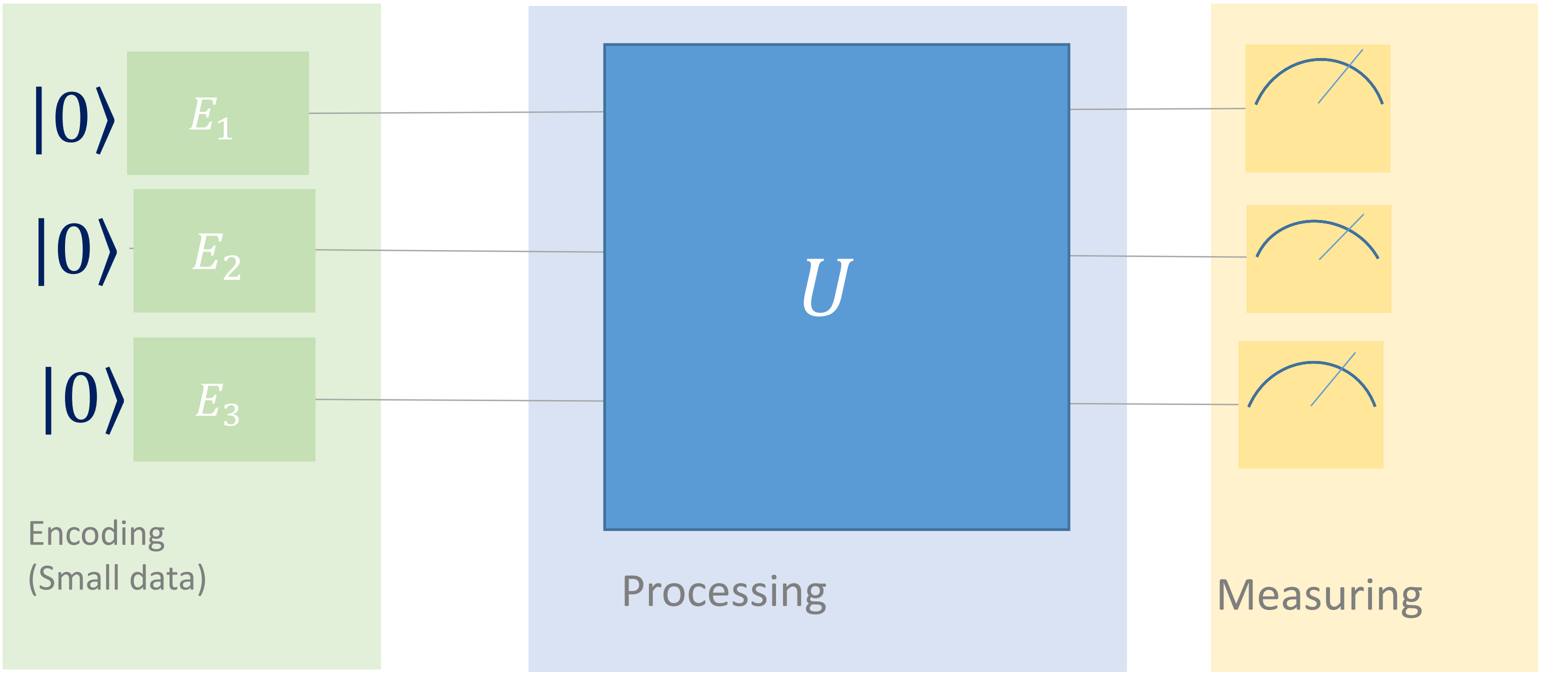
From the need to hybridize algorithmically to the need to integrate QPUs with CPUs

J. Mikael, EDF, E. Vergnaud, Teratec TQCI
Conference on QPU/CPU Integration

The first generation of Quantum Algorithms

QPU



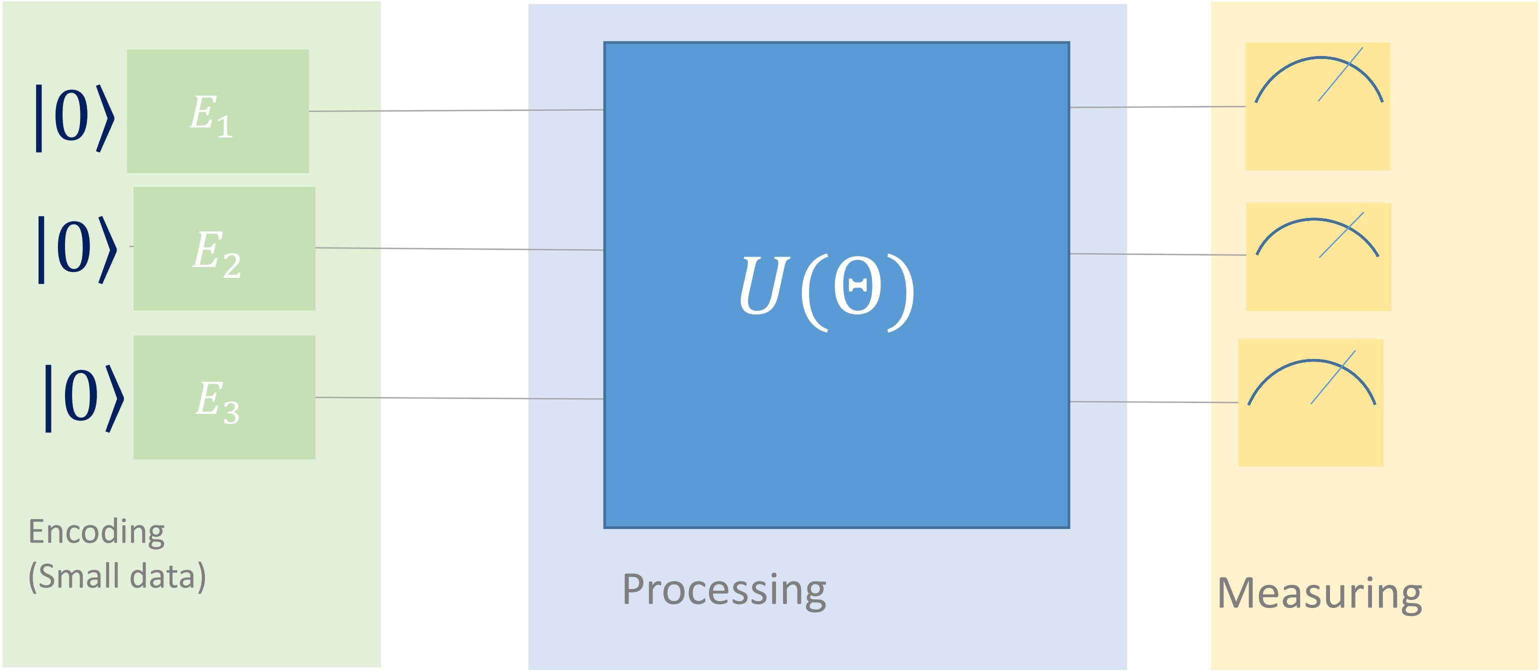
- Shor Factoring algorithm

-Grover (Unstructured DB Search)

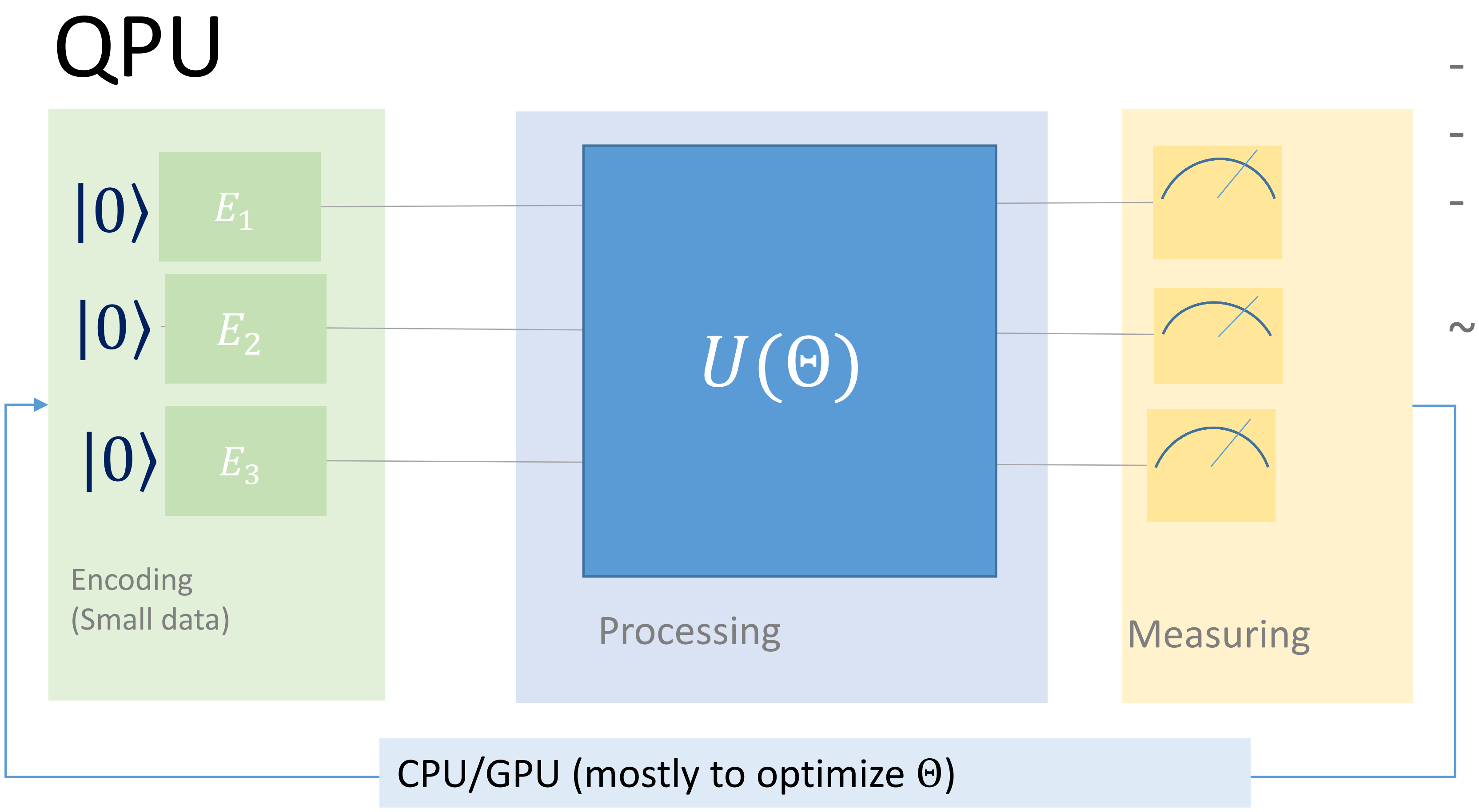
- HHL (Linear system of equations)
& Phase estimation

The first generation of Quantum Algorithms + Parameterization

QPU



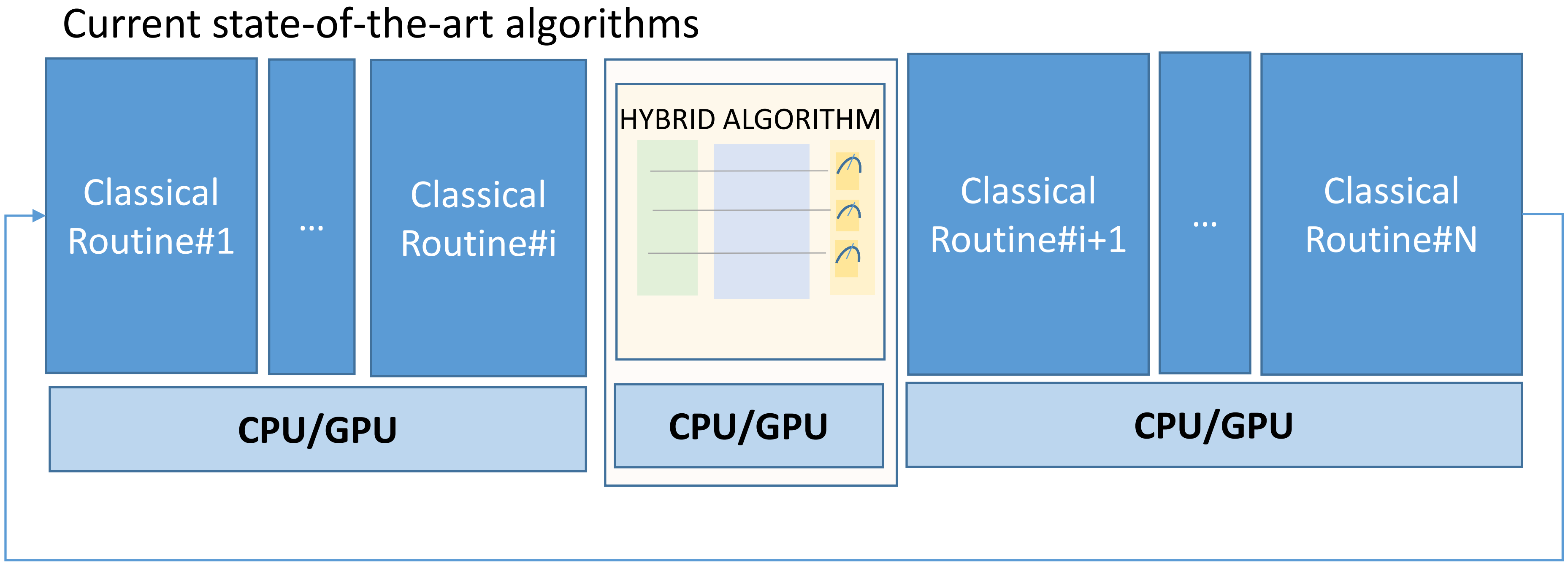
The second generation of Quantum Algorithms



- QAOA (Optimization)
- VQLS (Linear systems)
- PQC (involved in QML)

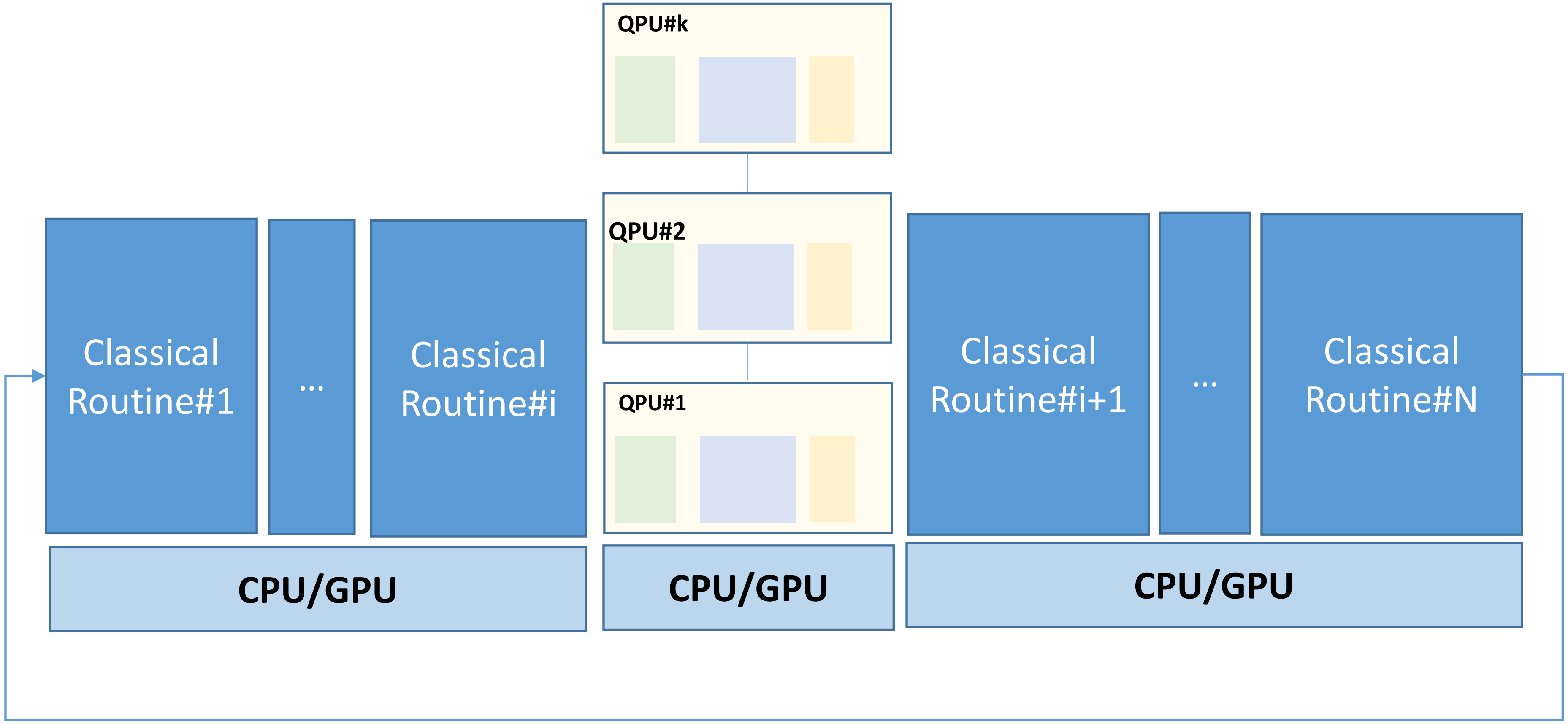
~ most of NISQ algorithms

The algorithms under study today



Branch and Price with RQAOA sub routine
Generative Adversarial Network with quantum update
Classical algorithm with Grover search
Topological Data Analysis

A possible future



Open Questions related to hybridation

How to build an architecture hybrid algorithms that maximizes the use of the expensive QPU?

How a clever integration can

- help producing more efficient algorithms?
- lead to energy consumption advantage?
- help reducing the NISQ uncertainty ?

What to put together with QPU? CPU? GPU?

How to link together QPUs?

Today’s Program / Morning

09:30 10:50 Session 1. Architecture, Code development, compilation and runtime - Chairman Cyril. Baudry :

- Quantum Computing in the Cloud with Amazon Braket : Sebastian Stern, Specialist Solution Architect Quantum Computing, AWS
- Large scale quantum/HPC hybridization with Atos QLM : Arnaud Gazda, Expert Quantum R&D Engineer, ATOS
- Quantum + Classical computation: Quantum Middleware : Ismael Faro, Distinguished Engineer, Chief Architect Quantum Computing Cloud and Software, IBM

10:50 11:20 Coffee Break

11:20 11:50 The impact of latency reduction on quantum algorithms, Vivien Londe, Microsoft

11:50 12:50 Presentation and objectives of the Quantum Energy Initiative – Focus on Hybridation and the case-study of cat-qubits (A. Auffèves from CNRS MajuLab and the QEI, O. Ezratty from the QEI and T. Péronnin from Alice&Bob)

#QEI
the quantum energy initiative

ALICE & BOB

Teratec Quantum Computing Initiatiue

Today's Program / Afternoon

13:45 16:45 Session2. On the hardware side; integration of CPU/QPU - Current experiments/Next Scientific and technical obstacles - Chairman Alain Refloch

- Bringing GPU acceleration to Quantum-Classical Computing : François Courteille, Principal Solutions Architect, NVidia
- Quantum interconnects to scale-up quantum technologies : Tom Darras, CEO WeLinq
- A quantum pricing-based column generation framework for hard combinatorial problems, Wesley da Silva Coelho, Louis-Paul Henry Quantum Application Engineer, Loïc Henriet, CTO, Pasqal
- Single-Photon based Quantum Computers available in the cloud : Metrics and Benchmarks" Shane Mansfield, Chief Research Officer and Jean Senellart, Chief Product Officer, Quandela
- Primitives and circuit optimization, Blake Johnson, Quantum Platform Lead, IBM
- An HPC & QC integrated platform : Jacques-Charles Lafoucrière, Head of HQI program, CEA
- Building the first of its kind hybrid quantum/classical hosting environment, Richard St Pierre, directeur général de la Zone d'Innovation Quantique de Sherbrooke, Laurent Bernou-Mazars, CTO et co-fondateur d'Exaion

16:45 17h:00 Coffee Break

Teratec Quantum Computing Initiatiue

Today’s Program / Afternoon

17:00 17h30 The impact of compilation in the implementation of quantum computing - Simon Martiel, Quantum Computing Researcher, Atos

17:30 18:00 Integrating High-Performance Computing with Quantum Computing - Scott Pakin, Scientist, Los Alamos National Laboratory
18:00 18:15 Conclusion (E. Décossin)

18h15 - 19h:00: Cocktail

Atos

Los Alamos NATIONAL LABORATORY EST. 1943

eDF

eDF Teratec Quantum Computing Initiative

Quantum Computing in the Cloud with Amazon Braket

Sebastian Stern
Quantum Specialist Solutions Architect
Amazon Web Services

Quantum Computing at AWS

AWS Center for Quantum Computing

Research and development

AWS Center for Quantum Networking

Research and development

AWS Partner Network

Community of quantum computing partners

Quantum Solutions Lab

State-of-the-art quantum and classical solutions

Amazon Braket

Fully managed quantum computing service

Amazon Braket – the AWS Quantum Computing Service

A fully managed service that makes it easy for scientists and developers to explore quantum computing

Build

- Amazon Braket SDK
- Jupyter notebooks
- Command line interface

Test

- Local simulators for rapid testing
- High-performance simulators

Run

- Access multiple quantum computers
- Combine quantum and classical resources

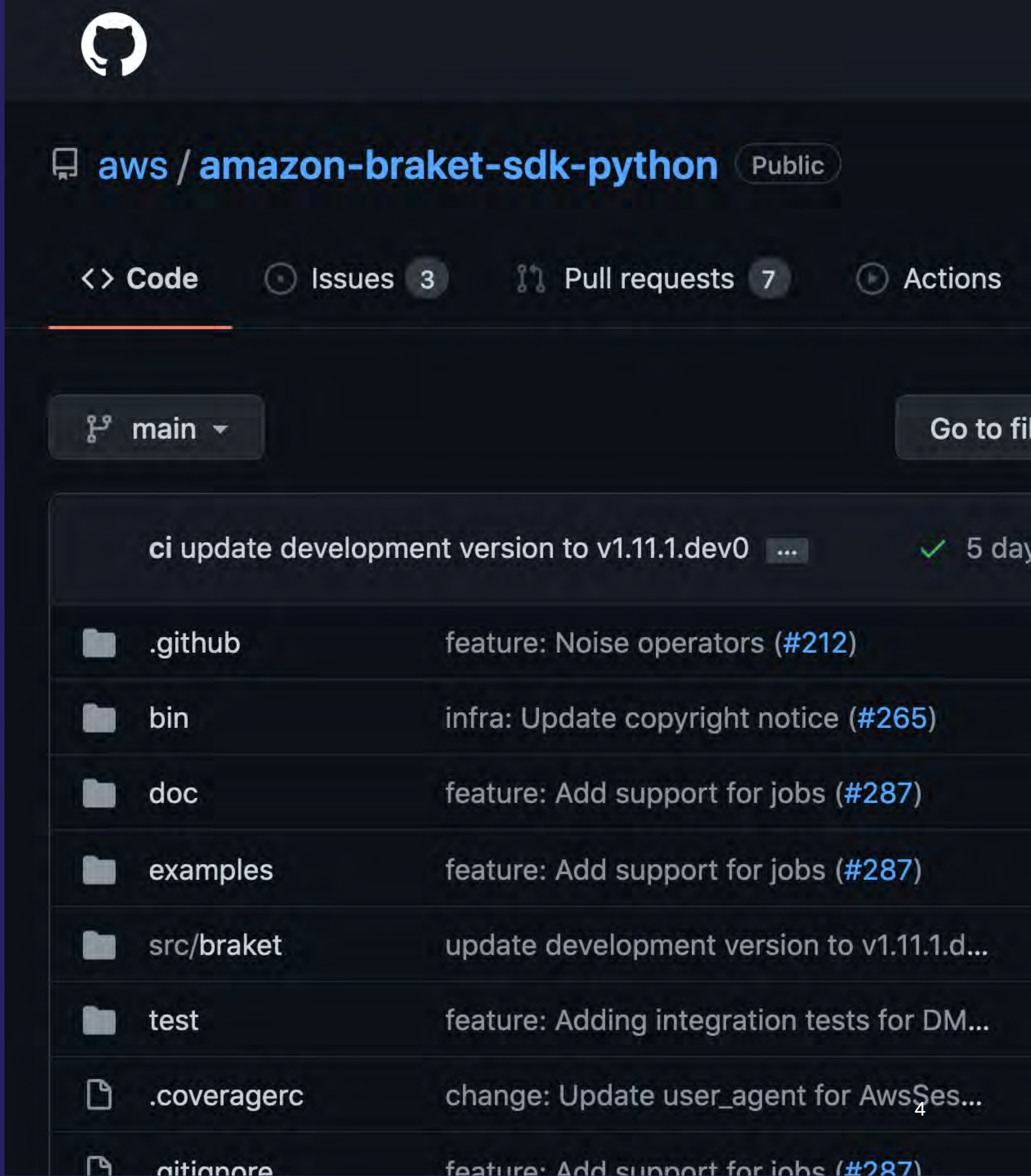
Analyze

- Monitor algorithms in almost real time
- Analyze algorithm results and performance

The Amazon Braket SDK

Open source Python library

- Design and build quantum circuits
- Submit tasks and jobs to devices
- Track and monitor their execution
- Local installation
  - pip install amazon-braket-sdk
- Local simulators (free)
  - braket_sv
  - braket_dm
  - braket_ahs



The Amazon Braket Workflow

aws_bell.py

```python
from braket.aws import AwsDevice
from braket.circuits import Circuit

device = AwsDevice("aws_device_ARN")

# Choose S3 bucket to store results
bucket = "amazon-braket-unique-aabbcdd"
prefix = "results"
s3_folder = (bucket, prefix)

bell = Circuit().h(0).cnot(0, 1)
print(bell)

task = device.run(bell, s3_folder, shots=1000)
print("Measurement Results")
print(task.result().measurement_counts)

```

Amazon Braket provides AWS customers access to multiple types of quantum computing technologies.

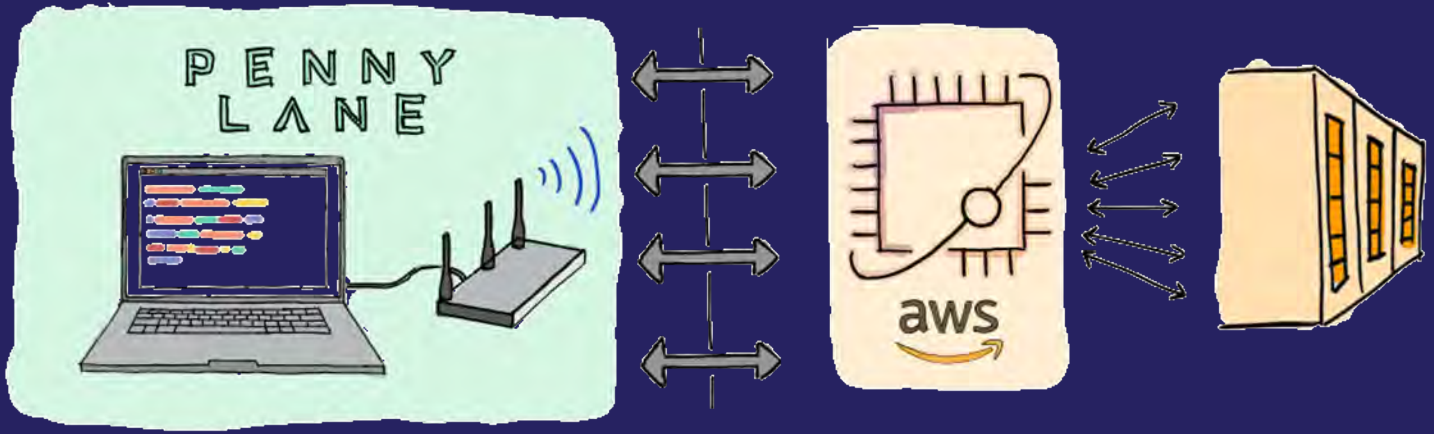
In Amazon Braket, a device represents a QPU or simulator.

Devices are selected using the device Amazon Resource Name (ARN).

https://docs.aws.amazon.com/braket/latest/developerguide/braket-devices.html

The Amazon Braket SDK

Enhance functionality with open-source plugins



A cross-platform Python library for differentiable programming of quantum computers.

Train a quantum computer the same way as a neural network.

https://pennylane.ai/qml

The Amazon Braket SDK

Enhance functionality with open-source plugins

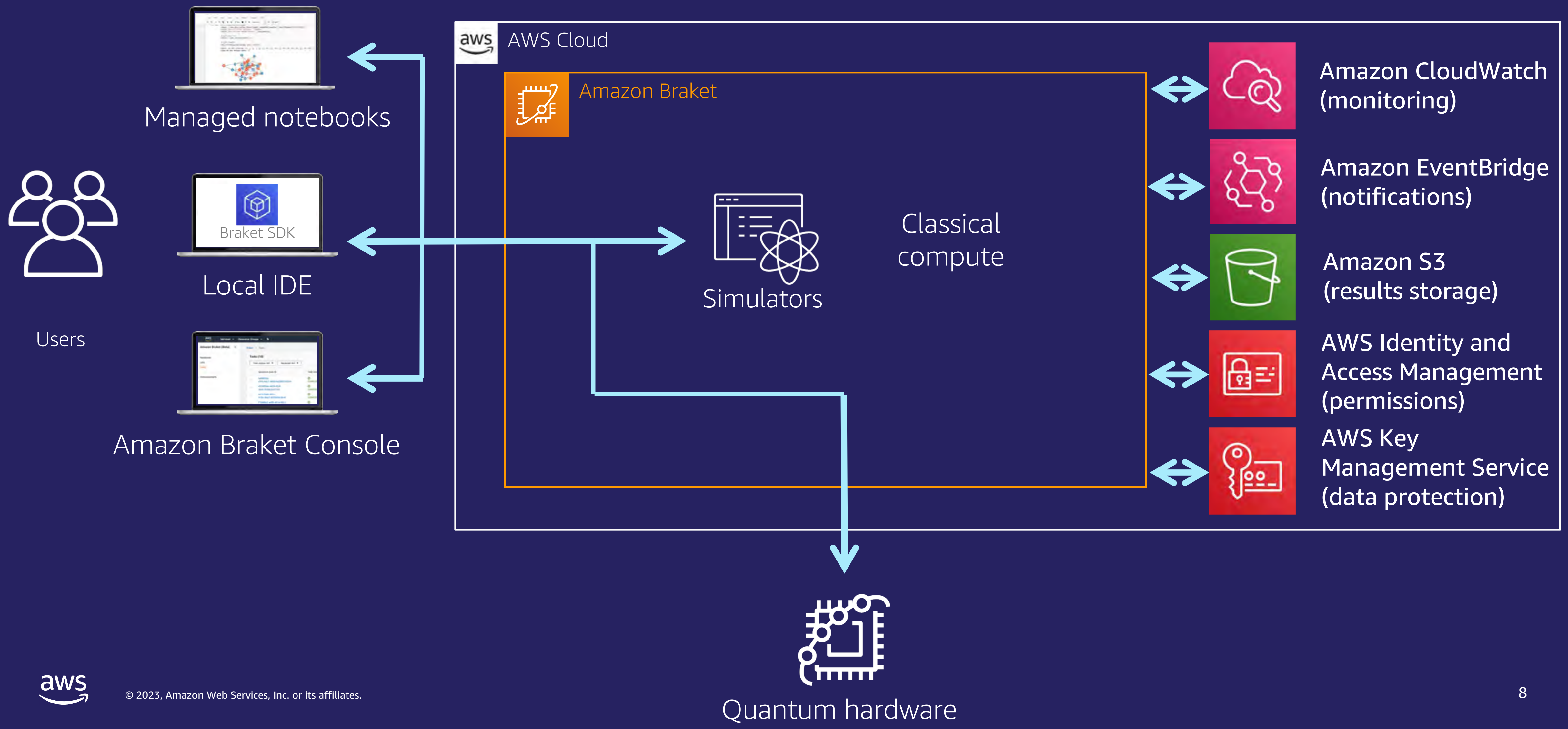
Run Amazon Braket code from Qiskit development toolkit

https://github.com/qiskit-community/qiskit-braket-provider

qiskit_connector.py

```python
from qiskit_braket_provider import BraketLocalBackend

local_simulator = BraketLocalBackend()
task = local_simulator.run(circuit, shots=1000)

```

The Amazon Braket Architecture



Local and On-Demand Simulators

Local simulator

Part of Braket SDK

Fast and convenient prototyping

Number of qubits based on hardware

SV1: State Vector simulator

Quantum circuit with up to 34 qubits

Stores the full wave function state

Concurrency: Default 35, max 50

TN1: Tensor Network simulator

Quantum circuit with up to 50 qubits

Encodes quantum circuits into a structured graph

Concurrency: Default 10, max 10

DM1: Density Matrix simulator

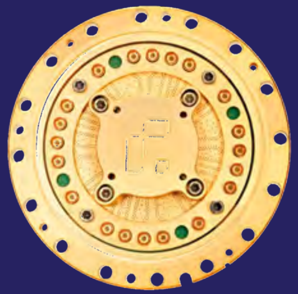
Quantum circuit with up to 17 qubits

Run multiple circuits in parallel with noise simulation

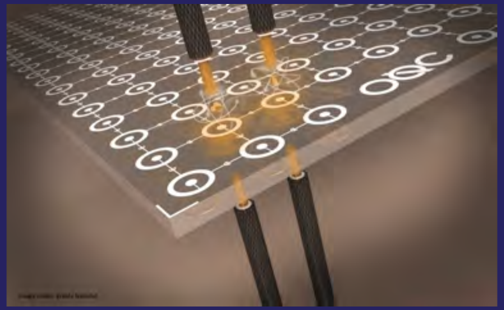
Concurrency: Default 35, max 50

Available Quantum Computers

IONQ

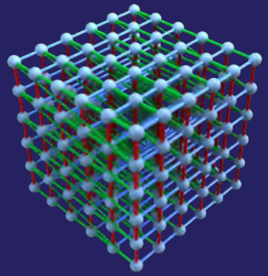


rigetti



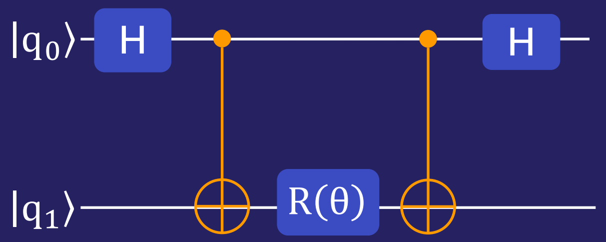
OQC

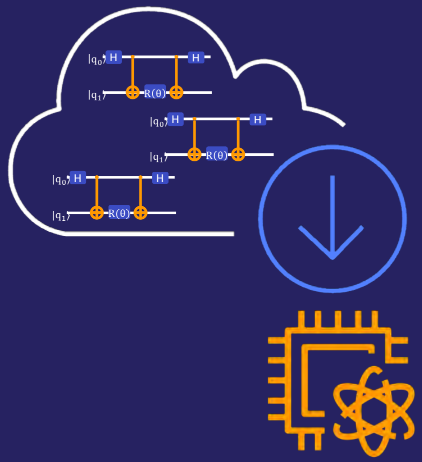
|QuEra>
COMPUTING INC.

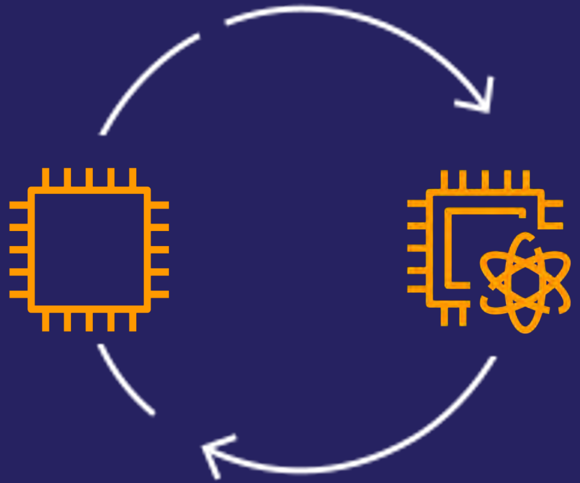


XANADU

Shots, Tasks, and now Hybrid Jobs







Shot

Single execution of quantum operation on a device

Task

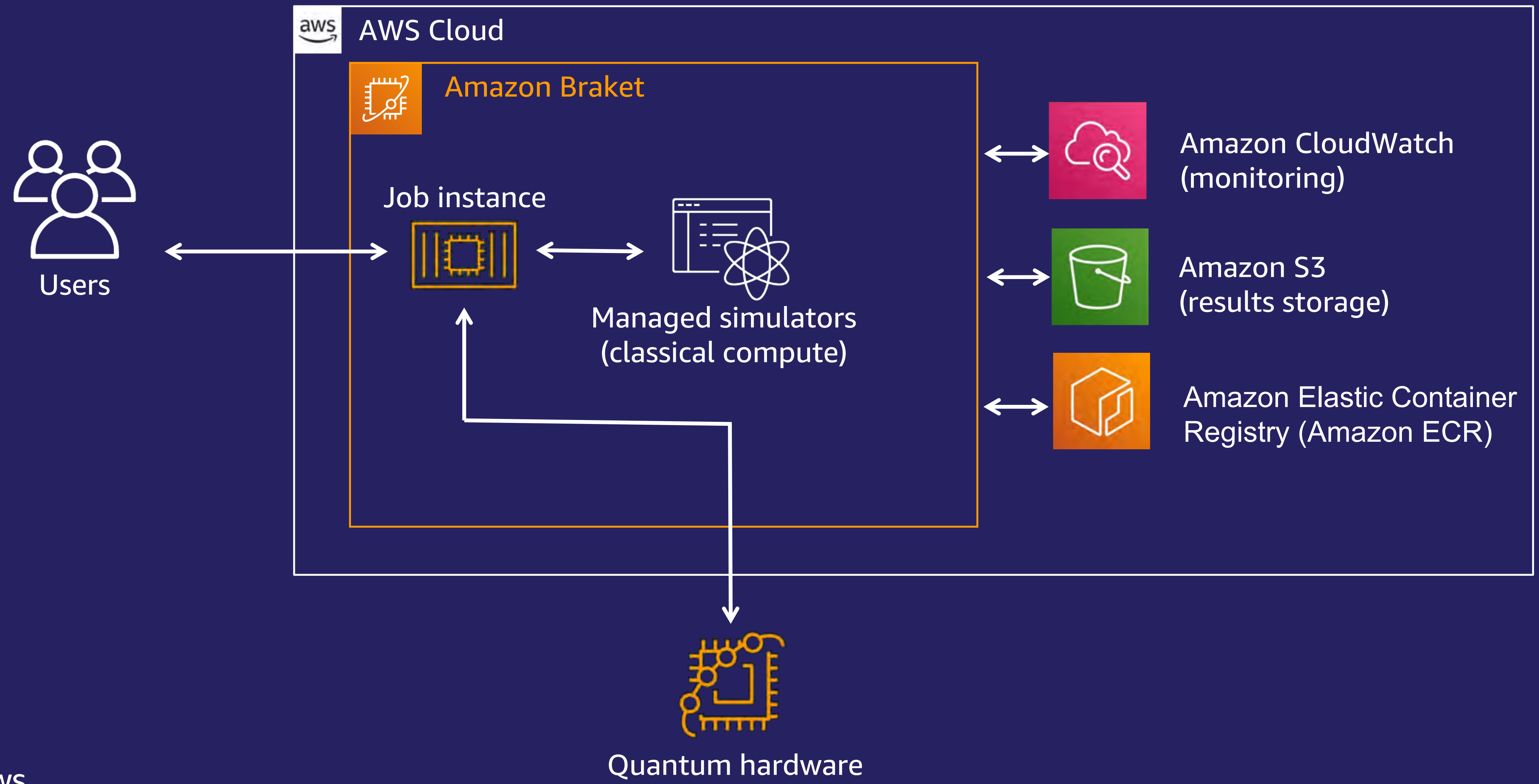
Series of repeated shots on a device (10s–10,000s shots per task)

Hybrid job

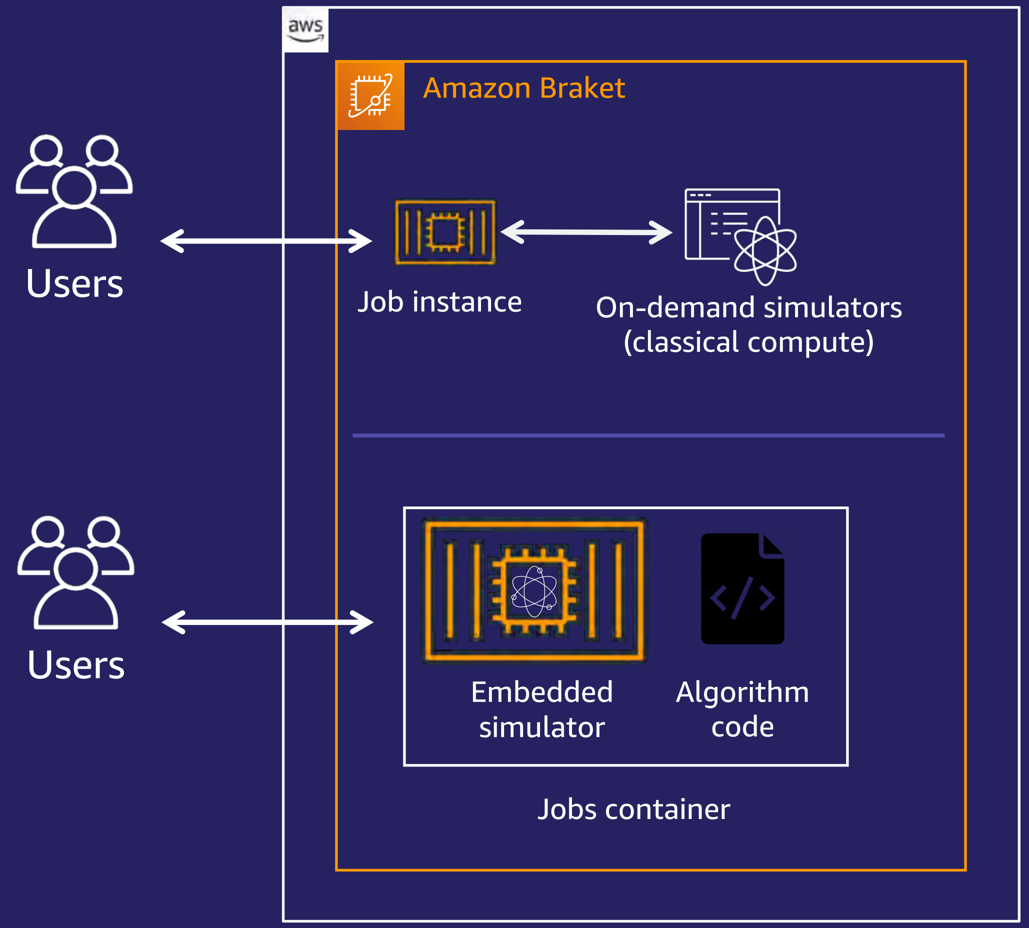
Sequence of classical and quantum compute cycles (10s to 1,000s of tasks per job)

https://docs.aws.amazon.com/braket/latest/developerguide/braket-jobs.html

Amazon Braket Hybrid Jobs



Embedded Simulators with Hybrid Jobs



Speed up demanding algorithms

- Bring code and simulator together in a single container for faster communication and distributed scale-out.
- NVIDIA cuQuantum state vector simulation library for highly entangled quantum circuits.
- Reduce number of iterations and lower memory usage, decreasing costs.
- lightning.qubit, lightning.gpu or BYOC

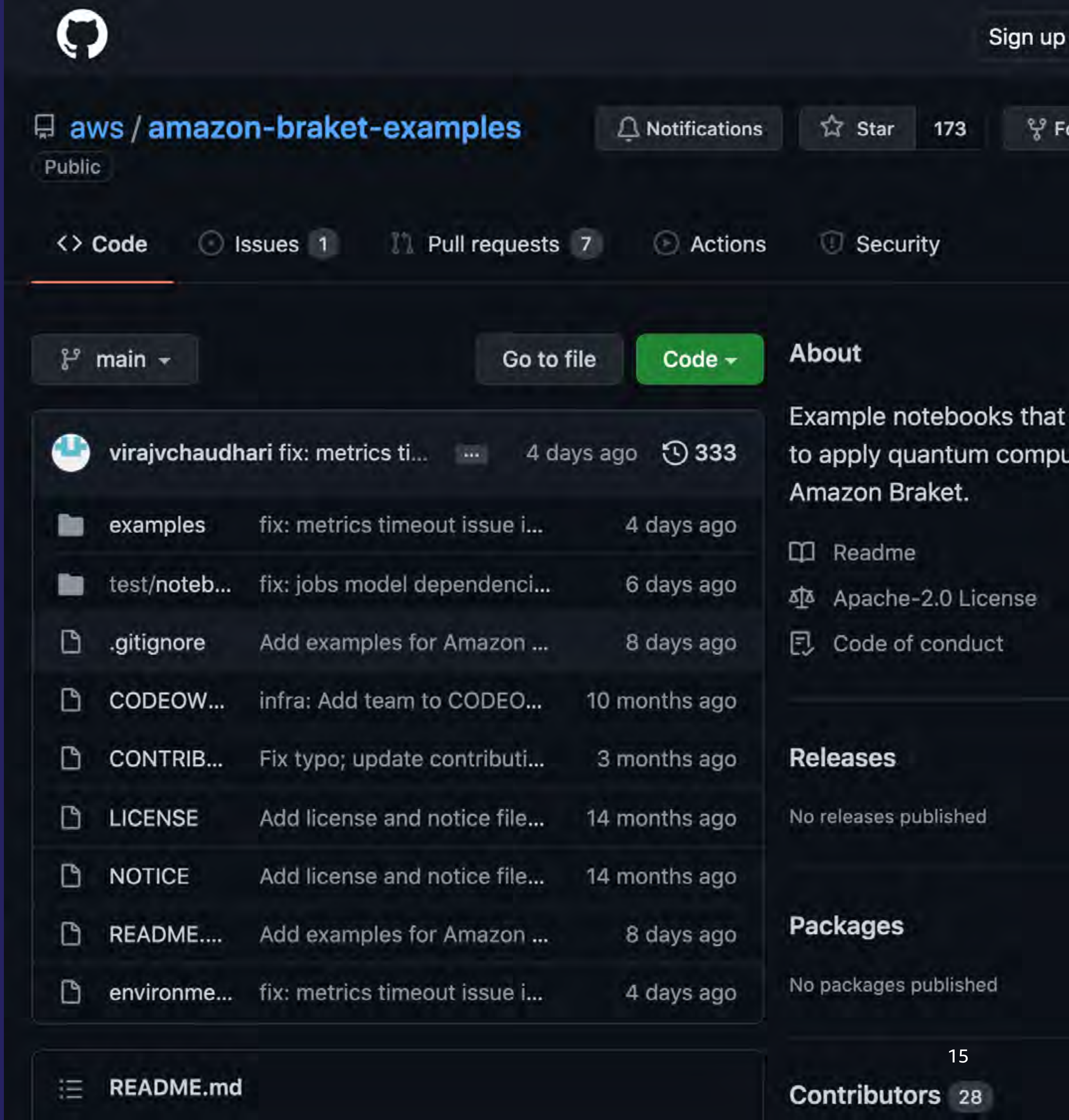
How to get started?

Amazon Braket Examples

Topics include:

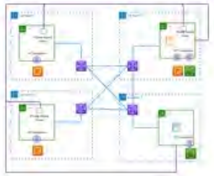
- Getting started
- Braket features
- Hybrid jobs
- Advanced circuits
- Hybrid algorithms
- PennyLane

https://github.com/aws/amazon-braket-examples



AWS Blogs

AWS Quantum Computing Blog

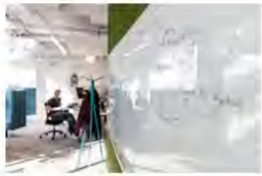


Setting up a cross-Region private environment in Amazon Braket

by Jagadeesh Pusapadi, Juan Moreno, and Niko Borodachuk | on 21 FEB 2022 | in Amazon Braket, Best Practices, Customer Solutions, Quantum Technologies, Technical How-to | Permalink | Share

At AWS we say that security is "job zero", by which we mean it is even more important than any number one priority. Amazon Braket is built around this philosophy; we often have conversations with customers who want to be able to work in a secured environment, and access Amazon Braket through private connections rather […]

Read More

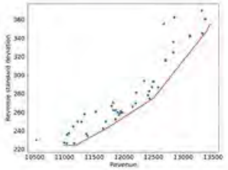


Quantum computing research in Poland with Amazon Braket

by Dariusz Matczak | on 31 JAN 2022 | in Amazon Braket, Announcements, Public Sector, Quantum

With a host of vibrant research centers, scientists and engineers in Poland have made
science over the past 40 years. Today, as a large number of quantum hardware techn
experimentation, Polish researchers are turning to Amazon Braket – the AWS quantu

Read More



Using quantum annealing on Amazon Braket for price optimization

by Feng Shi, Naz Levent, Helmut Katzgraber, Marco Guerriero, and Martin Schuetz | on 05 JAN 202
Intelligence, Customer Solutions, Quantum Technologies | Permalink | Share

Combinatorial Optimization is one of the most popular fields in applied optimizatio
every industry, including both private and public sectors. Examples include supply ch
planning, manufacturing layout design, facility planning, vehicle scheduling and rou
seasonal planning, telecommunication network […]

Read More

Winners announced in the BMW Group Quantum Computing Challenge

by James Goeders and Martin Schuetz | on 09 DEC 2021 | in Amazon Braket, Amazon Quantum Sol
Permalink | Share

The four winning teams of the BMW Quantum Computing Challenge were announce
Santa Clara, California. The challenge, focused on discovering potential quantum co
collaboration between the BMW Group and the Amazon Quantum Solutions Lab Pro

Read More

RESEARCH AREA

Quantum technologies

By harnessing the laws of quantum physics, quantum computers have the potential to solve problems that are beyond the reach of today's computers.

Explore more

AWS Center for Quantum Computing | Quantum computing | Quantum error correction | + 2 more

Recent publications

View all

Open quantum assembly language

Andrew Cross, Ali Javadi-Abhari, Thomas Alexander, Lev Bishop, Colm A. Ryan, Steven Heidel, Niel de Beaudrap, John Smolin, Jay M. Gambetta, Blake R. Johnson | 2022

Open quantum assembly language (OpenQASM 2) [1] was proposed as an imperative programming language for quantum computation based on earlier QASM dialects [2–6]. OpenQASM is one of the programming interfaces of the IBM Quantum services [7]. In the period since OpenQASM 2 was introduced, it has become something of a de… Read More

QUANTUM TECHNOLOGIES

Engineering fast bias-preserving gates on stabilized cat qubits

Qian Xu, Joseph Iverson, Fernando Brandão, Liang Jiang | 2022

Stabilized cat codes can provide a biased noise channel with a set of bias-preserving (BP) gates, which can significantly reduce the resource overhead for fault-tolerant quantum computing. All existing schemes of BP gates, however, require adiabatic quantum evolution, with performance limited by excitation loss and nonadiabatic errors during the… Read More

QUANTUM TECHNOLOGIES

Thank you!

Sebastian Stern
AWS Quantum Specialist Solutions Architect
sterseba@amazon.com

Quantum Centric Supercomputing

Ismael Faro
Distinguished Engineer, Tech Lead IBM Quantum Services

IBM **Quantum**

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 ✓
  - Machine learning | Natural science | Optimization
- Quantum Serverless
  - Intelligent orchestration
  - Circuit Knitting Toolbox
  - Circuit libraries

Kernel Developers

- Circuits ✓
- Qiskit Runtime ✓
  - Dynamic circuits ✓
  - Threaded primitives (on target)
  - Error suppression and mitigation
  - Error correction

System Modularity

Falcon ✓	Hummingbird ✓	Eagle ✓	Osprey ✓	Condor (on target)	Flamingo	Kookaburra	Scaling to
27 qubits	65 qubits	127 qubits	433 qubits	1,121 qubits	1,386+ qubits	4,158+ qubits	10K-100K qubits with classical and quantum communication
				Heron (on target) 133 qubits x p	Crossbill 408 qubits		

Development Roadmap

Executed by IBM
On target

IBM **Quantum**

Quantum-centric *supercomputing*

01 Modularity for quantum

02 Communication for quantum

03 Middleware for 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 — Machine learning | Natural science | Optimization

Quantum Serverless

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

Development Roadmap

Executed by IBM
On target

IBM **Quantum**

Quantum-centric *supercomputing*

01 Modularity for quantum

02 Communication for quantum

03 Middleware for 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 Machine learning | Natural science | Optimization			Quantum Serverless	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		

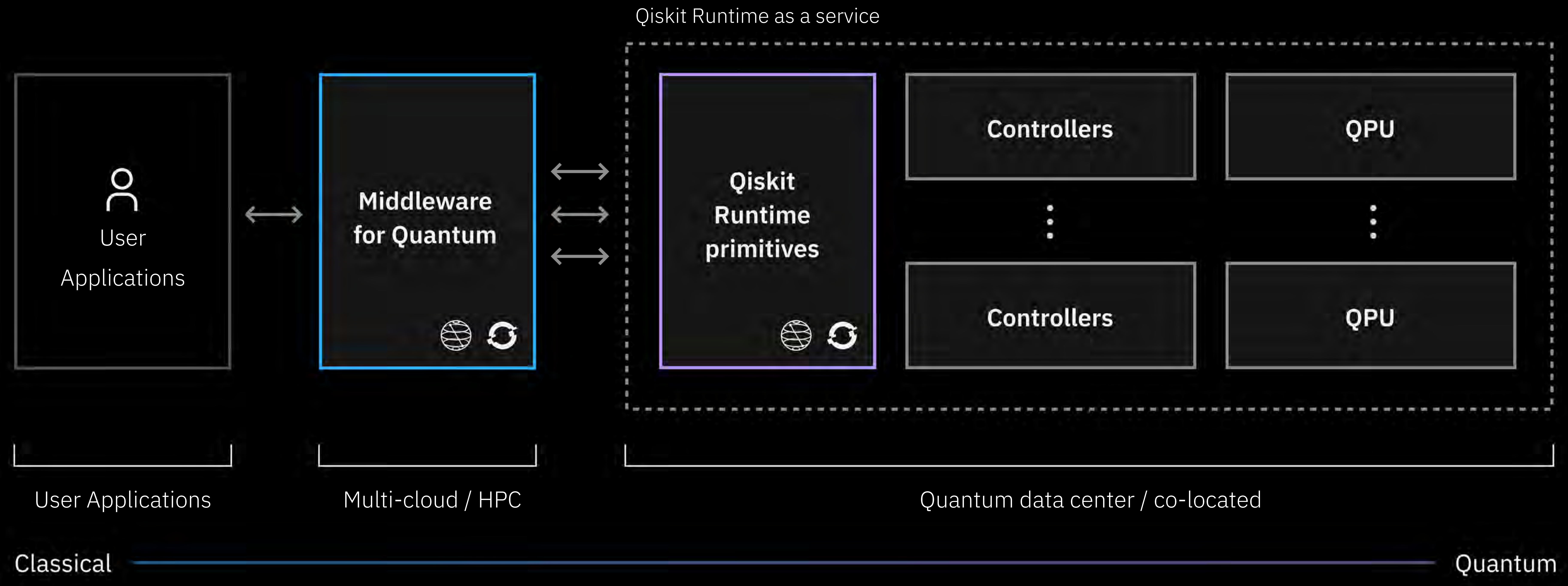
Middleware for quantum

Middleware for quantum

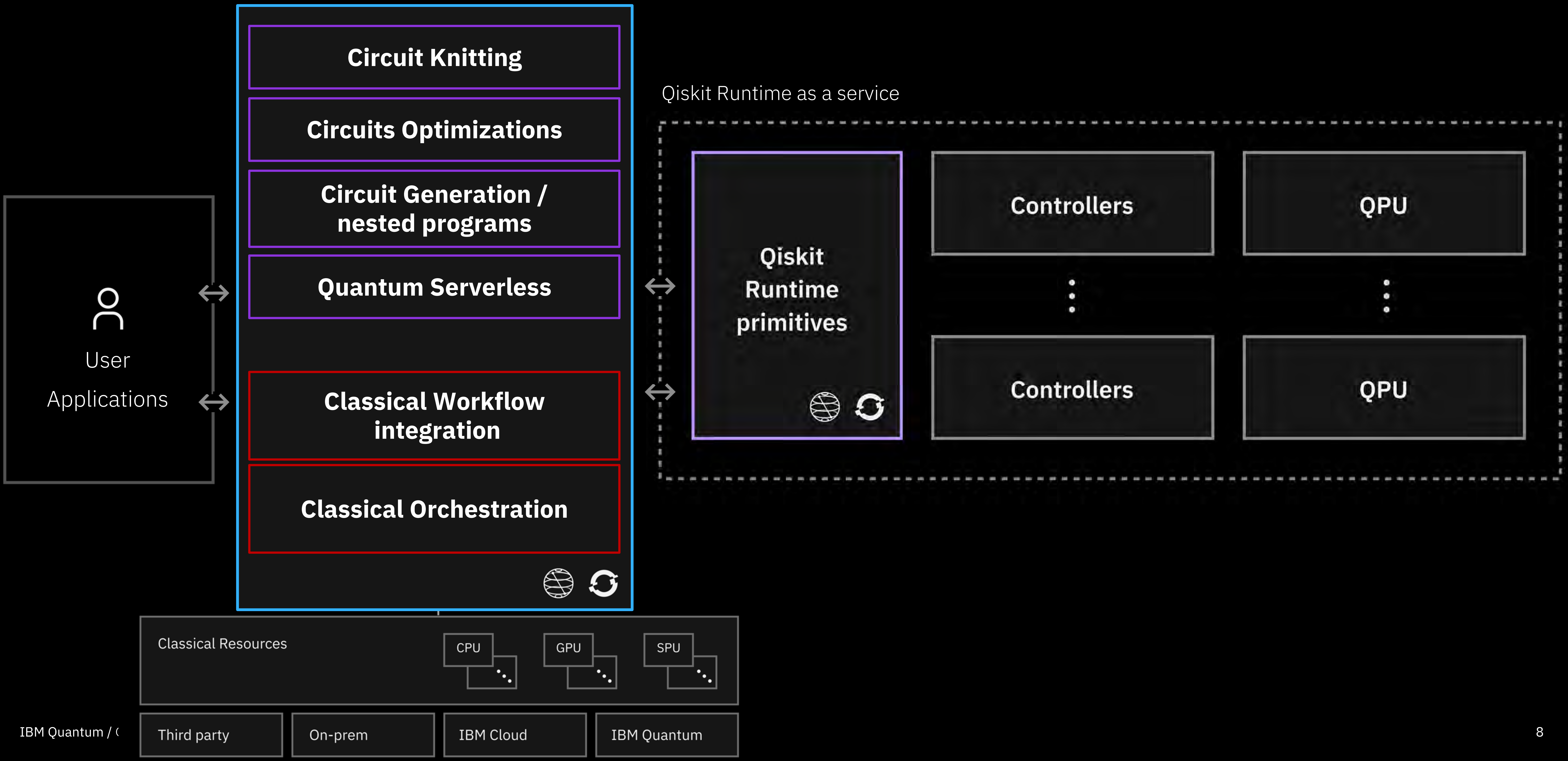
is the abstraction layer to enable a seamless integration between classical ecosystems of applications and quantum resources, based in open-source software and services in multi-cloud/classical environments.

Quantum + Classical Integration

IBM **Quantum**



Middleware for quantum



Middleware for quantum

01 Quantum Serverless

02 Circuit Knitting toolbox

Middleware for quantum

01 Quantum Serverless

- Enabling flexible quantum-classical resource orchestration
- Integration with other circuit tools such Circuit Knitting, Nested Program, Compilation, Synthesis, Layout & Routing, Optimization
- Multi-cloud/HPC (Ray.io, Knative, CodeEngine, CodeFlare, OpenShift, LSF, Slurm)

Middleware for quantum

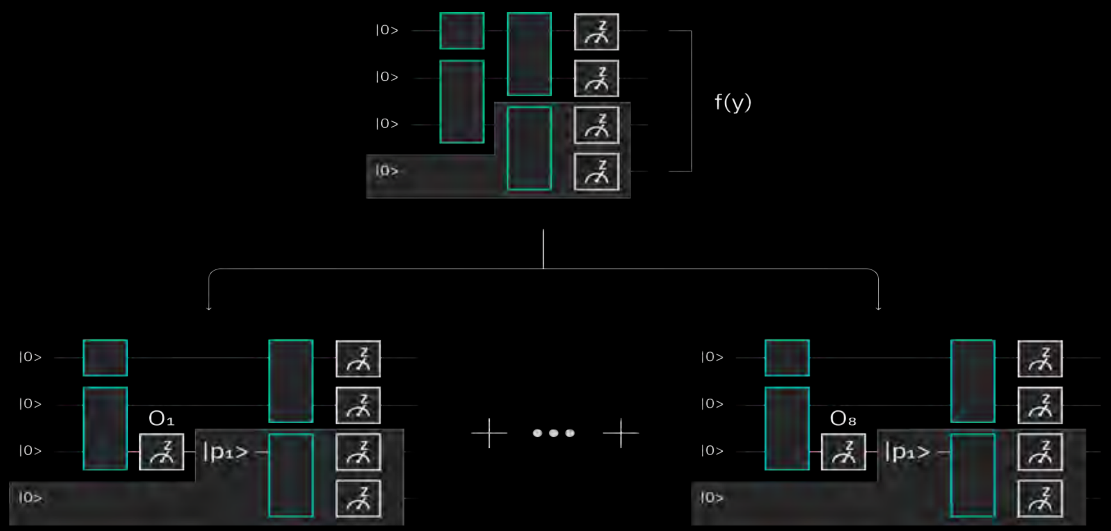
IBM **Quantum**

02 Circuit Knitting toolbox

Circuit cutting

Simulate a large quantum circuit using small QPUs by cutting the circuit into subcircuits, which are then sent to QPUs

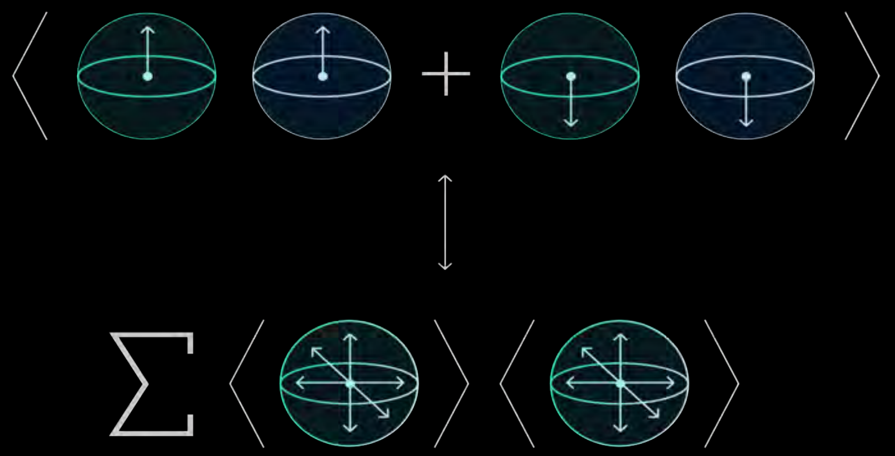
The output of the original circuit is built from classical post-processing of the subcircuits outputs



Entanglement forging

Break down a correlated system into smaller subsystems which can be tackled by smaller QPUs.

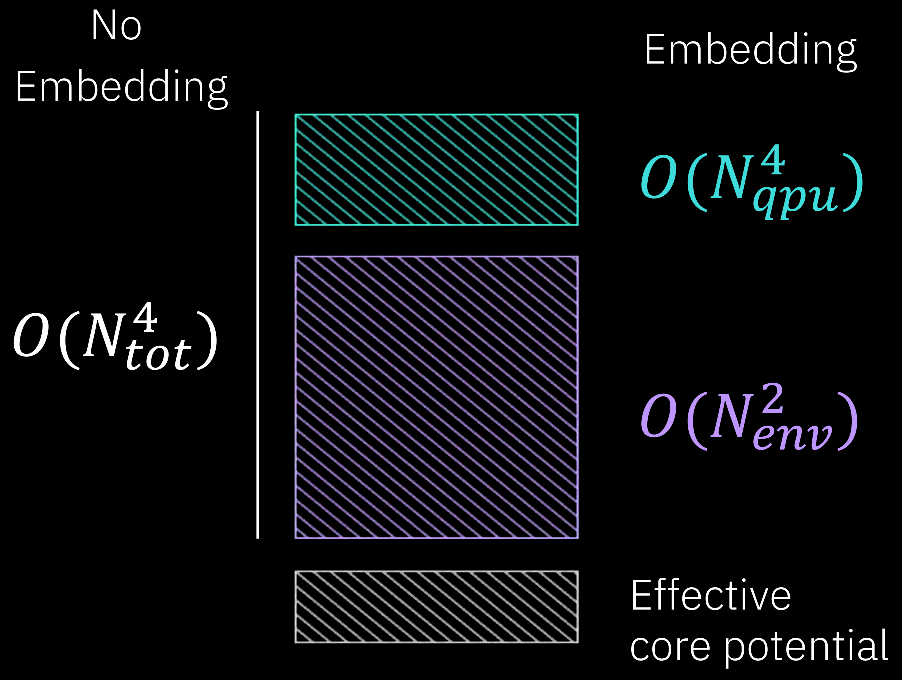
Recover the lost correlations with classical post-processing of the QPUs outputs



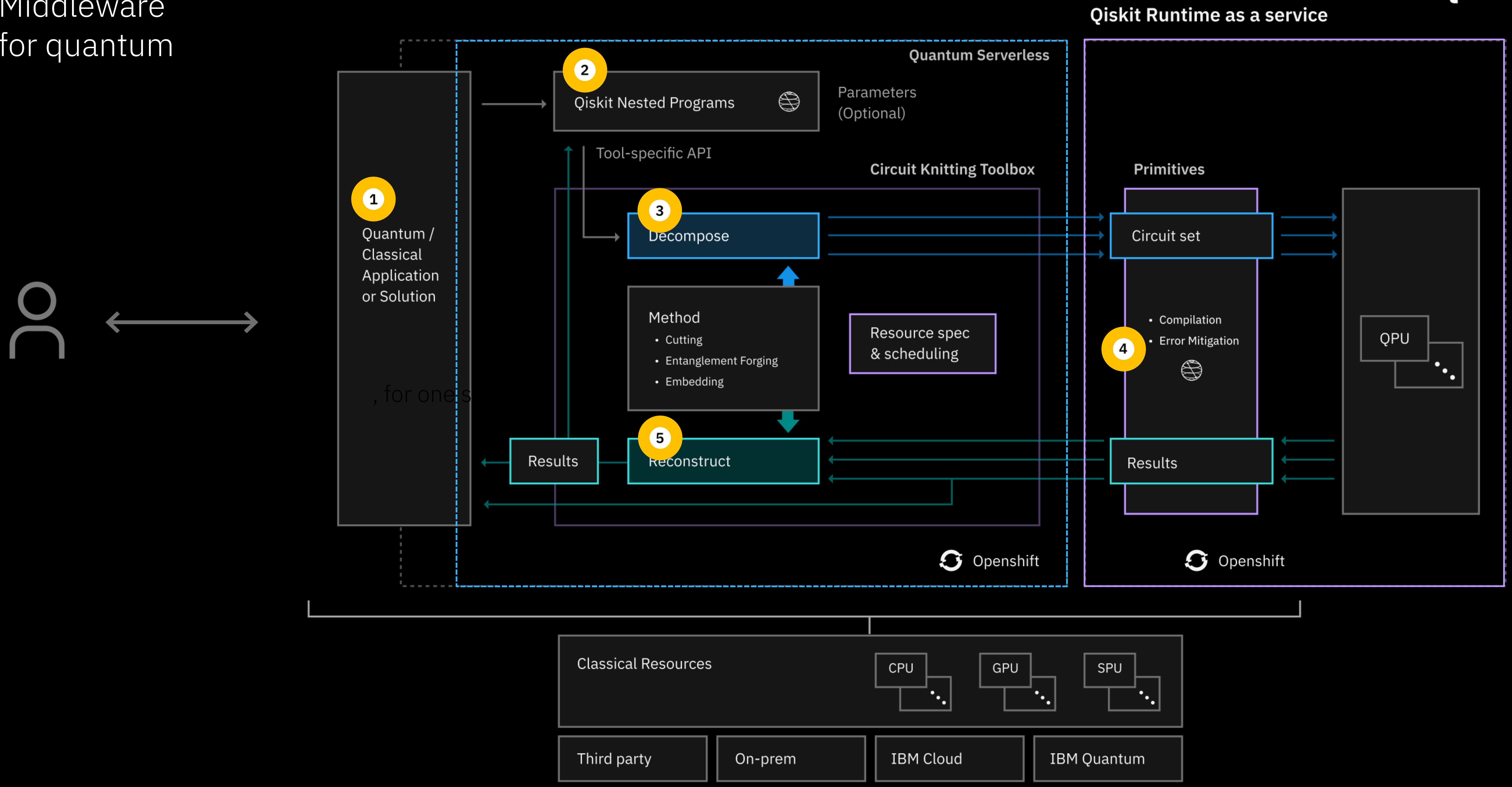
Embedding

Effective leveraging of QPU resources: Only the part of the problem which most benefits from exploiting entanglement is undertaken by the QPU

The CPU is efficient in tackling the remaining of the problem



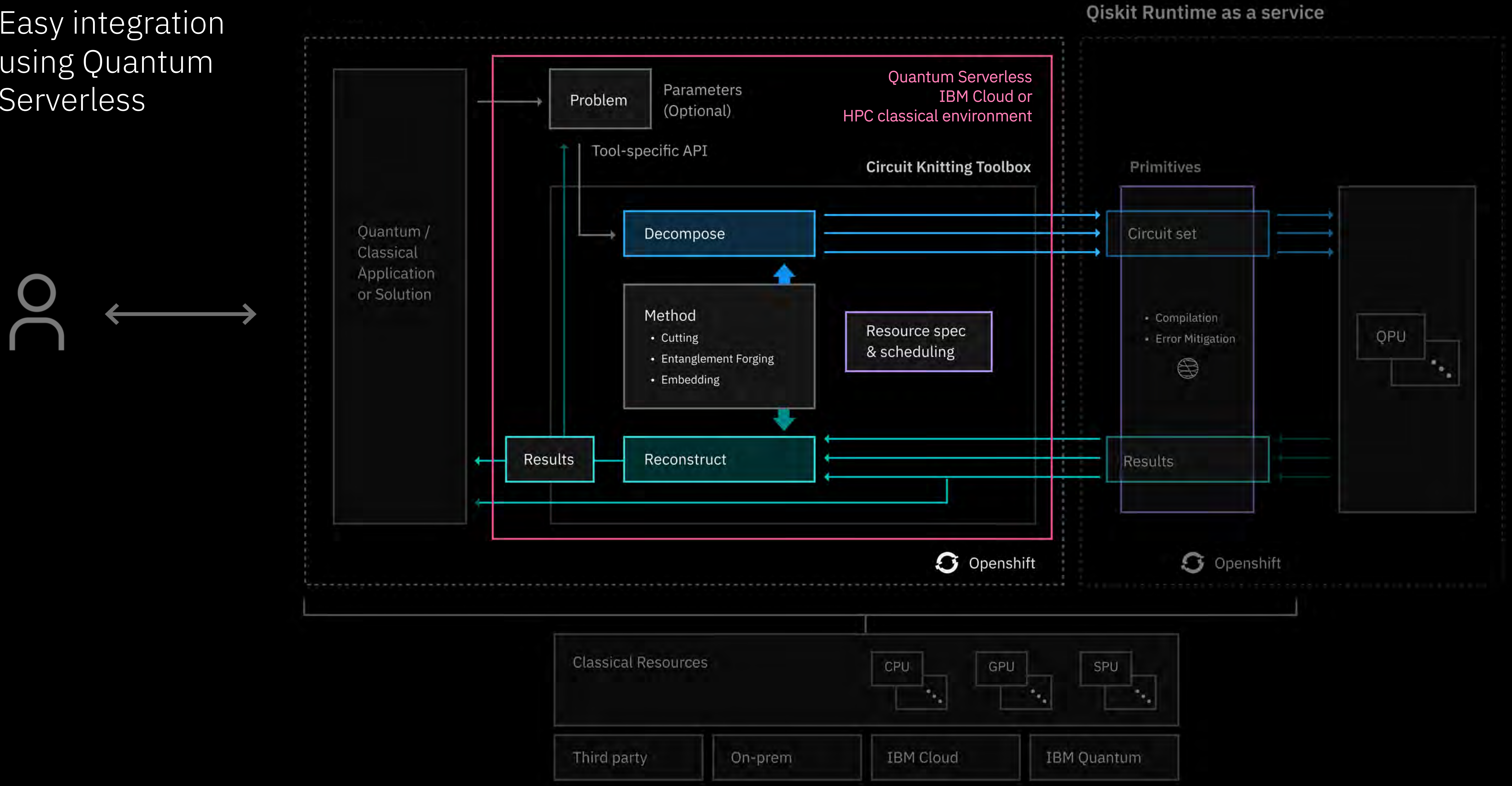
Middleware for quantum



Demo 01

Easy integration using Quantum Serverless

IBM **Quantum**



Demo 01 | Easy integration using Quantum Serverless


```
from quantum_serverless import run_qiskit_remote

@run_qiskit_remote()
def electronic_structure_problem(molecule, backend,
service)
    ...
    problem = ElectronicStructureProblem(driver)
    op = QubitConverter.convert(problem, ...)
    ...
    with Session(service, backend) as session:
        estimator = Estimator(session)
        vqe = EstimatorVQE(estimator, ansatz, ...)
        energy = vqe.compute_minimum_eigenvalue(op)
    return energy
```

Execute VQEs in parallel on the cloud:

```
electronic_structure_problem(molecule_0, backend_0)
```

```
electronic_structure_problem(molecule_1, backend_1)
```

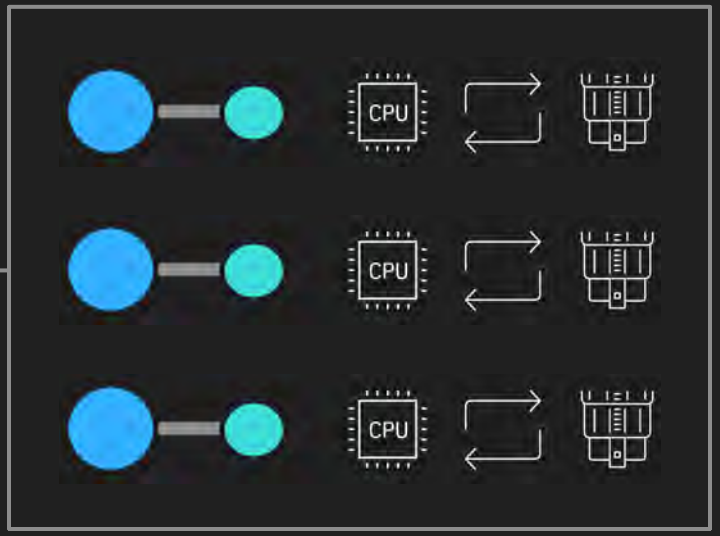
⋮

```
electronic_structure_problem(molecule_N, backend_M)
```

