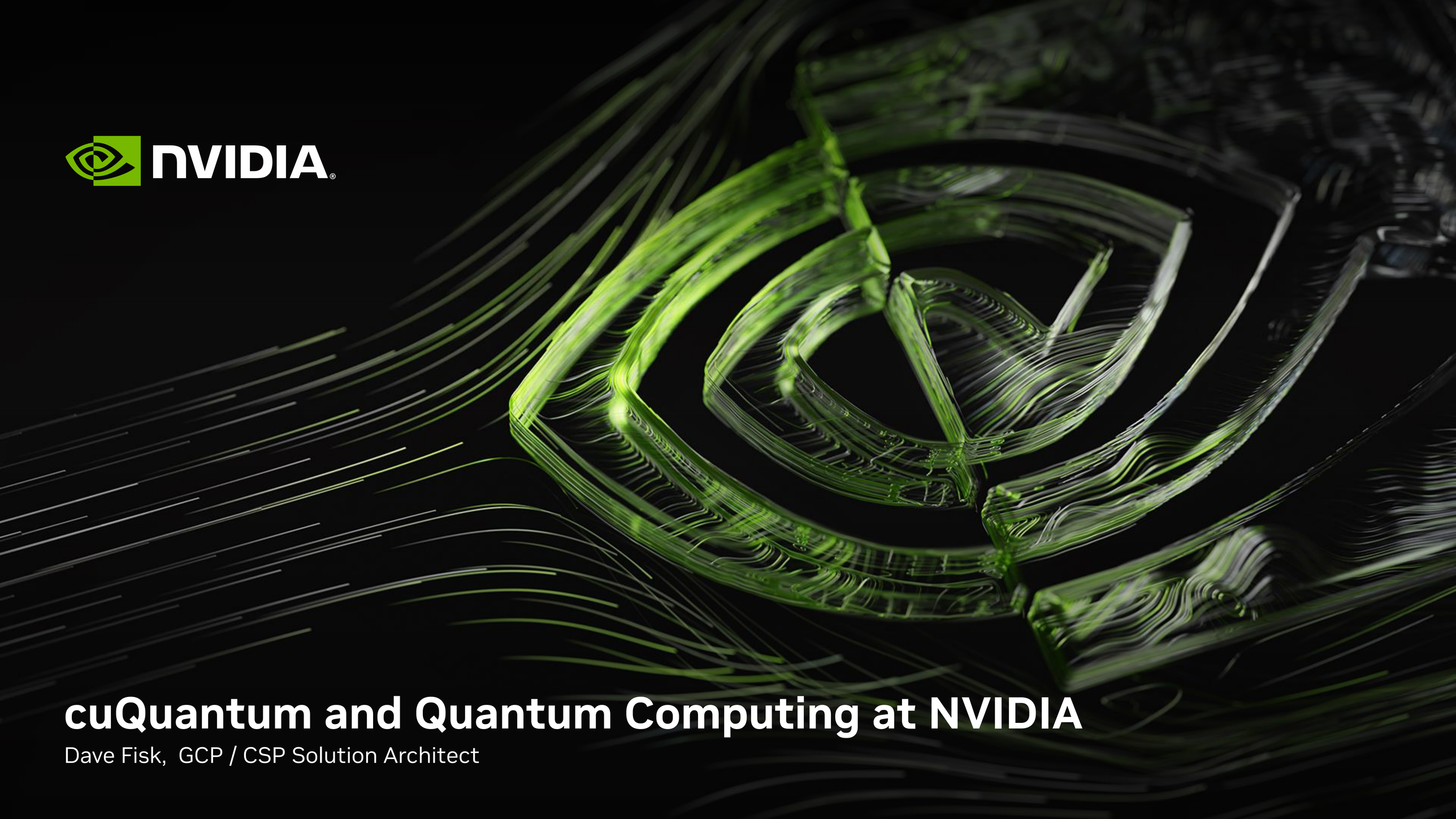


cuQuantum and Quantum Computing at NVIDIA

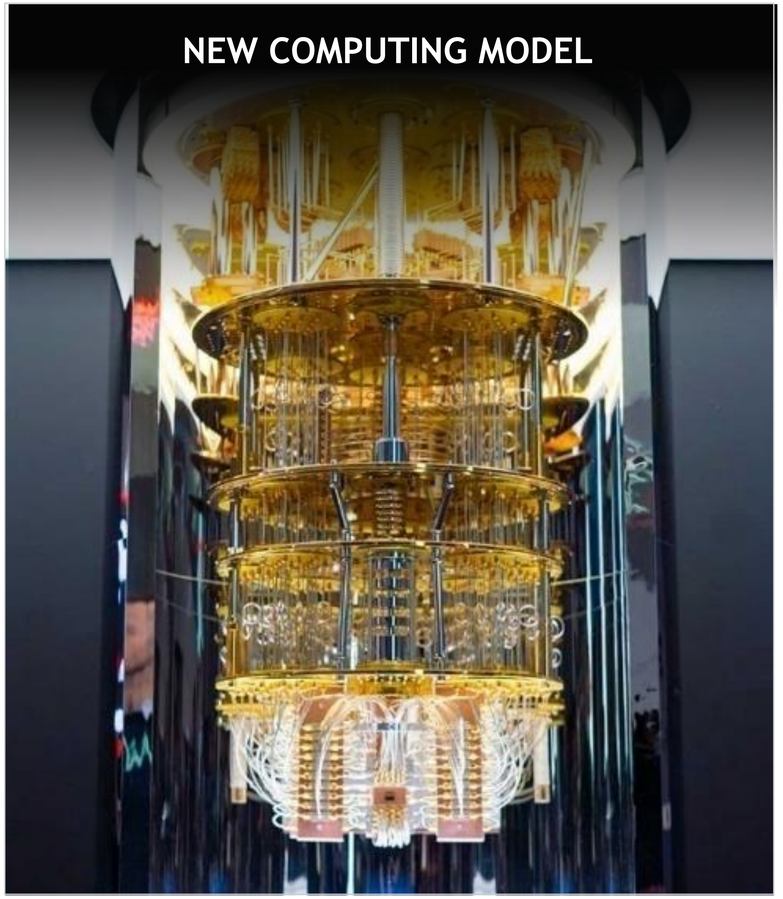
Dave Fisk, GCP / CSP Solution Architect

Course Agenda

- Introductory presentation
- Lab 1
- Lab 2
- cuTensorNet | Lab 3

Quantum Computing Overview

A New Computing Model – Quantum Computing



POTENTIAL USE CASES

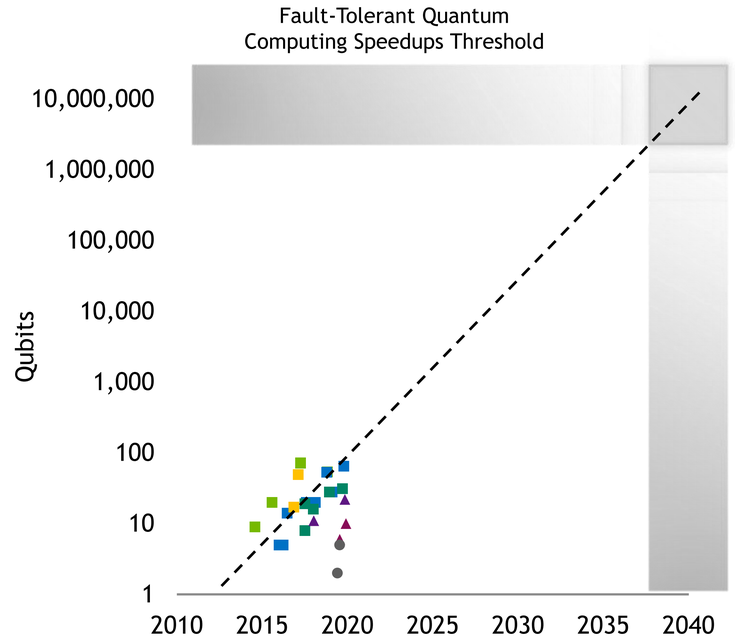
Computational Finance

Quantum Chemistry

Cryptography

Optimization

REQUIRES QUBITS SCALE TO DOUBLE EVERY YEAR

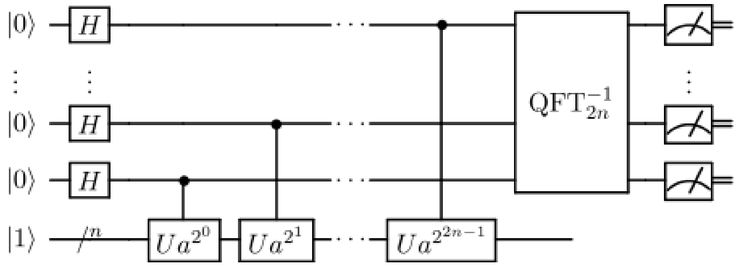


Far Term Applications

Rigorous proofs of advantage, many “perfect” qubits required

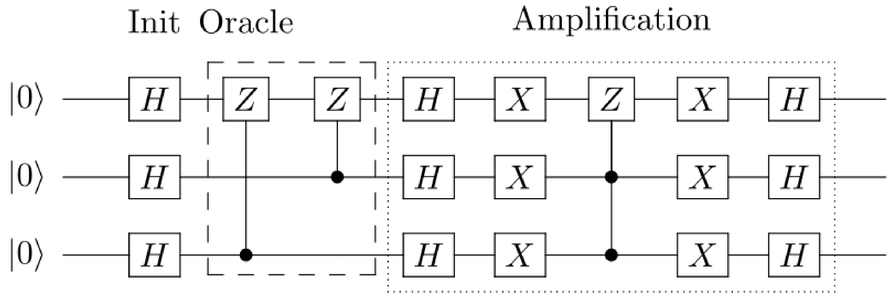
SHOR’S ALGORITHM

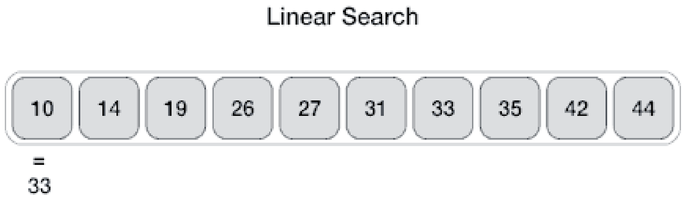
- Prime factorization of numbers - encryption
- Exponential speed-up



GROVER’S ALGORITHM

- Unstructured search
- Quadratic speed-up



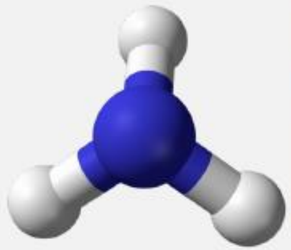


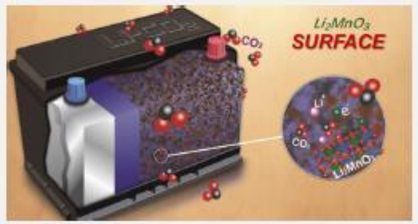
Near Term Application Potential

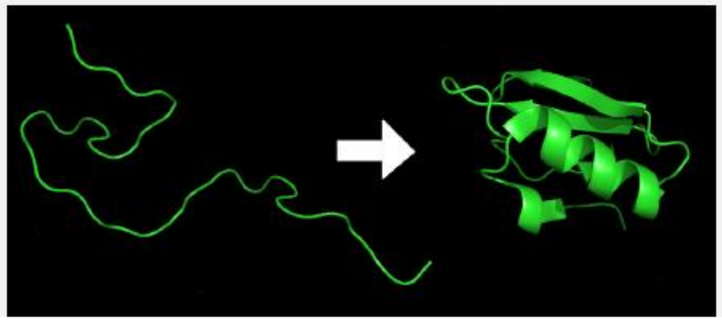
Applications with near term potential but quantum advantage is an open question

CHEMISTRY, MATERIALS SCIENCE, DRUG DISCOVERY

- Ground state energy calculations
- Protein folding
- Variational Quantum Eigensolver

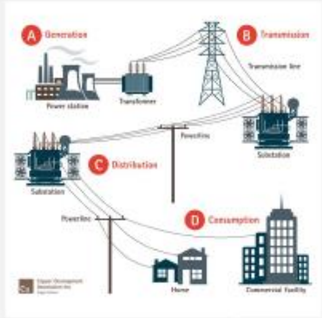






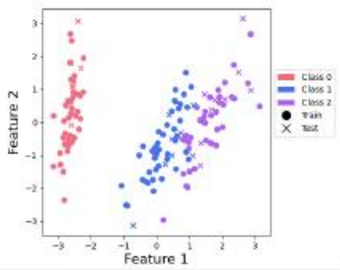
FINANCE, LOGISTICS, OPTIMIZATIONS

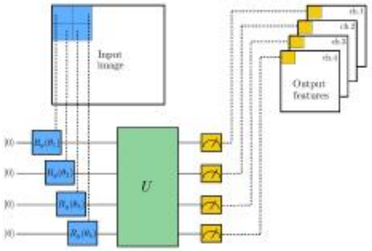
- Portfolio asset optimization
- Energy grid optimization
- Supply chain optimization

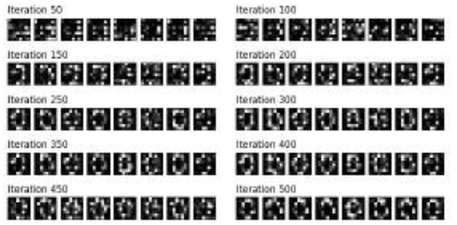


MACHINE LEARNING

- Classification problems
- Sampling problems







Source of images: PennyLane, Wikipedia, Lawrence Berkeley Lab, American Physical Society, Copper Development Association

