OAK RIDGE National Laboratory

Quantum-Accelerated High Performance Computing

Travis S. Humble
Quantum Computing Institute
Oak Ridge National Laboratory
humblets@ornl.gov

This presentation summarizes the architectural design of quantum-accelerated HPC system.

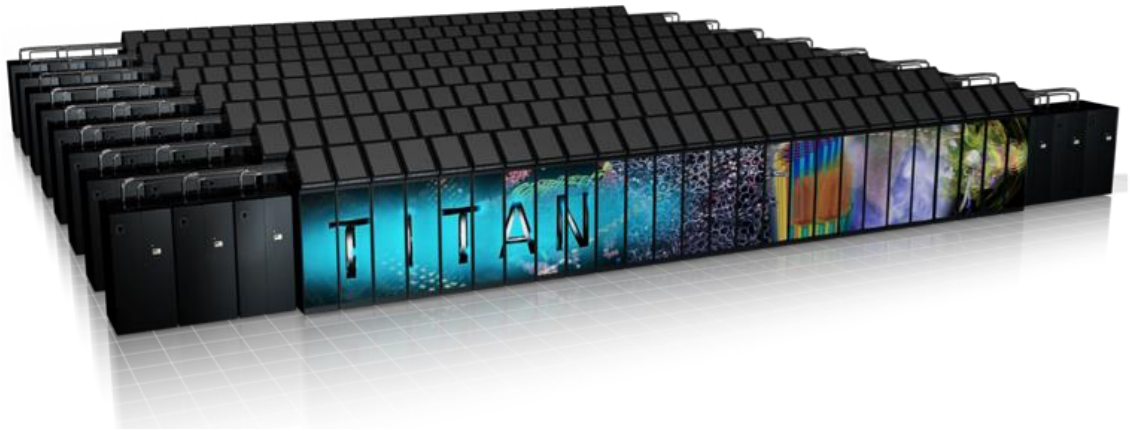
This research is supported by the Office of Science Early Career Research Program

U.S. DEPARTMENT OF ENERGY

17 September 2018

High-performance Computing

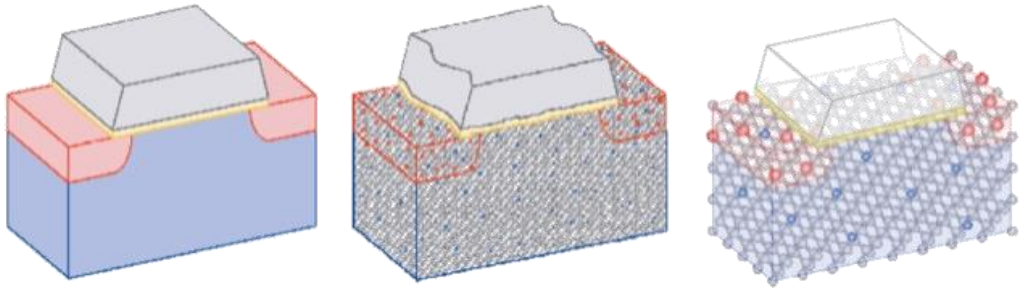
Scientific discovery and energy security depend on advances in computational capability

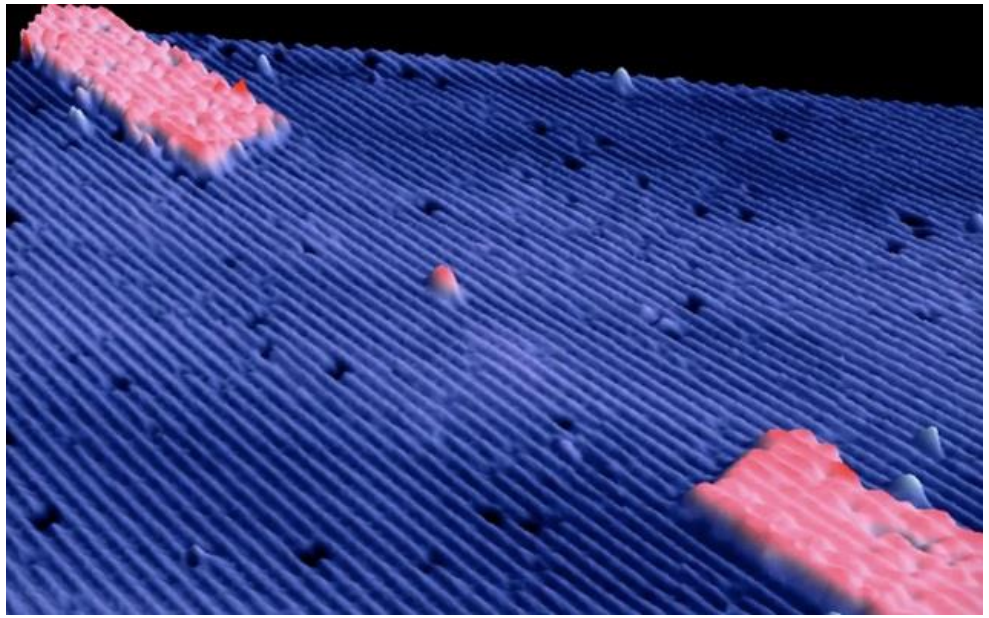


TITAN
Cray XK7, 18,688 Nodes
16-core AMD Interlagos + K20X
Gemini Network - 3D Torus Topology
27 PFLOPS peak, 9 MW

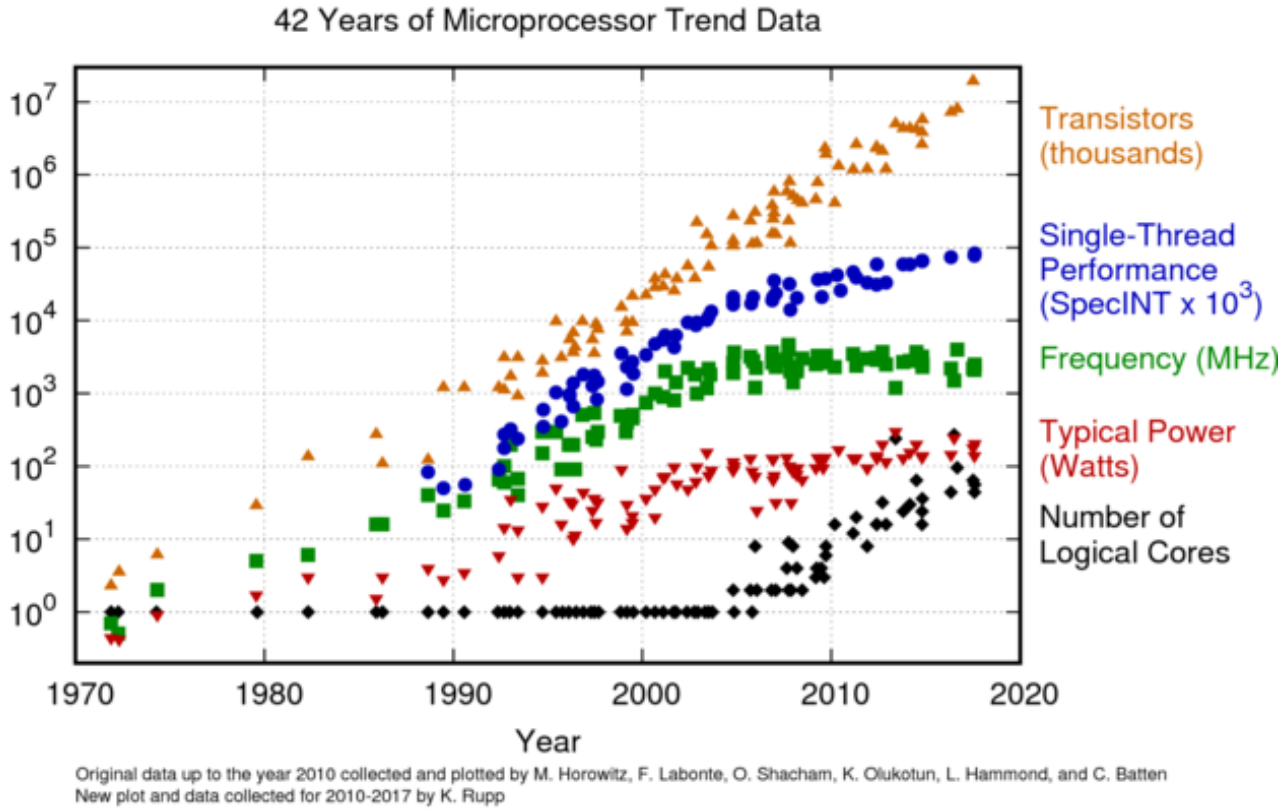
SUMMIT
IBM, 4,600 Nodes
2 Power9 + 6 NVidia Volta
Mellanox Network – Non-blocking Fat Tree
~200 PFLOPS, 13 MW

Planning for the Future of Computing





Single-atom transistor
from University of New South Wales, Australia

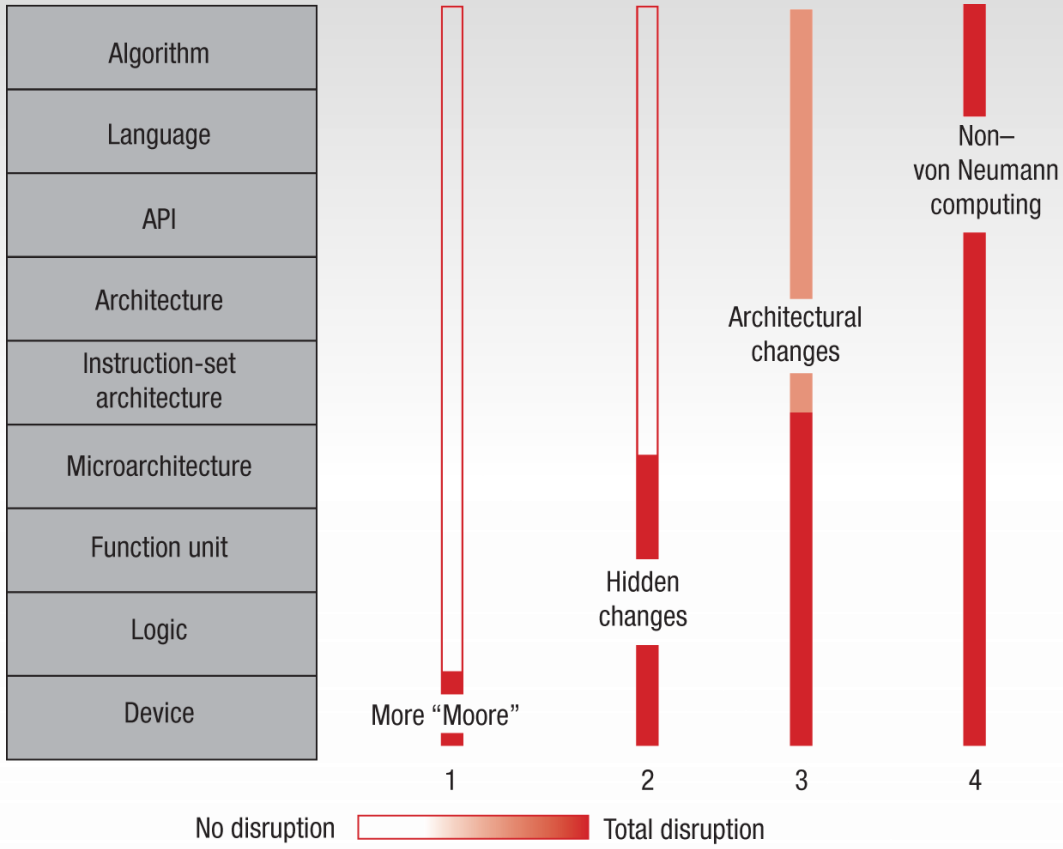


Future Technologies, Opportunities, and Challenges

Several alternative technology paths

- **More "Moore"**
  - New transistors, 3D devices
- **Hidden Changes**
  - New materials, cryogenic operation
- **Architectural Changes**
  - Special-purpose, probabilistic
- **Non-von Neumann**
  - Quantum computing
  - Neuromorphic computing

Risk mitigation is a dominating concern

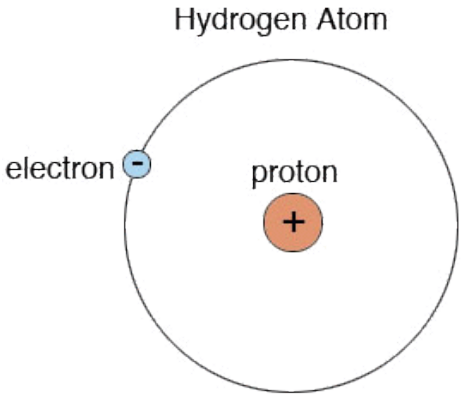


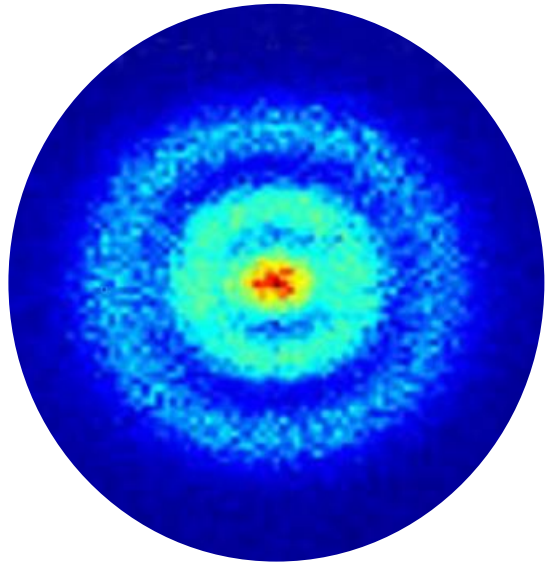
T. Conte et al., Rebooting Computing: The Road Ahead, Computer 50, 20-29 (2017)

What is Quantum Computing?

