

Christophe Combelles
Séminaire 10 ans Anybox
5 nov 2020

Quantum Computing

Où en est-on ?



Sommaire

- Pourquoi présenter ça à Anybox ?
- Quels usages ?
- Suprématie quantique ?
- La course mondiale
- Et en France ?
- **Pourquoi** est-ce plus rapide ?
- Simulateurs / bibliothèques Python
- Limites des simulateurs
- Quelques lignes de code
- Et ensuite ?

Pourquoi présenter ça à Anybox ?

On tient compte des appétences et centres d'intérêts

- Les miens
 - Machine Learning
 - Blockchain
 - **Informatique quantique**
 - Langages fonctionnels
 - Modélisation REA
 - Event-Sourcing et CQRS
 - Théorie des catégories
- Présenter est la meilleure façon d'apprendre et retenir !

Pourquoi présenter ça à Anybox ?

On peut l'utiliser aujourd'hui !

- Accès Cloud
- Simulateurs
-  python™

Quels usages ?

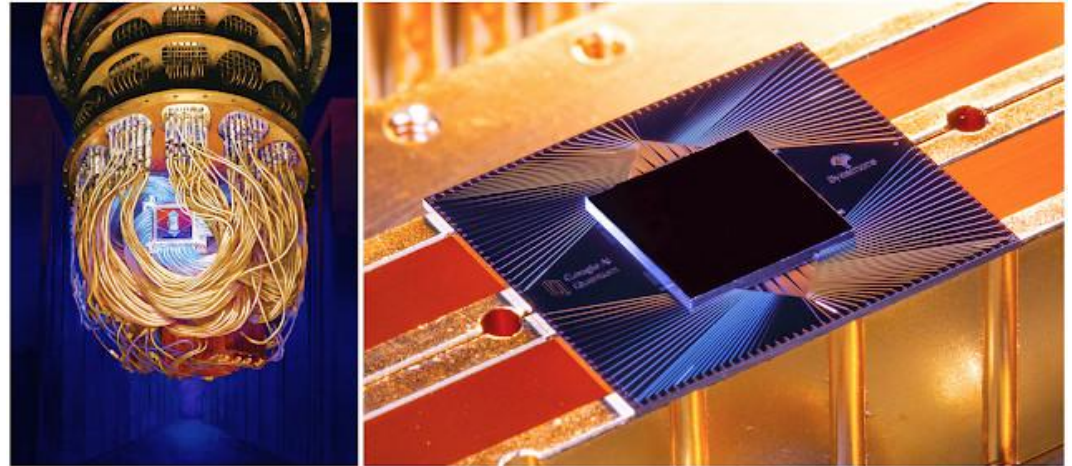
- Problèmes donnant lieu à une explosion combinatoire
- Problèmes d'optimisation
- Exemples :
 - Cyber sécurité
 - Nouveaux matériaux
 - Médicaments
 - Machine Learning / Prévisions (météo, financière, etc.)
 - Énergie / Changement climatique
(batteries, cellules photovoltaïques, supra-conducteurs)
 - Simulation de la mécanique quantique

Supr matie quantique ?