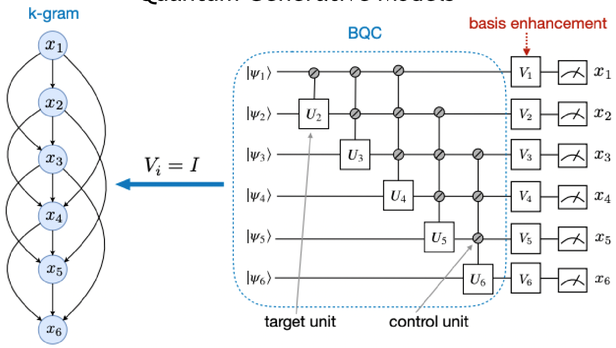
NVIDIA

Potential Near Term Quantum Computing Use-Cases

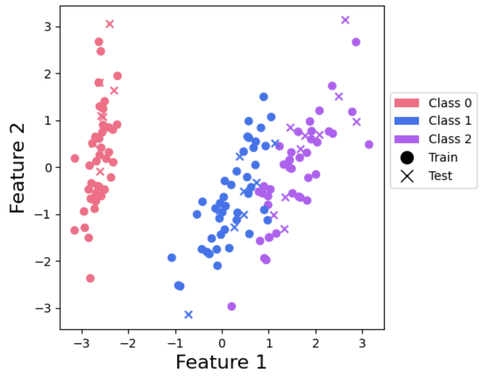
Applications with near term potential but quantum advantage is an open question

Quantum Machine Learning

Quantum Generative Models



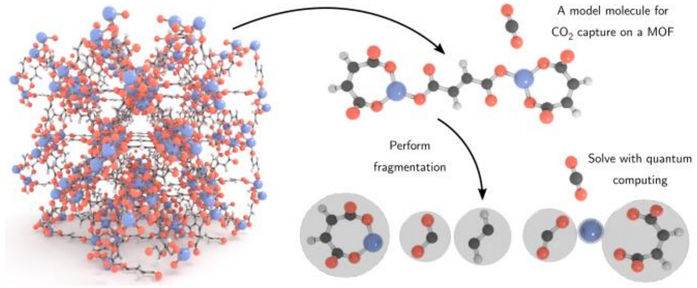
Quantum Support Vector Machine



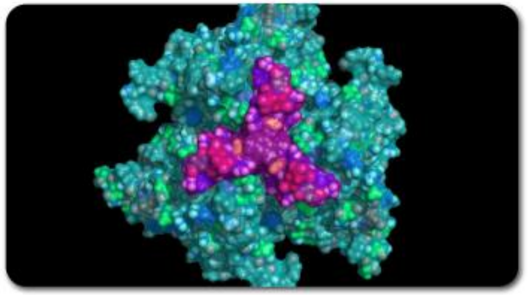
Gao, et al, Phys. Rev. X **12**, 021037
Pennylane.ai

Quantum Chemistry

Variational Quantum Eigensolver for carbon capture



Protein folding



Greene-Diniz, et al, arXiv:2203.15546,
Menten.ai

Combinatorial Optimization

QAOA for resource allocation

Logistics optimization

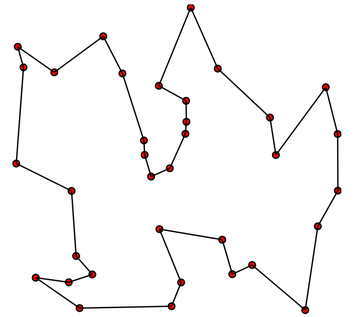
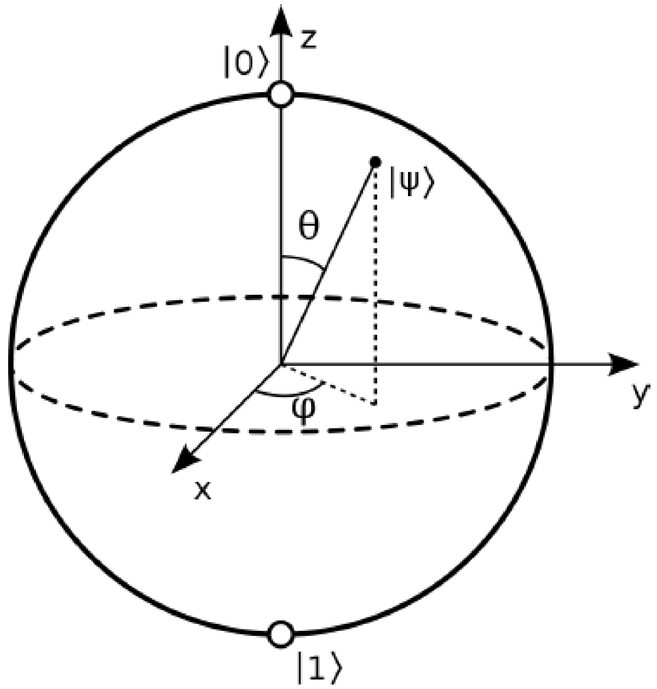


Image from ibm.com
Wikipedia.com

Quantum Computing Basic Operations

Superposition and Measurement

Bit



Qubit
(Bloch Sphere)

$$|\Psi\rangle = a|0\rangle + b|1\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$$

Measurement: wavefunction collapse
- measure only one state

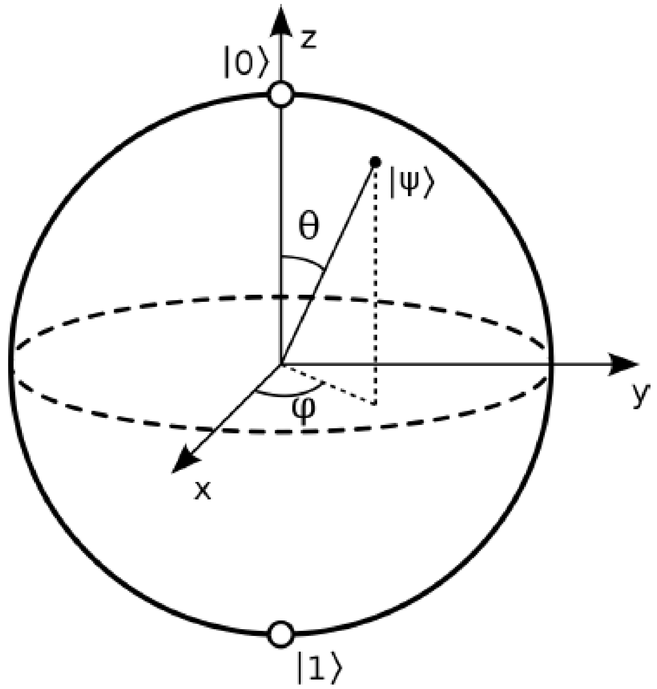
$P_0 = |a|^2$

$P_1 = |b|^2$

Quantum Computing Basic Operations

Superposition and Measurement

Bit



Qubit
(Bloch Sphere)

0

0	1

1	0	1

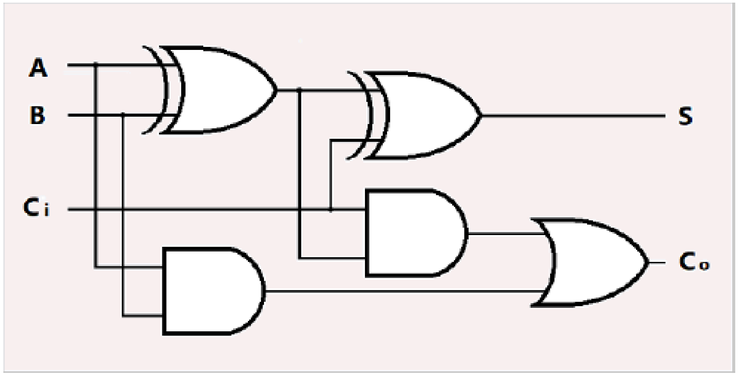
$$c_0|0\rangle + c_1|1\rangle = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$$

$$c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle = \begin{bmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{bmatrix}$$

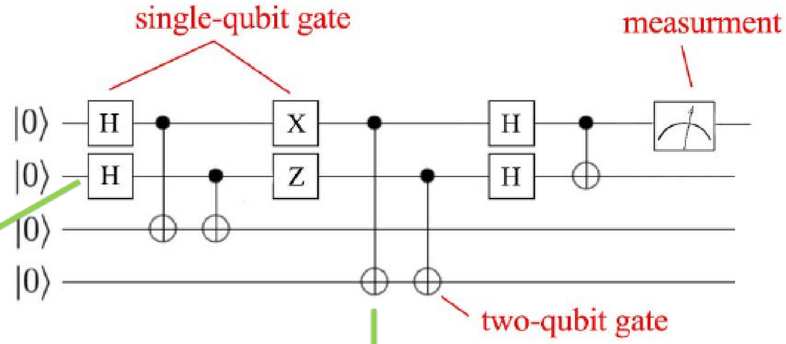
$$c_{000}|000\rangle + c_{001}|001\rangle + c_{010}|010\rangle + c_{011}|011\rangle + c_{100}|100\rangle + c_{101}|101\rangle + c_{110}|110\rangle + c_{111}|111\rangle = \begin{bmatrix} c_{000} \\ c_{001} \\ c_{010} \\ c_{011} \\ c_{100} \\ c_{101} \\ c_{110} \\ c_{111} \end{bmatrix}$$

Quantum Circuits

Classical Circuit



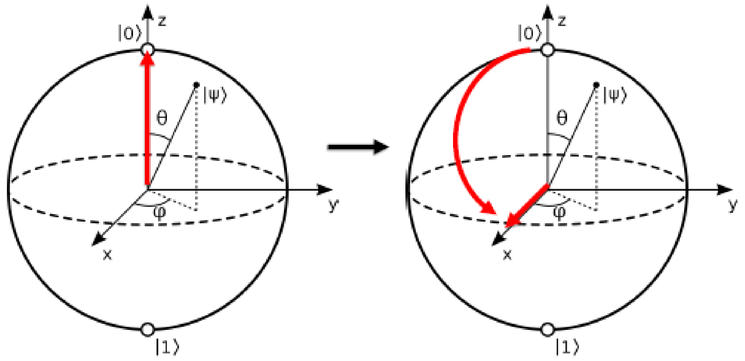
Quantum Circuit



Hadamard Gate:

$Had|0\rangle = |0\rangle + |1\rangle$

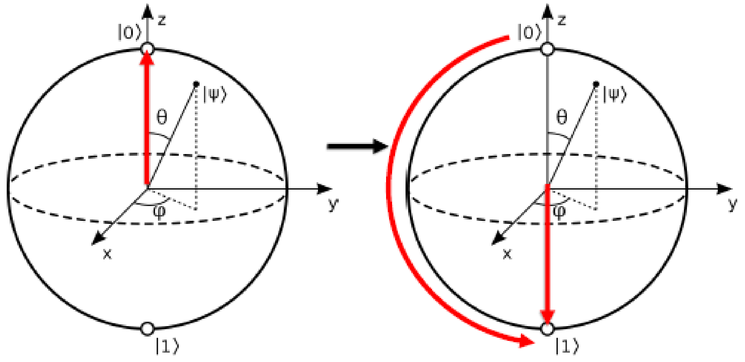
$Had|1\rangle = |0\rangle - |1\rangle$



CNOT Gate:

$CNOT|10\rangle = |11\rangle$

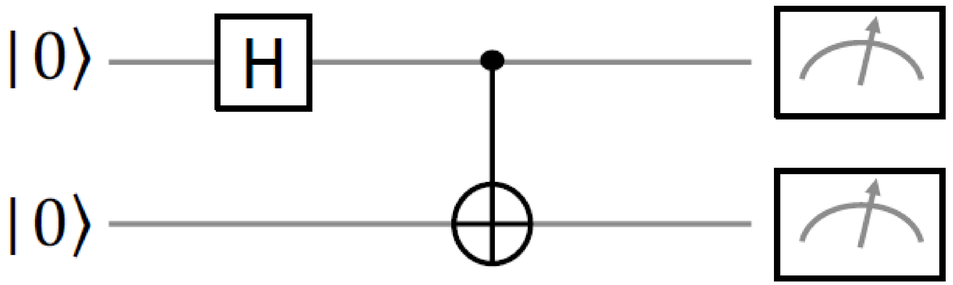
$CNOT|11\rangle = |10\rangle$



Quantum Entanglement

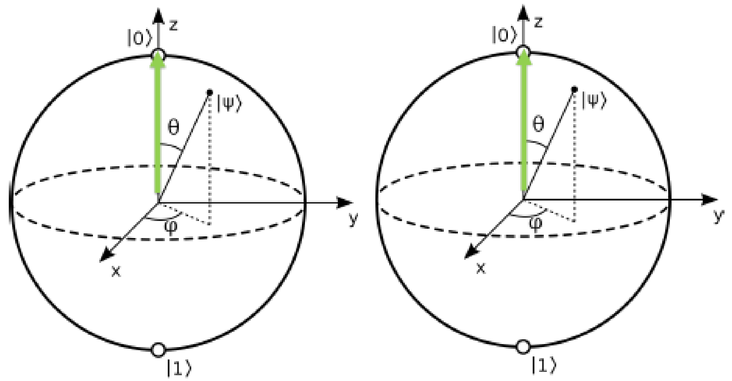
Hadamard Gate:
Had|0⟩ = |0⟩ + |1⟩
Had|1⟩ = |0⟩ - |1⟩

