

# Sujet de stage (Master 2)

## Parallélisation hiérarchique

**Encadrant:** Christophe Alias, [christophe.alias@ens-lyon.fr](mailto:christophe.alias@ens-lyon.fr)

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

### Contexte

Les limites technologiques liées à la miniaturisation des circuits imposent de dupliquer les unités de calcul pour obtenir davantage de performances. Or, découper un programme en unités parallèles et régler les communications est une activité difficile et coûteuse à la fois en temps et en moyens humains. La tendance actuelle est donc à l'automatisation de cette tâche de parallélisation au moyen d'un *compilateur paralléliseur*.

De nombreux problèmes doivent être résolus par un tel compilateur. Où se trouve le parallélisme dans le programme? Quelles parties paralléliser et comment? Comment allouer les ressources matérielles? Le *modèle polyédrique* permet de traiter toutes ces questions dans le même formalisme. Sous certaines contraintes, les iterations des boucles `for` peuvent être représentées par des polyèdres, puis un raisonnement géométrique permet de construire les analyses (dépendances, ordonnancement) avec de la programmation linéaire et des opérations géométriques (union, intersection, projection). Mais la précision à un coût: la complexité des calculs limite la taille des programmes compilables à quelques dizaines de lignes.

### Objectif

Dans ce stage, on se propose d'étudier comment faire passer à l'échelle le modèle polyédrique. On s'appuiera sur une approche *divide-and-conquer*: un programme est divisé en sous-programmes, chaque sous-programme est analysé séparément, puis on "recolle les morceaux". On étudiera le cas particulier du *tuilage de boucles*, une transformation polyédrique qui divise le domaine d'itération d'une imbrication de boucles en blocs atomiques.

- A la main: étudier les tuilages possibles sur plusieurs compositions de programmes polyédriques (e.g. plusieurs produits de matrices). Formuler les propriétés d'une bonne composition.
- Proposer un algorithme de tuilage hiérarchique qui reprend les propriétés trouvées.
- Implémenter et tester l'algorithme L'approche sera validée expérimentalement sur les benchmarks de la communauté [1].

**Compétences souhaitées.** Notions en compilation et en parallélisme.

## References

- [1] Louis-Noël Pouchet. Polybench: The polyhedral benchmark suite. *URL: <http://www.cs.ucla.edu/~pouchet/software/polybench/>*[cited July,], 2012.