

量子コンピュータを用いたプログラミング方法を学ぼう

DEVEL 比嘉 恵一朗

自己紹介と概要

【自己紹介】

氏名:比嘉恵一郎(ひがけいいちろう)
九州大学大学院数理学府数理学専攻修士課程卒
専門:量子情報・量子アルゴリズム
著書:IBMQで学ぶ量子コンピュータ
DEVEL株式会社・GLEAP株式会社代表
日本量子コンピューティング協会 正会員



【本講演の概要】

量子コンピュータをどのようにプログラムで操作するか、またどのような種類があるのかを知らない方は多いかもしれません。本講演では、実際にいくつかの量子コンピュータを動かしながら、プログラミングの流れを具体的に学んでいきます。

【今回使用するもの】

・Google colab
→実際に量子計算を実行します！！

【コメント】

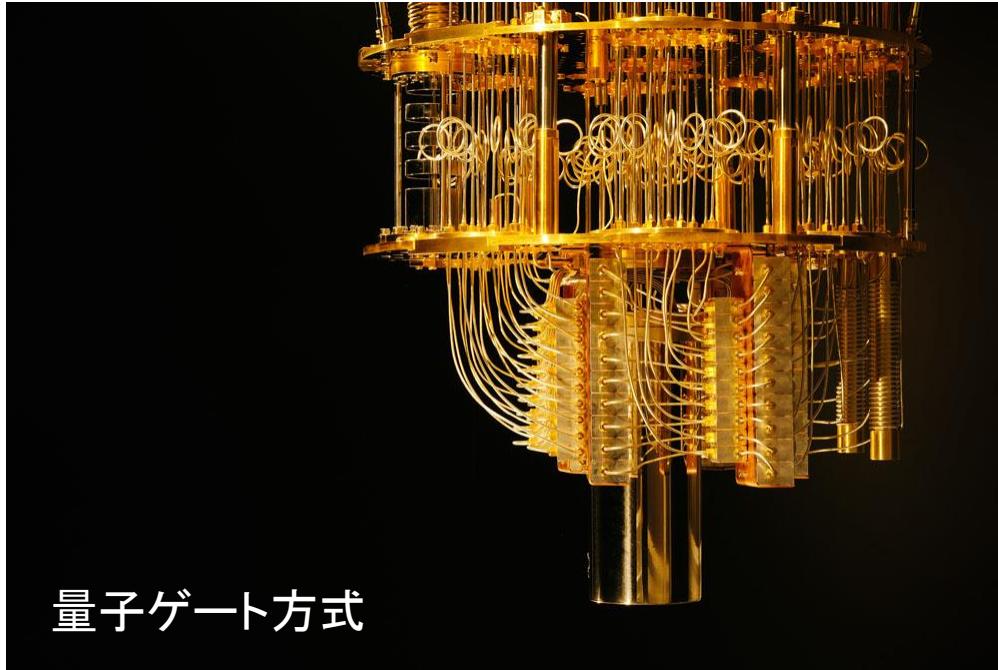
ここでは数式を一切使用せずに講義を行います。
量子コンピュータの計算の雰囲気味わってもらえたら幸いです。

目次

1. 量子コンピュータを実行する流れを理解しよう。
→量子コンピュータを動かすとは？
2. どのような量子コンピュータを使うことができるか見てみよう。
→量子コンピュータの方式は？どこで量子コンピュータを使うことができる？
3. プログラミングで量子コンピュータを動かす方法を理解しよう。
→量子コンピュータを動かすためのプログラミング言語って何？
4. 実際に量子コンピュータを動かしてみよう。(量子アニーリング編)
5. 実際に量子コンピュータを動かしてみよう。(量子ゲート編)
6. 実際の課題に対して量子コンピュータをどのように適応させるかを考えよう。
→量子コンピュータを使うためにはどういった課題設定を考えたら良い？
7. まとめ

量子コンピュータを実行する流れを理解しよう。

量子コンピュータを動かすとは??



