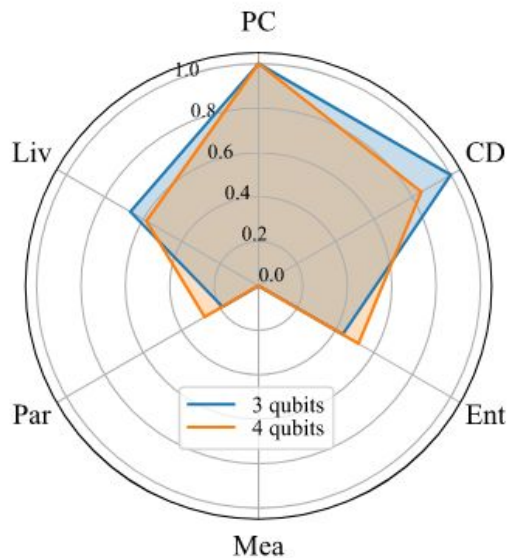
$$M = \frac{1}{2i} \left(\prod_{j=1}^n (\sigma_x^j + i\sigma_y^j) - \prod_{j=1}^n (\sigma_x^j - i\sigma_y^j) \right)$$

where σ_x^j and σ_y^j are the Pauli-X and -Y operators acting on the j -th qubit. If nature is quantum $\langle \phi | M | \phi \rangle = 2^{n-1}$ of this operator for an n qubit system is

- If nature is classical and obeys a theory of local-hidden variables, then expectation value of the Mermin operator is bounded by $\langle \phi | M | \phi \rangle \leq 2^{(n-(n \bmod 2))/2}$
- We measure performance by computing $(\langle \phi | M | \phi \rangle + 2^{n-1})/2^n$ as the benchmark score.

Benchmark 2 : Marmin-Bell: An example of bell-inequality test.

—After preparing the GHZ state, the remaining gates within the Mermin-Bell circuits rotate the quantum state into the shared basis of the Mermin operator such that the expectation of each term can be measured simultaneously.



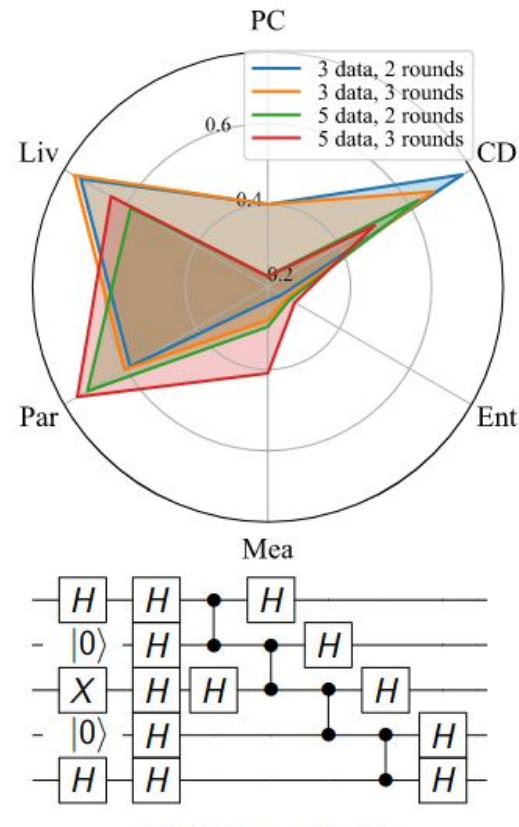
But unlike GHZ benchmark, the basis-change portion of the circuit begins to dominate the state preparation as the size of the benchmark increases.

Benchmark 3 : Phase Code Proxy-Application [Error - Correction Subroutine #1]

- The phase code benchmark is a phase flip repetition code parameterized by the number of data qubits and rounds of error correction.
- We therefore compute the Hellinger fidelity between the experimental and ideal distributions as a measure of performance.
- To measure performance, we first prepare the data qubits in initial