- Quantum mechanical computation
  - In quantum mechanics, the *wave function* describes all knowledge about the system
- Quantum computing manipulates the wave function to perform calculations
  - Quantum dynamical control of the Hamiltonian corresponds to computation

$$i\hbar \frac{\partial \Psi(t)}{\partial t} = H(t)\Psi(t)$$

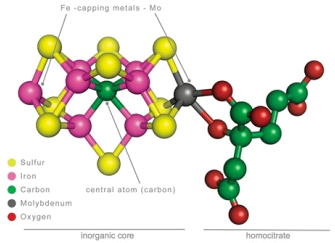


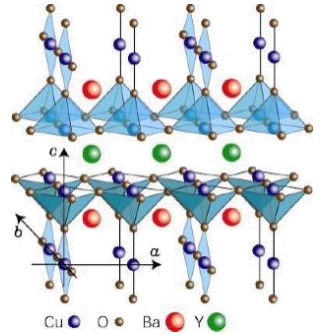


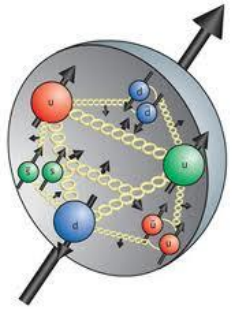
Stodolna et al. PRL 110, 213001 (2013)

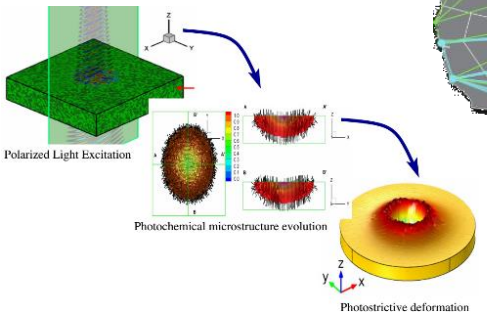
Scientific Applications of Quantum Computing

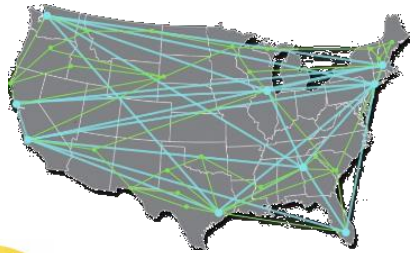
- Algorithms in the quantum computing model have been found to take fewer steps to solve problems
  - Quantum Simulation
  - Partition Functions
  - Discrete Optimization
  - Machine Learning
  - Factoring
  - Unstructured Search
  - Eigensystems
  - Linear Systems
- Several physical domains motivate quantum computing as a paradigm for scientific computing
  - High-energy Physics
  - Materials Science
  - Chemistry
  - Biological Systems
  - Artificial Intelligence
  - Data Analytics
  - Planning and Routing
  - Verification and Validation







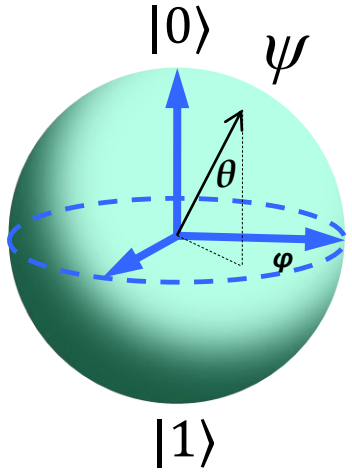




Basic Requirements of a Quantum Computer

D. DiVincenzo, "The Physical Implementation of Quantum Computation," (2000)

Qubit



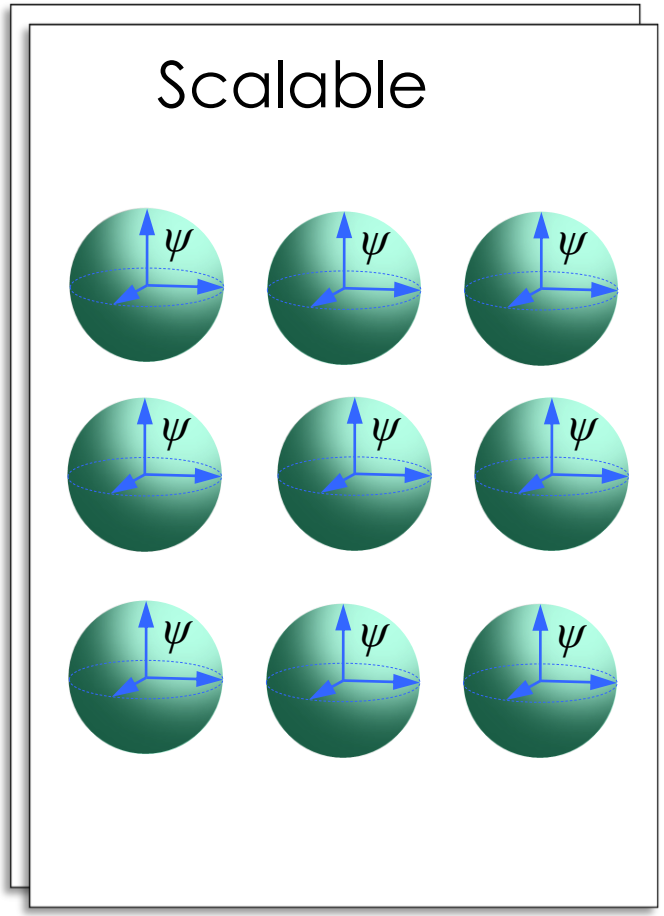
$$\psi = a|0\rangle + b|1\rangle$$

$$a = \cos\theta/2 \quad b = e^{i\phi}\sin\theta/2$$

- Applications require devices with a minimum of hundreds to thousands of logical qubits
- Programmability requires control to act more quickly than decoherence rate
- Resiliency requires fault-tolerant operation through redundant encoding
- System growth requires communication through transport and teleportation

Basic Requirements of a Quantum Computer

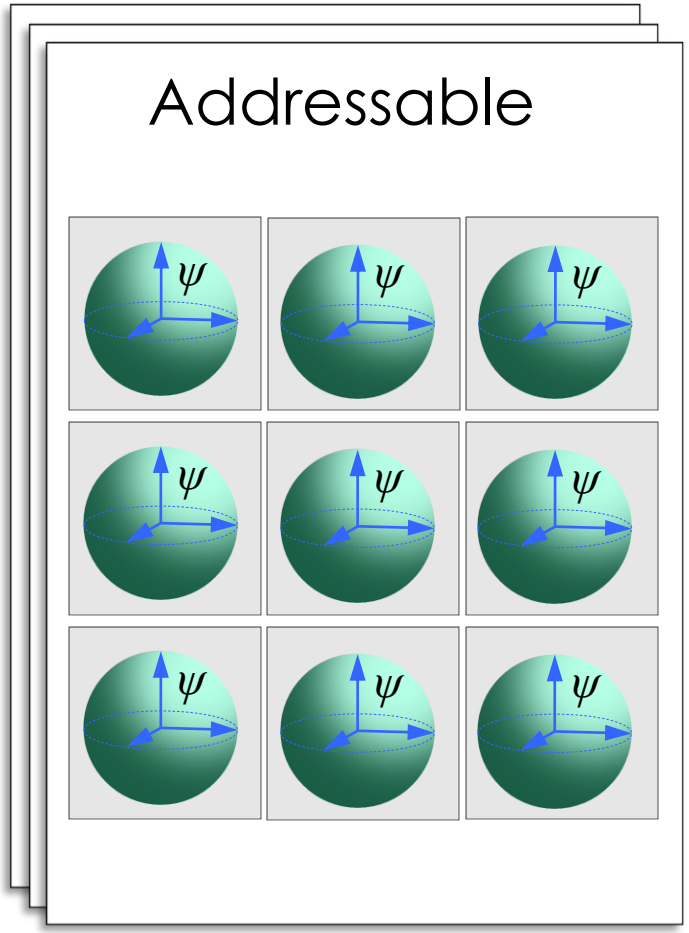
D. DiVincenzo, "The Physical Implementation of Quantum Computation," (2000)



- Applications require devices with a minimum of hundreds to thousands of logical qubits
- Programmability requires control to act more quickly than decoherence rate
- Resiliency requires fault-tolerant operation through redundant encoding
- System growth requires communication through transport and teleportation

Basic Requirements of a Quantum Computer

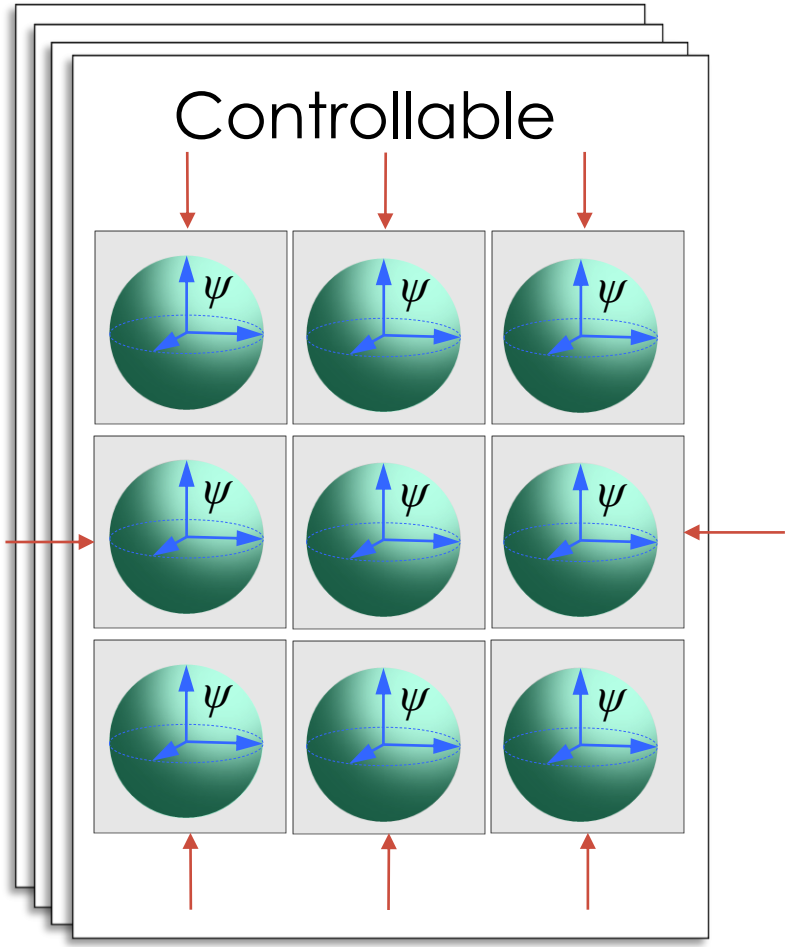
D. DiVincenzo, "The Physical Implementation of Quantum Computation," (2000)