CNOT Gate:
CNOT|10⟩ = |11⟩
CNOT|11⟩ = |10⟩

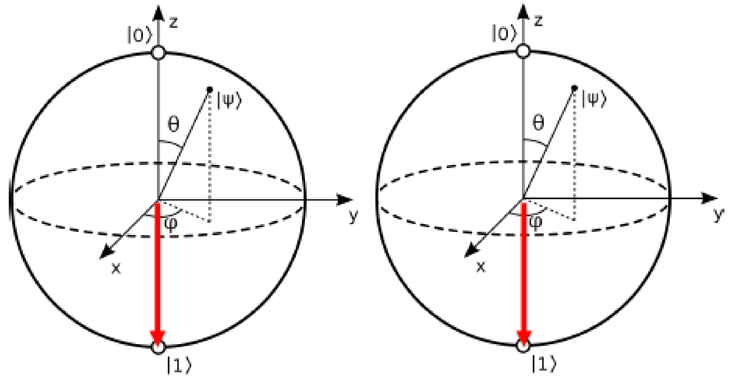


$$|00\rangle \rightarrow |00\rangle + |11\rangle$$

$|00\rangle$



$|11\rangle$



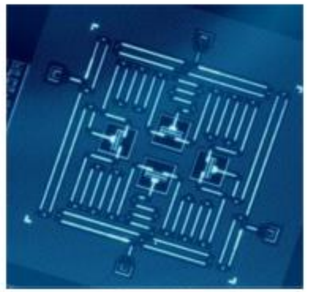
Leading Qubit Technologies

The challenge of engineering quantum hardware is to manipulate physical systems to implement superposition and entanglement (for a sufficiently long time)

SUPERCONDUCTORS

- **Principle:** Superconducting circuits based on Josephson junctions
- **Strengths:** Gate error rates <1%
- **Weaknesses:** Qubits only hold state ~100μs, fixed connectivity, cross-talk

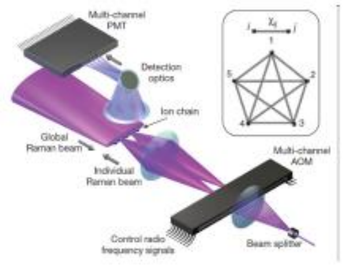
aws Google IBM intel QuTech rigetti



ION TRAPS

- **Principle:** Ions in a vacuum, trapped & rotated by lasers
- **Strengths:** Long coherence time, all-to-all connectivity
- **Weaknesses:** Scalability, slow read-out

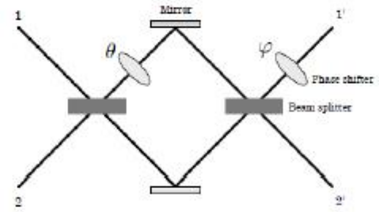
QUANTINUUM IONQ



SILICON PHOTONICS

- **Principle:** Store qubits as polarity of single photons, photonics for gates
- **Strengths:** Scalability, manufacturable
- **Weaknesses:** Photon sources/detectors, error rates, non-std computation model

PsiQuantum XANADU

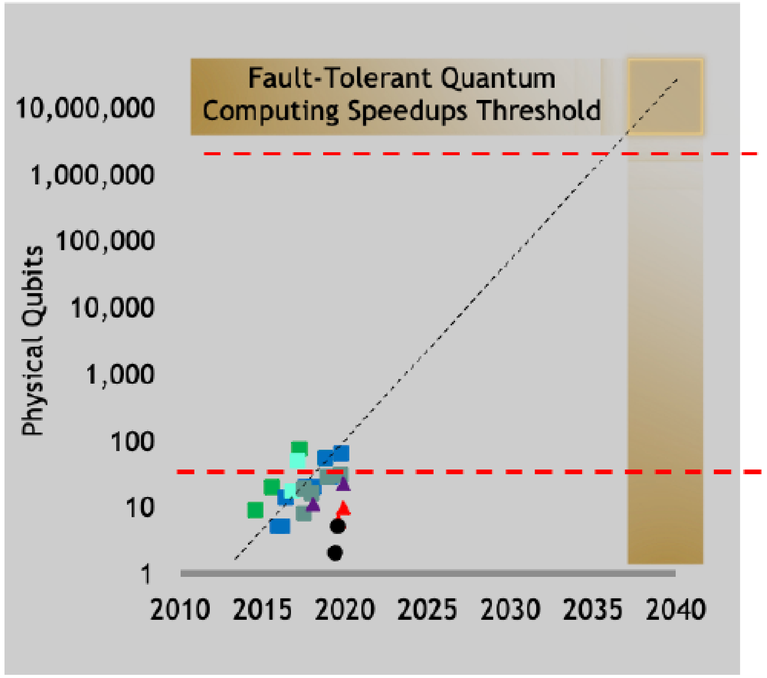


Other approaches: Neutral Atoms, Quantum Dots, Topological Qubit, Diamond Vacancies

Practical QC is expected to require scaling these technologies to millions of qubits, error correction and new quantum algorithm

Quantum Computing Research Roadmap

Large improvements in qubit quantity & quality, error correction, needed for wide adoption



Fault-Tolerant QC Era:

1000:1-10000:1 redundancy for error-corrected *logical* qubits.
[Fowler 2012][Reiher 2016]

Exponential speedups on a limited set of applications with hundreds to thousands of logical qubits (millions of physical qubits).

Active Research: What are the best error correction algorithms?

Noisy Intermediate Scale Quantum (NISQ) Era:

Quantum gates are noisy, errors accumulate. Qubits lose coherence.

QC hardware will mitigate errors by using tens to hundreds of redundant physical qubits per logical qubit to mitigate errors.

Active Research: Will NISQs have quantum advantage on useful workloads?

Quantum Supremacy Threshold: Experimental confirmation of quantum speedup on a well-defined (not necessarily *useful*) problem.

Qubits and quantum gates are very noisy, hardware not very usable.

Active Research: Can this be simulated efficiently on GPU supercomputers?

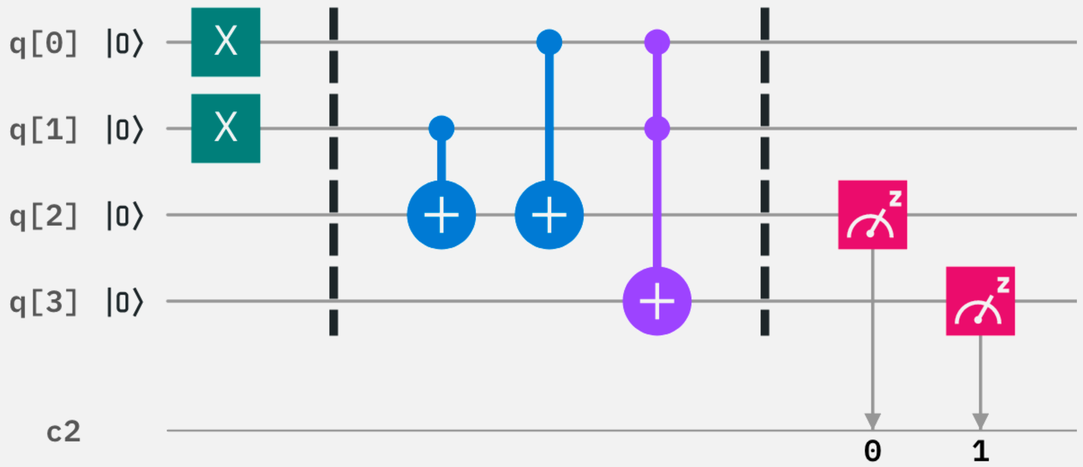
Quantum Computing Simulation

GPU-based Supercomputing in the Quantum Computing Ecosystem

Researching the quantum computer of tomorrow with the supercomputers of today

QUANTUM CIRCUIT SIMULATION

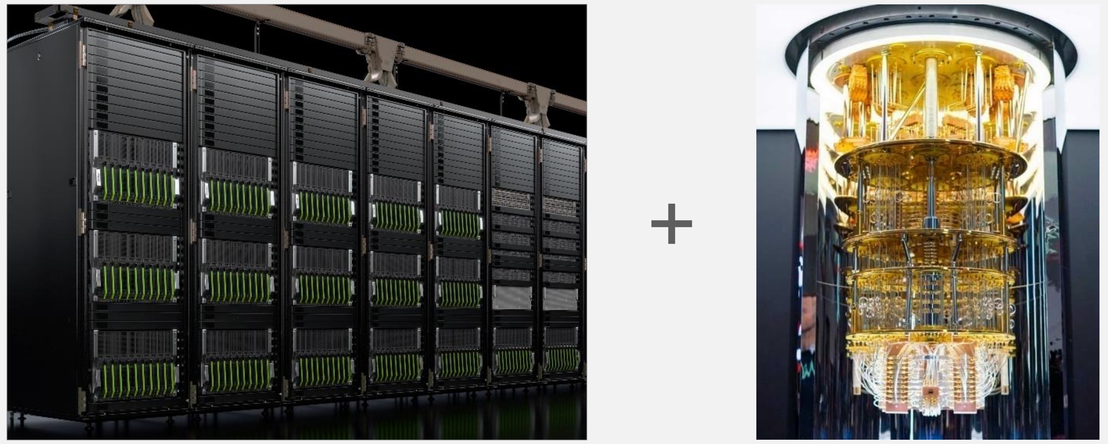
Critical tool for answering today's most pressing questions in Quantum Information Science (QIS):



- What quantum algorithms are most promising for near-term or long-term quantum advantage?
- What are the requirements (number of qubits and error rates) to realize quantum advantage?
- What quantum processor architectures are best suited to realize valuable quantum applications?

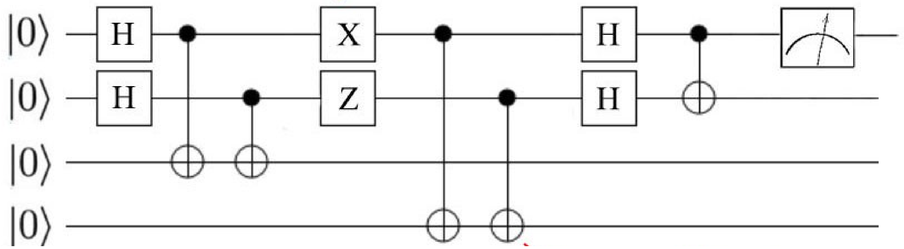
HYBRID CLASSICAL/QUANTUM APPLICATIONS

Impactful QC applications (e.g. simulating quantum materials and systems) will require classical supercomputers with quantum co-processors



- How can we integrate and take advantage of classical HPC to accelerate hybrid classical/quantum workloads?
- How can we allow domain scientists to easily test coprogramming of QPUs with classical HPC systems?
- Can we take advantage of GPU acceleration for circuit synthesis, classical optimization, and error correction decoding?

Two Leading Quantum Circuit Simulation Approaches



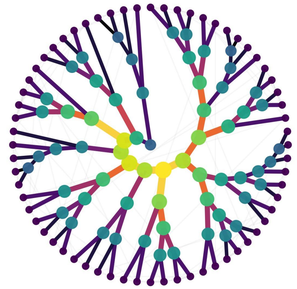
State vector simulation

“Gate-based emulation of a quantum computer”

- Maintain full 2^n qubit vector state in memory
- Update all states every timestep, probabilistically sample n of the states for measurement

Memory capacity & time grow exponentially w/ # of qubits - practical limit around 50 qubits on a supercomputer

Can model either ideal or noisy qubits



Tensor networks

“Only simulate the states you need”