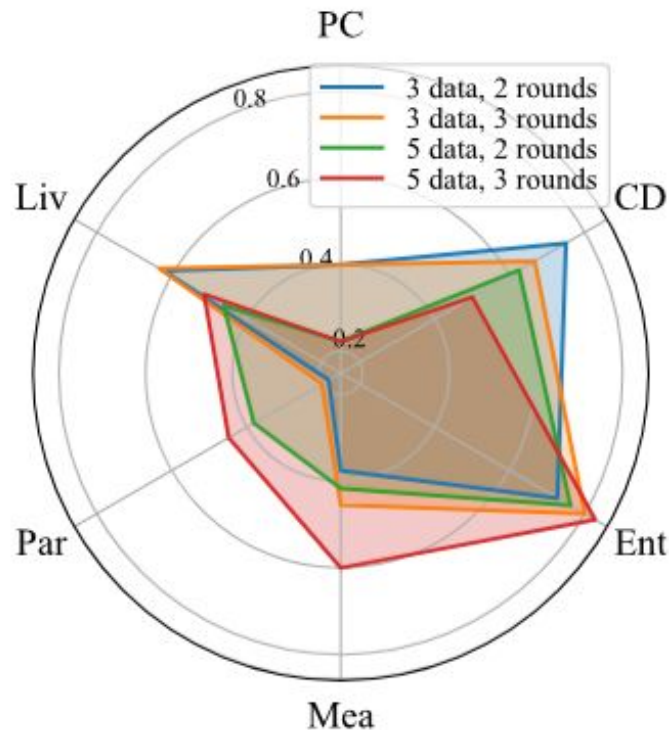
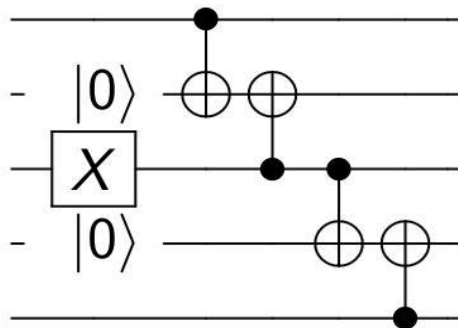
$$|+\rangle = (|0\rangle + |1\rangle) / \sqrt{2} \text{ OR } |-\rangle = (|0\rangle - |1\rangle) / \sqrt{2}$$

states followed by r rounds of error correction and finally a measurement of the final state.



Benchmark 4 : Bit Code Proxy-application: [Error - Correction Subroutine #2]

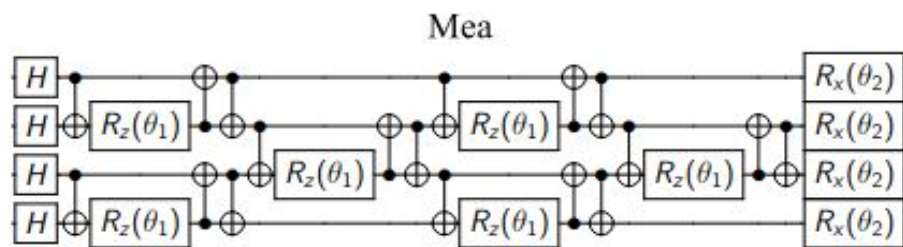
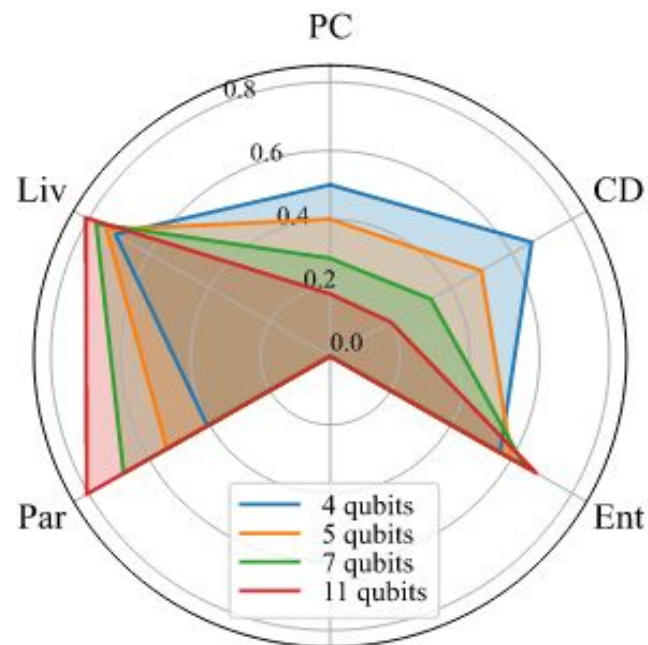
- The bit code detects bit flips on the data qubits.
- Bit code benchmark is also a bit flip repetition code that is parameterized by the number of data qubits and error correction rounds.
- A sample circuit with three data qubits initialized in the $|010\rangle$ state and a single round of error correction. Since the ideal final state is known a priori, also use the Hellinger fidelity as the score function for this benchmark.



