How can software engineers help building and using quantum computers?

Shelly Garion

shelly@il.ibm.com

May 2022

IBM **Quantum**

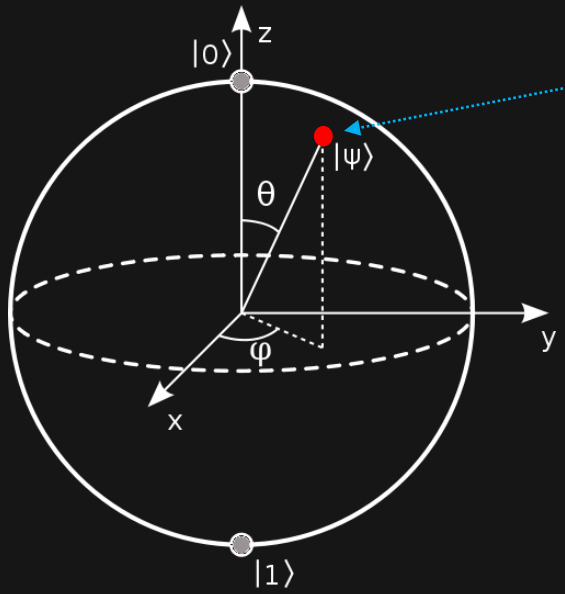
Bits vs Qubits

Classical Bit

0

1

Quantum Bit

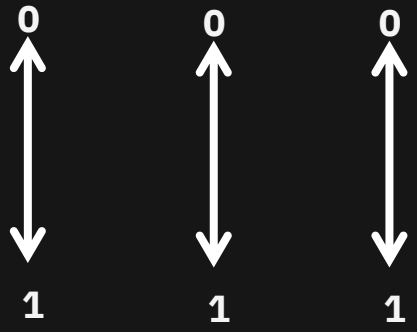


Multiple Bits/Qubits and states

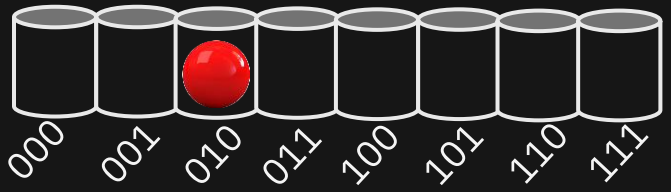
IBM **Quantum**

N bits/qubits→ 2^N combinations (states)

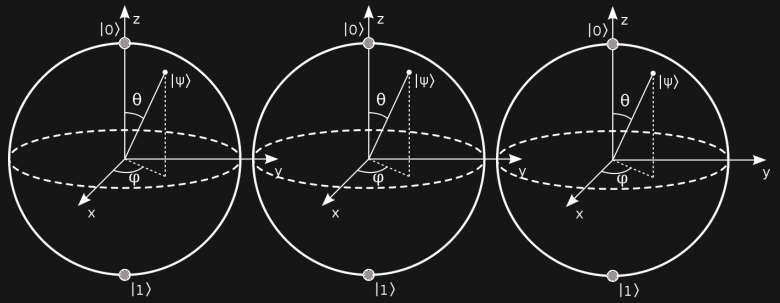
3 bits = 8 possible states



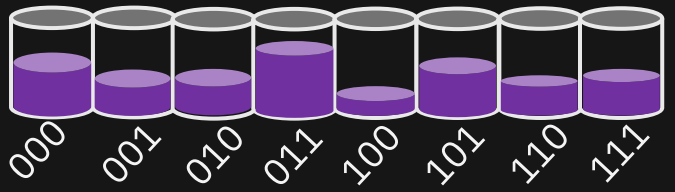
Computer **<u>at only 1 state</u>** at any given time



3 qubits = 8 possible states



Quantum computer **<u>at many (possible all) states</u>** at any given time



Comparison: Bits versus Qubits

Simulating a set of qubits demonstrates their potential advantage as information carriers.

Qubits	Digital bits required to represent an entangled state
2	512 bits
3	1024 bits
10	16 kilobytes
16	1 megabyte
20	17 megabytes
30	17 gigabytes
35	550 gigabytes
160	More than all the atoms of planet earth
280	More than all the atoms in the universe

What can we do with a quantum computer ?