1M CPU x 1Md transistors
10 000 ans



1 QPU x 53 qubits
200 secondes



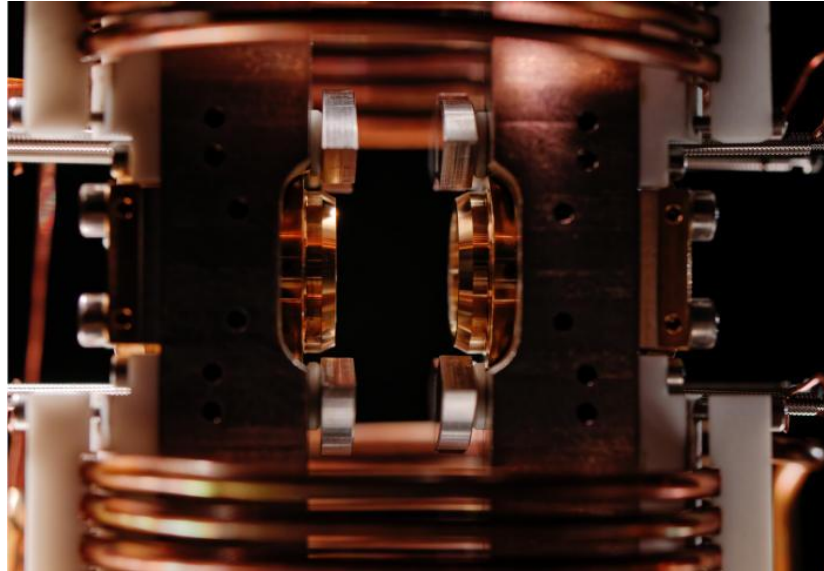
La course mondiale

- La course aux Qubits
 - Honeywell : 10 qubits
 - Rigetti : 19 qubits
 - Xanadu : 24 qubits
 - IonQ : 32 qubits (« parfaits » ?)
 - Intel : 49 qubits
 - IBM : 53 qubits
 - Google : 72 qubits
 - D-Wave : 5000 qubits
- La course au volume quantique
 - IBM : QV 64
 - Honeywell : QV 128
 - IonQ : QV 4000000



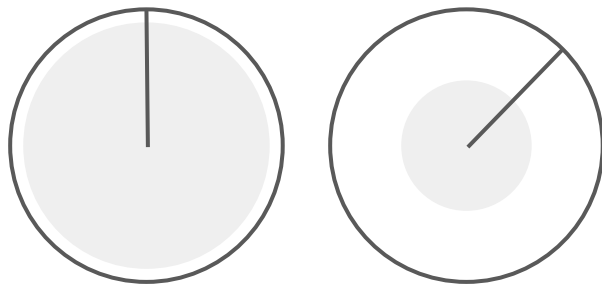
Et en France ?

- Pasqal
- Alice&Bob
- ... ?



Le qubit d'un point de vue logique

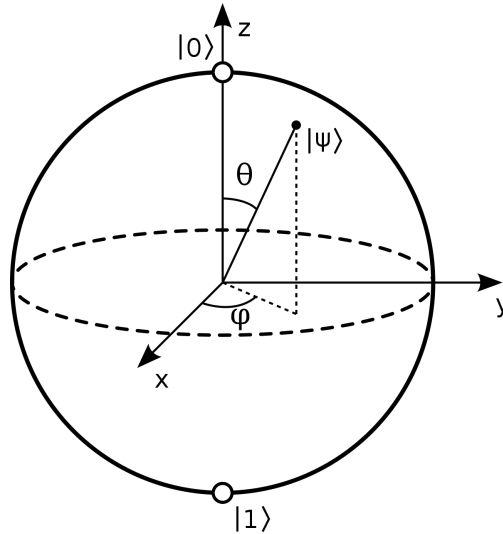
- Représentation d'un qubit sous forme de paire de cercles



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

Le qubit d'un point de vue logique

- Représentation d'un qubit sous forme de sphère de Bloch



Pourquoi est-ce plus rapide ?

- un bit = 0 ou 1 ← 1 information
- un qubit = 0 **et** 1 ← 2 informations
(+ la phase entre 0 et 1)

- Exemple :
 - 90% de chance = 1
 - 10% de chance = 0

Pourquoi est-ce plus rapide ?

- deux bits = 00 ou 01 ou 10 ou 11 ← 2 informations
- deux qubits = 00 **et** 01 **et** 10 **et** 11 ← **4 informations**
(+ la phase entre les 4 états)
-
- Exemple :
 - 50% de chance = 00
 - 0% de chance = 01
 - 0% de chance = 10
 - 50% de chance = 11

Pourquoi est-ce plus rapide ?

- Quantité d'information

1 bit	1 information
2 bits	2 informations
3 bits	3 informations
4 bits	4 informations
...	...
53 bits	53 informations :-)

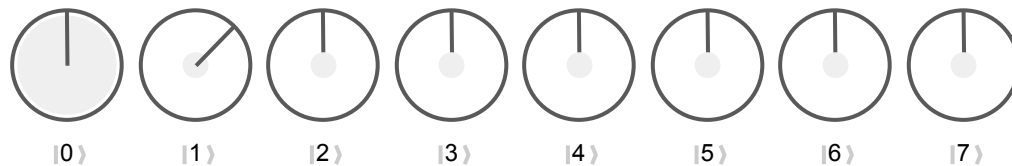
Pourquoi est-ce plus rapide ?

