COE 530
Quantum Computers and Architecture

Lecture 18

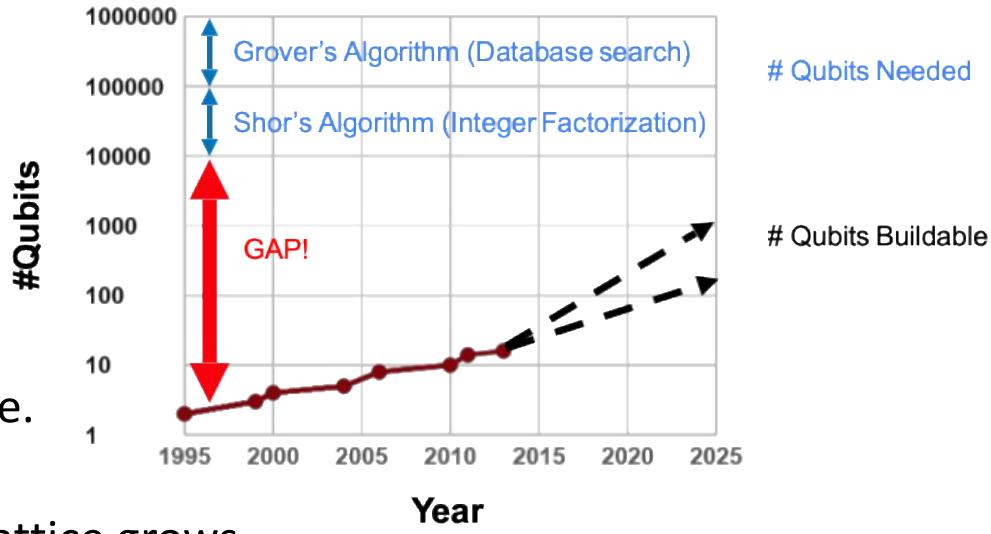
Distributed Quantum Computing

References:

Caleffi et al. (2024) — Distributed quantum computing: A survey. Computer Networks 254, 110672.

Limitations of Monolithic Quantum Computing

- Practical applications need ~10^6 – 10^7 physical qubits
- Today's NISQ chips ship 10^2 – 10^3



- A single QPU can't simply keep growing
  - **Cryogenic & control bottlenecks**
    - A single dilution refrigerator dissipates only mW at 15 mK
    - Control wiring scales linearly with qubits and saturates the fridge.
  - **Crosstalk & yield**
    - Cross-resonance and ZZ couplings degrade gate fidelity as the lattice grows
    - Chip yield falls super-linearly with die area
  - **Connectivity collapse**
    - Fixed 2D coupling maps force long SWAP chains
    - Circuit depth blows up super-linearly with width
  - **Manufacturing physics**
    - Reticle limits (~26 mm × 33 mm) cap monolithic die size