IBM **Quantum**

Some promising application domains for near time quantum computers:

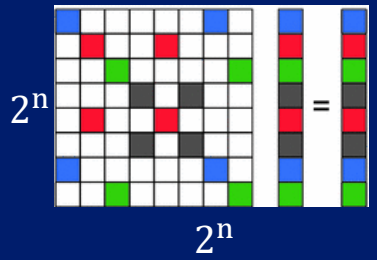
Quantum Simulations

Physics

Chemistry

Materials discovery

Linear Systems ($\mathbf{Ax = b}$)



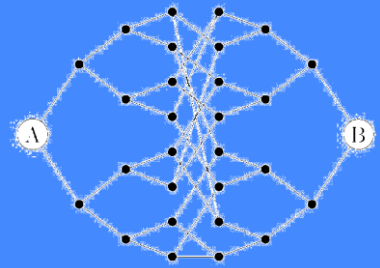
Network analysis

Differential equations

Option pricing, heat transfer

Classification (Machine Learning)

Quantum Walks



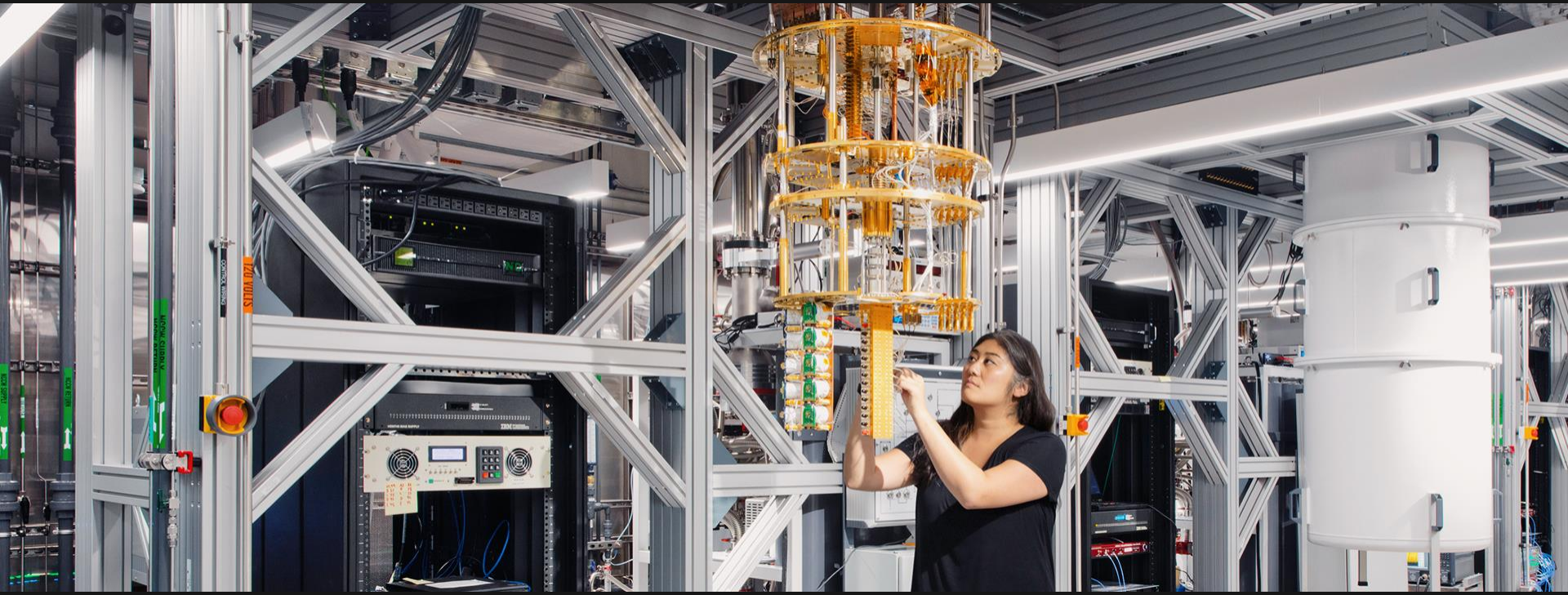
Graph properties (network flows, electrical resistance)

