

Travaux Pratiques - Découverte des GPUs

Alexandre Bordat

December 6, 2023

1 Introduction

Dans ces travaux pratiques, nous allons explorer les concepts liés aux unités de traitement graphique (GPU) et leur utilisation pour accélérer les calculs parallèles. Les GPU sont devenus des composants essentiels dans de nombreuses applications, notamment la science des données, la vision par ordinateur, la simulation, etc. Nous allons utiliser CUDA (Compute Unified Device Architecture) de NVIDIA pour travailler avec les GPU.

2 Travail Pratique 1 : Exploration des Capacités des GPU

2.1 Objectif

L'objectif de cette partie du travail pratique est de comparer les capacités de traitement parallèle d'un GPU à celles d'un CPU en effectuant des tâches de calcul intensif.

2.2 Exigences

- Un ordinateur équipé d'un GPU dédié.
- Connaissances de base en programmation (C ou C++).
- Accès à un kit de développement pour la programmation GPU (tel que CUDA, si vous avez un GPU NVIDIA disponible).

2.3 Description de l'exercice

Dans cette partie du travail pratique, vous allez effectuer une comparaison pratique des performances d'un CPU et d'un GPU en effectuant des opérations de calcul intensif. Les étapes à suivre sont les suivantes :

2.3.1 Tâche 1 : Comparaison de base des performances CPU vs GPU

1. Écrivez un programme simple qui effectue un grand nombre d'opérations mathématiques, telles que la multiplication de matrices. (Indice, cela implique une triple boucle for).
2. Exécutez ce programme en utilisant uniquement le CPU et mesurez le temps nécessaire pour terminer la tâche.
3. Modifiez le programme pour exécuter les mêmes opérations en utilisant le GPU. (Indice cela implique une simple boucle for et des indices de ligne et colonne savamment choisis).
4. Mesurez le temps pris par le GPU pour terminer la tâche et comparez-le avec les performances du CPU.

2.3.2 Tâche 2 : Analyse des Résultats

- Discutez des différences de performances observées entre le CPU et le GPU.
- Comprenez pourquoi le GPU a mieux (ou moins bien) performé pour cette tâche en tenant compte de facteurs tels que les capacités de traitement parallèle et la nature de la tâche.

2.4 Résultats d'Apprentissage

À la fin de cette partie du travail pratique, vous devriez être en mesure de :

- Acquérir une expérience pratique en programmation GPU de base.
- Comprendre les aspects pratiques des capacités de traitement parallèle des GPU.
- Analyser et interpréter les différences de performances entre le CPU et le GPU pour des tâches de calcul intensif.

3 Travail Pratique 2 : Comparaison de l'Entraînement de Modèles d'Apprentissage Automatique sur CPU vs. GPU

3.1 Objectif

Comprendre et quantifier les gains de performance dans l'entraînement de modèles d'apprentissage automatique lors de l'utilisation d'un GPU par rapport à un CPU.

3.2 Exigences

1. Un ordinateur équipé à la fois d'un CPU et d'un GPU (de préférence un GPU NVIDIA compatible avec CUDA pour une utilisation plus aisée avec de nombreux frameworks d'apprentissage automatique).
2. Un framework d'apprentissage automatique prenant en charge l'accélération GPU (comme TensorFlow, PyTorch).
3. Des connaissances de base en apprentissage automatique et de l'expérience en programmation en Python.

3.3 Description de la Tâche

1. Configuration de l'Environnement:

- (a) Installez un framework d'apprentissage automatique prenant en charge l'accélération GPU (on va utiliser TensorFlow).
- (b) Assurez-vous que les pilotes nécessaires et la trousse d'outils CUDA sont installés pour prendre en charge le GPU (si vous utilisez un GPU NVIDIA).

2. Sélection d'un Modèle d'Apprentissage Automatique et d'un Ensemble de Données:

- (a) Choisissez un modèle d'apprentissage automatique relativement simple à des fins de comparaison, comme un réseau neuronal convolutif (CNN) pour la classification d'images.
- (b) Utilisez CIFAR-10, couramment utilisé pour des benchmarks.

3. Entraînement du Modèle sur CPU:

- (a) Écrivez ou utilisez un script existant pour entraîner le modèle sélectionné sur l'ensemble de données choisi en utilisant uniquement le CPU.
- (b) Enregistrez le temps nécessaire pour entraîner le modèle pendant un nombre fixe d'époques (cycles d'entraînement). Notez les paramètres d'entraînement (comme la taille du lot et le taux d'apprentissage) pour garantir la cohérence.

