Avantage quantique

Quand est-ce qu'un ordinateur quantique sera plus performant qu'un ordinateur classique ?

1. L'avantage quantique

1.1. Une définition ?

L'***avantage quantique*** c'est deux choses : un ordinateur quantique et un problème à résoudre. Cet ordinateur quantique sait résoudre ce problème alors qu'aucun ordinateur classique ne peut le faire en temps raisonnable.

- La notion d'avantage quantique est un peu floue : par exemple, est-ce que le problème à résoudre doit être utile ou pas ? Ce que signifie résoudre un problème est clair : l'ordinateur quantique renvoie la bonne réponse en un temps raisonnable (disons en quelques heures ou quelques jours), mais il n'est pas clair de prouver qu'aucun algorithme ne peut résoudre ce même problème sur un ordinateur classique (peut-être qu'un bon algorithme n'a pas encore été trouvé).
- Certaines compagnies affirment avoir déjà dépassé le cap. Le consensus étant que cet avantage sera atteint durant la décennie 2020.
- Le terme « avantage quantique » est maintenant préféré à « suprématie quantique ». Outre l'aspect moins vindicatif du terme « avantage », à moyenne échéance, il est probable que les ordinateurs classiques et les ordinateurs quantiques cohabiteront ; on peut en effet imaginer que les ordinateurs classiques délégueront certaines tâches complexes aux ordinateurs quantiques.

D'autres définitions sont à inventer pour mesurer l'efficacité d'un ordinateur quantique et comparer les technologies mises en œuvre en tenant compte du nombre de qubits, des connexions entre ces qubits, du nombre de portes implémentées, du taux d'erreurs…

1.2. Factorisation

La factorisation des grands entiers est une bonne illustration. Rappelons qu'étant donné un entier il s'agit de lui trouver deux facteurs tels que $n = p \times q$. Tout d'abord c'est un problème utile, car la sécurité de nombreuses communications repose sur ce problème.

Ordinateurs classiques. Les meilleurs algorithmes actuels sur des ordinateurs classiques permettent de factoriser des entiers jusqu'à 250 chiffres (800 bits) (voir le chapitre « Arithmétique »). Les calculs se font sur des centaines d'ordinateurs en parallèle et prennent plusieurs semaines. La complexité de ces algorithmes de factorisation augmente de manière exponentielle avec le nombre de bits. La recommandation minimale pour la longueur d'une clé RSA sûre est actuellement de 2048 bits (600 chiffres). Une telle factorisation est hors de portée de tous les ordinateurs et algorithmes actuels pour encore plusieurs années.

Ordinateurs quantiques. L'algorithme de Shor démontre théoriquement l'avantage des ordinateurs quantiques car il permet de factoriser rapidement des grands entiers.

En 2020 les ordinateurs quantiques possèdent jusqu'à 50 qubits et savent factoriser des entiers à 5 chiffres (14 bits). Pour factoriser un entier de 2048 bits en quelques heures, il faudrait une machine quantique à 20 millions de qubits, ce qui ne sera pas atteint avant une ou deux décennies !

2. Simulation d'un ordinateur quantique

Les ordinateurs quantiques sont fondamentalement différents des ordinateurs classiques, cependant certains circuits quantiques simples peuvent être réalisés de façon efficace sur un ordinateur classique.

Théorème 1 (Gottesman – Knill)**.**
N'importe quel circuit quantique, composé uniquement de portes de Hadamard H, de portes $CNOT$, de portes de Pauli X, Y, Z et de portes de phase S, initialisé avec des états $|0\rangle$ et terminé par des mesures, peut être simulé efficacement par un ordinateur classique.

- « Efficacement » signifie en temps polynomial par rapport à la donnée du circuit.
- La porte S, appelé « porte phase » ou « porte $\frac{\pi}{4}$ », est définie par la matrice :
$$S = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}.$$
- En fait les portes de Pauli X, Y, Z peuvent être générées uniquement avec des portes H et S (c'est un bon exercice).
- Cependant les portes contenues dans l'énoncé ne permettent pas de générer toutes les portes quantiques : cet ensemble de portes n'est donc pas universel. Par exemple la porte de Toffoli, la porte $\sqrt{CNOT}$ ou la porte $\frac{\pi}{8}$ ne peuvent pas être générées à partir des portes du théorème.
- L'ordinateur classique doit être capable de simuler le hasard. Par exemple la mesure du qubit $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ donne 0 ou 1 avec probabilité $\frac{1}{2}$ ce qui revient à jouer à pile ou face.

Voici des exemples de circuits que nous avons rencontrés et qui peuvent être simulés efficacement sur un ordinateur classique :
- la communication par codage super-dense (voir le chapitre « Découverte de l'informatique quantique »),
- la téléportation quantique (voir le chapitre « Téléportation quantique »),
- les codes correcteurs d'erreur (voir la section « Détection d'un flip » du chapitre « Code correcteur »).

3. Arbre de calculs

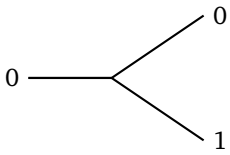
La simulation d'un ordinateur quantique par un ordinateur classique se confronte non seulement à des problèmes de temps de calculs mais aussi à des problèmes de mémoire. En effet, dès que l'on dépasse 50 qubits, il y a 2^{50} états de base, soit plus de 10^{15} états à stocker.
Nous allons voir une modélisation du calcul des états d'un circuit sous la forme d'un arbre. En parcourant l'arbre branche par branche, on teste toutes les possibilités sans utiliser trop de mémoire à chaque fois. La méthode n'apporte pas un gain de temps, qui reste exponentiel en fonction du nombre n de qubits, mais la taille de la mémoire utilisée est linéaire en n.