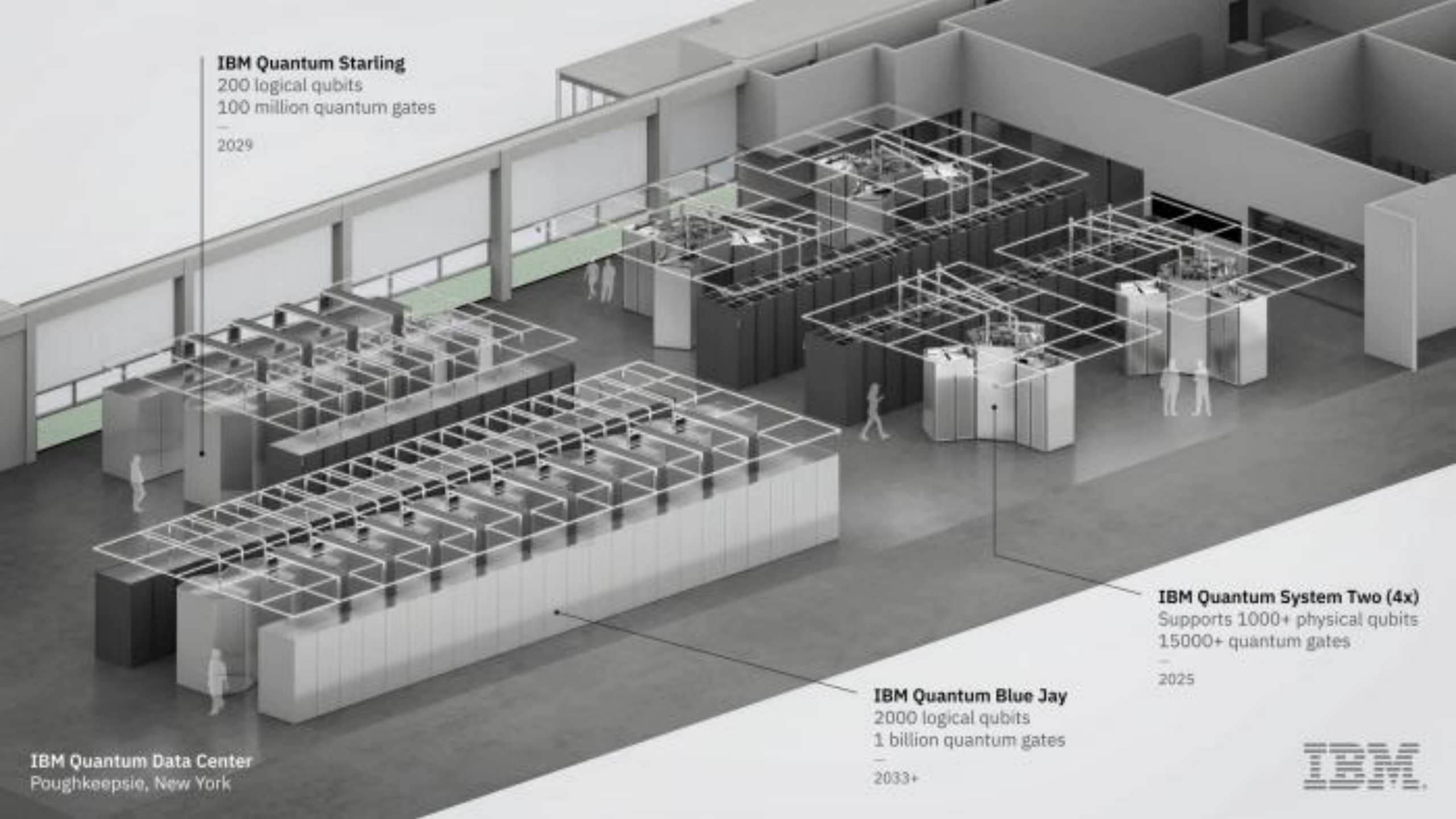
Development Roadmap

IBM **Quantum**

	2016–2019 ✓	2020 ✓	2021 ✓	2022 ✓	2023 ✓	2024	2025	2026	2027	2028	2029	2033+
	Ran quantum circuits on the IBM Quantum Platform	Released multi-dimensional roadmap publicly with initial aim focused on scaling	Enhanced quantum execution speed by 100x with Qiskit Runtime	Brought dynamic circuits to unlock more computations	Enhanced quantum execution speed by 5x with Quantum Serverless and execution modes	Improve quantum circuit quality and speed to allow 5K gates with parametric circuits	Enhance quantum execution speed and parallelization with partitioning and quantum modularity	Improve quantum circuit quality to allow 7.5K gates	Improve quantum circuit quality to allow 10K gates	Improve quantum circuit quality to allow 15K gates	Improve quantum circuit quality to allow 100M gates	Beyond 2033, quantum-centric supercomputers will include 1000's of logical qubits unlocking the full power of quantum computing
Data scientists						Platform						
						Code assistant ◉	Functions	Mapping collections	Specific libraries			General purpose QC libraries
Researchers					Middleware							
					Quantum Serverless ✓	Transpiler service ◉	Resource management	Circuit knitting x p	Intelligent orchestration			Circuit libraries
Quantum physicists			Qiskit Runtime									
	IBM Quantum Experience ✓		QASM 3 ✓	Dynamic circuits ✓	Execution modes ✓	Heron (5K) ◉	Flamingo (5K)	Flamingo (7.5K)	Flamingo (10K)	Flamingo (15K)	Starling (100M)	Blue Jay (1B)
	Early ✓ Canary 5 qubits Albatross 16 qubits Penguin 20 qubits Prototype 53 qubits	**Falcon** ✓ Benchmarking 27 qubits		**Eagle** ✓ Benchmarking 127 qubits		Error mitigation 5k gates 133 qubits Classical modular Up to 133x3 = 399 qubits	Error mitigation 5k gates 156 qubits Quantum modular Up to 156x7 = 1092 qubits	Error mitigation 7.5k gates 156 qubits Quantum modular Up to 156x7 = 1092 qubits	Error mitigation 10k gates 156 qubits Quantum modular Up to 156x7 = 1092 qubits	Error mitigation 15k gates 156 qubits Quantum modular Up to 156x7 = 1092 qubits	Error correction 100M gates 200 qubits Error corrected modularity	Error correction 1B gates 2000 qubits Error corrected modularity

✓ Executed by IBM

◉ On target


IBM Quantum Starling
200 logical qubits
100 million quantum gates
2029
IBM Quantum System Two (4x)
Supports 1000+ physical qubits
15000+ quantum gates
2025
IBM Quantum Blue Jay
2000 logical qubits
1 billion quantum gates
2033+
IBM Quantum Data Center
Poughkeepsie, New York

IonQ's Interconnect Roadmap

IONQ

Interconnect Generation & Scaling Trajectory (2026-2028)





	2026	2027	2028
INTERCONNECT GENERATION	**Speed-up** High-speed connections between QPU's	**Scale-up** Scaling to multiple lanes	**Scale-out** Connect fault-tolerant QC's Support long distance connections
RANGE	Datacenter Scale	Metropolitan scale (30 km)	Regional Scale (100+ km)
COMPATIBLE QPUS	Ions	Ions, Neutral Atoms	Ions, Neutral Atoms, Superconductors
APPLICATION AREAS	Quantum Data Center Hubs, heterogeneous networking R&D	Enterprise Grade Apps High-Frequency Trading (HFT), DI-QKD, VLBI.	Full Quantum Data Center Blind Quantum Computing

Distributed Quantum Computing (DQC) – Definition

- DQC is a paradigm in which k ≥ 2 small/moderate-scale Quantum Processing Units (QPUs), interconnected by a (classical + quantum) network, cooperatively execute a quantum circuit C whose width or depth exceeds the resources of any single QPU
- DQC involves non-local gates (i.e., operations between qubits belonging to different QPUs)
- Why can't we copy $|q_2\rangle$ to QPU1?

