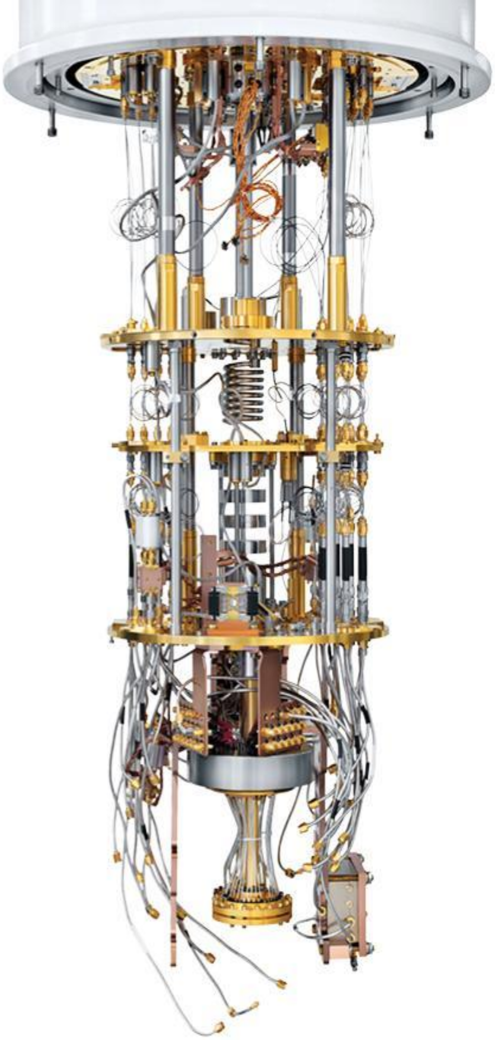
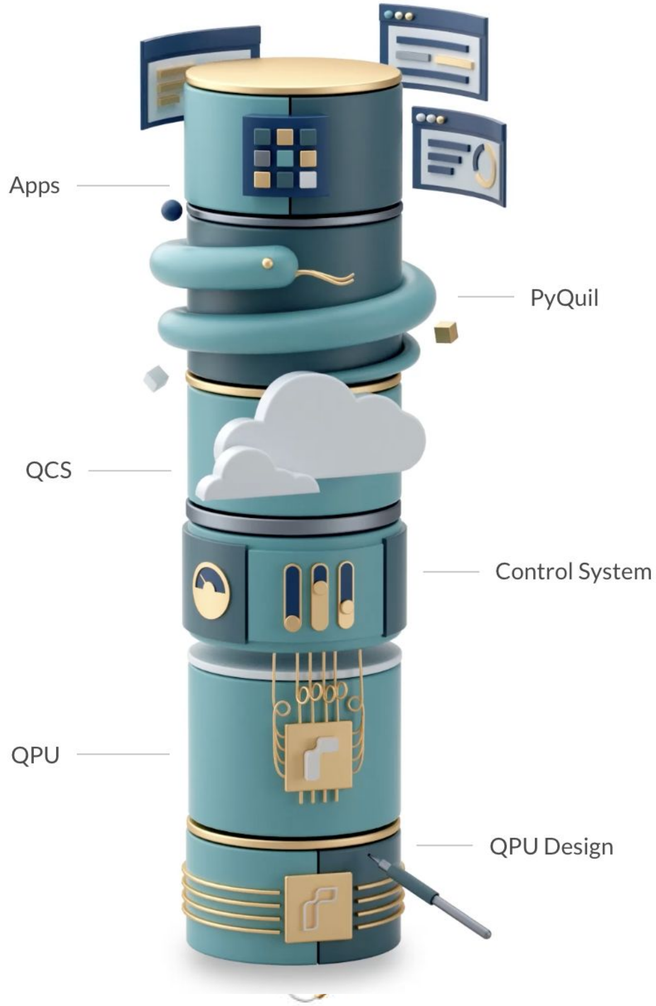


Quantum Computing and Programming Workshop

Rigetti Computing

02.25.2018

rigetti



rigetti

Full-stack quantum computing company.

8-qubit and **19-qubit** QPUs released on our cloud platform in 2017; **16-qubit Aspen-series** QPUs released fall 2018

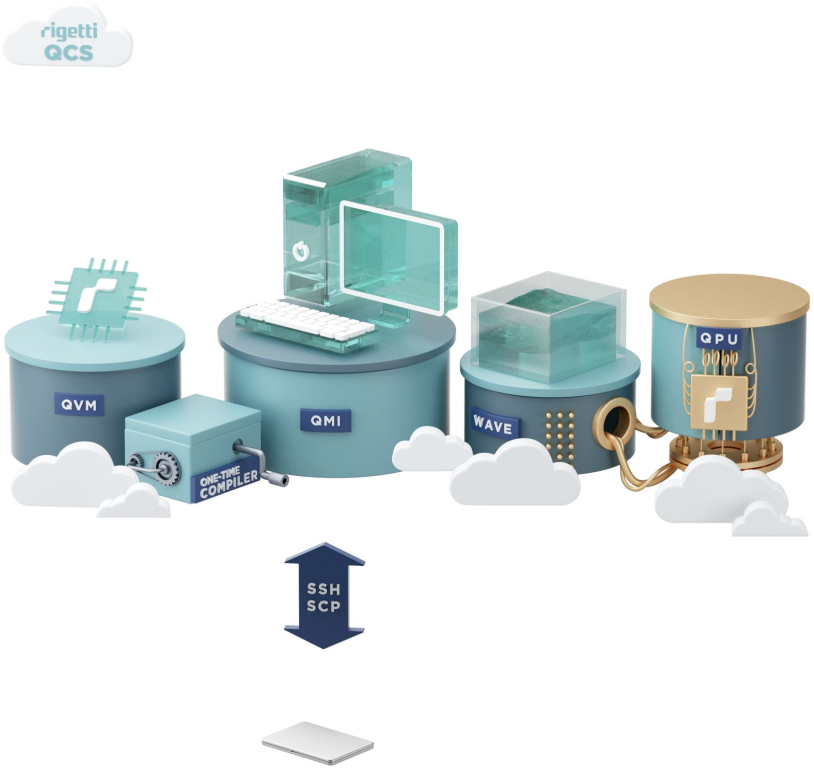
Quantum Cloud Services launched Fall 2018, with roadmap to **128-qubit systems**

100+ employees w/ $119M raised

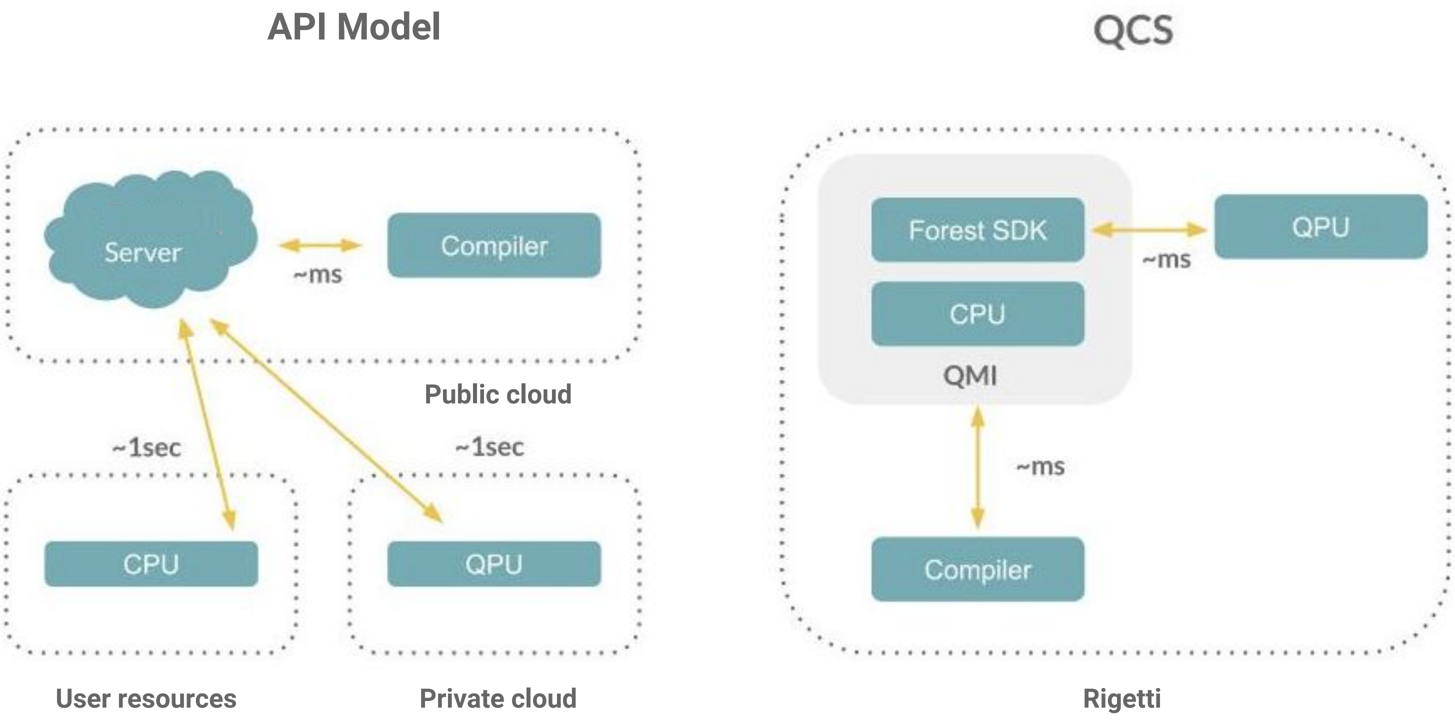
Home of Fab-1, the world's first commercial quantum integrated circuit fab

Located in Berkeley, Calif. (R&D Lab) and Fremont, Calif.