量子ゲート方式

引用元: https://newsroom.ibm.com/media-center?keywords=quantum#gallery_gallery_0:21747

量子コンピュータは通常のコンピュータとは異なりかなり大きい。
どうやってプログラムを用いて量子コンピュータを操作する？

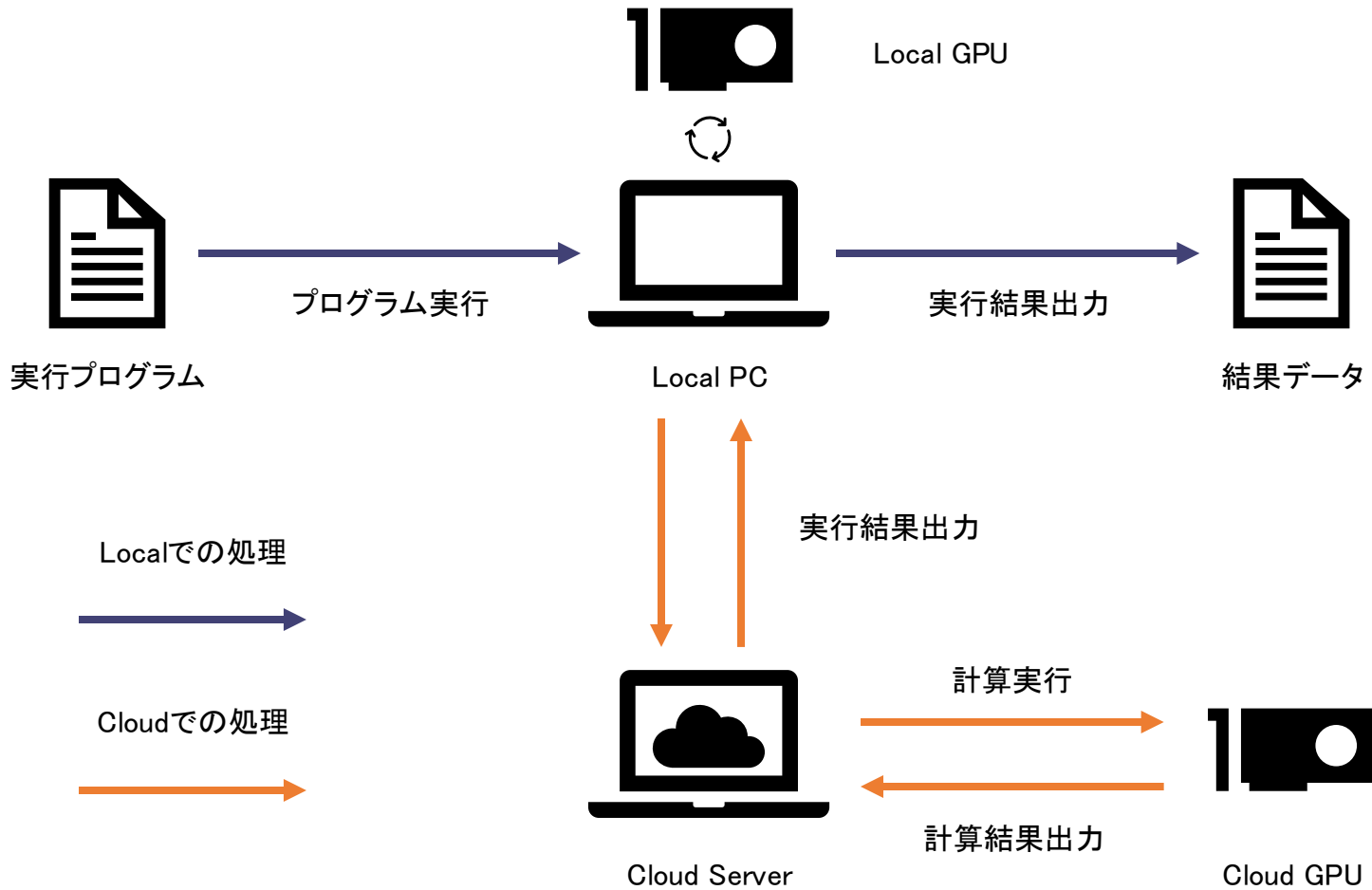


量子アニーリング方式

引用元:
https://dwavejapan.com/app/uploads/2020/10/DWave_Advantage_System-web-large.png

