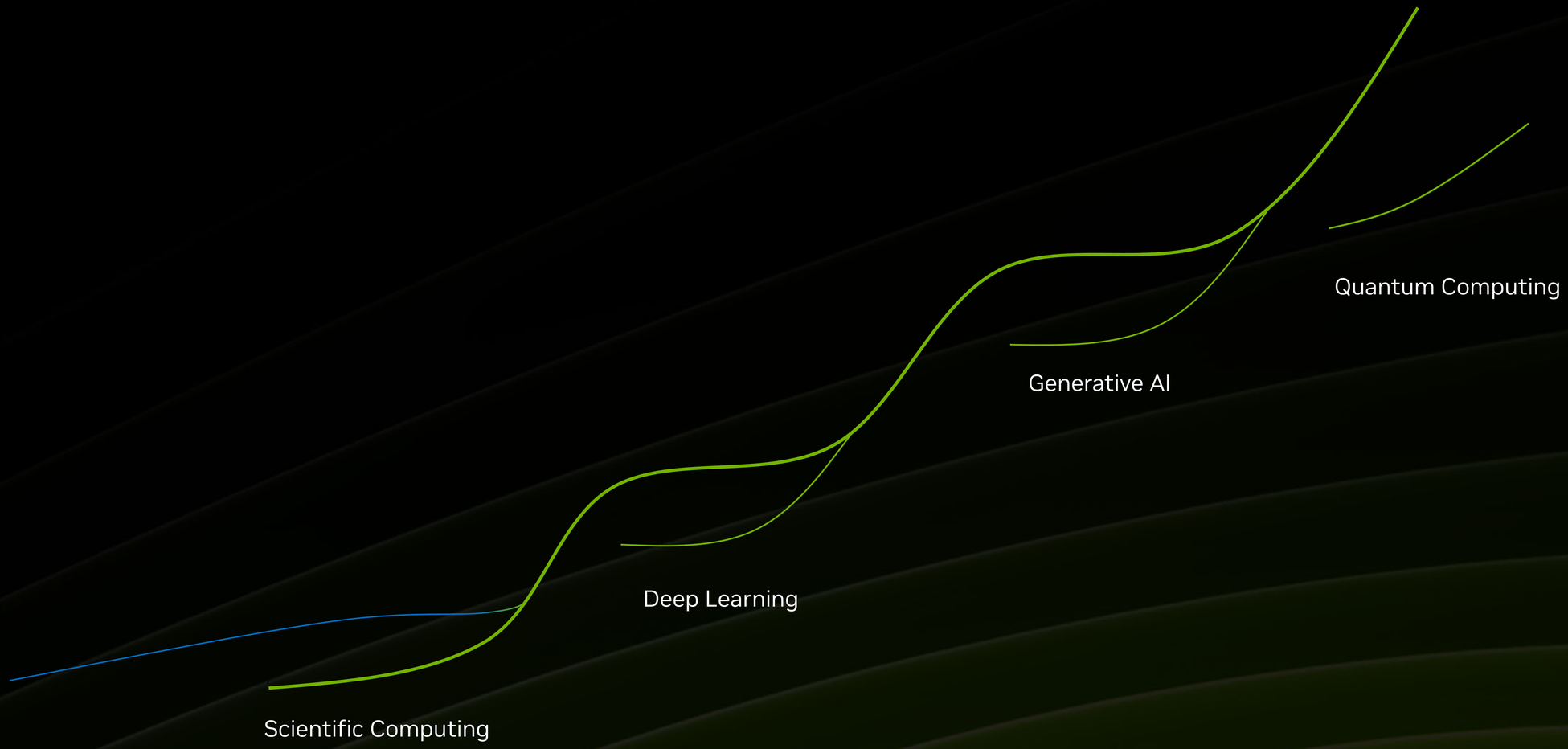
NVIDIA Quantum Platform Sharing

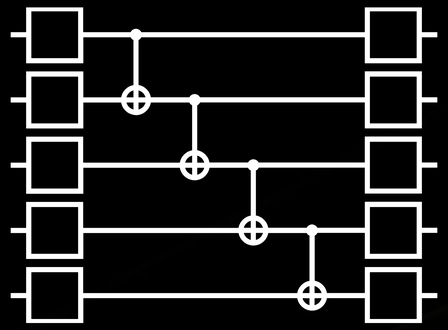
Pika Wang | Solution Architect | Quantum, NVAITC

Accelerated Computing



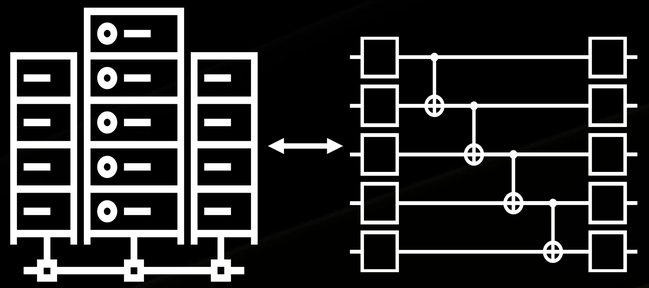
NVIDIA Quantum

Powering the Global Quantum Computing Community



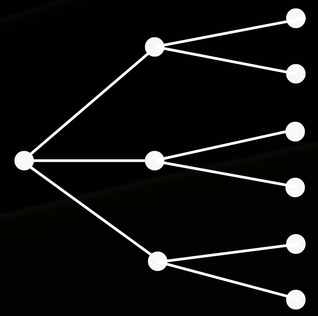
Simulation

Algorithm Design, Resource Estimation, QPU Design



HPC Quantum Integration

Integrated Applications, QEC, Sub-Microsecond Latency



AI for Quantum

QEC, Calibration, Algorithms

CUDA-Q

Libraries | Programming Model | Tools | Infrastructure

cuQuantum

Quantum Simulation

DGX-Q

Quantum Integrated Computing

GPU Supercomputing

NVIDIA Quantum Platform Portfolio

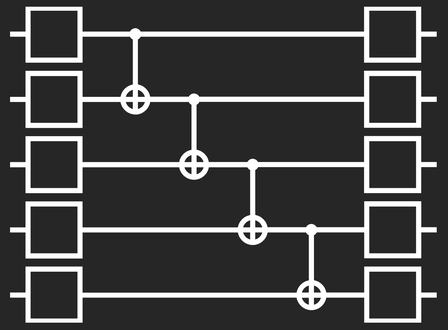
Research the computers of tomorrow with the most powerful computers of today

CUDAQ is the open-source quantum analog of CUDA. It enables domain scientists to easily and performantly coprogram CPU, GPU, and simulated or real QPUs

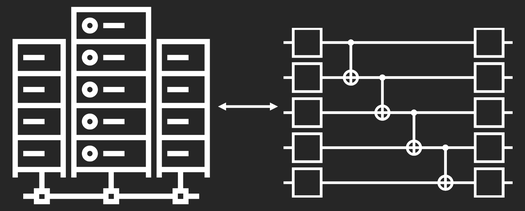
cuQuantum integrates with all quantum frameworks enabling GPU-accelerated, supercomputer scale quantum simulations

DGX-Q is an integrated system with a low-latency connection from Grace Hopper to a quantum control system, enabling tight physical GPU integration with any QPU