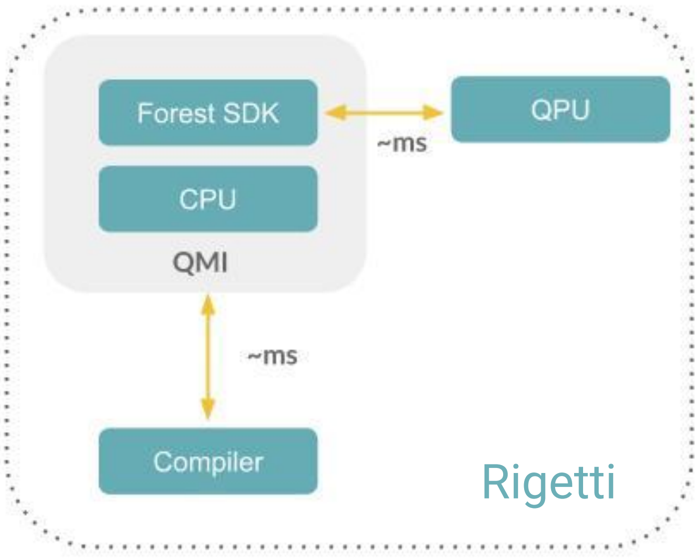
Quantum Cloud Services and the Rigetti Forest SDK



Rigetti Quantum Cloud Services



Rigetti QCS Addresses 3 Key Performance Bottlenecks



Colocation to reduce network latency between QPU and CPU.

Parameterized program compilation to reduce # of round trips to achieve solution.

Active reset of qubits in QPU to accelerate quantum runtime

new **Quantum Machine Image** access model

~30x faster than web API access models

Rigetti QCS Addresses 3 Key Performance Bottlenecks

Web API

~100

seconds

E.g. IBM Quantum Experience
Rigetti Forest 1.x

Colocation

to reduce network latency between QPU and CPU.

x 4.3

faster

+ Parametric Compilation

to reduce # of round trips to achieve solution.

x 13.5

faster

+ Active Reset

to accelerate quantum runtime

x 34.6

faster

Rigetti Forest SDK

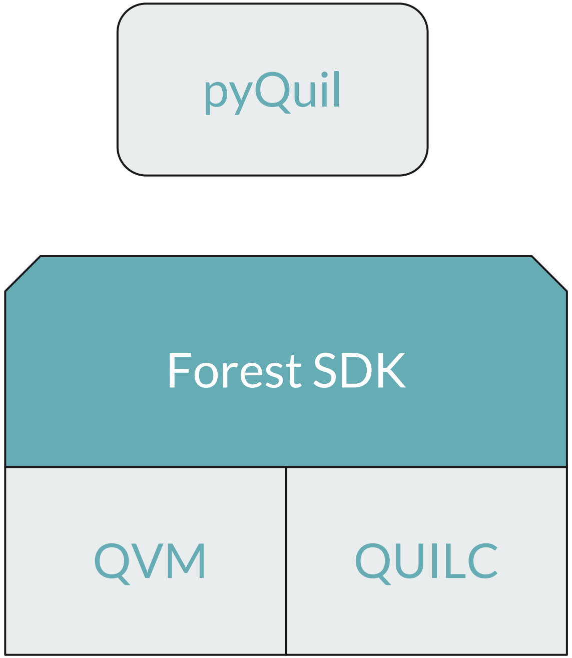
pyQuil

intuitive Python library for writing Quil programs to run on Forest SDK.

Forest SDK

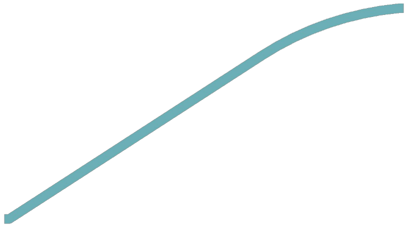
QVM: local simulator on **up to 26 qubits**

QUILC: compiler with the ability to **optimize programs to different architectures**



Why build a
quantum computer?

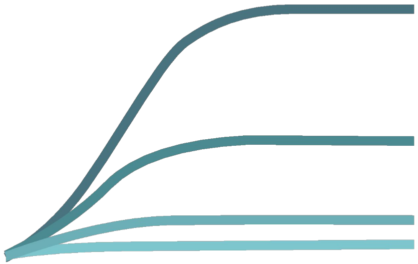
Classical computers have fundamental limits



Transistor scaling

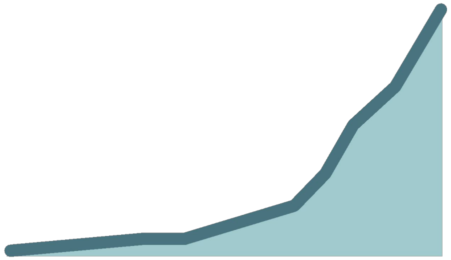
Economic limits with 10bn for next node fab

Ultimate single-atom limits



Returns to parallelization

Amdahl’s law



Energy consumption

Exascale computing project has its own power plant

Power density can melt chips

Quantum Advantage

Using quantum computers to solve problems **faster**, **cheaper** or **better** than otherwise possible.

1992

Quantum Algorithms' Progression

1992-4
First Quantum Algorithms w/ Exponential Speedup
(Deutsch-Jozsa, Shor's Factoring, Discrete Log, ...)

1996
First Quantum Database Search Algorithm (Grover's)

2007
Quantum Linear Equation Solving (Harrow, Hassidim, Lloyd)

2008
Quantum Algorithms for SVM's & Principal Component Analysis

These algorithms require big, perfect quantum computers

> 10,000,000 qubits for Shor's algorithms
to factor a 2048 bit number

2013
Practical Quantum Chemistry Algorithms (VQE)

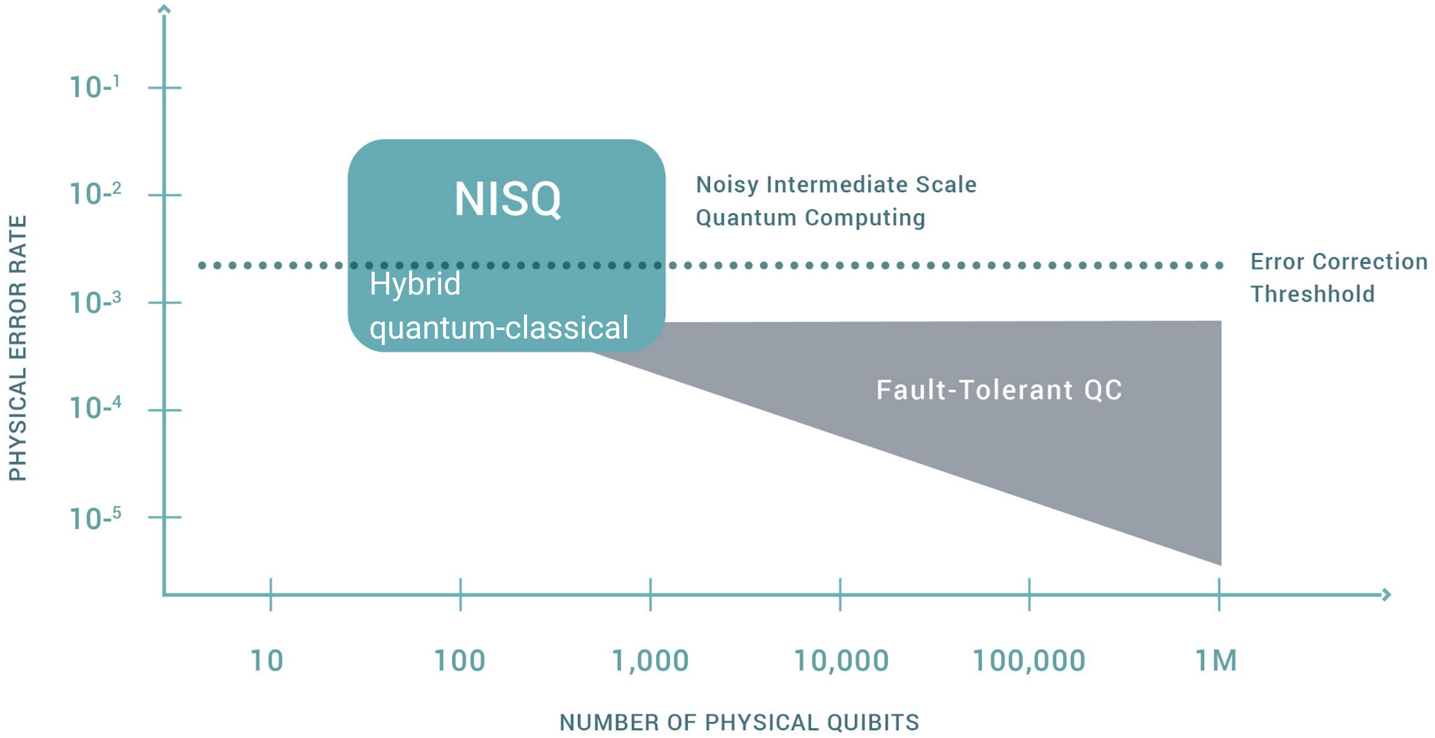
2016
Practical Quantum Optimization Algorithms (QAOA)
Simulations on Near-term Quantum Supremacy