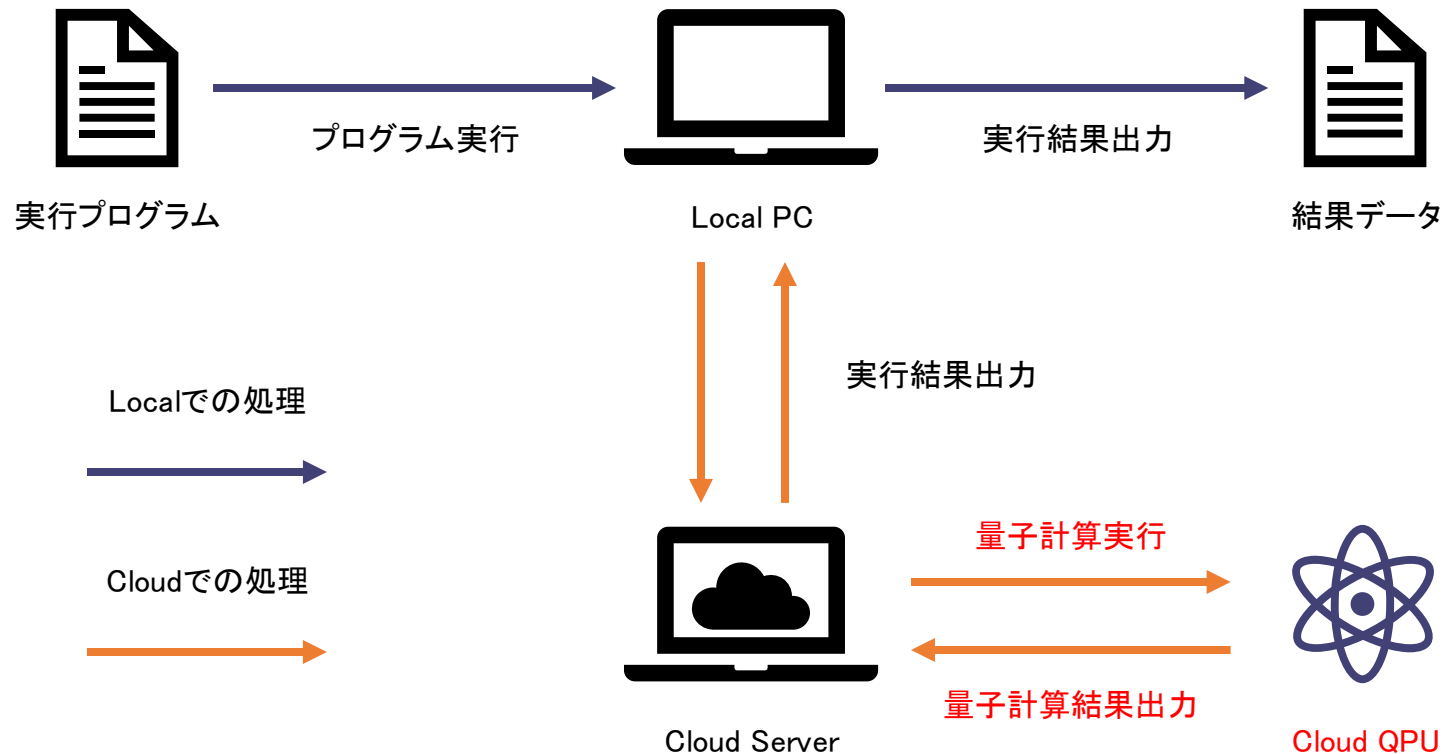
量子コンピュータを実行する流れを理解しよう。

古典コンピュータでプログラムを実行する場合。(Local or Cloud)



量子コンピュータを実行する流れを理解しよう。

量子コンピュータでプログラムを実行する場合。(Cloud)



手元の量子コンピュータというのは基本的には存在しないので、基本はクラウドのサーバ上で動かす。量子計算の部分のみを量子コンピュータに任せるので、イメージとしてはGPUに近い。