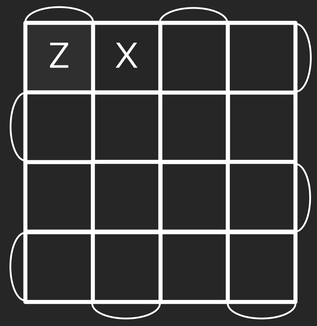
Quantum Algorithms Research



Quantum-Integrated Applications



Error Correction, Calibration, Control



cuQuantum
Accelerated Quantum Simulation

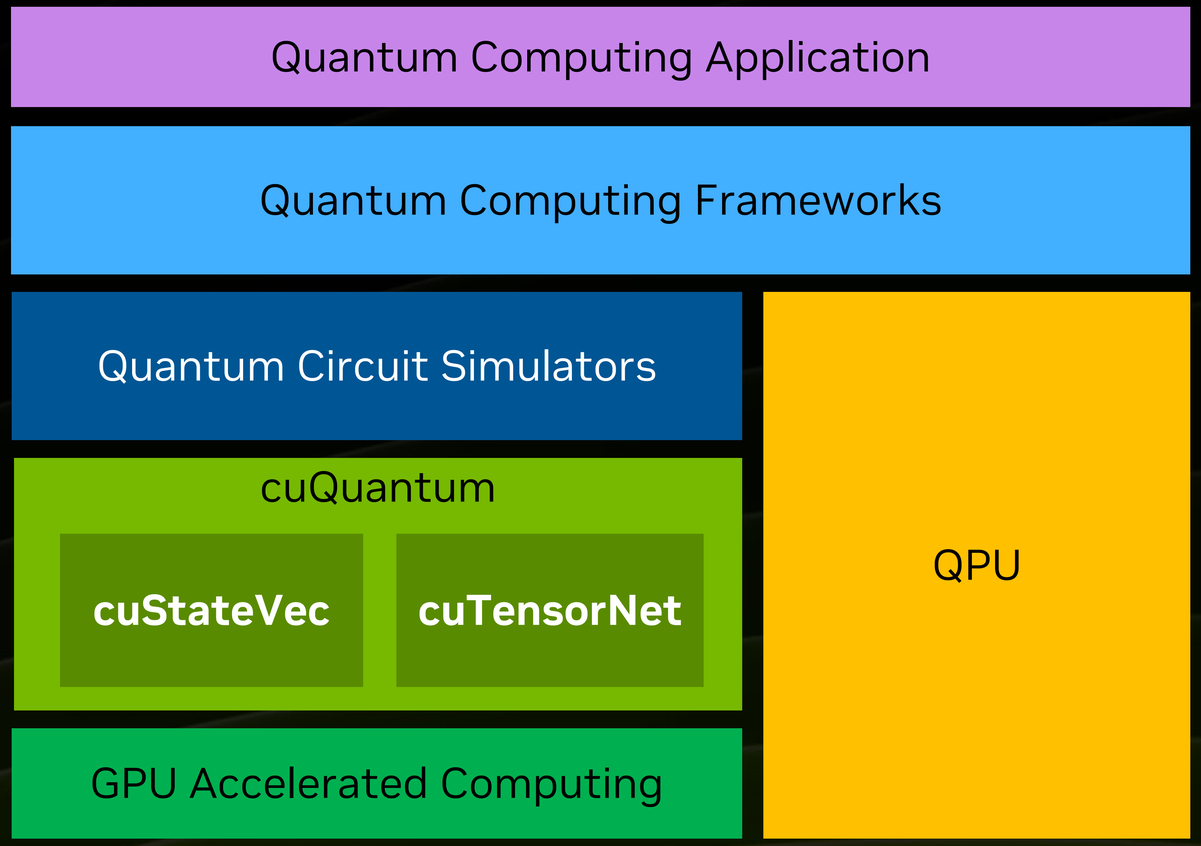
CUDA-Q
Quantum-Classical Developer Platform

Quantum Integrated GPU Supercomputing
DGX | HGX | **DGX-Q**

NVIDIA

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
- **cuQuantum** is **a collection of software libraries supporting simulator in existing frameworks** to speedup quantum simulation



CUDA-Q

- Seamless Integration of scientific computing ecosystem
  - Programming model extending C++ and Python with quantum kernels
  - Open programming model, open-source compiler
  - QPU Agnostic – includes superconducting, trapped ion, neutral atom, photonic, and nitrogen-vacancy center QPUs

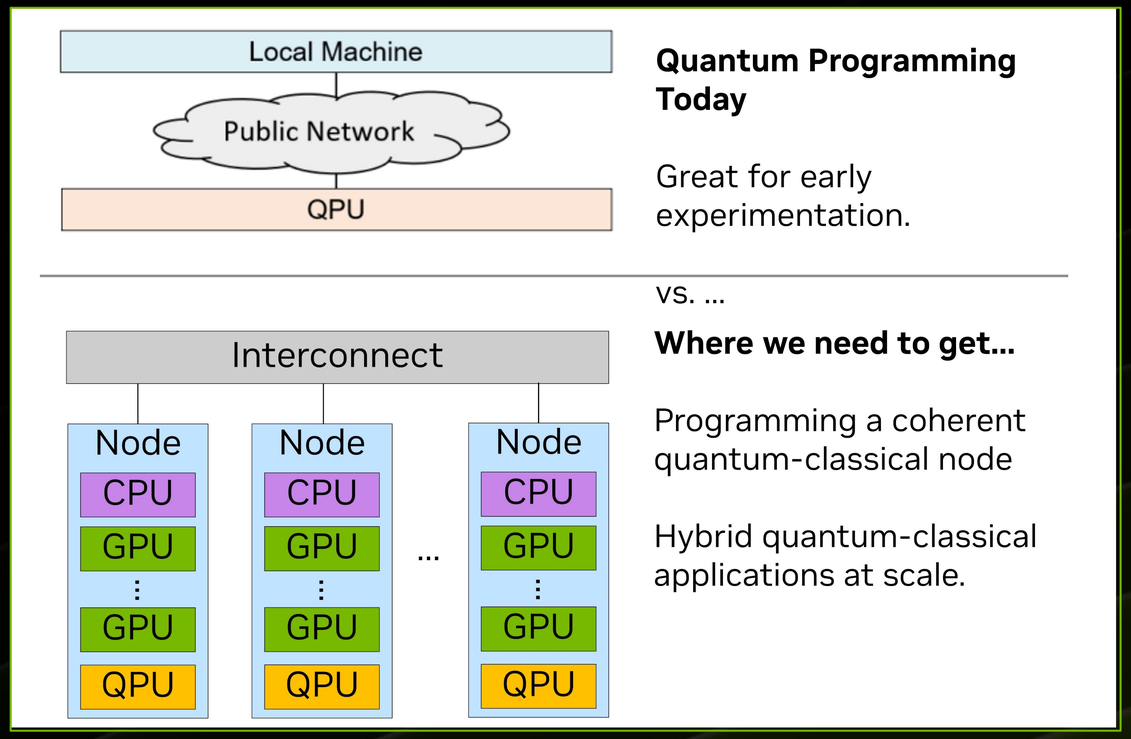
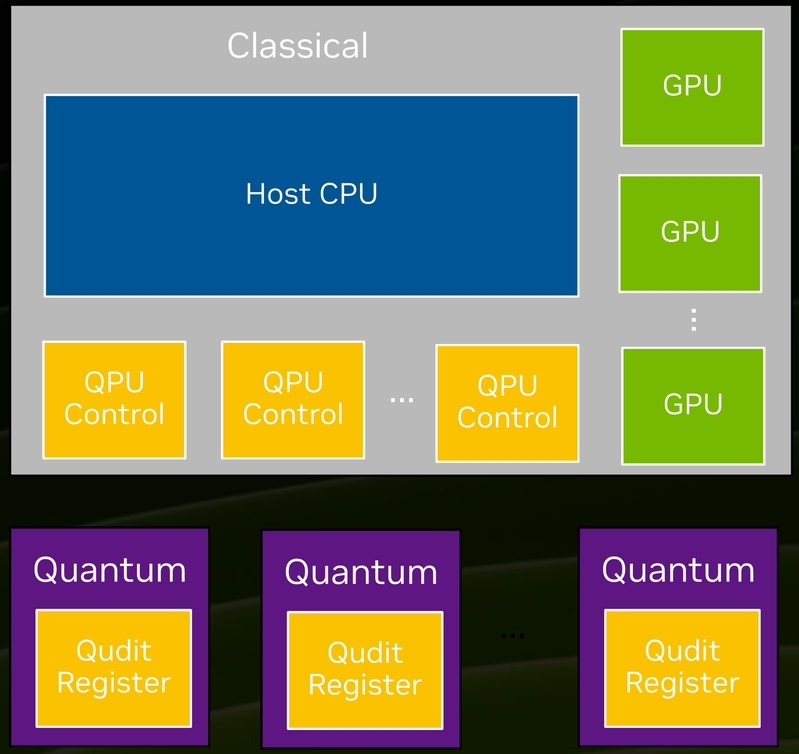


Figure adapted from:
Quantum Computers for High-Performance Computing. Humble, McCaskey, Lyakh, Gowrishankar, Frisch, Monz. IEEE Micro Sept 2021. 10.1109/MM.2021.3099140