Search

Collision finding

Quantum computer – behind the scenes

IBM **Quantum**

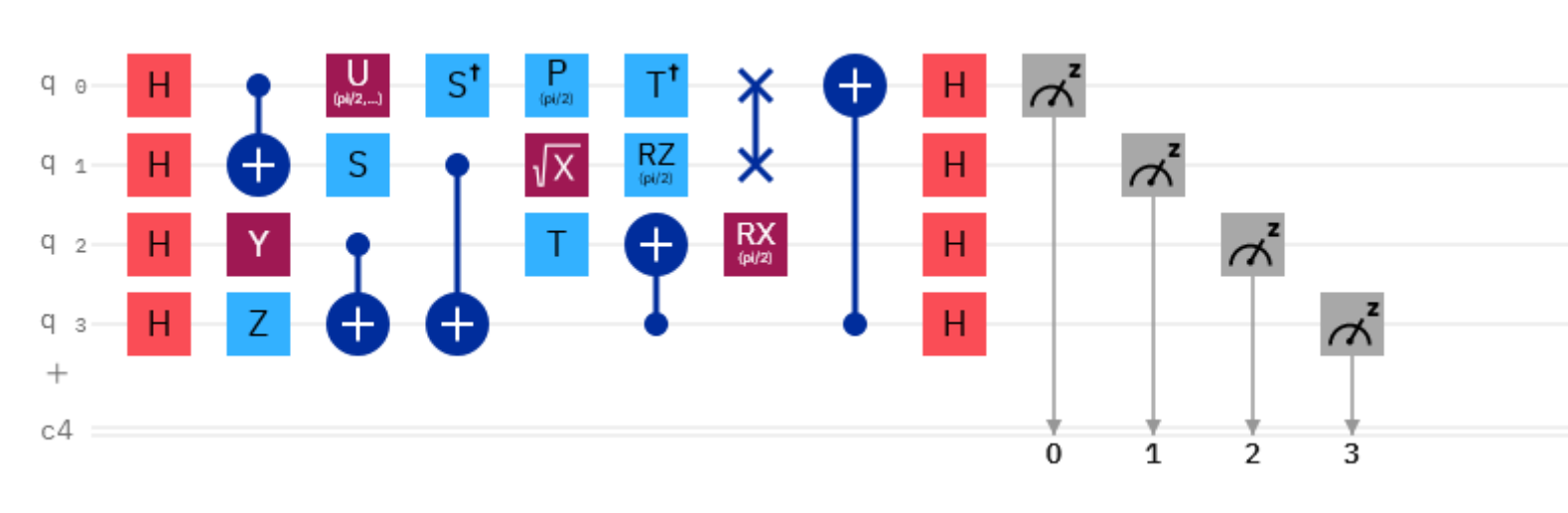


How can software engineers help building and using quantum computers?

- Quantum algorithms development
- Software Development Kit (e.g., Qiskit open-source)
- Cloud infrastructure (e.g., runtime env for quantum-classical programs)
- Backend services (e.g., the backend compiler, QASM and pulse)
- SW for the control HW
- UI/UX programming
- Security aspects

Quantum computer – users view

IBM **Quantum**



https://quantum-computing.ibm.com/

IBM Quantum Composer & IBM Quantum Lab

Interactive Web UI for building and running quantum circuits

Quantum computer – users view

IBM **Quantum**

ibmq_brooklyn

Exploratory

Details

65
Qubits

32
QV

753
CLOPS

Status:	Online
Total pending jobs:	40 jobs
Processor type:	Hummingbird r2
Version:	1.4.3
Basis gates:	CX, ID, RZ, SX, X
Your usage:	0 jobs

Avg. CNOT Error:	2.621e-2
Avg. Readout Error:	3.139e-2
Avg. T1:	76.58 us
Avg. T2:	77.03 us
Providers with access:	1 Providers
Supports Qiskit Runtime:	Yes

Your upcoming reservations 0

Calibration data

Last calibrated: an hour ago

Map view | Graph view | Table view

Qubit:

Frequency (GHz)

