Quantum Accelerated Supercomputing

A Step-by-Step Guide to Expanding Simulation Capabilities and Enabling Interoperability of Quantum Hardware

Monica VanDieren
NVIDIA

Jin-Sung Kim
NVIDIA

Ermal Rrapaj
NERSC at LBNL

Katie Klymko
NERSC at LBNL

Zohim Chandani
NVIDIA

Tommaso Macri
QuEra

Agenda

Morning Sessions

- Introductions
- Overview of CUDA-Q
- Onboarding to Perlmutter
- Accelerating Quantum Simulations Notebook
- NVIDIA Quantum Cloud and Local install of CUDA-Q
- Intro to large scale clusters, how to navigate and use them

Afternoon Sessions

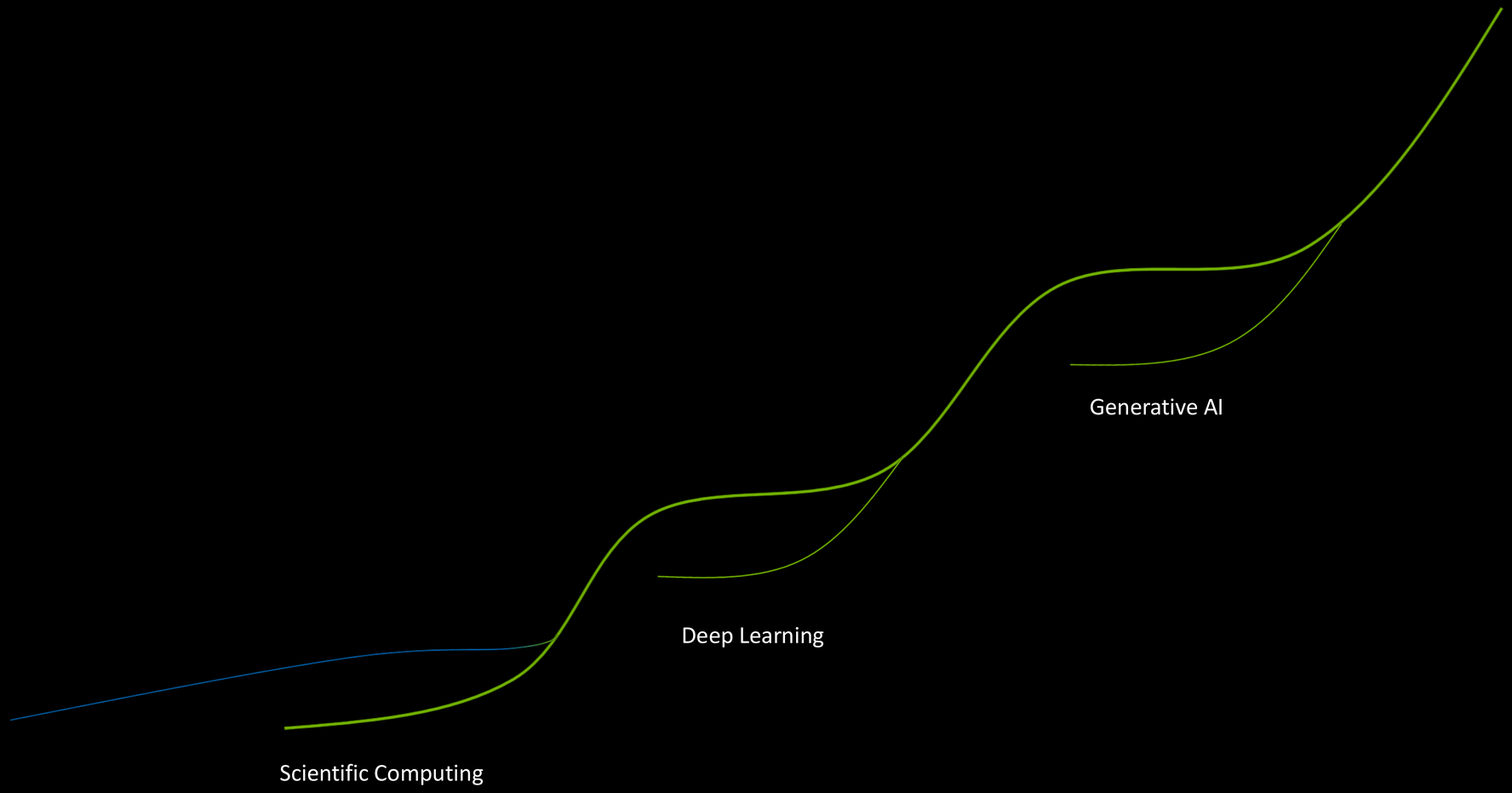
- Example: Quantum chemistry and nuclear physics at NERSC
- Industry Use Case: Simulating Hamiltonians of molecules with 30,000 terms
- QuEra’s Quantum Hardware
- Example: Quantum reservoir computing

CUDA-Q Overview

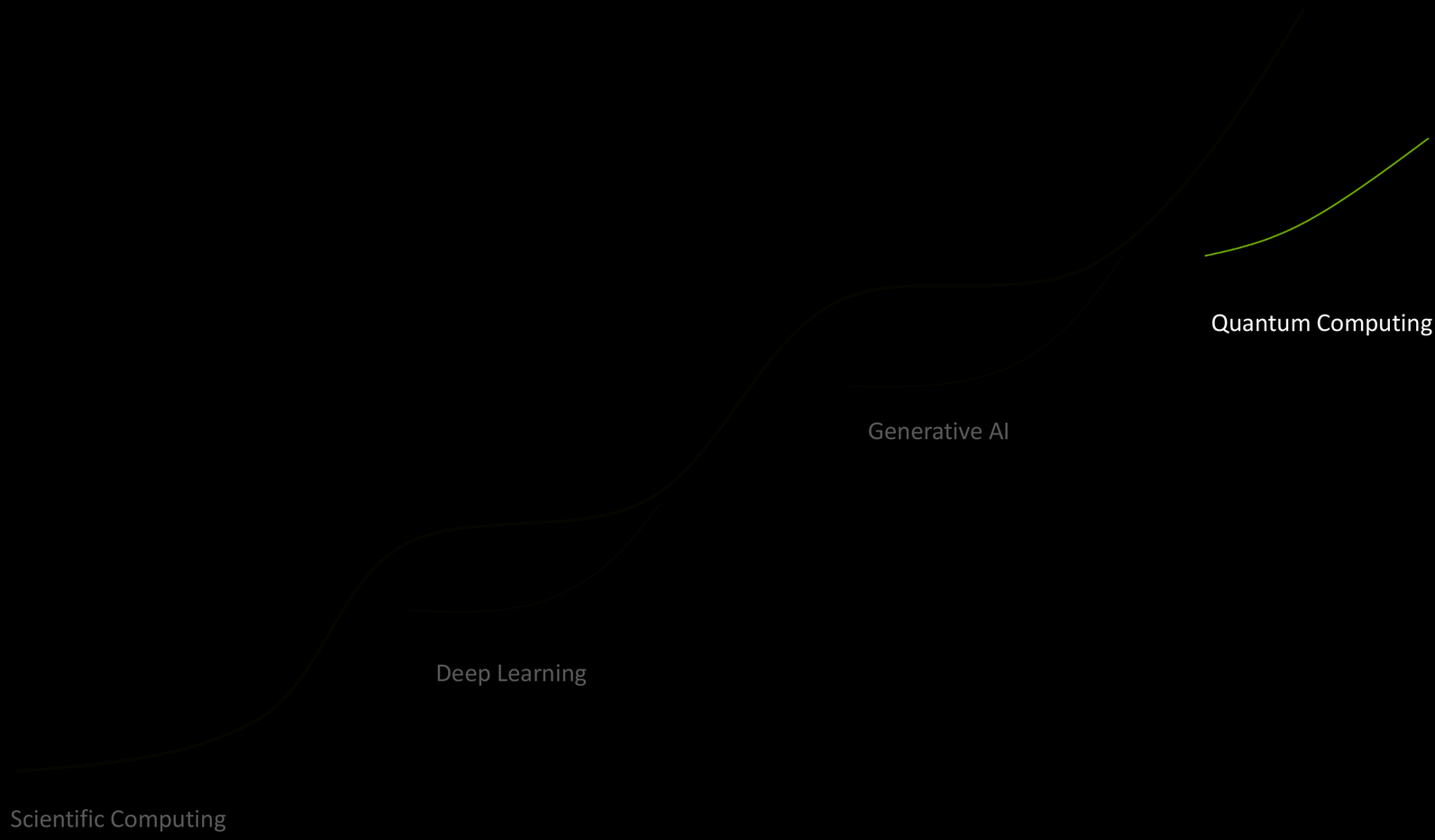
Agenda

- Accelerated Quantum Supercomputer
- CUDA-Q Platform
- Accelerating Quantum Simulations with GPUs
- Installing CUDA-Q and the NVIDIA Quantum Cloud
- CUDA-Q Resources

NVIDIA’s History of Enabling Computing Revolutions

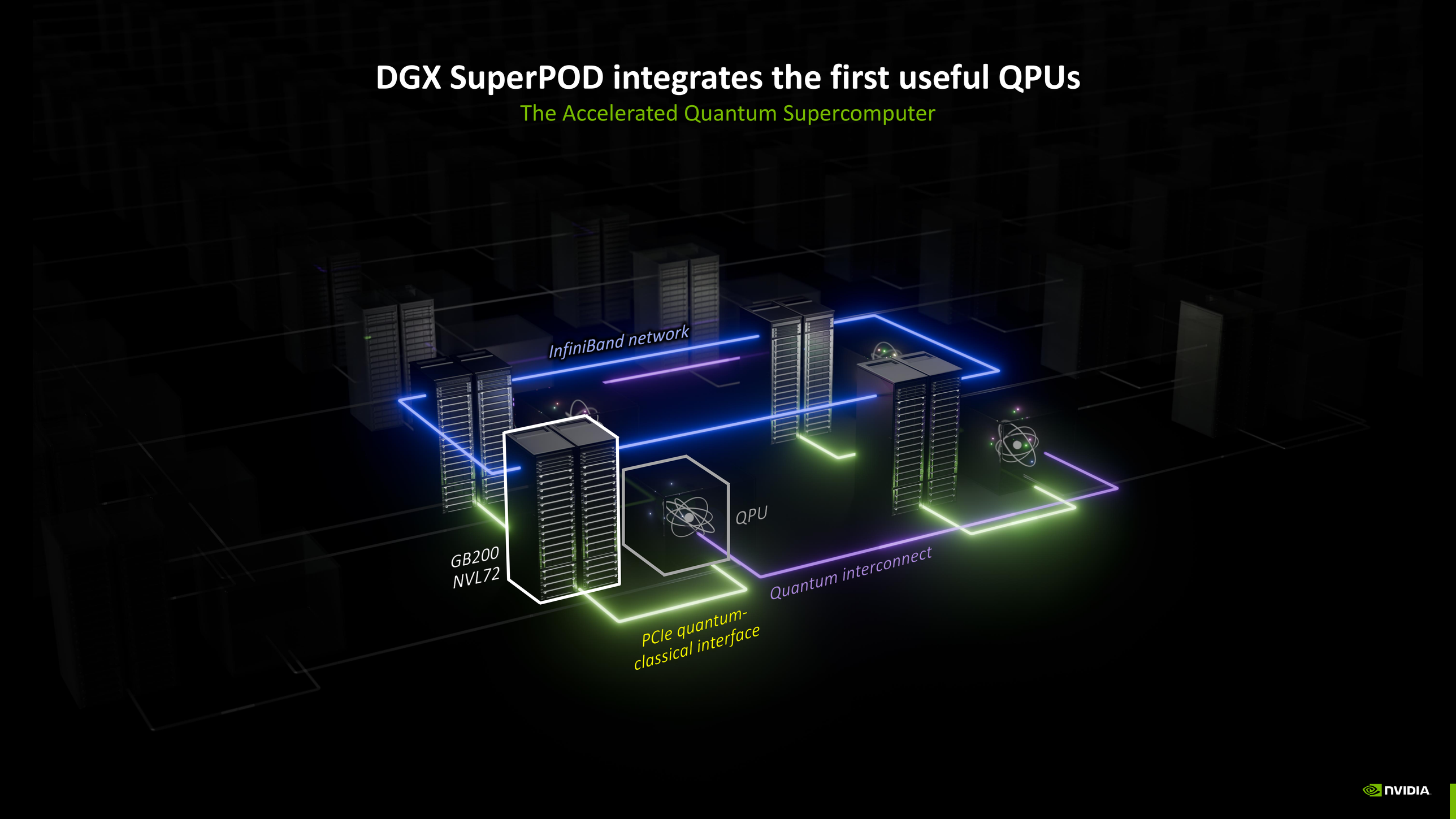


NVIDIA’s History of Enabling Computing Revolutions



DGX SuperPOD integrates the first useful QPUs

The Accelerated Quantum Supercomputer



DGX SuperPOD Integrates the First Useful QPUs

The Accelerated Quantum Supercomputer

- DGX SuperPODs **connect quantum hardware as a coprocessor**
- **Hybrid algorithms** need GPUs and QPUs (Quantum Processing Units)
- NVIDIA's **CUDA-Q software platform** connects applications seamlessly
- NVIDIA solutions **de-risk quantum** by being qubit-agnostic

Agenda

- Accelerated Quantum Supercomputer
- CUDA-Q Platform
- Accelerating Quantum Simulations with GPUs
- Installing CUDA-Q and the NVIDIA Quantum Cloud
- CUDA-Q Resources

NVIDIA Quantum

Enabling the quantum ecosystem

QPU vendors | **App developers** | **Academics** | • • •

CUDA-Q

Libraries | Programming Model | Tools | Infrastructure

cuQuantum

Quantum Simulation

DGX-Q

Quantum Integrated Computing

CUDA-X

World’s fastest AI and HPC libraries

NVIDIA

Agenda

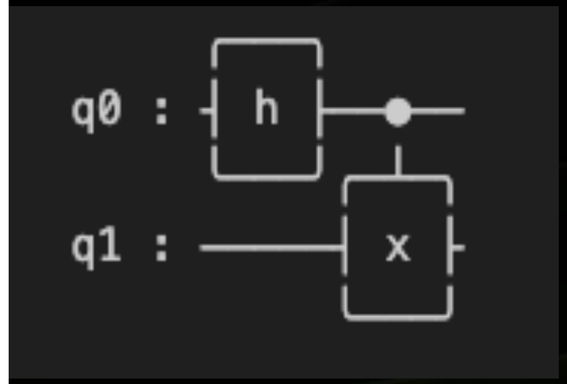
- Accelerated Quantum Supercomputer
- CUDA-Q Platform
- Accelerating Quantum Simulations with GPUs
- Installing CUDA-Q and the NVIDIA Quantum Cloud
- CUDA-Q Resources

Github Repository with tutorial notebooks:
https://github.com/NERSC/sc24-quantum-tutorial

Onboarding Perlmutter

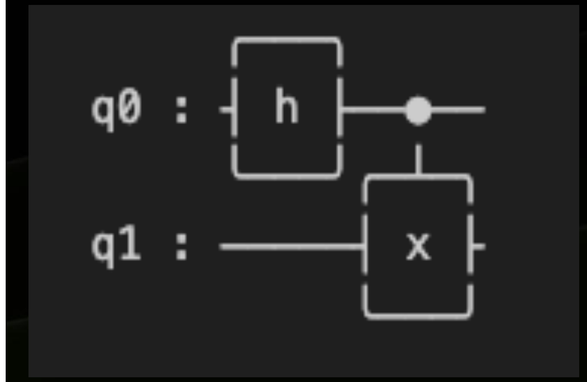
Building your First CUDA-Q Kernel

```python
import cudaq

qubit_count = 2

# Define the kernel
@cudaq.kernel
def my_kernel(qubit_count: int):
  # Allocate our `qubit_count` to the kernel.
  qubits = cudaq.qvector(qubit_count)

  # Apply a Hadamard gate to the qubit indexed by 0.
  h(qubits[0])

  # Apply a Controlled-X gate between qubit 0 (acting
  # as the control) and each of the remaining qubits.
  for i in range(qubit_count - 1):
    x.ctrl(qubits[i], qubits[i + 1])

  # Measure the qubits
  # If we don't specify measurements, all qubits are measured in the Z-basis by default.
  mz(qubits)

print(cudaq.draw(my_kernel, qubit_count))
```



Sampling your First CUDA-Q Kernel

```
# First set the backend for kernel execution
cudaq.set_target('qpp-cpu') # selects a CPU backend

if cudaq.num_available_gpus() > 0:
    cudaq.set_target(`nvidia') # selects a GPU backend

# cudaq.set_target('nvqc') # selects the NVIDIA Quantum Cloud
# cudaq.set_target('ionq')  # select an available QPU backend

qubit_count = 2

results =  cudaq.sample(my_kernel, qubit_count, shots_count = 10000)

print(results) # Example: {00:5005, 11: 4995}

print(results.most_probable()) # prints: `00`

print(results.probability(results.most_probable())) # prints: `0.5005`
```



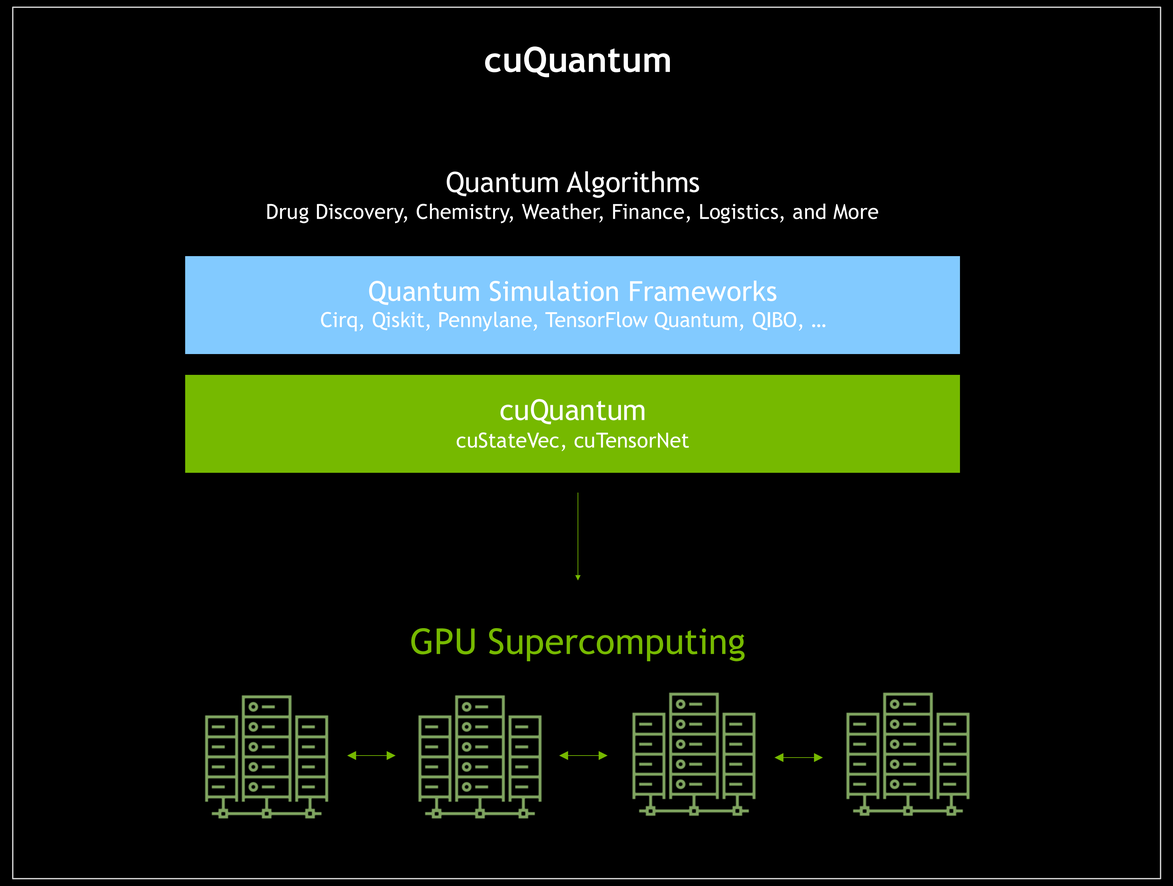
Accelerating Quantum Simulations

Jupyter Notebook on Github

https://github.com/NERSC/sc24-quantum-tutorial

cuQuantum

Research the Quantum Computer of Tomorrow on the most Powerful Computer Today



CUQUANTUM FEATURES

SDK for GPU Accelerated Quantum Simulation

Simulate Ideal or Noisy Qubits with State Vector or Tensor Network methods

Supports GPU Supercomputing with Multi-Node Multi-GPU Circuit Simulation

Accelerates all leading frameworks

Optimized frameworks in cuQuantum Appliance: catalog.ngc.nvidia.com/orgs/nvidia/containers/cuquantum-appliance

aws blueqat Cirq PENNYLANE QIBO Qiskit Torch Quantum

cuStateVec

Part of NVIDIA cuQuantum SDK

- cuStateVec: a library to accelerate statevector-based quantum circuit simulation
  - Most computations are “in-place” to reduce memory usage
  - Provides low-level primitives to cover common use cases:
    1) **Apply gate matrix**
    2) Apply diagonal/general permutation matrix
    3) Apply exponential of Pauli matrix product
    4) Expectation using matrix as observable
    5) Expectation on Pauli basis
    6) Sampling
    7) Measurement on a Z-product basis
    8) Batched single qubit measurement
    9) State vector segment extraction/update
    10) **Qubit reordering on single/multiple device(s)**
  - Easy integration & adoption for a wide variety of frameworks & programming languages
  - Also available in the cuQuantum Appliance container (standalone & Cirq/Qsim backend)