- System architecture discrete memory spaces
  - Available for hybrid quantum-classical code
  - GPU for pre- / post-processing, hybrid application workflows
  - Agnostic to the control system architecture



CUDA Quantum Python Bindings

Examples

- **Sample**

```
import cudaq

# Set the Simulator to cuQuantum
cudaq.set_target('nvidia')

# Create a Bell State Kernel
bell = cudaq.make_kernel()
qr = bell.qalloc(2)
bell.h(qr[0])
bell.cx(qr[0], qr[1]);
bell.mz(qr)

# Print the Quake Code
print(bell)

# JIT Compile and Execute
counts = cudaq.sample(bell)
print(counts)
```

- **Observe**

```
import cudaq

# Set the backend
cudaq.set_target('quantinuum')

# Create the kernel function signature
# here void(float)
ansatz, theta = cudaq.make_kernel(float)
q = ansatz.qalloc(2)
ansatz.x(q[0])
ansatz.ry(theta, q[1]);
ansatz.cx(q[0], q[1]);

h = cudaq.SpinOperator(...)

# API mirrors the C++
result = cudaq.observe(ansatz, h, .59)
print('<H> = ', result.expectation_z())
```

- **Noise Modeling**

```
import cudaq

# Set the backend
cudaq.set_target('density-matrix-cpu')

# Create a depolarization channel on
# X operations on qubit 0
depol = cudaq.DepolarizationChannel(.1)
noise = cudaq.NoiseModel()
noise.add_channel('x', [0], depol)

# Create your kernel code
bell = cudaq.make_kernel()
...

# Sample in the presence of noise
noisyCounts = cudaq.sample(bell,
                           noise_model=noise)
print(noisyCounts)
```

NVIDIA Quantum

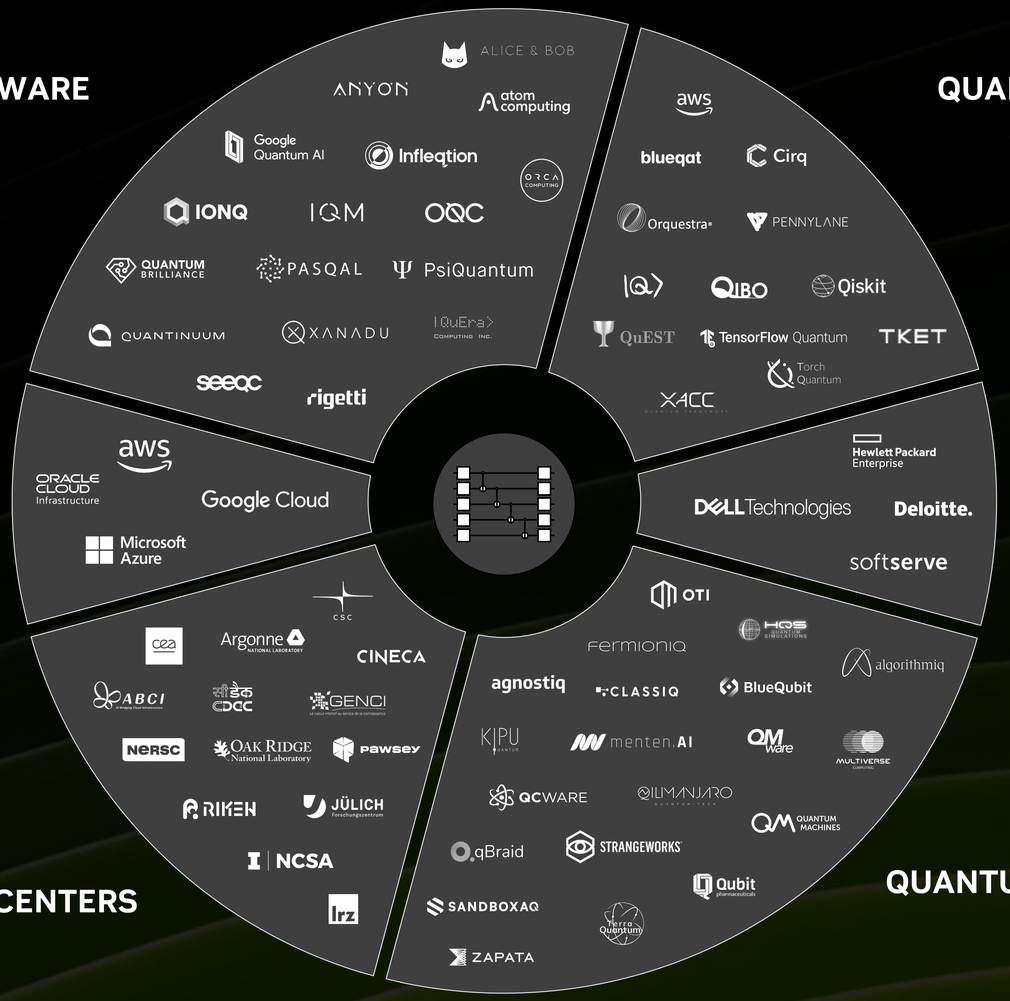
Powering the Global Quantum Computing Community

160+
Quantum Partners

>90%
Largest Startups

>75%
QPUs Integrating CUDA Q

15/17
Leading Frameworks Accelerated

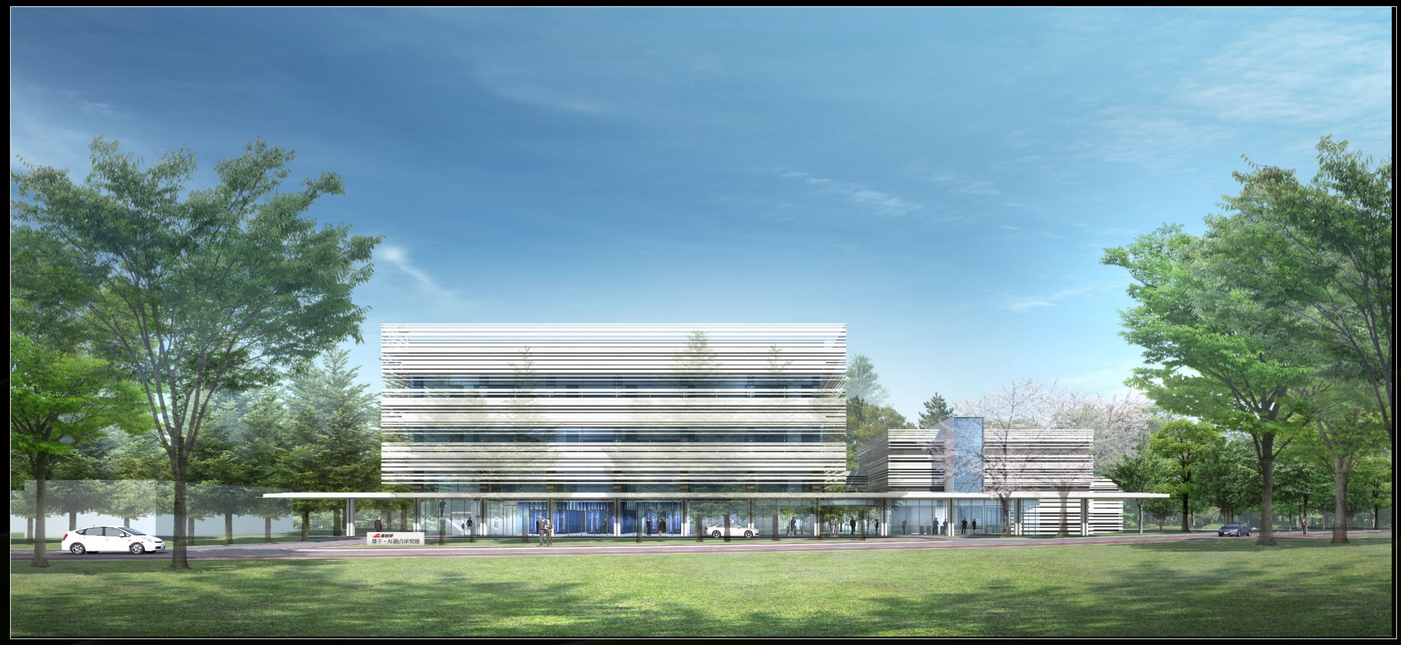


NVIDIA

Announcing ABCI-Q

G-QuAT/AIST Supercomputer for Quantum Research

- 2000+ H100 GPUs, over 500 nodes, connected by Infiniband and powered by CUDA-Q
- Built by Fujitsu, at the G-QuAT/AIST ABCI Supercomputing Center in Tsukuba
- A platform for the advancement of quantum simulation, the integration of quantum-classical systems, and the development of new algorithms inspired by quantum technology



https://nvidianews.nvidia.com/news/nvidia-powers-japans-abci-q-supercomputer-for-quantum-research

"ABCI-Q will let Japanese researchers explore quantum computing technology to test and accelerate the development of its practical applications. The NVIDIA CUDA-Q platform and NVIDIA H100 will help these scientists pursue the next frontiers of quantum computing research."