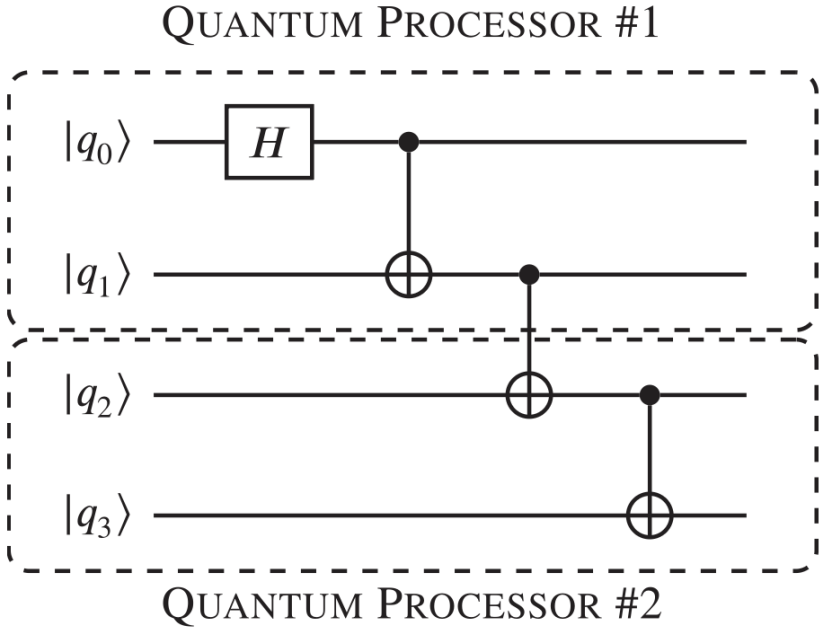
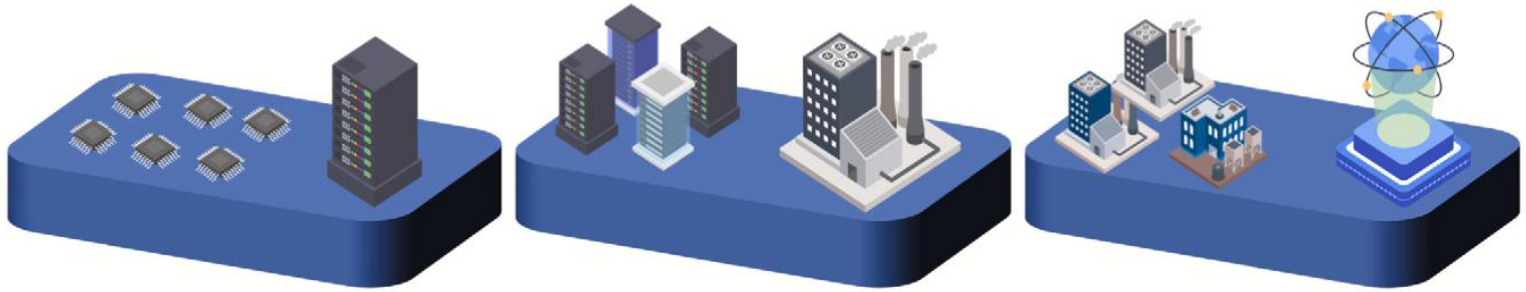


Fig. 4. Toy model for distributed quantum computation. The quantum circuit is composed by three two-qubit gates, i.e, CNOTs. First and last gates operate *locally*, namely, between qubits stored within the same QPU, whereas the intermediate gate operates *remotely*, namely, between qubits stored within different QPUs.

Archetypes for distributed quantum computing

- Multi-core (*same chip / same fridge)*
  - Also called Quantum Networks-on-Chip (QNoC)
  - Mostly homogenous QPUs
  - Physical distance between remote qubits is very short (mm to cm)
  - Limited number of qubits
  - Low latency (~10X ns)
- Multi-computer quantum architecture (*same room / same farm)*
  - Multiple quantum computers interconnected via some sort of Quantum LAN (QLAN)
  - Can be heterogenous
  - Longer distance (meters to 10X meters)
  - Mid latency (~ micros)
- Multi-farm quantum architecture
  - Multiple geographically-distributed quantum farms
  - Significantly heterogenous
  - Wide-scale connection (kms) – Quantum Internet
  - High latency (millis)

Archetypes for distributed quantum computing



	MULTI-CORE	MULTI-COMPUTER	MULTI-FARM
SCALE	SINGLE QUANTUM COMPUTER	QUANTUM FARM	GLOBAL
INTERCONNECTING	MULTIPLE QPU	MULTIPLE QUANTUM COMPUTERS	MULTIPLE QUANTUM FARMS
HARDWARE HETEROGENEITY	ABSENT OR VERY MODERATE	POSSIBLE	VERY LIKELY

