

Programmation sur Processeur Graphique – GPGPU

TD 9 : algorithme de réduction

Centrale Nantes

P.-E. Hladik, pehladik@ec-nantes.fr

—

Version bêta (19 décembre 2022)

1 Produit scalaire

Objectif 1.1

- observer l'influence des warps
- mettre en œuvre un algorithme d'arbre de réduction

(1.1) Travail à faire : Produit scalaire

Implémentez un *kernel* qui effectue le produit scalaire s (dot en anglais) pour deux vecteurs a et b , $s = \sum_{i=1..n} a_i \times b_i$.

Votre *kernel* devrait être capable de gérer des vecteurs d'entrée de longueur arbitraire. Pour vous aider, utilisez le squelette fourni dans le fichier `dot.cu` qui charge un tableau en mémoire partagée avec le produit de a et b (les conditions aux limites sont prises en charge ainsi que des vecteurs de grande taille – ce qui explique la présence du *while* au début du *kernel*).

Supposez aussi que les réductions produites par les blocs seront additionnées par le CPU (voir fin du code dans le fichier `dot.cu`).

Réalisez deux versions du calcul, l'une sans prendre en compte les warp et l'autre avec. Comparez les performances.