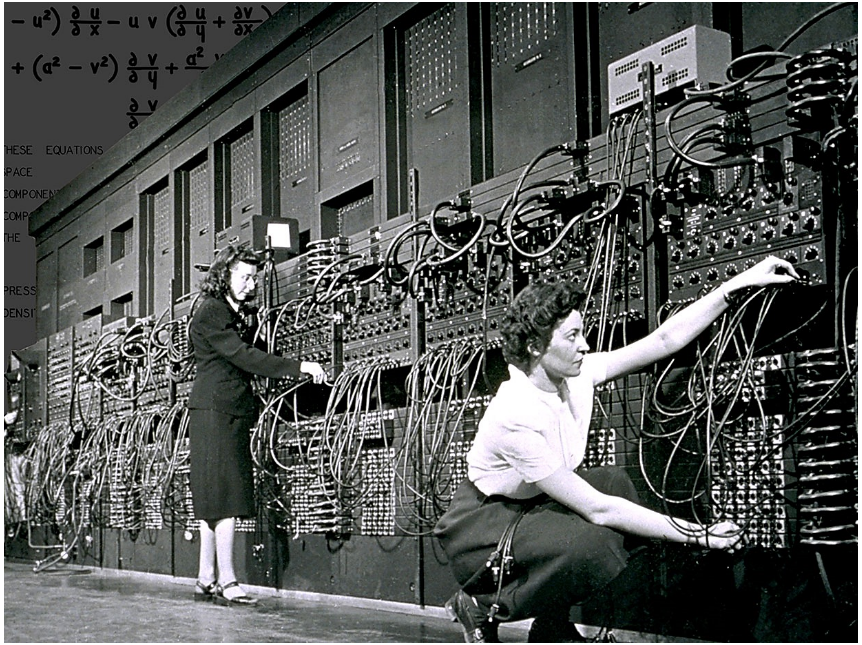
VTT

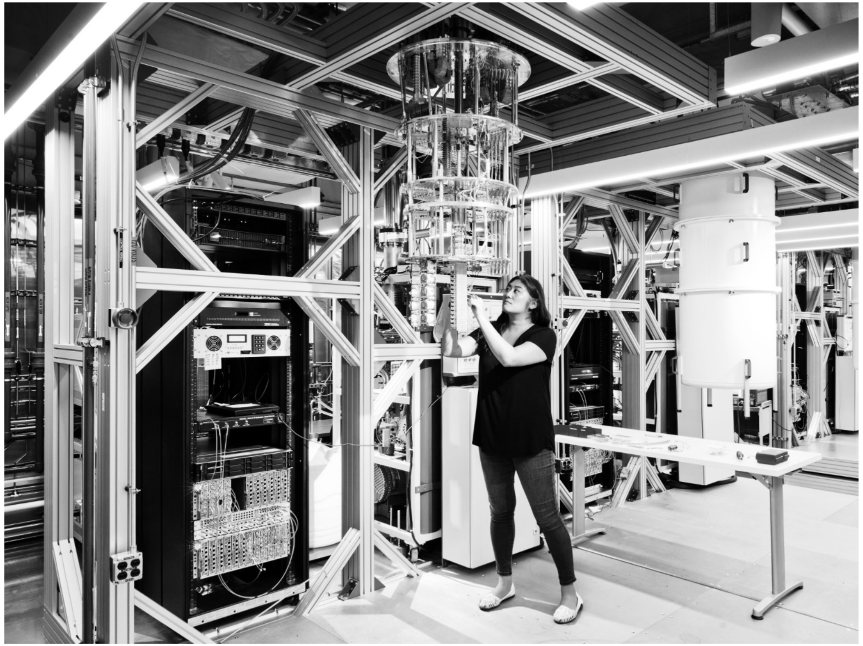
Programming quantum computers

Daniel Reitzner
Quantum algorithms and software / VTT

07/07/2022 VTT – beyond the obvious

Where we stand with current QPUs



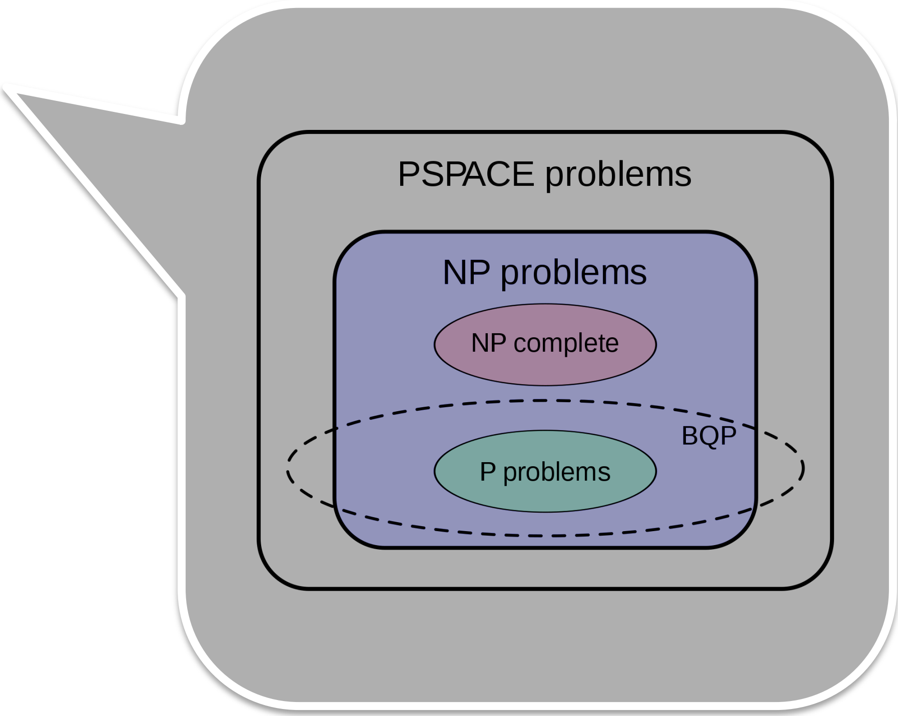


NISQ era = Noisy Intermediate-Scale Quantum

High-level programming workflow

Programing quantum computers workflow

1. Identify problem to be solved
2. Choose suitable platform and vendor
3. Prepare quantum program
4. Transpile
5. Submit



Programing quantum computers workflow

1. Identify problem to be solved
2. Choose suitable platform and vendor
3. Prepare quantum program
4. Transpile
5. Submit

- Quantum simulations (chemistry, material science, etc.) are suitable for adiabatic computers
- Quantum algorithmic tasks work well on gate-based platforms (superconductive, ion-based, optical, etc.)
- Optimization problems might benefit from various platforms

- Vendors may vary in quality in price and might offer also free computational time

Programing quantum computers workflow

1. Identify problem to be solved
2. Choose suitable platform and vendor
3. Prepare quantum program
4. Transpile
5. Submit

- Different vendors have their own libraries:
  - IBM: Qiskit
  - Google: Cirq
  - Xanadu: Strawberry Fields
  - Etc.
- Usually Python-based
- Many times offer wide variety of approaches to quantum computing
  - Simple gate-based
  - Hybrid (VQE, QAOA, etc.)
  - Supplemented by AI
  - Etc.

Programing quantum computers workflow

1. Identify problem to be solved
2. Choose suitable platform and vendor
3. Prepare quantum r
4. Transpile
5. Submit

- Important optimization task for efficient computation
- Many times hands-on
- Roughly composed of three main tasks:
  - routing: fitting the circuit to chosen device
  - decomposition: replacing gates with the native set
  - optimization: reducing the complexity of fitted circuit
- if we will want to get the most out of the QPU, we have to consider QPU's architecture, quality and (potentially) speed.

Programing quantum computers workflow

1. Identify problem to be solved
2. Choose suitable platform and vendor
3. Prepare quantum program
4. Transpile
5. Submit

- After transpilation quantum compilation transforms submission to some lower level language (OpenQASM2, etc.) and then to basic instruction set of QPU electronics
- These might have various bottlenecks
- Demand for
  - Quality
  - Fast access
  - Precompilation
  - Classical control
  - Intermediate measurements
  - Etc.

