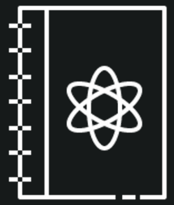
AWS のクラウド
量子コンピューティングサービス活用

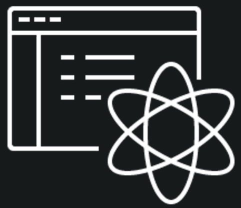
Shoko Utsunomiya, Ph.D.
Principal AI/ML Quantum Solutions Architect
Amazon Web Services

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



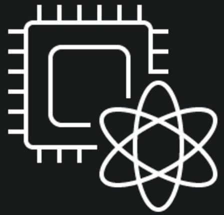
設計

マネージド開発環境



テスト

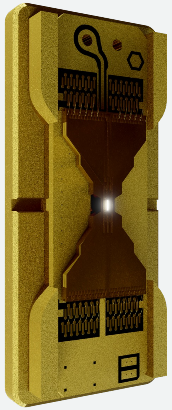
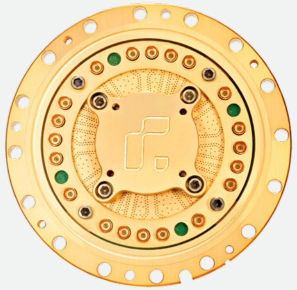
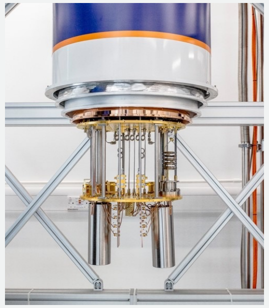
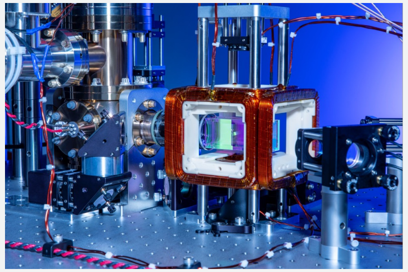
ハイパフォーマンスな
回路シミュレータ



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

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

- 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エア開発をスタート
- 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

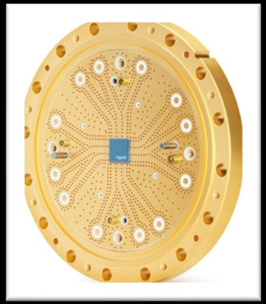
量子デバイス (QPU)

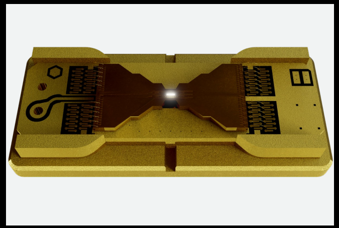
Gate-based QC

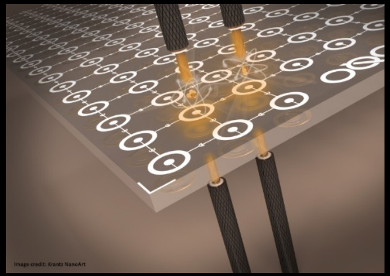
- **Rigetti :** 超伝導量子ビット
  - **Aspen-M-3:** 79 qubits, 2D topology architecture,
- **IonQ :** イオントラップ 、全結合
  - **Harmony:** 11 qubits
  - **Aria:** 25 qubits, error-mitigation
  - **Harmony :** 36 qubits, reservation only via Braket Direct
- **OQC :**超伝導量子ビット
  - **Lucy:** 8 qubits, 3D architecture with Coaximon

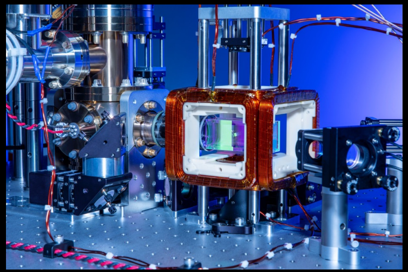
Analog QC

- **QuEra :** Rb 中性原子, Analog Hamiltonian Simulator (AHS)
  - **Aquila:** 256 qubits









利用量に応じた Amazon Braket の課金体系

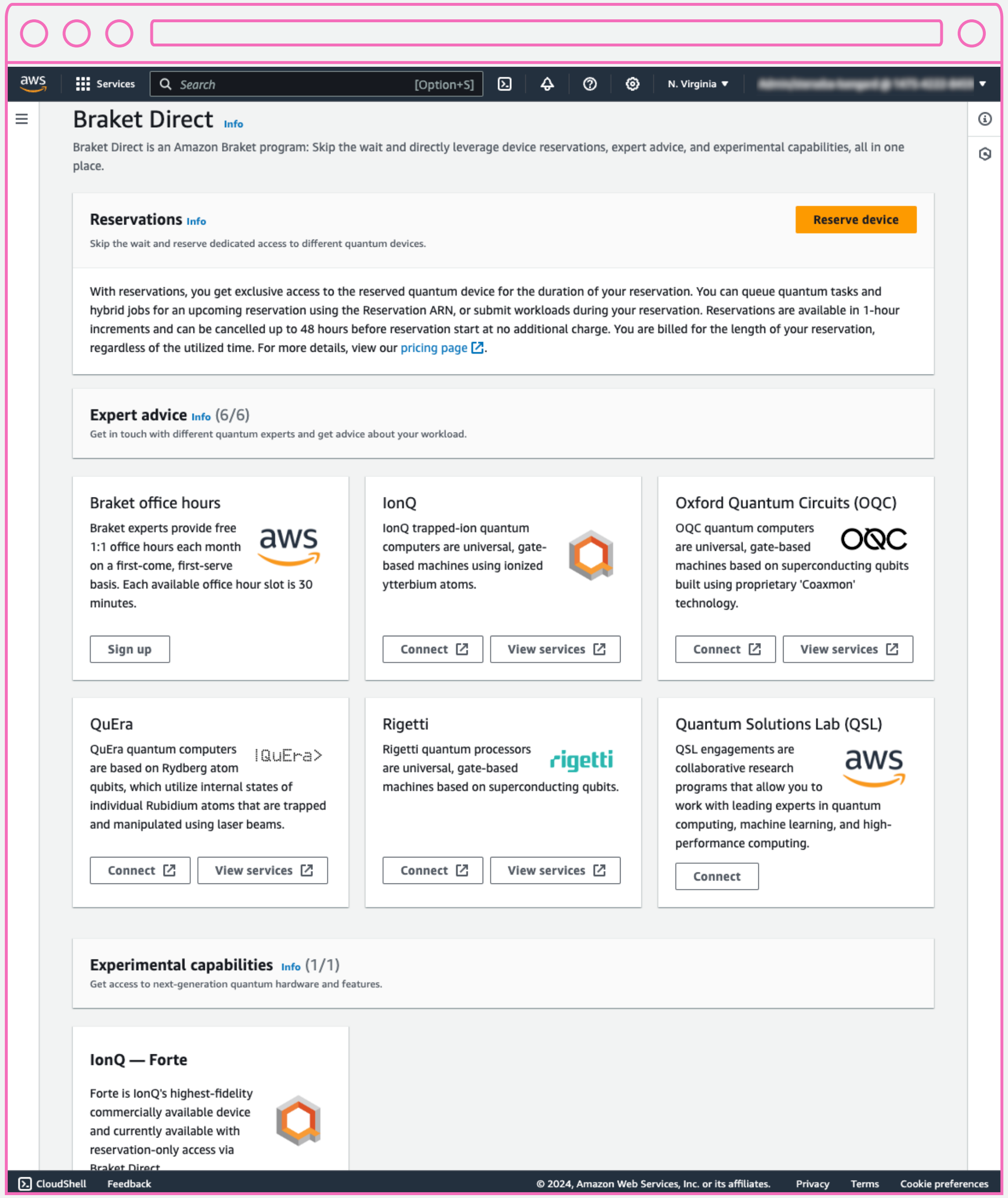
Hardware Provider	QPU family	Per-task price	Per-shot price
IonQ	Harmony	$0.30000	$0.01000
IonQ	Aria	$0.30000	$0.03000
OQC	Lucy	$0.30000	$0.00035
QuEra	Aquila	$0.30000	$0.01000
Rigetti	Aspen-M-3	$0.30000	$0.00035