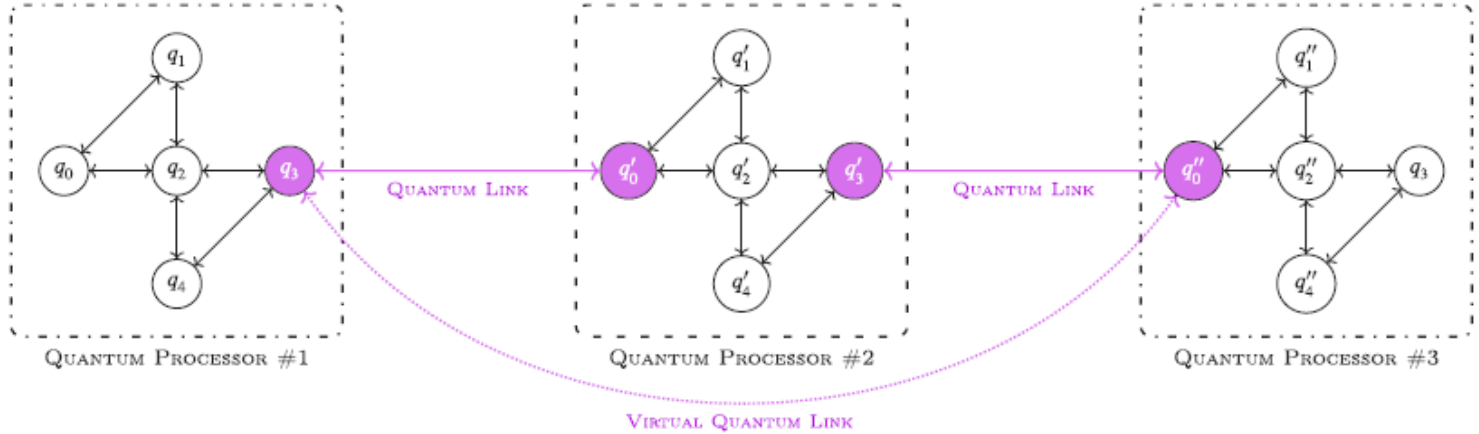
Fig. 5. Archetypes for Distributed Quantum Computing, with three different dimensions – i.e., scale, interconnection and heterogeneity – highlighted for the sake of comparison.

Distributed Quantum Computing — Jargon

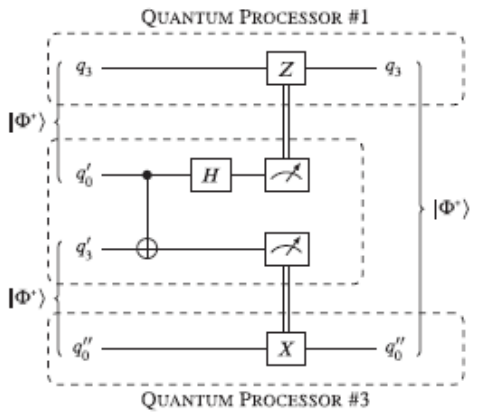
- Network primitives
  - Bell pairs (EPR) distribution
  - Entanglement swapping
  - Entanglement distillation
  - Classical 2-bit signalling
- Computation primitives
  - Local gates on each QPU
  - Non-local gates via TeleGate
  - Qubit movement via TeleData
  - Reversible message passing
- Resource accounting
  - Data qubits (computation)
  - ebits as a first-class resource

Entanglement: EPR pairs, swapping, distillation

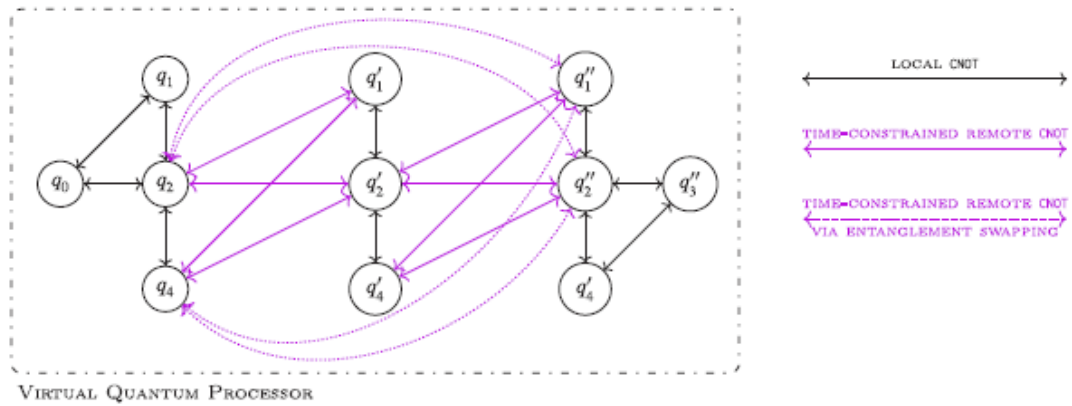
- Bell pairs (ebits)
  - $|\Phi^+\rangle = \frac{1}{2}(|00\rangle + |11\rangle)$ the unit of quantum communication
  - Can be generated by photonic emitters, atom–photon interfaces, or microwave-to-optical transducers
  - Each non-local gate consumes ≥ 1 ebit
  - ebit rate (ebit/second) is used as QoS of a quantum link
- Entanglement swapping
  - If A↔R and R↔B share Bell pairs, A↔B can be directly entangled be performing a Bell-state measurement (BSM) on R and broadcasts 2 cbits
  - Used to build long-range ebits from short-range ones (quantum repeaters)
- Entanglement distillation
  - Multiple noisy ebits → one high-fidelity ebit via local gates + measurement
  - Chain swap: A ↔ R_1 ↔ R_2 ↔ B → direct A ↔ B Bell pair (BSM at R_1, R_2 → 2 cbits each).