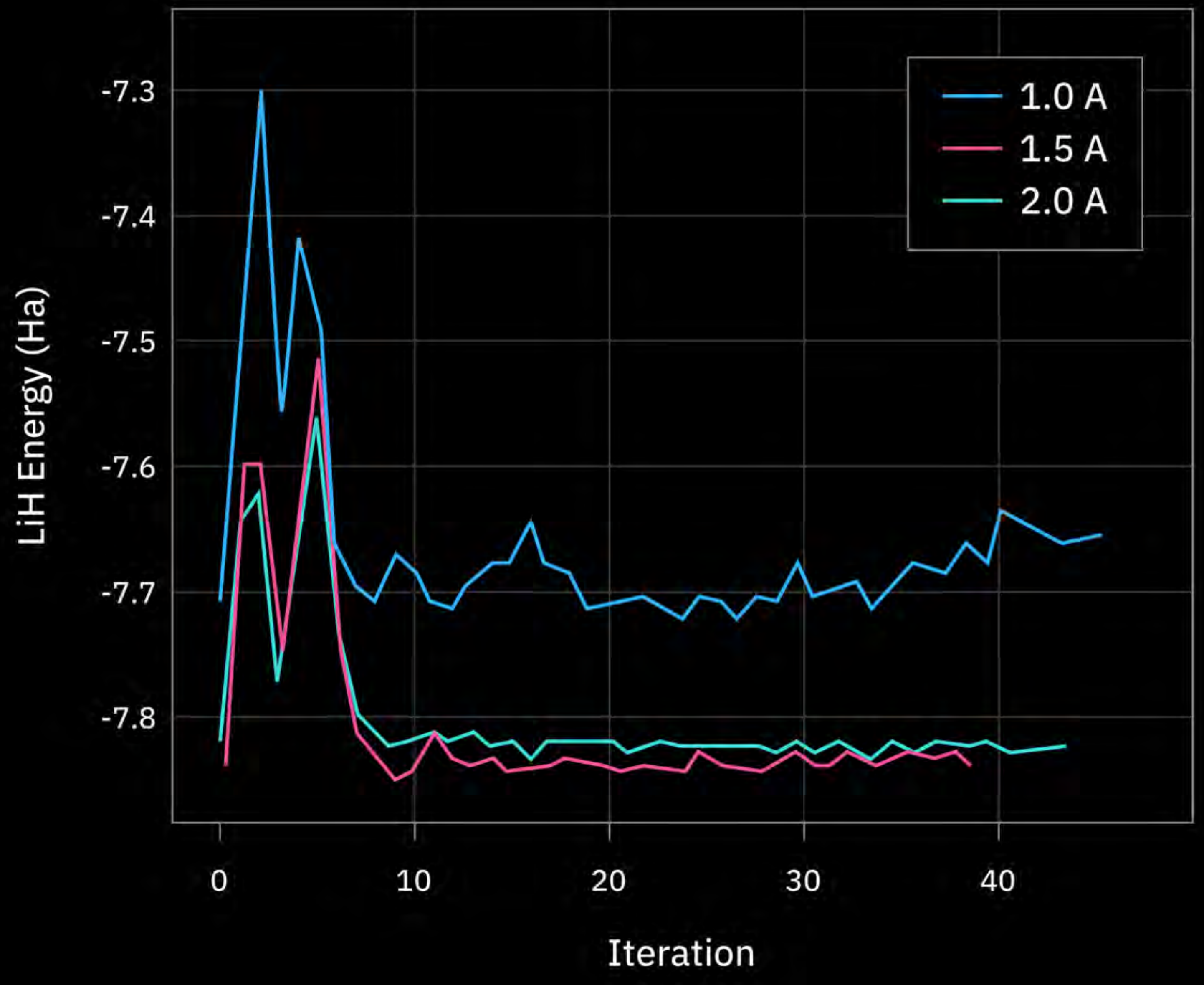
Demo 01 | Configuring a parallel workload

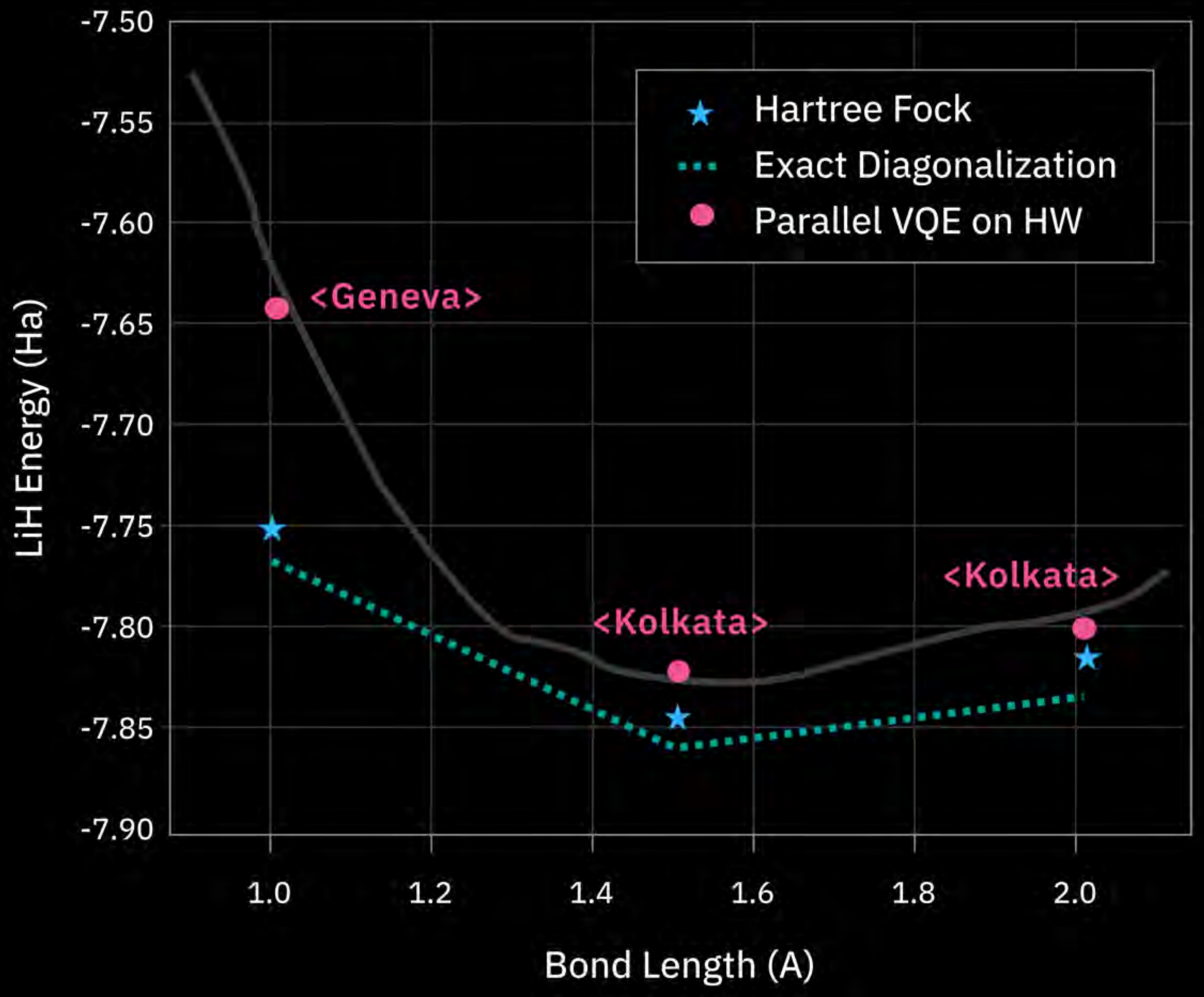

```
from qiskit_ibm_runtime import QiskitRuntimeService
from quantum_serverless import QuantumServerless

service = QiskitRuntimeService()
serverless = QuantumServerless(...)

with serverless.provider('ibm_cloud'):

    energies = electronic_structure_problem(
            molecules = [
                Molecule(geometry = [["Li", [0, 0, 0]], ["H", [0, 0, 1.0]]]),
                Molecule(geometry = [["Li", [0, 0, 0]], ["H", [0, 0, 1.5]]]),
                Molecule(geometry = [["Li", [0, 0, 0]], ["H", [0, 0, 2.0]]])
            ],

            backends = [
                service.backend('ibmq_geneva'),
                service.backend('ibmq_kolkata'),
                service.backend('ibm_auckland')
            ],
            ...
    )
```

→ Create serverless context with a cloud or HPC provider

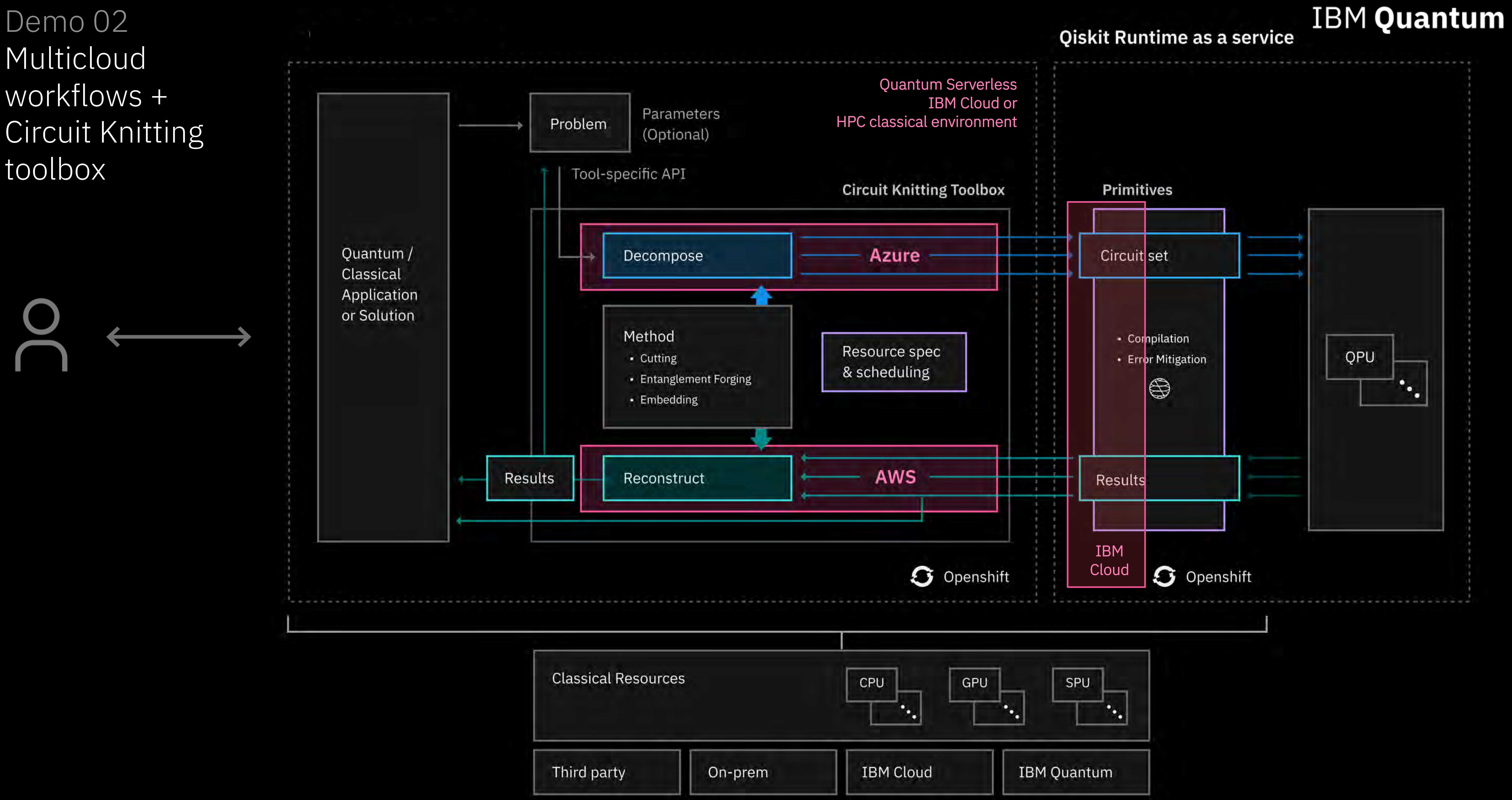


Demo 01 | Parallelized LiH ground state computations





Demo 02
Multicloud workflows + Circuit Knitting toolbox



Demo 02 | Multicloud workflows + Circuit Knitting toolbox

```
service = QiskitRuntimeService(...)
backends = [”ibmq_kolkata”, “ibm_auckland”]

from circuit_knitting_toolbox.circuit_cutting
import WireCutter
cutter = WireCutter(circuit, service, backends)
```

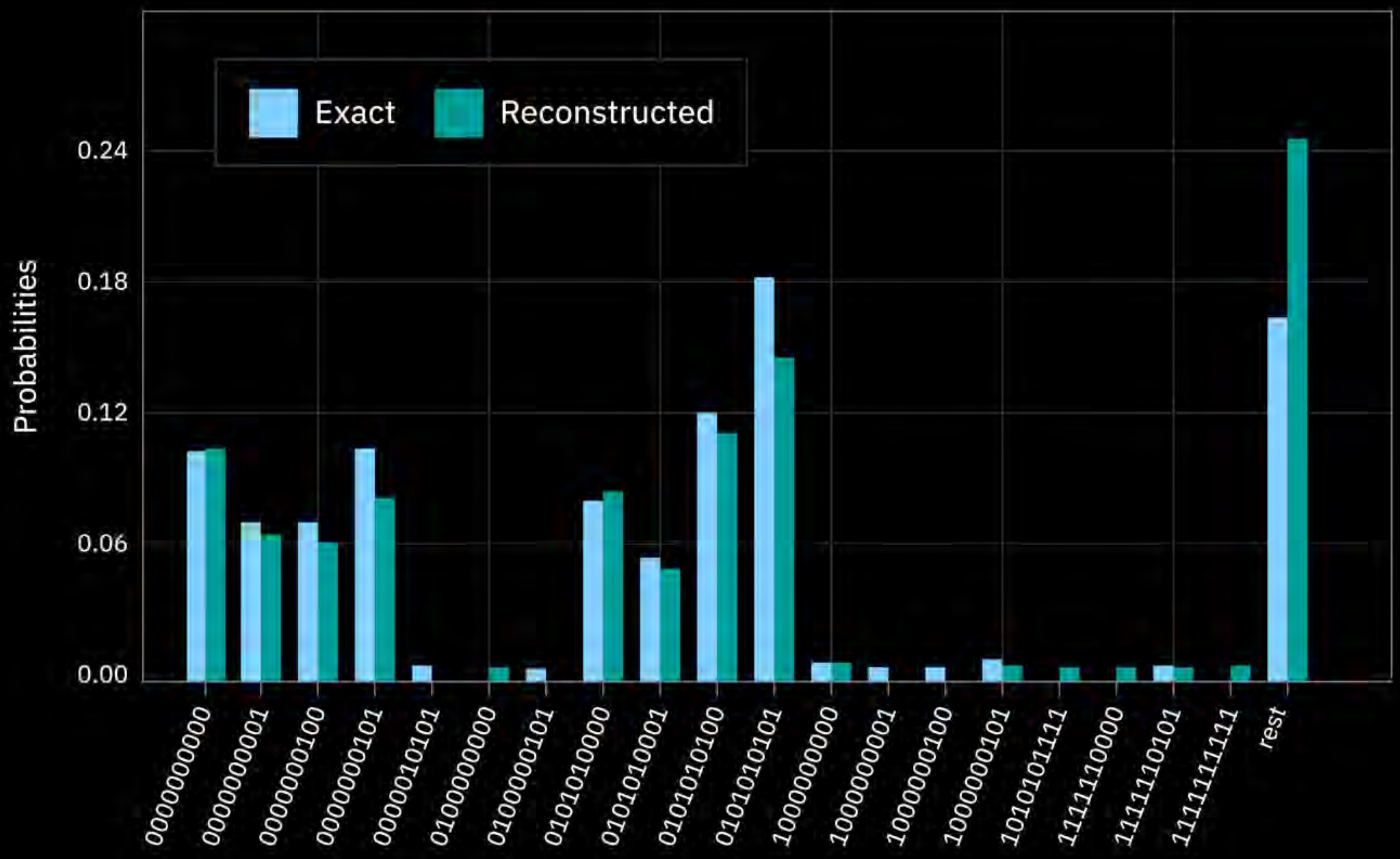
```
# cut (decompose) the circuit on Azure
with serverless.provider(‘azure’):
    cuts = cutter.decompose(“automatic”, ...)
```

```
# execute subcircuits using Qiskit Runtime Primitives
with serverless.provider(’ibm_cloud’):
    subcircuit_results = cutter.evaluate(cuts)
```

```
# reconstruct probabilities on AWS
with serverless.provider(‘aws’):
    reconstructed_result =
cutter.recompose(subcircuit_results, cuts)
```

Multi-cloud or HPC solutions

Reconstructed result



Middleware for quantum

IBM **Quantum**

01 Quantum Serverless

https://github.com/Qiskit-Extensions/quantum-serverless

02 Circuit Knitting toolbox

https://github.com/Qiskit-Extensions/circuit-knitting-toolbox

IBM **Quantum**

Large Scale Quantum / HPC hybridization with Atos QLM

Arnaud GAZDA

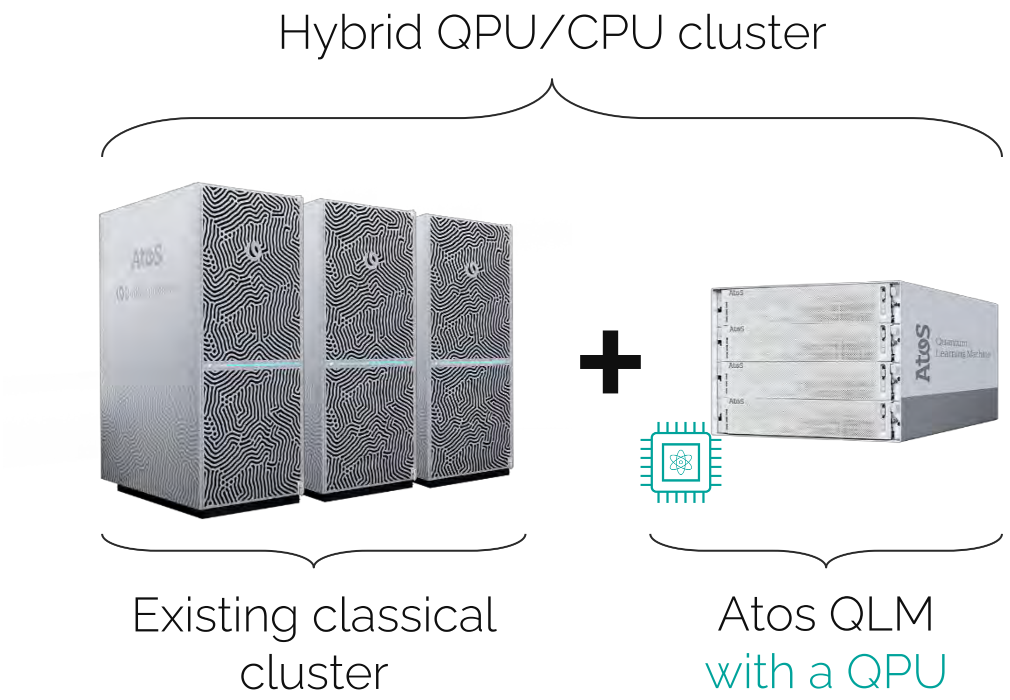
Atos QLM extends existing clusters to make them hybrid

Hybrid clusters

<u>Hybridization:</u> ability to perform advanced computations using both classical and quantum resources

HPC centers have classical resources. How to rely on these classical resources?

HPC centers are managed by advanced scheduling systems. How to rely on these systems.



Atos

Content Overview

01. Atos QLM and hybrid computation stacks

02. Accessing QPUs remotely

03. Scheduling hybrid jobs on the cluster

01. Atos QLM and hybrid computation stacks

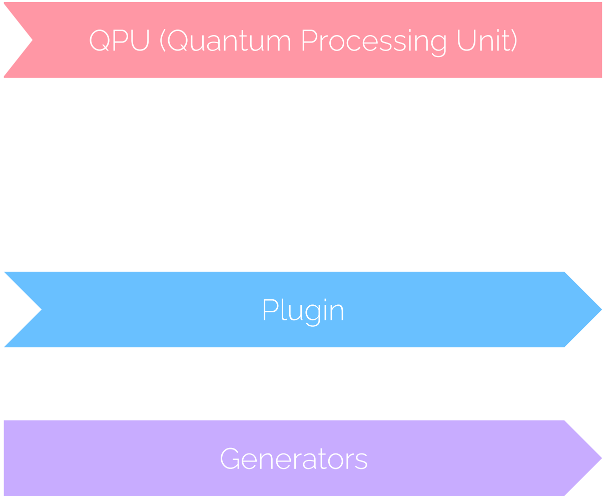
Atos

QLM computation chains can be composed of classical components

Design of Atos QLM

QLM defines 3 type of services

- Executes a quantum job, can either be:
  - *A simulator (running on CPU / GPU / …)*
  - *A QPU*
- Manipulates quantum jobs classically before and / or after their execution by a QPU
- Generates inputs (i.e. quantum jobs)



QLM computation chains can be composed of classical components

Design of Atos QLM

A computation chains can be built by stacking services. A chain is composed of:

- 1 QPU
- One or more plugins *optional*
- A list of quantum jobs *or* a strategy to build jobs

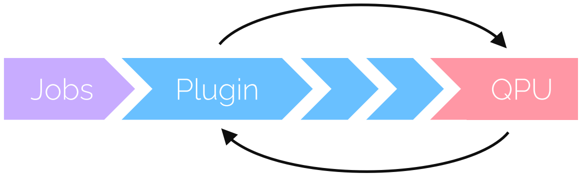
Atos

QLM computation chains can be composed of classical components
Design of Atos QLM

A computation chains can be built by stacking services. A chain is composed of:

- 1 QPU
- One or more plugins *optional*
- A list of quantum jobs *or* a strategy to build jobs

Moreover, plugins can resubmit quantum jobs:

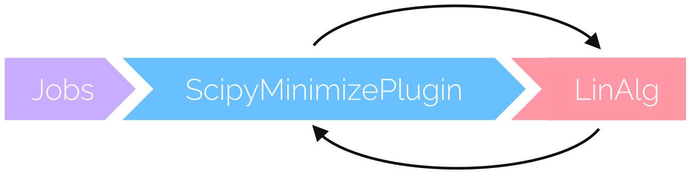


Plugins could be used to run variational jobs on the QLM

Example of QLM python code

```python
jobs = ...

from qat.qpus import LinAlg
from qat.plugins import ScipyMinimizePlugin

# Create a new QPU having classical capabilities
# The classical capabilities are used to perform
# classical optimization
stack = ScipyMinimizePlugin() | LinAlg()

# Submit "jobs" to "stack"
results = stack.submit(jobs)
```

- Create a quantum job
- Import plugins and QPUs used by the simulation
- Create a processing stack
- Submit the job

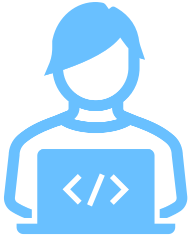


Atos

02. Accessing QPUs remotely

The hybrid cluster can be accessed remotely using Atos QLM

Introduction to myQLM Power Access



myQLM

- QPUs
- Plugins
- Generators
- ...

Local and remote computation use the same structure

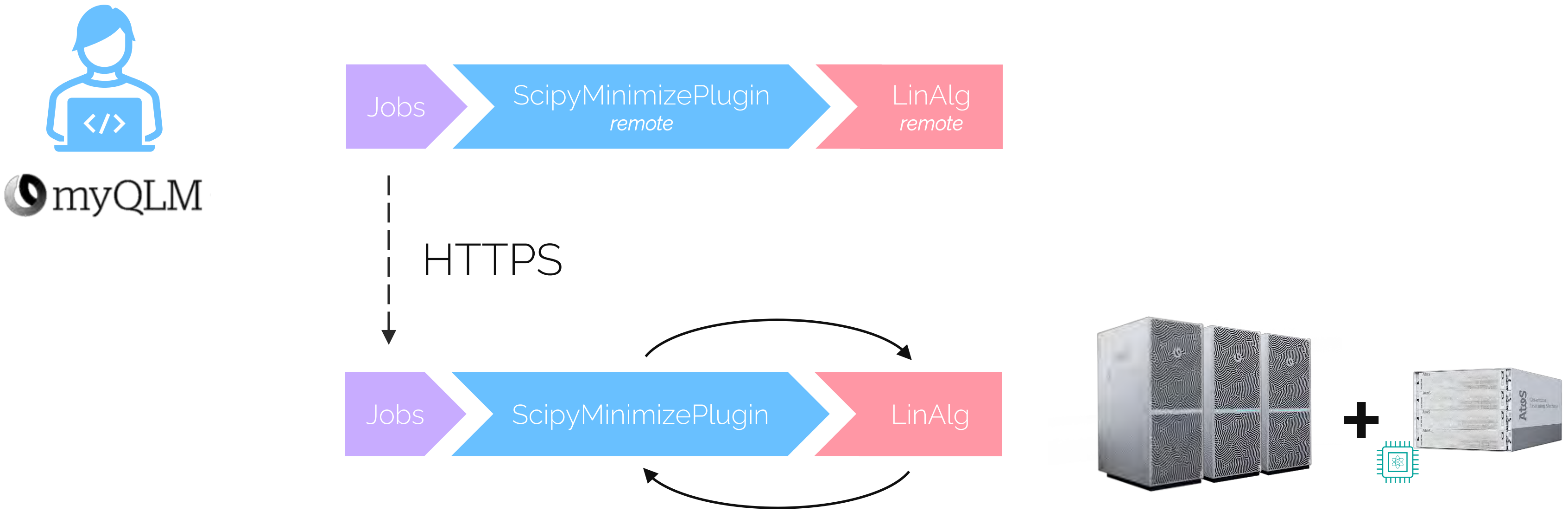
Example of QLM python code

```
jobs = ...

from qlmaas.qpus import LinAlg
from qlmaas.plugins import ScipyMinimizePlugin

# Create a new QPU having classical capabilities
# The classical capabilities are used to perform
# classical optimization
stack = ScipyMinimizePlugin() | LinAlg()

# Submit "jobs" to "stack"
future_results = stack.submit(jobs)
results = future_results.join()
```

- Create a quantum job
- Import plugins and QPUs used by the simulation
- Create a processing stack
- Submit the job

Remote stacks are rebuilt on the server side

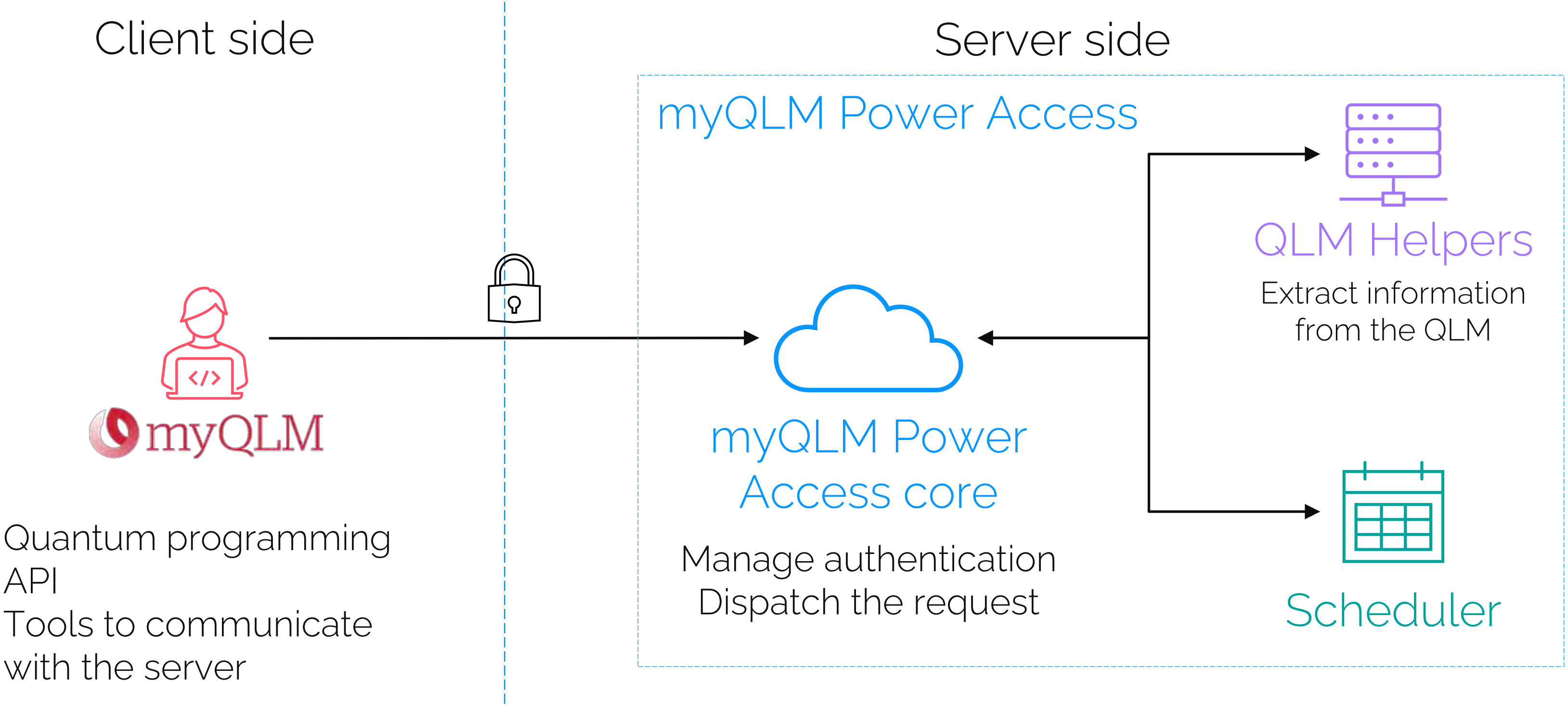
Example of QLM python code



03. Scheduling hybrid jobs on the cluster

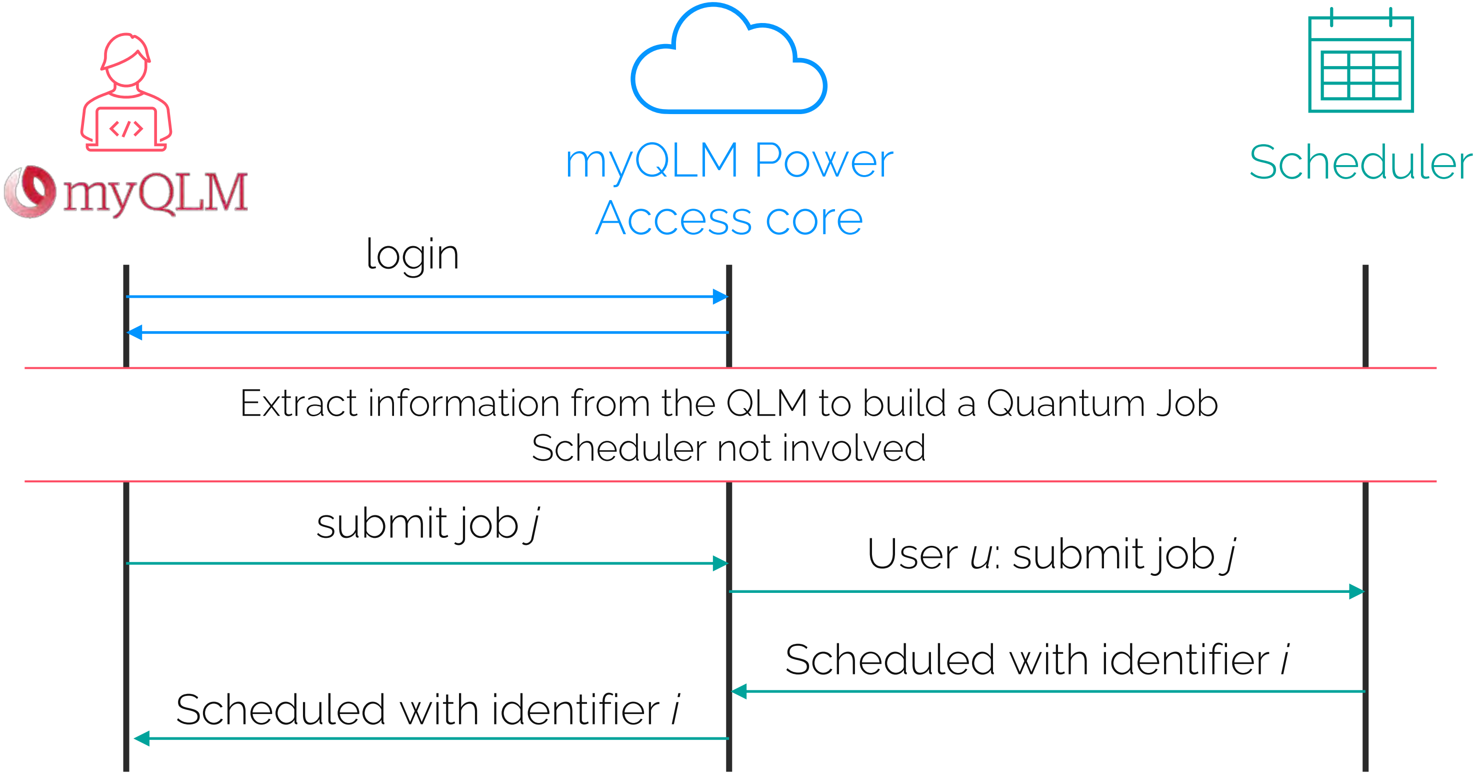
myQLM Power Access Overview

Components involved in myQLM Power Access



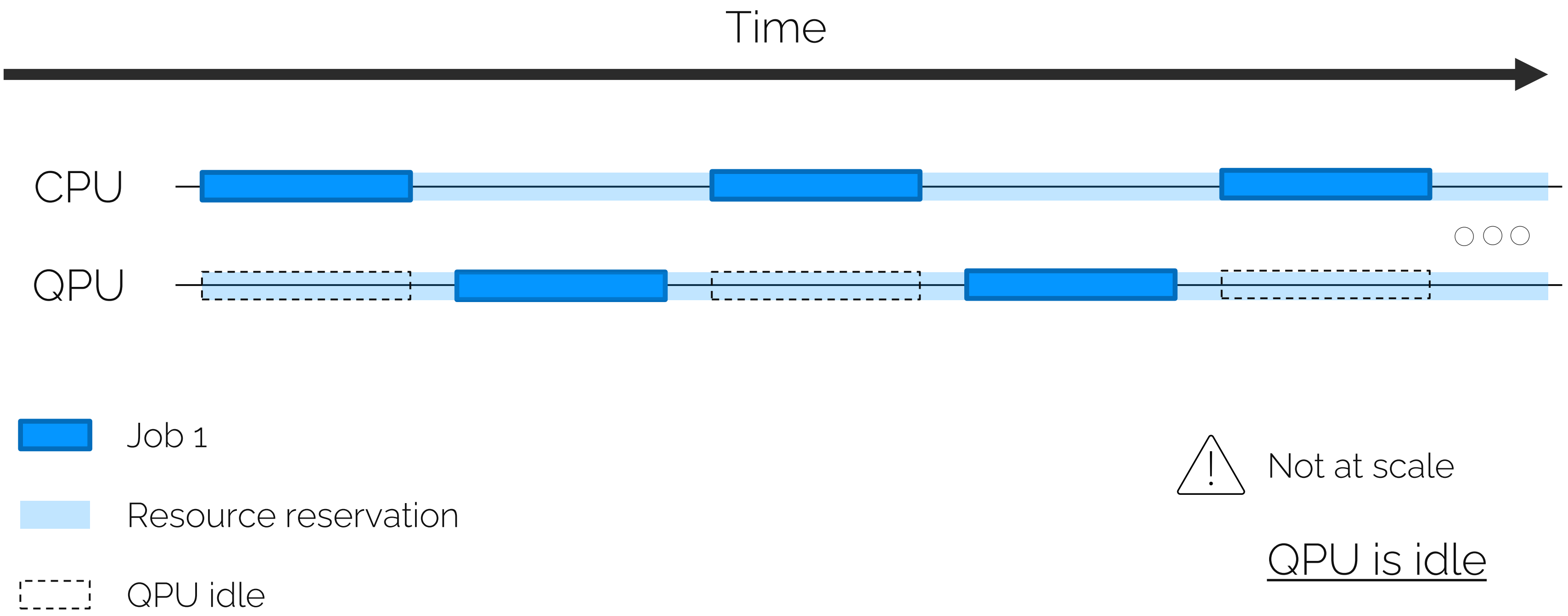
myQLM Power Access Overview

Components involved in myQLM Power Access



Quantum job scheduling

High-level scheduling and QPU idleness

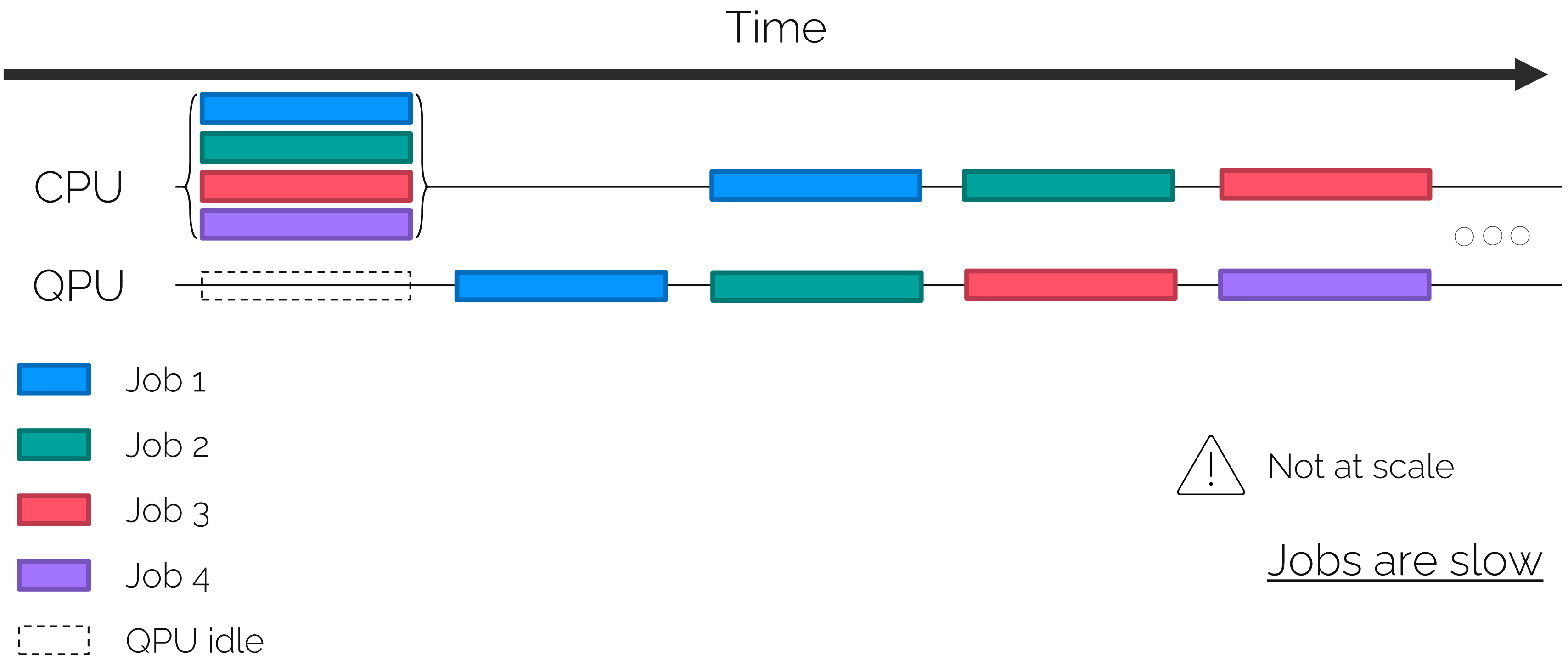


QPU is idle

Atos

Quantum job scheduling

Low-level scheduling and slow Quantum jobs



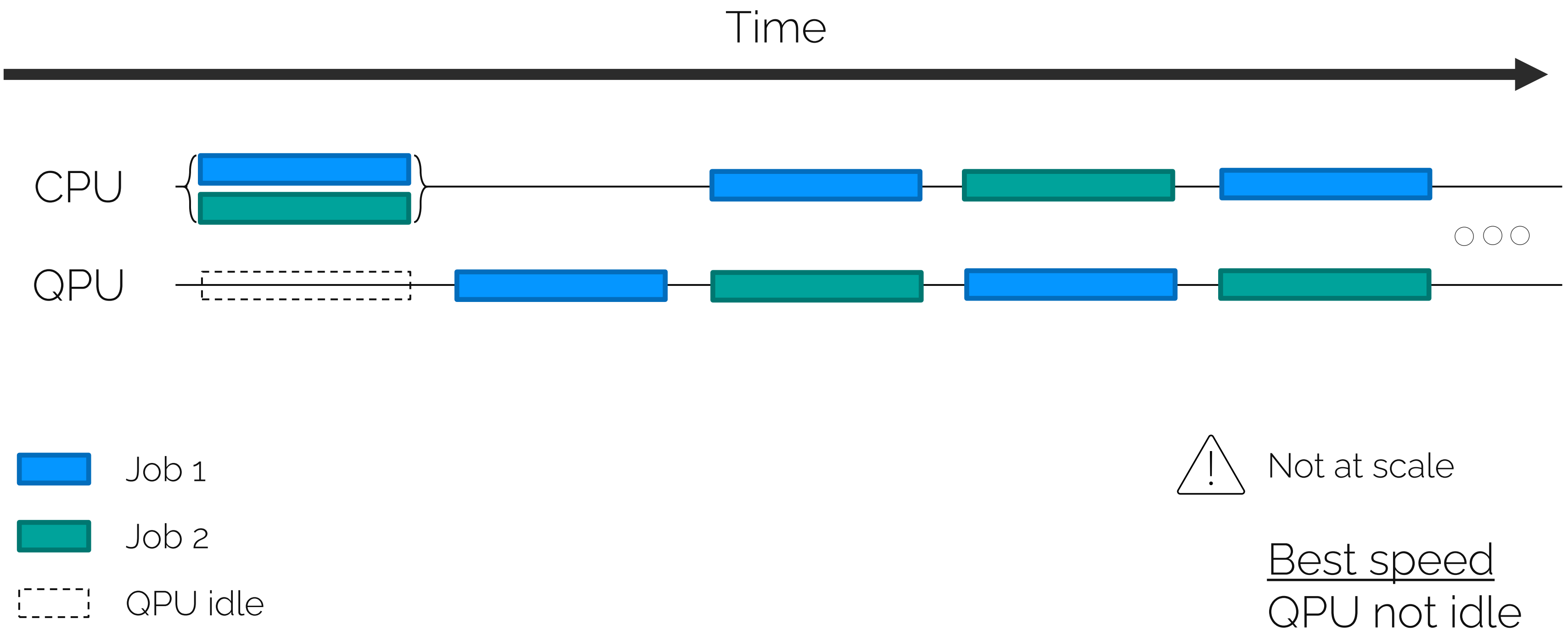
⚠ Not at scale

<u>Jobs are slow</u>

Atos

Quantum job scheduling

High/Low-level scheduling



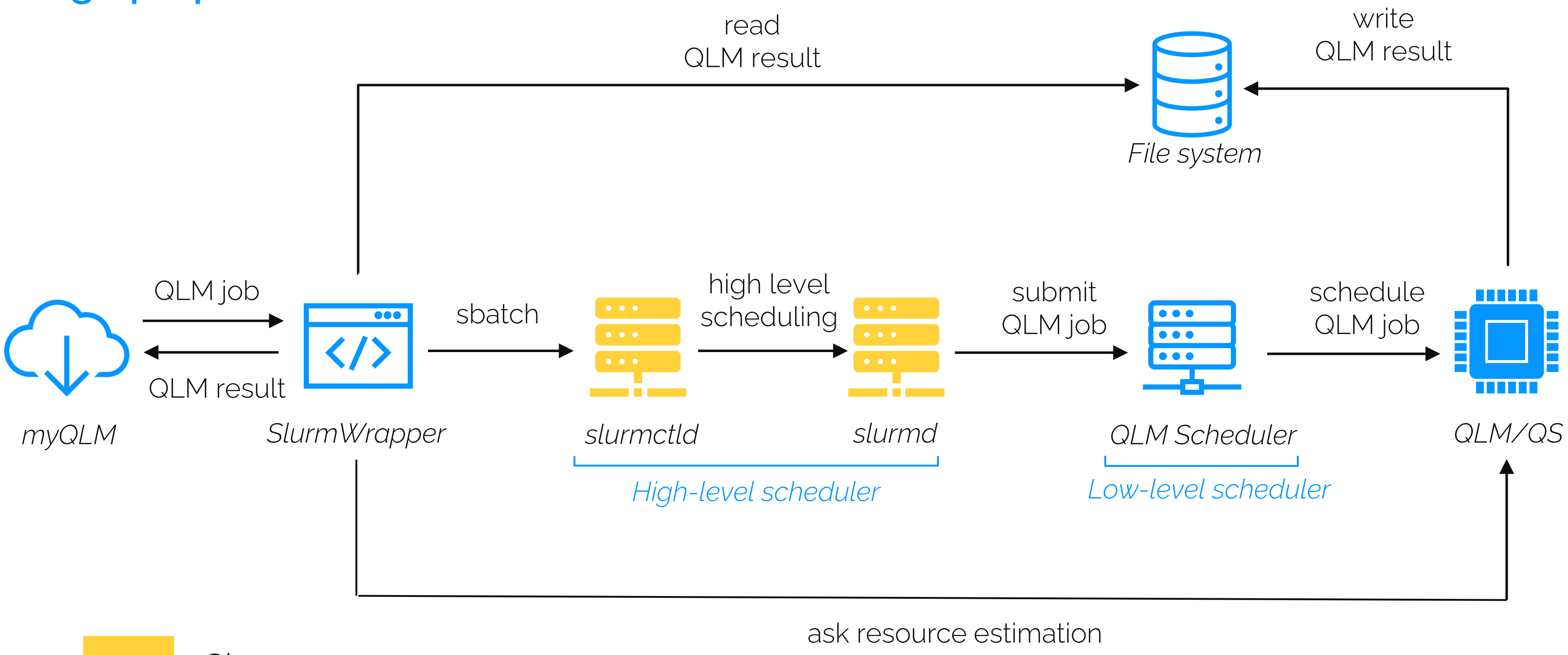
⚠ Not at scale

Best speed
QPU not idle

Atos

Quantum job scheduling

Design proposal



Slurm

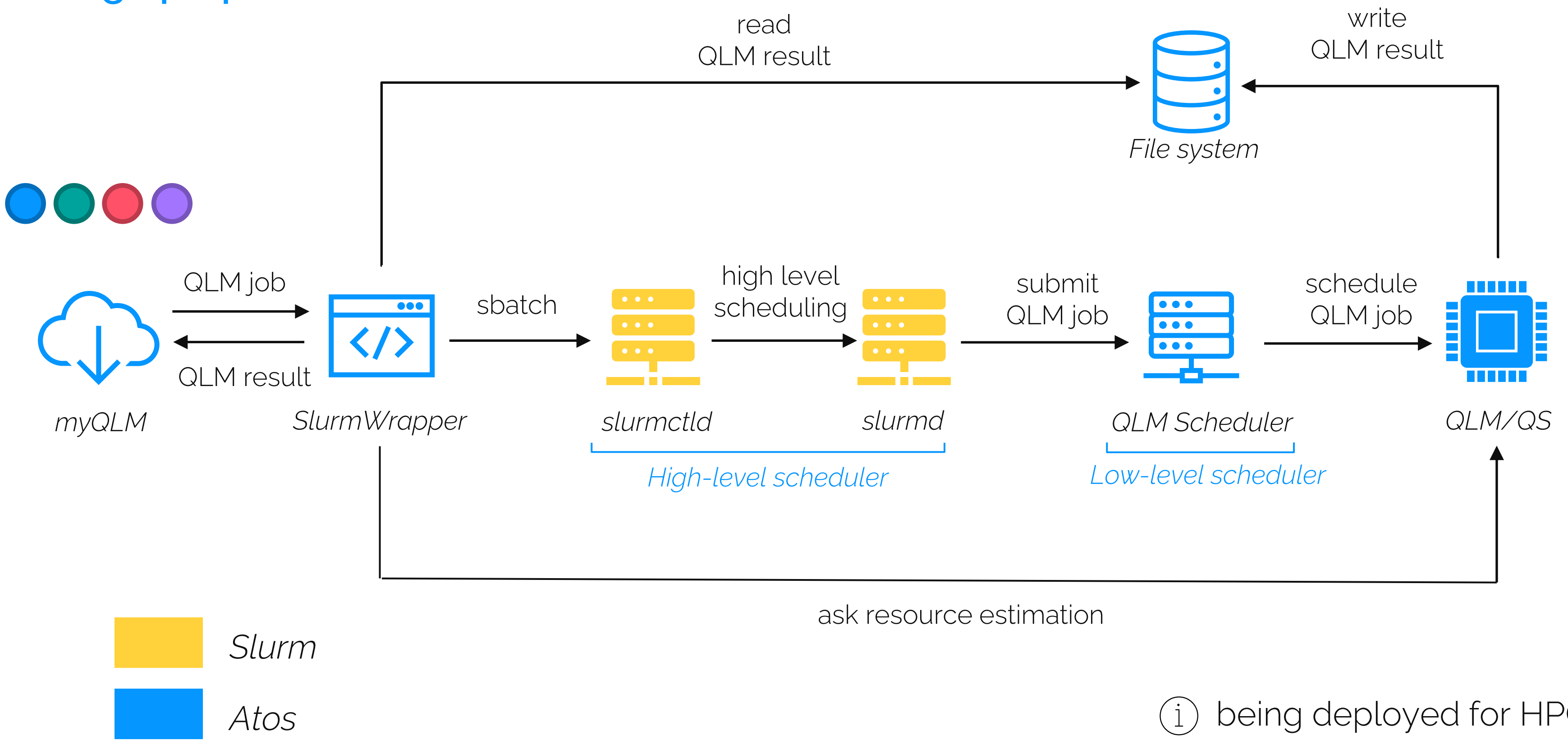
Atos

ⓘ being deployed for HPCQS

Atos

Quantum job scheduling

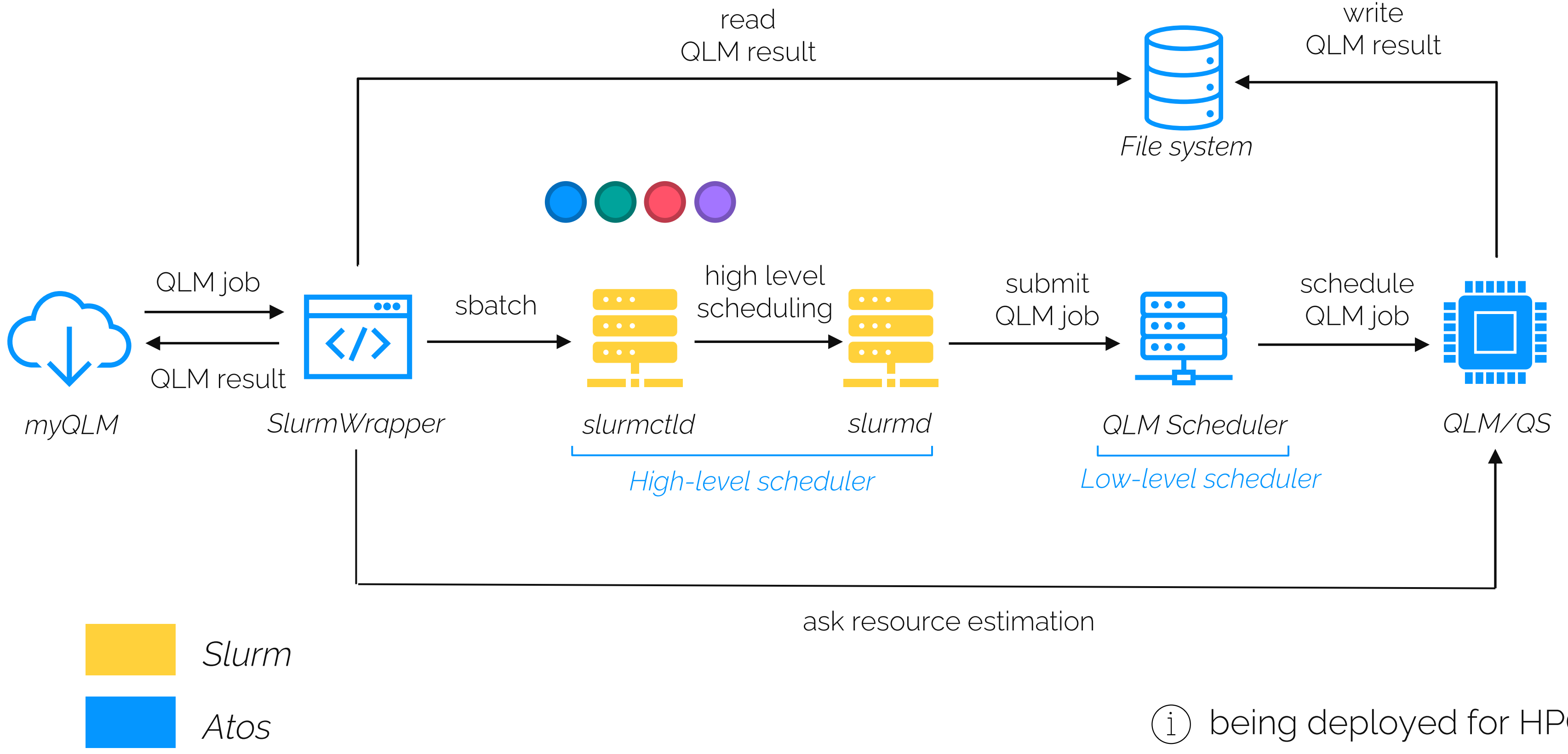
Design proposal



ⓘ being deployed for HPCQS

Quantum job scheduling

Design proposal

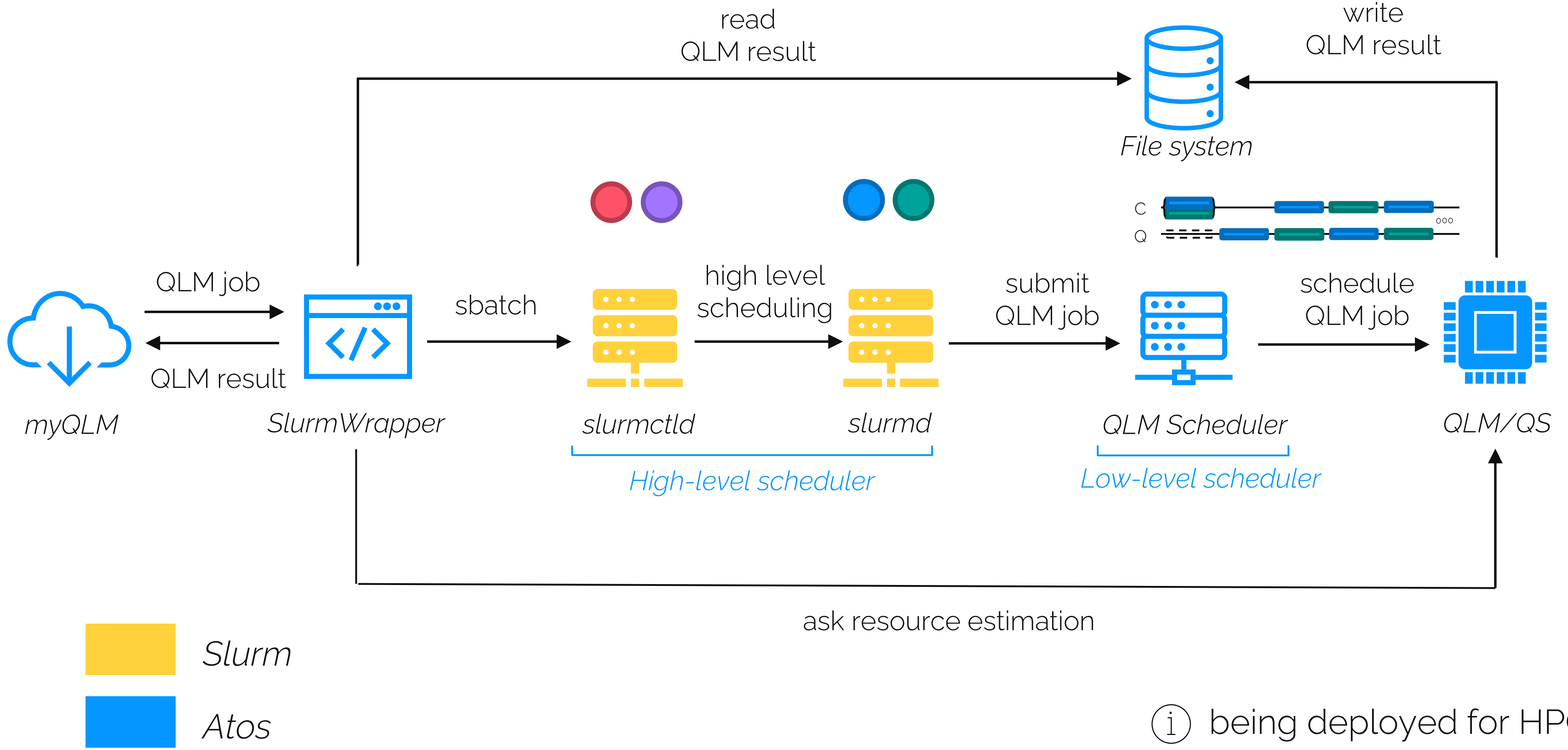


ⓘ being deployed for HPCQS

Atos

Quantum job scheduling

Design proposal



ⓘ being deployed for HPCQS

Questions

Thank you!

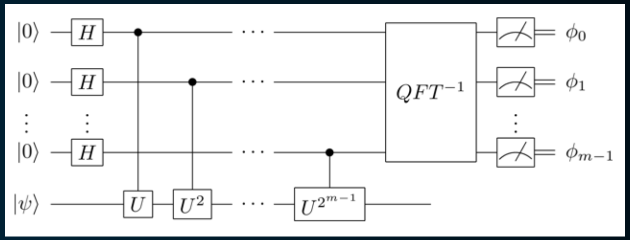
For more information please contact:
arnaud.gazda@atos.net

Phase estimation variants

and its implication for quantum/classical architecture

Outline

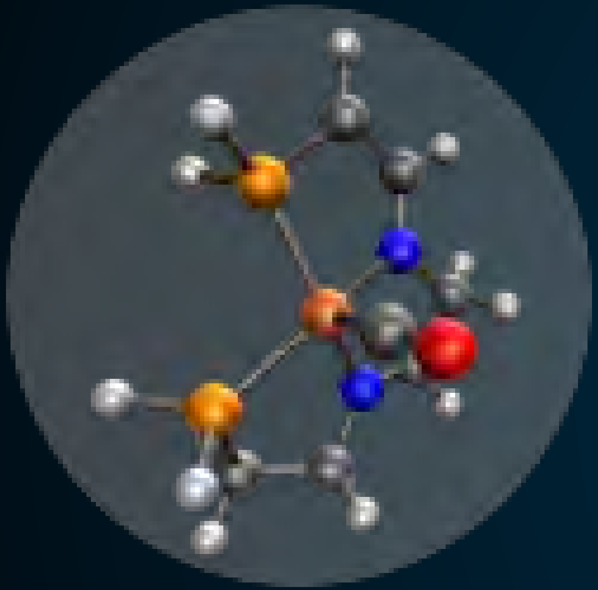
- Phase estimation variants and their architecture requirements



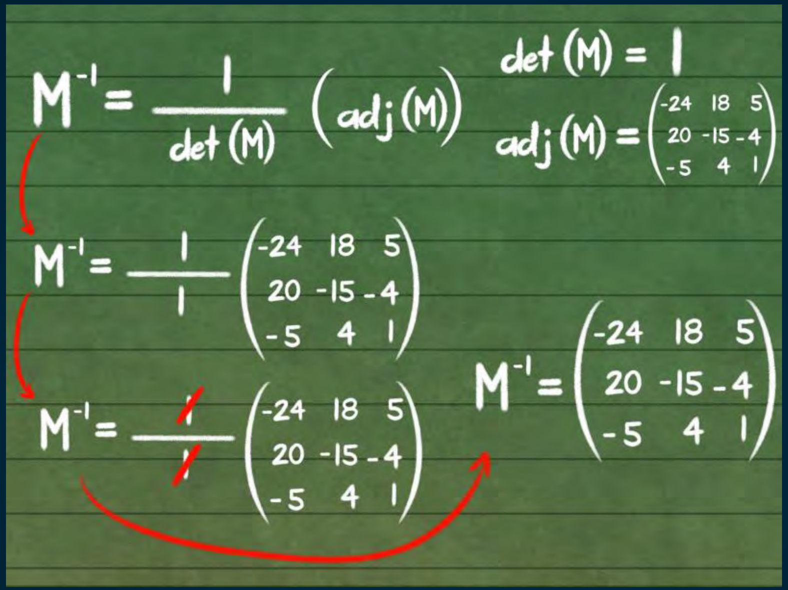
- 5 levels of quantum / classical integration

- The case for a hybrid Intermediate Representation

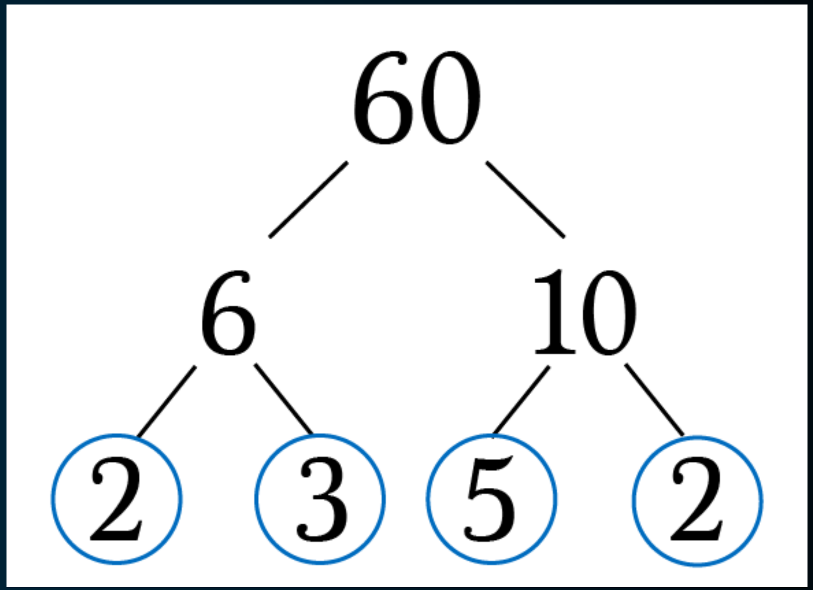
Quantum phase estimation: use cases



Molecular simulation
and
Material design

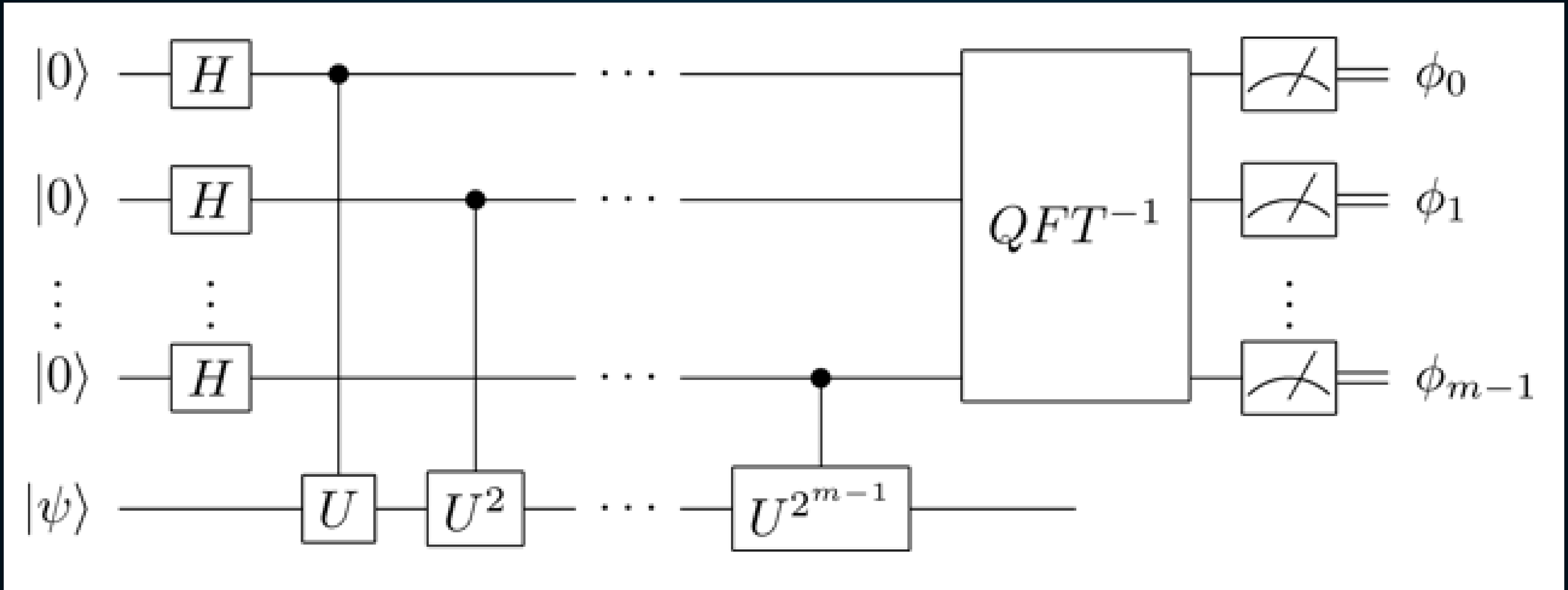


Linear system solving

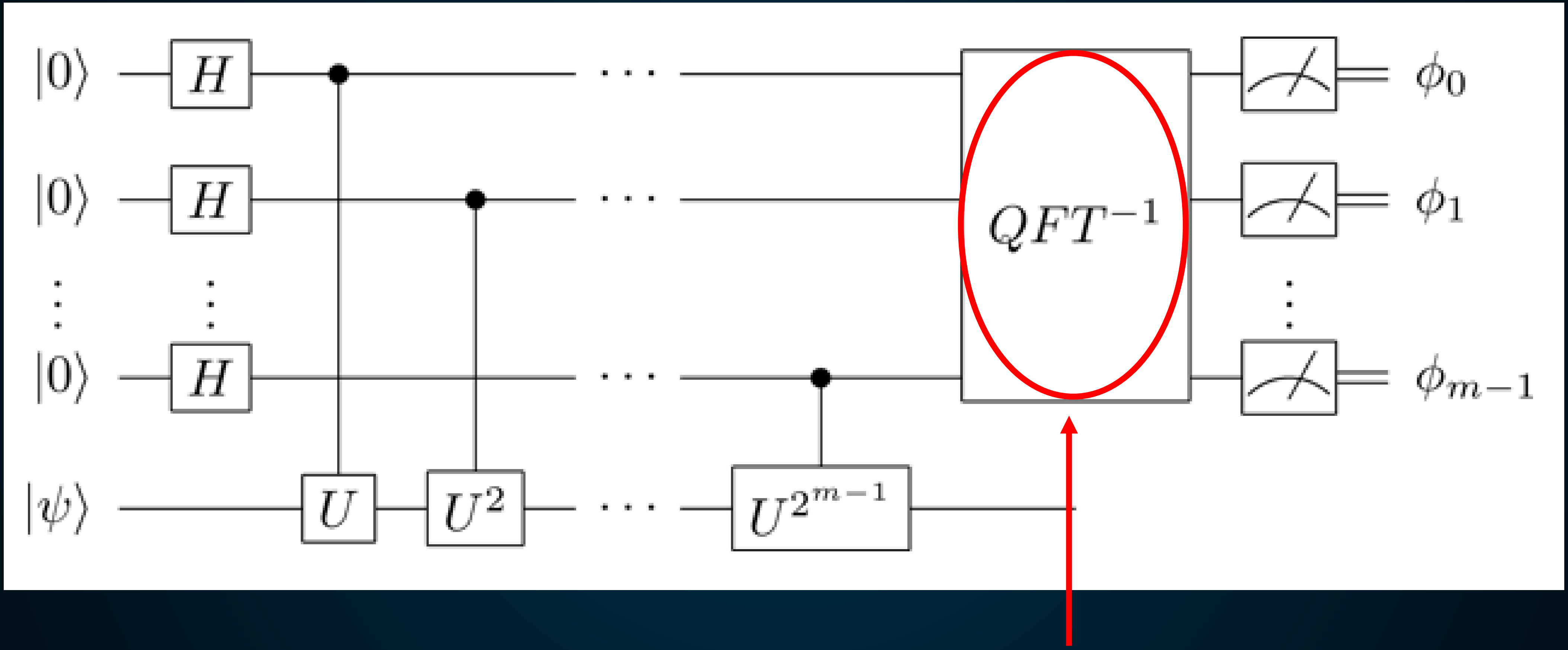


Number factoring

Quantum phase estimation: circuit

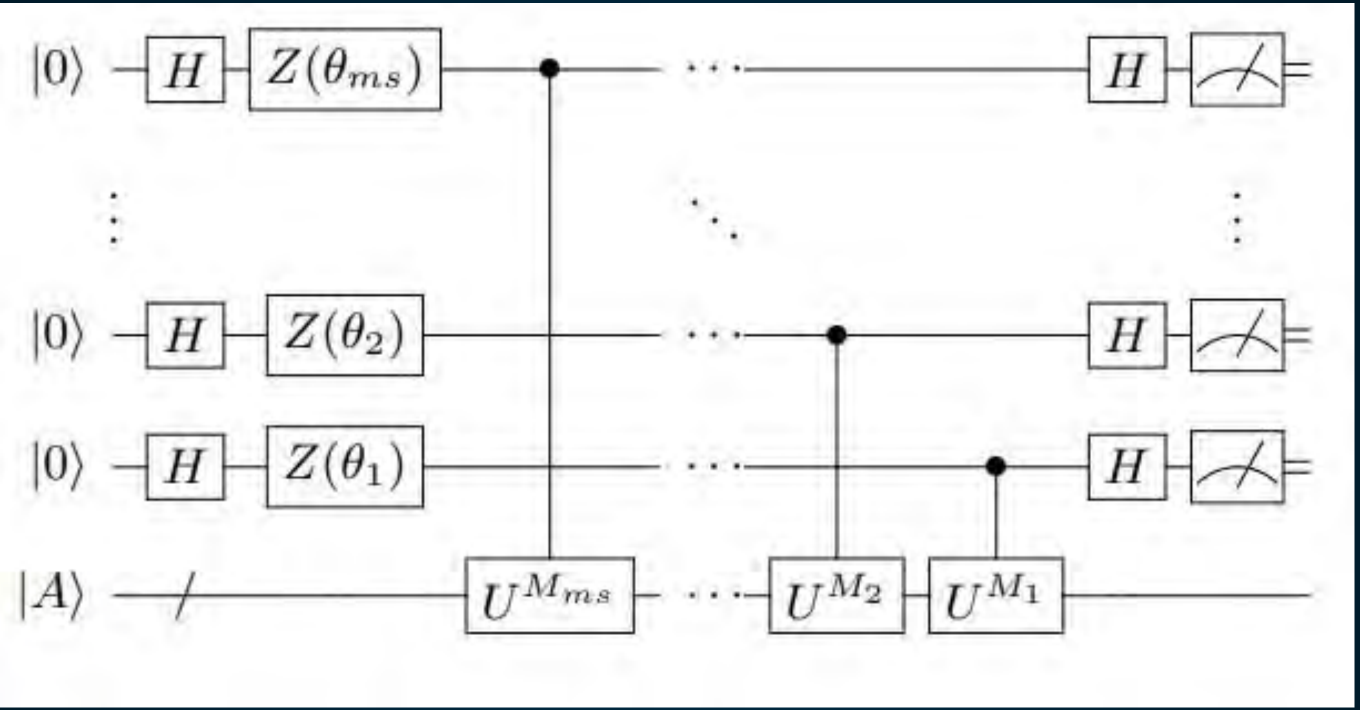


Quantum phase estimation

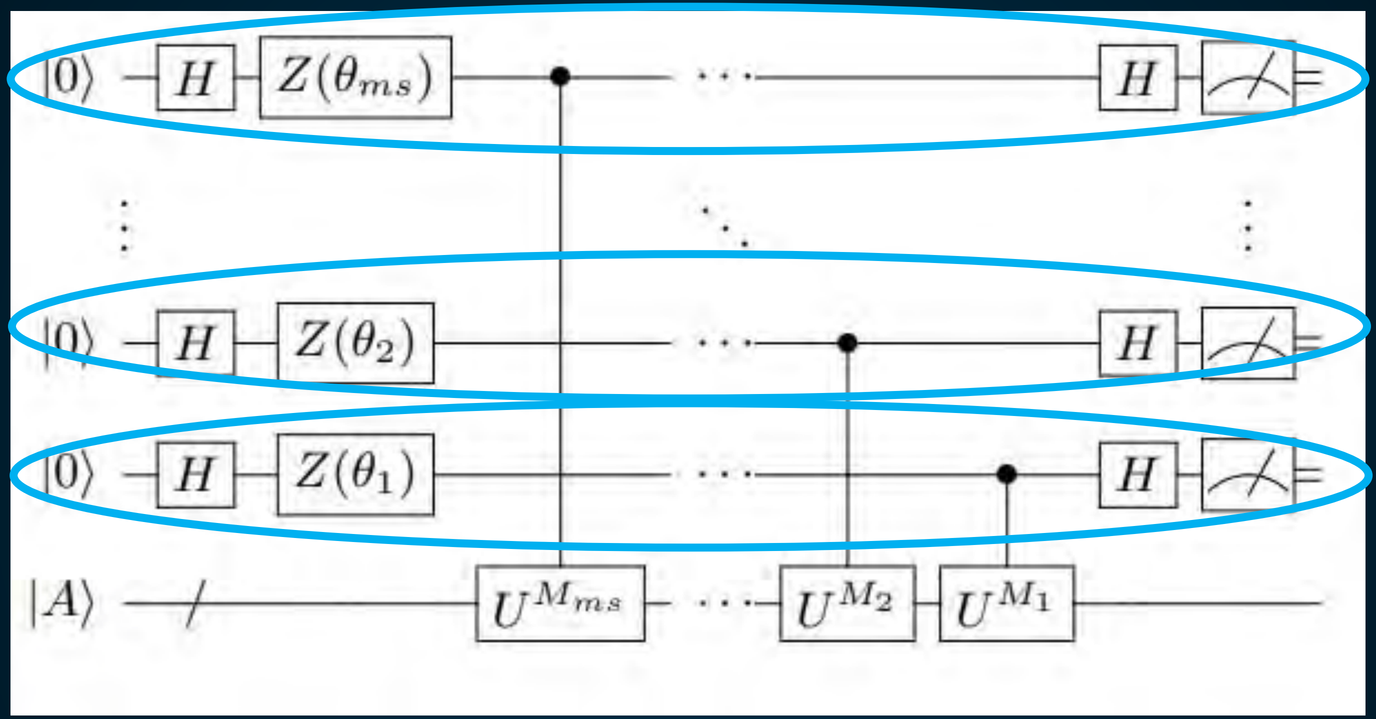


Auxiliary qubits are coupled by the inverse Fourier Transform.

