THE LINUX FOUNDATION

OPEN SOURCE SUMMIT INDIA

Building Hybrid Quantum-Classical Pipelines

Guncha Malik

Qiskit Advocate | IBM Quantum Technical Ambassador

Sainath Sativar, Amutamil E, Divya Singh - Qiskit Advocate

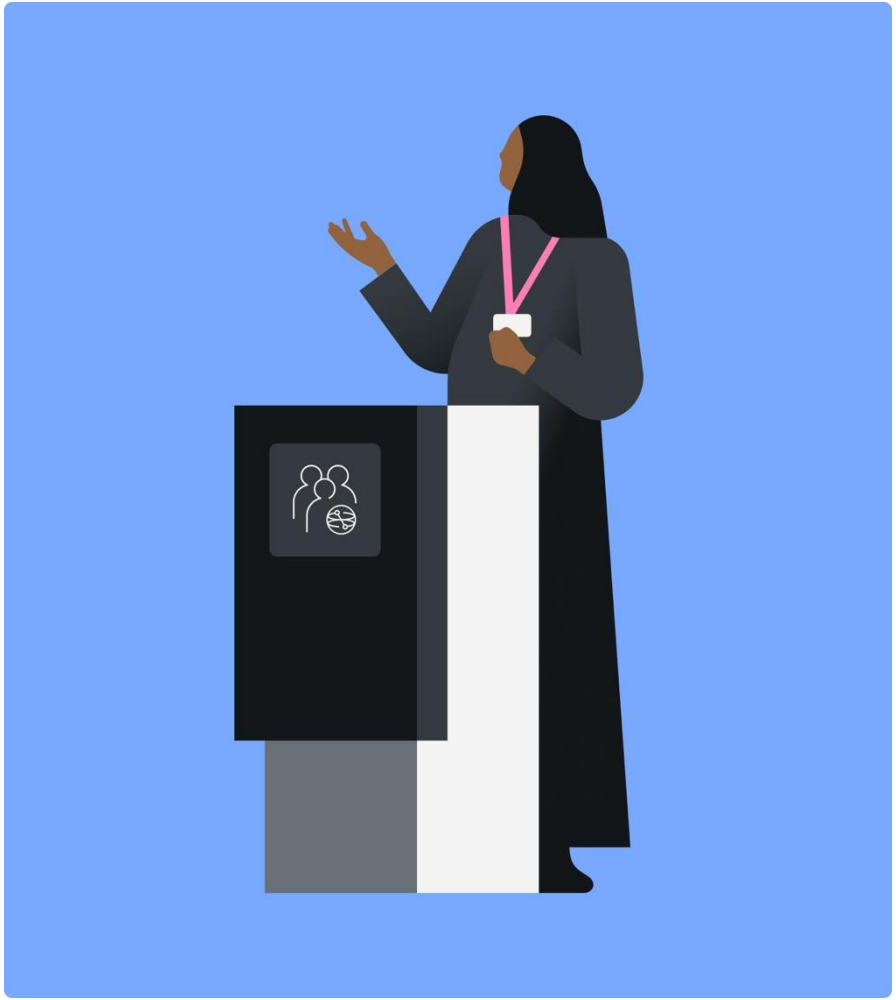
#OSSummit

THE LINUX FOUNDATION

Agenda

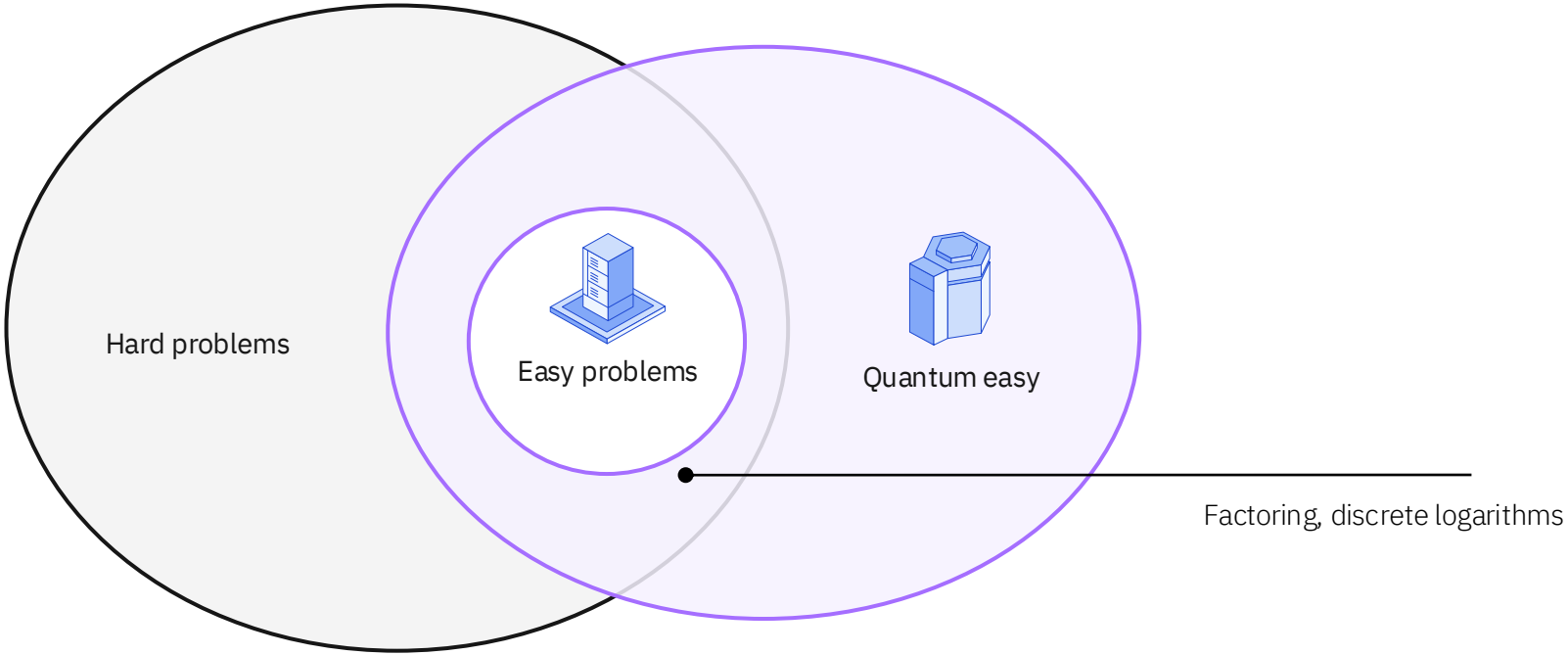
- Quantum-centric Super Computing
- Quantum fundamentals
- Qiskit
- Hybrid Pipeline Demo
- Q&A

Quantum-centric super computing



The source of quantum risk

There is a rich seam of problems that cannot be solved by classical and AI supercomputing, and never will. These are the trillion-dollar problems that quantum computing was designed to solve.