(a) By swapping the entanglement at an intermediate node – namely, quantum processor #2 – it is possible to distribute a Bell state between remote processors – namely, processors #1 and #3 – even if they are not directly connected through a quantum link. Hence, entanglement swapping enhances the quantum processors connectivity through *virtual quantum links*.



(b) Entanglement swapping circuit. A Bell state can be distributed between remote processors by swapping the entanglement at an intermediate node – as instance, processors #2 in Figure 8a – through local processing and classical communication.

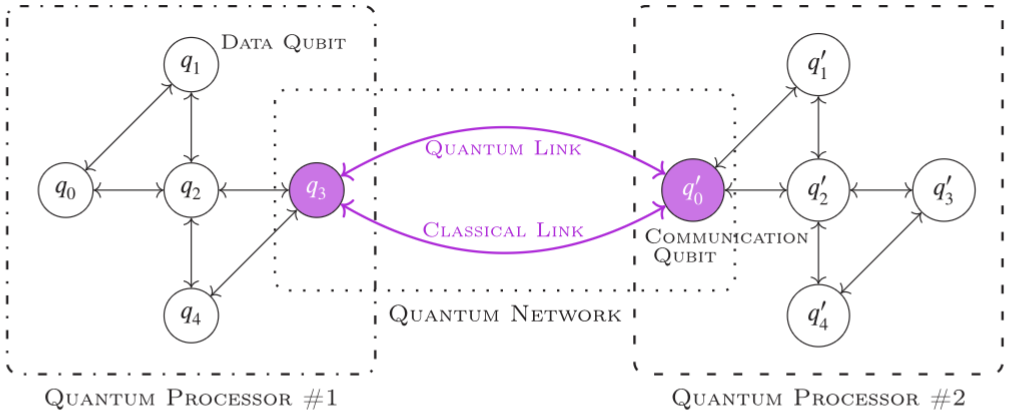


(c) Dynamic coupling map for the network topology shown in Figure 8a. The solid purple lines denote remote CNOTs between adjacent processors, whereas the dotted purple lines denote remote CNOTs between distant processors achievable via entanglement swapping.

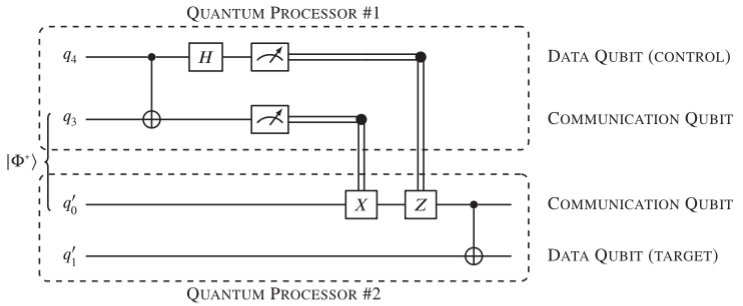
Fig. 8. Augmented connectivity. Entanglement swapping increases the connectivity between physical qubits, with a number of possible remote CNOTs that scales at least linearly with the number of processors.
Source: Figure reproduced from [13].

Quantum Networking

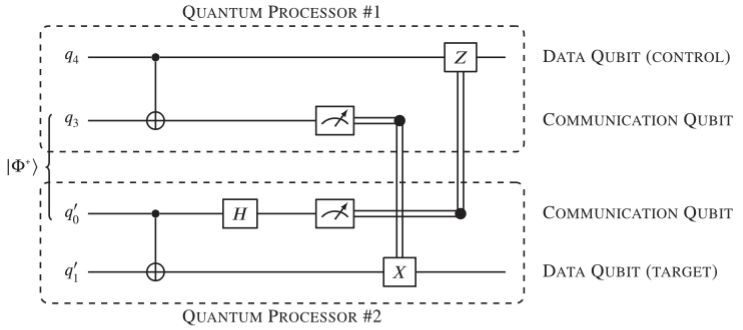
- DQC is enabled through quantum communication primitives
  - Directo qubit tranmission: entangle a qubit to a photon and transmit over fiber link
  - TeleData: quantum teleportation of an UNKNOWN data qubit $|\psi\rangle$ between QPUs, consuming one shared EPR pair $|\Phi^+\rangle$ and 2 classical bits
  - TeleGate: applies a non-local two-qubit gate (typically CNOT) between data qubits sitting on different QPUs, without moving either of them



(a) Two quantum processors given in Figure 3 interconnected through a quantum network, composed by a classical and a quantum link. The classical link is used to transmit classical information, whereas the quantum link is needed for distributing entangled states between the two remote processors to enable communication functionalities. Indeed, at least one physical qubit at each processor must be reserved for entanglement generation. This kind of qubits – purple-colored in the figure – are the *communication qubits* to distinguish them from the *data qubits* – white-colored in the figure.