Braket Direct

量子デバイスのキャパシティ
事前予約が可能に

量子コンピューティングの
スペシャリストと直接連携も可能

AWS で初めて利用可能となった
IonQ の最新36量子ビット QPU
「Forte」など、次世代ハードウェアや
機能へのアクセスが優先確保される



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

- ローカル環境で30量子ビットの量子回路を設計、SV1 ($0.075 / 分) でのシミュレーション実行に4分かかる場合
- Amazon Braket の利用開始から12ヶ月間は、ひと月あたり1時間の無料利用枠があるので、**$0**
- 利用開始から12ヶ月を過ぎると、($0.075 / 分) x 4 = **$0.3**

料金例 2 : Rigetti Aspen-M-3 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

- Aspen-M-3 で 10,000 ショットの計算
- タスクあたりの料金 $0.30 + (ショット単価 $0.00035 x 10,000 ショット) = **$3.80**

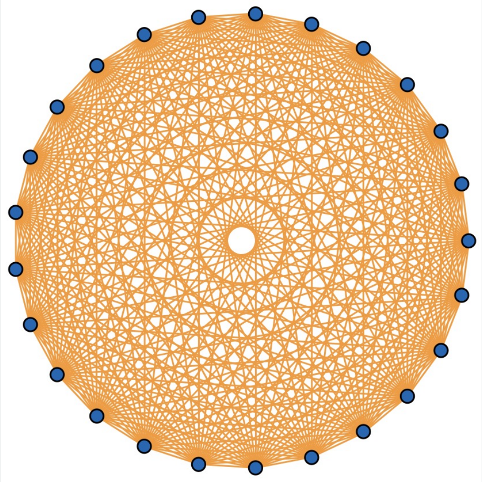
料金例 3 : QPU で Hybrid Job を実行する場合

- AWS US West (N. California) リージョンで実行、GPU を26分実行した場合
  - Task チャージ : 50 iterations x 2 tasks x $0.30 / task = $30.00
  - Shots チャージ : 50 iterations x 2 tasks x 100 shots x $0.00035 / shot = $3.50
  - GPUチャージ : ml.p3.8xlarge usage charges: ($14.688 / 60 minutes) x 26 minutes = $6.36
- 合計 **$39.86**

New!

IonQ Aria

Available Error mitigation with on-demand pricing (over 2,500 shots, \$0.03 per shot)



Trapped Ion 25 qubits
All to all connection

Arbitrary-Angle Mølmer-Sørensen (AAMS) Gate

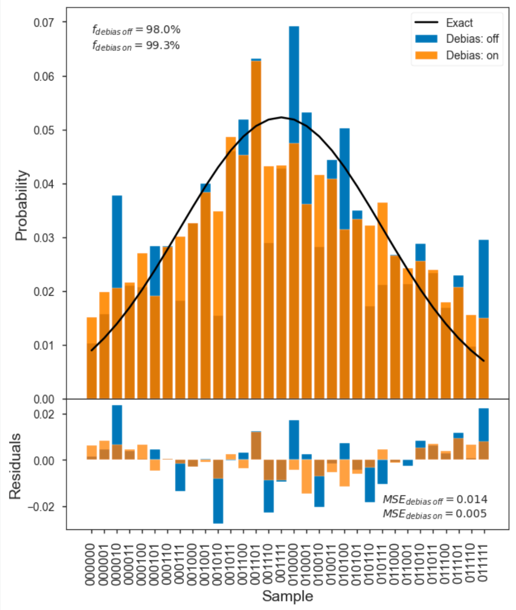
- Users can specify an arbitrary angle of rotation for Mølmer-Sørensen (MS) gate and run a partially entangling MS gate for the performance improvements to certain circuits from reducing the circuit's gate depth.

Error Mitigation with Debiasing

- Debiasing mitigates the effect of coherent errors by randomizing their effect across multiple circuit implementations. This improves the fidelity of the normal distribution generated by the Quantum Circuit Born Machine (QCBM) by 1.3% and the mean squared error (MSE) of the residuals by 3x.

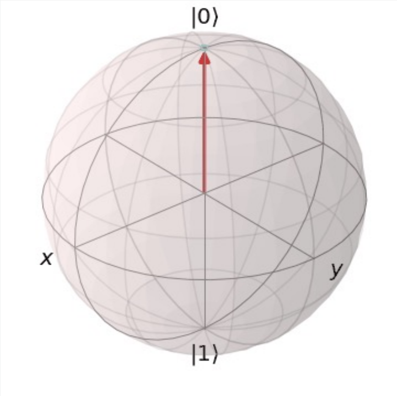
Error Mitigation with Sharpening

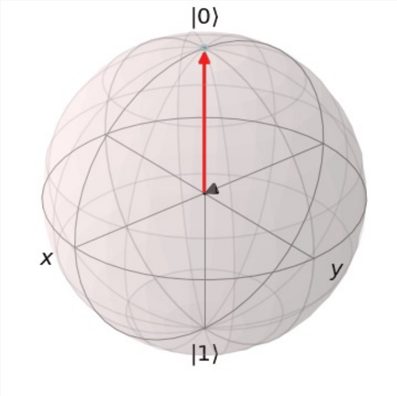
- For circuits with sparse output distribution, it is possible to ignore inconsistent measurements where only the number of times below the threshold value was obtained, and further improvements can be made from the results of debiasing.



Braket Pulse

- Pulse-level qubit control
  - Operation with analog pulses at a lower level access than quantum gates
  - Supported on Rigetti and OQC devices
  - Use-cases: qutrits, error mitigation, analog quantum algorithms
- Open sourced OQpy (OpenQASM 3 in Python)

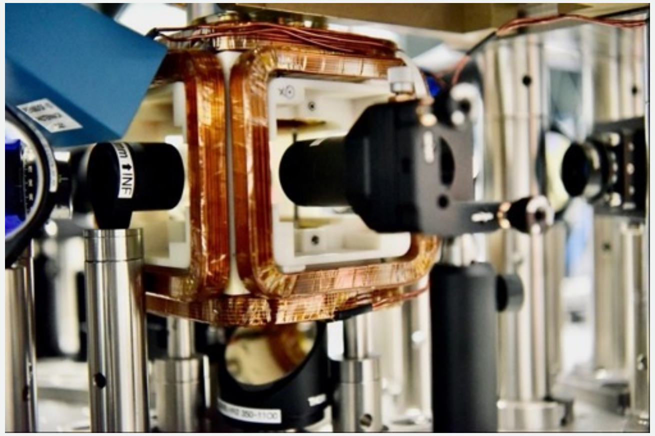




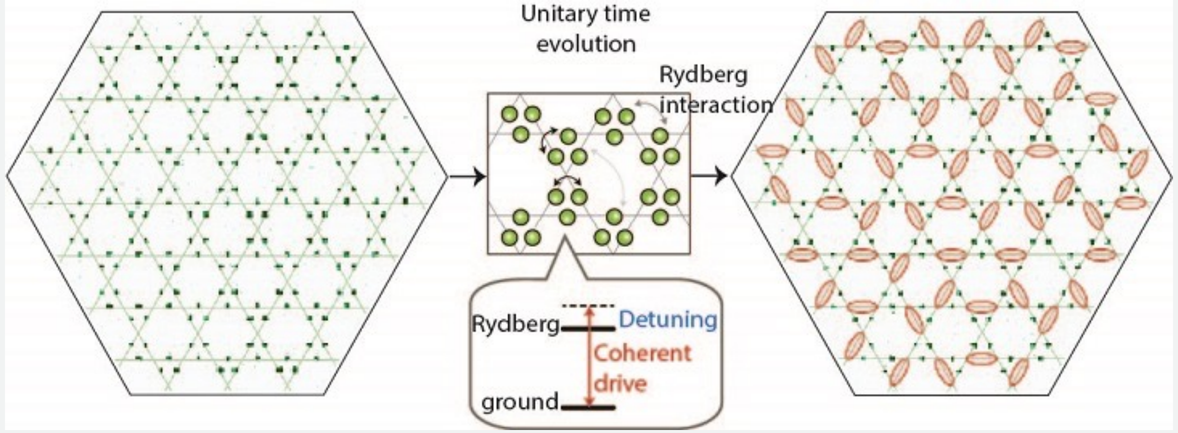
https://docs.aws.amazon.com/braket/latest/developerguide/braket-pulse.html