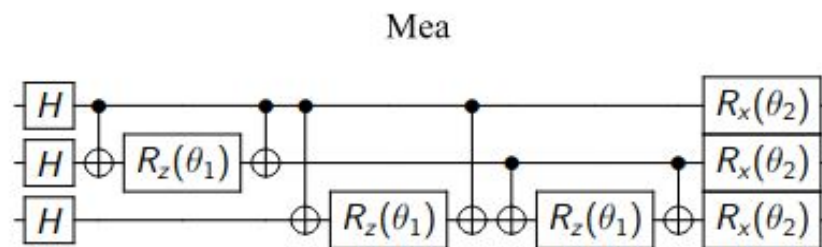
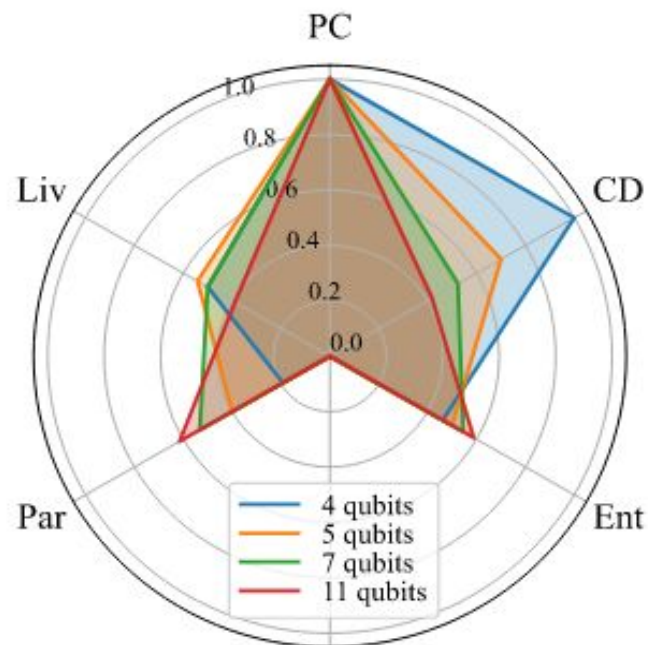
Benchmark 5 : Vanilla QAOA [Quantum Approximate Optimization Algorithm]:

- It is trained to output bit-strings to solve combinatorial optimization problem.
- Benchmarked QAOA for MaxCut on complete graphs with edge weights randomly drawn from $\{-1, +1\}$, known as Sherrington-Kirkpatrick(SK) model.
- Uses an ansatz; equivalent to SK model; a typical mutation of QAOA; required all to all connectivity.
- This ansatz is a natural choice for solving MaxCut on the SK model which requires an interaction between every pair of qubits (i.e., $n(n - 1)/2$ edges).
- Full QAOA benchmark would require thousands of iterations to reach convergence, evaluating the same becomes infeasible because of the wait times would be high.

**SK Model:: [Sherrington-Kirkpatrick Model](#)



ZZ-SWAP QAOA



Vanilla QAOA

Benchmark 6 : ZZ-SWAP QAOA:

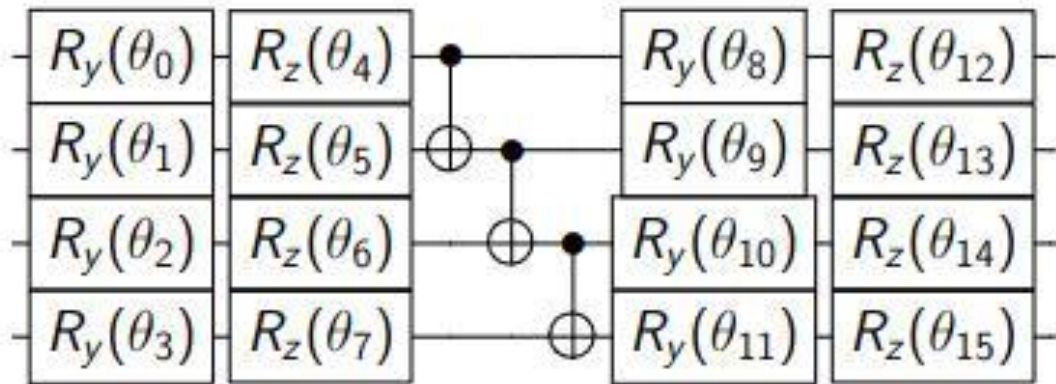
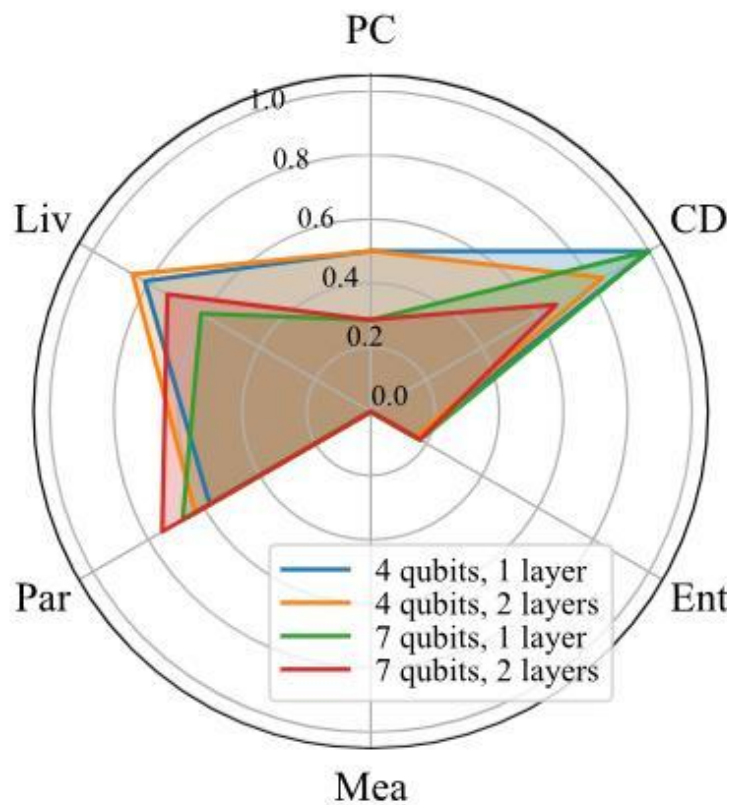
- The ZZ-SWAP QAOA benchmark implements a variational ansatz known as a SWAP network.
- This ansatz is a natural choice for solving MaxCut on the **SK model which requires an interaction between every pair of qubits (i.e., $n(n - 1)/2$ edges)
- The SWAP network (a sample circuit is shown in Figure before) is able to perform all $O(n^2)$ required interactions using a quantum circuit whose depth scales as $O(n)$.
- We compared the experimental and ideal results by measuring the expectation value, $\langle H \rangle$, and computing as the benchmark score.
$$1 - \left| \frac{\langle H \rangle_{ideal} - \langle H \rangle_{exper}}{2\langle H \rangle_{ideal}} \right| ;$$
- the performance measure for the full QAOA benchmark would be the final MaxCut value achieved after optimization, allows for straightforward comparisons with other quantum or classical MaxCut algorithms.

Benchmark 7 : Proxy-Application VQE:

- The goal of this algorithm is to find the lowest eigenvalue of a problem matrix by computing a difficult cost function on the QPU and feeding this value into an optimization routine running on a CPU.
- Typically, the problem matrix is the Hamiltonian governing a target system and the lowest eigenvalue corresponds to the system's ground state energy.
- Targeting the one dimensional transverse field Ising model (**TFIM, also called the transverse Ising chain) and use VQE to find its ground state energy.
- 1D **TFIM is a useful model for understanding phase transitions in magnetic materials, s a scalable benchmark because it is exactly solvable via classical methods.
- Converted into Proxy-application here, that measures performance for a single iteration of the VQE algorithm.
- Compared this energy with the value obtained classically and compute the same score function as the QAOA benchmark, and Hardware efficient ansatz used in this benchmark.

**TFIM = A one dimensional transverse field Ising model transverse Ising chain, is a useful model for understanding phase transitions in magnetic materials.

Benchmark 7 : Proxy-Application VQE:



Benchmark 8 : Hamiltonian Simulation

—Closing the gap between the algorithmic resource requirements and the capabilities of QC systems may lead to breakthroughs in the development of new batteries and catalysts.