- Quantité d'information

1 qubit	2 informations
2 qubits	4 informations
3 qubits	8 informations
4 qubits	16 informations
...	...
53 qubits	2^{53} informations = 9 millions de milliards = 8192 To

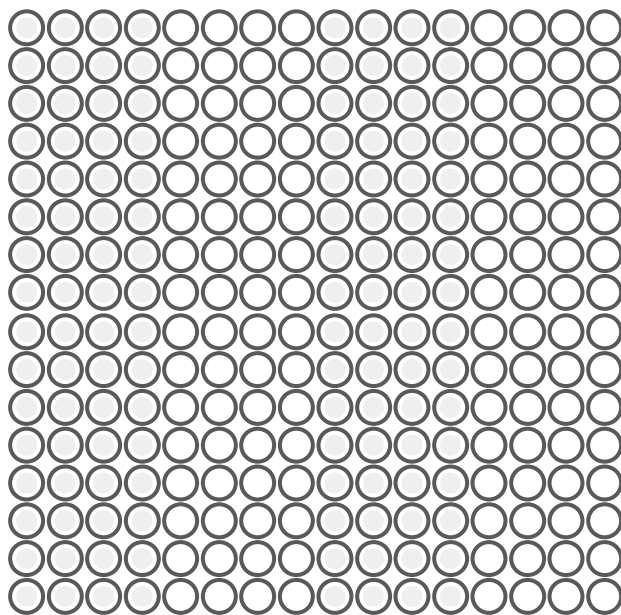
Pourquoi est-ce plus rapide ?

- Représentation logique de 3 qubits



Pourquoi est-ce plus rapide ?

- Représentation logique de 8 qubits



Pourquoi est-ce plus rapide ?

- 3 notions phares

1

La superposition

Les qubits ont plusieurs valeurs en même temps

2

L'intrication

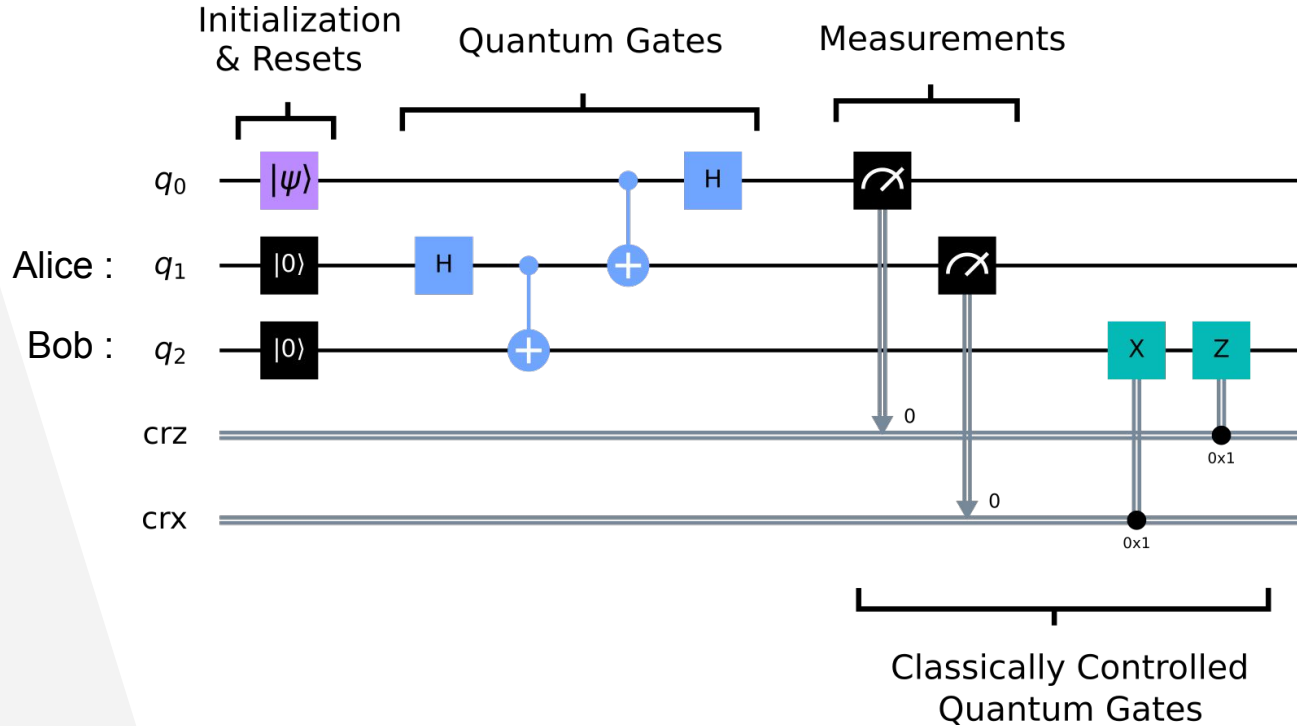
Des qubits peuvent avoir un lien intrinsèque et instantané à distance

