Module M1 STIC - Initiation à la recherche

Sujets proposés par Frédéric Gava et Frédéric Loulergue

1 Applications en Bulk Synchronous Parallel ML

Certains problèmes, comme la simulation de phénomènes physiques, biologiques ou chimiques ou la gestion de bases de données de grande taille, nécessitent des performances que seules les machines *massivement parallèles* peuvent offrir. Leur programmation demeure néanmoins plus difficile que celle des machines séquentielles. La conception de langages adaptés, la conception et implantation d'algorithmes parallèles sont des sujets de recherche actifs.

Le *parallélisme de données* est un paradigme de programmation parallèle dans lequel un programme décrit une séquence d'actions sur des valeurs parallèles. Le modèle *BSP* [11] vise à maximiser la portabilité des performances en ajoutant une notion de *processus explicites* au parallélisme de données. Un programme BSP est écrit en fonction du nombre de processeurs de l'architecture sur laquelle il s'exécute. Le modèle d'exécution BSP sépare synchronisation et communication et oblige les deux à être des *opérations collectives*. Il propose un modèle de coût fiable et simple permettant de prévoir les performances de façon réaliste et portable.

Bulk Synchronous Parallel ML est un langage fonctionnel pour la programmation parallèle BSP. Il est issu de recherches menées au LIFO (Orléans), dans le projet "Parallélisme Réalité virtuelle et Vérification de systèmes" (PRV) et au LACL (Paris XII), dans l'équipe "Systèmes communicants". Il existe une implantation partielle de ce langage sous forme d'une *bibliothèque* pour le langage Objective Caml. Cette bibliothèque permet d'écrire des programmes parallèles BSP portable sur une grande variété d'architectures allant du PC à deux processeurs au système massivement parallèle Cray comprenant plusieurs centaines de processeurs, en passant par des grappes de PC.

Les sujets qui suivent proposent *l'implantation* de divers algorithmes BSP en BSML. Les tests des implantations seront effectués tout d'abord sur un simulateur parallèle puis sur une grappe de PC.

1.1 Sujet 1 : Diagrammes de Voronoi

Les diagrammes de Voronoi sont très importants dans le domaine de la robotique par exemple. Ils permettent de fournir des parcours pour des robots à partir de la donnée de zones à éviter. [5] décrit un algorithme BSP pour la construction d'un diagramme de Voronoi. Le but de ce stage consiste à implanter cet algorithme, le comparer avec une version séquentielle [2] (celle-ci est de toute façon nécessaire pour la version parallèle) et de créer une petite application de visualisation des diagrammes, des zones à risques et d'un chemin possible.

1.2 Sujet 2 : Recherche de motifs

La recherche de motifs est sujet de recherche très actif ayant de nombreuses applications pratiques. La recherche d'un motif consiste (entre autres) à retrouver le (les) occurence(s) d'une suite de caractères dans un texte. L'application première fut la recherche d'un mot clé dans un texte ou une base de données. On peut aussi l'utiliser en biologie, par exemple pour la recherche de gènes dans une séquence ADN. [6] décrit un algorithme BSP pour la recherche de différents motifs dans un texte. Le but de ce sujet consiste à implanter cet algorithme, le comparer avec une version séquentielle [9, 2] (celle-ci est de toute façon nécessaire pour la version parallèle).

1.3 Sujet 3 : Transformée de Fourier Rapide

La transformée de Fourrier rapide (FFT) est extrêmement utilisée dans toutes les applications numériques, par exemple dans les télécommunications (votre téléphone portable l'utilise en permanence pour transformer le son en données numériques) ou en physique. [10, 3] proposent des algorithmes BSP classiques (et simples) pour calculer la FFT. [7] propose un autre algorithme plus efficace en théorie. Le but de ce stage est d'implémenter ces deux algorithmes et de les comparer. Une petite recherche bibliographique sur la FFT et ces applications pratiques sera aussi demandée.

1.4 Sujet 4 : Graphes

Les graphes sont une structure de donnés très importante. [4, 12] proposent divers algorithmes BSP sur les graphes. Le but de ce stage est d'implanter un maximum de ces algorithmes pour créer une bibliothèque d'algorithmes parallèles sur les graphes comme il en existe déjà en Objective Caml (mais en séquentiel).

2 Types de données et filtrage par application de fonctions

[8] propose une approche dans laquelle les structures de données et le filtrage de motifs dans un langage fonctionnel sont compilées sous forme d'application de fonctions. Ce type de codage existe depuis le λ-calcul [1], mais les auteurs indiquent que celui-ci est beaucoup plus efficace que les approches existantes.

L'objectif de ce sujet est de lire et comprendre cet article, de proposer une implantation en Objective Caml et de comparer l'efficacité avec des programmes Objective Caml habituels.

Références

[1] H. P. Barendregt. *The Lambda Calculus : Its Syntax and Semantics*. North Holland, 1984.

[2] D. Beauquier, J. Berstel, and Ph. Chrétienne. *Eléments d'algorithmique*. Masson, 1992.

[3] R. Bisseling. *Parallel Scientific Computation. A structured approach using BSP and MPI*. Oxford University Press, 2004.

[4] F. Dehne, A. Ferreira, A. Caceres, S. W. Song, and A. Roncato. Efficient Prallel Graph Algorithms for CGM and BSP. *Algorithmica*, 33(0) :183–200, 2002.

[5] M. Diallo, A. Ferreira, and A. Rau-Chaplin. A Note on Communication-Efficient Deterministic Parallel Algorithms for Planar Point Location and 2d Voronoï Diagram. *Parallel Processing Letters*, 11(2) :327–340, 2001.

[6] P. Ferragina and F. Luccio. String Search in Coarse-Grained Parallel Computers. *Algorithmica*, 24(0) :177–194, 1999.

[7] M. A. Inda and R. H. BIsselingq. A simple and afficient parallel FFT algorithm using the BSP model. *Parallel Computing*, 27(14) :1847–1878, 2001.

[8] J.M. Jansen, P. Koopman, and R. Plasmeijer. Data Types and Pattern Matching by Function Application. In *Workshop on Implementation of Functional Languages and Applications*, 2005.

[9] T. Lecroq, Ch. Charras, and C. Hancart. Algorithmes de recherche exacte d'un mot. Web pages at http ://www-igm.univ-mlv.fr/ lecroq, 2004.

[10] W. F. McColl. The BSP Approach to Architecture Independant Parallel Programming. Technical report, Oxford University, December 1994.

[11] D. B. Skillicorn, J. M. D. Hill, and W. F. McColl. Questions and Answers about BSP. *Scientific Programming*, 6(3) :249–274, 1997.

[12] S. W. Song. Paralle lgraph algorithms, 1999.