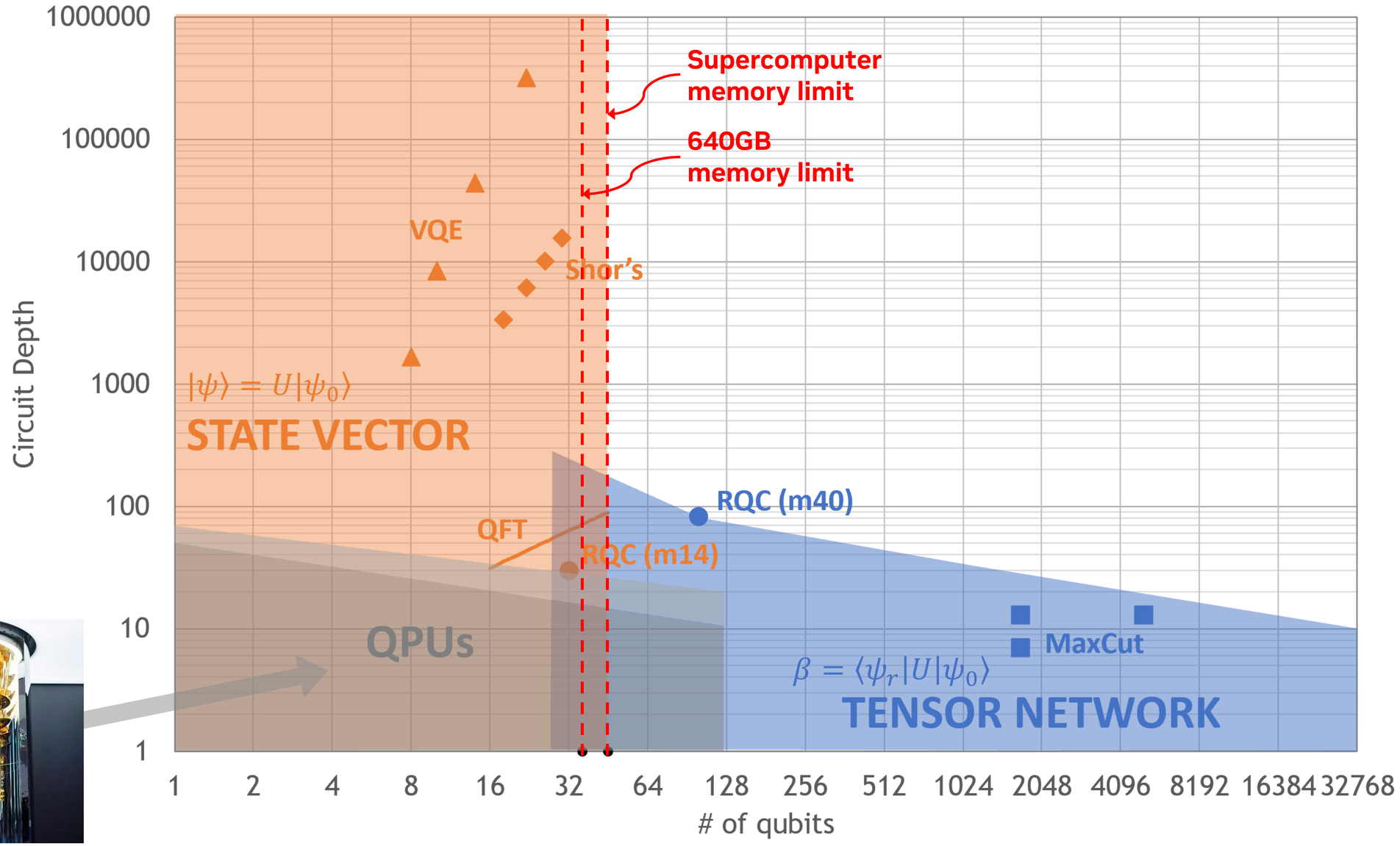
- Uses tensor network contractions to dramatically reduce memory for simulating circuits
- Can simulate 100s or 1000s of qubits for many practical quantum circuits

GPUs are a great fit for either approach

Source of Tensor Network image: Quimb - quimb.readthedocs.io

State Vector vs Tensor Network for Quantum Circuit Simulation

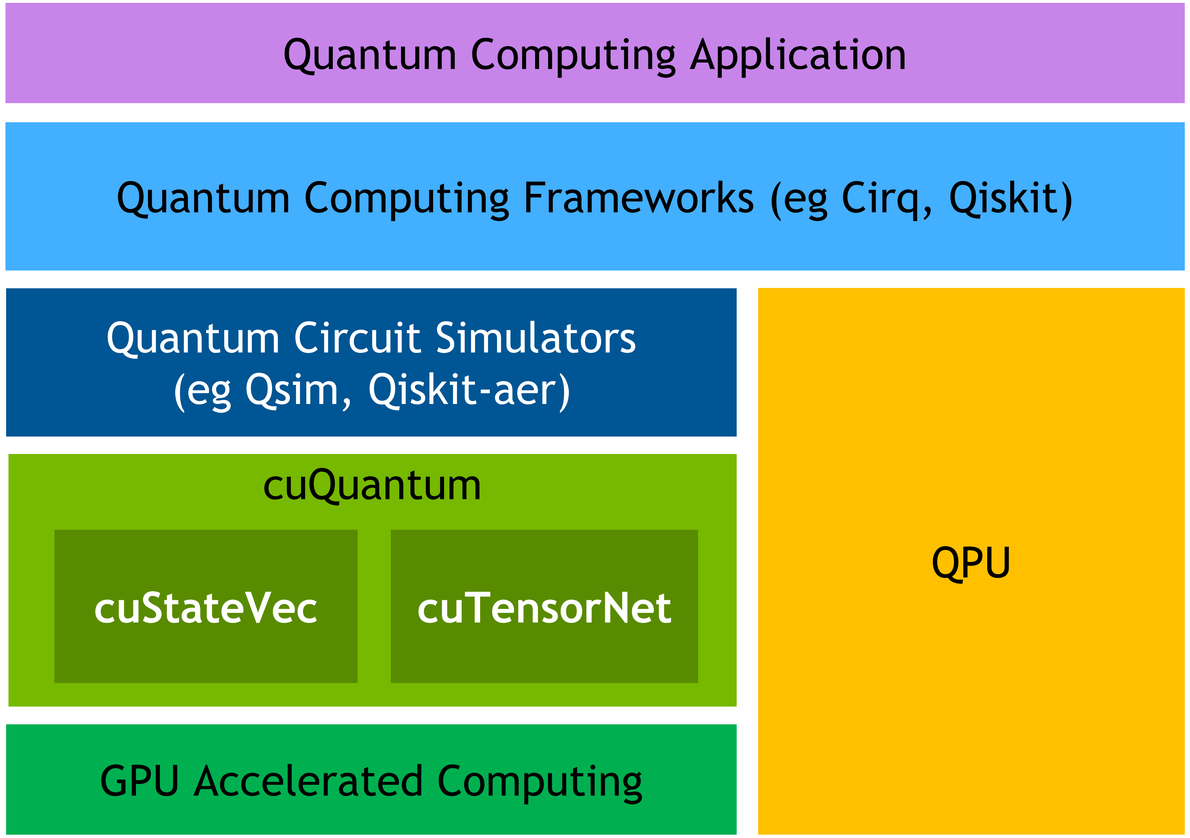
R&D for the computers of tomorrow requires powerful simulations today



NVIDIA

Introducing cuQuantum

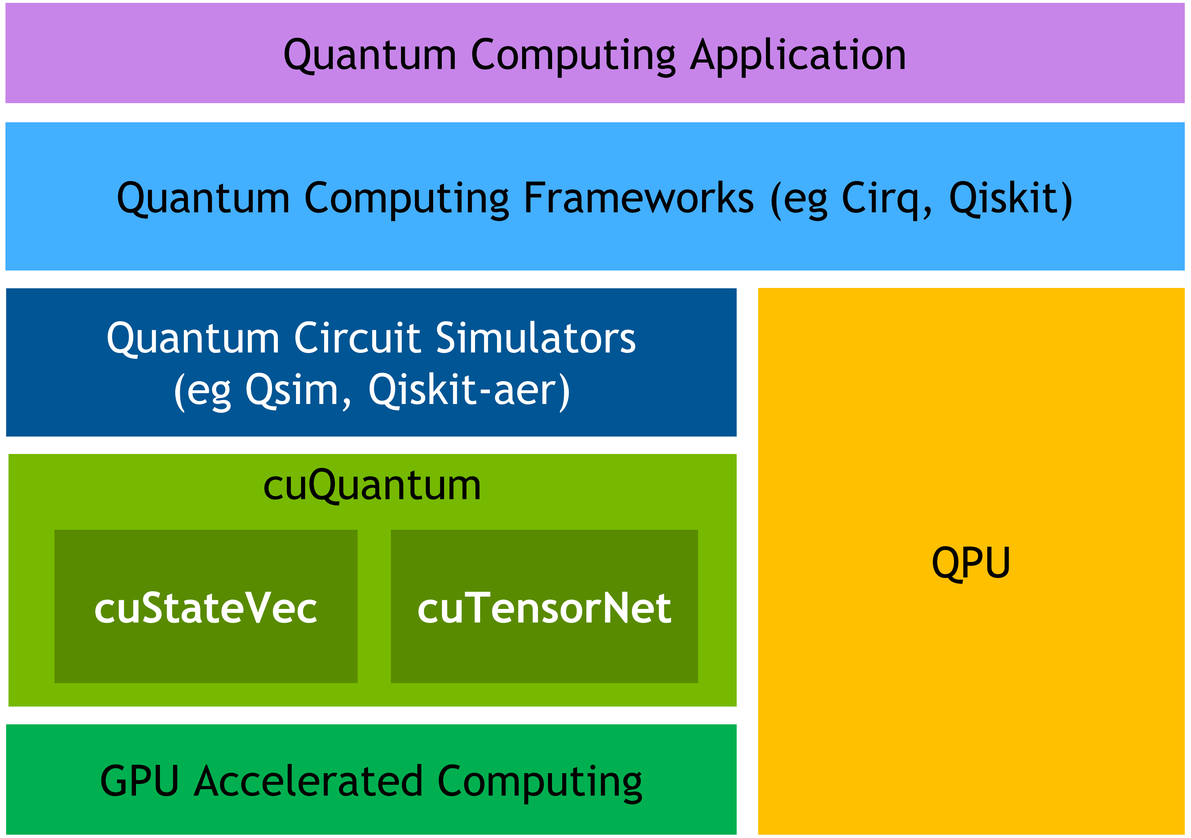
- cuQuantum is an SDK of **optimized libraries and tools** for accelerating Quantum Computing workflows
- cuQuantum is **not** a:
  - Quantum Computer
  - Quantum Computing Framework
  - Quantum Circuit Simulator



Introducing cuQuantum

- cuQuantum is a platform for Quantum Computing research
  - 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
- GA availability, integrated with
  - Google Cirq
  - IBM Qiskit
  - Xanadu PennyLane
- DGX Quantum Appliance container available on NGC: catalog.ngc.nvidia.com/orgs/nvidia/containers/cuquantum-appliance
- Full documentation at docs.nvidia.com/cuda/cuquantum