C API

```
custatevecStatus_t custatevecApplyMatrix(
    custatevecHandle_t handle,
    void *sv,
    cudaDataType_t svDataType,
    const uint32_t nIndexBits,
    const void *matrix,
    cudaDataType_t matrixDataType,
    custatevecMatrixLayout_t layout,
    const int32_t adjoint,
    const int32_t *targets,
    const uint32_t nTargets,
    const int32_t *controls,
    const int32_t *controlBitValues,
    const uint32_t nControls,
    custatevecComputeType_t computeType,
    void *extraWorkspace,
    size_t extraWorkspaceSizeInBytes)
```

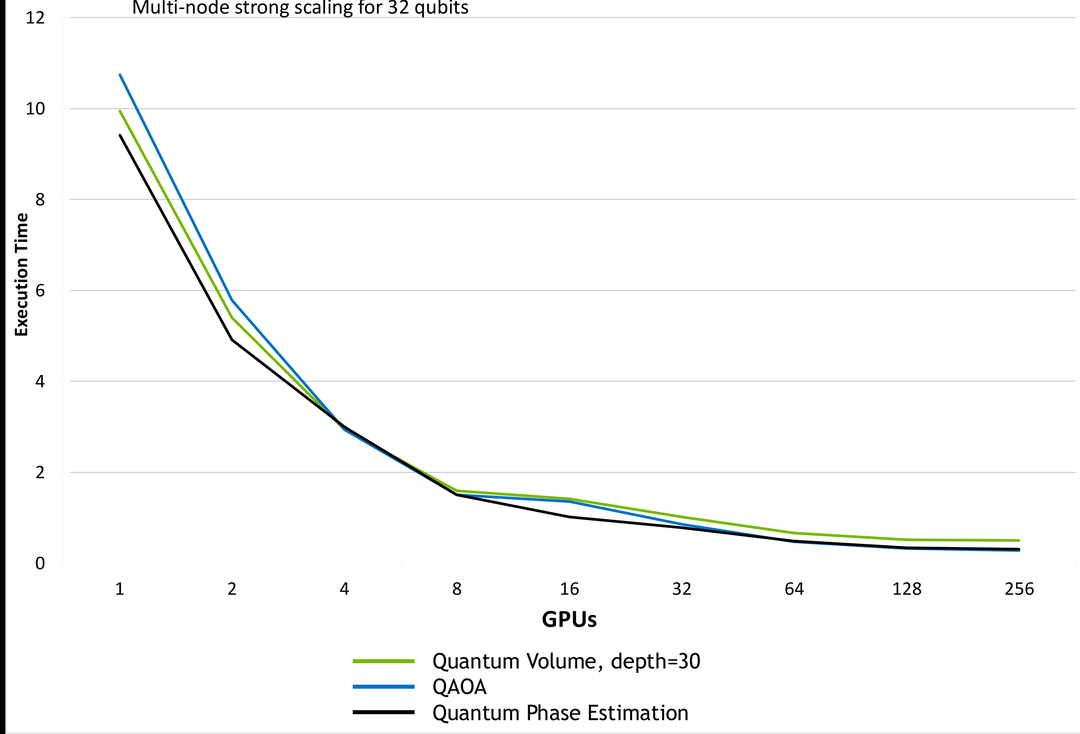
Python API

```
cuquantum.custatevec.apply_matrix(
    handle,
    sv,
    sv_data_type,
    n_index_bits,
    matrix,
    matrix_data_type,
    layout,
    adjoint,
    targets,
    n_targets,
    controls,
    control_bit_values,
    n_controls,
    compute_type,
    workspace,
    workspace_size)
```

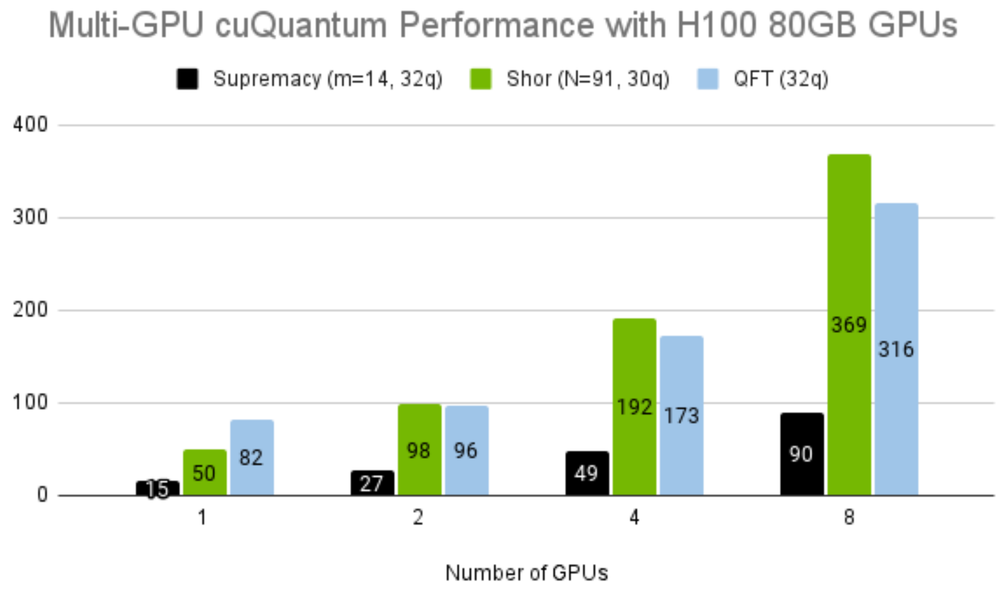
cuStatevec

Research the Quantum Computer of Tomorrow on the most Powerful Computer Today

World Class Performance with Now with Multi-Node Multi-GPU



World Class Performance

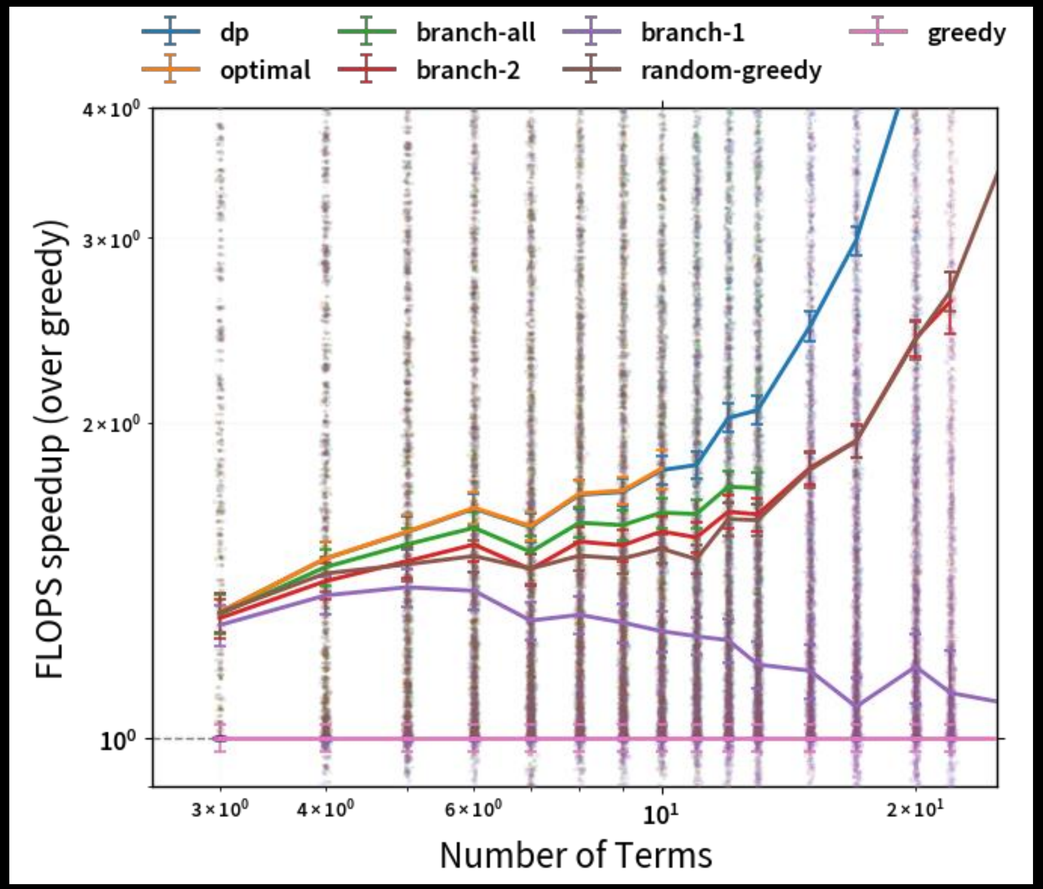


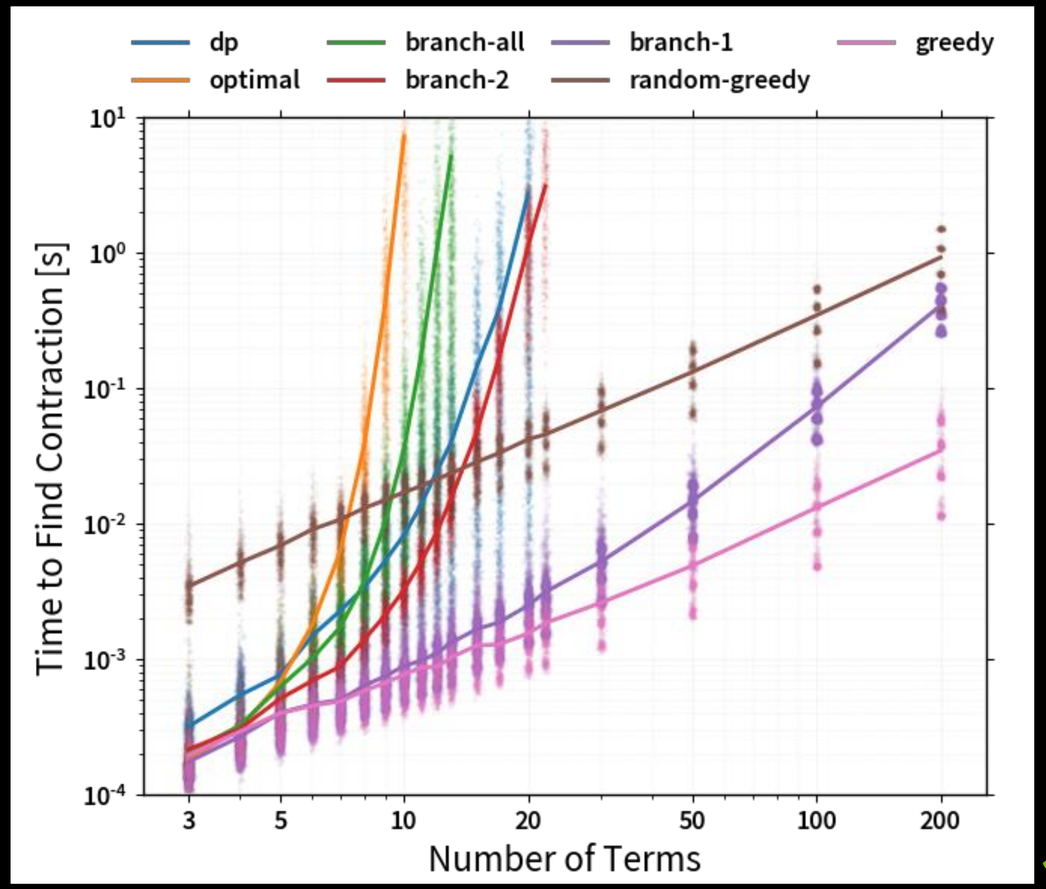
cuTensorNet

A Library to accelerate tensor network based quantum circuit simulation

Motivation: What is contraction path optimization and why is it needed ?

- Reordering the contraction is important to minimize flops as well as memory requirements.
- Finding the optimal path is NP hard problem, which is why they only can be used for small number of tensors (terms)



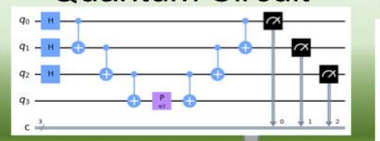


cuTensorNet

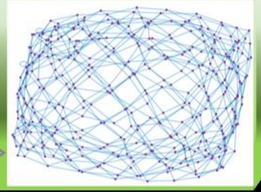
A Library to accelerate tensor network based quantum circuit simulation

Quantum Algorithm

Quantum Circuit



Tensor Network



cuTensorNet

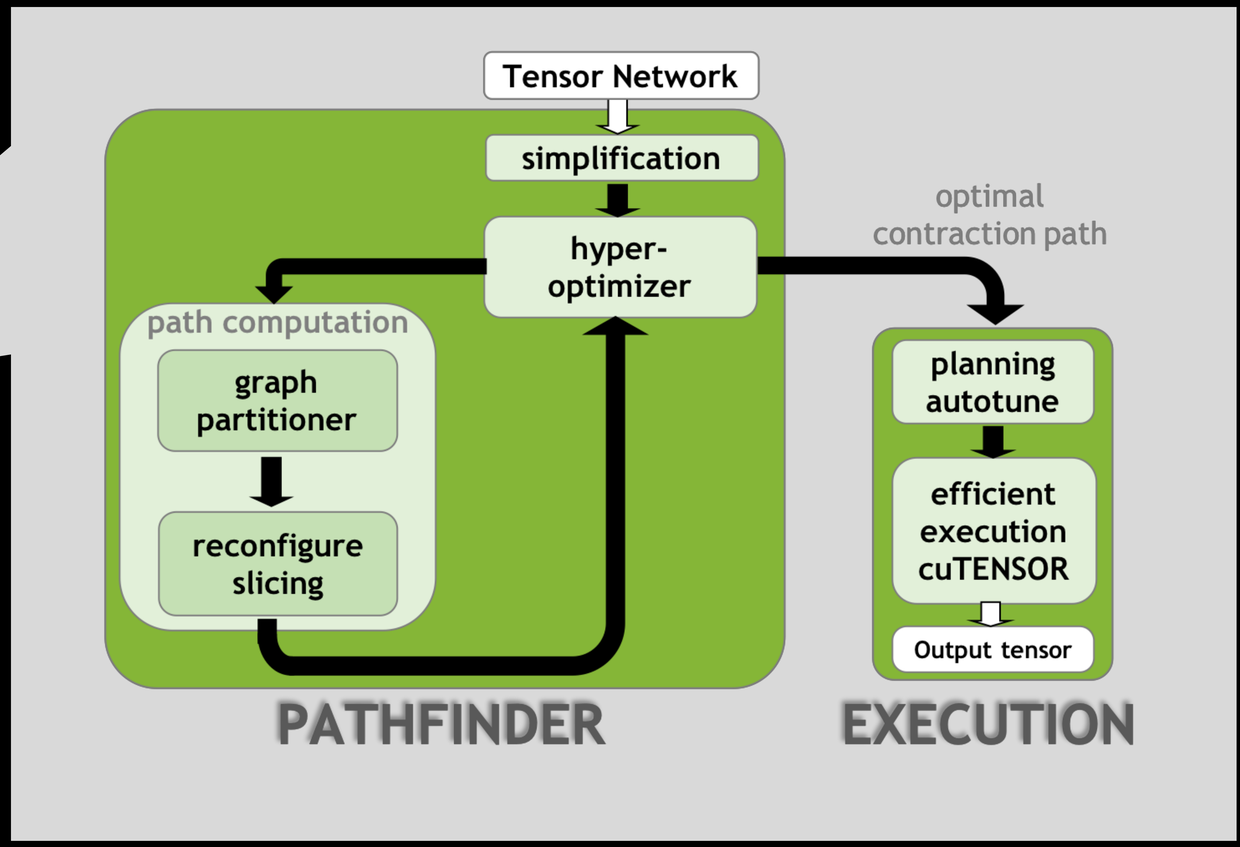
Python APIs

C APIs

Algorithmic advancements	Optimal contraction	Kernels optimization

NVIDIA GPU

Supercomputer



cuTensorNet

Research the Quantum Computer of Tomorrow on the most Powerful Computer Today

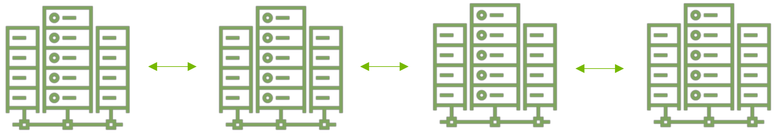
cuQuantum

Quantum Algorithms
Drug Discovery, Chemistry, Weather, Finance, Logistics, and More

Quantum Simulation Frameworks
Cirq, Qiskit, Pennylane, TensorFlow Quantum, QIBO, …

cuQuantum
cuStateVec, cuTensorNet

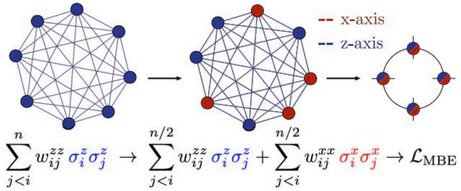
GPU Supercomputing



Supercomputing Simulation at FTQC Scale

Perfect Qubit Count

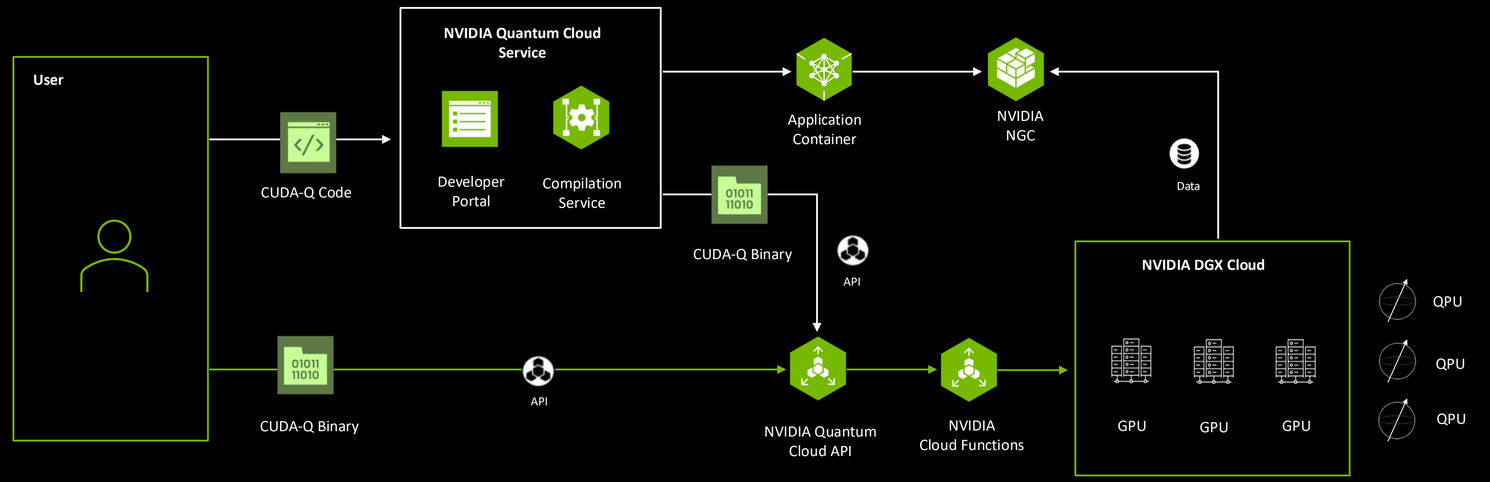
MaxCut Problem



[1] Danylo Lykov et al, Tensor Network Quantum Simulator With Step-Dependent Parallelization, 2020
https://arxiv.org/pdf/2012.02430.pdf
[2] Taylor Patti et al, Variational Quantum Optimization with Multibasis Encodings, 2022
https://arxiv.org/abs/2106.13304

NVIDIA Quantum Cloud

Seamless Access to the most Powerful Quantum Resources

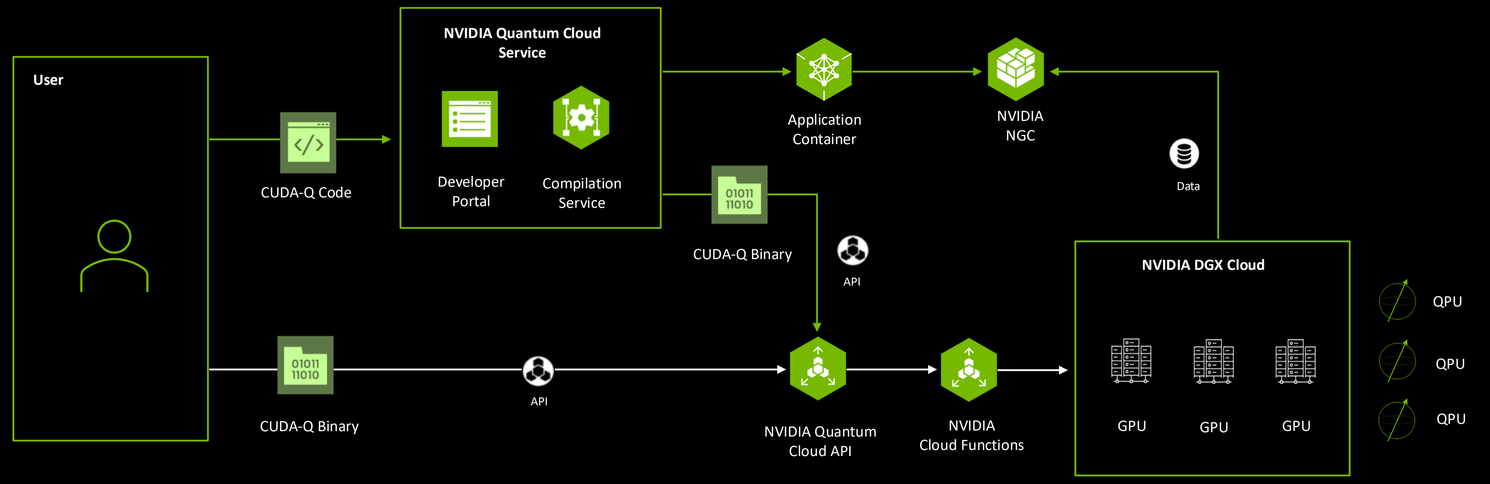


- Access to the most powerful Quantum Resources
- Develop locally, run any CUDA-Q app seamlessly in Cloud
- Call pre-built Quantum Cloud APIs from your application
- Run workloads on GPU Supercomputers
- Integrated ISV applications
- EA Available Today! Apply for Early Access at developer.nvidia.com/quantum-cloud-early-access-join