- Masahiro Horibe, deputy director of G-QuAT/AIST

Announcing Gefion

Novo Nordisk Foundation Supercomputer for Research in Quantum, Healthcare and the Green Transition

- 1500+ H100 GPUs in DGX SuperPOD AI Supercomputer powered by H100 Tensor Core GPUs and Infiniband Networking
- Will support the NQCP, NNF's 12- year initiative to build a quantum computer capable of solving important problems in the life sciences.
- Danish research to be accelerated by NVIDIA SW platforms including CUDA-Q and BioNemo

“Gefion will be an essential tool to support our full stack quantum computing research as part of the NQCP - NNF’’s 12- year initiative to enable the development a quantum computer capable of solving important problems in the life sciences."

- Peter Krogstrup, CEO Novo Nordisk Foundation Quantum Computing Programme

First DGX-Q Deployment

Israeli Quantum Computing Center to Deploy World's First Tightly Integrated Quantum-Classical System

- IQCC to deploy DGX-Q, combining NVIDIA Grace Hopper Superchips with Quantum Machines OPX Quantum Control
- System to be connected to multiple QPUs with sub-microsecond latency from GPU to QPUs from ORCA, Quantware and Rigetti
- Enables research in Real-Time Accelerated Error Correction

ORCA COMPUTING

QUANTUM MACHINES

rigetti

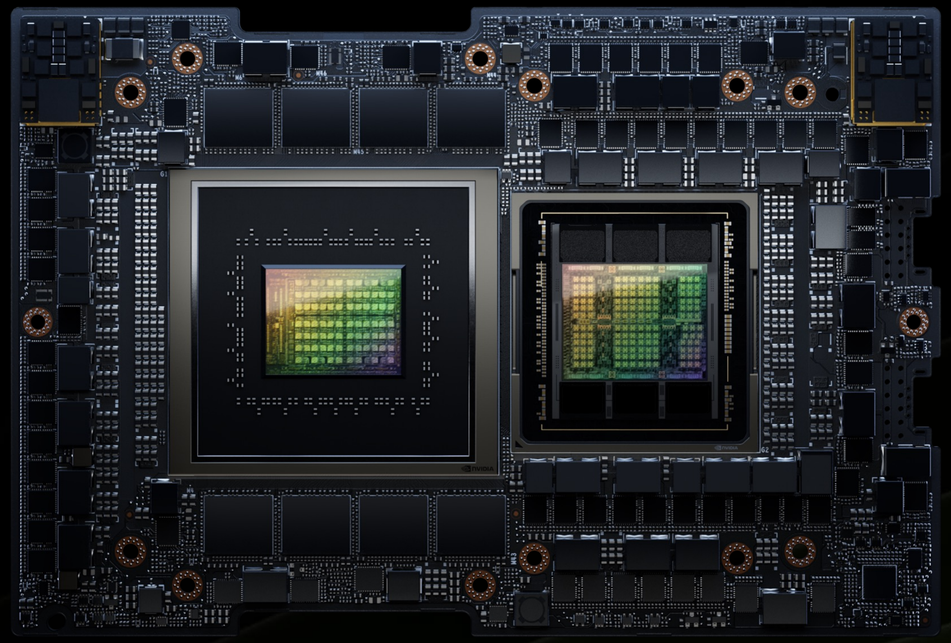
"With DGX Quantum, QM-NVIDIA collaboratively develop a game-changing capability that's essential to reach quantum advantage. We are thrilled about this technology, which will enable quantum computer builders and researchers to unleash the next wave of massive performance improvements. Pioneers in quantum error mitigation and quantum error correction now have a brand-new playing field thanks to ultra low latency feedback and high throughput processing."

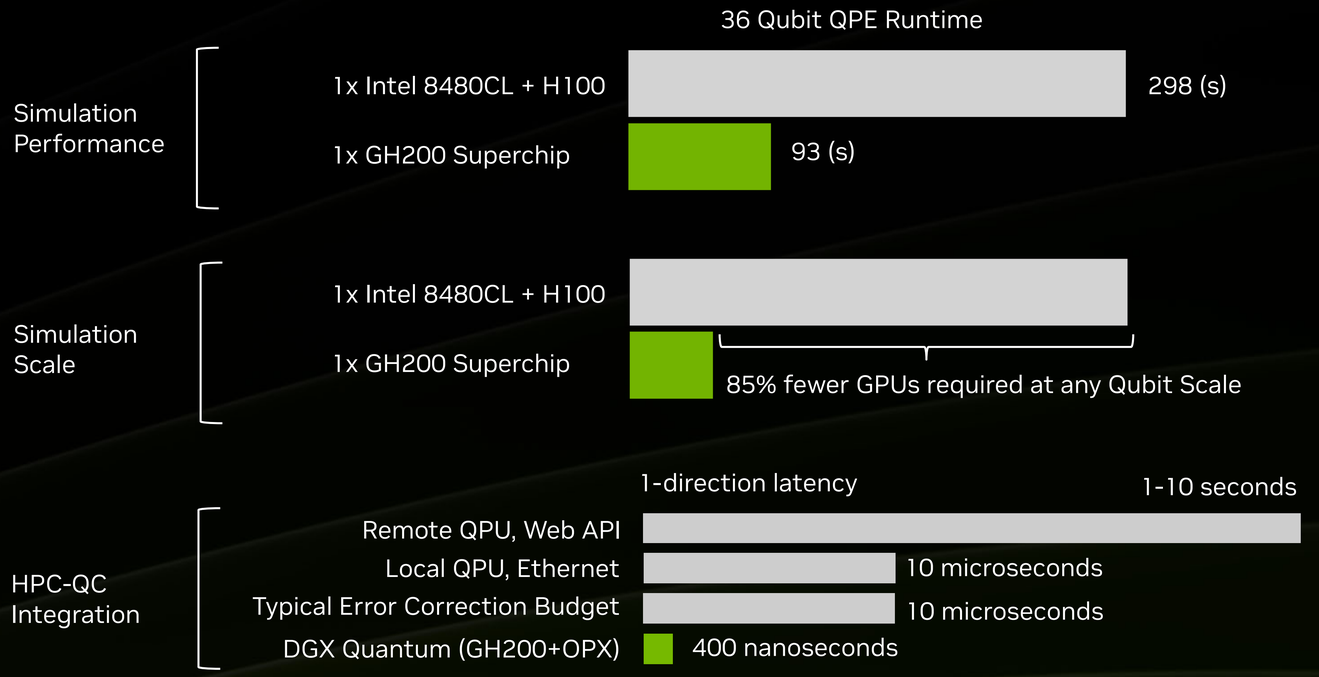
- Itamar Sivan, CEO of Quantum Machines

NVIDIA

Grace Hopper: The Engine for Quantum Research

New Grace Hopper Deployments for Quantum Research





FCAT FIDELITY CENTER for APPLIED TECHNOLOGY.