Cirq Qiskit PENNYLANE



cuQuantum Ecosystem

Frameworks

aws Amazon Braket

Cirq

Qiskit

PENNYLANE

Orquestra®

QIBO

XACC
QUANTUM FRAMEWORK

HPC Centers

Argonne NATIONAL LABORATORY

JÜLICH Forschungszentrum

OAK RIDGE National Laboratory

NERSC

Other Power Users

IBM

Google Quantum AI

rigetti

QUANTINUUM

IONQ

XANADU

PsiQuantum

PASQAL

QUANTUM BRILLIANCE

CLASSIQ

QCWARE

ZAPATA

menten.AI

cuQuantum Performance

Enabling speedups for a range of use cases and users

Faster Quantum Algorithm for Physics-ML

100X
Faster Time-to-solution

24X
More Circuit Depth

New PennyLane Integration via AWS Braket

900X
Faster Time-to-solution

3.5X
Lower Costs

Orquestra Platform Integration

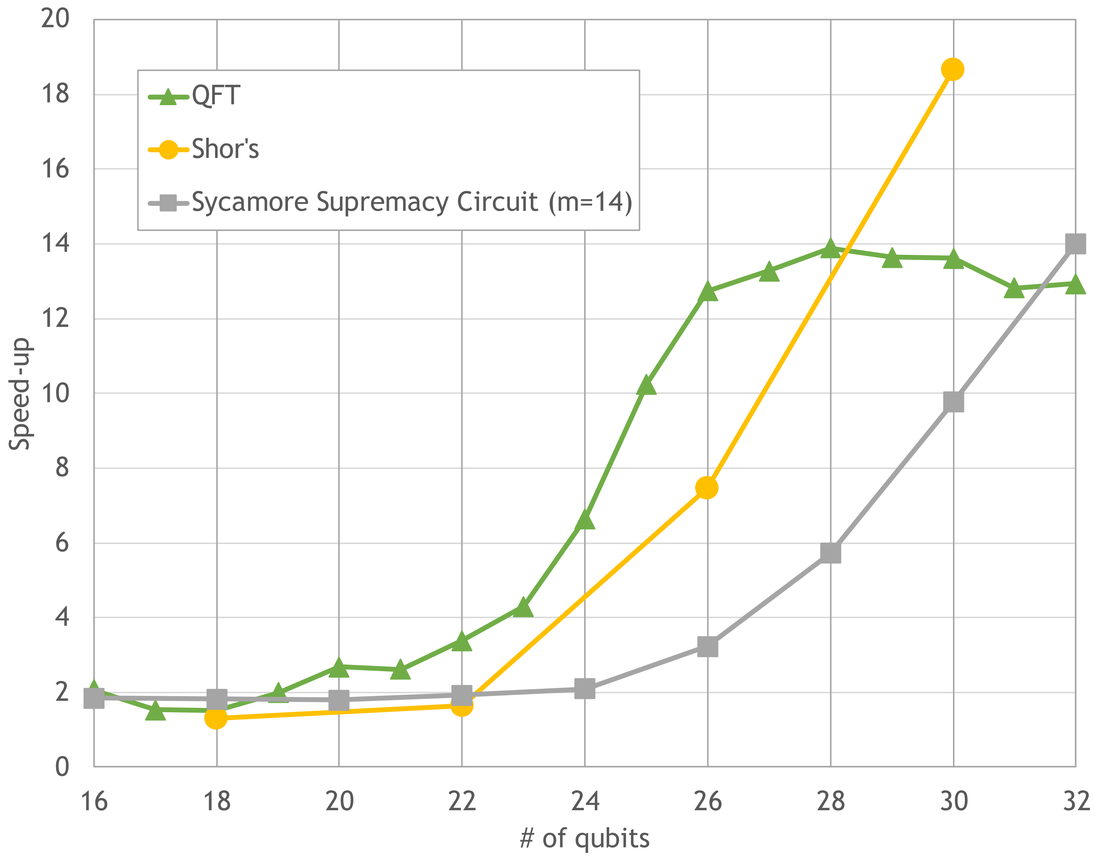
100X
Faster Time-to-solution

1.5X
More Qubits

cuStateVec – Single GPU Performance

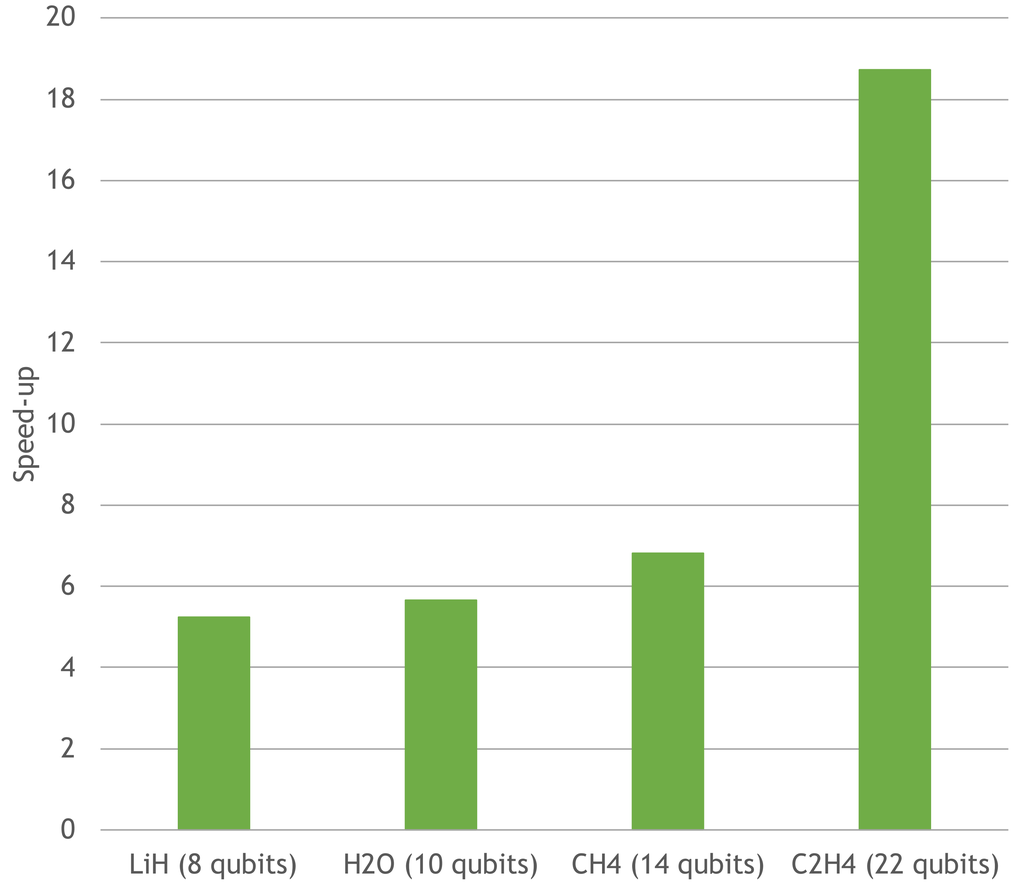
Preliminary performance of Cirq/Qsim + cuStateVec on NVIDIA A100

A100 80G vs 64 core CPU



Benchmarks run using cirq/qsim with modifications to integrate cuStateVec
CPUs used were AMD EPYC 7742 with 64 cores
QFT circuit with 32 qubits and depth 63
Shor's circuit with 30 qubit and depth 15560 (integer factorized: 65)
Sycamore supremacy circuit m=14 with 7480 gates

VQE speed-up relative to single CPU

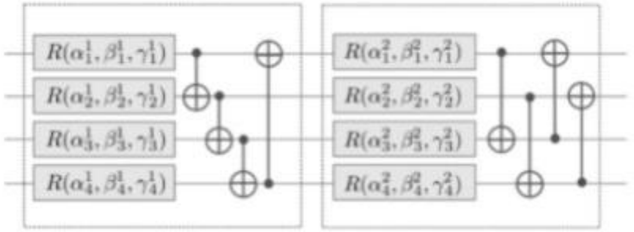


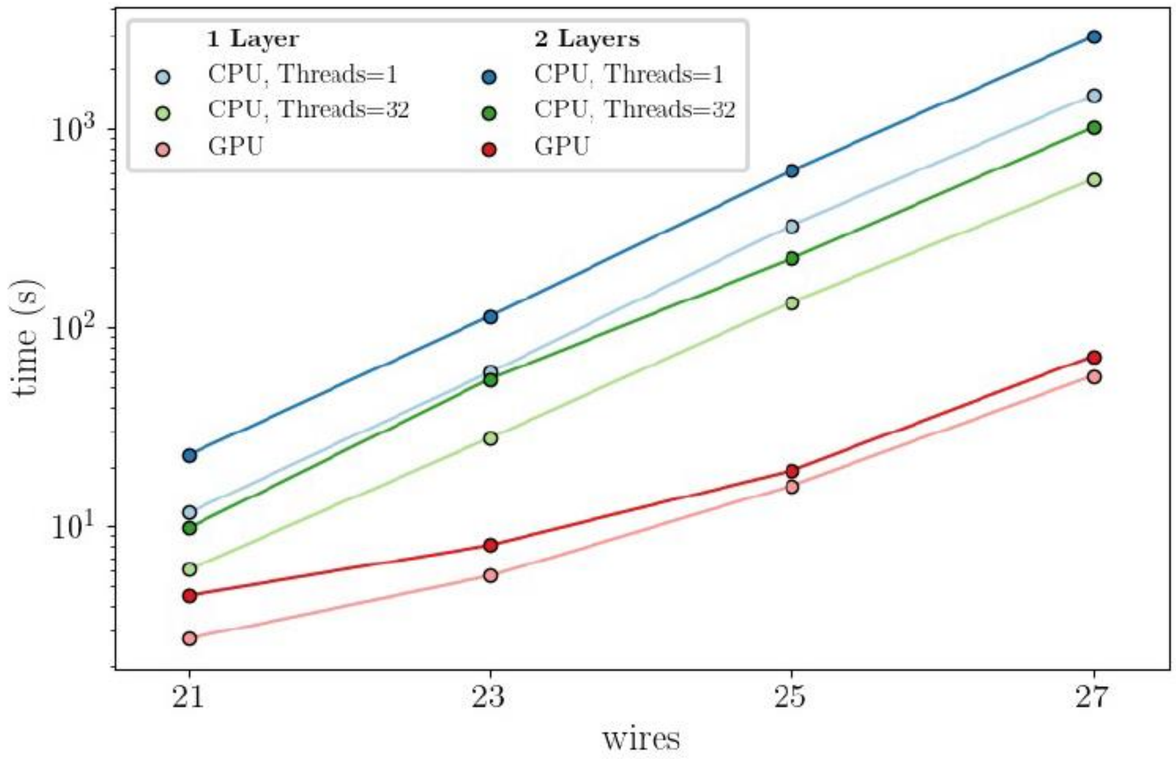
VQE benchmarks have all orbitals and results were measured for the energy function evaluation

cuQuantum Support for PennyLane

- Leading open-source framework for quantum machine learning and quantum chemistry, built by Xanadu
  - Train Quantum Computers in the same way as Neural Networks
- New simulator *lightning.gpu* with cuQuantum support, available now:
  - xanadu.ai/products/lightning
- 10x speedup for QML circuits

PENNYLANE

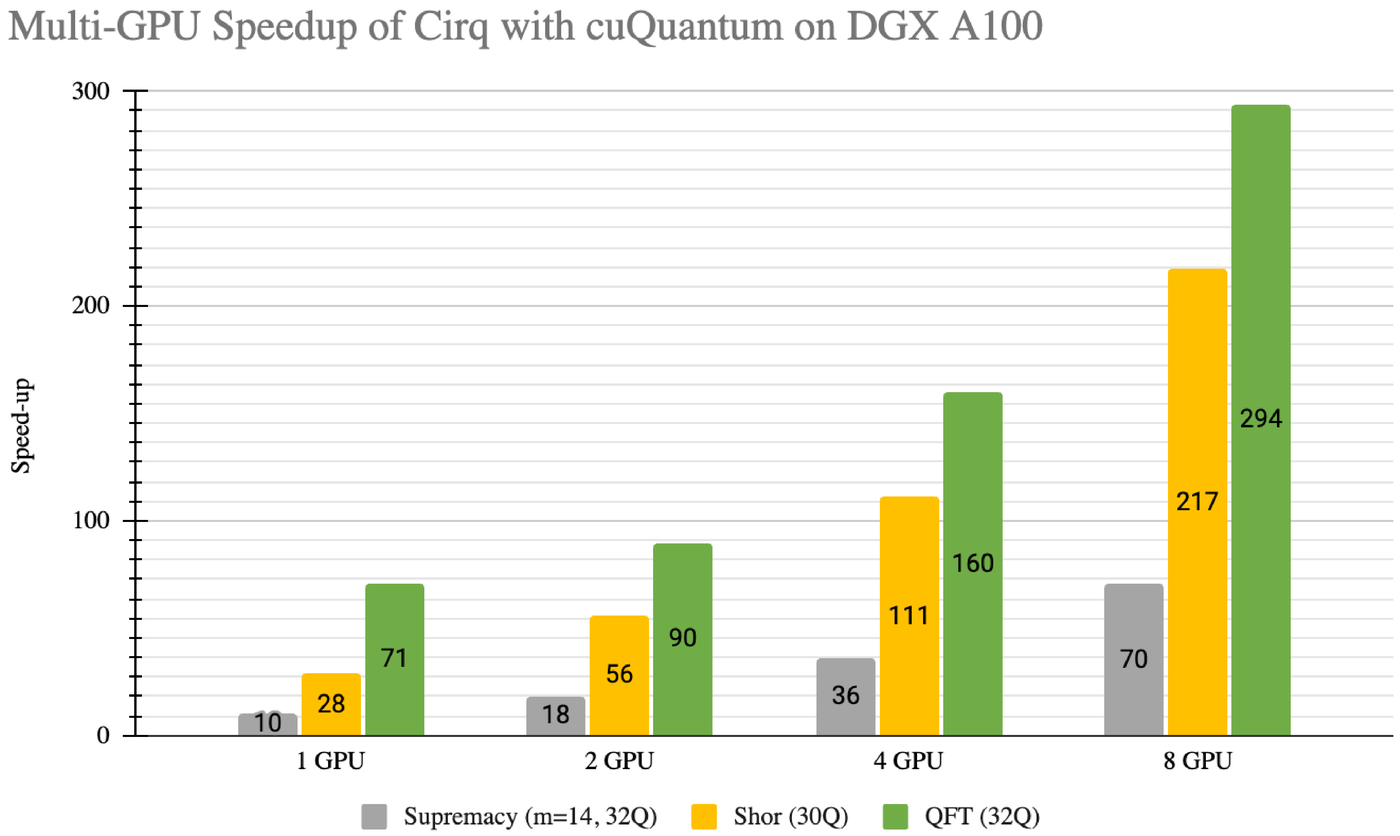




DGX cuQuantum Appliance

Multi-GPU container with cuQuantum + integrated Cirq/Qsim

- Full Quantum Simulation stack with a Cirq/Qsim frontend
  - other frontends will be available in future releases
- World class performance on key quantum algorithms
- Available now on NGC: catalog.ngc.nvidia.com/orgs/nvidia/containers/cuquantum-appliance