VTT

Programing quantum computers workflow

1. Identify problem to be solved
2. Choose suitable platform and vendor
3. Prepare quantum program
4. Transpile
5. Submit

Quantum teleportation → gate-based (superconductive) → IBMQ

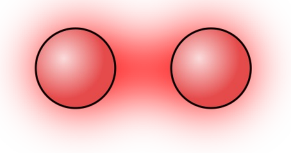
Let's have a look at Jupyter notebook

Quantum teleportation

Quantum teleportation

- With shared entanglement and 2 classical bits of information we can “teleport” a quantum state

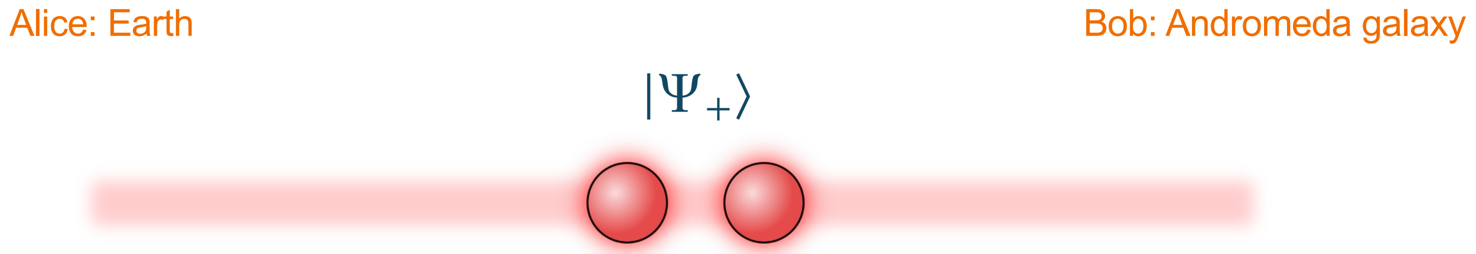
$$|\Psi_+\rangle$$



C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, W. K. Wootters – Teleporting an Unknown Quantum State via Dual Classical and Einstein–Podolsky–Rosen Channels, Phys. Rev. Lett. 70, 1895–1899 (1993)

Quantum teleportation

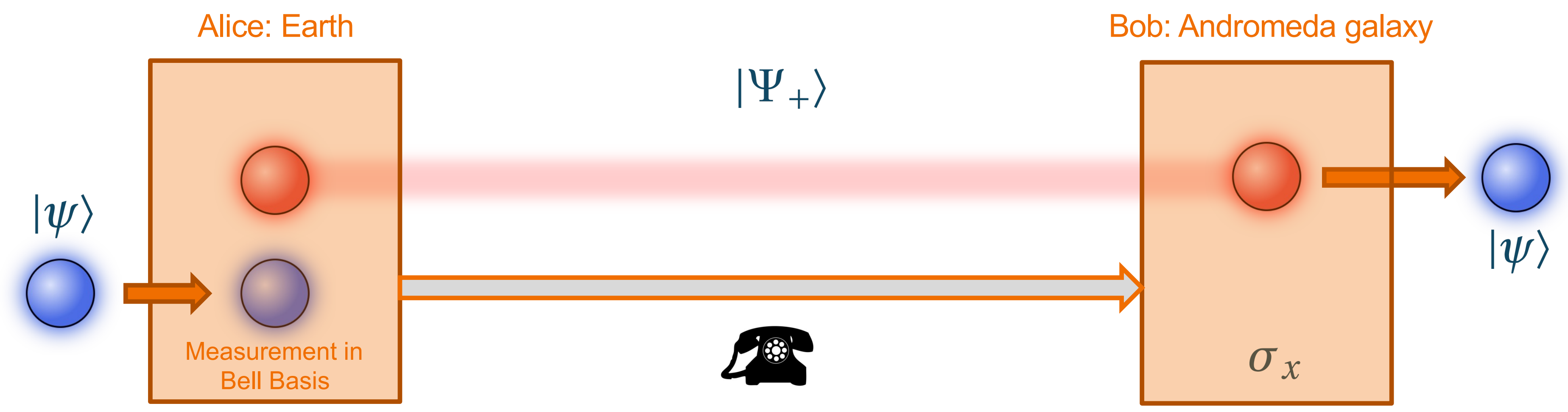
- With shared entanglement and 2 classical bits of information we can “teleport” a quantum state



