

TP3

Objectifs :

- Streams / mémoire shared

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

- Modifier l'exemple d'addition de vecteurs afin de d'utiliser une version utilisant les **streams** CUDA
 - o Chaque **stream** opère sur une sous-partie du vecteur (e.g deux streams).
 - o Utiliser les fonctions *CudaMallocHost*, *CudaStreamCreate* + destructeurs et *CudaMemcpyAsync*
 - o Commencer avec seul bloc

Exercice 2

- Rajouter l'utilisation de la mémoire **shared** à l'exemple précédent (un bloc)
- Evaluer le gain (ou l'absence de gain) en temps de calcul pour différentes tailles de problèmes.

Exercice 3

- Modifier l'exercice 2 afin de transformer l'addition de deux vecteurs en produit scalaire
- Utiliser la mémoire **shared**.
- Synchroniser les threads au niveau du kernel avant l'opération de réduction.
- Ecrire une première version en utilisant 1 seul bloc et sans utiliser les **streams**.

Exercice 4

- Compléter les exemple 2, 3 et 4 en utilisant **plusieurs blocs** afin de saturer la carte graphique