

TP2

Objectifs : Organisation des threads

Ressources

- <http://docs.nvidia.com/cuda/index.html>
- https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf

Setup

- haswell-cuda ou 192.168.80.64
- Login : prenom.nom (minuscule)
- Passwd : cuda (à changer)
- Vérification bashrc, PATH et LD_LIBRARY_PATH

Git : <https://github.com/hpc-apps/Cuda.git>

Exercice 1

1. Créer un code permettant de récupérer le nombre de carte graphique disponible en utilisant **CudaGetDeviceCount()**
2. En utilisant **CudaGetDeviceProperties()** récupérer :
 - le nombre maximum de threads par bloc,
 - les dimensions maximales pour les blocs et les grilles,
 - le nombre de multiprocesseurs.

Exercice 2

1. Reprendre l'exemple d'addition de vecteur du TP1 et l'adapter pour que le nombre de blocs et de threads par bloc soit déterminé automatiquement à partir de la taille des données.
2. Optimiser la décomposition en fonction de la taille des données et les caractéristiques du GPU.

Exercice 3

1. Créer deux matrices de taille N x N et définir un kernel permettant de les additionner à partir de blocs 2D de threads.
2. Optimiser en fonction de la taille des données ainsi que des caractéristiques du GPU.
3. Tracer une courbe représentant le temps d'exécution du kernel en fonction de la taille du problème.