Compilation pour les ordinateurs quantiques de première génération

Caroline Collange – Inria

15 décembre 2019

Résumé

Après plusieurs décennies de développement, la première génération d'ordinateurs quantiques universels est désormais disponible. Au-delà des enjeux théoriques déjà étudiés depuis les années 1990, la disponibilité concrète d'ordinateurs quantiques ouvre des possibilités et des questions nouvelles pour la communauté langage et compilation. Nous abordons quelques pistes de recherche liées aux enjeux posés par ces technologies.

1 Contexte

Le calcul quantique promet d'exploiter les phénomènes de la physique quantique pour effectuer des tâches de traitement de l'information. Cette direction est particulièrement intéressante d'un point de vue théorique. D'une part, elle constitue une rupture fondamentale dans l'histoire de l'informatique : tous les autres dispositifs de calcul proposés jusqu'ici, qu'ils soient numériques ou analogiques, mécaniques ou électroniques, sont basés sur les règles de la physique classique. D'autre part, le potentiel par rapport au calcul classique est établi : pour certains problèmes, on connaît un algorithme quantique de meilleure complexité asymptotique que les algorithmes classiques connus. En particulier, l'algorithme quantique de Shor remet en cause une grande partie de la cryptographie actuelle en promettant de factoriser des nombres en temps polynomial [16].

Si l'ordinateur quantique relève toujours de la science fiction dans l'inconscient collectif, les développements récents montrent que le concept est viable. Depuis 2016, IBM met un prototype en libre accès à disposition des programmeurs [5]. Google prévoyait en 2017 une rentabilité commerciale dans des domaines de niche dans les 5 ans [10], et annonce la suprémacie quantique par un calcul sur un ordinateur à 54 qubits [1]. En France, Atos constuit des simulateurs d'ordinateurs quantiques [2]. À l'heure actuelle, la technologie la plus populaire est basée sur des dérivés de qubits supraconducteurs [3]. Ces développements concrets promettent de mettre finalement en application les travaux théoriques menés depuis les années 1990 sur le calcul quantique. .

Néanmoins, il existe un gouffre entre les capacités des architectures quantiques envisageables dans un futur proches et les prérequis des algorithmes quantiques. Pour le combler, il faudra non seulement des progrès dans les capacités du matériel, mais aussi réduire nos ambitions sur les applications et les couches logicielles dans un premier temps. En effet, les ordinateurs quantiques de première génération restent très limités. On atteint aujourd'hui une cinquantaine de qubits utilisables, mais ceux-ci sont très sensibles au bruit. Par exemple, les auteurs de l'expérience de suprémacie quantique précitée estiment à 0,2% seulement la chance de succès d'un calcul qui implique 53 qubits et opère 1113 opérations sur un qubit individuel et 430 opérations sur une paire de qubits ainsi qu'une mesure sur chaque qubit [1].

Malgré ces limitations, il est crucial de considérer dès maintenant la programmation de ces ordinateurs quantiques. En effet, l'industrie vise actuellement une rentabilité à court terme des ordinateurs quantiques sur quelques applications ciblées [14]. En cas de succès, cela permettra de justifier des investissements importants qui permettront d'envisager des applications plus ambitieuses. À l'inverse, un échec de la première génération d'ordinateurs quantiques à être utilisables provoquerait certainement un désintérêt pour le domaine et un « hiver quantique » à l'instar de l'hiver de l'IA.

L'enjeu actuel consiste donc non seulement à développer les capacités du matériel quantique, mais aussi à adapter les environnements logiciels pour les capacités modestes de la première génération d'ordinateurs quantiques. Ce domaine de recherche encore peu exploré mais en croissance rapide consiste à revisiter l'architecture des ordinateurs, des systèmes, et la compilation pour l'adapter au nouveau contexte du calcul quantique.

