

Visualisation radiale : approche parallèle entre CPU et GPU

Tianyang Liu*, Fatma Bouali**,*, Gilles Venturini*

* Université François-Rabelais de Tours, Laboratoire d'Informatique
64 avenue Jean Portalis, 37200 Tours, France, venturini@univ-tours.fr

** Université de Lille2, IUT, Dpt STID
25-27 Rue du Maréchal Foch, 59100 Roubaix, France, fatma.bouali@univ-lille2.fr

Résumé. Dans cet article, nous proposons une parallélisation sur CPU et GPU d'une méthode de visualisation radiale à base de points d'intérêt. Nous montrons que cette approche peut visualiser avec des temps très courts des millions de données sur des dizaines de dimensions, et nous étudions l'efficacité de la parallélisation dans différentes configurations.

1 Introduction

Traiter des données de plus en plus volumineuses fait partie des défis que se donne la communauté des chercheurs en fouille de données. Une des manières de répondre à ce défi est de définir des méthodes de fouille de données mettant en oeuvre le parallélisme, sur une ou plusieurs machines, et en utilisant le processeur (CPU) et/ou le calculateur graphique (GPU). Un certain nombre d'algorithmes ont déjà donné lieu à parallélisation sur GPU à la fois en fouille de données (voir un aperçu dans (Jian et al., 2011)), en visualisation scientifique (Weiskopf, 2006) ou en visualisation d'informations et plus spécifiquement en affichage de graphes (Frishman et Tal, 2007).

Dans le domaine de la fouille visuelle de données, les approches pouvant afficher le plus de données relèvent soit des approches orientées pixels et assimilées (voir (Keim et al., 1995) où 530.000 valeurs sont visualisées), soit des approches utilisant des densités (voir (Fua et al., 1999) où les coordonnées parallèles peuvent afficher jusqu'à 200.000 données). Cependant on peut constater que les interactions sont le point d'achoppement de ces méthodes visuelles pour le traitement de grands volumes de données (voir discussion dans (Florek, 2006)). Lorsque l'utilisateur formule une requête graphique, le système doit bien souvent faire des calculs supplémentaires et réafficher les données, partiellement ou entièrement, ce qui peut ralentir l'utilisateur dans son processus d'exploration (et rendre la méthode inutilisable).

Des approches de visualisation ont été étudiées pour remédier à ces inconvénients, comme pour le MDS (Ingram et al., 2009) en abaissant à la fois la complexité algorithmique et en parallélisant la méthode sur GPU. Cependant cette complexité est encore importante si l'on veut traiter des millions de données et il s'agit d'une approche peu interactive (sans redéfinition de l'espace de visualisation). Dans (Florek, 2006), l'objectif a été cette fois d'obtenir des temps d'affichage permettant les interactions. Le facteur d'accélération entre l'implémentation sur CPU (non parallélisée) et l'implémentation sur GPU va jusqu'à 60 (150.000 données en dimension 4 sont affichées en 0.3s sur GPU au lieu de 20s sur CPU).