Amazon Braket に QuEra が追加されました！

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

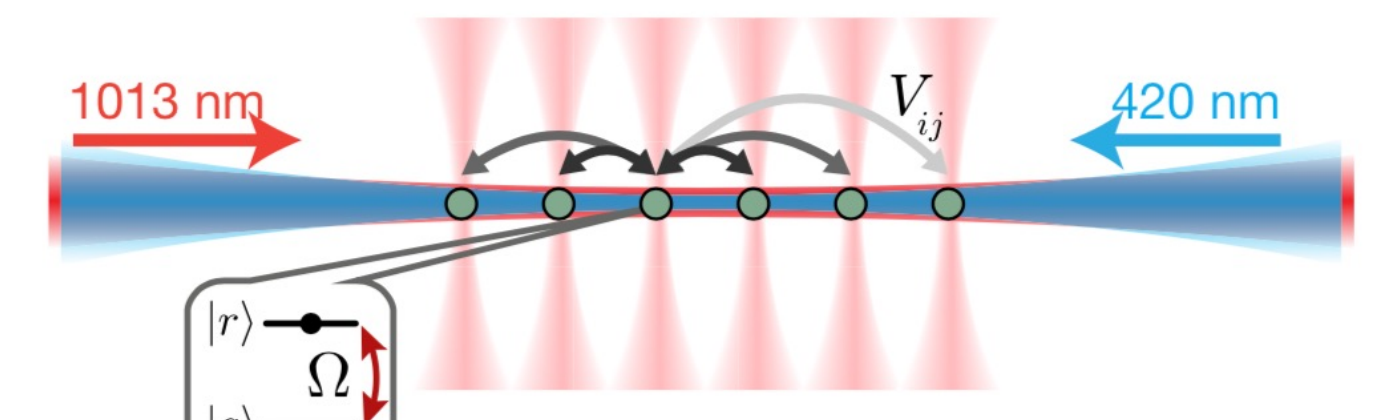


QuEra Computing with
Rydberg atom arrays



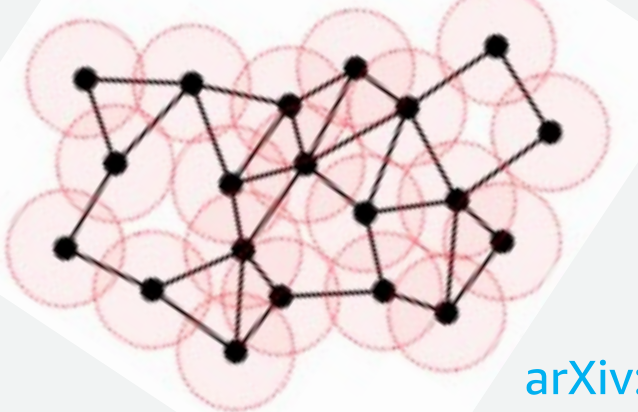
Physics of creating spin-pairs in a frustrated lattice.

What is a neutral atom quantum computer?



arXiv:1707.04334

$$H = \sum_i \Omega \sigma_i^x - \Delta \sigma_i^z + \sum_{i<j} V_{ij} n_i n_j$$



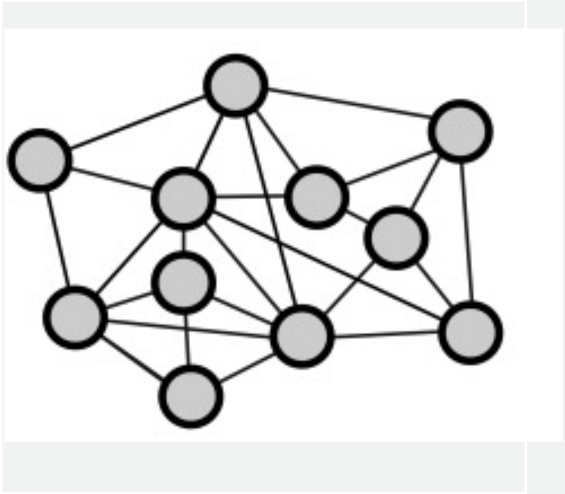
arXiv:2205.08500

Key features

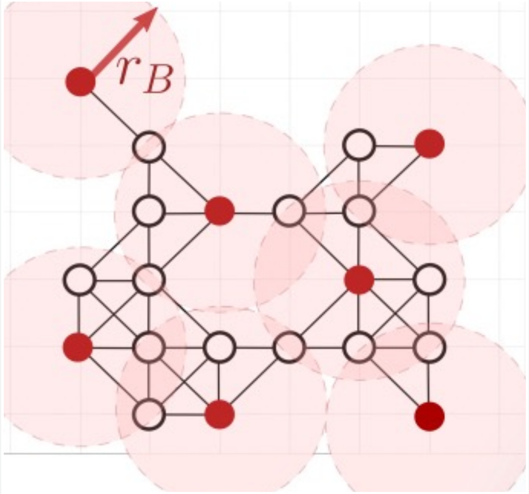
- Individual control of Rydberg atoms via lasers held by a trapping potential in 3D
- Strong Rydberg interactions lead to a unit disk graph connectivity
- Dynamical control of the Hamiltonian parameters