– The Hamiltonian for this system, consisting of N spins, may be written as

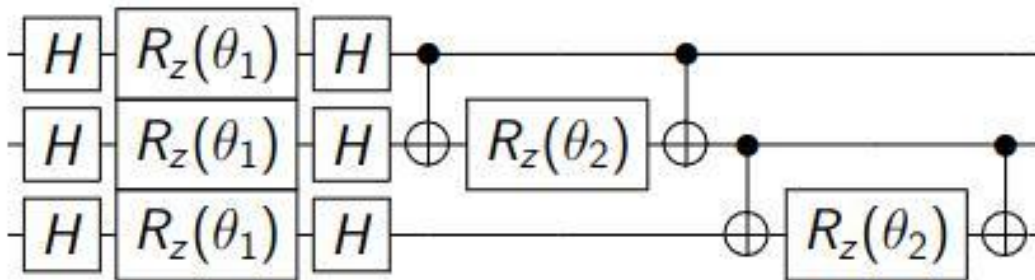
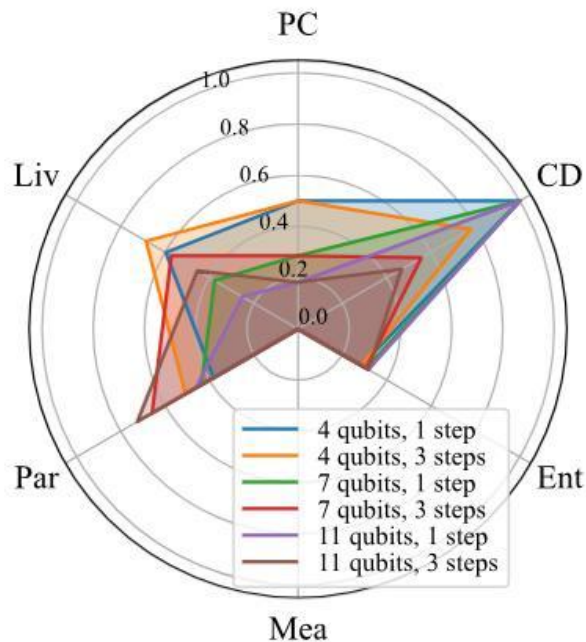
$$H = - \sum_{i=1}^N (J_z \sigma_z^i \sigma_z^{i+1} + \epsilon_{ph} \cos(\omega_{ph} t) \sigma_x^i)$$

Where J_z = coupling constant, determines the strength of the nearest-neighbor interactions.

– Second term of above written equation describes the time-varying magnetic field.

Benchmark 8 : Hamiltonian Simulation

— The experimentally obtained average magnetization is then compared to the exact value obtained classically.



Coverage Analysis :

- To find the coverage of a given set of applications, we compute the volume of the convex hull defined by their feature vectors: each shape in the feature maps which corresponds to a single vector within the higher dimensional feature space.

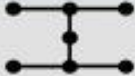
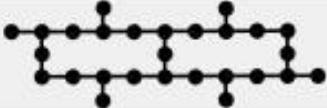
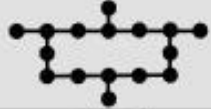
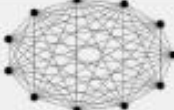

Suite	Volume	Circuits
SupermarQ (this work)	9.0e-03	52
QASMBench [30]	4.0e-03	62
Synthetic	1.4e-03	6
CBG2021 [84]	1.6e-08	10476
TriQ [17]	4.1e-14	12
PPL+2020 [16]	1.0e-15	9

- Only SupermarQ and QASM Bench attain coverage superior to the synthetic benchmark suite.

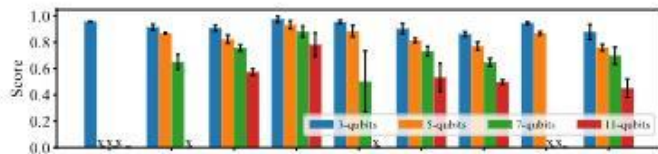
Characteristics of the QC systems, to evaluate benchmarks :

— The IBM and IonQ data was taken from the public documentation available through their respective cloud providers (IBM Qiskit and AWS Braket),2021.

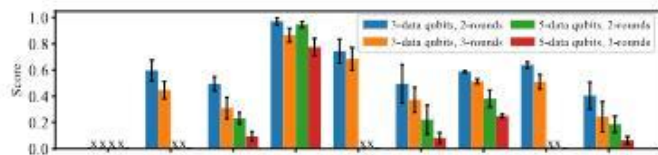
—The device statistics for the IBM QPUs not pictured here are available online through IBM Quantum [85]. The AQT system properties were obtained via randomized benchmarking ,2021