- Applications require devices with a minimum of hundreds to thousands of logical qubits
- Programmability requires control to act more quickly than decoherence rate
- Resiliency requires fault-tolerant operation through redundant encoding
- System growth requires communication through transport and teleportation

Basic Requirements of a Quantum Computer

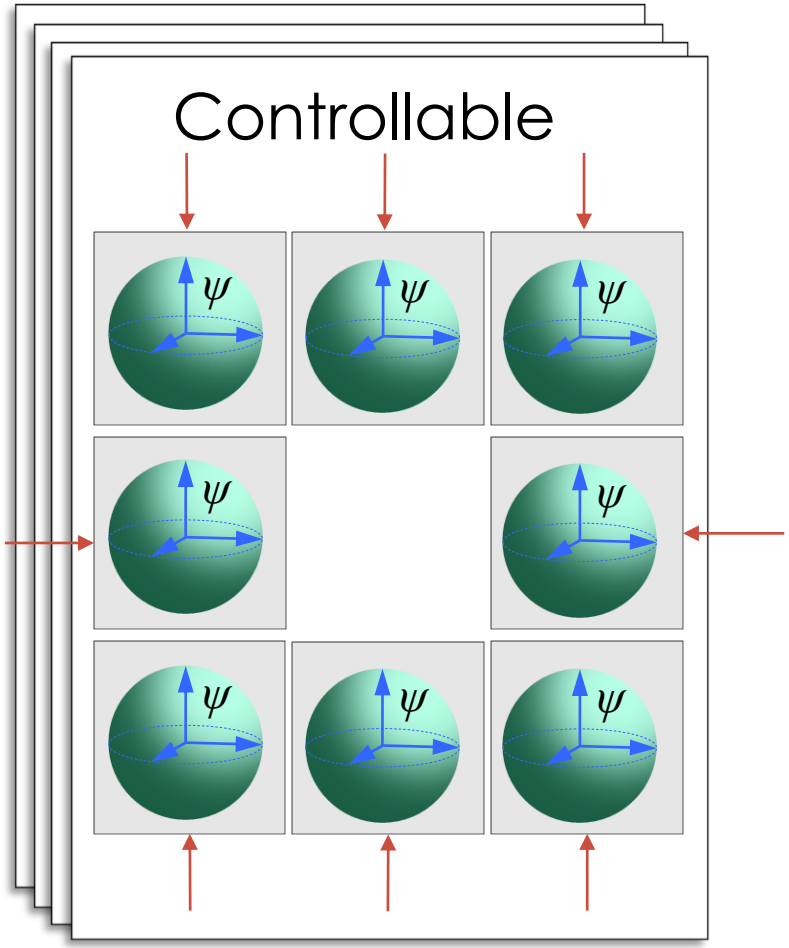
D. DiVincenzo, "The Physical Implementation of Quantum Computation," (2000)



- Applications require devices with a minimum of hundreds to thousands of logical qubits
- Programmability requires control to act more quickly than decoherence rate
- Resiliency requires fault-tolerant operation through redundant encoding
- System growth requires communication through transport and teleportation

Basic Requirements of a Quantum Computer

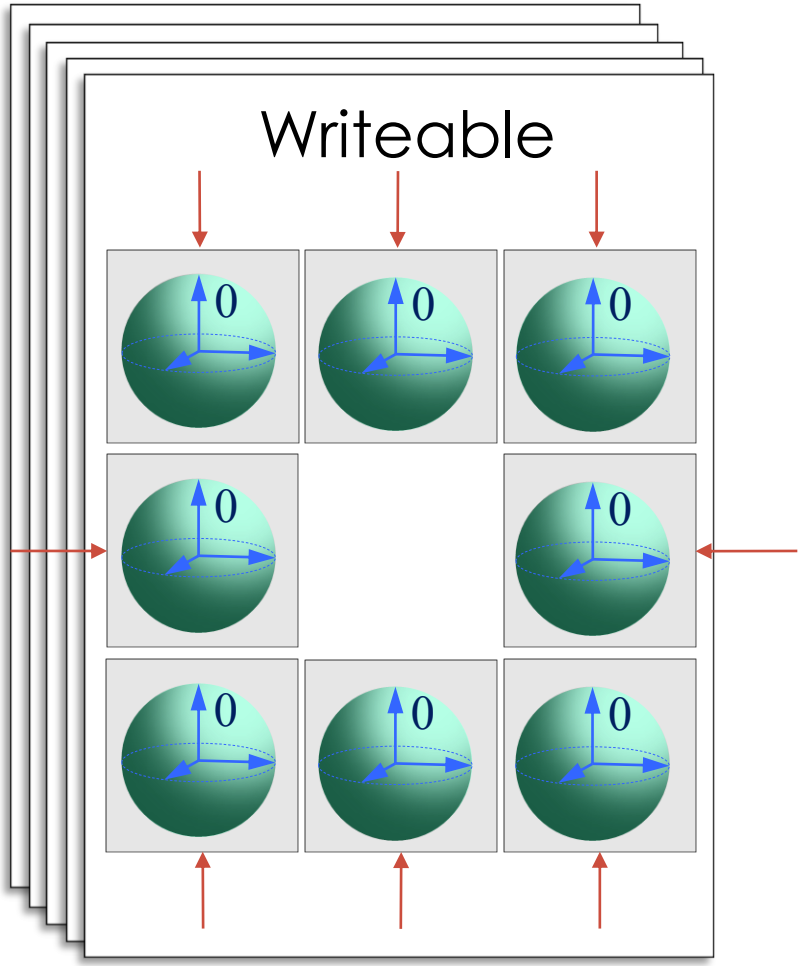
D. DiVincenzo, "The Physical Implementation of Quantum Computation," (2000)



- Applications require devices with a minimum of hundreds to thousands of logical qubits
- Programmability requires control to act more quickly than decoherence rate
- Resiliency requires fault-tolerant operation through redundant encoding
- System growth requires communication through transport and teleportation

Basic Requirements of a Quantum Computer

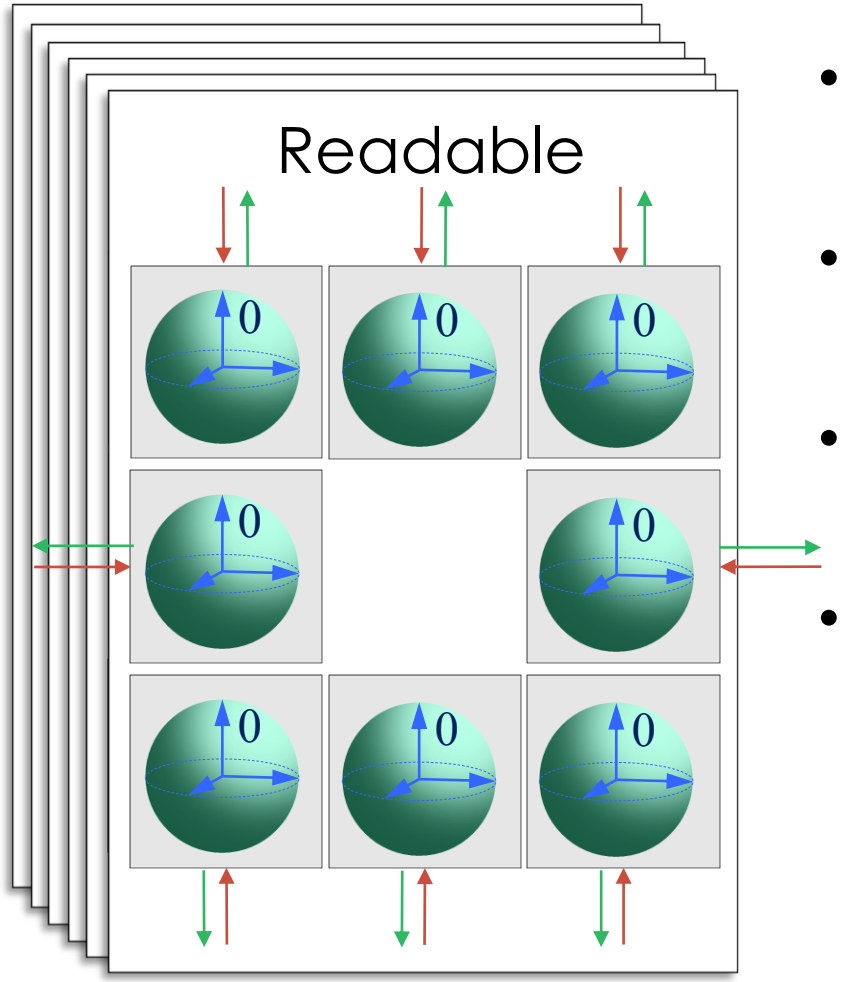
D. DiVincenzo, "The Physical Implementation of Quantum Computation," (2000)



- Applications require devices with a minimum of hundreds to thousands of logical qubits
- Programmability requires control to act more quickly than decoherence rate
- Resiliency requires fault-tolerant operation through redundant encoding
- System growth requires communication through transport and teleportation

Basic Requirements of a Quantum Computer

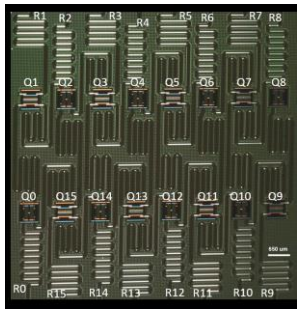
D. DiVincenzo, "The Physical Implementation of Quantum Computation," (2000)



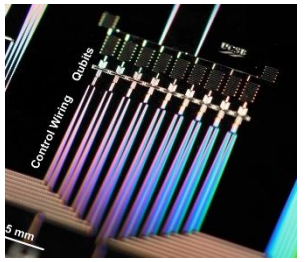
- Applications require devices with a minimum of hundreds to thousands of logical qubits
- Programmability requires control to act more quickly than decoherence rate
- Resiliency requires fault-tolerant operation through redundant encoding
- System growth requires communication through transport and teleportation

Quantum Processing Units

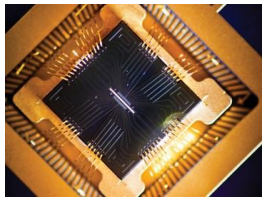
- QPU's are devices that implement the principles of digital quantum computing
  - Several different maturing technologies
  - Small register sizes (0-22)
  - Moderately high control fidelities (0.99+)
- Applications limited by noise, controllability
  - Limited connectivity with good addressability
  - Limited coherence times
  - Poor scalability
- Early stage vendors are offering access
  - D-Wave, IBM, IonQ, Google, Rigetti
  - Client-server interaction, "cloud" model
  - Very loose integration with modern computing