Iterative phase estimation

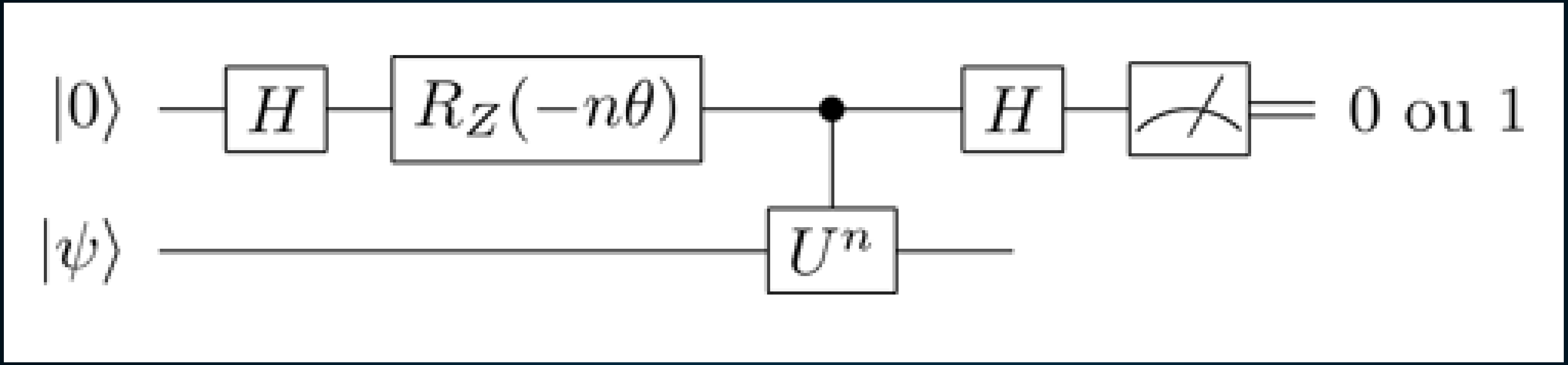


Iterative phase estimation



- Replacing the Fourier transform by a Hadamard transform decouples auxiliary qubits and opens the door to parallelization.
- But using a Hadamard transform requires classical postprocessing to compute the estimated phase.

Iterative phase estimation



- Smaller circuits:

  - only one auxiliary qubit.
  - reduced depth.

-> lower requirement for quantum hardware.

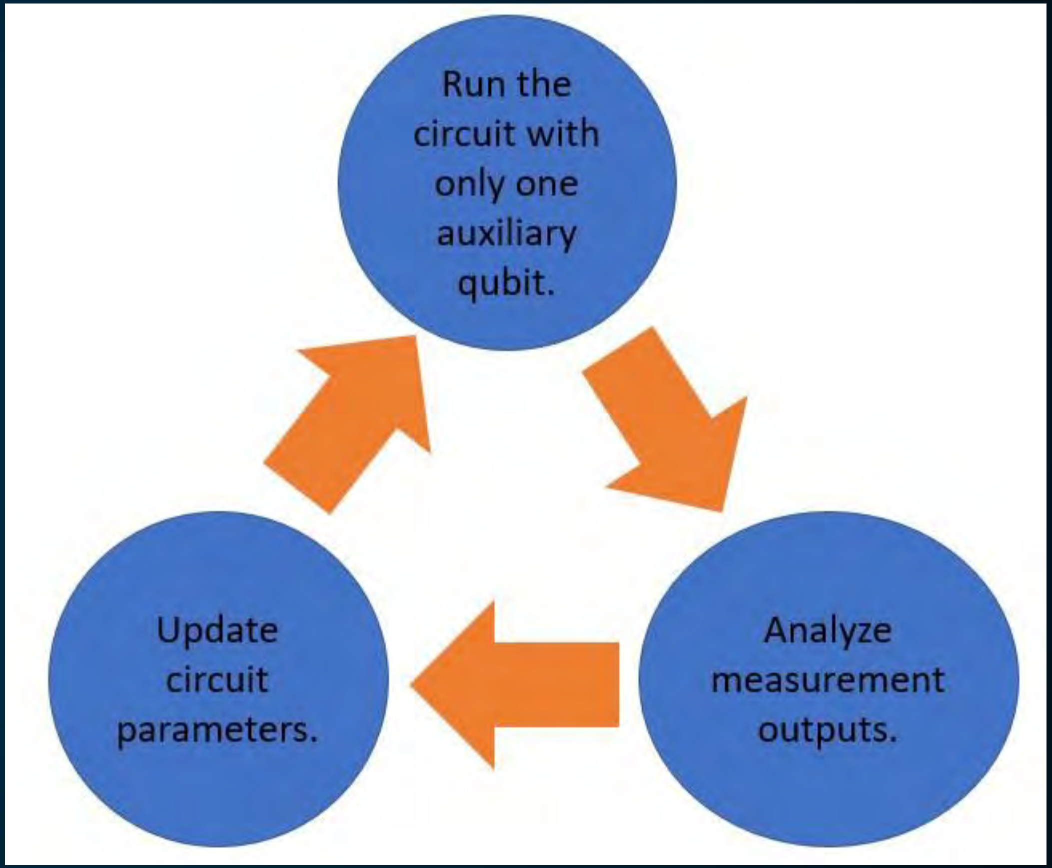
$$\Pr(0) = \cos^2(n\frac{\phi - \theta}{2}).$$

$$\Pr(1) = \sin^2(n\frac{\phi - \theta}{2}).$$

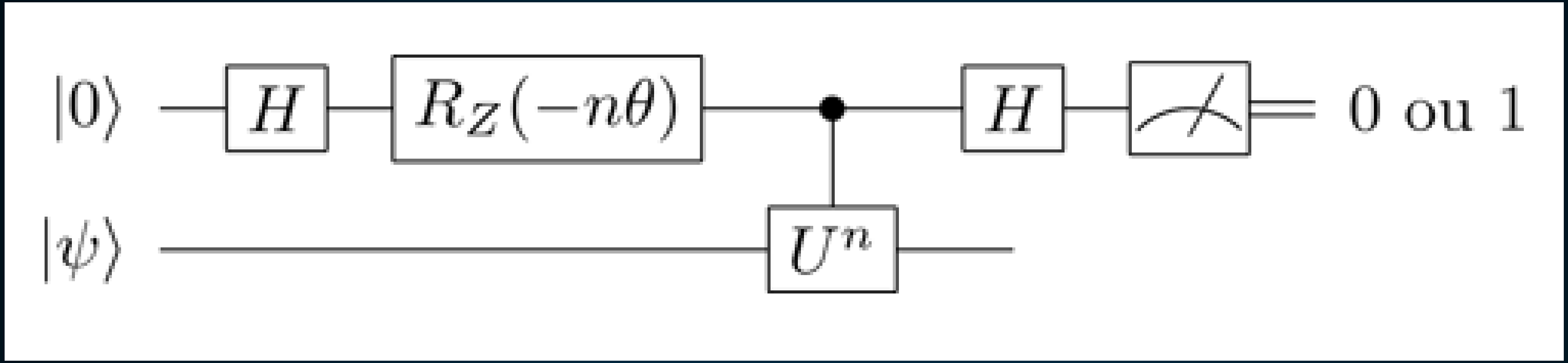
Iterative phase estimation

- Smaller circuits:
  - only one auxiliary qubit.
  - reduced depth.

-> lower requirement for quantum hardware.



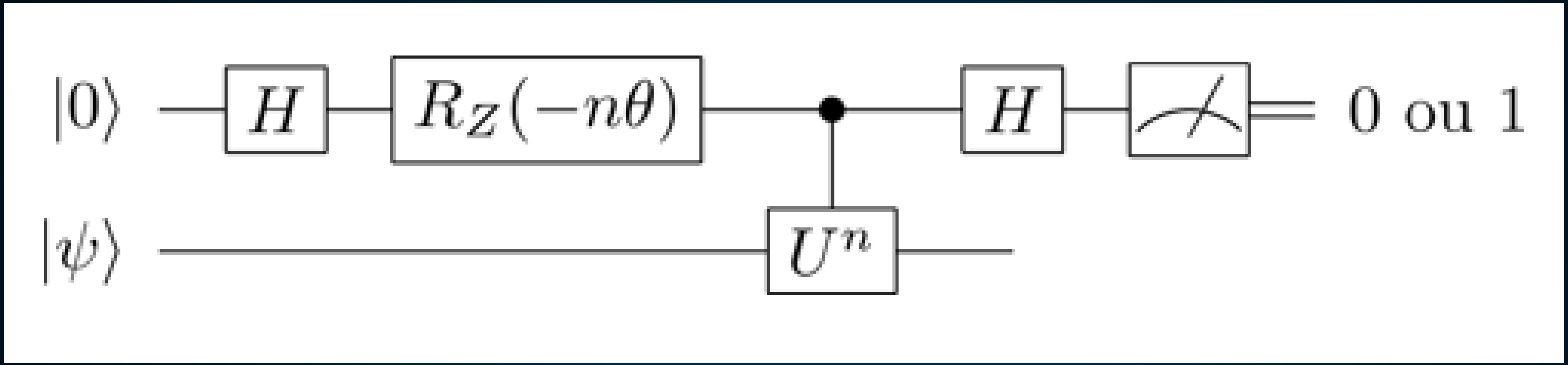
Iterative phase estimation



How should we choose the sequence of values for θ and n ?

-> different tradeoffs lead to different flavors of iterative phase estimation.

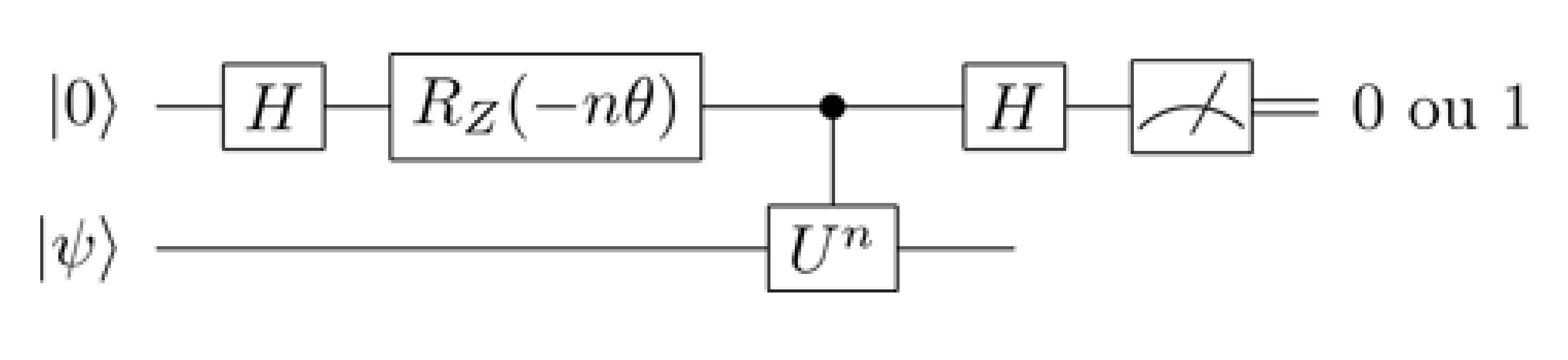
Bayesian phase estimation



- For m bits of precision for the estimated phase ϕ:
  - 2^m classical values are considered.
  - Updating our current knowledge of ϕ requires O(2^m) classical computation.
  - Optimal in the number of queries to U.

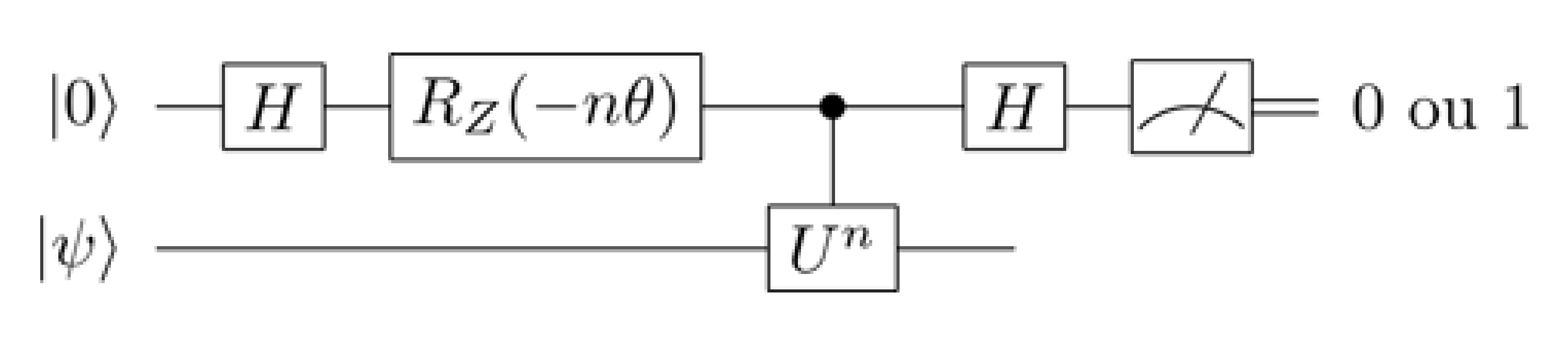
$$\Pr(\phi_a|d) = \frac{\Pr(d|\phi_a)\Pr(\phi_a)}{\Pr(d)}.$$

Robust phase estimation



- Efficient in terms of classical computing.
- More measurements to learn most significant bits of ϕ than to learn least significant bits.
- Robust to state preparation and measurement (SPAM) errors.
- Precision in ϕ scales like $\frac{c}{Q}$. (optimal scaling is $\frac{d}{Q}$ with $d < c$. Q is the number of queries to U.)

Random walk phase estimation



- Estimated ϕ is modeled by a Gaussian distribution (2 parameters only).
- Efficient scaling of the precision of ϕ with respect to:
  - Queries to U.
  - Classical postprocessing.
- Adaptive algorithm : subsequent values of n and θ depend on previous measurement outcomes.

-> requires close integration of quantum and classical computations.

Resources to go further

- Iterative and Bayesian phase estimation:
  - [1304.0741] Faster Phase Estimation (arxiv.org)
  - Quantum/BayesianPhaseEstimation.qs at main · microsoft/Quantum (github.com)

- Robust phase estimation:
  - RobustPhaseEstimation operation - Q# reference - Microsoft Quantum | Microsoft Docs
  - [1502.02677] Robust Calibration of a Universal Single-Qubit Gate-Set via Robust Phase Estimation (arxiv.org)
  - QuantumLibraries/Robust.qs at main · microsoft/QuantumLibraries (github.com)

- Iterative and Bayesian phase estimation:
  - [1309.0876] Hamiltonian Learning and Certification Using Quantum Resources (arxiv.org)
  - Quantum-NC/RandomWalkPhaseEstimation.qs at main · microsoft/Quantum-NC (github.com)

- Blog post (in French): Quantum Computing | Estimation de phase : forces et faiblesses des variantes (vivienlonde.github.io)

Hybrid Quantum Computing Architectures

1. Batch quantum computing

Shor

2. Interactive batch quantum computing

VQE

3. Integrated quantum computing

Adaptive phase estimation

4. High performance quantum computing

Rejection sampling

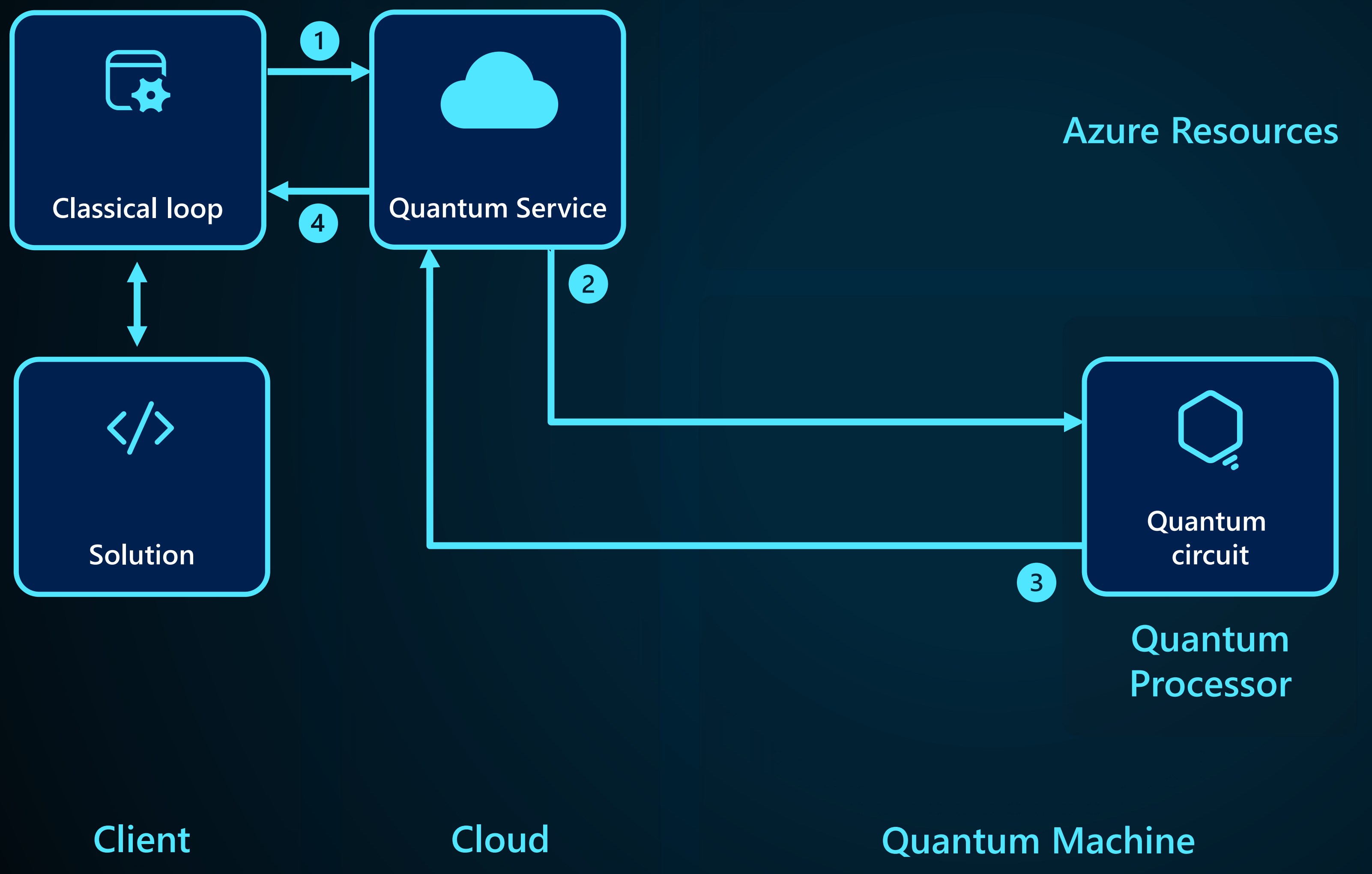
5. Distributed quantum computing

Materials modeling

Increasingly tighter and more complex interaction between quantum & classical processing

Level 1: Batch quantum computing

Quantum circuit with classical pre- and post-processing

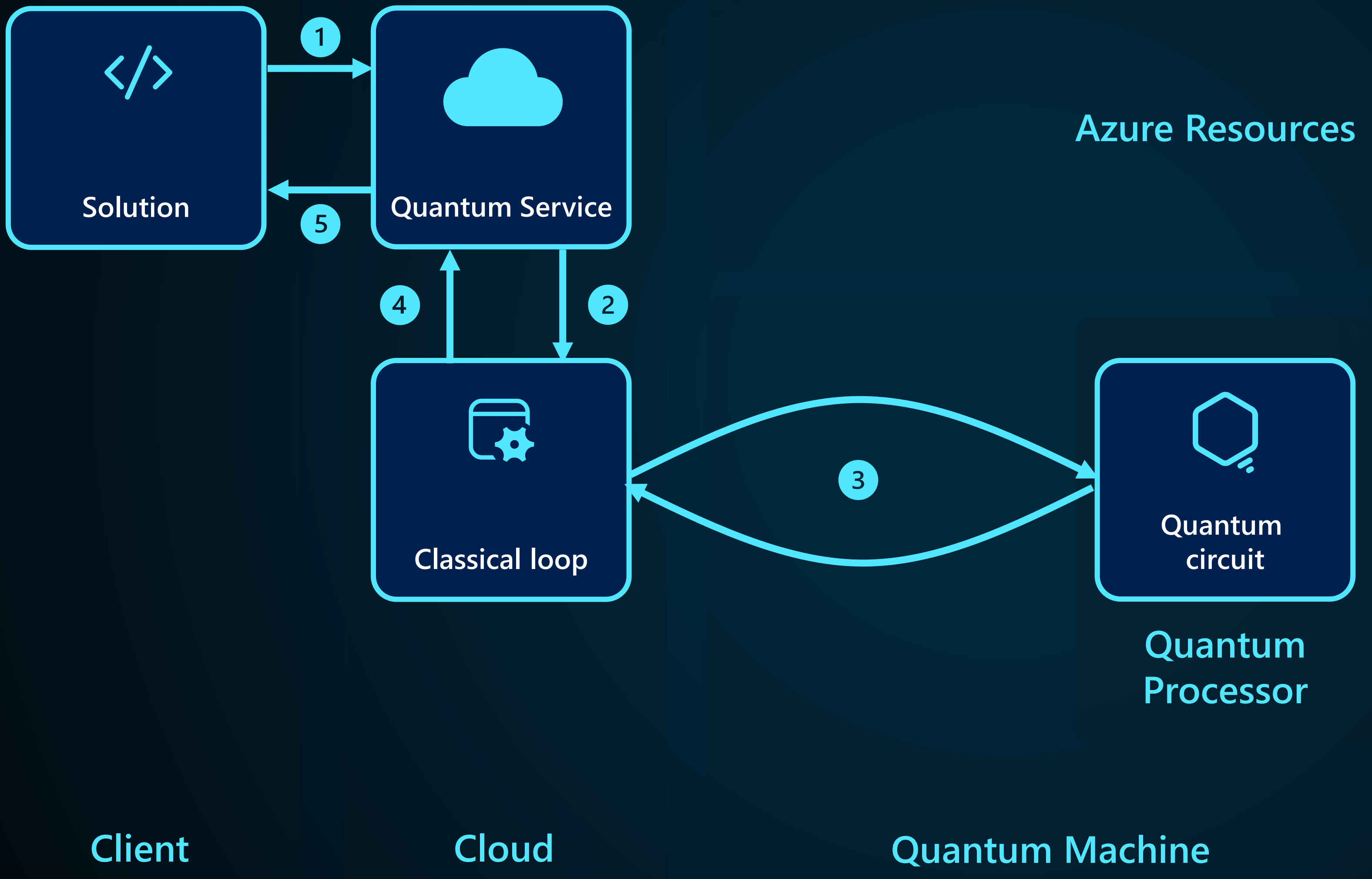


Examples

- Shor’s algorithm (cryptanalysis)
- Simple quantum phase estimation

Level 2: Interactive batch quantum computing

Parameterized quantum circuit in a classical driver loop that runs in the cloud



3 Prioritized loop

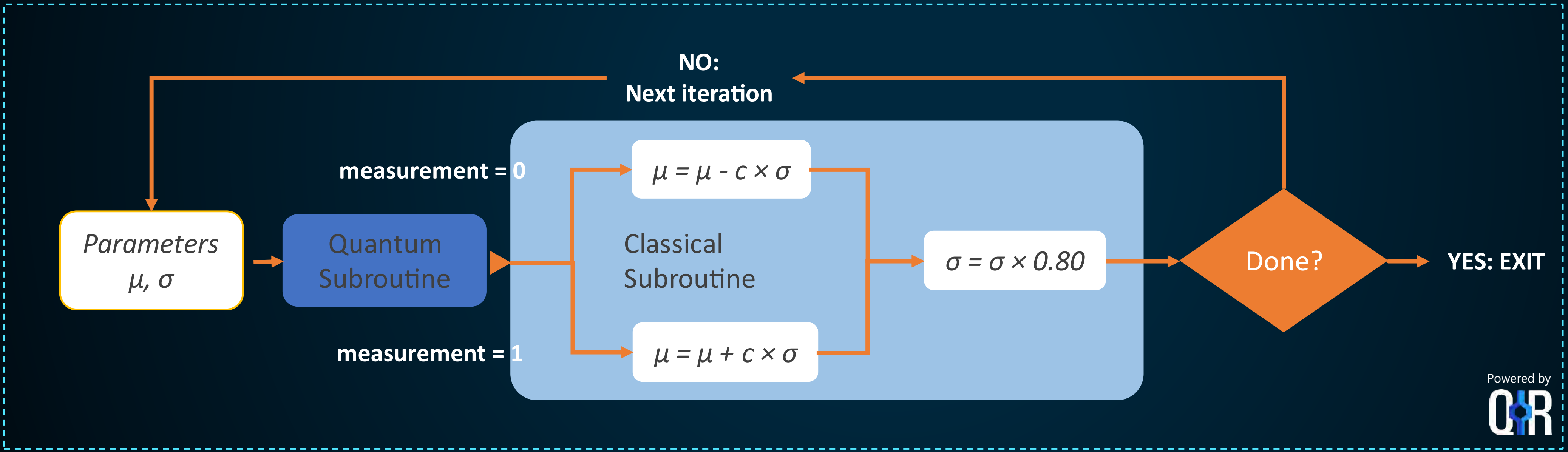
Examples

- Variational quantum eigensolvers (VQE)
- Quantum approximate optimization algorithms (QAOA)

3. Integrated quantum computing

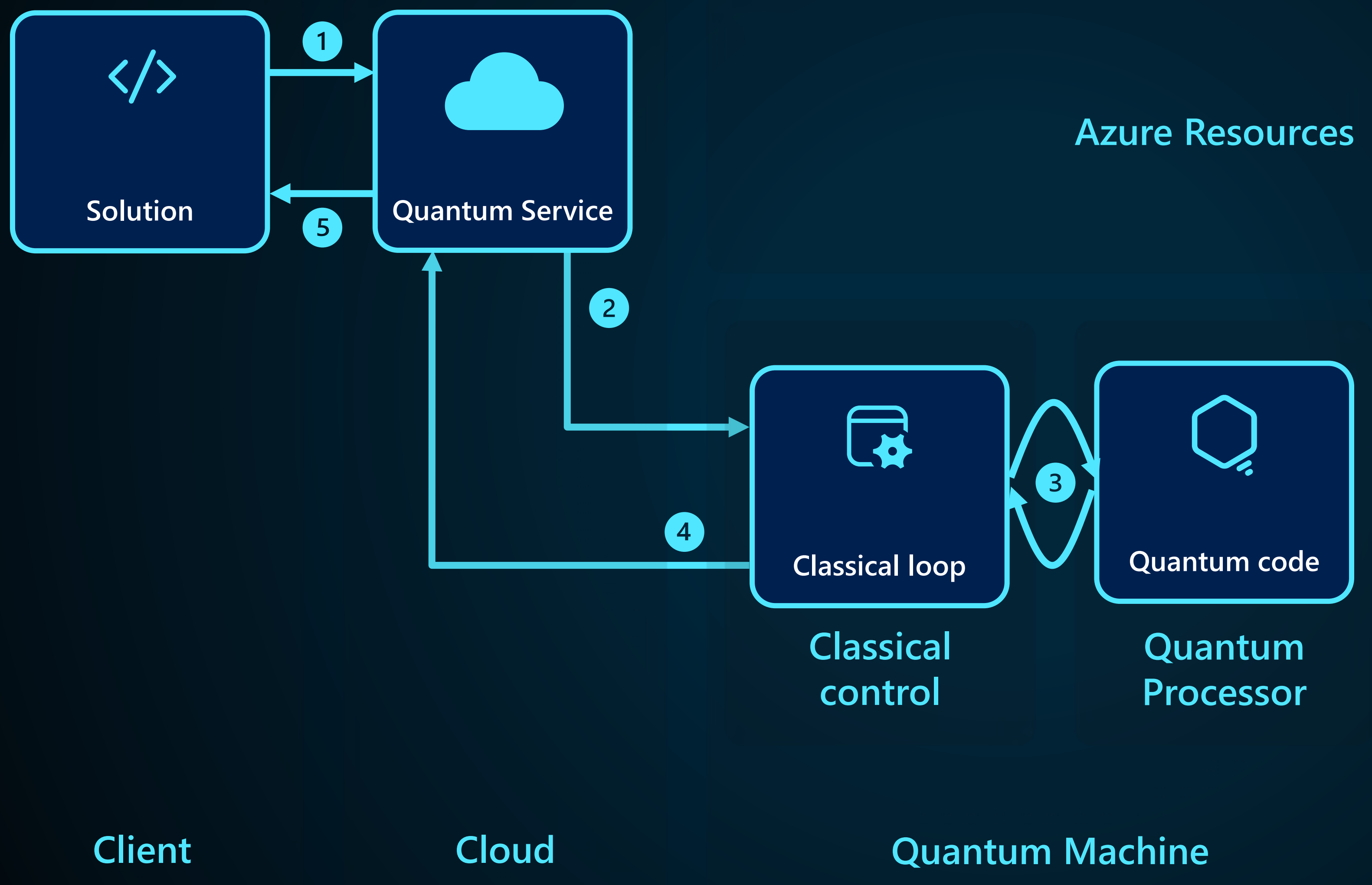
In the quantum machine:

- Arbitrary control flow,
- Classical computations by back-end while physical qubits are alive



Level 3: Integrated quantum computing

Coherent quantum coroutine with a classical driver loop



③ Physical qubit remains alive

Limited classical control

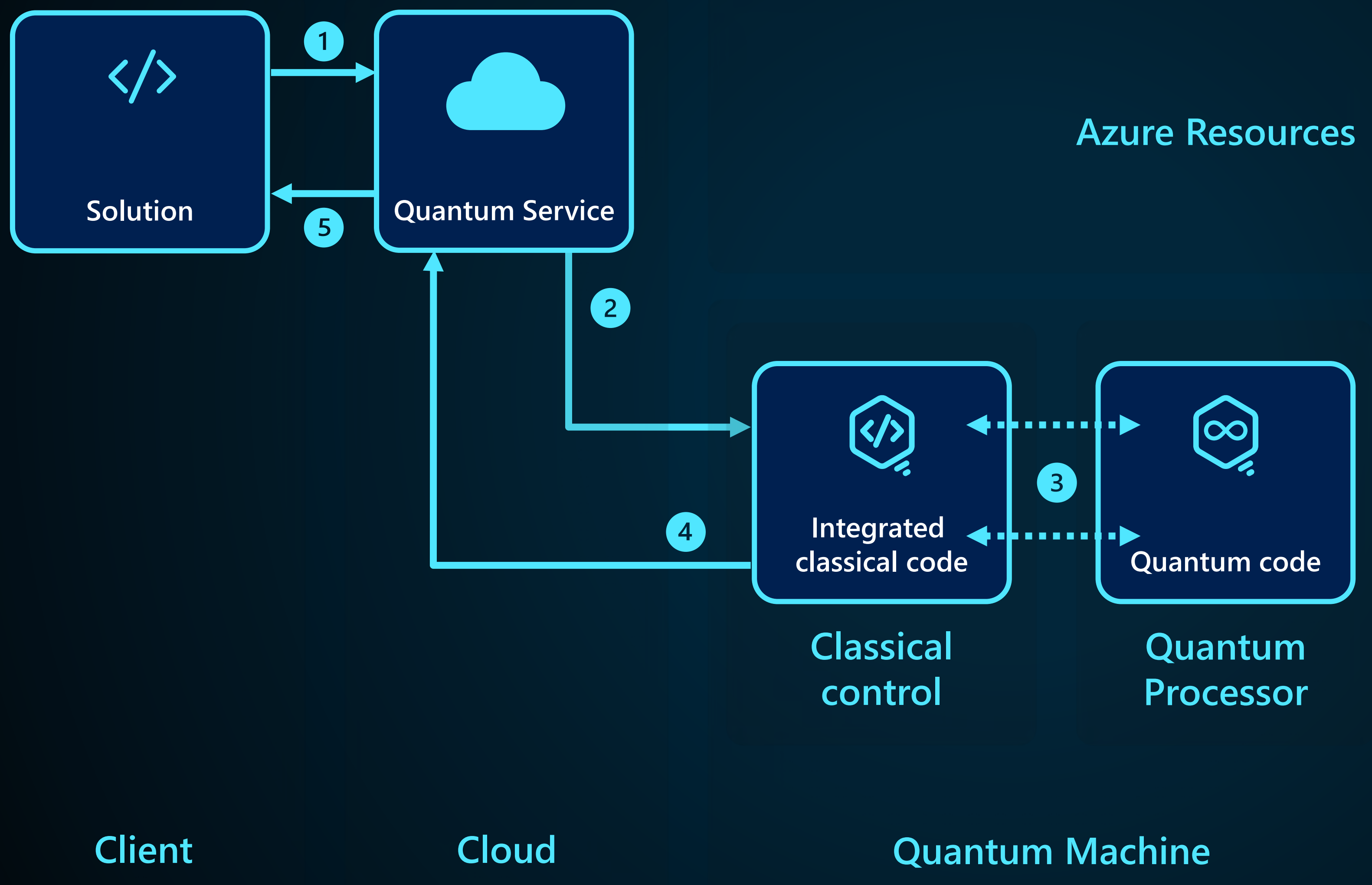
Example

Parameterized quantum coroutine:

- Adaptive phase estimation techniques such as random walk PE or Bayesian PE
- Error Correction

Level 4: High Performance quantum computing

Full classical compute next to QPU



3 Logical qubit with indefinite lifetime

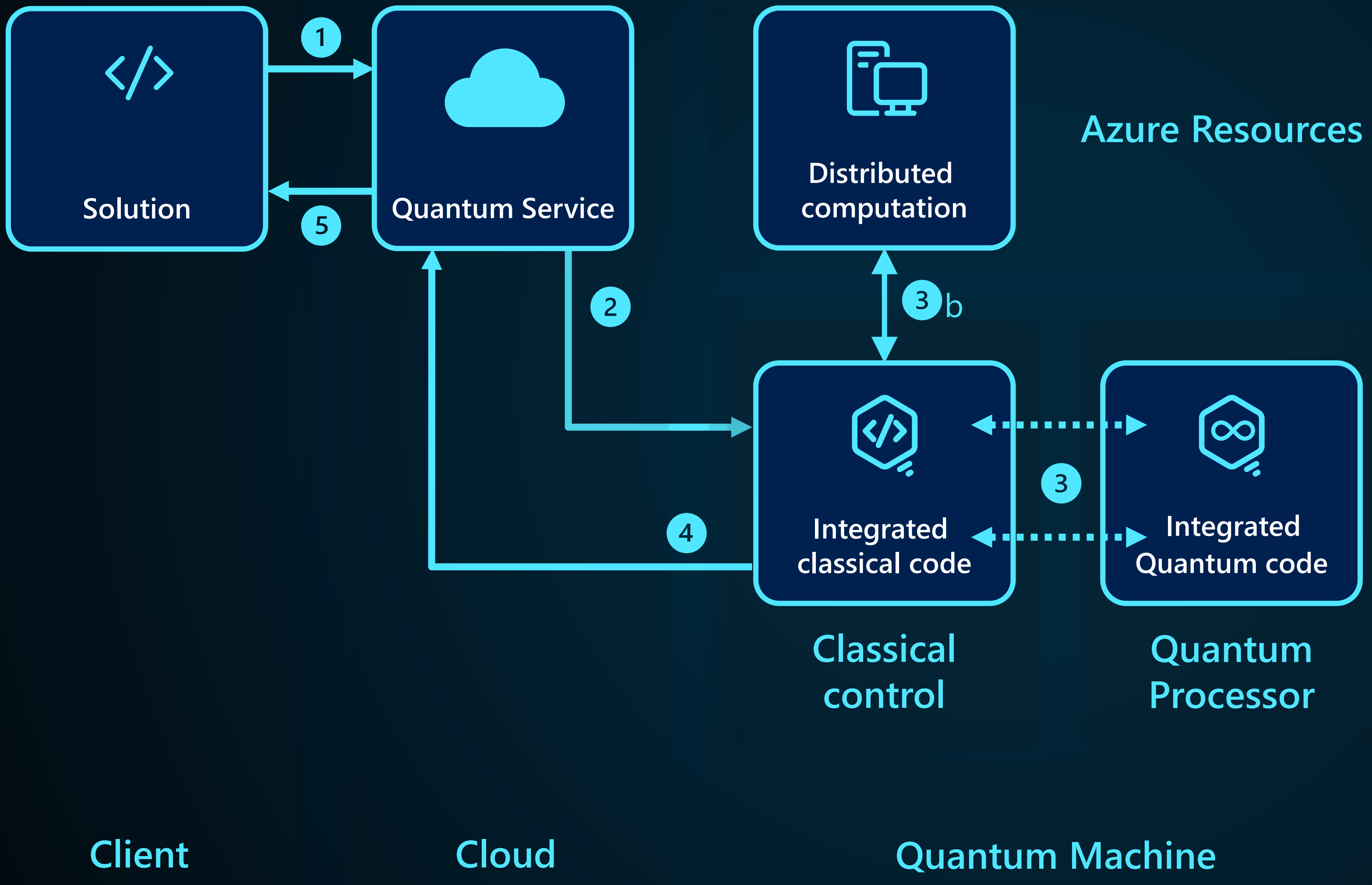
Full classical control

Examples

- Repeat until success gadgets
- Rejection sampling

Level 5: Distributed quantum computing

Distributed quantum/classical processing



3b Real-time cloud processing

Examples:

- Complex materials modelling
- Catalysis

QIR: Quantum Intermediate Representation

- Intermediate representation that expresses quantum and classical computations together.
- Frontends:
  - Qiskit
  - Q#
  - Cirq
  - …
- Backends:
  - Superconducting qubits
  - Photonic qubits
  - Trapped ions and cold atom qubits
  - Topological qubits
  - Simulators
  - Resource estimators
  - …

Q2B2021 presentation: Q2B 2021 | Empowering Heterogeneous Quantum Computing with QIR | Panel - YouTube

QIR: Quantum Intermediate Representation

- Intermediate representation that expresses quantum and classical computations together.

- Frontends:
  - Qiskit
  - Q#
  - Cirq
  - …

- Optimizations :
  - Of a hybrid program
  - Hardware agnostic first
  - Hardware specific then
  - Reuses LLVM tooling

- Backends:
  - Superconducting qubits
  - Photonic qubits
  - Trapped ions and cold atom qubits
  - Topological qubits
  - Simulators
  - Resource estimators
  - …

Q2B2021 presentation: [Q2B 2021 | Empowering Heterogeneous Quantum Computing with QIR | Panel - YouTube]

QIR: Quantum Intermediate Representation

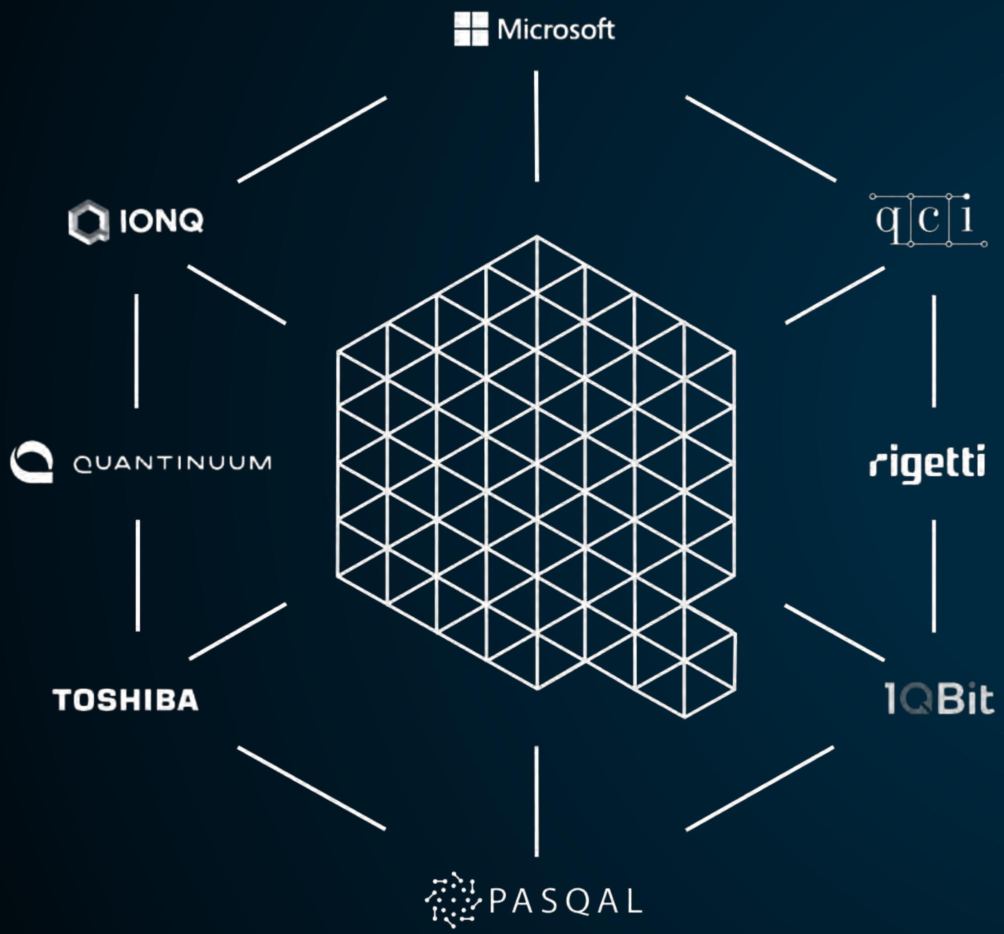
- Intermediate representation that expresses quantum and classical computations together.
- Frontends:
  - Qiskit
  - Q#
  - Cirq
  - …
- Backends:
  - Superconducting qubits
  - Photonic qubits
  - Trapped ions and cold atom qubits
  - Topological qubits
  - Simulators
  - Resource estimators
  - …
- Optimizations :
  - Of a hybrid program
  - Hardware agnostic first
  - Hardware specific then
  - Reuses LLVM tooling
- Joint effort :
  - Quantinuum
  - QCI
  - Rigetti
  - Microsoft
  - Oak Ridge National Laboratory
  - Nvidia
  - …

Q2B2021 presentation: [Q2B 2021 | Empowering Heterogeneous Quantum Computing with QIR | Panel - YouTube]

Microsoft

Azure Quantum

The High-Performance hybrid QC cloud platform



- ✓ Unique high-performance hybrid capabilities
- ✓ Start path to fault-tolerance with Resource Estimation
- ✓ Choice of hardware from a single cloud service
- ✓ Q#, Qiskit and Cirq support
- ✓ $500 free credits for all users, larger grants up to $10,000
- ✓ Write once, run on multiple platforms

Microsoft

Thank you

quantum computing energetics, from NISQ to FTQC

alexia auffèves
Director of CNRS MajuLab, Singapore,
QEI cofounder
alexia.auffeves@cnrs.fr

olivier ezratty
consultant and author,
QEI cofounder
olivier@oezratty.net

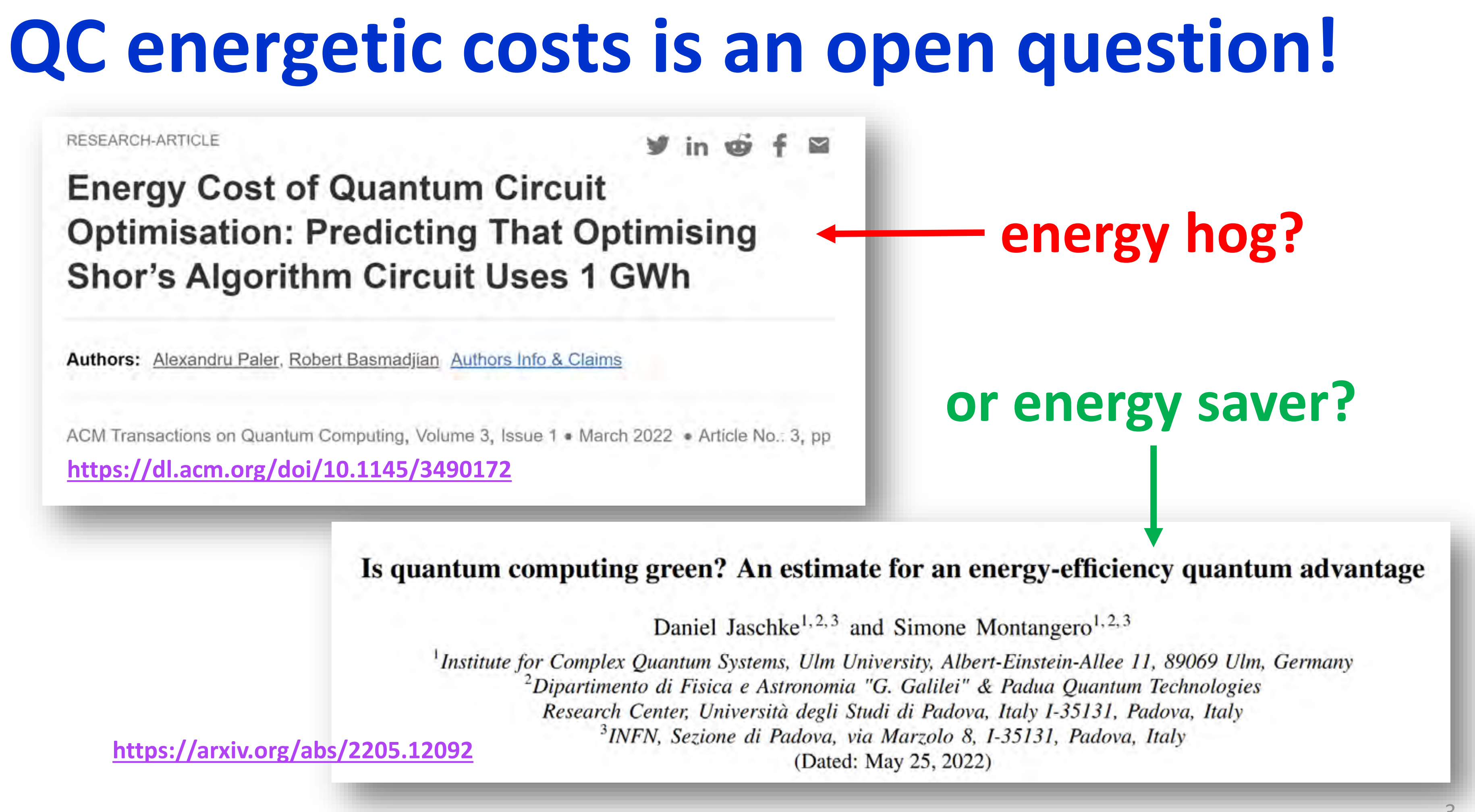
théau péronnin
CEO and cofounder of Alice&Bob
theau.peronnin@alice-bob.com

EDF, Palaiseau, January 11th, 2023

QEI proposed methodology

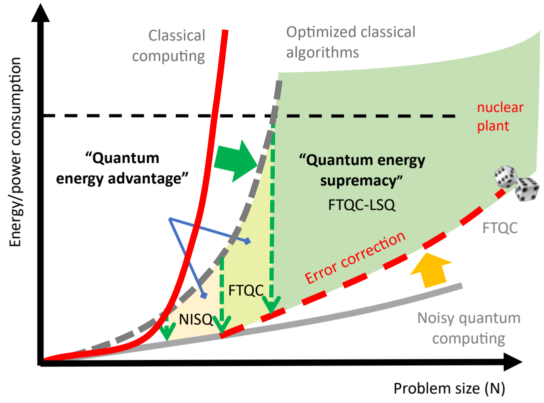
FTQC perspective

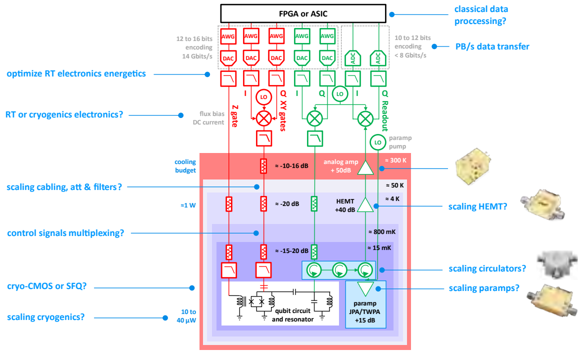
NISQ perspective

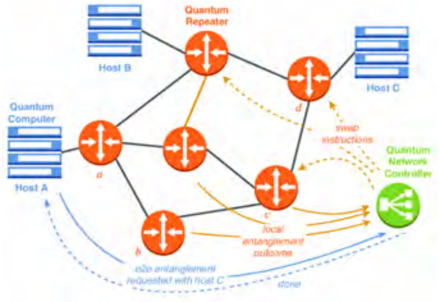

QC energetic costs is an open question!
RESEARCH-ARTICLE
Energy Cost of Quantum Circuit Optimisation: Predicting That Optimising Shor's Algorithm Circuit Uses 1 GWh
Authors: Alexandru Paler, Robert Basmadjian Authors Info & Claims
ACM Transactions on Quantum Computing, Volume 3, Issue 1 • March 2022 • Article No.: 3, pp
https://dl.acm.org/doi/10.1145/3490172
energy hog?
or energy saver?
Is quantum computing green? An estimate for an energy-efficiency quantum advantage
Daniel Jaschke1,2,3 and Simone Montangero1,2,3
1Institute for Complex Quantum Systems, Ulm University, Albert-Einstein-Allee 11, 89069 Ulm, Germany
2Dipartimento di Fisica e Astronomia "G. Galilei" & Padua Quantum Technologies Research Center, Università degli Studi di Padova, Italy I-35131, Padova, Italy
3INFN, Sezione di Padova, via Marzolo 8, I-35131, Padova, Italy
(Dated: May 25, 2022)
https://arxiv.org/abs/2205.12092

#QEI
the quantum energy initiative

key scientific questions







is there a **quantum energy advantage** vs classical computing as quantum processors scale up?

how different is it from the **quantum computational advantage**?

what is the fundamental **minimal energetic cost** of quantum computing?

how to **avoid energetic dead-ends** on the road to LSQ?

will **other quantum technologies** present energetic challenges?

quantum communications and sensors

Quantum Energy Initiative vision paper

PRX QUANTUM

a Physical Review journal

Highlights Recent Accepted Authors Referees Search About Scope Staff

Perspective Open Access

Quantum Technologies Need a Quantum Energy Initiative

Alexia Auffèves

PRX Quantum **3**, 020101 – Published 1 June 2022

Article References No Citing Articles PDF HTML Export Citation

ABSTRACT

Quantum technologies are currently the object of high expectations from governments and private companies, as they hold the promise to shape safer and faster ways to extract, exchange, and treat information. However, despite its major potential impact for industry and society, the question of their energetic footprint has remained in a blind spot of current deployment strategies. In this Perspective, I argue that quantum technologies must urgently plan for the creation and structuration of a transverse quantum energy initiative, connecting quantum thermodynamics, quantum information science, quantum physics, and engineering. Such an initiative is the only path towards energy-efficient, sustainable quantum technologies, and to possibly bring out an energetic quantum advantage.

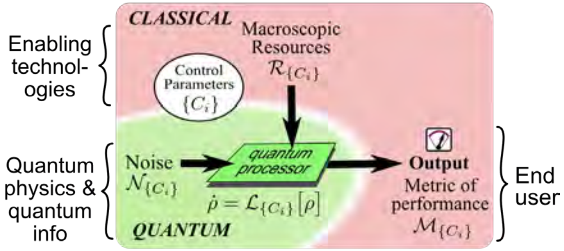
https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.3.020101

#QEI goals & missions

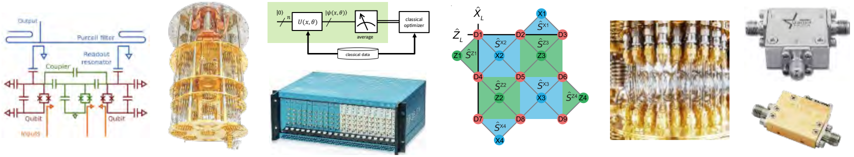
the quantum energy initiative

create a **worldwide community** working on quantum technologies energetics associating fundamental research and industry vendors.

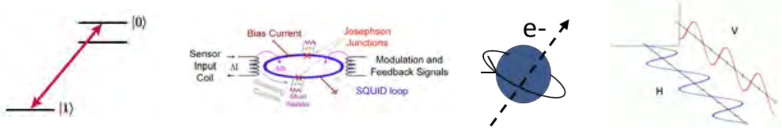
(a) Metric-Noise-Resource (MNR) methodology for the full-stack of a quantum computer



propose **optimization methodologies, frameworks** and **benchmarks** for quantum technologies, enabling technologies and software engineering



create a new **transversal line of research** and **collaborative projects**.

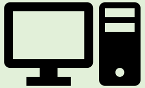


cover **all qubit types**, **programming paradigms**, and other quantum technologies (communications, sensing)

Generic definition of resource efficiency

Machine efficiency

$$\boldsymbol{\eta = \frac{M}{R}}$$



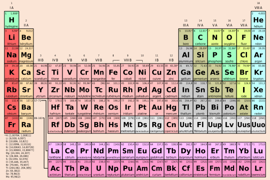
$e.g.\quad \eta = \frac{FLOPs}{W} \qquad \eta = \frac{km}{l}$

Target result: operations, distance, etc.

Performance metric

Physical resource cost

Materials

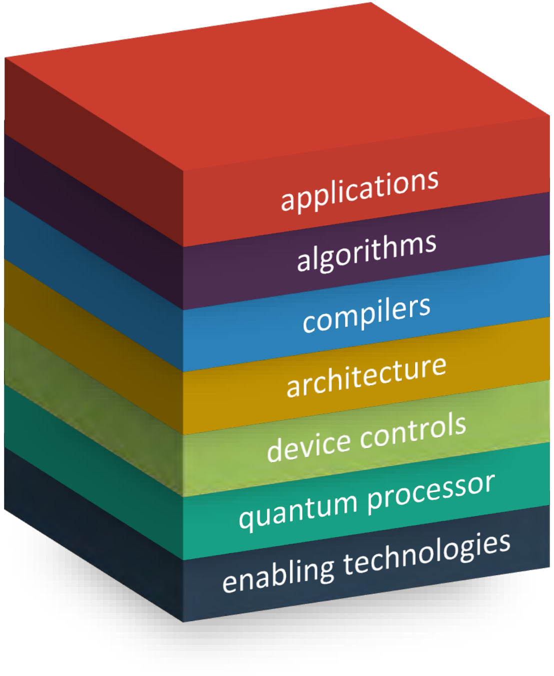


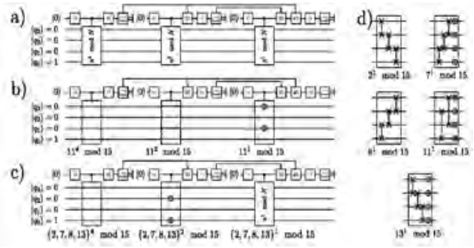
Energy or power

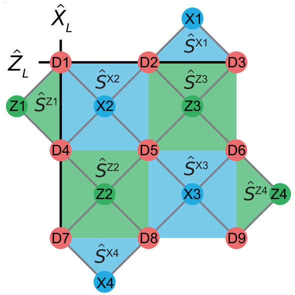
A full-stack quantum computer

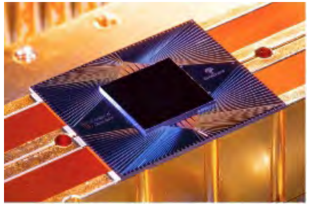
Hardware agnostic

Hardware dependent



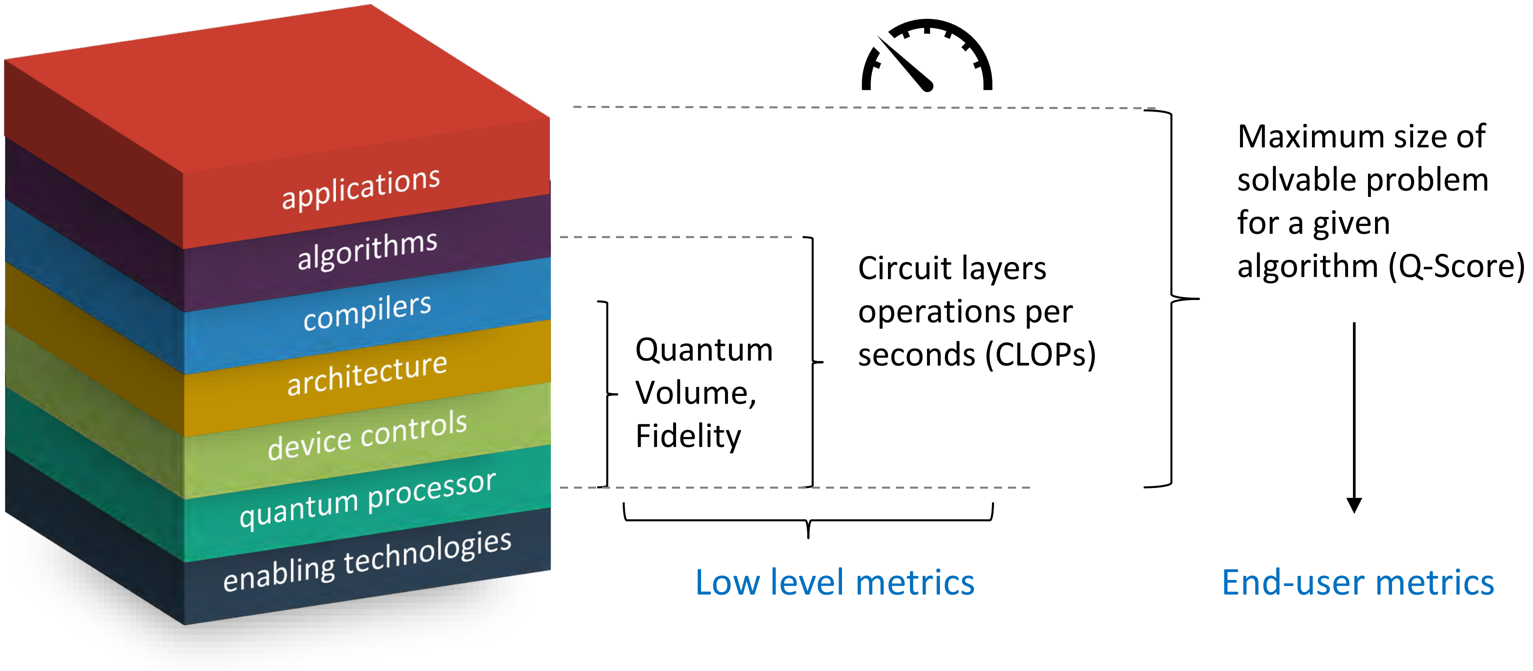






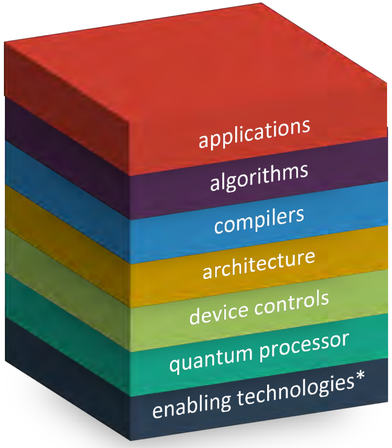
Metrics of performance

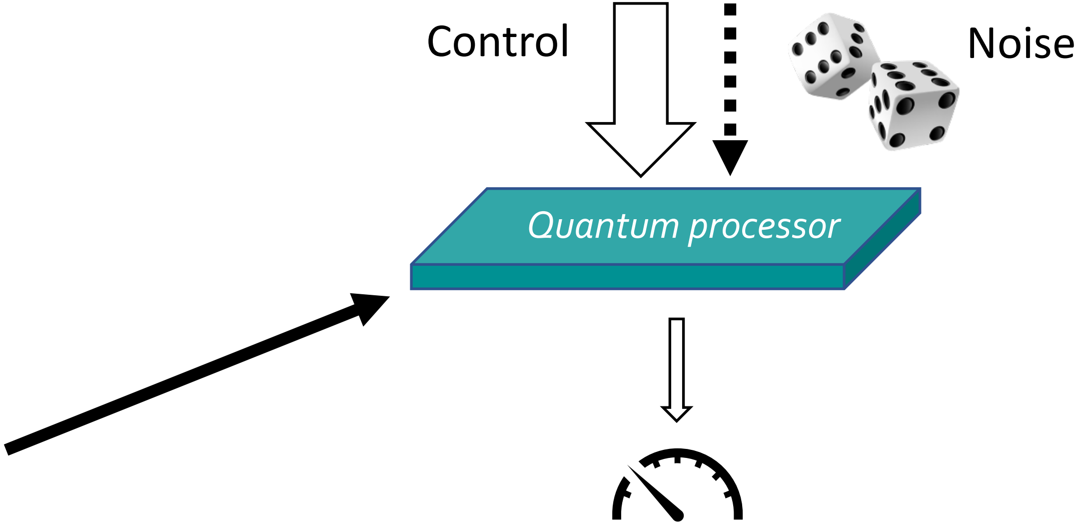
- Intense work nowadays to define metrics of performances & standards
- Performances can be defined at low level or at end-user level (w.r. to applications)



Metrics of performance

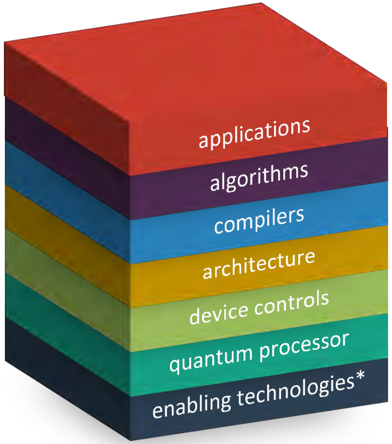
➢ All metrics depend on the level of control reached over the noisy processor
➢ Fight [control: noise] @**quantum level of the stack**





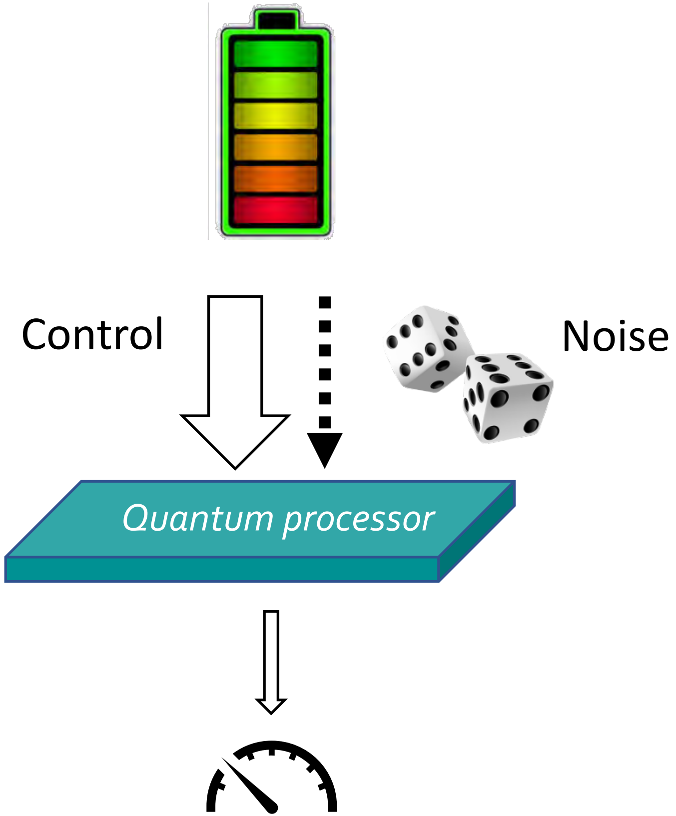
Resource costs

- ➢ Good performances mandate resources **@ each level of the stack**
- ➢ Hybrid efficiencies $\eta = \frac{M}{R}$



- **@Quantum level:** fundamental bounds
- Involve quantum control, reservoir engineering...

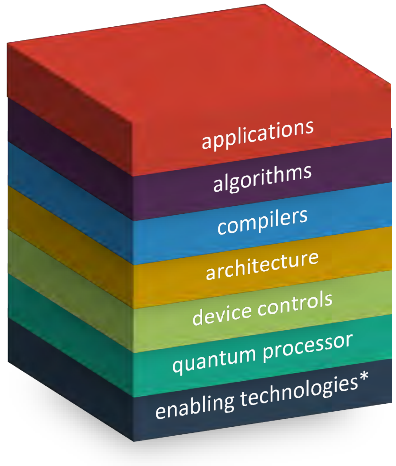
➢ *Must connect to hardware-agnostic levels*

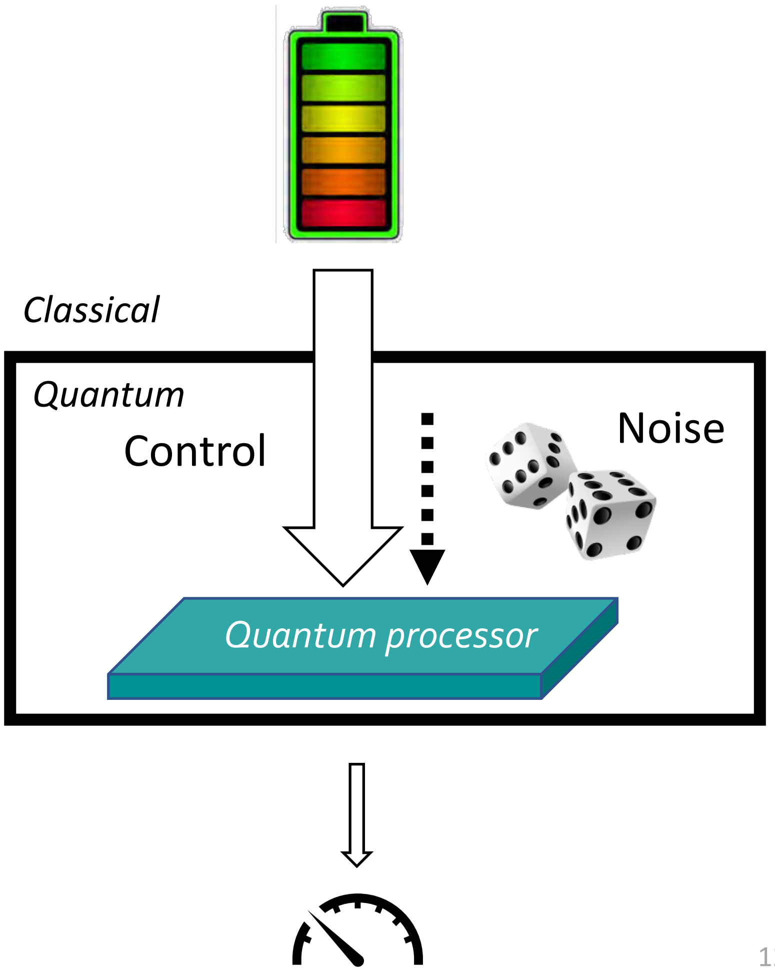


Resource costs

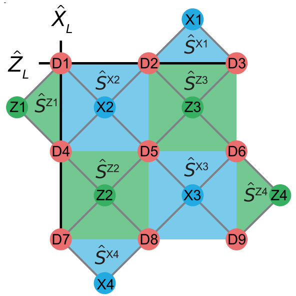
@All hardware-dependent levels:

- Cost of trapping and controlling a Schrödinger cat
- Macroscopic resources spent by enabling tech and control chains
- *Must connect quantum and macroscopic levels*



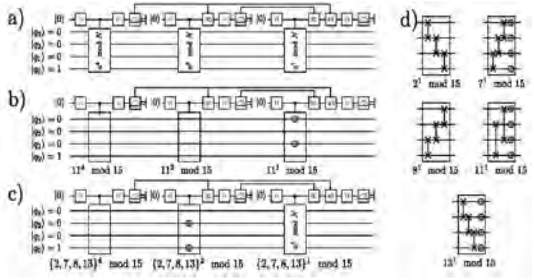


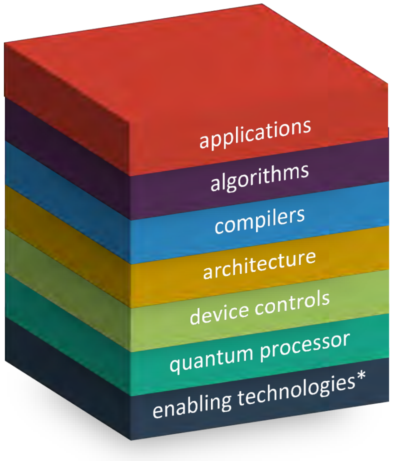
Resource costs



@hardware-agnostic levels:

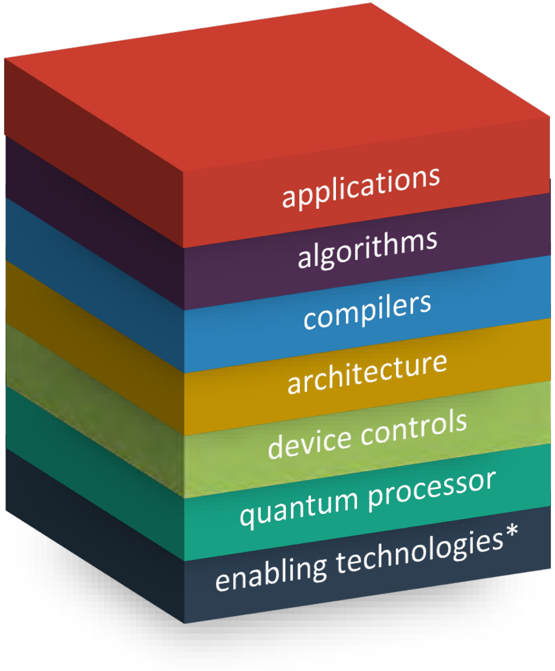
- Number of logical operations and qubits...
- Number of physical qubits per logical qubit, code connectivity, complexity of encoding/decoding operations...