The new wave of computing

Classical computer
Well suited for many problems

Quantum computer
Unlock classically intractable problems

What are these problems?

Modeling molecules, atoms, electrons, and quarks with unprecedented accuracy

Developing lighter, longer-lasting batteries for electric vehicles, electronics, and energy grid storage

Designing lighter, stronger materials to allow planes to be more efficient and to need less maintenance

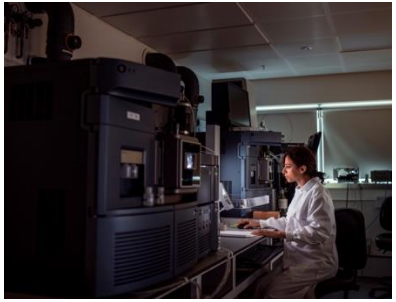
Discovering new classes of antibiotics to counter the emergence of multidrug-resistant bacterial strains

Designing optimal superconductors for MRI, electromobility, and renewable energies

What are these problems?

Solving algebra in [exponential] spaces.
Finding hidden patterns in structured problems.

Improving understanding of physical and chemical properties of materials

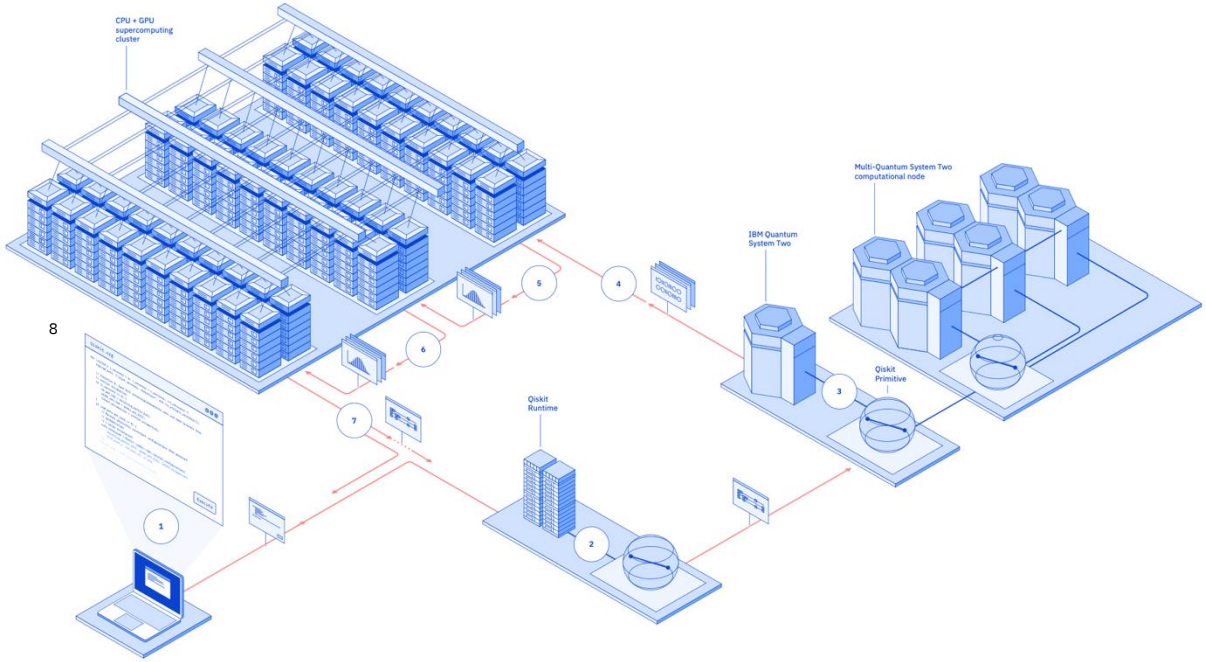


Improving patient outcomes by designing optimal cell-centric therapeutics

Strengthening risk management through better time series and sequence prediction

Optimizing vehicle routing and scheduling for large-scale logistics networks

CPUs
+ GPUs
+ QPUs



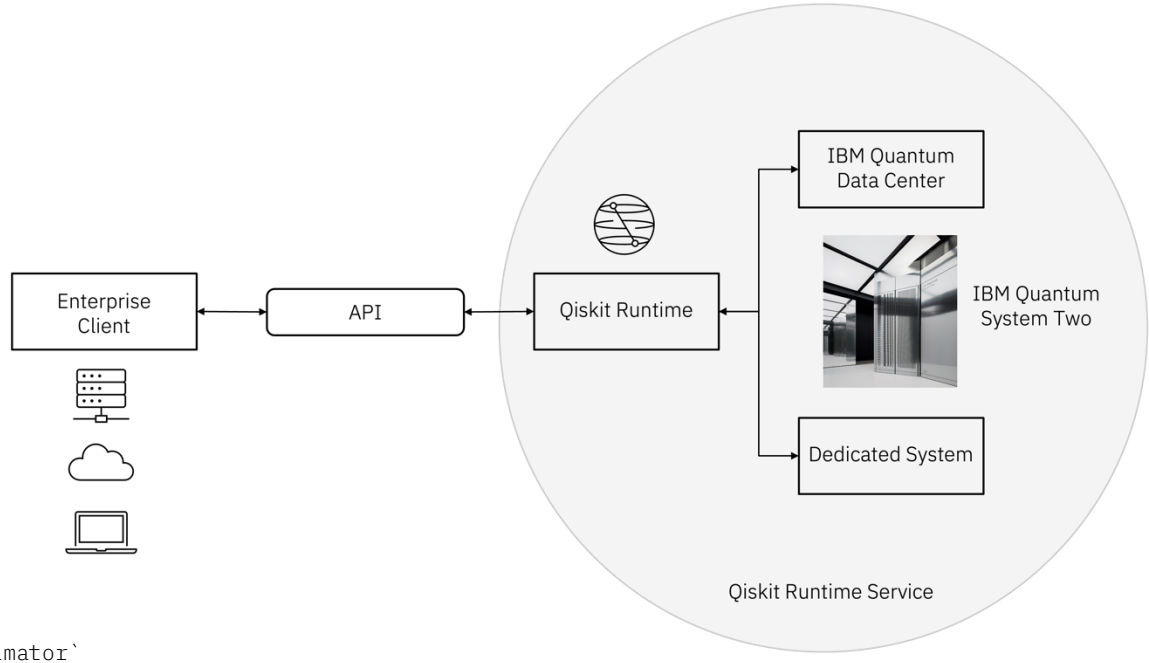
Quantum-centric supercomputing

for accelerated scientific discovery

Qiskit Runtime

Qiskit Runtime is a cloud-based service for executing quantum computations on IBM Quantum hardware that streamlines quantum computations and provides optimal implementations of Qiskit primitives.