Hybrid quantum/classical algorithms

noise-robust, empirical speedups

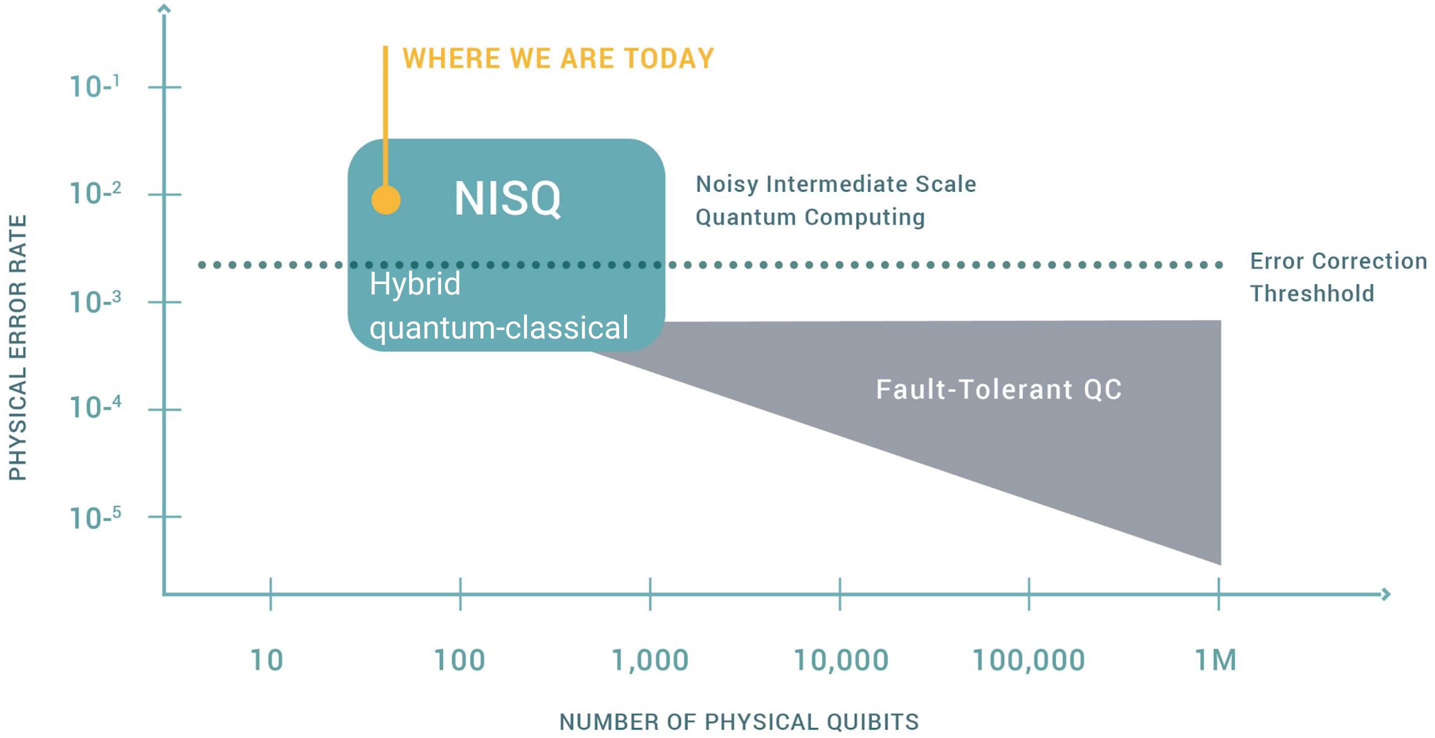
TODAY

Quantum Advantage will be achieved in the next 3-5 years



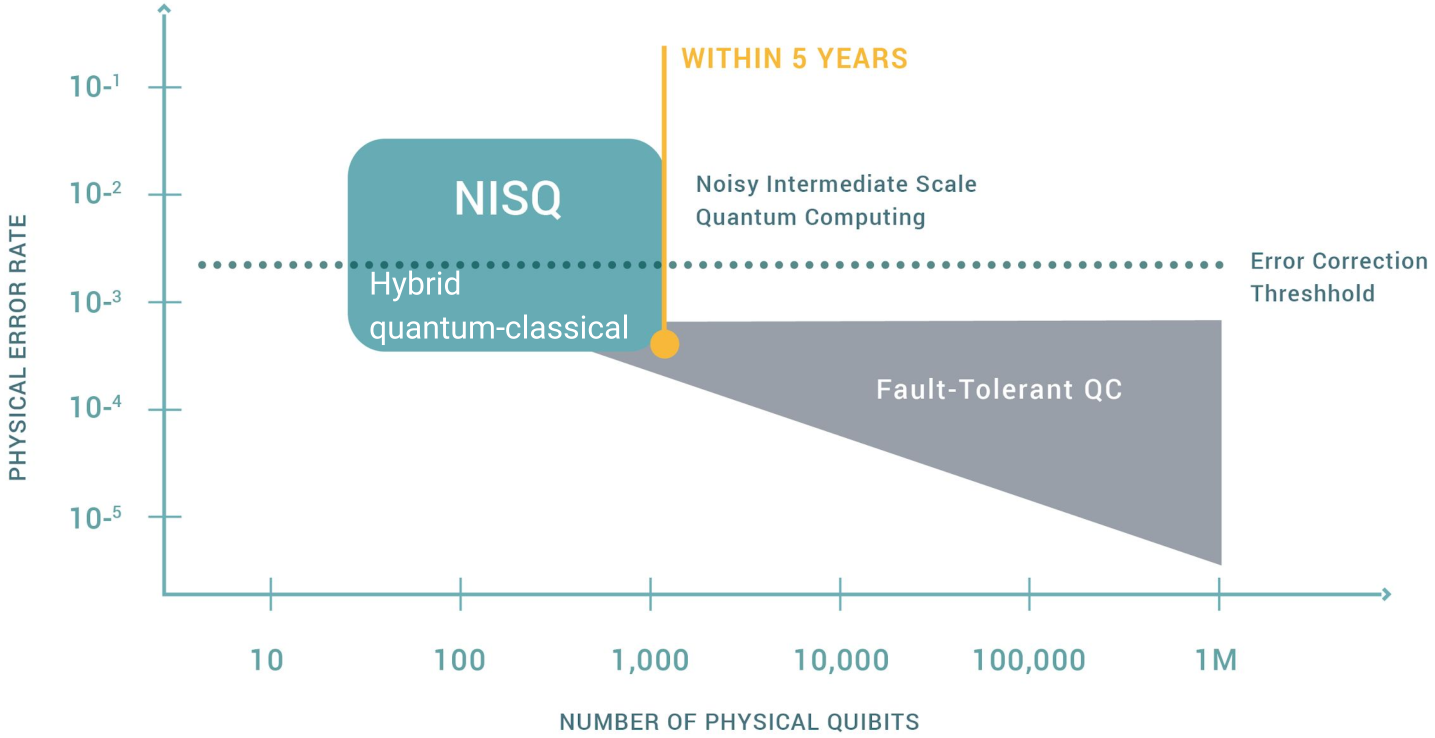
“Quantum computing in the NISQ era and beyond” Preskill, 2018
https://arxiv.org/abs/1801.00862

