Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

CEG 4536 Architecture des ordinateurs III
Automne 2024

A soumettre le 3 décembre à 23h59.

Lab4 : Optimisation des kernels CUDA pour une utilisation efficace de la mémoire partagée

1. Introduction

Ce projet explore l'utilisation avancée de la mémoire partagée de CUDA afin d'améliorer les performances des programmes accélérés par GPU. Les étudiants examineront des techniques pour optimiser l'utilisation de la mémoire partagée, éviter les conflits de banque, et tirer parti du padding de mémoire et des instructions de shuffle de warp pour une gestion efficace des données. Le projet repose sur les concepts de la hiérarchie de mémoire de CUDA et de l'optimisation de la mémoire partagée.

2. Objectifs

Les objectifs principaux sont :

- Comprendre et implémenter les opérations de mémoire partagée dans les kernels CUDA.
- Optimiser les performances des kernels en réduisant les conflits de banque et les accès à la mémoire globale.
- Analyser les effets du padding et des instructions de shuffle de warp dans les réductions parallèles et les opérations matricielles.

3. Réalisation

Les étudiants devront :

1. Implémenter un kernel de transposition de matrice en utilisant la mémoire partagée et le padding pour éliminer les conflits de banque.
2. Créer des kernels de réduction parallèle optimisés en tirant parti de la mémoire partagée et des instructions de shuffle de warp.
3. Comparer les performances des kernels optimisés et non optimisés à l'aide de `nvprof` ou d'autres outils de profilage.
4. Évaluer l'impact de l'alignement mémoire et des stratégies de padding.

4. Application Exemple

Un kernel GPU qui transpose une matrice en utilisant la mémoire partagée avec des dispositions de mémoire en ligne et en colonne :

- Implémenter le kernel avec et sans padding mémoire.
- Profiler et comparer les performances pour les deux implémentations, en se concentrant sur la réduction des conflits de banque et les schémas d'accès à la mémoire.

-

5. Livrables

1. Code source pour les kernels de transposition de matrice et de réduction parallèle.
2. Un rapport d’analyse des performances comprenant :
    a. Les statistiques sur les conflits de banque.
    b. Les schémas d'accès à la mémoire.
    c. Les comparaisons des temps d'exécution.
3. Des visualisations de la disposition mémoire et des conflits de banque sous forme de diagrammes.
4. Une vidéo de démonstration illustrant l'exécution du kernel et les résultats du profilage.

6. Évaluation

Le projet sera évalué sur les critères suivants :

- **Fonctionnement (30%)** : L’implémentation respecte les bonnes pratiques de programmation CUDA.
- **Optimisation (30%)** : Utilisation efficace de la mémoire partagée et élimination des conflits de banque.
- **Performances (20%)** : Amélioration significative par rapport aux kernels non optimisés.
- **Présentation (20%)**: Documentation claire et explication des résultats dans le rapport et la vidéo.