Année académique / Academic year 2022-2023

Sujet de stage SERMA / SERMA internship topic N° – Laboratoire/Laboratory : LTSD

Titre du stage

DÉVELOPPEMENT D'UN ORDONNANCEUR DYNAMIQUE DE CALCUL HYBRIDE GPU (CUDA)/CPU

Internship title

DEVELOPMENT OF A HYBRID GPU (CUDA)/CPU DYNAMIC SCHEDULER

Type de sujet / Topic type*

- *Études et benchmarking Studies and benchmarking*
- *Développement de méthodes et de codes de calcul / Methods and computational codes development*

Contexte du stage

Patmos est un code-maquette de simulation de transport de particules pour la physique des réacteurs utilisant la méthode Monte-Carlo [1]. Cette méthode consiste à échantillonner aléatoirement les trajectoires des particules et leurs interactions avec le milieu dans lequel elles se déplacent, selon des lois de probabilités déterminées par la physique sous-jacente. Le Monte-Carlo n'introduit quasiment pas d'approximations dans la résolution de l'équation de transport et n'a pas besoin de discrétiser l'espace des phases, contrairement aux méthodes dites déterministes ; pour cette raison, le Monte-Carlo est considérée la méthode de référence dans le domaine.

Les probabilités déterminant les interactions particule-matière sont décrites par des grandeurs élémentaires appelées **sections efficaces**. Les sections efficaces proviennent de bibliothèques de données nucléaires, fournies en entrée de l'application. Toutefois, stocker la totalité des sections efficaces s'avère extrêmement coûteux en terme de consommation mémoire, notamment pour les applications multi-physiques, où des jeux de sections efficaces pour chaque isotope présent dans les compositions matérielles du réacteur et pour toute température (effets d'élargissement Doppler) doivent être utilisés pendant la simulation. **L'élargissement Doppler « au vol »** est une technique proposée récemment, permettant de calculer les sections efficaces à une température donnée à partir de données nucléaires à une température de référence, par le biais d'un calcul effectué à chaque collision des particules. Le fait de devoir stocker uniquement un jeu de sections efficaces par isotope (et non plus par isotope et par température) rend l'encombrement mémoire nettement moins important. Cette approche implique néanmoins une étape de calcul (l'élargissement Doppler au vol, à chaque collision) représentant une part importante du temps total de calcul de la simulation de la vie des particules.

Patmos est développé au CEA dans le but de s'adapter aux nouvelles architectures logicielles et matérielles [2]. Les architectures **GPU** sont fortement représentées dans le paysage des supercalculateurs modernes. Cette technologie offre une puissance de calcul conséquente, capable de réduire considérablement le temps de calcul d'une simulation. C'est pourquoi le **portage du calcul de l'élargissement Doppler des sections efficaces** sur GPU a été étudié dans des travaux de thèse [3] : cela permet de réduire considérablement le temps de calcul en comparaison de la réalisation de ce même calcul sur **CPU**. Cependant, les architectures GPU actuelles se composent toujours d'une partie CPU, indispensable au déroulement normal d'une application. L'implémentation actuelle de portage sur GPU permet d'utiliser les ressources GPU mises à disposition, mais implique une inactivité CPU (en termes de calcul) durant la phase de calcul sur GPU et donc une utilisation des ressources non optimale. Il est donc souhaitable de définir **une stratégie de portage permettant d'exploiter la totalité des ressources**, y compris CPU.

Internship context

Patmos is a particle transport simulation code-model for reactor physics using the Monte-Carlo method [1]. This method consists of randomly sampling the trajectories of particles and their interactions with the medium in which they move, according to probability laws determined by the underlying physics. Monte-Carlo introduces almost no approximations into the resolution of the transport equation and does not need to discretize the phase

space, unlike so-called deterministic methods; for this reason, Monte-Carlo is considered the reference method in the field. The probabilities determining the particle-matter interactions are described by elementary quantities called cross sections. The cross sections come from nuclear data libraries, provided as input to the application. However, storing all the cross sections is extremely expensive in terms of memory consumption, especially for multi-physics applications, where sets of cross sections for each isotope present in the material compositions of the reactor and for any temperature (Doppler broadening effects) should be used during the simulation. **"On-the-fly" Doppler broadening** is a recently proposed technique for calculating cross sections at a target temperature based on nuclear data at a reference temperature, by means of a calculation performed at each particle collision. The fact of having to store only one set of cross sections per isotope (and no longer per isotope and per temperature) makes the memory footprint much smaller. This approach nevertheless involves a calculation step (Doppler broadening of the cross sections on the fly, at each collision) representing a significant part of the total calculation time involved in the simulation of the life of the particles.
Patmos/Tripoli-5 is developed at the CEA in order to adapt Monte-Carlo algorithms to new software and hardware architectures [2]. GPU architectures are widely represented in the landscape of modern supercomputers. This technology offers substantial computing power, capable of considerably reducing the calculation time of a simulation. This is why the offload of the calculation of the Doppler broadening of the cross sections on GPU has been studied in a PhD thesis at CEA [3]: this makes it possible to considerably reduce the calculation time in comparison with the realization of this same calculation on CPUs. However, current GPU architectures always consist of a CPU part, which is essential for the normal running of an application. Offloading to GPU allows using all the available GPU resources, but implies CPU inactivity (in terms of calculation) during the calculation phase on GPU and therefore a non-optimal use of resources. It is therefore desirable to define an offload strategy in order to exploit all the resources, including CPU.