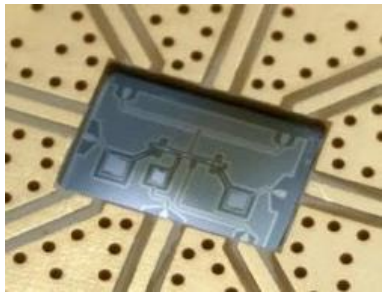
Superconducting chip from IBM



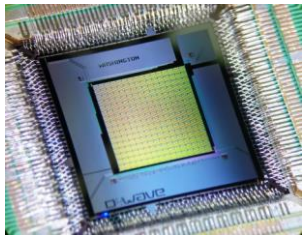
Superconducting chip from Google



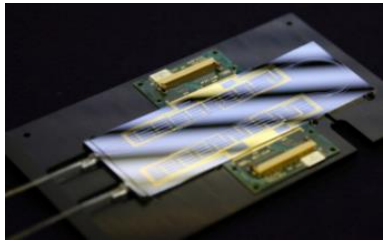
Ion trap chip from Sandia



Superconducting chip from Rigetti



Superconducting chip from D-Wave Systems



Linear optical chip from Univ. Bristol/QET Labs

“Quantum acceleration” as a path beyond exascale

Today’s Systems

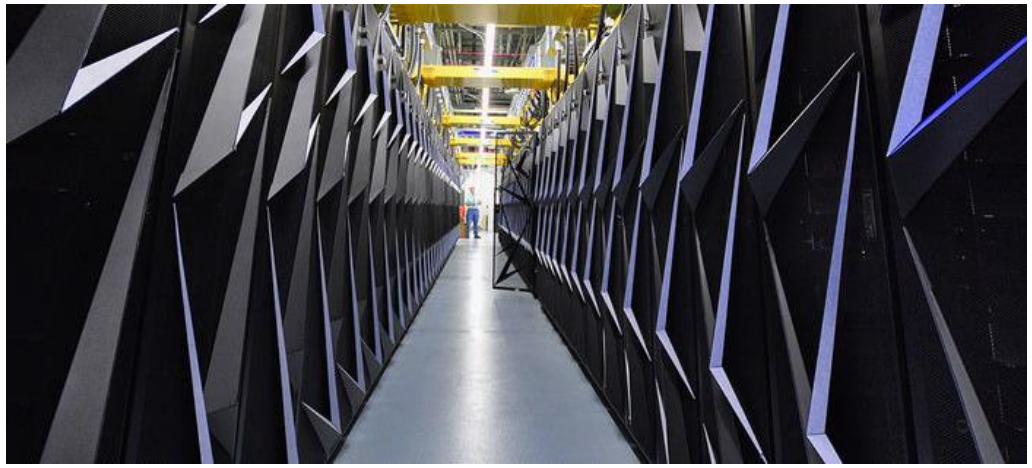
Summit supercomputer with integrated GPU acceleration

Tomorrow’s Technology

Prototype quantum computing hardware from multiple vendors validates the concept of quantum accelerated computing

Modern Scientific Computing

- State-of-the-art scientific computing is dominated by massively parallel processing
  - Large-scale linear algebra, PDES for complex, multi-scale models
  - Application codes are parallelized and capable of utilizing distributed resources
  - Constrained by programming complexity, memory latency, and power consumption

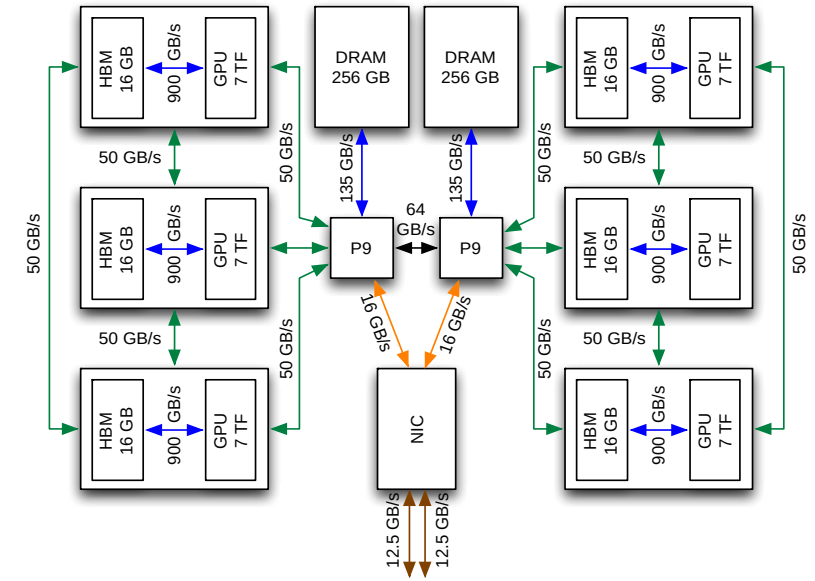


System	R_{max}(PF)	Mem (TiB)	Pow (MW)
Summit	122.3	2,801	8.8
TaihuLight	93.0	1,310	15.4
Sierra	71.6	1,382	UNK
Tianhe-2	33.8	1,375	18.4
ABCI	19.9	340	1.6

Top 5 HPC systems ranked by LINPACK benchmark (Jun 2018)

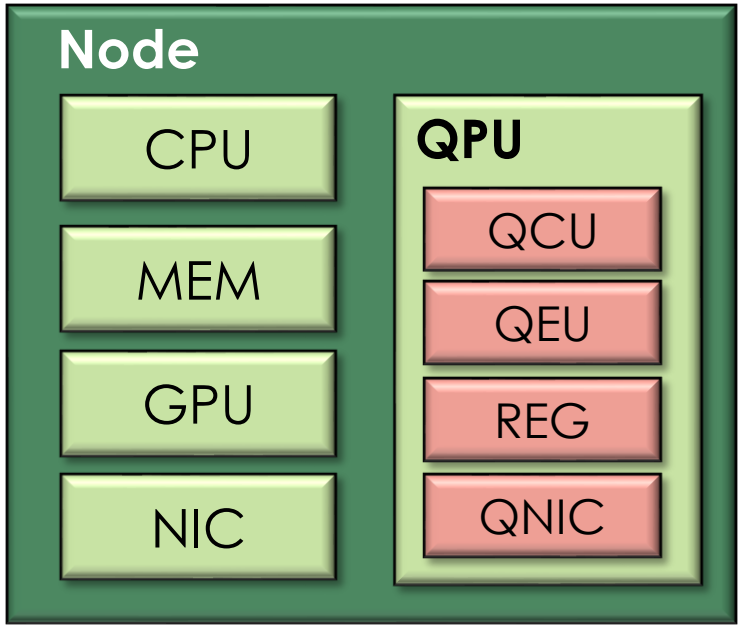
Quantum-accelerated High-Performance Computing

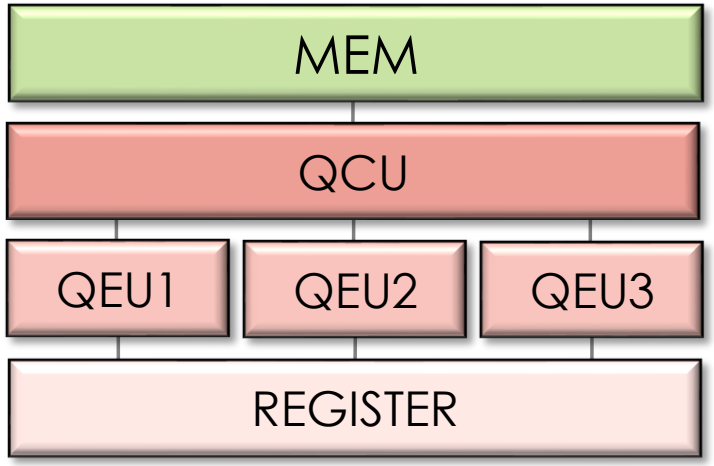
- Are QPUs compatible with modern scientific computing?
  - When do QPU's accelerate applications relative to state of the art HPC?
  - What are the behavioral and functional requirements placed on the processor?
- We are designing quantum-accelerated high-performance computing systems
  - We are testing and tuning new programming and execution models
  - We are benchmarking against physical, computational, and scientific metrics



Quantum Accelerator Node: Component Model

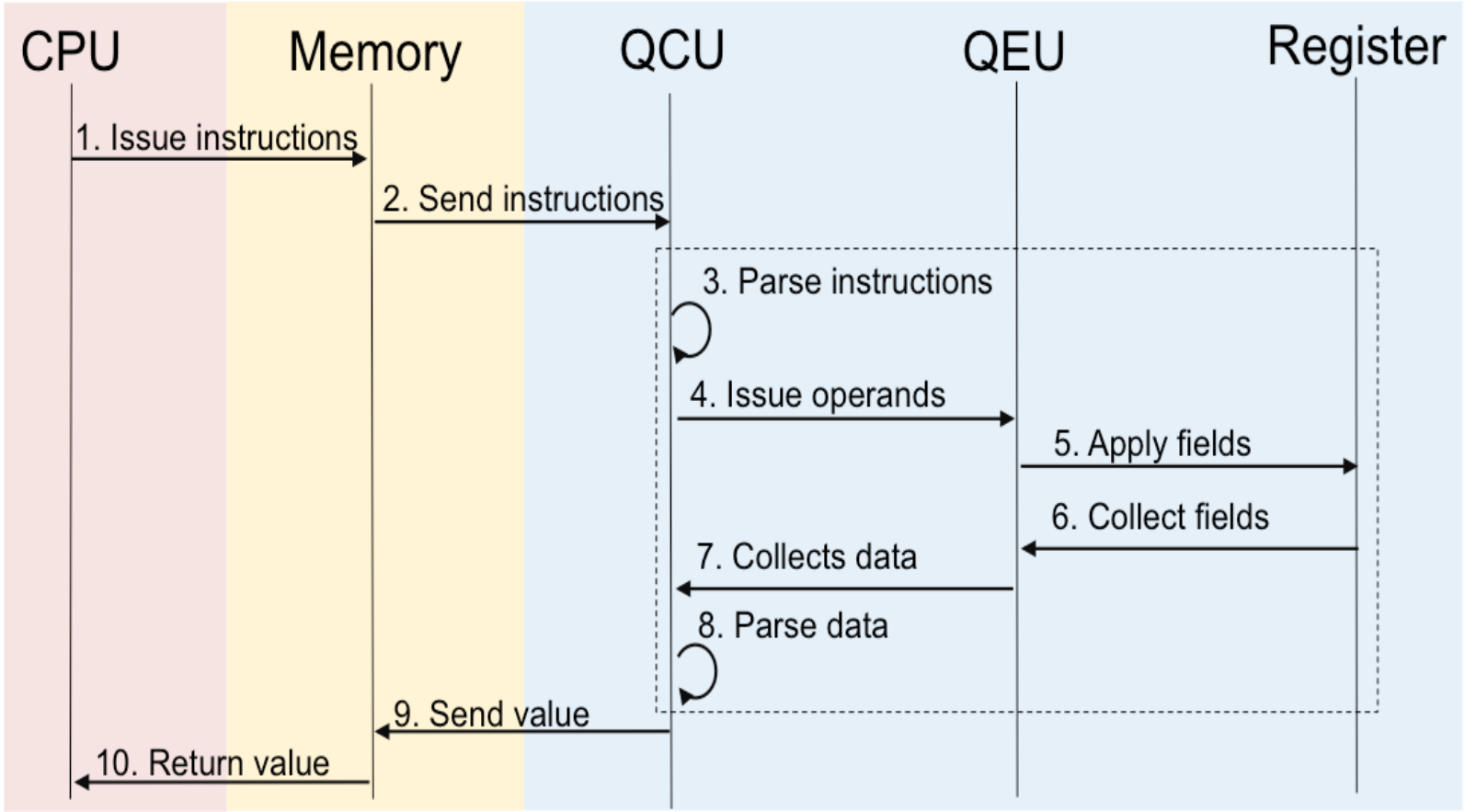
- A **node** may be composed from CPUs GPUs, and memory hierarchies as well as QPUs.
- The **quantum processing unit** (QPU) encompasses methods for parsing and executing quantum programs.
- The **quantum control unit** (QCU) parses instruction sent by the CPU.
- A **quantum execution unit** (QEU) applies fields to initiate gates. There may be multiple QEU's.
- Applied fields drive changes in the **quantum register**. The register state stores the value of the computation.
- **I/O** is based on fields to prepare and measure the register in computational basis states.
- **Network interfaces** for the conventional (NIC) and quantum (QNIC) interconnects support communication





Quantum Accelerator Node: Execution Model

- Quantum execution models define how the HPC system carries out the program instructions