(b) `TeleData`. To perform a `TeleData` between remote processors – say to move the quantum state $|\varphi\rangle$ stored by data qubit q_4 in Figure 7a to communication qubit q_0' – a Bell state such as $|\Phi^+\rangle$ must be distributed through the quantum link so that each pair member is stored within the *communication qubit* at each processor. Once $|\varphi\rangle$ is teleported at q_0' (with local quantum operations and classical transmission), the remote operation – for instance, a `CNOT` with $|\varphi\rangle$ as control and the state stored by qubit q_1' as target as represented by the last `CNOT` in the figure – can be executed through local operations.



(c) `TeleGate`. A `TeleGate` enables a direct gate between remote physical qubits stored at different processors without the need of quantum state teleportation, as long as a Bell state such as $|\Phi^+\rangle$ is distributed through the quantum link. As instance, a remote `CNOT` between q_4 and q_1' in Figure 7a can be implemented with two local `CNOT`s between the *data* and the *communication qubit* at each processor, followed by a conditional gate on the data qubit depending on the measurement of the remote communication qubit.

Fig. 7. Remote quantum operations through either `TeleData` or `TeleGate`. Fig. 7(a) shows the network topology along with the processors coupling maps, whereas Figs. 7(b) and 7(c) illustrate the quantum circuit detailing the classical (2 bits) and the quantum (the Bell state) resources needed to execute a `TeleData` and a `TeleGate`, respectively. *Source:* Figure reproduced from [13].

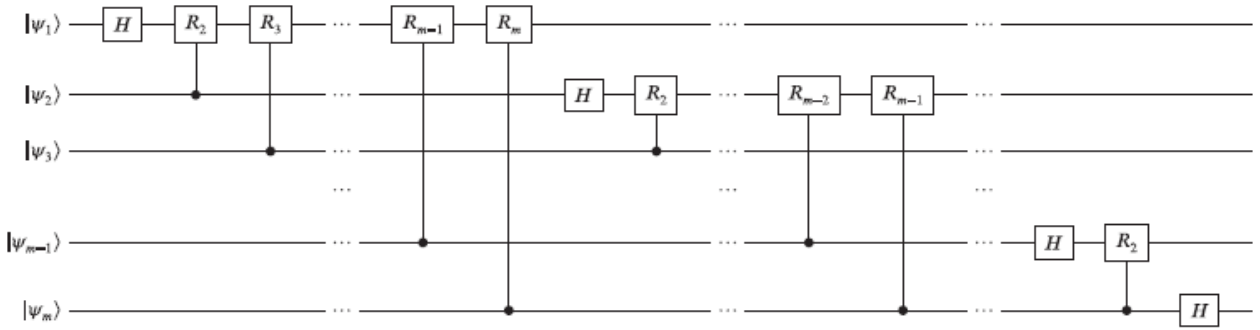
Hardware layer – Inter-QPU Communication

- All-microwave (cryo coax)
  - QPUs share a single dilution refrigerator OR are wired with cryogenic coaxial
- Microwave–optical transducer
  - Convert SC microwave qubits → telecom photons → fibre → re-convert
  - Conversion efficiency today ~5–30 % (DQC bottleneck)
- Native photonic / matter-flying
  - Trapped ions, NV-centres, neutral atoms emit photons at telecom or visible band directly
- Fault-tolerance is the gating constraint, not the link itself
- Physical 2-qubit error today ≈ 10^{-3}, while algorithms need ≈ 10^{-10} logical → ~10^{3}–10^{4} physical per logical (surface code)
- Hardware engineers optimise EBIT-PER-SECOND and EBIT-FIDELITY
  - Compilers must treat both as first-class budgets

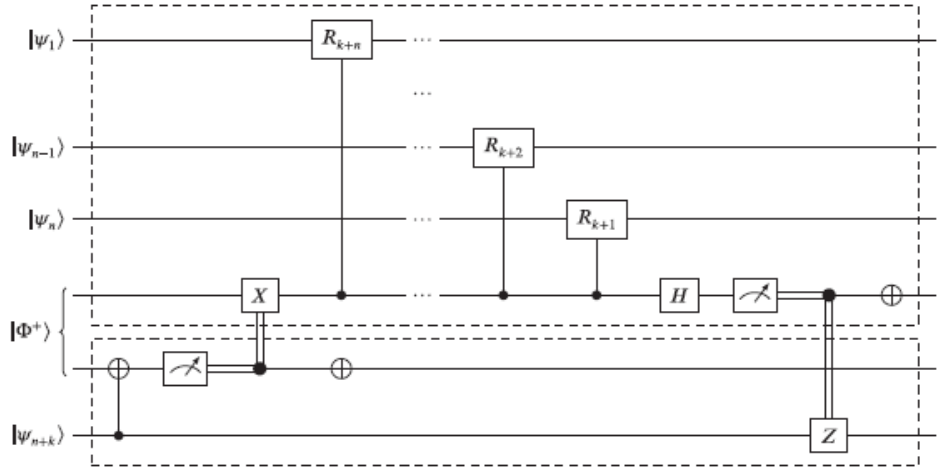
Data qubits vs communication qubits — the budget