Avg 5.133

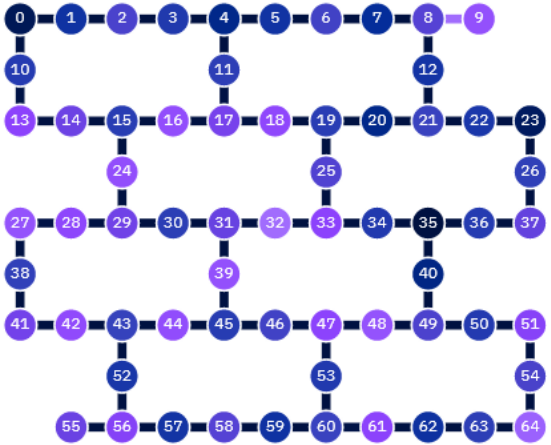
min 4.814 max 5.401

Connection:

CNOT error

Avg 2.621e-2

min 6.794e-3 max 1.000e+0



The three key metrics for measuring quantum computing performance

IBM **Quantum**

Scale

Measured by **number of qubits** which indicates the amount of information we can encode in the quantum system.

High coherence, high reliability, lower cost

2019	Today	2021
27 qubits	65 qubits	127 qubits

Quality

Measured by **Quantum Volume** which indicates quality of circuits and how faithfully circuits are implemented in hardware.

Need low operation errors, meaning large Quantum Volume

2019	Today	2021
32 QV	128 QV	256 QV

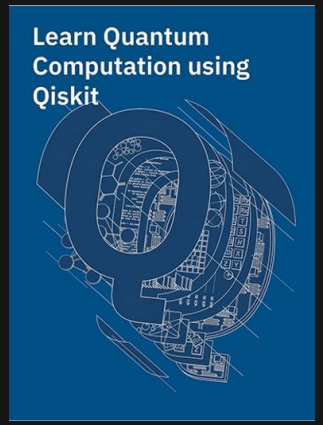
Speed

Measured by **CLOPS (Circuit Layer Operations Per Second)** which indicates how many circuits can run on hardware in a given time.

Seamless synchronization of quantum and classical circuits increases execution rate

2019	TODAY
200 CLOPS (inferred)	1,400 CLOPS

IBM’s open-source tools

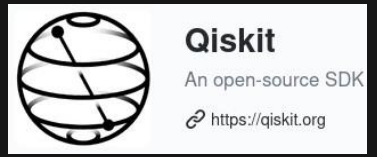


[IBM Qiskit textbook]

Learn quantum via hands-on experience with Qiskit

[Qiskit YouTube channel]

Lectures from introductory to advanced quantum computing topics

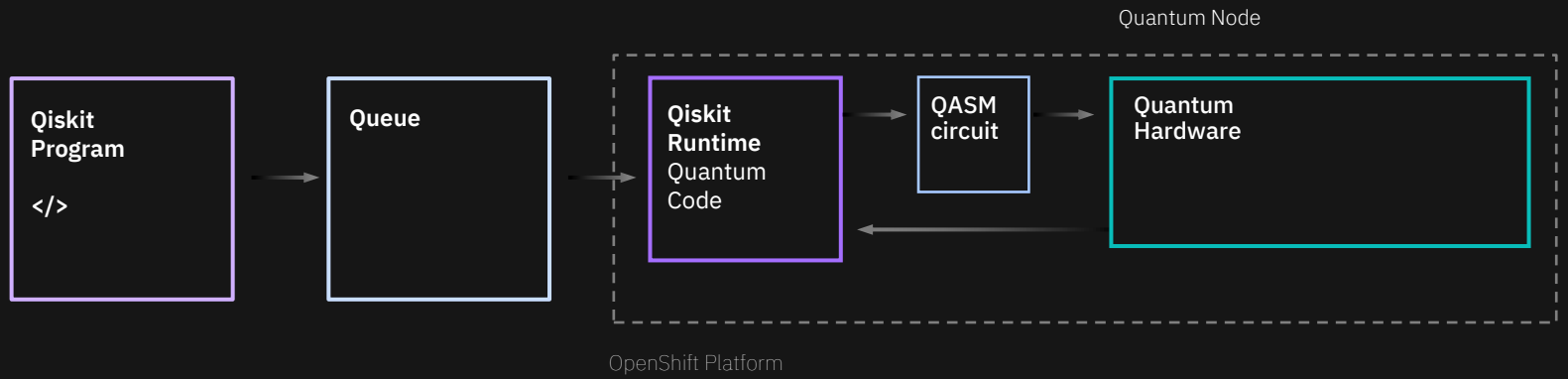


[Qiskit open-source]

Quantum computing software development mostly in Python

https://github.com/Qiskit

Qiskit Runtime
Quantum-classical programs



120x speedup for an example quantum chemistry algorithm

How does our team help building and using quantum computers?

