

## Proposition de stage de recherche (Master 2)

### Analyse de WCET et compilation des réseaux de processus

**Encadrant:** Christophe Alias, chargé de recherche à l'INRIA

mail: [Christophe.Alias@ens-lyon.fr](mailto: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 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? On répond à la première question en traduisant le programme en un réseau de processus communicants [1, 2, 3]. En particulier, les Data-aware Process Network (DPN) [1] offrent la possibilité de "régler" le parallélisme et la granularité de l'application, ce qui en fait une représentation intermédiaire adaptée.

L'objet de ce stage est d'apporter des éléments de réponse à la deuxième question. Etant donné un réseau de processus DPN obtenu à partir d'un programme, il s'agit de concevoir des algorithmes pour:

- calculer la latence (worst-case execution time, WCET) et la bande passante optimale du réseau
- trouver un chemin critique du réseau
- trouver les bons réglages de parallélisation du réseau pour atteindre la latence optimale tout en minimisant la taille du réseau.

L'approche sera validée expérimentalement sur les benchmarks de la communauté [4].

**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] Gilles Kahn. The semantics of simple language for parallel programming. In *IFIP Congress 74*, pages 471–475, 1974.
- [3] Edward A Lee and David G Messerschmitt. Synchronous data flow. *Proceedings of the IEEE*, 75(9):1235–1245, 1987.
- [4] Louis-Noël Pouchet. Polybench: The polyhedral benchmark suite. *URL: <http://www.cs.ucla.edu/~pouchet/software/polybench/>*[cited July,], 2012.