

Lab n° 1 : Introduction à OpenCL

L'objectif de ce TP est de se familiariser avec la bibliothèque *OpenCL*. Pour cela, vous utiliserez le langage *C++* ou le langage *C*. Le code devra être documenté.

Le rendu, que vous devrez commenter, peut-être soumis par courrier électronique pour une correction.

Exercice 1. Analyse de l'environnement

Le but de cet exercice est de manipuler l'environnement *OpenCL* et d'obtenir les informations concernant le matériel que vous allez utiliser dans la suite du cours.

Il est fortement recommandé de garder une version compilée de cet exercice afin de vérifier au cours de vos développements futurs les capacités de la carte que vous manipulez.

1. Etudier le fichier `DeviceInfo.c` fourni. Compiler le et exécuter le afin d'obtenir les informations sur l'environnement d'exécution.

Exercice 2. Addition de vecteurs

L'objectif de cet exercice est de vérifier que vous pouvez analyser et exécuter un noyau *OpenCL*.

Le fichier source `vadd.c.c` contient une implémentation en *C* de l'addition de deux vecteurs en utilisant un accélérateur.

1. Analyser le fichier source afin d'identifier les différents éléments qui permettent d'exécuter un noyau *OpenCL*.

Le fichier contient des fonctions d'affichage et de parsing des erreurs que seront réutilisés par la suite.

Exercice 3. Addition de vecteurs

L'objectif de cet exercice est de vérifier que vous pouvez analyser et exécuter un noyau *OpenCL*.

Le fichier source `vadd.cpp` contient une implémentation en *C* de l'addition de deux vecteurs en utilisant un accélérateur.

1. Analyser le fichier source afin d'identifier les différents éléments qui permettent d'exécuter un noyau *OpenCL*.
2. Comparer la complexité des API du code et du code CXX.

Le fichier contient des fonctions d'affichage et de parsing des erreurs que seront réutilisés par la suite.

Exercice 4. Addition en chaîne de vecteurs

L'objectif de cet exercice est de vérifier que vous comprenez l'invocation des noyaux et les buffers en *OpenCL*. Pour cela, vous pouvez partir de l'implémentation de l'addition de votre choix.

1. Ajouter des buffers complémentaires et associer les à des vecteurs complémentaires définis sur la partie hôte.
2. Chainer les additions $C = A + B$, $D = C + E$, $F = D + G$.
3. Lisez le résultat final et vérifier qu'il est correct.

Exercice 5. Produit matriciel

L'objectif de cet exercice est d'écrire un premier noyau *OpenCL* qui multiplie une paire de matrices entre elles.

Le fichier source *matmul.cpp* contient une implémentation en *C++* de la multiplication de deux matrices en séquentiel.

1. Créer un noyau de multiplication en *OpenCL*.
2. Modifier le programme hôte afin de pouvoir utiliser votre noyau.
3. Vérifier les résultats.

Exercice 6. Calcul de π

L'objectif de cet exercice est de comprendre les mécanismes de synchronisation entre les work-items en *OpenCL*.

Le fichier source *pi.cpp* contient une implémentation en *C++* du calcul de π par intégration numérique en séquentielle.

1. Créer un noyau de calcul de π en *OpenCL*.
2. Modifier le programme hôte afin de pouvoir utiliser votre noyau.
3. Vérifier les résultats.
4. Donner des pistes d'amélioration les performances de la méthodes.

Références : [1] <https://www.khronos.org/files/opencl-1-2-quick-reference-card.pdf>