Adiabatic optimization with the QuEra device

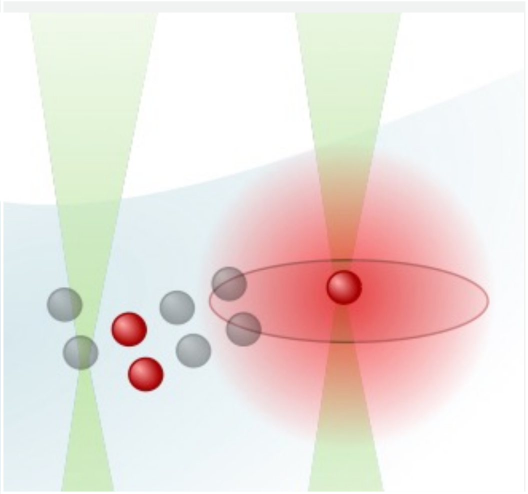
arXiv:2209.03965



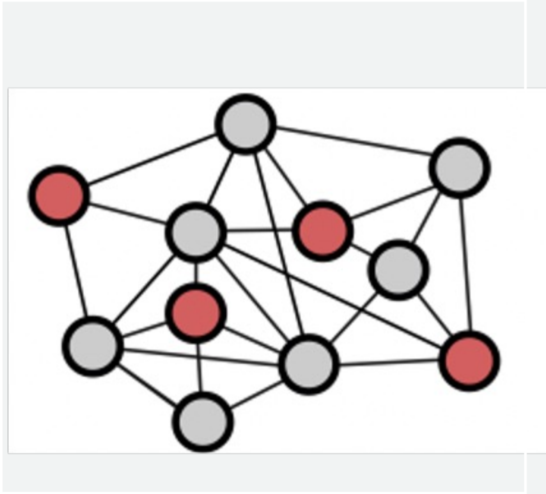
Problem



UD Encoding



MIS on device

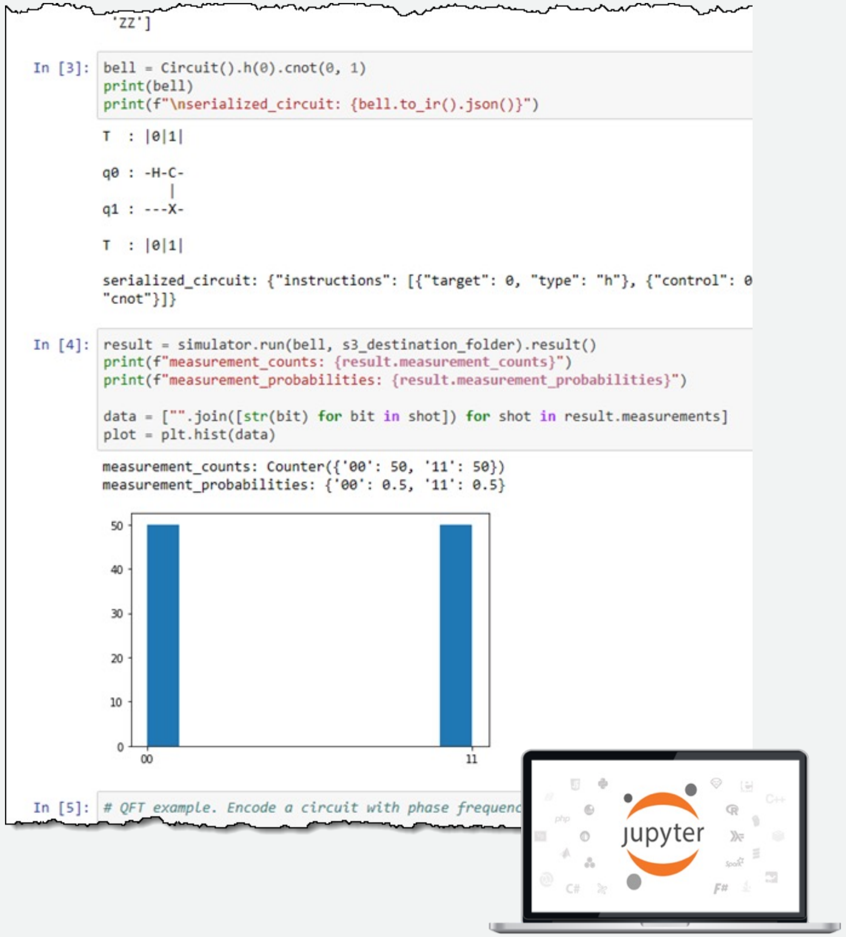


Result

Analog optimization

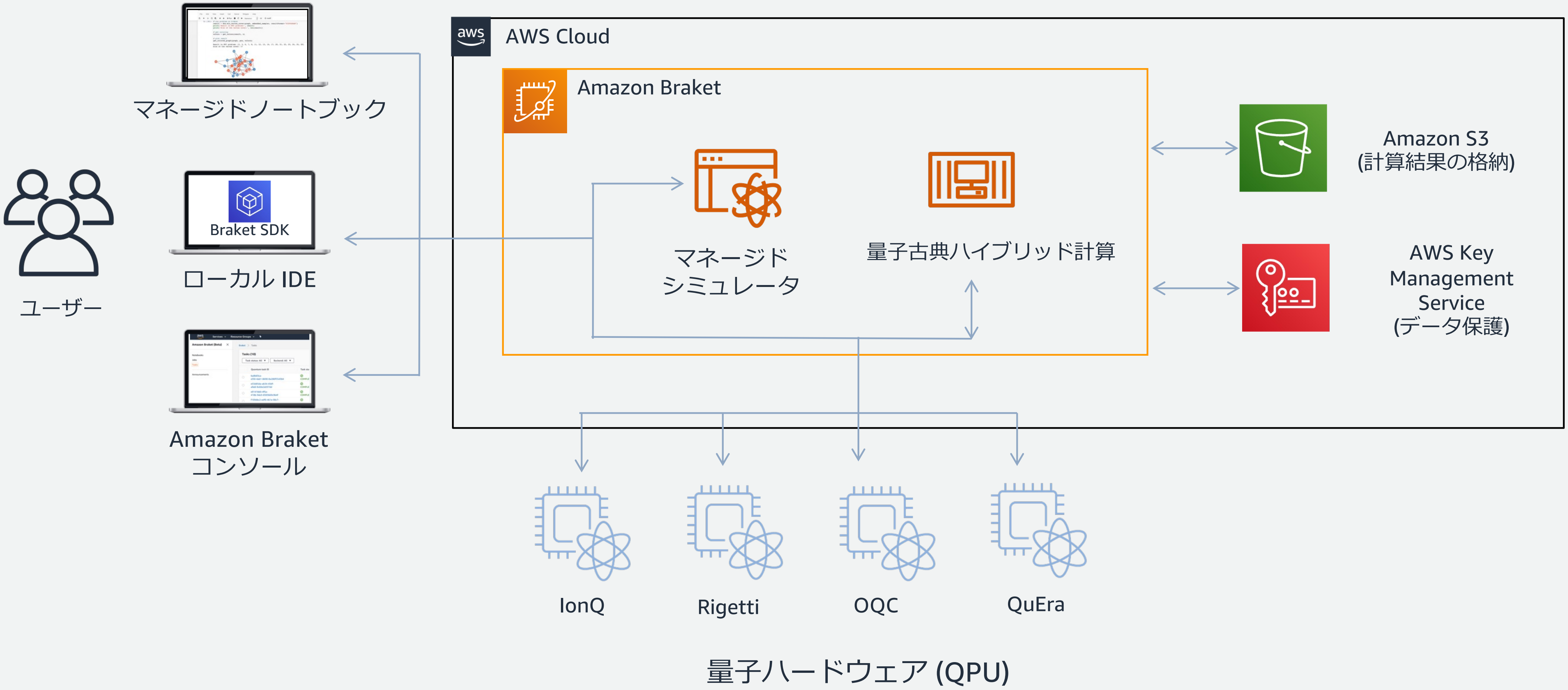
- The problem graph is encoded in a unit-disk (UD) graph
- Laser tweezers are used to position the atoms
- After the anneal the result can be read out