- QPU Partner Integrations coming soon
- Quantum Cloud Web Developer Portal coming soon

NVIDIA Quantum Cloud

Seamless Access to the most Powerful Quantum Resources

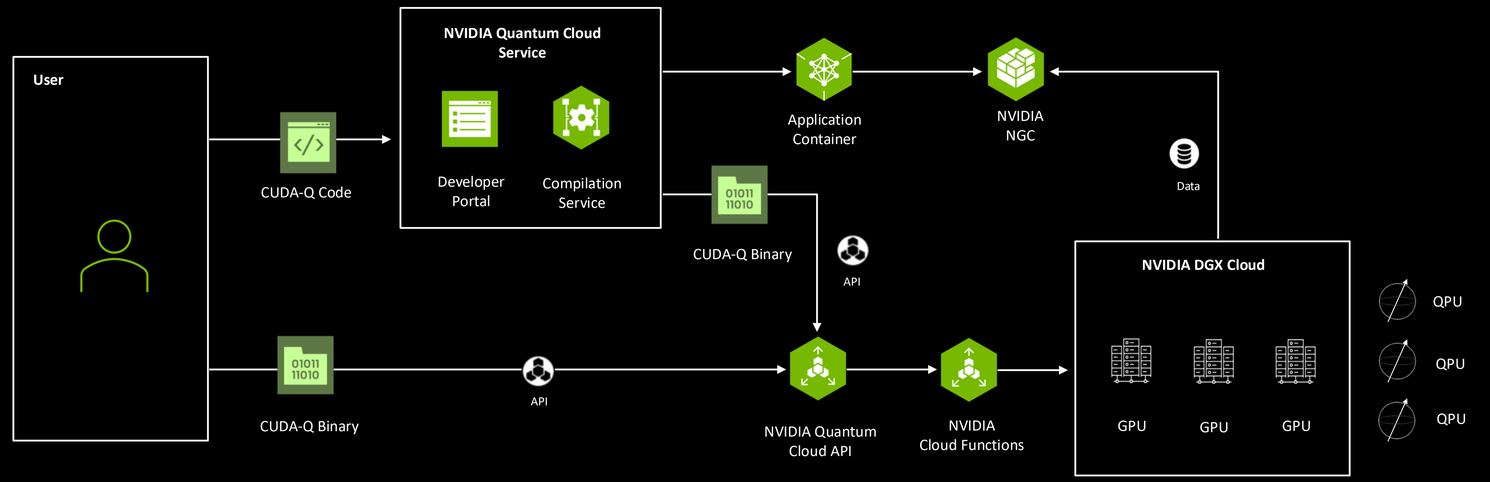


- Access to the most powerful Quantum Resources
- Develop locally, run any CUDA-Q app seamlessly in Cloud
- Call pre-built Quantum Cloud APIs from your application
- Run workloads on GPU Supercomputers
- Integrated ISV applications
- EA Available Today! Apply for Early Access at developer.nvidia.com/quantum-cloud-early-access-join

- QPU Partner Integrations coming soon
- Quantum Cloud Web Developer Portal coming soon

NVIDIA Quantum Cloud

Seamless Access to the most Powerful Quantum Resources



algorithmiq · ALICE & BOB · BlueQubit · CLASSIQ · Infleqtion · IONQ · IQM

OQC · ORCA Computing · PENNYLANE · qBraid · QCWARE · QM ware · QUANTINUUM

QUANTUM BRILLIANCE · Qubit PHARMACEUTICALS · IQuEra · rigetti · seeqc · Terra Quantum · ZAPATA AI

NVIDIA

Find out more

NVIDIA Quantum
https://www.nvidia.com/en-us/solutions/quantum-computing/

CUDA-Q Academic

Educational resources for CUDA-Q
(https://github.com/NVIDIA/cuda-q-academic)

NVIDIA Quantum Cloud

Early Access - https://www.nvidia.com/en-us/solutions/quantum-computing/cloud/


NVIDIA

Agenda

Morning Sessions

- Introductions
- Overview of CUDA-Q
- Onboarding to Perlmutter
- Accelerating Quantum Simulations Notebook
- NVIDIA Quantum Cloud and Local install of CUDA-Q
- Intro to large scale clusters, how to navigate and use them

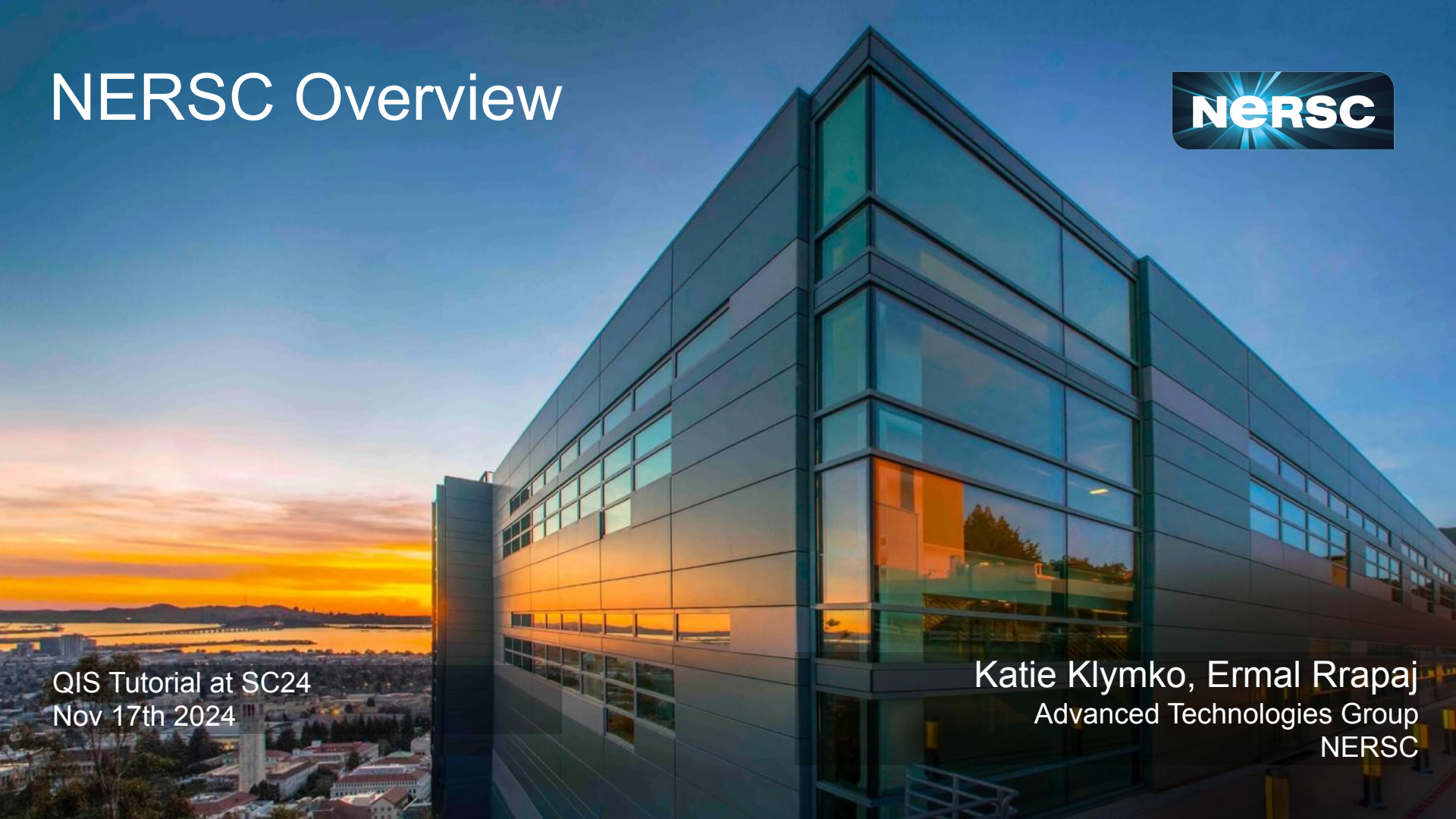
Afternoon Sessions

- Example: Quantum chemistry and nuclear physics at NERSC
- Industry Use Case: Simulating Hamiltonians of molecules with 30,000 terms
- QuEra’s Quantum Hardware
- Example: Quantum reservoir computing

Outline for NERSC Section of the CUDA-Q Tutorial

- Description of NERSC systems
- How to access NERSC training accounts
- Physics and Chemistry Variational Quantum Eigensolver examples: https://github.com/NERSC/sc24-quantum-tutorial

NERSC Overview



QIS Tutorial at SC24
Nov 17th 2024

Katie Klymko, Ermal Rrapaj
Advanced Technologies Group
NERSC

National Energy Research Scientific Computing Center

- NERSC is a national supercomputer center funded by the U.S. Department of Energy Office of Science (SC)
  - Supports SC research mission
  - Part of Berkeley Lab
- Researchers with funding from SC who need supercomputing resources can use NERSC
  - Other researchers can apply if research is in SC mission
- NERSC supports 10,000 users, 1,000 projects
  - From all 50 states + international; 60% from universities
  - Hundreds of users log on each day

NERSC User Demographics

~10,000 Annual Users from ~800 Institutions + National Labs

32% Graduate Students

19% Postdoctoral Fellows

15% Staff Scientists

13% University Faculty

8% Undergraduate Students

5% Professional Staff

60% Universities

29% DOE Labs

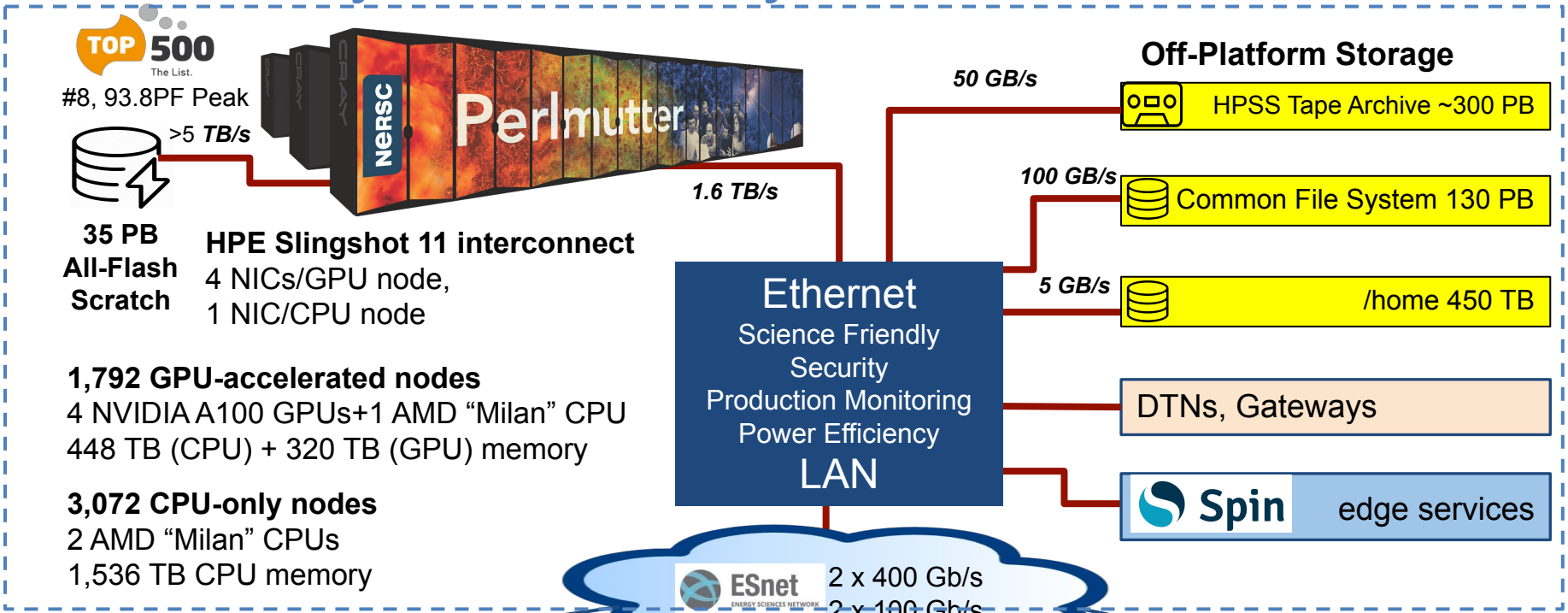
5% Other Government Labs

4% Industry

1% Small Businesses

<1% Private Labs

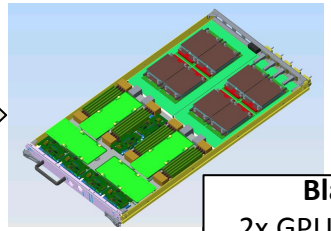
NERSC Systems Ecosystem

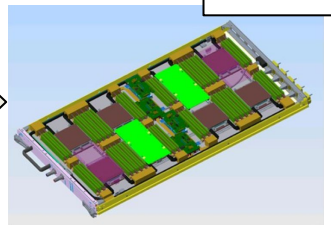


Perlmutter system configuration

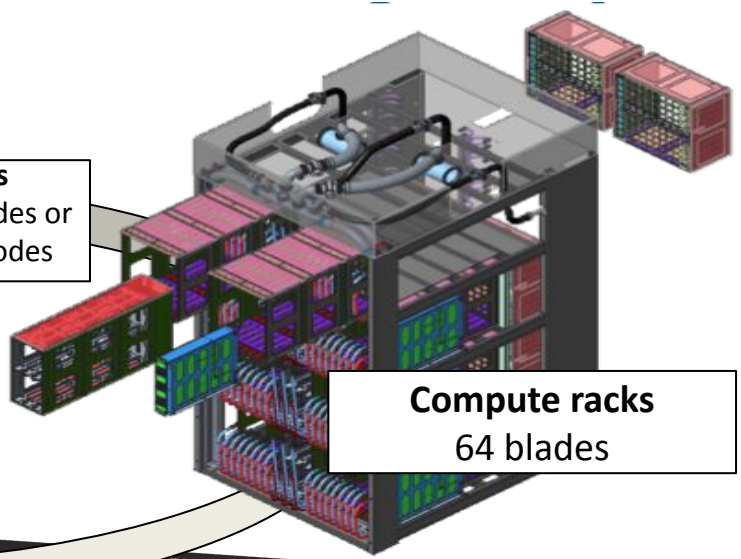
NVIDIA "Ampere" GPU Nodes
4x GPU + 1x CPU
40 GiB HBM + 256 GiB DDR
4x 200G "Slingshot" NICs

AMD "Milan" CPU Node
2x CPUs
> 256 GiB DDR4
1x 200G "Slingshot" NIC





Blades
2x GPU nodes or
4x CPU nodes

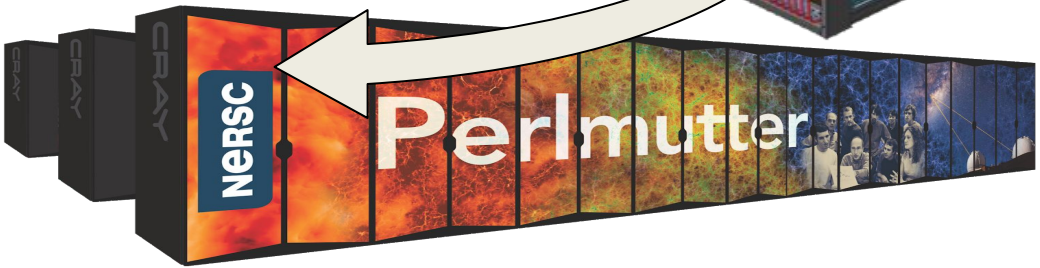


Compute racks
64 blades

Centers of Excellence
Network
Storage
App. Readiness
System SW

→

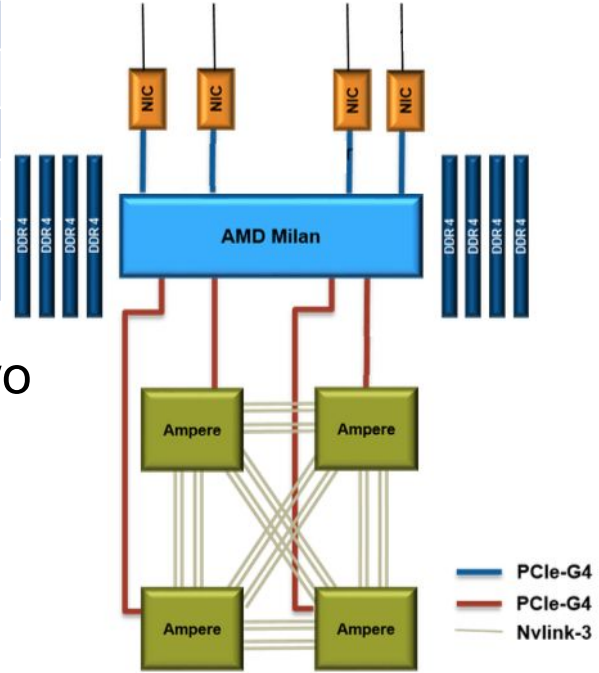
Perlmutter system
GPU racks
CPU racks
~6 MW



Node Types

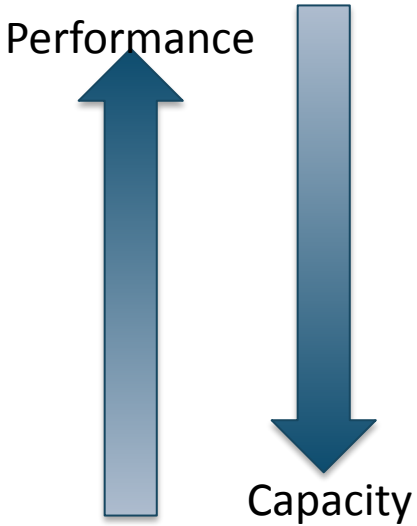
Partition	Nodes	CPU	RAM	GPU	NIC
GPU	1536	1x AMD EPYC 7763	256GB	4x NVIDIA A100 (40GB)	4x HPE Slingshot 11
	256	1x AMD EPYC 7763	256GB	4x NVIDIA A100 (80GB)	4x HPE Slingshot 11
CPU	3072	2x AMD EPYC 7763	512GB	–	1x HPE Slingshot 11
Login	40	1x AMD EPYC 7713	512GB	4x NVIDIA A100 (40GB)	–
Large Memory	4	1x AMD EPYC 7713	1TB	4x NVIDIA A100 (40GB)	1x HPE Slingshot 11

GPU Node Architecture:



- Each Milan CPU has 64 2.45 GHz cores with two hardware threads
- Access is managed via Slurm partitions
- Login nodes are GPU-enabled

Simplified NERSC File Systems



Memory

Burst Buffer

Scratch

Community

HPSS

Global Common

Global Home

35 PB Flash Scratch
Lustre >5 TB/s
temporarily (purge)
xGB (x=Total RAM) In-Memory Burst Buffer
/tmp RamFS
157 PB HDD Community
Spectrum Scale (GPFS)
150 GB/s, permanent
150 PB Tape Archive
HPSS Forever
20 TB SSD Software
Spectrum Scale
Permanent
Faster compiling / Source Code

Global File Systems

Global Home

- Permanent, relatively small storage
- Mounted on all platforms
- NOT tuned to perform well for parallel jobs
- Quota cannot be changed
- Snapshot backups (7-day history)
- **Perfect for storing data such as source code, shell scripts**
- **Addressed using `$HOME`**

Community File System (CFS)

- Permanent, larger storage
- Mounted on all platforms
- Medium performance for parallel jobs
- Quota can be changed
- Snapshot backups (7-day history)
- **Perfect for sharing data within research group**
- **Addressed using `$CFS`**

Local File Systems

Scratch

- Large, temporary storage
- Optimized for read/write operations, NOT storage
- Not backed up
- Purge policy (8 weeks)
- **Perfect for staging data and performing computations**
- **Addressed using `$SCRATCH`**

Burst Buffer

- Temporary storage
- High-performance in-memory file system
- **Perfect for getting good performance in I/O-constrained codes**
- Not shared between nodes

There are many different ways to access NERSC. To use our resources, you need to either:

1. Log into the login nodes and interact with Slurm (covered here)
2. Use services (eg. Jupyter) that expose web interfaces (covered later)
3. Interact via our REST API – called the “Superfacility API” (see: https://docs.nersc.gov/services/sfapi/)

