



NISQ 時代の量子コンピュータ

Shoko Utsunomiya

Solutions Architect
Amazon Web Services Japan G.K.

宇都宮聖子, Ph.D.

アマゾン ウェブ サービス ジャパン合同会社
技術統括本部
シニア機械学習・量子ソリューションアーキテクト

2018年よりAWSにジョイン
AWS 機械学習・量子コンピュータを担当

前職は、自動車 OEM にて機械学習を用いた自動運転開発
前々職は、国立情報学研究所にて量子コンピュータの研究者

好きな AWS サービス: Amazon SageMaker, Amazon Braket



Twitter: shokout

Contents

- 量子計算の近況
- 量子計算を実現するのに必要な要素
- 量子コンピューティングと NISQ
- AWS Center for Quantum Computing
- Amazon Braket
 - 量子ハードウェア
 - 量子コンピューティングシミュレータ
 - 量子ソフトウェア開発のためのライブラリ
 - Amazon Quantum Solutions Lab
 - 量子アプリケーション構築のための AWS パートナー
 - お客様事例

量子コンピュータを使った研究開発

お客様からよくいただく質問

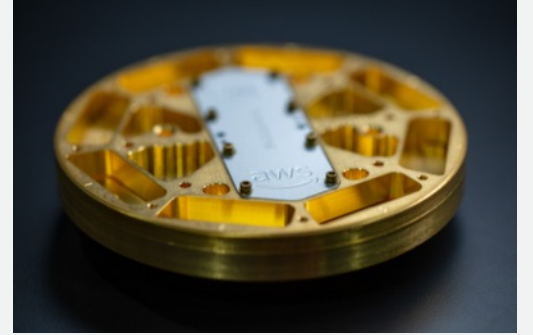
- 既存のソリューションや機械学習を凌駕？すぐに置き換えられますか？
- 量子コンピューティングはどんな分野に使える？
- 量子コンピューティングの実力は？課題は？
- クラウドで量子コンピューティング、どうやって？何が嬉しい？安全？
- どんな人が Amazon Braket を使っている？ユースケースは？
- ハードルが高そう、どうすれば量子ソフトウェア開発を始められる？

量子計算の近況



量子コンピュータで既存のソリューションを凌駕できる？

- 現状開発者が使えるハードウェアのサイズは発展途上
 - ハードウェアの性能やスケールは急速に進化
 - 量子古典ハイブリッド手法など有望なアプローチも
- 量子アルゴリズムの探索は学術的にも高難易度
 - 素因数分解、検索、量子機械学習、量子化学計算など
- アプリケーション探索の敷居の高さ
 - 限られたアルゴリズムを適用した[ビジネスアプリケーションの検討](#)は量子コンピュータの専門知識がないと難しい
 - ビジネスニーズから出発すると適用先がかなり限定される

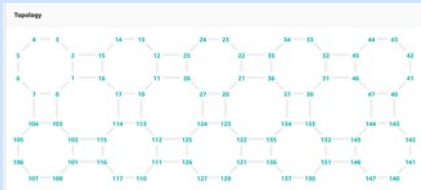
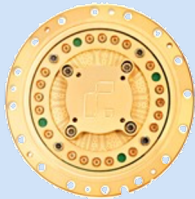
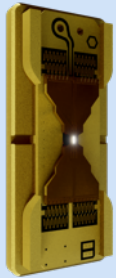


量子ビット: 量子演算の最小単位

量子計算の種類

ゲートベース 量子コンピュータ

ユニバーサルゲートセット
を用いて量子計算を行う

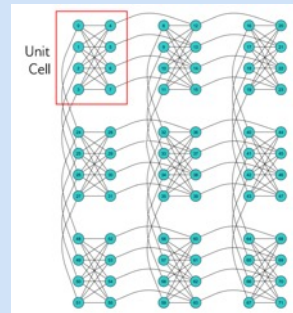
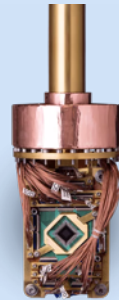


Rigetti M-2 80 qubits

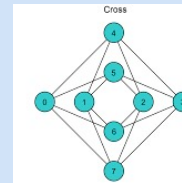
量子アニーリング

量子アニーリング
で解けるイジング問題に実装する

c.f. 量子断熱計算



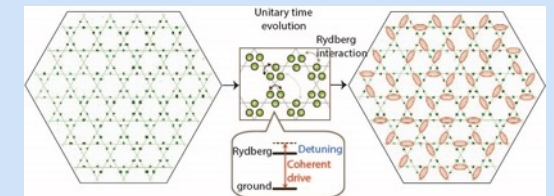
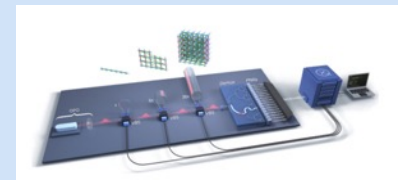
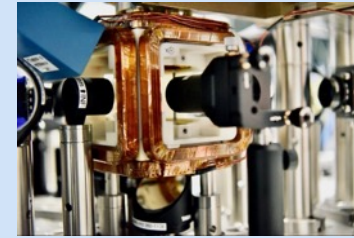
Chimera Graph



Pegasus Graph

量子 シミュレーション

コントロール可能な
量子状態を用いて別の
量子系の状態をシミュレートする



Quantum spin liquids with
a Rydberg AHS device



物理系を計算機でシミュレートする (量子シミュレーション)



“物理系をシミュレートする計算機を見つけるというのは、僕にとっては素晴らしい取り組みに思える
(中略)

自然界は「古典」ではない、自然をシミュレートしようとするなら量子力学を使った方が良い。コンピュータで自然を正確にシミュレートするための可能性について話したい”

R. Feynman, Simulating Physics with Computers, 81

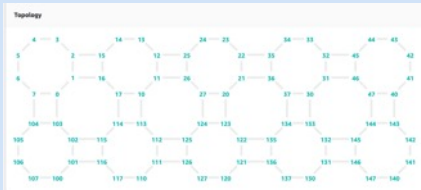
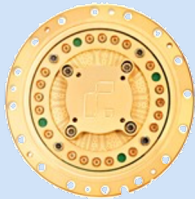
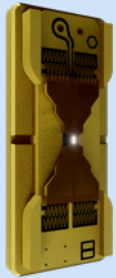
re:Invent 2019 (CMP213-R) から抜粋

<https://www.youtube.com/watch?v=BV9TZWuAwyk>

量子計算の種類

ゲートベース 量子コンピュータ

ユニバーサルゲートセット
を用いて量子計算を行う

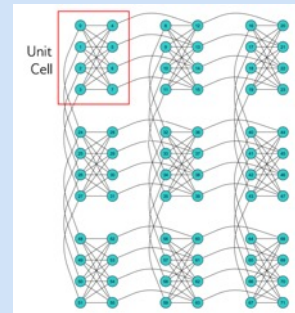
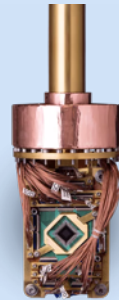


Rigetti M-2 80 qubits

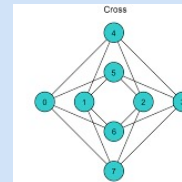
量子アニーリング

量子アニーリング
で解けるイジング問題に実装する

c.f. 量子断熱計算



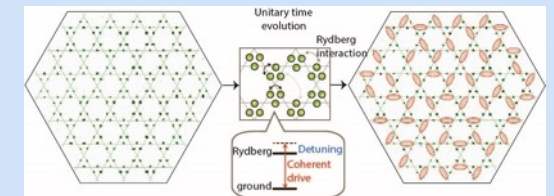
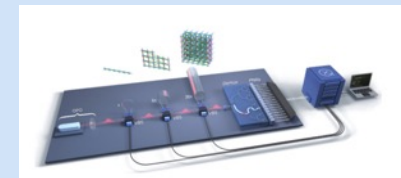
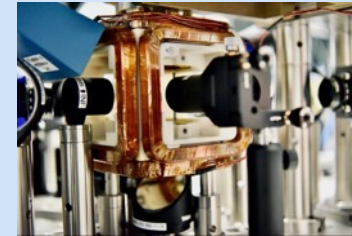
Chimera Graph



Pegasus Graph

量子 シミュレーション

コントロールラブルな
量子状態を用いて別の
量子系の状態をシミュレートする



Quantum spin liquids with
a Rydberg AHS device



量子コンピュータ どんな分野に応用できる？

- 期待されている量子コンピュータのユースケース



- 量子化学計算、創薬、材料化学

- 量子機械学習

- 金融（ポートフォリオ最適化、リスク計算、オプション価格決定）



- 交通系（配送計画最適化、交通経路探索）

- 最適化（レコメンデーション、人材配置）

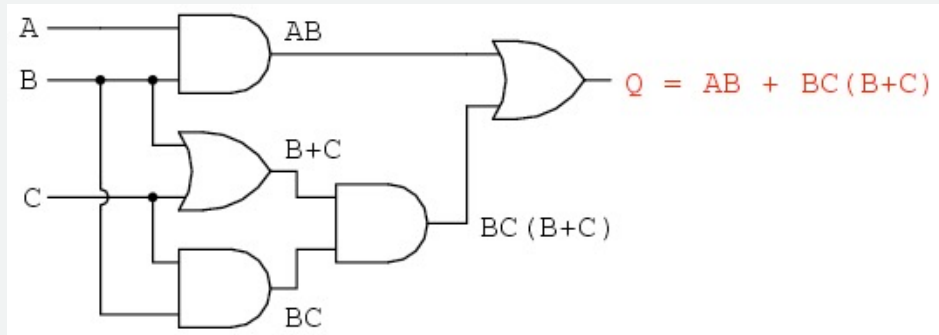


利用できるハードウェアは限られた量子ビット数
アルゴリズム・アプリケーションも研究段階

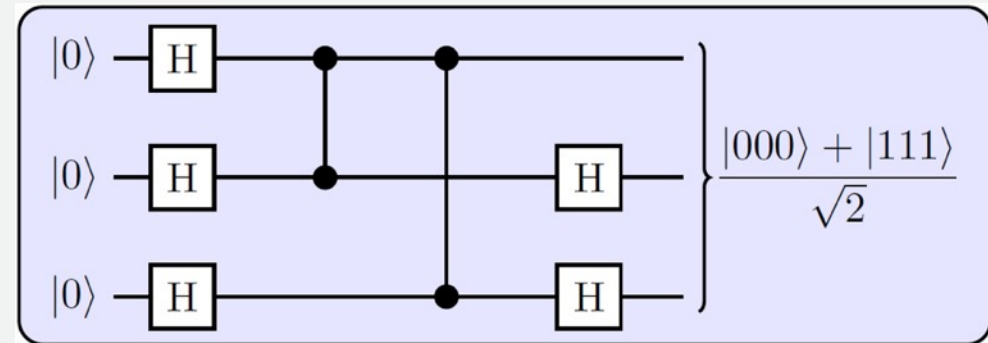
量子計算の要素

古典回路と量子回路

- 量子コンピューティングの基本操作 = 量子ゲート操作
- 操作可能な量子系を計算に用いる

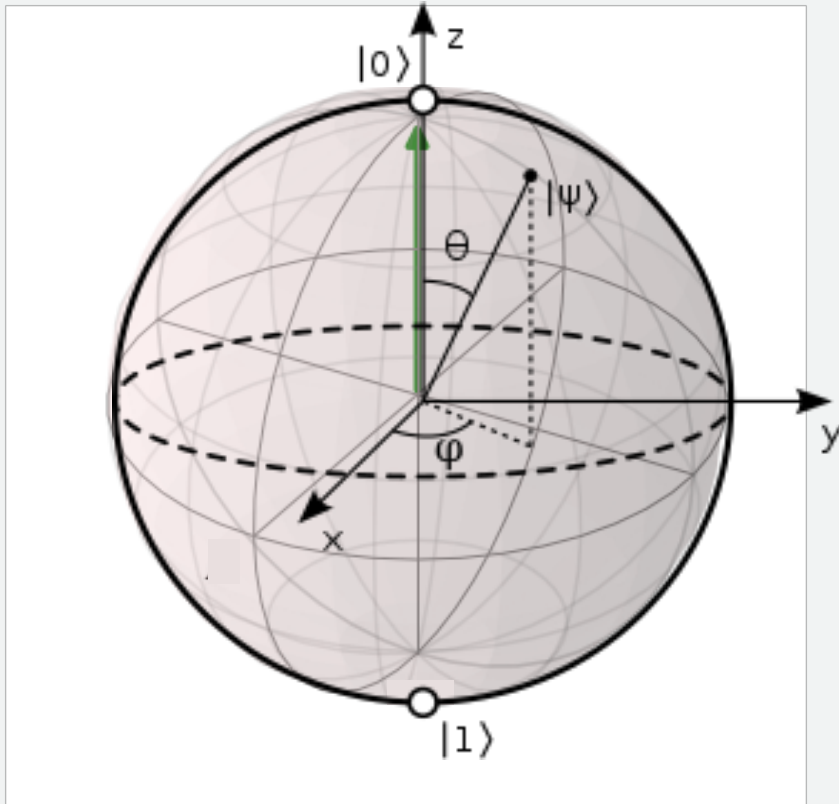


ブール回路



量子回路

量子ビット (Qubit) とヒルベルト空間



ブロッホ球 (Bloch Sphere)

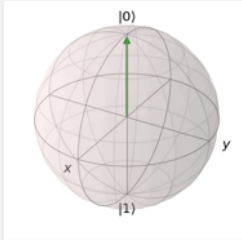
二つの直交する純粋状態の重ね合わせで表現できる量子状態を単位球面上に表す表記法

$$\begin{aligned} |\psi\rangle &= \cos(\theta/2)|0\rangle + e^{i\phi} \sin(\theta/2)|1\rangle \\ &= \cos(\theta/2)|0\rangle + (\cos \phi + i \sin \phi) \sin(\theta/2)|1\rangle \end{aligned}$$

<https://ja.wikipedia.org/wiki/%E3%83%96%E3%83%AD%E3%83%83%E3%83%9B%E7%90%83>

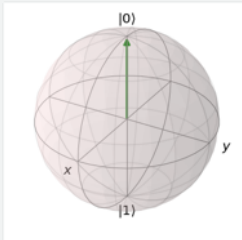
ユニバーサルゲート・量子論理ゲート

1 量子ビット操作と2量子ビット操作



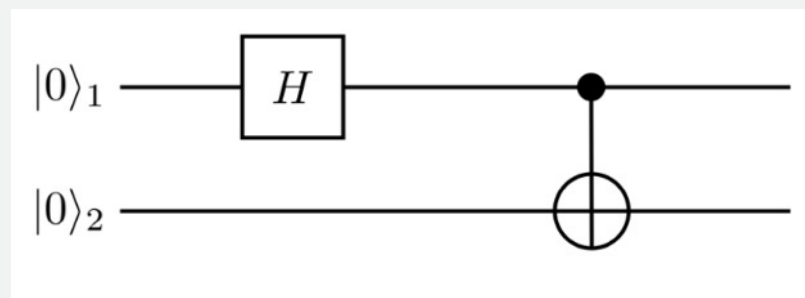
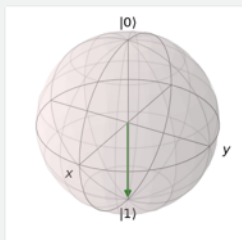
1 量子ビット操作

量子ビットに対するパウリ演算
位相操作、ビット反転など



2 量子ビット操作

Control Qubit に対する Target Qubit の操作

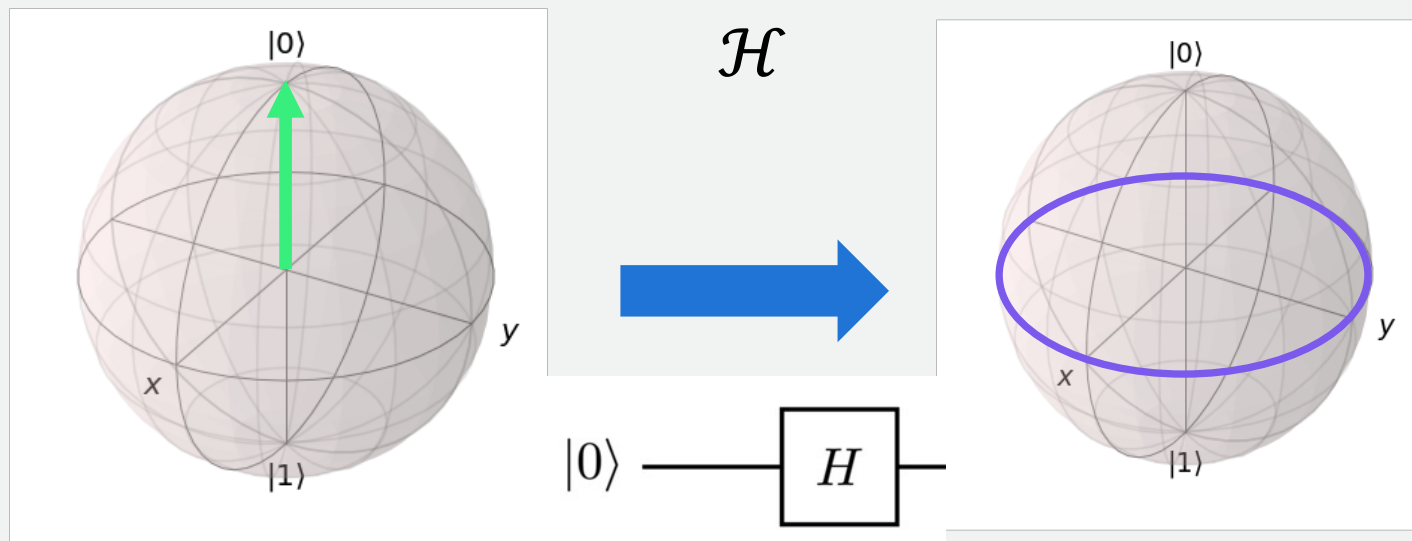


Operator	Gate(s)	Matrix
Pauli-X (X)	\oplus	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

https://en.wikipedia.org/wiki/Quantum_logic_gate#/media/File:Quantum_Logic_Gates.png

Hadamard 変換: 量子力学的な重ね合わせの作成

Hadamard 変換



$$|\psi\rangle = |0\rangle$$

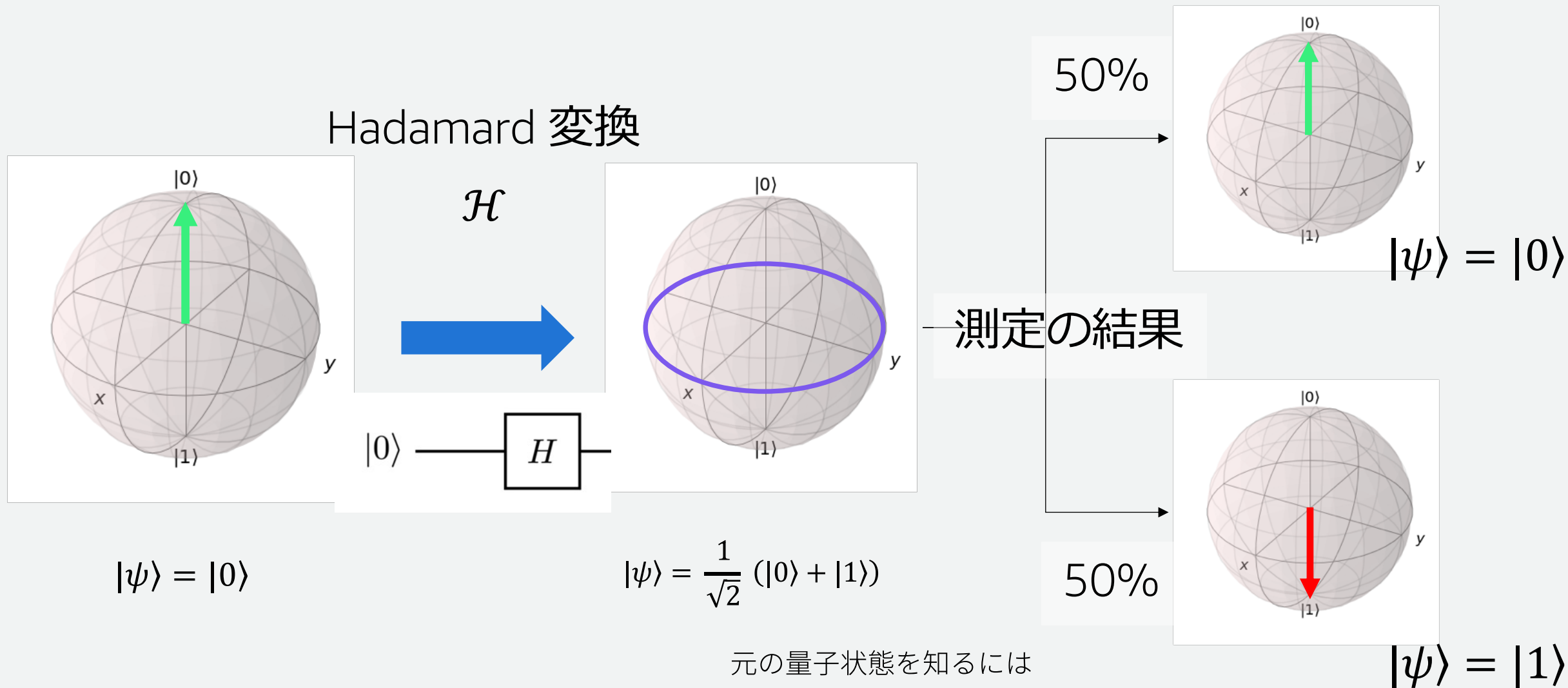
$$|\psi\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$$