fermioniq

INSTITUT QUANTIQUE UNIVERSITÉ DE SHERBROOKE

JÜLICH Forschungszentrum

OQC

pawsey

qBraid

QUANTUM DATA CENTER

QUANTUM MACHINES

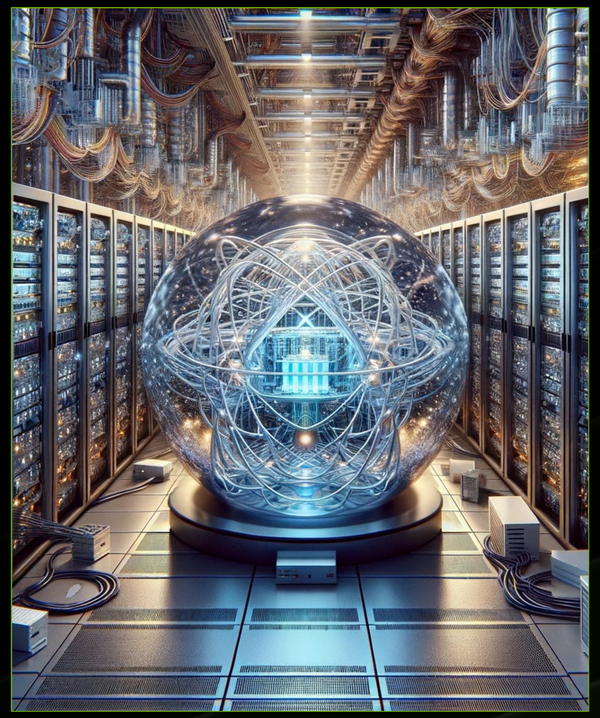
Rensselaer

XANADU

NVIDIA

NVIDIA Quantum

Platform Advancements



QPU Support

- Newly Completed CUDA-Q QPU Integrations: OQC, IQM

IQM and Oxford Quantum Circuits QPU backends

A QPU backend is a hardware computing device that acts as a quantum processing unit and can run quantum workloads. CUDA-Q is integrated with several quantum hardware providers' QPUs.

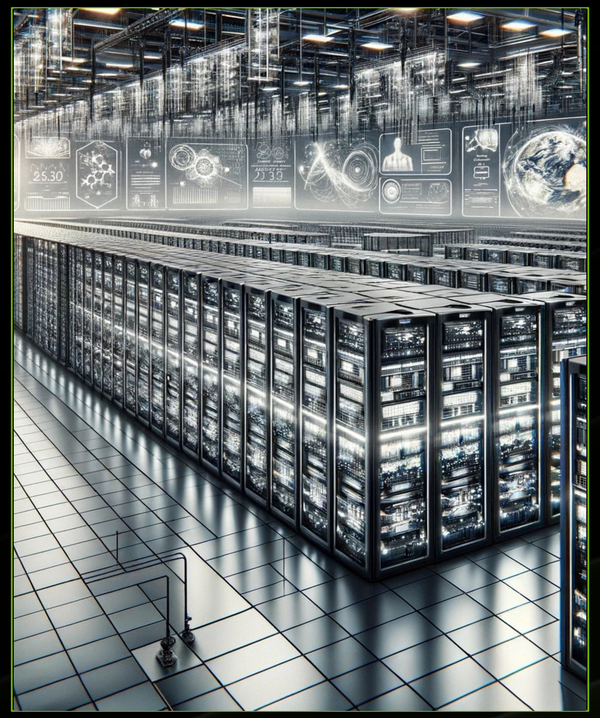
IQM and Oxford Quantum Circuits (OQC) quantum computers are now supported as QPU backends in CUDA-Q. This is a great addition to the already supported quantum computers from Quantinuum and IonQ, which enable you to run CUDA-Q code on a variety of different quantum technologies available today.