Connecting with SSH

- To access Perlmutter, use:
  `saul.nersc.gov`
  `perlmutter.nersc.gov`

```
blaschke@laptop:~$ ssh <user>@saul.nersc.gov
blaschke@laptop:~$ ssh <user>@perlmutter.nersc.gov
```

- To be able to open GUI applications on the login nodes use: `ssh -Y`

```
blaschke@laptop:~$ ssh -Y <user>@saul.nersc.gov
```

Connecting with SSH

After successfully logging on, you be greeted by the terms of use, and the command-line input prompt:

```
For past outages, see: https://my.nersc.gov/outagelog-cs.p
blaschke@perlmutter:login17:~>
```

From here you can interact with perlmutter…

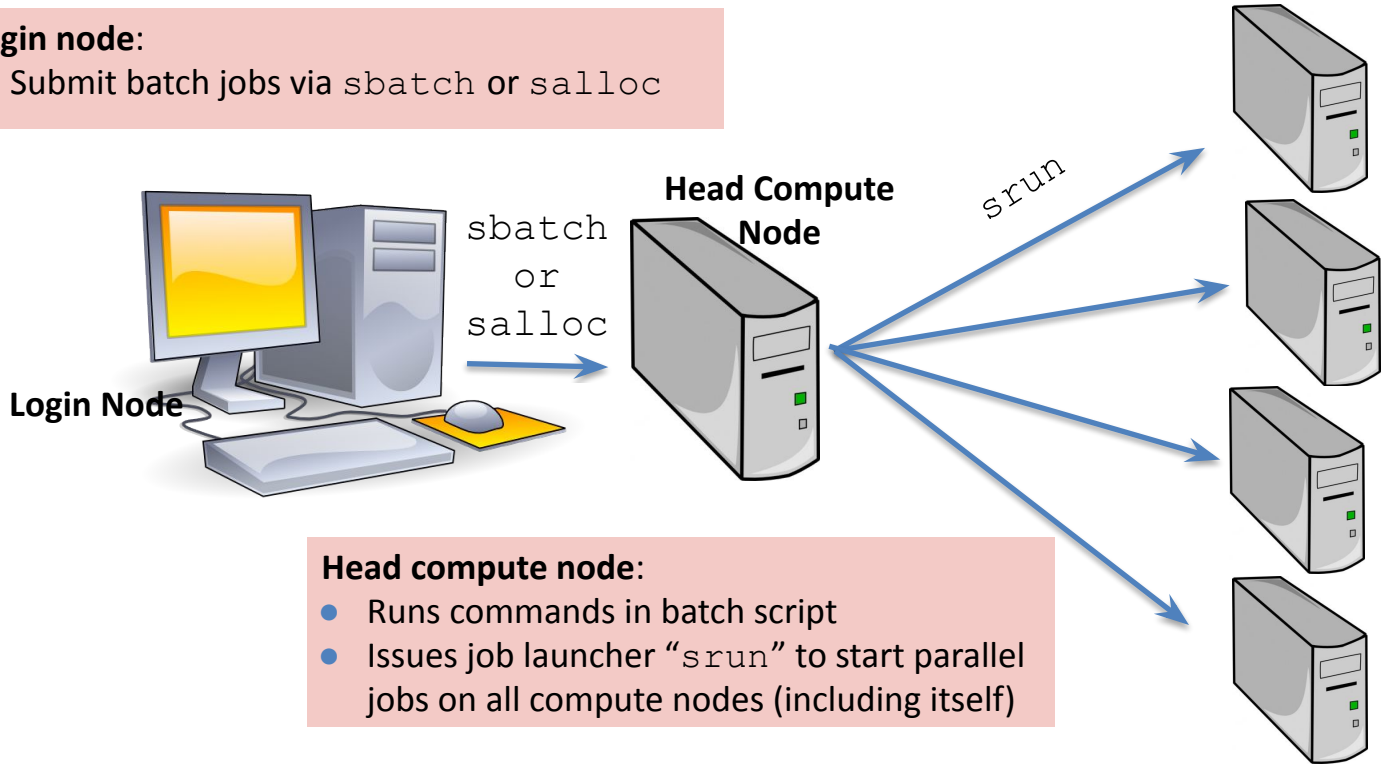
```
blaschke@laptop:~$ ssh blaschke@saul.nersc.gov
********************************************************************************
                        NOTICE TO USERS

Lawrence Berkeley National Laboratory operates this computer system under
contract to the U.S. Department of Energy.  This computer system is the
property of the United States Government and is for authorized use only.
Users (authorized or unauthorized) have no explicit or implicit
expectation of privacy.

Any or all uses of this system and all files on this system may be
intercepted, monitored, recorded, copied, audited, inspected, and disclosed
to authorized site, Department of Energy, and law enforcement personnel,
as well as authorized officials of other agencies, both domestic and foreign.
By using this system, the user consents to such interception, monitoring,
recording, copying, auditing, inspection, and disclosure at the discretion
of authorized site or Department of Energy personnel.

Unauthorized or improper use of this system may result in administrative
disciplinary action and civil and criminal penalties. By continuing to use
this system you indicate your awareness of and consent to these terms and
conditions of use. LOG OFF IMMEDIATELY if you do not agree to the conditions
stated in this warning.

********************************************************************************

Login connection to host x3114c0s15b0n0:


################################################################################
                _                                  _   _
               | |                                | | | |
 _ __   ___ _ __| |_ _ __ ___  _   _| |_| |_ _ __ ___ _ __
| '_ \ / _ \ '__| | '_ ` _ \| | | | __| __/ _ \ '__|
| |_) |  __/ |  | | | | | | | |_| | |_| ||  __/ |
| .__/ \___|_|  |_|_| |_| |_|\__,_|\__|\__\___|_|
| |
|_|

Hello, World!

Welcome to perlmutter!
################################################################################

For all planned outages, see: https://www.nersc.gov/live-status/motd/

For past outages, see: https://my.nersc.gov/outagelog-cs.php/
blaschke@perlmutter:login17:~>
```

Submitting Jobs

- Jobs can be submitted to queueing system through sbatch or salloc:
  - `sbatch <my_job_script>`
  - `salloc <options>`
- The above methods list details about resources needed for a job and for how long
- Eg.: a request for a cpu node to be allocated for 5 mins:
  - `salloc -N 1 -C cpu -q debug -t 5 -A <project>`

Submitting Jobs

Login node:
- Submit batch jobs via `sbatch` or `salloc`



Head compute node:
- Runs commands in batch script
- Issues job launcher "`srun`" to start parallel jobs on all compute nodes (including itself)

*figure courtesy Helen (2020 NERSC Training)

My First “Hello World” Job Script

debug queue

2 nodes

10 min “walltime”

run on haswell partition

use SCRATCH file system

run 64 processes in parallel

my_batch_script.sh

```
#!/bin/bash
#SBATCH -q debug
#SBATCH -N 2
#SBATCH -t 10:00
#SBATCH -C haswell
#SBATCH -L SCRATCH
#SBATCH -J myjob
srun -n 64 ./helloWorld
```

To run via batch queue

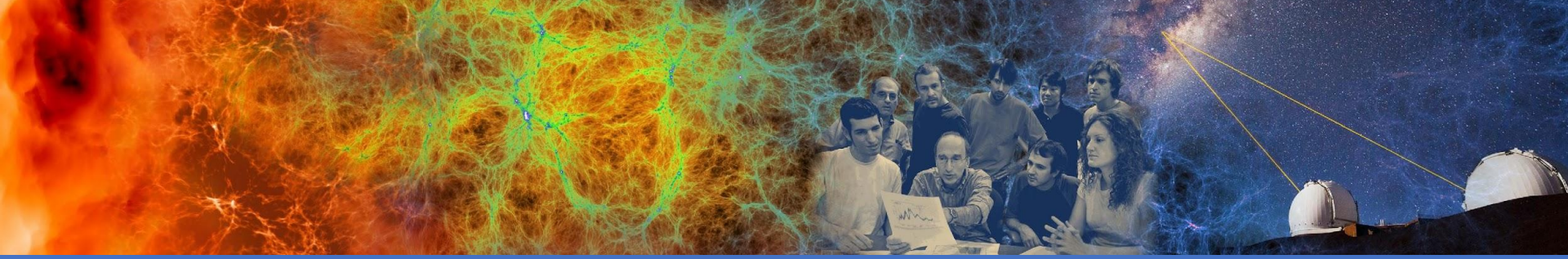
```
% sbatch my_batch_script.sh
```

To run via interactive batch

```
% salloc -N 2 -q interactive -C gpu -t 10:00
```

<wait_for_session_prompt. Land on a compute node>

```
% srun -n 64 ./helloWorld
```



NERSC Training Accounts

for *The CUDA-Q Tutorial On Perlmutter*

NERSC

BERKELEY LAB

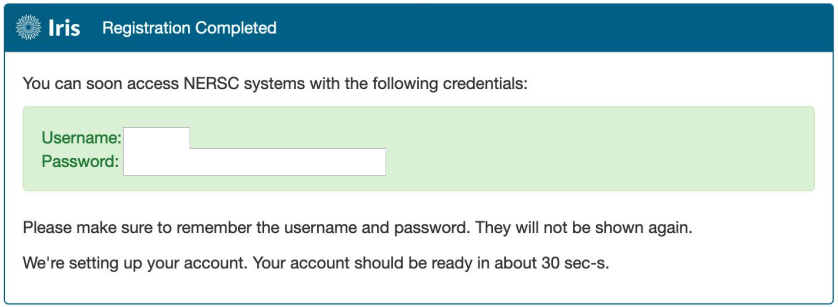
U.S. DEPARTMENT OF ENERGY | Office of Science

Activate your Training Account

- Go to: https://iris.nersc.gov/train

- Enter the Training Code: "*ep81*" along with your details
  Note: if you have previously registered an account using a different code please recreate it.

- **Important: make a copy of your login and password** you won't be able to change these, nor recover them later!!!



Access to Perlmutter and Use Cuda-Q

- NERSC users have been added to `ntrain10` project
- Non-users were sent the instruction to get a training account
  - Account valid through Nov 18th
- Login to Perlmutter: ssh username@perlmutter.nersc.gov
- Cuda modules:
  - `% module load cudatoolkit`
- Running Jobs examples:
  - https://docs.nersc.gov/jobs/

Compute Node Reservations

- GPU node reservation: Sun Nov 17th and Mon Nov 18th
  - To use 1 GPU only (sample flags for `sbatch` or `salloc`):
    - `-A ntrain1 --reservation=julia_sc24 -C gpu -N 1 -c 32 -G 1 -t 30:00 -q shared`
  - To use multiple nodes (sample flags for sbatch or salloc):
    - `-A ntrain1 --reservation=julia_sc24 -C gpu -N 2 -t 30:00 -q regular`
- Outside of reservation, use:
  - To use 1 GPU only (sample flags for sbatch or salloc):
    - `-A <project> -C gpu -N 1 -c 32 -G 1 -t 30:00 -q shared`
  - To use multiple nodes (sample flags for sbatch or salloc):
    - `-A <project> -C gpu -N 2 -t 30:00 -q regular` (or `-q interactive` for salloc)

Reminder about file Systems

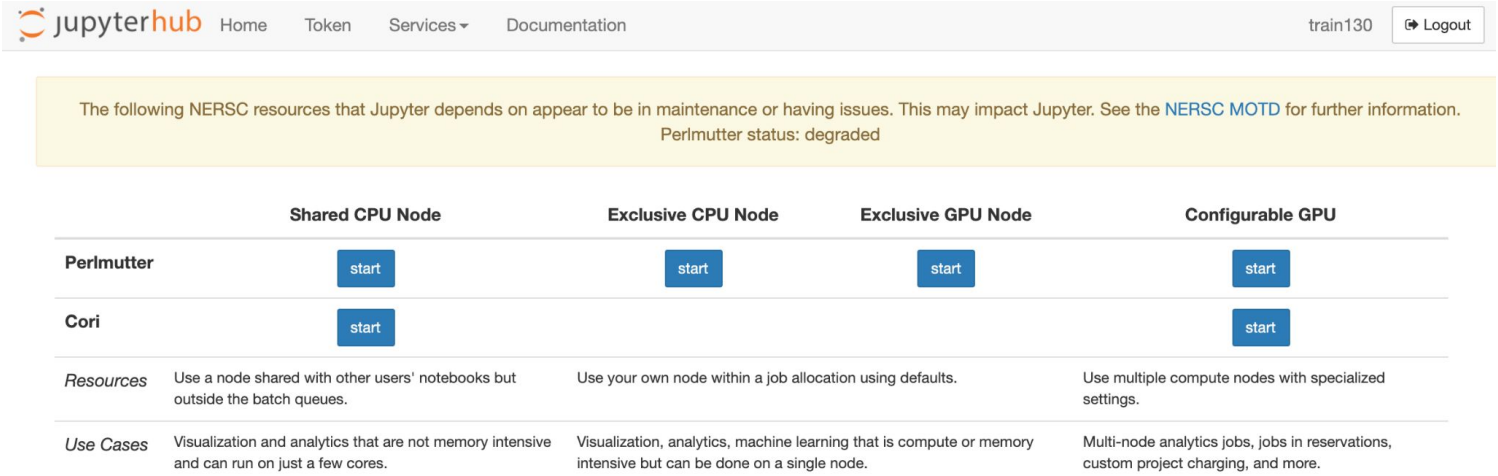
`$HOME` is not the best file system for running jobs at scale. Consider therefore replacing all occurrences of `$HOME` with `$SCRATCH`.

https://github.com/NERSC/sc24-quantum-tutorial

More detailed information in the github repository

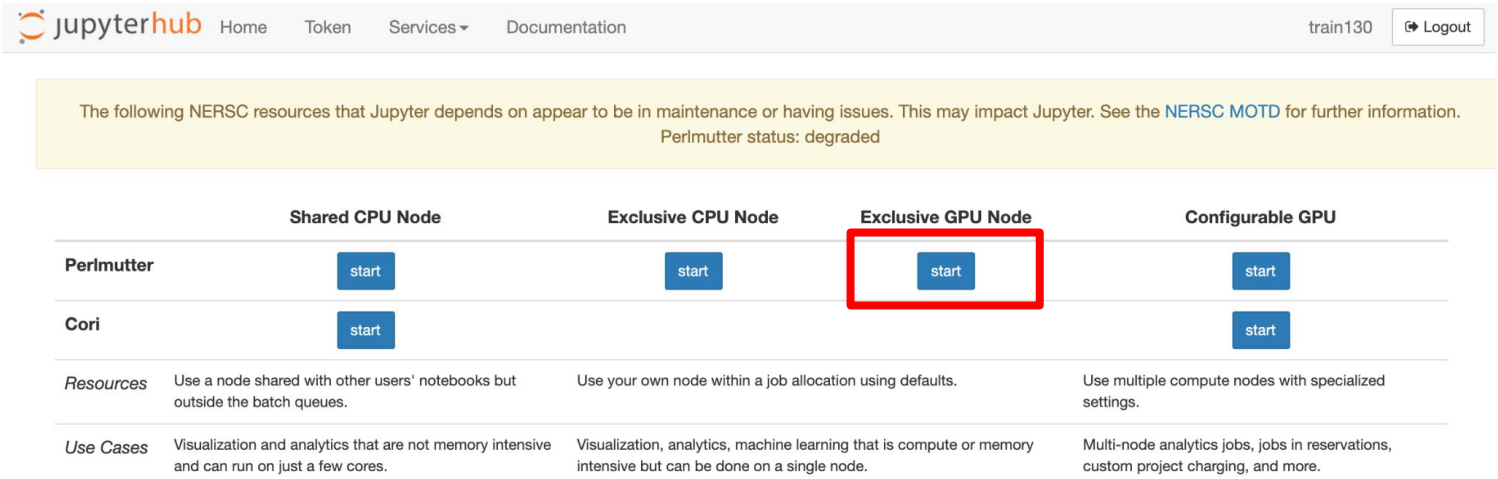
Logging into Jupyter

- Go to https://jupyter.nersc.gov
- Sign in using your training account credentials
- Select your preferred jupyter instance:



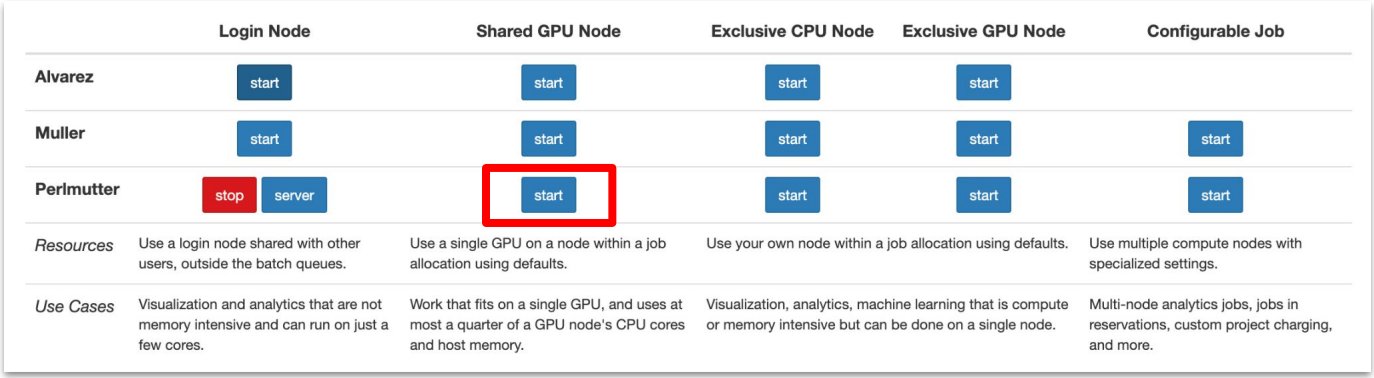
Setup

- Go to https://jupyter.nersc.gov and select: "Exclusive GPU Node" in the "Perlmutter" row



Setup

If you're spending a long time in the queue waiting for an exclusive GPU node, "Shared GPU Node"s are also available – eg:

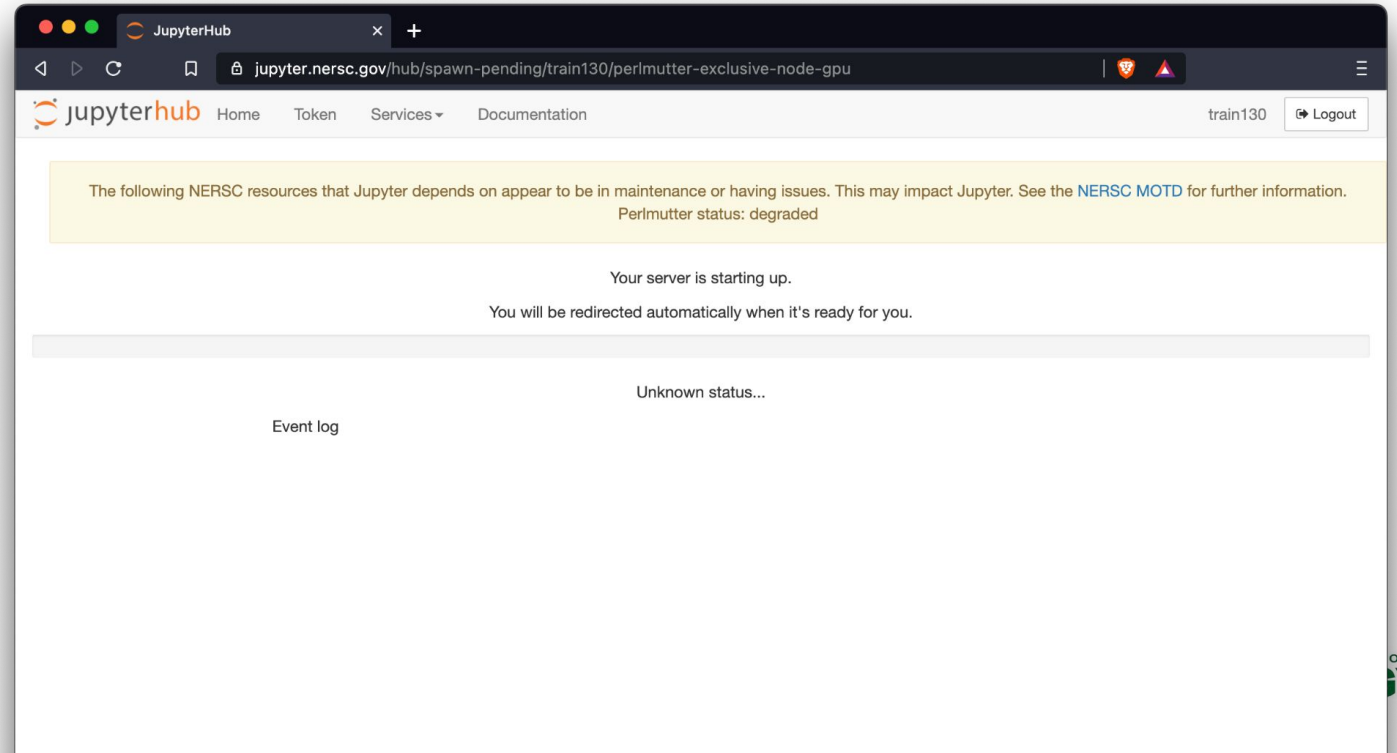


	Login Node	Shared GPU Node	Exclusive CPU Node	Exclusive GPU Node	Configurable Job
Alvarez	start	start	start	start	
Muller	start	start	start	start	start
Perlmutter	stop server	start	start	start	start
Resources	Use a login node shared with other users, outside the batch queues.	Use a single GPU on a node within a job allocation using defaults.	Use your own node within a job allocation using defaults.		Use multiple compute nodes with specialized settings.
Use Cases	Visualization and analytics that are not memory intensive and can run on just a few cores.	Work that fits on a single GPU, and uses at most a quarter of a GPU node's CPU cores and host memory.	Visualization, analytics, machine learning that is compute or memory intensive but can be done on a single node.		Multi-node analytics jobs, jobs in reservations, custom project charging, and more.