Quantum Advantage will be achieved in the next 3-5 years



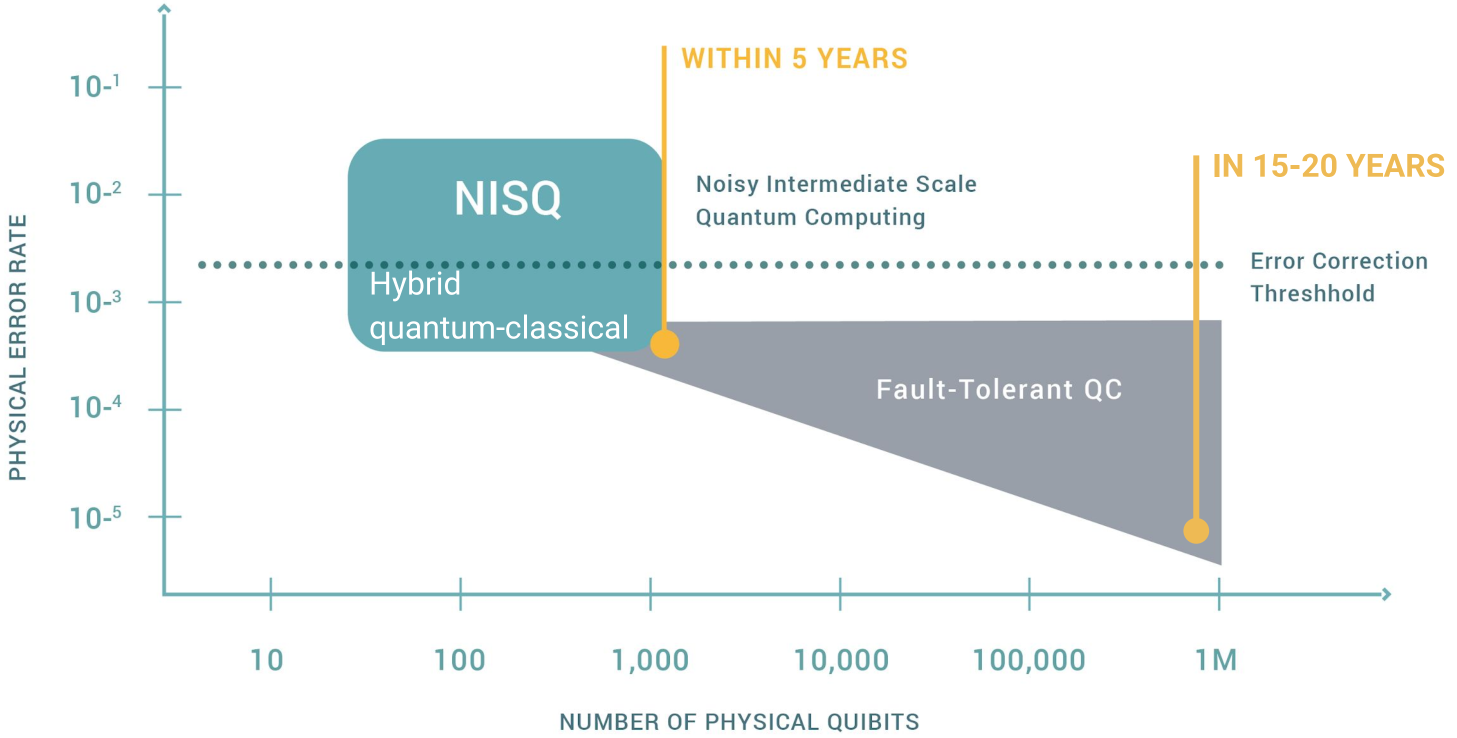
"Quantum computing in the NISQ era and beyond" Preskill, 2018
https://arxiv.org/abs/1801.00862

Quantum Advantage will be achieved in the next 3-5 years



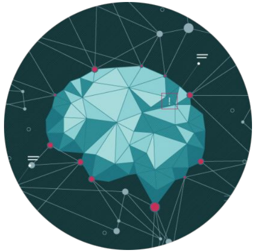
"Quantum computing in the NISQ era and beyond" Preskill, 2018
https://arxiv.org/abs/1801.00862

Quantum Advantage will be achieved in the next 3-5 years



"Quantum computing in the NISQ era and beyond" Preskill, 2018
https://arxiv.org/abs/1801.00862

Potential Applications for Hybrid Quantum-Classical Approach



Machine Learning

Development of new training sets and algorithms

Classification and sampling of large data sets

Supply Chain Optimization

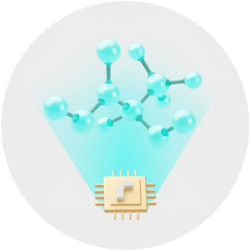
Forecast and optimize for future inventory demand

NP-hard scheduling and logistics map into quantum applications

Robotic Manufacturing

Reduce manufacturing time and cost

Maps to a Traveling Salesman Problem addressable by quantum constrained optimization



Computational Materials Science

Design of better catalysts for batteries

Quantum algorithms for calculating electronic structure

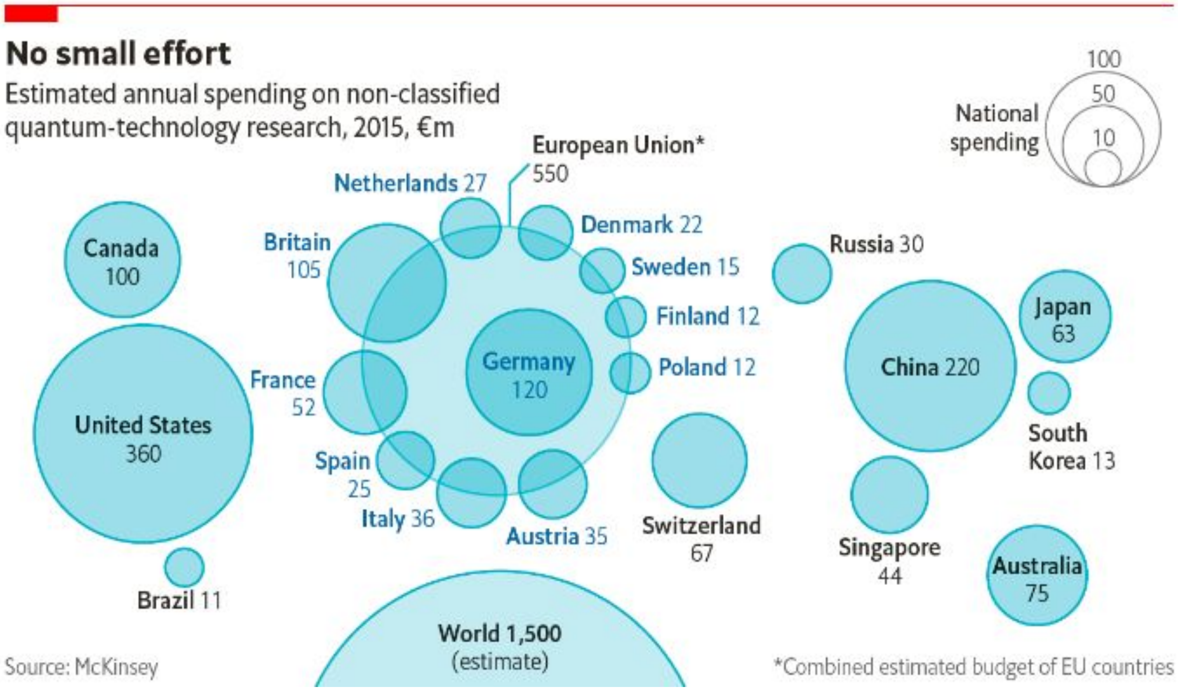
Alternative Energy Research

Efficiently convert atmospheric CO2 to methanol

Powered by existing hybrid quantum- classical algorithms + machine learning

Who is building and investing in quantum computers?

