Proposition de stage de recherche (Master 2)

Abstraction des entrées/sorties pour la parallélisation automatique

Encadrant: Christophe Alias, chargé de recherche à l'INRIA.
mail: `Christophe.Alias@ ens-lyon.fr`
web: `http://perso.ens-lyon.fr/christophe.alias`

Durée: de 4 à 6 mois (gratification $\approx$ 500 euros/mois)

Lieu: Laboratoire de l'Informatique du Parallélisme (LIP)
École Normale Supérieure de Lyon

Les technologies utilisées dans les super-calculateurs pour le stockage de données (DDR, SSD, etc) et les communications (QPI, HT, etc) ont une bande passante très en dessous des besoins des unités de calcul. La solution classique est de cacher les données réutilisées dans une mémoire rapide pour éviter d'avoir à les recharger. Plus la distance entre le premier chargement d'une donnée et sa dernière utilisation est grande, plus la mémoire rapide doit être grande; ce qui impose de transformer le programmer pour que les données logent dans la mémoire rapide [2]. Un compromis doit également être trouvé avec le degrée de parallélisation qui impose d'autant plus de mémoire rapide.

Depuis les travaux récents sur le modèle DPN [1], on sait régler *optimalement* la bande passante et la taille de la mémoire rapide pour n'importe quel programme régulier *indépendamment* du degré de parallélisation, qui peut être aussi grand que le calcul le permet. Malheureusement l'optimalité à un coût: le domaine des données à charger sur chaque tuile – une union de polyèdres convexes – devient rapidement très complexe.

Le but de ce stage est de trouver une abstraction réglagle des domaines de données, entre le domaine original (taille mémoire rapide minimale, mais contrôle complexe) et une sur-approximation grossière (mémoire rapide plus grande, contrôle simple). Il s'agit de:

- Définir ce qu'est une approximation correcte des chargements.
- Quantifier l'impact d'une approximation (taille mémoire, complexité du contrôle).
- Trouver un algorithme qui réalise une approximation en fonction de l'impact toléré.

Compétences souhaitées. Notions en compilation et en parallélisme.

References

[1] Christophe Alias and Alexandru Plesco. Data-aware Process Networks. Research Report RR-8735, Inria - Research Centre Grenoble – Rhône-Alpes, June 2015.

[2] François Irigoin and Remi Triolet. Supernode partitioning. In *Proceedings of the 15th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 319–329. ACM, 1988.