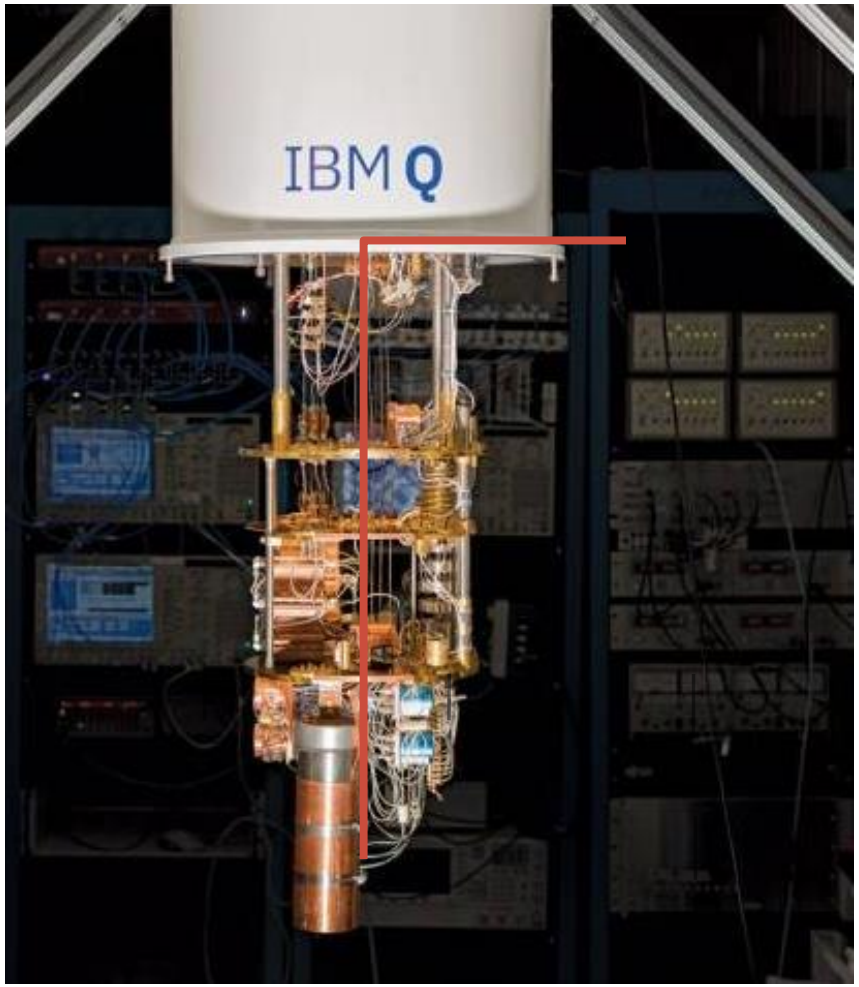


Language Hierarchy

- **Programming Language**
- **Executable Binary**
- **Instruction Set Architecture**
- **Opcodes**
- **Gate Fields**

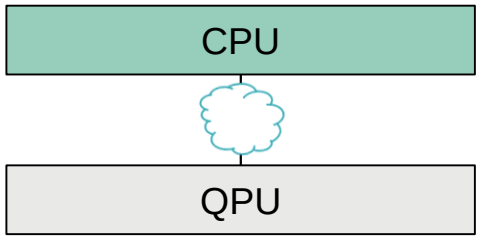
Quantum Accelerator Model

- Example: IBM Q QPU
  - Programmable integrated superconducting circuit with 5, 16, and 20-qubit register
  - Electromagnetic shielding, ultra-high vacuum, cryogenically cooled to ~13mK
  - General-purpose, gate model operation
  - Gate errors<1%, measure errors ~ 5%
  - Authentication and management
  - Job queuing system
  - Cloud-based access, HTTPS



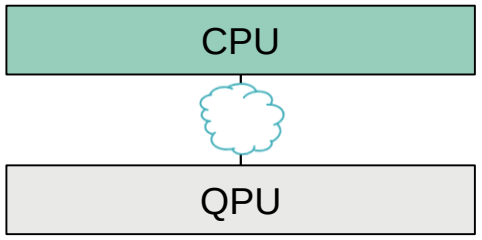
Quantum Accelerator Model

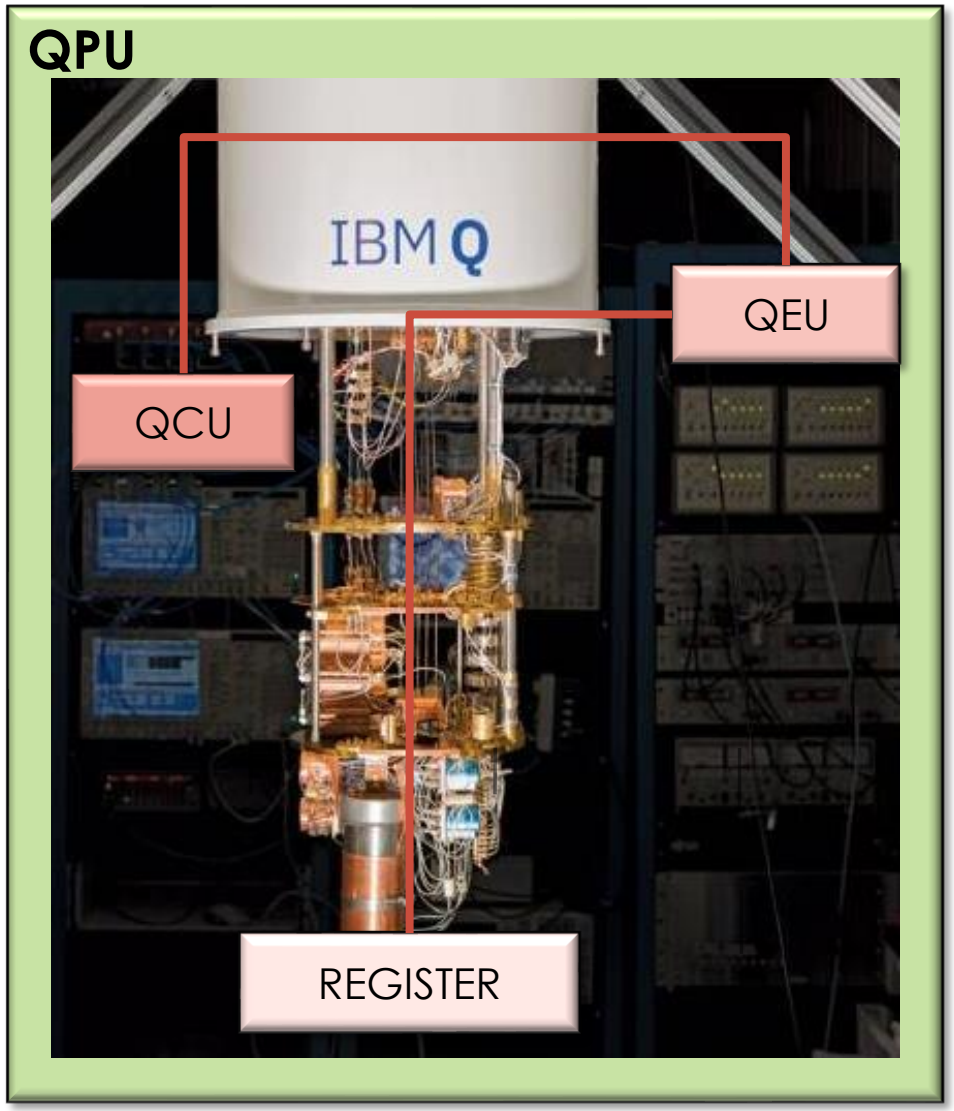
- Example: IBM Q QPU
  - Programmable integrated superconducting circuit with 5, 16, and 20-qubit register
  - Electromagnetic shielding, ultra-high vacuum, cryogenically cooled to ~13mK
  - General-purpose, gate model operation
  - Gate errors<1%, measure errors ~ 5%
  - Authentication and management
  - Job queuing system
  - Cloud-based access, HTTPS



Quantum Accelerator Model

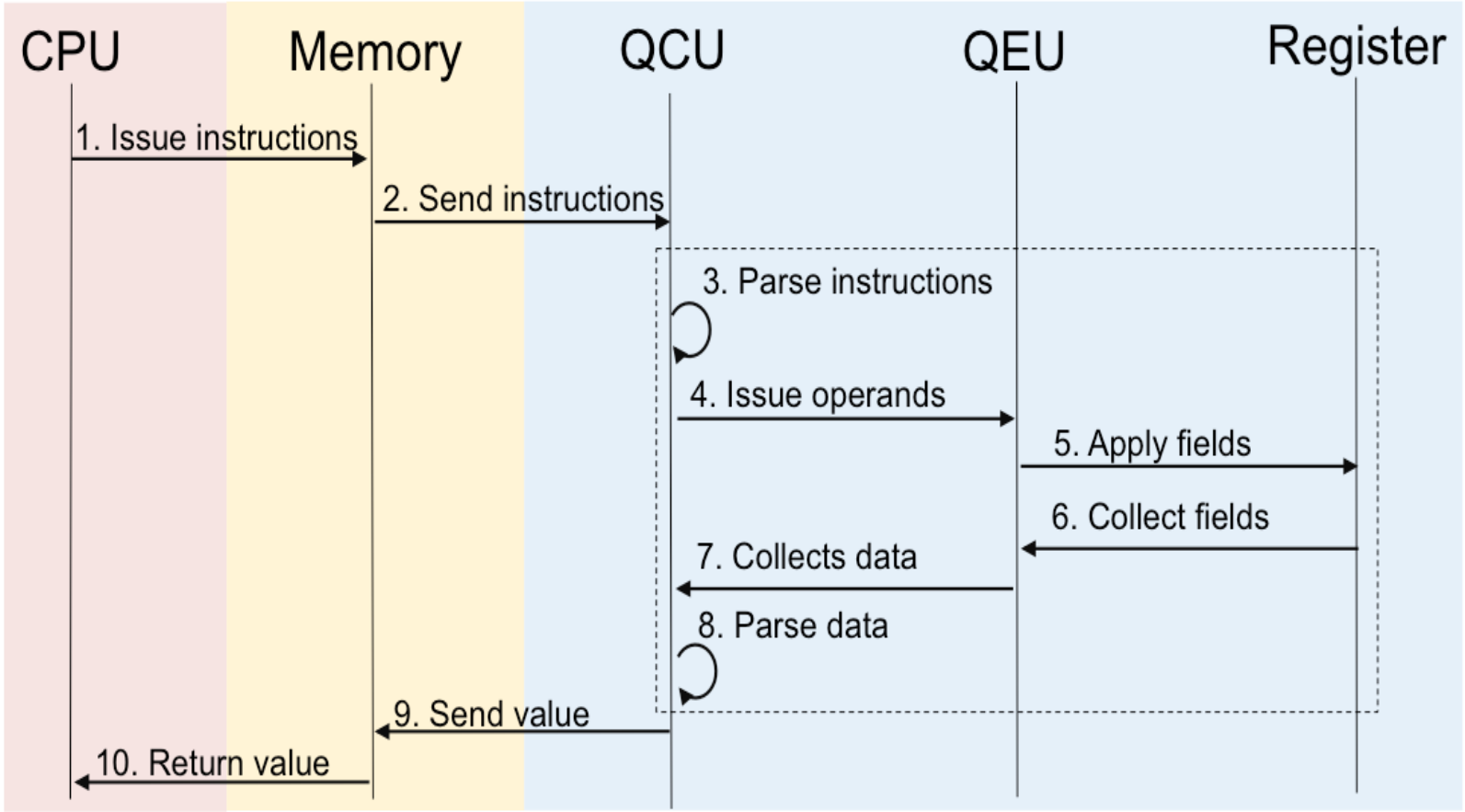
- Example: IBM Q QPU
  - Programmable integrated superconducting circuit with 5, 16, and 20-qubit register
  - Electromagnetic shielding, ultra-high vacuum, cryogenically cooled to ~13mK
  - General-purpose, gate model operation
  - Gate errors<1%, measure errors ~ 5%
  - Authentication and management
  - Job queuing system
  - Cloud-based access, HTTPS





Quantum Accelerator Node: Execution Model

- Quantum execution models define how the HPC system carries out the program instructions