S631123 – Integrating CUDA Quantum with Quantum Computers

Vladimir Kukushkin, IQM

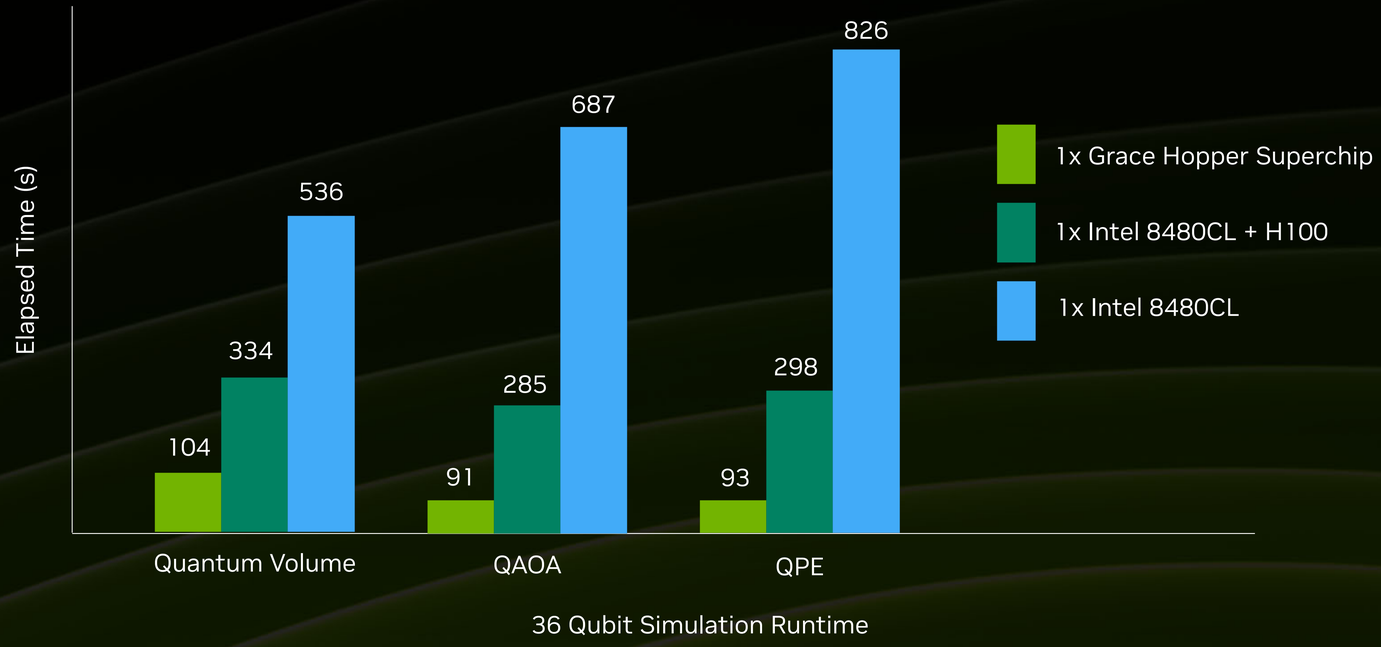
NVIDIA Quantum

Platform Advancements



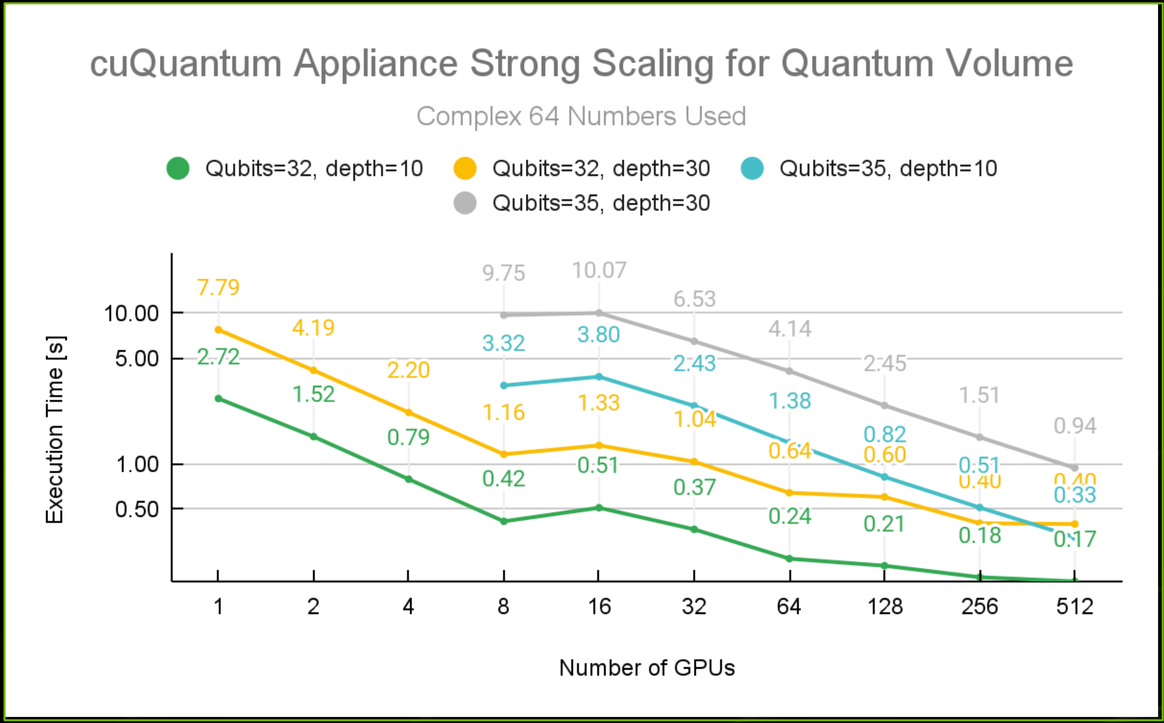
Scale and Performance

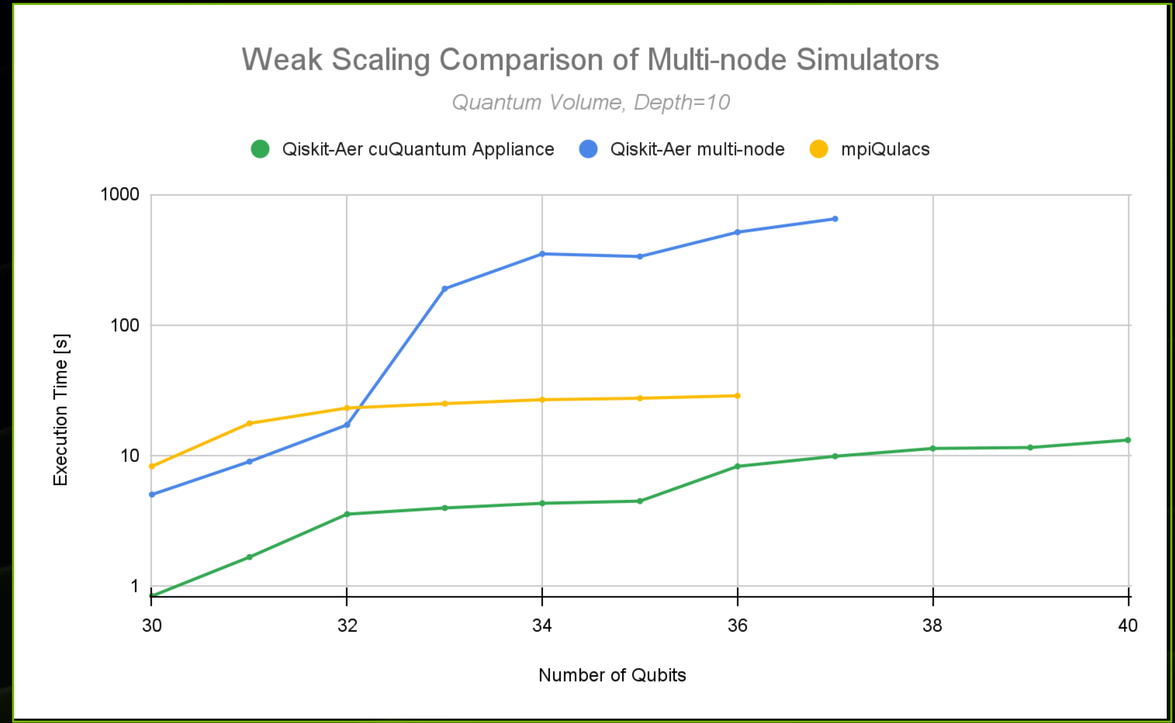
- CUDA-Q Support for Multi-Node Multi-GPU Simulation
- MPI Interoperability
- Grace Hopper Shared Memory Support



NVIDIA Quantum

Best-in-class performance on ABCI supercomputer





https://developer.nvidia.com/blog/best-in-class-quantum-circuit-simulation-at-scale-with-nvidia-cuquantum-appliance/

- cuQuantum-integrated simulator in the popular framework (Qiskit) can further speedup
- Benchmark against the competitor (mpiQulacs)

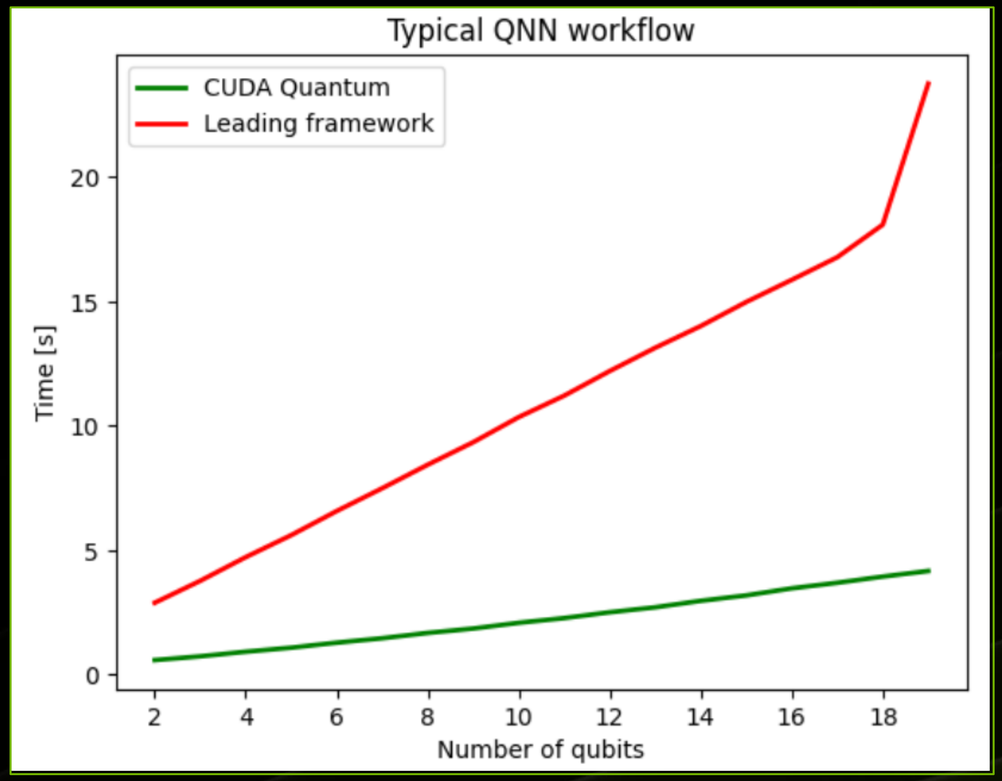
The simulation was held to 32 and 35 qubits and distributed across 512 NVIDIA A100 40GB GPUs on the ABCI Compute Node (A)

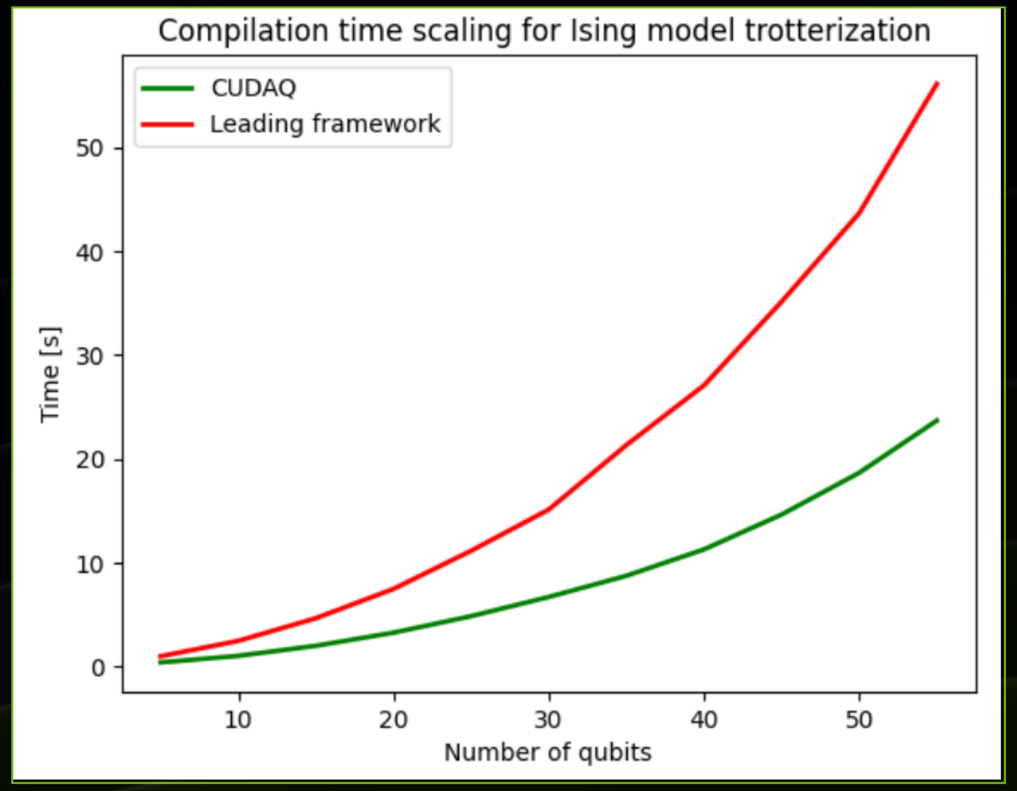
mpiQulacs is a fast, multi-node, full-state vector quantum circuit simulator developed to run on the Fujitsu A64FX CPU architecture

NVIDIA

NVIDIA Quantum

CUDA-Q Compile Time Speedup





- **5x speedup**: Comparison between CUDA Quantum with a leading quantum computing SDK, both leveraging the NVIDIA cuQuantum backend to optimally offload circuit simulation onto NVIDIA GPUs.