開発環境と Amazon Braket Python SDK



- 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

Amazon Braket アーキテクチャ



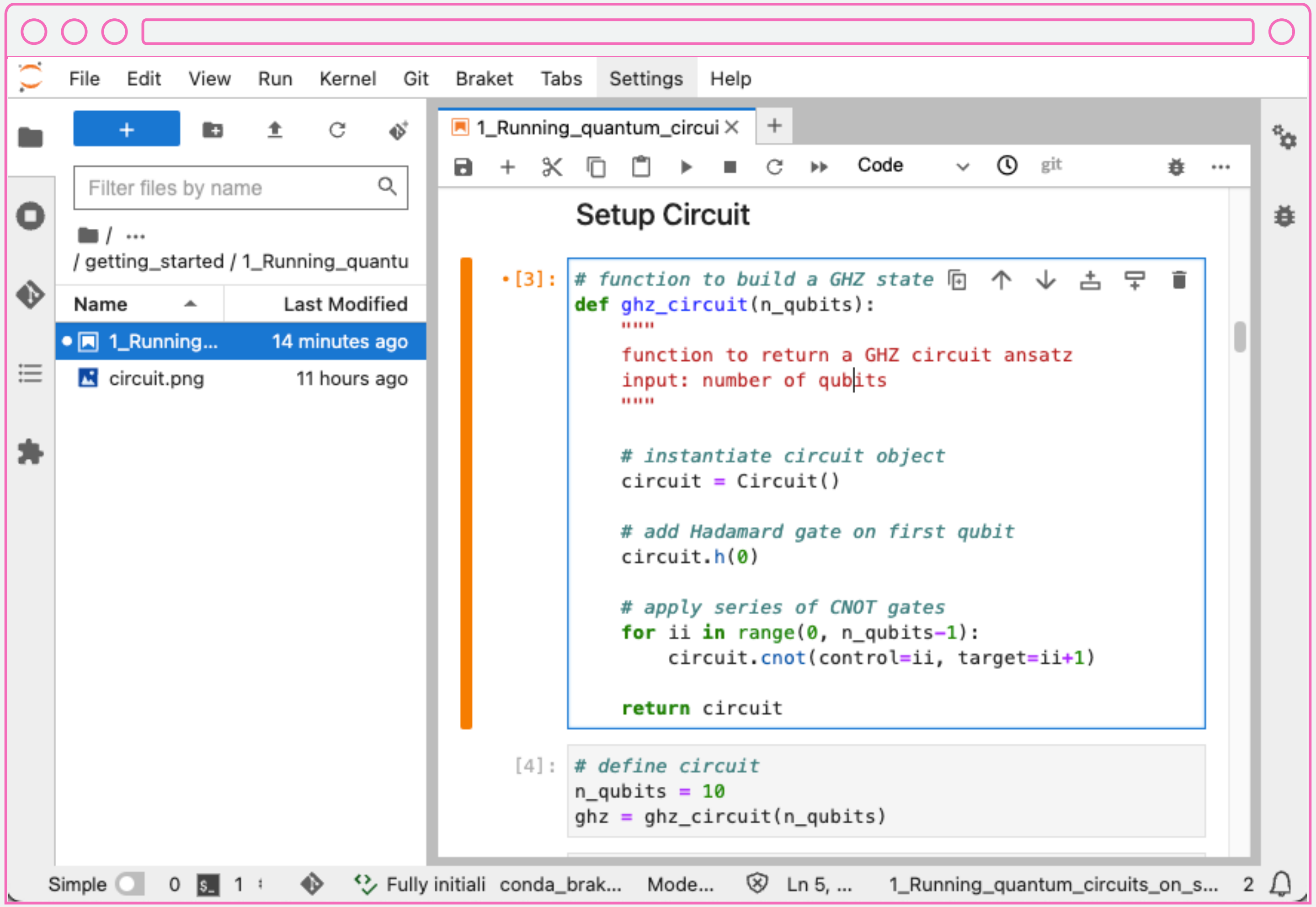
Managed JupyterLab environments

今すぐはじめよう

フルマネージドな開発環境

- プリインストールされたサンプルコードや Amazon Brakt SDK をはじめとした開発者ツール

docs.aws.amazon.com/braket/latest/developerguide



Amazon Braket SDK

オープンソース Python ライブラリ

- 量子回路の設計と構築
- タスクやジョブをデバイスに投入
- 実行状況の追跡と監視

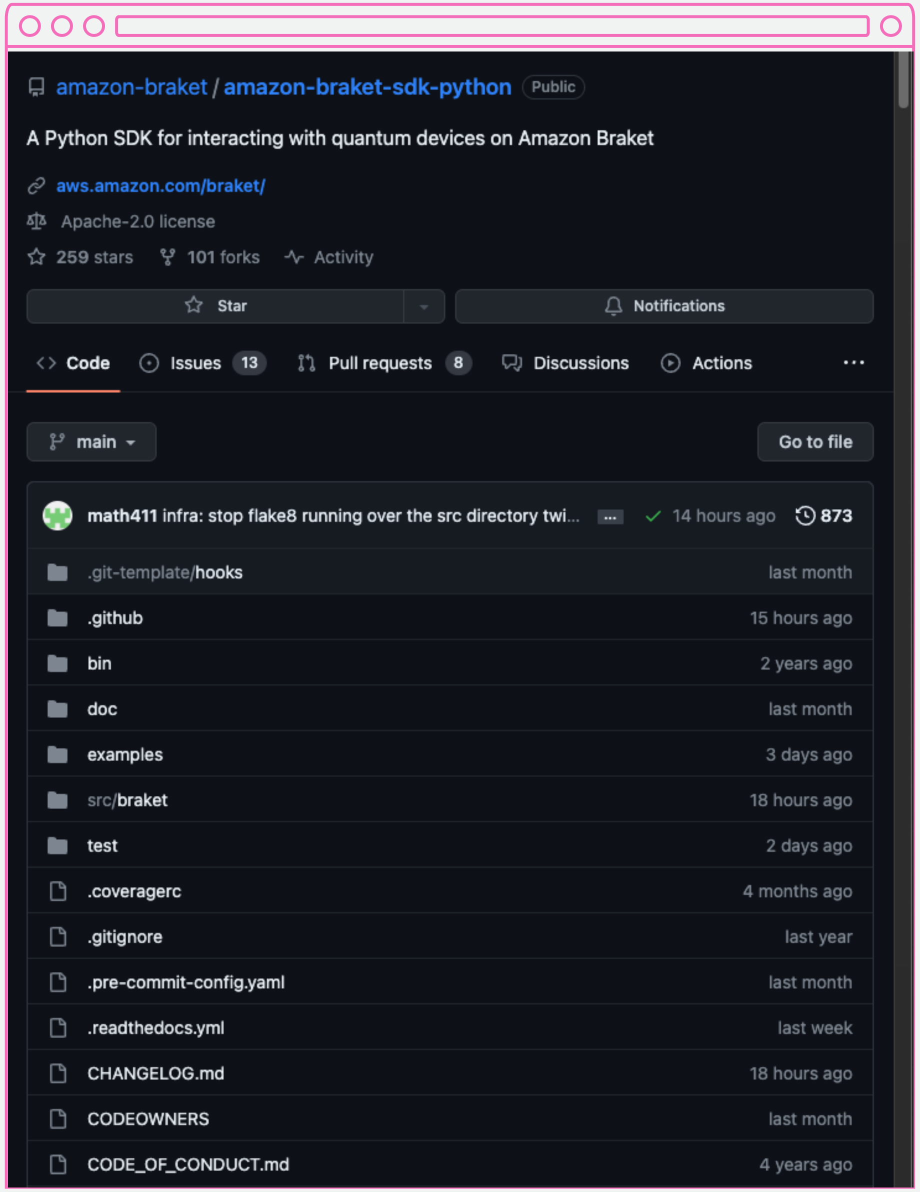
ローカル環境へのインストール

```
pip install amazon-braket-sdk
```

ローカルシミュレータ

- braket_sv
- braket_dm
- braket_ahs

[amazon-braket/amazon-braket-sdk-python](https://github.com/amazon-braket/amazon-braket-sdk-python)