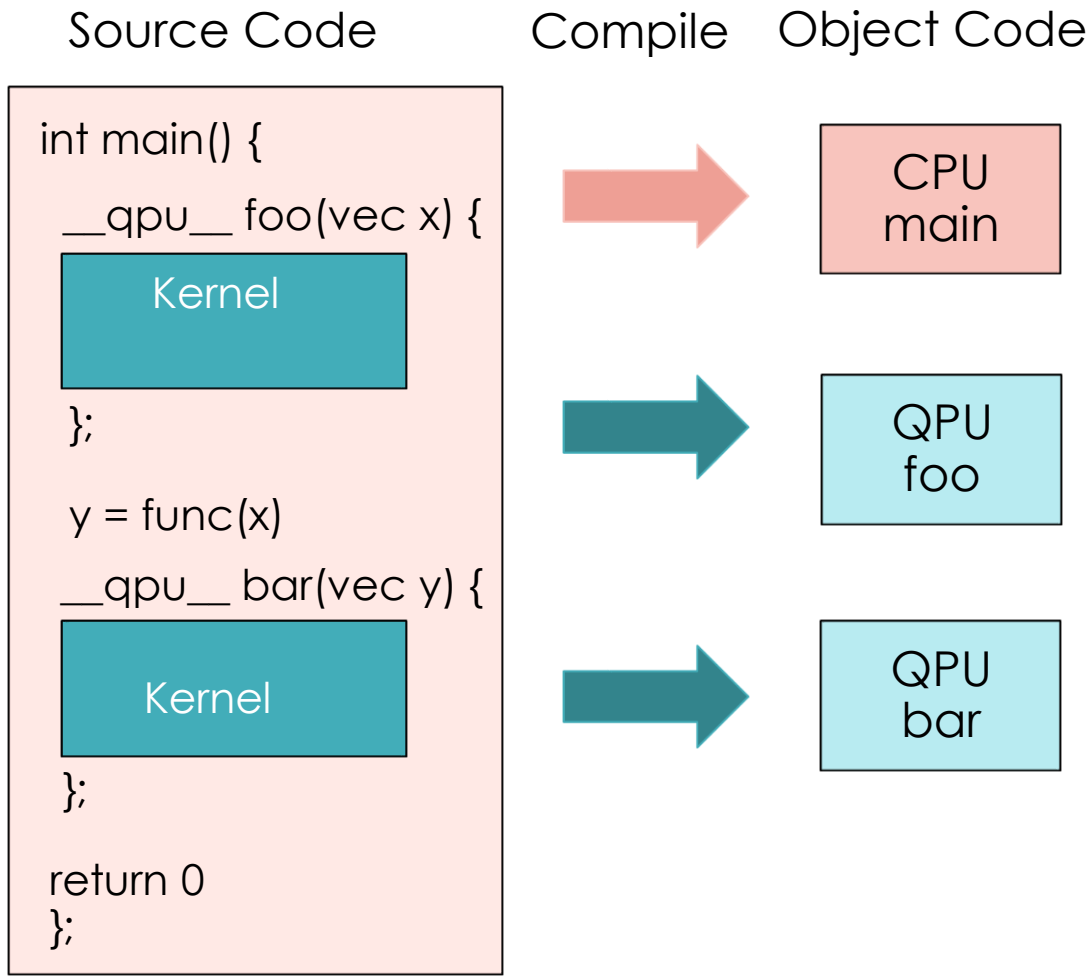


Language Hierarchy

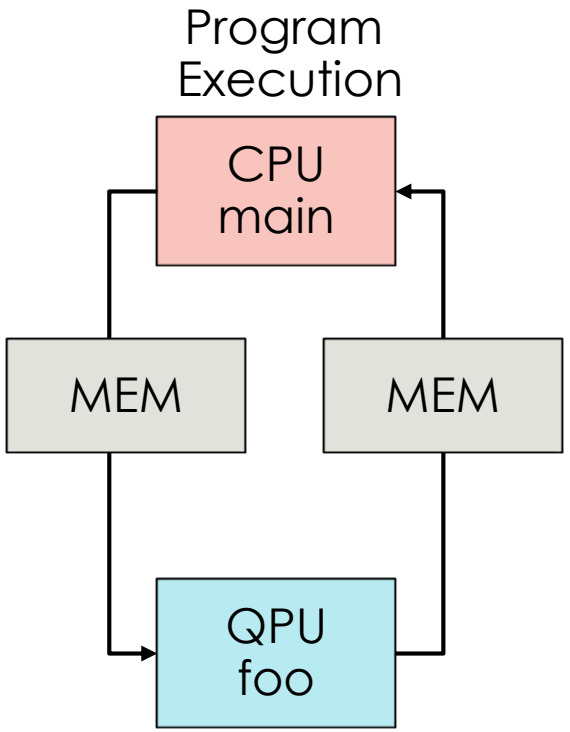
- **Programming Language**
- **Executable Binary**
- **Instruction Set Architecture**
- **Opcodes**
- **Gate Fields**

Quantum Accelerator Node: Programming Model

- Program flow is controlled by the CPU
  - Directive-based languages permits offloading
  - Quantum computational kernels are compiled to a target device representation
  - Memory buffer is 'shared'
- XACC is a mixed-language, directive-based framework
  - LLVM toolchain triggers backend compilers
  - Example: C/C++ host program with Quil kernels compiled to OpenQASM for IBM QPU
  - Intermediate representations enable transformations that target device



Quantum Accelerator Node: Programming Model

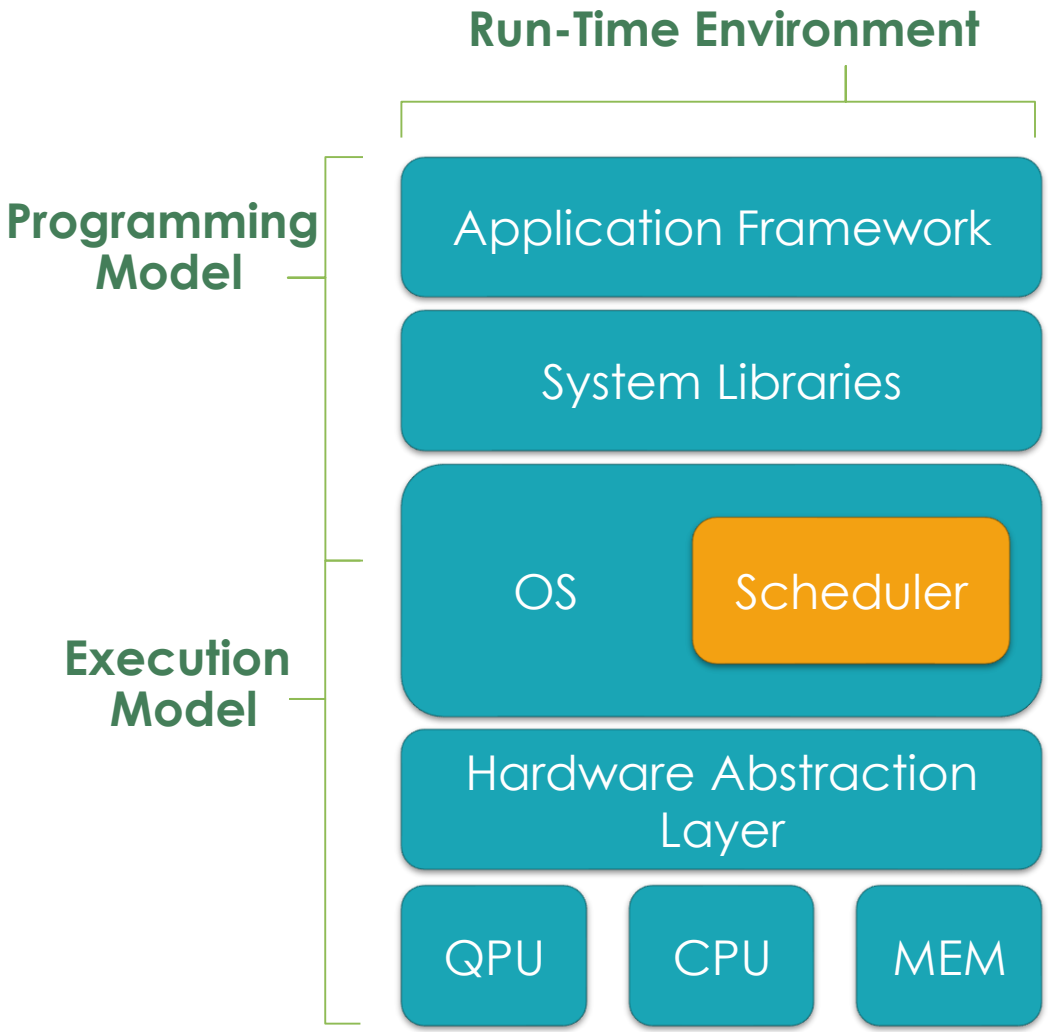



```
// Get reference to the QPU, and allocate a buffer of qubits
auto qpu = xacc::getAccelerator("ibm");
auto buffer = qpu->createBuffer("qreg", 2);

// Create and compile the Program from the kernel
// source code. Get executable Kernel source code
auto kernelSourceCode = "__qpu__ foo(double theta) {....}";
xacc::Program program(qpu, kernelSourceCode);
program.build();
auto kernel = program.getKernel<double>("foo");

// Execute over theta range
for (auto theta : thetas) kernel.execute(buffer, theta);
```

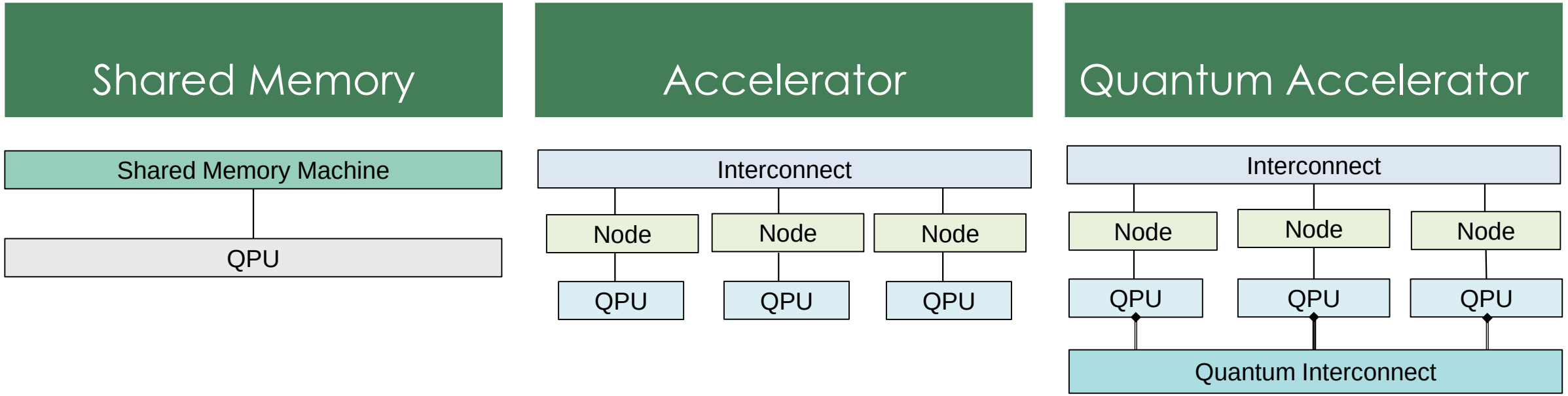
https://ornl-qci.github.io/xacc/

Quantum Accelerator Node: Run-time Environment

- At run-time, the node must manage the QPU and schedule instructions for execution
  - Current QPUs are loosely integrated with host node, e.g., client-server interactions
  - Server backend serves role of OS, scheduler
- The run-time system manages resources, errors, permissions, and execution
  - CPU issues instructions to the QCU via memory managed by the OS
  - Some program control statements require measurement feedback for CPU evaluation



Quantum Accelerated HPC: System Architectures

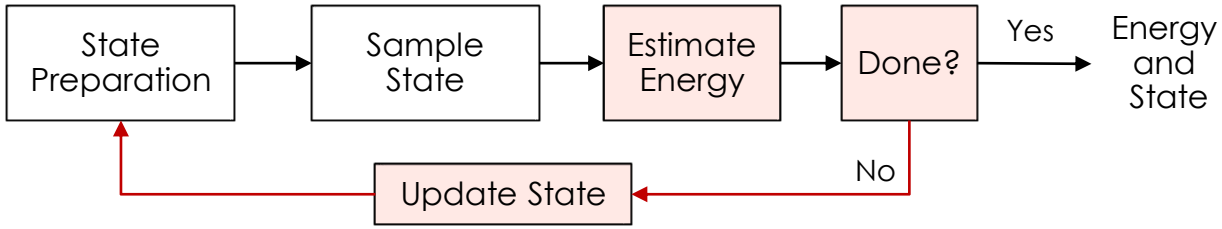


- Machine architecture differentiates how quantum processing can be allocated.
- Application performance depends on domain decomposition and algorithmic design.
- Tighter technology integration is advancing rapidly driven by progress in quantum control.