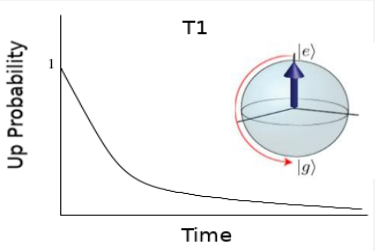
- Experimentalists tools: characterization and benchmarking of the quantum hardware
- Circuits & transpilation: efficiently compiling quantum circuits

Characterization for experimentalists

IBM **Quantum**

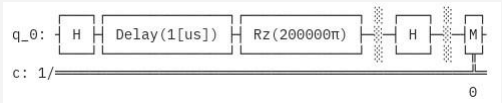
https://github.com/Qiskit/qiskit-experiments

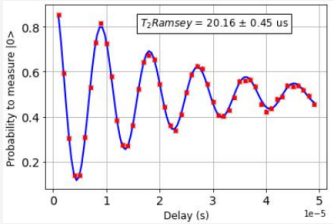
T1 – relaxation



$$P(t) = Ae^{-t/T_1} + B$$

T2* - detuning

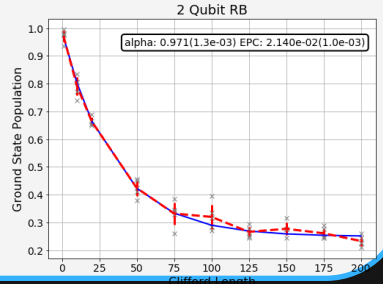




Randomized Benchmarking

$$A_0\alpha^m + B_0$$

$$r = 1-\alpha-\frac{1-\alpha}{2^n}$$



- State, Process and Gate Set Tomography
- Quantum Volume

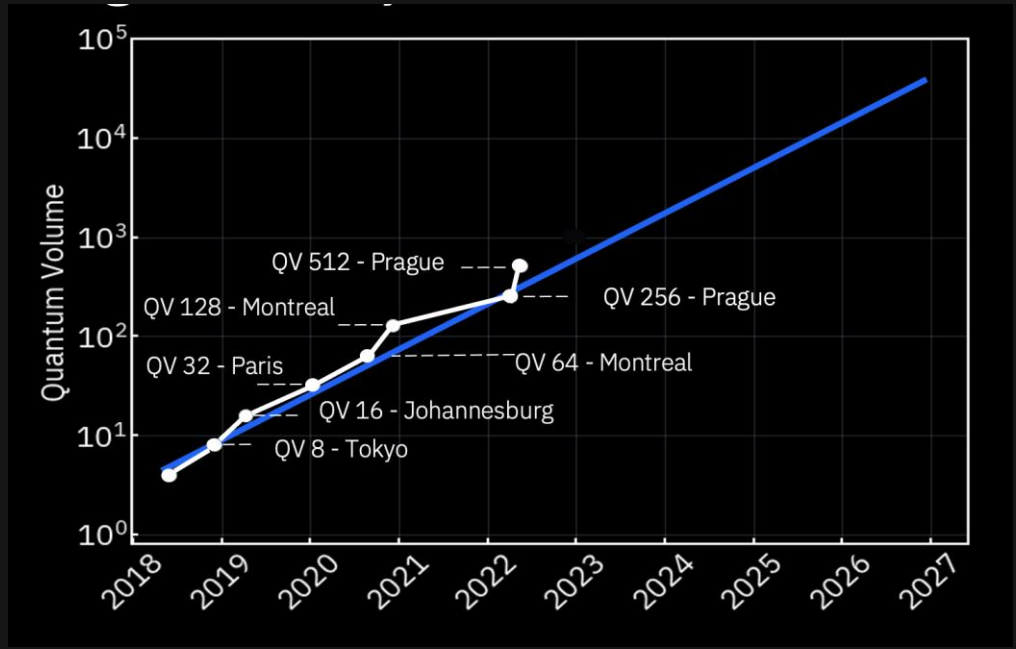
R&D goals: **robust** and **scalable** characterization methods