3

La téléportation

On peut transférer un état quantique à distance de manière instantanée

Programmation : les circuits quantiques



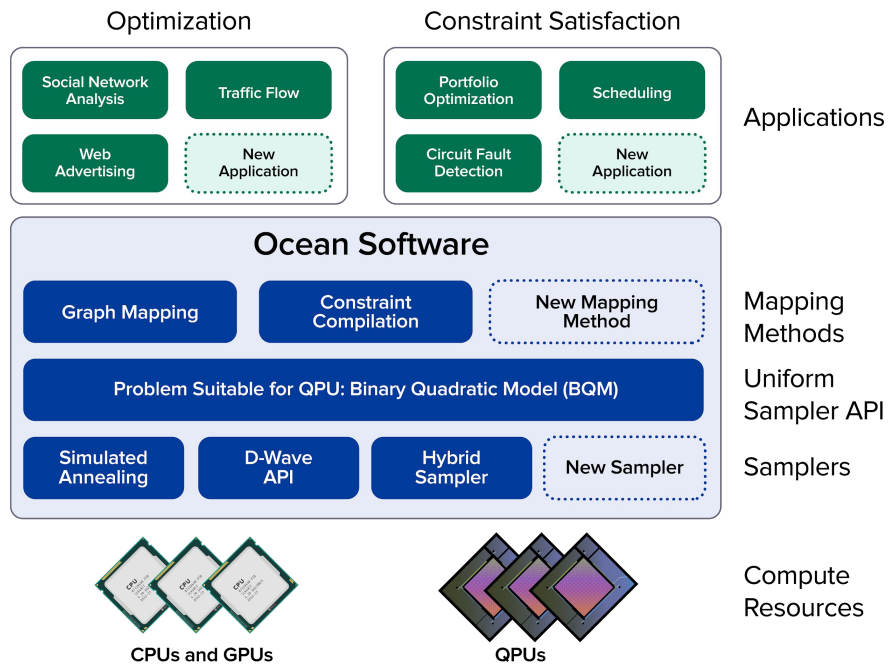
Simulateurs / bibliothèques

QisKit : <https://qiskit.org/>



Simulateurs / bibliothèques

Ocean stack : <https://docs.ocean.dwavesys.com>

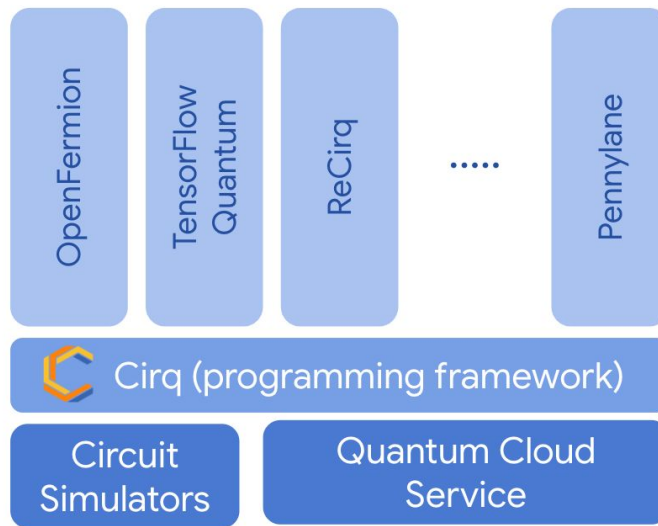


Simulateurs / bibliothèques

Cirq : <https://github.com/quantumlib/Cirq>

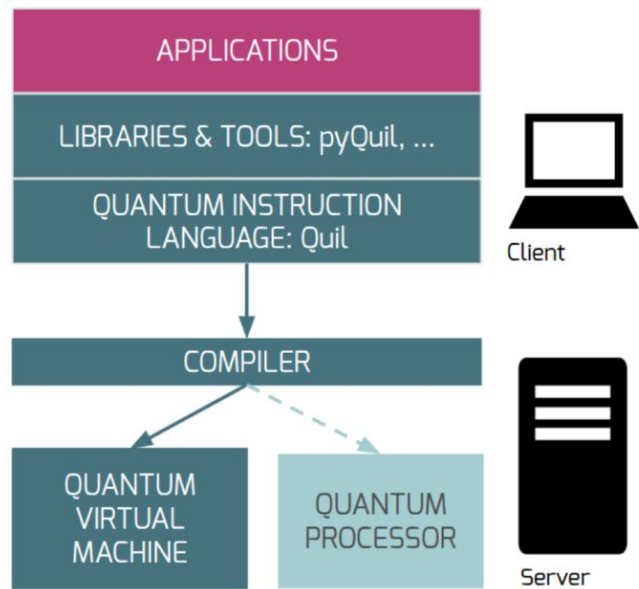


Research
libraries &
Tools



Simulateurs / bibliothèques

Pyquil : <https://pyquil-docs.rigetti.com>

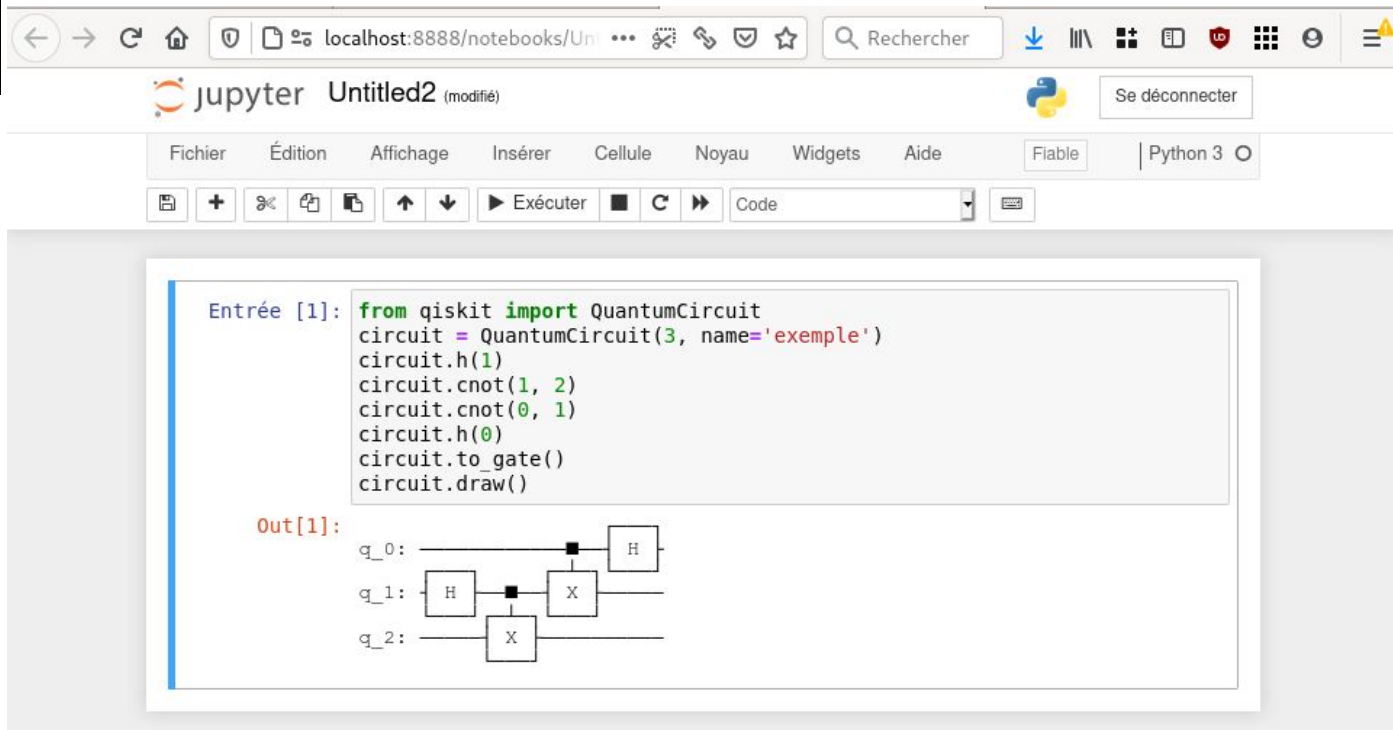


Limites des simulateurs

- Simuler un QPU
 - plus de 48 qubits → prend trop de mémoire
 - moins de 48 qubits → ne sert à rien

Quelques lignes de code

```
$ python3 -m venv qc
$ source qc/bin/activate
(qc) $ pip install jupyter qiskit
(qc) $ jupyter notebook
```