Alternatively please use a reservation (details later)

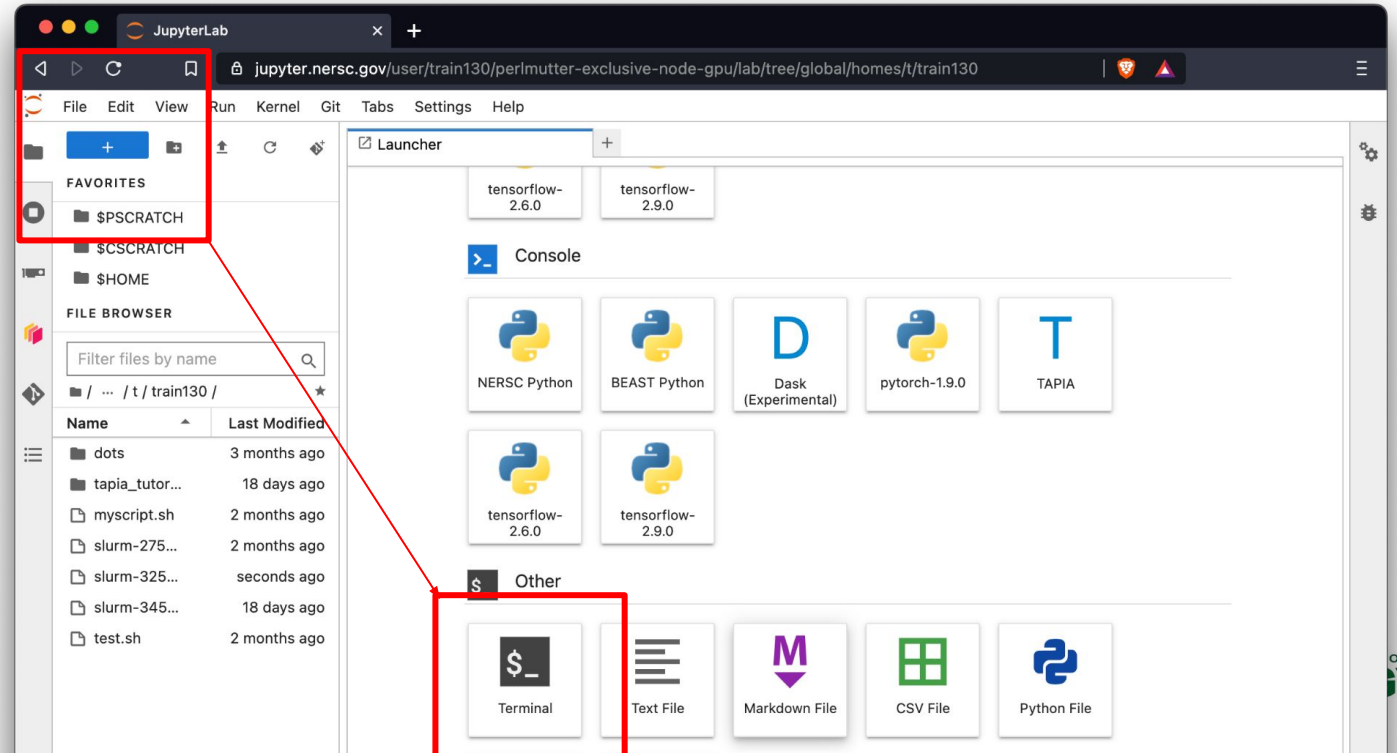
Setup

- Jupyter should take a minute to start:



Setup

- (If you don't see a terminal), select "+" followed by "Terminal"

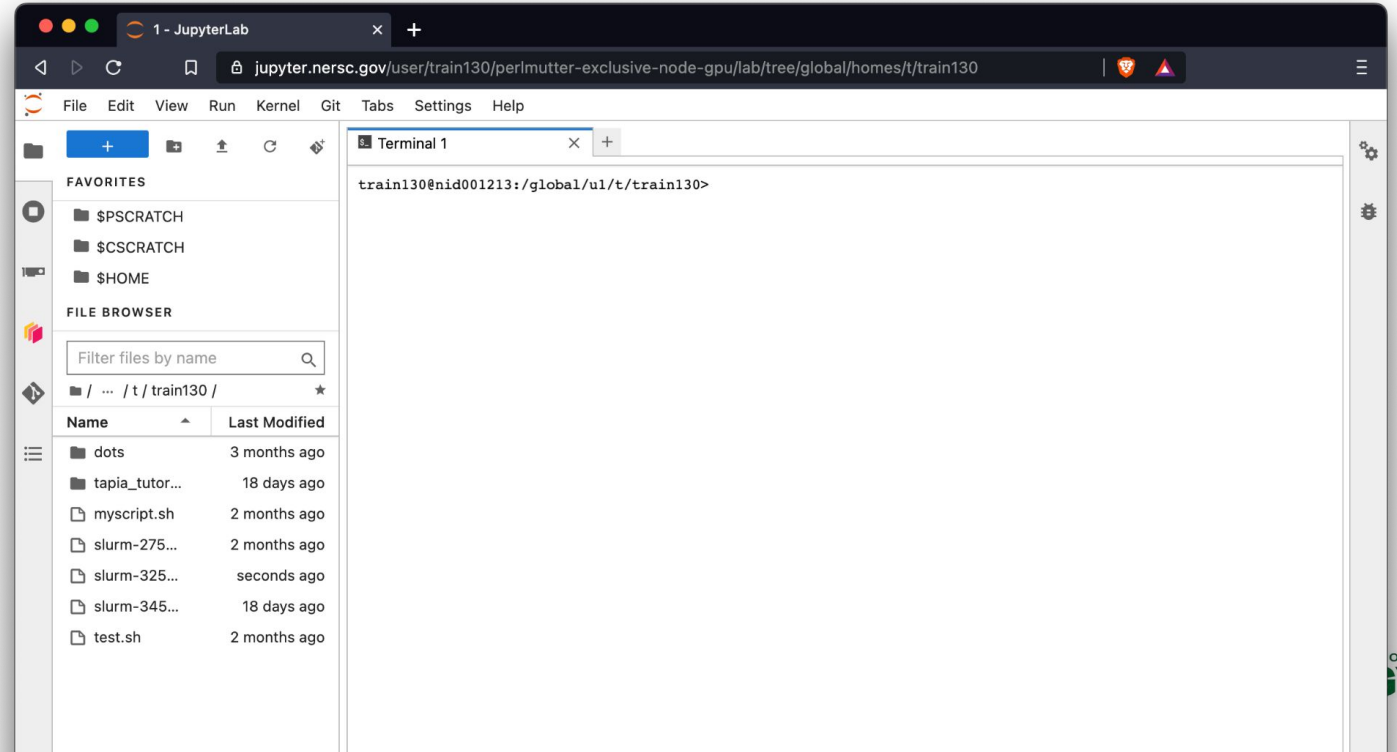


NERSC

Office of Science

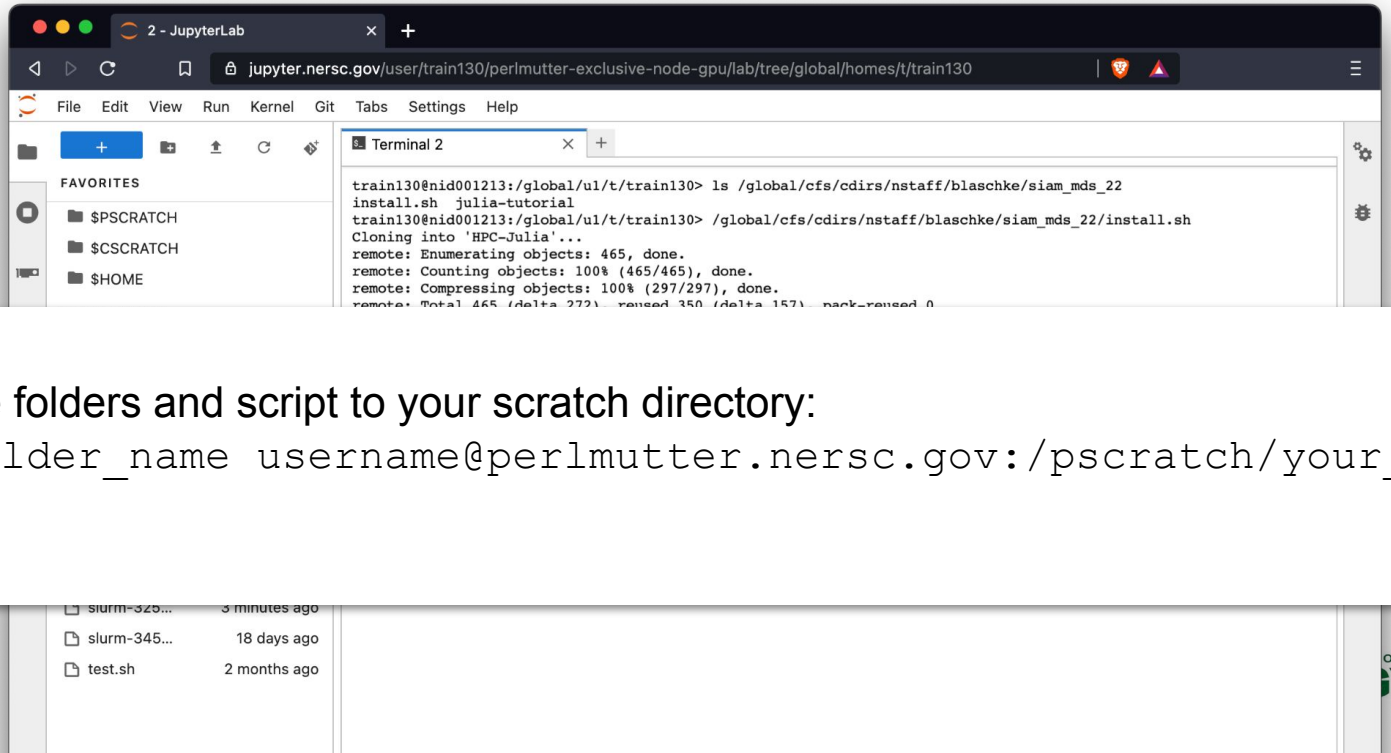
Setup

- Once started you should see a terminal



Setup

- (In the terminal), follow installation instructions

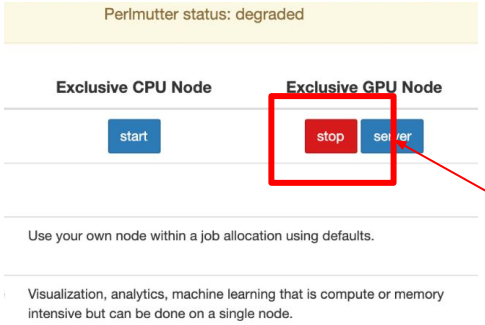


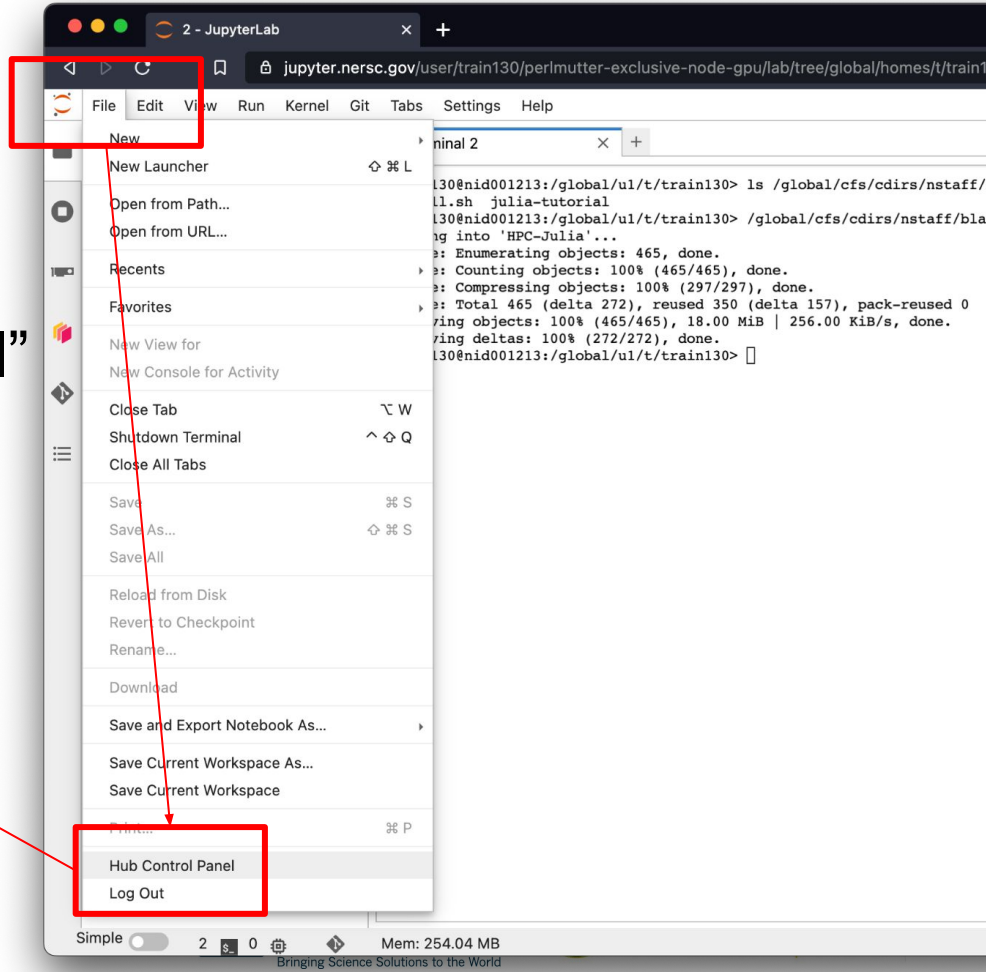
Upload the folders and script to your scratch directory:

```
$ scp folder_name username@perlmutter.nersc.gov:/pscratch/your_path
```

Setup

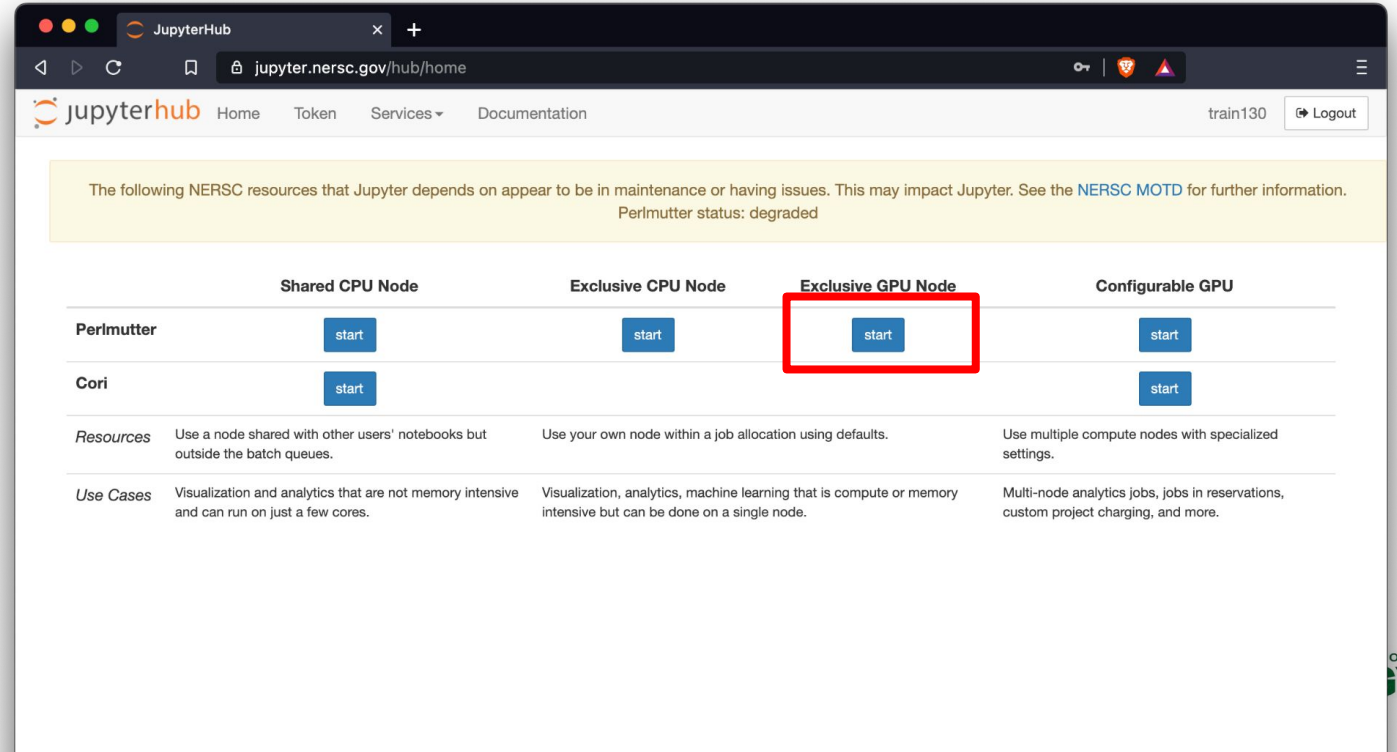
- Stop your server by going to: “File” > “Hub Control Panel”
- The selecting “Stop”





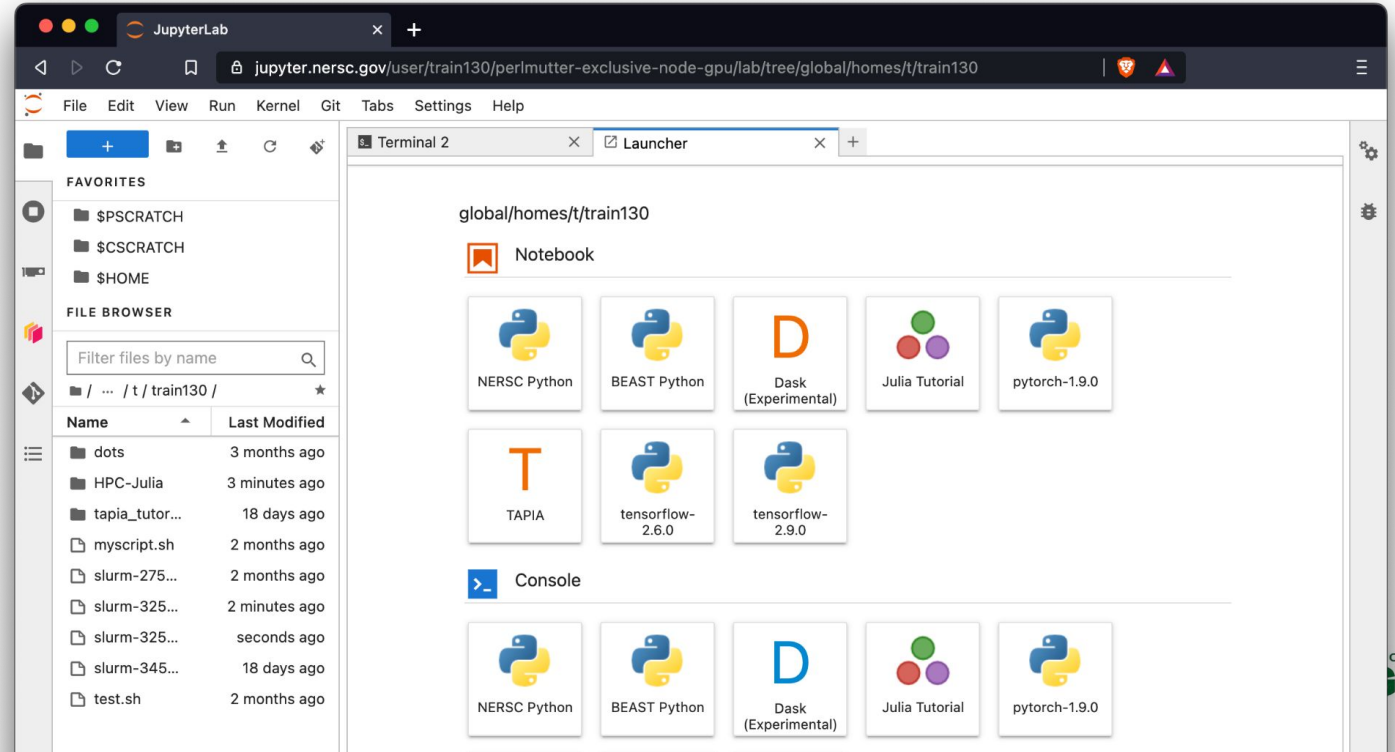
Setup

- Start a fresh exclusive node on Perlmutter



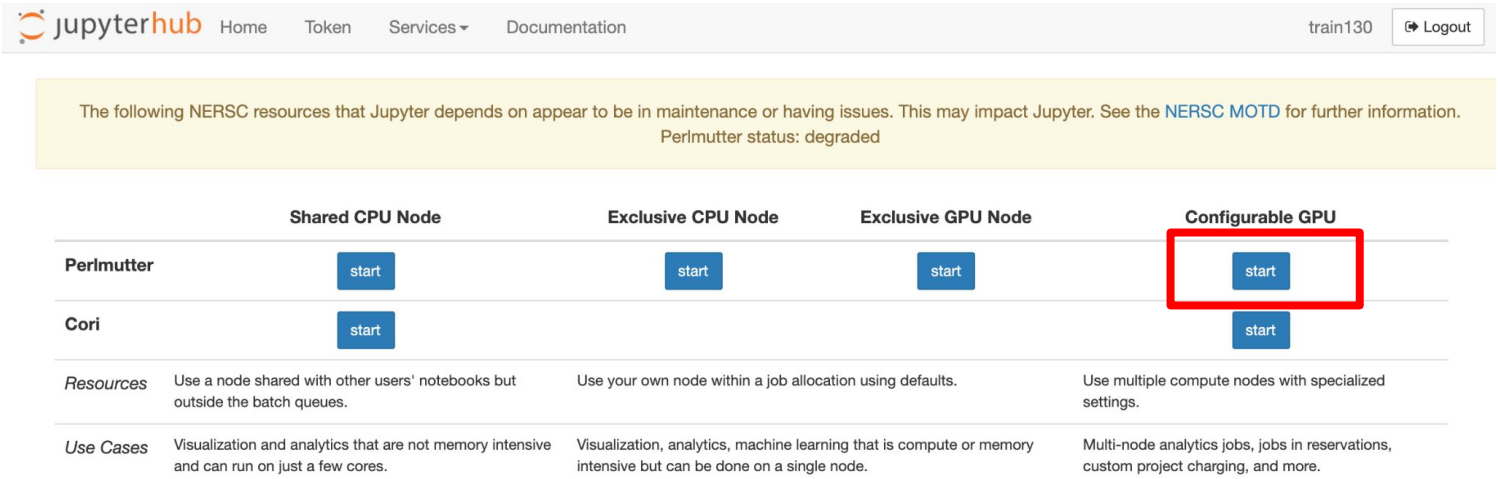
Setup

- Select the launcher, you should see the Cuda-Q Kernel :)