量子コンピュータの方式は2種類存在する。

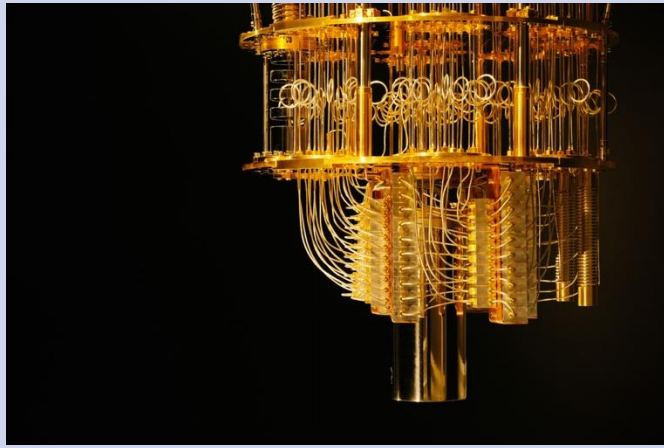
量子アニーリング方式



D-Wave Systems社

組み合わせ最適化問題に特化した量子コンピュータ

量子ゲート方式



IBM社

一般的に量子コンピュータといふところを指す。超電導回路や半導体量子ドットや光、イオン、冷却原子(中性原子)など、様々な実現手法が提案されている。

これらの量子コンピュータは基本クラウドを経由して利用を行う。
色々な会社が量子コンピュータをクラウド上で動かせるようにサービスを提供している。

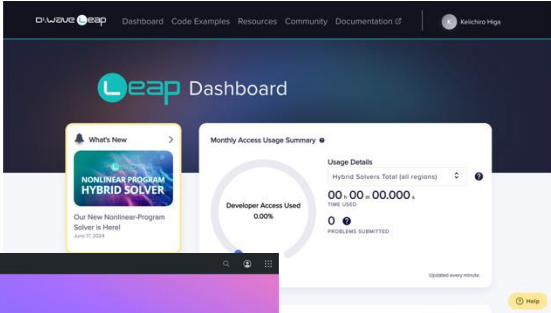
どのような量子コンピュータを使うことができるか見てみよう。

誰でも使えるクラウドサービスをここでは紹介する。

【量子アニーリング】

・D-Wave Leap

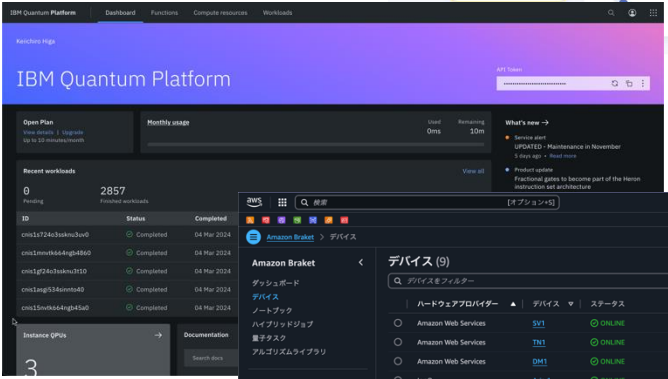
<https://cloud.dwavesys.com/leap/login?next=/leap>



【量子ゲート】

・IBM Quantum Platform

<https://quantum.ibm.com/>



・Amazon Bracket

<https://aws.amazon.com/jp/braket/>



・Azure Quantum

<https://azure.microsoft.com/ja-jp/products/quantum>

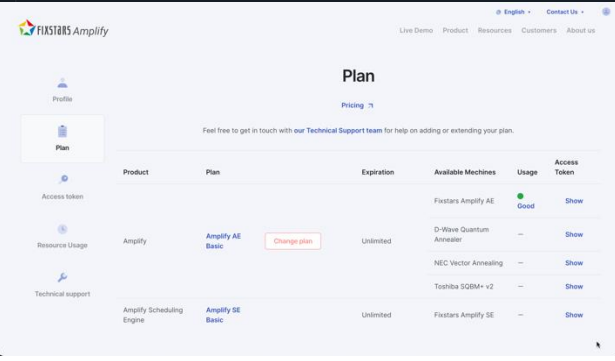
・Google Cloud Platform

<https://console.cloud.google.com/?hl=ja>

【擬似量子アニーリング】

・Fixstars Amplify

<https://amplify.fixstars.com/en/>



プログラミングで量子コンピュータを動かす方法を理解しよう。 **DEVEL**

手元のPCもしくは使用しているサーバ上で実行するプログラムはどのようなもの??