Investments across academia, government, and industry are global and growing



Bits vs.
Probabilistics Bits vs.
Qubits

Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$		

Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$	Real vector $\vec{v} = a\vec{0} + b\vec{1}$ $a, b \in \mathbb{R}_+$ $a + b = 1$	

Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$	Real vector $\vec{v} = a\vec{0} + b\vec{1}$ $a, b \in \mathbb{R}_+$ $a + b = 1$	

Probability of 0

Probability of 1

Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$	Real vector $a, b \in \mathbb{R}_+$ $\vec{v} = a\vec{0} + b\vec{1}$ $a + b = 1$	Complex vector $\alpha, \beta \in \mathbb{C}$ $\lvert\psi\rangle = \alpha\lvert 0\rangle + \beta\lvert 1\rangle$ $\lvert\alpha\rvert^2 + \lvert\beta\rvert^2 = 1$

Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$	Real vector $a, b \in \mathbb{R}_+$ $\vec{v} = a\vec{0} + b\vec{1}$ $a + b = 1$	Complex vector $\alpha, \beta \in \mathbb{C}$ $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$ $\|\alpha\|^2 + \|\beta\|^2 = 1$

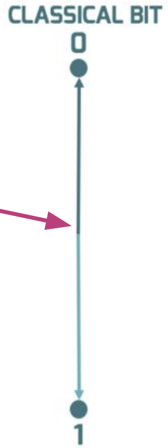
$|\alpha|^2$ = Probability of 0

$|\beta|^2$ = Probability of 1

Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$	Real vector $a, b \in \mathbb{R}_+$ $\vec{v} = a\vec{0} + b\vec{1}$ $a + b = 1$	Complex vector $\alpha, \beta \in \mathbb{C}$ $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$ $\|\alpha\|^2 + \|\beta\|^2 = 1$

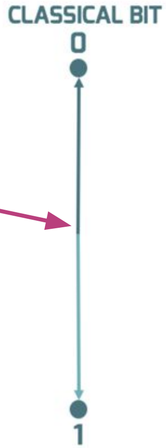
$$\vec{v}_{\text{coin}} = \frac{1}{2}\vec{0} + \frac{1}{2}\vec{1}$$



Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$	Real vector $a, b \in \mathbb{R}_+$ $\vec{v} = a\vec{0} + b\vec{1}$ $a + b = 1$	Complex vector $\alpha, \beta \in \mathbb{C}$ $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$ $\|\alpha\|^2 + \|\beta\|^2 = 1$

$$\vec{v}_{\text{coin}} = \frac{1}{2}\vec{0} + \frac{1}{2}\vec{1}$$



$$|\psi\rangle_{\text{coin}} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$|\psi\rangle_{\text{coin}} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

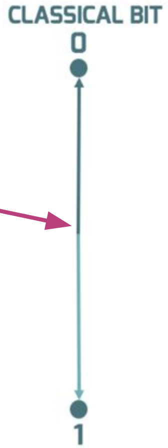
$$|\psi\rangle_{\text{coin}} = \frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle$$

...

Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$	Real vector $a, b \in \mathbb{R}_+$ $\vec{v} = a\vec{0} + b\vec{1}$ $a + b = 1$	Complex vector $\alpha, \beta \in \mathbb{C}$ $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$ $\|\alpha\|^2 + \|\beta\|^2 = 1$

$$\vec{v}_{\text{coin}} = \frac{1}{2}\vec{0} + \frac{1}{2}\vec{1}$$



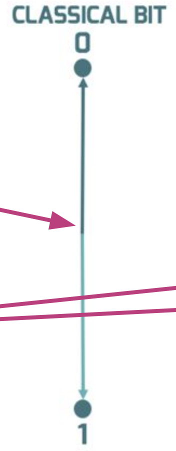
$$|\psi\rangle_{\text{coin}} = \frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\theta}}{\sqrt{2}}|1\rangle$$

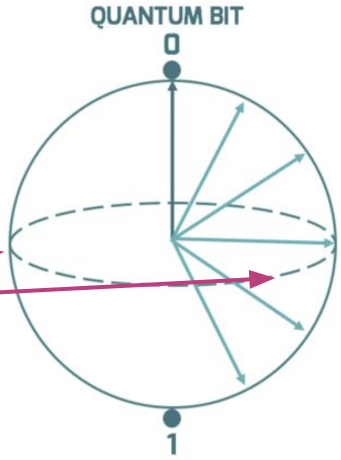
Bits vs. Probabilistic Bits vs. Qubits

	Bits	Probabilistic Bits	Qubits
State (single unit)	Bit $\in \{0, 1\}$	Real vector $a, b \in \mathbb{R}_+$ $\vec{v} = a\vec{0} + b\vec{1}$ $a + b = 1$	Complex vector $\alpha, \beta \in \mathbb{C}$ $\lvert\psi\rangle = \alpha\lvert 0\rangle + \beta\lvert 1\rangle$ $\lvert\alpha\rvert^2 + \lvert\beta\rvert^2 = 1$

$$\vec{v}_{\text{coin}} = \frac{1}{2}\vec{0} + \frac{1}{2}\vec{1}$$

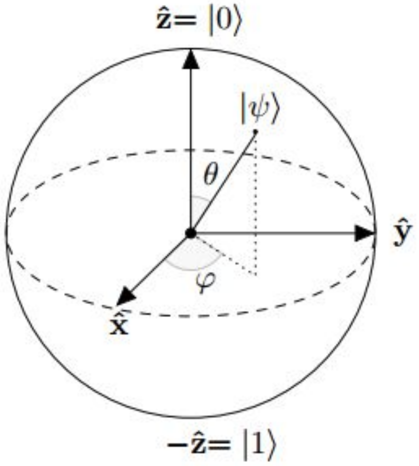
$$|\psi\rangle_{\text{coin}} = \frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\theta}}{\sqrt{2}}|1\rangle$$





Qubit States

Qubit States: Bloch Sphere



$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

Qubit States

Qubit: $$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C}, \qquad |\alpha|^2 + |\beta|^2 = 1$$

Kets: $$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad |\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Measurement yields:

- ‘0’ with probability $|\alpha|^2$
- ‘1’ with probability $|\beta|^2$

Qubit States

Qubit: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C}, \qquad |\alpha|^2 + |\beta|^2 = 1$

Kets: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad |\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$

Bras: $\langle 0| = (1,\ 0) \qquad \langle 1| = (0,\ 1) \qquad \langle\psi| = (\overline{\alpha}, \overline{\beta})$

Brackets
(Inner Product):

$$|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$$
$$\langle\phi|\psi\rangle = \overline{\gamma}\alpha + \overline{\delta}\beta = \overline{\langle\psi|\phi\rangle}$$

Qubit States: Multi-qubit states

Multiple qubits:

$$|\psi\rangle_{n-1} \otimes \ldots \otimes |\psi\rangle_2 \otimes |\psi\rangle_1 \otimes |\psi\rangle_0$$

Tensor product:

$$|\psi\rangle \otimes |\phi\rangle = (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle)$$
$$= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle$$

Vector form:

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \alpha_0 \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \\ \alpha_1 \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \alpha_0\,\beta_0 \\ \alpha_0\,\beta_1 \\ \alpha_1\,\beta_0 \\ \alpha_1\,\beta_1 \end{pmatrix}$$

Qubit States: Multi-qubit states

Associative: $$(|\psi\rangle \otimes |\phi\rangle) \otimes |\gamma\rangle = |\psi\rangle \otimes (|\phi\rangle \otimes |\gamma\rangle)$$

Not commutative: $$|\psi\rangle \otimes |\phi\rangle \neq |\phi\rangle \otimes |\psi\rangle$$

Single Qubit Operations

Qubit Operations

Unitary Operators (Gates):

$$UU^\dagger = U^\dagger U = I$$

$$|\psi\rangle \rightarrow |\psi'\rangle = U|\psi\rangle$$

Preserve inner product:

$$\langle\phi| \rightarrow \langle\phi'| = \langle\phi|U^\dagger$$

$$\langle\phi|\psi\rangle \rightarrow \langle\phi'|\psi'\rangle = \langle\phi|U^\dagger U|\psi\rangle = \langle\phi|\psi\rangle$$

Qubit Operations: Pauli gates

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Qubit Operations: Identity gate

$$I|0\rangle = |0\rangle \qquad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$I|1\rangle = |1\rangle \qquad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Qubit Operations: Pauli-X (NOT) gate

$$X|0\rangle = |1\rangle$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$X|1\rangle = |0\rangle$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Qubit Operations: Pauli-Y gate

$$Y|0\rangle = i|1\rangle$$

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = i \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$Y|1\rangle = -i|0\rangle$$

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -i \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Qubit Operations: Pauli-Z gate

$$Z|0\rangle = |0\rangle$$

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$Z|1\rangle = -|1\rangle$$

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Qubit Operations: Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Qubit Operations: Hadamard gate

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Multi-Qubit Operations

Qubit Operations: Multiple qubits

Examples:

$$I \otimes X \left(|0\rangle \otimes |0\rangle\right) = |0\rangle \otimes |1\rangle = I_1 X_0 |00\rangle = |01\rangle$$

$$X \otimes H \left(|0\rangle \otimes |0\rangle\right) = |1\rangle \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right) = X_1 H_0 |00\rangle = \frac{1}{2} \left(|10\rangle + |11\rangle\right)$$