線形重ね合わせ状態

観測するまで量子状態は複数の状態にまたがっている

複数回の観測によって元の状態の確率分布が求められる

波動関数の収縮により、確率的に測定結果が確率的に求まる

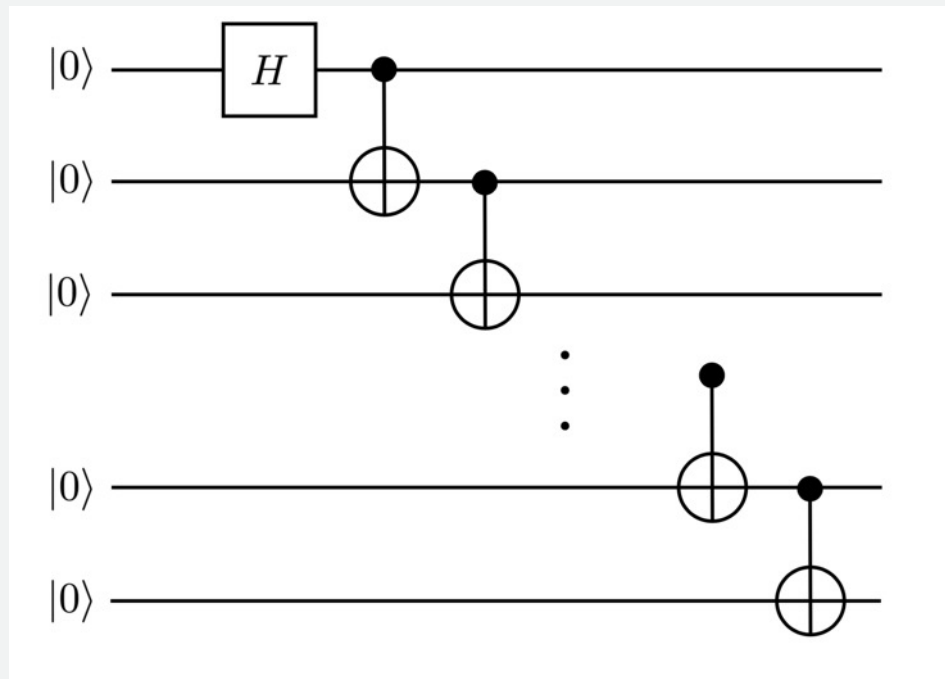


元の量子状態を知るには

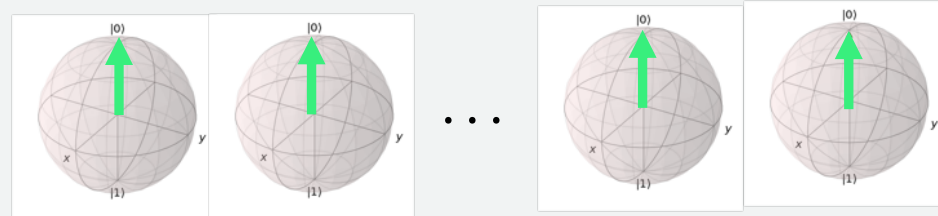
統計的に十分な数の実験が必要になる

GHZ State - 量子もつれ状態を作る

Greenberger-Horne-Zeilinger state



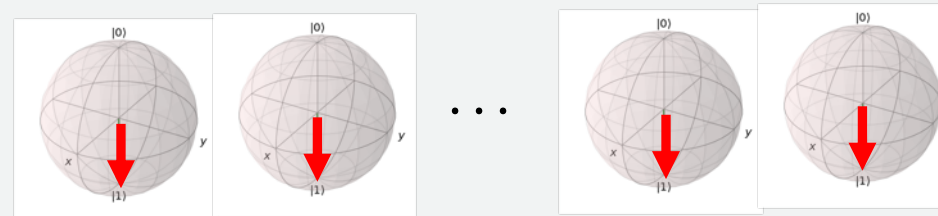
50%



$$|\psi\rangle = |00 \dots 00\rangle$$

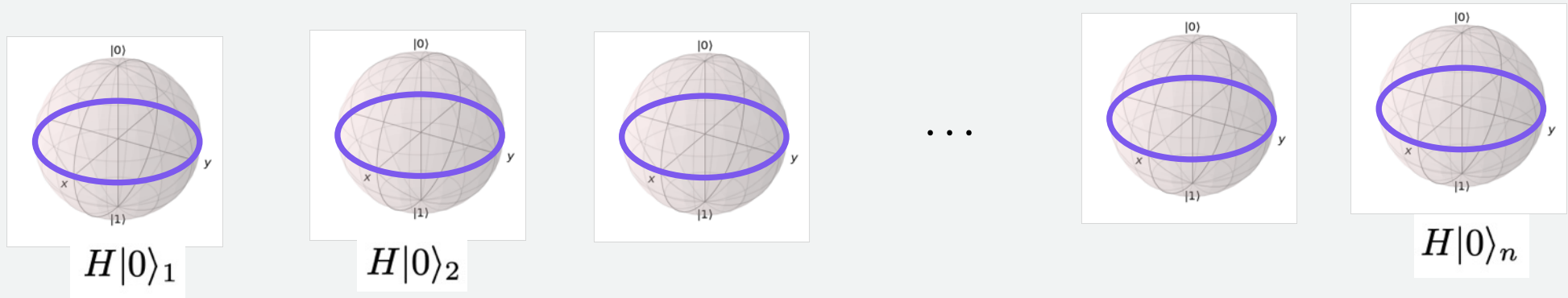
測定の結果

50%

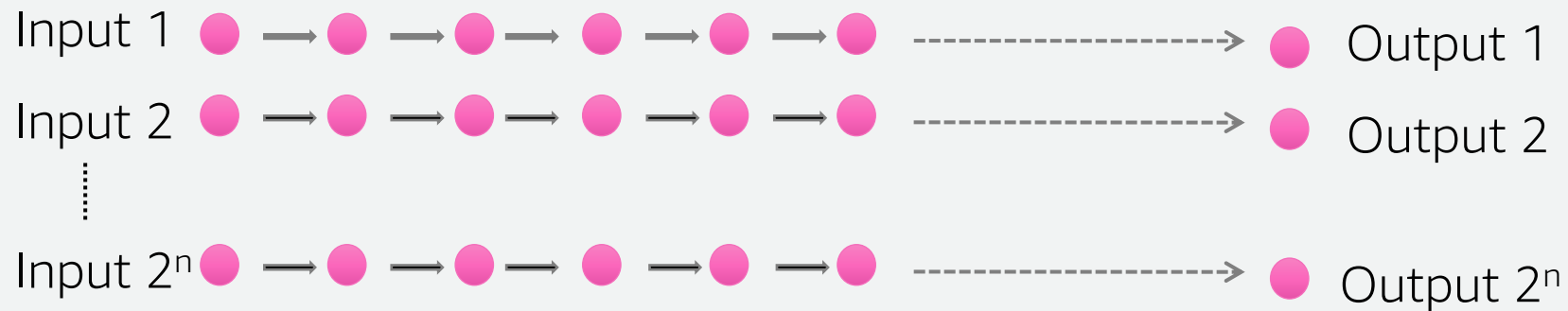


$$|\psi\rangle = |11 \dots 11\rangle$$

量子計算パラレルリズム



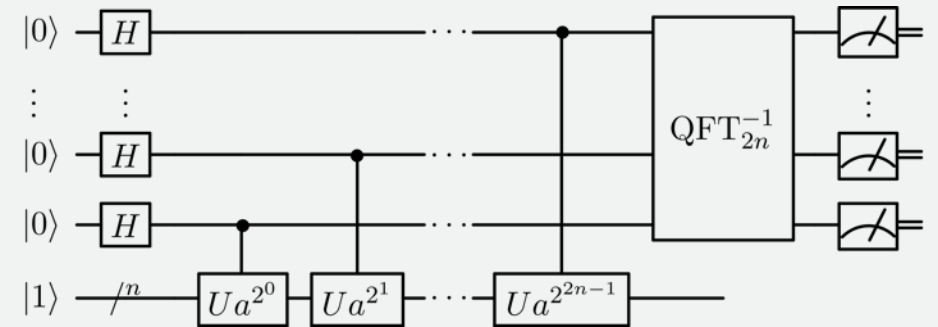
$$\begin{aligned}
 H|0\rangle_1 \otimes H|0\rangle_2 \otimes \cdots \otimes H|0\rangle_n &= \frac{1}{\sqrt{2}}(|0\rangle_1 + |1\rangle_1) \otimes \frac{1}{\sqrt{2}}(|0\rangle_2 + |1\rangle_2) \otimes \cdots \otimes \frac{1}{\sqrt{2}}(|0\rangle_n + |1\rangle_n) \\
 &= \frac{1}{\sqrt{2^n}}(|00 \cdots 000\rangle + \cdots + |01 \cdots 010\rangle + \cdots + |11 \cdots 111\rangle)
 \end{aligned}$$



n qubit

波束の収縮

Shor's algorithm



https://en.wikipedia.org/wiki/Shor%27s_algorithm

量子計算を実現するのに 必要な要素

(実用的な) 量子計算を実現するのに必要な要素

- 量子ビットの大規模化
 - 量子ビット・量子ゲートの高精度化
 - 量子ビットの大規模化・スケーラビリティ
- 誤り訂正技術
 - より少ない量子ビットで大規模化ができる誤り訂正符号
 - Shor's algorithm に必要な量子ビット数は 10万～100万量子ビット
- アルゴリズム
 - 利用可能な量子コンピュータで実装できるアルゴリズム
- 量子コンピュータアーキテクチャ
 - スケーラブルなアーキテクチャ・トポロジー
 - 誤り訂正符号やアルゴリズムの実装 etc…

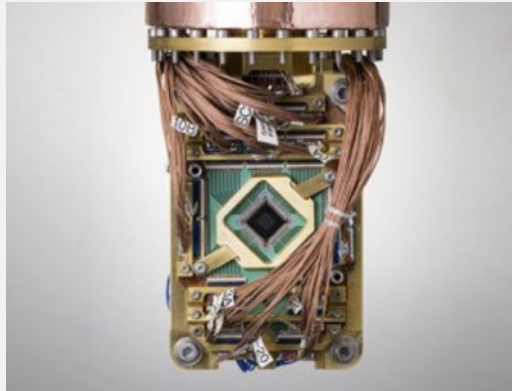
さまざまな条件が揃う必要がある

量子コンピュータの大規模化が難しい理由

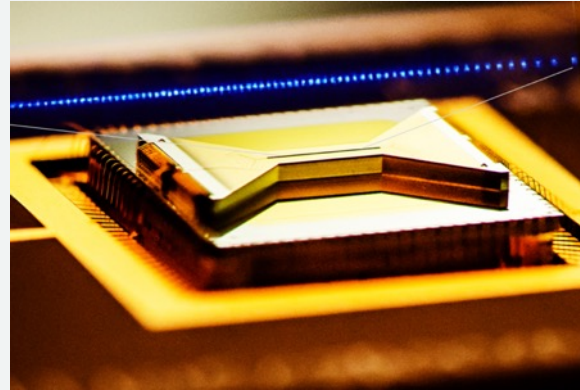
- 量子ビットの操作が完全ではない
 - 有限のエラーを含むゲート操作
 - 量子ゲートの欠陥
 - 量子ビット・量子ゲートにばらつきがある
- 量子ビットの寿命時間（コヒーレンス時間）が有限（計算時間に比べて十分長い必要がある）
- 誤り訂正可能でスケーラブルかつ実装可能なアーキテクチャ

 量子誤り訂正が必要

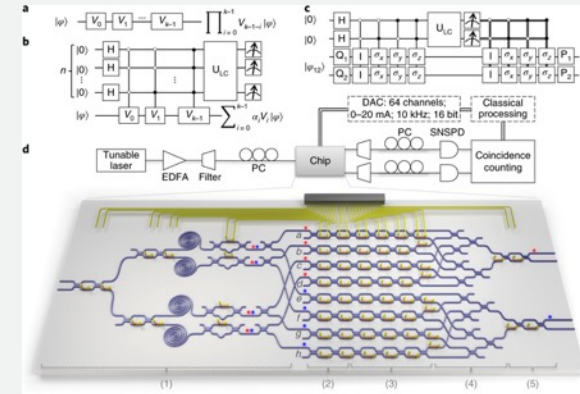
様々なデバイスで量子コンピュータの大規模化を目指す



Superconducting qubits
超伝導量子ビット



Trapped ions
イオントラップ



Photonics
光量子ビット

商用デバイスで利用可能な量子ビット数は数量子ビット～数10ビット程度

re:Invent 2019 (CMP213-R) から抜粋
<https://www.youtube.com/watch?v=BV9TZWuAwyk>
<https://www.nature.com/articles/s41566-018-0236-y>

既知の量子アルゴリズム Quantum Algorithm Zoo

Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@microsoft.com. (Alternatively, you may submit a pull request to the [repository](#) on github.) Your help is appreciated and will be [acknowledged](#).

Algebraic and Number Theoretic Algorithms

Algorithm: Factoring

Speedup: Superpolynomial

Description: Given an n -bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\tilde{O}(n^3)$ time [82, 125]. The fastest known classical algorithm for integer factorization is the general number field sieve, which is believed to run in time $2^{\tilde{O}(n^{1/3})}$. The best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n^{1/4+o(1)}})$ via the Pollard-Strassen algorithm [252, 362]. Shor's factoring algorithm breaks RSA public-key encryption and the closely related quantum algorithms for discrete logarithms break the DSA and ECDSA digital signature schemes and the Diffie-Hellman key-exchange protocol. A quantum algorithm even faster than Shor's for the special case of factoring "semiprimes", which are widely used in cryptography, is given in [271]. If small factors exist, Shor's algorithm can be beaten by a quantum algorithm using Grover search to speed up the elliptic curve factorization method [366]. Additional optimized versions of Shor's algorithm are given in [384, 386]. There are proposed classical public-key cryptosystems not believed to be broken by quantum algorithms, cf. [248]. At the core of Shor's factoring algorithm is order finding, which can be reduced to the [Abelian hidden subgroup problem](#), which is solved using the quantum Fourier transform. A number of other problems are known to reduce to integer factorization including the membership problem for matrix groups over fields of odd order [253], and certain diophantine problems relevant to the synthesis of quantum circuits [254].

Algorithm: Discrete-log

Speedup: Superpolynomial

Description: We are given three n -bit numbers a , b , and N , with the promise that $b = a^s \pmod N$ for some s . The task is to find s . As shown by Shor [82], this can be achieved on a quantum computer in $\text{poly}(n)$ time. The fastest known classical algorithm requires time superpolynomial in n . By similar techniques to those in [82], quantum computers can solve the discrete logarithm problem on elliptic curves, thereby breaking elliptic curve cryptography [109, 14]. A further optimization to Shor's algorithm is given in [385]. The superpolynomial quantum speedup has also been extended to the discrete logarithm problem on semigroups [203, 204]. See also [Abelian hidden subgroup](#).

Algorithm: Pell's Equation

Speedup: Superpolynomial

Description: Given a positive nonsquare integer d , Pell's equation is $x^2 - dy^2 = 1$. For any such d there are infinitely many pairs of integers (x, y) solving this equation. Let (x_1, y_1) be the pair that minimizes $x + y\sqrt{d}$. If d is an n -bit integer (i.e. $0 \leq d < 2^n$), (x_1, y_1) may in general require exponentially many bits to write down. Thus it is in general impossible to find (x_1, y_1) in polynomial time. Let $R = \log(x_1 + y_1\sqrt{d})$. $[R]$ uniquely identifies (x_1, y_1) . As shown by Hallgren [49], given a n -bit number d , a quantum computer can find $[R]$ in $\text{poly}(n)$ time. No polynomial time classical algorithm for this problem is known. Factoring reduces to this problem. This algorithm breaks the Buchman-Williams cryptosystem. See also [Abelian hidden subgroup](#).

Navigation

[Algebraic & Number Theoretic](#)

[Oracular](#)

[Approximation and Simulation](#)

[Optimization, Numerics, & Machine Learning](#)

[Acknowledgments](#)

[References](#)

Translations

This page has been translated into:

[Japanese](#)

[Chinese](#)

Other Surveys

For overviews of quantum algorithms I recommend:

[Nielsen and Chuang](#)

[Childs](#)

[Preskill](#)

[Mosca](#)

[Childs and van Dam](#)

[van Dam and Sasaki](#)

[Bacon and van Dam](#)

[Montanaro](#)

[Hidary](#)

Terminology

If there exists a positive constant α such that the runtime $C(n)$ of the best known classical algorithm and the runtime $Q(n)$ of the quantum algorithm satisfy $C = 2^{\Omega(Q)}$, then I call the speedup superpolynomial. Otherwise I call it polynomial. For a review of the $O, \Omega, \Theta, \tilde{O}, \dots$ notations see the [Wikipedia article](#).

About

Author:

Quantum Algorithm Zoo全訳 オラクルアルゴリズム

本記事はQuantum Algorithm Zoo全訳シリーズの一部です。

以下のリンクから他の記事に飛べます。

- [前書き](#)
- [代数アルゴリズム、数論アルゴリズム](#)
- [オラクルアルゴリズム\(本記事\)](#)
- [近似アルゴリズム、シミュレーションアルゴリズム](#)

訳者注：この項目では、入力に対してある決まった規則で出力を返すオラクル(神託機械)が与えられたとき、できるだけ少ない回数オラクルに問い合わせ(クエリ)をすることでオラクルに関する何らかの性質を求める問題を考える。この時、問題を解くためにオラクルに何回問い合わせをする必要があるかを、問い合わせ計算量(query complexity)と呼ぶ。問い合わせ計算量のSpeedupは、必ずしも同じスケールでの時間計算量の改善を意味するわけではないことに注意。

Algorithm: 探索 (Grover Search, グローバー探索)

Speedup: 多項式的

詳細

N 種類の入力を受け付けるオラクルが与えられる。或る入力 w のみが"当たり"であり、これに対応する出力は1である。それ以外の全ての入力に対応する出力は0である。探索問題は、 w を見つける問題である。古典計算では、 $\Omega(N)$ の回数の問い合わせが必要

- [代数アルゴリズム、数論アルゴリズム](#)
- [オラクルアルゴリズム](#)
- [近似アルゴリズム、シミュレーションアルゴリズム](#)

<https://quantumalgorithmzoo.org/>

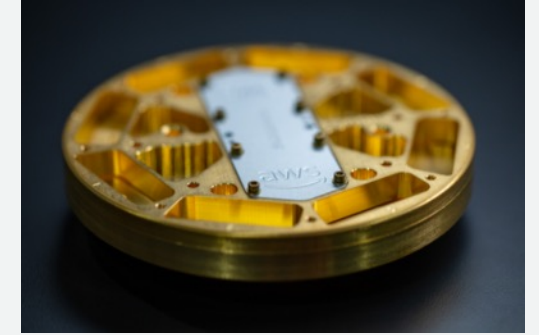
<https://www.qmedia.jp/algorithm-zoo/>



量子誤り訂正とNISQ

量子コンピュータの開発に誤り訂正がなぜ重要か

- 量子ビットの操作の不完全さ
 - 有限のエラーを含むゲート操作
 - 量子ゲートの欠陥
 - 量子ビット・量子ゲートにばらつきがある
- 量子ビットのコヒーレンス時間（寿命時間）が有限
 - 計算時間に比べてコヒーレンス時間が十分長い必要がある



量子ビット: 量子演算の最小単位

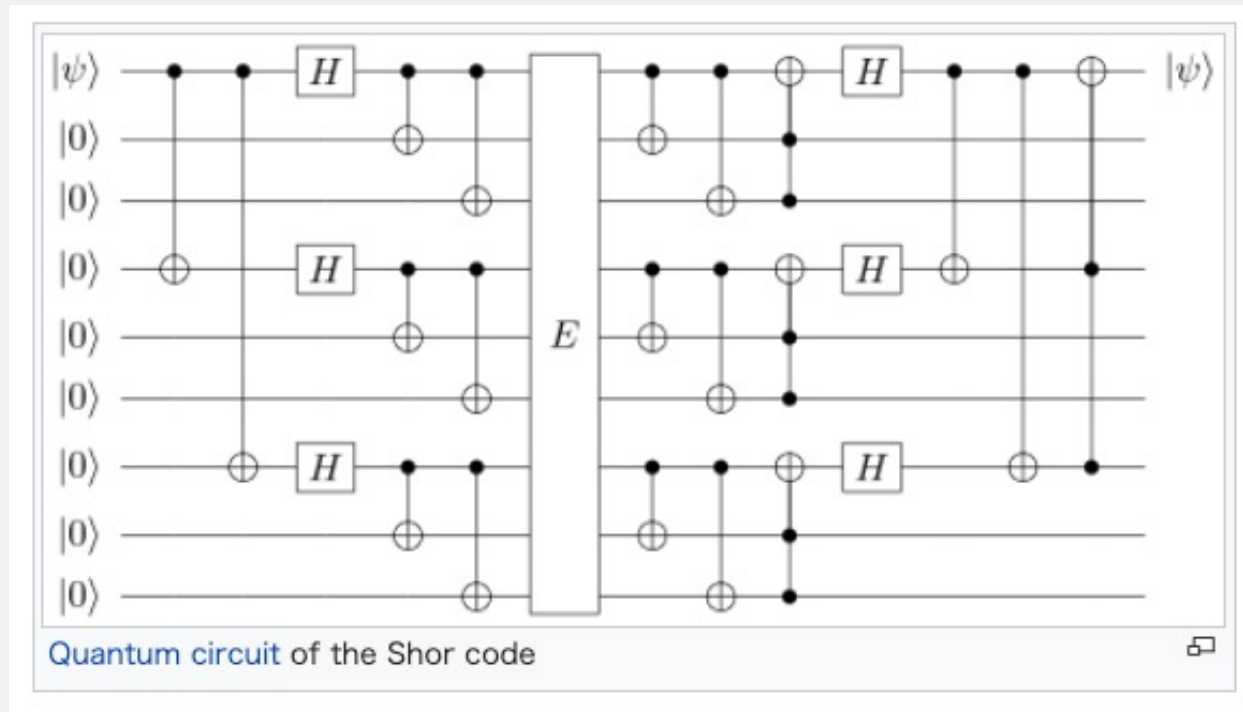
➡ 量子誤り訂正 が必要

誤り耐性があり、スケーラブルかつ実装可能なアーキテクチャの探索が必要

量子誤り訂正 (Quantum Error Correction)

- 複数の量子ビットで冗長化、1論理量子ビットを表現
- エラーを検知して正しい状態に復元する機能

Shor's 9 qubit code ['95]



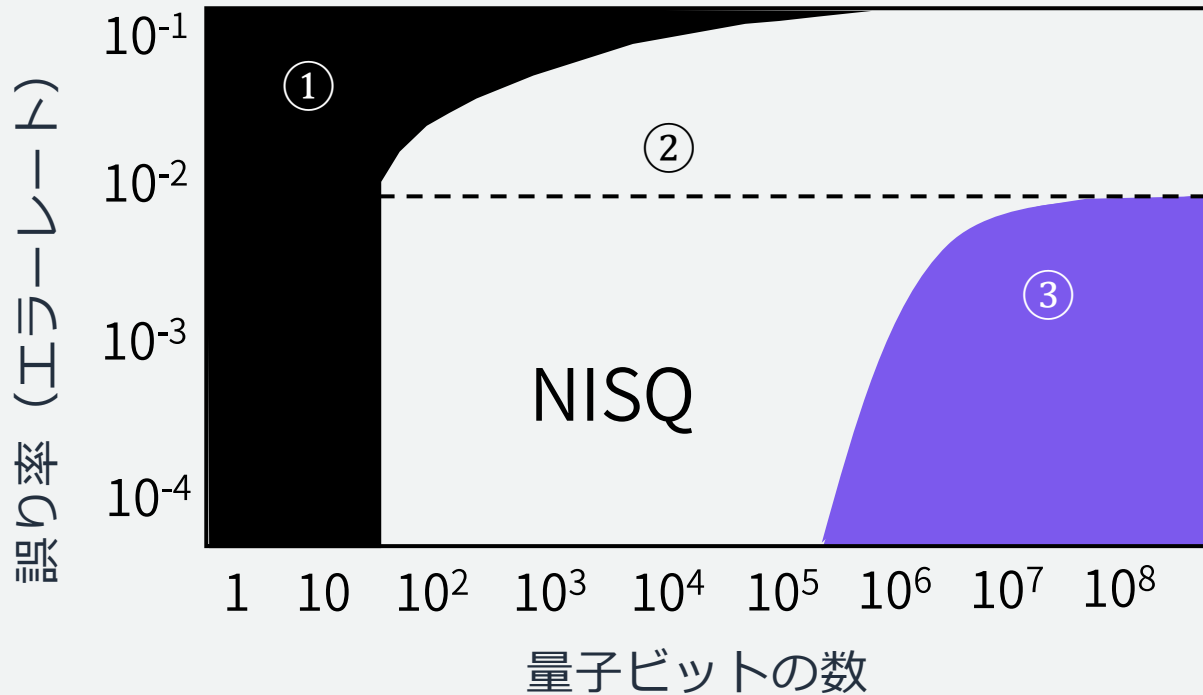
- 9量子ビットで1論理量子ビットをエンコード
- 符号・位相の両エラーを訂正できる

$$|0_s\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)$$

$$|1_s\rangle = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)$$

NISQ : Noisy Intermediate-Scale Quantum Technology

- ノイズあり中規模量子デバイス -



- 1: 古典計算機でシミュレーション可能
- 2: Noisy Intermediate-Scale Quantum (NISQ)
- 3: エラー訂正を前提としたコンピュータ

① < 50 量子ビット:

古典計算機でシミュレーション可能

② NISQ

有望なアプリケーション実現が期待される

いましてすぐ量子コンピュータを使って開発はじめよう!