【プログラミング言語】

量子コンピュータを動かすためのプログラミング言語は主に以下のものが存在する。

Python

C++

C#

Q#

先ほど紹介したサービスは全てPythonのプログラムを実行することが可能。
→特に選択する理由がなければ基本的にはPythonでプログラムを作成すべき。

【量子フレームワーク】

言語そのものは少ないが、デバイスあるいはサービスごとに対応しているフレームワークが異なる。
したがって、基本的にはデバイスごとに量子回路を設計するプログラムも異なる。

例)

D-Wave Leap → dimod, IBM Quantum → Qiskit, Fixsters Amplify → Amplify SDK
→各フレームワークの記法を学ぶ必要があるため、学習コストが大きい。

次のページに量子フレームワークの一覧を紹介。

プログラミングで量子コンピュータを動かす方法を理解しよう。 **DEVEL**

量子フレームワーク	言語	量子ゲート	量子アニーリング	開発企業・団体
Qiskit	Python	○	-	IBM
Cirq	Python	○	-	Google
pyQuil	Python	○	-	Rigetti
Quantum Development Kit	C#, Q#	○	-	Microsoft
dimod	Python	-	○	D-Wave System
Amplify SDK	Python	-	○	Fixstars
Amazon Braket Python SDK	Python	○	○	Amazon
Intel Quantum SDK	C++	○	○	Intel
openqemist	Python	○	-	1Qbit
ProjectQ	Python	○	-	ETH Zurich
PennyLane	Python	○	-	Xanadu
strawberryfields	Python	○	-	Xanadu
FlamingPy	Python	○	-	Xanadu
OpenJij	Python	-	○	Jij