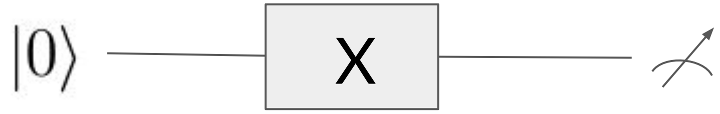
Qubit Operations: Multiple qubits

$$A \otimes B = \begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix} \otimes \begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix} = \begin{pmatrix} A_{00}[B] & A_{01}[B] \\ A_{10}[B] & A_{11}[B] \end{pmatrix}$$

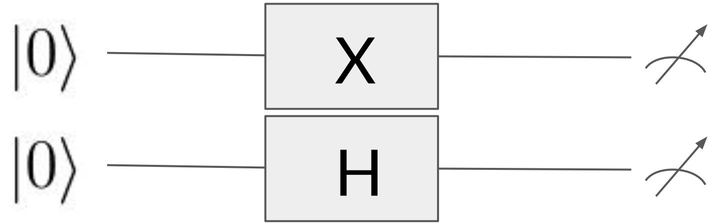
$$= \begin{pmatrix} A_{00}B_{00} & A_{00}B_{01} & A_{01}B_{00} & A_{01}B_{01} \\ A_{00}B_{10} & A_{00}B_{11} & A_{01}B_{10} & A_{01}B_{11} \\ A_{10}B_{00} & A_{01}B_{01} & A_{11}B_{00} & A_{11}B_{01} \\ A_{10}B_{10} & A_{11}B_{11} & A_{11}B_{10} & A_{11}B_{11} \end{pmatrix}$$

Qubit Operations: Quantum Circuits

$$X|0\rangle = |1\rangle$$



$$X_1 H_0 |00\rangle$$



Qubit Operations: Controlled Operations

$$cU = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$$

If qubit *1* is in the state |0> , apply *I* (identity) to qubit *0*
Else if qubit *1* is in the state |1>, apply *U* to qubit *0*

For example,

$$CNOT = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Entangled States

A state that cannot be written as a product state, i.e.

$$|\psi\rangle \neq |\xi\rangle \otimes |\phi\rangle$$

An example of a state that is not entangled:

$$\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) = |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

An example of a state that is entangled:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Qubit Operations: Projections

Project a ket/bra along a given ket/bra via its corresponding projection operator.

For some $|\psi\rangle$ the corresponding projection operator is given by the outer product

$$P_\psi = |\psi\rangle\langle\psi| = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} (\overline{\alpha}, \overline{\beta}) = \begin{pmatrix} |\alpha|^2 & \alpha\overline{\beta} \\ \beta\,\overline{\alpha} & |\beta|^2 \end{pmatrix}$$

e.g.

$$P_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, P_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, P_+ = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Measurement in the standard basis

Computational basis: $\{|0\rangle, |1\rangle\}$

Measurement yields:

- '0' with probability $\langle\psi|P_0|\psi\rangle = \langle\psi|0\rangle\langle 0|\psi\rangle = |\langle 0|\psi\rangle|^2$
- '1' with probability $\langle\psi|P_1|\psi\rangle = \langle\psi|1\rangle\langle 1|\psi\rangle = |\langle 1|\psi\rangle|^2$

Measurement in some arbitrary basis

Some other basis: $\{|0'\rangle = U|0\rangle, |1'\rangle = U|1\rangle\}$ $\quad |\psi'\rangle = U^\dagger|\psi\rangle$

Measurement yields:

- 0' with probability $$\langle\psi|P_{0'}|\psi\rangle = \langle\psi|0'\rangle\langle 0'|\psi\rangle = \langle\psi|U|0\rangle\langle 0|U^\dagger|\psi\rangle$$ $$= \langle\psi'|0\rangle\langle 0|\psi'\rangle = |\langle 0|\psi'\rangle|^2$$
- 1' with probability $$\langle\psi|P_{1'}|\psi\rangle = \langle\psi|1'\rangle\langle 1'|\psi\rangle = \langle\psi|U|1\rangle\langle 1|U^\dagger|\psi\rangle$$ $$= \langle\psi'|1\rangle\langle 1|\psi'\rangle = |\langle 1|\psi'\rangle|^2$$

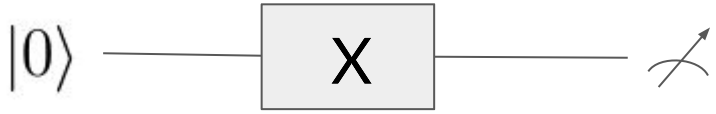
Measurement of $|\Psi\rangle$ in some basis {U|0>, U|1>}

= Measurement of $U^\dagger|\Psi\rangle$ in standard computational basis {|0>, |1>}

Quantum Programs

Quantum Programs

$$X|0\rangle = |1\rangle$$

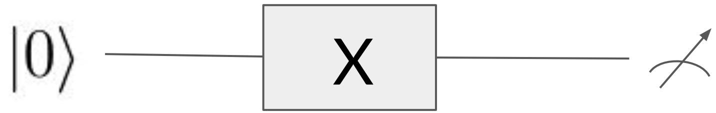



```
from pyquil import Program, get_qc
from pyquil.gates import X

p = Program(X(0))
qc = get_qc('9q-generic-qvm')
results = qc.run_and_measure(p, trials=10)[0]

print (results)
```

```
[1 1 1 1 1 1 1 1 1 1]
```

Quantum Programs

$$X|0\rangle = |1\rangle$$

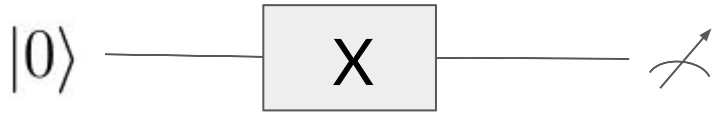



```
from pyquil import Program, get_qc
from pyquil.gates import X, MEASURE

p = Program()
p.declare('ro', 'BIT', 1)
p.inst(X(0))
p.inst(MEASURE(0, 'ro'))
p.wrap_in_numshots_loop(shots=10)

qc = get_qc('9q-generic-qvm')
results = qc.run(qc.compile(p))

print (p)
```

```
DECLARE ro BIT[1]
X 0
MEASURE 0 ro[0]
```

Quantum Programs

$$X|0\rangle = |1\rangle$$

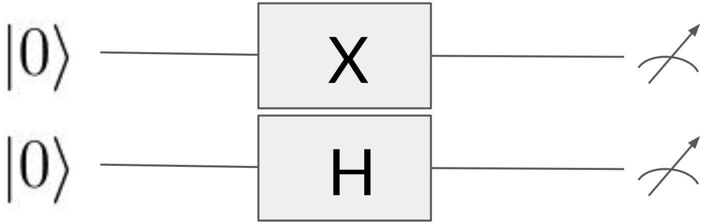



```
from pyquil import Program, get_qc
from pyquil.gates import X, MEASURE

p = Program()
p.declare('ro', 'BIT', 1)
p.inst(X(0))
p.inst(MEASURE(0, 'ro'))
p.wrap_in_numshots_loop(shots=10)

qc = get_qc('9q-generic-qvm')
results = qc.run(qc.compile(p))

print (results)
```

```
[[1]
 [1]
 [1]
 [1]
 [1]
 [1]
 [1]
 [1]
 [1]
 [1]]
```

Quantum Programs

$$X_1 H_0 \; |0\rangle_1 |0\rangle_0$$

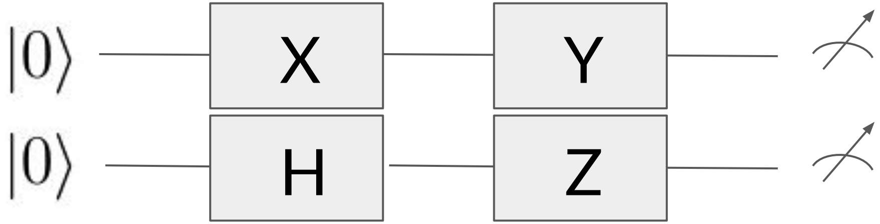



```
from pyquil import Program, get_qc
from pyquil.gates import X, H, MEASURE

p = Program()
ro = p.declare('ro', 'BIT', 2)
p.inst(H(0))
p.inst(X(1))
p.inst(MEASURE(0, ro[0]))
p.inst(MEASURE(1, ro[1]))
p.wrap_in_numshots_loop(shots=10)

qc = get_qc('9q-generic-qvm')
results = qc.run(qc.compile(p))

print (results)
```

```
[[0 1]
 [1 1]
 [0 1]
 [1 1]
 [1 1]
 [0 1]
 [1 1]
 [0 1]
 [0 1]
 [0 1]]
```