Using Reservations in Tutorials

- Go to https://jupyter.nersc.gov and select: "Configurable GPU" in the "Perlmutter" row



Jupyter Options:

Leave defaults, except:

Account:

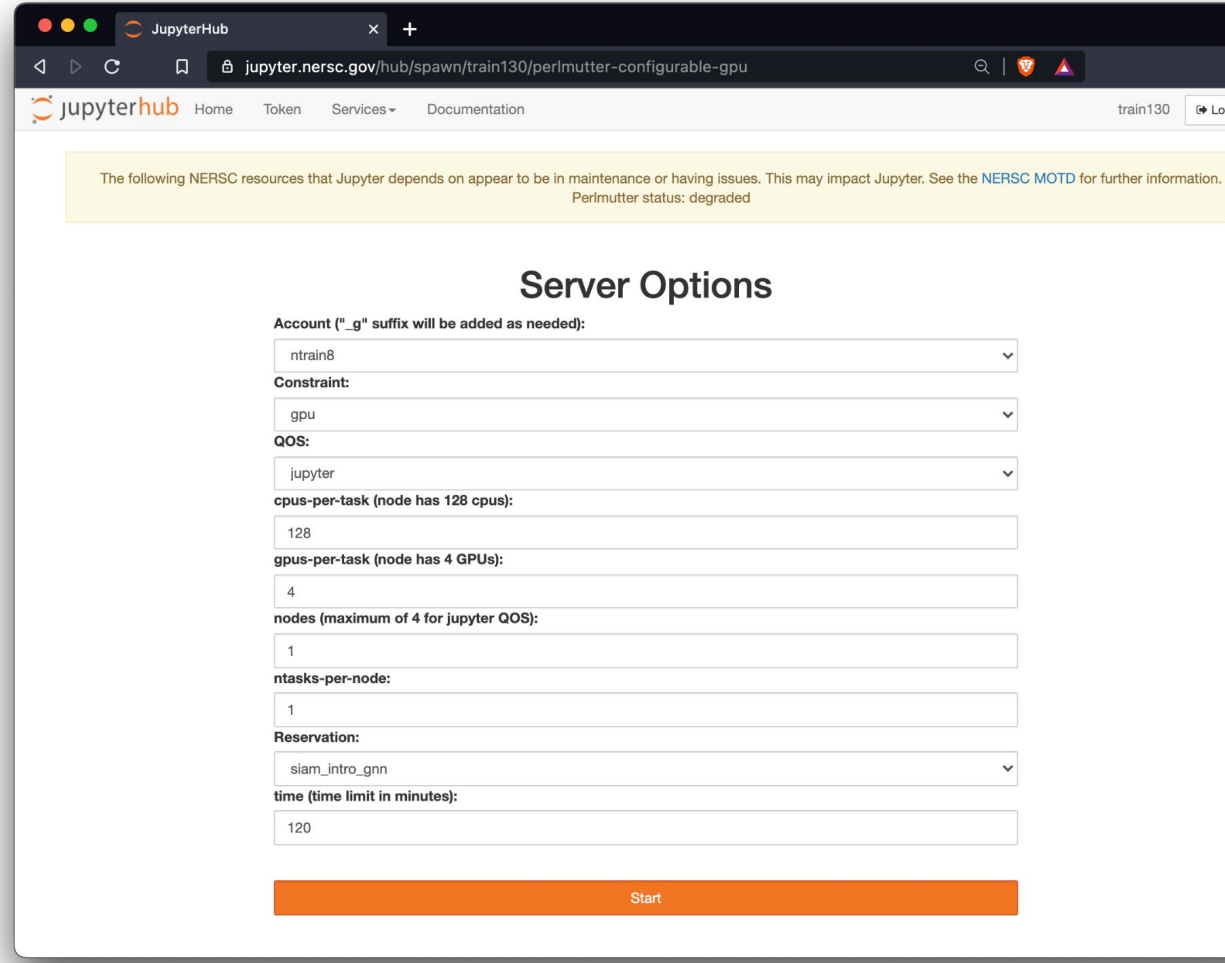
`ntrain10`

Reservation

`TBD`

Time

`TBD`



Note:

- The reservation is active from until 5pm EST today. If you request a node during this time, make sure that the current time + requested time does not exceed the reservation's end time (5pm).

- Training accounts are fully fledged NERSC accounts
- Will remain available until Tuesday – please copy any data and code to your computer before then

Agenda

Morning Sessions

- Introductions
- Overview of CUDA-Q
- Onboarding to Perlmutter
- Accelerating Quantum Simulations Notebook
- NVIDIA Quantum Cloud and Local install of CUDA-Q
- Intro to large scale clusters, how to navigate and use them

Afternoon Sessions

- Example: Quantum chemistry and nuclear physics at NERSC
- Industry Use Case: Simulating Hamiltonians of molecules with 30,000 terms
- QuEra’s Quantum Hardware
- Example: Quantum reservoir computing

What's the problem?

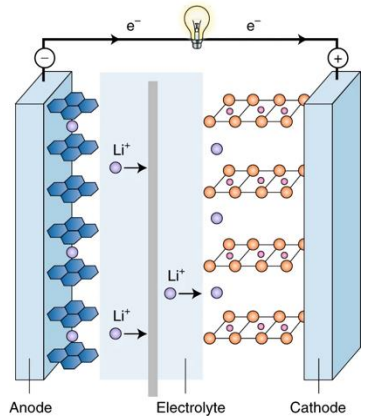
- The goal is finding the ground state of a many-body Hamiltonian (describing e.g., a many-electron molecule)

External potential (nuclei-electrons)

$$H = \sum_i \left(-\frac{\hbar^2}{2m}\nabla_i^2 + V_{\text{ext}}(r_i) \right) + \sum_{i,j} V_{\text{e-e}}(|r_i - r_j|)$$

Kinetic part for the electrons

Interaction (electrons-electrons)

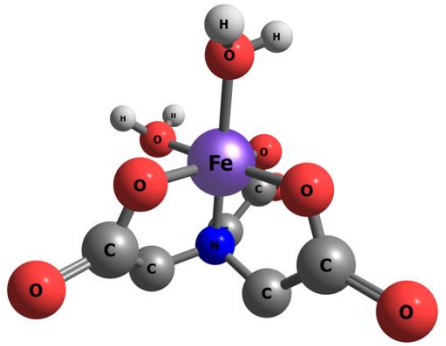


Nat. Electron.

- In general, $H = H_1 + H_2$
  - H_1 contains only one-body operators (kinetic terms, external potentials, …). Easy
  - H_2 contains the two-body operators (Coulomb interactions, …). Complicated

- Solving for the ground state is a difficult task:
  - Space of solutions grows exponentially
  - Perturbation theory often fails

- Need for numerical / approximate techniques

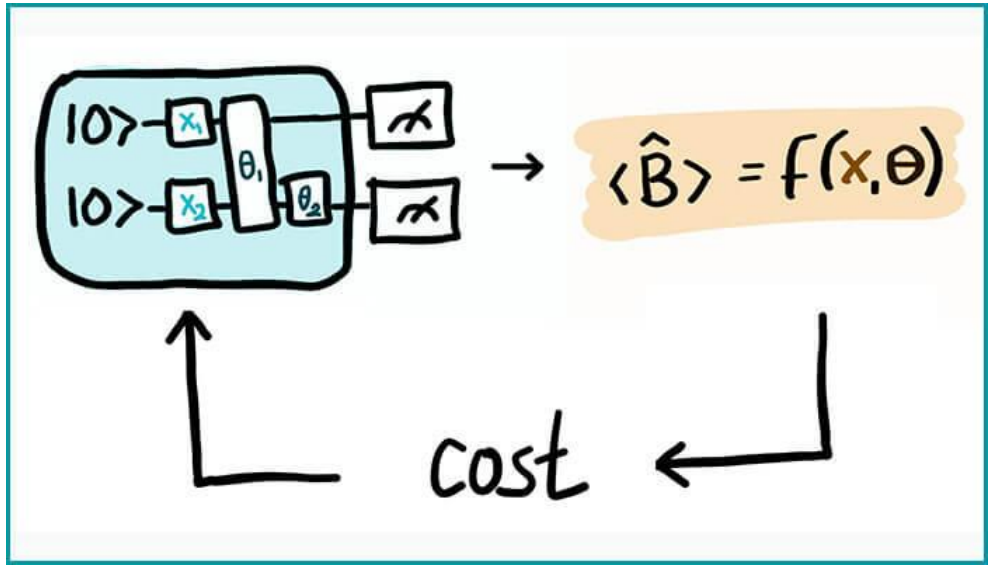


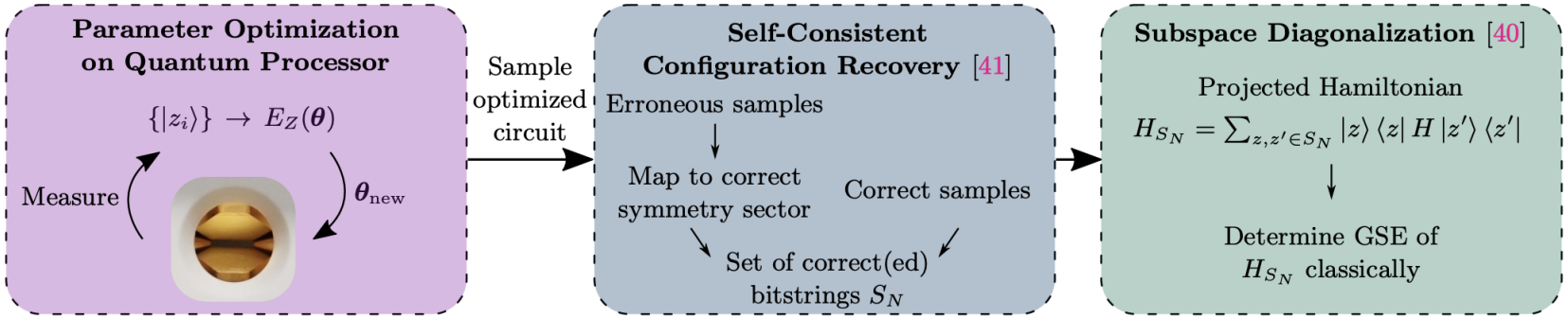
arxiv:2408.10801

VQE: One possible solution

- In NISQ devices, VQE with hybrid approaches:

$$E = \min_{\theta} \langle \psi(\theta) | H | \psi(\theta) | \rangle$$





Solving an Industrially Relevant Quantum Chemistry Problem on Quantum Hardware

Ludwig Nützel,[1,*] Alexander Gresch,[2,3] Lukas Hehn,[4] Lucas Marti,[1] Robert Freund,[5] Alex Steiner,[5] Christian D. Marciniak,[5] Timo Eckstein,[1,6] Nina Stockinger,[1,7] Stefan Wolf,[1] Thomas Monz,[5,†] Michael Kühn,[4] and Michael J. Hartmann[1,6]

[1] *Department of Physics, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany*
[2] *Heinrich Heine University Düsseldorf, Faculty of Mathematics and Natural Sciences, Germany*
[3] *Institute for Quantum Inspired and Quantum Optimization, Hamburg University of Technology, Germany*
[4] *BASF SE, Next Generation Computing, Pfalzgrafenstr. 1, 67061 Ludwigshafen, Germany*
[5] *Universität Innsbruck, Institut für Experimentalphysik, Innsbruck, Austria*
[6] *Max Planck Institute for the Science of Light, Staudtstraße 2, 91058 Erlangen, Germany*
[7] *Department of Chemistry, Technical University of Munich, Garching, Germany*

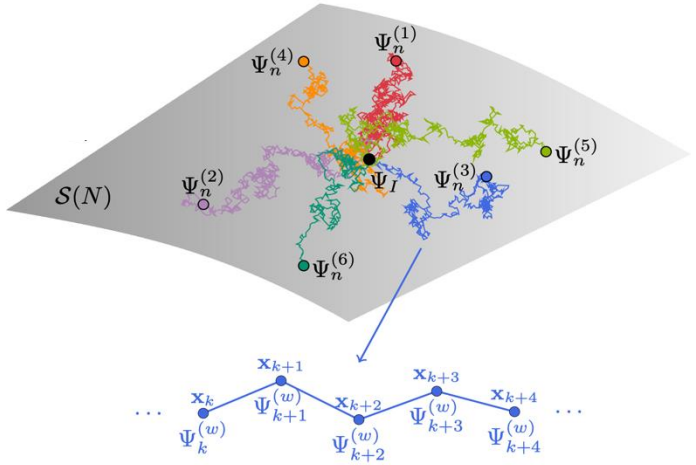
Auxiliary Field QMC

M Motta, S Zhang, WIREs Comput Mol Sci 2018, 8:e1364

- AFQM transforms the “complicated” interaction terms to an infinite sets of non-interacting terms
- Strongly correlated systems not well described by lower level of theory (HF, DFT, …)
- Monte Carlo methods proved useful in many other applications

AFQMC: basic ingredients

- Imaginary time evolution
- Random walk in the space of Slater determinants
- Good trial (initial) wave function



AFMQC: Imaginary time evolution

AFQMC is a projection quantum Monte Carlo technique

- Our target is the ground state $|\Psi_0\rangle$ of H
- Let's consider an initial state

Excited states of *H*

$$|\Phi_I\rangle = c_0|\Psi_0\rangle + c_1|\Psi_1\rangle + c_2|\Psi_2\rangle + \dots$$

- Apply $e^{-\beta H}$

$$E_0 < E_1 < E_2 < \dots$$

Eigenvalues of *H*

$$e^{-\beta H}|\Phi_I\rangle \sim c_0|\Psi_0\rangle + c_1 e^{-\beta(E_1-E_0)}|\Psi_1\rangle + c_2 e^{-\beta(E_2-E_0)}|\Psi_2\rangle + \dots$$

- All the high-energy states are exponentially suppressed
- We project onto the ground state when $\beta \to \infty$.

J Lee, H Q Pham, D R. Reichman, arXiv:2208.01280

Why do we need a quantum computer?

- For extracting physical quantities like the g.s. energy, a **trial wf** $|\Psi_T\rangle$ is generally used

$$E_0 = \langle \Psi_T | H e^{-\beta H} | \Phi_I \rangle$$

- $|\Psi_T\rangle$ can be a high quality wf hard to simulate on a classical computer, e.g.:

$$|\Psi_T\rangle = |\Psi_{CC}\rangle = \exp(T - T^\dagger)\,|\phi_0\rangle$$

(unitary coupled cluster)

$$T = \sum_{P,Q} t_{PQ} \hat{a}_P^\dagger \hat{a}_Q + \sum_{P,Q,R,S} t_{PQ}^{RS}\ \hat{a}_P^\dagger a_Q^\dagger \hat{a}_R \hat{a}_S$$

- What do we need from the QC?

$$c = \langle \Psi_T | e^{-\beta H} | \Phi_I \rangle$$

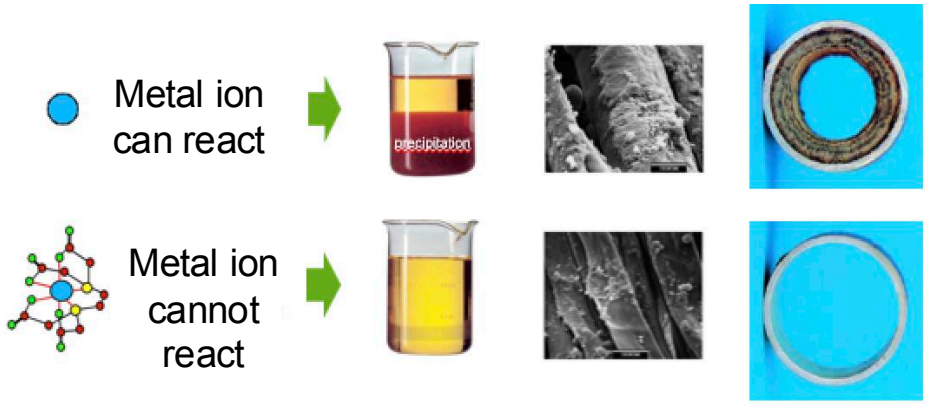
overlaps

Computed via Hadamard test or shadow techniques

FeNTA chelating agent for industrial applications

- Chelating agents are used to shield the undesired influence of metal ions in various industrial processes and environmental systems.

- They are used
  - in the detergent industry, to remove hard water ions and improve cleaning ability of detergents;
  - in paper and pulp production, to improve the effectiveness of the bleaching chemicals;
  - in the mining industry, they form stable complexes with metal ions, making it easier to separate and extract valuable metals such as copper, gold, and silver from the ore;
  - in the textile industry to remove metal ions from fabrics, preventing discoloration and damage, and improving the dyeing process;
  - in the photographic industry, to avoid unpleasant deposits on the photographic layers and contamination of the developing equipment;
  - in the agricultural industry, they continuously supply the plants with very low concentrations of metal ions required for healthy growth.



BASF has been producing these chemicals for many decades. The range and application possibilities grew continuously in line with customer requirements. Today, BASF is one of the leading international producers of organic chelating agents.

Agenda

Morning Sessions

- Introductions
- Overview of CUDA-Q
- Onboarding to Perlmutter
- Accelerating Quantum Simulations Notebook
- NVIDIA Quantum Cloud and Local install of CUDA-Q
- Intro to large scale clusters, how to navigate and use them

Afternoon Sessions

- Example: Quantum chemistry and nuclear physics at NERSC
- Industry Use Case: Simulating Hamiltonians of molecules with 30,000 terms
- QuEra’s Quantum Hardware
- Example: Quantum reservoir computing

Applications of Neutral-Atom Analog Quantum Computing

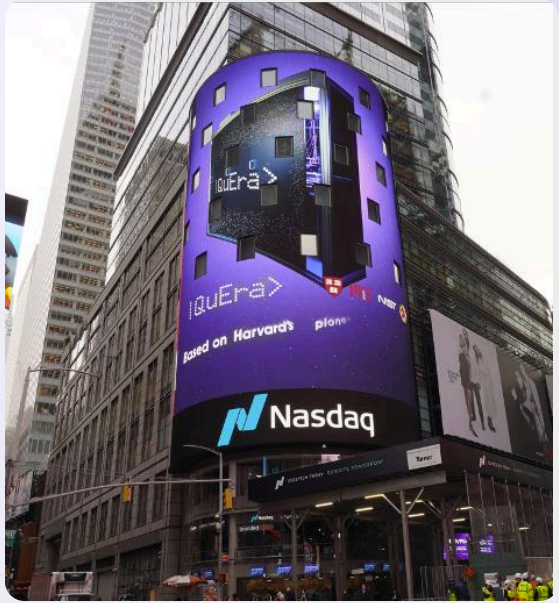
Tommaso Macrì, QuEra Computing

November 2024

SC’24 Tutorial

About QuEra

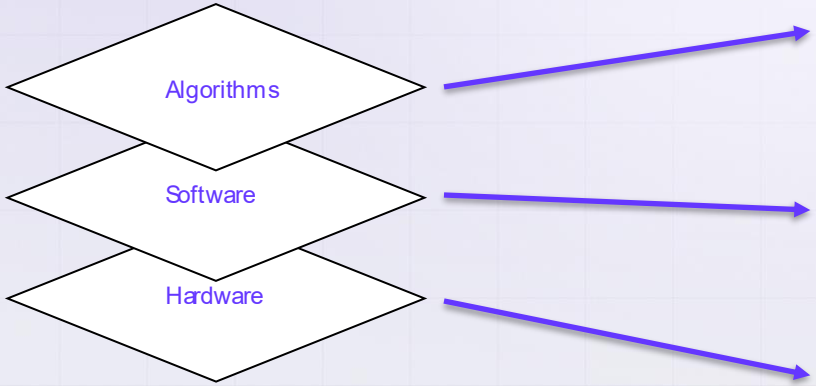
- Based in Boston, we employ 50+ scientists and engineers from Harvard, MIT, and other top-tier institutions.
- We have a working analog quantum computer on a public cloud and our mission is to build and commercialize **large-scale, error-corrected digital devices**.
- We productize innovations from QuEra and our MIT/Harvard collaborators at **record speed**.
- Dozens of organizations and hundreds of users already execute jobs on our quantum computer.
- We are hiring, check current job opportunities!