まず始めに量子アニーリング方式のデバイスとして、D-Waveの量子コンピュータを使ってみます。

【量子計算の流れ】

量子アニーリング方式の量子計算を実行するためには以下の流れで行います。

1. 問題をQUBOに変換する。
2. 1で作成したQUBOをもとにアニーリングを実行する。
2. 実行結果をもとに評価を行う。

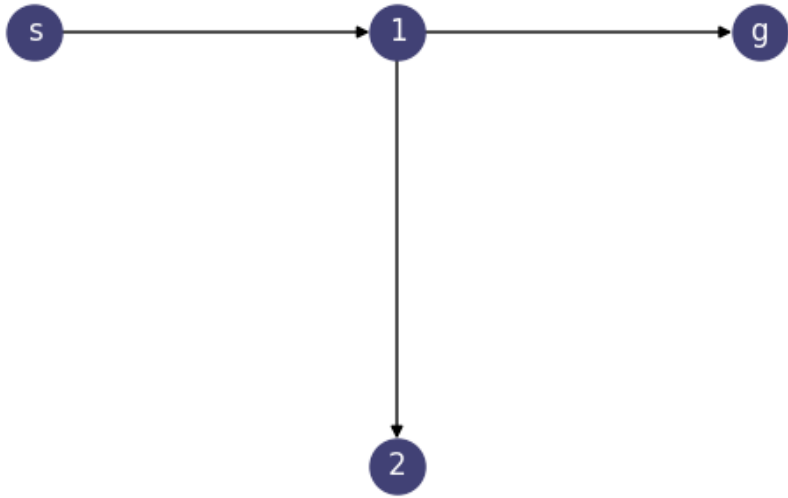
今回は以下の簡単なグラフを用いて、最短経路問題を解いていきます。

【グラフ】

S: スタート地点、G: ゴール地点
各経路の長さは全て等しい。

【問題設定】

S → Gへ向かう経路の内、最短なものを考える。

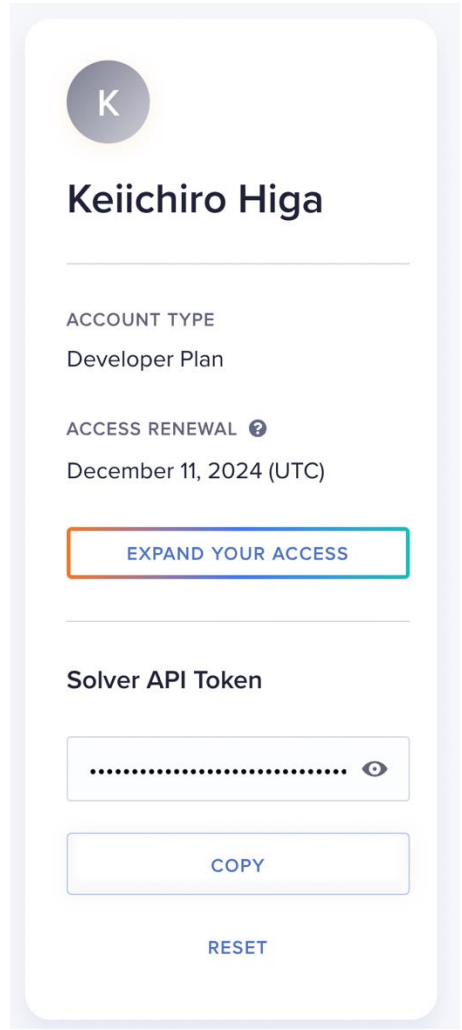
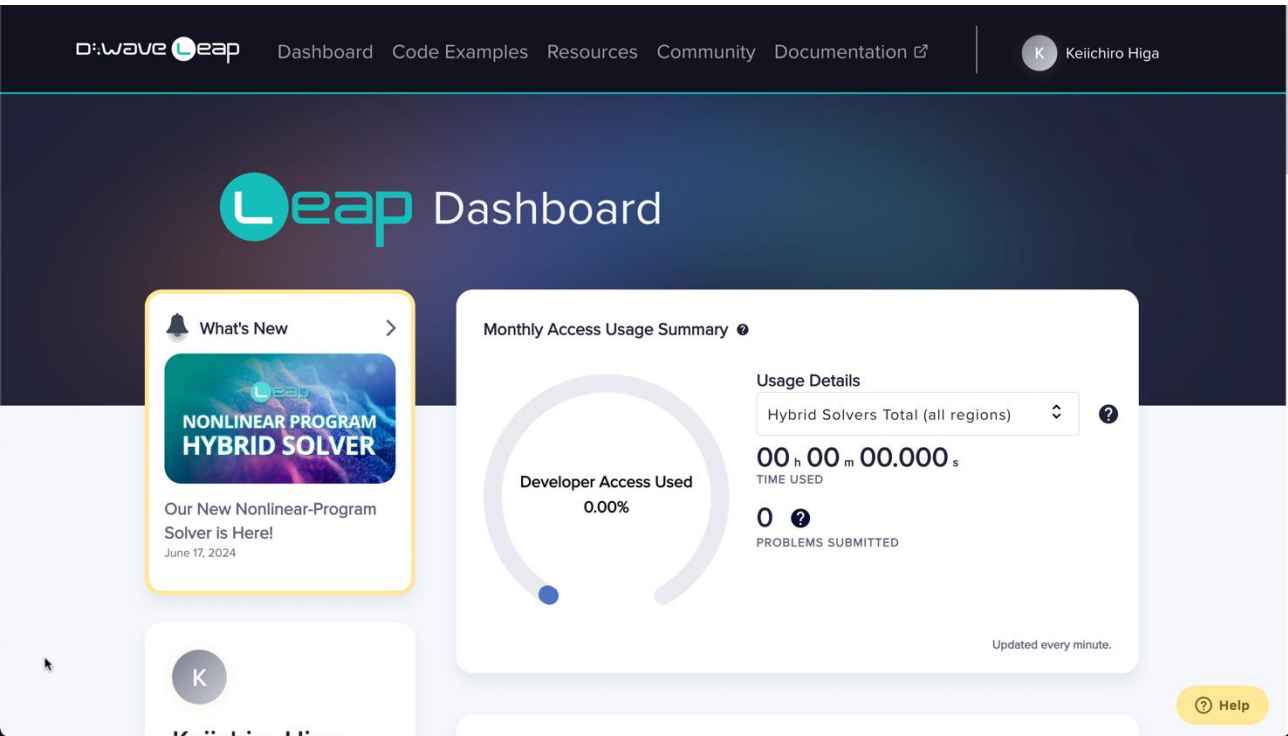