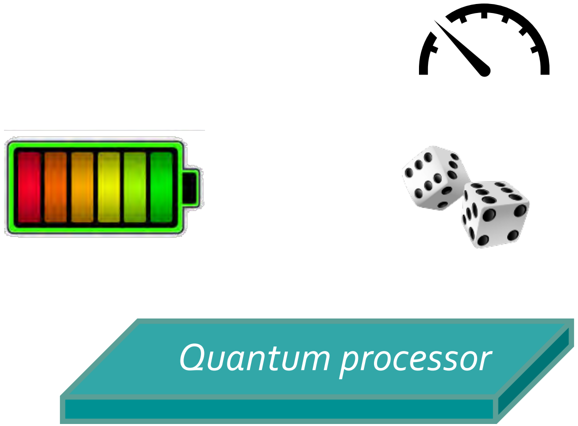


- ➢ Currently: architecture optimizations for fixed hardware noise properties
- ➢ *But the noise depends on the circuit architecture*
- ➢ *Must connect to hardware-dependent levels*

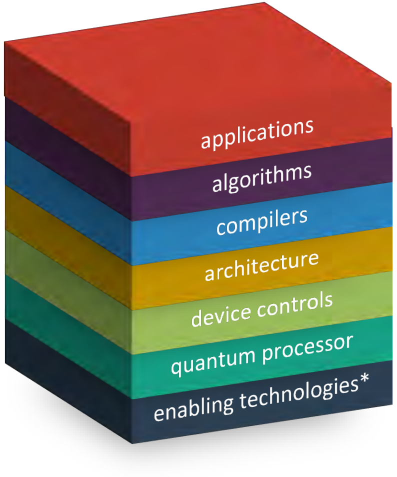
Need for a holistic approach



- Must connect inputs from all levels of the stack to optimize quantum computing energy efficiencies
- Need for common language and methodology



Metric-Noise-Resource (MNR) methodology



- Specify the control parameters $\{C_i\}$ and the metric of performance M
- Model the processor dynamics $\dot{\rho} = L_{\{C_i\}}[\rho]$ => get the metric $M(\{C_i\})$
- Model the resource cost $R(\{C_i\})$

- Set a target metric $M(\{C_i\}) = M_0$ => Implicit relation on $\{C_i\}$
- Minimize the resource cost $R(\{C_i\})$ under the constraint $M = M_0$
- Maximize the resource efficiency $\eta(M_0) = \frac{M_0}{R_{min}(M_0)}$

#QEI
the quantum energy initiative

first partners

research

- PCQT – Paris Centre for Quantum Technologies
- MajuLab
- CQT – Centre for Quantum Technologies, National University of Singapore
- WACQT – Wallenberg Centre for Quantum Technology
- Quantum Information Center – Alliance Sorbonne Université

industry

- parityqc
- Atos
- Qilimanjaro
- PASQAL
- Alice & Bob
- ATEM – customized connectivity
- aQuantum® Software Engineering
- IQM
- siquance
- C12 Quantum
- Quandela
- Radiall™
- Viqthor

HPC service providers

- France Hybrid HPC Quantum Initiative
- CSC

industry associations

- llq

#QEI proposed structure and overview

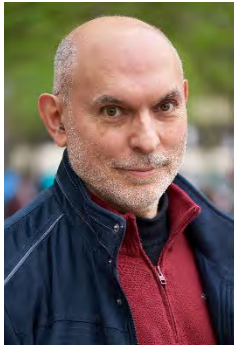
the quantum energy initiative

community buildup	**QEI board creation** **more QEI partners**	**QEI video series** **QEI research inventory**	**QEI workshops** **coordinate collaborative projects RFPs responses**
creating collaborative research	**Quantum physics** **Quantum thermodynamics**	**Quantum information** (algorithms, compilers, software engineering, error correction)	**Qubit types** **Quantum matter**
	Quantum computing	**Quantum communications**	**Quantum sensing**
	Enabling technologies (cryogeny, electronics, cabling, …)		
standardization	**Quantum energetics standardization** CENELEC, IEEE, ISO, national standards bodies		**Benchmarks** QG500

#QEI
the quantum energy initiative

contacts

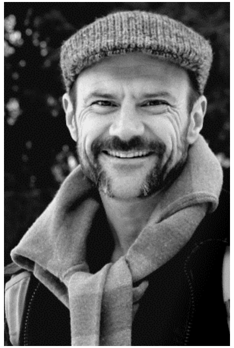
contact@quantum-energy-initiative.org



Olivier Ezratty
Quantum Technologies
Consultant and Author
olivier@oezratty.net

Alexia Auffèves
CNRS Research Director
MajuLab, Singapore
alexia.auffeves@neel.cnrs.fr

Janine Splettstoesser
Professor
Chalmers University
janines@chalmers.se



Robert Whitney
Researcher
CNRS LPMMC Grenoble
robert.whitney@lpmmc.cnrs.fr

QEI proposed methodology

FTQC perspective

NISQ perspective

Optimizing resource efficiencies for scalable full-stack quantum computers

Marco Fellous-Asiani,[1,2,*] Jing Hao Chai,[2,3] Yvain Thonnart,[4] Hui Khoon Ng,[5,3,6,†] Robert S. Whitney,[7,‡] and Alexia Auffèves[2,6,§]

[1] *Centre for Quantum Optical Technologies, Centre of New Technologies, University of Warsaw, Banacha 2c, 02-097 Warsaw, Poland*
[2] *Université Grenoble Alpes, CNRS, Grenoble INP, Institut Néel, 38000 Grenoble, France*
[3] *Centre for Quantum Technologies, National University of Singapore, Singapore*
[4] *Université Grenoble Alpes, CEA-LIST, F-38000 Grenoble, France*
[5] *Yale-NUS College, Singapore*
[6] *MajuLab, International Joint Research Unit UMI 3654, CNRS, Université Côte d'Azur, Sorbonne Université, National University of Singapore, Nanyang Technological University, Singapore*
[7] *Université Grenoble Alpes, CNRS, LPMMC, 38000 Grenoble, France.*

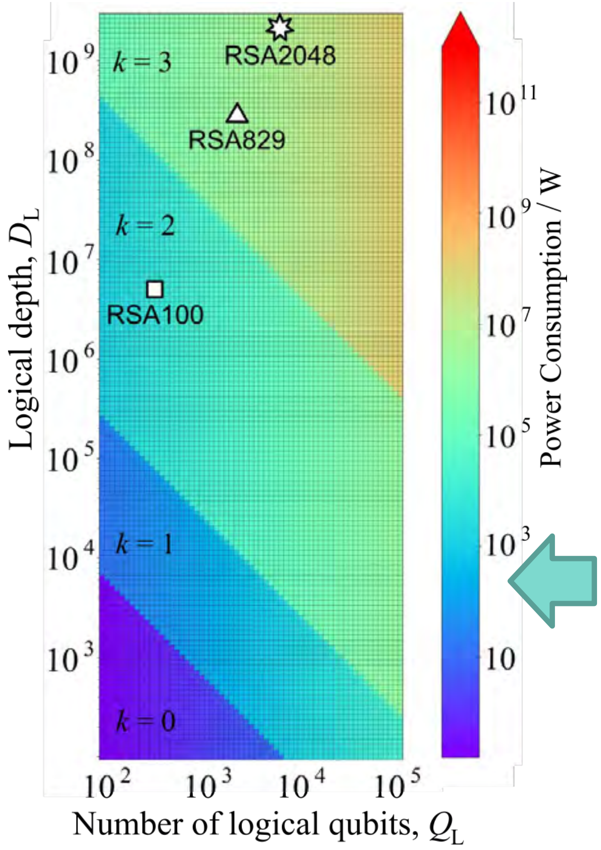
https://arxiv.org/abs/2209.05469

early findings applying the MNR methodology in a particular example

1. **energy advantage** may show up before **computing advantage**.
2. x10 qubit fidelities => **x100 energy savings**.
3. **quantum error correction codes** impact energetic footprint.
4. in FTQC, **control electronics** consumes more energy than cryogeny.
5. significant progress needed in control electronics (room temperature, cryo-electronics, cabling, multiplexing).

it's only a beginning, with many outstanding challenges in all quantum technologies

Minimizing full stack power consumption



Methodology

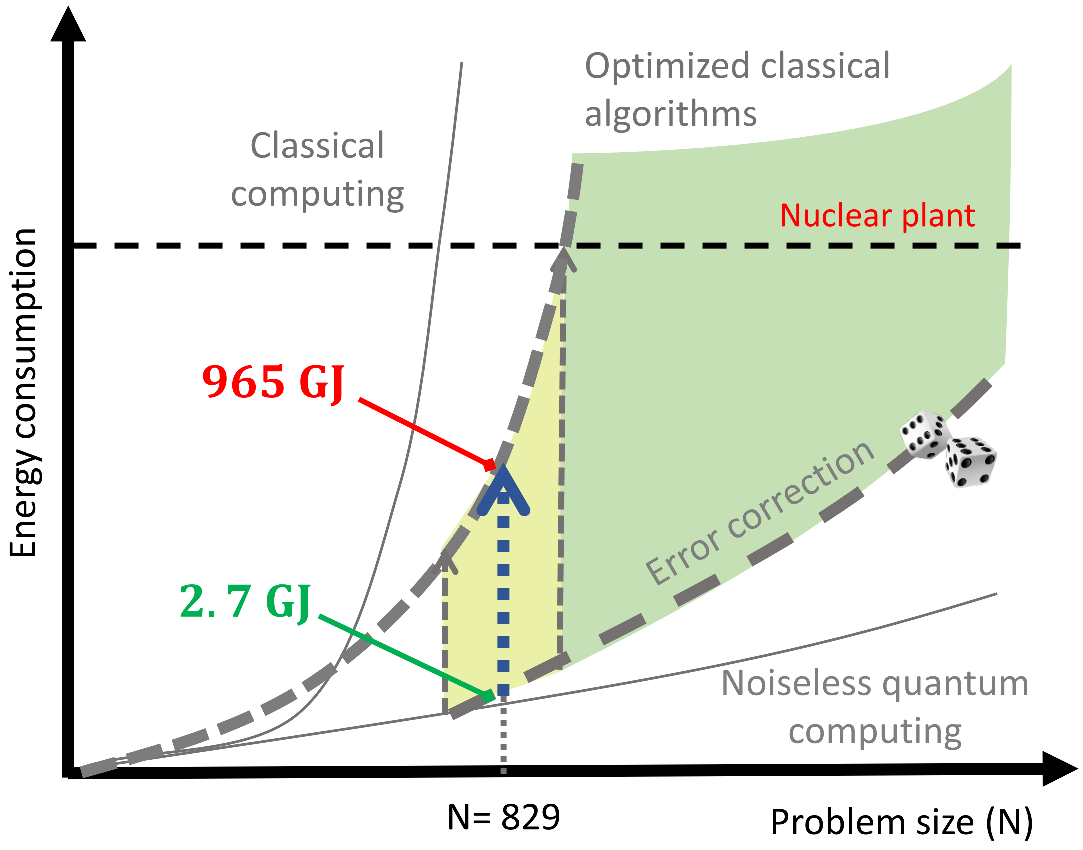
- ✓ Pick a generic circuit $Q_L * D_L$
- ✓ Set $P_{success} = 2/3$
  - $\Rightarrow$ Implicit relation between (A, T_{gen}, T_{qb}, k)
- ✓ Minimize P_{FT} as a function of (A, T_{gen}, T_{qb}, k)

$P_{FT}(Q_L, D_L)$

- ✓ $1/\gamma = 50$ ms (top quality qubits)
- ✓ CMOS electronics

- ➢ Model useful algorithms with our generic circuit
- ➢ RSA-n with $Q_L(n)$ and $D_L(n)$ from Gidney&Ekera, Quantum 5, 433 (2021)

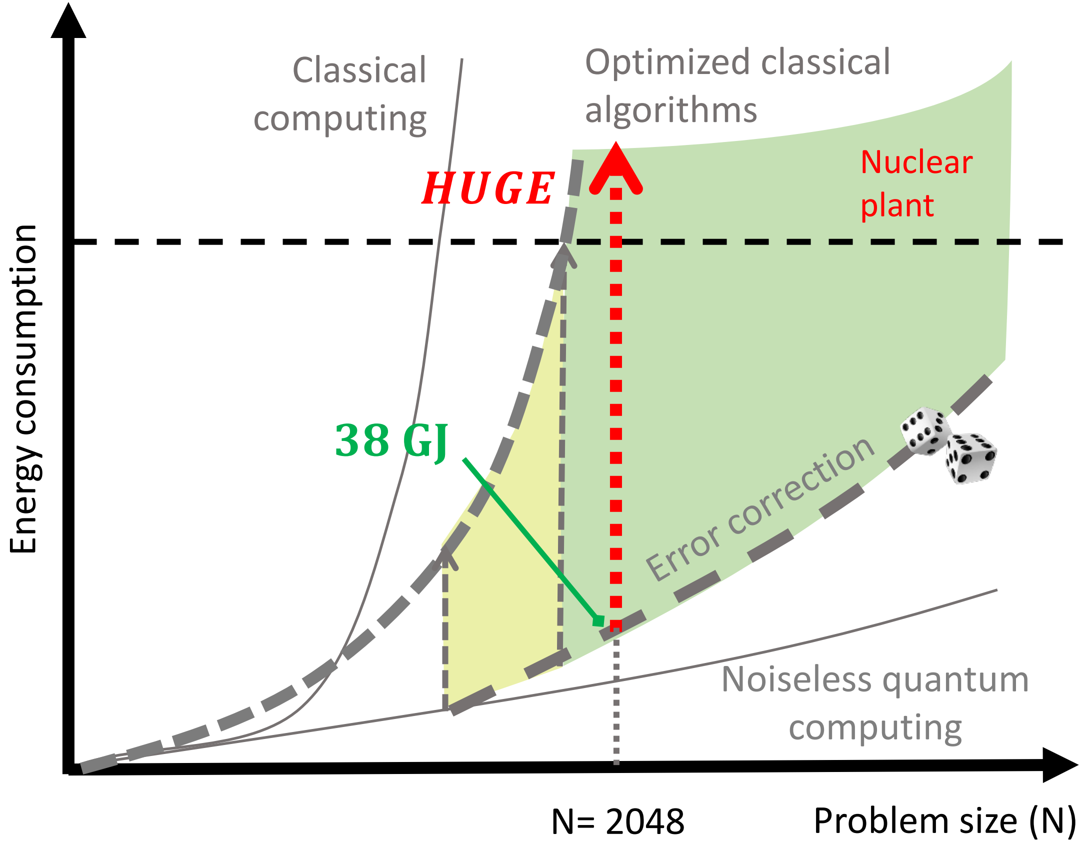
First estimates on quantum energy advantage



Breaking RSA-829 key

- Classical supercomputer (Inria 2021): 965 GJ ≈ 1.3MW in 8.6 days
- Quantum computer with top quality qubits + Steane code 2.7GJ = 2.9 MW in 16 min

First estimates on quantum energy advantage



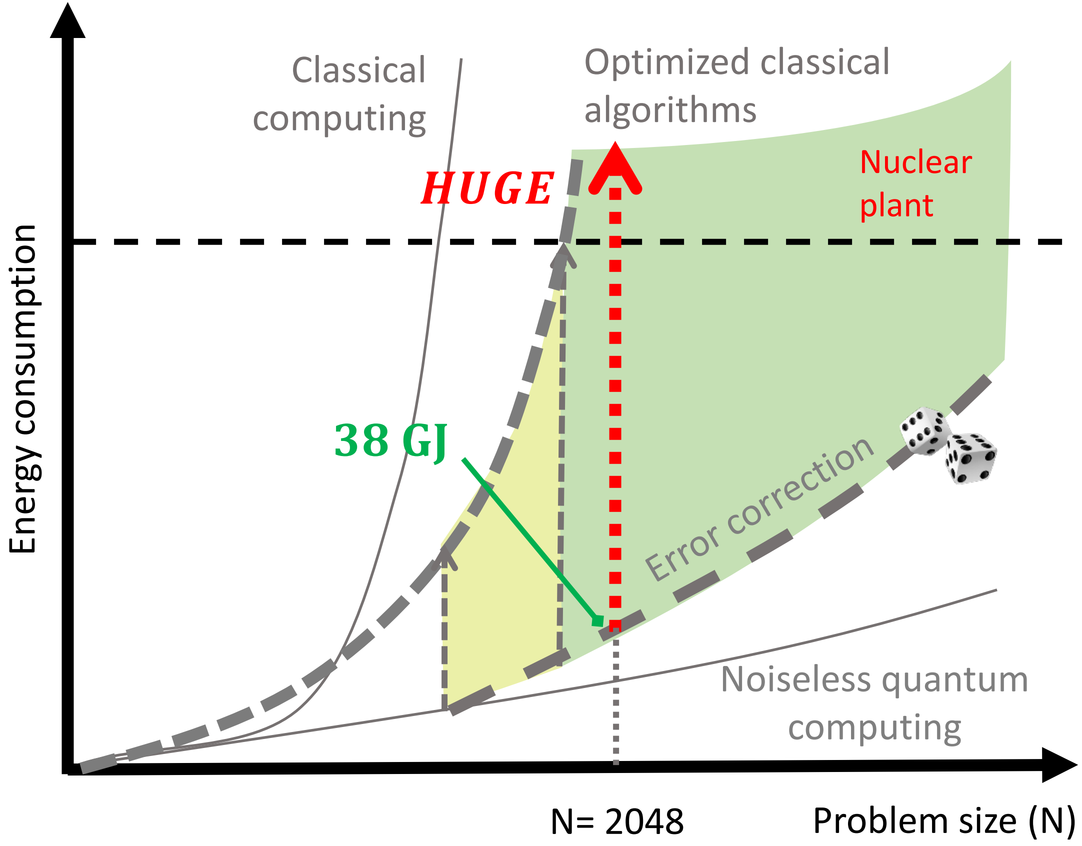
Breaking RSA 829 key

- Classical supercomputer
  965 GJ ≈ 1.3MW in 8.6 days
- Quantum computer with top quality qubits (2000 better than Sycamore) + Steane code
  2.7GJ = 2.9 MW in 16 min

Breaking RSA 2048 key

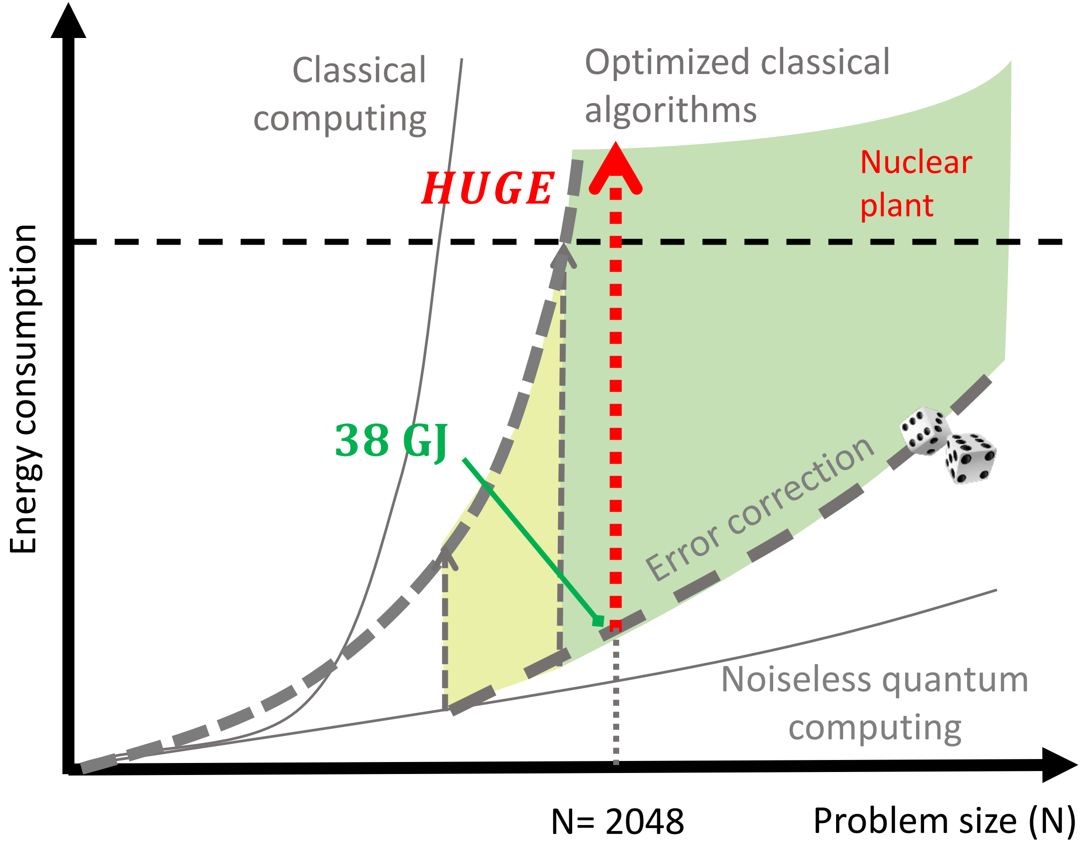
- Classical supercomputer
  TOO MUCH
- Quantum computer (Steane code)
  38 GJ = 7 MW in 1.5 hours

First estimates on quantum energy advantage



- Potential for a quantum energy advantage, but
  - ✓ to consolidate on more realistic qubits/architectures/ full-stack energy costs
  - ✓ in a coordinated way

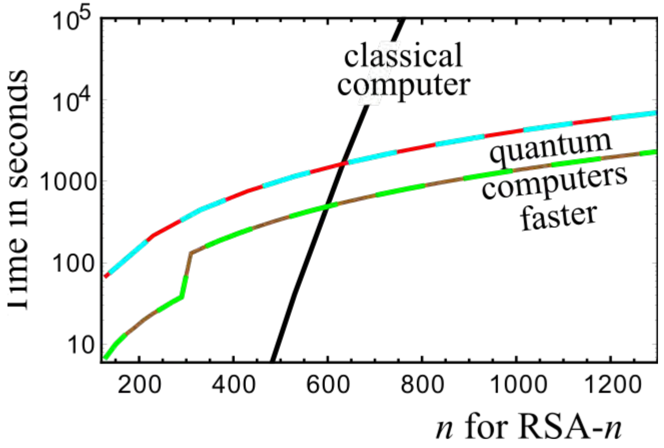
Energy vs computational advantage



- FAQ: « But isn't that enough to optimize the computational advantage? Lowering the computing time will automatically lower the energy cost! »
- Relation between quantum energy advantage and quantum computational advantage?

Energy vs computational advantage

Estimated computing time (s)

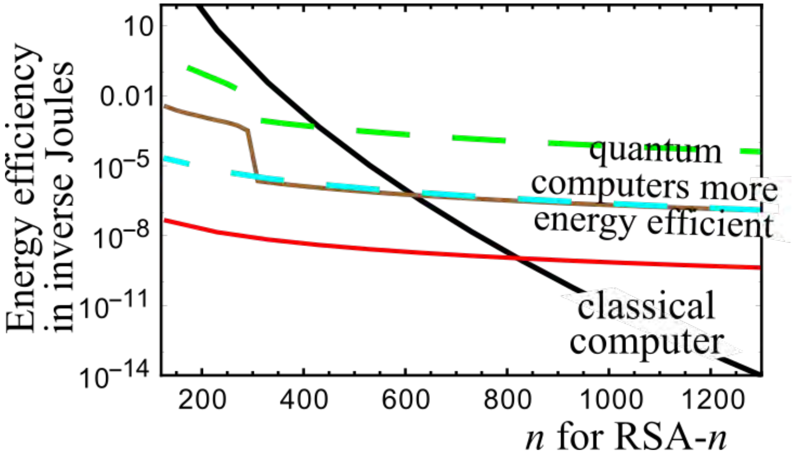


- ✓ red: $1/\gamma = 5$ ms, CMOS control electronics
- ✓ cyan: $1/\gamma = 5$ ms, SFQ control electronics
- ✓ braun: $1/\gamma = 50$ ms, CMOS control electronics
- ✓ green: $1/\gamma = 50$ ms, SFQ control electronics

Energy vs computational advantage

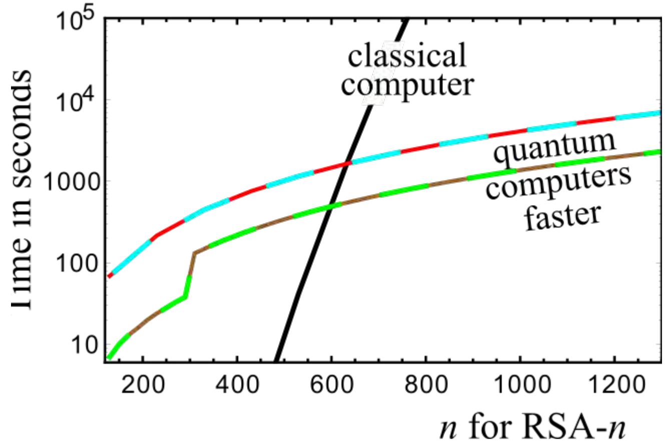
- ✓ red: $1/\gamma = 5$ ms, CMOS control electronics
- ✓ cyan: $1/\gamma = 5$ ms, SFQ control electronics
- ✓ braun: $1/\gamma = 50$ ms, CMOS control electronics
- ✓ green: $1/\gamma = 50$ ms, SFQ control electronics

Efficiency = key size/energy (/J)

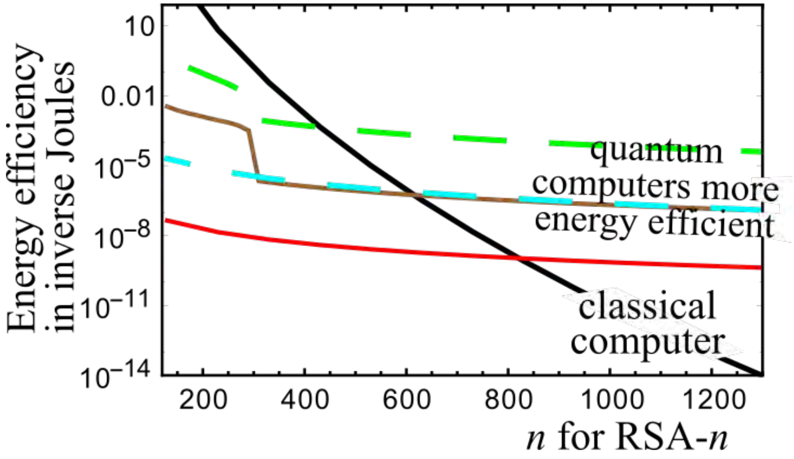


Energy vs computational advantage

Estimated computing time (s)



Efficiency = key size/energy (/J)



- Energy advantage (power*time) ≠ Computational advantage (time) : a practical advantage of different nature!
- One may save energy before saving time...

Take home messages

- **Quantum energy advantage** = a huge practical interest of quantum computing
  - Different from the quantum computational advantage
  - To explore and optimize now
  - Need to articulate different levels of description in an interdisciplinary research line **#QEI**
- **New benchmark**: Quantum computing energy efficiency $\eta = M/R$
  - New tool for optimizations software/hardware; fundamental/full stack
  - Qubits benchmarking
  - Towards a « Q-Green 500 »

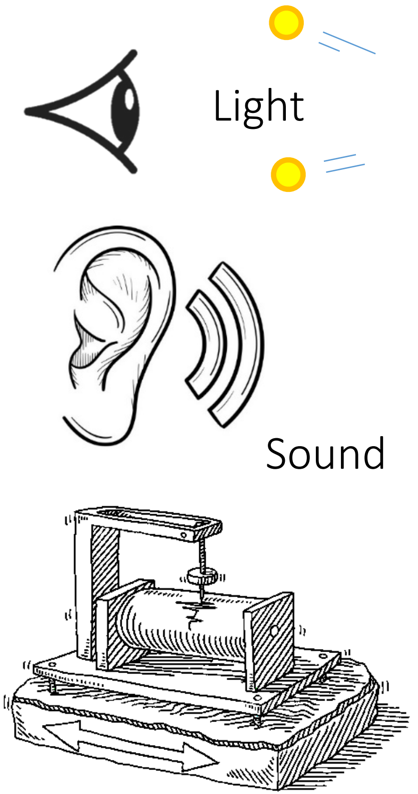
ALICE & BOB

Confidential

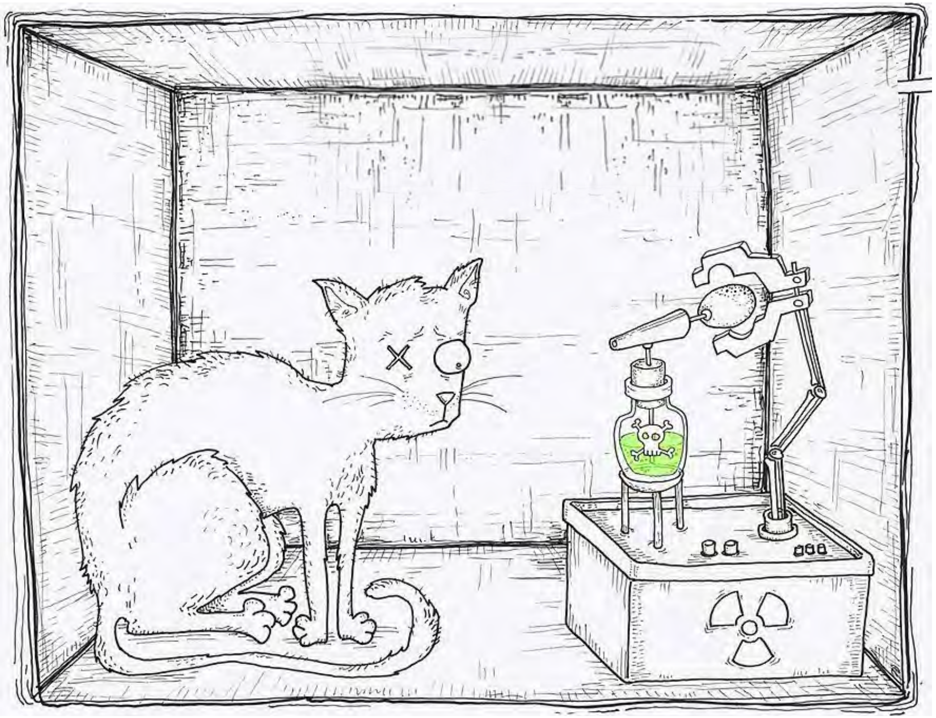
November 2022

PROVIDE EXPONENTIAL QUANTUM COMPUTING POWER ACROSS INDUSTRIES

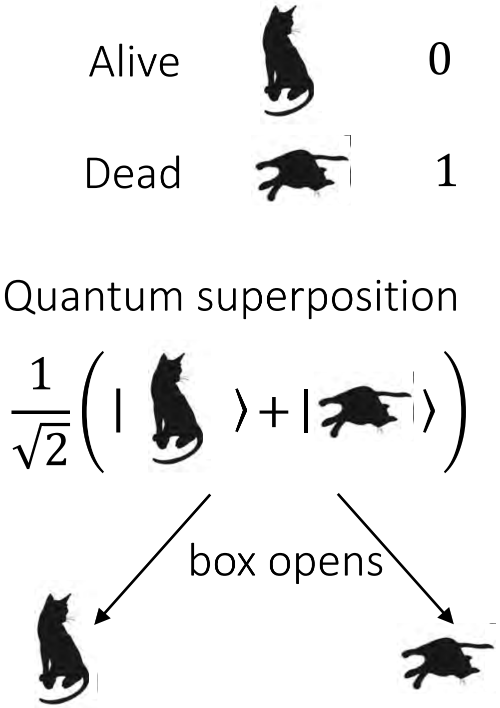
Why are quantum computers faulty ?



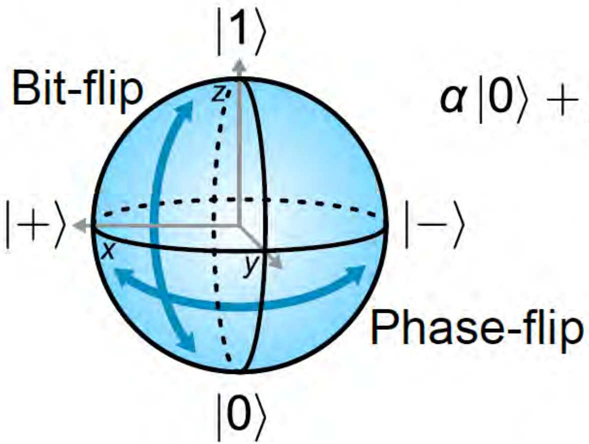
Vibrations

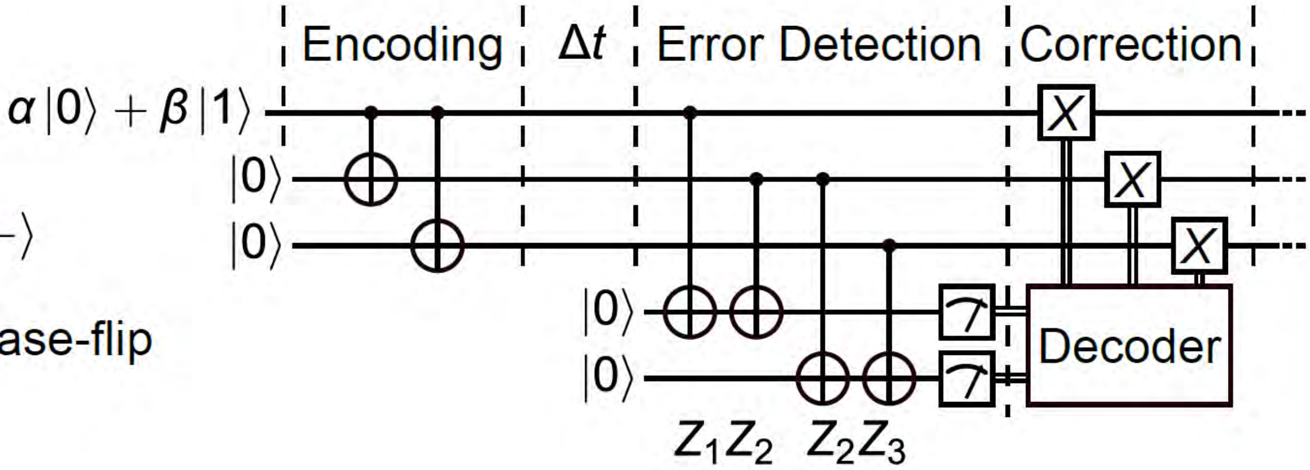


By ADA and Neagoe

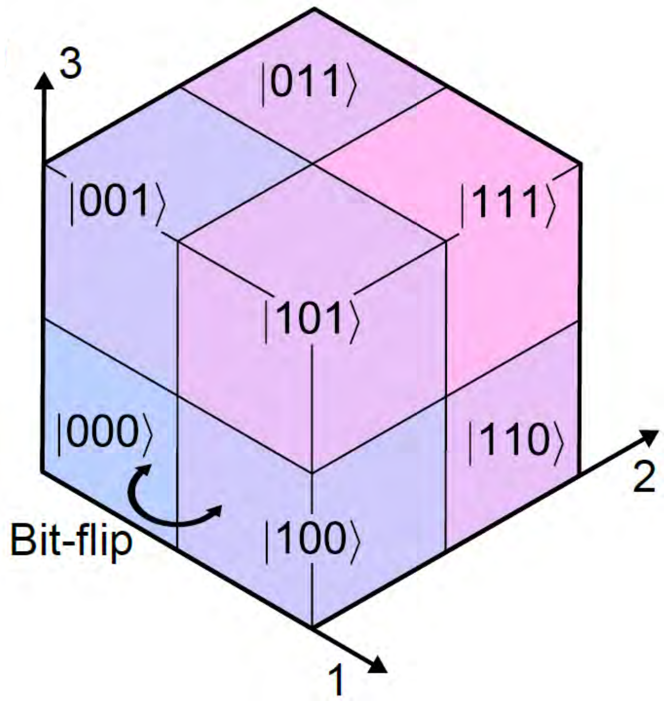


Bit-flip correction circuit





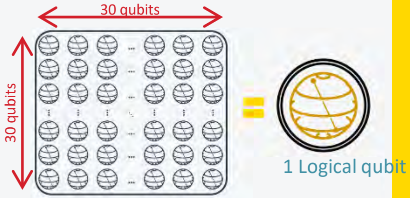
Bit-flip protection



We need a big Hilbert space to protect against errors

QUANTITATIVE APPROACH TO REDUCE ERRORS

30 physical qubits to correct each type of error



DEFINITION

Logical qubit

Qubit able to store quantum information with sufficiently low error probability

Universal qubit

Logical qubit able to perform any type of operation

Universal Quantum Computer would require 1,000,000+ physical qubits

1,000 Physical qubits → 30 x 30 → Bit-flips correction / Phase-flips correction → 1 Logical qubit

100,000 Physical qubits → 100 Logical qubits → 1 Universal qubit

1,000,000+ Physical qubits → 10+ Universal qubits → 1 Universal Quantum Computer

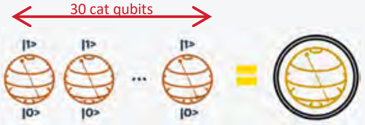
Players following this strategy

Google | IBM | rigetti | IONQ | intel | PsiQuantum

With 128 physical qubits

A&B'S QUALITATIVE APPROACH

Only 30 cat qubits needed to correct phase-flip errors



1,000x fewer physical qubits to build a Universal Quantum Computer

01.

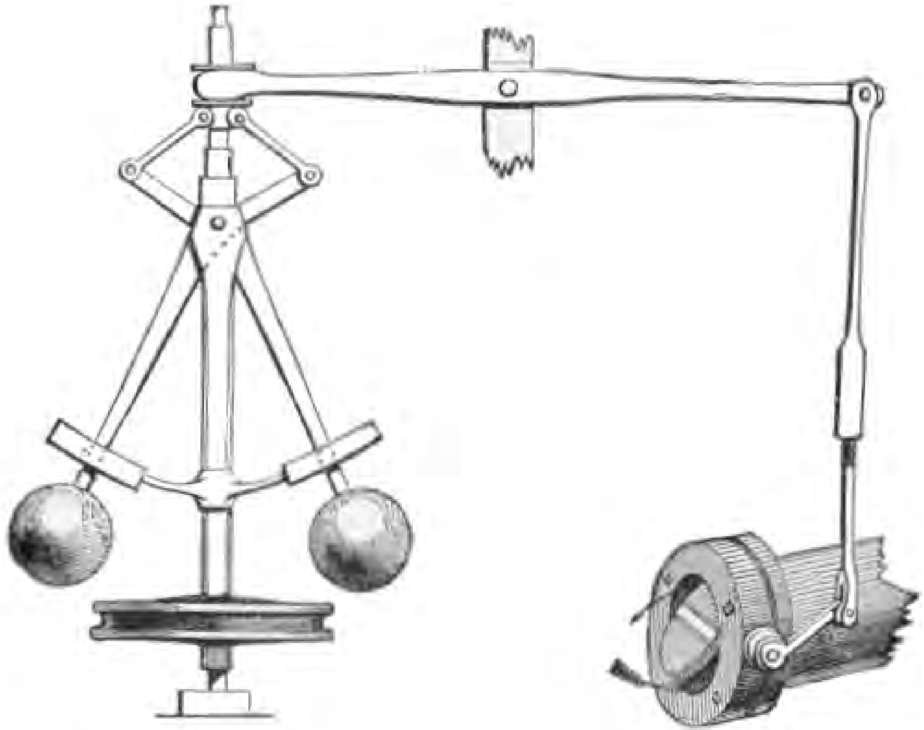
Cat Qubit:
Autonomous error-correction of bit flips by design

02.

Shortcut to universality

	"QUANTITATIVE" APPROACH: STANDARD QUBITS		BY DESIGN APPROACH: CAT QUBITS
1 LOGICAL QUBIT	1,000 physical qubits	vs / 30	30 cat qubits
1 UNIVERSAL QUBIT	100,000 physical qubits	vs **/ 1,000**	90 cat qubits

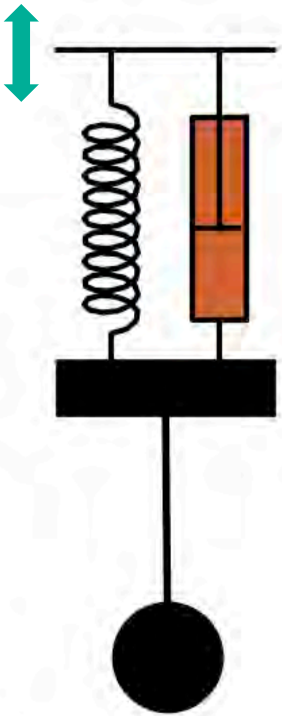
Autonomous regulation



The Watt regulator autonomously controls the speed of a steam engine

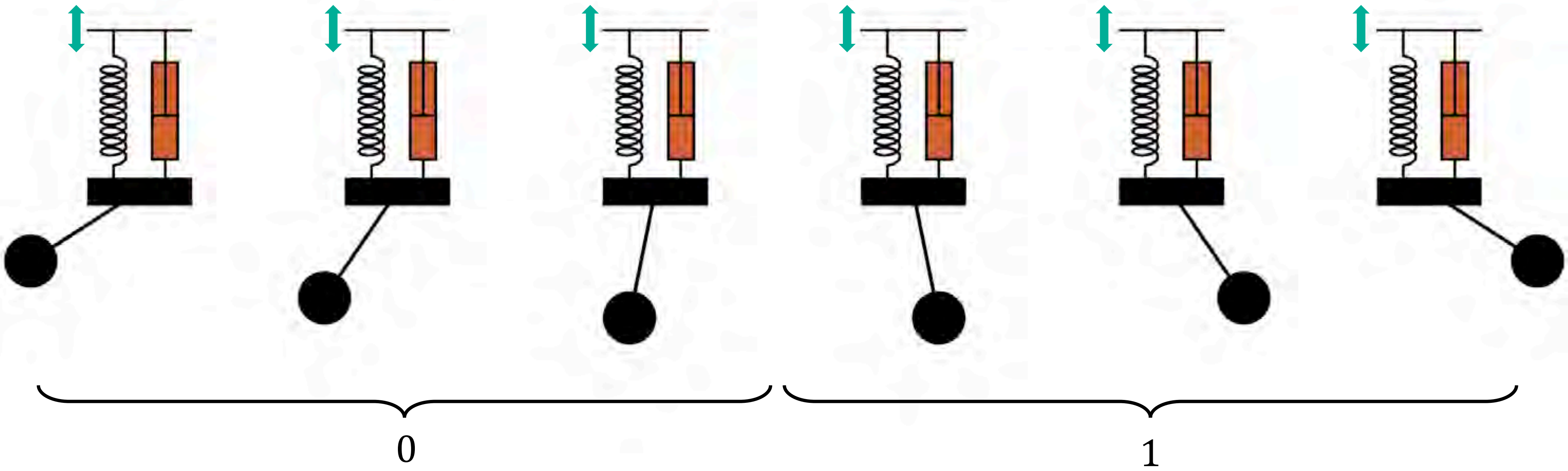
Can we build a quantum system that self corrects ?

Adding dissipation



Dissipation prevents the motion to diverge and **stabilizes** it to a given state.

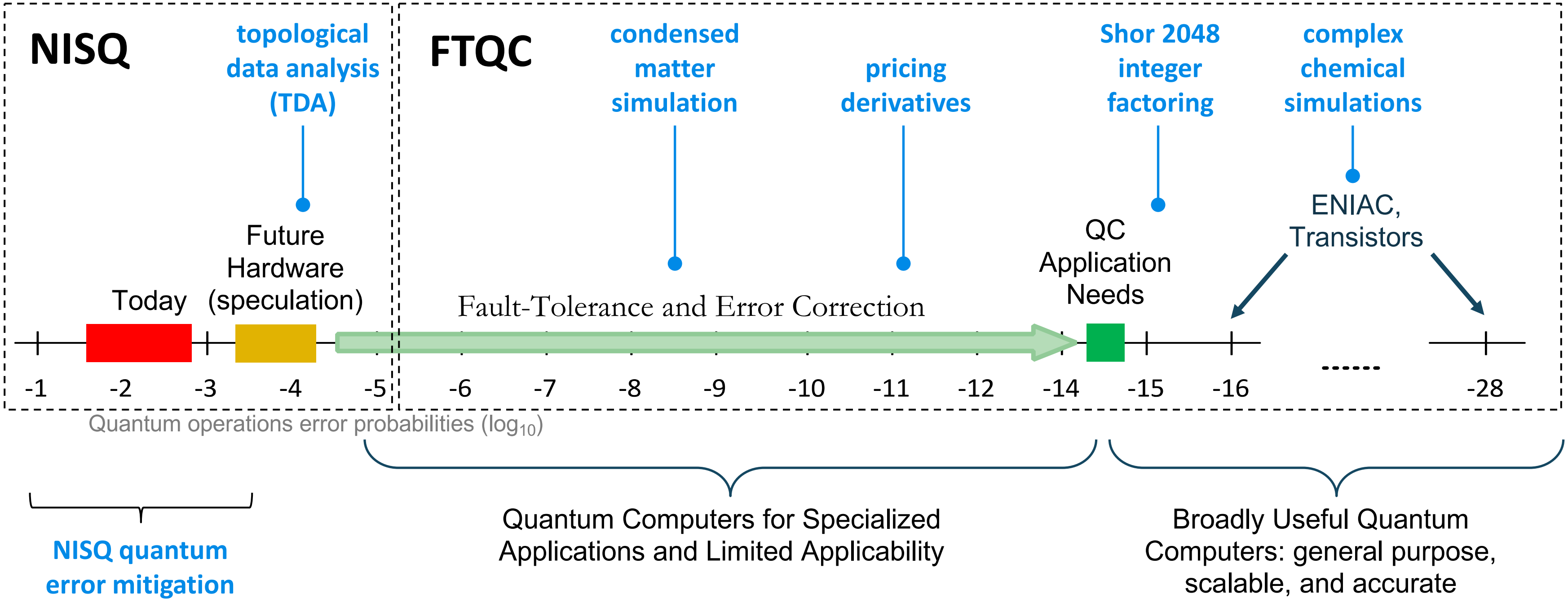
A bi-stable system



There are **2 steady states** in which we can encode information

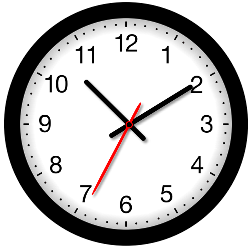
QEI proposed methodology
FTQC perspective
NISQ perspective

from NISQ to FTQC

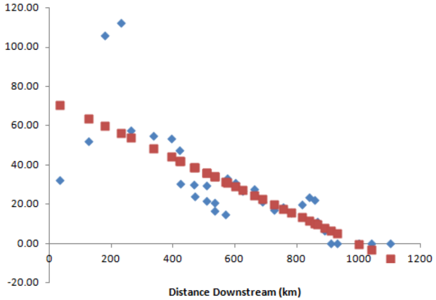


source: How about quantum computing? by Bert de Jong, DoE Berkeley Labs, June 2019 (47 slides) + Olivier Ezratty additions.

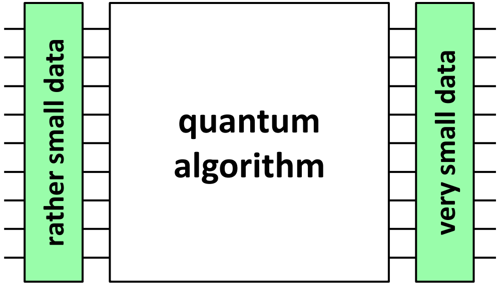
NISQ figures of merit



what speedup advantage?



what precision advantage?



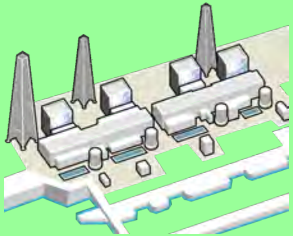
smaller input data?

cost/benefit of quantum error mitigation?

fully-burdened cost with classical+QPU?

analog vs gate-based differences?

which algorithms?



energetic advantage?

total cost?

Financial Risk Management on a Neutral Atom Quantum Processor

Lucas Leclerc[1,2,*] Luis Ortiz-Gutiérrez[1], Sebastián Grijalva[1], Boris Albrecht[1], Julia R. K. Cline[1], Vincent E. Elfving[1], Adrien Signoles[1], and Loïc Henriet[1†]
[1] *PASQAL, 7 rue Léonard de Vinci, 91300 Massy, France and*
[2] *Université Paris-Saclay, Institut d'Optique Graduate School, CNRS, Laboratoire Charles Fabry, 91127 Palaiseau, France*

Gianni Del Bimbo[3,*] Usman Ayub Sheikh[3,*] Maitree Shah[4], Luc Andrea[5], Faysal Ishtiaq[3], Andoni Duarte[3], Sam Mugel[4], Irene Cáceres[3], Michel Kurek[5], and Roman Orús[3,6,7]
[3] *Multiverse Computing, Parque Científico y Tecnológico de Gipuzkoa, Paseo de Miramón 170, 20014 San Sebastián, Spain*
[4] *Centre for Social Innovation, 192 Spadina Ave, Suite 509, M5T 2C2 Toronto, Canada*
[5] *WIPSE Paris-Saclay Enterprises 7, rue de la Croix Martre 91120 Palaiseau, France*
[6] *Donostia International Physics Center, Paseo Manuel de Lardizabal 4, E-20018 San Sebastián, Spain and*
[7] *Ikerbasque Foundation for Science, Maria Diaz de Haro 3, E-48013 Bilbao, Spain*

Achraf Seddik[8], Oumaima Hammami[8], Hacene Isselnane[8], and Didier M'tamon[8]
[8] *Crédit Agricole Corporate and Investment Bank, 12 Place des États-Unis, 92545 Montrouge, France*
(Dated: December 7, 2022)

Machine Learning models capable of handling the large datasets collected in the financial world can often become black boxes expensive to run. The quantum computing paradigm suggests new optimization techniques, that combined with classical algorithms, may deliver competitive, faster and more interpretable models. In this work we propose a quantum-enhanced machine learning solution for the prediction of credit rating downgrades, also known as fallen-angels forecasting in the financial risk management field. We implement this solution on a neutral atom Quantum Processing Unit with up to 60 qubits on a real-life dataset. We report competitive performances against the state-of-the-art Random Forest benchmark whilst our model achieves better interpretability and comparable training times. We examine how to improve performance in the near-term validating our ideas with Tensor Networks-based numerical simulations.

https://arxiv.org/abs/2212.03223

CRÉDIT AGRICOLE CORPORATE & INVESTMENT BANK

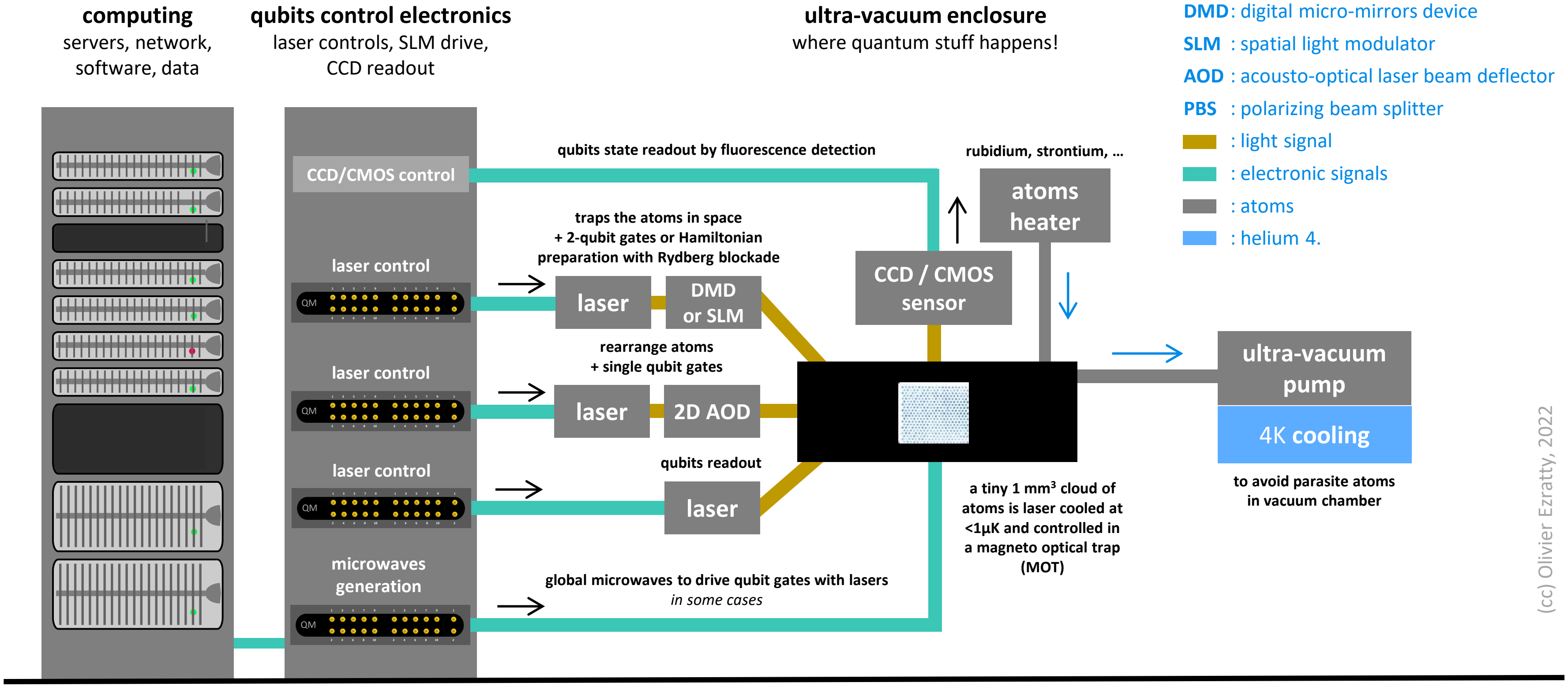
PASQAL MULTIVERSE

QBoost hybrid algorithm used to predict « fallen angels », businesses who could fail in loans reimbursements. Quantum algorithm reduced to a QUBO problem.

data set: 20 years + 90 000 items with 150 features on 2000 companies in 10 verticals and 100 sub-verticals from 70 countries. 65 000 items in training data and 26 000 items for tests.

quantum advantage: could show up with 150 - 342 neutral atoms when compared to a best-in-class classical tensor network, 2800 atoms for the more precise subsampling method.

inside a neutral atoms QC



gate-based cold atoms quantum computer simplified view *

* : AOD and SLM beams can be assembled through a PBS (polarizing beam splitter), atoms cooling lasers not shown.

computing paradigms and algorithms

quantum advantage

some useful cases

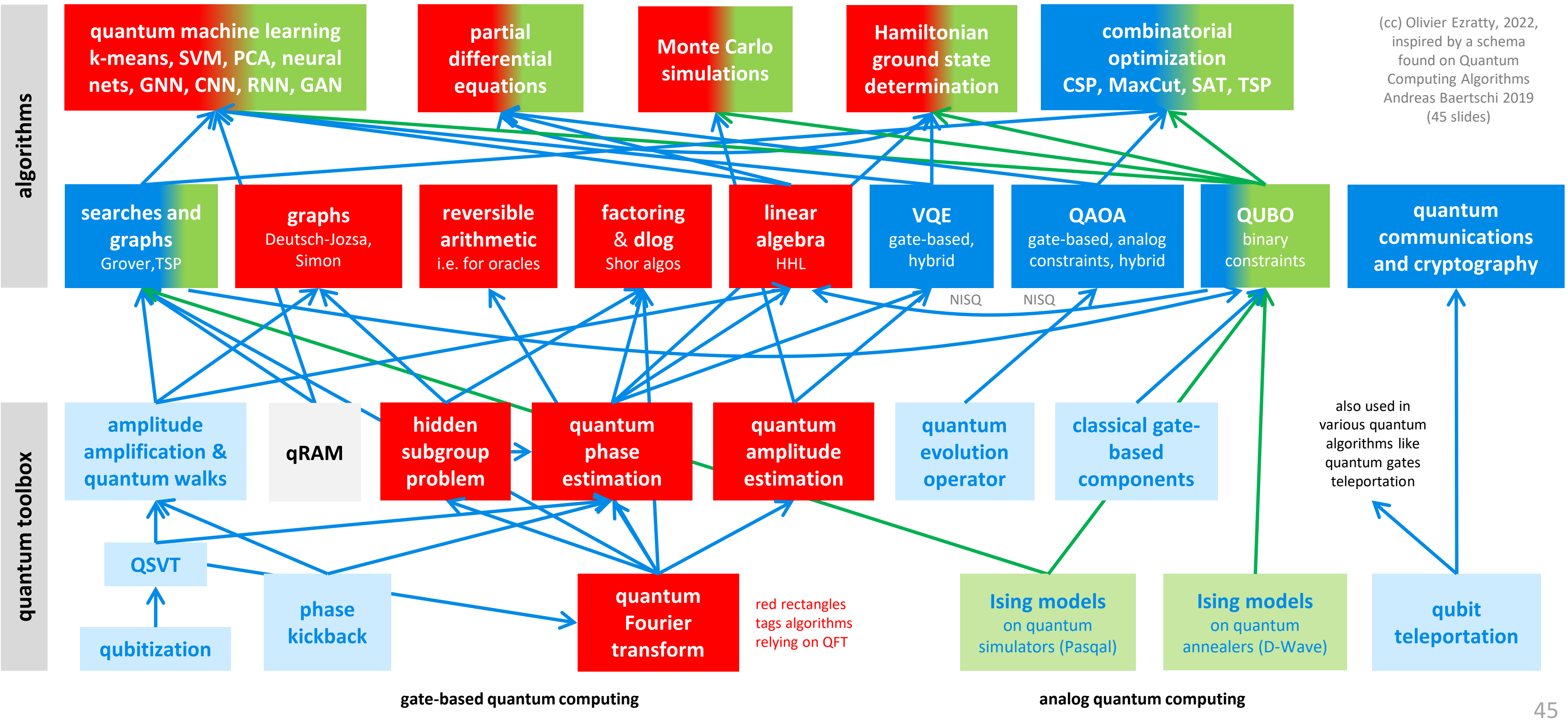
no useful case so far

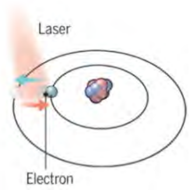
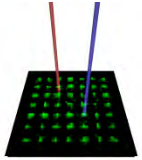
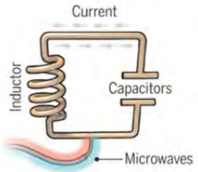
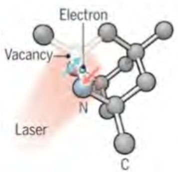
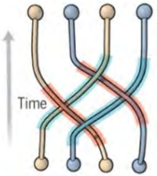
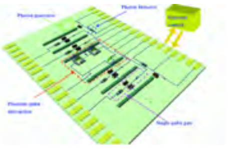
				gate-based quantum computers	
	classical computers	quantum annealer	quantum simulators	NISQ	FTQC
search algorithms	hybrid algorithms				polynomial speedup
optimization algorithms	hybrid algorithms				may require qRAM
quantum machine learning	hybrid algorithms				may require qRAM
physics simulation	hybrid algorithms				100 to 10K logical qubits
organic chemistry simulation	hybrid algorithms				100 to 10K logical qubits
integer factoring	hybrid algorithms	6-digit record to date		15-digit record to date (QAOA, China)	4K-6K logical qubits
quantum inspired algorithms					
	now	now	soon	later	much later

« quantumness » and arrow of time availability

NISQ: noisy intermediate scale quantum computer, FTQC : lfault tolerant quantum computer

quantum algorithms map

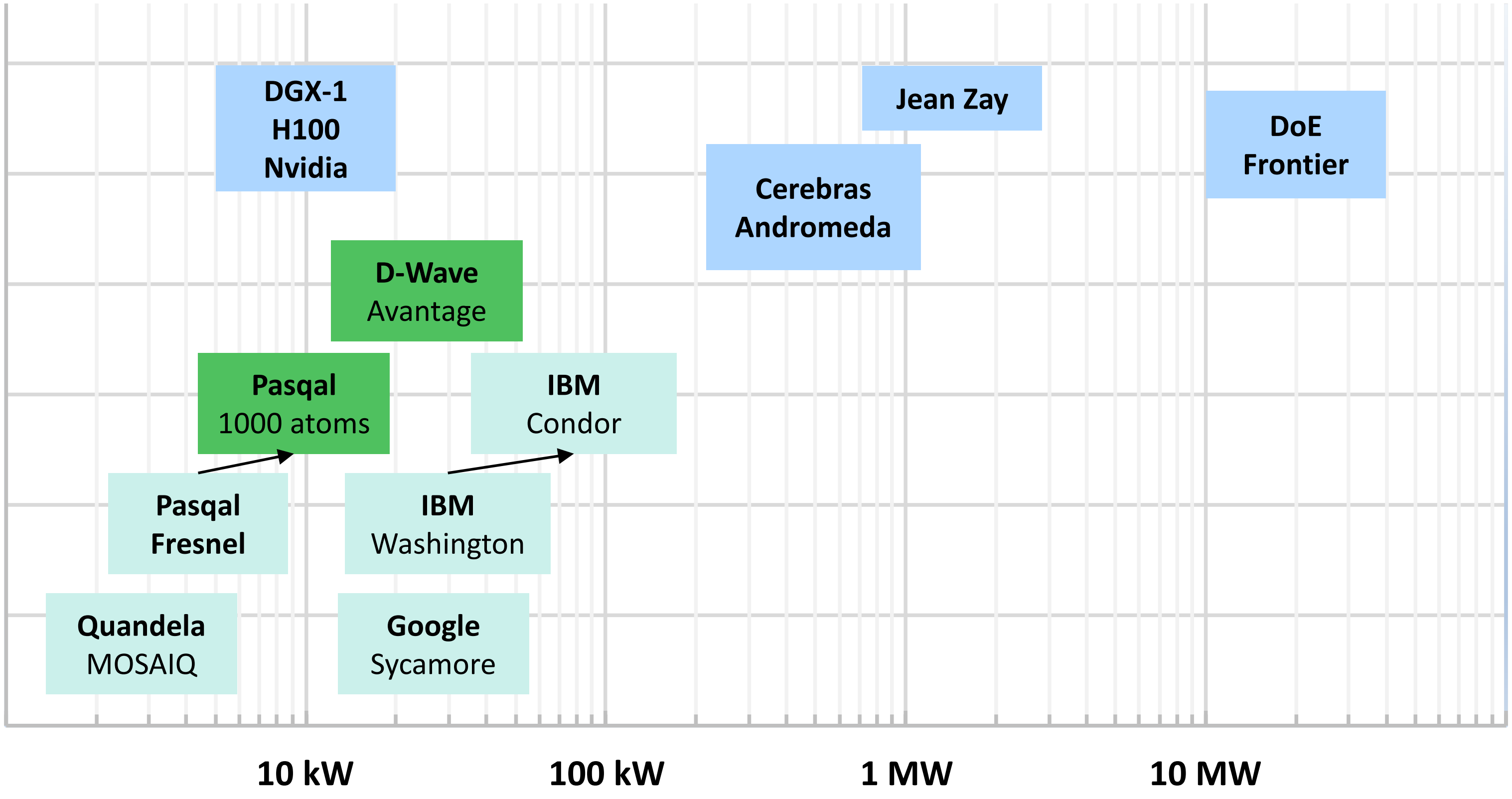


	atoms		electron superconducting loops & controlled spin				photons
							
qubit type	**trapped ions**	**cold atoms**	**super-conducting**	**silicon**	**NV centers**	**Majorana fermions**	**photons**
cryogeny	<300W	N/A	16 KW	12 KW	< 1 KW	16 KW	3KW
vacuum pumps[1]	Vacuum	ultra-vacuum 100W	vacuum	vacuum	vacuum	vacuum	vacuum
qubits gate controls	<1.4KW ions heating, lasers, micro-aves generation, CMOS readout electronics	5,8KW atoms heater, lasers, control (SLM, etc) and readout image sensor + electronics	from 20 mW to 100W / qubit depending on architectures with micro-wave generation outside or inside the cryostat		N/A	N/A	300 W for photons sources and detectors, qubit gates controls
computing	300W	1 KW	1 KW	1 KW	<1 KW	1 KW	700 W
# qubits used	24	100-1000	53-433	4	N/A	N/A	20
total	**2 KW** (4)	**7 KW** (1)	**25 KW** (2)	**21 KW**	**N/A**	**N/A**	**4 KW** (3)

[1] : fixed energetic cost, for preping stage

typical configurations for Pasqal (1), Google Sycamore with 53 qubits (2), Quandela/QuiX (3), AQT (4) rough estimates for others

QPU + classical energetics scale



QC energetics benchmarking

LABORATOIRE NATIONAL DE MÉTROLOGIE ET D'ESSAIS

IEEE Advancing Technology for Humanity

QEI WG (C/S2ESC/QEI)

discussion

Séminaire quantique EDF TQCI
11 Janvier 2023

NVIDIA

Bringing GPU acceleration to Hybrid Quantum-Classical Computing

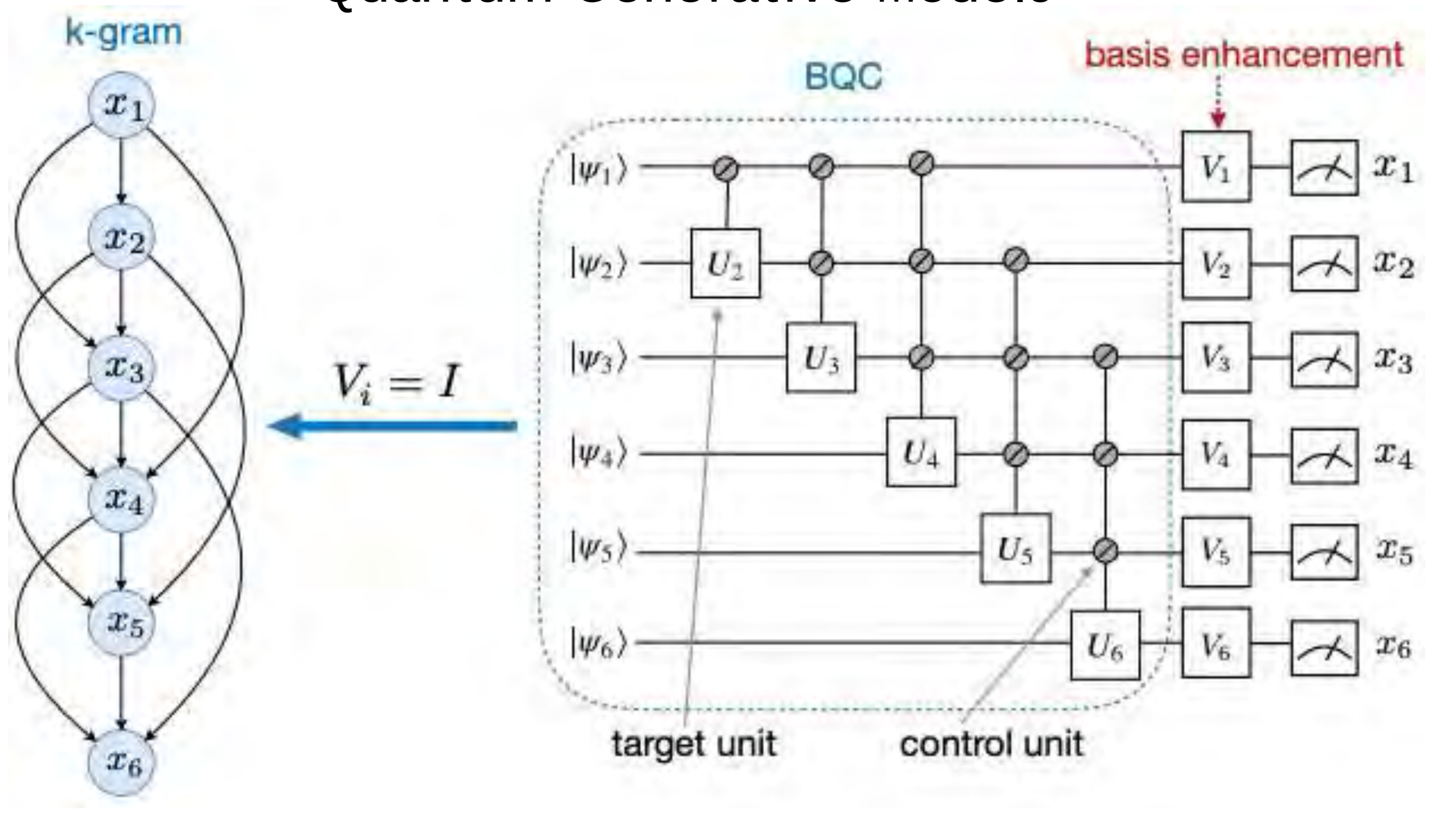
François Courteille |Principal Solutions Architect, NVIDIA | fcourteille@nvidia.com
Sam Stanwyck |Senior Product Manager for Quantum Computing Software, NVIDIA | sstanwyck@nvidia.com

POTENTIAL NEAR-TERM QUANTUM USE-CASES

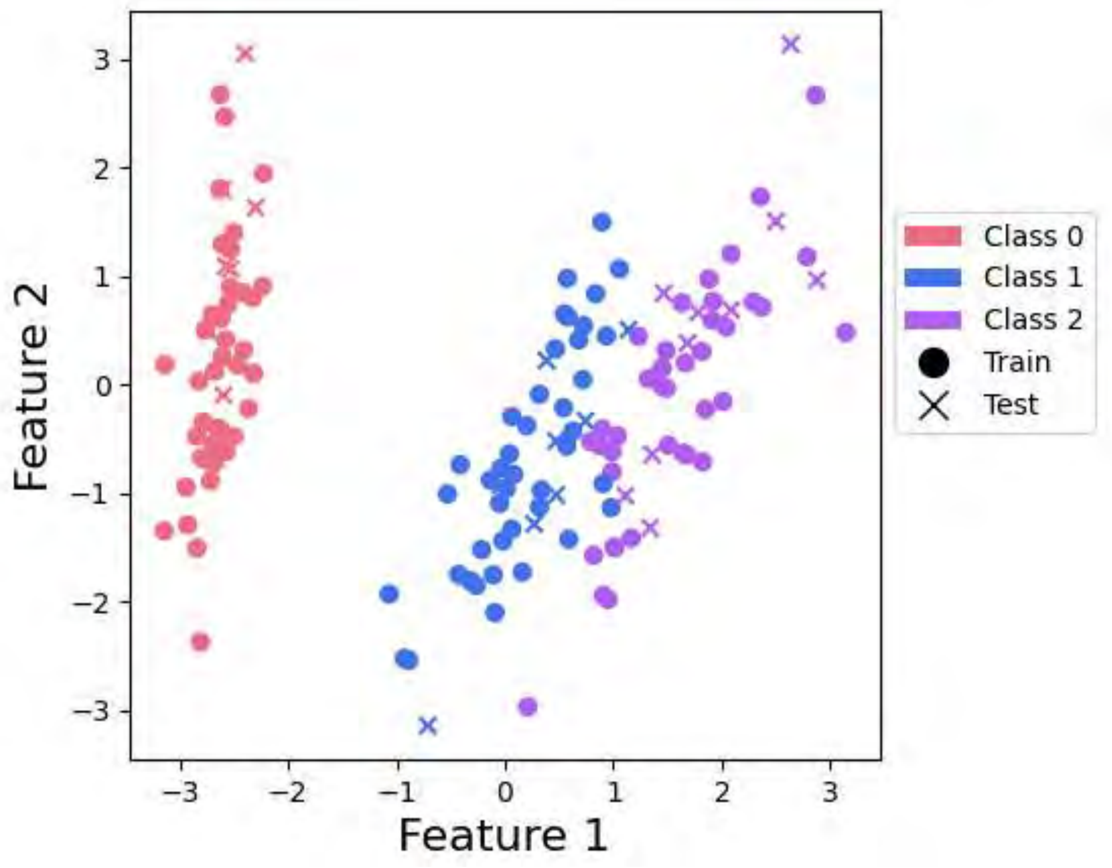
Applications with near-term potential, but quantum advantage is an open question

Quantum Machine Learning

Quantum Generative Models



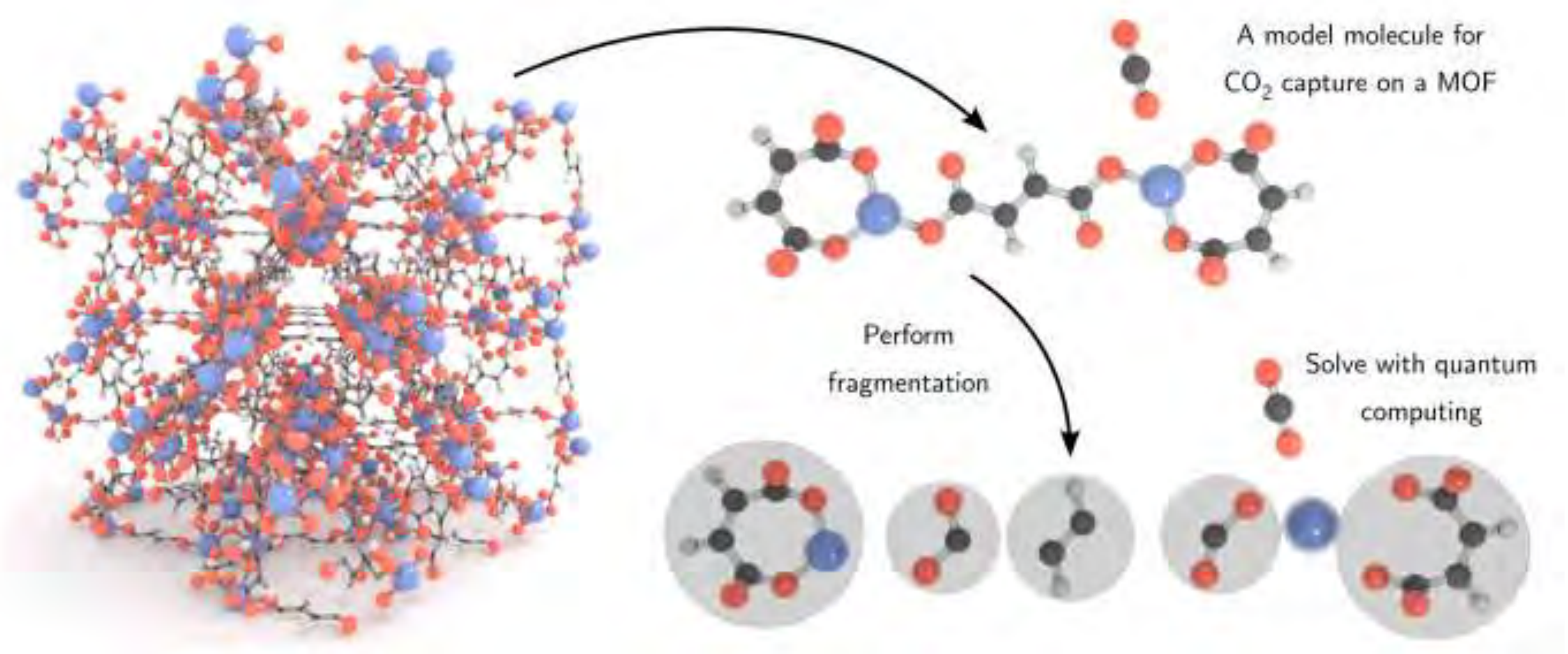
Quantum Support Vector Machine



Gao, et al, Phys. Rev. X 12, 021037
Pennylane.ai

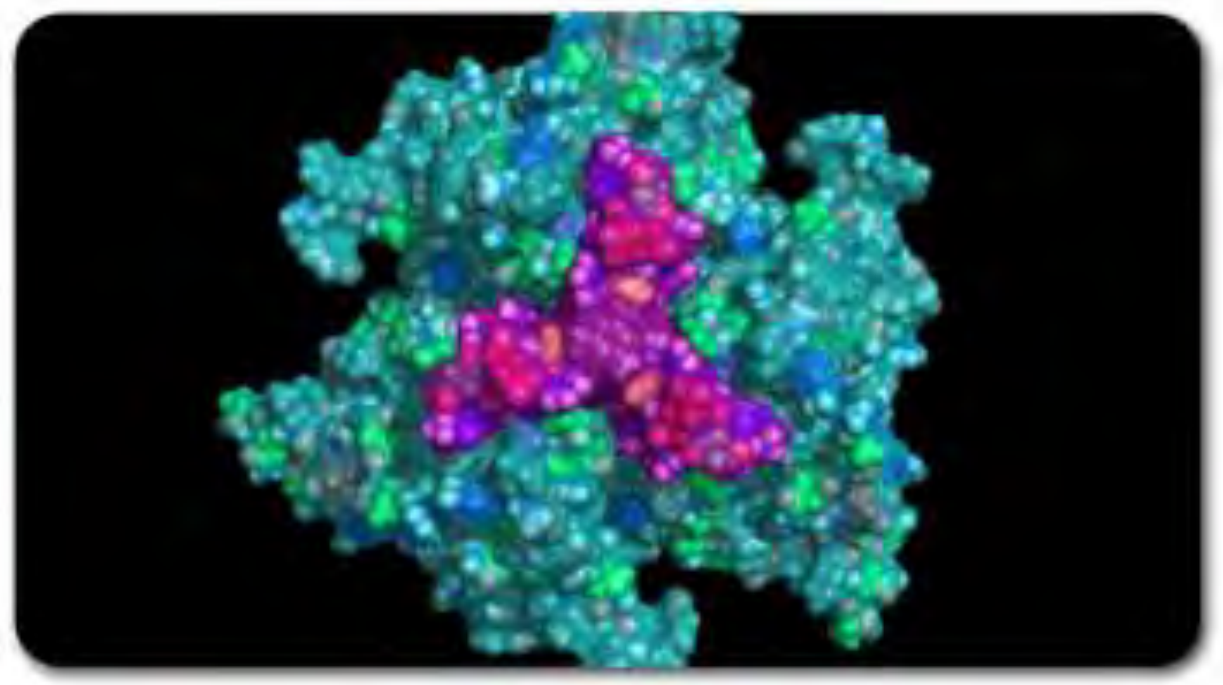
Quantum Chemistry

Variational Quantum Eigensolver for carbon capture



Greene-Diniz, et al, arXiv:2203.15546,

Protein folding



Menten.ai

Combinatorial Optimization

QAOA for resource allocation

Logistics optimization

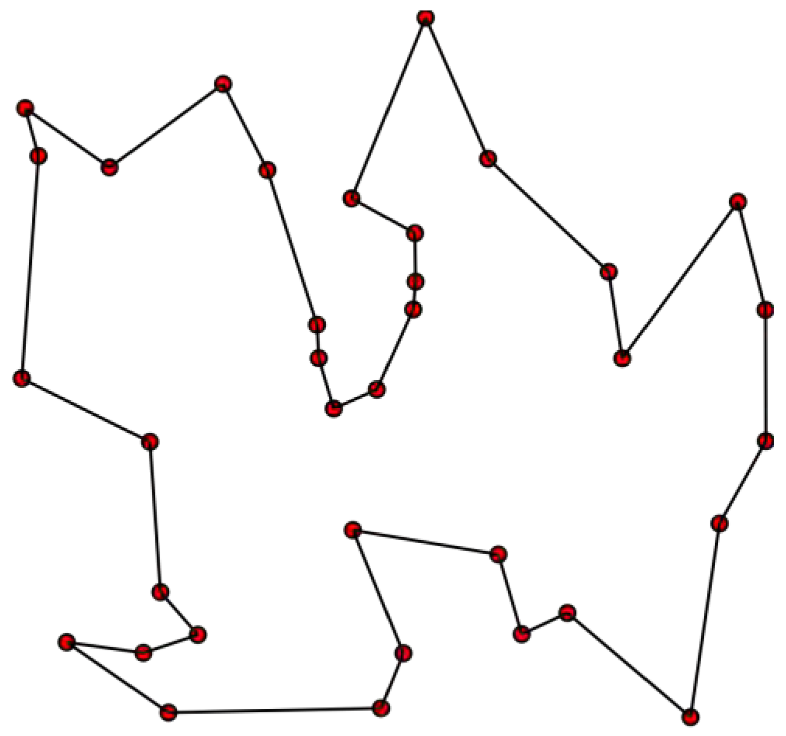


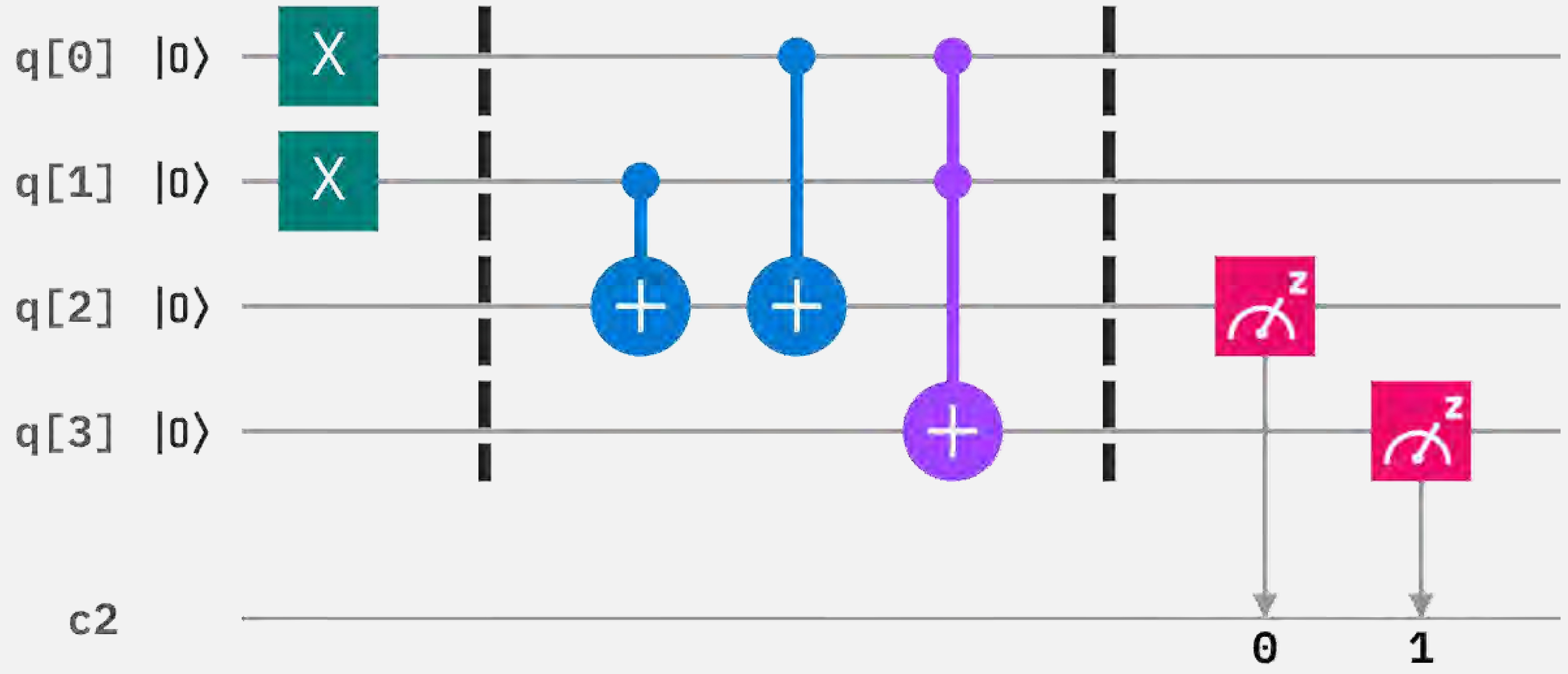
Image from ibm.com
Wikipedia.com

Quantum Computing @ NVIDIA

Quantum Computing at NVIDIA

Researching the Quantum Computers of Tomorrow with the Supercomputers of Today

QUANTUM SIMULATION - CUQUANTUM



- Develop algorithms at scale of valuable quantum computing
- Discover use cases with quantum advantage
- Design and validate future hardware