- Each QPU partitions its physical qubits into TWO disjoint sets and the split is a hyperparameter of the architecture
- Data qubits (n_d)
  - Carry the algorithmic state
  - Drive the WIDTH of distributable circuits per node
  - Long coherence requirements (memory)
  - Bigger n_d → fewer SWAPs / partitioning overhead
- Communication qubits (n_c)
  - Hold one half of an EPR pair shared with a peer
  - Drive the THROUGHPUT of remote gates / teleportations
  - High-bandwidth photonic / matter-flying interface
  - bigger n_c → more parallel TeleGate/TeleData per layer
- First-order resource model
  - $n_{phys} = n_d + n_c$
  - $R_{remote} \leq \min(n_c, BW_{link} \cdot t_{layer})$
  - $T_{circuit} \approx T_{local} + (G_{remote} - R_{remote}) \cdot T_{ebit}$

Distributed Quantum Algorithm



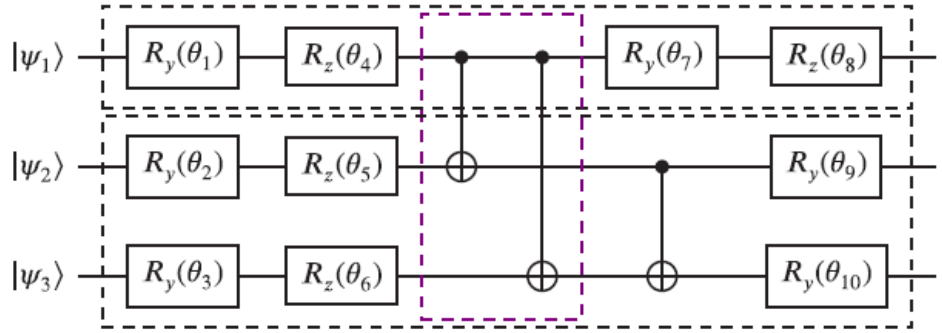
(a) Monolithic QFT circuit. The i-th qubit is obtained through an Hadamard gate followed by $m-i$ controlled R_m operations – with $R_i = P_{2\pi/i}$ denoting the phase gate mapping $|1\rangle \rightarrow e^{j2\pi/i}|1\rangle$ while leaving $|0\rangle$ unchanged – with the controlled operations controlled by the $m-i$ higher-order qubits.



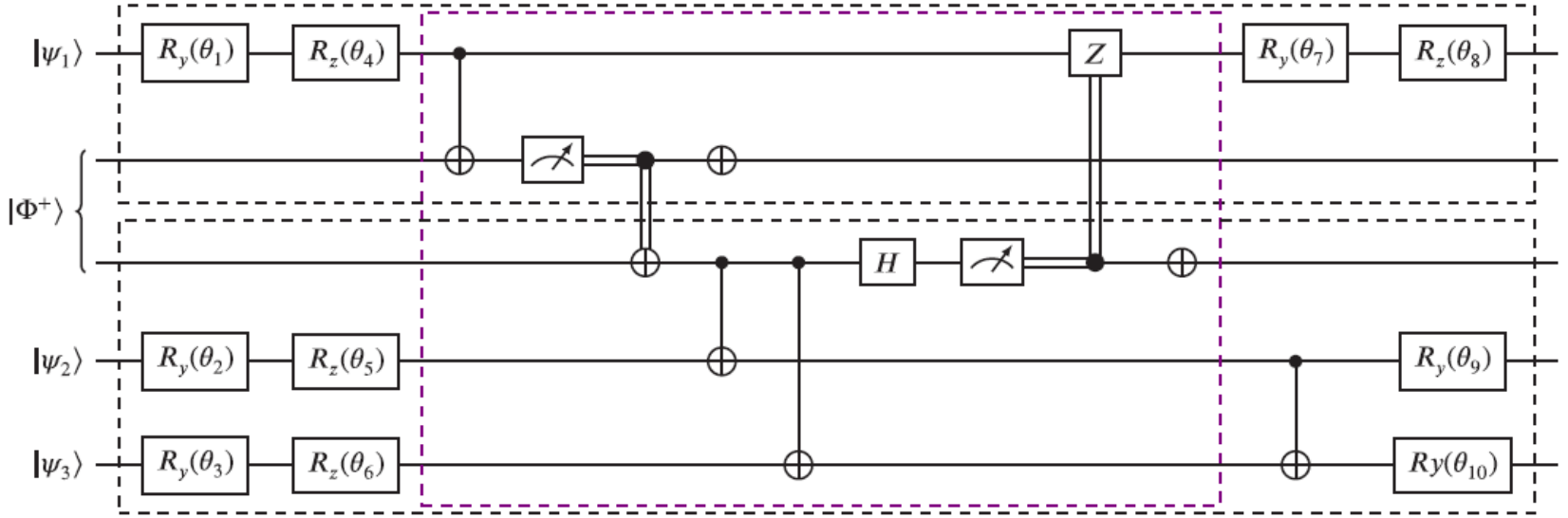
(b) Part of a distributed QFT as shown in [92]. Here, n non-local operations on n qubits are performed using a single shared entangled state. The control is $|\psi_{n+k}\rangle$, the targets are $|\psi_1\rangle, \ldots, |\psi_n\rangle$. With respect to Figure 9, a subset of the m qubits is considered (i.e., $n+k<m$).