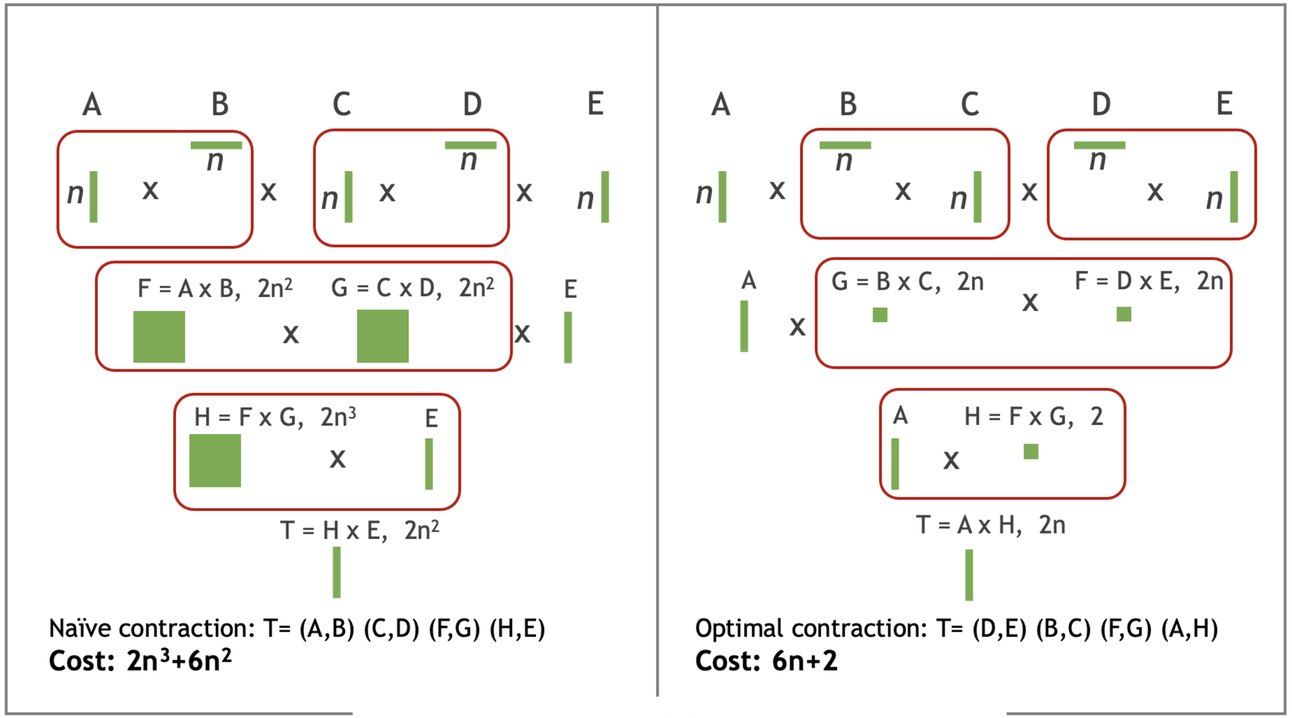


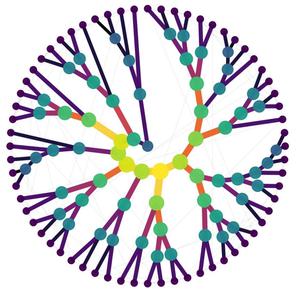
Tensor Networks & MaxCut

cuTensorNet

A library to accelerate Tensor Network based Quantum Circuit simulation

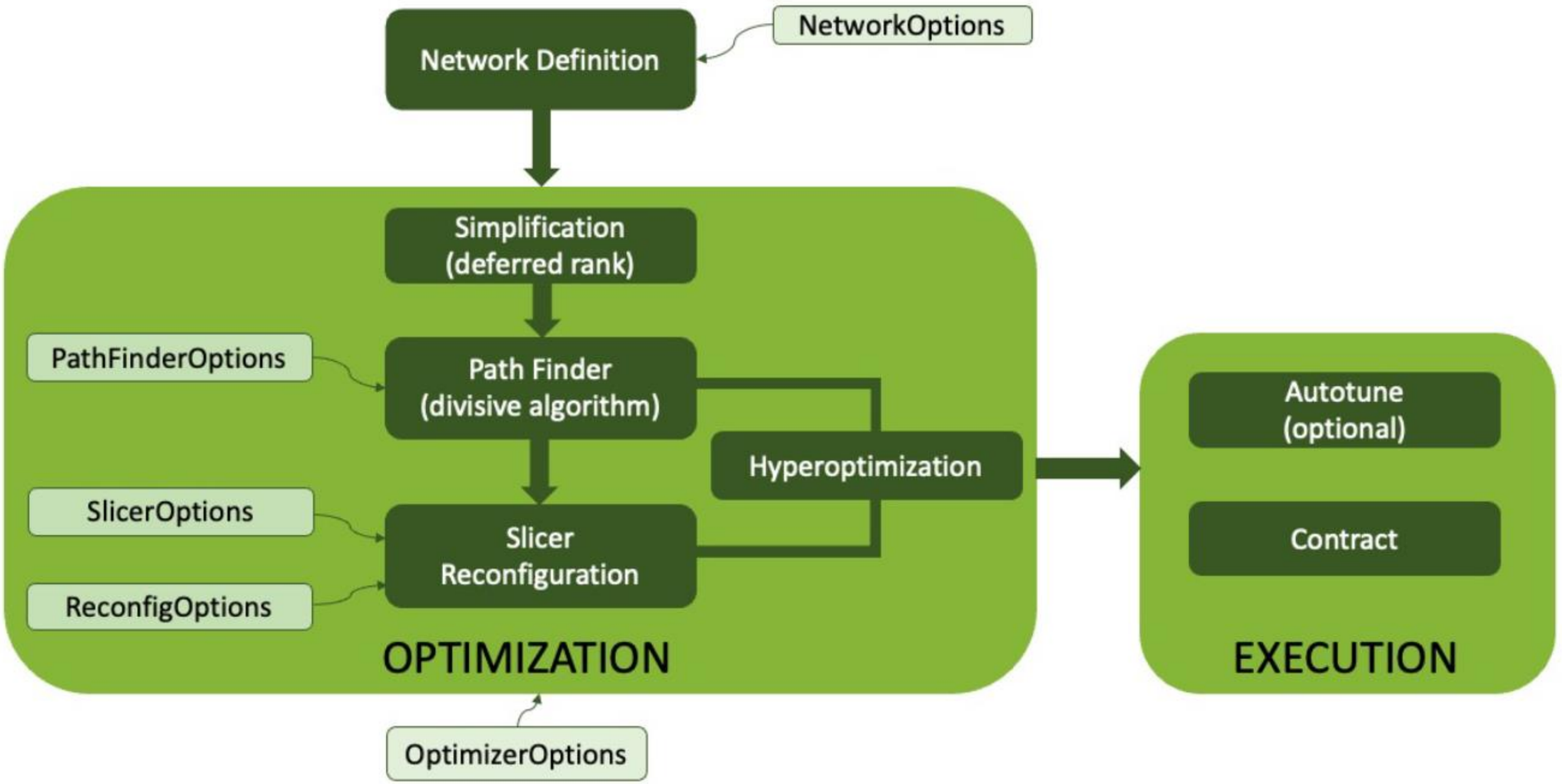
- For many practical quantum circuits, tensor networks enable scaling of simulation to 100s or 1000s of qubits
- cuTensorNet provides APIs to:
  - convert a circuit written in Cirq or Qiskit to a tensor network
  - calculate an optimal path for the contraction
    - hyper-optimization is used to find contraction path with lowest total cost (eg FLOPS or time estimate)
    - slicing is introduced to create parallelism or reduce maximum intermediate tensor sizes
  - calculate an execution plan and execute the TN contraction
    - leverages cuTENSOR heuristics
- Checkout technical blogpost on NVIDIA Devblog: developer.nvidia.com/blog/scaling-quantum-circuit-simulation-with-cutensornet





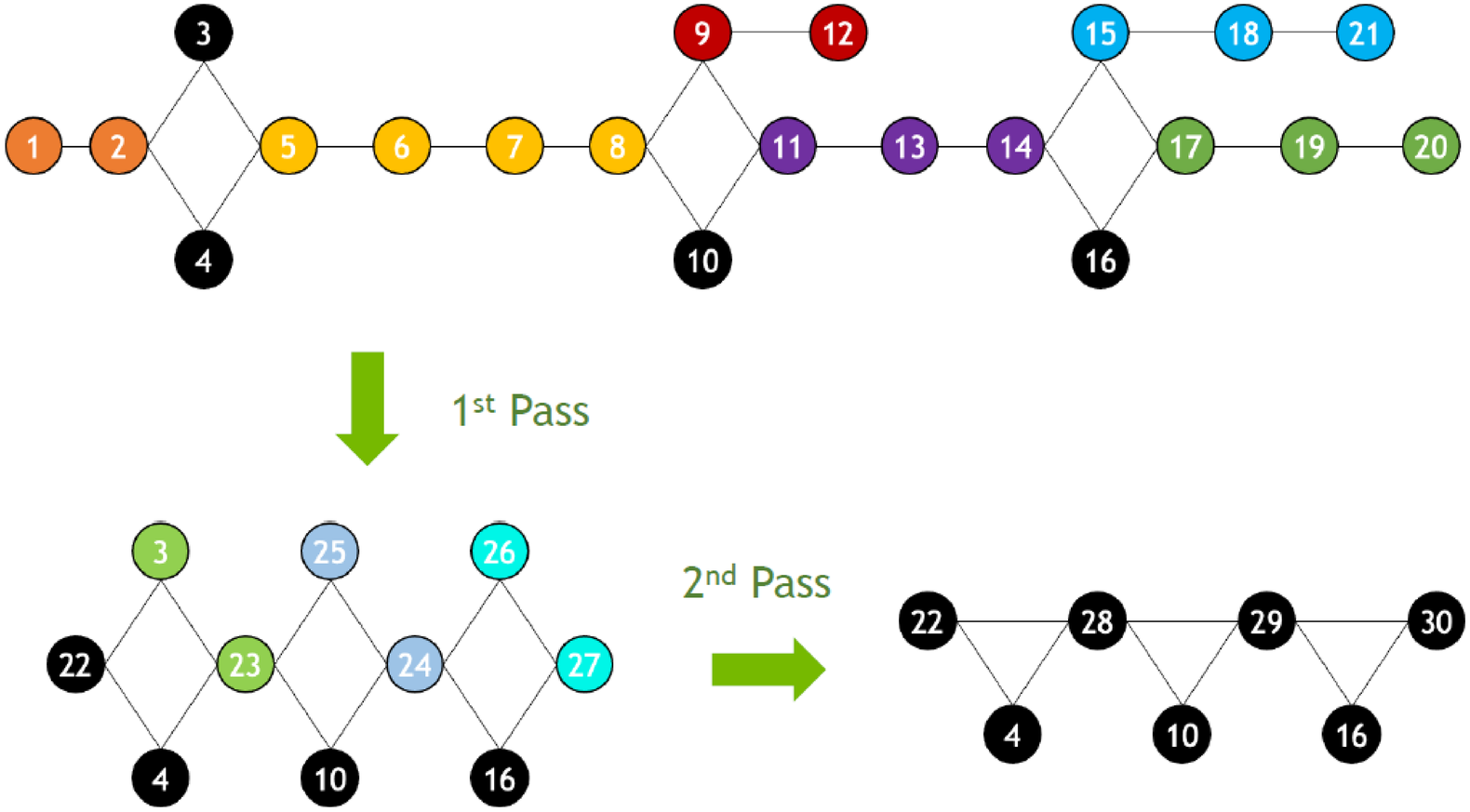
Source of Tensor Network image: Quimb - quimb.readthedocs.io

cuTensorNet Optimization & Flowchart



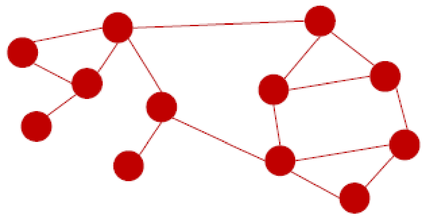
Tensor Network Simplification

- Simplification aims to reduce the computational cost of contracting the tensor network through preprocessing.
- cuTensorNet implements deferred rank-simplification, which identifies those pairwise contractions that do not increase the rank (number of dimensions) of the resulting tensor and sequences them to be performed first as a path prefix. This essentially creates a smaller network for the divisive algorithm as well as for reconfiguration to process.

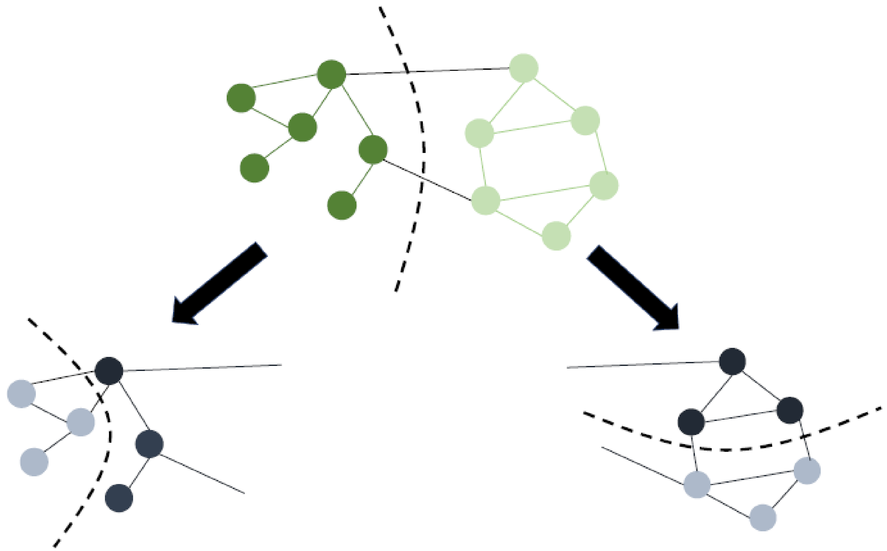


cuTensorNet Path Finder (Divisive Algorithm)

- The tensor network is represented as a graph, with tensors as the vertices and modes that are contracted as the edges.



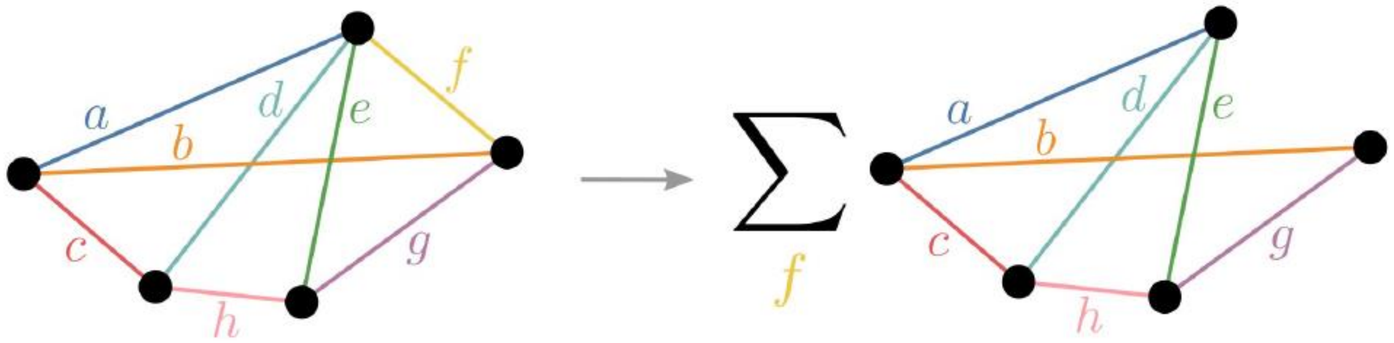
- The graph is partitioned into the specified number of partitions (2 shown) recursively until the size of each partition is less than or equal to the specified cutoff size (3 shown). Exhaustive search or an agglomerative algorithm is used to find the contraction order within as well as between partitions, from which the contraction order for the complete tensor network is built.