The screenshot shows a Jupyter Notebook interface in a web browser. The browser address bar shows the URL `localhost:8888/notebooks/Untitled2`. The notebook title is "jupyter Untitled2 (modifié)". The interface includes a menu bar with options like "Fichier", "Édition", "Affichage", "Insérer", "Cellule", "Noyau", "Widgets", and "Aide". Below the menu bar is a toolbar with icons for file operations and execution. The main content area displays a code cell labeled "Entrée [1]:" containing the following Python code:

```
from qiskit import QuantumCircuit
circuit = QuantumCircuit(3, name='exemple')
circuit.h(1)
circuit.cnot(1, 2)
circuit.cnot(0, 1)
circuit.h(0)
circuit.to_gate()
circuit.draw()
```

Below the code cell, the output is labeled "Out[1]:" and shows a quantum circuit diagram for three qubits (q_0, q_1, q_2). The circuit consists of the following operations:

- Qubit 1 (q_1) has an H gate.
- Qubit 0 (q_0) has an H gate.
- Qubit 2 (q_2) has an X gate.
- A CNOT gate with control on qubit 1 and target on qubit 2.
- A CNOT gate with control on qubit 0 and target on qubit 1.
- A CNOT gate with control on qubit 2 and target on qubit 0.

Quelques lignes de code