QuEra: A Full-Stack Approach

Tightly integrated hardware, software, and applications

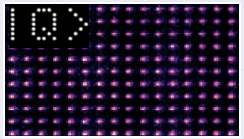
To leverage unique hardware features and maximize the quantum compute power



Unique algorithms for optimization, machine learning, and simulation

BLOQADE

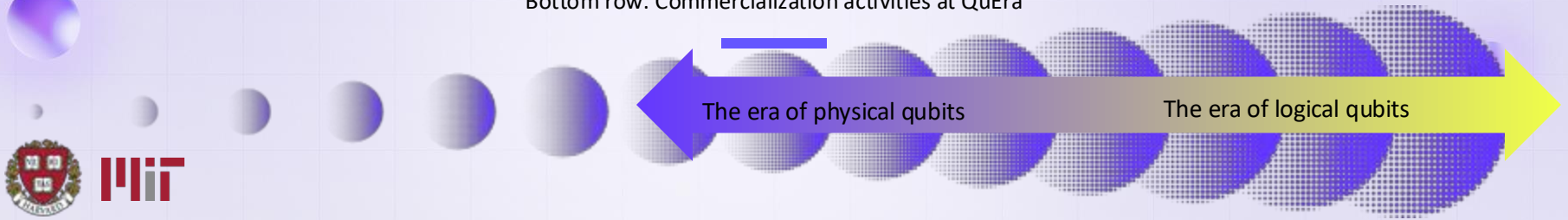
Development and emulation environment to test quantum software: Available in Julia and Python



Flexible, large-scale quantum hardware with unique features

Transitioning From Physical to Logical Qubits

Top row: Academic demonstrations at Harvard and MIT
Bottom row: Commercialization activities at QuEra



Nov '17
Programmable quantum processor with 51 qubits

Ref: Bernien (Harvard) et al., Science 2017

Jul '19
QuEra founded as a spin-off of Harvard-MIT

Jul '21
First demonstration of control over 256 atoms with scientific discoveries

Ref: Ebadi (Harvard) et al., Nature 2021

Nov '22
QuEra neutral-atom 256-qubit machine launches on AWS, the largest quantum computer on the public cloud

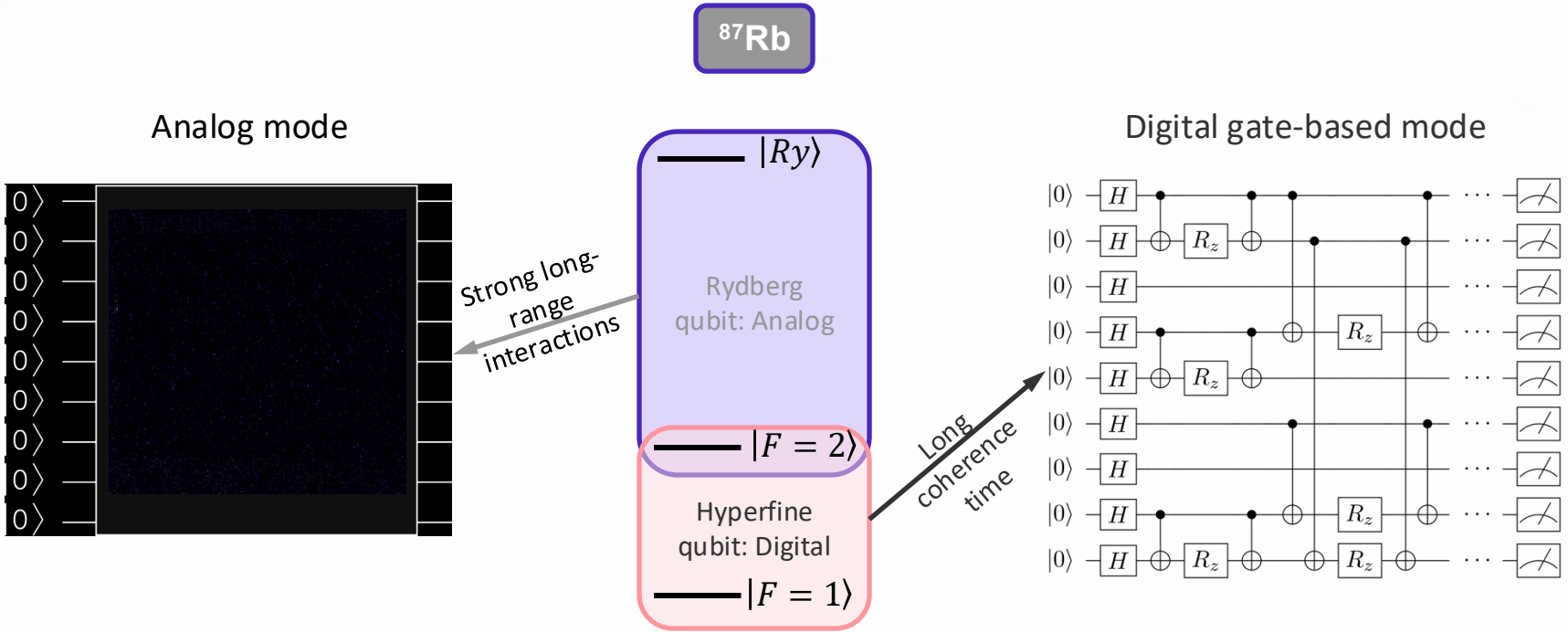
Apr '23
World record 99.5% two-qubit gate fidelity
a critical breakthrough for QEC

Ref: Evered (Harvard) et al., Nature 2023

Dec '23
Breakthrough demonstration of algorithms with multiple logical qubits

Ref: Bluvstein (Harvard) et al., Nature 2023

Analog and Digital Operation

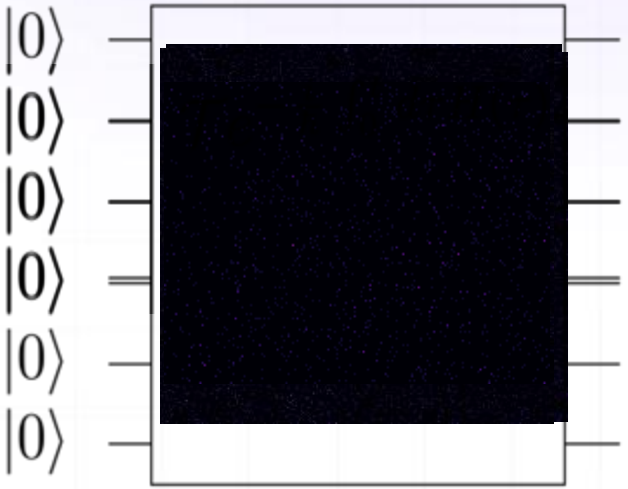


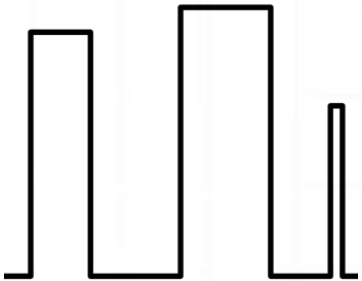
Analog quantum computing system

Analog waveforms for time-dependent dynamics
User-defined waveforms execute many different protocols.

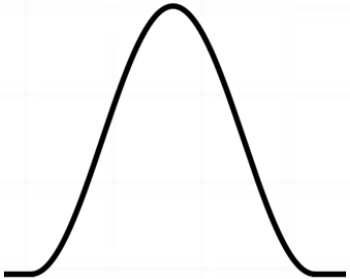
$$\hat{H}/\hbar = \sum_i \frac{\Omega(t)}{2}\left(e^{i\varphi(t)}\left|1\right\rangle_i\left\langle r\right| + h.c.\right) - \Delta_i(t)n_i + \sum_{i<j} V_{ij}n_in_j$$

Programmable connectivity
Encode problems in the spatial positions of atoms

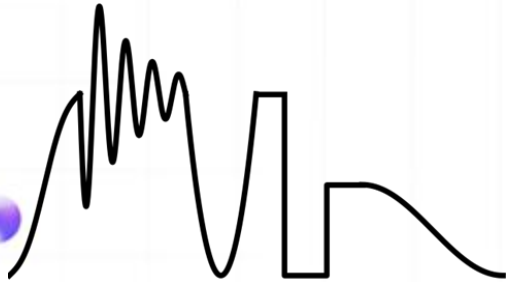




Gates



Annealing



Arbitrary

Aquila: Our 256-Qubit Analog Quantum Computer

- The first and still the world's **only** publicly-accessible quantum computer based on neutral atoms.
- Increased availability from 10 to over **100 hours per week.**
- Key applications: **simulation, optimization, machine learning.**
- The use of Aquila has resulted in several exciting scientific publications.
- Through "premium access", customers can engage directly with QuEra scientists.

Overview of New Features & Outlooks

New Aquila features for Analog Hamiltonian Simulation (AHS)

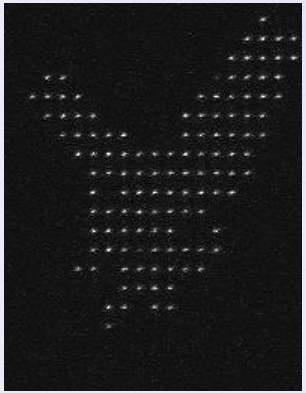
- Local detuning
  - Programmable individual qubit detunings in AHS
- More flexible qubit arrangements
  - Tighter row spacing 4->2 um, larger height 75 ->100 um

New opportunities in analog quantum simulation:

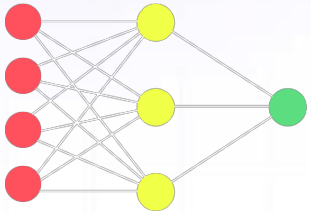
- Programmable disorder, initial state
- Non-equilibrium dynamics
- Exotic lattices
- Larger system sizes for 1D and quasi-1D problems

New opportunities in **digital quantum simulation**:

- Material, chemistry, condensed matter system
- Qubit shuttling, long-range connectivity
- Large scale
- Early fault tolerant digital quantum simulation
- Quantum advantage?
- …

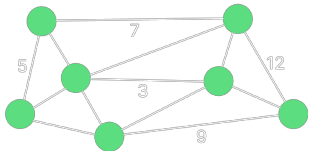


What problems can QuEra's Aquila solve?



Machine learning

- Classification
- Regression and prediction
- Reservoir computing



Optimization

- Maximum independent set
- Non-native optimization, e.g. QUBO
- Commercially-relevant problems

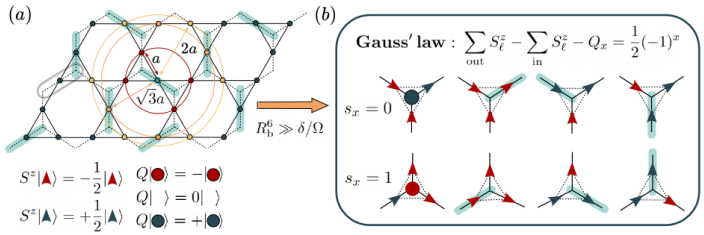
Simulation

- Equilibrium quantum phase of matter
- Non-equilibrium quantum dynamics
- Lattice gauge theories

Check out the Aquila Whitepaper: Wurtz et al., arXiv:2306.11727

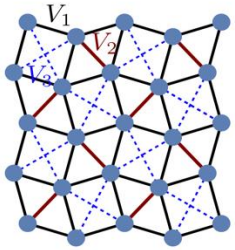
Quantum Simulations Demonstrations on Aquila

String Breaking



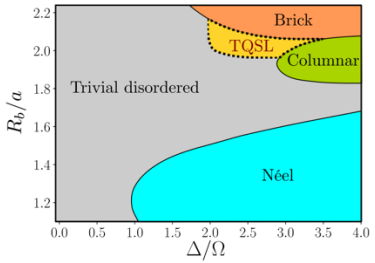
Collab. with Peter Zoller (Innsbruck)

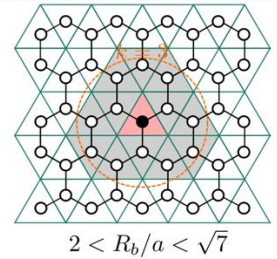
Shastry-Sutherland



Collab. With Shalaev, Boltasseva, Banerjee groups (Purdue & QSC)

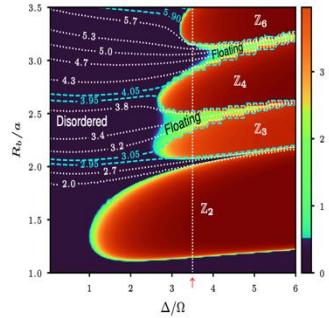
Trimer Quantum Spin Liquids





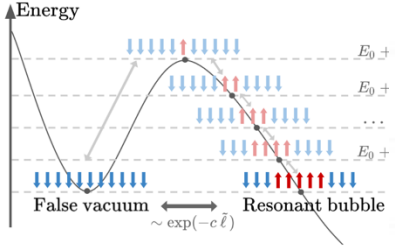
M. Kornjaca et al., Comm. Phys. 2023

Quantum Floating Phases



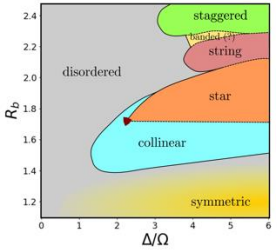
Collab. with Yannick Meurice (Iowa), Shan-Wen Tsai (UC Riverside), arXiv:2401.08087

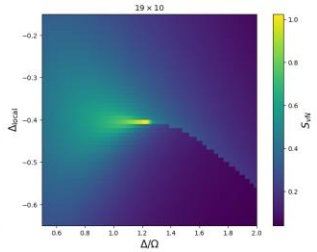
False Vacuum Dynamics



Collab. with NERSC
S. Darbha et al, PRB (2024)

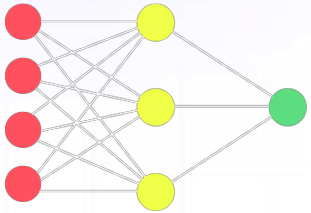
Multi-Criticality and Analog Higgs





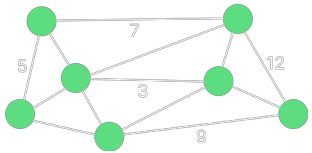
Collab. with NERSC
S. Darbha et al, PRB (2024)

What problems can QuEra's Aquila solve?



Machine learning

- Classification
- Regression and prediction
- Reservoir computing



Optimization

- Maximum independent set
- Non-native optimization, e.g. QUBO
- Commercially-relevant problems

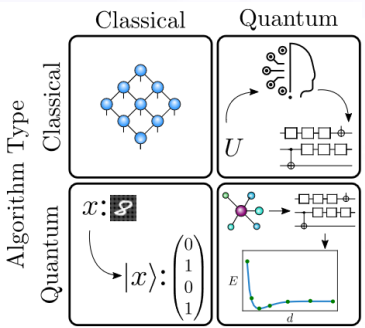
Simulation

- Equilibrium quantum phase of matter
- Non-equilibrium quantum dynamics
- Lattice gauge theories

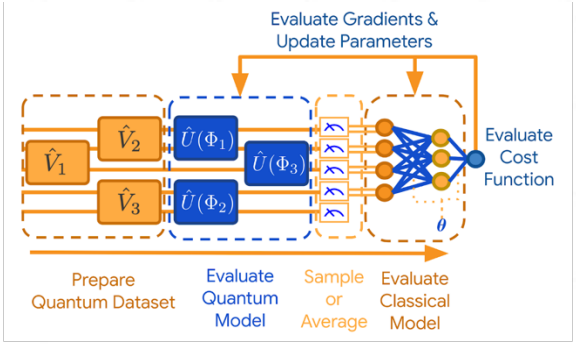
Check out the Aquila Whitepaper: Wurtz et al., arXiv:2306.11727

Overview

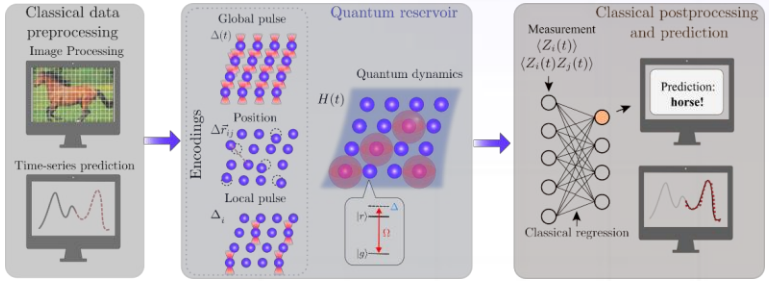
- Why quantum machine learning?



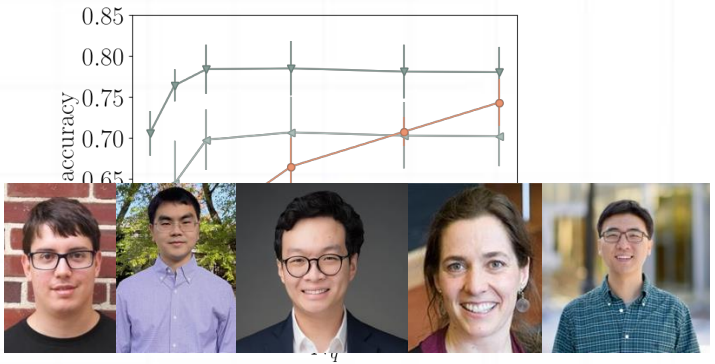
- Current quantum algorithms.



- Our approach – quantum reservoir learning.



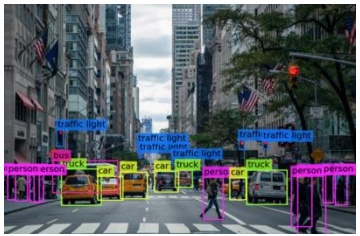
- Results – **largest quantum machine learning experiments (108 qubits)**, kernel advantage.

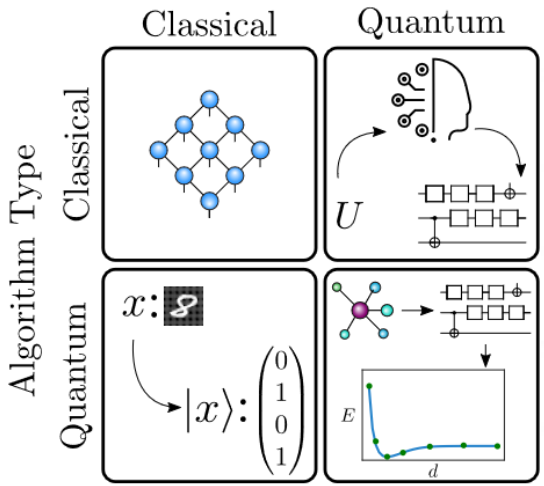


M. Kornjaca et al. arXiv:2407.02553 (2024)

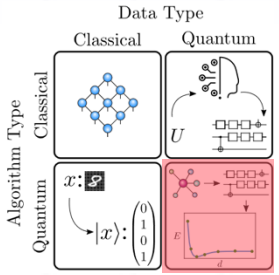
Quantum machine learning

- Machine learning – “unreasonably” successful **data-based** computation paradigm.
- Quantum machine learning – **critical** to the future of the field.
- Prospect for advantage – **quantum data**.

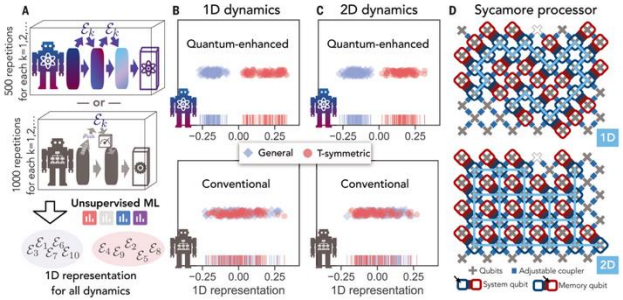




Quantum machine learning – promise



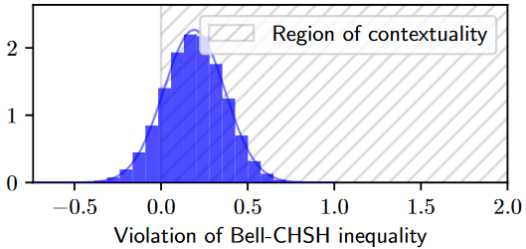
- Explicit – learning quantum dynamics.



[Science 376, 6598 (2022)] ***40 qubits**

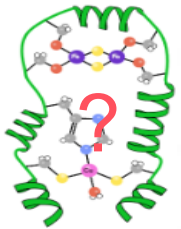
- Implicit – language ambiguities.

A and B belong to the same [cannibalistic / herbivorous]$_1$ species of animal. In a hot afternoon in south Sahara, ***one of them***$_1$ *was very hungry. They notice each other when they were roaming in the field. In a while,* ***one of them***$_2$ *is no longer [hungry / alive]$_2$.*



[EPTCS 384, 187 (2023)]

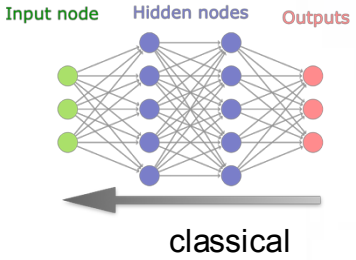
- Potential – materials and drug properties?