C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, W. K. Wootters – Teleporting an Unknown Quantum State via Dual Classical and Einstein–Podolsky–Rosen Channels, Phys. Rev. Lett. 70, 1895–1899 (1993)

Quantum teleportation

- With shared entanglement and 2 classical bits of information we can “teleport” a quantum state



C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, W. K. Wootters – Teleporting an Unknown Quantum State via Dual Classical and Einstein–Podolsky–Rosen Channels, Phys. Rev. Lett. 70, 1895–1899 (1993)

Quantum teleportation

- How does it work? Composite state before Alice measures it is $|\Phi\rangle = |\psi\rangle \otimes |\Psi_+\rangle$
- If $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ then $|\Phi\rangle = \frac{1}{\sqrt{2}}\left(\alpha|000\rangle + \beta|100\rangle + \alpha|011\rangle + \beta|111\rangle\right)$
- We measure in the Bell basis and use:

$$|00\rangle = \frac{1}{\sqrt{2}}(|\Psi_+\rangle + |\Psi_z\rangle) \qquad |01\rangle = \frac{1}{\sqrt{2}}(|\Psi_x\rangle + \mathrm{i}|\Psi_y\rangle)$$

$$|11\rangle = \frac{1}{\sqrt{2}}(|\Psi_+\rangle - |\Psi_z\rangle) \qquad |10\rangle = \frac{1}{\sqrt{2}}(|\Psi_x\rangle - \mathrm{i}|\Psi_y\rangle)$$

- Now

$$|\Phi\rangle = \frac{1}{2}\left[|\Psi_+\rangle \otimes |\psi\rangle + |\Psi_x\rangle \otimes \sigma_x|\psi\rangle + |\Psi_y\rangle \otimes \sigma_y|\psi\rangle + |\Psi_z\rangle \otimes \sigma_z|\psi\rangle\right]$$

Quantum teleportation

- Measuring the state

$$|\Phi\rangle = \frac{1}{2}\left[|\Psi_+\rangle \otimes |\psi\rangle + |\Psi_x\rangle \otimes \sigma_x|\psi\rangle + |\Psi_y\rangle \otimes \sigma_y|\psi\rangle + |\Psi_z\rangle \otimes \sigma_z|\psi\rangle\right]$$

in Bell basis gives each result with the same probability ¼ and gives two bits of classical information about the transformation Bob needs to make to transform his state $|\phi\rangle$ to $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$|\Psi_+\rangle \to \mathbb{1}|\phi\rangle = |\psi\rangle \qquad |\Psi_x\rangle \to \sigma_x|\phi\rangle = |\psi\rangle$$

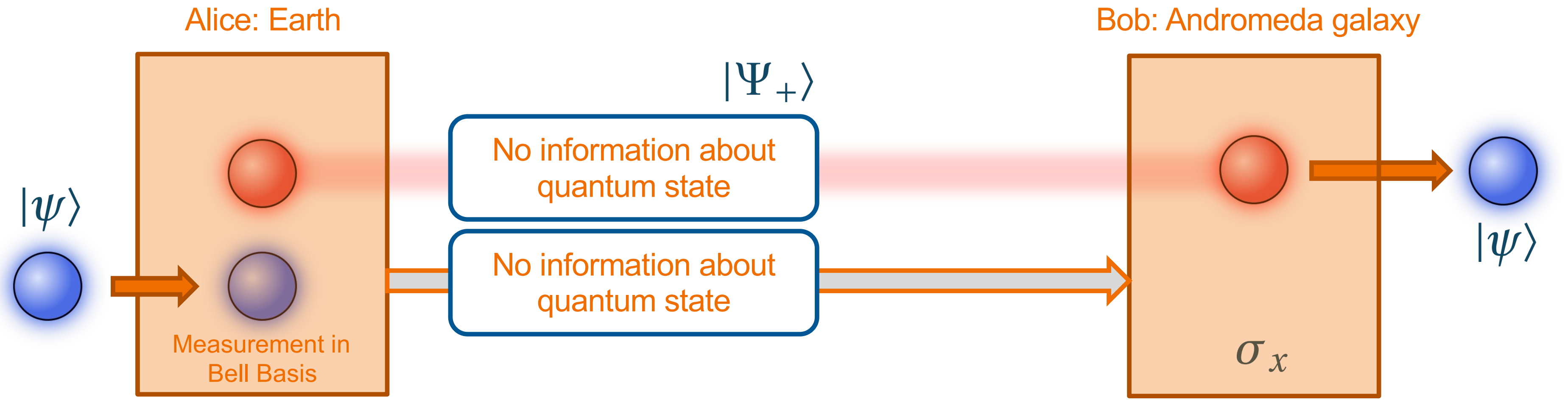
$$|\Psi_y\rangle \to \sigma_y|\phi\rangle = |\psi\rangle \qquad |\Psi_z\rangle \to \sigma_z|\phi\rangle = |\psi\rangle$$