Quantum volume

Quantum Volume (QV) is a single-number metric that measures the largest quantum computational space a device can “explore”

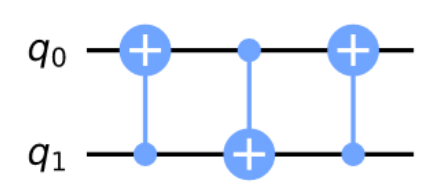
QV = 2^N where N=number of qubits

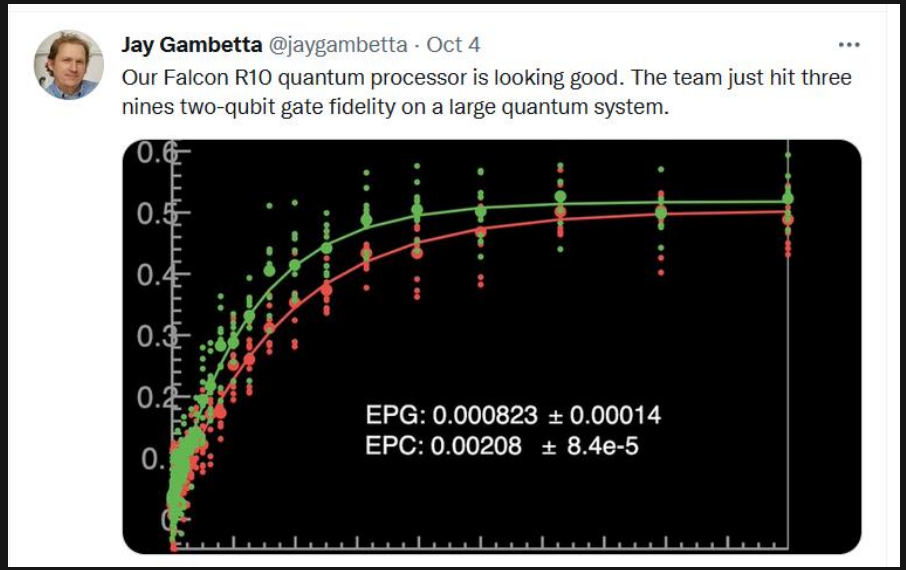
- High-fidelity operations
- High connectivity
- Large calibrated gate sets
- Low single-qubit errors
- Low measurement errors
- Minimal cross-talk
- Smart circuit rewriting software
- Stable control electronics
- ....



Quantum circuit synthesis challenges

- Reduce the circuit depth
- Reduce the number of 2-qubit gates (noisier than 1-qubit gates)
- The quantum device does not have all-to-all connectivity
- One SWAP gate = Three CNOT gates





Development Roadmap

Executed by IBM
On target

IBM Quantum

	2019	2020	2021	2022	2023	2024	2025	2026+
	Run quantum circuits on the IBM cloud	Demonstrate and prototype quantum algorithms and applications	Run quantum programs 100x faster with Qiskit Runtime	Bring dynamic circuits to Qiskit Runtime to unlock more computations	Enhancing applications with elastic computing and parallelization of Qiskit Runtime	Improve accuracy of Qiskit Runtime with scalable error mitigation	Scale quantum applications with circuit knitting toolbox controlling Qiskit Runtime	Increase accuracy and speed of quantum workflows with integration of error correction into Qiskit Runtime
Model Developers					Prototype quantum software applications → Quantum software applications		Machine learning | Natural science | Optimization	
Algorithm Developers		Quantum algorithm and application modules			Quantum Serverless			
		Machine learning | Natural science | Optimization			Intelligent orchestration	Circuit Knitting Toolbox	Circuit libraries	
Kernel Developers	Circuits		Qiskit Runtime					
				Dynamic circuits	Threaded primitives	Error suppression and mitigation		Error correction
System Modularity	Falcon 27 qubits	Hummingbird 65 qubits	Eagle 127 qubits	Osprey 433 qubits	Condor 1,121 qubits	Flamingo 1,386+ qubits	Kookaburra 4,158+ qubits	Scaling to 10K-100K qubits with classical and quantum communication
					Heron 133 qubits x p	Crossbill 408 qubits		

Thank you !

IBM **Quantum**

https://quantum-computing.ibm.com