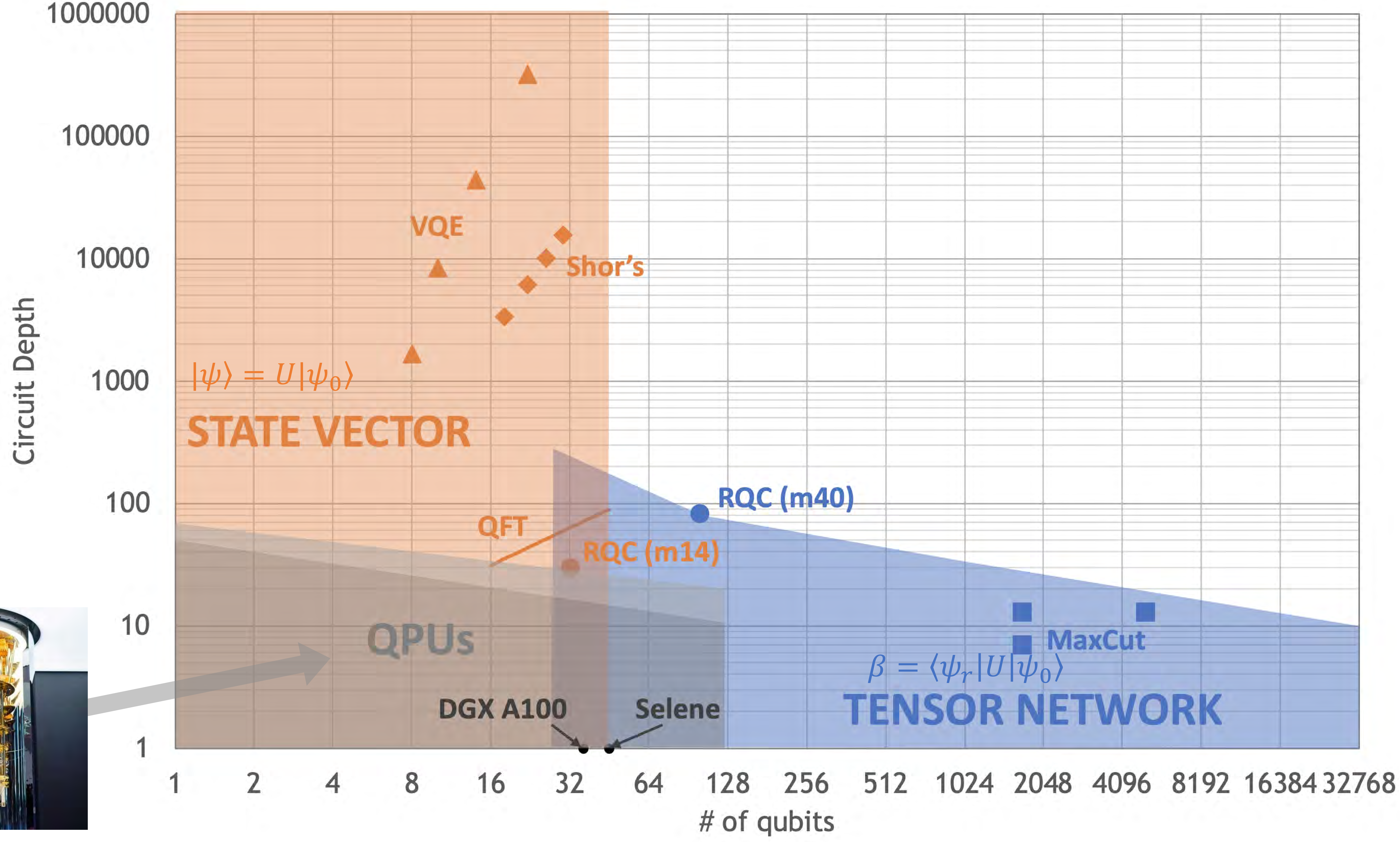
HYBRID QUANTUM-CLASSICAL COMPUTING - QODA

+

- Develop quantum applications by integrating quantum into leading accelerated applications
- Build a platform that is familiar to domain scientists
- Unparalleled performance and scientific productivity using the best resource for the task

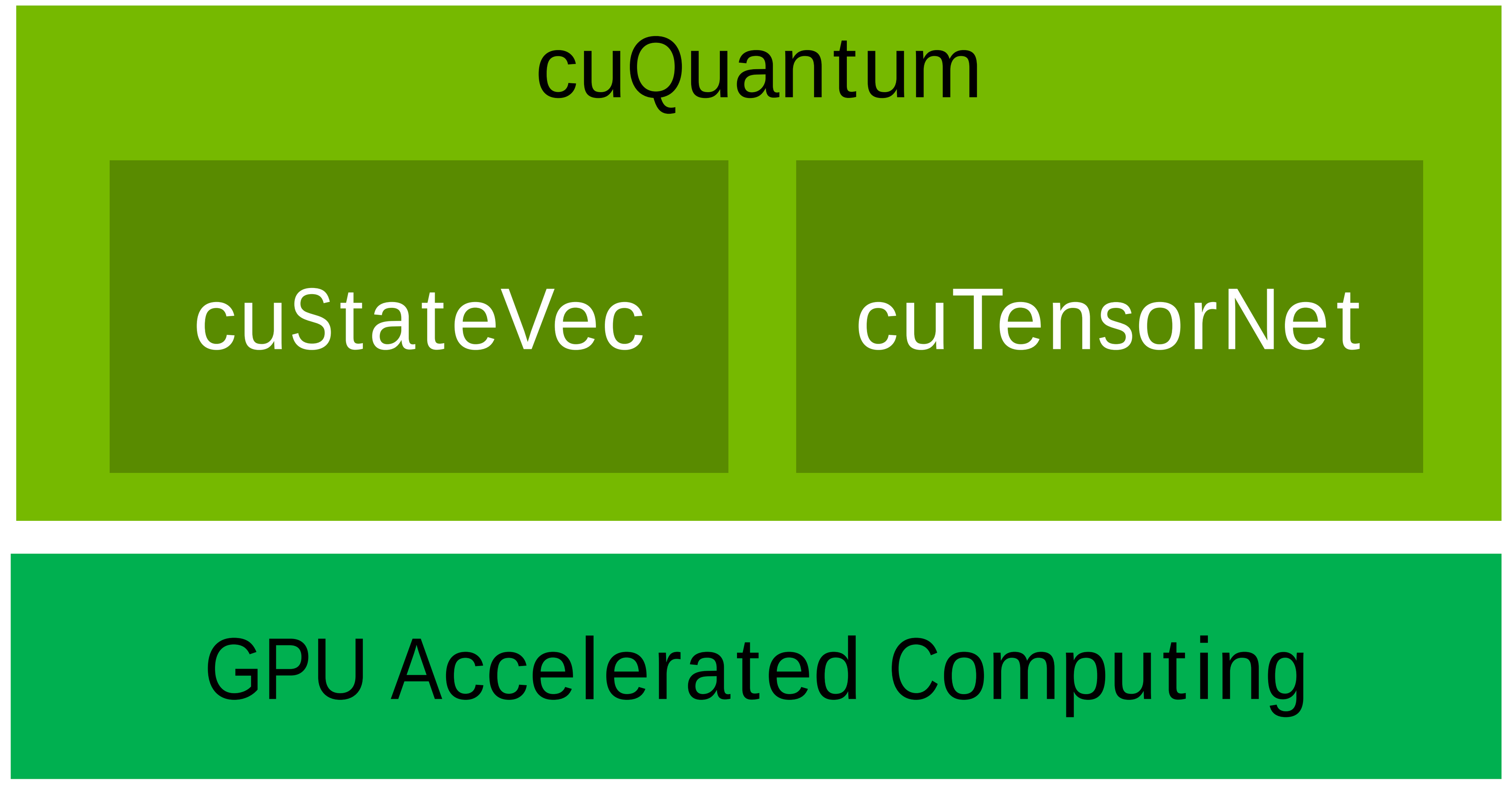
POWERFUL SIMULATIONS WITH cuQuantum

Researching & Developing the Computers of Tomorrow Requires Powerful Simulations Today



cuQuantum

- SDK for Quantum Circuit Simulation
  - Accelerate Quantum Circuit Simulators on GPUs
  - Simulate ideal or noisy qubits
  - Enable algorithms research with scale and performance not possible on quantum hardware, or on simulators today
- cuQuantum available now
  - Integrated into leading quantum computing frameworks Cirq, Qiskit, and Pennylane
  - C and Python APIs
  - Available today at developer.nvidia.com/cuquantum



cuQuantum Appliance

Continuous Performance Improvement

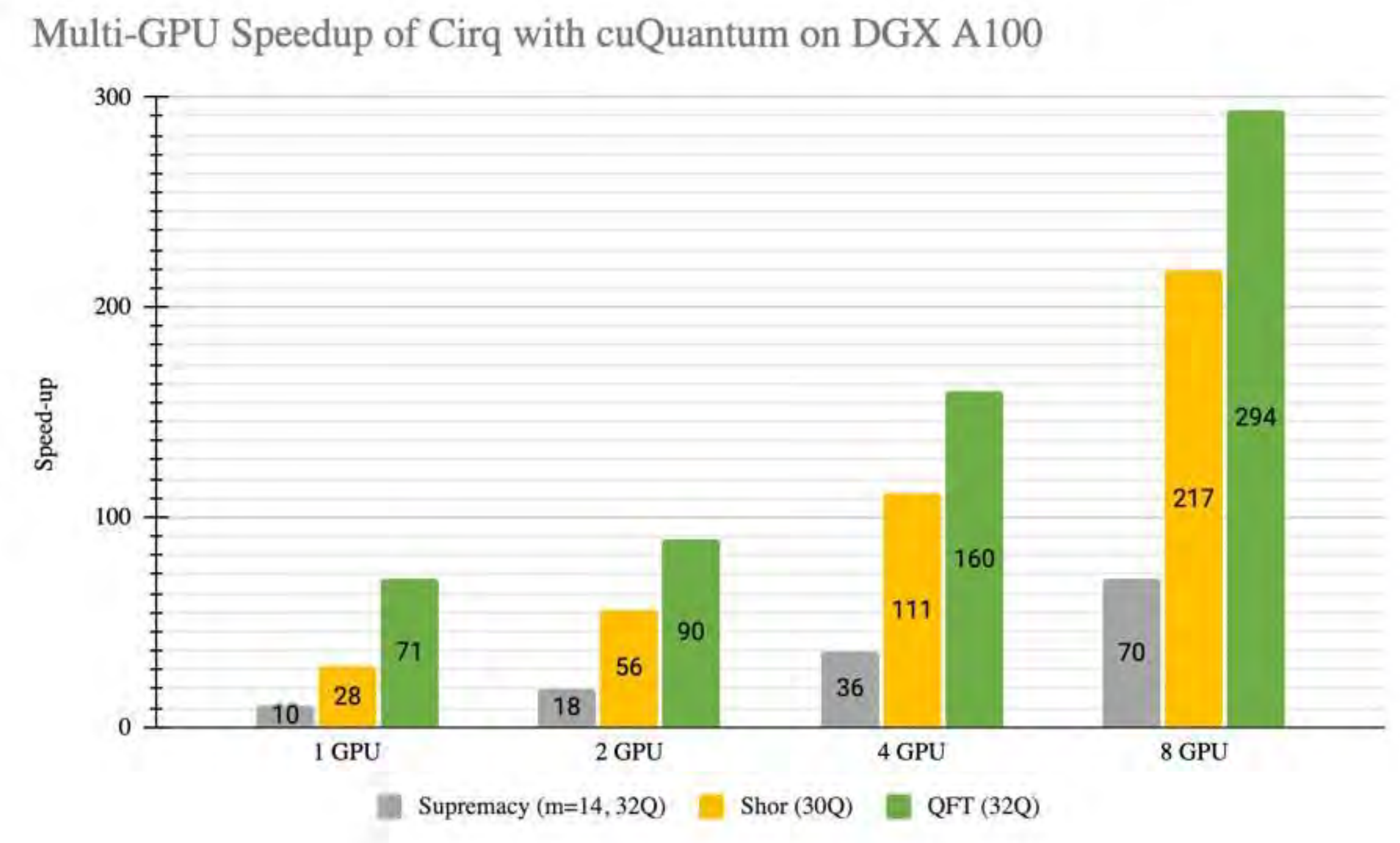
Fully integrated quantum simulation solution

- State-of-the-art *performance*
- Unmatched simulation *scale*

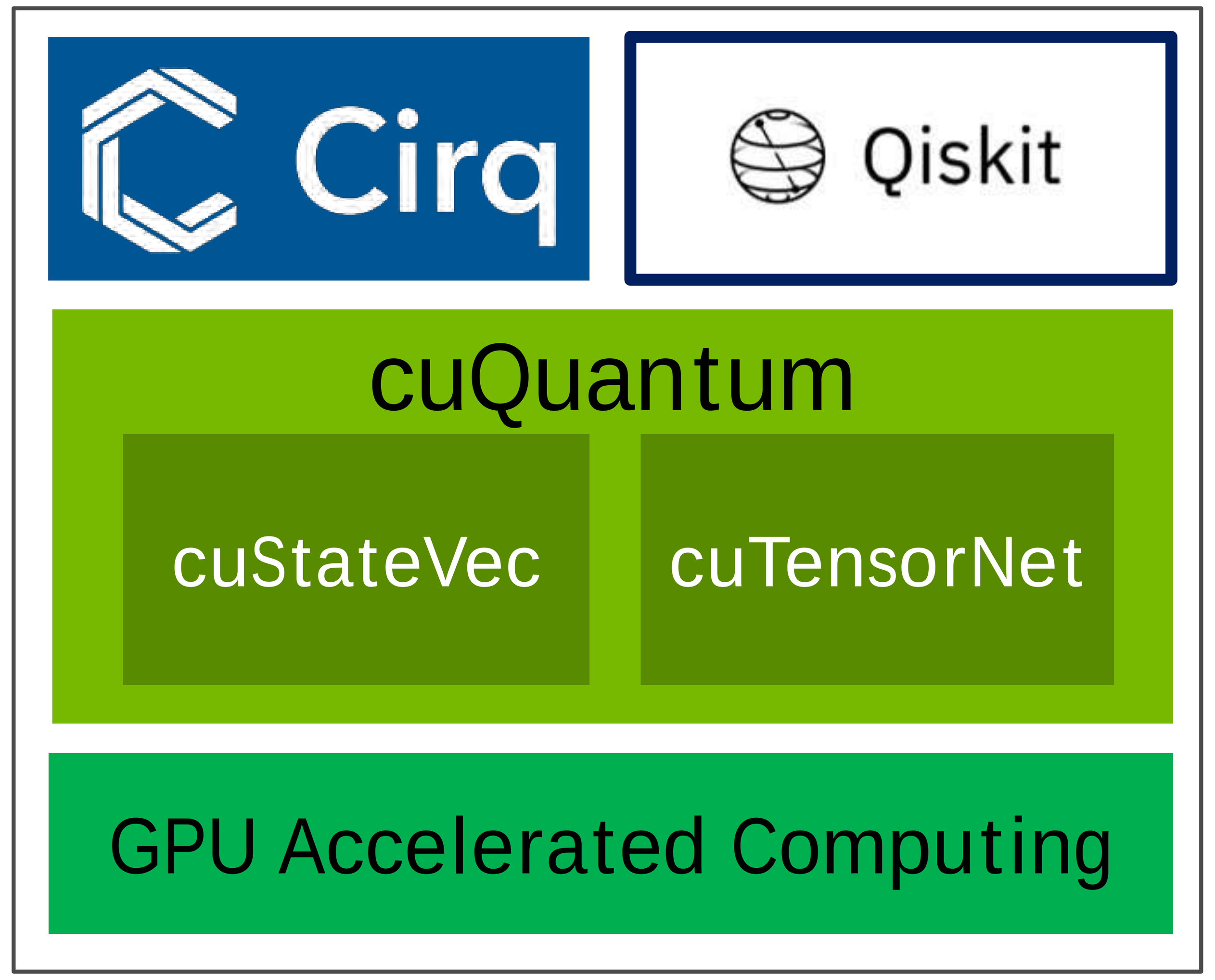
Reduce the simulation time by *orders of magnitude*

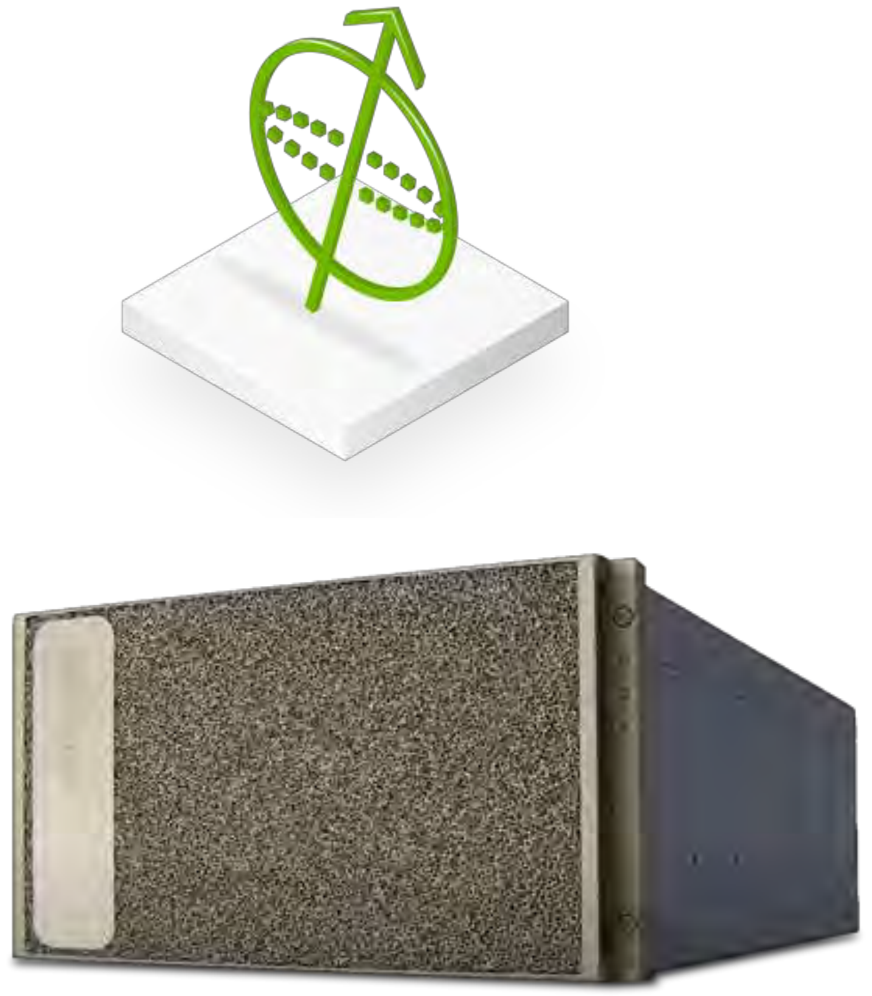
Simulate *thousands* of perfect or noisy qubits

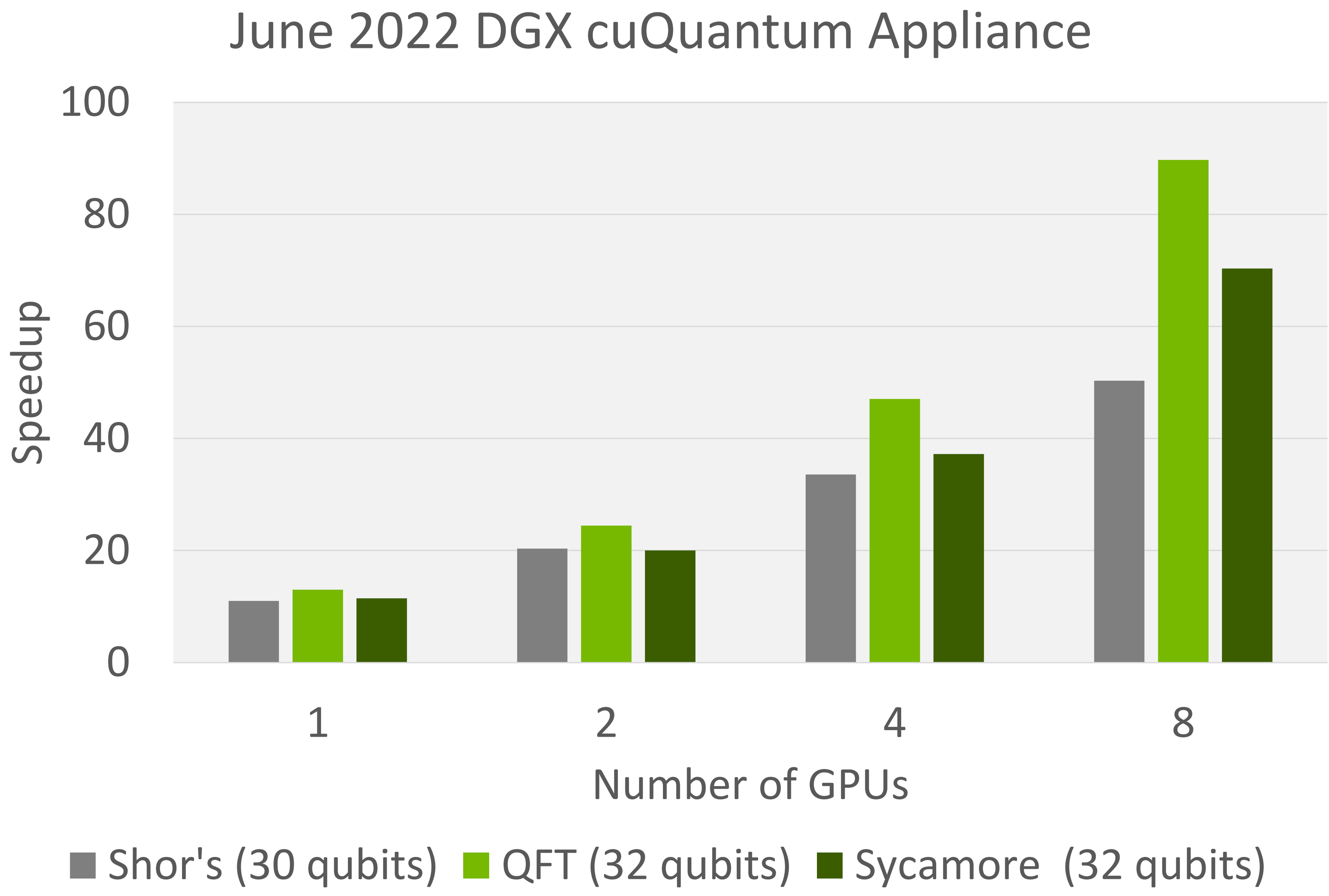
Up to 4.4X Perf since June



DGX cuQuantum Appliance





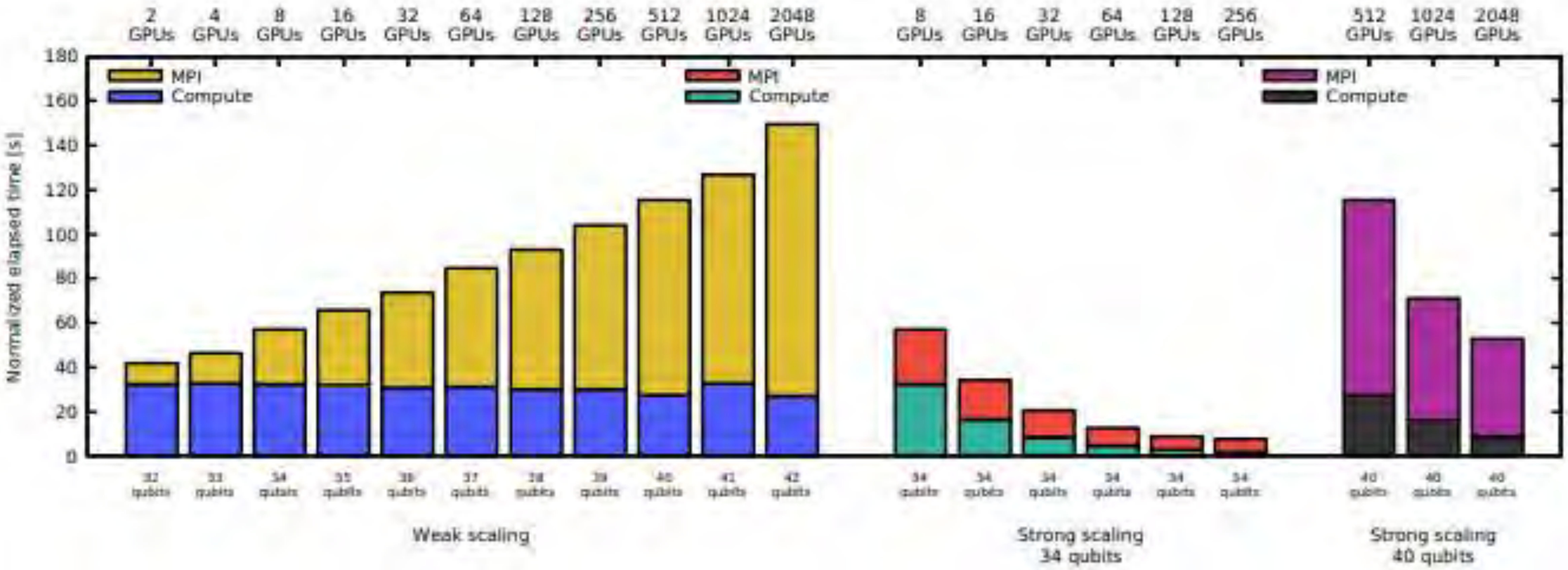


Speedup over AMD EPYC 7742 with 64 cores

NVIDIA

Accelerating MultiNode SV Simulation

MPI Dominates Processing Time

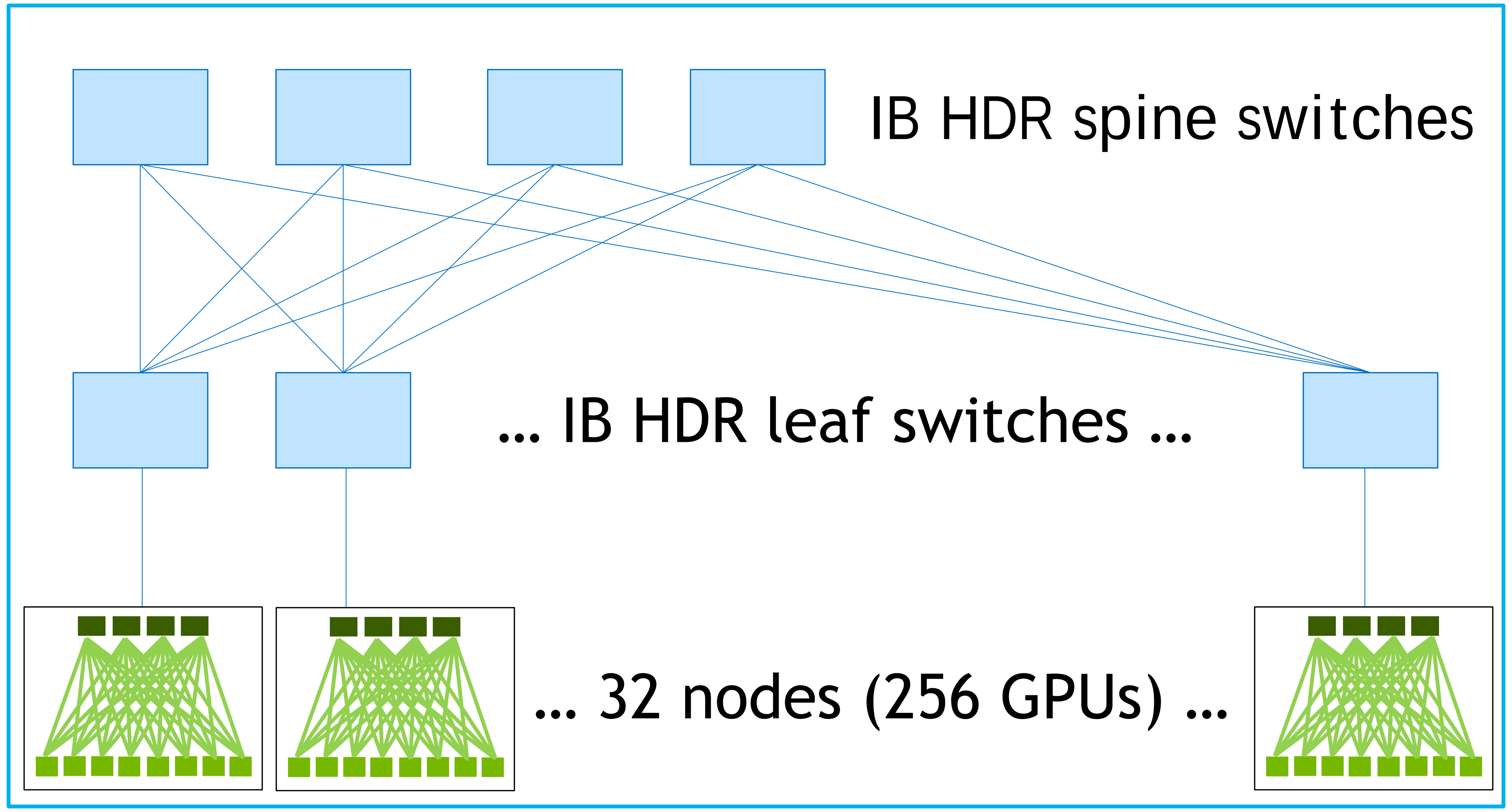


***GPU-accelerated simulations of quantum annealing and the quantum approximate optimization algorithm,** D. Willsch, M. Willsch, F. Jin, K. Michielsen and H. De Raedt, Computer Physics Communications 278, 108411 (2022)

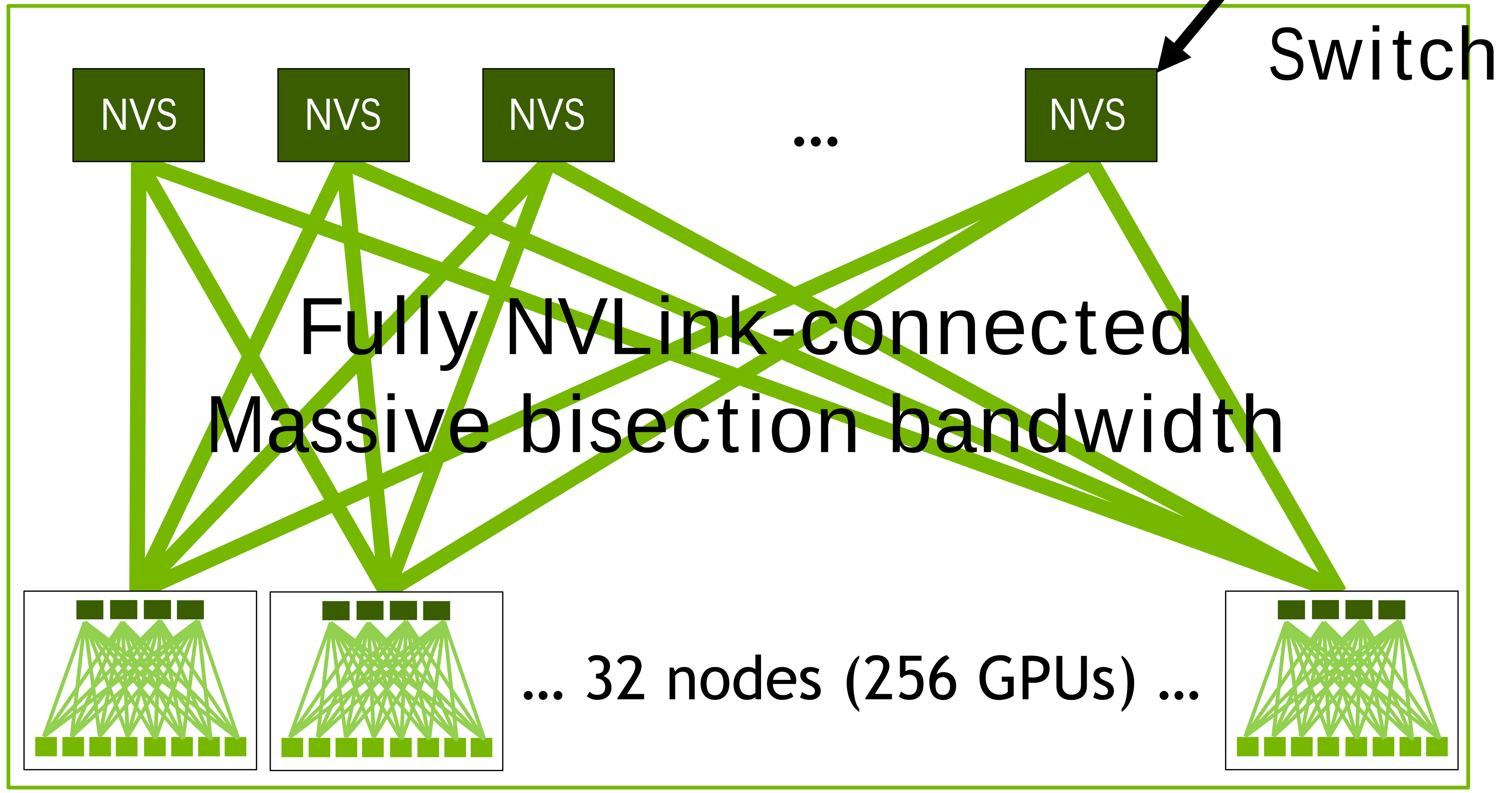
NVIDIA

SCALE-UP WITH NVLINK NETWORK

DGX A100 256 Pod



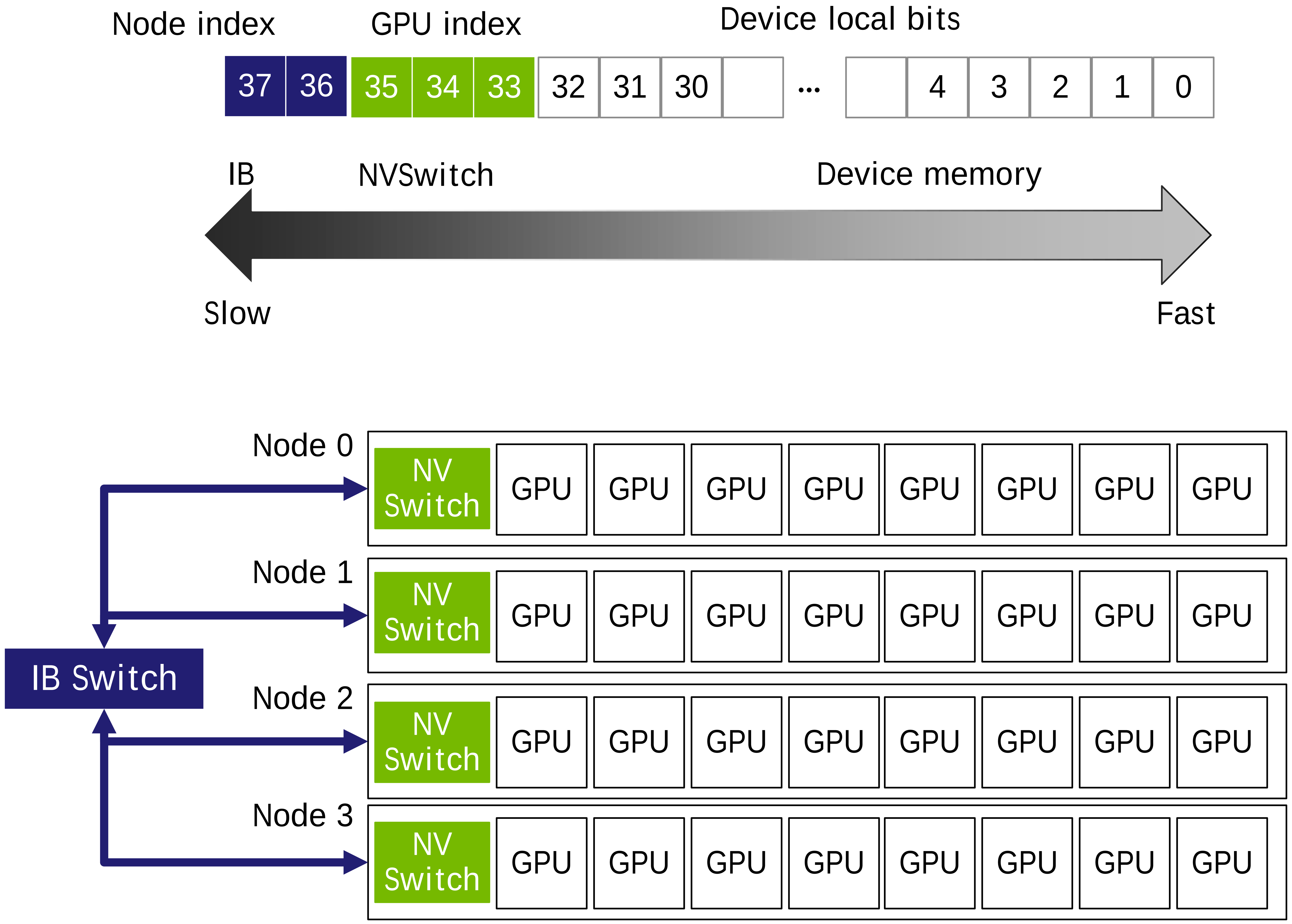
DGX H100 256 Pod



	A100 SuperPod			H100 SuperPod			Speedup	
	Dense PFLOP/s	Bisection [GB/s]	Reduce [GB/s]	Dense PFLOP/s	Bisection [GB/s]	Reduce [GB/s]	Bisection	Reduce
1 DGX / 8 GPUs	2.5	2,400	150	16	3,600	450	**1.5x**	**3x**
32 DGXs / 256 GPUs	80	6,400	100	512	57,600	450	**9x**	**4.5x**

COMING SOON: MULTINODE CUQUANTUM

- Multi-node simulation limited by internode gate operations requiring slower data transfer
- Our APIs will use index bit swapping along with recursive gate grouping to move as many gates as possible to within node, minimizing these transfers
- Can also use smart scheduling, overlapping gates with data transfer on other bits

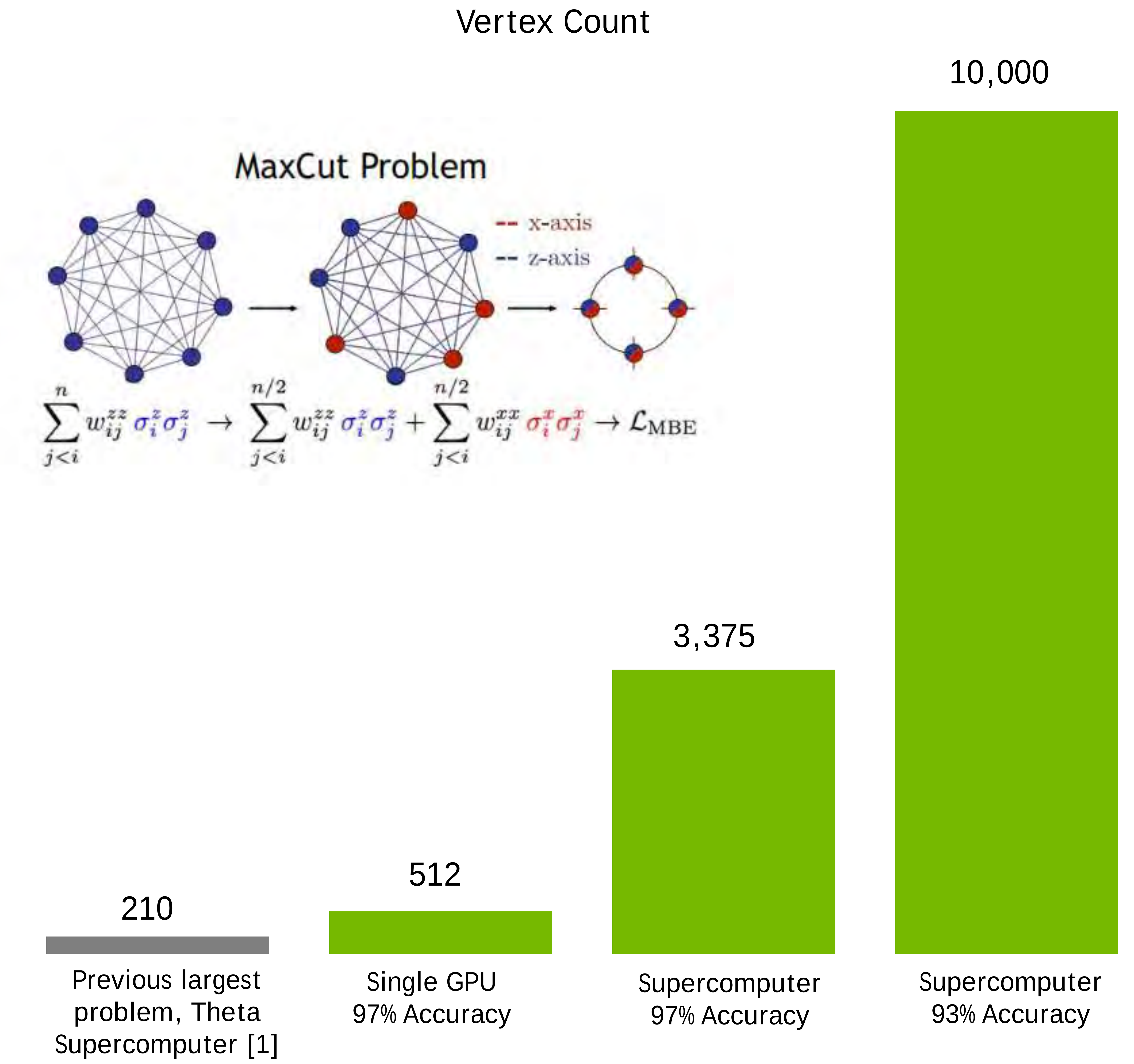


SCALING SIMULATIONS TO A SUPERCOMPUTER

NVIDIA's Selene DGX SuperPOD based supercomputer

- Using NVIDIA's Selene supercomputer
- Solved a 3,375 vertex problem (1,688 qubits/896 A100) with 97% accuracy
- Solved a 10,000 vertex problem (5,000 qubits) with 93% accuracy

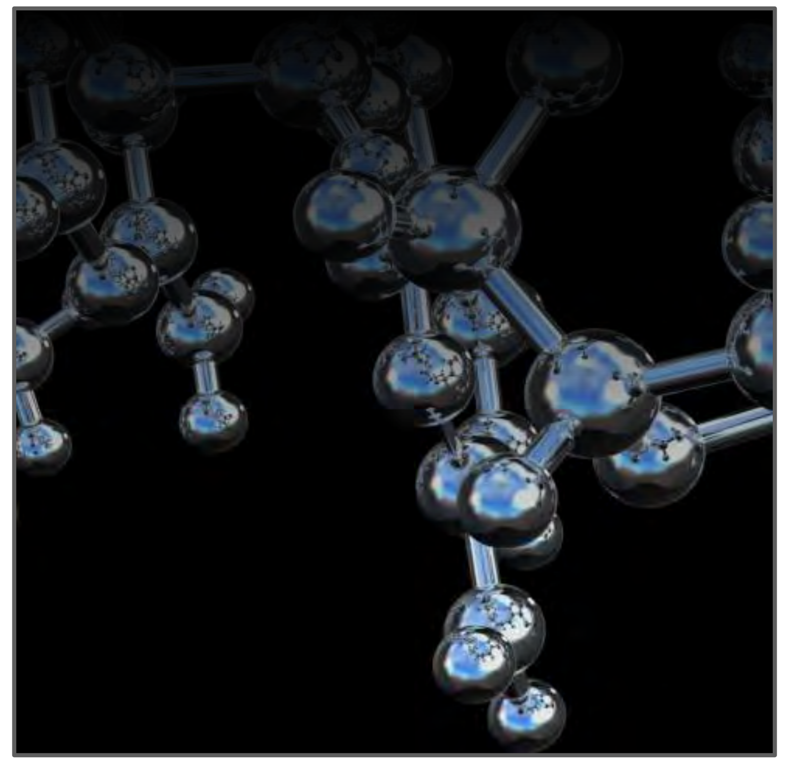
Patti et. al.(2021)/NVIDIA Research
Paper: https://arxiv.org/pdf/2106.13304.pdf
Code: https://github.com/tensorly/quantum



[1] Danylo Lykov et al, Tensor Network Quantum Simulator With Step-Dependent Parallelization, 2020
https://arxiv.org/pdf/2012.02430.pdf

WHAT ARE USERS DOING?

ABCI
AI Bridging Cloud Infrastructure



Grand Challenge

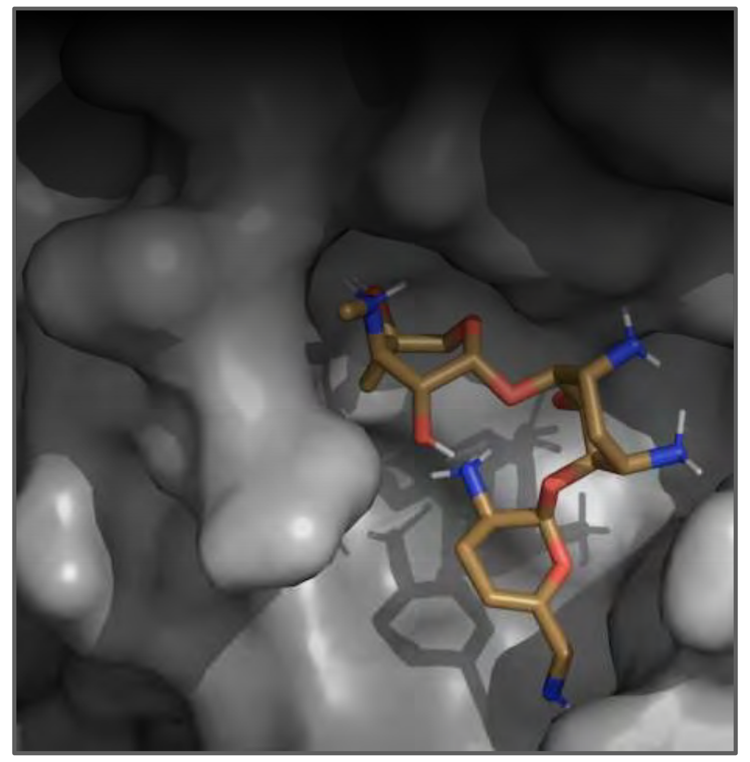
41 qubit full statevector simulation on 512 GPUs

XANADU

129 Qubits

Novel cut technique enables 129 qubit QAOA simulation on Perlmutter

Johnson & Johnson



VQE

100X Speedup for VQE applied to 7MER Protein Folding in collaboration with Strangeworks

NVIDIA

NVIDIA cuQuantum Ecosystem

aws · BMW · Deloitte. · FUJIFILM · IBM · Google Quantum AI · ORACLE · softserve

CLASSIQ · ColdQuanta · IONQ · PASQAL · PsiQuantum · QCWARE

QUANTINUUM · QUANTUM BRILLIANCE · rigetti · seeqc · XANADU · ZAPATA

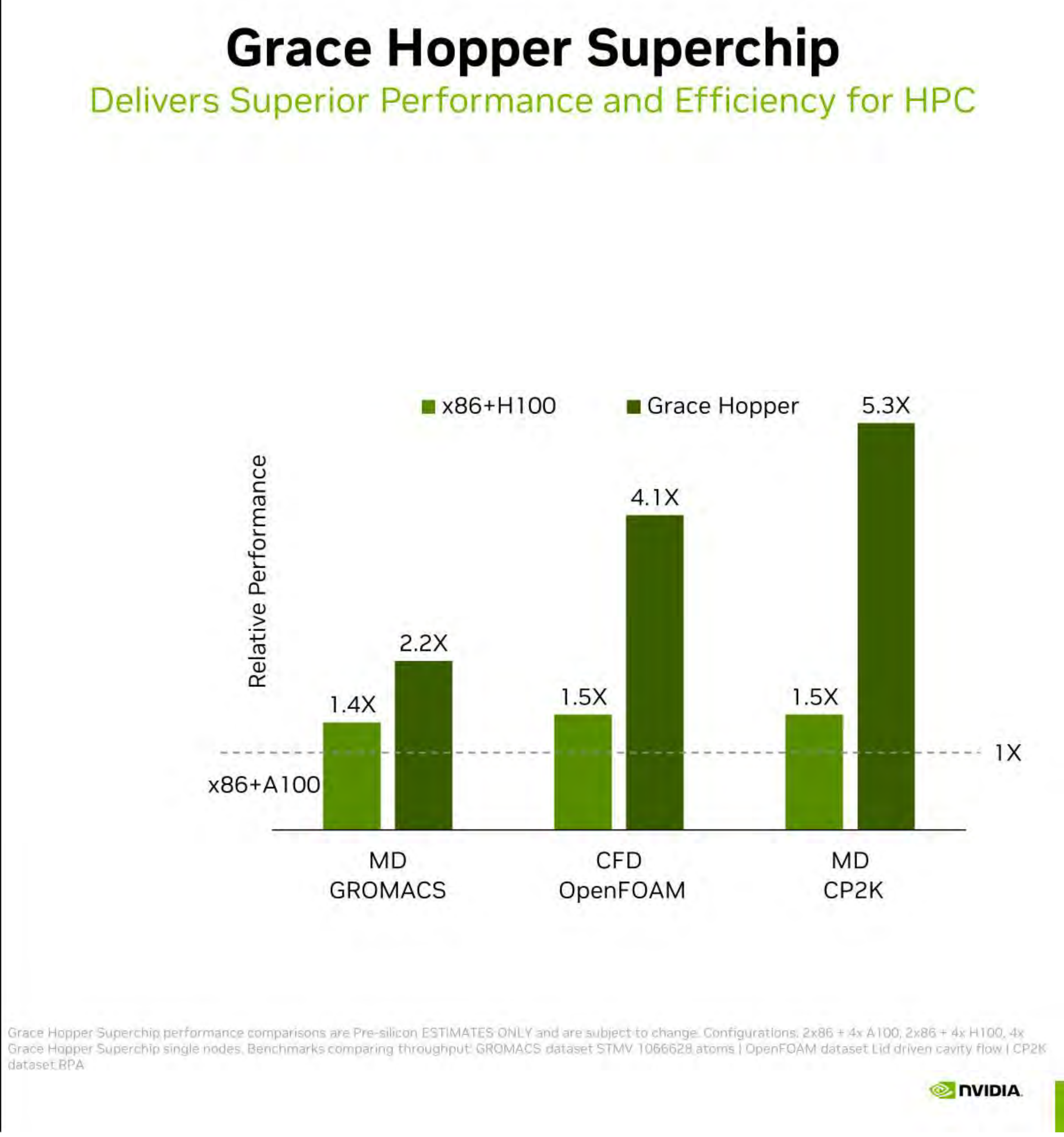
blueqat · Cirq · Orquestra® · PENNYLANE · QIBO · Qiskit · XACC QUANTUM FRAMEWORK

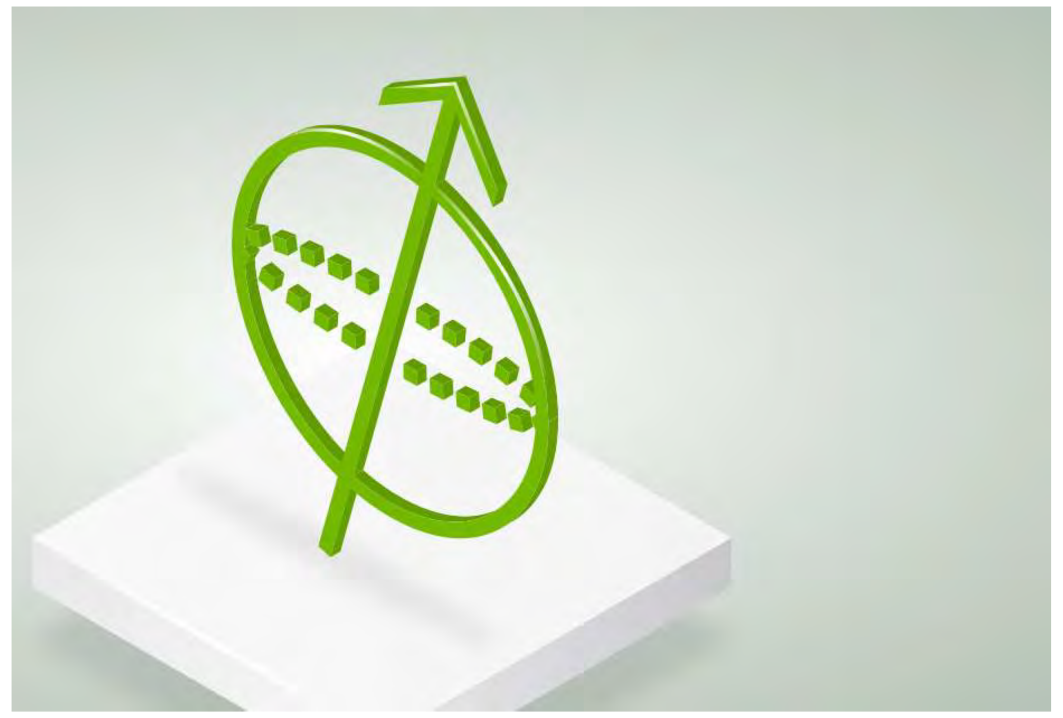
Argonne NATIONAL LABORATORY · JÜLICH Forschungszentrum · NERSC · OAK RIDGE National Laboratory

WHAT TO EXPECT NEXT?

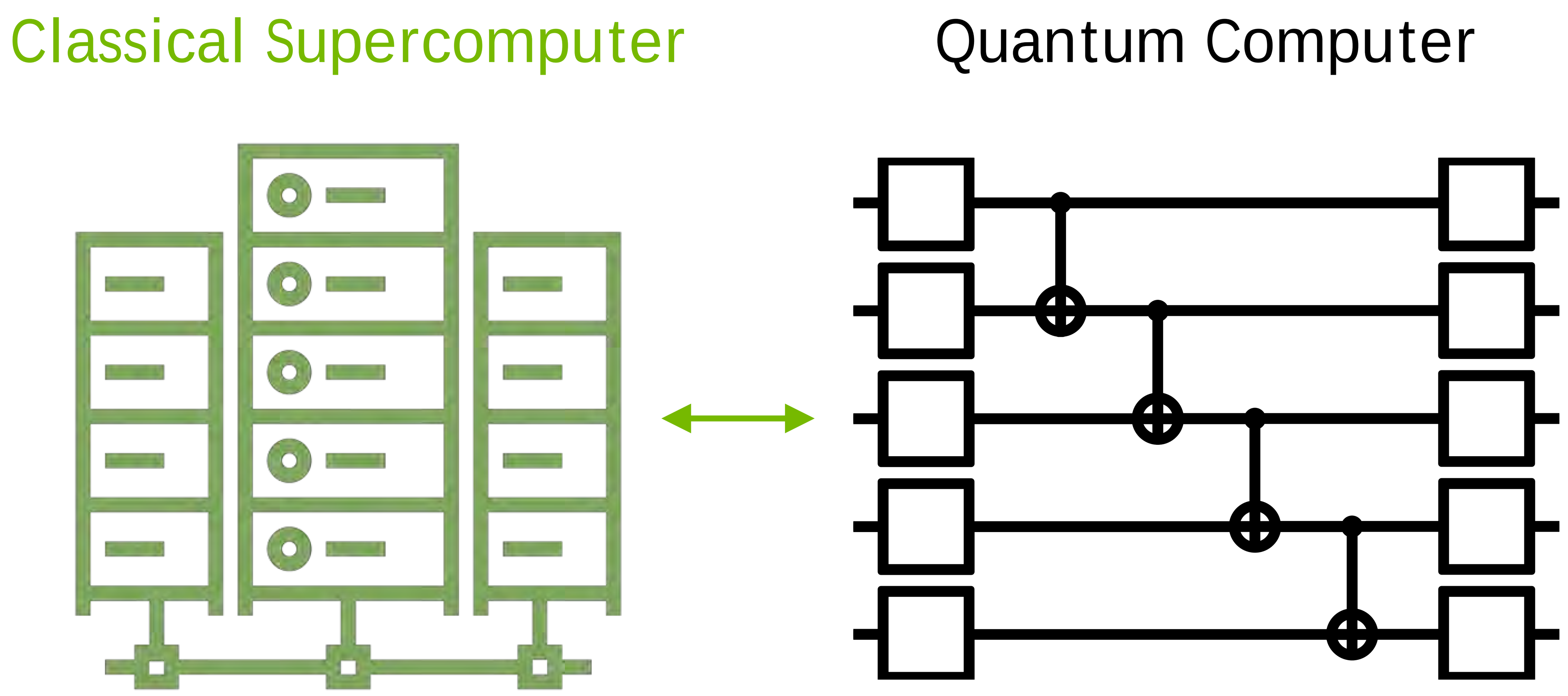
- November 2022 - cuQuantum 22.11 with
  - Support for approximate TN contractions (Matrix Product State formulation)
  - Container for multinode state vector simulation (Accelerating Qiskit-aer)

- March 2023 - cuQuantum 23.03
  - Library APIs for multinode state vector simulation
  - Support for batched simulation - enabling parallelized noisy simulations
- June 2023 - cuQuantum 23.06
  - Grace Hopper-optimized quantum circuit simulations





ECOSYSTEM GAPS LIMIT THE PROGRESS OF HYBRID QUANTUM APPLICATIONS



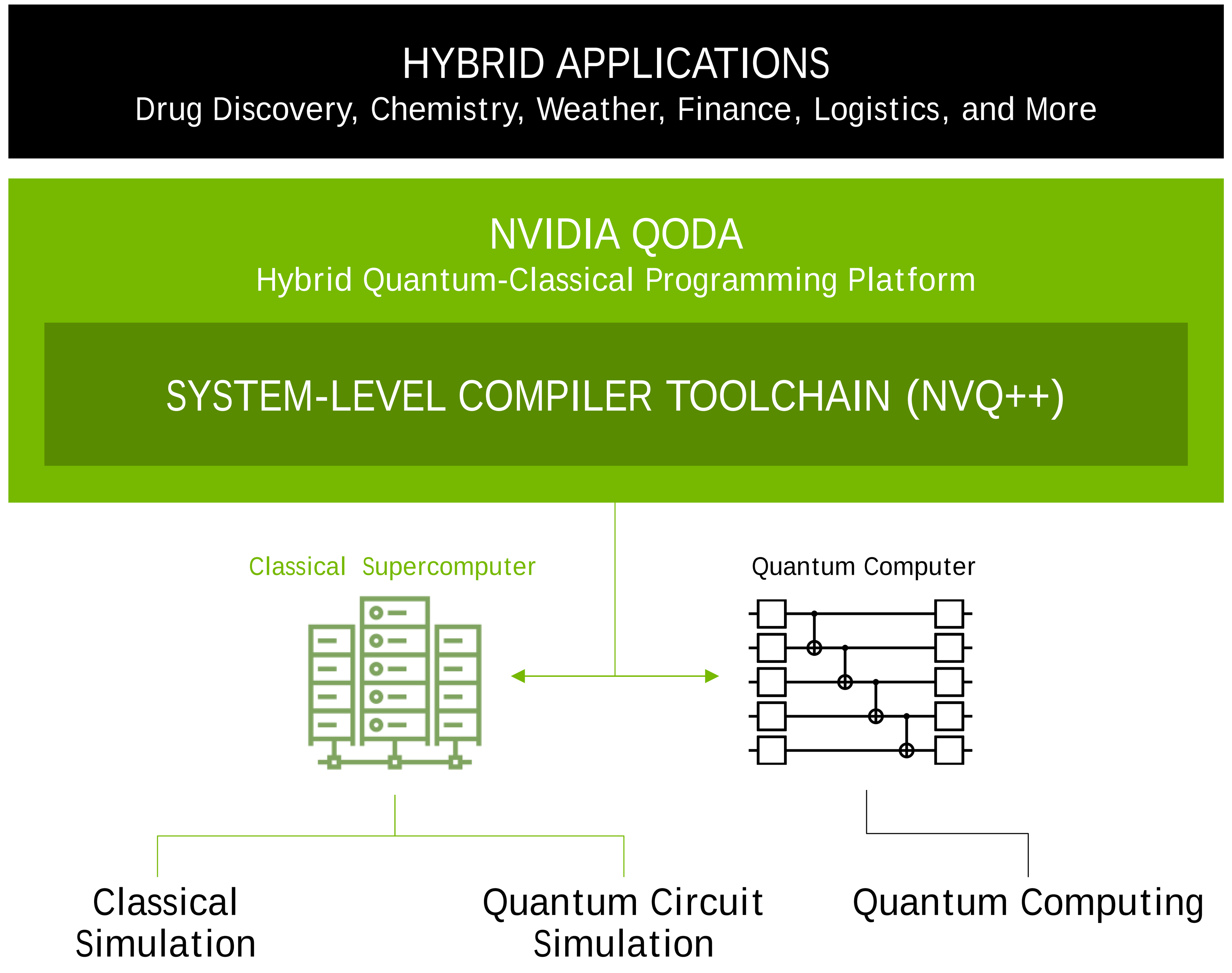
ECOSYSTEM CHALLENGES

- No Performant Software Stack
- Not Accessible To Domain Scientists
- Hybrid System Bottlenecks

ANNOUNCING NVIDIA QODA

THE PLATFORM FOR HYBRID QUANTUM-CLASSICAL COMPUTING

NVIDIA QODA PLATFORM



QODA FEATURES

Open and interoperable with today's applications

QPU-agnostic , emulated or physical

Compiler toolchain for hybrid systems

Kernel-based programming model

Single-source C++ and Python

Standard library of quantum algorithmic primitives

Unlocks performance, developer productivity, and cross-platform Interoperability

ANNOUNCING NVIDIA QODA

THE PLATFORM FOR HYBRID QUANTUM-CLASSICAL COMPUTING

NVIDIA QODA PLATFORM

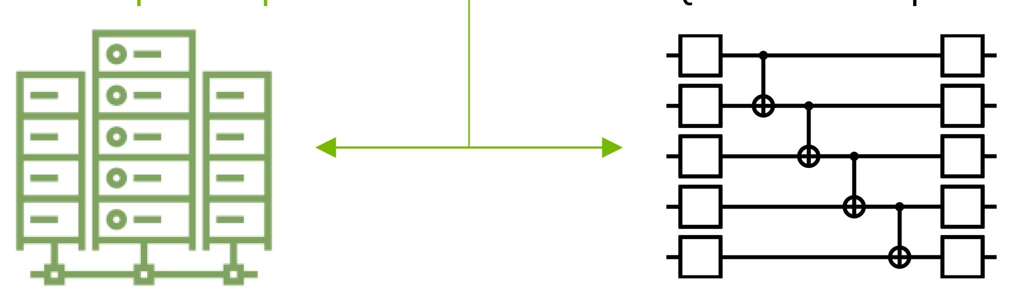
HYBRID APPLICATIONS
Drug Discovery, Chemistry, Weather, Finance, Logistics, and More

NVIDIA QODA
Hybrid Quantum-Classical Programming Platform

SYSTEM-LEVEL COMPILER TOOLCHAIN (NVQ++)

Classical Supercomputer

Quantum Computer



Classical Simulation

Quantum Circuit Simulation

Quantum Computing

QODA PLATFORM ECOSYSTEM

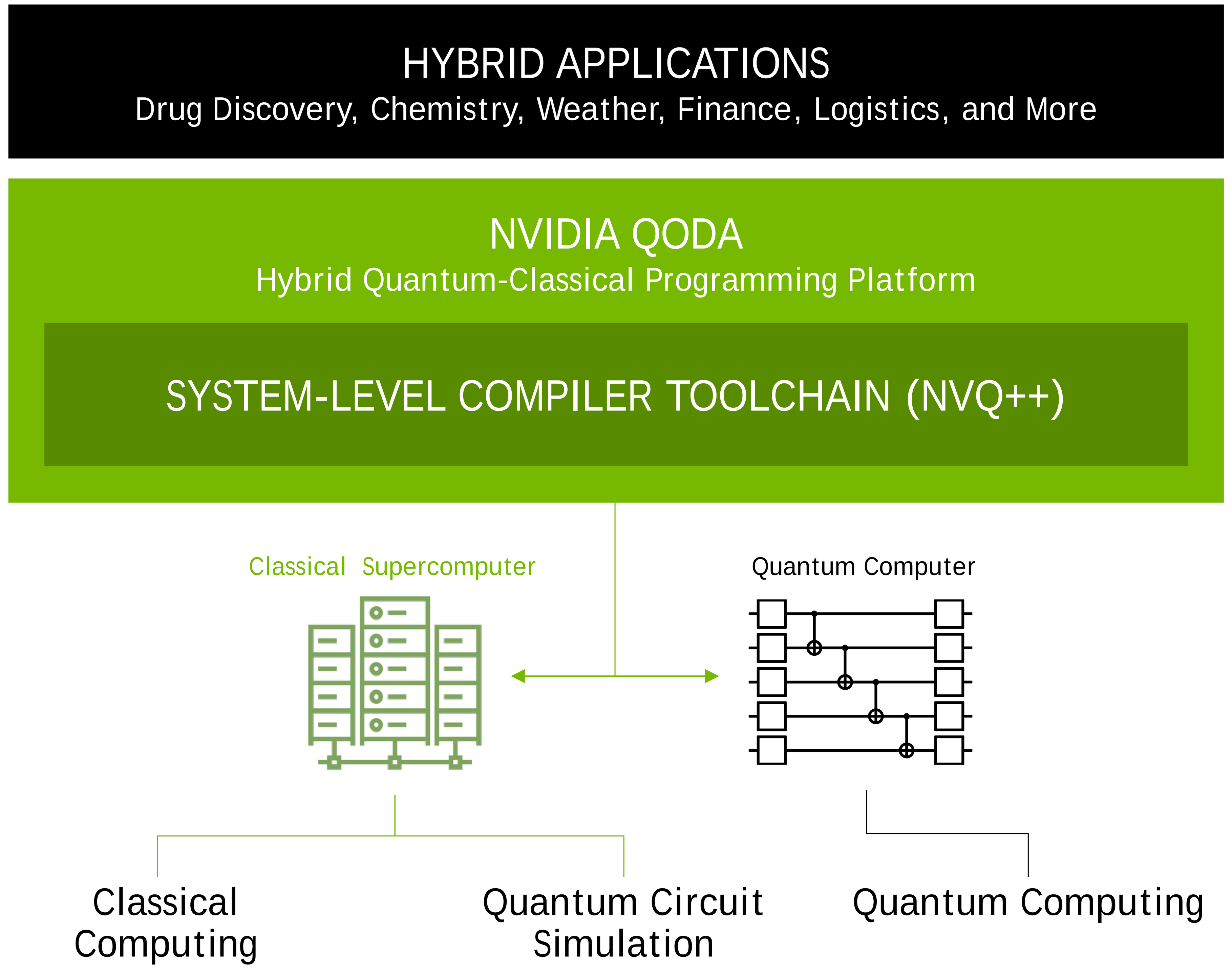
IQM
PASQAL
QUANTUM BRILLIANCE
QUANTINUUM
rigetti
XANADU
Microsoft
Qubit pharmaceuticals
CLASSIQ
QCWARE
ZAPATA
JÜLICH Forschungszentrum
NERSC
OAK RIDGE National Laboratory

NVIDIA

ANNOUNCING NVIDIA QODA

THE PLATFORM FOR HYBRID QUANTUM-CLASSICAL COMPUTING

NVIDIA QODA PLATFORM



Familiar to Domain Scientists

```
// Define a QODA Quantum Kernel.
auto ansatz = [] (double theta) __qpu__ {
  qoda::qbit q, r;
  x(q);
  ry(theta, r);
  cnot(r, q);
};

// Define the Hamiltonian via Pauli tensor products.
qoda::spin_op H = 5.907 - 2.1433 * x(0) * x(1)
                - 2.1433 * y(0) * y(1)
                + .21829 * z(0) - 6.125 * z(1);

qoda::nlopt::cobyla opt;

// Run Variational Quantum Eigensolver with 1 param.
auto [min_e, opt_p] = qoda::vqe(ansatz, H, opt, 1);
```

ANNOUNCING NVIDIA QODA

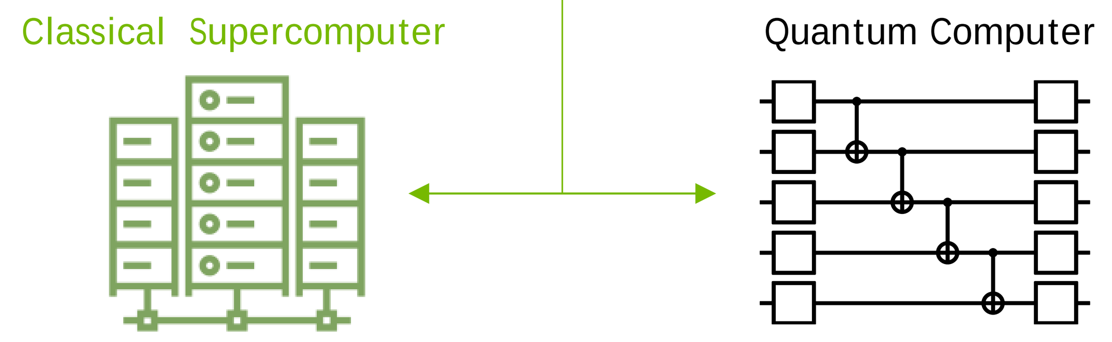
THE PLATFORM FOR HYBRID QUANTUM-CLASSICAL COMPUTING

NVIDIA QODA PLATFORM

HYBRID APPLICATIONS
Drug Discovery, Chemistry, Weather, Finance, Logistics, and More