Quantum Programs

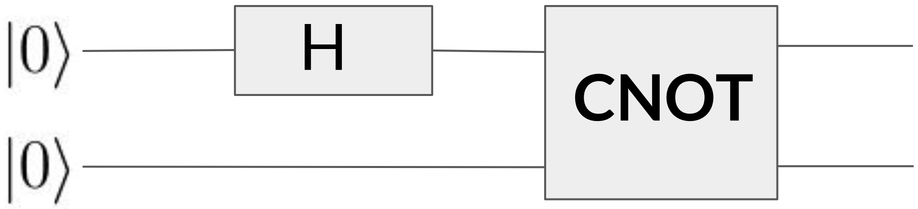
$$Y_1 Z_0 X_1 H_0 \; |0\rangle_1 |0\rangle_0$$




```
from pyquil import Program, get_qc
from pyquil.gates import X, Y, Z, H, MEASURE

p = Program()
ro = p.declare('ro', 'BIT', 2)
p += Program(H(0), X(1), Z(0), Y(1), MEASURE(0, ro[0]), MEASURE(1, ro[1]))
p.wrap_in_numshots_loop(shots=10)

qc = get_qc('9q-generic-qvm')
results = qc.run(qc.compile(p))

print (results)
```

```
[[1 0]
 [1 0]
 [0 0]
 [1 0]
 [1 0]
 [0 0]
 [1 0]
 [1 0]
 [0 0]
 [0 0]]
```

Bell State via CNOT



$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

```
from pyquil import Program
from pyquil.gates import H, CNOT
from pyquil.api import WavefunctionSimulator

p = Program(H(1))
p += Program(CNOT(1, 0))
wfn = WavefunctionSimulator().wavefunction(p)

print (wfn)
```

```
(0.7071067812+0j)|00> + (0.7071067812+0j)|11>
```

Quantum Programming Examples

Quantum Die Example

Example: Quantum Die

Goal: Create a fair N-sided die

Example: Quantum Die

Question: What gate would we use?

$$H^{\otimes n} |0\rangle^{\otimes n} = (H |0\rangle)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} |z\rangle$$

Example: Quantum Die

Question: How many qubits would we use?

$$2^n = N \Rightarrow n = log_2 N$$

Quantum Teleportation Example

Example: Quantum Teleportation

Goal: Teleport a qubit state from Alice to Bob

Example: Quantum Teleportation

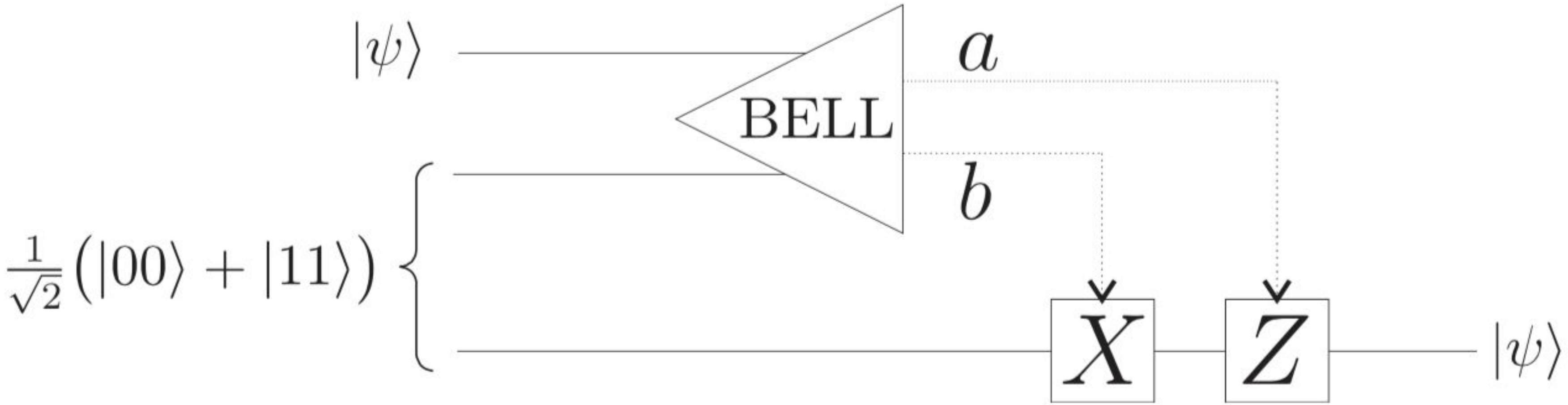
Goal: Teleport a qubit state from Alice to Bob

Scenario: Alice is in possession of a qubit $|\Psi>$, which she would like to teleport over to Bob, who is at some distant location.

Protocol:

- Create a Bell state, giving one qubit each to Alice and Bob
- Have Alice measure both her qubits in the Bell basis, and send her results to Bob
- Have Bob conditionally apply gates to his qubits, based off Alice's measurements, to reconstruct the original qubit at his location

Example: Quantum Teleportation



Classical Control in pyQuil

```
from pyquil import Program
from pyquil.gates import I, X
from pyquil.api import WavefunctionSimulator

p = Program(X(0))
ro = p.declare('ro', 'BIT', 1)
p.measure(0, ro[0]).if_then(ro[0], Program(X(1)), Program(I(1)))
wfn = WavefunctionSimulator().wavefunction(p)

print (wfn)
```

```
(1+0j)|11>
```

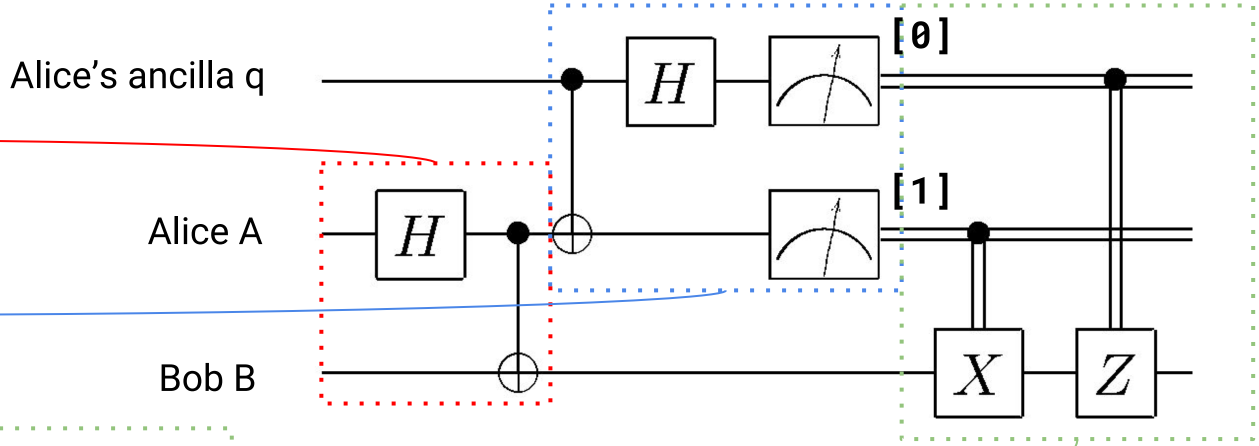
Classical Control in pyQuil

```
from pyquil import Program
from pyquil.gates import I, X
from pyquil.api import WavefunctionSimulator

p = Program(I(0))
ro = p.declare('ro', 'BIT', 1)
p.measure(0, ro[0]).if_then(ro[0], Program(X(1)), Program(I(1)))
wfn = WavefunctionSimulator().wavefunction(p)

print (wfn)
```

```
(1+0j)|00>
```

Example: Quantum Teleportation

```
DEFCIRCUIT TELEPORT A q B:
    # Bell pair
    H          A
    CNOT       A B

    # Teleport
    CNOT       q A
    H          q
    MEASURE    q [0]
    MEASURE    A [1]

    # Classically communicate measurements
    JUMP-UNLESS @SKIP [1]
    X B
    LABEL @SKIP
    JUMP-UNLESS @END [0]
    Z B
    LABEL @END

# If Alice's qubits are 0 and 1
# and Bob's is 5
TELEPORT 0 1
```



Quantum Approximate Optimization Algorithm (QAOA)

Quantum Approximate Optimization Algorithm

Goal: Given binary constraints over bitstrings

$$z \in \{0,1\}^n$$

$$C_\alpha(z) = \begin{cases} 1 & \text{if } z \text{ satisfies the constraint } \alpha \\ 0 & \text{if } z \text{ does not} \end{cases}$$

