Analyse et amélioration des performances d'un intergiciel JAVA pour l'algorithmique des graphes distribués

Sid Touati*et Luc Hogie†

Janvier 2015

Sujet adaptable pour trois, deux ou un seul étudiant motivé

Description du sujet

Lieu

INRIA-Sophia, équipe-projet AOSTE.

Prérequis

Maîtrise du langage de programmation Java. Connaissances en programmation parallèle.

Informations complémentaires

BigGrph est un intergiciel pour la recherche et le développement d'algorithmes distribués sur les grands graphes. Sur ce thème, il se pose en concurrent des intergiciels Giraph et GraphX. Par rapport à ces derniers, BigGrph a pour objectif de proposer à ses utilisateurs :

- une utilisation simplifiée autant que possible (simplifier la configuration du logiciel distribué) ;
- le support de nombreux modèles de programmations (centralisé, map/reduce, BSP, asynchronous message-passing, mobile agent-based), les autres intergiciels ne supportant que Map/Reduce et BSP ;
- une exécution plus rapide grace à de nombreuses optimisations réalisés au niveau des structures de données et des protocoles de communication ;
- le support de machines hétérogènes de calcul (grace à JAVA)
- une exécution multi-utilisateur (plusieurs chercheurs peuvent l'utiliser en meme temps sur le meme cluster de calcul) ;
- un API réduite qui permet une immersion plus rapide du développeur.

BigGrph est un intergiciel codé en Java par Luc Hogie. Il est développé depuis mi-2014 au laboratoire I3S de l'Université de Nice-Sophia Antipolis, en collaboration avec l'INRIA-Sophia.

Programme du stage : Dans le cadre de ce stage, nous proposons de faire une étude des performances de lui BigGrph et apporter des améliorations. Le programme de travail est le suivant :

*Professeur à l'UNS, Sid.Touati@inria.fr
†Ingénieur de recherche au CNRS, Luc.Hogie@inria.fr

1. Éffectuer un profilage des performances de BigGrph sur des machines installées à l'INRIA. Cette première étape permettrait de détecter les goulots d'étranglement de BigGrph, d'évaluer sa capacité à traiter des graphes de grande taille, etc.
2. Modifier un peu le code source JAVA de BigGrph pour améliorer ses performances. Dans ce cadre, étudier si l'utilisation de *threads* JAVA permettrait d'améliorer le temps d'exécution de certaines routines de la librairie.
3. Étudier la possibilité d'optimiser certain paramètres de la JVM pour accélérer le temps d'exécution des routines de la librairie.