③ > 100k 量子ビット:

エラー訂正前提で量子計算が古典計算を凌駕

しかし大規模化開発には時間がかかる

NISQ アルゴリズム

NISQ (Noisy Intermediate Scale Quantum)

- 50~数百量子ビットを用いて、誤り訂正が実現するまでに有用な量子アプリケーションを探す
- 耐ノイズ性を考慮したノイズ耐性アルゴリズムを設計する必要がある
- 量子古典ハイブリッドソリューションでは、古典系で VQE, QAOA, Quantum Simulation etc.
- 少なくとも近い将来、量子コンピューターは特殊用途のデバイスになる可能性が高く、ほとんどのユーザーがクラウド経由でアクセスする
- NISQ だけで世界を変えることはない、近い将来の量子プラットフォームの目標は、将来のより高度な量子デバイスによって実現されるより大きな道を開くこと

Quantum Computing in the NISQ era and beyond

John Preskill

Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics,
California Institute of Technology, Pasadena CA 91125, USA

30 July 2018

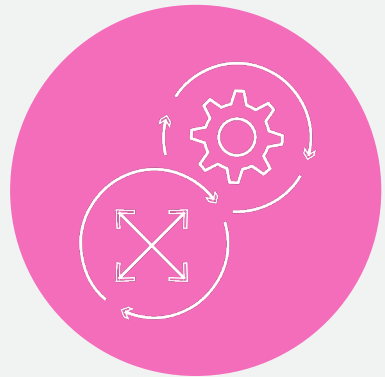
Noisy Intermediate-Scale Quantum (NISQ) technology will be available in the near future. Quantum computers with 50-100 qubits may be able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates will limit the size of quantum circuits that can be executed reliably. NISQ devices will be useful tools for exploring many-body quantum physics, and may have other useful applications, but the 100-qubit quantum computer will not change the world right away — we should regard it as a significant step toward the more powerful quantum technologies of the future. Quantum technologists should continue to strive for more accurate quantum gates and, eventually, fully fault-tolerant quantum computing.

Quantum. 2: 79. [doi:10.22331/q-2018-08-06-79](https://doi.org/10.22331/q-2018-08-06-79). [S2CID 44098998](https://arxiv.org/abs/1808.07352)

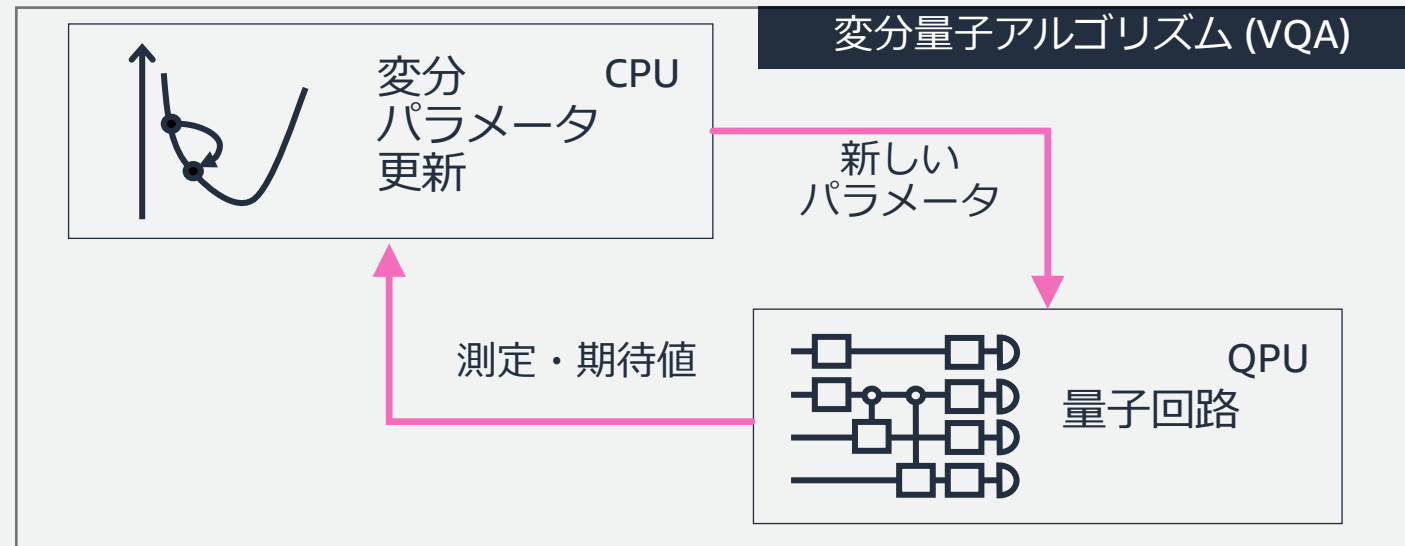
“NISQ is not likely to change the world all by itself. Instead, the primary goal for nearterm quantum platforms should be to pave the way for bigger payoffs which will be realized by more advanced quantum devices down the road.”

NISQ デバイスでの量子・古典ハイブリッド計算

- Noisy Intermediate Scale Quantum (NISQ)
- 限られた量子ビット (50~数百) を用いて現実的な問題を解く
- 量子コンピュータをコプロセッサとして利用する量子・古典ハイブリッド計算
 - 変分量子固有値法 (Variational Quantum Eigensolver: VQE)
 - QAOA (Quantum Approximate Optimization Algorithm)
 - 量子機械学習 (Quantum Machine Learning)



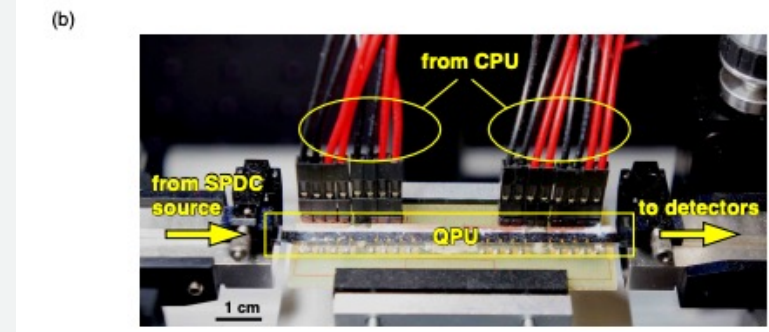
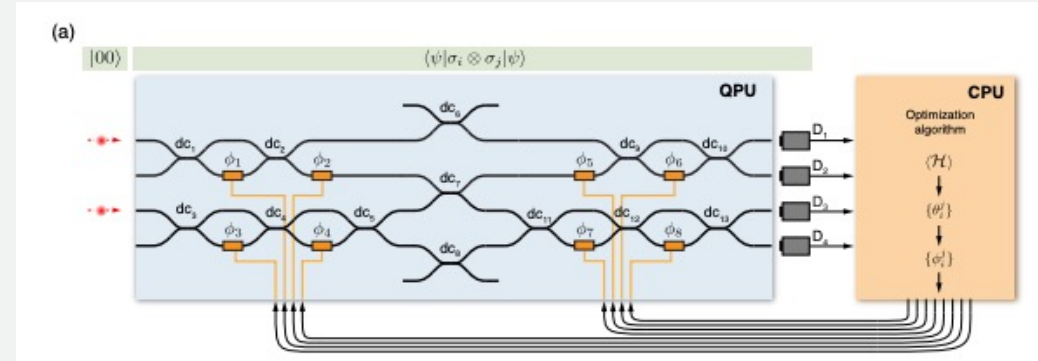
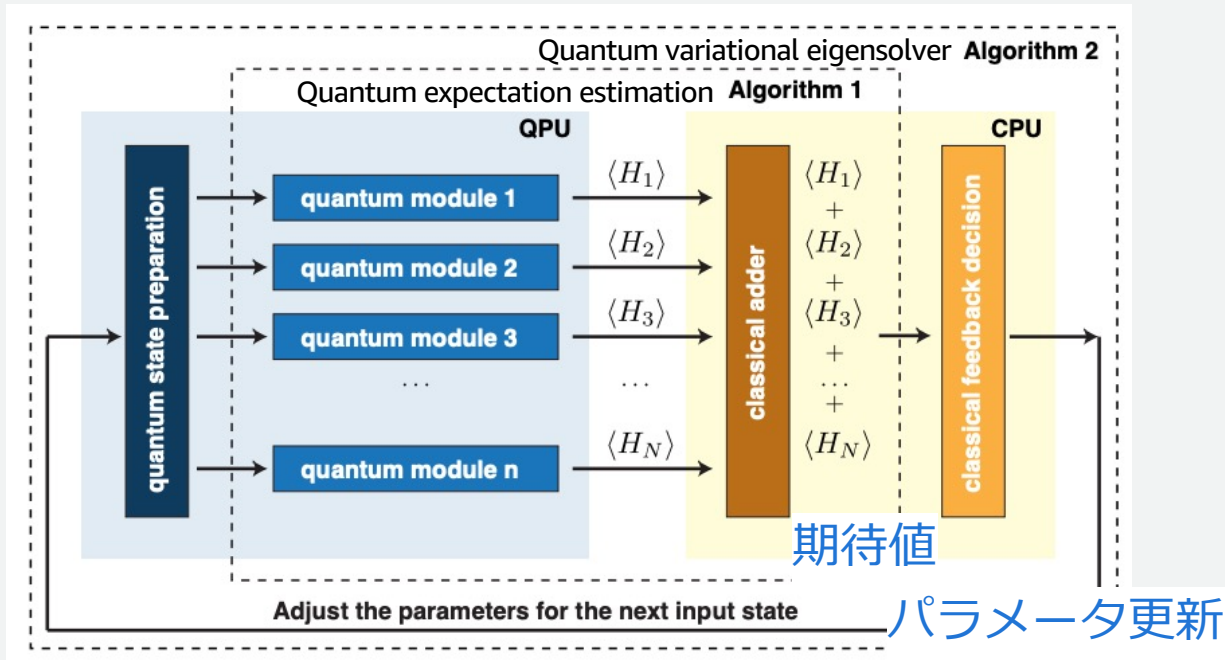
Hybrid
algorithms



VQE(Variational Quantum Eigensolver)

変分量子固有値ソルバーアルゴリズム

量子化学, 量子シミュレーション, 最適化のための量子古典ハイブリッド計算



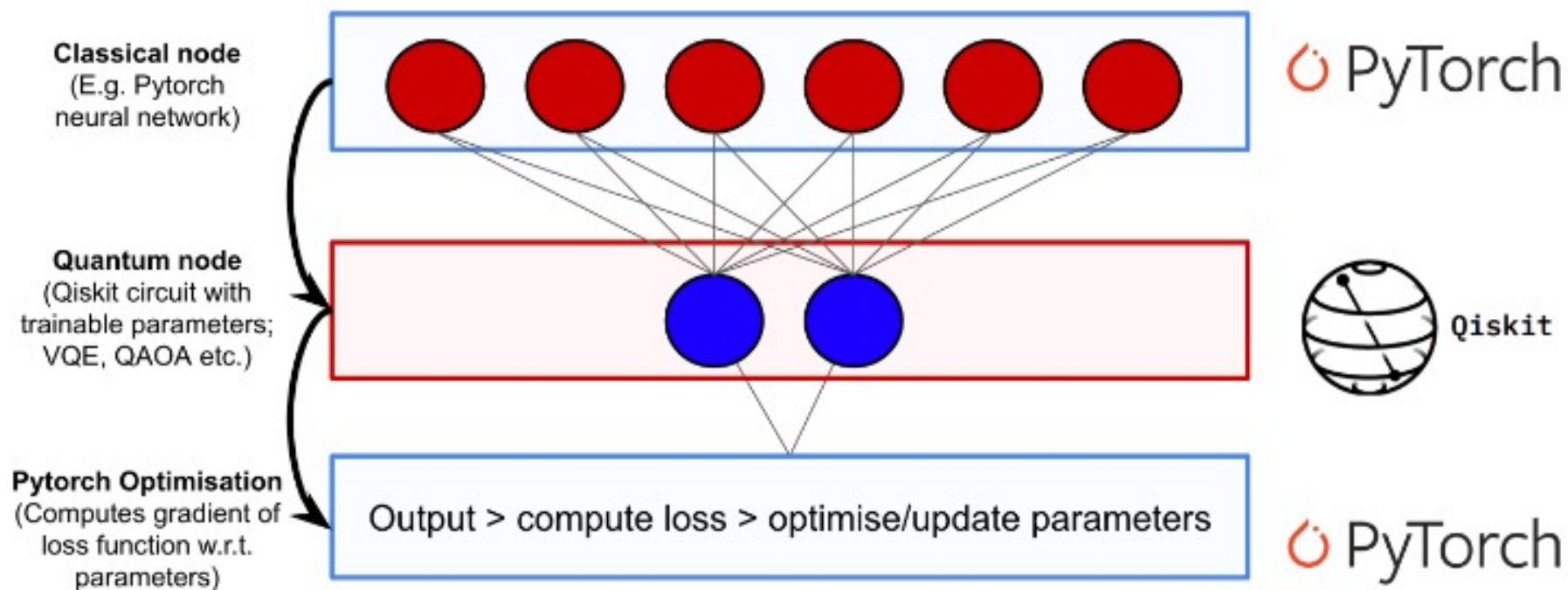
1. 量子状態の準備
2. QPU 上で量子期待値推定 $|\psi(\vec{\phi})\rangle = U(\vec{\phi})|0^n\rangle$
3. CPU 上でパラメータの決定・更新 $\min \langle \psi(\vec{\phi}) | H | \psi(\vec{\phi}) \rangle$

- 量子状態準備
- 測定により量子期待値推定 $\langle \psi | \sigma_i \otimes \sigma_j | \psi \rangle$ を実行

[A. Peruzzo et. al., arXiv:1304.3061v1 \[quant-ph\] 10 Apr 2013](#)

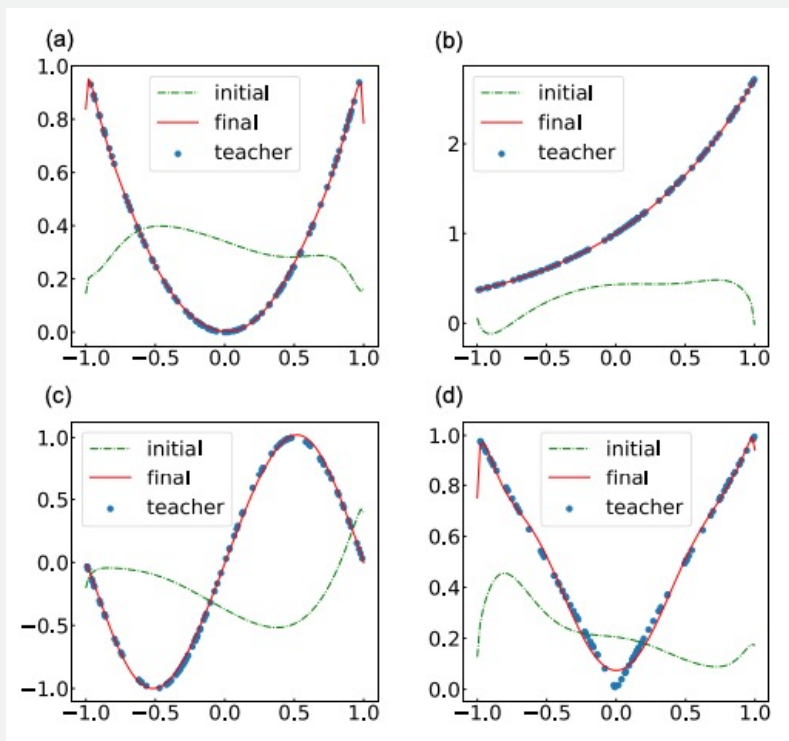
量子機械学習

1. どのように動作するのか?



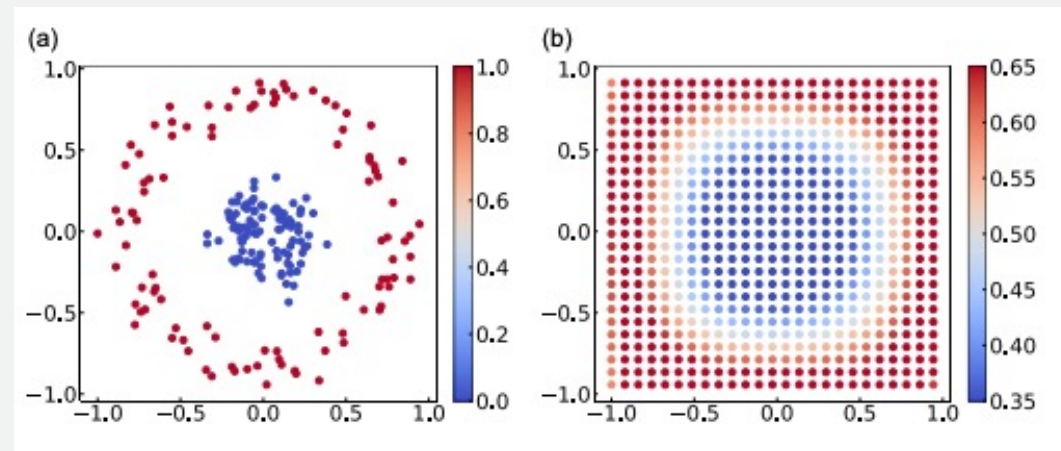
<https://qiskit.org/textbook/ja/ch-machine-learning/machine-learning-qiskit-pytorch.html>

NISQ 型量子回路学習



非線形関数の学習 (汎化)

- 変分量子回路を用いた教師あり学習
- テンソル積構造による非線型性とユニタリ変換
- 量子回路の解析微分による勾配の計算



二値文類問題の学習

[K. Mitarai et al., Phys. Rev. A 98, 032309 \(2018\)](#)

VQE, QML に便利なライブラリ

量子機械学習等の古典最適化、量子化学計算、量子機械学習などが

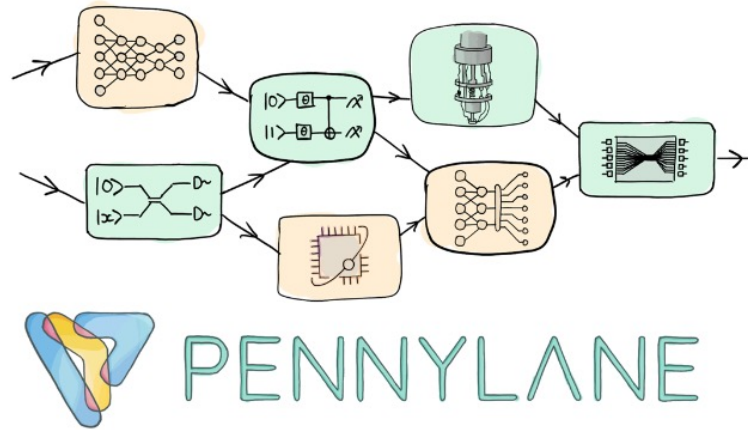
```
import pennylane as qml
from pennylane import numpy as np

# create a quantum device
dev1 = qml.device('default.qubit', wires=1)

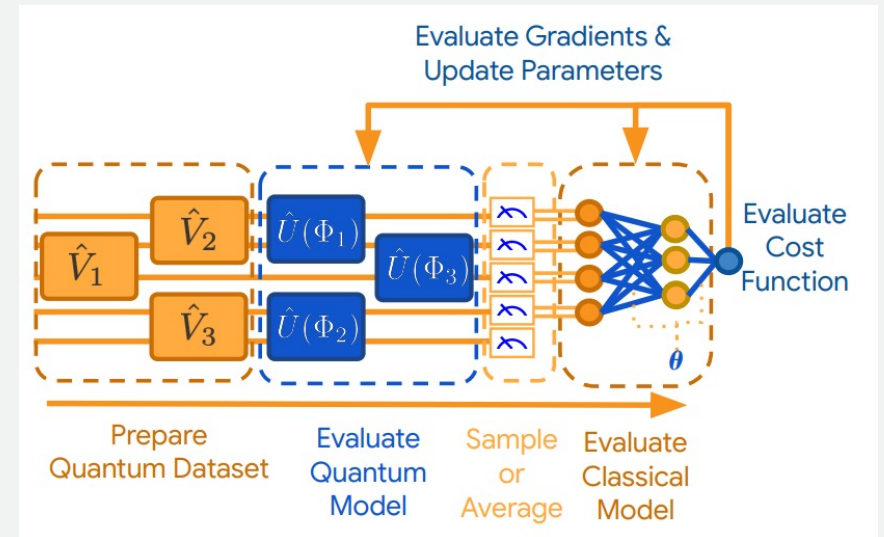
@qml.qnode(dev1)
def circuit(phi1, phi2):
    # a quantum node
    qml.RX(phi1, wires=0)
    qml.RY(phi2, wires=0)
    return qml.expval(qml.PauliZ(0))

def cost(x, y):
    # classical processing
    return np.sin(np.abs(circuit(x, y))) - 1

# calculate the gradient
dcost = qml.grad(cost, argnum=[0, 1])
```



