Sujet de thèse :
Modèles de programmation innovants pour le développement d'applications scientifiques sur architectures exascale.

Ce sujet de thèse s'adresse aux étudiant.e.s détenteurs.ses d'un master 2 en informatique ou calcul scientifique avec une forte composante calcul haute-performance.

Encadrant : Mathieu Lobet
Directeur de thèse : Edouard Audit
Laboratoire : Maison de la Simulation, CEA Saclay (CEA Saclay, 91191 Gif-sur-Yvette Cedex)
Contact : Mathieu Lobet (mathieu.lobet@cea.fr)

Contexte

Les supercalculateurs sont de plus en plus hétérogènes. Les nœuds de calcul se composent de plusieurs sockets CPU couplés à plusieurs accélérateurs le plus souvent GPGPU (General Purpose Graphics Processing Units). Cette hybridation des nœuds de calcul a progressivement commencé dans les années 2010 pour atteindre aujourd'hui presque 50% des 100 supercalculateurs les plus puissants au monde. Cette tendance s'accentue depuis quelques années dans un contexte d'amélioration croissante des technologies GPU pour le calcul et l'intelligence artificielle à la fois au niveau matériel (performance, consommation énergétique, mémoire embarquée) mais aussi logiciel (maturité des modèles de programmation, développement de bibliothèques, etc). Si pendant longtemps NVIDIA a dominé le marché GPU du Calcul Haute Performance (HPC), d'autres acteurs viennent aujourd'hui diversifier les technologies disponibles comme AMD ou Intel. Ces nouvelles architectures sont notamment favorisées par la course à l'Exascale, visant à atteindre une puissance de calcul de 10^{18} opérations arithmétiques par secondes.

La démocratisation croissante des GPUs, notamment pour le calcul scientifique, s'est accompagnée de l'émergence de nouveaux modèles de programmation. Par simplification, nous distinguons suivant trois catégories majeures : les langages relativement bas niveau fournis par les constructeurs ou supportés par certains d'entre eux (CUDA, OpenCL, etc), les modèles à base de directive (OpenACC, OpenMP) et les modèles à haut niveau d'abstraction (Kokkos, RAJA, Alpaka, SYCL, etc). Tous ces modèles se comparent sur la base de plusieurs critères que sont la performance (par rapport à la puissance crête de la machine), la portabilité (capacité du code à pouvoir s'adapter à

plusieurs architectures avec un minimum d'effort), la maturité du modèle, la facilité d'apprentissage, la facilité de mise en production, la maintenabilité, la modularité et plus encore. Avec l'arrivée de multiples architectures GPU, la portabilité est devenue un enjeu prioritaire dans le choix d'un modèle pour la modernisation ou l'écriture d'un code. Elle permet à partir d'une implémentation unique de tourner sur un grand nombre d'architectures à la fois CPU et GPU, mais aussi de s'assurer une compatibilité avec les futures technologies à venir. Les développeurs, souvent scientifiques, minimisent ainsi les efforts de réécritures au profit d'une meilleure productivité.

Les modèles de programmation à haut niveau d'abstraction permettent, comme leur nom l'indique, d'abstraire les structures de données, la gestion de la mémoire et le parallélisme sur les architectures visées. Pour cela, ils utilisent en arrière-plan les langages bas niveau (backends CUDA par exemple) et au premier plan la puissance des derniers standards C++ (notamment C++17). Ils permettent en peu de lignes de code de proposer des implémentations à la fois portables et performantes. Développés depuis presque 10 ans pour certains, ils sont encore relativement peu utilisés dans le milieu scientifique sur des codes de production. Néanmoins, un grand nombre d'entre eux arrivent maintenant à maturité. Ils sont poussés et supportés par la plupart des constructeurs.

Objectifs

Cette thèse a pour objectif premier l'exploration de ces nouveaux modèles de programmation dans un contexte de développement d'applications scientifiques pour les futures architectures exascales. Le modèle qui nous intéresse le plus est SYCL développé par le Khronos Group (https://www.khronos.org/sycl/). Il a l'avantage d'être aujourd'hui adopté par de nombreux constructeurs (AMD, Intel, XILINX et d'autres) et d'être ainsi supporté par de nombreuses bibliothèques logicielles parallèles. Il est l'un des principaux modèles proposés par Intel au sein de son implémentation DPC++ pour programmer ses futurs GPU (Ponte Vecchio) mais aussi ses futures cartes FPGA (Agilex).

Le ou la candidate aura pour rôle d'explorer l'implémentation SYCL de plusieurs noyaux de calcul scientifique. Un des premiers noyaux envisagés correspond à la méthode ``Particle-In-Cell' utilisée pour la simulation des plasmas dans plusieurs codes au CEA. D'autres noyaux provenant de codes développés ou supportés à la Maison de la Simulation pourront être explorés (astrophysique, dynamique moléculaire, dynamique des fluides). SYCL et les noyaux implémentés seront testés sur des systèmes en développement pour l'Exascale européen. Un partenariat avec SiPearl et Intel dans le cadre du projet EoCoE-III permettra de tester la solution sur des nœuds hétérogènes couplant le processeur ARM Rhea et les GPU Ponte Vecchio. Les futures architectures des autres constructeurs seront également testées progressivement avec l'arrivée des futures machines européennes et française.

Le second objectif de la thèse sera d'explorer la programmation des accélérateurs FPGA (Field Programmable Gate Array) toujours dans un contexte scientifique. Dans un but de performance et d'efficacité énergétique, l'architecture FPGA pourrait devenir une technologie accélératrice pertinente aux côtés des CPUs et GPUs et peut-être équiper les futures machines post-exascale. De nombreux constructeurs poussent dans cette direction même si l'on est aujourd'hui encore en phase expérimentale dans un cadre HPC. Intel est ici encore un acteur en première ligne et propose depuis peu une nouvelle carte accélératrice Agilex programmable grâce au modèle de programmation DPC++ basé sur SYCL pour les applications industrielles et scientifiques. Le ou la candidate aura accès

par l'intermédiaire d'un partenariat entre le CEA et Intel à des cartes Agilex afin d'étudier la performance des noyaux scientifiques sur ce type d'architecture.

Pour chaque objectif, le modèle sera évalué suivant divers critères à définir plus précisément au cours de la thèse comme la portabilité réelle, la performance par rapport aux autres modèles de programmation, la facilité de prise en main, la maturité, la complexité du code obtenu, etc. En fonction de son affinité et de son avancement, le ou la candidate retenue pourra choisir les noyaux ou les architectures sur lesquels renforcer ses études. En revanche, le but n'est pas de développer un nouveau code scientifique.

Cette thèse porte sur un sujet qui convient à un ou une candidate désireuse de mener un projet de recherche en informatique et de poursuivre ensuite sa carrière dans la recherche aussi bien que dans l'industrie.

Compétences

- Connaissance approfondie de la programmation logicielle en C++ moderne
- Connaissance en programmation parallèle (programmation des accélérateurs GPU est un plus)
- Appétence pour la recherche
- Bonne capacité à travailler en anglais (à l'écrit et à l'oral)
- Un intérêt pour les mathématiques appliqués et la simulation numérique est un plus