The colors map to the partitioning level, and the shades at each level distinguish different partitions.

Tensor Network Slicing for Parallelism & Minimizing Memory Requirements

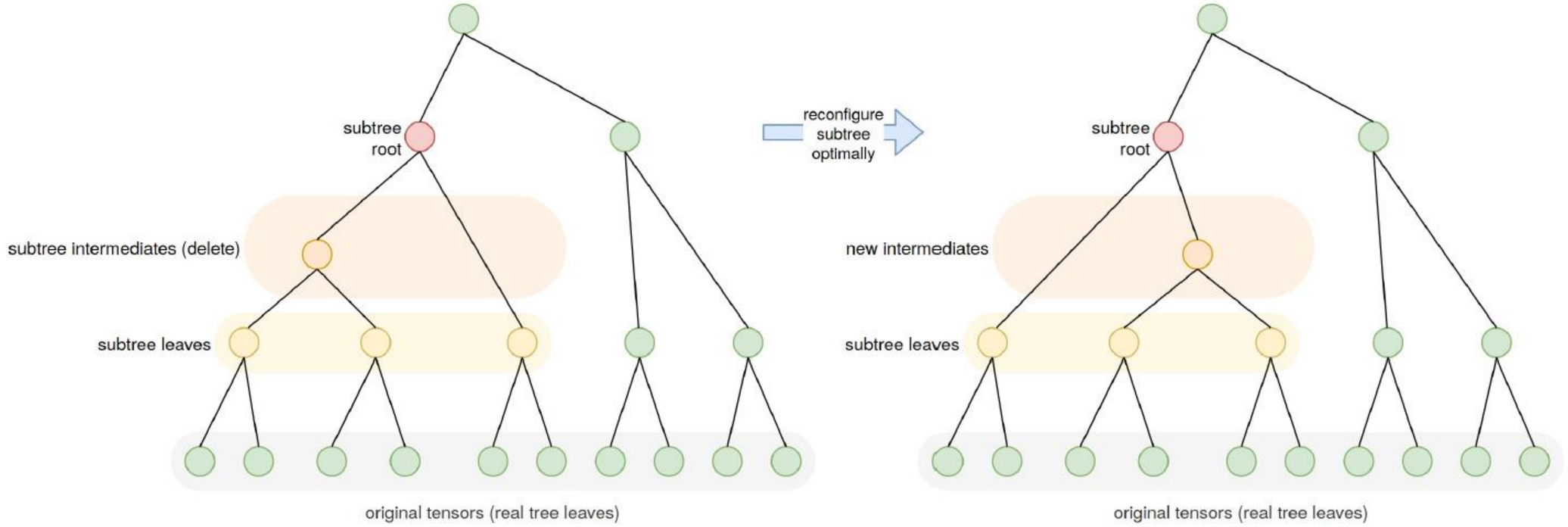
- *Slicing* is a technique to select a subset of edges from a tensor network (corresponding to mode labels) for explicit summation.
- A sliced network:
  - 1. results in lower memory requirements (often with some computational overhead), and
  - 2. allows for parallel execution.
- cuTensorNet implements *dynamic slicing*, which interleaves slicing with reconfiguration.



Source of image: github.com/jcmgray/cotengra by Johnnie Gray

Tensor Network Reconfiguration

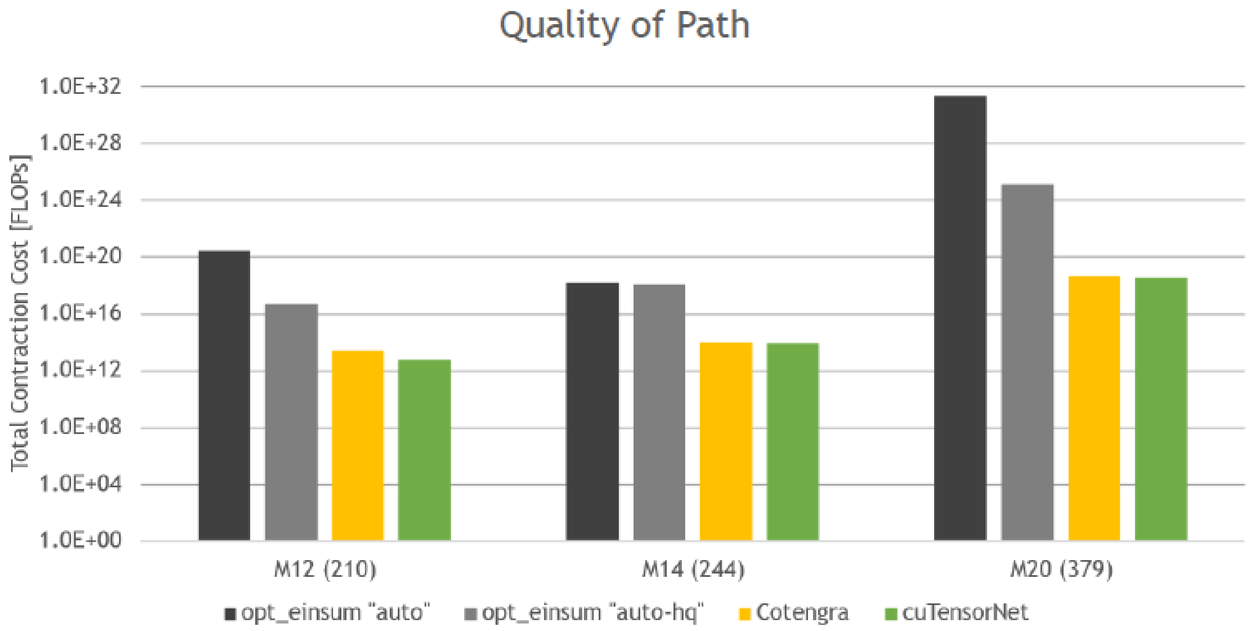
- The divisive algorithm computes a contraction path, which is a linearization of the contraction tree. The basic idea behind reconfiguration is to reduce the total contraction cost by reducing the contraction cost of portions (subtrees) of the contraction tree. The number of leaves in the subtree is typically chosen to be small enough so that the optimal algorithm can be used, and multiple iterations of reconfiguration are performed on different subtrees.
- As mentioned earlier, if slicing is active cuTensorNet interleaves reconfiguration with slicing to keep the contraction cost low.

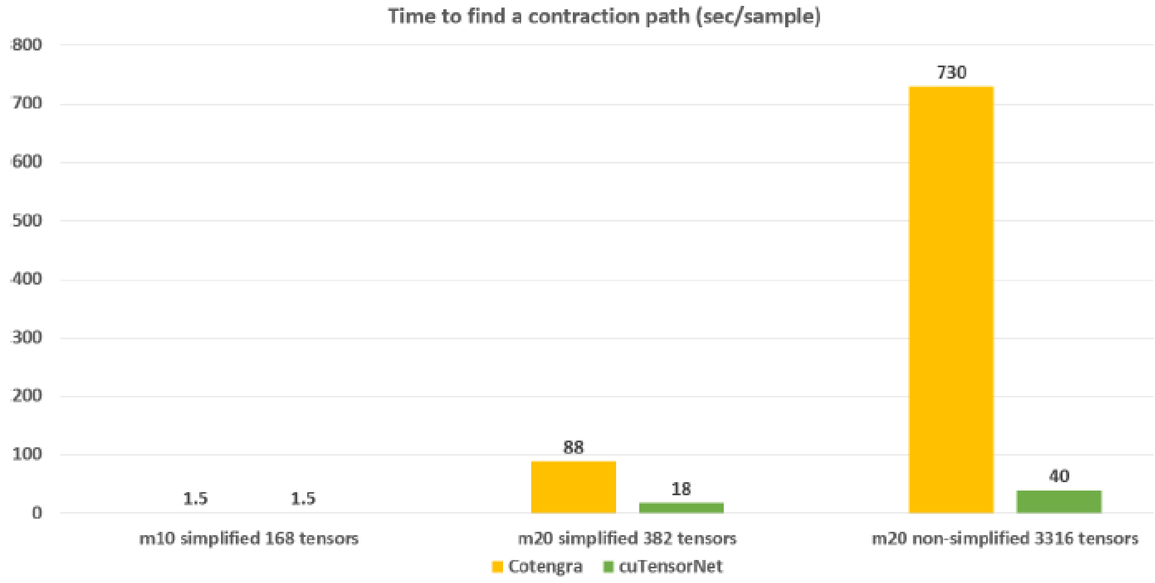


Source of image: github.com/jcmgray/cotengra by Johnnie Gray

cuTensorNet

Tensor Network path optimization performance

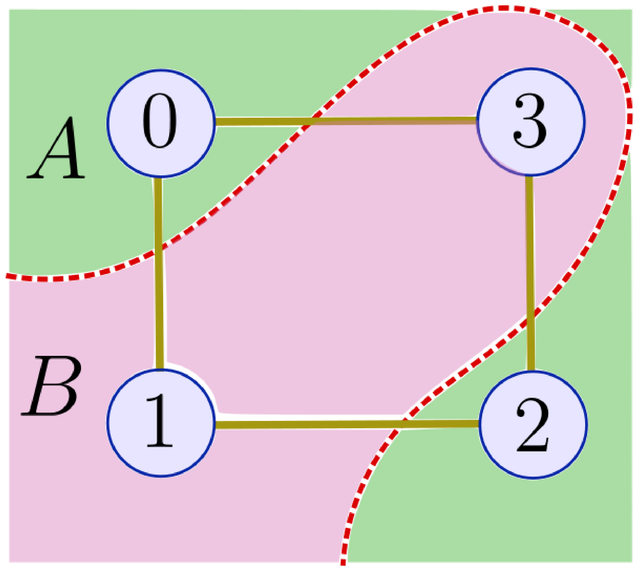


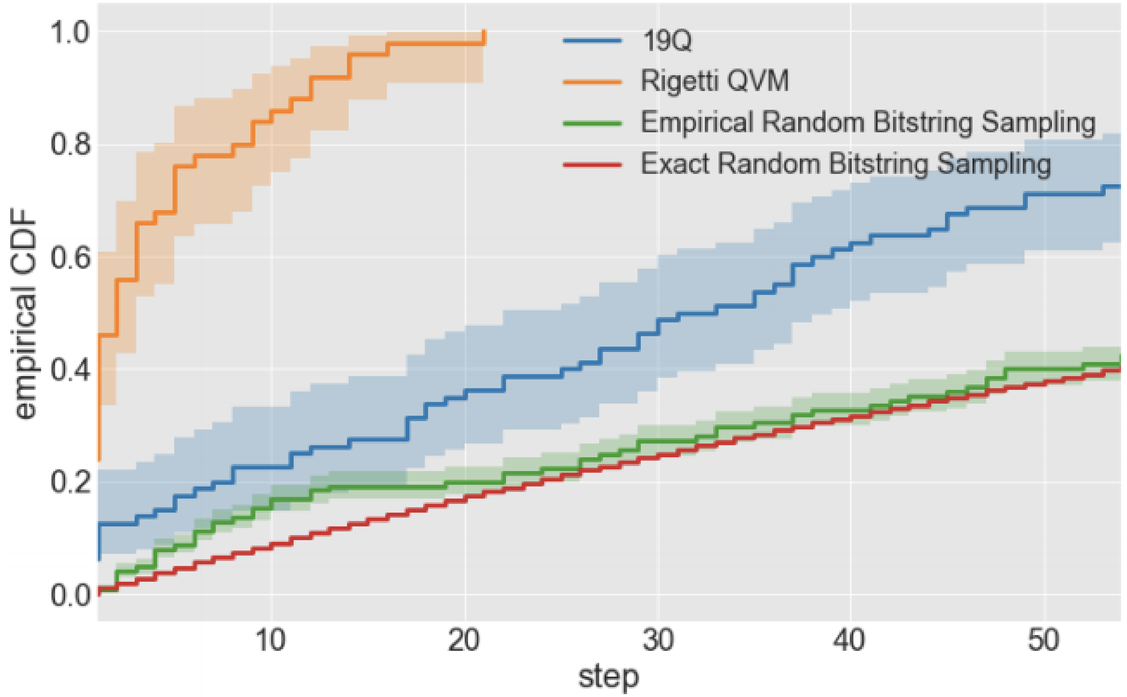


cuTensorNet achieves SotA pathfinding results dramatically faster, and does better with more complex networks

[1] Gray & Kourtis, Hyper-optimized tensor network contraction, 2021. URL: quantum-journal.org/papers/q-2021-03-15-410/pdf
[2] opt-einsum, URL: pypi.org/project/opt-einsum

The MaxCut Problem





- NP-Complete combinatorial optimization problem
- Applications include clustering, network design, Statistical Physics, and more