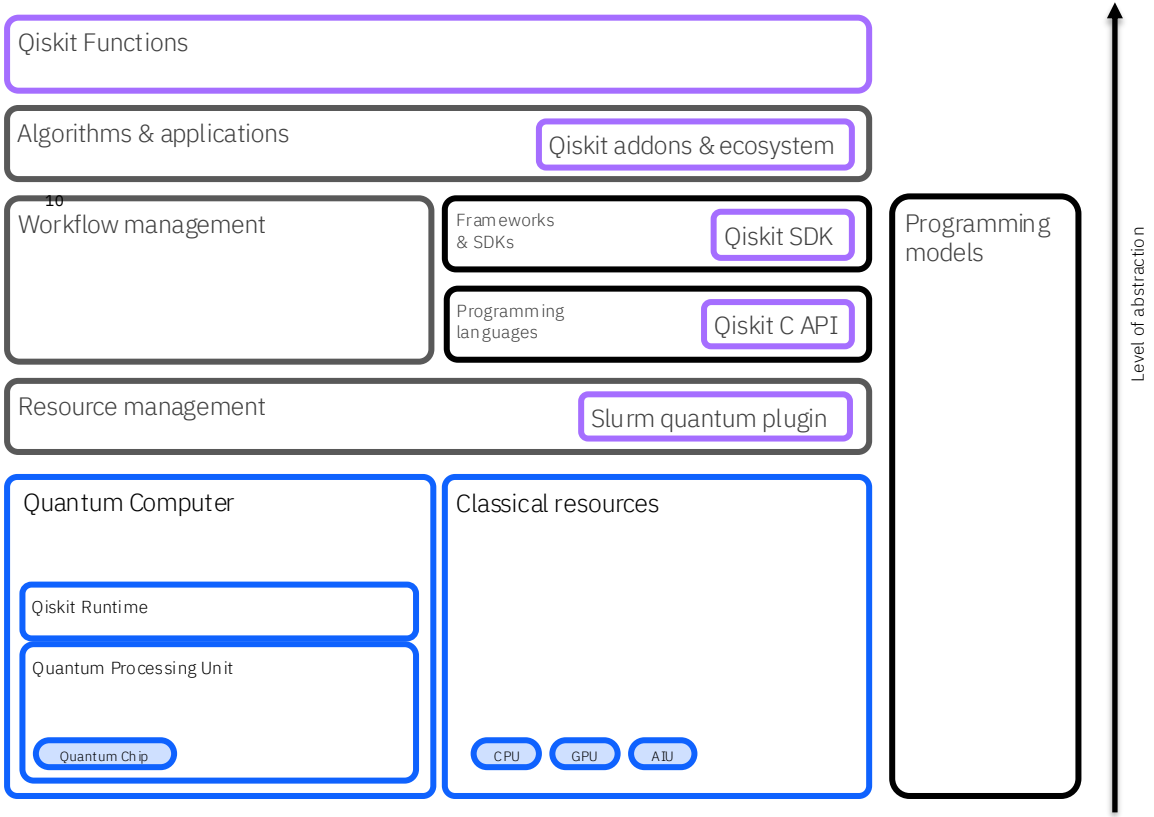
- Runs quantum programs on IBM Quantum hardware through IBM Quantum Platform or IBM Cloud
- Uses additional classical and quantum compute resources including error suppression and error mitigation techniques
- Offers three execution modes: Job, Session, and Batch for different use cases
- Simple installation via pip:
  `pip install qiskit-ibm-runtime`



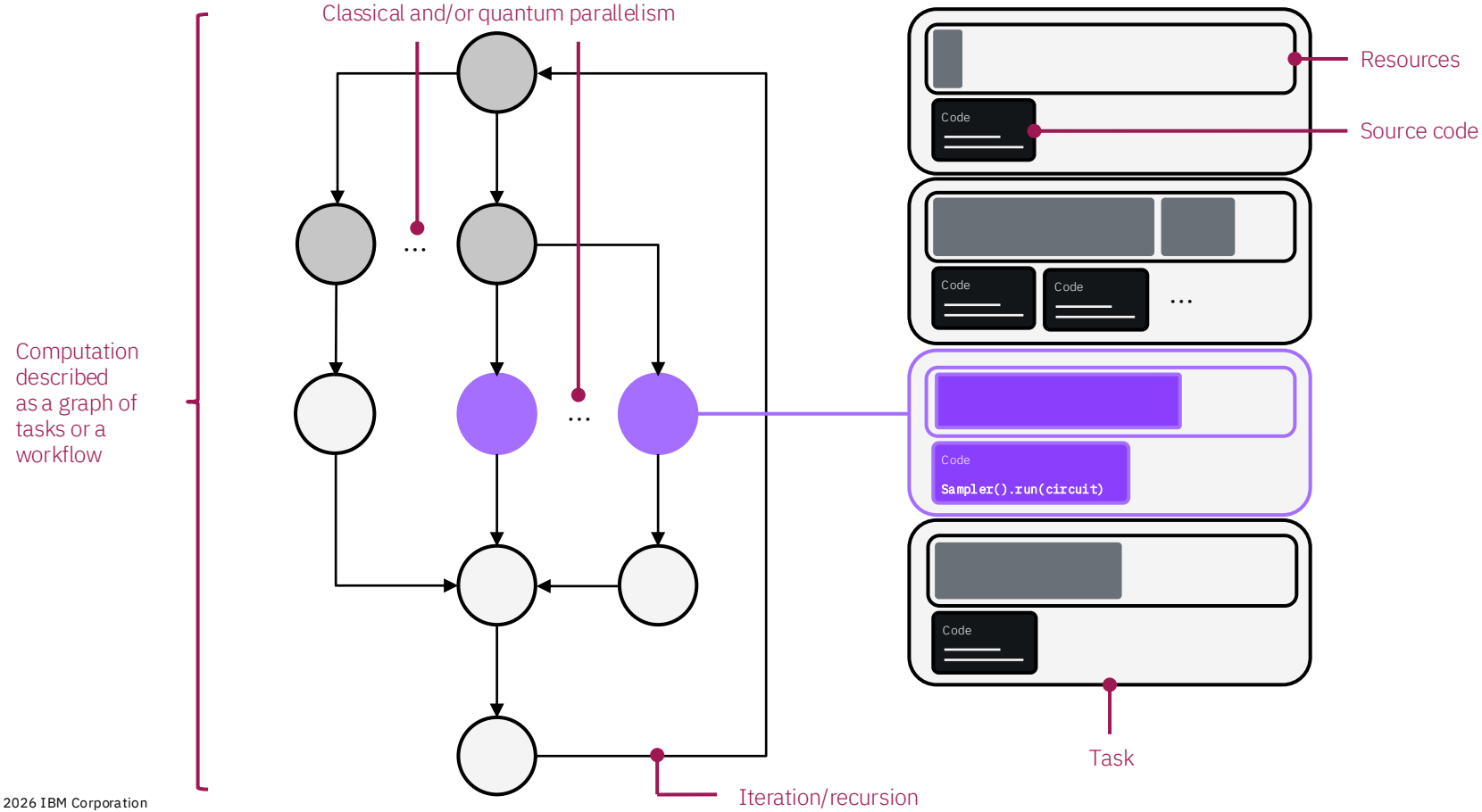
How to import runtime primitives:

```
from qiskit_ibm_runtime import EstimatorV2 as Estimator
from qiskit_ibm_runtime import SamplerV2 as Sampler
```