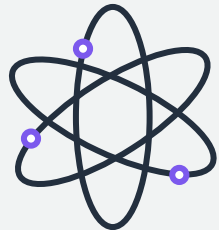
<https://pennylane.ai/>



[arXiv:2003.02989v2 \[quant-ph\]](https://arxiv.org/abs/2003.02989v2) 26 Aug 2021

AWS における量子コンピューティングの 取り組み全体像

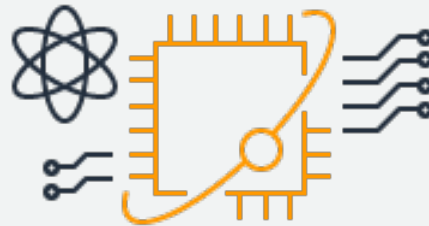
AWS Center for Quantum Computing



Push the Boundaries

アルゴリズムと
ハードウェアの研究

Amazon Braket



Democratize Quantum
Computing

アプリケーション探索と
実験を容易にするサービス

Amazon Quantum Solutions Lab



Provide Expert Guidance

実践的・学際的な
サポートとコラボレーション

AWS Center For Quantum Computing



AWS Center for Quantum Computing

- カリフォルニア工科大学（Caltech） に隣接する **AWS の量子研究センター**
- 世界をリードする量子コンピューティングの研究者とエンジニアを集めて、量子コンピューティングの**ハードウェア**と**ソフトウェア**の開発を加速

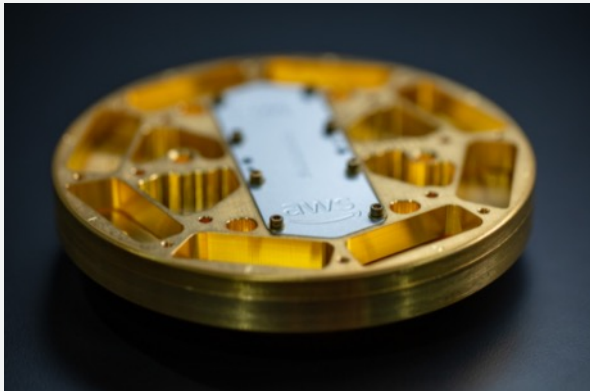


Pasadena の
Caltech キャンパスに隣接

<https://aws.amazon.com/jp/blogs/quantum-computing/announcing-the-opening-of-the-aws-center-for-quantum-computing/>

AWS Center for Quantum Computing

- Oskar Painter, Fernando G.S.L. Brandão, John Preskill ら、世界をリードする量子コンピューティングの研究者とエンジニアを集めて量子コンピューティングのハードウェアとソフトウェアの開発を加速



マイクロ波パッケージの
AWS量子プロセッサ



超伝導量子ビット

Quantum Computing in the NISQ era and beyond

John Preskill

Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics,
California Institute of Technology, Pasadena CA 91125, USA
30 July 2018

Noisy Intermediate-Scale Quantum (NISQ) technology will be available in the near future. Quantum computers with 50-100 qubits may be able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates will limit the size of quantum circuits that can be executed reliably. NISQ devices will be useful tools for exploring many-body quantum physics, and may have other useful applications, but the 100-qubit quantum computer will not change the world right away — we should regard it as a significant step toward the more powerful quantum technologies of the future. Quantum technologists should continue to strive for more accurate quantum gates and, eventually, fully fault-tolerant quantum computing.

<https://arxiv.org/pdf/1801.00862.pdf>

AWS Center for Quantum Computing

- Oskar Painter, Fernando G.S.L. Brandão, John Preskill ら、
世界をリードする量子コンピューティングの研究者とエンジニアを集めて
量子コンピューティングのハードウェアとソフトウェアの開発を加速

Quantum Computing in the NISQ era and beyond

John Preskill

Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics,
California Institute of Technology, Pasadena CA 91125, USA
30 July 2018

Noisy Intermediate-Scale Quantum (NISQ) technology will be available in the near future. Quantum computers with 50-100 qubits may be able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates will limit the size of quantum circuits that can be executed reliably. NISQ devices will be useful tools for exploring many-body quantum physics, and may have other useful applications, but the 100-qubit quantum computer will not change the world right away — we should regard it as a significant step toward the more powerful quantum technologies of the future. Quantum technologists should continue to strive for more accurate quantum gates and, eventually, fully fault-tolerant quantum computing.

<https://arxiv.org/pdf/1801.00862.pdf>

Building a fault-tolerant quantum computer using concatenated cat codes

Christopher Chamberland,^{1,2} Kyungjoo Noh,¹ Patricio Arrangoiz-Arriola,^{1,*}
Earl T. Campbell,^{1,*} Connor T. Hann,^{1,3,*} Joseph Iverson,^{1,*} Harald Putterman,^{1,*} Thomas
C. Bohdanowicz,^{1,2} Steven T. Flammia,¹ Andrew Keller,¹ Gil Refael,^{1,2} John Preskill,^{1,2}
Liang Jiang,^{1,4} Amir H. Safavi-Naeini,^{1,5} Oskar Painter,^{1,2} and Fernando G.S.L. Brandão^{1,2}

¹AWS Center for Quantum Computing, Pasadena, CA 91125, USA

²IQIM, California Institute of Technology, Pasadena, CA 91125, USA

³Department of Physics, Yale University, New Haven, CT 06511, USA

⁴Pritzker School of Molecular Engineering, The University of Chicago, Illinois 60637, USA

⁵Department of Applied Physics and Ginzton Laboratory, Stanford University, Stanford, CA 94305, USA

We present a comprehensive architectural analysis for a fault-tolerant quantum computer based on cat codes concatenated with outer quantum error-correcting codes. For the physical hardware, we propose a system of acoustic resonators coupled to superconducting circuits with a two-dimensional layout. Using estimated near-term physical parameters for electro-acoustic systems, we perform a detailed error analysis of measurements and gates, including CNOT and Toffoli gates. Having built a realistic noise model, we numerically simulate quantum error correction when the outer code is either a repetition code or a thin rectangular surface code. Our next step toward universal fault-tolerant quantum computation is a protocol for fault-tolerant Toffoli magic state preparation that significantly improves upon the fidelity of physical Toffoli gates at very low qubit cost. To achieve even lower overheads, we devise a new magic-state distillation protocol for Toffoli states. Combining these results together, we obtain realistic full-resource estimates of the physical error rates and overheads needed to run useful fault-tolerant quantum algorithms. We find that with around 1,000 superconducting circuit components, one could construct a fault-tolerant quantum computer that can run circuits which are intractable for classical supercomputers. Hardware with 32,000 superconducting circuit components, in turn, could simulate the Hubbard model in a regime beyond the reach of classical computing.

int-ph] 7 Dec 2020

<https://arxiv.org/pdf/2012.04108.pdf>

“Building a fault-tolerant quantum computer using concatenated cat codes”

by Christopher Chamberland, Kyungjoo Noh, Patricio Arrangoiz-Arriola, Earl T. Campbell, Connor T. Hann, Joseph Iverson, Harald Putterman, Thomas C. Bohdanowicz, Steven T. Flammia, Andrew Keller, Gil Refael, John Preskill, Liang Jiang, Amir H. Safavi-Naeini, Oskar Painter, Fernando G.S.L. Brandão

[arXiv:2012.04108](https://arxiv.org/abs/2012.04108) [quant-ph]

<https://aws.amazon.com/blogs/quantum-computing/designing-a-fault-tolerant-quantum-computer-with-cat-qubits/>

Building a fault-tolerant quantum computer using concatenated cat codes

Christopher Chamberland,^{1,2} Kyungjoo Noh,¹ Patricio Arrangoiz-Arriola,^{1,*} Earl T. Campbell,^{1,*} Connor T. Hann,^{1,3,*} Joseph Iverson,^{1,*} Harald Putterman,^{1,*} Thomas C. Bohdanowicz,^{1,2} Steven T. Flammia,¹ Andrew Keller,¹ Gil Refael,^{1,2} John Preskill,^{1,2} Liang Jiang,^{1,4} Amir H. Safavi-Naeini,^{1,5} Oskar Painter,^{1,2} and Fernando G.S.L. Brandão^{1,2}

¹AWS Center for Quantum Computing, Pasadena, CA 91125, USA

²IQIM, California Institute of Technology, Pasadena, CA 91125, USA

³Department of Physics, Yale University, New Haven, CT 06511, USA

⁴Pritzker School of Molecular Engineering, The University of Chicago, Illinois 60637, USA

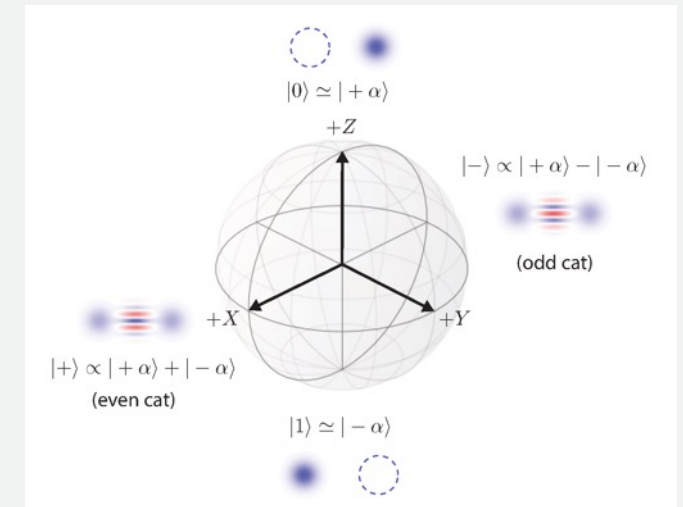
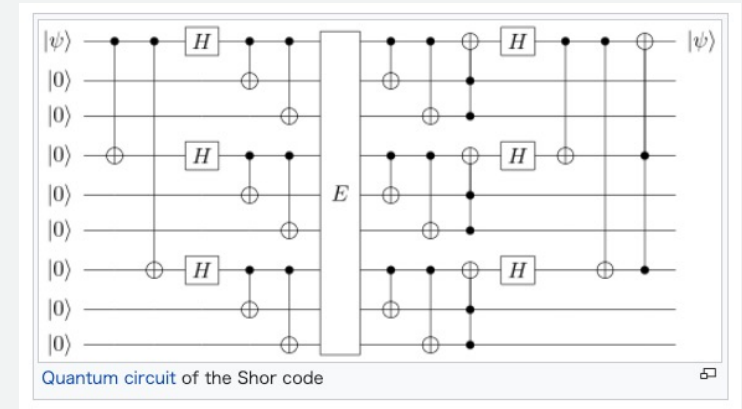
⁵Department of Applied Physics and Ginzton Laboratory, Stanford University, Stanford, CA 94305, USA

We present a comprehensive architectural analysis for a fault-tolerant quantum computer based on cat codes concatenated with outer quantum error-correcting codes. For the physical hardware, we propose a system of acoustic resonators coupled to superconducting circuits with a two-dimensional layout. Using estimated near-term physical parameters for electro-acoustic systems, we perform a detailed error analysis of measurements and gates, including CNOT and Toffoli gates. Having built a realistic noise model, we numerically simulate quantum error correction when the outer code is either a repetition code or a thin rectangular surface code. Our next step toward universal fault-tolerant quantum computation is a protocol for fault-tolerant Toffoli magic state preparation that significantly improves upon the fidelity of physical Toffoli gates at very low qubit cost. To achieve even lower overheads, we devise a new magic-state distillation protocol for Toffoli states. Combining these results together, we obtain realistic full-resource estimates of the physical error rates and overheads needed to run useful fault-tolerant quantum algorithms. We find that with around 1,000 superconducting circuit components, one could construct a fault-tolerant quantum computer that can run circuits which are intractable for classical supercomputers. Hardware with 32,000 superconducting circuit components, in turn, could simulate the Hubbard model in a regime beyond the reach of classical computing.

quant-ph] 7 Dec 2020

Quantum Error Correction (QEC)

- Active QEC
 - 量子ビットを冗長化し、論理量子ビットを表現
 - エラーレートを十分低く抑えられるが、その分オーバーヘッドが発生
 - bit-flip エラーと phase-flip エラーの両方を同時に抑えようとすると、多くの物理量子ビットを消費
- Passive QEC
 - エラーに対して安定な状態を論理量子ビットとして採用
 - 本質的にエラーが起こりづらい状況をつくる
 - 例: Schrödinger cat qubit や Gottesman-Kitaev-Preskill (GKP) qubit



Active + Passive QEC の組み合わせでより少ない量子ビットによる誤り耐性符号を提案

Announcing the AWS Center for Quantum Networking

by Denis Sukachev and Mihir Bhaskar | on 21 JUN 2022 | in [Amazon Quantum Solutions Lab](#), [Announcements](#), [AWS Center For Quantum Computing](#), [Quantum Technologies](#) | [Permalink](#) | [Share](#)

Over the last decade, governments and technology companies have invested heavily in research and development of *quantum computers* that have the potential to [revolutionize](#) science and technology. While there is still a long way ahead, these investments have already transformed quantum computers: They have evolved from delicate laboratory systems accessible to only a few research institutions to increasingly reliable and powerful commercial machines available to researchers, developers, and even quantum enthusiasts worldwide via cloud services like [Amazon Braket](#).

While quantum computing continues to be a major area of investment and progress for academic and industry researchers alike, it is only one component of a broader class of quantum technologies. To unlock the full potential of quantum devices, they need to be connected together into a *quantum network*, similar to the way today's devices are connected via the internet. Despite not receiving the same level of attention as quantum computers, quantum networks have fascinating possible applications. One of them is enabling global communications protected by quantum key distribution with privacy and security levels not achievable using conventional encryption techniques. Quantum networks will also provide powerful and secure cloud quantum servers by connecting together and amplifying the capabilities of individual quantum processors.

Quantum networks will leverage some technologies already deployed for modern optical communications, such as lasers, fibers, and detectors. However, instead of strong laser beams, quantum networks require single photons — smallest building blocks of light — to connect quantum devices together. Single photons enable many of the special capabilities of quantum networks and simultaneously pose the great challenge: quantum mechanics prohibits their amplification limiting the network range. Also, the weakness of single photons complicates interfacing them with today's quantum computing devices. This means that special new technologies, such as quantum repeaters and transducers, will need to be developed in

Announcing a research alliance between AWS and Harvard University

by Antia Lamas-Linares, Mihir Bhaskar, and Denis Sukachev | on 12 SEP 2022 | in [Announcements](#), [Quantum Technologies](#) | [Permalink](#) | [Share](#)

Quantum science is progressing rapidly: in recent decades it has emerged from textbooks to enable tools we use in everyday life, such as lasers and GPS. Amazon, which is already involved in [quantum computing](#), recently announced the creation of the AWS Center for Quantum Networking. That effort will focus on a more overlooked facet of quantum technology: networking and communication, aiming to connect quantum computers into a network, like the internet. Quantum networks could enable new technologies such as secure communication channels and powerful quantum computing clusters. With this new technology, there are both engineering and scientific challenges to be resolved on the way to building global quantum networks.

Today, [AWS is announcing a research alliance with Harvard University](#) to address the fundamental scientific challenges associated with building quantum networks. For more than 20 years, Harvard researchers have been on the leading edge of quantum technology, with a diverse research portfolio in solid-state physics, nanofabrication, quantum optics, and quantum computing and communications. Harvard researchers have been at the forefront of solving important technical problems in quantum networking. [In 2020, scientists from the Harvard Quantum Initiative performed a groundbreaking demonstration](#) increasing the usable range of a quantum network beyond the standard transmission limit using a [quantum memory](#). In addition to the cutting-edge research, Harvard has a nanofabrication facility at the [Center for Nanoscale Systems](#), which is available for industrial partners, and forms an ecosystem for many successful high-tech start-ups led by Harvard researchers.

[The AWS Center for Quantum Networking is located in the Boston area](#), allowing for cooperation between scientists at AWS and Harvard. This new AWS research alliance will further bolster scientific research and support education of a diverse group of quantum scientists and engineers by providing funding for advanced quantum networking research, engineering, and fabrication. As noted in President Biden's recent [directives](#), these advances in quantum technology will not be possible without qualified researchers.

To support the development of a diverse quantum workforce, AWS established the "AWS Generation Q Fund at the Harvard Quantum Initiative." Generation Q will enable the Harvard Quantum Initiative to offer new fellowship programs for post-baccalaureates, graduate students, and postdocs to train the next-generation of quantum scientists and engineers. [Mikhail Lukin](#), a Harvard Physics Professor and the co-director of the Harvard Quantum Initiative, underscores the importance of the Generation Q Fund: ["There is a shortage of qualified quantum-educated workforce, and it's not just physicists but engineers and even people involved in running these businesses. We're in a unique position to contribute: Essentially, all major quantum research centers in the US and abroad have several faculty members and group leaders who have been educated at Harvard."](#)

This alliance will add a new chapter to a successful story of collaboration between Amazon and leading research institutions and [academics](#), which spans the fields from machine learning and artificial intelligence through health science and quantum physics.

量子コンピュータを使った研究開発

お客様からよくいただく質問

- 量子コンピューティングはどんな分野に使える？
- 既存のソリューションや機械学習を凌駕？すぐに置き換えられますか？
- 量子コンピューティングの実力は？課題は？
- クラウドで量子コンピューティング、どうやって？何が嬉しい？安全？
- どんな人が Amazon Braket を使っている？ユースケースは？
- ハードルが高そう、どうすれば量子ソフトウェア開発を始められる？

Amazon Web Services クラウドコンピューティング 200 を超える幅広いサービスであらゆるワークロードをサポート

AWSの提供する90%のサービスや機能はお客様からの意見をもとに開発、残りの10%もお客様の潜在的な要望を汲み取って作られています



コンピューティング



機械学習



アプリケーション統合



モバイル



IoT



Game Tech



ARとVR



ロボット工学



量子テクノロジー



エンドユーザー
コンピューティング



ビジネスアプリケーション



カスタマーエンゲージメント



ストレージ



メディアサービス



移行と転送



データベース



分析



ブロックチェーン



ネットワークとコンテンツ配信



マネジメントとガバナンス



セキュリティ, ID, コンプライアンス



AWS コスト管理



開発者用ツール



人工衛星

日本で数十万のお客様※ (世界で数百万以上)



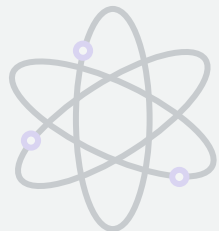
aws
※ お客様とはアクティブカスタマー数を指します。アクティブカスタマーとは、AWSクラウド無料利用枠を含むAWSアカウントの先月の使用状況のあるアマゾン会員でない対象アカウントです。
© 2022, Amazon Web Services, Inc. or its affiliates.

Amazon Braket

フルマネージド型の量子コンピューティングサービス

AWS における量子コンピューティングの 取り組み全体像

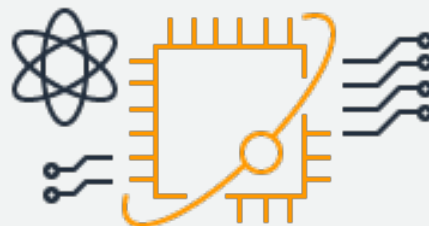
AWS Center for Quantum Computing



Push the Boundaries

アルゴリズムと
ハードウェアの研究

Amazon Braket



Democratize Quantum
Computing

アプリケーション探索と
実験を容易にするサービス

Amazon Quantum Solutions Lab

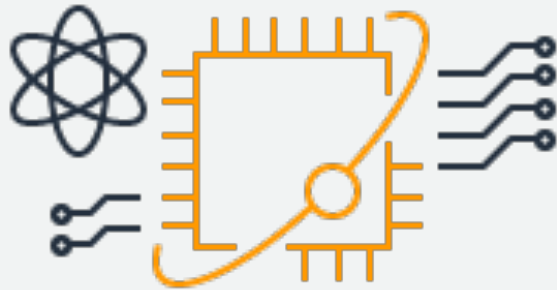


Provide Expert Guidance

実践的・学際的な
サポートとコラボレーション

量子コンピュータの敷居の高さ

量子コンピュータごとに
断片化した開発者ツール



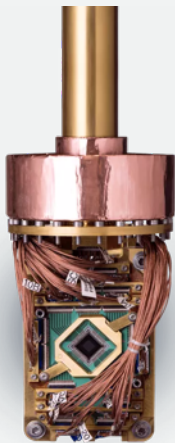
量子ハードウェアは
貴重な計算資源

ハードごとに個別契約が必要
アクセスが難しい

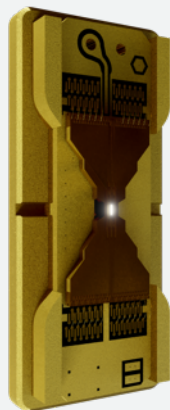
量子コンピュータのシミュレーションは
専門知識を有する上に
計算リソースが必要

Amazon Braket を使うメリット AWS クラウド量子コンピューティング

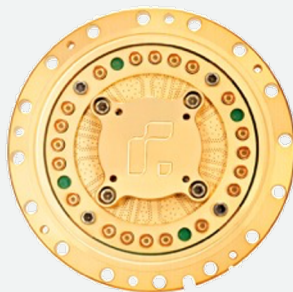
D:wave
The Quantum Computing Company™



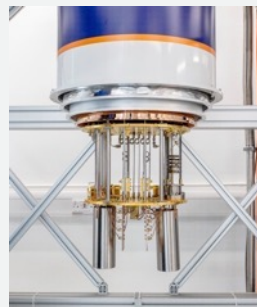
IONQ



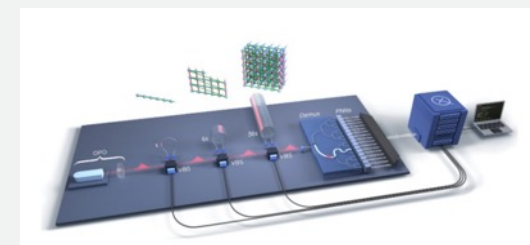
rigetti



OQC



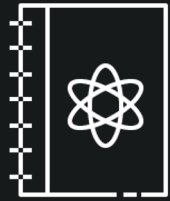
XANADU



- さまざまな量子ハードウェアを利用できる（QPU は随時アップデート）
- AWS のアカウントを取得すれば誰でも今すぐ量子ソフトウェア開発をスタート
- 初期費用なし、オンデマンド・従量課金のためスモールスタートできる
- AWS の各種サービスとのシームレスな連携
- Amazon Braket SDK やライブラリ Hybrid Jobs などのデベロッパーツール