2 Quelques enjeux pour la compilation

Du point de vue de la communauté langages et compilation, le calcul quantique demande de revisiter en profondeur les environnements de compilation. Nous abordons ici quelques exemples d'enjeux de recherche axés sur la compilation bas-niveau, sans prétendre à l'exhaustivité.

Gestion du bruit En l'absence de mécanisme de correction d'erreur, le temps d'exécution d'un programme quantique est limité par le bruit et le phénomène de décohérence. Plus un programme quantique met de temps à s'exécuter et plus il opère d'opération, moins son résultat est précis. Les optimisations de compilation minimisant le temps d'exécution et la complexité du calcul sont donc cruciales, non seulement pour la vitesse mais surtout pour la précision. Notons que si le temps de décohérence est de l'ordre de 1000 fois le délai d'une porte quantique dans un ordinateur à 50 qubits, il n'est possible d'opérer que 20 portes par qubit en moyenne si celles-ci sont effectuées de manière séquentielle. Il est donc important de permettre l'exécution en parallèle de portes quantiques pour permettre le passage à l'échelle.

Les codes correcteurs d'erreurs quantiques représentent un champ de recherche actif qui promet à long terme de pallier à ces limitations [8]. Cependant, le surcoût imposé par la plupart des techniques de correction d'erreur rend leur emploi rédhibitoire sur les premières générations d'ordinateurs quantiques. Il est probable qu'il faille se contenter de techniques de mitigation d'erreur ou de correction partielle d'erreur à court et moyen terme. Au niveau langage, il faudra permettre de spécifier des algorithmes partiellement tolérants au bruit, afin de profiter de codes correcteurs d'erreurs à couverture partielle.

Placement et allocation de qubits Comme son équivalent classique l'allocation de registres, l'allocation de qubits consiste à placer les variables du programme sur les ressources matérielles. Cette allocation doit respecter les contraintes physiques et pratiques imposées par la machine. Dans le cas des qubits supraconducteurs, la principale contrainte est la connectivité entre qubits. En effet, le couplage entre deux qubits qui permet de les intriquer est possible seulement entre certaines paires de qubits, suivant un graphe de couplage spécifique à la machine. Le compilateur doit donc opérer des transformations sur le circuit pour obéir à ces contraintes de connectivité [17].

En l'absence d'un équivalent quantique de la mémoire, l'allocation de qubits s'effectue sur la globalité du programme. Si elle est actuellement praticable sur les premiers programmes quantiques de taille réduite, cette approche risque de rencontrer des limites en termes de scalabilité. Il sera nécessaire d'identifier des manières de partitionner les programmes pour limiter la complexité. On observe par ailleurs que le taux d'erreur peut différer d'un ordre de grandeur entre différents qubits du même ordinateur quantique [15]. Une piste pour améliorer l'allocation de qubits consiste à prendre en compte l'hétérogénéité des caractéristiques des qubits.

Sélection et ordonnancement de portes quantiques La sémantique des opérations quantiques est traditionnellement exprimée par des calculs d'algèbre linéaire. De fait, la synthèse de transformations arbitraires peut se concevoir comme la décomposition d'une matrice en une combinaison de produits de matrices et produits tensoriels de matrices de rotations simples [4]. Cette approche bas-niveau peut être vue comme l'analogue quantique de l'optimisation de formules booléennes. Cependant, son passage à l'échelle reste incertain.

Une approche plus proche de la compilation classique consiste à exposer des ensembles d'opérations prédéfinies en plusieurs couches à différents niveaux d'abstraction, sous la forme de bibliothèques. Cela ouvre des possibilités de simplifications sous réserve de vérifier des propriétés globales sur le programme. Par exemple une porte de Toffoli peut être réalisée plus simplement au prix de l'introduction d'un déphasage, qui doit alors être compensé dans le reste du circuit [9].