We are actively building toward a quantum-centric supercomputing future



Making quantum a native capability in HPC environments



IBM Quantum blog

Researchers use quantum-centric supercomputing to simulate 12,635-atom protein complex

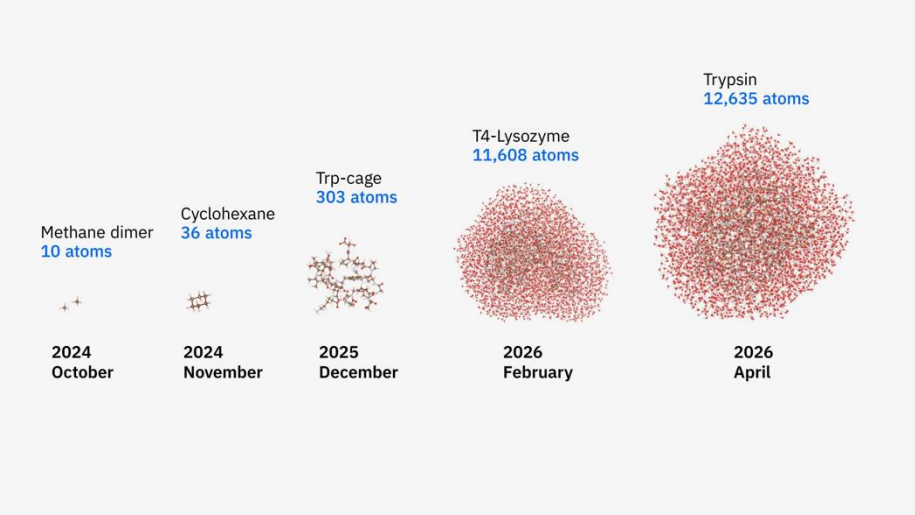
Date
5 May 2026

Authors
Rafi Letzter

Cleveland Clinic, RIKEN, IBM debut the largest heterogeneous quantum-classical (HQC) electronic-structure calculation to date, using up to 94 qubits.

Researchers used a quantum-centric supercomputing workflow to simulate protein–ligand chemistry, with the largest simulation reaching over 12,000 atoms (30,000 orbitals).

The work modeled the proteins T4-Lysozyme and Trypsin in solution with binding agents.



Researchers achieved a 40-fold increase in system size and 210x improvement in accuracy over previous results, driven by new algorithm developments and tight integration of QPUs, CPUs, and GPUs.

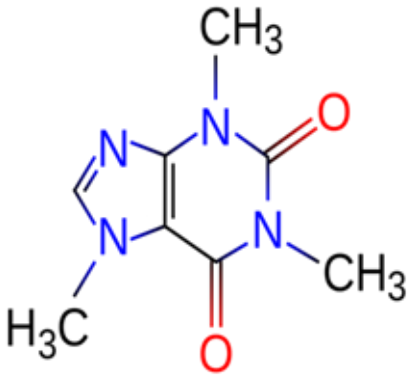
The result shows quantum computing is already useful for real chemistry problems.

https://ibm.biz/cleveland-clinic-riken-chemistry

Quantum Fundamentals

Computing with caffeine

If our best classical computers are so powerful, shouldn't we be able to perfectly simulate molecules and chemical reactions?
This would allow us to accelerate discovery of new compounds and processes for healthcare, materials, alloys, and sustainable energy creation.
Let's consider caffeine …

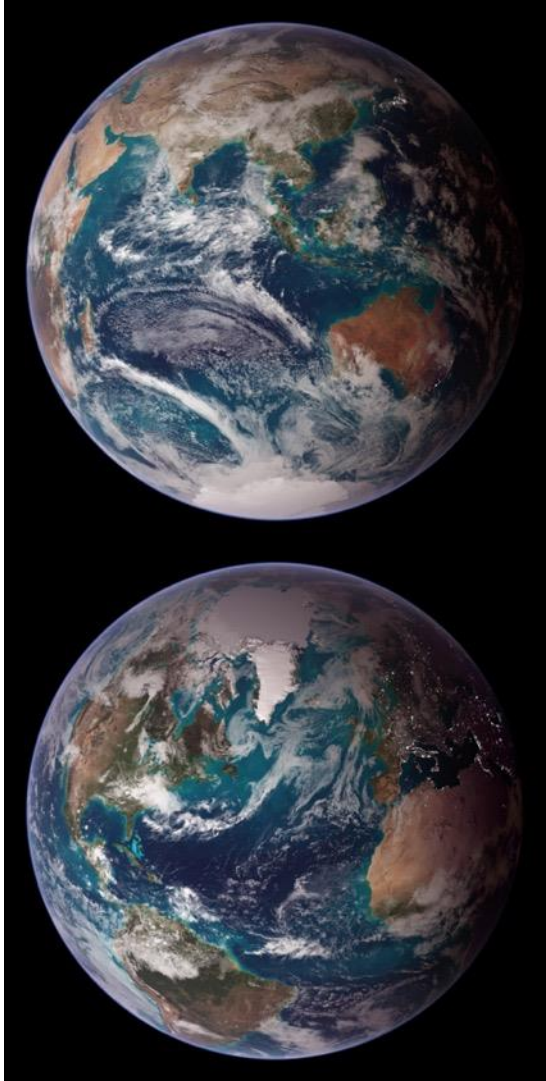


Computing with caffeine

We would need approximately 10^{48} bits to represent the energy configuration of a single caffeine molecule at a single instant in a classical computer.