Exemple concret d'accélération d'algorithme :

[4, 5, 6, 2, 7, 8, 9, 0, 3, 1] ← trouver la position de : 3

- **Algorithme classique avec un CPU :**

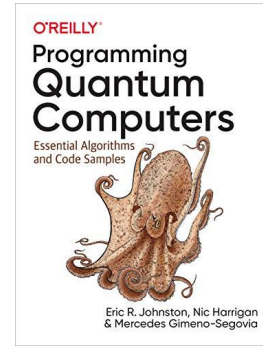
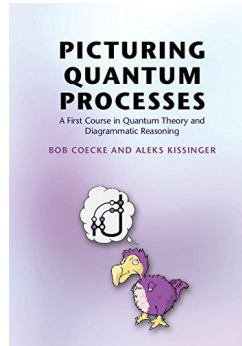
```
def oracle(n):  
    return True if n==3 else False
```

→ Ici besoin de 9 exécutions (complexité en **$O(N)$**)

- **Algorithme de Grover avec un QPU :**

→ Besoin de 2 à 3 exécutions (complexité en **$O(\sqrt{N})$**)

Et ensuite ?



- Livres
- Formations en ligne
 - edX : <https://www.edx.org/learn/quantum-computing>
 - MIT xPRO : <https://learn-xpro.mit.edu/quantum-computing>
 - Coursera : <https://fr.coursera.org/courses?query=quantum%20computing>
 - Future Learn : <https://www.futurelearn.com/courses/intro-to-quantum-computing>
- Événements : FOSDEM, Meetups ?, etc.
- Youtubers
- Jeux et simulateurs (en ligne et sur smartphone)
 - <https://quantumgame.io/>
 - <https://strawberryfields.ai/interactive/>
 - Sur Android : Hello Quantum, Quantum Computing, Quantum, etc....