Projet de Recherche

Abstractions Qualitatives pour l'Ordonnancement *Exascale*

Raphaël Bleuse (raphael.bleuse@imag.fr)

8 décembre 2017

Le prochain jalon que la communauté du calcul haute performance s'est posé est l'*exascale* (10^{18} Flop/s [1]). Atteindre une telle puissance de calcul tout en maintenant les coûts de construction et d'exploitation raisonnables nécessite d'une part de changer la conception des machines et d'arriver à utiliser au mieux les ressources disponibles d'autre part.

Les machines hybrides – constituées d'unité de calcul plus ou moins spécialisées pour certaines tâches (calculs, entrées/sorties, ...) – sont une approche envisagée par les constructeurs pour augmenter les performances et l'efficacité énergétique des nouvelles architectures.

Diminuer les coûts peut aussi se faire en réorganisant les réseaux de communication. Cette réorganisation peut se faire via deux axes orthogonaux : via la conception et l'étude de nouvelles topologies ou via la fusion des réseaux de communication inter processus et d'entrées/sorties. Néanmoins, cette multiplication des types de ressources de calcul, ainsi que la fusion des différents flux réseau constitue un **défi pour les couches logicielles chargées de gérer les ressources**.

Dans le cadre de mon doctorat, nous nous sommes intéressé au problème de l'ordonnancement d'applications sur les machines parallèles. Une application est représentée comme un ensemble de tâches élémentaires à effectuer : trouver un ordonnancement revient à déterminer pour chaque tâche où et quand s'exécuter. Les impacts des évolutions susmentionnées sur l'ordonnancement se font à plusieurs échelles : les évolutions architecturales des ressources de calcul ont un impact à grain fin (c.-à-d. au sein d'un unique nœud de calcul) alors que les évolutions des réseaux se font ressentir au niveau de la machine complète. Au vue de la taille des machines actuelles, utiliser une description détaillée et quantitative des machines ne semble pas raisonnable et ne passera vraisemblablement pas à l'échelle.

1 Principaux résultats obtenus

Le début de mon doctorat a été consacré à l'étude d'ordonnancements à grain fin. Nous nous sommes intéressé à l'ordonnancement de tâches séquentielles indépendantes au sein d'un nœud composé de deux types de ressources de calcul. Nous avons proposé une notion d'*affinité* comme indicateur qualitatif pour rendre compte du degré de liaison entre les différentes tâches d'une application parallèle. L'idée sous-jacente était que des tâches en fortes interactions puissent être allouées sur des ressources voisines. Ces travaux ont été présentés pendant la conférence Euro-Par 2014 [2]. Dans la même veine, nous avons implémenté des algorithmes d'ordonnancement proposant de meilleures garanties au prix d'une complexité plus grande. Confirmant nos intuitions, le surcout de calcul induit est trop important et dégrade les performances globales. Ces travaux ont conduit à une publication dans le journal CCPE [4].

La suite naturelle de ces travaux a été l'extension du modèle d'exécution des tâches : les tâches sont considérées *moldables* sur un type de ressource de calcul. Le problème a été abordé

1. Flop/s : opération flottante par seconde

à l'aide d'un algorithme reposant sur un programme linéaire rapide, mais dont la complexité en pire cas est inconnue. Nous avons aussi proposé un algorithme relaxé de faible complexité. Suite à une première évaluation encourageante, un article en révision favorable et décrivant ces travaux vient d'être soumis au journal TPDS [3].

Dans le cadre d'une collaboration au sein du partenariat inter laboratoire JLESC [2], j'ai eu accès à Blue Waters – la machine opérée par le NCSA[3] La topologie réseau particulière de Blue Waters présente des défis pour l'ordonnancement à l'échelle de la machine [6]. À partir des traces d'exécution de la machine, nous avons proposé une modélisation des machines prenant en compte les contraintes imposées par les topologies hétérarchiques [5]. Ces topologies ont la particularité de proposer une vision de la machine beaucoup moins hiérarchique que les topologies habituelles (*fat tree*, par exemple). Par exemple, les nouvelles topologies réseaux proposées proposent des hétérarchies locales connectées dans une hiérarchie de faible profondeur [1,7]. Les contraintes dérivent alors principalement de l'agencement des nœuds hétérogènes au sein de la topologie. La prise en compte des contraintes n'est pas faite de manière quantitative, mais au travers de propriétés géométriques : les allocations sont contraintes à être convexes.

2 Axes de recherche prévus

Le modèle théorique évoqué ci dessus a été bien reçu par la communauté. Deux pistes à explorer dans le cadre de ce modèle ont été identifiées.

L'entrelacement des flux d'entrée/sortie des différentes applications au sein d'un réseau perturbe les performances. La convexité des allocations permet de limiter ces interférences. Obliger les applications à se situer près des nœuds d'entrée/sortie peut permettre d'éliminer complètement ces interactions néfastes. Quelles stratégies peuvent être mises en place pour ajouter ces contraintes à un cout raisonnable ? Quel est l'impact de ces contraintes supplémentaires sur le taux d'utilisation des machines ?

D'autre part, pour un certain nombre de ressources demandées par une application, il existe plusieurs formes convexes qui répondent aux besoins. Une piste que nous souhaitons explorer est d'étudier quel choix de forme permet de mieux exploiter les machines. Dans la même optique, nous souhaiterions étudier si allouer plus de ressources que demandé permettrait de simplifier l'étape d'allocation sans sans perdre trop de puissance de calcul.

Références

[1] Besta, M., and Hoefler, T. Slim Fly : A Cost Effective Low-diameter Network Topology. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Piscataway, NJ, USA, 2014), SC '14, IEEE Press, pp. 348–359.

[2] Bleuse, R., Gautier, T., Lima, J. V. F., Mounié, G., and Trystram, D. Scheduling Data Flow Program in XKaapi : A New Affinity Based Algorithm for Heterogeneous Architectures. In *Euro-Par 2014 Parallel Processing* (Aug. 2014), F. Silva, I. Dutra, and V. Santos Costa, Eds., vol. 8632, Springer International Publishing, pp. 560–571.

[3] Bleuse, R., Hunold, S., Kedad-Sidhoum, S., Monna, F., Mounié, G., and Trystram, D. Scheduling Independent Moldable Tasks on Multi-Cores with GPUs. *Manuscript submitted for publication* (2016).

2. *Joint Lab. for Extreme Scale Computing* – cf. `http://publish.illinois.edu/jointlab-esc/`
3. *National Center for Supercomputing Applications* – cf. `http://www.ncsa.illinois.edu/`

[4] Bleuse, R., Kedad-Sidhoum, S., Monna, F., Mounié, G., and Trystram, D. Scheduling Independent Tasks on Multi-Cores with GPU Accelerators. *Concurrency and Computation : Practice and Experience 27*, 6 (2015), 1625–1638.

[5] Bleuse, R., Lucarelli, G., and Trystram, D. Convex Allocations under IO Constraints, Mar. 2016. Presented at *New Challenges in Scheduling Theory*, Aussois.

[6] Enos, J., Bauer, G. H., Brunner, R., Islam, S., Steed, M., Jackson, D., and Fiedler, R. Topology-Aware Job Scheduling Strategies for Torus Networks. *Cray User Group* (2014).

[7] Tuncer, O., Leung, V. J., and Coskun, A. K. PaCMap : Topology Mapping of Unstructured Communication Patterns Onto Non-contiguous Allocations. In *Proceedings of the 29th ACM on International Conference on Supercomputing* (New York, NY, USA, 2015), ICS '15, ACM, pp. 37–46.