Amazon Braket で計算を実行

aws_bell.py

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

device = AwsDevice("aws_device_ARN")

bell = Circuit().h(0).cnot(0, 1)
print(bell)

task = device.run(bell, shots=1000)
print("Measurement Results")
print(task.result().measurement_counts)
```

- Braket SDKモジュールのインポート
- デバイス（シミュレータ/QPU ）を選択する
- 量子回路等、量子プログラムをコード化する
- 選択したデバイスでプログラムを実行
- 計算完了時に計算結果を取得

Qiskit-Braket Provider

Qiskit開発ツールキットからAmazon Braketコードを実行する

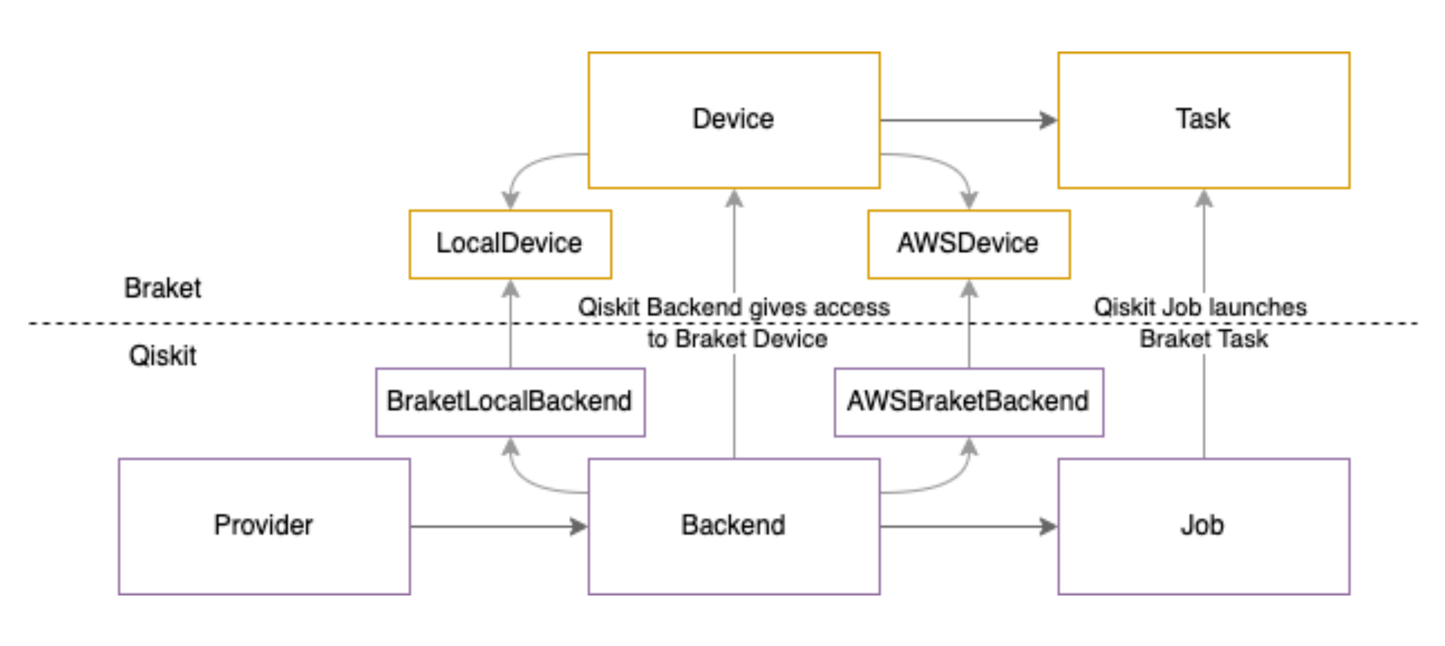
https://github.com/qiskit-community/qiskit-braket-provider

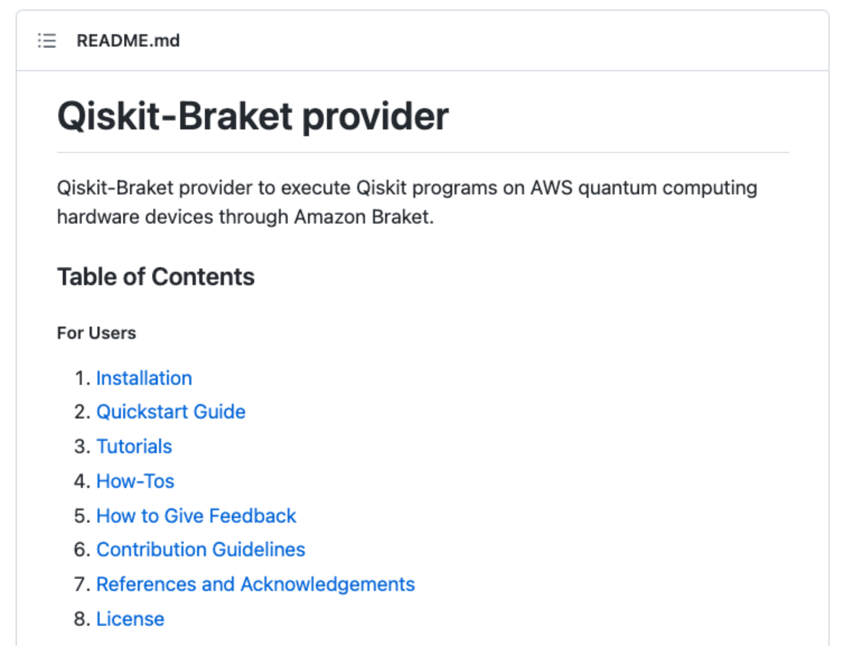
qiskit_connector.py