This is 1 to 10% of the total number of atoms in the Earth.

10^{48} = 1,000,000,000,000,000,
000,000,000,000,000,000,
000,000,000,000,000



Computing with caffeine

Although it’s impossible to completely represent the molecular configuration of caffeine on today’s most powerful super computers, we could represent it using 160 logical qubits.

The limit of bits

For decades we've been simplifying nature into 1s and 0s because that was the only way we could manage to create a useful and scalable system of computation.

```
00100110111001001000100100100110010011
00101110011111001010010001110001000100
10000100100101010100101010111001001101
10010010001001001001100100111001011100
11110010100100011100010001001010001001
01010101001010101110111001110010101111
```

The limit of bits

For decades we've been simplifying nature into **1**s and **0**s because that was the only way we could **manage** to create a useful and scalable system of computation.

```
00100110111001001000100100100110010011
00101110011111001010010001110001000100
10000100100101010100101010111001001101
10010010001001001001100100111001011100
11110010100100011100010001001010001001
01010101001010101110111001110010101111
```

But the future isn't just **1**s and **0**s.

Bits and classical logic circuits

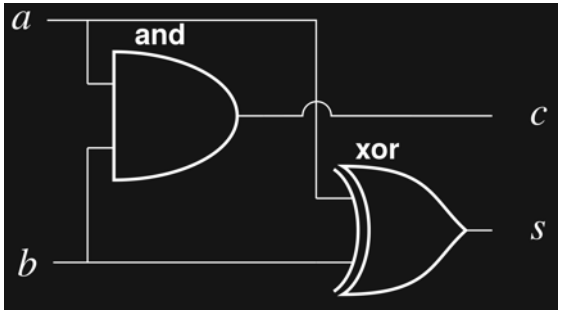
0

•

•

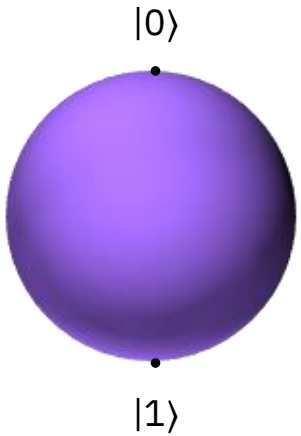
1

A bit is a controllable classical object that is the unit of information

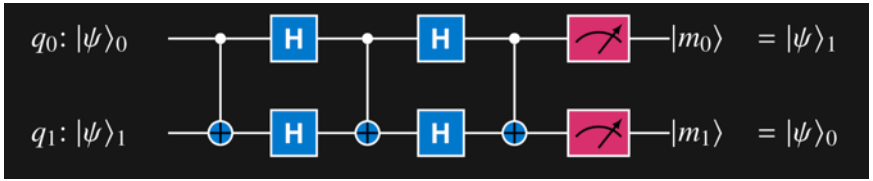


A classical logic circuit is a set of gate operations on bits and is the unit of computation

Quantum bits (qubits) and quantum circuits

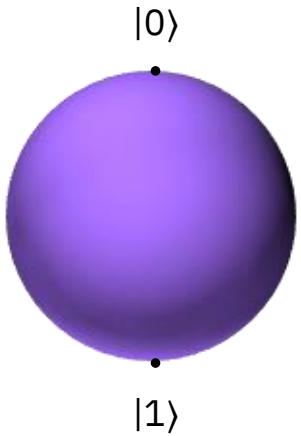


A quantum bit or qubit is a controllable quantum object that is the unit of information



A quantum circuit is a set of quantum gate operations on qubits and is the unit of computation

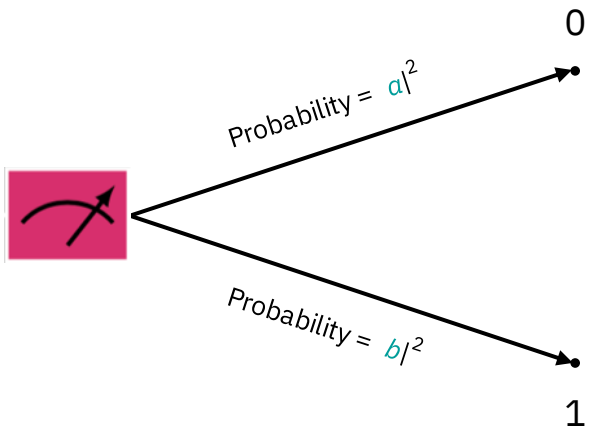
Bits and qubits



A qubit's state is a combination of |0⟩ and |1⟩:

$a\,|0\rangle + b\,|1\rangle$

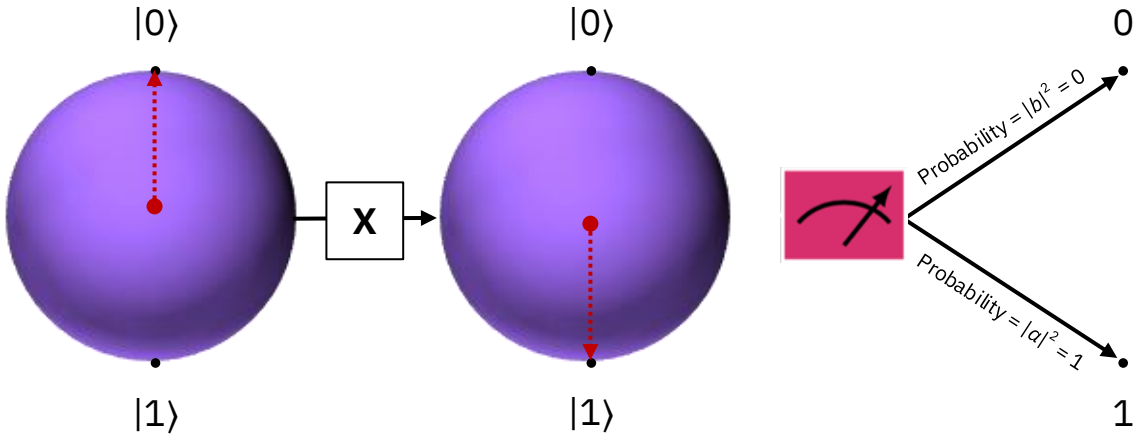
This means that a single qubit contains two pieces of information.