Google colaboratory での作業に移行します。

今回はその中でも量子アニーリングマシンとして、Dwaveの量子コンピュータを使ってみます。

以下のサイトよりDwaveにログイン

<https://cloud.dwavesys.com/leap/login?next=/leap>



ここに実機を動かすために必要なTokenがあります。→

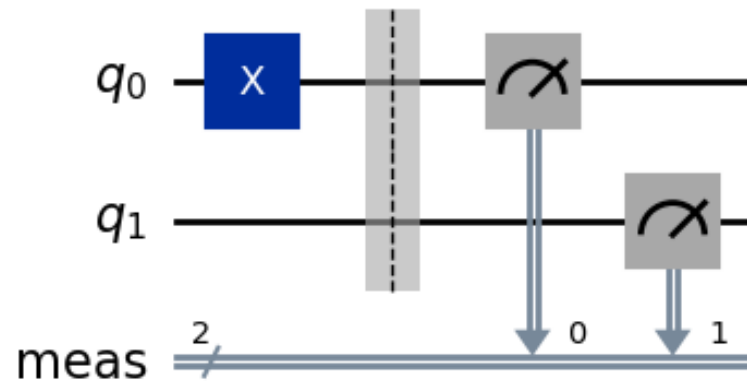
次に量子ゲート方式のデバイスとして、IBMの量子コンピュータを使ってみます。

【量子計算の流れ】

量子ゲート方式の量子計算を実行するためには以下の流れで行います。

1. 量子ビットの数を設定して量子回路の初期状態を準備する。
2. 1で準備した量子回路に量子ビットゲートを作用させる。
3. 2で作成した量子回路を用いて量子計算を実行する。
4. 実行結果をもとに評価を行う。

今回は以下の簡単な量子回路を作成しながら、量子計算操作を一つずつ説明していきます。



Google colaboratory での作業に移行します。

今回はその中でも量子ゲートマシンとして、IBMの量子コンピュータを使ってみます。

以下のサイトよりIBMQにログイン

<https://quantum.ibm.com/>

ここに実機を動かすために必要なTokenがあります。

実際に量子コンピュータを動かしてみましたが、現実の社会課題に対してどのように当てはめたら良いのだろうか？

【量子アニーリング方式】

組み合わせ最適化問題に特化したデバイスなので、最適化できる課題を考えなければならない。
→組み合わせ最適化問題とは、様々な制約の下で多くの選択肢の中から、ある指標(価値)を最も良くする変数の値(組合せ)を求めること。

複数のパターンの中から最も良いものや条件を考えたい場合に活用できる可能性が高い。

【量子ゲート方式】

汎用計算ができるとはいえ、実行の流れではGPUと同じイメージだと述べた。

GPUは画像処理装置としての高い計算処理能力を備えており、単純なタスクの並列処理に優れていた。

→GPUでは大規模な計算部分やタスクの並列処理を行うため、実行イメージが同じ量子コンピュータの場合でも同様に扱うべき。

古典計算処理の中にある大規模計算部分を処理したい場合に活用できる可能性が高い。
(組み合わせ最適化問題においてはアニーリング方式でも同様のことが言える。)

まとめ

- ・基本的に量子コンピュータはクラウドのサーバ上で動かす。
量子計算の部分のみを量子コンピュータに任せるので、イメージとしてはGPUに近い。
- ・方式としては量子アニーリング方式と量子ゲート方式の2種類だが用いるためのサービスは多く存在する。
- ・量子コンピュータを動かすためのプログラミング言語は限られているが、デバイスやサービスに依存したフレームワークによって量子計算の書き方が異なる。
- ・D-Wave Leap と IBM Quantum で量子アニーリング方式と量子ゲート方式のデバイスをそれぞれ実行して動作を確認した。
- ・量子アニーリング方式では複数のパターンの中から最も良いものや条件を考えたい場合に活用できる可能性が高い。
- ・量子ゲート方式では古典計算処理の中にある大規模計算部分を処理したい場合に活用できる可能性が高い。

本日はご参加いただきありがとうございました！！