Analyses de compilation Comme dans le cas classique, les optimisations de compilation profitent voire nécessitent des informations obtenues par des analyses statiques. Certaines analyses sont analogues au cas classique : par exemple identifier les dépendances entre qubits pour leur allocation [17]. D'autres analyses sont spécifiques aux programmes quantiques : par exemple, déterminer quels qubits sont intriqués et quels qubits sont séparables [13, 6]. Cela ouvre un champ de recherche consistant à proposer des analyses pertinentes et les réaliser de manière précise et efficace.

Quelles représentations internes ? Enfin, une question clé dans la conception de compilateurs est celle du choix des abstractions et représentations internes employées pour représenter les programmes. Quelle serait « la forme SSA du calcul quantique » ? Il s'agit d'identifier une ou plusieurs représentations internes ou structures de données permettant de faciliter les analyses et optimisations de compilation. Ces représentations doivent à la fois pouvoir être mises en œuvre de façon efficace et pouvoir être formalisées.

Les représentations traditionnelles se basent sur l'abstraction des circuits quantiques, dont la sémantique s'exprime en termes d'opérations d'algèbre linéaire. En revanche, la simulation de circuit quantiques sur des machines classiques s'appuie typiquement sur des réseaux de tenseurs [12]. Le ZX-Calculus est une construction axiomatique qui a été récemment prouvée complète [11, 7], qui pourrait constituer une représentation intermédiaire plus souple que les circuits quantiques.

Références

[1] Frank Arute, Kunal Arya, and Ryan Babbush et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574 :505–510, 2019.

[2] Atos. *Quantum Learning Machine*, 2018.

[3] Caroline Collange. Ordinateurs quantiques : ouvrons la boîte. In *COMPAS 2019 - Conférence d'informatique en Parallélisme, Architecture et Système*, Anglet, France, June 2019.

[4] Christopher M. Dawson and Michael A. Nielsen. The Solovay-Kitaev algorithm, 2005.

[5] Simon J. Devitt. Performing quantum computing experiments in the cloud. *Phys. Rev. A*, 94(3) :032329, 2016.

[6] Ali JavadiAbhari, Shruti Patil, Daniel Kudrow, Jeff Heckey, Alexey Lvov, Frederic T Chong, and Margaret Martonosi. ScaffCC : a framework for compilation and analysis of quantum computing programs. In *Computing Frontiers*, page 1, New York, NY, USA, 2014. ACM.

[7] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Diagrammatic reasoning beyond Clifford+T quantum mechanics, 2018.

[8] Daniel A Lidar and Todd A Brun. *Quantum error correction*. Cambridge University Press, Cambridge, UK, 2013.

[9] Dmitri Maslov. Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization. *Physical Review A*, 93(2), Feb 2016.

[10] M Mohseni, P Read, H Neven, S Boixo, V Denchev, R Babbush, A Fowler, V Smelyanskiy, and J Martinis. Commercialize early quantum technologies. *Nature*, 543(7644) :171, 2017.

[11] Kang Feng Ng and Quanlong Wang. A universal completion of the ZX-calculus, 2017.

[12] Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, Erik W. Draeger, Eric T. Holland, and Robert Wisnieff. Breaking the 49-qubit barrier in the simulation of quantum circuits, 2017.

[13] Simon Perdrix. Quantum entanglement analysis based on abstract interpretation. In *SAS*, pages 270–282, Heidelberg, Germany, 2008. Springer.

[14] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2 :79, Aug 2018.

[15] Swamit S. Tannu and Moinuddin Qureshi. A case for variability-aware policies for nisq-era quantum computers. In *ASPLOS*, 2019.

[16] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *Journal on Computing*, 26(5) :1484–1509, 1997.

[17] Marcos Yukio Siraichi, Vinicius Fernandes dos Santos, Caroline Collange, and Fernando Magno Quintão Pereira. Qubit allocation as a combination of subgraph isomorphism and token swapping. In *OOPSLA*, Athens, Greece, October 2019.