When we measure a qubit, it becomes **0** or **1** based on probability.

Bits and qubits: the effect of the **X** gate on $|0\rangle$

IBM **Quantum**



The **X** gate reverses $|0\rangle$ and $|1\rangle$:

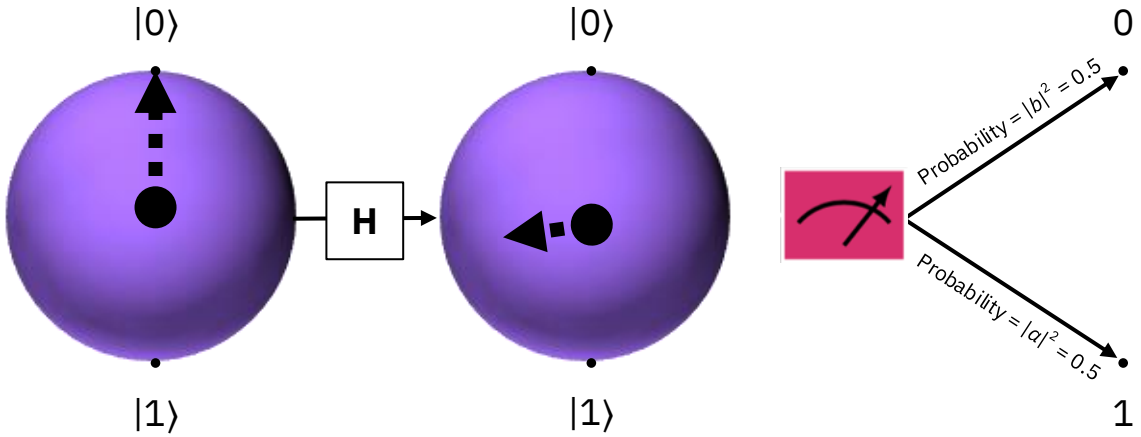
$a\,|0\rangle + b\,|1\rangle \;\mapsto\; b\,|0\rangle + a\,|1\rangle$

$a = 1$ and $b = 0$, so $|0\rangle$ is mapped to $|1\rangle$.

When measured, the result is **1** with 100% probability.

Bits and qubits: the effect of the **H** gate on |0⟩

IBM **Quantum**



The **H** gate maps |0⟩ via

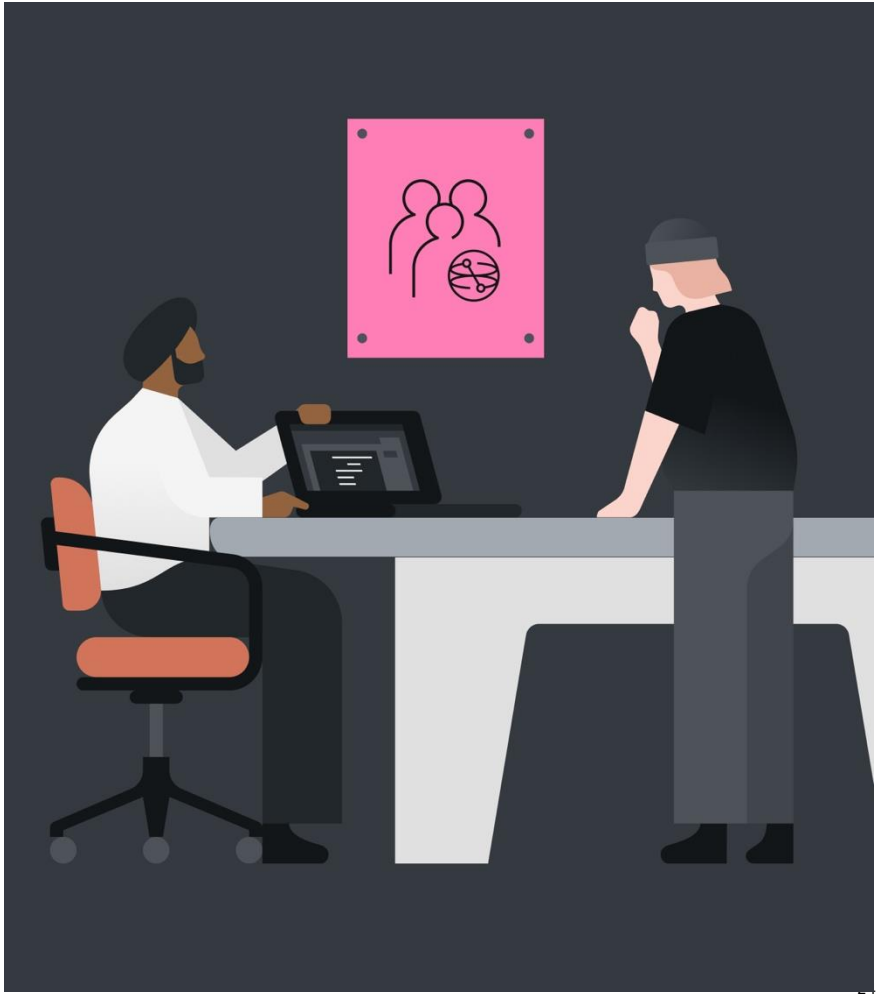
$|0\rangle \mapsto (1/\sqrt{2})\,|0\rangle + (1/\sqrt{2})\,|1\rangle = a\,|0\rangle + b\,|1\rangle$

Since $a = b = 1/\sqrt{2}$, $|a|^2 = |b|^2 = ½$.

When measured, the probability of getting **0** or **1** is the same, 0.5.

Quantum randomness!