Quantum Accelerated HPC: Application Example

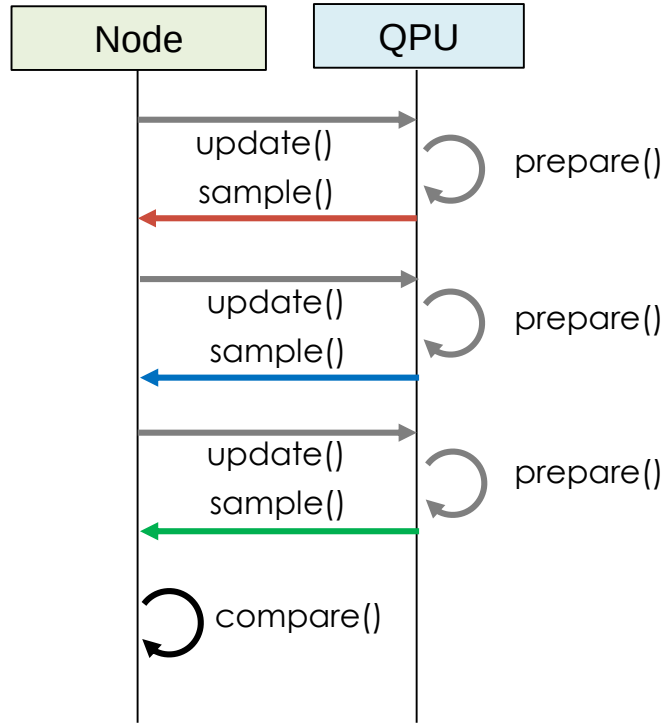
- Variational Quantum Eigensolver is a sampling method used for multiple problem types
  - Chemistry, Nuclear Physics, HEP, etc.

Input: Hamiltonian and Hilbert space
Output: Ground state and energy



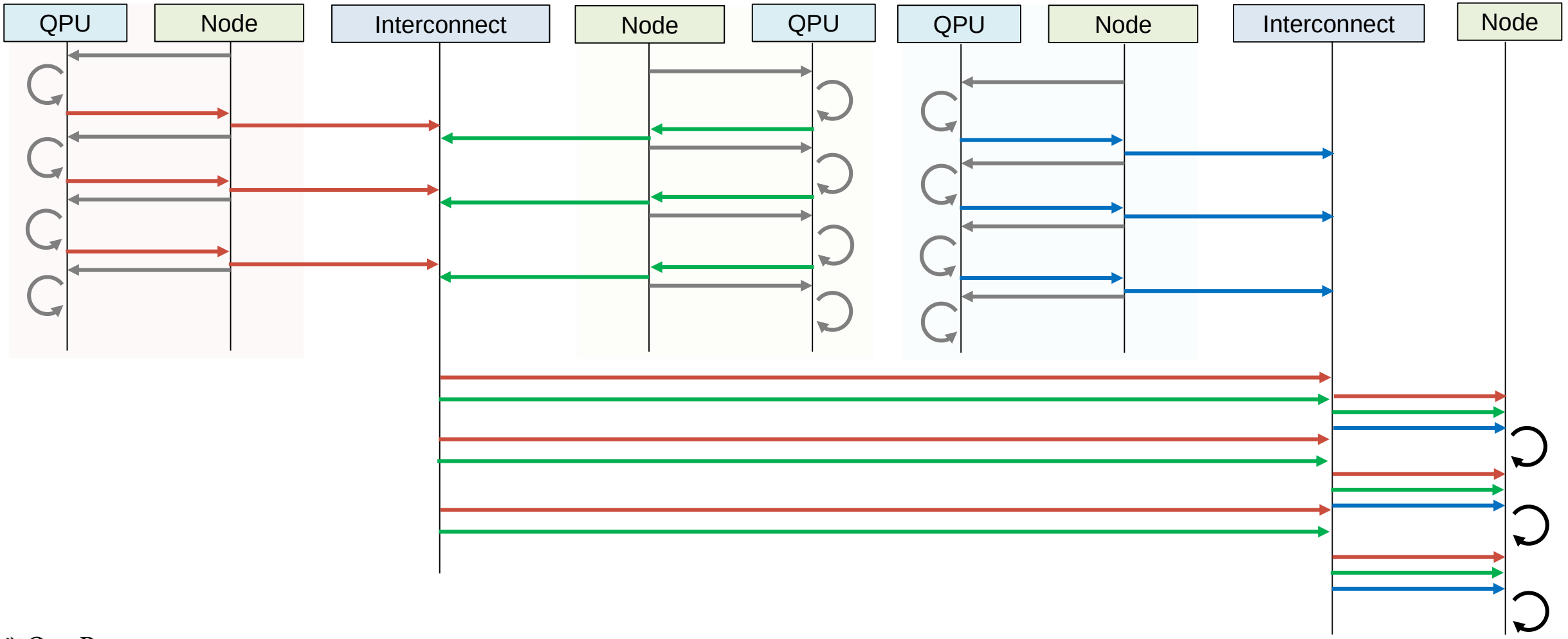
  - Accuracy determined by state preparation
  - Precision determined by sample size
  - Time dominated by search and sampling

Serial Execution



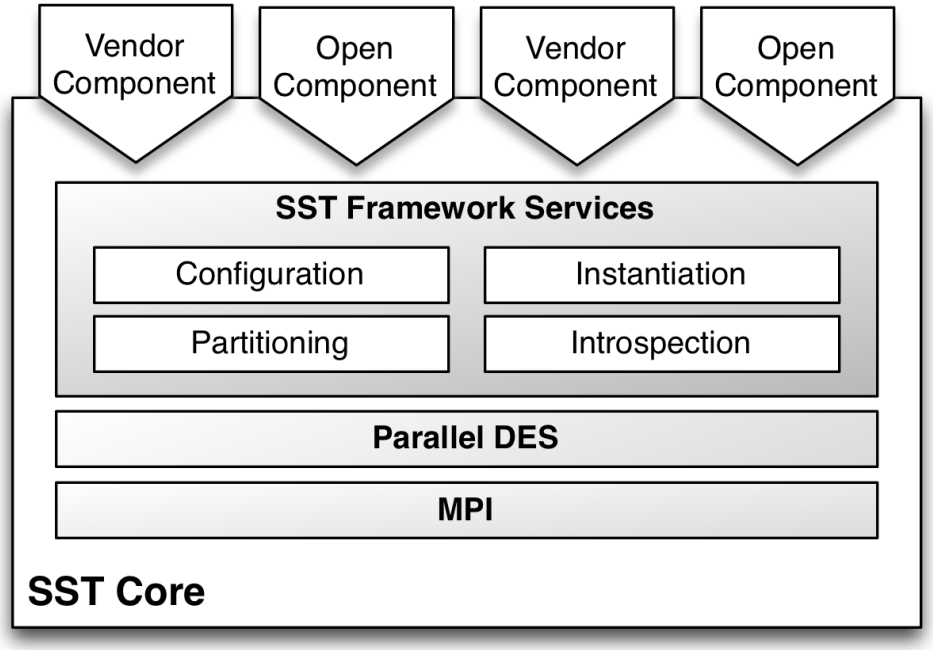
Quantum Accelerated HPC: Application Example

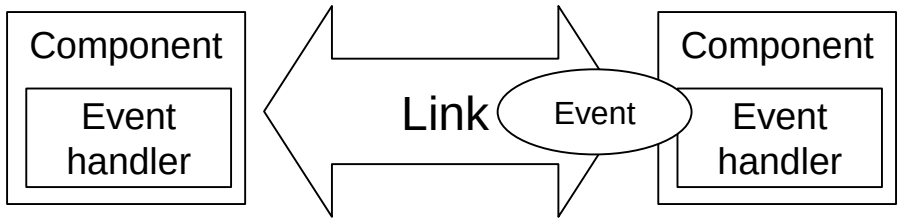
Parallel Execution Model with Message Passing



Modeling and Simulation of Quantum Accelerated HPC

- We use the Structural Simulation Toolkit (SST) to model nodes, memory, network
  - SST is a discrete-event simulations used to model conventional computing systems
  - It has existing models that account for data movement, latency, and power consumption
  - We use it to profile applications against architecture and device parameters
- We are building models for QA-HPC and simulating these using SST
  - CPU-QPU
  - CPU-MEM-QPU
  - (CPU+CPU)-MEM-QPU
  - (CPU+CPU)-MEM-(QPU+QPU)

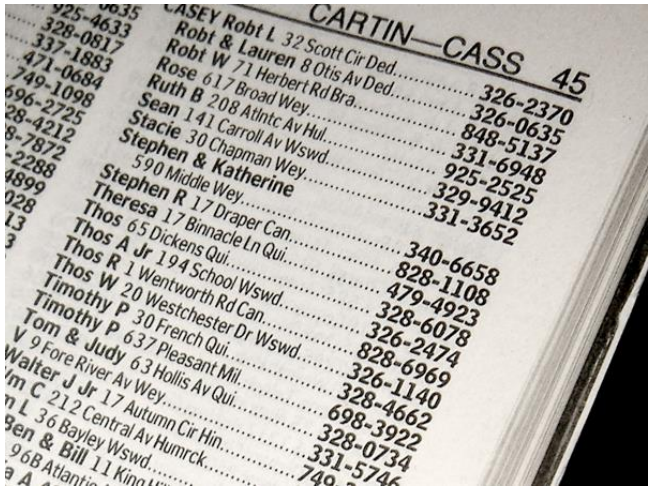




http://sst-simulator.org/

Test Case: Energy Requirements for Unstructured Search

- Problem: Find a specific item in an unstructured database
  - The optimal classical algorithm to find a marked item requires *N*/2 queries for an *N*-item database
  - Parallelizable across *K* system nodes with *K*-fold gather as the last step
- Quantum search is a method for finding an item in an unstructured database
  - First proposed by Grover (1996)
  - Sqrt(*N*) queries to find a single marked item
  - Sqrt(*N*/*M*) queries to find one of *M* marked items
  - Partial search: decompose *N* into *K* subsets, find the subset containing the marked item
- What are the energy requirements?



Example: Function inversion
$y = h(x) \Rightarrow h^{-1}(y) = x$

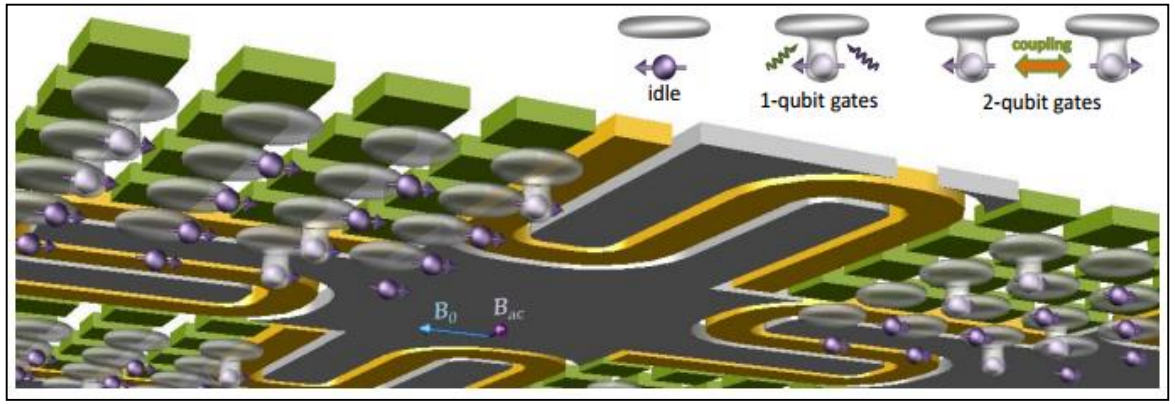
Bitcoin mining
Hash(x) = SHA256(SHA256(x))
x => 4 bytes = nonce

