Modèles de complexité calculatoire dans le nuage

Charles Bouillaguet, Sorbonne université / LIP6 (charles.bouillaguet@lip6.fr)

Context

La complexité des algorithmes est traditionnellement mesurée en comptant le nombre d'opérations « élémentaires » effectuées dans un modèle de calcul abstrait, qui est une simplification très grossière d'un véritable ordinateur. Par exemple, il est bien connu que le *tri fusion* nécessite $\mathcal{O}(n \log n)$ opérations.

Le modèle calculatoire habituellement enseigné et utilisé est généralement décrit de manière très floue. C'est (une variante de) celui de la *Random Access Machine*. Dans ce modèle, effectuer des opérations arithmétiques sur des opérandes de petites tailles est une « opération élémentaire » qui prend un temps constant, tout comme lire ou écrire la mémoire à une adresse donnée.

Ce modèle est bien adapté à l'étude de petits calculs qui se déroulent à l'intérieur d'un ordinateur de petite taille. Il prédit de manière nettement moins fiable le comportement de calculs qui se déroulent sur de grandes machines parallèles. En effet, il ignore complètement toute la problématique de la complexité de communication. Par exemple, il est empiriquement faux qu'accéder à une mémoire arbitrairement grande prend un temps constant, indépendamment de l'adresse et de la distance du composant mémoire avec le processeur qui effectue la requête. De plus, ceci viole les lois connues de la physique.

Dans le calcul scientifique, et de manière plus générale sur des machines de calcul parallèle, il est au contraire habituel de réfléchir au coût des communications, et des modèles spécifiques (e.g. LogP [4]) ont été développés pour réfléchir « sur le papier » au coût des algorithmes.

Exécuter des calculs dans le nuage, c'est-à-dire en louant des machines virtuelles à l'opérateur d'un nuage public, pose le problème du coût des calculs d'une manière radicalement différente. En effet, la notion de « coût « des calcul a alors une acception directement fiduciaire, car il faut payer la location des ressources utilisées — le coût est alors proportionnel à la somme des durées des locations des machines virtuelles utilisées. Ceci est très comparable à la métrique des « heures-CPU » utilisées dans les centres de calcul scientifique.

Le point de départ de ce sujet de thèse est l'observation que les coûts asymptotiques dans le nuage ne sont prouvablement pas les mêmes que ceux qu'on enseigne habituellement dans tous les cours d'algorithmique. Si un algorithme nécessite une (grosse) mémoire M pour fonctionner, supérieure à la capacité d'un seul serveur de calcul, alors pour en exécuter une implantation distribuée, il faut louer $\Omega(M)$ machines virtuelles pour l'exécuter. Si celle-ci était une adaptation naïve d'un algorithme séquentiel, alors son temps d'exécution serait proportionnel au nombre d'opérations, et le « coût-nuage » serait de l'ordre de TM — soit le *produit* de la complexité temporelle et spatiale.

Personne ne s'amuserait cependant à louer un *cluster* de calcul pour exécuter des algorithmes séquentiels, car ce serait très inefficace. De manière générale, faire de gros calcul nécessite des algorithmes parallèles. Le coût-nuage dépend alors du temps d'exécution de la version parallèle, et celui-ci est directement lié à l'efficacité de la parallélisation mise en oeuvre et à l'efficacité des communications. Autant dire que le coût-nuage obtenu n'a pas forcément grand-chose à voir avec le nombre d' »opération élémentaires » nécessaires.

Pour revenir sur le cas du tri fusion, Thompson a démontré en 1978 [13] que la complexité de communication du tri entraine une borne inférieure sur son coût dans le modèle VLSI, de la forme $AT \geq n^{1.5}$ (où A désigne la surface d'un circuit plat qui effectue le tri et T désigne le temps nécessaire). Ceci implique vraissemblablement que trier dans le nuage coûte asymptotiquement plus cher que dans le modèle de la *Random access Machine*.

Beaucoup d'opérations usuelles (multiplication, FFT, convolution, etc.) sont également concernées par de telles bornes inférieures prouvées dans les années 1970–1980 [14, 2, 3, 13, 11].

La complexité de nombreux algorithmes est donc plus élevée dans le nuage, et le modèle de calcul classique ne le reflète pas.

Objectifs de la thèse

Dans le modèle de calcul usuel, de nombreux « progrès » algorithmiques consistent à utiliser plus de mémoire pour faire baisser le nombre d'opérations. Comme la complexité est habituellement mesurée en nombre d'opérations, il y a manifestement un gain. Mais la plupart du temps, ces algorithmes sont décrits sur le papier et jamais implantés sur de vraies machines, en tout cas pas à grande échelle.