Nous expliquons cette modélisation par des exemples. Nous partons d'un circuit, avec un état initial. Il s'agit d'obtenir tous les états possibles que l'on pourrait obtenir après mesure.
La brique fondamentale à comprendre est l'arbre pour une porte H de Hadamard.

> **Exemple.**
> Soit le circuit quantique initialisé par $|0\rangle$ suivi d'une simple porte de Hadamard (sans mesure sur la figure

de gauche, avec mesure à droite) :

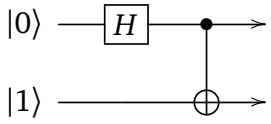
La sortie est $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Si on termine par une mesure alors la sortie est 0 ou 1.
Voici l'arbre de calculs.



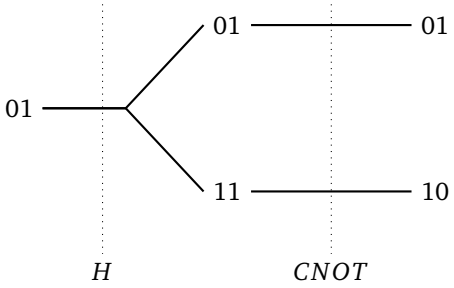
L'arbre de calculs représente tout simplement les deux possibilités. Lorsque l'on passe une porte H les feuilles de l'arbre 0 et 1 représentent la superposition des états $|0\rangle$ et $|1\rangle$. Autrement dit les feuilles représentent toutes les mesures possibles. On a simplifié l'écriture en omettant les coefficients $\frac{1}{\sqrt{2}}$.

Voici un exemple avec deux lignes quantiques.

Exemple.



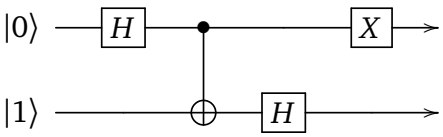
La sortie est le qubit $|\psi\rangle = \frac{1}{\sqrt{2}}(|0.1\rangle + |1.0\rangle)$. Une mesure donnerait donc 0.1 ou bien 1.0, ce que l'on retrouve aux feuilles de notre arbre.

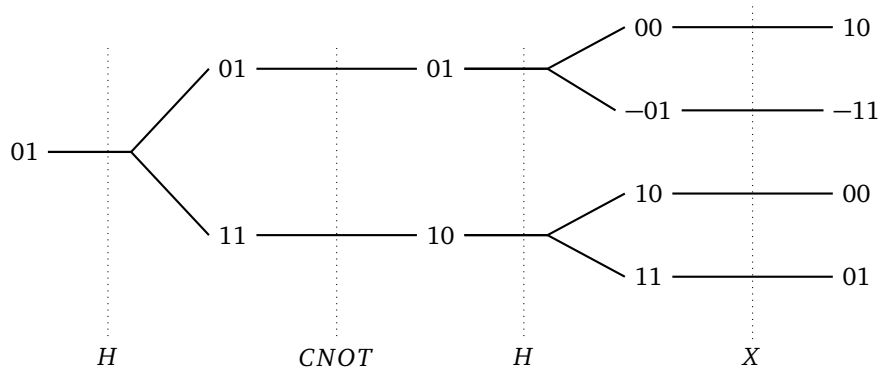


Noter qu'au passage de la porte $CNOT$ l'arbre ne se ramifie pas (chaque branche se poursuit en une seule branche).

Continuons avec un exemple pour comprendre le fonctionnement : à chaque porte H correspond une bifurcation en deux branches. L'intérêt de cet arbre est que les calculs d'une branche sont mis en commun tant qu'il n'y a pas de bifurcation.

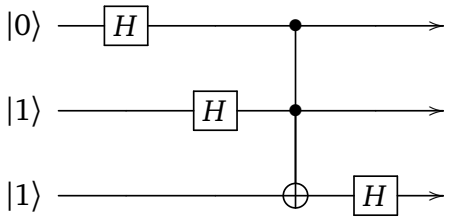
Exemple.



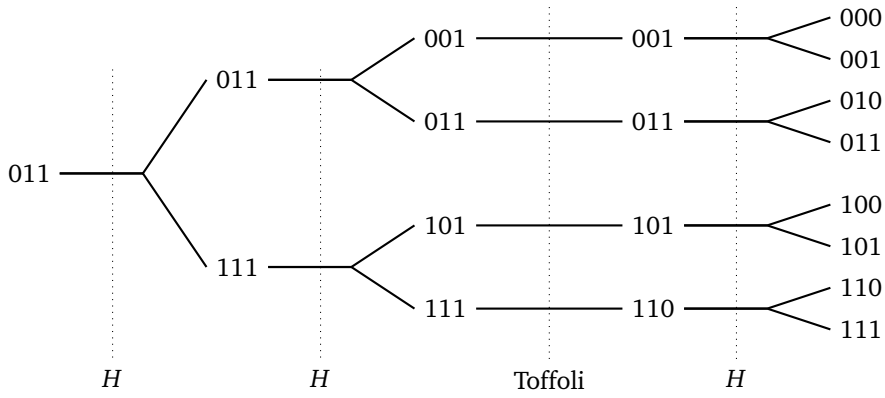


Le qubit de sortie est $|\psi\rangle = \frac{1}{2}(|0.0\rangle + |0.1\rangle + |1.0\rangle - |1.1\rangle)$

Exemple.
Terminons avec un circuit contenant une porte de Toffoli.



Pour simplifier l'arbre ci-dessous, on omet les signes et les coefficients devant les qubits.



Notes. La référence pour les arbres de calculs est Andrew Shi, *Recursive path-summing simulation of quantum computation* (2017).