Estimating Energy Costs for a Silicon Quantum Computer

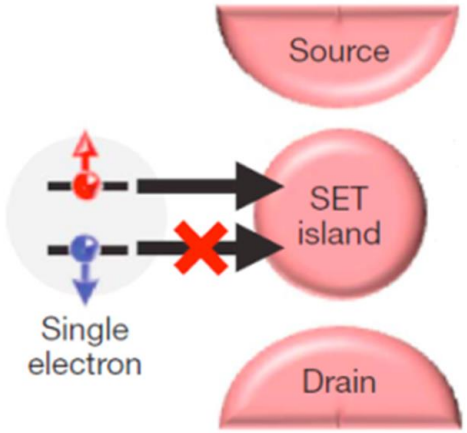
- Silicon technology architecture
  - Two-qubit gates induced through long-range resonators
  - Single-electron transistor (SET) is used to readout and initialize spin state
  - Tosi et al., Nature Comms 8, 450 (2017)

Gate time and energy

Qubit	Time	Power	Energy
Flip-flop	40 ns	0.1 pW	4 zJ



Measurement time and energy

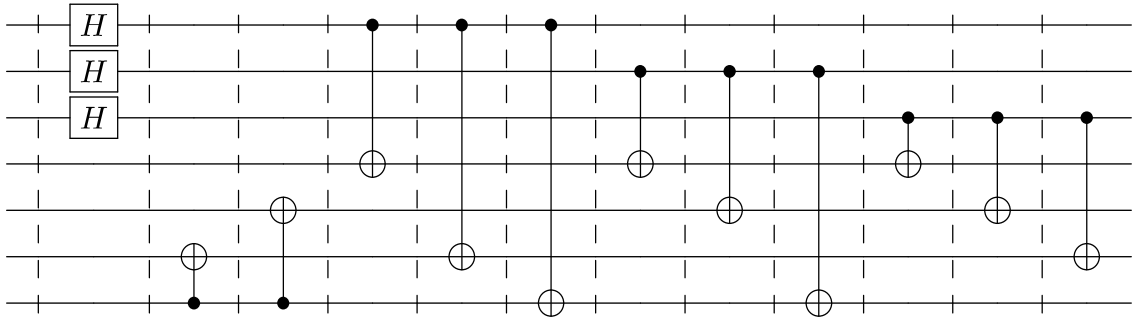


V_{SD} = 100 uV
I_{SET} = 1 nA

T_{read} = 100 us
E_{read} = 5 aJ

T_{init} = 300 us
E_{init} = 5 aJ

Fault-tolerant quantum error correction



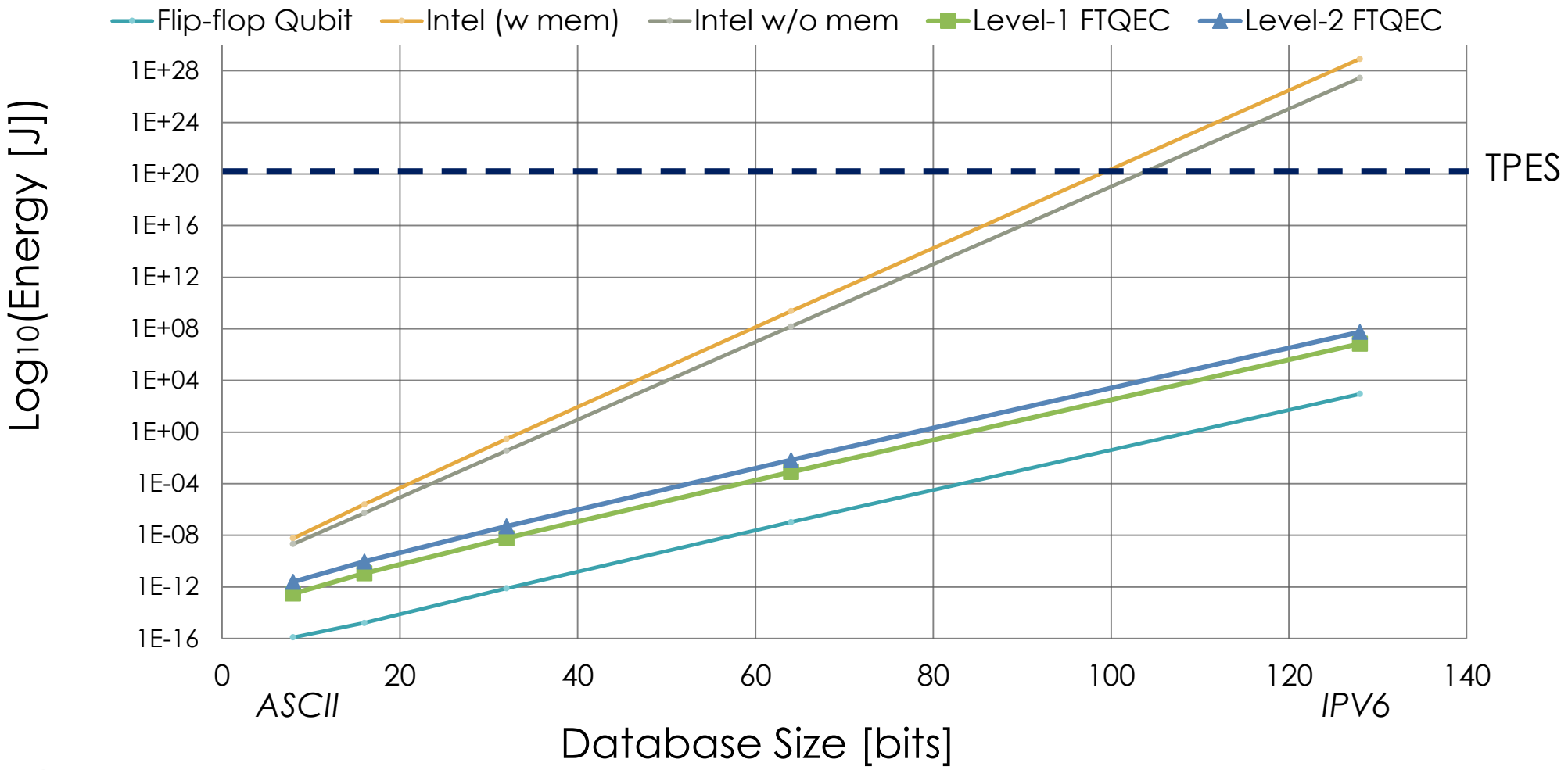
Conventional Computing Baseline

- A linear search application
  - We compile a loop over string comparisons into assembly instructions
  - 6 Instructions per iteration
  - *N*/2 loop iterations, average-case
- We estimate energy per instruction
  - Intel Core i7-6700K
  - 2.73 pJ Energy Per Instruction (EPI)
    - 1.35V, 1.5 pF, 91W
  - 3.7pJ/bit for DRAM read (best)
    - Deng et al., ASPLOS 2011
  - log(*N*) bits per read

```
1   start:   cmp bx,[si]
2            je  found
3            add si,2
4            inc cx
5            dec dx
6            jnz start
```

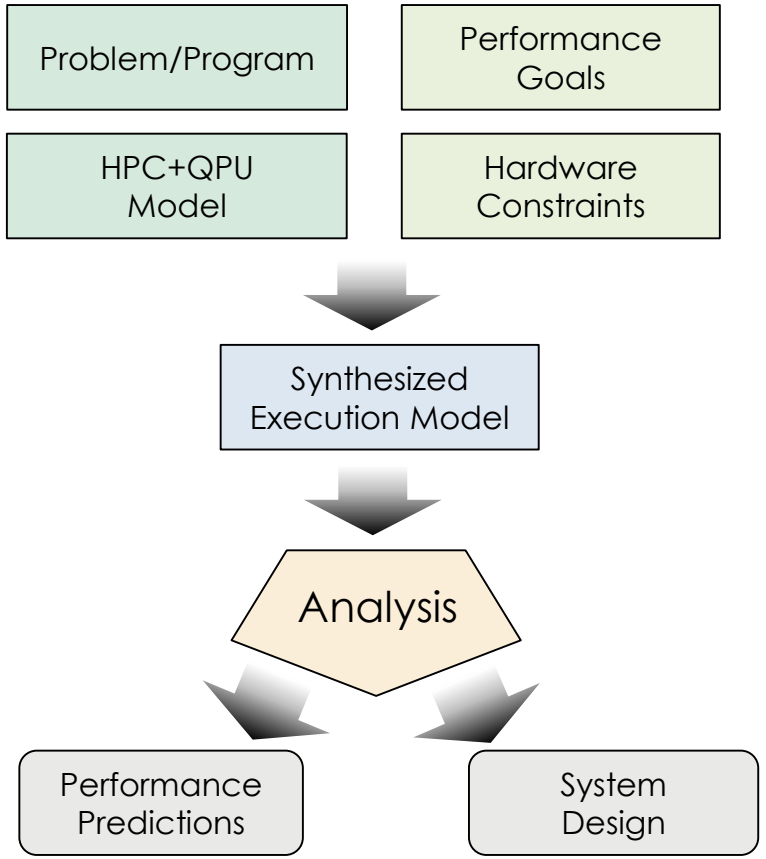
CPU vs QPU Energy Estimates (Steane FTQEC)

- Our comparative analysis reveals an exponential separation in total energy costs



Overarching Goals for Quantum Accelerator Research

- Develop quantum execution models for HPC systems and applications using abstract machine models and simulations
- Simulate quantum run-time environments for QPU's and quantum interconnects within machine models with networking
- Design quantum networking protocols for interconnected QPU's using software-defined networking principles
- Demonstrate resiliency of quantum program and run-time models using quantum network coding
- Define methods and metrics for benchmarking quantum-enabled HPC systems based on resource consumption
- Predict performance and scaling of scientific applications using programs executed within hybridized machine models



Broader Community Engagement

Software Working Group

- With Eclipse Foundation
- A user community to develop use cases, requirements, and application profiling across system stacks.

Developers
Programmers
Engineers
Vendors

https://code.ornl.gov/QCSWG

Device Benchmarking Group

- With IEEE Rebooting Computing Initiative
- A community to define and evaluate metrics for quantum processors

Vendors
End-Users
International
Engineers

http://icrc.ieee.org/

Green Quantum Computing

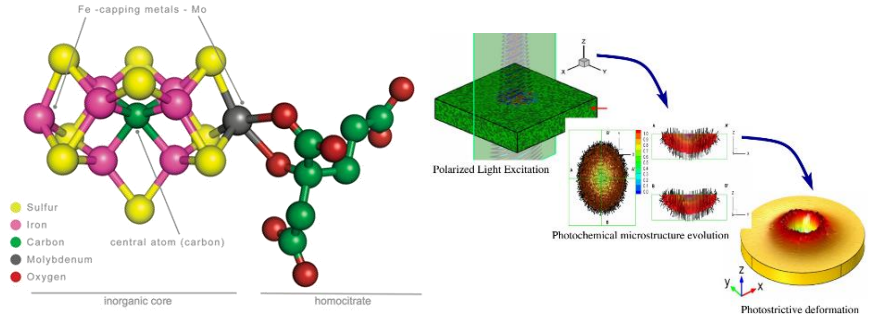
- With International Green and Sustainable Computing Conference
- Toward Quantum Computing for Sustainable Computing

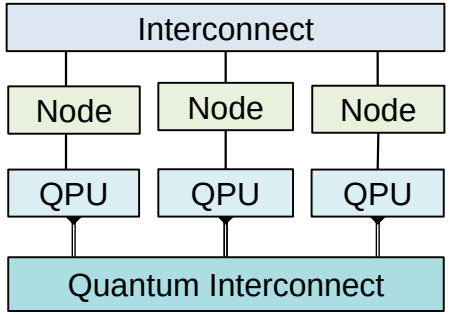
Scientists
Engineers
System Admins
Architects

https://www.igscc.org/

In Conclusion…

- We are using modeling and simulations to design and demonstrate Quantum Accelerated HPC
- New models for programming, execution, and run-time are used to evaluate this potential
- Continuing research addresses model refinement, new technologies, and applications
- Quantum acceleration may provide a path to beyond exascale computing
- But evaluating quantum computing systems requires broad support for this new technology





Our weekly newsletter provides the latest updates!
quantum@ornl.gov

Discussion