Description du sujet du stage

Pour résoudre le problème de l'inutilisation des cœurs CPU durant la phase de calcul sur GPU une approche **hybride GPU/CPU** est possible. Le nombre de sections efficaces à calculer dans une étape de calcul étant relativement élevé, il est aisé de répartir la charge de calcul sur les composants matériels (CPU et GPU). Néanmoins, la question de la proportion de calcul à porter sur GPU se pose. S'il est admis qu'une répartition équitable du calcul serait non optimale – le GPU étant à priori plus rapide – nous ne pouvons pas facilement et de manière précise définir à l'avance la meilleure répartition. La solution pour répondre à ce problème est donc un **ordonnancement dynamique du calcul des sections efficaces.** Pour cela, plusieurs approches ont été étudiées dans la littérature [4]. Dans le cas du calcul des sections efficaces nous nous intéressons notamment aux approches **d'ordonnancement en fonction des performances relatives des CPU/GPU**.
L'objectif de ce stage est de proposer un **ordonnancement dynamique du calcul des sections efficaces** sur **CPU** et **GPU** en utilisant une approche de **performances relatives**. En particulier, le stagiaire proposera une solution hybride pour l'implémentation **CUDA** en s'appuyant notamment sur l'API fournie [5]. Plusieurs algorithmes d'ordonnancement pourront être mis en œuvre. Ces derniers pourront notamment s'inspirer des solutions existantes dans la littérature [4]. Ce développement sera réalisé au sein d'application Patmos.
L'ordonnanceur sera évalué en termes de performances sur des cas représentatifs et sur une machine HPC. Ces tests permettront de mettre en lumière la plus-value de la solution proposée. Aussi, des tests de validation des résultats physiques seront effectués afin de garantir l'absence d'influence de l'ordonnancement sur la physique simulée.

Internship topic description

To solve the problem of unused CPU cores during the calculation phase on GPU, a **hybrid GPU/CPU** approach is possible. The number of cross sections to be calculated in a calculation step being relatively high, it is easy to distribute the calculation load on the hardware components (CPU and GPU). Nevertheless, the question of the proportion of calculation to carry on GPU arises. If it is admitted that a fair distribution of the calculation would be non-optimal – the GPU being considered faster – we cannot easily and precisely define in advance the best distribution. The solution to this problem is therefore a **dynamic scheduling of the calculation of the cross**

sections. For this, several approaches have been studied in the literature [4]. In the case of the calculation of cross sections, we are particularly interested in **scheduling approaches according to the relative performances of GPU/CPU**.
The objective of this internship is to propose a **dynamic scheduling of the calculation of cross sections** on **CPU** and **GPU** using a relative performance approach. In particular, the student will propose a hybrid solution for the **CUDA** implementation based in particular on the provided API [5]. Several scheduling algorithms can be implemented. The latter can in particular be inspired by existing solutions in the literature [4]. This development will be carried out within the Patmos application.
The scheduler will be evaluated in terms of performance on representative cases and on an HPC machine. These tests will highlight the added value of the proposed solution. Also, validation tests of the physical results will be carried out in order to guarantee the absence of influence of the scheduling on the simulated physics.

Bibliographie - Références / Bibliography - References

[1] I. Lux, L. Koblinger, Monte Carlo particle transport methods (CRC press, 1990).
[2] E. Brun, S. Chauveau, F. Malvagi, PATMOS: A prototype Monte Carlo transport code to test high performance architectures, in Proc. M&C 2017, Jeju, Korea, April 16-20 (2017).
[3] T. Chang, Evaluation of programming models for anycore and/or heterogeneous architectures for Monte Carlo neutron transport codes, PhD thesis, Institut polytechnique de Paris (2020).
[4] Mittal, Sparsh, and Jeffrey S. Vetter, A survey of CPU-GPU heterogeneous computing techniques. ACM Computing Surveys (CSUR) 47, no. 4 (2015).
[5] Doc Nvidia CUDA : https://docs.nvidia.com/cuda/cuda-runtime-api/index.html

Ouverture éventuelle sur un sujet de thèse / Possible opening on a thesis proposal

Oui/Yes Non/No

Profil du stagiaire

Master 2 ou 3ème année d'école d'ingénieur en informatique avec une spécialisation HPC (GPU, OpenMP). Une sensibilisation à la physique des réacteurs serait un plus appréciable.

Applicant profile

Master of Science or Engineering diploma in computer science with an HPC specialization (GPU, OpenMP). An awareness of reactor physics would be appreciated.

Localisation du stage / Internship location

Commissariat à l'énergie atomique et aux énergies alternatives (CEA), Centre de Saclay
DES/ISAS/DM2S/SERMA – Bât. 470
91191 Gif-Sur-Yvette Cedex

Personne(s) contact(s) / Contact person(s)

Nom/Name : Gonçalves
Prénom/First name : Thomas
e-mail : thomas.goncalves@cea.fr
Téléphone/phone number : 01 69 08 95 44
Affiliation : DES/ISAS/DM2S/SERMA/LTSD