La question de fond que pose ce sujet de thèse est : dans le nuage ces « progrès » ne sont-ils pas des regressions ? Ne font-ils pas *augmenter* le coût (en augmentant la quantité de ressources matérielles nécessaires) ?

Ce n'est pas facile de répondre à cette question, car les algorithmes sont généralement décrits dans le modèle classique qui est purement séquentiel, alors que leur exécution à grande échelle dans le nuage nécessite des implantations parallèles. Trancher ce genre de question expérimentalement est difficile, car cela demande un non seulement un travail (potentiellement non-trivial) d'adaptation algorithmique mais aussi une maître sérieuse de l'art de la programmation des ordinateurs. De plus, un certain nombre de facteurs « cachés » interviennent, comme la topologie et la qualité du réseau déployé dans le nuage, qui ne sont généralement pas connus.

Par exemple, on peut réfléchir à la questio, en s'attaquant à un des problèmes calculatoires les plus classiques et les plus anciens : le problème SUBSET-SUM est l'un des 21 problèmes NP-complets que Karp a exhibés en 1972 (sous le nom de « sac-à-dos ») [10]. Étant donné n grands entiers, y en a-t-il un sous-ensemble dont la somme vaut une valeur « cible » donnée ? Le cas le plus dur est celui où on a affaire à n entiers de n bits. De nombreux algorithmes ont été décrits pour tenter de résoudre le problème le plus vite possible, ou bien en faisant des compromis temps-mémoire [7, 5, 6, 1, 9, 12, 8]. Mais quel algorithme coûterait le moins cher dans le nuage ?

On peut essayer de répondre à la question d'un point de vue théorique (en faisant des raisonnements asymptotiques lorsque n tend vers l'infini dans un modèle de calcul abstrait de nuage qui reste à définir) et en pratique (avec des expériences sur de vrais *clusters* de calcul).

Références

[1] Anja Becker, Jean-Sébastien Coron, and Antoine Joux. Improved generic algorithms for hard knapsacks. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 364–385. Springer, 2011.

[2] Gianfranco Bilardi and Franco P. Preparata. Area-time lower-bound techniques with applications to sorting. *Algorithmica*, 1(1) :65–91, 1986.

[3] R. P. Brent and H. T. Kung. The chip complexity of binary arithmetic. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC '80, pages 190–200, New York, NY, USA, 1980. ACM.

[4] David E. Culler, Richard M. Karp, David A. Patterson, Abhijit Sahay, Klaus E. Schauser, Eunice E. Santos, Ramesh Subramonian, and Thorsten von Eicken. Logp : Towards a realistic model of parallel computation. In Marina C. Chen and Robert Halstead, editors,

Proceedings of the Fourth ACM SIGPLAN Symposium on Principles & Practice of Parallel Programming (PPOPP), San Diego, California, USA, May 19-22, 1993, pages 1–12. ACM, 1993.

[5] Claire Delaplace, Andre Esser, and Alexander May. Improved low-memory subset sum and LPN algorithms via multiple collisions. In Martin Albrecht, editor, *Cryptography and Coding - 17th IMA International Conference, IMACC 2019, Oxford, UK, December 16-18, 2019, Proceedings*, volume 11929 of *Lecture Notes in Computer Science*, pages 178–199. Springer, 2019.

[6] Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems. *IACR Cryptol. ePrint Arch.*, page 217, 2012.

[7] Andre Esser and Alexander May. Low weight discrete logarithm and subset sum in $2^{0.65n}$ with polynomial memory. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 94–122. Springer, 2020.

[8] Ellis Horowitz and Sartaj Sahni. Computing partitions with applications to the knapsack problem. *J. ACM*, 21(2) :277–292, 1974.

[9] Nick Howgrave-Graham and Antoine Joux. New generic algorithms for hard knapsacks. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 235–256. Springer, 2010.

[10] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[11] John E. Savage. Area—time tradeoffs for matrix multiplication and related problems in vlsi models. *Journal of Computer and System Sciences*, 22(2) :230 – 242, 1981.

[12] Richard Schroeppel and Adi Shamir. A $T = \mathcal{O}\left(2^{n/2}\right)$, $S = \mathcal{O}\left(2^{n/4}\right)$ Algorithm for Certain NP-Complete Problems. *SIAM J. Comput.*, 10(3) :456–464, 1981.

[13] C. D. Thompson. Area-time complexity for vlsi. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 81–88, New York, NY, USA, 1979. ACM.

[14] J. Vuillemin. A combinatorial limit to the computing power of vlsi circuits. *IEEE Trans. Comput.*, 32(3) :294–300, March 1983.