Find the bitstring that maximizes the objective function

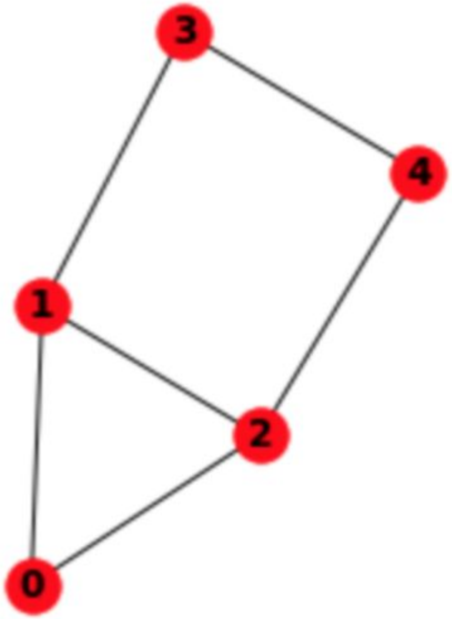
$$\text{argmax}_z C(z) = \text{argmax}_z \sum_{\alpha=1}^{m} C_\alpha(z)$$

MaxCut problem:

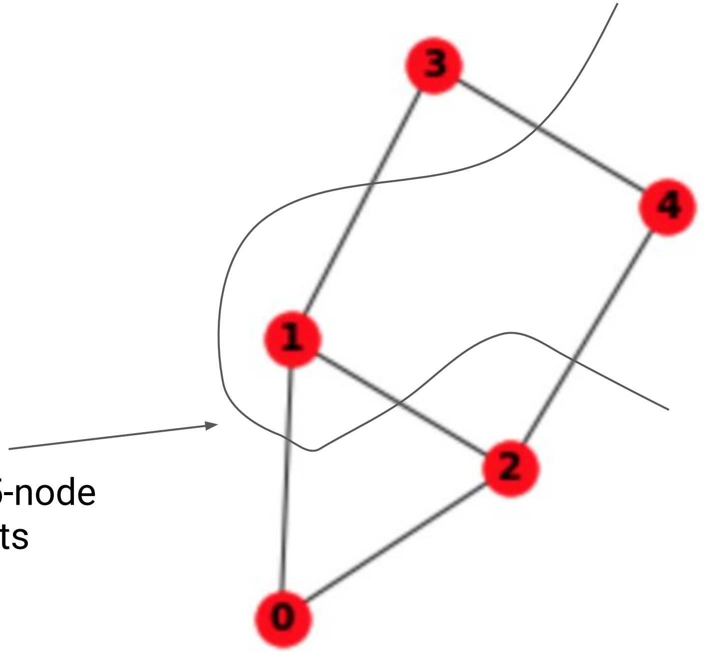
Given some undirected graph with arbitrary (non-negative) weights, find a partition $(S, \overline{S})$ of the graph's nodes (a 'cut' of the graph) that maximizes the weights along the cut

$$\sum_{i \in S, j \in \overline{S}} w_{ij}$$

Quantum Approximate Optimization Algorithm



Quantum Approximate Optimization Algorithm



MaxCut for simple 5-node graph with all weights either 0 or 1.

MaxCut solution (as a bitstring):

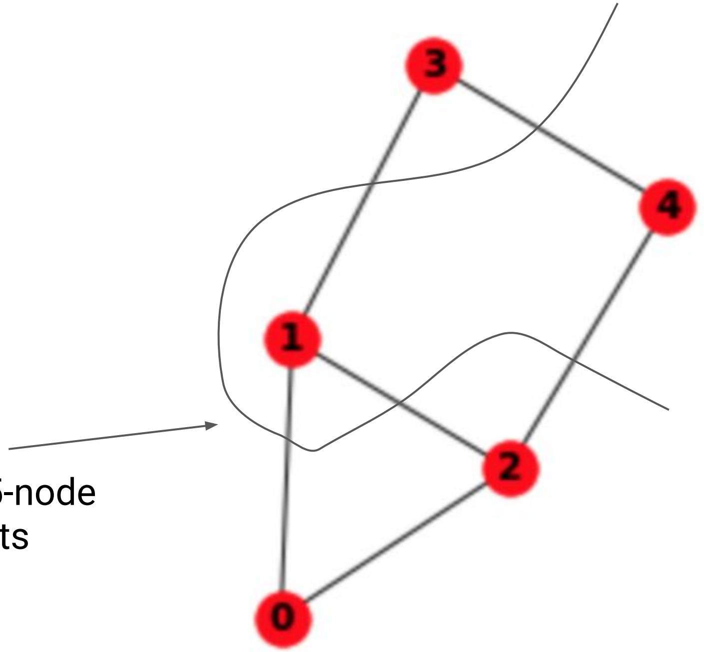
01001 OR 10110

On a quantum computer:

$$\frac{1}{\sqrt{2}}(|01001\rangle + |10110\rangle)$$

(ideally)

Quantum Approximate Optimization Algorithm



MaxCut for simple 5-node graph with all weights either 0 or 1.

MaxCut objective function:

$$\sum_{i \in S, j \in \overline{S}} w_{ij}$$

On a quantum computer:

$$\frac{1}{2} \sum_{i, j \in V} w_{ij} \frac{1 - Z_i Z_j}{2}$$

Noise and Quantum Computation

‘Pure’ quantum states

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

evolve via Unitary operations

$$|\psi\rangle \rightarrow |\psi'\rangle = U|\psi\rangle \qquad UU^\dagger = U^\dagger U = I$$

More generally, quantum states are described by “Density Matrix”

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

evolving via Kraus operations (“quantum channel”)

$$\rho \to \sum_i K_i \rho K_i^\dagger, \qquad \sum_i K_i^\dagger K_i = I$$

For example,

$$\rho = \frac{1}{2}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right) = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Not to be confused with

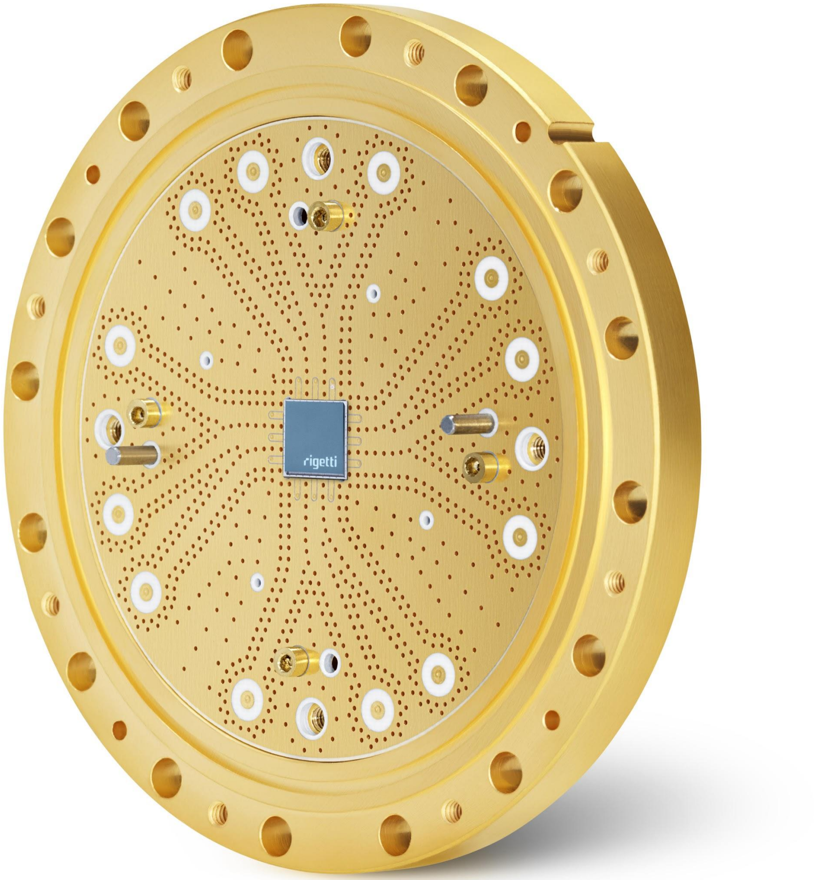
$$\rho = \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)\left(\frac{1}{\sqrt{2}}(\langle 0| + \langle 1|)\right) = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Example of quantum channel/set of Kraus operators/noise model:

$$\{\sqrt{p}X, \sqrt{1-p}Z\}$$

Quantum state passing through the channel/experiencing the noise transforms to:

$$\rho \rightarrow pX^\dagger \rho X + (1-p)Z^\dagger \rho Z$$



Thank you, and keep in touch!

amy@rigetti.com

Join our Slack channel:

rigetti-forest.slack.com

rigetti