NVIDIA QODA
Hybrid Quantum-Classical Programming Platform

SYSTEM-LEVEL COMPILER TOOLCHAIN (NVQ++)



Classical Simulation | Quantum Circuit Simulation | Quantum Computing

Interoperable with GPU Supercomputing

```
auto cnts = qoda::sample(q, …);
```

ISO Standard Parallelism

```
std::sort(std::execution::par, …);
```

CUDA

```
kernel<<<…>>>(…);
cudaDeviceSynchronize();
```

OpenMP

```
#pragma omp target teams loop
for (…) …
```

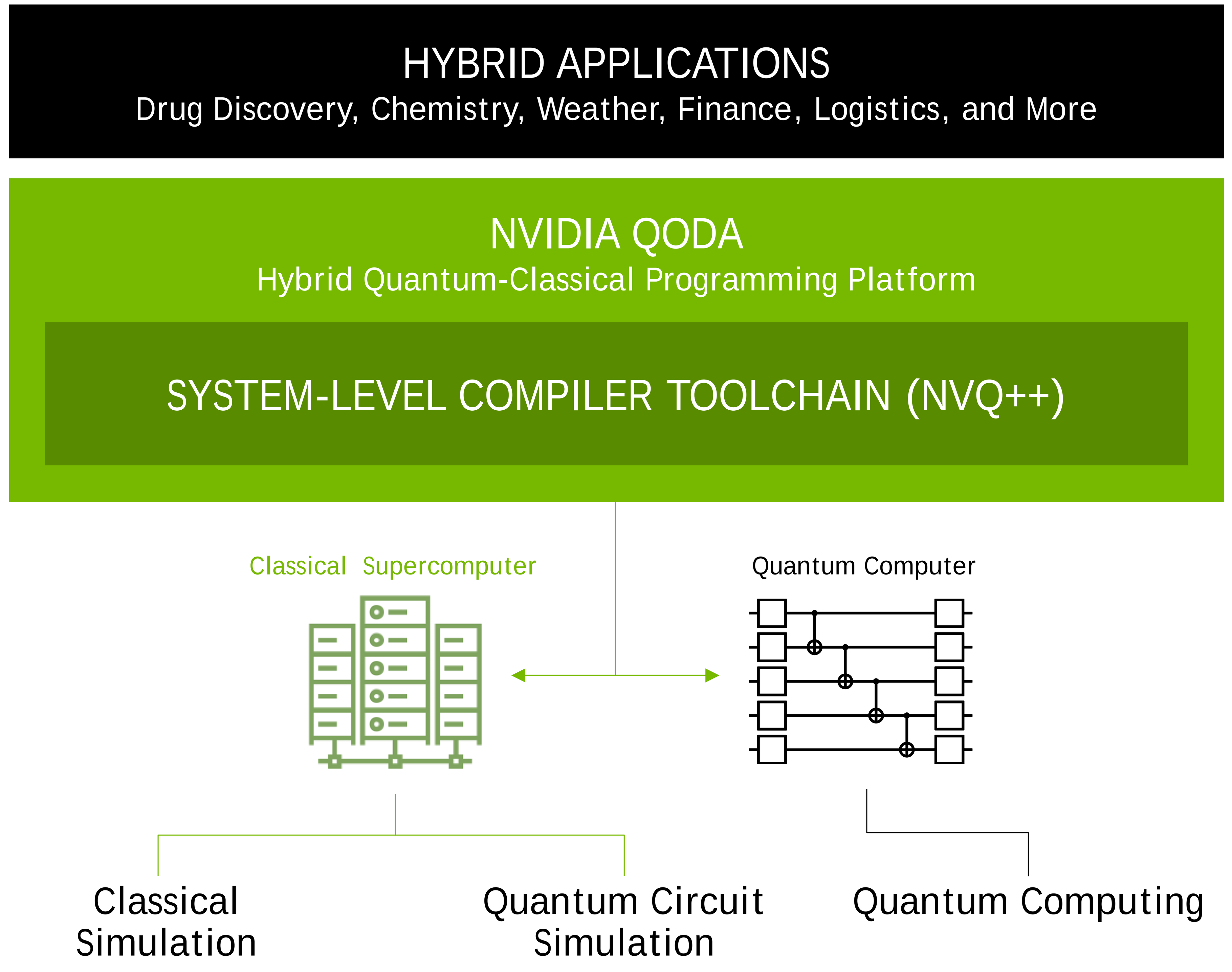
OpenACC

```
#pragma acc parallel loop
for (…) …
```

ANNOUNCING NVIDIA QODA

THE PLATFORM FOR HYBRID QUANTUM-CLASSICAL COMPUTING

NVIDIA QODA PLATFORM

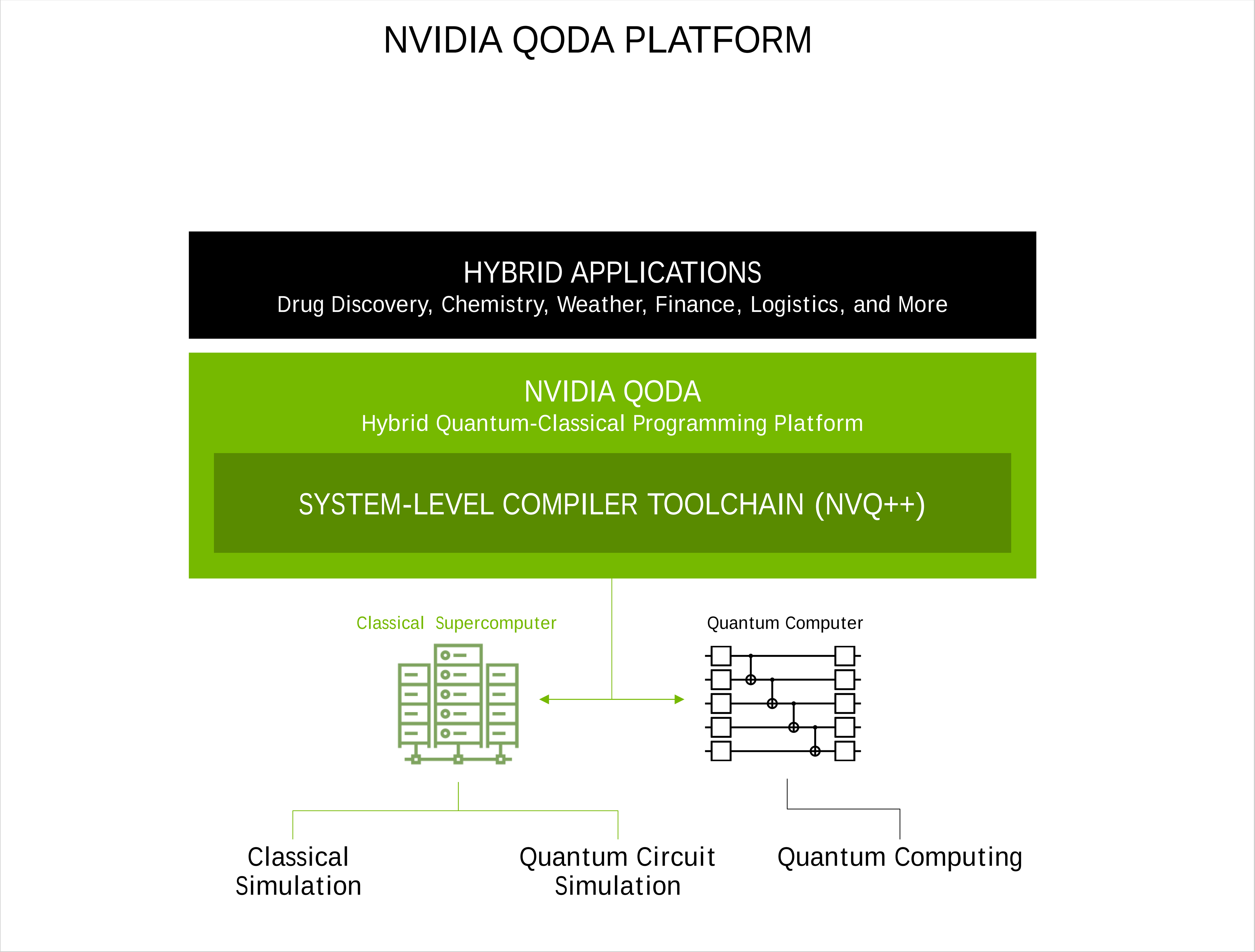


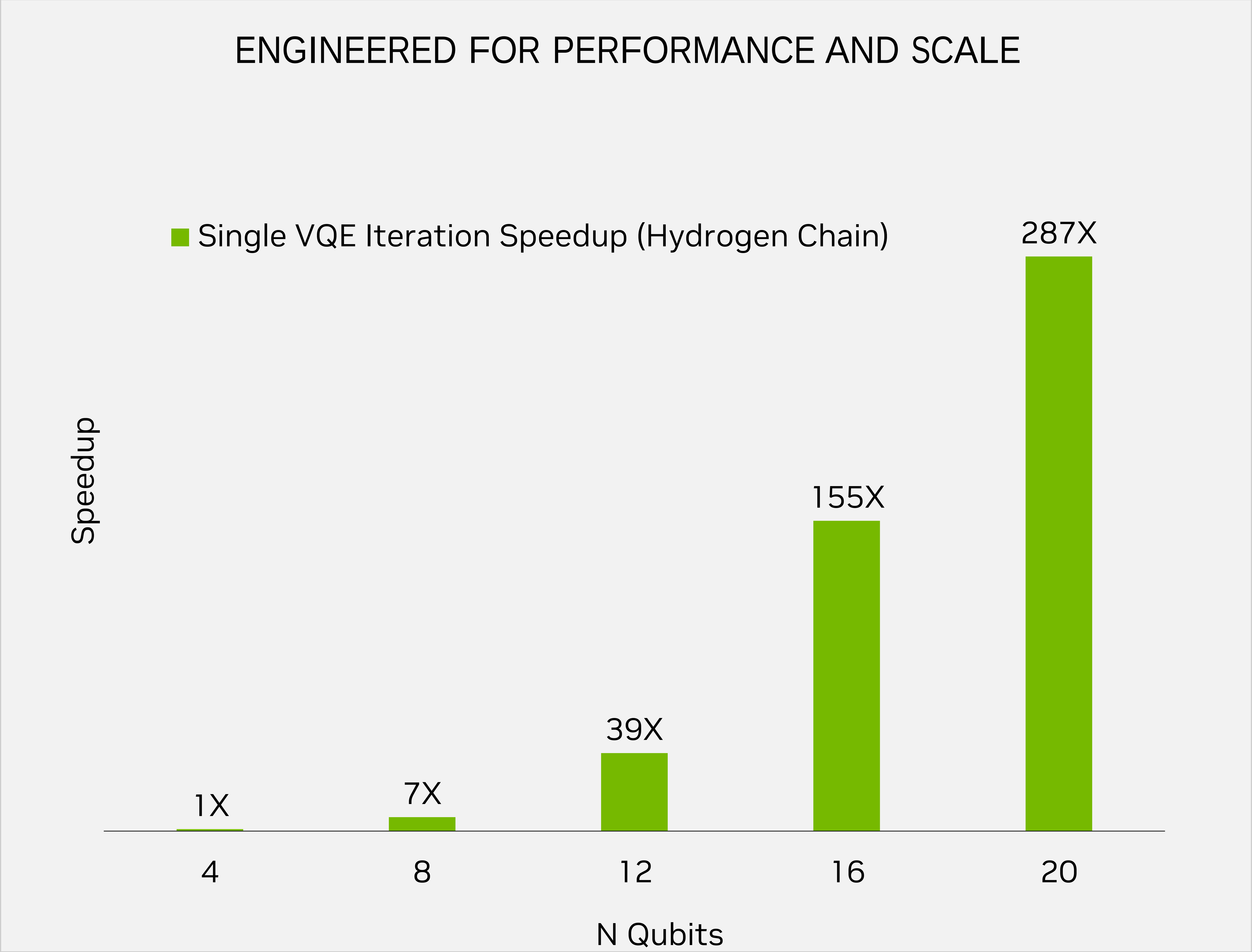
Interoperable with GPU Supercomputing

```
// Compute expectation values with QPU.
qoda::spin_op h = ...;
std::vector<double> sig_exps;
for (auto& pauli_op : generate_pauli_permutations(h.n_qubits()))
  sig_exps.push_back(qoda::observe(qite, pauli_op, h.n_qubits()));

...
// Compute LU Factorization of S_mat on the GPU.
auto dim = std::pow(2, h.n_qubits());
cusolverDnXgetrf(handle, params, dim, dim, CUDA_C_64F, S_mat,
                 lda, NULL, CUDA_C_64F, buffer_on_device,
                 bytes_on_device, buffer_on_host,
                 bytes_on_host, info);
```

ANNOUNCING NVIDIA QODA

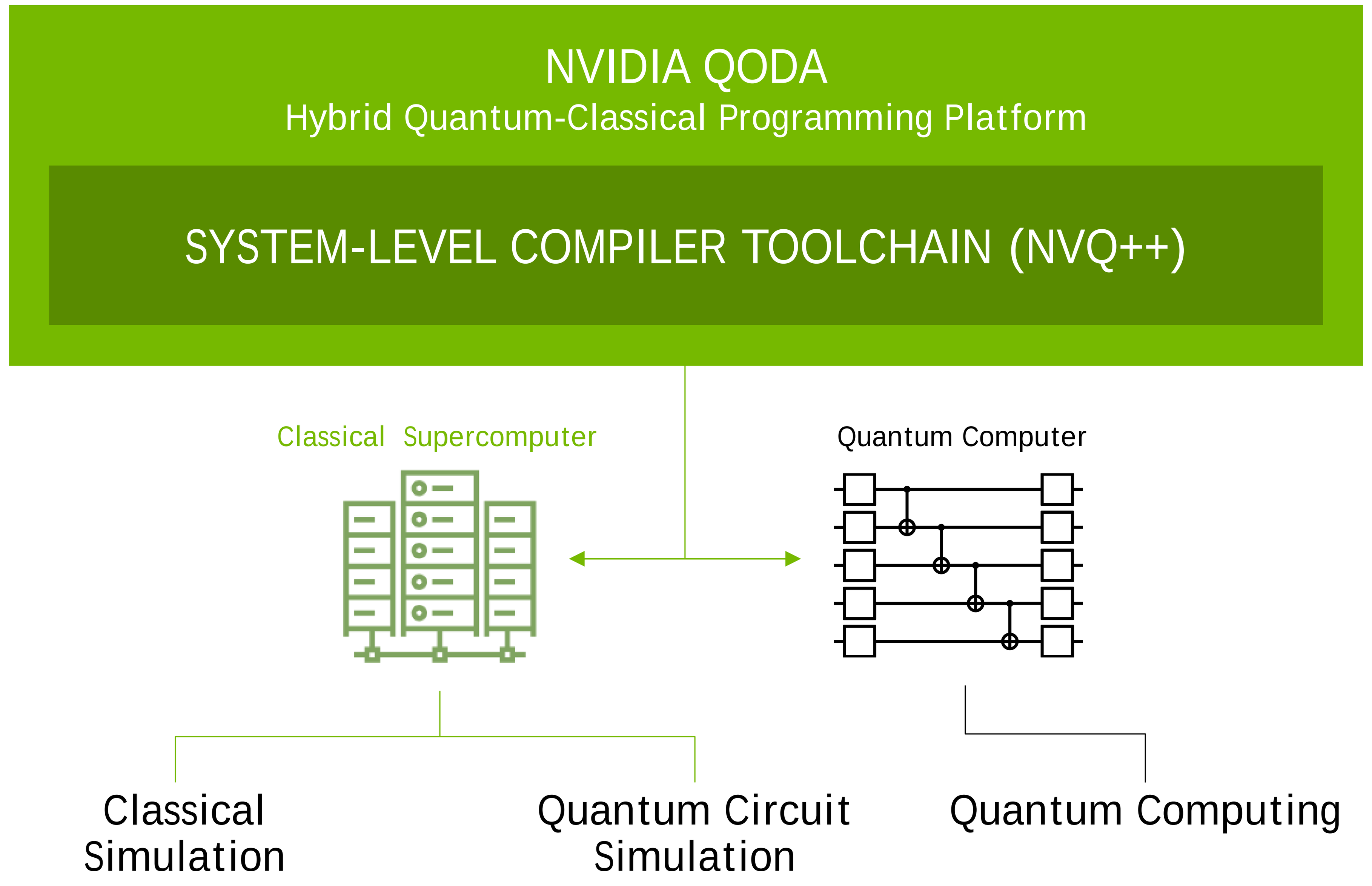
THE PLATFORM FOR HYBRID QUANTUM-CLASSICAL COMPUTING



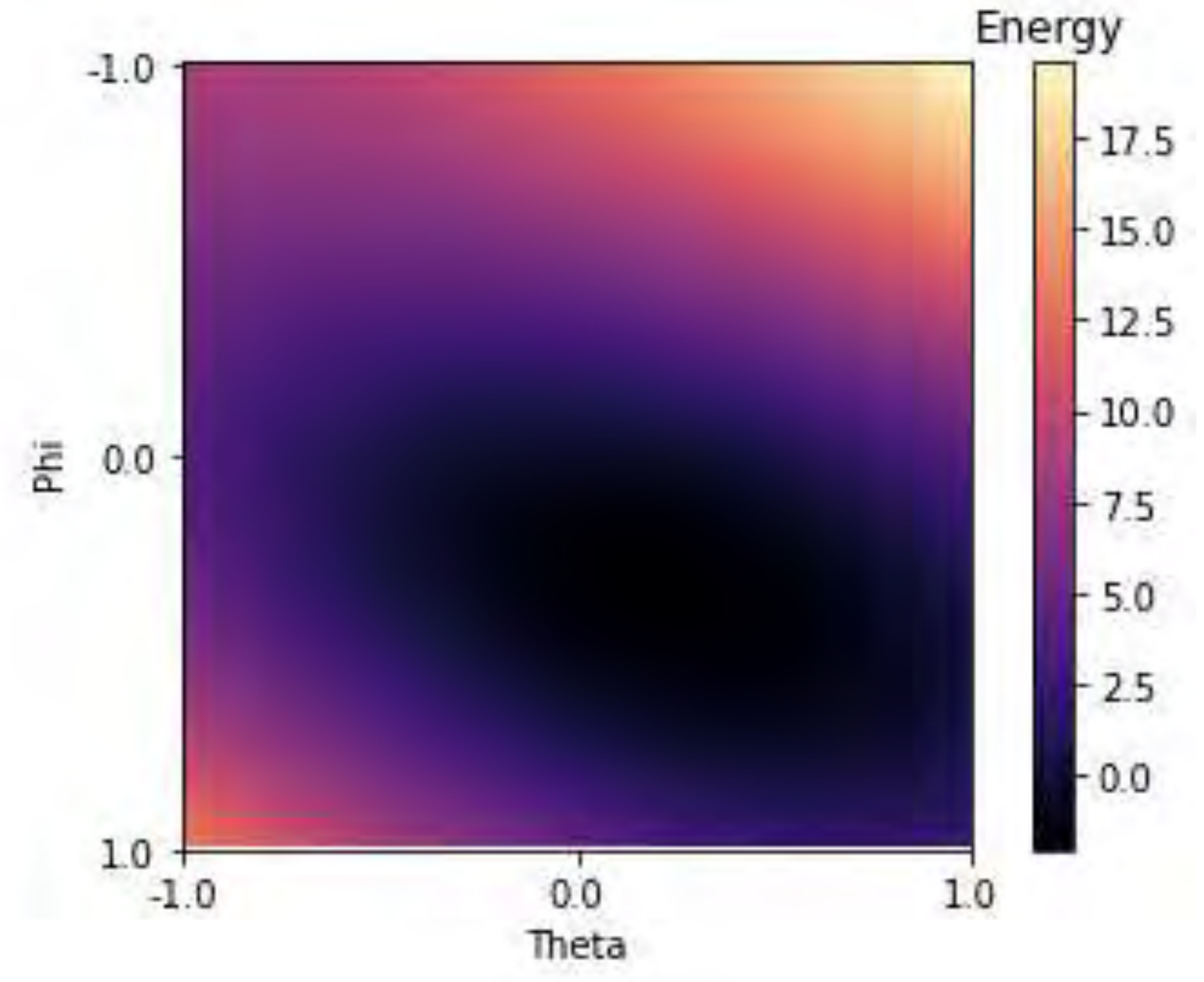


VQE: NVQ++/cuStateVec on A100 GPU vs Leading Pythonic Framework and SImulator with Thrust on A100 GPU

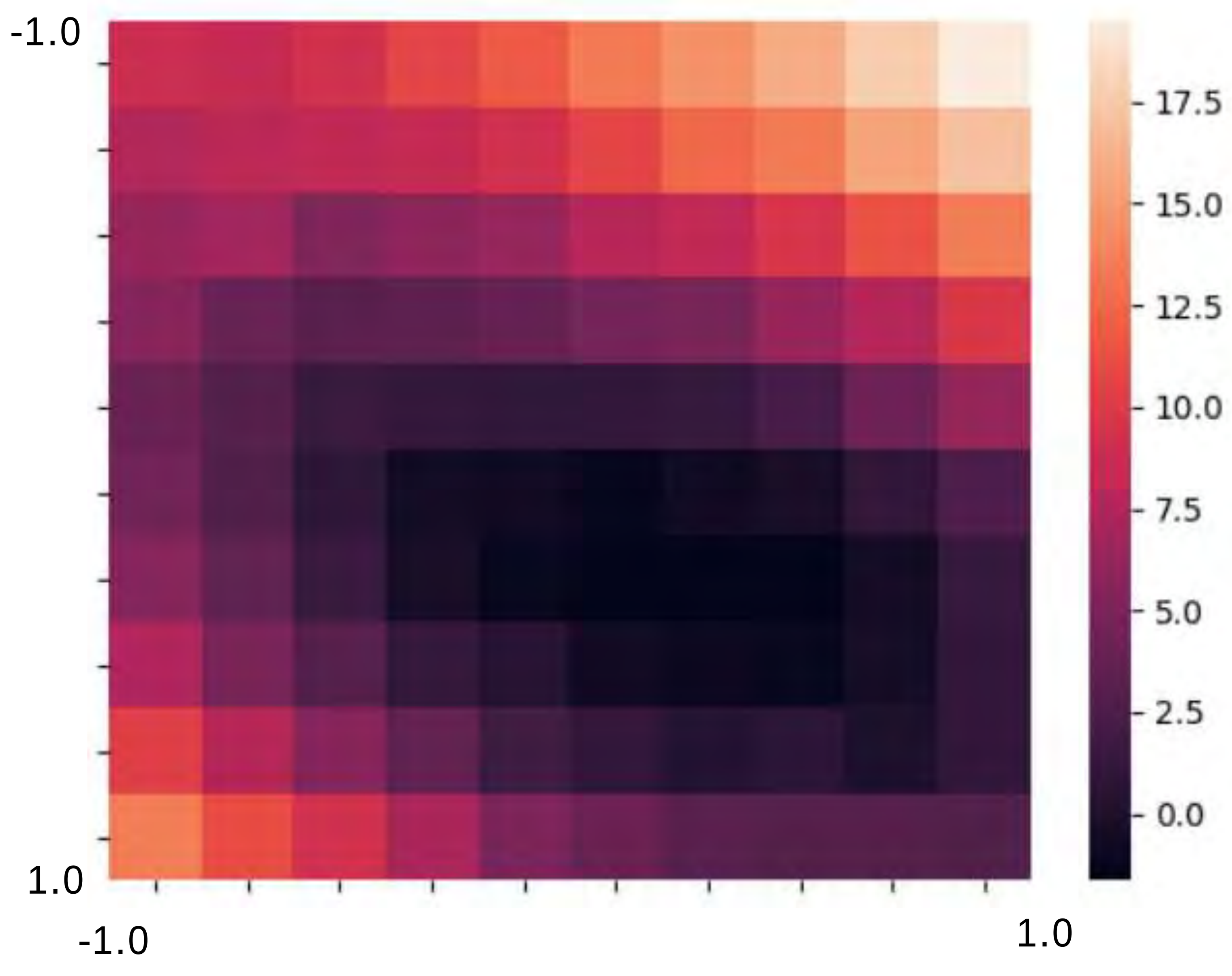
QODA ON QUANTINUUM



`nvq++ -qpu=cuquantum vqe.cpp`



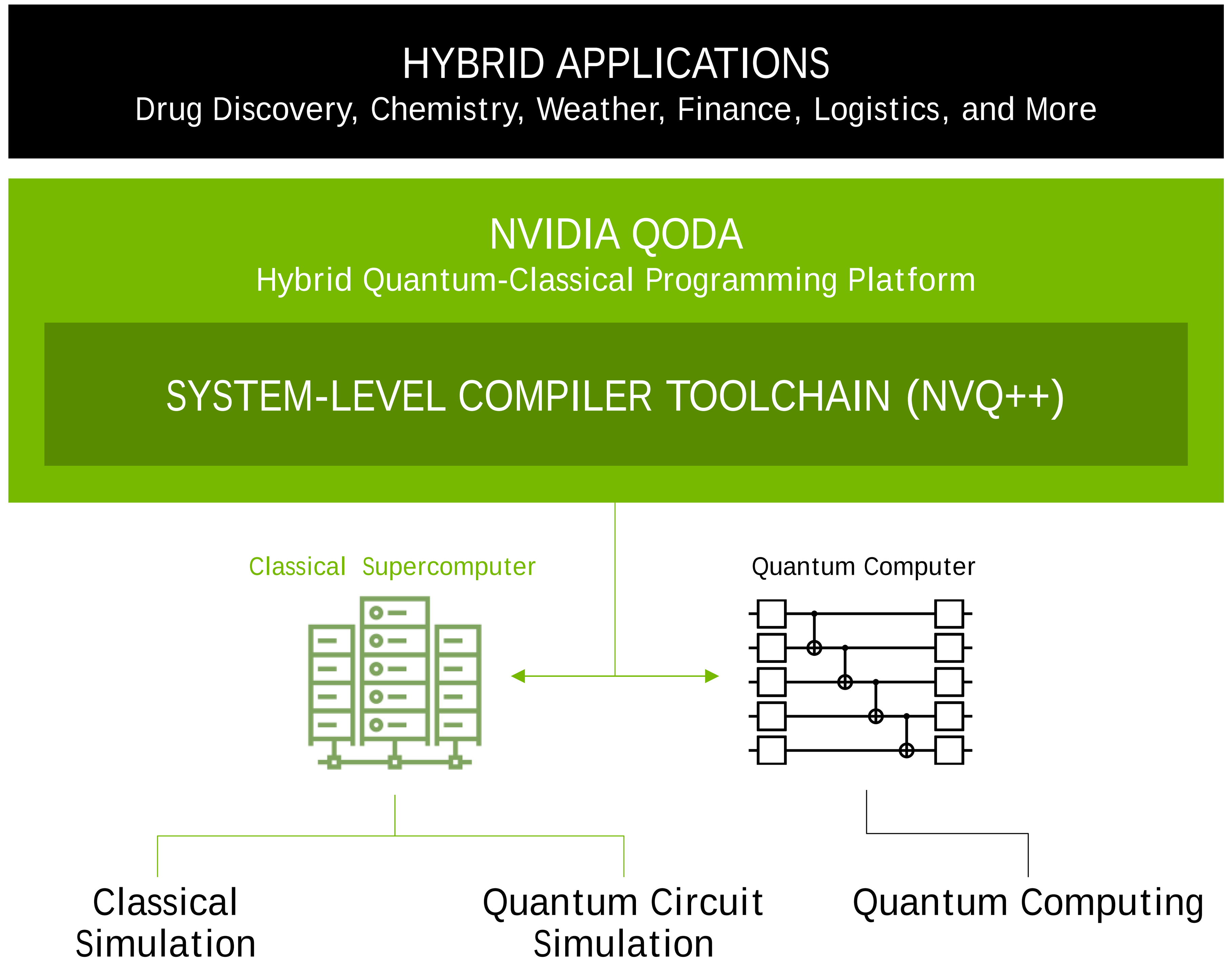
`nvq++ -qpu=quantinuum:h1 vqe.cpp`



Introducing NVIDIA QODA

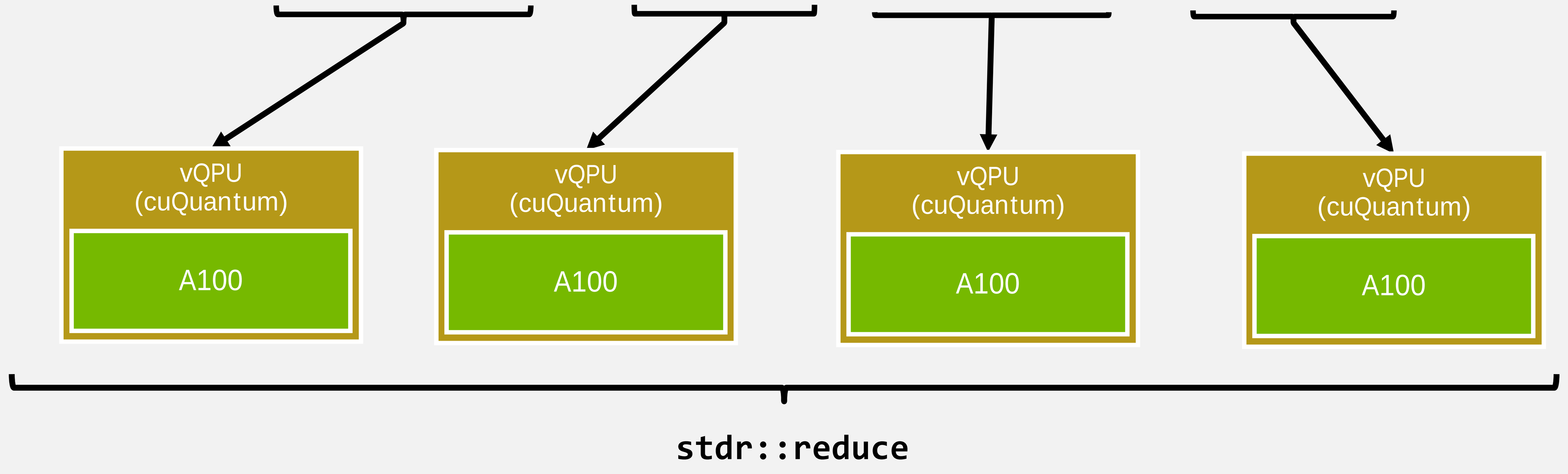
Enabling Innovative Quantum Systems Research

NVIDIA QODA PLATFORM




```
std::vector<qoda::observe_sender<double>> subs;
for (auto qpu : qoda::all_qpus()) {
 auto sub_H = H.subspan(qpu.idx() * terms_per_qpu, (qpu.idx() + 1) *
terms_per_qpu);
  subs.emplace_back(
    qoda::observe_async(qpu, sub_H, ansatz, ...));
}
auto sum = stdr::reduce(std::execution::par, qoda::when_all(subs), 0.0);
```

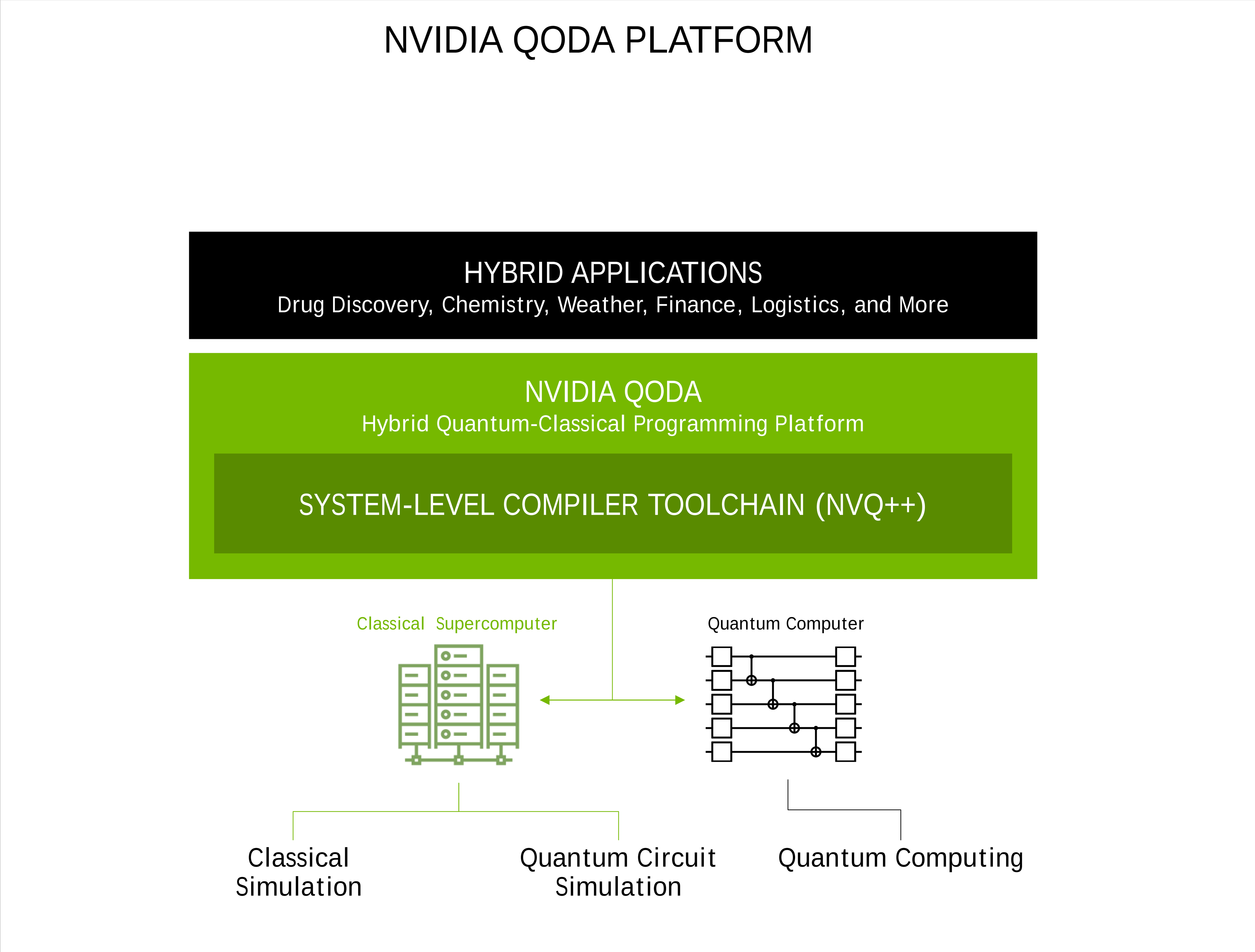
$$H = P_0 + P_1 + P_2 + \ldots + P_i + P_{i+1} \ldots + P_j + P_{j+1} + \ldots + P_{n-1} + P_n$$

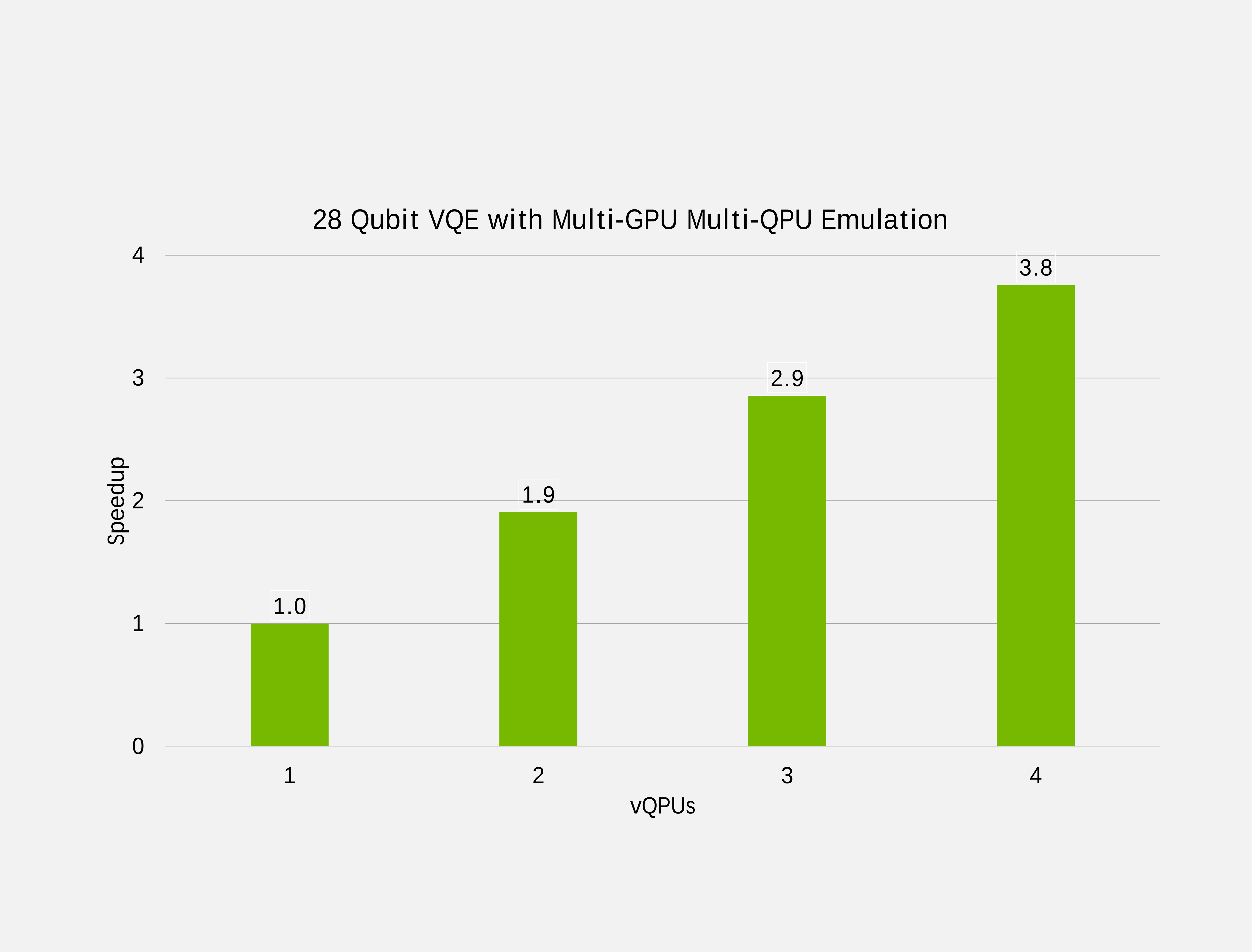


VQE: NVQ++/cuStateVec on A100 GPU vs Leading Pythonic Framework and SImulator with Thrust on A100 GPU

Introducing NVIDIA QODA

Enabling Innovative Quantum Systems Research





VQE: NVQ++/cuStateVec on A100 GPU vs Leading Pythonic Framework and SImulator with Thrust on A100 GPU

NVIDIA DGX QUANTUM APPLIANCE AND QODA DEVELOPMENT PLATFORM

A Scalable Platform for Hybrid Quantum Computing Research and Deployment

CLOUD PROVIDERS: Alibaba Cloud, aws, BAIDU AI CLOUD, Google Cloud, Microsoft Azure, ORACLE Cloud Infrastructure, Tencent Cloud

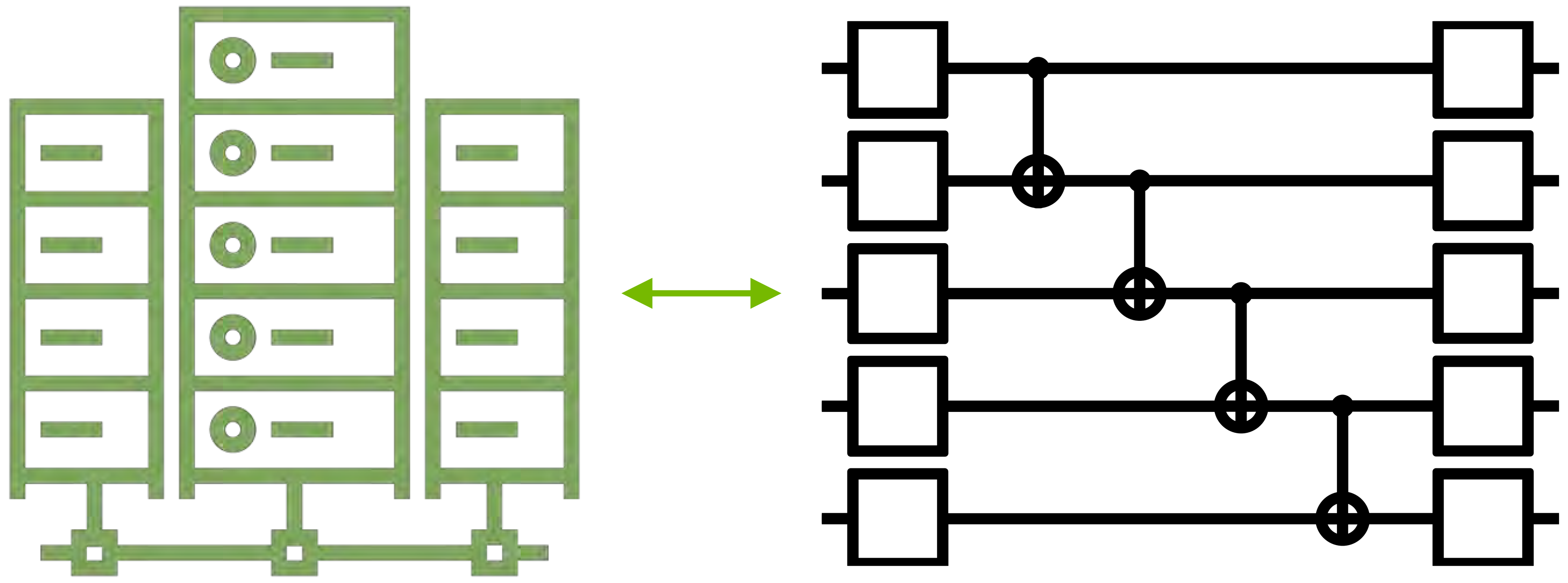
SYSTEM PROVIDERS: Atos, DELL Technologies, FUJITSU, GIGABYTE, H3C, Hewlett Packard Enterprise, inspur, Lenovo, Nettrix 宁畅, SUPERMICRO

ANNOUNCING NVIDIA QODA

THE PLATFORM FOR HYBRID QUANTUM-CLASSICAL COMPUTING

Classical Supercomputer

Quantum Computer



SUMMARY

- Simulating hybrid QC-classical systems requires large scale infrastructure with fast interconnect

- All Valuable Quantum Applications Will Be Hybrid
- A Platform For Real, Valuable Quantum Computing
- For Entire Quantum Computing Ecosystem

To go further:

- Qoda Blog
- Quoda-early-access
- Download CuQuantum


NVIDIA

Towards efficient hybrid algorithms using neutral atoms

Hybridation Quantum Processor Unit QPU / CPU – GPU
January 11 2023

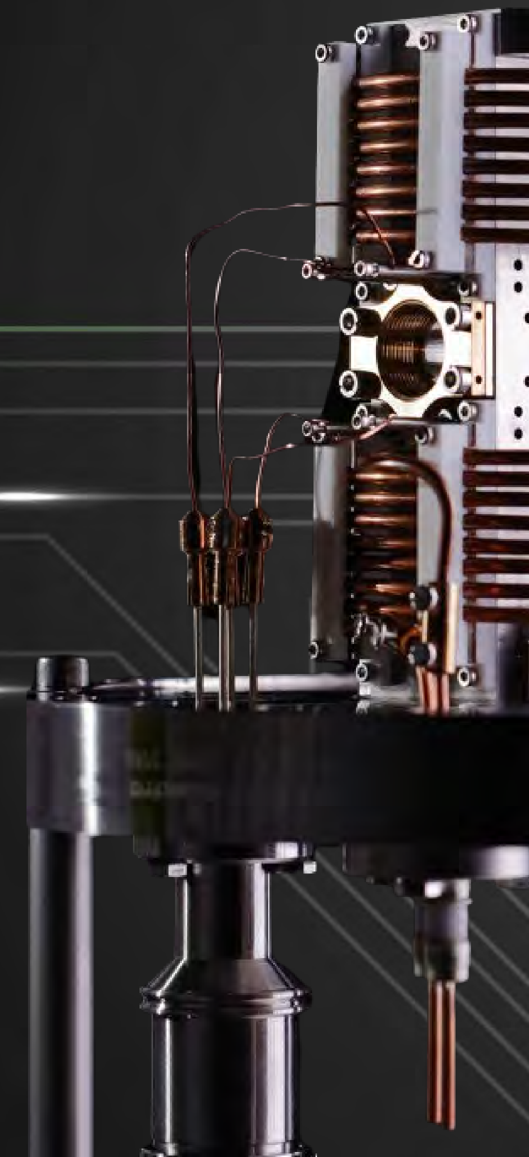
PASQAL
www.pasqal.com
office@pasqal.com
2, av. Augustin Fresnel
91120 Palaiseau
France

Agenda

1	Why do we need hybrid architectures
2	An example of hybrid algorithm
3	How would it work in practice

PASQAL

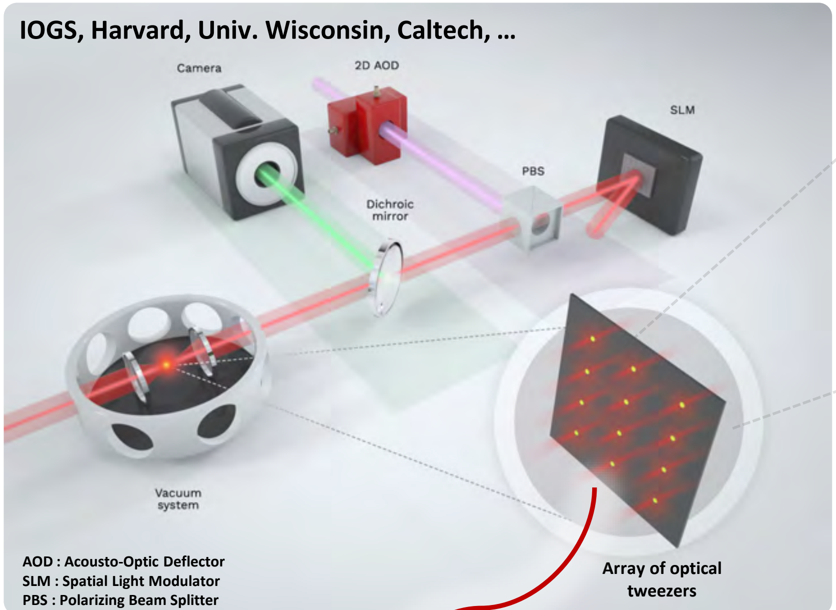
About PASQAL



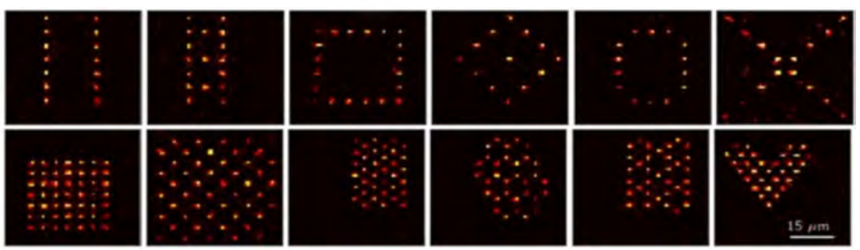
PASQAL in a few words

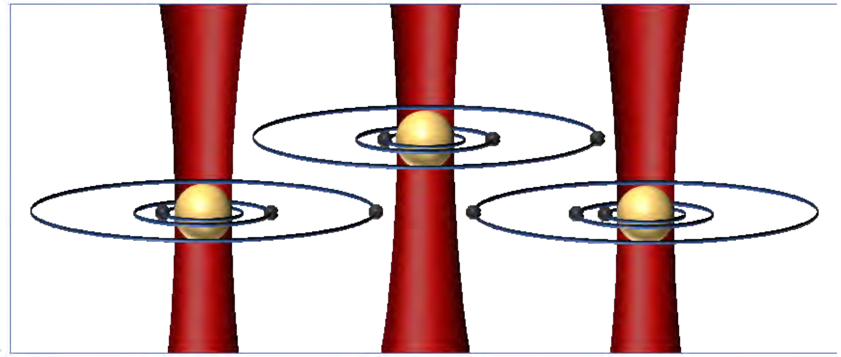
- **Our Legacy**
  - A word leading hub of Quantum science & Technology in Paris-Saclay.
  - Spin-out from Antoine BROWAEYS & Thierry LAHAYE's Lab
  - Cofounded by Alain Aspect
- **Our Solutions**
  - Neutral atoms (^{87}Rb) trapped with optical tweezers, driven by lasers and interacting using Rydberg physics.
  - Multi-purpose, flexible, 100 – 1000 qubit Quantum Processing Units (QPUs) for analog and digital QC.
- **Our Strengths**
  - Neutral atoms QPUs with many qubits of tunable geometries with unrivaled performance.
  - A full surrounding software stack for QPUs ready to be exploited on-premise or from the Cloud.
- **Our Team**
  - 120+ people; more than 80 Ph.D. Researchers and Engineers
  - Based on offices worldwide: France (Headquarter), Netherlands, United Kingdom, Canada, USA, …

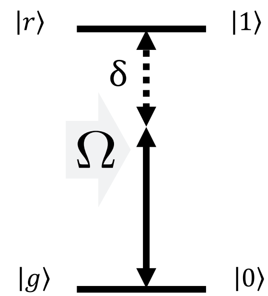
Rydberg atoms

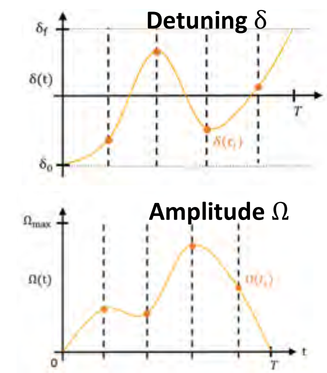


Schlosser et al., Nature (2001)







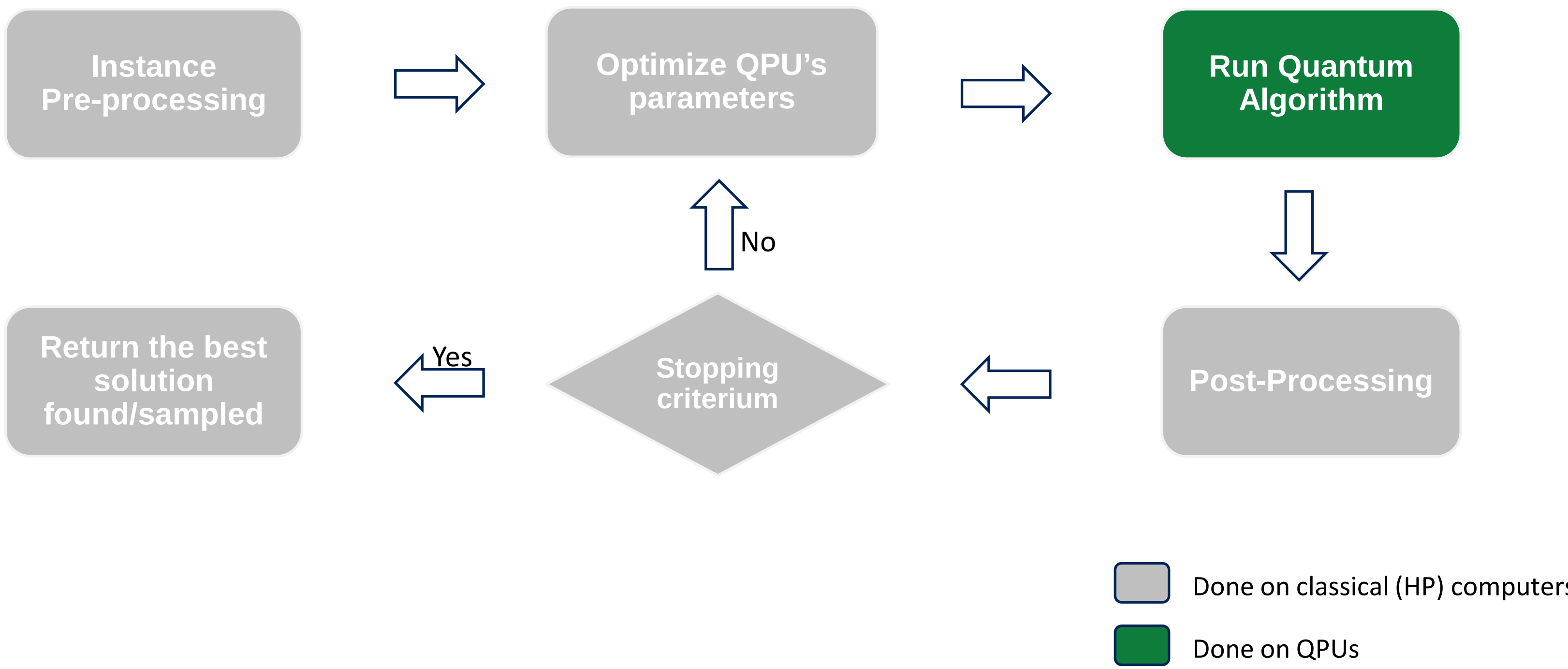


- **Strengths of neutral atoms platform :**
  - *High scalability, tunable connectivity, versatility (different settings)*
- **What can we do today?**
  - *Analog Quantum Computing*
  - *Quantum Simulation*

PASQAL

How CPUs and QPUs are normally coupled to solve optimization problems

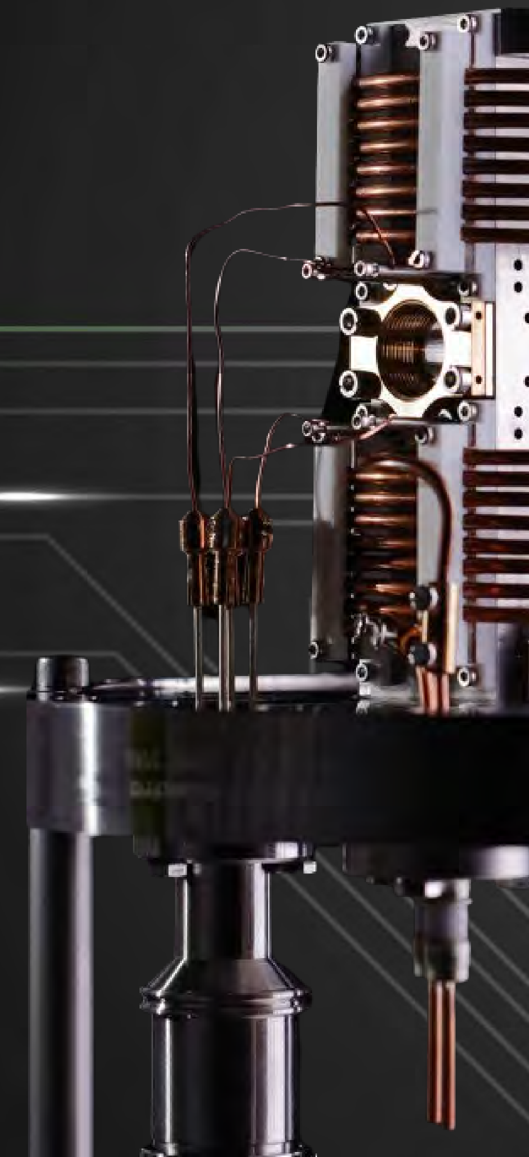
- Many QC algorithms involve sighnificant pre- or post-processing
- Many also include costly close-loop optimization



- New hybrid, with more intricate combination of hardware are being developped
- We present here an example of such an approach

Hybrid Computing

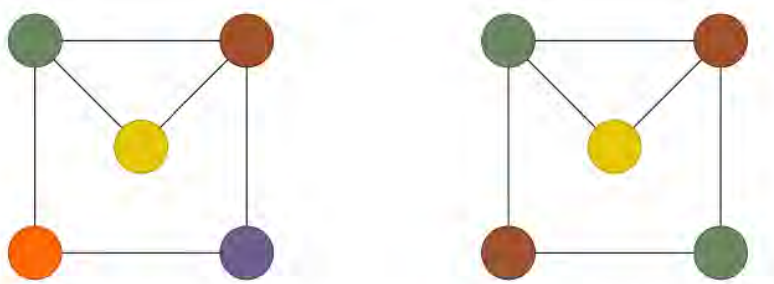
A quantum sampler to speed up classical solvers



The Minimum Vertex Coloring problem

Let G = (V,E) be a graph with a set V of nodes and a E of edges.
Also, let C be a set of available colors.

The **Minimum Vertex Coloring Problem** (MVCP) consists in **coloring the vertices of G with exactly one color from C** in a such way that **the number of used colors is minimized** while ensuring that **no two adjacent vertices have the same color**.



(a) Trivial coloring. (b) Optimal coloring.

Graph coloring enjoys many practical applications such as

- Scheduling problems
- Portfolio optimization
- Resources assignment
- Sudoku puzzles

Any sub-set of nodes colored with the same color is an Independent Set* !!

* Independent Set : a sub-set of vertices that are not connected to each other.

An extended formulation for the Minimum Vertex Coloring Problem

Let **G** be an input **graph** with a **set V of nodes** and **a set E of edges**

Let **S** be the set of **all independent sets** in the graph

Let y_s **be a binary variable that takes 1 if the independent set s ∈ S is selected**; 0 otherwise.

Let b_{us} **be a parameter that holds 1 if the vertex u ∈ V** is present in the independent set s ∈ S; 0 otherwise.

$$min \sum_{s \in S} y_s$$

$$\sum_{s \in S} b_{us} y_s = 1, \qquad \forall u \in \mathcal{V}$$

$$y_s \in \{0, 1\}, \qquad \forall s \in S$$

Finding all independent sets in a graph is a very hard task and can be very time and resource-consuming, even for small graphs.

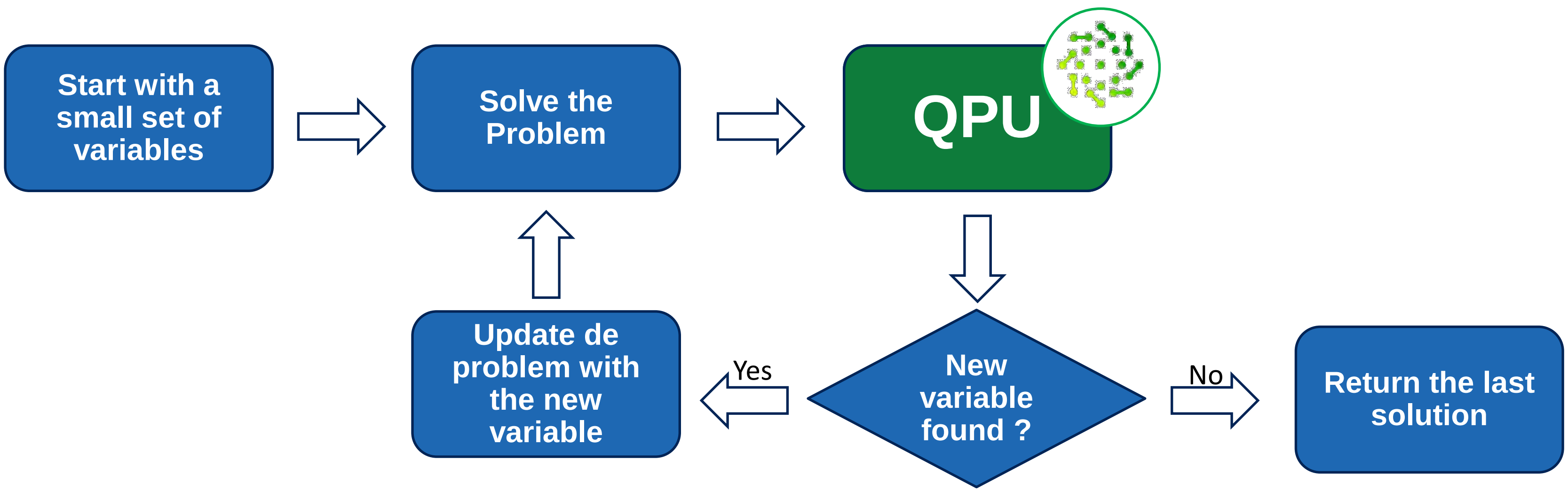
- The number of such sets, and therefore the number of **y** variables, can be extremely large.
- Most of them are not selected in any feasible solution, especially in the optimal one.

PASQAL

* Independent Set : a sub-set of vertices that are not connected to each other.

A column generation for the Minimum Vertex Coloring Problem

We don't need to know all elements (independent sets) in advance

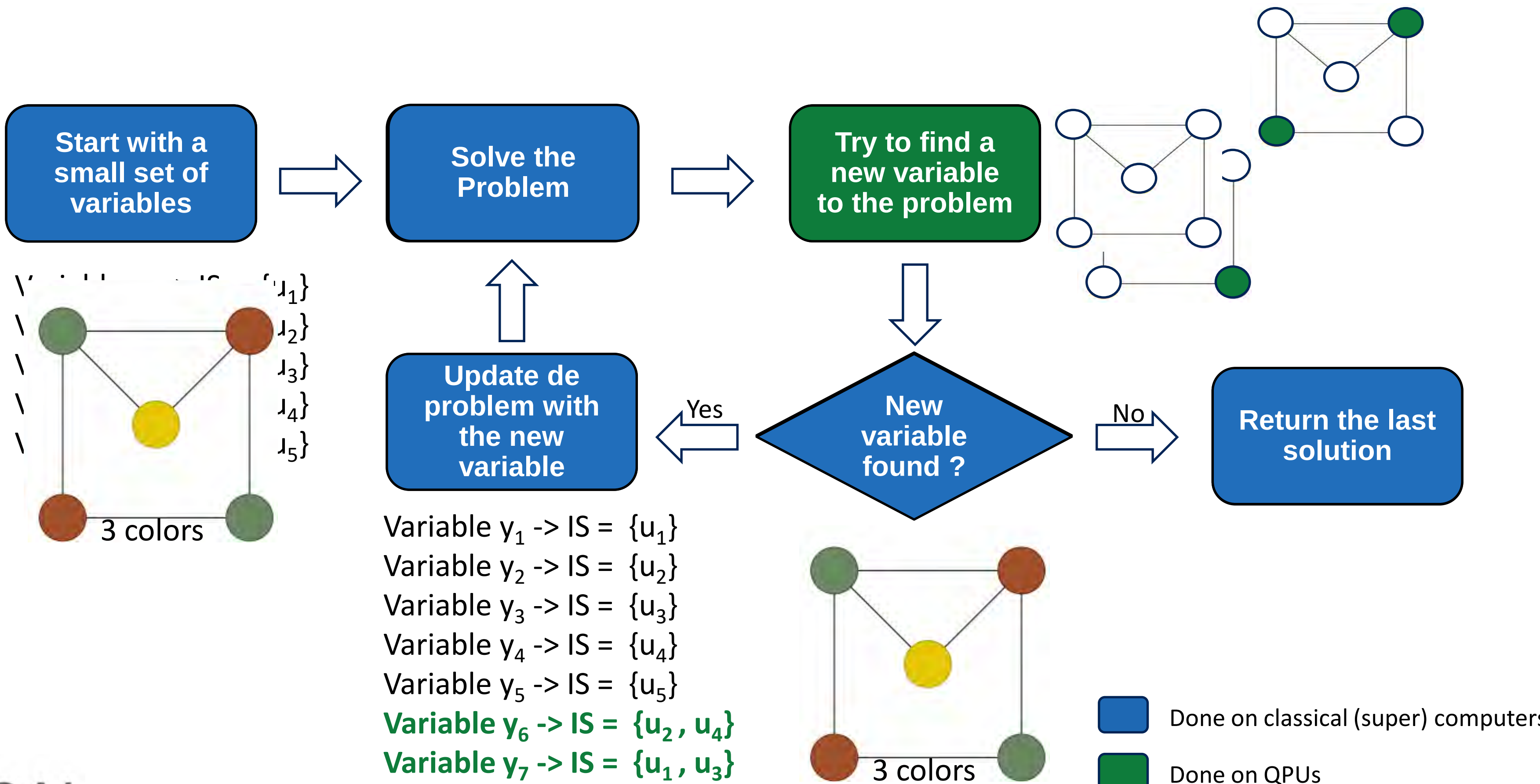


But, how to **efficiently generate new elements** to be converted into variables within the mathematical formulation?

* Independent Set : a sub-set of vertices that are not connected to each other.

A column generation for the Minimum Vertex Coloring Problem

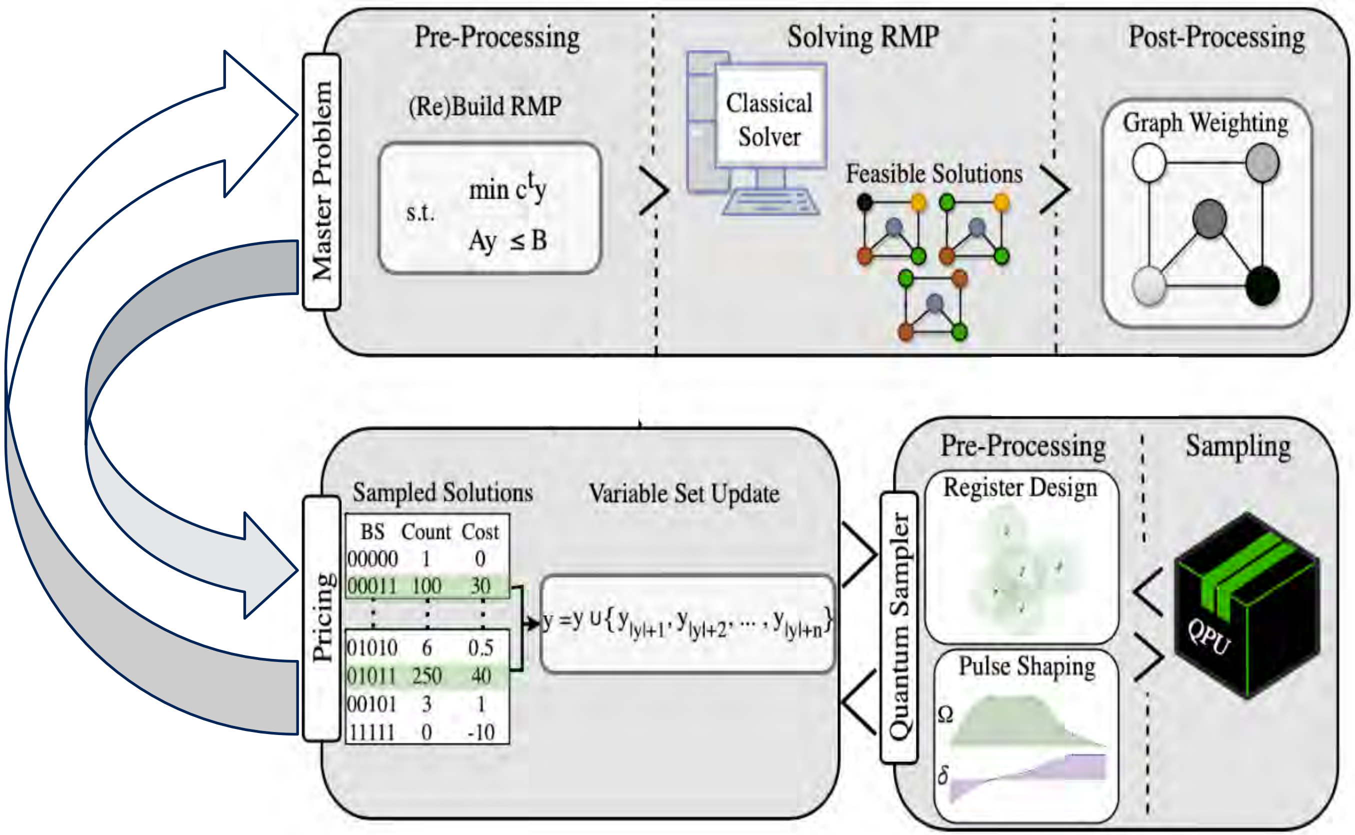
In the proposed hybrid framework, a quantum sampler is specifically tailored to provide new variables to the master problem



PASQAL

Independent Set (IS) : a sub-set of vertices that are not connected to each other.

An overview of the hybrid column generation framework for solving hard combinatorial problems



Classical Computer

Finds the best coloring with the current set of elements (variables)

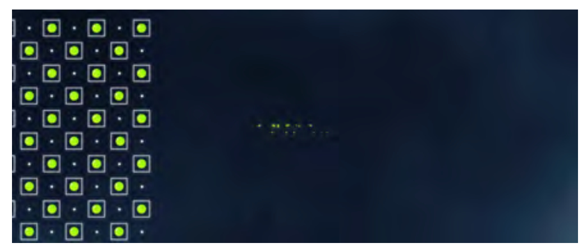
Provides instances to the pricing routine

Runs the Atom Register and Pulse Sequence design algorithms

Quantum Computer

Samples good solutions for each pricing sub-problem

Provides new elements to the RMP as new variables

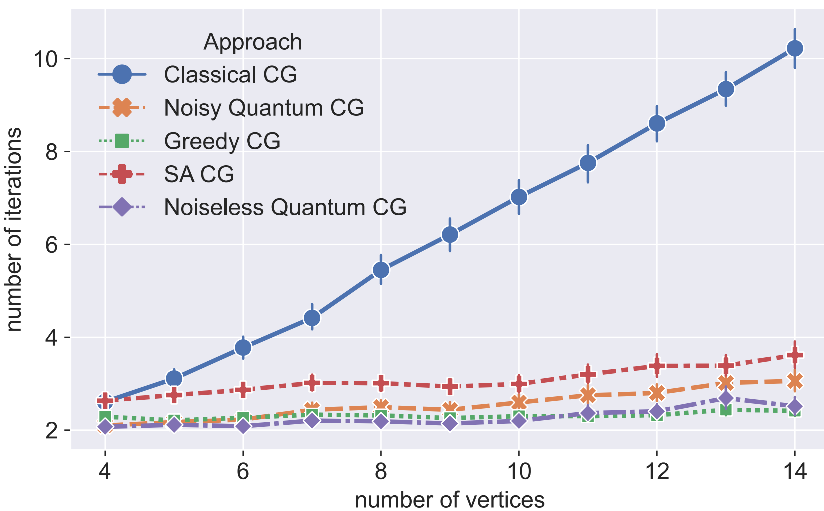


QPU : Quantum Processing Unit
RMP : Reduced Master Problem
BS: Bitstring

Results

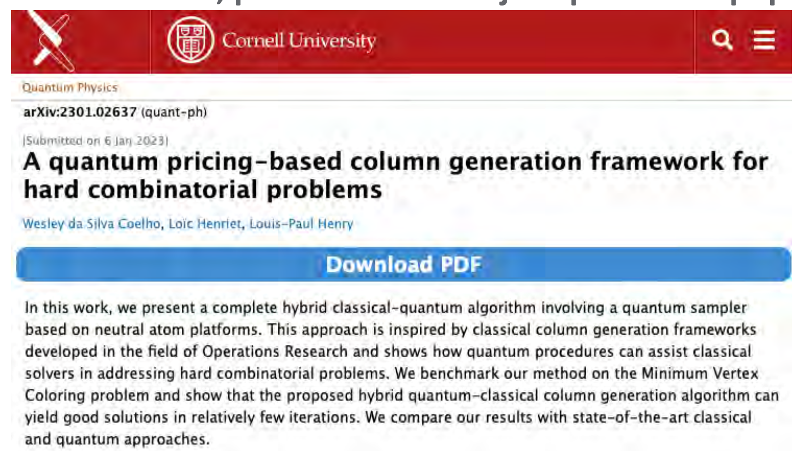
We compared our hybrid approach against different column generation frameworks where the pricing sub-problems were solved by:

- Classical Solver (exactly solving the associated MWIS problem on each pricing instance)
- Stochastic Greedy algorithm
- Simulated Annealing-based sampler



Number of iterations before finding the optimal solution by applying different approaches on different graph orders (from 4 up to 14 vertices), classes (UD and non-UD) and densities (20%, 50% 80%).

For more details, please check our just-published paper



Quantum Physics

arXiv:2301.02637 (quant-ph)

[Submitted on 6 Jan 2023]

A quantum pricing-based column generation framework for hard combinatorial problems

Wesley da Silva Coelho, Loic Henriet, Louis-Paul Henry

Download PDF

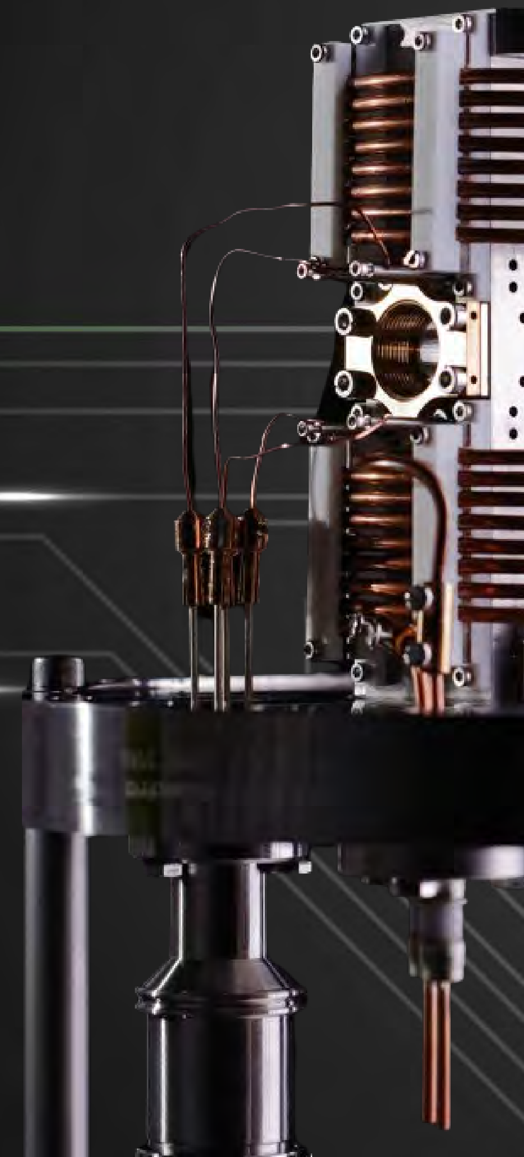
In this work, we present a complete hybrid classical-quantum algorithm involving a quantum sampler based on neutral atom platforms. This approach is inspired by classical column generation frameworks developed in the field of Operations Research and shows how quantum procedures can assist classical solvers in addressing hard combinatorial problems. We benchmark our method on the Minimum Vertex Coloring problem and show that the proposed hybrid quantum-classical column generation algorithm can yield good solutions in relatively few iterations. We compare our results with state-of-the-art classical and quantum approaches.