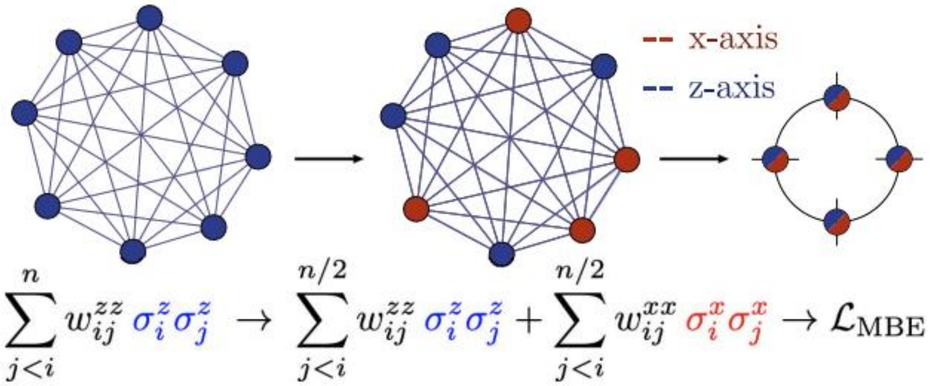
- Early target for hybrid variational quantum algorithms
- QAOA proposed by Farhi et al: arXiv:1411.4028
- Several HW demonstrations, including on Rigetti 19Q chip in 2017

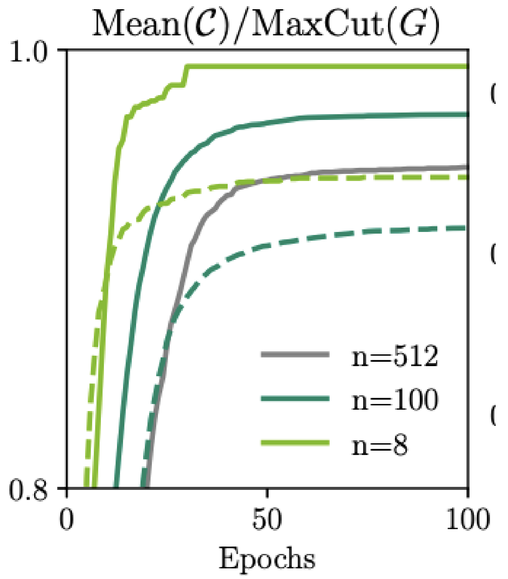
Source of images: Xanadu Inc. - pennylane.ai/qml/demos/tutorial_qaoa_maxcut.html (left) | Otterbach et al, Unsupervised Machine Learning on a Hybrid Quantum Computer, arXiv:1712.05771

Simulating MaxCut using Tensor Networks

- Tensor Networks are a natural fit for MaxCut
  - Fried et. al. (2017) arxiv.org/abs/1709.03636
  - Huang et. al (2019) arxiv.org/abs/1909.02559
  - Lykov et. al. (2020) arxiv.org/abs/2012.02430

- Patti et. al.(2021): NVIDIA Research proposes a novel variational quantum algorithm
  - Based on 1D tensor ring representation
  - Multibasis encoding
  - Able to find accurate solution for 512 vertices (256 qubits) on a single GPU
  - Paper: arxiv.org/abs/2106.13304
  - Code: github.com/tensorly/quantum

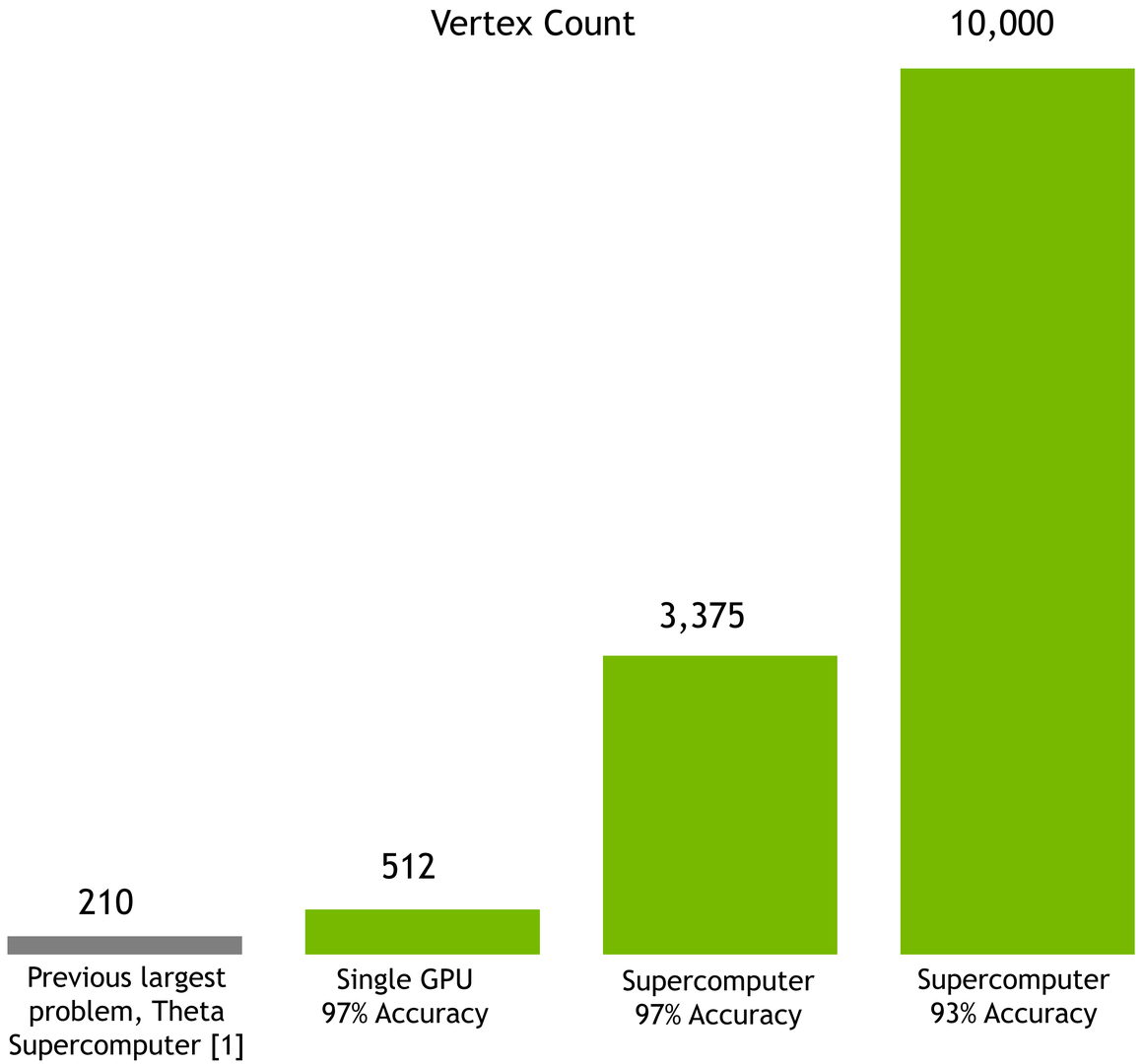




Scaling to a GPU Supercomputer: NVIDIA DGX SuperPOD

NVIDIA's Selene DGX SuperPOD based supercomputer

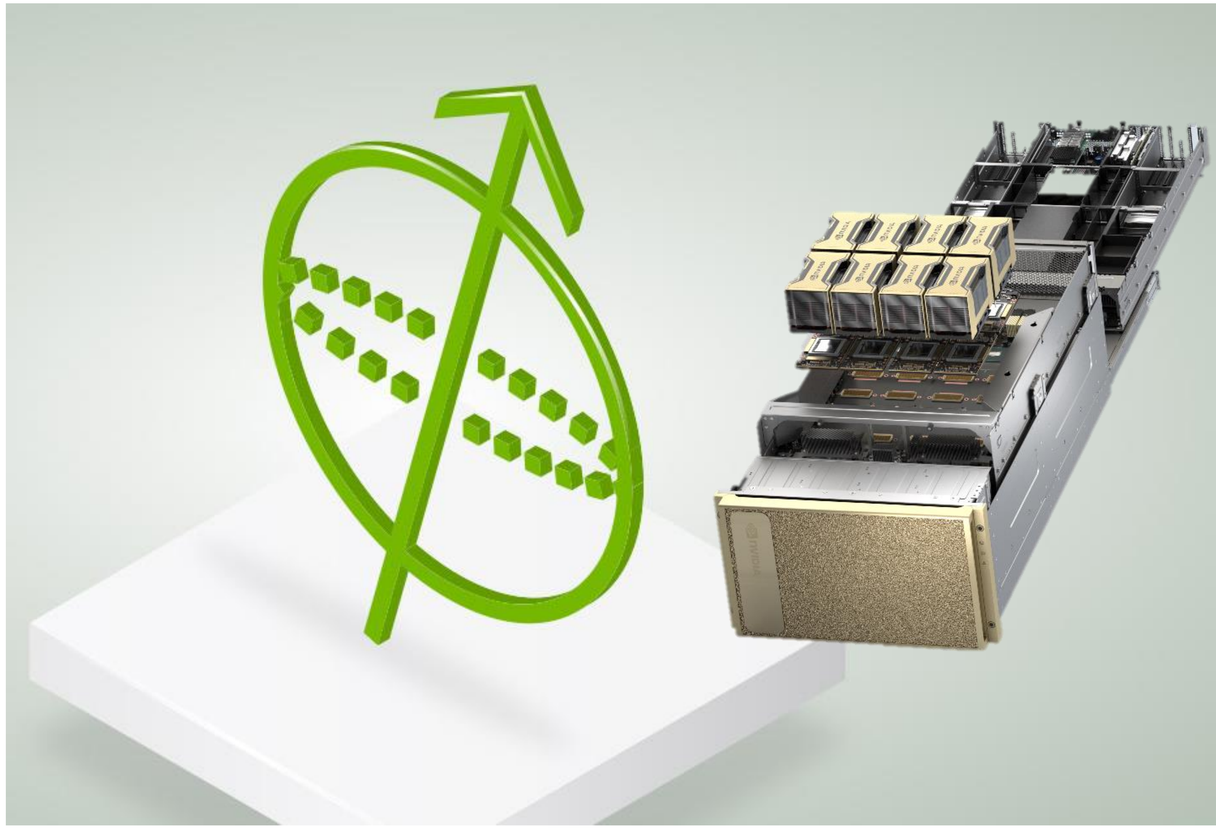
- Using NVIDIA's Selene supercomputer
- Solved a 3,375 vertex problem (1,688 qubits) with 97% accuracy
- Solved a 10,000 vertex problem (5,000 qubits) with 93% accuracy

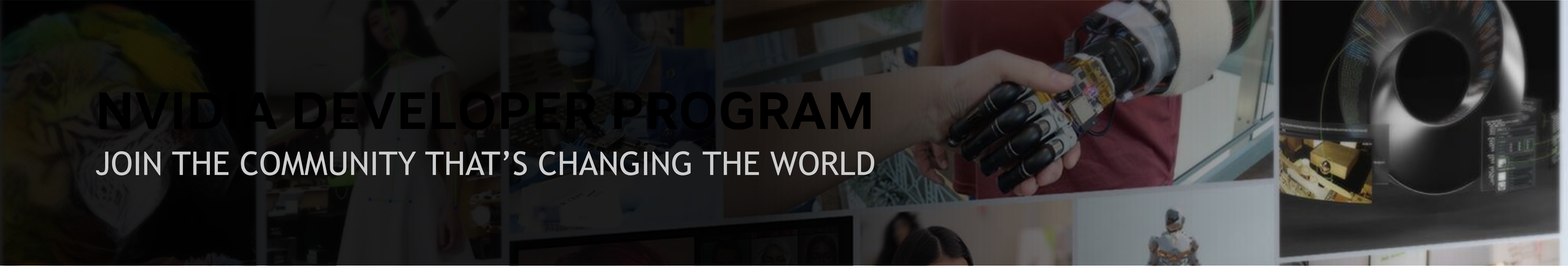


[1] Danylo Lykov et al, Tensor Network Quantum Simulator With Step-Dependent Parallelization, 2020 arxiv.org/abs/2012.02430

Summary

- Quantum circuit simulation is an approach to conduct quantum computation with classical computer processors like CPUs and GPUs
- cuQuantum makes it easy for anyone with NVIDIA hardware to accelerate and scale their simulations more than previously possible
- An expanding ecosystem is using cuQuantum to enable quantum research
- Get stated with cuQuantum today by pulling our container from NGC, downloading the SDK from our DevZone, via pip or conda install, or through other frameworks





NVIDIA DEVELOPER PROGRAM

JOIN THE COMMUNITY THAT'S CHANGING THE WORLD

TOOLS

- Get exclusive access to an extensive library of NVIDIA software, spanning all of NVIDIA's technology platforms.
- Save time with ready-to-run, GPU-optimized software, model scripts, and containerized apps from the NVIDIA NGC™ catalog.
- Participate in early access programs where you can be one of the first to experience the latest NVIDIA technology.

TRAINING

- Take advantage of research papers, technical documentation, developer blogs, and industry-specific resources.
- Choose from a broad catalog of training options through the NVIDIA Deep Learning Institute (DLI).
- Get unlimited access to NVIDIA On-Demand, the home for NVIDIA resources from GTCs and other leading industry events.

COMMUNITY

- Network with like-minded developers, engage with GPU experts, and contribute to discussions in the developer forums.
- Attend exclusive meetups, GPU hackathons, and events.
- Connect with NVIDIA experts through developer-focused webinars and Instructor-led workshops.

Join the Free Program developer.nvidia.com/join

Thank you!

dfisk@ nvidia.com