Qiskit

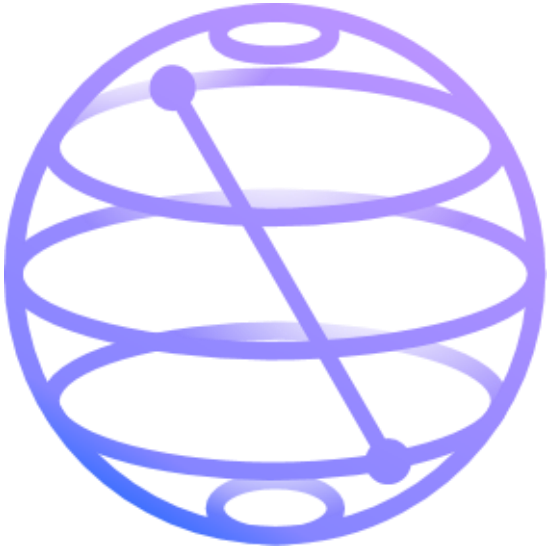


What is Qiskit?

The name "Qiskit" is a general term referring to a collection of software for executing programs on quantum computers.

Qiskit SDK is an open-source SDK for working with quantum computers at the level of extended quantum circuits, operators, and primitives.

github.com/Qiskit/qiskit
quantum.cloud.ibm.com/docs



The **Qiskit Ecosystem** is a collection of software and projects that build on or extend Qiskit.

qiskit.github.io/ecosystem

Qiskit 2.0

Released April 2025

Qiskit 2.0 is the first release to introduce a C API beyond the usual Python interface, marking the beginning of multi-language support. This sets the foundation for accessing Qiskit from C directly and building bindings for other programming languages on top of C interface.

Two modes of operation:

- Standalone C usage (no Python runtime dependency) - Installation Guide
- Embedded extension, for calling the C API from a Python context Python - Extension Guide

Currently, the C interface is minimal but growing: C API Reference

Other Qiskit 2.0 highlights:

- Rust-powered performance boosts: Up to 2× faster circuit construction and ~20% faster transpilation.
- Support for abstract/deferred timing in dynamic circuits: Enables use of stretches for complex timing logic.
- New 'Box' support: Atomic blocks for the transpiler; useful for twirling, noise learning, and more.

Quantum Resource Management Interface (QRMI) an open standard for integrating quantum resources

Thin and vendor-agnostic layer to access, control, and monitor underlying on-prem or cloud quantum systems

Contributors:

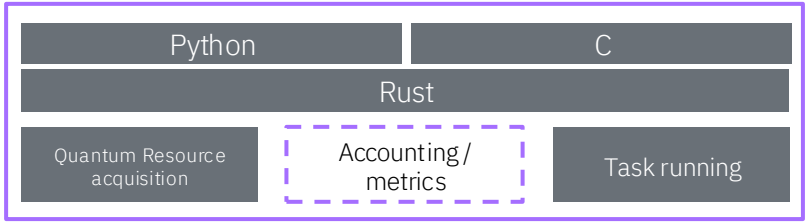
- IBM Quantum
- Rensselaer Polytechnic Institute
- Pasqal
- STFC Hartree Centre
- Cleveland Clinic
- Oak Ridge National Laboratory
- Jülich Supercomputing Centre
- Poznań Supercomputing and Networking Center
- ParTec

HPC Plugins

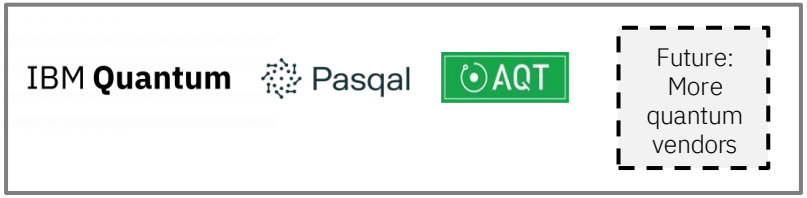
slurm workload manager

PREFECT

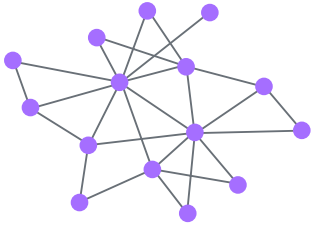
QRMI https://github.com/qiskit-community/qrmi



Hardware Support



Qiskit Pattern:
The anatomy of a quantum algorithm



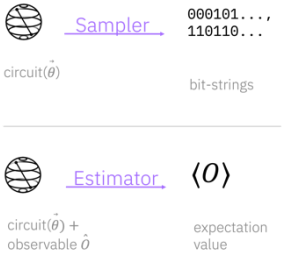
01

Map problem instance to quantum circuits and operators

```
PassManager([UnitarySynthesis(),
             BasisTranslator(),
             EnlargeWithAncilla(),
             AISwap(),
             Collect1qRuns(),
             Optimize1qGates(),
             Collect2qBlocks(),
             ConsolidateBlocks()])
```

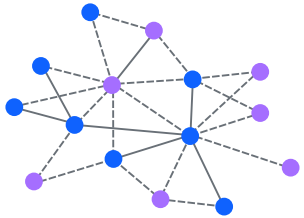
02

Optimize for target hardware execution



03

Execute via Qiskit Runtime



04

Result processing

Map

Optimize

Execute

Post-Process

Qiskit Installation Instructions

`To Install Qiskit:`
You can use an online jupyter lab environment (quantum.cloud.ibm.com/docs/en/guides/online-lab-environments) or install Qiskit locally.

Follow the Qiskit installation guide (docs.quantum.ibm.com/guides/install-qiskit) to complete the following steps:

1. Install python

2. Install Qiskit

- Create a virtual environment and activate it
- Install qiskit including the visualization extra packages : `` `pip install qiskit[visualization]` ``
- Install qiskit-ibm-runtime: `` `pip install qiskit-ibm-runtime` ``
- Install jupyter: `` `pip install jupyter` ``