MWIS : Maximum Weighted Independent Set
SA : Simulated Annealing

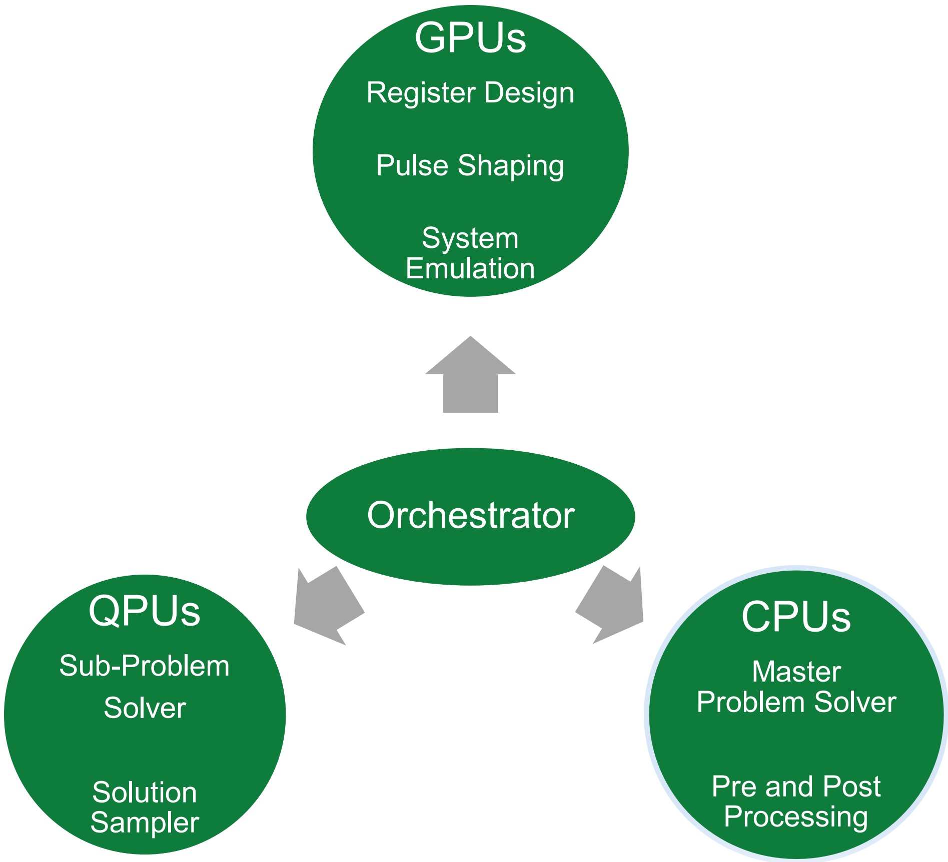
CG : Column Generation
UD : Unit-Disk

HPC environment

How to integrate QPUs into HPC frameworks



HPC Environment



- Classical computers (CPUs and GPUs) are still important in solving hard optimization problems
  - They can solve efficiently several of them in a reasonable runtime
  - Import helper in optimizing the QPU's parameters
  - They are useful in emulating quantum systems
  - Still needed to pre- and post-processing the data provided by QPUs

- Quantum computers can solve more efficiently some classes of problems
  - They can be "plugged" to classical computers to help them in solving complex problems
  - They can provide valuable/inaccessible information about the problem

PASQAL

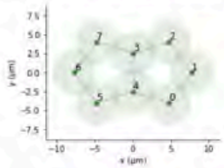
How to program the QPU?

Pulser

https://github.com/pasqal-io

Pulser is a framework for composing, simulating and executing pulse sequences for neutral-atom quantum devices.

```
pentagons_register.draw(draw_graph=True, blockade_radius=a*1.1, draw_half_radius=True)
```

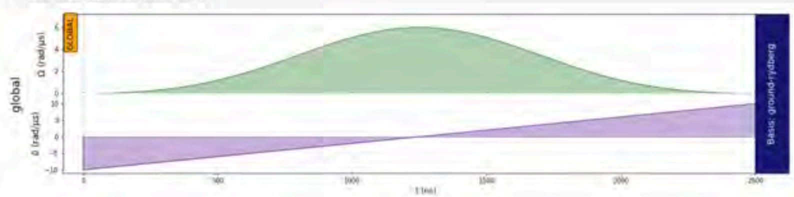



```
duration = 2500 # 2500 ns = 2.5 µs
Omega = 2*np.pi # rad/µs
delta = 10.
```

```
from pulser import Pulse
from pulser.waveforms import BlackmanWaveform, RampWaveform

amp_wf = BlackmanWaveform(duration, Omega)
det_wf = RampWaveform(duration, -delta, delta)

pulse = Pulse(amp_wf, det_wf, phase=0)
```

```
from pulser.devices import Chadoq2

device = Chadoq2
```

```
from pulser import Sequence

pentagons_sequence = Sequence(pentagons_register, device)
pentagons_sequence.available_channels

pentagons_sequence.declare_channel('global','rydberg_global')

pentagons_sequence.add(pulse, 'global')

pentagons_sequence.measure(basis="ground-rydberg")


pentagons_sequence.draw()
```



Pulser studio

https://pulserstudio.pasqal.cloud/

- Enables users to graphically build quantum registers and design pulse sequences without coding knowledge.
- The platform allows the creation of novel insights into quantum computing and neutral atoms using an original user experience.
- Pulser Studio includes a built-in emulator that will simulate sequences for small systems directly in the browser.
- These registers and pulse sequences can be executed on quantum processors.
- Pulser Studio is already open and free to corporate and academic users
- A powerful tool to quickly try your ideas

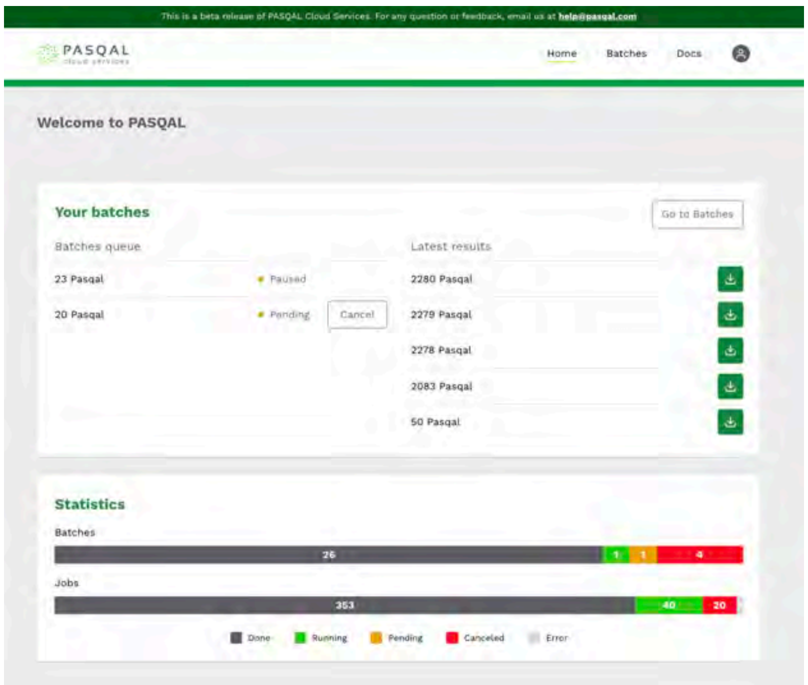
PASQAL

How to use PASQAL Cloud Services to send Pulser jobs on the QPU?

User Portal

https://portal.pasqal.cloud

- Account & Project administration
- Get API keys
- Monitor jobs
- Download results



Python SDK

https://github.com/pasqal-io/cloud-sdk

- Pulser sequences (jobs) that share common parameters or register layouts can be grouped into a single batch.
- SDK is used to send a batch for execution on the QPU via the cloud platform
- SDK is used to retrieve batch & jobs results

```
# Define Pulser sequence
reg = Register.square(2, prefix="q")
seq = Sequence(reg, devices.Chadoq2)
seq.declare_channel("raman", "raman_local")
seq.target("q0", "raman")
simple_pulse = Pulse.ConstantPulse(200, 2, -10, 0)
seq.add(simple_pulse, "raman")
sequence_builder = seq.serialize()

# Init connection to PASQAL cloud
endpoints: Endpoints = Endpoints(
    core="https://apis.dev.pasqal.cloud/core",
    account="https://apis.dev.pasqal.cloud/account",
)
client_id = ""
client_secret = ""
sdk = SDK(
    client_id=client_id,
    client_secret=client_secret,
    endpoints=endpoints,
)

jobs = []
for _ in range(10):
    jobs.append({"runs": 10})

# Launch batch and wait for results
batch = sdk.create_batch(
    sequence_builder, jobs, device_type=DeviceType.EMU_FREE, wait=True
)
```

Hybrid in HPC centers

PASQAL works in close collaboration with HPC centers, in order to develop hybrid platforms

Sold 2 QPUs to be installed directly
into HPC centers
in France and Germany

GENCI
Le calcul intensif au service de la connaissance
TGCC

JÜLICH
Forschungszentrum

Created **QuaTERA** (Quantum Technologies Energy Result Accelerator)
alongside EDF, Exaion Inc. and the Quantum Innovation Zone

“The first open center of excellence to develop sustainable energy solutions using the combined capabilities of HPC and quantum computing”

exaion
GROUPE EDF

ZONES D'INNOVATION QUÉBEC
SHERBROOKE QUANTIQUE

PASQAL

Thank you

PASQAL
www.pasqal.com
office@pasqal.com
2, av. Augustin Fresnel
91120 Palaiseau
France

QUANDELA

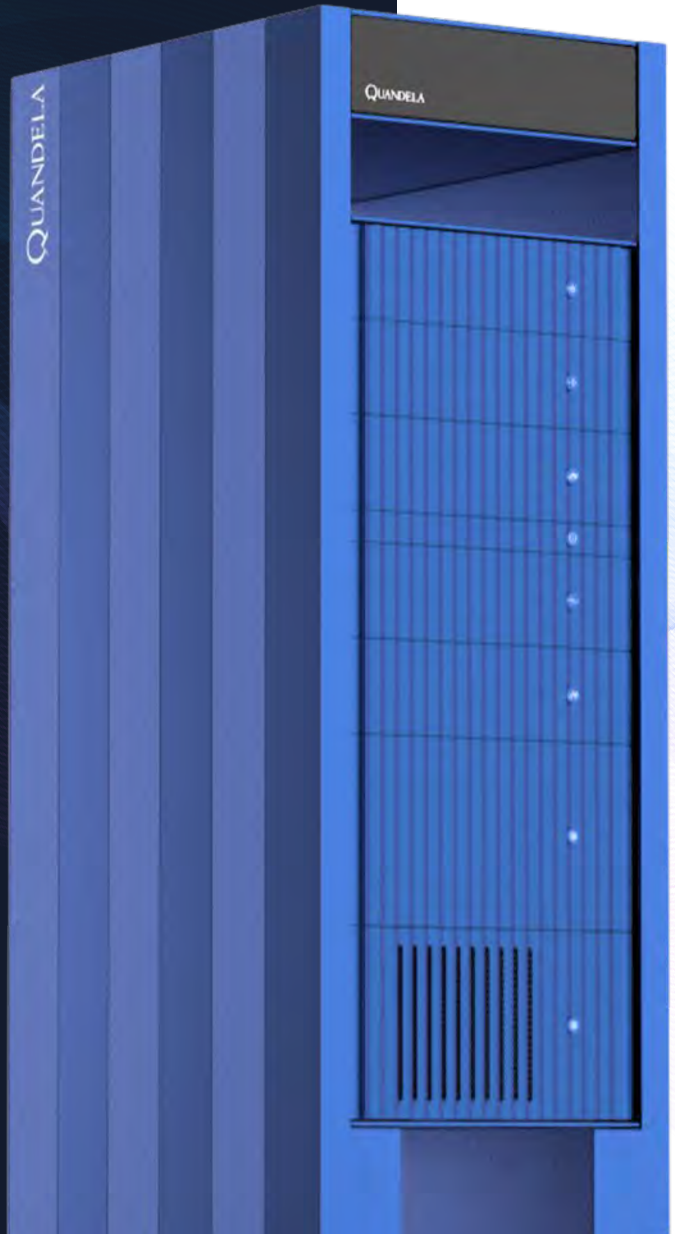
Single-Photon based Quantum Computers available in the cloud

Metrics and Benchmarks

Shane Mansfield

Jean Senellart

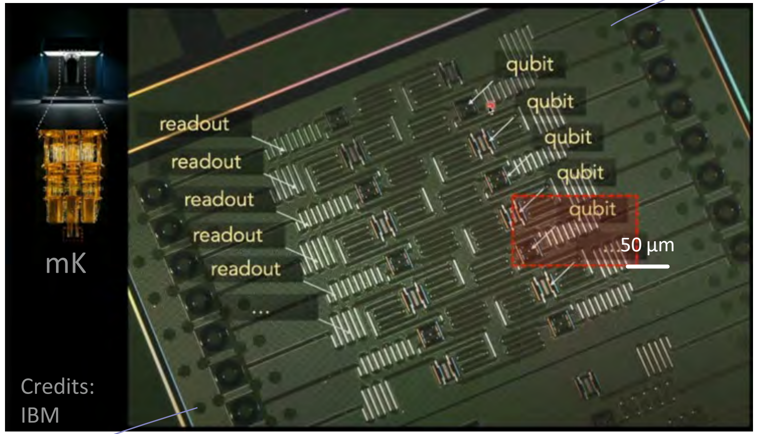
2023-01-11



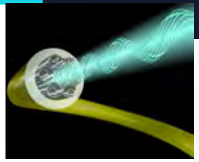
Digital Approaches to Quantum Computing

1 Matter Qubits: Ions, Superconductors, Cold Atoms, etc

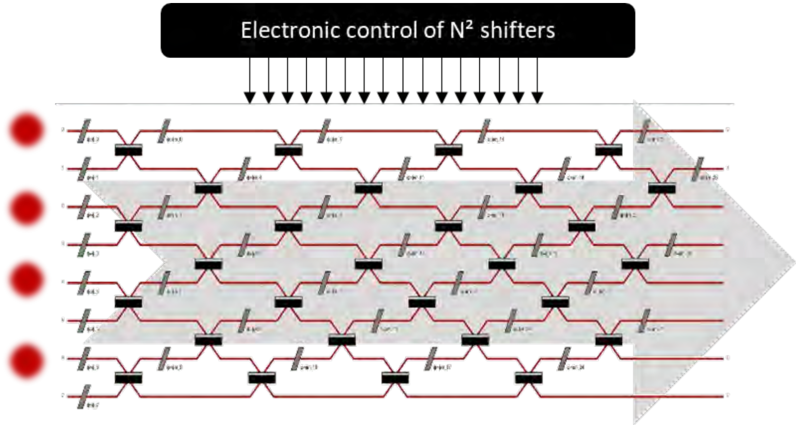
- Static qubits
- Objects located on a QPU



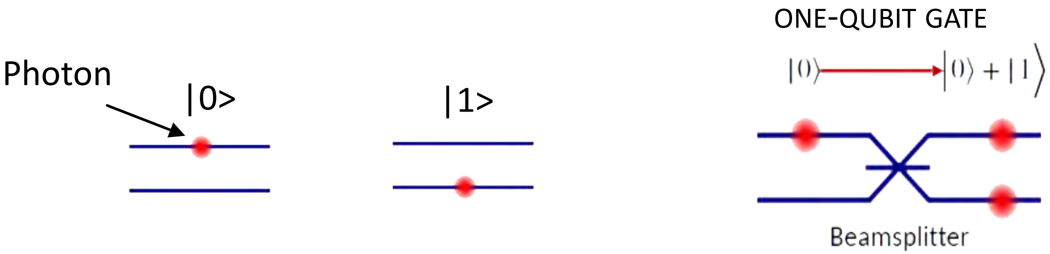
2 Photonic Qubits



- 'Flying' qubits
- Passing through a QPU
- Optical fibres, integrated linear optical circuits



1 qubit (dual-rail encoding):



Photons and Qubits

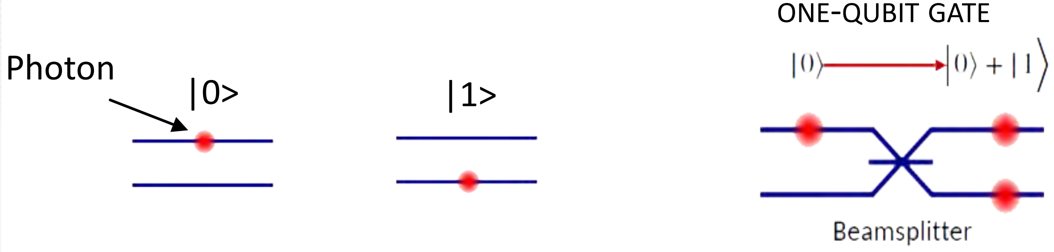
Encoded qubits

Photons

- More degrees of freedom than a qubit
- More computational states:
  - $\binom{n-1}{m-1}$ states of n photons in m modes
  - 2^n sates of n qubits
- Photonic quantum algorithms can exploit this
- Why demonstrations of quantum advantage to date are either photonic or have a strong photonic flavour
- This also means there are many ways to **encode qubits into photons**!

Photonic Qubits

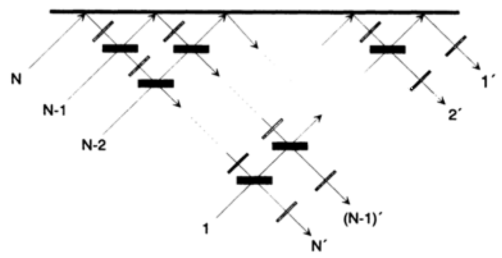
Ex 1: Dual-rail Encoding



Ex 2:

Reck–Zeilinger–Bernstein–Bertani (RZBB) Decomposition

- Any *qubit circuit* can be described by a *unitary matrix*
- Any *unitary matrix* can be constructed as a *linear optical circuit*



Universal Chip Design
Ascella – A General Purpose QPU

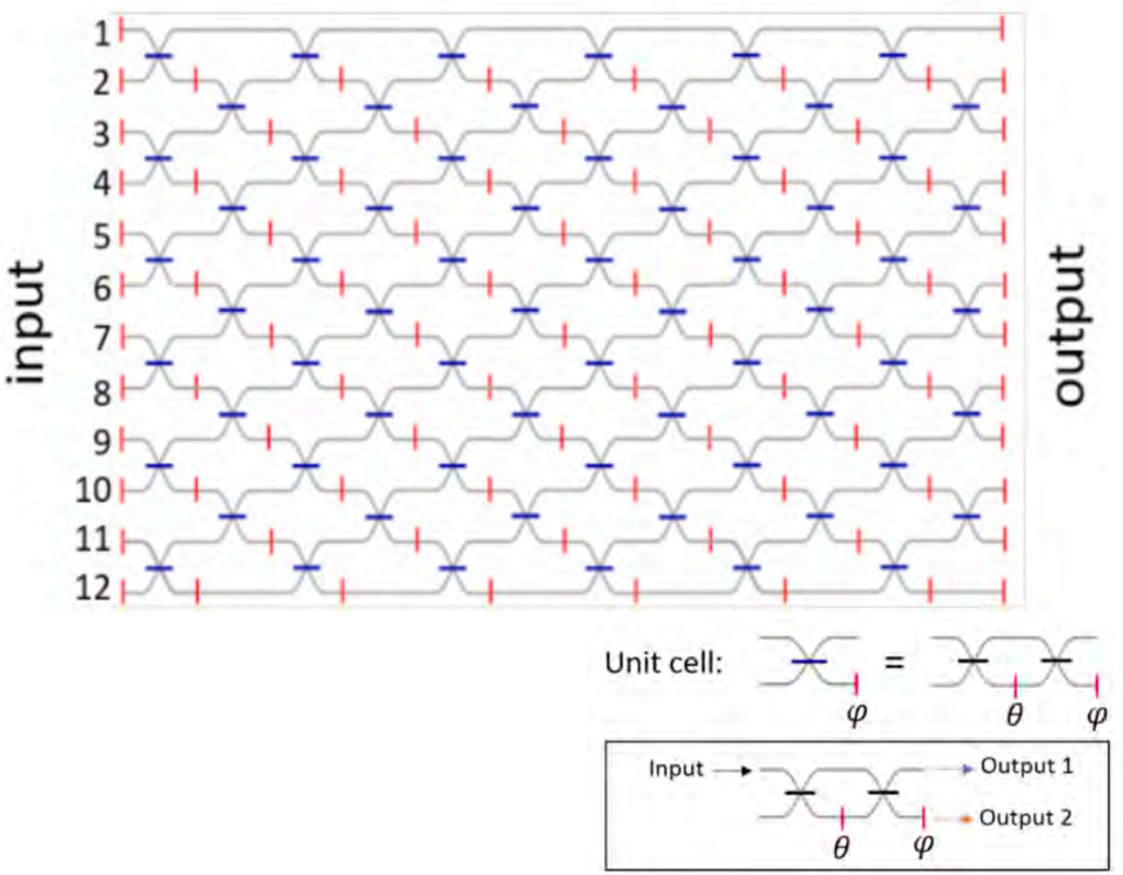
Running Computations

- Chip is designed to have a “universal” architecture
- Similar to the RZBB decomposition

- Can run **photon-native algorithms**
- Quandela’s Differential Equation Solver, or Boson Sampling

- Can also run general **qubit circuits**
- *Perceval* software will automatically translate qubit circuits or *Qiskit* code to run on our chip

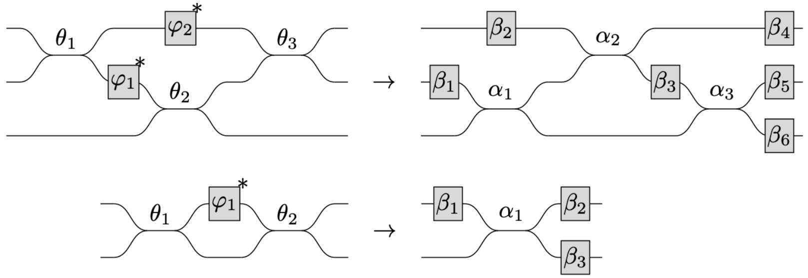
12 modes – up to 6 photons – 126 parameters



Specialized Chip Design
Altair - A Machine Learning QPU

Application Specific Integrated Circuits

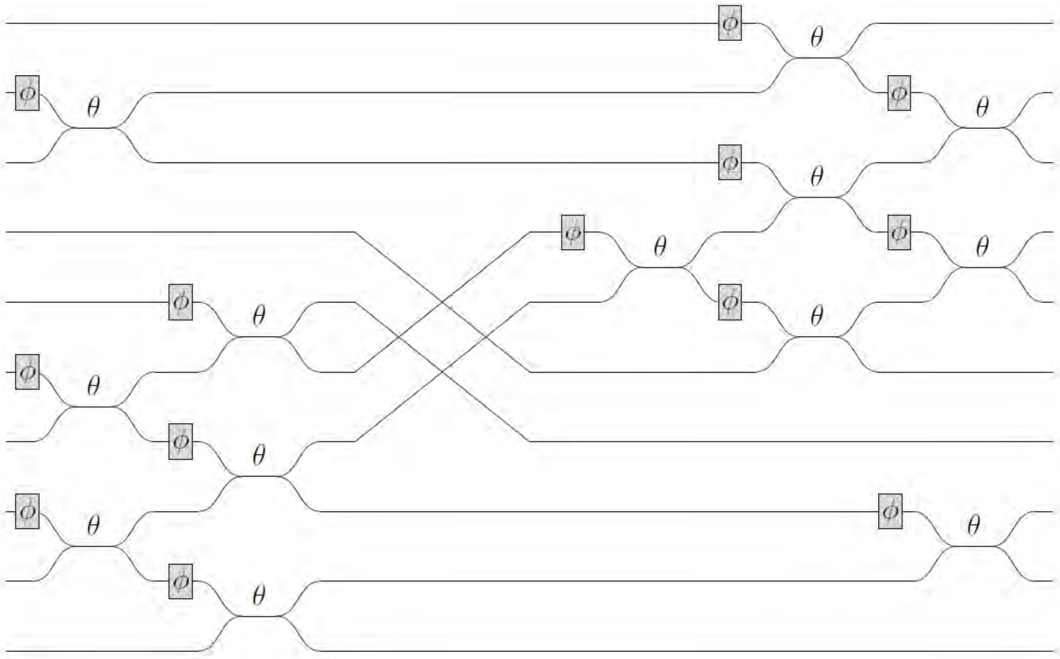
- Accelerated performance with QPUs that are **optimized to specific tasks**
- **Reduced parameters** compared to "universal" architecture
- Powerful design tools such as the **LOv-calculus**



Examples:

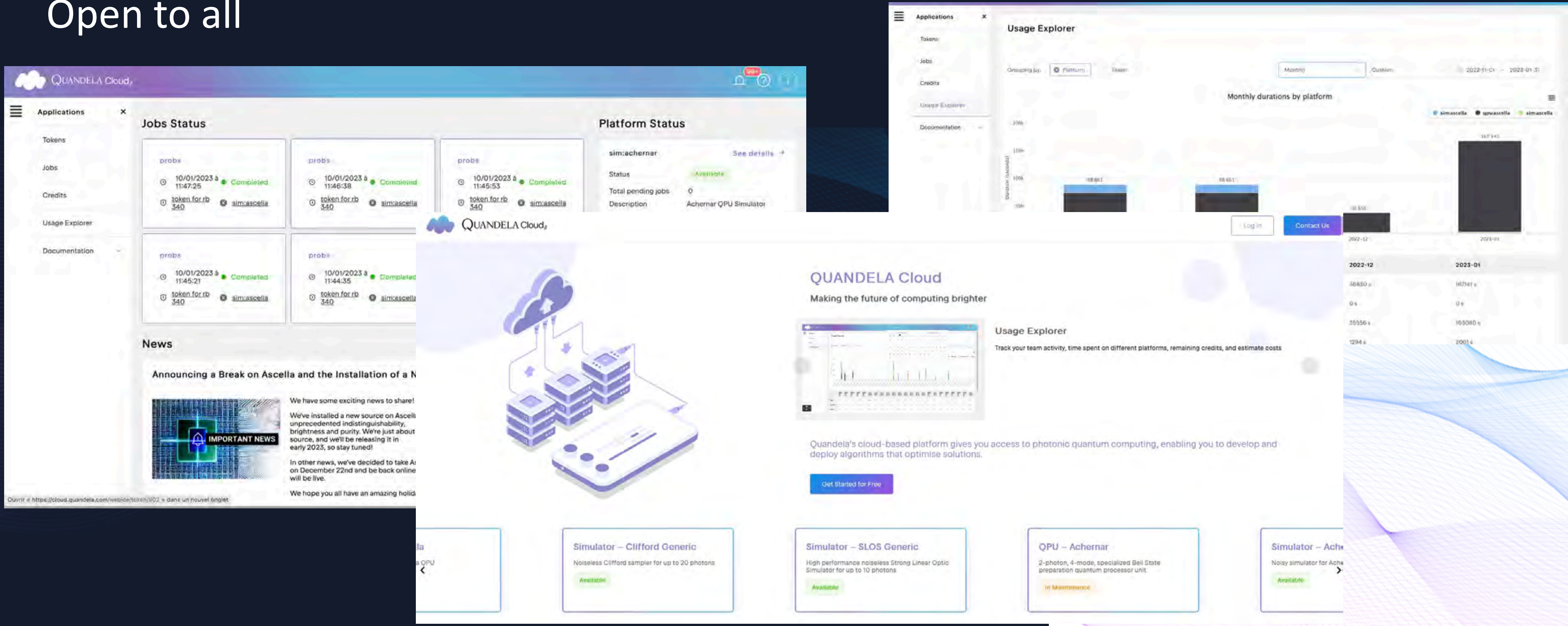
- *Altair* is optimized for ansatz circuit generation
- Quandela's *Entropy* device contains circuits optimized for certified random number generation

10 modes – 3 photons – 26 parameters



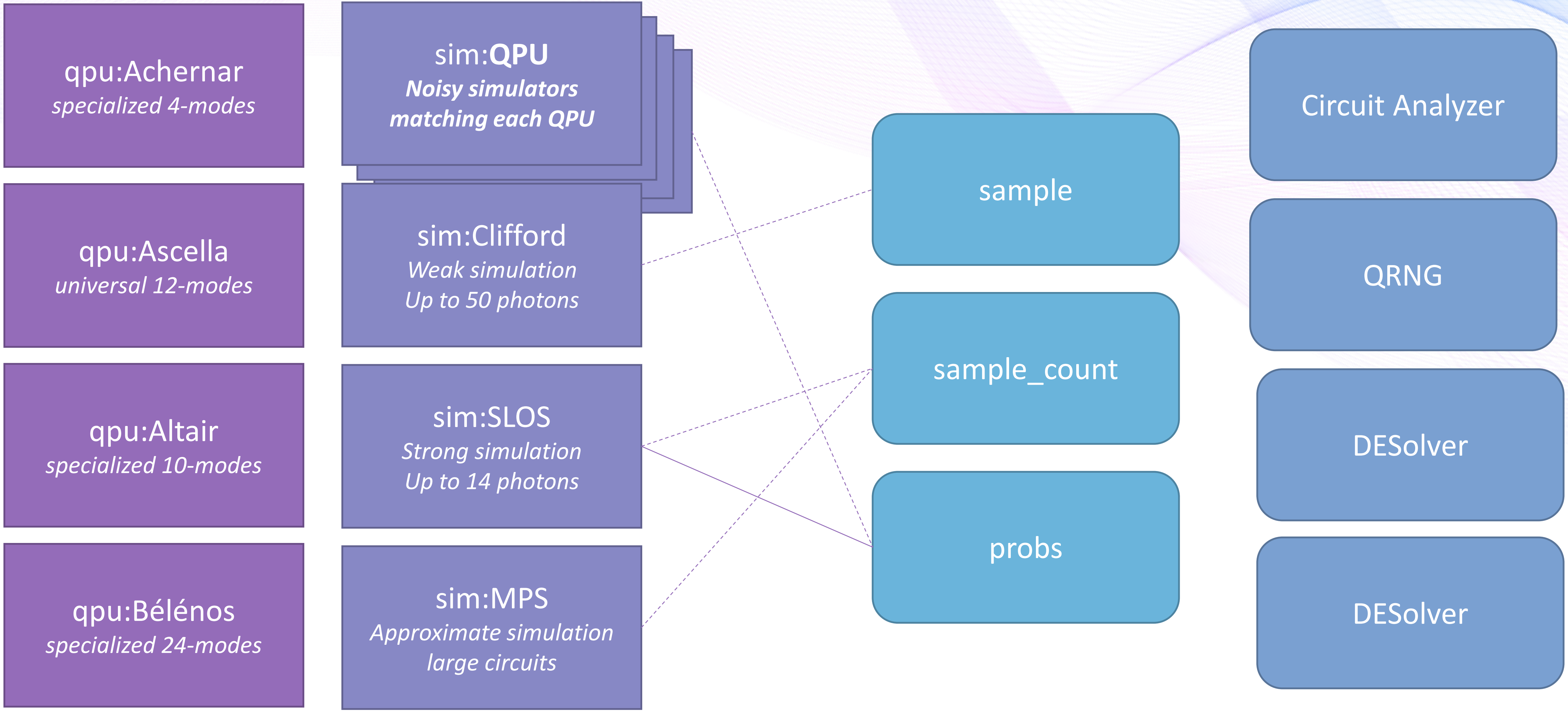
Quandela: the first European QPU provider in the cloud

Open to all



Test our QPUs available in the cloud
https://cloud.quandela.com

Available Processors and Primitives

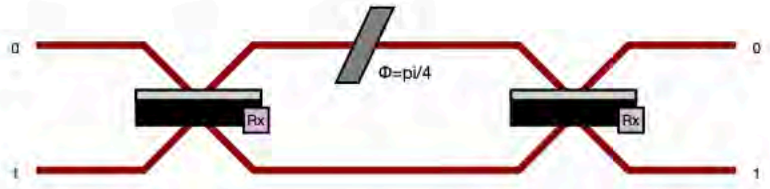


Running a quantum program on the cloud

A sample program on a photonic processor !

```python
import perceval as pcvl
from perceval.algorithm import Sampler
```

```python
input_state = pcvl.BasicState([1, 1])

c = pcvl.Circuit(2)
c.add(0, pcvl.BS())
c.add(0, pcvl.PS(phi = np.pi/4))
c.add(0, pcvl.BS())

pcvl.pdisplay(c)
```




```python
# Use your key here to let the system know who you are
token_qcloud = 'MY_SECRET_KEY'
remote_qpu = pcvl.RemoteProcessor("qpu:ascella", token_qcloud)
```

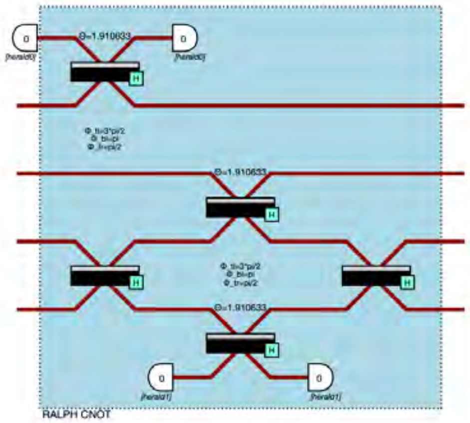
```python
remote_qpu = pcvl.RemoteProcessor("qpu:ascella", token_qcloud)
remote_qpu.set_circuit(c)
remote_qpu.with_input(input_state)
remote_qpu.mode_post_selection(1)

sampler_on_qpu = Sampler(remote_qpu)

nsample = 10000
job = sampler_on_qpu.sample_count(nsample)
```

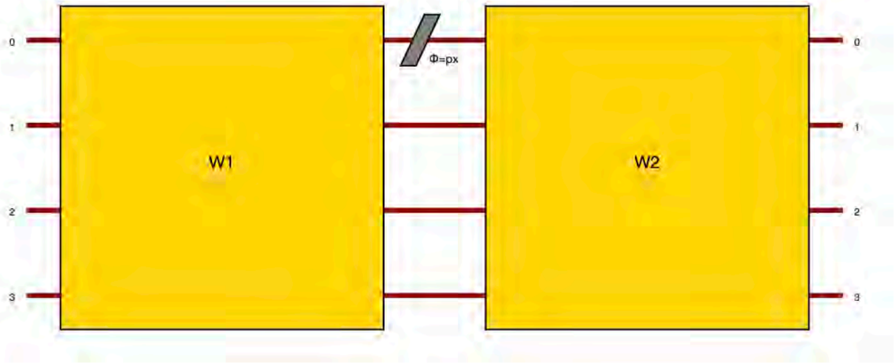
```python
results = job.get_results()
print(results['results'])
```

```
{
  |0,1>: 13979
  |1,0>: 9627
  |1,1>: 51
}
```

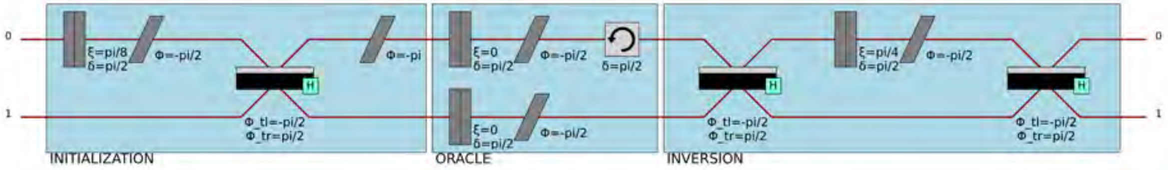
Example Circuits

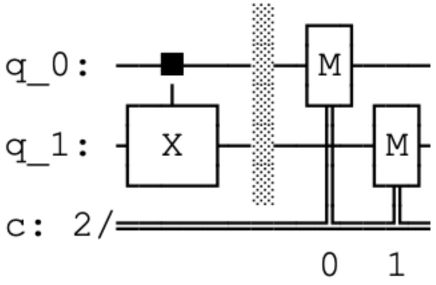
Native Photonic Circuit

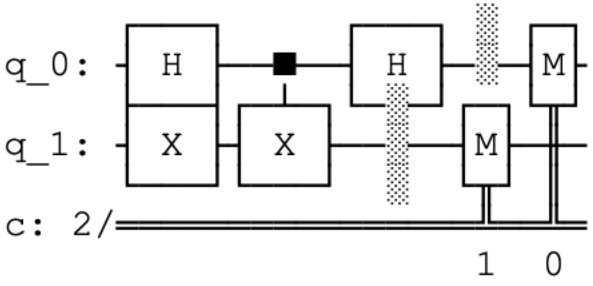




Other Supported circuit

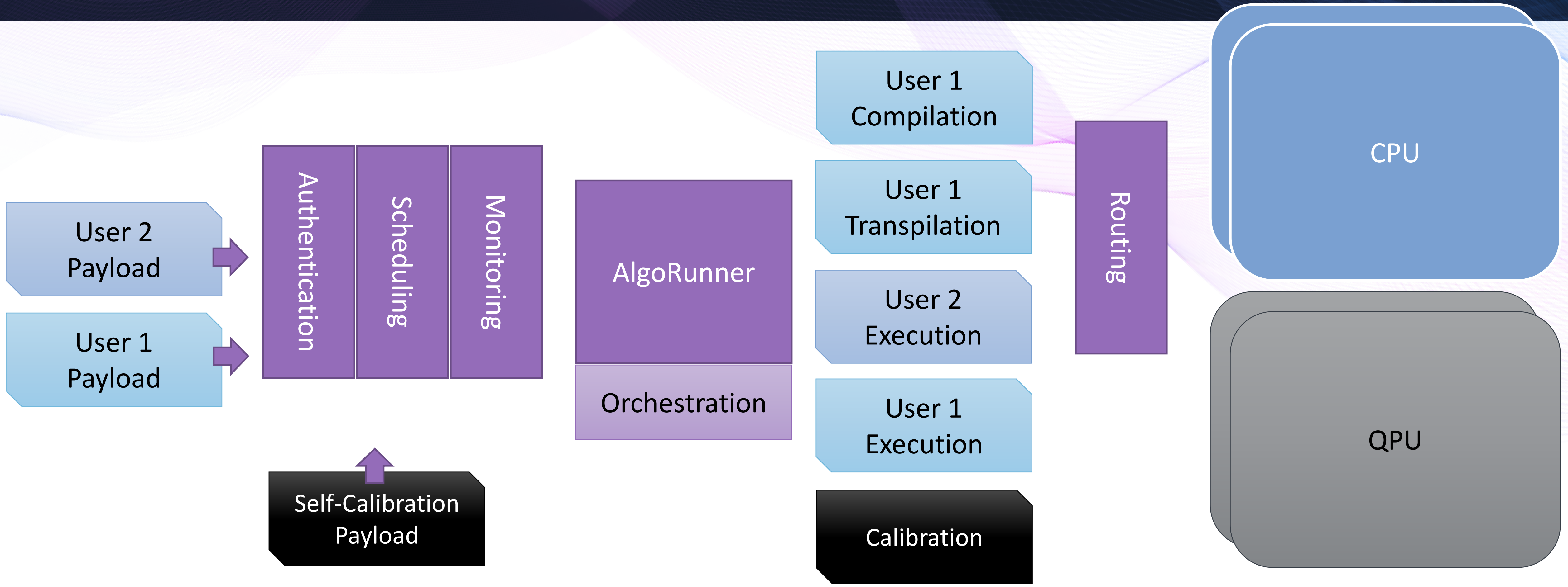






Behind the scene

Step-by-Step request processing



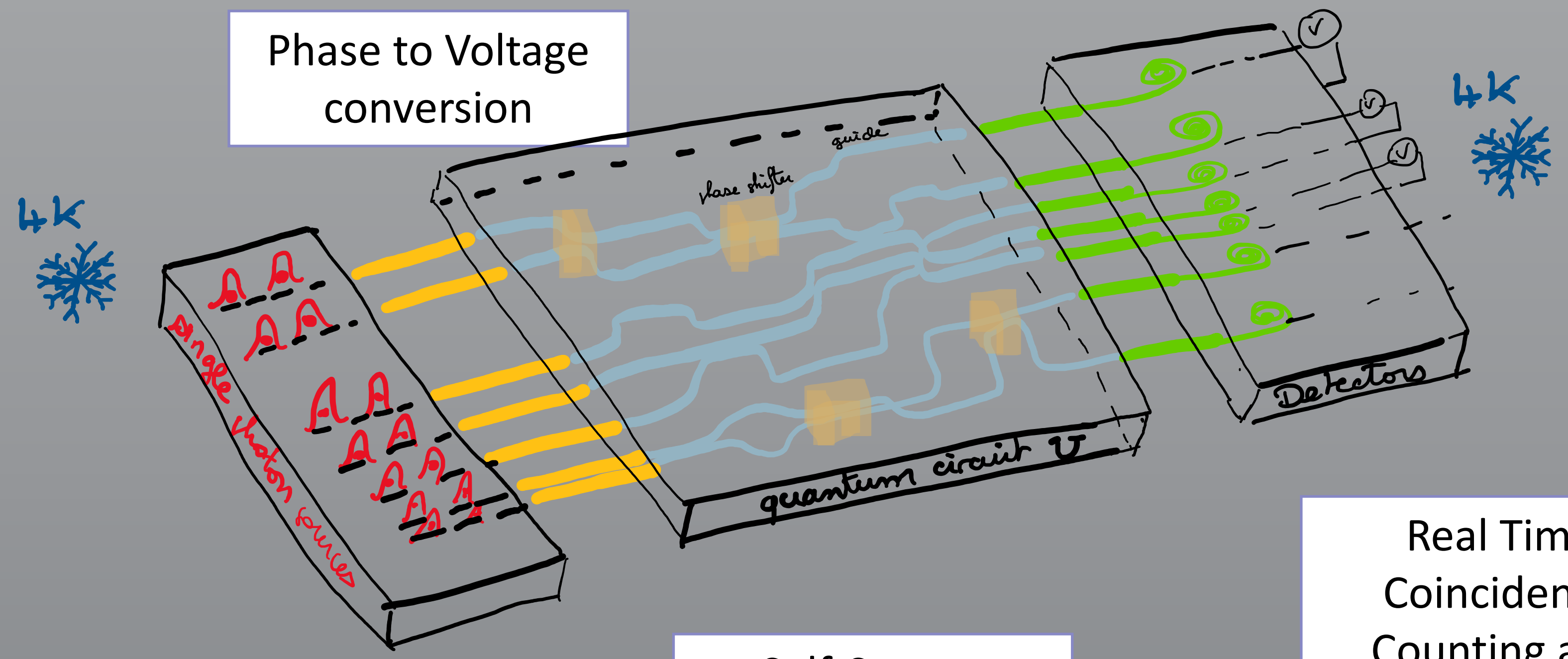
Full complementarity between classical and quantum algorithms

Behind the scene

Step-by-Step request processing

QPU

Phase to Voltage conversion



User 2 Execution

Circuit
Input State

Photon Routing Control

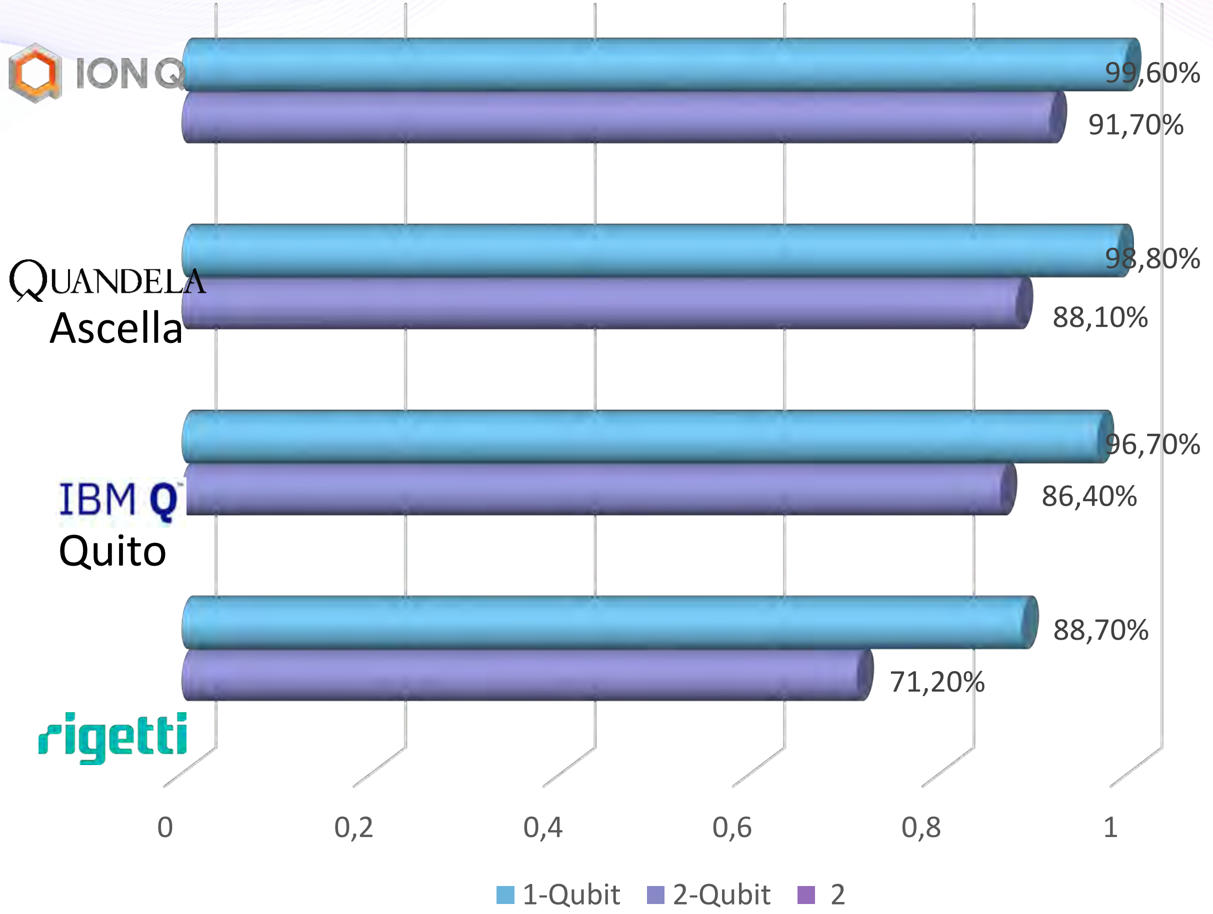
Self-System Monitoring

Real Time Coincidence Counting and Aggregation

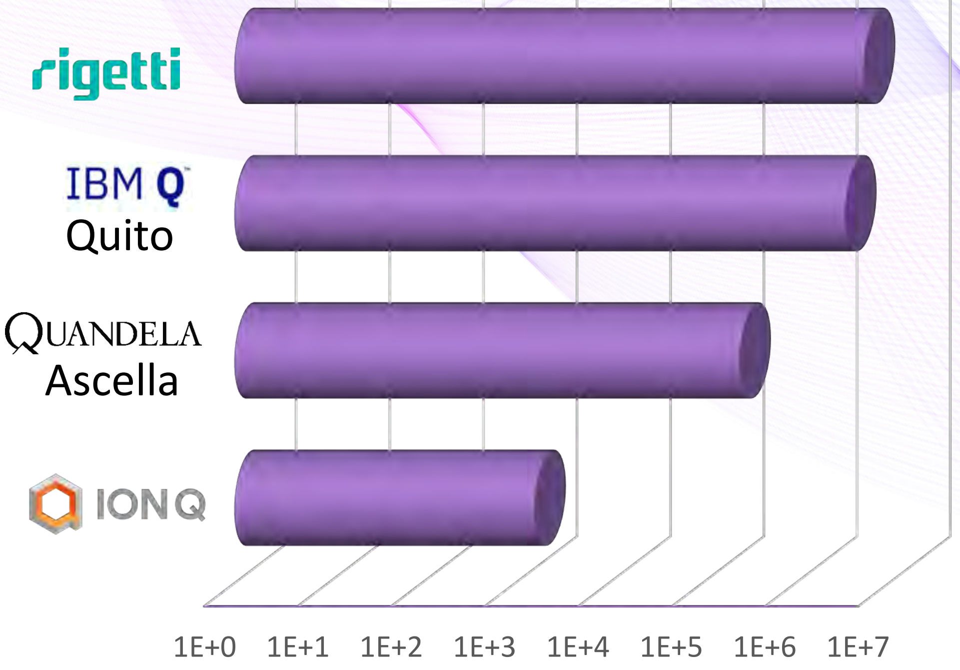
Full complementarity between classical and quantum algorithms

Benchmarking with other online platforms

N-Qubit Gate Fidelity



Number of 2-Qubit gates per second



500+ hours online

Availab lity 88%

17000+ jobs

QUANDELA

Merci !

shane.mansfield@quandela.com

jean.senellart@quandela.com

Quantum-classical computing at multiple time scales

Blake R Johnson, PhD
Distinguished Research Scientist, Quantum Engine Lead

IBM **Quantum**

May 2016:

IBM **Quantum**

First programmable quantum computer in the cloud with a simple composer for developing quantum circuits

May 3, 2016
Posted in: Quantum Computing

Quantum computing: It's time to build a quantum community

The New York Times

IBM Wants Everyone to Try a Quantum Computer

By John Markoff

WIRED

CADE METZ BUSINESS 05.04.16 12:01 AM

IBM IS NOW LETTING ANYONE PLAY WITH ITS QUANTUM COMPUTER

TechCrunch

IBM launches quantum computing as a cloud service

Ron Miller

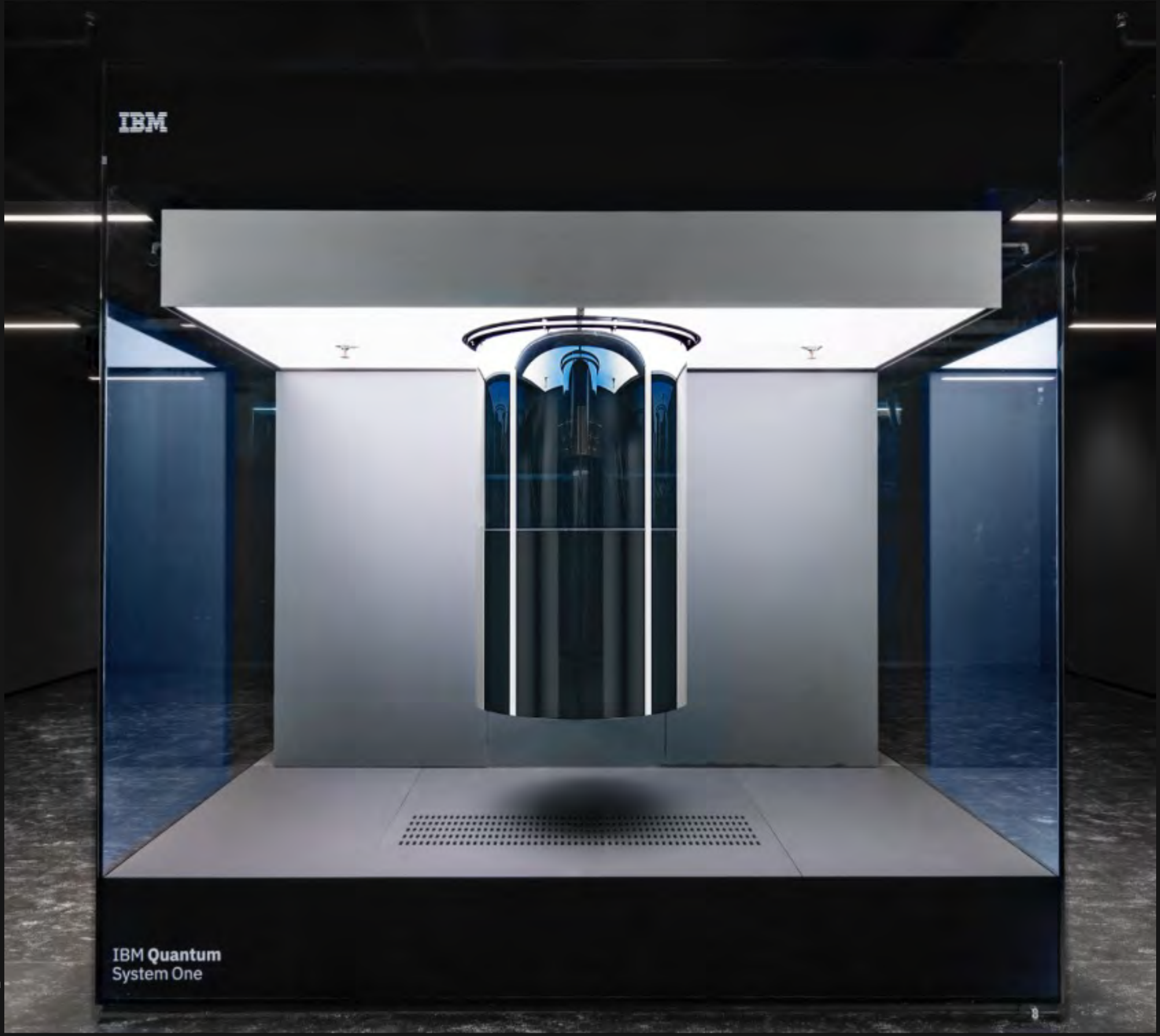
Today: published 1500+ papers using IBM Quantum.

Today: 4 billion quantum circuits each day are run over the cloud

IBM **Quantum**

2019

IBM Quantum System One



2022

IBM Quantum data center

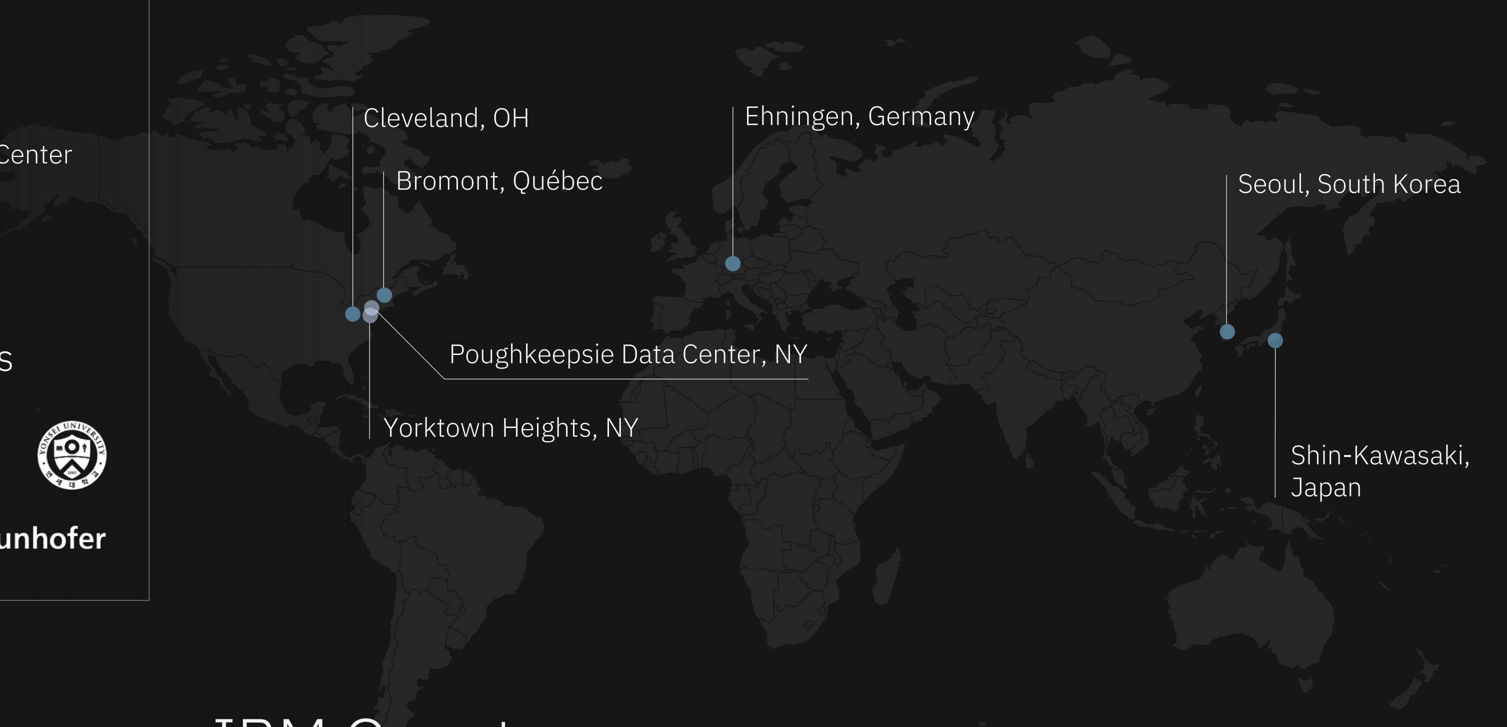
IBM Quantum Datacenters

IBM Poughkeepsie Data Center

Global Deployment of IBM Quantum Computation Centers

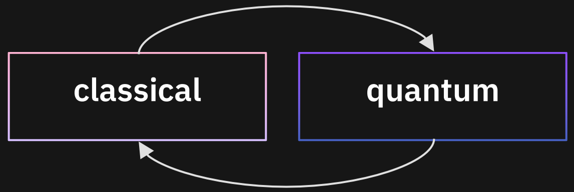
Cleveland Clinic PINQ²

東京大学 THE UNIVERSITY OF TOKYO Fraunhofer



60+ IBM Quantum systems deployed since 2016

Structure of a typical workload



Real workloads are not purely quantum, but rather require interaction between quantum and classical compute resources.

Time scales and resources for quantum computing

User programs

User programs
Seconds to days

Model developers
Enterprises

Quantum serverless

Elastic
100 miliseconds to hours

Algorithm Developers
Model developers

Qiskit Runtime

Near-time
Microseconds to seconds

Kernel Developers
Algorithm Developers

Dynamic circuits

Real-time
Nanoseconds to microseconds

Quantum researchers
Kernel developers

Time scales and resources for quantum computing

User programs

User programs
Seconds to days

Model developers
Enterprises

Quantum serverless

Elastic
100 miliseconds to hours

Algorithm Developers
Model developers

Qiskit Runtime

Near-time
Microseconds to seconds

Kernel Developers
Algorithm Developers

Dynamic circuits

Real-time
Nanoseconds to microseconds

Quantum researchers
Kernel developers

Time scales and resources for quantum computing

User programs

User programs
Seconds to days

Model developers
Enterprises

Quantum serverless

Elastic
100 miliseconds to hours

Algorithm Developers
Model developers

Qiskit Runtime

Near-time
Microseconds to seconds

Kernel Developers
Algorithm Developers

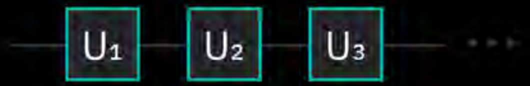
Dynamic circuits

Real-time
Nanoseconds to microseconds

Quantum researchers
Kernel developers

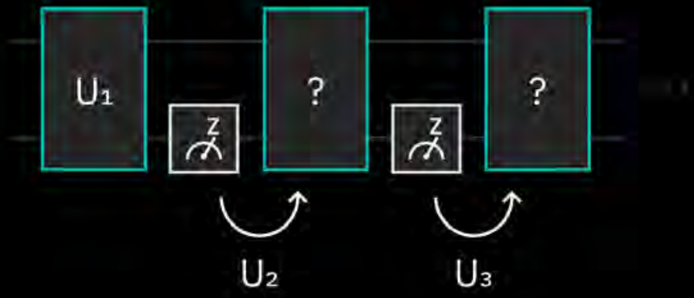
What are dynamic circuits ?

01 Regular quantum circuit



02 Dynamic circuit

New features:
- Mid-circuit measurements
- Real-time classical computing
- Feedforward



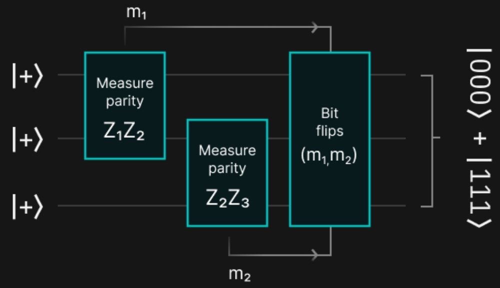
Power of Dynamic Circuits

IBM **Quantum**

Reducing Circuit Depth $N^2 \rightarrow O(1)$

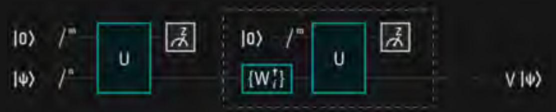
Cost N^2 auxiliary qubits

State preparation

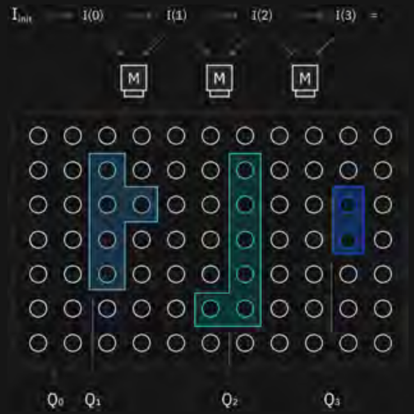


Alternative models for algorithms

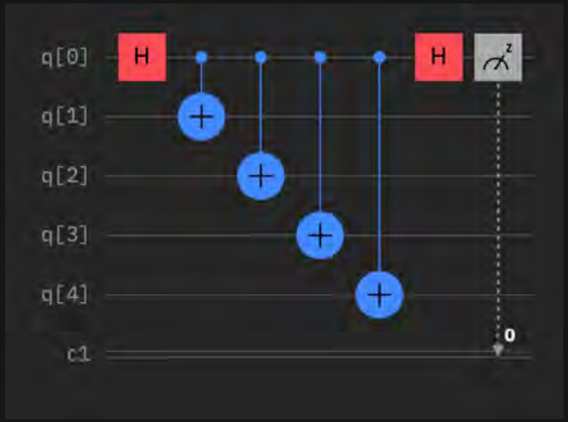
Repeat until success[1]



Measurement model[2]



Parity checks for QEC

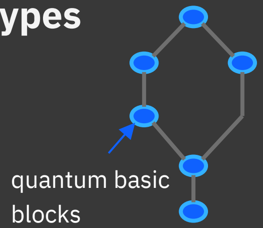


[1] Bocharov et al, PRL **114**, 080502 (2015)
[2] Raussendorf et al, PRA **68**, 022312 (2003)

OpenQASM3

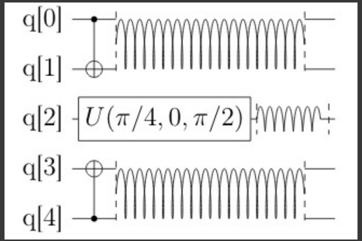
A language for expressing dynamic circuits.

Classical control flow, instructions, and data types



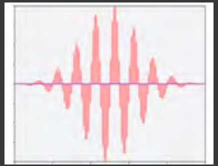
TeX-inspired timing relationships

Boxes and glues in TeX



Boxes and stretches in OpenQASM3

Connect pulse-level descriptions to gates



```
OPENQASM 3.0;
qubit q;   // phase estimation qubit
qubit r;   // target qubit for the controlled-unitary gate
angle[16] c;  // phase estimation bits


// prepare uniform superposition of eigenvectors of phase
h r;

// iterative phase estimation loop
for i in [1:n] {
 reset q;
 h q;
 ctrl @ pow[2 ** i] @ U q, r;
 inv @ phase(c) q;
 h q;
 measure q -> c[0];
 // next iteration acts on the next bit
 c <<= 1;
}
```

A. Cross et al, arXiv:2104.14722
accepted to ACM Transactions on QC

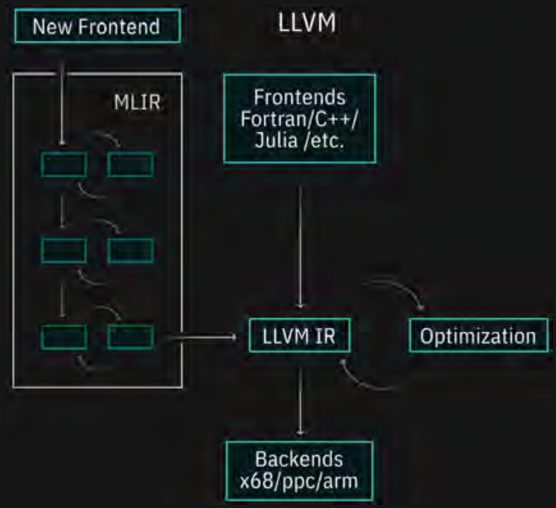
Enabling Technology for Dynamic Circuits

3rd Generation Control System

OpenQASM3 – *circuit language description*

```
OPENQASM 3.0;
qubit q;
qubit r;
angle[16] c;

h r;
for i in [1:n] {
  reset q;
  h q;
  ctrl @ pow[2 ** i] @ U q, r;
  inv @ phase(c) q;
  h q;
  measure q -> c[0];
  c <<= 1;
}
```

OpenQASM3 – *native compiler*



Time scales and resources for quantum computing

IBM **Quantum**

User programs

User programs
Seconds to days

Model developers
Enterprises

Quantum serverless

Elastic
100 miliseconds to hours

Algorithm Developers
Model developers

Qiskit Runtime

Near-time
Microseconds to seconds

Kernel Developers
Algorithm Developers

Dynamic circuits

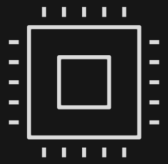
Real-time
Nanoseconds to microseconds

Quantum researchers
Kernel developers

Core Primitives

IBM **Quantum**

Sampler

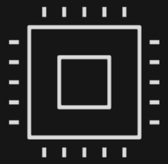


Circuit

$$\tilde{p}(\hat{x})$$

Quasi-probability distribution

Estimator

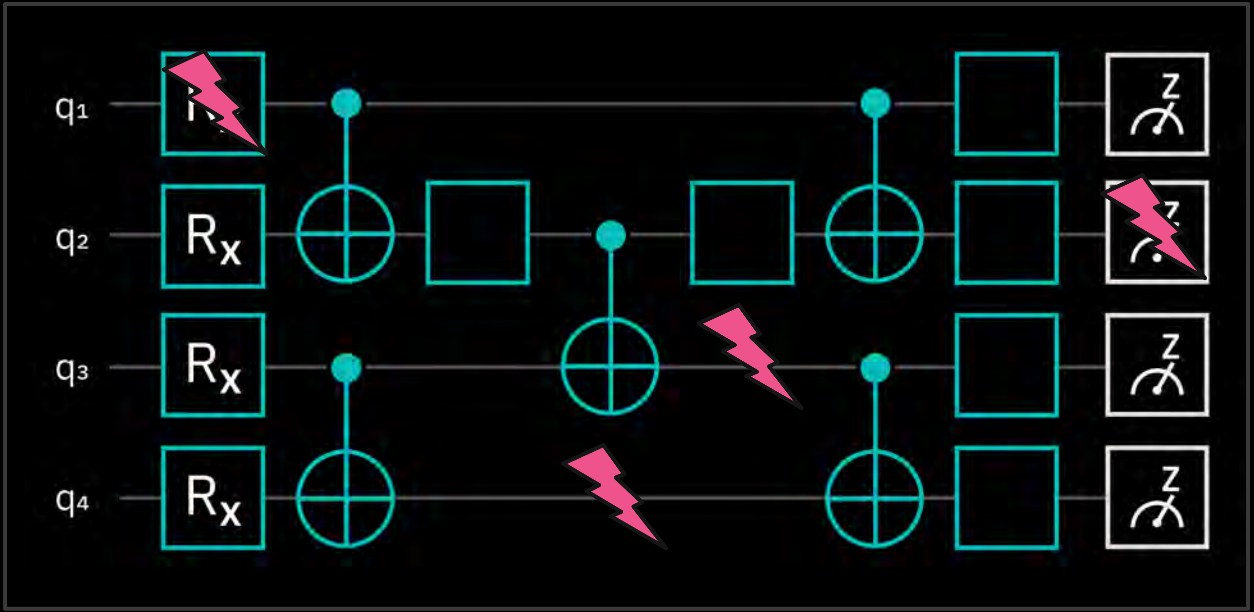


Circuit
+
Observable
$\tilde{O}$

$$\langle \tilde{O} \rangle$$

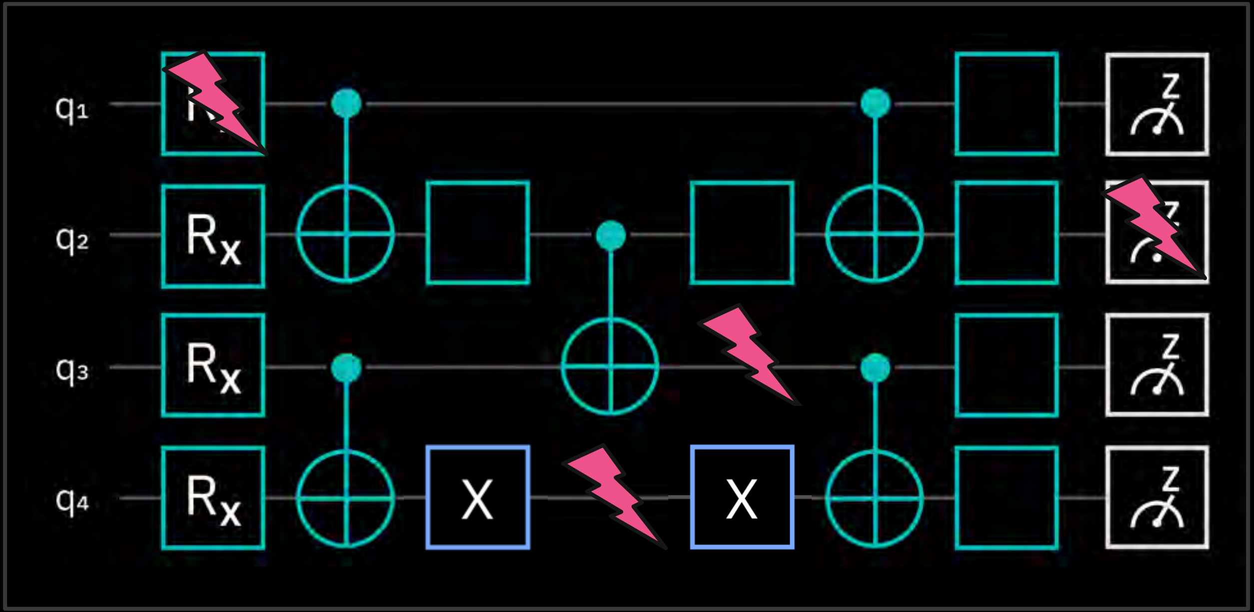
Expectation value

In quantum computation,
we must deal with errors.



In quantum computation, we must deal with errors.

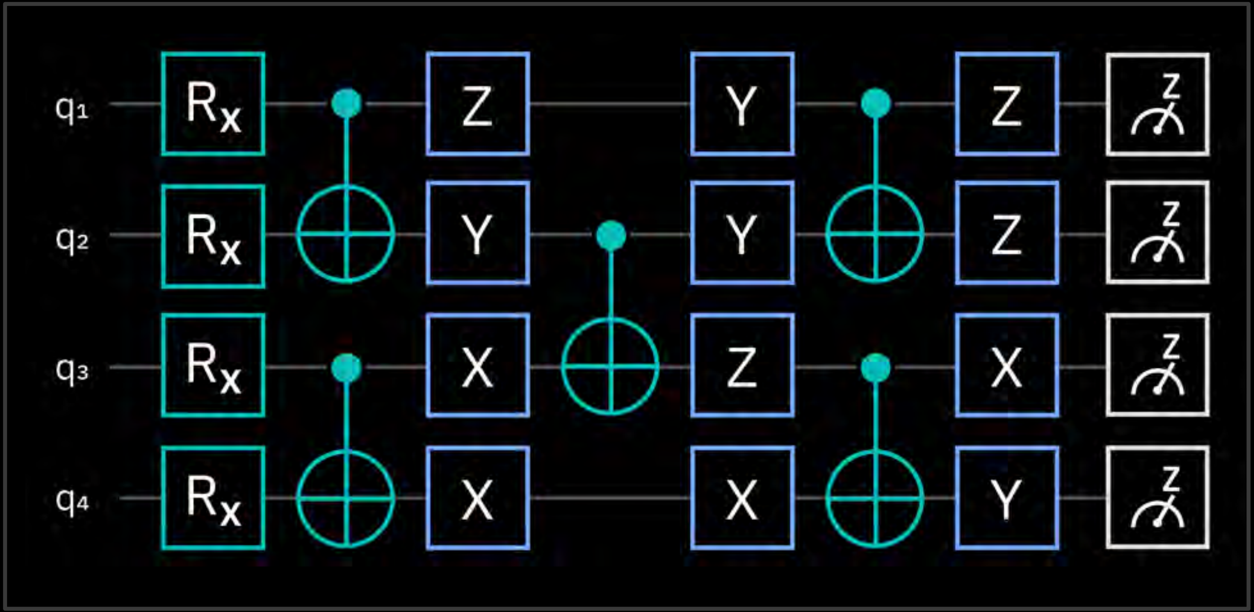
Error suppression reduces errors by modifying the circuit.



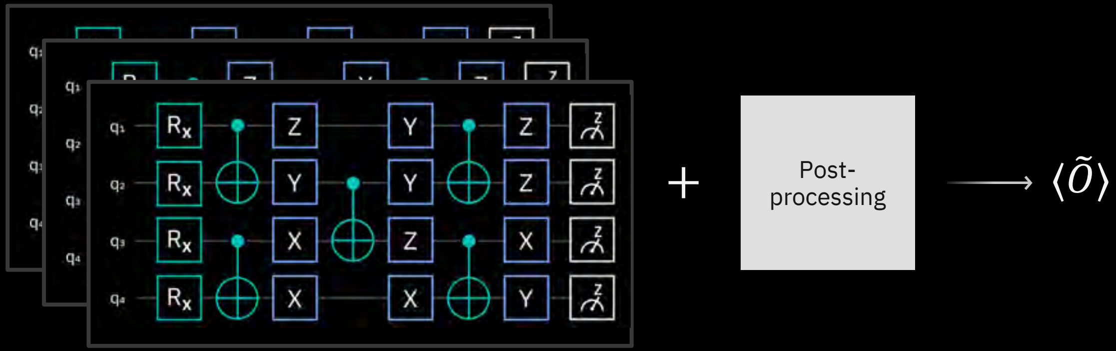
In quantum computation, we must deal with errors.

Error suppression reduces errors by modifying the circuit.

Error mitigation uses outputs from ensembles of circuits to increase accuracy in expectation values.



Error mitigation uses outputs from ensembles of circuits to increase accuracy in expectation values.