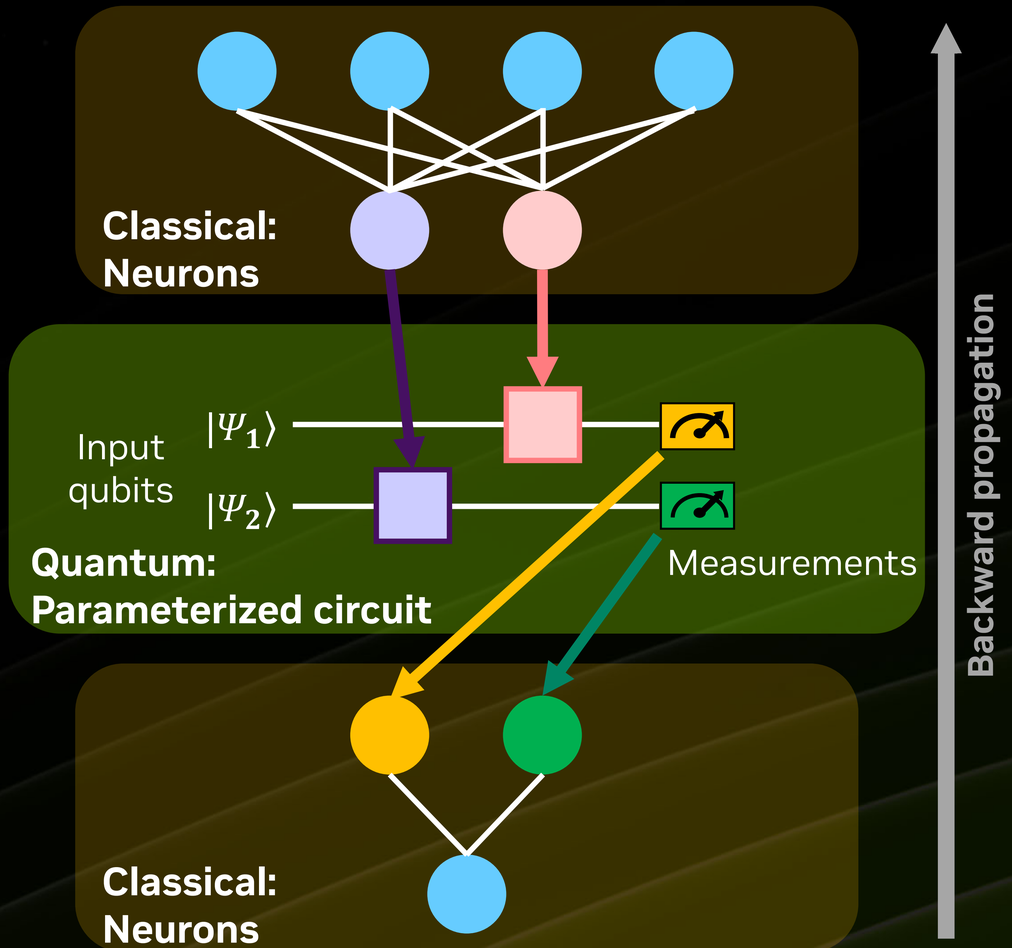
- **2.4x Faster**: The nvq++ compiler used by CUDA Quantum is on average faster compared to its competition. Compilation involves circuit optimization, decomposing into the native gate sets supported by the hardware and qubit routing.

https://developer.nvidia.com/blog/programming-the-quantum-classical-supercomputer/

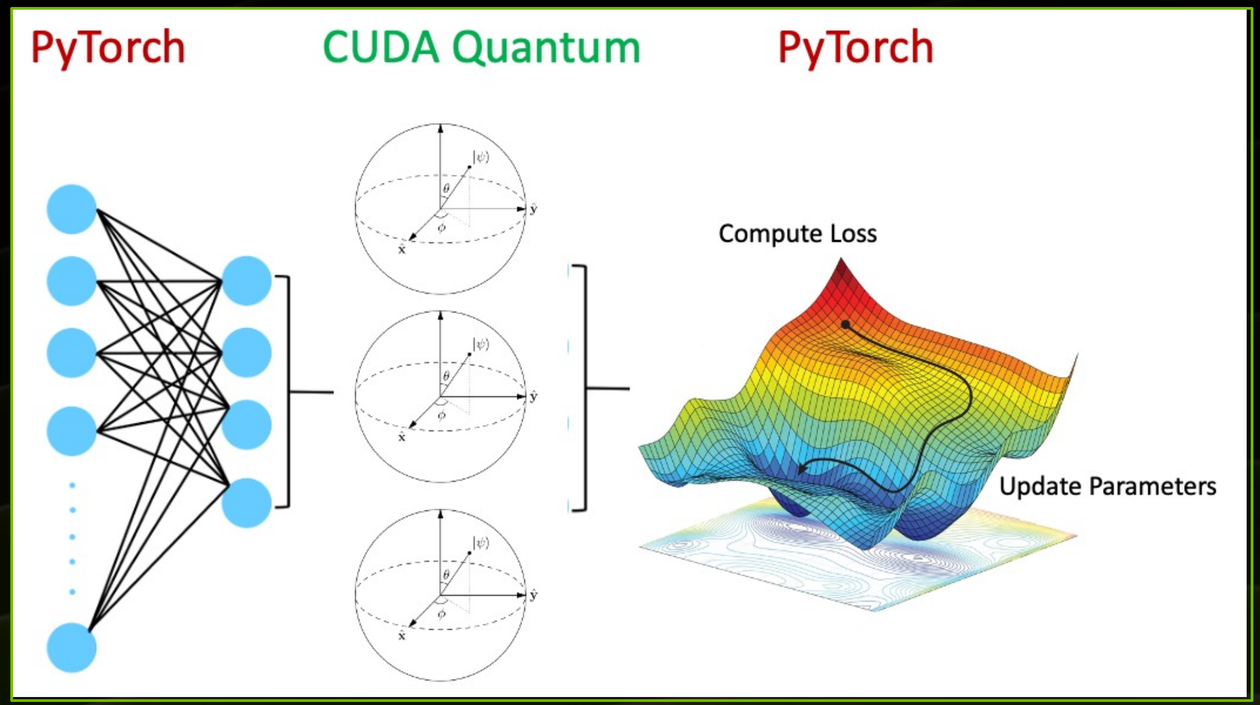
Seamless Integration of CUDA-Q and AI Framework

Hybrid quantum neural network architecture accelerated by GPUs made possible by CUDA Quantum

- Explore hybrid quantum-classical algorithms
- Combine quantum circuit and neural network



