

TP1

Objectifs : Base de la programmation CUDA

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 : toto (à changer)
- Vérification bashrc, PATH et LD_LIBRARY_PATH

Depot Git :

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

Partie 1

Exercice 1

- Utiliser l'utilitaire *deviceQuery* afin d'identifier les « Compute Capabilities » .

Exercice 2

- Tester les exemples du répertoire Ex2
- Compilation : *nvcc -o exemple exemple.cu -std=c++11*
- Modifier la taille des tableaux, ceux-ci ne doivent pas dépasser 1024

Partie 2

Exercice 3

- Créer un programme qui initialise deux vecteurs, puis effectue leur somme.

Exercice 4

- Créer un programme qui initialise deux vecteurs, puis effectue leur somme.

Exercice 5

- *Rajouter la gestion des erreurs pour l'addition de deux vecteurs.*
 - Fonctions ***cudaMalloc***, ***cudaMemcpy***, . . . et valeur de type ***cudaError_t***
- Produire les erreurs suivantes et vérifier leur gestion :
 - Nombre de threads supérieur à 1024 pour le lancement du kernel.
 - Inversion des arguments pour ***cudaMemcpy*** (***HostToDevice***, ***DeviceToHost***).

Exercice 6

- L'API CUDA fournit donc des outils spécifiques pour mesurer le temps d'exécution
 - ***CUDA runtime API/Event management***
- Instrumenter le code et mesurer les temps d'exécution du kernel pour l'addition de deux vecteurs de taille 1024.