- This holds because

$$\mathbb{1}^2 = \sigma_x^2 = \sigma_y^2 = \sigma_z^2 = \mathbb{1}$$

Quantum teleportation

- With shared entanglement and 2 classical bits of information we can “teleport” a quantum state

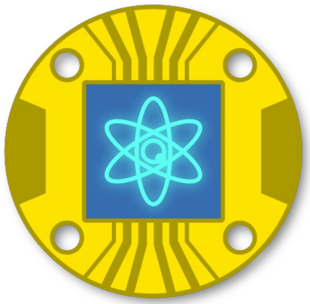


C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, W. K. Wootters – Teleporting an Unknown Quantum State via Dual Classical and Einstein–Podolsky–Rosen Channels, Phys. Rev. Lett. 70, 1895–1899 (1993)

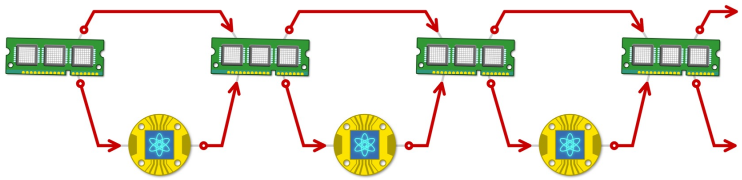
Beyond transpilation

Transpilation

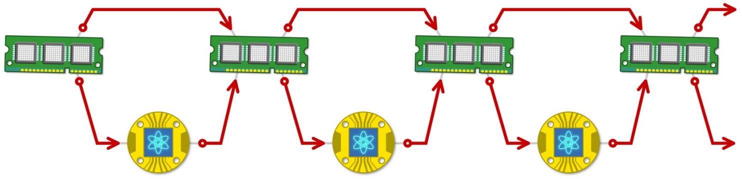
- routing: fitting the circuit to chosen device
- decomposition: replacing gates with the native set
- optimization: reducing the complexity of fitted circuit



- Current devices are noisy and small
- Useful for smaller tasks → hybrid computation
- New bottlenecks



Hybrid computations



- Current devices are noisy and small
- Useful for smaller tasks → hybrid computation
- Old bottlenecks:
  - Size – number of qubits
  - Quality (decrease noise, error correction, etc.)
- New bottlenecks:
  - Initialization speed (loading the data to the electronics)
  - Parametric circuits and pre-compilation (VQE, QAOA, etc.)
  - Fast communication buses (integration to HPC framework)
  - Active reset (decrease the measurement times)
- Practical demands: classical control, intermediate measurements, etc.

Holistic quality and speed measures:
- Quantum volume
- CLOPS
- Algorithmic qubits

Summary

- High level programming of QPUs can be simple
- But more time you put into preparations, more quality you can gain in computation (will it be enough?)
- Specifics of NISQ devices mean that alternative (hybrid) approaches are studied widely – these require new ways of quantifying the quality of the quantum resources

VTT

beyond
the obvious

Daniel Reitzner
daniel.reitzner@vtt.fi

@VTTFinland

www.vtt.fi