`To Access Hardware:`

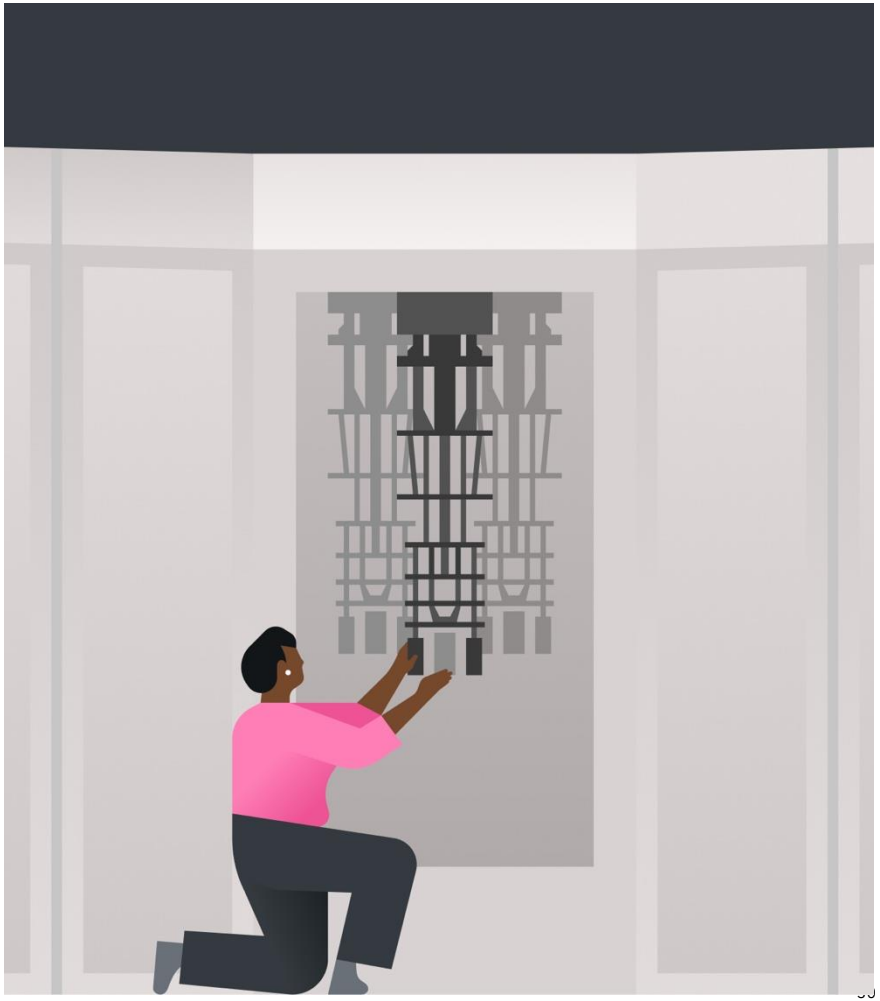
Follow this guide (docs.quantum.ibm.com/guides/setup-channel) to setup an IBM Quantum account

1. Register IBM Quantum account using your institutional email

2. Save token to your local computer

qisk.it/install

Demo: end-to-end hybrid quantum-classical execution pipeline



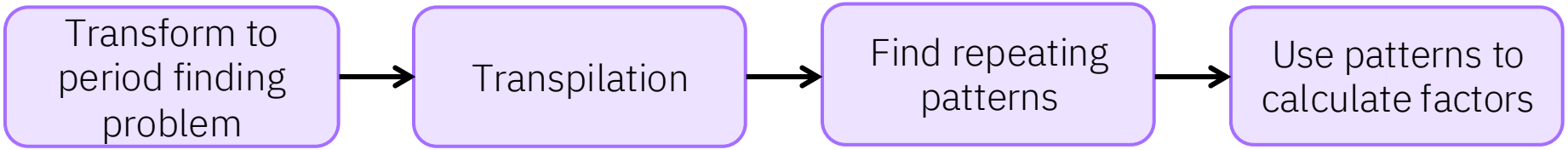
Today’s classical security protocols will be obsolete tomorrow

Prime factors	2048-bit composite integer	Expected computation time
= p x q	25195908475657893494027183240048398571429282126204032 02777713783604366202070759555626401852588078440691829 06412495150821892985591491761845028084891200728449926 87392807287776735971418347270261896375014971824691165 07761337985909570009733045974880842840179742910064245 86918171951187461215151726546322822168699875491824224 33637259085141865462043576798423387184774447920739934 23658482382428119816381501067481045166037730605620161 96762561338441436038339044149526344321901146575444541 78424020924616515723350778707749817125772467962926386 35637328991215483143816789988504044536402352738195137 86365643921201039712282212072035 7	The most powerful computer today: **Millions of years** Shor’s quantum algorithm: **Hours**

Public key encryption • Digital signatures • Key exchange algorithms

RSA • DSA • ECC • ECDSA • DH

Qiskit Pattern:
The anatomy of a quantum algorithm



01
Map problem instance to quantum circuits and operators

Map

02
Optimize for target hardware execution

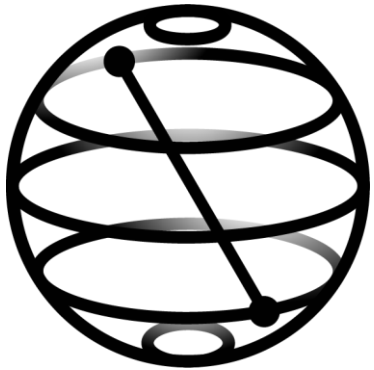
Optimize

03
Execute via Qiskit Runtime

Execute

04
Result processing

Post-Process



THE LINUX FOUNDATION

OPEN SOURCE SUMMIT INDIA

Order-finding by phase-estimation

To connect the order-finding problem to phase estimation, consider a system whose classical state set is $\mathbb{Z}_N$.

For a given element $a \in \mathbb{Z}_N^*$, define an operation as follows:

$$M_a|x\rangle = |ax\rangle \qquad \text{(for each } x \in \mathbb{Z}_N\text{)}$$

This is a *unitary operation* — but only because $\gcd(a, N) = 1$!

Example

Let $N = 15$ and $a = 2$. The operation M_a has this action:

$$M_2|0\rangle = |0\rangle \qquad M_2|5\rangle = |10\rangle \qquad M_2|10\rangle = |5\rangle$$
$$M_2|1\rangle = |2\rangle \qquad M_2|6\rangle = |12\rangle \qquad M_2|11\rangle = |7\rangle$$
$$M_2|2\rangle = |4\rangle \qquad M_2|7\rangle = |14\rangle \qquad M_2|12\rangle = |9\rangle$$
$$M_2|3\rangle = |6\rangle \qquad M_2|8\rangle = |1\rangle \qquad M_2|13\rangle = |11\rangle$$
$$M_2|4\rangle = |8\rangle \qquad M_2|9\rangle = |3\rangle \qquad M_2|14\rangle = |13\rangle$$