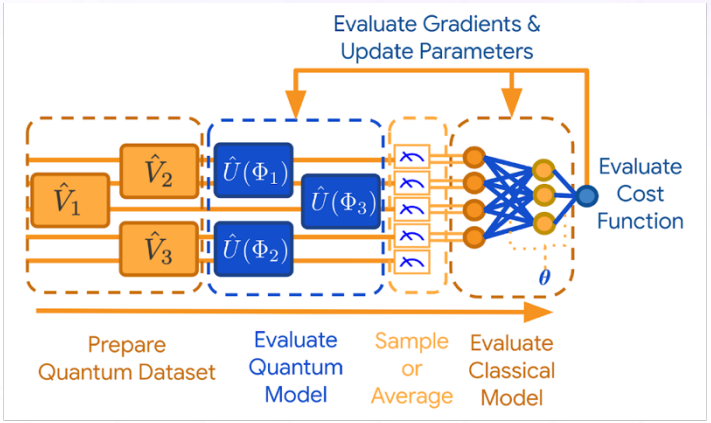
- What about **algorithms**?

Quantum machine learning – practice

- **Variational QML** - quantum neutral network.
- Dominate theoretical/numerical explorations.

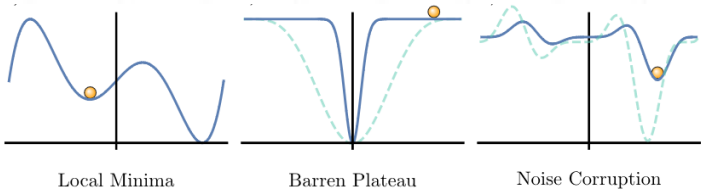


classical



quantum [Nat. Commun. 12, 6961 (2021)]

- Major issues:
  - **Cost** – gradient estimation loop.
  - **Trainability** – barren plateaus.

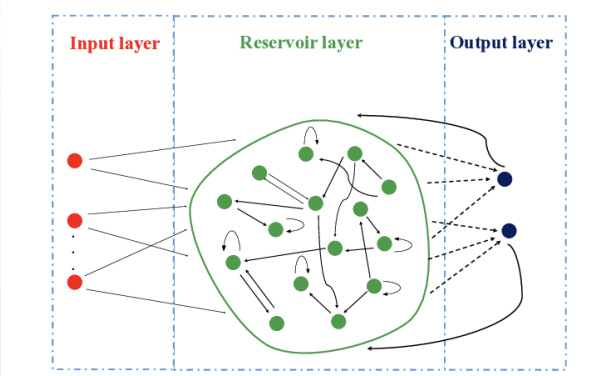


[Nat. Comput. Sci. 2, 567 (2022)]

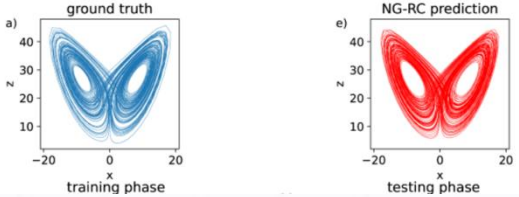
- **Experiments –** rare, small-scale, limited advantage potential**.**
- Our approach – **quantum reservoir computing**.

Classical reservoir computing

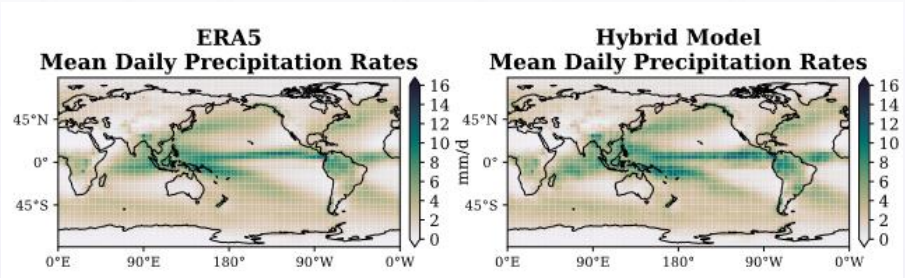
- Predictive power from a dynamical system – a **reservoir.**
- **Gradient-free** training.



- Successful **applications**:
  - Chaotic dynamics prediction.
  - Hybrid deployment – climate modeling.



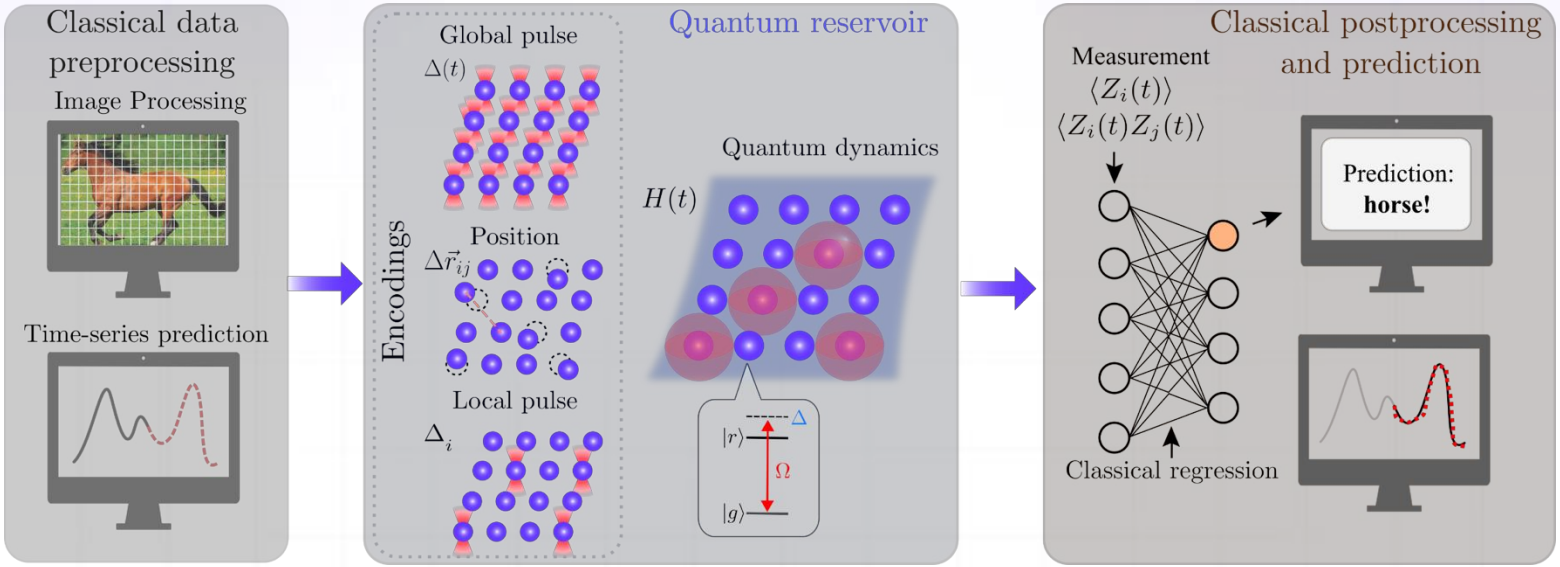
[Nat. Commun. 12, 5564 (2021)]



[Geophys. Res. Lett. 50, 102649 (2023)]

Quantum reservoir computing

Neutral atom quantum reservoir architecture:



QRC advantages (over variational QML):

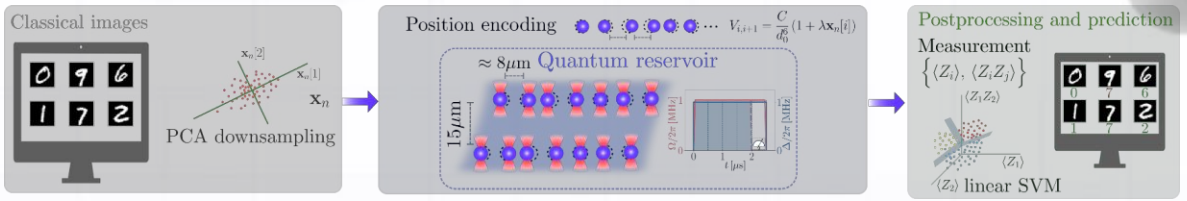
- Trainable
- Resource frugal
- Noise-resilient
- Scalable
- General purpose
- Classically intractable

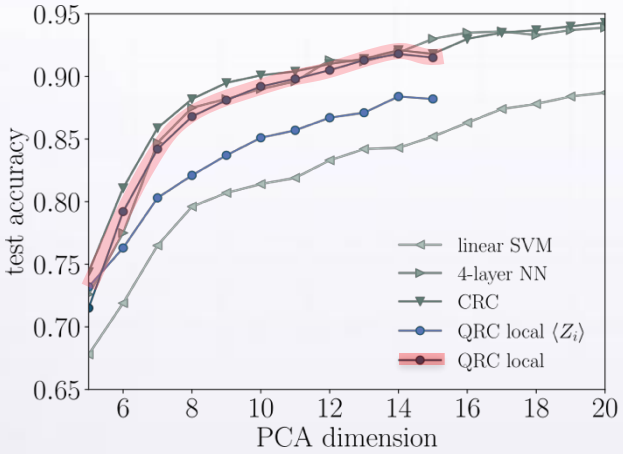
Proof of concept – simulations

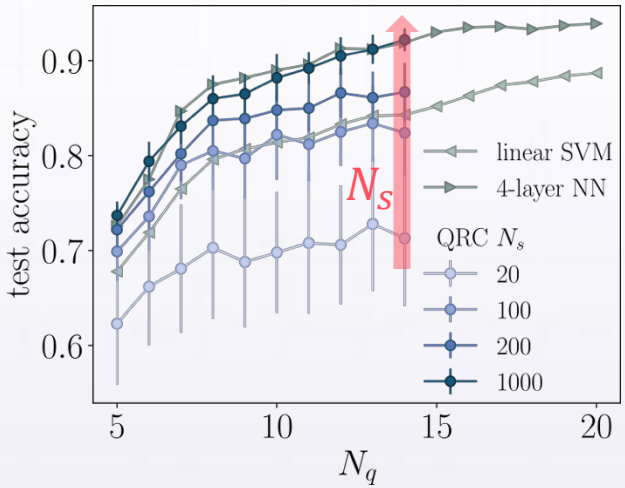
BLOQADE

Performance in simulation on MNIST:

- Exact simulation – reaches problem thresholds
- Practice – sample dependent performance

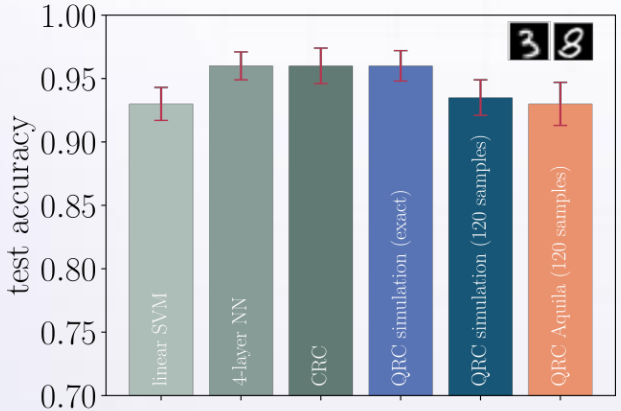


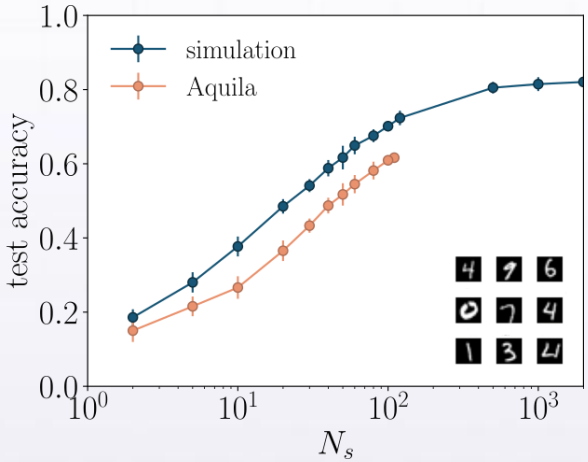




Proof of concept – Aquila

- **Performance on quantum hardware:**
  - 93% test accuracy on Aquila – binary task.
  - Within 0.5% of sampled simulation.
  - 62% accuracy – 10-class MNIST.
  - Sample requirements higher for multiclass.

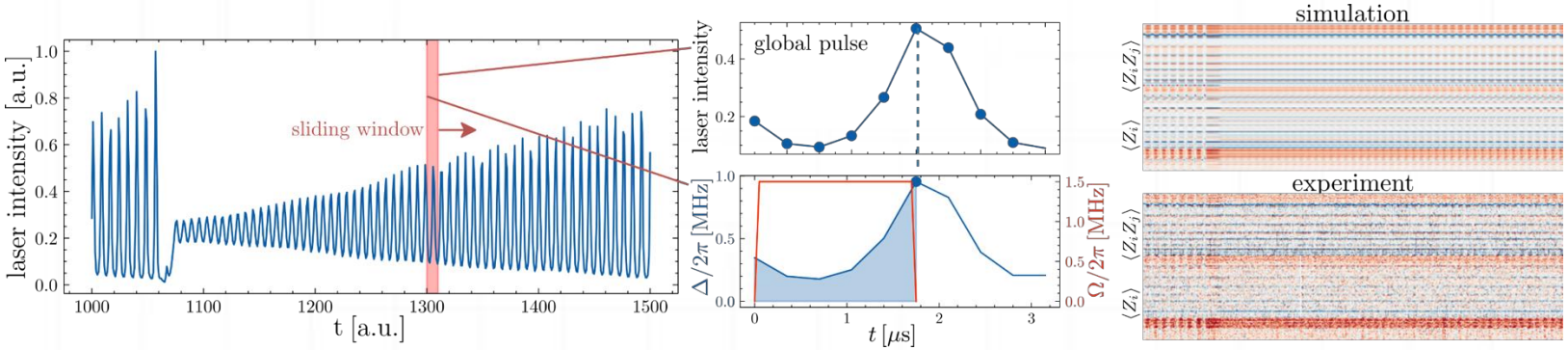




QRC – timeseries and encoding comparison

Look into QRC embeddings:

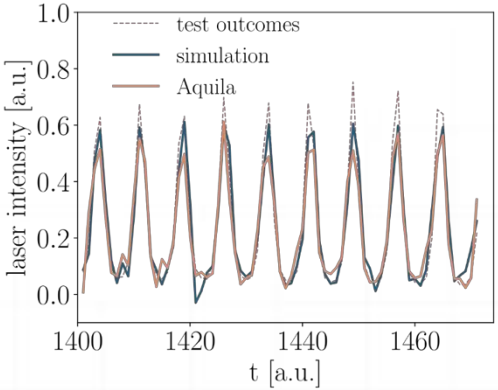
- Embeddings – new expressions of the data.
- Simulation and experiment agree.

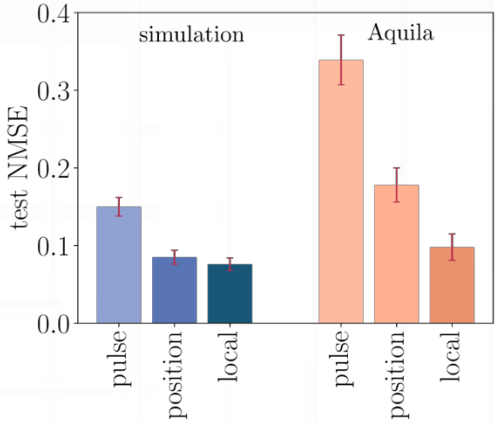


QRC – timeseries and encoding comparison

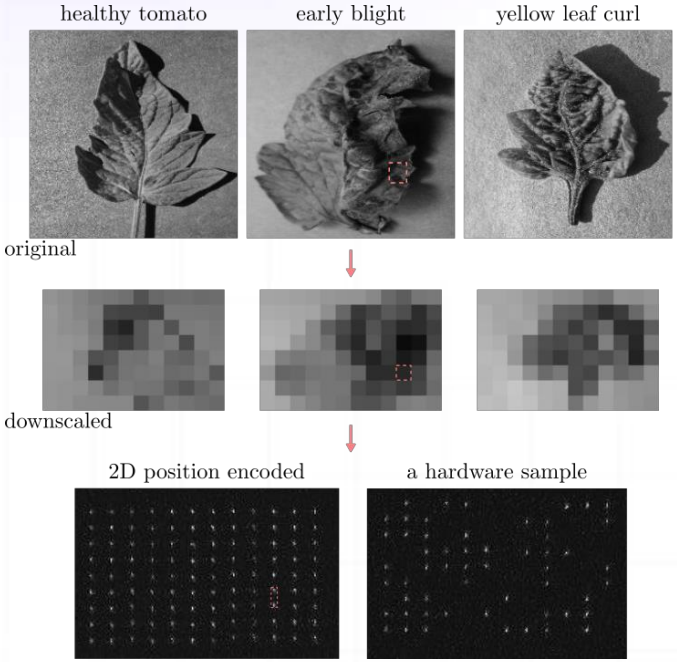
Time series usecase:

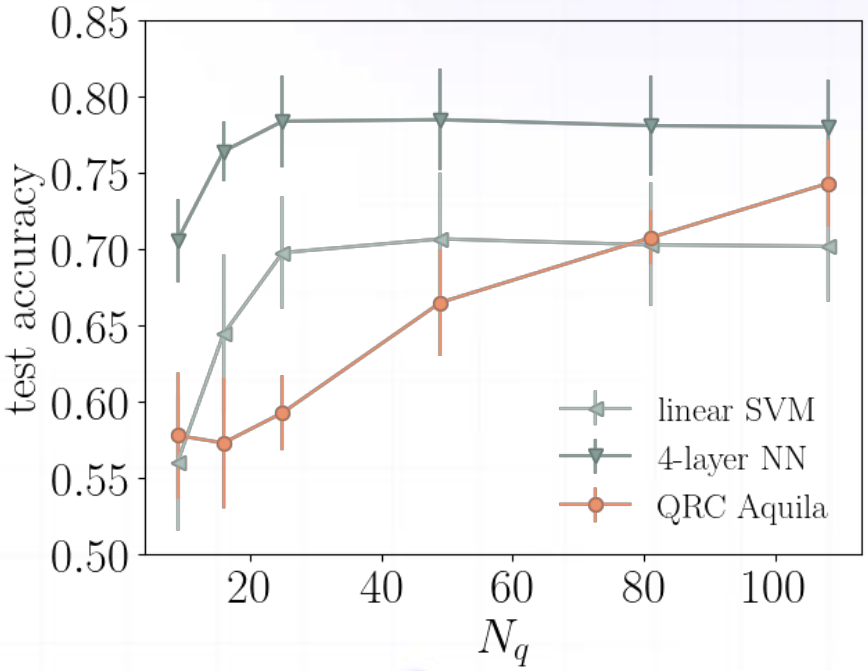
- Experiment performing as well as simulation.
- Encodings compared – **local detuning** the best.
- One of the first results with **expanded Aquila capabilities.**





Scaling up to > 100 qubits – Aquila experiments





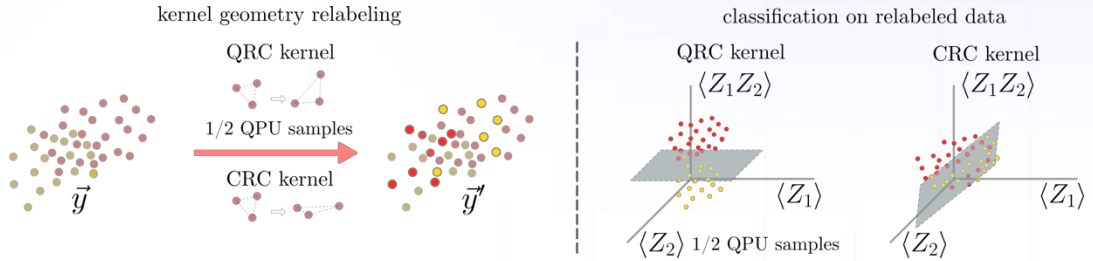
Main results:

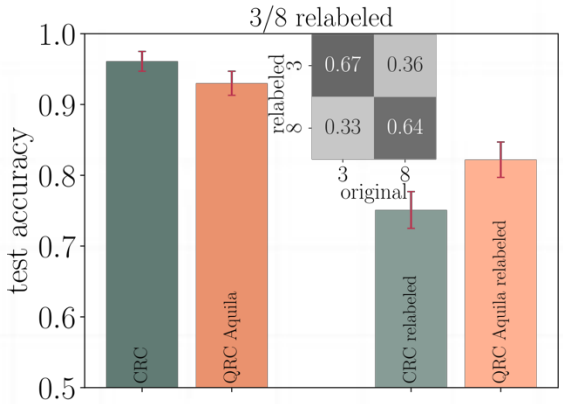
- Consistent scaling with increasing qubit number
- **The biggest successful QML experiment to date – 108 qubits**

QRC – Aquila experiments

Why QRC over classical methods?

- Potential classical intractability
- Data geometry changed in a fundamentally different way
- “Kernel relabeling” construction [Nat. Commun. 12: 2631 (2021)]
- **Comparative quantum kernel advantage**
- **Works on Aquila**





Thank you!