Machine	Qubits	Coherence Time (μ s) (T1, T2)	Gate Times (μ s) (1Q, 2Q, Meas)	Gate Errors (%) (1Q, 2Q, Meas)	Topology
IBM-Casablanca	7	91.21, 125.23	0.035, 0.443, 5.9	0.028, 0.83, 2.09	
IBM-Montreal	27	104.14, 86.88	0.035, 0.423, 5.2	0.052, 1.76, 1.96	
IBM-Guadalupe	16	99.52, 104.99	0.035, 0.416, 5.4	0.043, 1.03, 2.79	
IonQ	11	>1e7, 2e5	10, 210, 100	0.28, 3.04, 0.39	
AQT	4	62, 37	0.03, 0.152, 1.02	0.083, 2.1, 1.25	

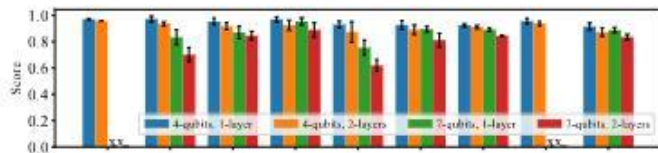
Benchmark Results evaluated across Superconducting & Trapped-Ion Devices :



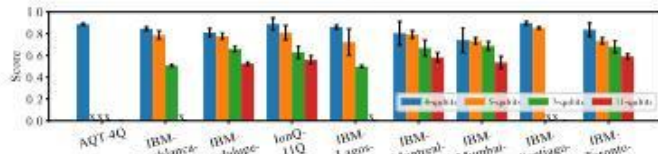
(a) GHZ



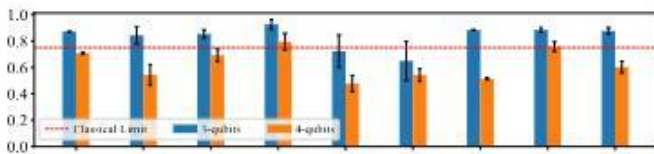
(c) Bit Code



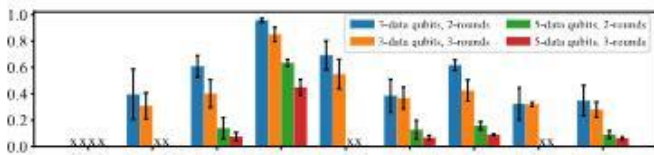
(e) VQE



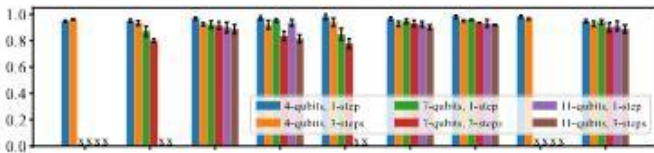
(g) ZZ-SWAP QAOA



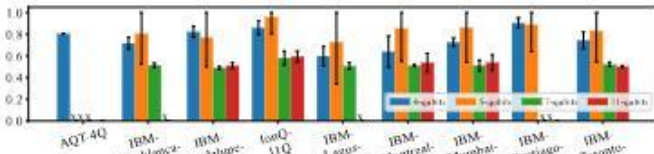
(b) Mermin-Bell



(d) Phase Code



(f) Hamiltonian Simulation



(h) Vanilla QAOA

– In every benchmark run, the execution 2000 shots on the IBM devices, 1024 on the AQT device, and 35 on the IonQ processor.

– Shot counts selection is to maintain a reasonable cost budget for collecting the benchmark results.

Benchmarking for All

- SupermarQ to simultaneously submit the same quantum circuit instance to multiple available devices via the cloud.
- Specified at the level of OpenQASM, a popular intermediate representation for quantum circuits.
- Adhering to the principle of full-system evaluations, SupermarQ uses optimizations that are publicly available.
- Include the transpilation of OpenQASM to native gates, noise-aware qubit mapping, SWAP insertions, reordering of commuting gates, and cancellation of adjacent gates.
- optimizations included are those that are automatically applied when using cloud-based platforms like Qiskit, cirQ. This matches the level of optimization that would be available to the typical user.

Conclusion

- SupermarQ also features an error correction benchmark, a proxy for practical readiness that indicates how far a device has to go on the road to fault tolerance.
- SupermarQ defines a set of feature vectors to quantify coverage, selects applications from a variety of domains to ensure the suite is representative of real workloads, and collects benchmark results.
- sophisticated developers can also refer to performance features (like critical depth or entanglement ratio) that help them project how each of the benchmarked systems (IonQ versus Advanced Quantum Testbed, for example) would perform on *any* algorithm, even custom algorithms outside the benchmark set.