Fig. 9. Quantum Fourier Transform (QFT) circuit compilation for DQC.

Distributed Quantum Algorithm



(a) 3-qubit variational form.



(b) 3-qubit variational form distributed across two devices.

Fig. 10. Variational Quantum Algorithm (VQA) circuit compilation for DQC.

DQC Compiler

- Compiling for DQC = compiling for monolithic QPU + an additional partitioning + remote-gate scheduling stage.
- 1. Qubit assignment (partitioning)
  - Map n logical qubits → k QPU partitions to minimize inter-partition gates
  - Models: hypergraph k-cut, weighted graph K-L
  - Static (decide once) vs Dynamic (re-partition on a moving window via TeleData)
  - NP-hard in general
- 2. Non-local gate handling
  - For each cut edge, decide TeleGate vs TeleData and synthesise the implementation
  - TeleGate uses cat-entangler / cat-disentangler
  - Optimization objective is to minimize # ebits and depth
- 3. Local circuit compilation + scheduling
  - Per-QPU: gate decomposition into native set, qubit routing (SWAP insertion), noise-aware mapping
  - Cross-QPU: schedule remote operations in time, respect ebit-generation rate, hide latency behind local clock
  - Co-design: treat ebit availability as a constraint in the local scheduler, not a post-hoc patch

DQC Challenges

- Decoherence vs latency
  - Coherence time ~100 μs (SC) or ~1 s (ion). inter-farm RTT can be ms
- NP-hard partitioning
  - Optimal qubit assignment is NP-complete even on a single chip
  - In DQC the search space grows combinatorially with k QPUs and a time-varying coupling map
- Noise-aware DQC compilation
  - single-QPU compilers are noise-aware
  - Currently, no production compiler is end-to-end noise-aware across multiple QPUs.
- Distillation vs throughput
  - higher-fidelity ebits need more rounds → more communication qubits → less compute budget
- Multi-tenant scheduling
  - quantum cloud must arbitrate concurrent jobs across multiple QPUs and a shared entanglement fabric
- Verification & debugging
  - cannot copy or peek at quantum state; debugging a distributed QC program is itself a research problem

DQC Opportunities

- Exponential power scaling
  - linear in interconnected QPUs $\Longrightarrow$ exponential $\uparrow$ in joint Hilbert-space dimension
- Quantum-centric supercomputing
  - tight coupling of CPU + GPU + QPU + EPR fabric
  - QPUs become a HPC accelerator within familiar SLURM-style schedulers
- Multi-vendor heterogeneity
  - Match each subroutine to its best-fit hardware: ions for high-fidelity logic, SC for fast Clifford, photonic for memory.
- New algorithm classes
  - Measurement-based one-way QC and multipartite-entanglement protocols become natural in a networked setting (proactive ebit distribution).
- Geographic + secure compute
  - DQC + QKD = secure delegated quantum computation across organisations
- Modular fault-tolerance
  - Logical qubits stitched across modules via lattice surgery → break the single-die qubit ceiling without re-engineering each generation

Take-aways and references

- Five lecture take-aways
  1. Monolithic QPUs hit physical, thermal, and yield walls long before the qubit counts that practical algorithms need
  2. DQC scales out: theoretically equivalent to monolithic QC up to polylog depth, but pays a real cost in ebits and classical signalling
  3. Three architectural archetypes (multi-core / multi-computer / multi-farm) trace a natural deployment path
  4. TeleData + TeleGate plus an ebit-aware scheduler are the minimal compiler primitives
  5. The bottleneck is no longer 'enough qubits' but 'enough high-fidelity ‘ebits per second’
- Reading list
  - Caleffi et al. (2024) — Distributed quantum computing: A survey. Computer Networks 254, 110672.
  - Ferrari et al. (2021) — Compiler design for distributed quantum computing. IEEE Trans. Quantum Eng. 2.
  - Häner, Steiger, Hoefler, Troyer (SC '21) — Distributed quantum computing with QMPI. ACM/IEEE SC '21.
  - Beals et al. (2013) — Efficient distributed quantum computing. Proc. R. Soc. A 469.
  - Vendor roadmaps (2026) — ibm.com/roadmaps/quantum · quantinuum.com/blog · quantumai.google/roadmap · ionq.com/news · psiquantum.com