```
from qiskit_braket_provider import BraketLocalBackend

local_simulator = BraketLocalBackend()
task = local_simulator.run(circuit, shots=1000)

```

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.

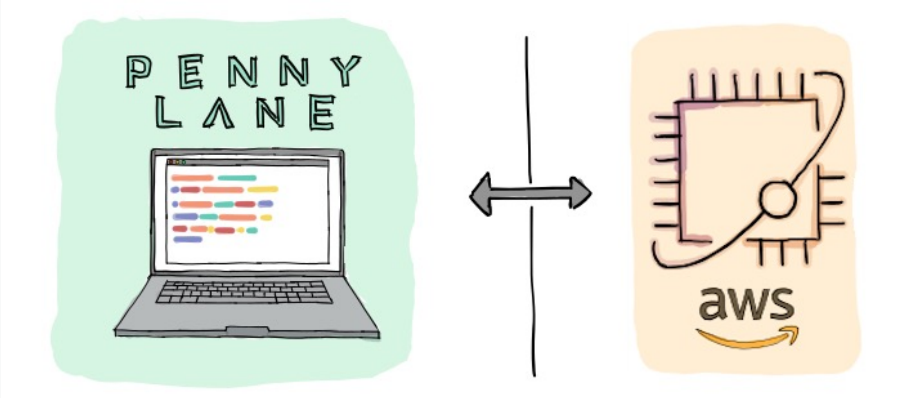
Table of Contents

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/

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.

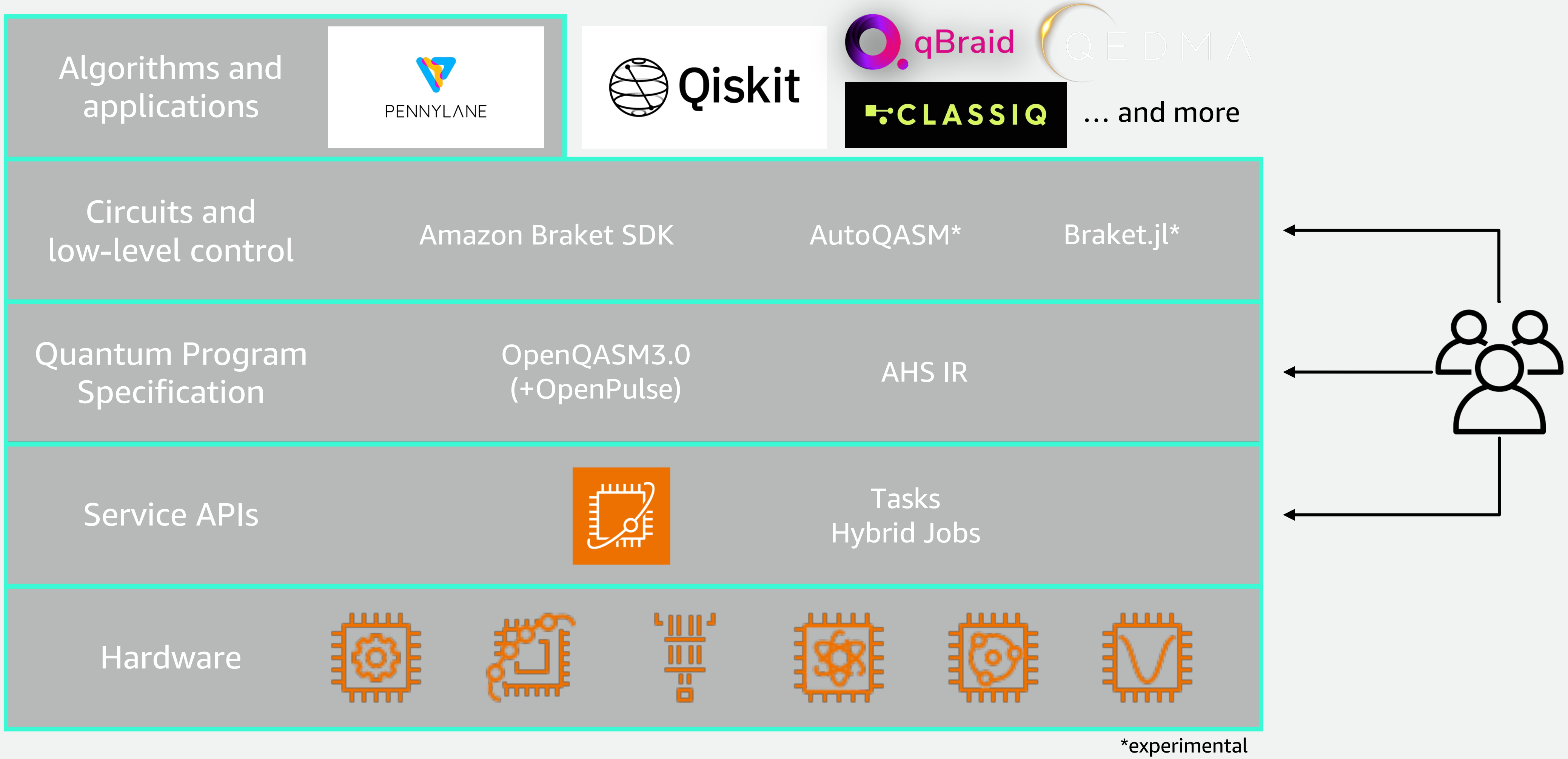
機械学習ツールを量子コンピューティングに

- 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

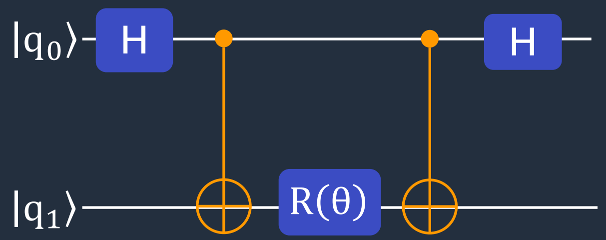
ハイパフォーマンスな計算の実装が簡単に

- 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

Amazon Braket Stack

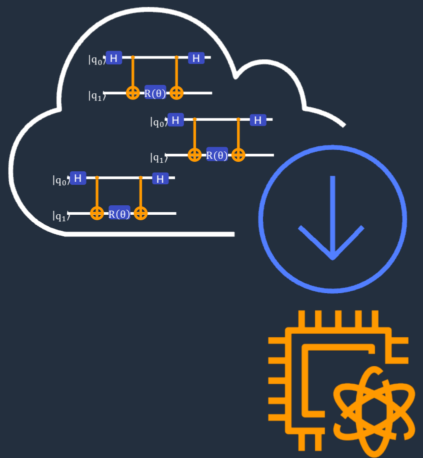


Shots, tasks and jobs



Shot

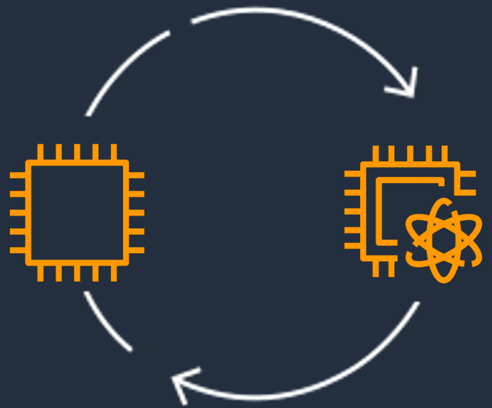
Single circuit execution with measurement instructions on a device



Task

Series of repeated shots on a device

(10s to 10,000s shots per task)



Hybrid job

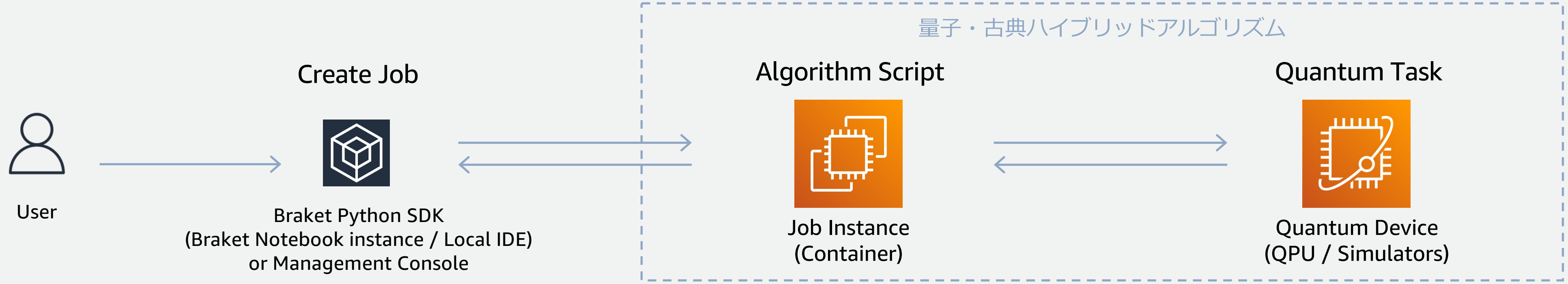
Sequence of classical and quantum compute cycles

(10s to 1,000s of tasks per job)

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なため、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 Hybrid Jobs で実行する VQE

量子・古典ハイブリッドアルゴリズム

User

Create Job

Braket Python SDK
(Braket Notebook instance / Local IDE)
or Management Console

Algorithm Script

Job Instance
(Container)

Quantum Task

Quantum Device
(QPU / Simulators)

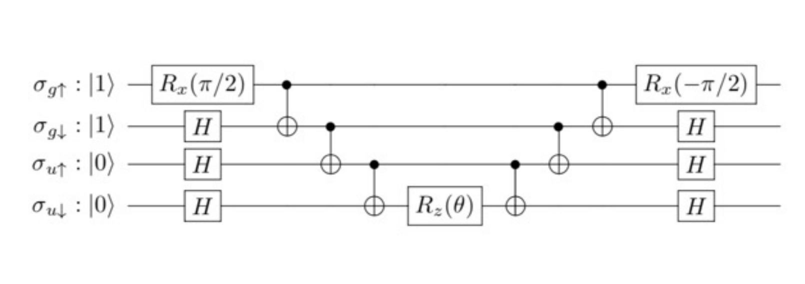
古典計算機によるパラメータ計算

- 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)
```

