Accélération multi-GPU pour l'évaluation polynomiale

Le but de ce stage M2 est d'exploiter la puissance de calcul des GPU pour obtenir l'évaluation polynomiale la plus rapide possible, cette opération étant majoritaire en temps dans plusieurs problèmes cruciaux en calcul formel. Il permet d'acquérir une première expérience significative en programmation GPU et en HPC.

Encadrants :

Pierre Fortin	`Pierre.Fortin@lip6.fr`	`http://lip6.fr/Pierre.Fortin`
François Lemaire	`Francois.Lemaire@univ-lille.fr`	`http://www.fil.univ-lille1.fr/~lemairef`

Laboratoire d'accueil :
CRIStAL (`http://www.cristal.univ-lille.fr/`), Université de Lille

Equipe d'accueil :
équipe CFHP (*Calcul Formel et Haute Performance*, `http://cfhp.univ-lille.fr/`)

Durée : 5 à 6 mois, à partir de février ou mars 2020

Financement : indemnité standard de l'Université de Lille

Keywords : GPU, évaluation polynomiale, arithmétique modulaire, MPI, HPC

Contexte scientifique :
Les temps de calcul des solutions de problèmes cruciaux en calcul formel, comme la factorisation polynomiale [1] et le calcul du PGCD de deux polynômes [1], sont dominés par les évaluations partielles de polynômes en plusieurs variables via de l'arithmétique modulaire.

Dans le cadre d'une collaboration avec le Professeur M. Monagan (Simon Fraser University, Canada), dont le but est d'accélérer d'un facteur 1 million de tels algorithmes, nous étudions les implémentations HPC de tels calculs. Après avoir apporté l'an passé un facteur 10 sur chaque coeur CPU, nous souhaitons désormais étudier l'apport des GPU pour accélérer ces évaluations polynomiales.

Objectifs :
A partir d'un code de référence sur CPU, le but de ce stage sera donc de déployer puis d'optimiser ce code sur GPU. Afin de tirer parti au maximum de la puissance de calcul des GPU, il faudra s'attaquer aux difficultés suivantes : implémentation efficace de l'arithmétique modulaire sur GPU, et minimisation des coûts liés aux transferts CPU-GPU et aux accès mémoire au sein du GPU. On utilisera des GPU HPC (NVIDIA Tesla P100 et V100) disposant de 3584 à 5120 coeurs de calcul via une programmation CUDA [2].

On visera ensuite à exploiter conjointement plusieurs GPU, d'abord sur un seul noeud de calcul puis sur plusieurs noeuds grâce à la mise en place d'une programmation parallèle multi-processus (via le standard MPI [3]) sur la plate-forme nationale Grid'5000 [4]. Enfin, le dernier objectif d'une

exécution hybride CPU multi-coeur / GPU sur plusieurs noeuds de calcul nécessitera de s'attaquer aux problématiques d'équilibrage de charge associées.

Compétences requises : programmation en C, goût pour le HPC (programmation et algorithmique) et pour le calcul scientifique, forte motivation, rigueur et capacités d'analyse. Aucune compétence en programmation GPU n'est requise.

Possibilités de poursuite en thèse dans le domaine du HPC et/ou du calcul formel.

Références :

[1] Knuth DE. The Art of Computer Programming. Volume 2 (3rd Ed.) : Seminumerical Algorithms. Boston, MA, USA : Addison-Wesley, 1997.

[2] `https://developer.nvidia.com/cuda-zone`

[3] `https://www.mpi-forum.org`

[4] `https://www.grid5000.fr`