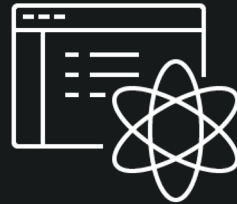
Amazon Braket

- 全ての開発者・科学者の手に量子コンピューティングを



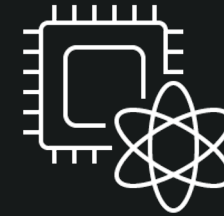
設計

マネージド開発環境



テスト

ハイパフォーマンスな
回路シミュレータ



実行

セキュア・オンデマンドな
量子ハードウェアで
ハイブリッド計算

量子デバイス (QPU)

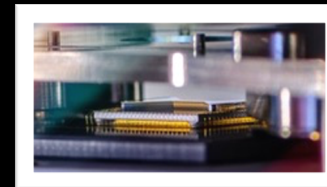
- ゲートベース量子コンピュータ
 - Rigetti : 超伝導量子ビット (The Rigetti Aspen-M1)
 - 80 量子ビット、結合は部分的
 - IonQ : イオントラップ量子ビット (IonQ linear trap)
 - 11 量子ビット、全結合が特徴
 - OQC : 超伝導量子ビット (Lucy)
 - 8 量子ビット
- 光量子コンピュータ
 - Xanadu : 光量子 (Borealis)
 - 216量子モード、ガウシアンボゾンサンプラー (GBS)
- 量子アニーリング
 - D-Wave : 超伝導量子ビット
 - DW_2000Q_6 (2048量子ビット)
 - Advantage_system4, Advantage_system6 (5760量子ビット)



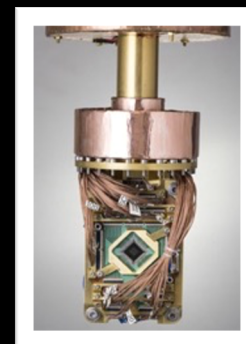
© 2022, Amazon Web Services, Inc. or its affiliates.



rigetti



IONQ

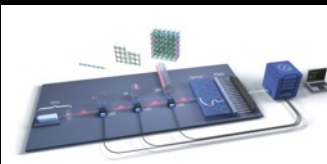


OQC



XANADU

D:wave

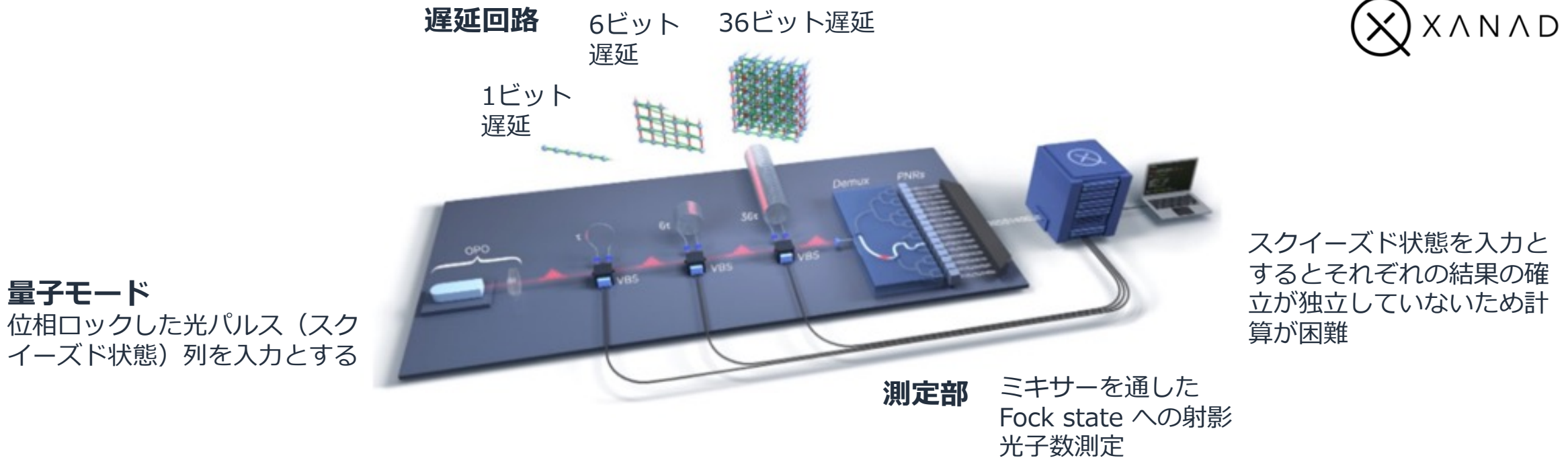


デバイスの開発・構築実行は非常に高度な技術を必要とする上、運用も高コスト。クラウドベースのオンデマンドに適している。

<https://aws.amazon.com/braket/hardware-providers/>

量子超越を実現した光量子計算を Amazon Braket で提供

- Xanadu 社 Borealis : 216 モードのスクイーズド光パルス
- プログラム可能なガウシアンボゾンサンプラー (GBS)
- 3次元のエンタングル状態を用いて、9000年以上かかる計算を 36 μ s で実現

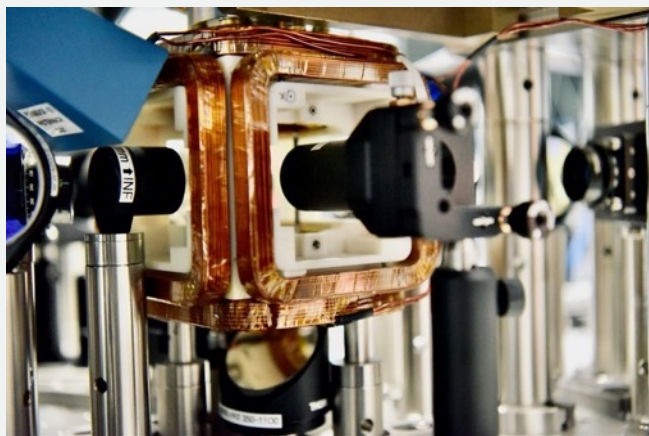


<https://www.nature.com/articles/s41586-022-04725-x>
<https://aws.amazon.com/jp/blogs/quantum-computing/explore-quantum-computational-advantage-with-xanadus-borealis-device-on-amazon-braket/>

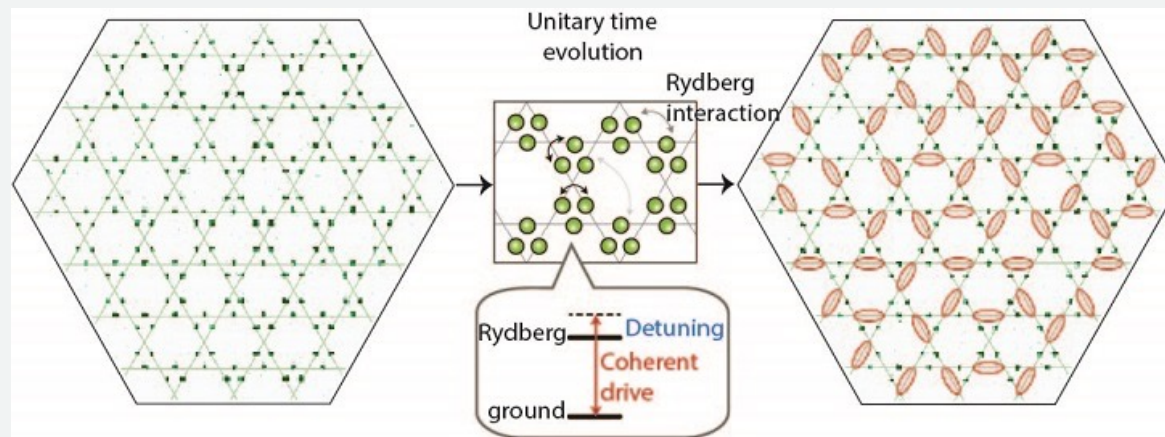
Amazon Braket に QuEra を追加予定

QuEra Computing

- Rydberg (リュードベリ) 原子によるアナログハミルトニアンシミュレーション (AHS)
- プログラマブルな量子デバイス (カゴメ格子上に配置した Rydberg 原子) を用いて、興味のある量子系 (量子スピン液体) を実験的に模擬し、そのふるまいを調べる [[Semeghini, et al., Science \(2021\), arXiv:2104.04119](#)]



QuEra Computing with Rydberg atom arrays



Physics of creating spin-pairs in a frustrated lattice

<https://aws.amazon.com/jp/blogs/quantum-computing/realizing-quantum-spin-liquid-phase-on-an-analog-hamiltonian-rydberg-simulator/>

開発環境と Amazon Braket Python SDK

```
'ZZ']

In [3]: bell = Circuit().h(0).cnot(0, 1)
print(bell)
print(f"\nserialized_circuit: {bell.to_ir().json()}")

T : |0|1|

q0 : -H-C-
      |
q1 : ---X-

T : |0|1|

serialized_circuit: {"instructions": [{"target": 0, "type": "h"}, {"control": 0, "target": 1, "type": "cnot"}]}

In [4]: result = simulator.run(bell, s3_destination_folder).result()
print(f"measurement_counts: {result.measurement_counts}")
print(f"measurement_probabilities: {result.measurement_probabilities}")

data = ["".join([str(bit) for bit in shot]) for shot in result.measurements]
plot = plt.hist(data)

measurement_counts: Counter({'00': 50, '11': 50})
measurement_probabilities: {'00': 0.5, '11': 0.5}

In [5]: # QFT example. Encode a circuit with phase frequenc
```



- マネージドな開発環境を数クリックで立ち上げ
- Jupyter 環境で量子アルゴリズムの構築・テスト・実行
- 事前構築された開発環境（サンプルノートブック）
- Amazon Braket SDK を通じてデバイスに依存しない量子回路設計が可能
 - ゲート型: Rigetti, IonQ, OQC, アニーリング: D-Wave
 - シミュレータ: SV1, TN1, DM1
 - 量子回路計算とハイブリッドジョブ

デバイスタイプ切り替えで簡単にQPUにアクセスできる

```
import boto3
from braket.circuits import Circuit
from braket.aws import AwsDevice

aws_account_id = boto3.client("sts").get_caller_identity()["Account"]

device = AwsDevice("arn:aws:braket:::device/qpu/rigetti/Aspen-8")
s3_folder = (f"amazon-braket-output-{aws_account_id}", "RIGETTI")

bell = Circuit().h(0).cnot(0, 1)
task = device.run(bell, s3_folder)
print(task.result().measurement_counts)
```

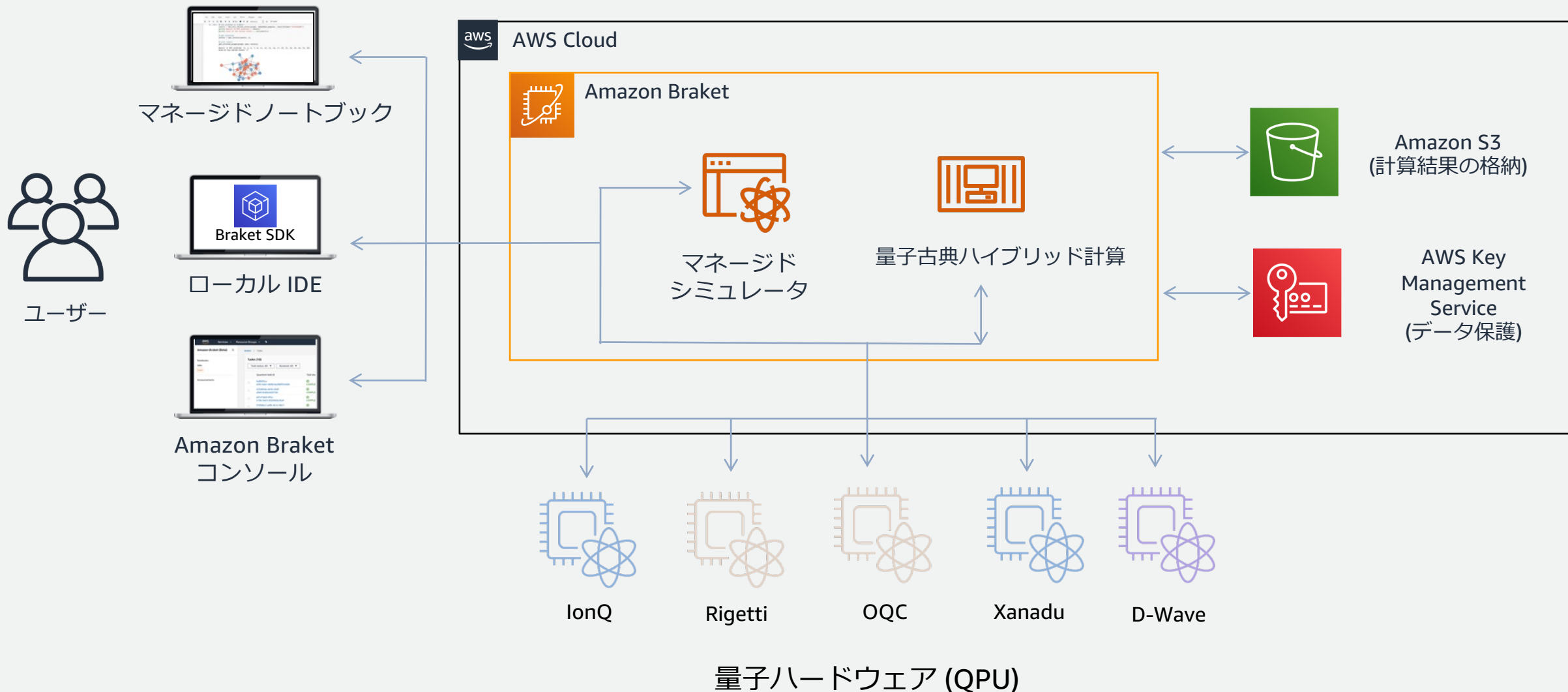
IonQ "arn:aws:braket:::device/qpu/ionq/ionQdevice"

Rigetti "arn:aws:braket:::device/qpu/rigetti/Aspen-8"

D-Wave (See the next section in this document for more information about using D-Wave.)

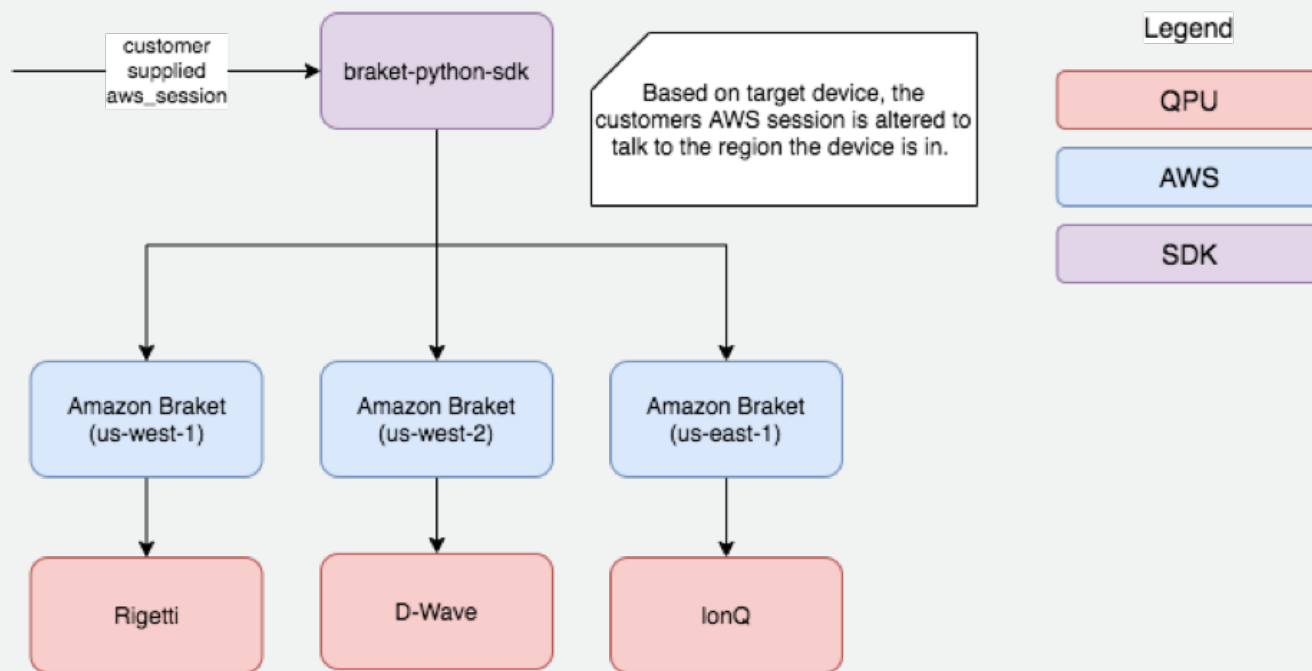
- **D-Wave 2000Q_6** "arn:aws:braket:::device/qpu/d-wave/DW_2000Q_6"
- **D-Wave Advantage_system1** "arn:aws:braket:::device/qpu/d-wave/Advantage_system1"

Amazon Braket アーキテクチャ



Amazon Braket で利用可能なリージョン

Amazon Braket で利用可能なリージョン			
Region Name	Region	Braket Endpoint	QPU
US East (N. Virginia)	us-east-1	braket.us-east-1.amazonaws.com	IonQ
US East (N. Virginia)	us-east-1	braket.us-east-1.amazonaws.com	Xanadu
US West (N. California)	us-west-1	braket.us-west-1.amazonaws.com	Rigetti
US West (Oregon)	us-west-2	braket.us-west-2.amazonaws.com	D-Wave
EU West 2 (London)	eu-west-2	braket.us-west-2.amazonaws.com	OQC



Amazon Braket SDK で任意のリージョンで実行された量子タスクは自動的に正しいリージョンにセッションを作成、ルーティングされる

Amazon Braket からアクセス可能な QPU とシミュレータ

Quantum Processing Units (QPUs) Hide retired devices

QPU Name	Qubits	Status	Region	Next available
D-Wave — Advantage_system6.1	5760	ONLINE	us-west-2	AVAILABLE NOW
D-Wave — Advantage_system4.1	5760	ONLINE	us-west-2	AVAILABLE NOW
Xanadu — Borealis	216 Qumodes	ONLINE	us-east-1	13:29:50
D-Wave — DW_2000Q_6	2048	ONLINE	us-west-2	AVAILABLE NOW
IonQ	11	ONLINE	us-east-1	AVAILABLE NOW
Rigetti — Aspen-11	38	OFFLINE	us-west-1	UNAVAILABLE
Rigetti — Aspen-M-1	80	ONLINE	us-west-1	13:29:50
Oxford Quantum Circuits — Lucy	8	ONLINE	eu-west-2	07:29:50

Simulators

Simulator Name	Qubits	Status	Region	Next available
Amazon Web Services — SV1	34	ONLINE	us-east-1, us-west-1, us-west-2	AVAILABLE NOW
Amazon Web Services — TN1	50	ONLINE	us-east-1, us-west-2	AVAILABLE NOW
Amazon Web Services — DM1	17	ONLINE	us-east-1, us-west-1, us-west-2	AVAILABLE NOW

QPU: IonQ: 11 Qubits, Universal gate model, イオントラップ
Rigetti: 80 Qubits, Universal gate model, 超伝導
Xanadu: 216 Qumode 光プロセッサ
D-Wave: 5760 & 2048 Qubits, 量子アニーリング

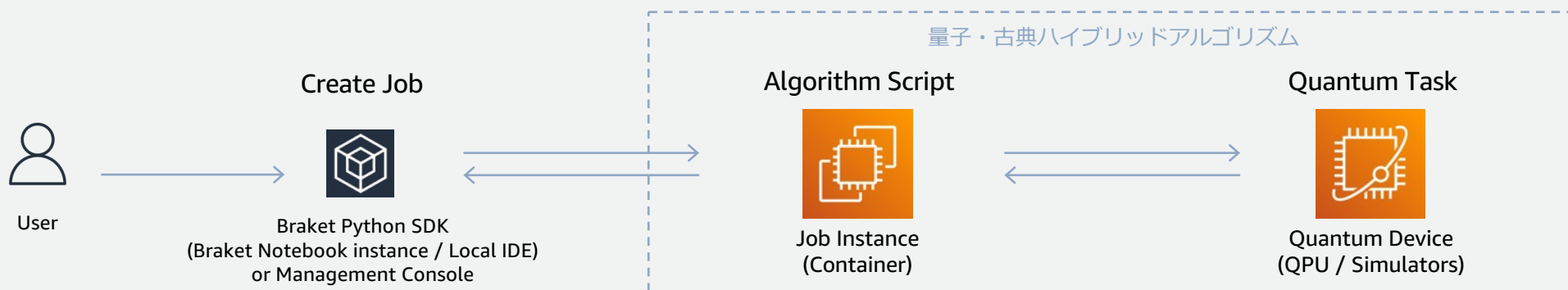
シミュレータ
SV1: 34 Qubit
DM1: 17 Qubit
TN1: 50 Qubit



