

Les objectifs de ce TD sont :

- Compréhension du modèle d'exécution
- Emulation sur CPU
- Calcul d'indice global

I Calcul d'indice global

Q.1: En utilisant les variables prédéfinies par CUDA dans un kernel, calculer l'indice global d'un block dans une grille 2D

Q.2: En utilisant les variables prédéfinies par CUDA dans un kernel, calculer l'indice global d'un block dans une grille 3D

Q.3: En utilisant les variables prédéfinies par CUDA dans un kernel, calculer la taille (nombre de threads) d'un bloc défini en 3D

Q.4: En utilisant les variables prédéfinies par CUDA dans un kernel, calculer l'indice global d'un thread dans un bloc 3D

Q.5: En vous servant des questions précédentes, calculer l'indice global d'un thread dans une grille 3D composée de blocs 3D

II Modèle d'exécution et SDK CUDA

Dans cette partie nous considérons le fichier td2.cu.

Q.6: Quelle partie du programme doit s'exécuter sur l'hôte ? Quelle partie sur le device ?

Q.7: Que calcule ce programme ? (si vous ne savez pas répondre à cette question, répondre à la suivante pourra vous aider).

Q.8: Combien y a-t-il de blocs au total ? Combien de threads par blocs ? Combien de threads au total ?

Q.9: Émuler sur CPU le comportement du GPU sans utiliser le SDK CUDA. Pour ce faire, réécrire le programme en C/C++ avec les contraintes suivantes :

1. utilisation de nouveaux tableaux à utiliser pour le "kernel"
2. copie des données en entrées et en sorties du "kernel"
3. utilisation d'une fonction `kernel`
4. utilisation des grilles de blocs et de threads
5. calcul d'un indice global pour accéder aux cases des tableaux