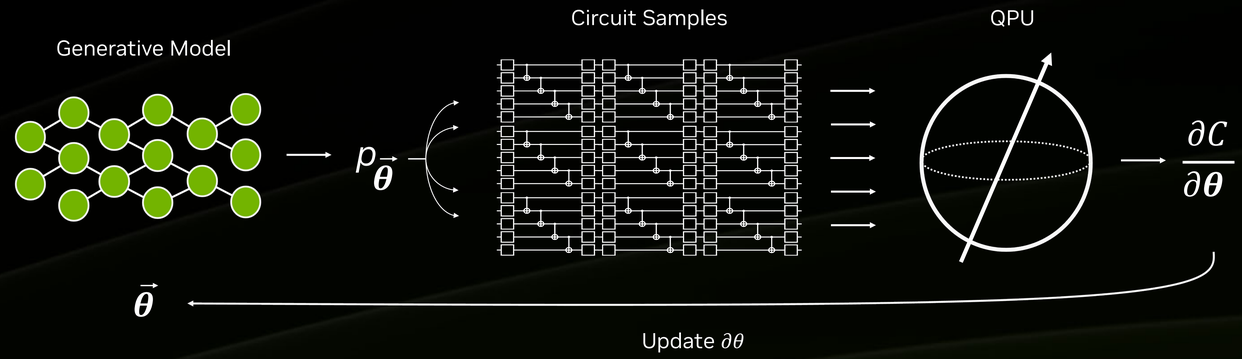
- CUDA Quantum also seamlessly integrates with existing machine learning frameworks



Generative AI + Quantum Algorithms

University of Toronto, St Jude's, and NVIDIA partner to invent GPT-QE

- Generative Pre-Trained Transformer-based (GPT) method for computing the ground state energies
- First GPT-generated quantum circuit
- Run via CUDA-Q on NERSC Perlmutter



UNIVERSITY OF TORONTO

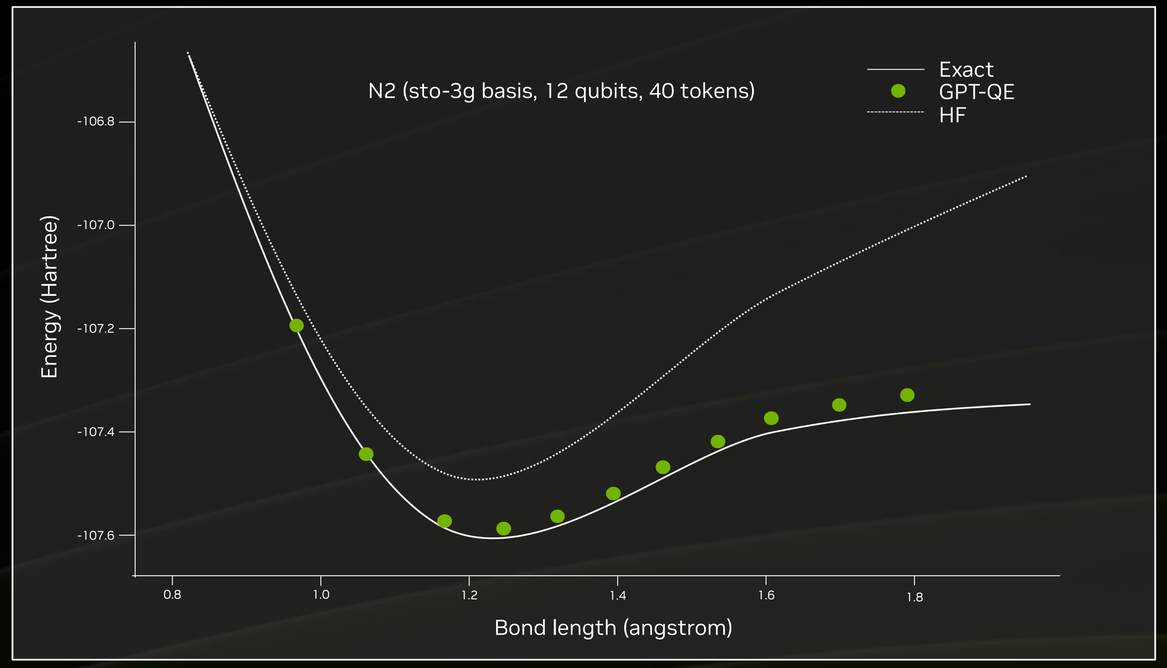
NVIDIA

St. Jude Children's Research Hospital

Generative AI + Quantum Algorithms

University of Toronto, St Jude's, and NVIDIA partner to invent GPT-QE

- Generative Pre-Trained Transformer-based (GPT) method for computing the ground state energies
- First GPT-generated quantum circuit
- Run via CUDA-Q on NERSC Perlmutter



S62497 – Combining Machine Learning with Quantum Computing for a New Generation of Algorithms

Alan Aspuru-Guzik, Univ of Toronto

UNIVERSITY OF TORONTO

NVIDIA

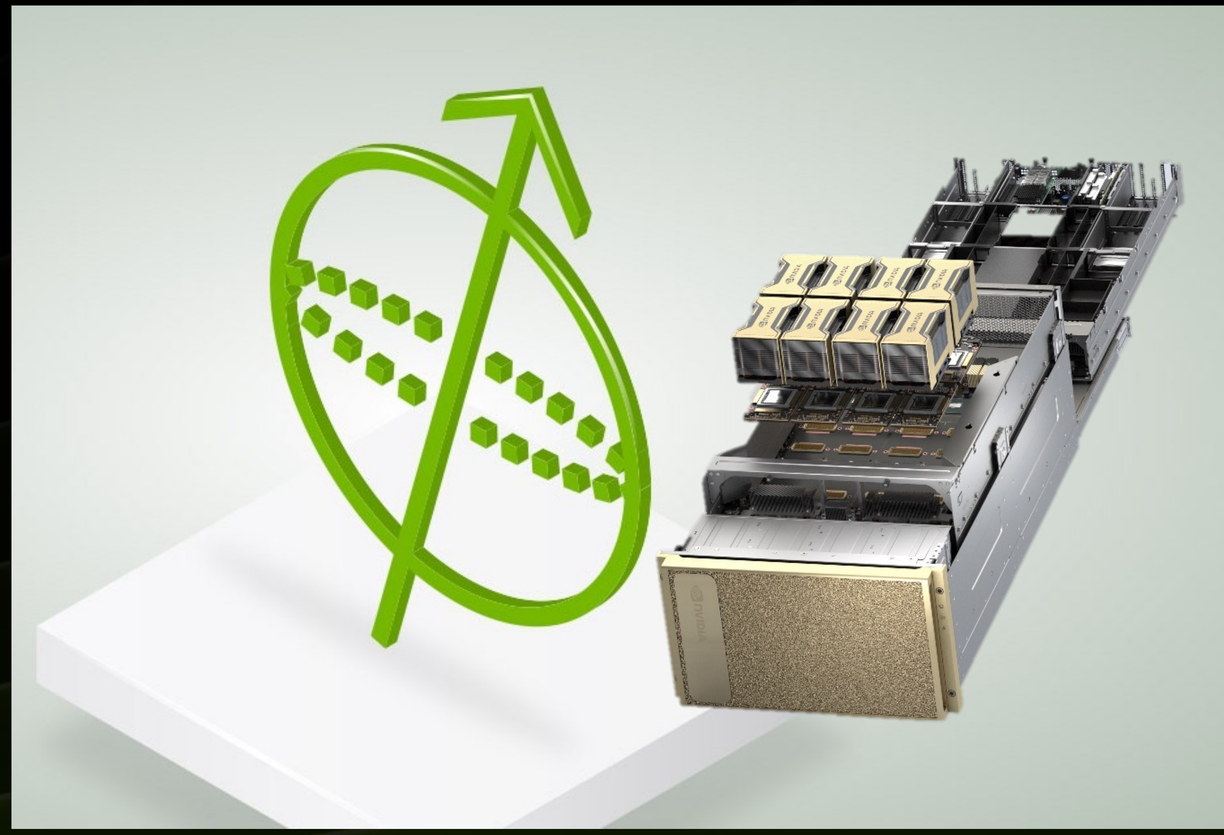
St. Jude Children's Research Hospital

SUMMARY

- **cuQuantum:** An SDK designed to accelerate and scale quantum circuit simulations, enabling researchers to simulate larger quantum systems and explore new capacities in quantum computing.

- **CUDA-Q**: An open-source programming model and compiler toolchain for integrating and programming across quantum processing units (QPUs), GPUs, and CPUs

- **DGX-Q:** The first unified system for integrated quantum-classical computing, developed by NVIDIA and Quantum Machines. It combines NVIDIA's Grace Hopper Superchip with OPX for ultra-low latency quantum control, offering am optimized platform for quantum research




NVIDIA