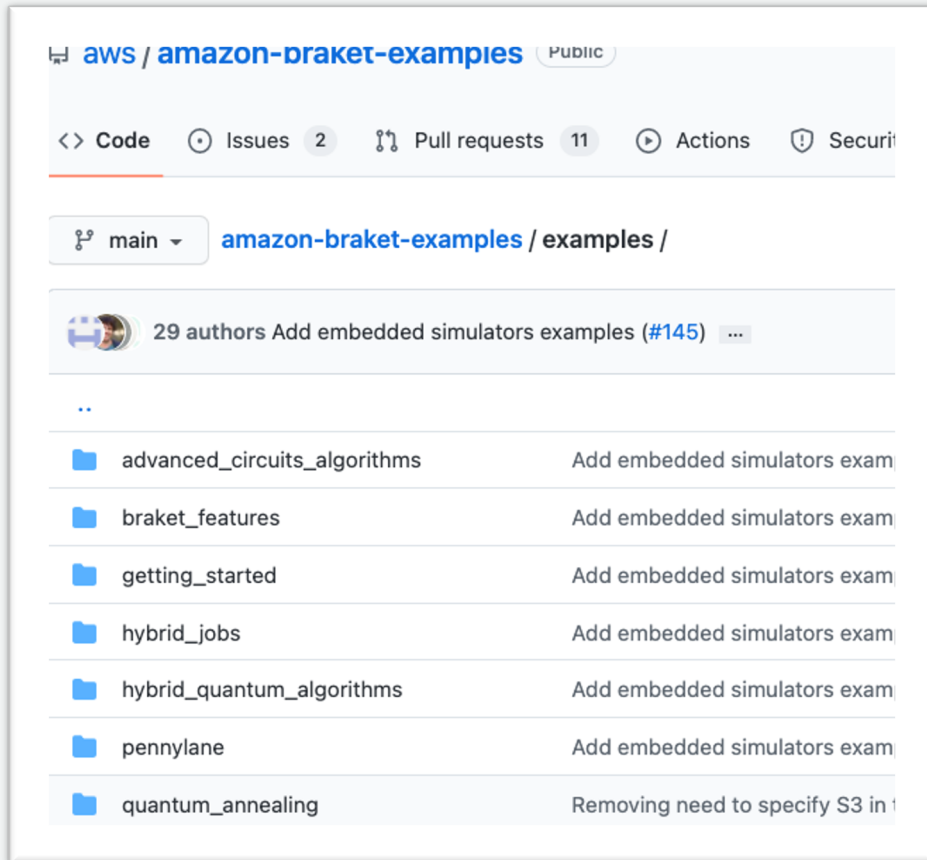
Amazon Braket Hybrid Jobs

フルマネージドな量子・古典ハイブリッドアルゴリズムの実行

- Variational Quantum Eigensolver (VQE, 固有値計算) や Quantum Approximate Optimization Algorithm (QAOA, 組合せ最適化) などの量子・古典ハイブリッドアルゴリズムを簡単に実行
- アルゴリズムは Python (Amazon Braket SDK または PennyLane) で記述しビルド済みコンテナを利用、BYOC も可能。
- 量子・古典アルゴリズムの起動時にジョブ実行用コンテナが起動、実行が終わるとインスタンスを解放するよう設計、従量課金制
- 実験期間中は選択された QPU に優先的なアクセスが可能のため、より速く、予測可能な時間でハイブリッドアルゴリズムを実行



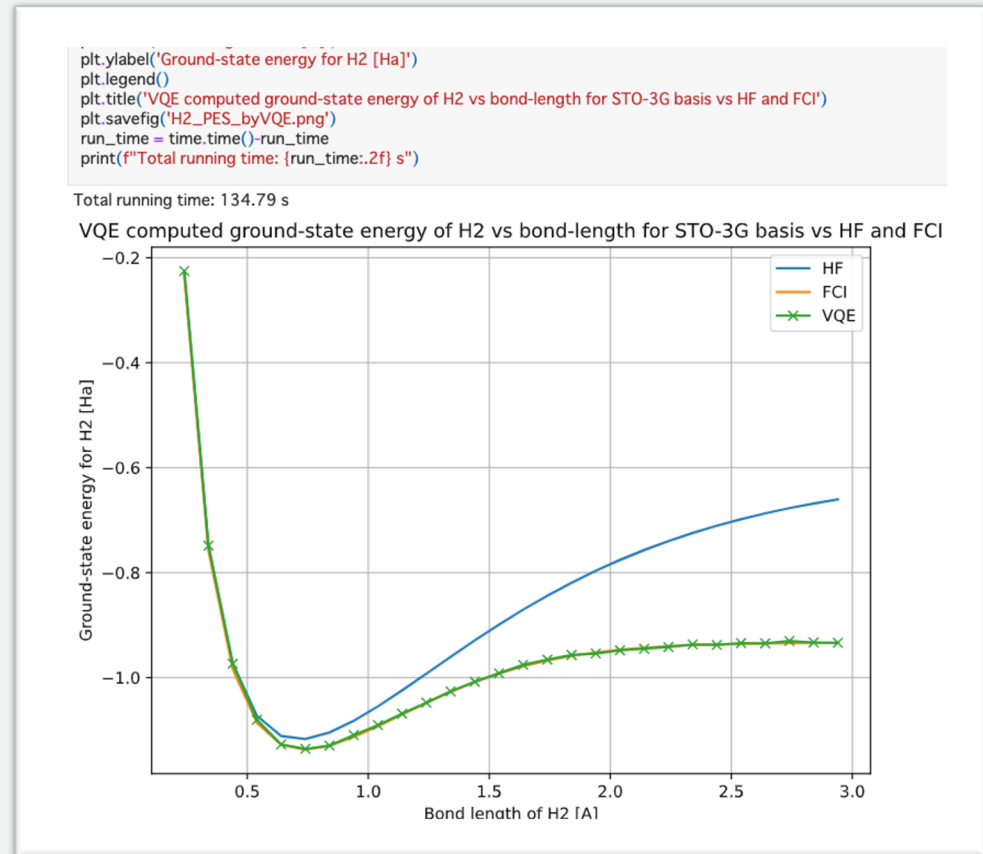
Amazon Braket examples : サンプルノートブック



The screenshot shows the GitHub repository for 'aws / amazon-braket-examples'. It is a public repository with 2 issues, 11 pull requests, and a security advisory. The current branch is 'main'. There are 29 authors. A list of example folders is provided:

- advanced_circuits_algorithms: Add embedded simulators exam
- braket_features: Add embedded simulators exam
- getting_started: Add embedded simulators exam
- hybrid_jobs: Add embedded simulators exam
- hybrid_quantum_algorithms: Add embedded simulators exam
- pennylane: Add embedded simulators exam
- quantum_annealing: Removing need to specify S3 in t

Amazon Braket を今すぐにスタートできる
サンプルノートブックを多数提供



VQE による 量子古典ハイブリッド手法
水素分子のポテンシャルエネルギー計算

VQE (Variational Quantum Eigensolver) on Amazon Braket

1. 量子ビットによる表現

Jordan-Wigner を使ったフェルミ粒子から量子ビットへのマッピング

2. Ansatz と初期トライアル関数:

量子状態 $|\psi\rangle$ を生成するための量子回路

UCCSD ansatz とパラメータ θ を用いて量子状態を生成

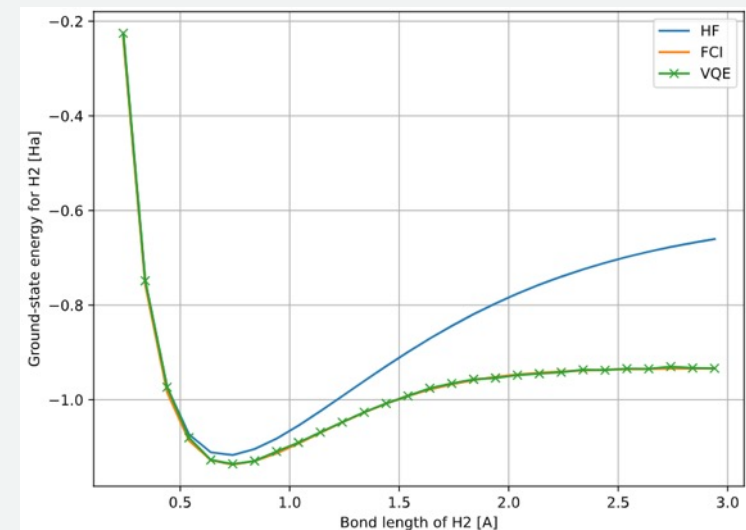
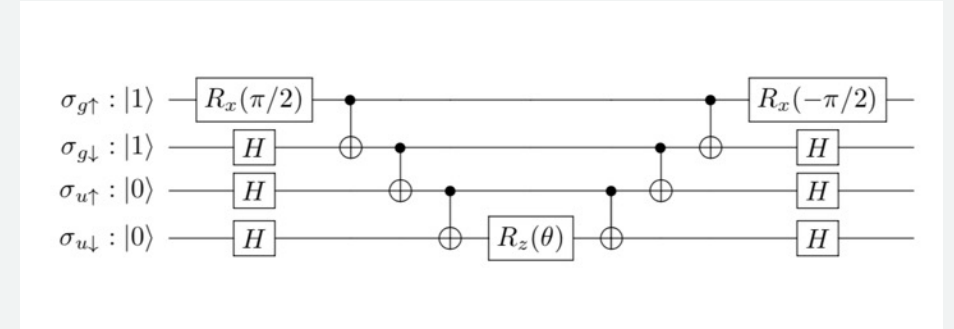
3. 測定

期待値を測定

4. 様々なボンド長におけるエネルギー計算

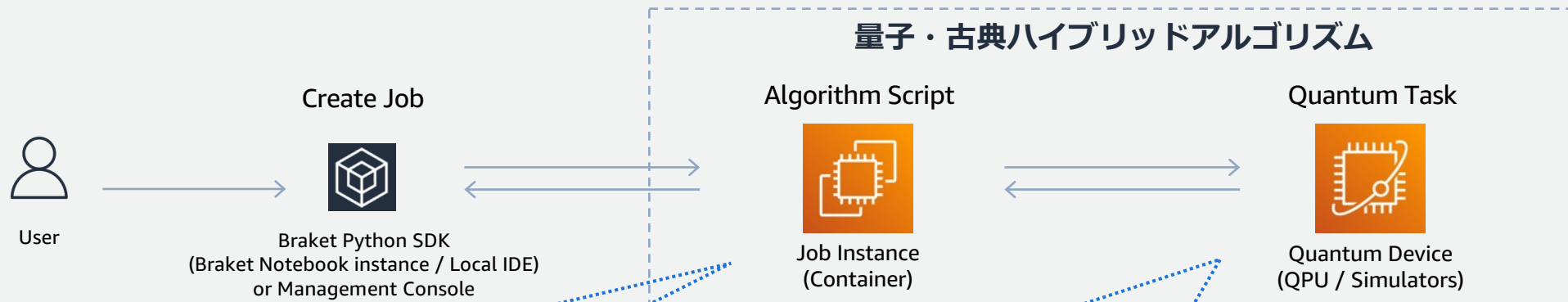
古典計算機側 (CPU) でパラメータの更新

https://github.com/aws/amazon-braket-examples/blob/main/examples/hybrid_quantum_algorithms/VQE_Chemistry/VQE_chemistry_braket.ipynb



VQE computed ground-state energy of H2 vs bond-length for STO-3G basis vs HF and FCI

Amazon Braket Hybrid Jobs で実行する VQE



古典計算機によるパラメータ計算

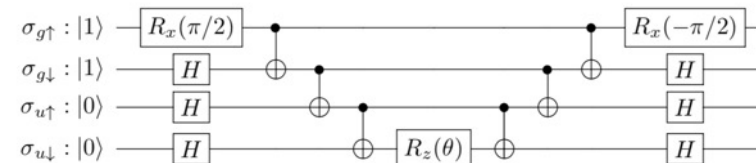
- パラメータの最適化・アップデート

勾配降下法などを用いて次のパラメータを計算

```
opt = qml.GradientDescentOptimizer(stepsize=0.4)
```

PennyLane ライブラリでVQEを簡単に記述できる

Ansatz を用いた期待値の量子計算



```
def circuit(params, wires):  
    qml.templates.AllSinglesDoubles(  
        params, wires, hf_state, singles, doubles)
```

[https://github.com/aws/amazon-braket-examples/tree/main/examples/pennyLane/3 Quantum chemistry with VQE](https://github.com/aws/amazon-braket-examples/tree/main/examples/pennyLane/3%20Quantum%20chemistry%20with%20VQE)

QPU の特徴

QPU 稼働時間

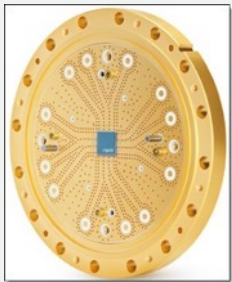
QPU	Availability UTC	Availability JST (日本時間)
D-Wave Advantage_system6.1 Advantage_system4.1 DW_2000Q_6	Everyday, 00:00 - 23:59 UTC (24 Hours)	Everyday, 00:00:00 - 23:59:59 JST (24 Hours)
Rigetti Aspen-M-2	Weekdays, 04:00 – 06:00 UTC Weekdays, 15:00 - 18:00 UTC	Monday-Friday, 13:00 – 15:00 JST Tuesday-Saturday, 0:00 – 5:00 JST
IonQ	Weekdays, 13:00 - 02:00 UTC	Tuesday-Saturday, 22:00-11:00 JST
Oxford Quantum Circuits - Lucy	Weekdays, 09:00 - 12:00 UTC	Tuesday-Saturday, 18:00 – 9:00 JST
Xanadu - Borealis	Weekdays, 15:00 - 17:00 UTC	Tuesday-Saturday, 0:00 – 2:00 JST

Rigetti



Rigetti Computing 汎用ゲート型超伝導量子ビット

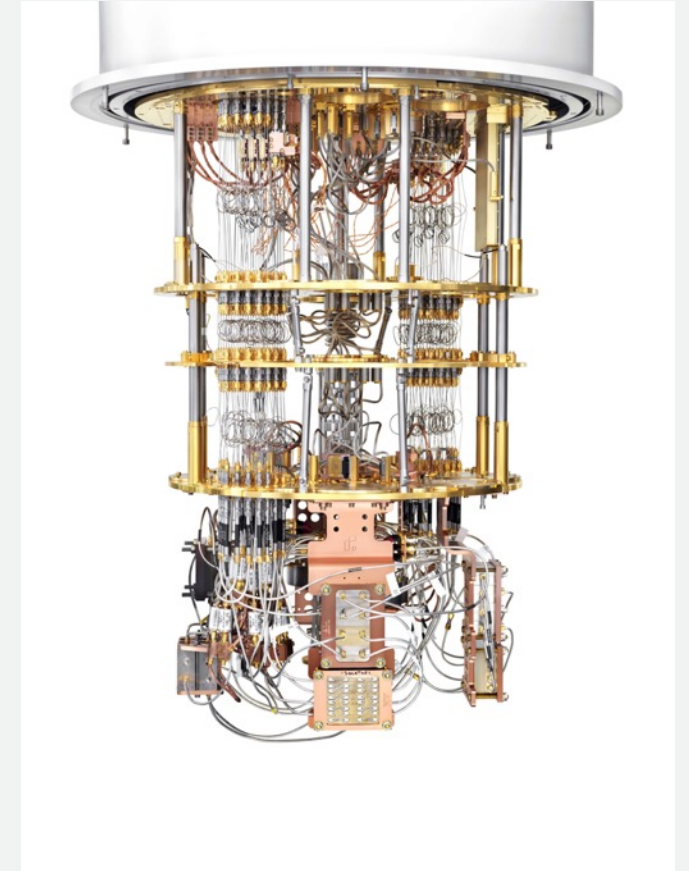
- 超伝導量子ビットに基づく汎用ゲートモデルの量子マシン
- Aspenシリーズチップ:
 - 固定量子周波数と可変超伝導量子ビットによる構成による精度、スケール、速度の強力な組み合わせ
 - 8角形グラフ接続で量子ビット接続性は3倍
 - 最近傍でない隣接量子ビットを自動的にリンク



The Rigetti 16Q Aspen-4



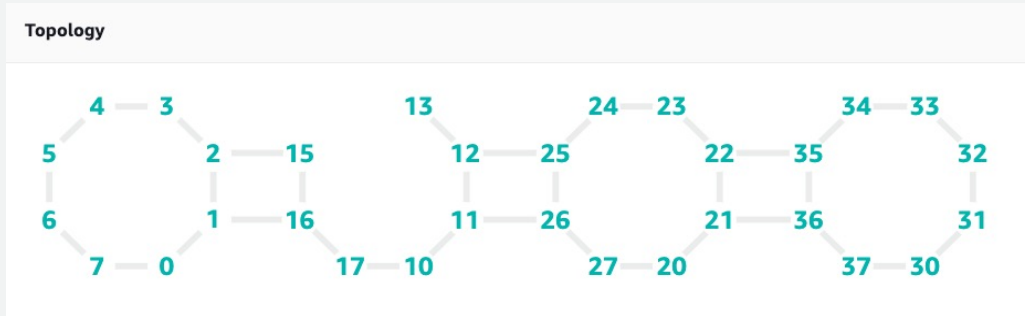
Rigetti Aspen シリーズチップ:
スケーラブルな回路トポロジ。base-8でラベル付けされたグラフ構造の量子ビット間接続



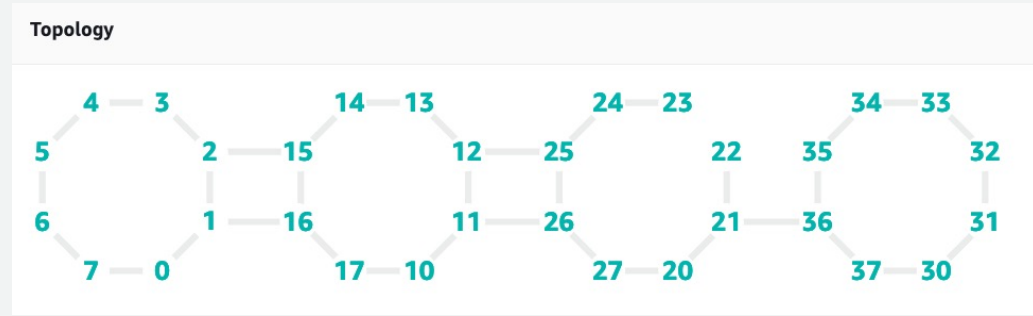
<https://aws.amazon.com/braket/hardware-providers/rigetti>

Rigetti チップ構成

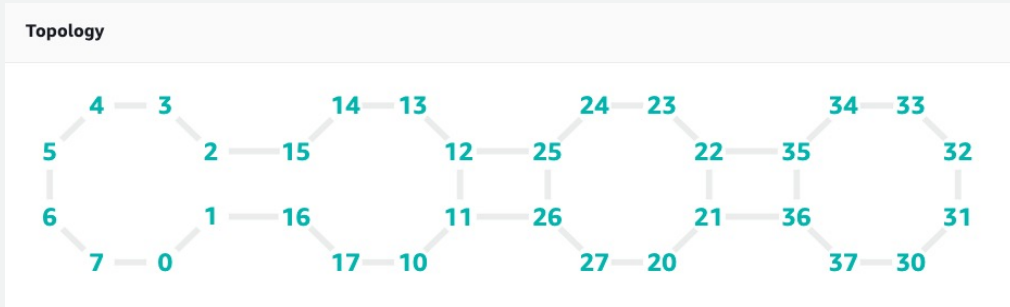
Aspen-8



Aspen-10



Aspen-9

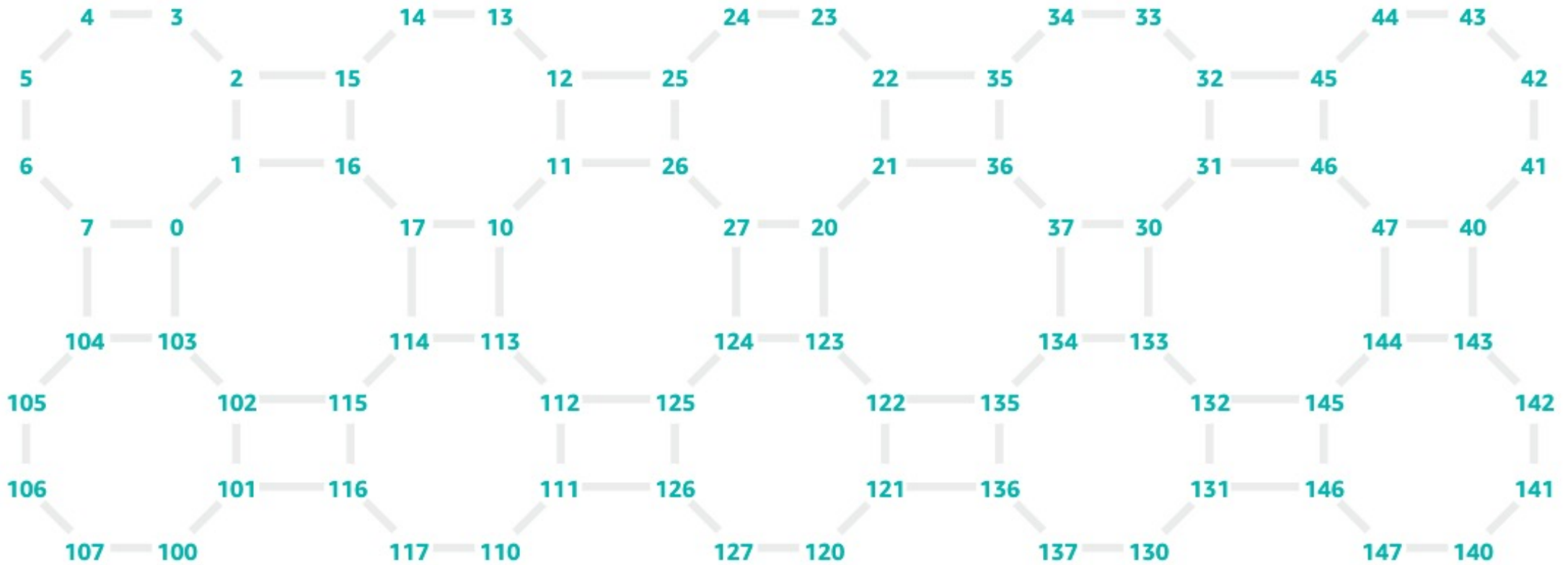


Aspen-11



Rigetti Aspen-M-2 チップ構成

Topology



Rigetti Chip Calibration Aspen-M-2

1 量子ビットゲートに対する Calibration

Calibration						
Last updated: Sep 19, 2022 21:16 (UTC) Info						
<input type="text" value="Qubit"/>						
Qubit	T1 (μs) Info	T2 (μs) Info	Fidelity (RB) (%) Info	Fidelity (simultaneous RB) (%) Info	Readout fidelity (%) Info	Active reset fidelity (%) Info
0	12.869	21.538	97.489 ± 0.363	98.235 ± 0.164	78.800	93.800
1	56.364	7.088	99.697 ± 0.018	99.579 ± 0.034	92.800	97.300
2	21.237	21.585	99.899 ± 0.007	99.818 ± 0.008	98.600	99.750
3	69.485	36.988	99.916 ± 0.020	99.885 ± 0.043	94.700	97.700
4	26.360	37.637	99.916 ± 0.011	99.793 ± 0.011	98.600	99.650
5	47.744	110.757	99.734 ± 0.034	99.908 ± 0.120	97.400	99.200