Simple interfaces for adjusting accuracy

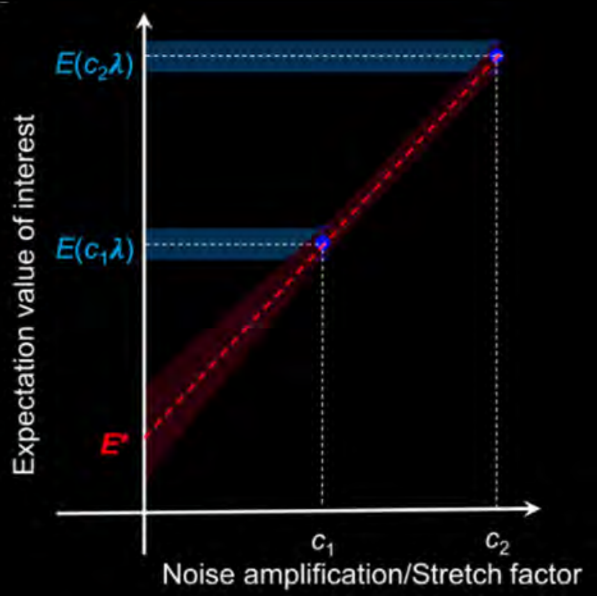
Resilience Level

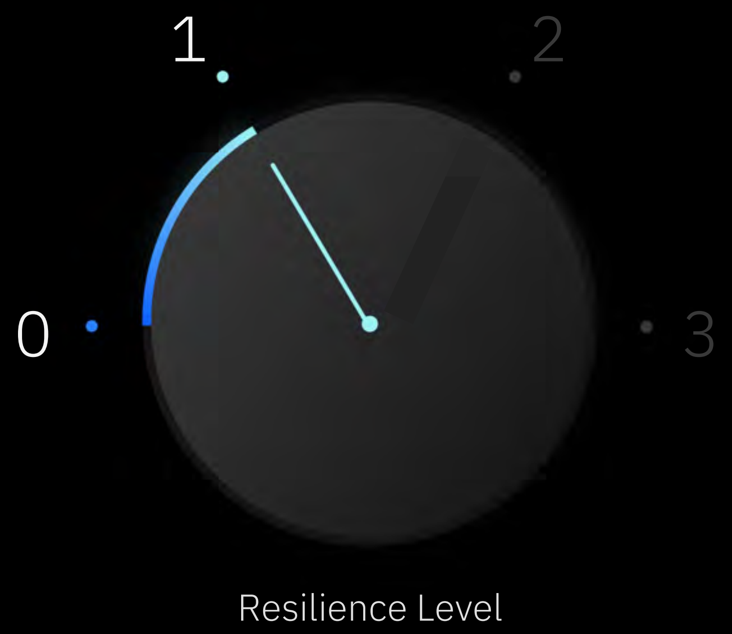
Measurement error mitigation is almost “free”

Primitive	Error Mitigation Method
Sampler	Model-based measurement error mitigation (M3)
Estimator	Model-free measurement error mitigation (TREX)

Error mitigation extends further at the cost of sampling overhead

Zero noise extrapolation

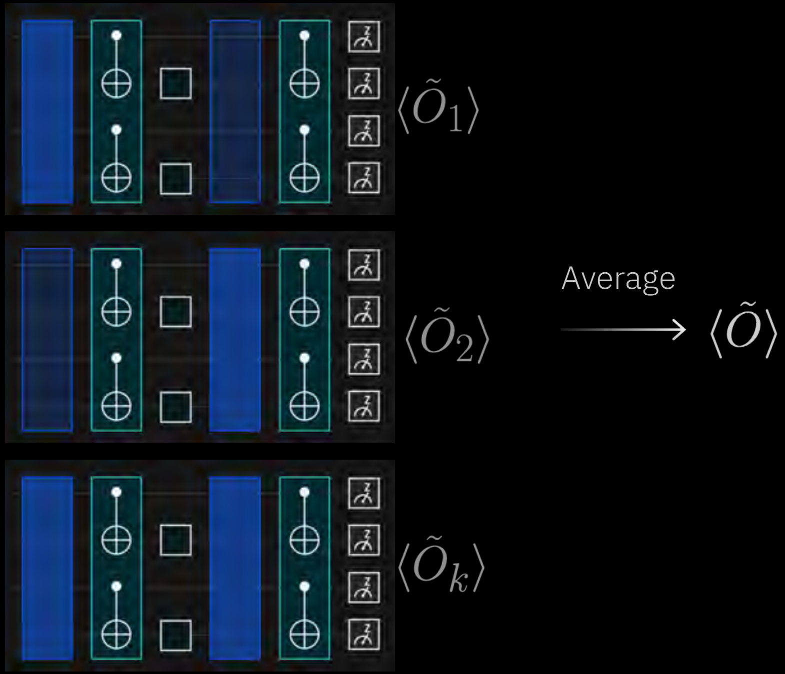


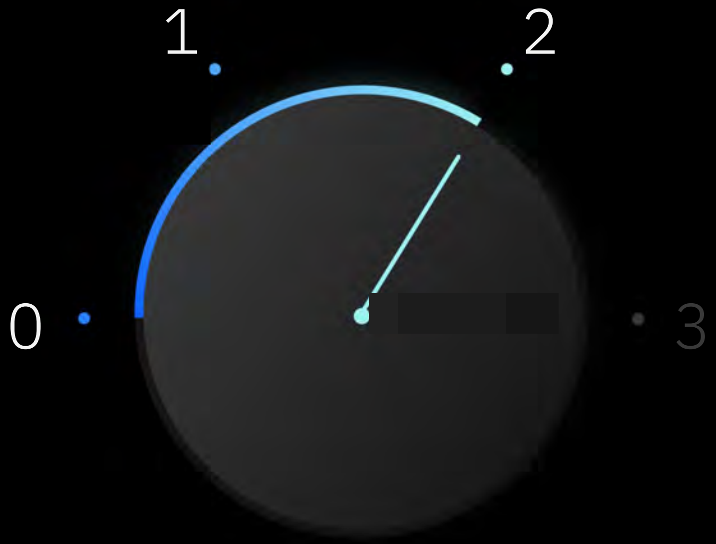


E.g. A. Kandala et al, Nature 567, 491 (2019)

More cost, more accuracy

Probabilistic Error Cancellation



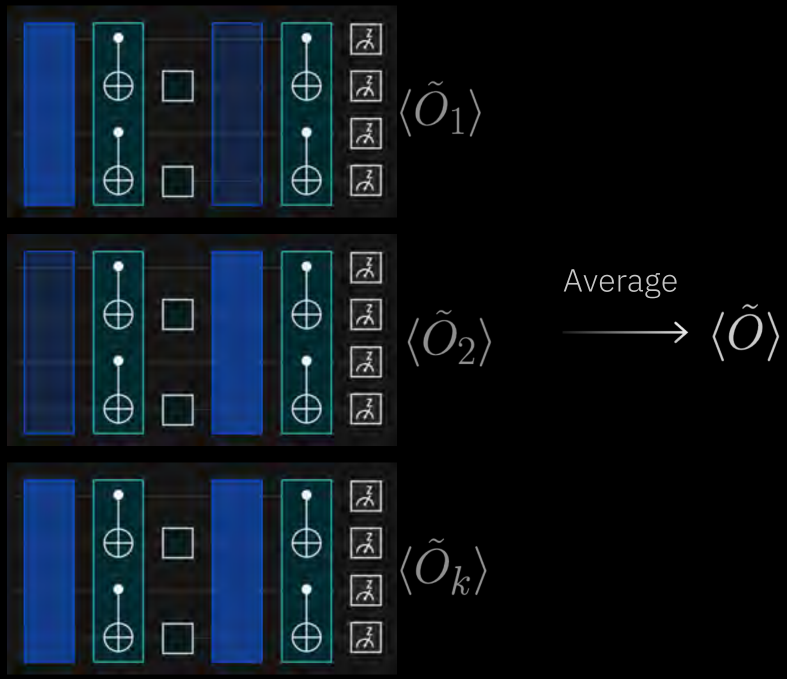


Resilience Level

E. van de Berg, Z. Minev, K. Temme, A. Kandala, arXiv:2201.09866

Or in code...

Probabilistic Error Cancellation




```
options = Options(resilience_level=3)

with Session(service) as session:

    estimator = Estimator(session, options)

    job = estimator.run(circuits=[psi1],
                        observables=[H1],

parameter_values=[theta])
```

E. van de Berg, Z. Minev, K. Temme, A. Kandala, arXiv:2201.09866

Time scales and resources for quantum computing

IBM **Quantum**

User programs

User programs
Seconds to days

Model developers
Enterprises

Quantum serverless

Elastic
100 miliseconds to hours

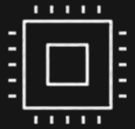
Algorithm Developers
Model developers

Qiskit Runtime

Near-time
Microseconds to seconds

Kernel Developers
Algorithm Developers

Dynamic circuits



Real-time
Nanoseconds to microseconds

Quantum researchers
Kernel developers


IBM Quantum
Hummingbird

FRANCE
HYBRID
HPCQUANTUM
INITIATIVE

HQI: HPC & QC integrated platform

A HYBRID HPC-QC APPROACH

Coupling supercomputers with QPUs

- Quantum computing is **an accelerator** pour for targeted **HPC/AI** applications and algorithms that will be **offloaded to the QPU**
- A **workload evaluation** that must be adapted on existing middleware environments
- A **well-known access procedure**

A **central platform** to build programming environments, develop and provide access to scalable and interconnected quantum computers as well as applications.

GENCI's Joliot-Curie supercomputer operated at TGCC/CEA

FRANCE HYBRID HPCQUANTUM INITIATIVE

QUANTUM COMPUTING: HQI INITIATIVE

Scope: 2022-2026

A **production hybrid HPC-QC platform**

and

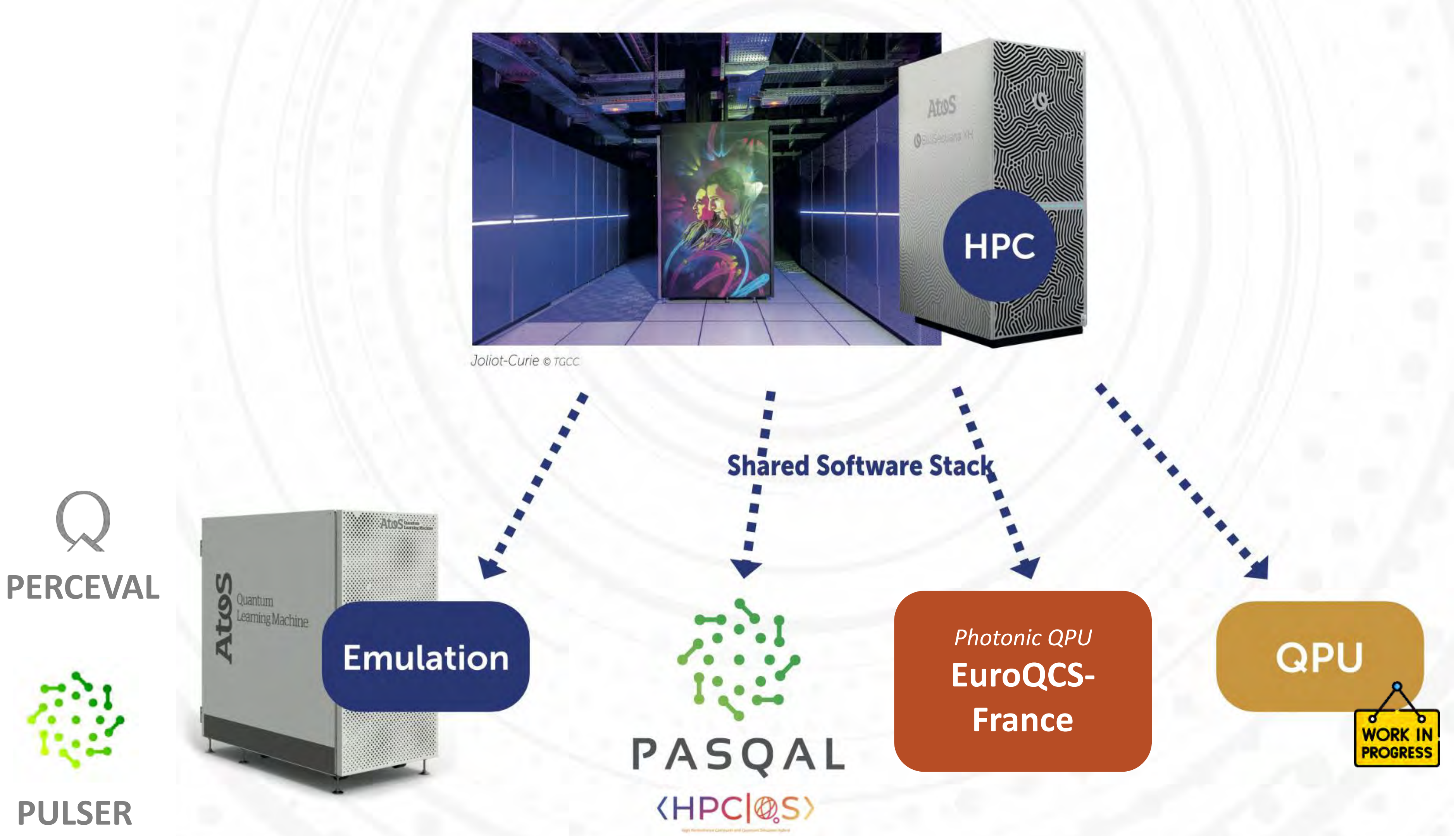
An **academic and industrial research programme**

TGCC
Très Grand Centre de calcul du CEA

Procurement and deployment of QC platforms	GENCI (36.3 M€)	cea
Academic research	cea (36,0 M€)	Atos, cea, cnrs, GENCI, Inria
Industrial research		
Dissemination and end-user community support		

HQI: PROCUREMENT AND DEPLOYMENT OF QC PLATFORMS

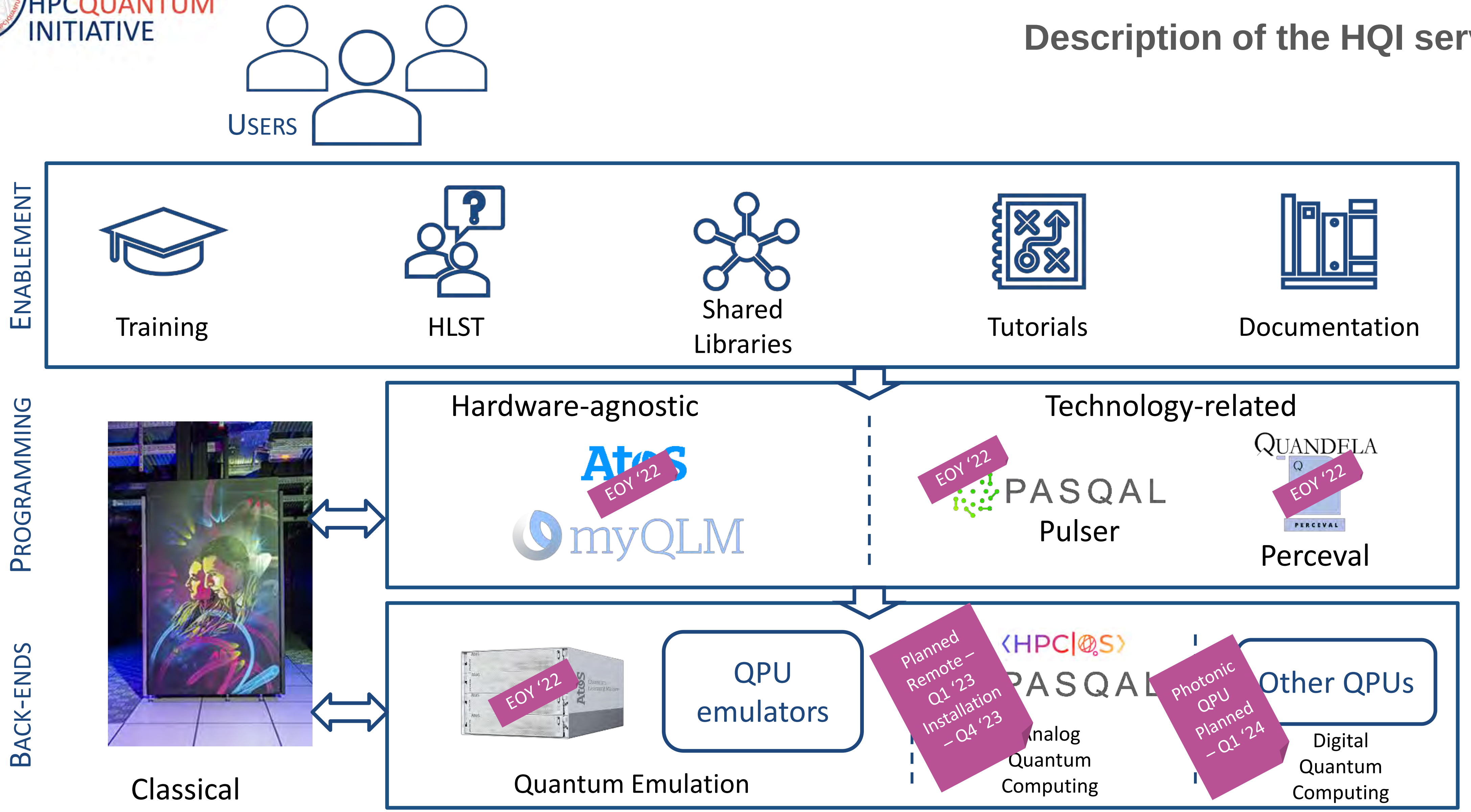
A scalable and open platform



FRANCE HYBRID HPCQUANTUM INITIATIVE

OVERVIEW OF THE HQI ENVIRONMENT

Description of the HQI services



Main stack is based on Atos/QLM software

- To work on quantum algorithms (HQI integration)
  - myQLM runs on supercomputer
  - Offload quantum runtime to QLM appliance to QPUs or to QPUs directly
  - Jupyter notebooks or command line interface
  - QPU emulators run on supercomputer
- To work on hybrid computing in applications (HQI R&D)
  - Integration of quantum runtime in HPC environment
  - Use of API or compiler directive

Direct access to vendor stack is available

- Allow low level interaction with QPUs
- For advance R&D purpose

FRANCE HYBRID HPCQUANTUM INITIATIVE

HPC/QC QPU INTEGRATION

Three levels of integration

Weak coupling

- A step within a processing workflow or an simulation/data analysis run
- Uses the computing centre network
- Requires ability to share the QPU between runs (QPUs do not use time sharing)
- Only solution if the QPU is very expensive

QC are separate systems

HQI initial setup

Medium coupling

- A step within a processing workflow or an simulation/data analysis run
- A new supercomputer local resource
- Uses the supercomputer network
- Allow allocation of a group of QPUs to classic processing

QC are racks in a supercomputer

HQI target

Strong coupling

- Like GPU-type accelerators, can by used in each compute loop
- Uses either compute node buses or the supercomputer network
- Requires high density, low cost QPUs
- Need to put CPU in cryostat?

QC are cards on mother boards

HQI dream

QC COMPUTING CENTRE INTEGRATION

A new computer room in CEA/TGCC

Quantum Computers are simple to integrate but bring new constraints

- (+) Few electrical power (hundreds of kW)
- (+) Few cooling (cryostat are embedded)
- (-) Qubit quantum state need to be protected
  - Required few dust
  - Required few vibrations
  - Required few electromagnetic radiations
  - Required clean/stable electrical power

Classical Computer

- Room air cooling systems generate a lot of dust
- Cooling systems generate a lot of vibrations (pumps, fans)
- Computer power supply supports noisy current
- Large computers with CPU/GPU generate electrical variation (visible on led lighting)

CEA/TGCC strategy is to setup a dedicated computer room in existing facility

- Close to main computer room
- Compatible with short term usage (weak connection)

FRANCE HYBRID HPCQUANTUM INITIATIVE

HQI: ACADEMIC AND INDUSTRIAL RESEARCH

Overview

Pilot design and implementation

- Design and realized a converged hybrid Classic/Quantum computer
  - Based on Atos QLM and Atos XH3000 SuperComputer
  - Emulators
  - LSQ Distributed Heterogeneous Programming
- Improve the software-programming environment (tools for developers)

Applications

- Algorithms for Optimization, Machine Learning, Cryptanalysis
- Quantum simulation of nuclear many-body systems, Quantum computing applied to theoretical and quantum chemistry, Quantum many-body dynamics and entanglement, Quantum simulations of strongly correlated materials, Partial Differential Equation

Exploration

- Noise characterization and mitigation
- Quantum links for computing

HQI: DISSEMINATION AND END-USER COMMUNITY SUPPORT

Helping communities get their hands on the platform

Choice of a sovereign Cloud Services Provider to provide access to the platform

Community platforms - websites, wiki, Slack forum

Organization of dissemination **events**

Development of **International relationships**

Hands-on training for end-users

Setting up an HQI platform end-user **high-level support team**

Development of **use cases** through national and international extensions of the Quantum Pack initiative

Creation of a network of excellence centers: **Houses of Quantum**

FRANCE HYBRID HPCQUANTUM INITIATIVE

HQI France

@HQI_France

Thank you

For more information on the HQI initiative, please contact:

Jacques-Charles Lafoucriere
HQI Programme Manager
jacques-charles.lafoucriere@cea.fr

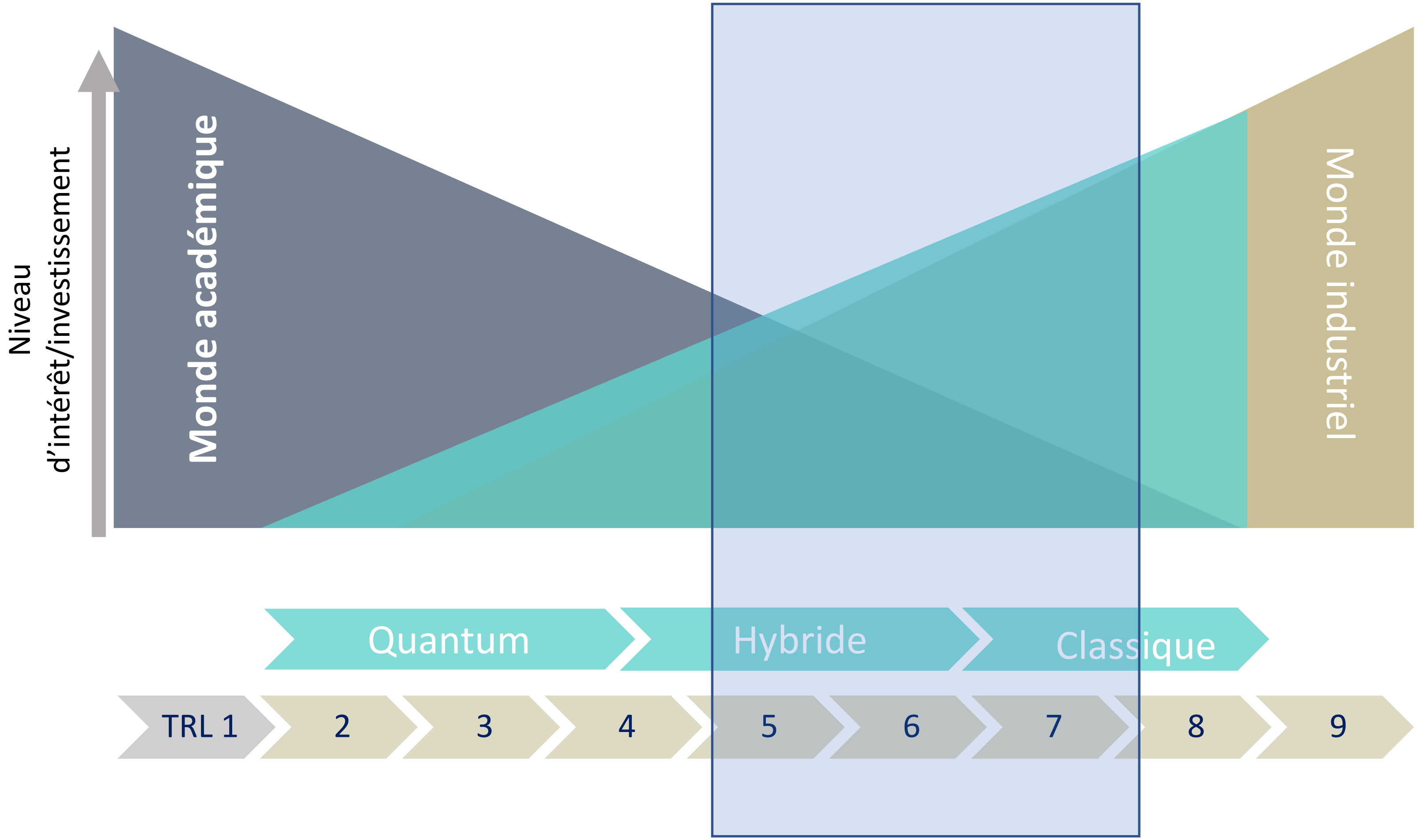
ZONES D'INNOVATION QUÉBEC

SHERBROOKE QUANTIQUE

435 millions $

MANUFACTURIER INNOVANT
3 Usine CINQ
4 Espace Quantique 1
Zone T.I.
Cégep de Sherbrooke
Ville de Sherbrooke
Quartier patrimonial
Sporobole (Hub numérique culturel)
Pôle multimodal
Chambre de commerce et industrie de Sherbrooke
Gorge de la rivière Magog
Espaces de travail collaboratif
Espace-INC
Productique Québec
Outillage spécialisé
Amélioration de la productivité, automatisation et numérique
Conception et fabrication d'équipements automatisés
Quartier général de l'entrepreneuriat
Sherbrooke Innopole
Destination Sherbrooke
Entreprendre Sherbrooke
Groupe des partenariats d'affaires (9 partenaires)
Ancienne zone industrielle à reconvertir en manufacturier innovant
Vieille gare revitalisée
Berges de la rivière Saint-François à revitaliser
Corridor de mobilité durable et active
Éco-quartier Humano District (Sainte-Famille)
CENTRE-VILLE
5 Quartier général de l'entrepreneuriat
Université de Sherbrooke
CTA Centre de technologies avancées BRP - UdeS
Centre de mise à l'échelle
ACET Banque Nationale
3IT Institut interdisciplinaire d'innovation technologique
Parc solaire
Mont-Bellevue
PARC INNOVATION - ACELP
1 Carrefour d'innovation technologique et d'économie circulaire
2 Institut quantique
Boul. de Portland
King O.
King E.
Galt O.
Galt E.
Terrill

« Dé-risquer » l’innovation commerciale, axées sur solutions numériques classiques, hybrides et quantiques



** Source Etudes de Marché iA7- mars 2021- portant sur le marché nord-américain des infrastructures de calcul de haute performance*

UdS Université de Sherbrooke

INSTITUT QUANTIQUE

ZONES D'INNOVATION QUÉBEC

SHERBROOKE QUANTIQUE

PASQAL

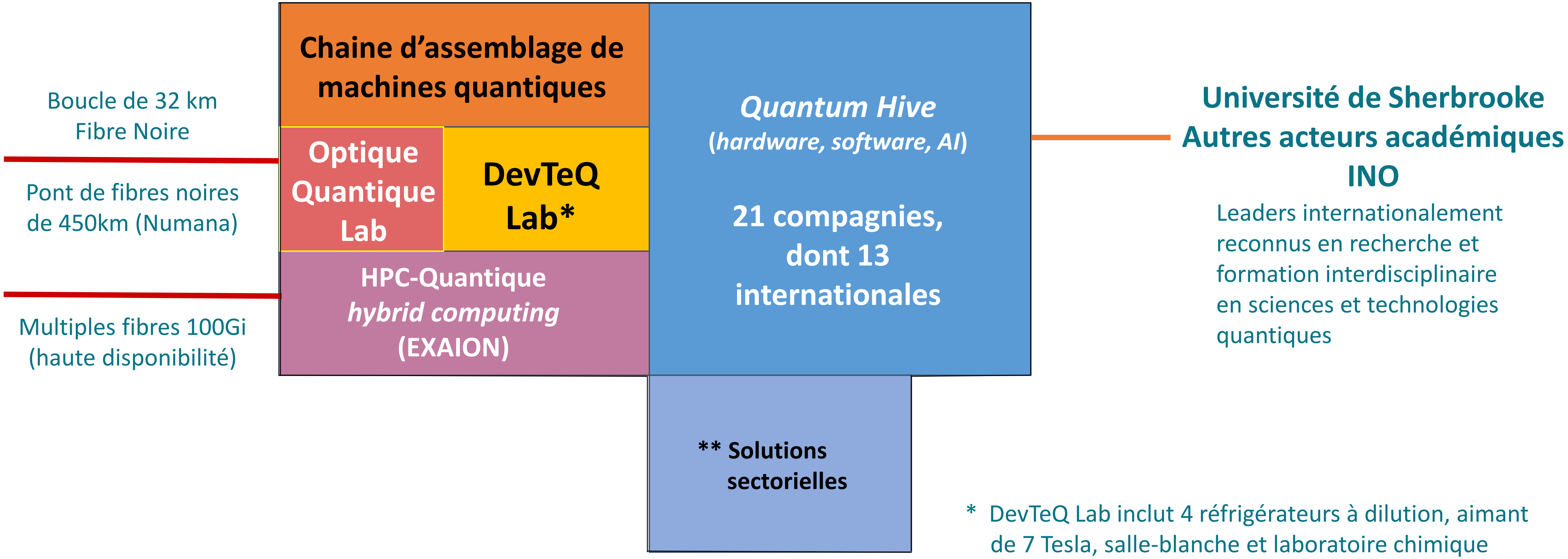
EDF

Espace Quantique 1

Solutions d'Industrie Précompétitives Utilisant des Technologies Quantiques

- Immeuble dédié au développement d'un écosystème quantique de 6 100 mètres2 (66 000 pi2) de solutions quantiques pour l'industrie (énergie, chimie, minière, etc.)
- Espace existant de 48 000 pi2 en cours de rénovation additionné d'un agrandissement (voir carré bleu) de 18 000 pi2 locatifs supplémentaires doublant ainsi la valeur des loyers potentiels
- Activités TRL 6-8
(aucune activité académique)

Espace Quantique 1



* DevTeQ Lab inclut 4 réfrigérateurs à dilution, aimant de 7 Tesla, salle-blanche et laboratoire chimique

** Zone agrandissement par industrie

Centre d'excellence QuaTERA

Quantum Technologies Energy Result Accelerator

PASQAL

About News Resources Careers Contact

November 15, 2022 in **Latest News**

EDF, Exaion Inc., PASQAL and the Quantum Innovation Zone Bring High-Performance Computing and Quantum Technologies to the Forefront of Energy Solutions

(Sherbrooke, Quebec, Canada, November 15, 2022) — EDF, Exaion Inc., PASQAL and the Quantum Innovation Zone are joining forces to create the first open center of excellence to develop sustainable energy solutions using the combined capabilities of HPC and quantum computing. As the world is facing unprecedented energy challenges, we aim to provide the energy industry with sustainable quantum-based solutions by 2024.

Partenaires impliqués et engagés

Nord Quantique
BLUEFORS
1QBit
PASQAL
multiverse
SBQuantum
ESPACE IBM Q
INNO
FAIRE LA LUMIÈRE
SHEDDING LIGHT
Chemia Discovery
Bell
3IT Institut interdisciplinaire d'innovation technologique
UNIVERSITÉ DE SHERBROOKE
QUANTO NATION
Québec
quantino
Quantacet
BDC
ACET
ENTREPRENEURIAT LAVAL
COMPLICE DE VOS PROJETS
EN TECH
t0.technology
CMC MICROSYSTEMS
PINQ²
PLATEFORME D'INNOVATION NUMÉRIQUE ET QUANTIQUE
QUÉBEC QUANTIQUE
C2MI
Centre de Collaboration MiQro Innovation
NUMANA

The impact of compilation in the implementation of quantum computing

Software to the rescue of hardware

Simon Martiel
Researcher @ Atos Quantum
11/01/2023

Atos

When software helps hardware: quantum circuit compilation

Quantum software is a (very) broad topic:

- HPC integration
- Cloud integration
- Quantum Programming languages
- Formal methods (ZX calculus, static analysis)

In this talk we will focus on Quantum Circuit compilation/optimization

Atos

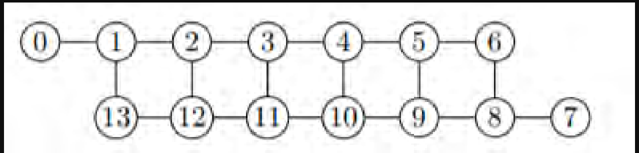
Quantum Circuit Compilation/Transpilation
But why ?

Typical quantum circuits:

- Contain large gates (C.....CNOTs) and black-boxed primitives
- Contain all kind of weird gates

Typical quantum hardware (NISQ setting):

- Comes with gate-set limitation ({ CNOT, U3 }, { CZ, $RX(\frac{\pi}{2})$, RZ }, ...)
- (Usually) comes with connectivity restrictions
- Each operation has some (large) error rate



<u>A good compiler needs to reduce gate count/depth while matching all those constraints !</u>

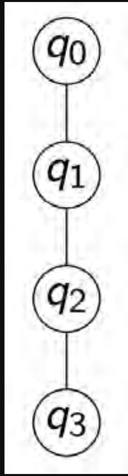
Focus on qubit routing

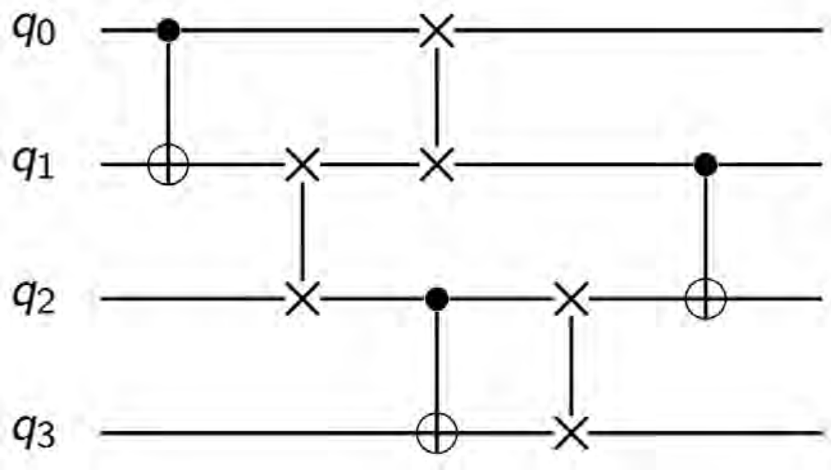
The problem

Input : Some circuit and some connectivity graph

Output : Some equivalent circuit matching the connectivity

Standard approach: SWAP insertion



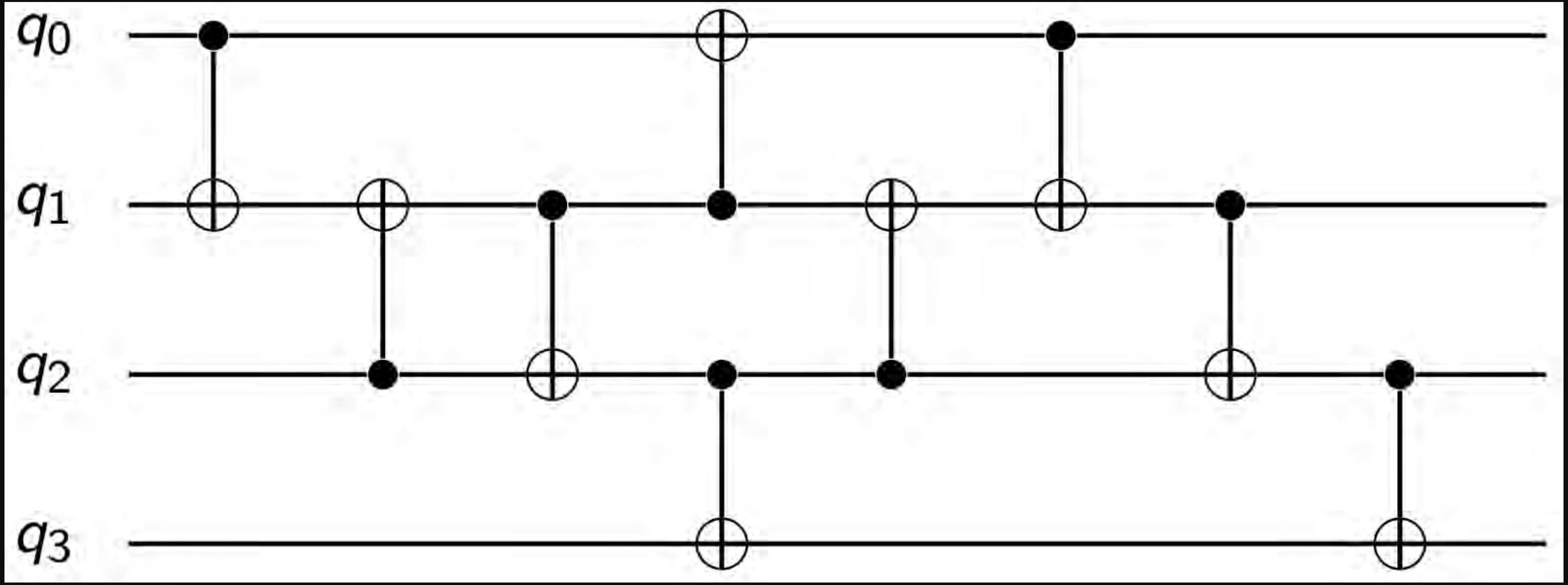


Final cost: 12 CNOTs
(3 + 3 SWAPS)

Focus on qubit routing

Ad hoc synthesis as an alternative to SWAP insertion

Can we be smarter (in that example) ?



This circuit is equivalent and contains 9 CNOTs !

[Kissinger et al. (2019)]

Atos

Focus on qubit routing

Ad hoc synthesis as an alternative to SWAP insertion

We know how to synthesize circuits for:

- Qubit permutations (SWAP circuits)
- Boolean linear maps (CNOT circuits)
- Phase Polynomials (CNOT + RZ circuits)
- Clifford (CNOT + H + S)

We don’t known how to do it for arbitrary circuits

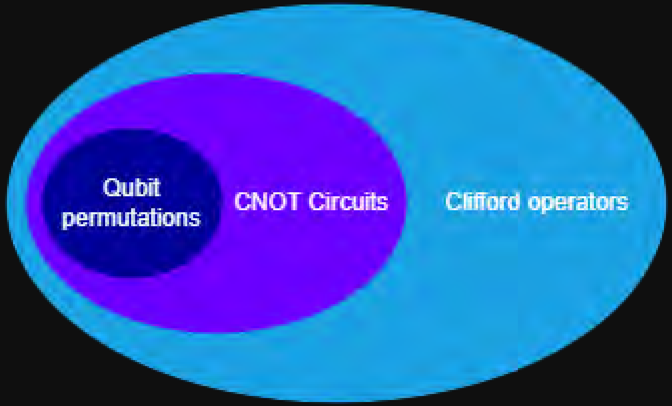
Our take on qubit routing

With Timothée Goubault de Brugière

Our take:

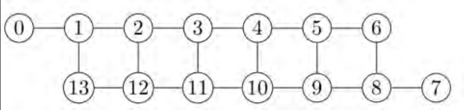
- Work with a particular set of operators
- Read circuit from left to right:
  - If the gate is in the set of operators => free cost (update current) operator
  - If not, lazily synthesize a piece of the operator using standard techniques

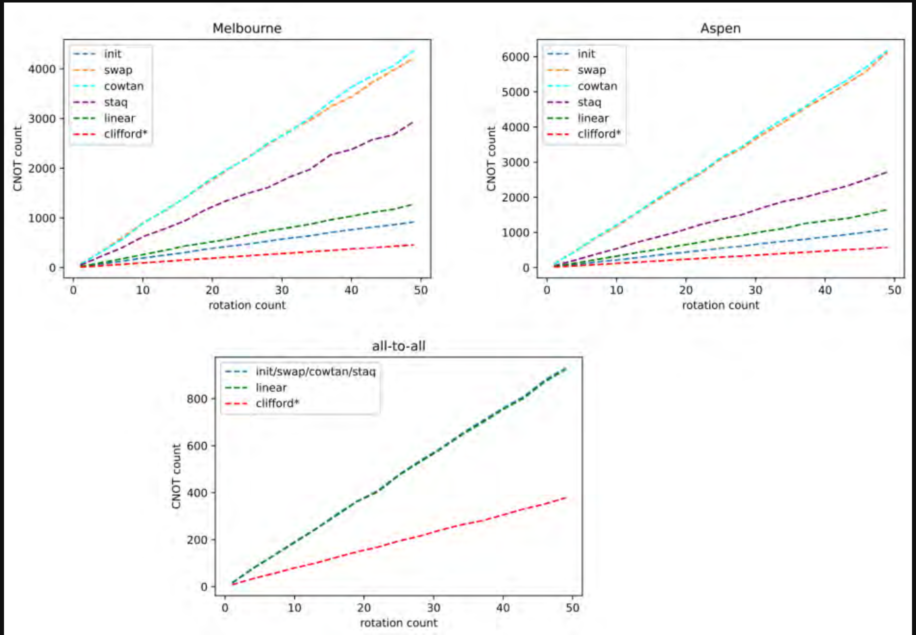
Implemented for the following operators:

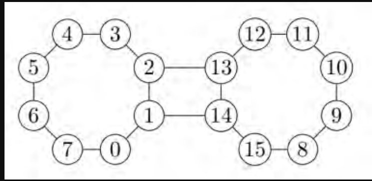


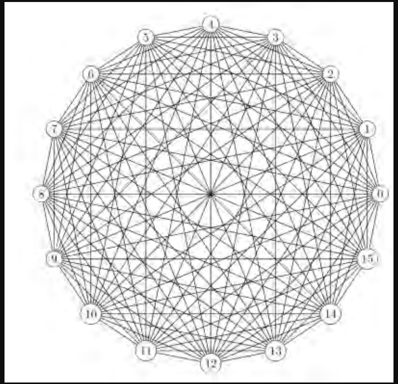
Quick benchmarks of our compiler

With Timothée Goubault de Brugière





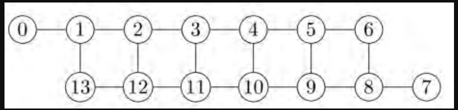




Quick benchmarks of our compiler

With Timothée Goubault de Brugière

circuit	init	swap	linear	clifford	clifford⋆	clifford†	clifford⋆†	cowtan	staq
tof_3	18	116.7%	150.0%	138.9%	77.8%	127.8%	72.2%	133.3%	77.8%
barenco_tof_3	24	75.0%	66.7%	66.7%	50.0%	25.0%	-4.2%	100.0%	45.8%
mod5_4	28	117.9%	60.7%	25.0%	-3.6%	0.0%	-21.4%	171.4%	117.9%
tof_4	30	110.0%	150.0%	120.0%	103.3%	116.7%	83.3%	160.0%	100.0%
tof_5	42	135.7%	276.2%	226.2%	214.3%	157.1%	109.5%	150.0%	54.8%
qft_4	46	176.1%	60.9%	28.3%	19.6%	-23.9%	-19.6%	117.4%	56.5%
barenco_tof_4	48	112.5%	170.8%	87.5%	87.5%	12.5%	0.0%	150.0%	60.4%
mod_mult_55	48	337.5%	345.8%	220.8%	181.2%	172.9%	168.8%	193.8%	306.2%
vbe_adder_3	70	107.1%	60.0%	38.6%	11.4%	-32.9%	-17.1%	120.0%	135.7%
barenco_tof_5	72	112.5%	245.8%	119.4%	127.8%	41.7%	20.8%	137.5%	59.7%
rc_adder_6	93	180.6%	76.3%	31.2%	31.2%	-7.5%	-10.8%	112.9%	221.5%
gf2^4_mult	99	184.8%	278.8%	205.1%	93.9%	180.8%	84.8%	197.0%	381.8%
mod_red_21	105	165.7%	204.8%	116.2%	105.7%	79.0%	58.1%	171.4%	210.5%
hwb6	116	196.6%	169.0%	91.4%	64.7%	67.2%	52.6%	152.6%	205.2%
grover_5	288	116.7%	210.4%	245.1%	166.3%	194.4%	91.7%	121.9%	92.0%
hwb8	7129	224.2%	168.5%	169.7%	156.6%	134.4%	114.1%	183.6%	280.5%



Lazy operator synthesis:

- A framework for quantum circuit compilation
- More than competitive for VQE like circuits
- Almost always outperforms SWAP insertion

Published in Quantum

Architecture aware compilation of quantum circuits via lazy synthesis, S. M., Timothée Goubault de Brugière, Quantum

Atos

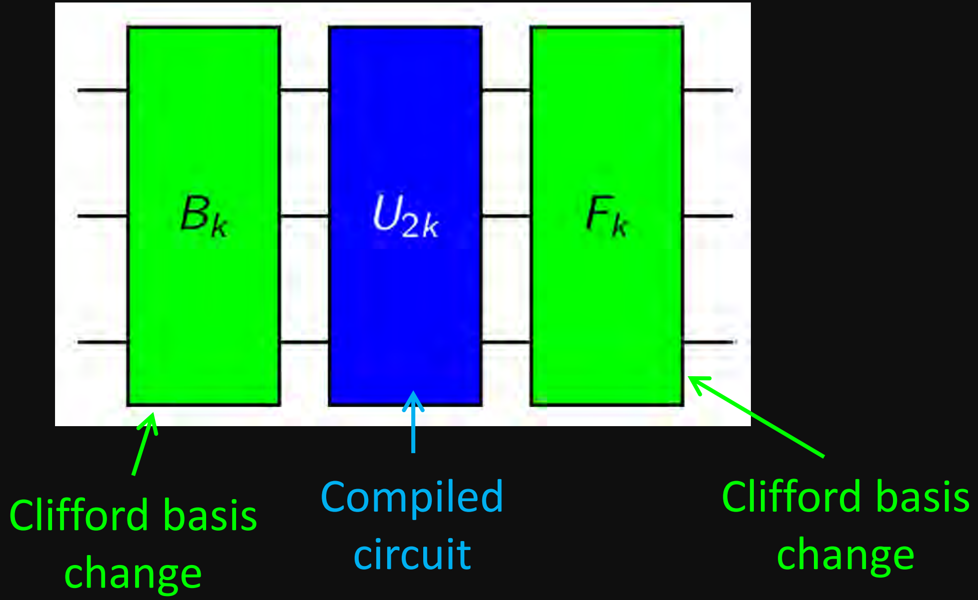
Extension to bidirectional normalization

With Arnaud Gazda, Timothée Goubault de Brugière, and Christophe Vuillot

Idea: explore Clifford basis to optimize the circuit

B_k is synthesized via
stabilizer state synthesis

F_k is synthesized via
Pauli operators co-diagonalization
+ classical post-processing



A graph-state based synthesis framework for Clifford isometries, Timothée Goubault de Brugière, S.M., Christophe Vuillot, Pre-print

Atos

A all-in-one compiler

Initial Transpilation
qat-pbo

Breaking down large gates

Rotation pruning
lazy synthesis

Bidirectional renormalization
lazy synthesis

Lazy operator synthesis (compilation & optimization)

2-qubit gate transpilation
qat-pbo

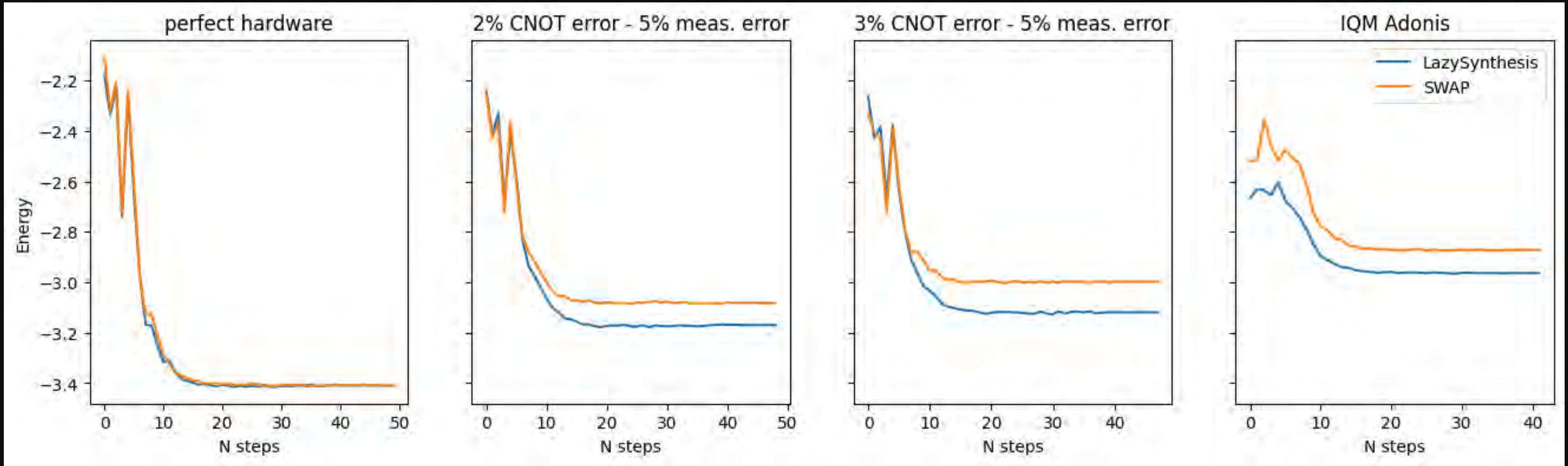
1-qubit gate compression
qat-pbo

Circuit rewriting + 1-qbit gate compression

A all-in-one compiler

Simulation and real hardware runs - Combinatorial Optimization applications

QAOA – MaxCut – $G(5,\frac{1}{2})$ – avg. over 100 runs

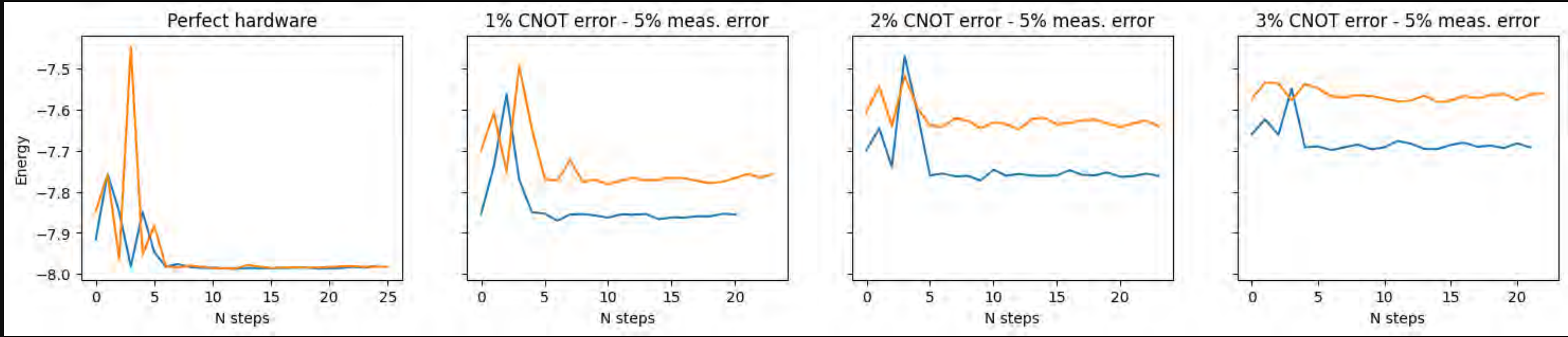


Atos

A all-in-one compiler

Simulation - Quantum Chemistry applications

UCCSD VQE for LiH

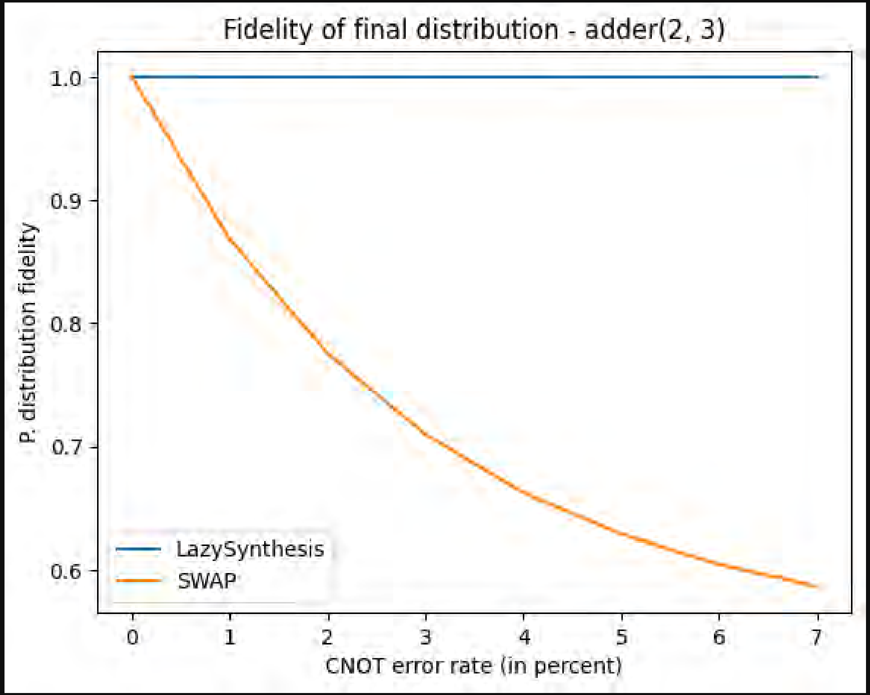


Atos

A all-in-one compiler

Simulation - Arithmetic quantum circuit

QFT-based adder(2,3)



Conclusion and perspectives

We presented:

- A generic compilation framework
- And its embedding in an all-in-one compiler that covers most usages

Benchmarks show that the compiler does *increase the algorithmic performances* of the QPU

Available in the QLM framework : NISQCompiler plugin

Further work:

- A more subtle target metric (here we minimized the gate count)
- Scalability improvements (we are limited to less than 50 qbits)

Thank you

simon.martiel@atos.net

Atos

Los Alamos NATIONAL LABORATORY

Integrating High-Performance Computing with Quantum Computing

Scott Pakin

11 January 2023

LA-UR-22-29744

Focus of This Talk

- Experience and opportunities for using quantum computing to accelerate high-performance computing applications
- Experience and opportunities for using high-performance computing to help develop quantum computing applications

Complementary Roles

- High-performance computing is good for computations that…
  - Input, manipulate, and output large amounts of data
  - Involve many tasks cooperating to solve a large problem
  - May be floating-point intensive
- Quantum computing is good for computations that…
  - Input and output minimal amounts of data but perform extreme amounts of work on that data
  - Reveal global properties of data
  - Work with discrete data

More general-purpose

At best constant speedup over sequential, classical execution (but possibly a very large constant)

More specialized

At best exponential speedup over sequential, classical execution (but typically coming with a constant-time performance penalty)

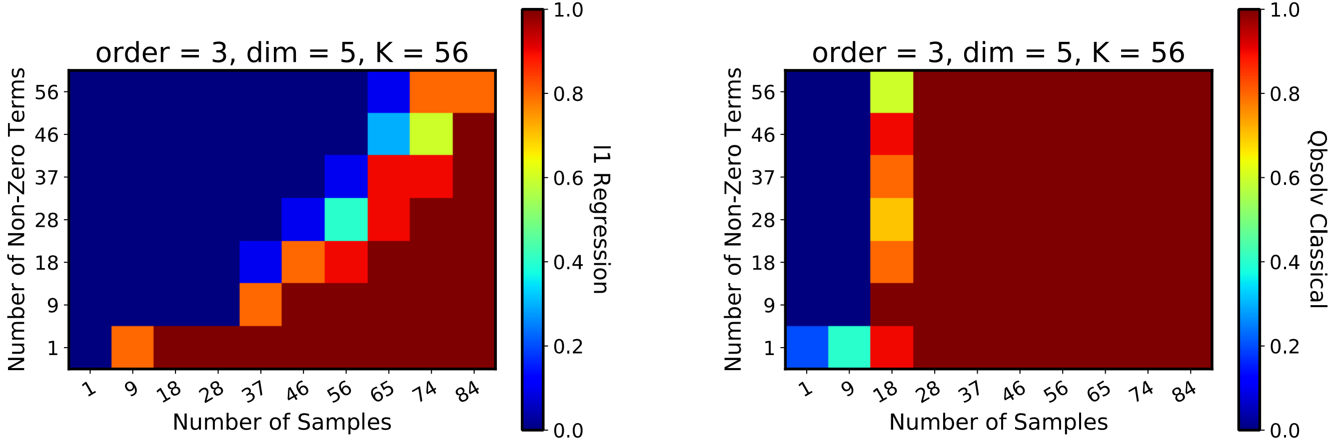
QC for HPC

Quantum Optimization for Uncertainty Quantification

- Problem
  - Want the $\bar{x}$ that minimizes $f(\bar{x})$
  - $f(\bar{x})$ is expensive to evaluate, typically requiring a long-running HPC simulation
  - How to evaluate only those $\bar{x}$ with a good chance of minimizing $f(\bar{x})$?
- Solution
  - Use a 0–1 linear combination of many basis functions as a surrogate for $f(\bar{x})$
  - Have a quantum computer find which combination of basis functions best fits the known $\{\bar{x}, f(\bar{x})\}$
  - Minimize the surrogate function also using a quantum computer
  - Evaluate the real f at the point that minimizes the surrogate
  - Repeat the process including the new $\{\bar{x}, f(\bar{x})\}$ pair

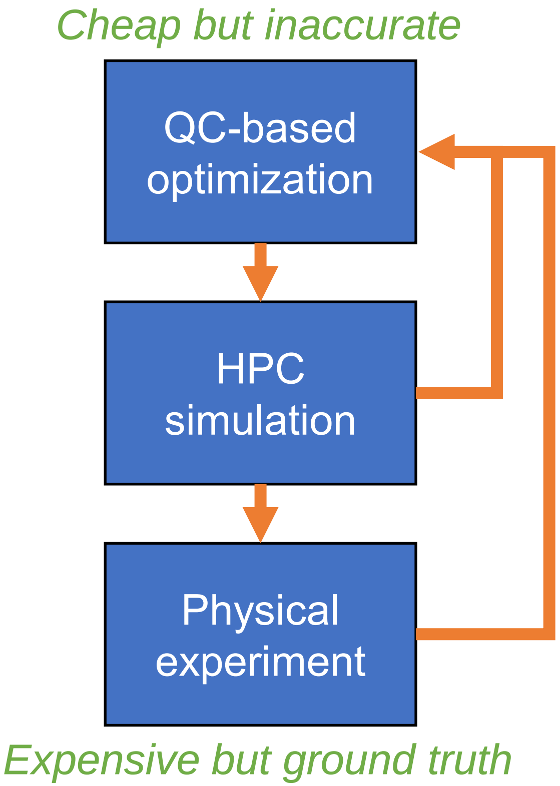
Quantum Optimization for Uncertainty Quantification (cont.)

- Work by Bert Debusschere, Khachik Sargsyan, and Ojas Parekh (Sandia National Laboratories) on LANL's D-Wave System
- Better recovery when using 0–1 coefficients on D-Wave (right) than the more traditional real-valued coefficients on a classical computer (left)



A Variation: Model Parameterization

- (A project that unfortunately never got off the ground)
- Problem
  - Need to find values for a parameterized model that best fit the experimental data
  - Experimental data are expensive and time-consuming to acquire and therefore in short supply
  - HPC simulations are faster but not 100% accurate
- Solution
  - As in the UQ example, use quantum computing to optimize a surrogate function
  - Based on the quantum computer's recommendation, select HPC simulations to run
  - Only when finding model parameters that look very promising, gather more experimental data

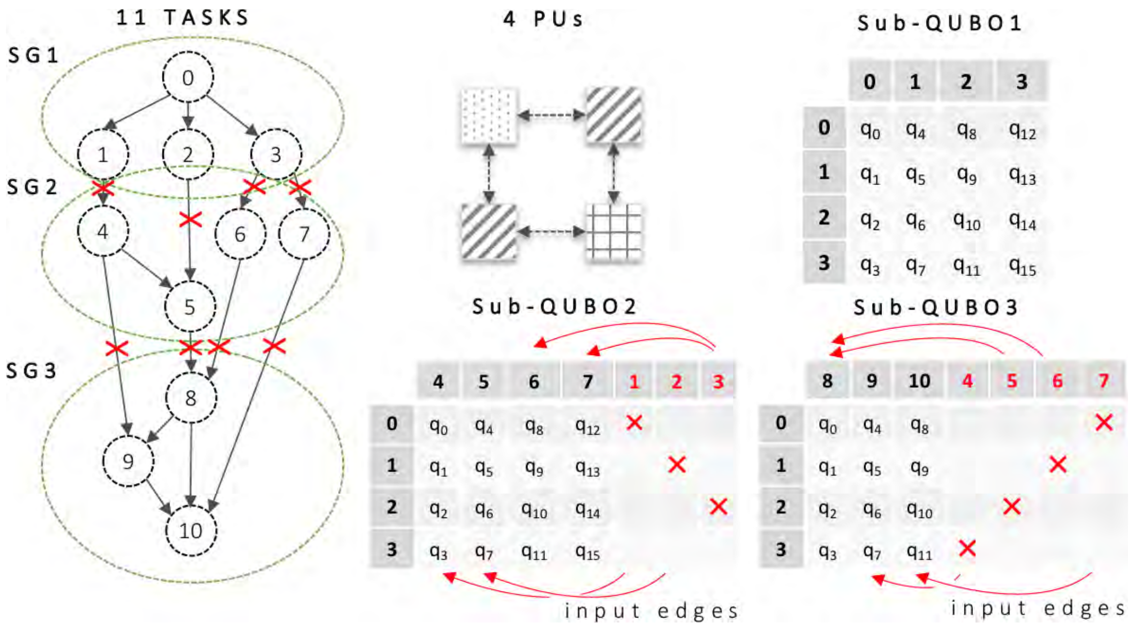


Mapping Compute Tasks to HPC Hardware

- Problem
  - Have many, possibly small, communicating tasks that need to run on heterogeneous HPC hardware (e.g., including CPUs and GPUs)
  - If tasks are maximally spread across the system, parallelism is maximized (good), but communication overhead can dominate execution time (bad)
  - If tasks are packed onto few processors, communication is minimized (good), but available parallelism is not exploited (bad)
- Solution
  - Encode a QUBO that maps each task to exactly one processing unit
  - Partition the task graph based on dependency levels
  - Represent inter-task communication with a quadratic coefficient proportional to the communication cost
  - Use a quantum annealer to find the mapping that maximizes performance

Mapping Tasks to HPC Hardware (cont.)

- Work by Anastasiia Butko (Lawrence Berkeley National Laboratory) on LANL's D-Wave System
- Found promising performance and scalability relative to classical solvers



Partitioning a task communication graph and mapping it to multiple QUBOs

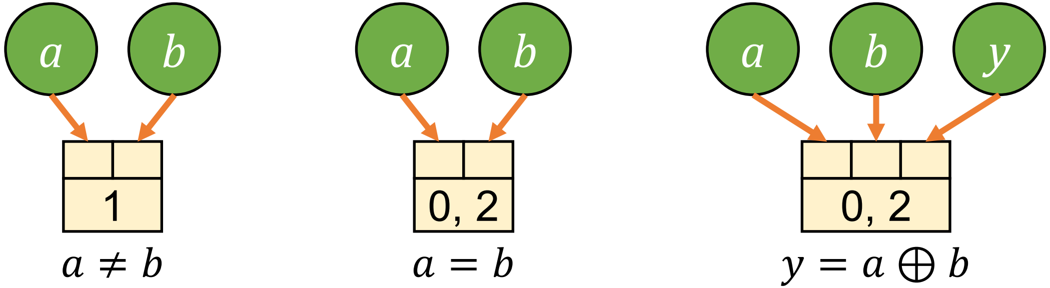
HPC for QC

My Current Project: A Classical Programming Model for Quantum Computers

- Goals
  - Relatively easy to use by traditional HPC developers
  - Applicable to a range of problems
  - Portable across different QCs and (for development) even classical computers
  - Able, at least potentially, to deliver a performance benefit
- Work in progress: **NchooseK**
  - New constraint-programming system
  - Designed for simplicity of expressing problems and of compiling code to both circuit-model and annealing-model QCs

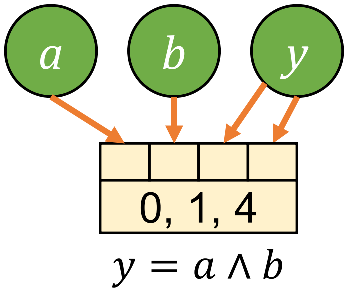
NchooseK Semantics

- Only one basic primitive: $nck(N, K)$
  - N is a multiset of variables, e.g., $\{a, a, b\}$
  - K is a set of numbers, e.g., $\{1, 3\}$
- Interpretation: "I want any $k \in K$ of the Boolean variables listed in N to be TRUE"
- Examples:
  - $nck(\{a, b\}, \{1\})$: "I want exactly one of a and b to be TRUE (i.e., $a \neq b$)"
  - $nck(\{a, b\}, \{0, 2\})$: "I want either both or neither of a and b to be TRUE (i.e., $a = b$)"
  - $nck(\{a, b, y\}, \{0,2\})$: "I want either zero or two of a, b, and y to be TRUE (i.e., $y = a \oplus b$)"
    - Only FFF, FTT, TFT, and TTF have either 0 or 2 TRUE



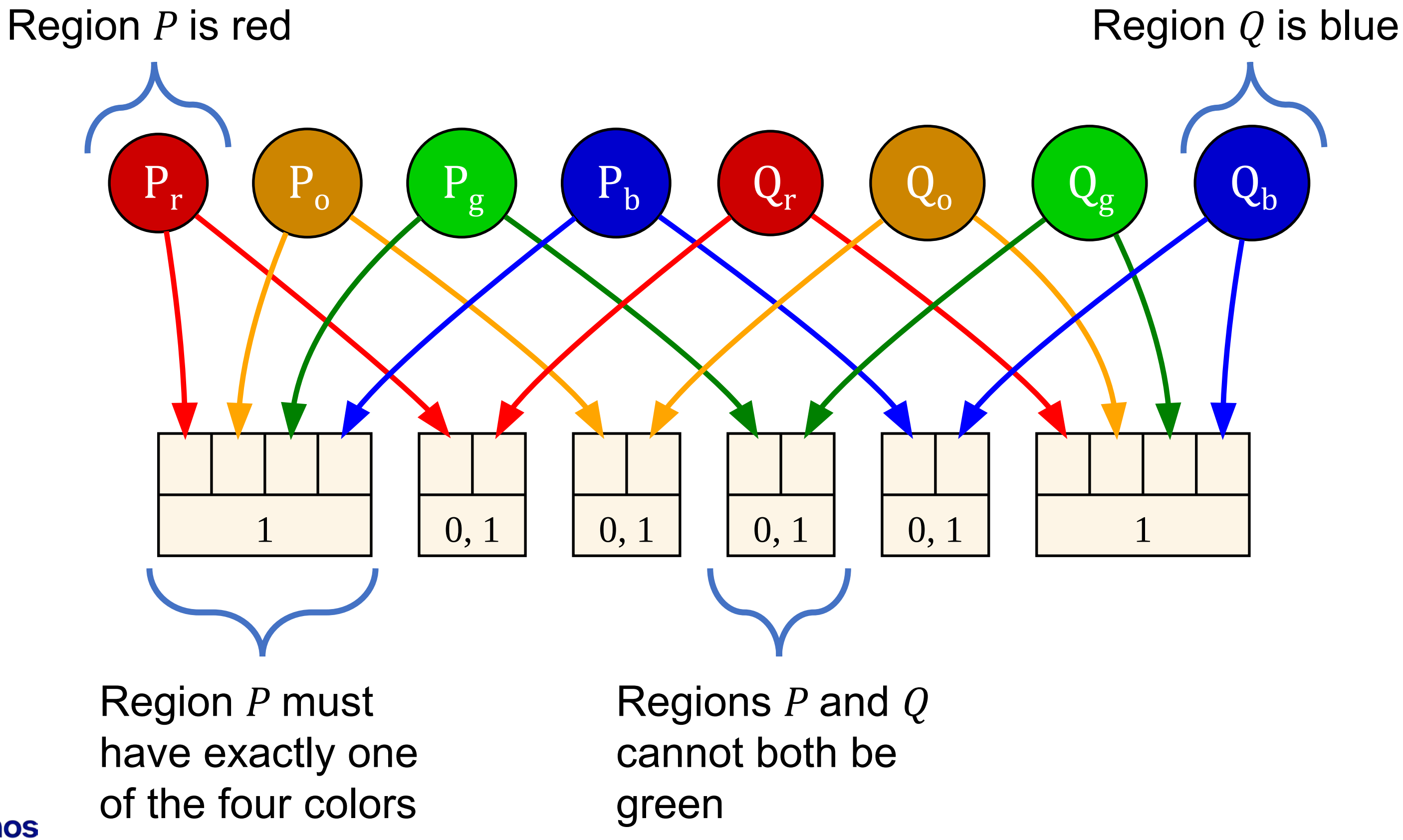
Shared Variables

- Variables that are shared within or across nck constraints will be assigned the same value
- Example of using a variable more than once within a constraint:
  - $nck(\{a, b, y, y\}, \{0,1,4\})$: "I want exactly 0, 1, or 4 of a, b, y, and y to be TRUE, and the two ys must have the same value (i.e., $y = a \wedge b$)"
    - Only FFFF, FTFF, TFFF, and TTTT honor 0, 1, or 4 TRUE and have the last two values equal



- Example of using a variable more than once across constraints:
  - $nck\left(\{P_r, P_o, P_g, P_b\}, \{1\}\right) \wedge nck(\{P_r, Q_r\}, \{0,1\}) \wedge nck(\{P_o, Q_o\}, \{0,1\}) \wedge nck\left(\{P_g, Q_g\}, \{0,1\}\right) \wedge nck(\{P_b, Q_b\}, \{0,1\}) \wedge nck\left(\{Q_r, Q_o, Q_g, Q_b\}, \{1\}\right)$
    - Two adjacent regions, P and Q, of a map four-coloring problem

Two Regions from a Map Four-Coloring Problem



Implementation Overview

NchooseK constraints

↓

QUBO

Given an upper-triangular $Q \in \mathbb{R}^{N \times N}$, find $\underset{x}{\arg\min}(x^T Q x)$ over all $x \in \{0,1\}^N$

(via QAOA)

Quantum annealers (using D-Wave's Ocean)

Circuit-model quantum computers (using IBM's Qiskit)

Z3

Classical computers (using Microsoft's Z3 SMT solver)

Why High-Performance Computing is Required

- Convert constraint → truth table → QUBO
  - Use a constraint-programming solver to find the QUBO matrix
- May need to augment the truth table with additional columns
  - Want to add as few columns as possible because columns cost qubits
  - Minimum number of additional columns is unknown
  - Boolean values with which to populate the additional columns are unknown
- Challenge
  - There are an exponential number of ways to populate those additional columns
  - Specifically, $2^{c \cdot 2^{|N|}}$ possibilities for a constraint $nck(N, K)$ with c additional columns
  - Intractable with brute force

Why High-Performance Computing is Required (cont.)

- Very tail-heavy distribution of execution times for converting constraints to optimal QUBOs, even when using a sophisticated CP solver
  - In most cases, QUBO generation is fast (a fraction of a second)
  - In a few cases, QUBO generation takes seconds or minutes
  - In rare cases, QUBO generation takes many, many hours
- Solution (rather, workaround): Use an HPC system to precompute many QUBOs in parallel
  - Store results in a database for future use
  - For the rare, super-slow cases, have each core work on the same problem but with a different ordering of the search space

Why High-Performance Computing is Required (cont.)

- Example: Convert $nck([A,B,C,D,E,F,G,H],\{0,3,4,5,6,7\})$ to a QUBO
- After 25 hours (!) running on 300+ cores, the following solution was found:

$$
\begin{array}{c}
\quad\quad A \quad\; B \quad\; C \quad\; D \quad\; E \quad\; F \quad\; G \quad\; H \quad\; \alpha \quad\; \beta \quad\; \gamma \\
Q = \begin{pmatrix}
-9 & 15 & 4 & 4 & 4 & 3 & 4 & 4 & -17 & 11 & -22 \\
0 & -14 & 15 & 15 & 15 & 11 & 15 & 15 & -63 & 43 & -83 \\
0 & 0 & -9 & 4 & 4 & 3 & 4 & 4 & -17 & 11 & -22 \\
0 & 0 & 0 & -9 & 4 & 3 & 4 & 4 & -17 & 11 & -22 \\
0 & 0 & 0 & 0 & -9 & 3 & 4 & 4 & -17 & 11 & -22 \\
0 & 0 & 0 & 0 & 0 & -7 & 3 & 3 & -13 & 8 & -16 \\
0 & 0 & 0 & 0 & 0 & 0 & -9 & 4 & -17 & 11 & -22 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -9 & -17 & 11 & -22 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 83 & -47 & 93 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -15 & -62 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 122
\end{pmatrix}
\end{array}
$$

Circuit Synthesis

- Given a unitary matrix, generate a high-quality quantum circuit
  - E.g., short circuit depth
- Given a quantum circuit, generate a superior quantum circuit that performs the same function
- Extremely time-consuming task
- Use HPC resources to accelerate the search for quality circuits

BQSKit

- QSearch: Optimal depth synthesis up to four qubits
- LEAP: Best quality of solution synthesis up to six qubits
- QFAST: Scales good solution quality synthesis up to eight qubits
- QGO: Optimizing compiler combining partitioning and synthesis
- QUEST: Scalable circuit approximations
- QFactor: Fastest quantum circuit optimizer using tensor networks

Debugging Quantum Applications

- Not possible to stop quantum execution, inspect and manipulate state, and resume execution
  - Measurement collapses the wave function
- Instead, co-opt quantum simulators for use as quantum debuggers
  - Add support for setting breakpoints and watchpoints, single-stepping, querying/modifying the state vector, and other features helpful for understanding
- As qubit counts increase, time and/or memory requirements increase exponentially
- HPC-based quantum simulators can enable debugging of larger quantum systems than desktop-based quantum simulators

Summary

- Quantum computing and high-performance computing are mutually beneficial
- Symbiosis is not limited to using quantum computing as an accelerator for certain subroutines in an HPC program
- Examples considered
  - Quantum optimization to suggest the most fruitful HPC-based simulation to run next
  - Quantum optimization for improving parallel task placement in an HPC application
  - Using HPC resources to precompute costly code transformations for use in compilation of code for quantum computers
  - Using HPC resources to generate highly optimized quantum circuits
  - Simulating quantum circuits on an HPC system to debug quantum applications
- As high-performance computers and quantum computers become increasingly integrated, more opportunities for joint usage will assuredly arise