

HPC pour BLINEIDE

Encadrants :

François Boulrier	Francois.Boulrier@univ-lille.fr	http://cristal.univ-lille.fr/~boulrier
Pierre Fortin	Pierre.Fortin@lip6.fr	http://lip6.fr/Pierre.Fortin
François Lemaire	Francois.Lemaire@univ-lille.fr	http://cristal.univ-lille.fr/~lemaire

Laboratoire d'accueil :

Sorbonne Université, Laboratoire d'Informatique de Paris 6 (LIP6), Jussieu, Paris.

Equipe d'accueil :

Equipe PEQUAN (PERformance et QUalité des Algorithmes Numériques)

Durée : 5 à 6 mois, à partir de février ou mars 2019

Financement : indemnité standard de Sorbonne Université

Keywords : HPC, OpenCL, algorithmique numérique, intégration numérique, GPU, SIMD, programmation par tâches

Contexte scientifique :

L'équipe CFHP est spécialiste du traitement des équations différentielles ordinaires (ODE en anglais). Dans le cadre d'une application à la biologie qui vise à mieux comprendre et modéliser l'interaction Neurone-Astrocyte qui joue un rôle important lors des accidents vasculaires cérébraux, l'équipe s'intéresse aux équations intégro-différentielles (IDE) [1]. Ces IDE ont la particularité de faire intervenir des termes intégraux, et permettent de modéliser des phénomènes complexes dépendant de l'état passé du système, comme par exemple celui de la toxicité d'un produit qui diminuerait dans le temps. La bibliothèque BLINEIDE [2] (développée dans l'équipe) permet d'intégrer numériquement les IDE en s'appuyant sur les méthodes d'Euler, Runge-Kutta, et de Simpson [3].

Cette bibliothèque vient d'être déployée sur CPU multi-cœur via du parallélisme de tâches, et nous souhaitons désormais l'accélérer davantage en exploitant d'autres niveaux de parallélisme et d'autres architectures de calcul haute performance (*high performance computing* - HPC).

Objectifs :

Le premier objectif est de concevoir un code portable sur diverses architectures HPC : CPU multi-cœur, GPU discret, GPU intégré, voire FPGA à plus long terme. On utilisera pour cela le standard OpenCL [4]. L'enjeu sera ici de concevoir une plateforme logicielle adaptée à ces différentes architectures, et de pouvoir intégrer automatiquement des IDE fournies par l'utilisateur.

On s'intéressera ensuite à l'optimisation des performances sur ces diverses architectures, avec des enjeux spécifiques sur chacune. Sur CPU multi-cœur on visera une vectorisation effective des noyaux de calcul sur les unités SIMD des cœurs CPU : ces unités SIMD (AVX2 ou AVX-512) peuvent en

effet offrir des gains de performance allant jusqu'à 16x. Sur GPU, on s'intéressera à l'impact sur les performances des transferts de données via le bus PCI : on pourra ainsi comparer les GPU discrets (puissants mais accessibles via le bus PCI) aux GPU intégrés (moins puissants mais ne nécessitant pas ces transferts PCI).

Enfin, on pourra étudier une exécution hybride CPU-GPU. Pour cela, on pourra étendre le parallélisme par tâche existant dans BLINEIDE aux architectures hybrides à l'aide de systèmes de tâches avancés comme StarPU [5]. Ceci pourra aussi permettre d'optimiser les performances sur GPU discrets en exploitant tous les niveaux de parallélisme offerts par BLINEIDE.

Le matériel adéquat sera mis à disposition : un serveur HPC avec deux processeurs à 18 coeurs chacun (deux CPU Intel Xeon E5-2695 v4, avec 72 threads matériels), ainsi que des serveurs intégrant des GPU haut de gamme pour le HPC (GPU NVIDIA K40 et P100, avec respectivement 2880 et 3584 coeurs GPU), et des GPU intégrés AMD et Intel.

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 n'est requise en algorithmique numérique.

Des compétences en HPC sont bienvenues, mais non nécessaires.

Ce stage permet d'acquérir une première expérience significative en HPC et en algorithmique numérique, et offre des possibilités de poursuite en thèse.

Références :

- [1] François Boulier, Hélène Castel, Nathalie Corson, Valentina Lanza, François Lemaire, Adrien Poteaux, Alban Quadrat, and Nathalie Verdière. *Symbolic-Numeric Methods for Nonlinear Integro-Differential Modeling*. Computer Algebra and Scientific Computing (CASC) 2018.
- [2] François Boulier. BLINEIDE. <http://cristal.univ-lille.fr/~boulier/BLINEIDE>
- [3] François Boulier. Cours de calcul numérique.
<http://cristal.univ-lille.fr/~boulier/polycopies/CNUM2A5/support.pdf>
- [4] Standard OpenCL. <https://www.khronos.org/opencv/>
- [5] StarPU, a task programming library for hybrid architectures.
<http://starpu.gforge.inria.fr/>