2 量子ビットゲートに対する Calibration

Calibration			
Last updated: Sep 19, 2022 21:16 (UTC) Info			
<input type="text" value="Qubit"/>			
Edge (Qubit pair)	C-Phase gate fidelity (%) Info	XY gate fidelity (%) Info	CZ gate fidelity (%) Info
0-1	81.904 ± 1.123	82.015 ± 1.995	85.986 ± 0.836
0-7	82.563 ± 1.255	87.779 ± 0.860	83.724 ± 1.869
1-16	96.284 ± 0.737	97.439 ± 1.234	93.162 ± 2.466
1-2	88.008 ± 0.758	84.935 ± 0.703	81.605 ± 1.045
10-11	91.250 ± 1.037	93.434 ± 0.932	96.237 ± 0.521
10-113	95.341 ± 1.117	-	98.923 ± 0.387
10-17	97.315 ± 0.881	96.516 ± 0.814	96.684 ± 0.691

量子ビットのマニュアル割当 (Native Gate Support)



Rigetti Aspen-11 のトポロジー

- Rigetti, OQC, IonQ で利用可能
- より高精度な量子操作を明示的にコントロール

```
# create a random state with neighboring qubits
```

```
circuit = Circuit()
```

```
circuit.rz(0,np.pi/2).cnot(1,2).x(3)
```

```
print(circuit)
```

```
rigetti_task = device.run(circuit, s3_folder, shots=100, disable_qubit_rewiring=True)
```

← タスク実行時に引数で指定

https://github.com/aws/amazon-braket-examples/blob/main/braket_features/Allocating_Qubits_on_QPU_Devices.ipynb

<https://aws.amazon.com/jp/about-aws/whats-new/2020/11/amazon-braket-supports-manual-qubit-allocation/>

Set Verbatim Compilation

- Rigetti QPU 利用時に、指定した量子回路を勝手に(効率的なものに)変換せず Error mitigation やベンチマークが可能
- Rigetti, OQC, IonQ で利用可能
- `disable_qubit_rewiring` を True にセット

```
from braket.aws import AwsDevice
from braket.device_schema.device_action_properties import DeviceActionType
device = AwsDevice("arn:aws:braket:us-west-1::device/qpu/rigetti/Aspen-M-2")
device.properties.action[DeviceActionType.OPENQASM].supportedPragmas
```

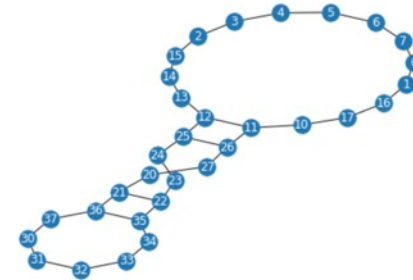
<https://docs.aws.amazon.com/braket/latest/developerguide/braket-constructing-circuit.html?#verbatim-compilation>

Programming verbatim circuits onto the device topology

As we have mentioned above, to build circuits with multi-qubit gates, you need to take into consideration the connectivity graph of the device. When a circuit contains a verbatim box, automatic qubit rewiring has to be disabled, and you have manually allocate the qubits on the device that you want to use for your circuit. You can access the connectivity graph on the [device detail page](#) in the Amazon Braket Console, or by using the code below.

```
In [10]: # access and visualize the device topology
print(rigetti.properties.paradigm.connectivityGraph)
nx.draw_kamada_kawai(rigetti.topology_graph, with_labels=True, font_color="white")

{'0': ['1', '7'], '1': ['0', '16'], '10': ['11', '17'], '11': ['10', '12', '26'], '12': ['11', '13', '25'], '13': ['12', '14'], '14': ['13', '15'], '15': ['14', '2'], '16': ['1', '17'], '17': ['10', '16'], '2': ['15', '3'], '20': ['2', '1', '27'], '21': ['20', '22', '36'], '22': ['21', '23', '35'], '23': ['22', '24'], '24': ['23', '25'], '25': ['12', '24', '26'], '26': ['11', '25', '27'], '27': ['20', '26'], '3': ['2', '4'], '30': ['31', '37'], '31': ['30', '32'], '32': ['31', '33'], '33': ['32', '34'], '34': ['33', '35'], '35': ['22', '34', '36'], '36': ['21', '35', '37'], '37': ['30', '36'], '4': ['3', '5'], '5': ['4', '6'], '6': ['5', '7'], '7': ['0', '6']}
```



From the connectivity graph, you can see that qubits 11, 10, and 17 are connected in a line, and with the code in the next cell you can access their respective 2-qubit gate fidelities to make sure you have selected a high-quality qubit subset.

```
In [11]: print(rigetti.properties.provider.specs["2Q"]["10-11"])
print(rigetti.properties.provider.specs["2Q"]["10-17"])

{'fCPHASE': 0.9370584109675574, 'fCPHASE_std_err': 0.00795074764345174, 'fCZ': 0.9299243089381443, 'fCZ_std_err': 0.014825613277350928, 'fXY': 0.9745254923082562, 'fXY_std_err': 0.008192514619096408}
{'fCPHASE': 0.9814191643655765, 'fCPHASE_std_err': 0.003640156618078282, 'fCZ': 0.9901793460716128, 'fCZ_std_err': 0.002845237063339863, 'fXY': 0.9619550297476629, 'fXY_std_err': 0.005187344784704715}
```

Note: At the time when you run this notebook the fidelity numbers may be different as QPU devices are periodically recalibrated

After selecting the qubits and validating their gate fidelities, you can now construct a circuit and run it.

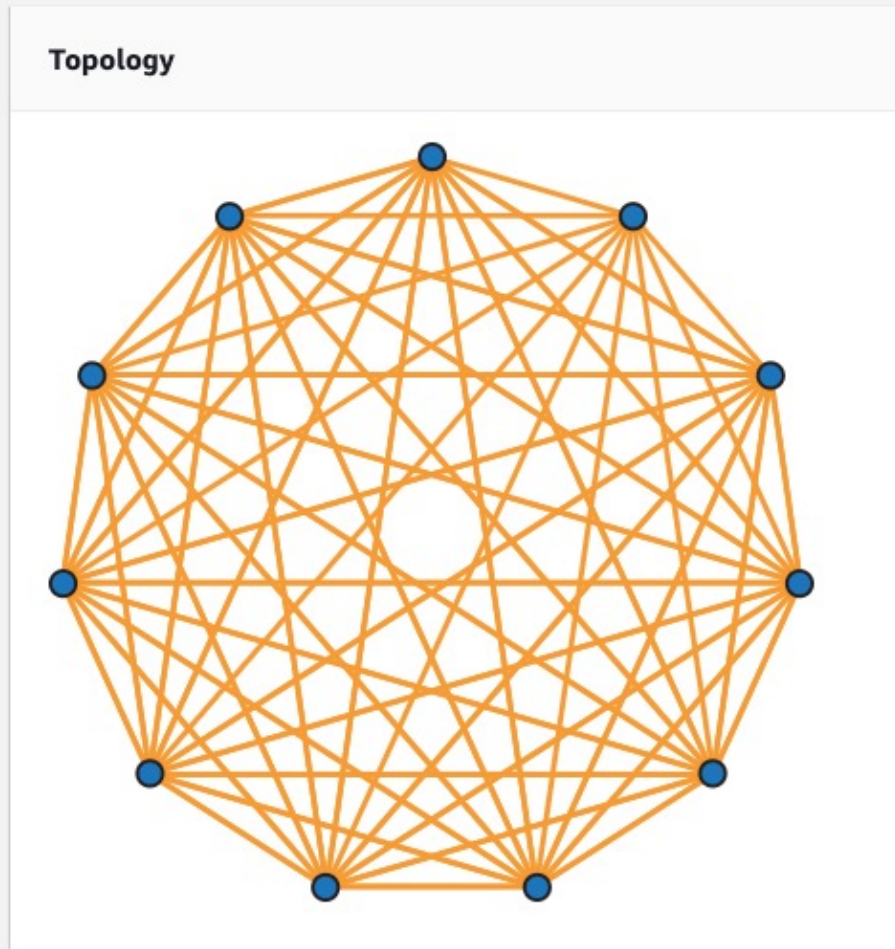
```
In [12]: circ = Circuit().xy(10,11,pi/4).xy(10,17,pi/2).rx(10,pi).xy(10,11,pi/4)
verbatim_circ = Circuit().add_verbatim_box(circ)
print(verbatim_circ)
```

```
T : | 0 | 1 | 2 | 3 | 4 | 5 |
q10 : -StartVerbatim-XY(0.785)-XY(1.57)-Rx(3.14)-XY(0.785)-EndVerbatim-
q11 : |-----XY(0.785)-----|-----XY(0.785)-----|-----
q17 : -*****-----XY(1.57)-----*****
T : | 0 | 1 | 2 | 3 | 4 | 5 |
```

IonQ



IonQ トポロジーとキャリブレーション



Calibration
Last updated: Aug 07, 2020 12:47 (UTC)

```
1- {  
2-   "braketSchemaHeader": {  
3-     "name": "braket.device_schema.ionq.ionq_provider_properties",  
4-     "version": "1"  
5-   },  
6-   "fidelity": {  
7-     "1Q": {  
8-       "mean": 0.99717  
9-     },  
10-    "2Q": {  
11-      "mean": 0.9696  
12-    },  
13-    "spam": {  
14-      "mean": 0.9961  
15-    }  
16-  },  
17-  "timing": {  
18-    "T1": 10000,  
19-    "T2": 0.2,  
20-    "1Q": 0.000011,  
21-    "2Q": 0.00021,  
22-    "readout": 0.000175,  
23-    "reset": 0.000035  
24-  }  
25- }
```

11 量子ビット 全結合 (レーザー光によるオンデマンドな結合)

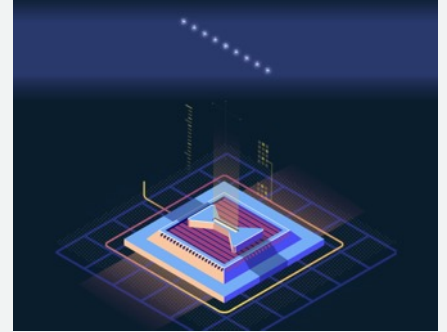
イオントラップを用いた IonQ の特徴

• Ion Trap

- 量子ビット：イオン化された Yb^+ (イッテルビウム) 原子
- 100本程度の電極により正確にデザインされた、*linear Ion Trap* (線形イオントラップ) 技術を使い、イオンを捕獲する
- 電磁場によって3次元空間にトラップすることにより、環境ノイズや緩和に強くなる
- 10^{-11} Torr の真空チャンバで動作

• Gate Configuration

- レーザー照射による**オンデマンドな結合** (物理的な結合を伴わない) を形成することによって全結合が可能に
- 同一チップに量子ビットトラップを再構成可能
 - 最大79量子ビットイオンチェーン、任意の全結合は11量子ビットまで
- 共鳴光の照射により状態を読み出し



IonQ の性能とベンチマーク

Performance Benchmarks†

Qubits

Single-qubit gates on

79 Qubits

Two-qubit gates on all pairs up to

11 Qubits

Average Fidelity

Single-qubit gates

>99%

Two-qubit gates

>98%*

Best Fidelity

Single-qubit gates

>99.97%

Two-qubit gates

>99.3%*

Minimum Fidelity

Single-qubit gates

>99%

Two-qubit gates

>96%*

Coming Soon: 32 Qubits

We are currently gathering detailed performance data on our latest system, which features 32 fully-connected qubits and an expected quantum volume of over 4 million.

[more about this historic milestone →](#)

† these data are from our [benchmarking paper published in nature communications](#). For the most up-to-date performance data, we recommend using the information provided by our [cloud partners](#).

* not corrected for state preparation and measurement errors [?](#)

D-WAVE

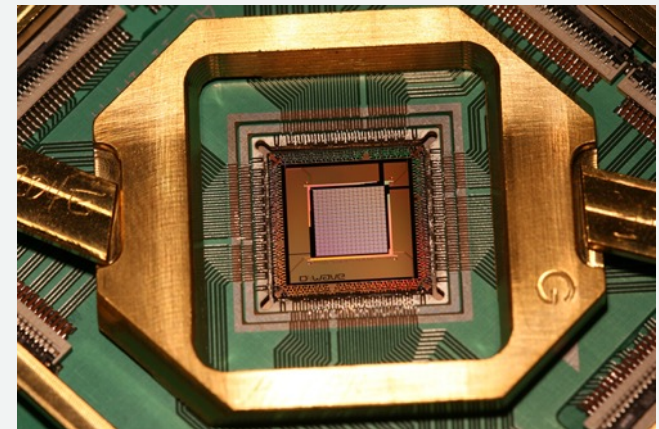


D-Wave 量子アニーリング

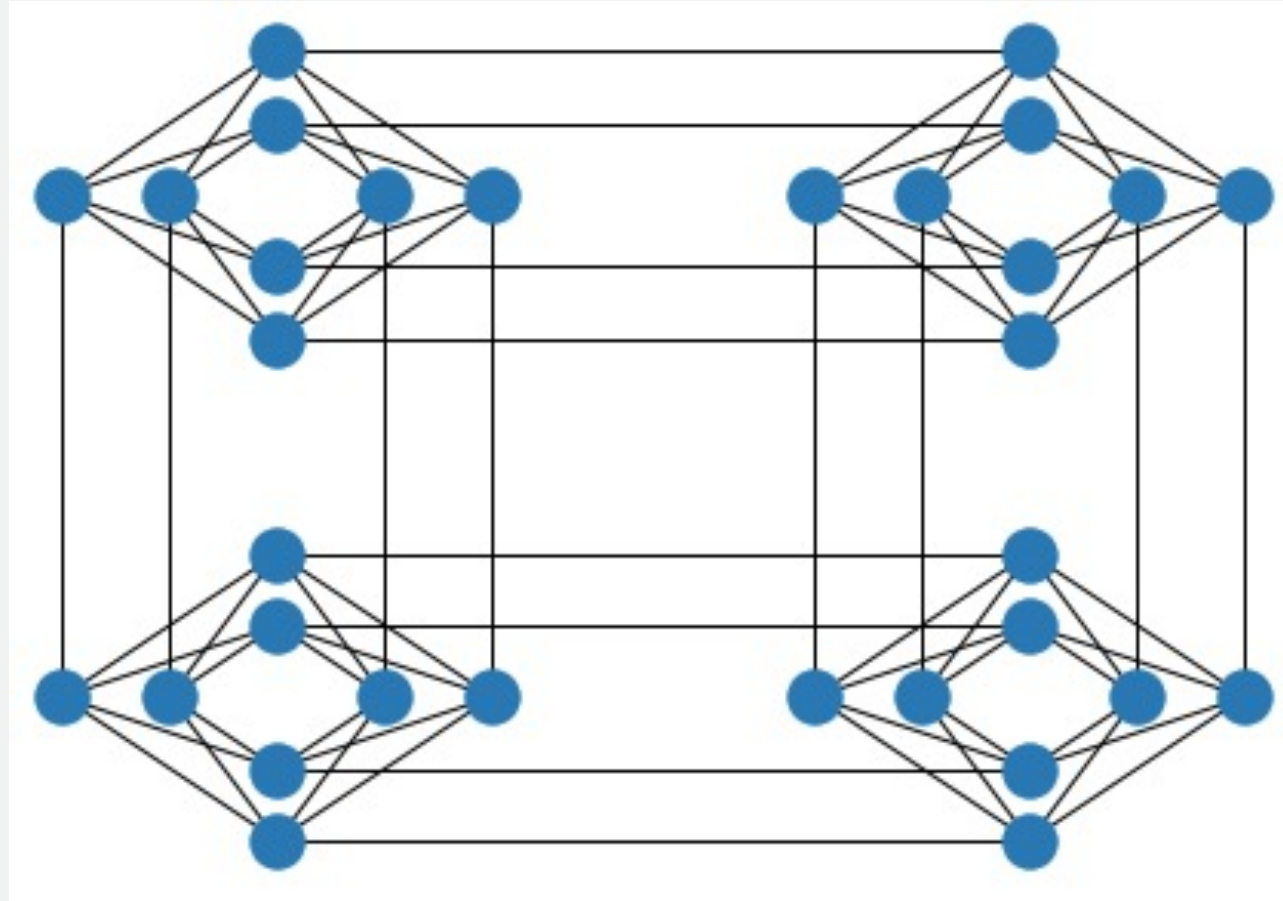
- 超伝導量子ビットに基づくアニーリングモデルの量子マシン
- 相互に接続された超伝導フラックス量子ビットのネットワーク
- 各量子ビットはジョセフソン接合素子
- 2種類のQPU
 - D-Wave 2000Q (2048量子ビット)
 - Advantage_system4.1 (5760量子ビット)

$$H_{ising} = \underbrace{-\frac{A(s)}{2} \left(\sum_i \hat{\sigma}_x^{(i)} \right)}_{\text{Initial Hamiltonian}} + \underbrace{\frac{B(s)}{2} \left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j} \hat{\sigma}_z^{(i)} \hat{\sigma}_z^{(j)} \right)}_{\text{Final Hamiltonian}}$$

横磁場を徐々に弱める

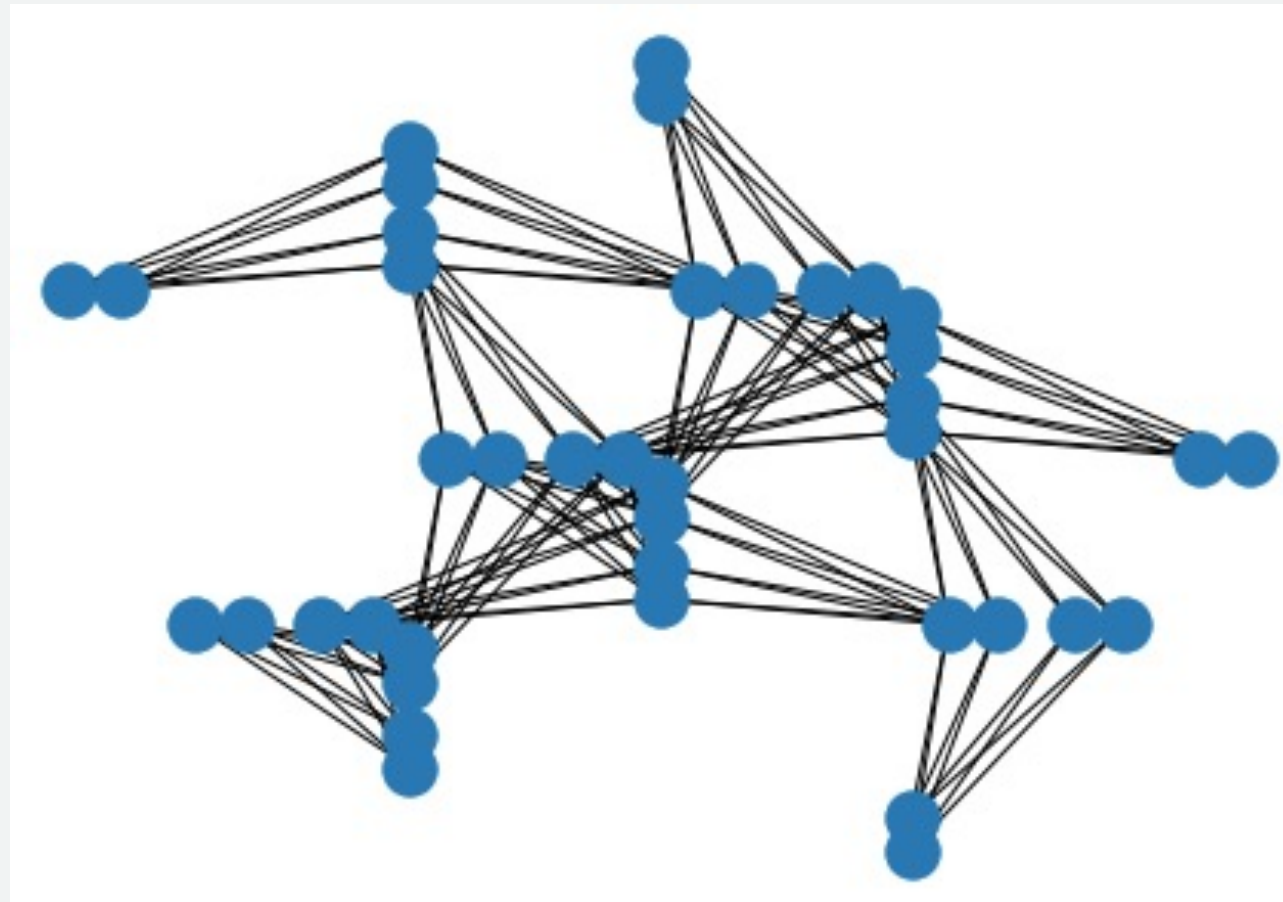


D-Wave 2000Q トポロジ



キメラグラフ

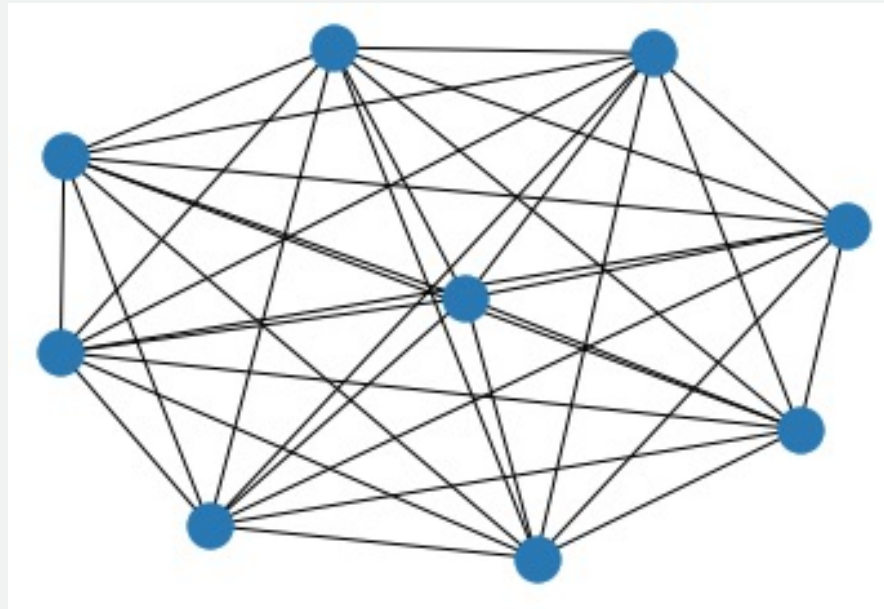
Advantage_system4.1 トポロジ



ペガサスグラフ

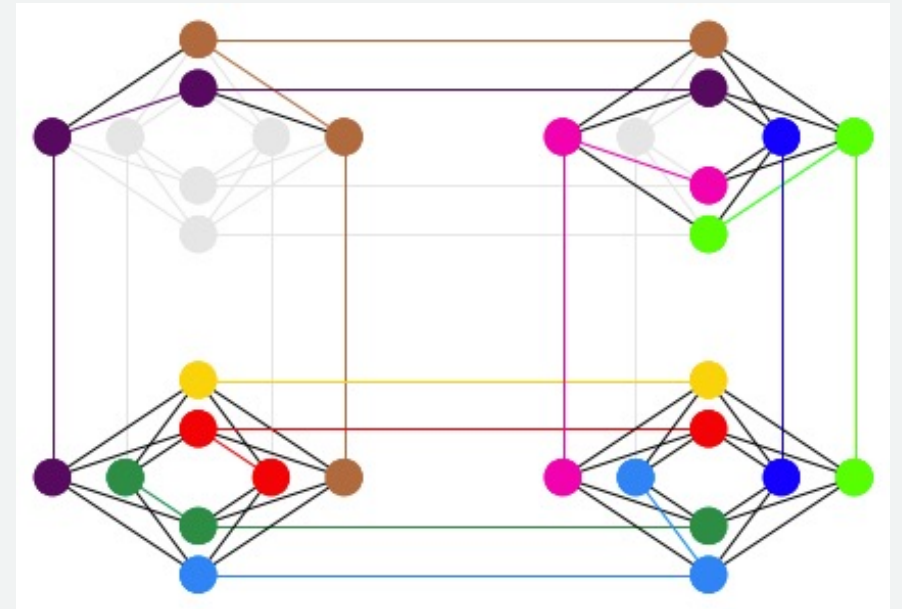
© 2022, Amazon Web Services, Inc. or its affiliates.