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)
```

PennyLane ライブラリでVQEを簡単に記述できる

https://github.com/aws/amazon-braket-examples/tree/main/examples/pennylane/3_Quantum_chemistry_with_VQE
https://github.com/aws/amazon-braket-examples/blob/main/examples/hybrid_jobs/2_Using_PennyLane_with_Braket_Jobs/Using_PennyLane_with_Braket_Jobs.ipynb

VQE (Variational Quantum Eigensolver) on Amazon Braket

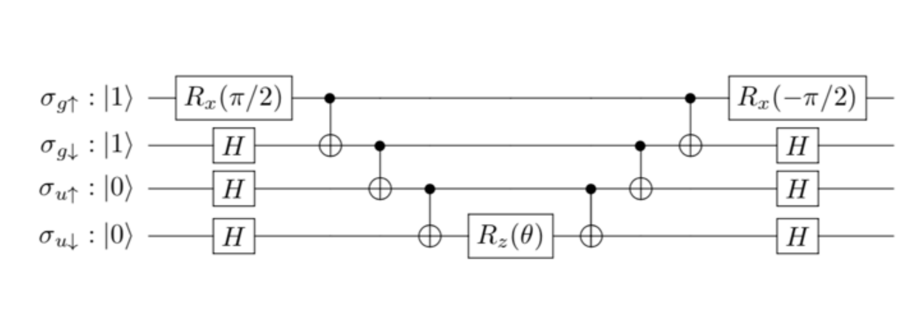
1. 量子ビットによる表現
   Jordan-Wigner を使ったフェルミ粒子から量子ビットへのマッピング

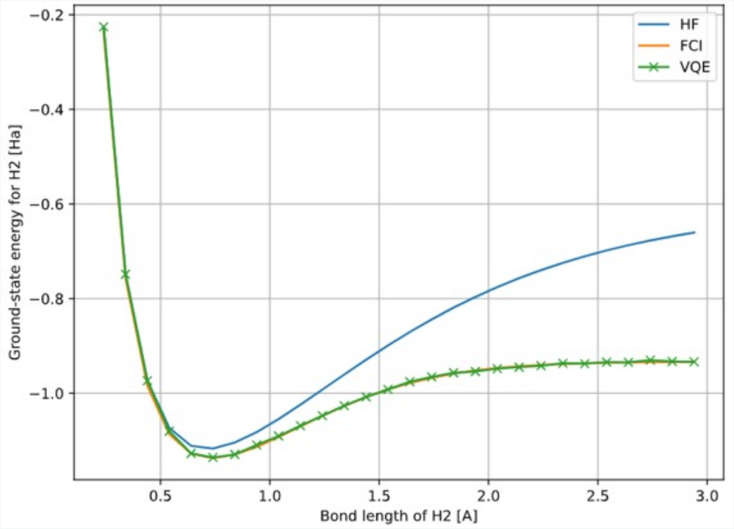
2. Ansatz と初期トライアル関数:
   量子状態 l ψ〉を生成するための量子回路
   UCCSD ansatz とパラメータ θ を用いて量子状態を生成

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 QPU 稼働時間

QPU	Availability UTC	Availability JST（日本時間）
Rigetti Aspen-M-3	Everyday, 04:00 – 06:00 UTC Everyday, 15:00 - 18:00 UTC	Everyday, 13:00 – 15:00 JST Everyday, 0:00 – 5:00 JST
IonQ (Aria, Harmony)	Weekdays, 12:00 - 03:00 UTC	Tuesday-Saturday, 21:00-12:00 JST
Oxford Quantum Circuits - Lucy	Weekdays, 09:00 - 12:00 UTC	Tuesday-Saturday, 18:00 –21:00 JST
QuEra	Monday 01:00 - 23:59:59 UTC Tuesday 00:00 – 12:00 UTC Wednesday 00:00 - 12:00 UTC Friday 00:00 - 23:59:59 UTC Saturday 00:00 – 23:59:59 UTC Sunday 00:00 – 12:00 UTC	Tuesday 10:00 – 08:59:59 JST Wednesday 10:00 - 21:00 JST Friday 10:00 - 21:00 JST Saturday 10:00 – 08:59:59 JST Sunday 10:00 – 08:59:59 JST Monday 10:00 - 21:00 JST

デバイスの稼働時間外にもジョブの実行は可能、
実行順に待ち行列に並び、稼働時間になったら順次ジョブが実行され結果を取得可能

お客様事例

BMW Group ロボットアームのトラジェクトリ計画を高速化 AWS プロフェッショナルサービスによる支援

BMW GROUP

Customer: BMW Group
Industry: Automotive
Country: Germany
Employees: 129,000
Website: www.bmw.com

Challenges

- Finding an optimal robot-motion production plan for polyvinyl chloride (PVC) sealing of the car body is a very hard combinatorial optimization problem.
- The goal is to process all seams within a minimal time span.
- Automation of robot program generation, increasing the efficiency of production facilities, and streamlining manufacturability tests during the early design phase.
- BMW wanted to build internal capabilities and expertise in quantum computing.

Solution

- Implementation of the nature-inspired Biased Random Key Genetic Algorithm (BRKGA) providing a comprehensive solution to the PVC optimization use case.
- Extended Bayesian hyperparameter optimization on hosted Amazon SageMaker notebooks.
- Implementation of a quantum-ready PoC on Amazon Braket, using quantum annealing and hybrid quantum-classical approaches.
- Proposal for a hybrid stacked model where myriad of solvers (such as quantum annealer) seeds input to BRKGA.

Results

- Enabler for increased efficiency, with about 10% improvement in projected runtime for sealing an entire car body in production.
- BMW Group gets access to a general purpose, portable optimization algorithm with endemic encoding for any sequencing problem.
- Deeper understanding where and when quantum computing will become relevant for BMW's business.

The Goldman Sachs Group, Inc.
量子内点法とポートフォリオ最適化：エンドツーエンドのリソース分析

Customer: The Goldman Sachs Group, Inc.
Industry: Financial Services
Country: United States
Segment: Large Enterprise
Website: www.goldmansachs.com/

Customer Profile:

The Goldman Sachs Group, Inc. (GS) is an American multinational investment bank and financial services company headquartered in New York City. It offers services in investment management, securities, asset management, Prime brokerage, and securities underwriting.

Challenges

- Understand the utility of established quantum algorithms for solving real-world financial use cases.
- Carefully quantify the quantum computational resources required for portfolio optimization.
- Build on existing algorithms to optimize performance and reduce computational overhead.
- Perform a true end-to-end resource analysis from problem input to problem output.

Solution

- Map the portfolio optimization problem to an instance of second order cone programming (SOCP).
- Use a quantum algorithm for SOCP to solve the problem, which is based on a quantum interior point method (QIPM).
- Dive deep into the algorithm and perform a careful accounting of all quantum computational resources required at each step.
- Develop new algorithmic optimizations and quantum circuits for implementing each step of the QIPM.

Results

- Two publications:

1. Quantum Resources Required to Block Encode a Matrix of Classical Data (arXiv: 2206.03505).
2. End-to-end resource analysis for quantum interior point methods and portfolio optimization (arXiv: 2211.12489).

- Detailed accounting of the number of logical qubits, the T-depth, and the T-count (i.e., the computationally expensive quantum resources) required to perform portfolio optimization.

Amazon Braket を使った事例

豊田中央研究所

「量子コンピュータ×量子化学計算の産業応用に向けた取り組みとAmazon Braketの活用事例」

- 産業応用のための量子化学計算
- 分子の基底状態を計算する方法を Amazon Braket の NISQ(Noisy Intermediate Scale Quantum) で検証
- 励起状態を計算する Variational Quantum Deflation (VQD) の研究

大阪大学

「量子コンピュータ x 量子化学計算とAmazon Braket」

- Amazon Braket を使った VQE のノイズ耐性調査（高い軌道エネルギー状態を考慮した計算）
- 量子化学計算ツール QunaSys Qamuy のAmazon Braket 連携の紹介
- Amazon Braket の使用量モニタリングツールの紹介

日本経済新聞社/お茶の水女子大学

「産学共同研究での Amazon Braket 活用」

- Amazon Braket D-Wave を使ったニュースレターにおける読者に合わせた記事の個別最適化
- D-Waveを使うと、SA (Simulated Annealing) に比べて短い時間で多くのトライアルができ、より高いスコア

https://aws.amazon.com/jp/blogs/news/event-report-braket-case-study-2022/

Amazon Braket Digital Badge

AWSを通じて追加費用なしで
オンデマンド配信される
量子コンピューティングコース

Braket上で最初の量子回路を
構築し、実行する

https://explore.skillbuilder.aws/learn/learning_plan/view/1990/amazon-braket-knowledge-badge-readiness-path-amazon

Thank you!

Amazon Braket Technologies Blog

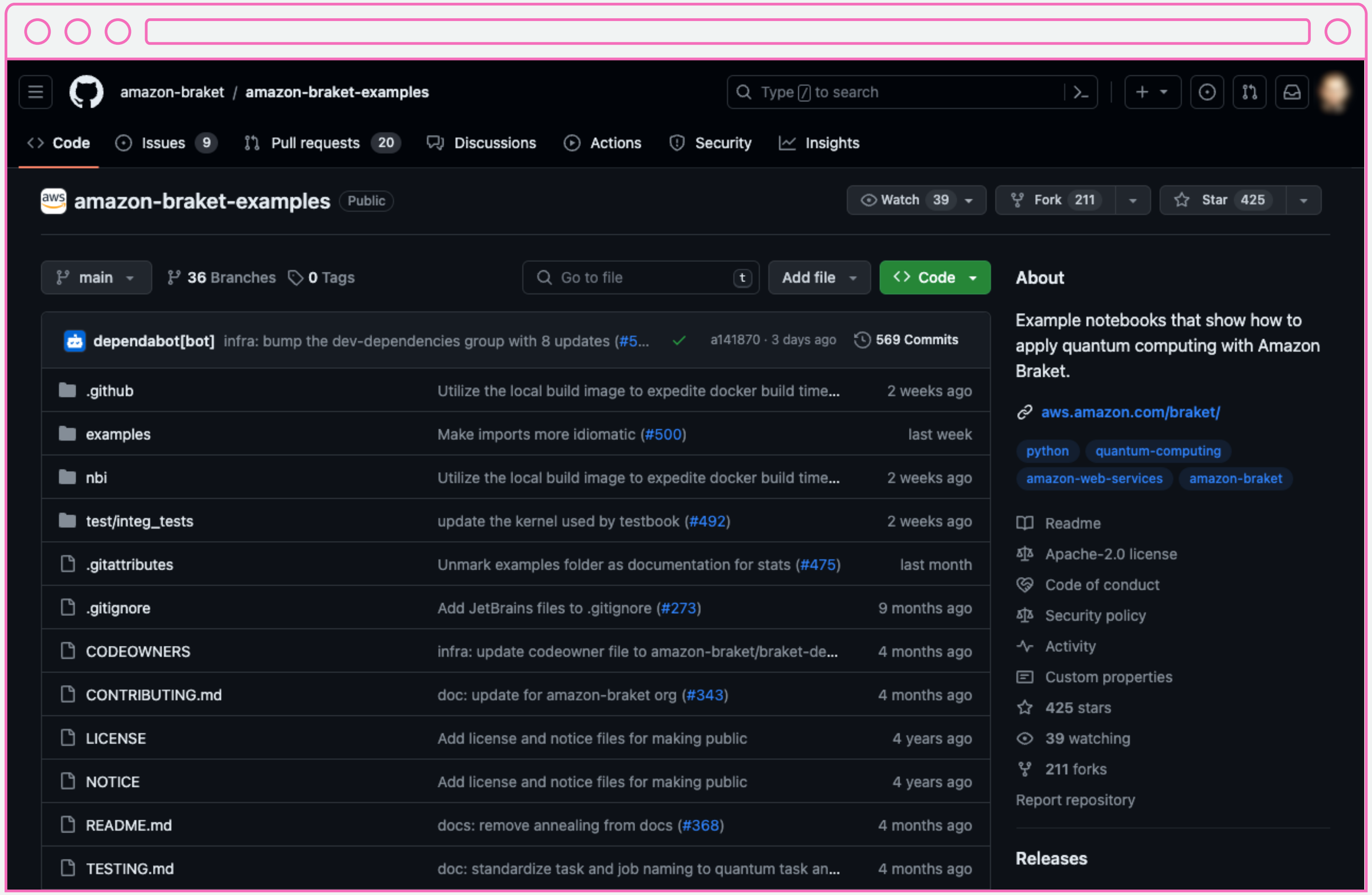
https://aws.amazon.com/jp/blogs/quantum-computing/

Amazon Braket GitHub Samples

https://github.com/aws/amazon-braket-examples

Amazon Braket Examples

https://github.com/amazon-braket/amazon-braket-examples



AWS Quantum Blog Channel

https://aws.amazon.com/blogs/quantum-computing/