論理グラフを物理グラフに埋め込む



論理グラフ
(解きたい問題)

埋め込み



同色を1論理ビットと見立てた
物理グラフ

量子回路シミュレータについて

フルマネージドなゲート量子計算シミュレータ

```
import boto3
from braket.aws import AwsDevice
from braket.circuits import Circuit

aws_account_id = boto3.client("sts").get_caller_identity()["Account"]

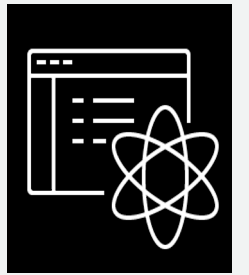
device = AwsDevice("arn:aws:braket:::device/quantum-simulator/amazon/sv1")
s3_folder = (f"amazon-braket-output-{aws_account_id}", "folder-name")

bell = Circuit().h(0).cnot(0, 1)
task = device.run(bell, s3_folder, shots=100)
print(task.result().measurement_counts)
```

Universal Quantum gate based computing simulator:

- Local simulator:
 - 25量子ビットまでのクイックなプロトタイピングに最適
- SV1: 量子計算の状態ベクトルシミュレータ
- TN1: テンソルネットワークシミュレータ
- DM1: 密度行列シミュレータ

量子ゲートモデルのマネージドシミュレータ



Simulator

- SV1 (状態ベクトルシミュレータ)
 - 34量子ビットまで
 - 高密度など任意の回路をシミュレート可能
 - Square Circuit (深さ34)の場合 1~2 時間の計算時間
 - デフォルトで35並列タスク (最大50並列) まで実行可能
- TN1 (テンソルネットワークシミュレータ)
 - 50量子ビットまで
 - 回路の深さ100以下
 - スパース回路、ローカルゲートを備えた回路、QFT回路などに最適
 - 最大実行時間は6時間
- DM1 (密度演算シミュレータ)
 - ノイズコントロール可能
 - 最大17量子ビットまで
 - デフォルトで35並列タスク (最大50並列) まで実行可能
 - 最大実行時間は6時間

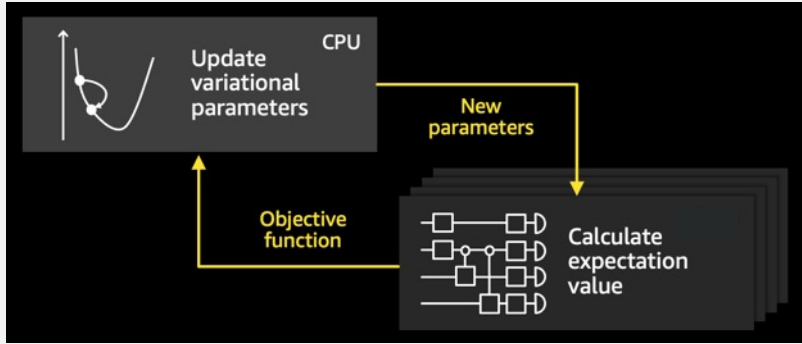
<https://docs.aws.amazon.com/braket/latest/developerguide/braket-devices.html>

<https://docs.aws.amazon.com/braket/latest/developerguide/braket-result-types.html>

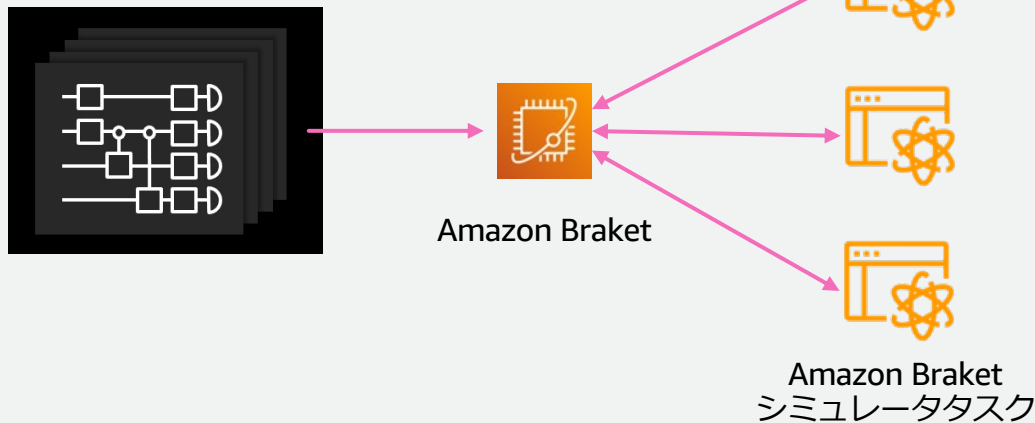


10倍の高速化を実現する Amazon Braket シミュレータ

変分アルゴリズムでは、すべてのステップで多くの回路実行が必要

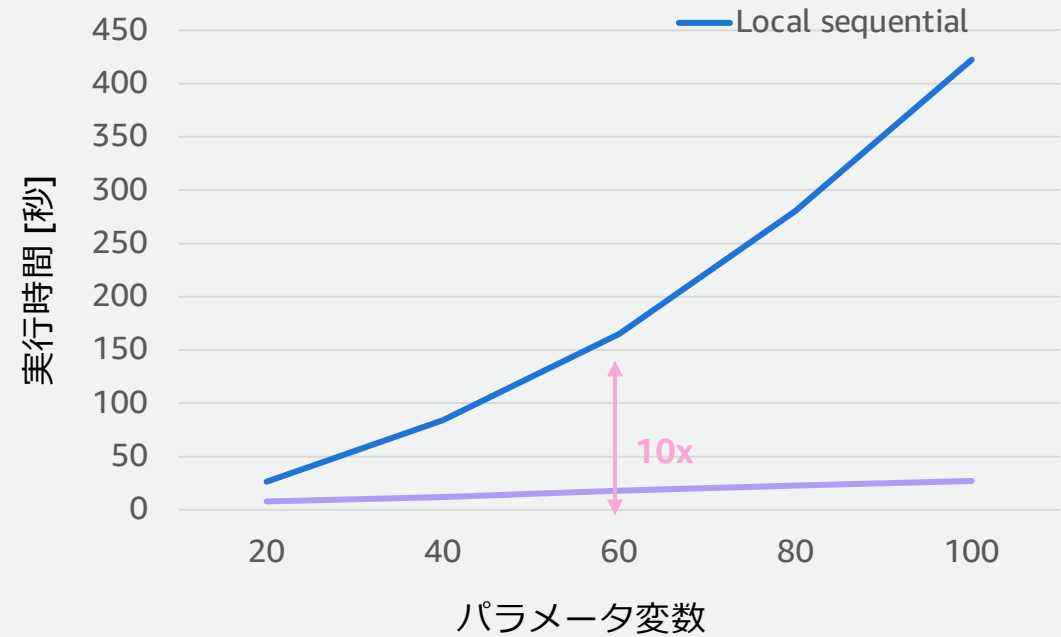


Amazon Braket シミュレータで回路の実行を並列化する



トレーニング時間を10倍短縮し、アルゴリズムの研究とチューニングを加速

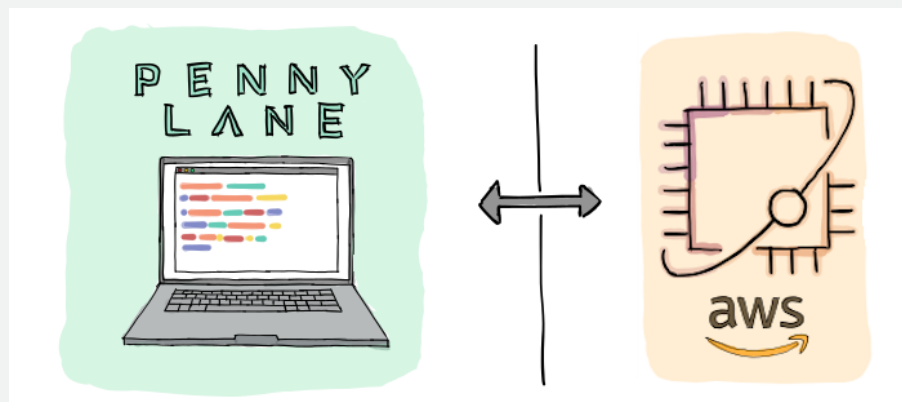
単一の勾配評価の時間*



□ ーカルシミュレータ: m5.12xlarge EC2 インスタンス + PennyLane SV1: 30までの並行タスク



PennyLane が Amazon Braket で量子回路を学習



Defining a circuit

We will choose a simple two-qubit circuit with two controllable rotations and a CNOT gate.

```
@qml.qnode(dev)
def circuit(params):
    qml.RX(params[0], wires=0)
    qml.RY(params[1], wires=1)
    qml.CNOT(wires=[0, 1])
    return qml.expval(qml.PauliZ(1))
```

The `qml.qnode(dev)` decorator binds the circuit to the local Braket device. Now, every time that `circuit()` is called, the quantum computation defined in the function above will be executed with Braket.

機械学習ツールを量子コンピューティングに

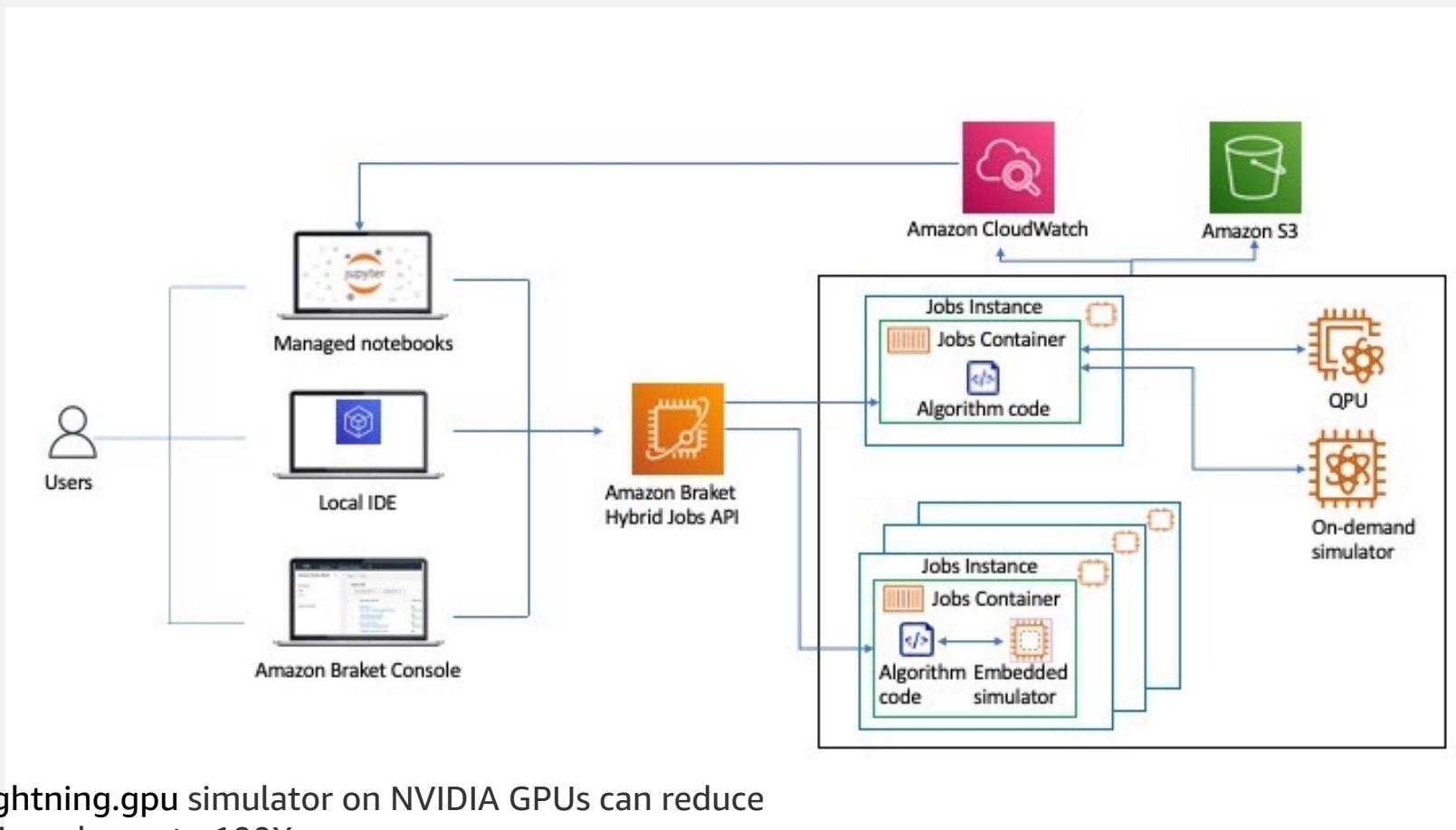
- › 量子微分プログラミングを PyTorch や TensorFlow を使ってより簡単に実装できる
- › 量子古典ハイブリッド計算を実装

ハイパフォーマンスな計算の実装が簡単に

- › Amazon Braket で事前定義されたチュートリアルですぐに開始
- › 化学、最適化、機械学習等のアプリケーションライブラリにアクセス
- › Amazon Braket を使って10倍も高速な学習

Embedded Simulators

Accelerate hybrid quantum-classical algorithms on Amazon Braket using embedded simulators from Xanadu's PennyLane featuring NVIDIA cuQuantum



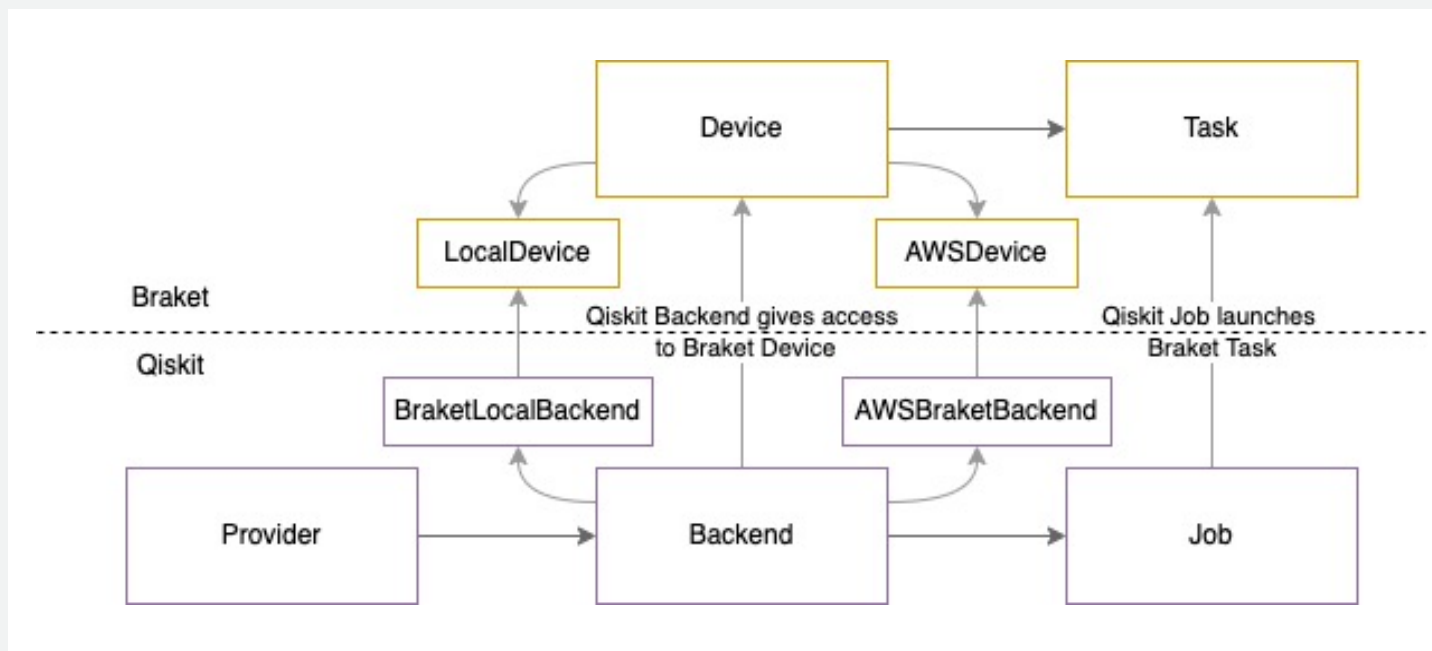
PennyLane's `lightning.gpu` simulator on NVIDIA GPUs can reduce algorithm runtimes by up to 100X

Amazon Braket Hybrid Jobs architecture:



Qiskit-Braket Provider が Amazon Braket にも対応

- Qiskit から Amazon Braket に接続、IonQ や Aspen-M-1 などさまざまな QPU を実行可能
- Qiskit VQE などのアルゴリズムを Braket backend で実行可能



☰ README.md

Qiskit-Braket provider

Qiskit-Braket provider to execute Qiskit programs on AWS quantum computing hardware devices through Amazon Braket.

Table of Contents

For Users

1. [Installation](#)
2. [Quickstart Guide](#)
3. [Tutorials](#)
4. [How-Tos](#)
5. [How to Give Feedback](#)
6. [Contribution Guidelines](#)
7. [References and Acknowledgements](#)
8. [License](#)

<https://aws.amazon.com/jp/blogs/news/introducing-the-qiskit-provider-for-amazon-braket/>
<https://github.com/qiskit-community/qiskit-braket-provider/>

AWS が提供する 量子コンピューティングの範囲

Quantum Computing

Amazon Braket

ユニバーサルゲートモデル

- Rigetti (超伝導) - 80 量子ビット
- IonQ (イオントラップ) - 11 量子ビット
- OQC (超伝導) - 8 量子ビット

古典シミュレータ

- Local Simulator – 25 量子ビット
- SV1 - 34 量子ビット, DM1 – 17 量子ビット
- TN1 - 50 量子ビット

光量子コンピュータ

- Xanadu (光量子) – 216 量子モード
 - Strawberry Fields library

量子アニーリング

- D-Wave 2000 (超伝導) 2048 量子ビット
- D-Wave Advantage (超伝導) 5760 量子ビット
 - Ocean SDK

イジングシミュレータ

- QUBO solver (Meta-Analytics) – 400,000 ノード
- SBM (Toshiba) – 10,000 ノード

料金

• 量子コンピュータ (QPU)

Hardware Provider	QPU family	Per-task price	Per-shot price
D-Wave	2000Q, Advantage	\$0.30000	\$0.00019
IonQ	IonQ device	\$0.30000	\$0.01000
Rigetti	Aspen-M-1	\$0.30000	\$0.00035
OQC	Lucy	\$0.30000	\$0.00035
Xanadu	Borealis	\$0.30000	\$0.0002

• 量子回路シミュレータ

- SV1 simulator \$ 0.075 / min (\$ 4.5 / hr)
- DM1 simulator \$ 0.075 / min (\$ 4.5 / hr)
- TN1 simulator \$ 0.275 / min (\$ 16.5 / hr)
- 開発環境 Amazon SageMaker notebook 実行は別途課金

料金について

料金例 1: Amazon Braket マネージドシミュレーターSV1 を利用して回路をシミュレートする場合

- 30量子ビットを使用する量子回路を設計し、シミュレーションの実行には69分（1.15時間）かかった場合。
- 5.175ドル = 1時間あたり4.50ドル x 1.15
- （SV1の実行にかかる時間単位、最小15秒単位の秒課金となります）

料金例 2: D-Wave2000Q 量子コンピューターで量子アニーリング問題を実行する場合

- アニーリング問題の2,000の結果サンプルが含まれていた場合のタスクの実行コスト
- 0.68ドル = タスクあたりの料金 0.30ドル + (ショットあたりの価格は0.00019ドル x 2000ショット)

料金例 3: Rigetti Aspen-M-1 量子コンピューターで量子アルゴリズムを実行する場合

- 回路設計の10,000回の繰り返しショットが含まれている場合のタスクの実行コスト
- 3.80ドル = タスクあたりの料金 0.30ドル + (ショットあたりの価格は0.00035ドル x 10,000ショット)

マネージドシミュレーターとRigettiデバイスに許可されるショットの最大数は100,000。DwaveおよびIonQデバイスの場合は10,000。



參考資料

- Amazon Braket documentation
 - <https://docs.aws.amazon.com/braket/latest/developerguide/what-is-braket.html>
- Amazon Braket Python SDK - Read the Docs
 - <https://amazon-braket-sdk-python.readthedocs.io/en/latest/>
- Boto3 SDK
 - <https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/braket.html>
- GitHub – Python Braket SDK
 - <https://github.com/aws/amazon-braket-sdk-python>
- GitHub – Amazon Braket Examples
 - <https://github.com/aws/amazon-braket-examples>

まとめ

- 特定の用途の量子計算に対して量子超越が実現
- NISQ 時代の量子計算
 - 大規模な量子誤り訂正が実現するまでの量子計算の活路
 - VQE 等を活用したアルゴリズム
 - 量子シミュレーション
- Amazon Braket クラウド活用による量子コンピューティング
 - 誰でもいますぐに量子計算を試せる時代に（量子超越も）
 - ライブラリ等 developer friendly なツールも充実