4. Entraînement du Modèle sur GPU:

- (a) Modifiez le script pour entraîner le modèle sur le GPU. Les frameworks comme TensorFlow utilisent automatiquement le GPU s'il est disponible et configuré correctement.
- (b) Assurez-vous que les paramètres d'entraînement sont identiques à ceux de l'entraînement sur CPU pour une comparaison équitable.
- (c) Enregistrez le temps d'entraînement pour le même nombre d'époques que l'entraînement sur CPU.

5. Analyse et Comparaison:

- (a) Comparez les temps d'entraînement entre le CPU et le GPU.
- (b) Calculez le facteur d'accélération, qui est le rapport entre le temps d'entraînement sur CPU et le temps d'entraînement sur GPU.

6. Rapport et Discussion:

- (a) Rédigez un rapport discutant des résultats. Incluez des détails sur le modèle, l'ensemble de données, les paramètres d'entraînement, les spécifications matérielles et l'accélération observée.
- (b) Discutez pourquoi le GPU offre un avantage de performance, en mettant l'accent sur des aspects tels que les capacités de traitement parallèle du GPU.

3.4 Livrables

- 1. Un script ou un notebook contenant le code pour l'entraînement du modèle sur CPU et GPU.
- 2. Un rapport documentant la configuration de l'expérience, les résultats et l'analyse de la différence de performance.

3.5 Résultats d'Apprentissage

À la fin de cette partie du travail pratique, vous devriez être en mesure de :

- 1. Comprendre l'impact de l'accélération GPU dans l'entraînement de modèles d'apprentissage automatique.
- 2. Acquérir de l'expérience pratique dans l'entraînement de modèles en utilisant à la fois le CPU et le GPU.
- 3. Développer des perspectives sur la pertinence de l'utilisation du GPU pour différents types de tâches d'apprentissage automatique.

4 Travail Pratique 3 : Application du Flou Gaussien à une Image en Utilisant le GPU

4.1 Objectif

L'objectif de cette partie du travail pratique est d'appliquer un flou gaussien à une image en utilisant le GPU pour démontrer l'accélération des opérations de traitement d'images parallèles.

4.2 Exigences

1. Un ordinateur équipé d'un GPU compatible avec CUDA (de préférence une carte graphique NVIDIA).
2. Connaissance de base en traitement d'images.
3. Accès à l'environnement de développement CUDA (installation préalable, y compris les pilotes et la trousse d'outils CUDA).

4.3 Description de la Tâche

1. Préparation de l'Environnement:

- (a) Assurez-vous d'avoir installé et configuré CUDA sur votre ordinateur avec un GPU NVIDIA.
- (b) Téléchargez une image de votre choix pour l'expérience.

2. Écriture du Programme de Flou Gaussien:

- (a) Écrivez un programme en utilisant CUDA C/C++ qui applique un filtre de flou gaussien à l'image. Vous pouvez utiliser la convolution avec un noyau de flou gaussien pour ce faire.
- (b) Assurez-vous de paralléliser le calcul du filtre de flou gaussien en utilisant les threads CUDA pour traiter simultanément différents pixels de l'image.

3. Chargement et Traitement de l'Image:

- (a) Chargez l'image sélectionnée dans votre programme.
- (b) Appliquez le filtre de flou gaussien à l'image en utilisant le GPU. Mesurez le temps nécessaire pour cette opération.

4. Sauvegarde de l'Image Floutée:

- (a) Enregistrez l'image résultante après l'application du flou gaussien sur le disque.

5. Analyse des Performances:

- (a) Comparez le temps nécessaire pour appliquer le flou gaussien en utilisant le GPU avec celui qui serait requis en utilisant uniquement le CPU.
- (b) Calculez l'accélération obtenue grâce à l'utilisation du GPU.

6. Visualisation et Discussion:

- (a) Visualisez l'image floutée.
- (b) Discutez des résultats obtenus et de l'efficacité de l'utilisation du GPU pour accélérer les opérations de traitement d'images.

4.4 Livrables

1. Le code source du programme de flou gaussien utilisant CUDA.
2. L'image originale et l'image floutée.
3. Un rapport documentant l'expérience, les résultats, et l'analyse des performances.

4.5 Résultats d'Apprentissage

À la fin de cette partie du travail pratique, vous devriez être en mesure de :

1. Comprendre comment utiliser le GPU pour accélérer les opérations de traitement d'images parallèles.
2. Écrire un programme CUDA pour appliquer un filtre de flou gaussien à une image.
3. Mesurer et analyser les performances du GPU par rapport au CPU dans le contexte du traitement d'images.

Cette expérience vous permettra de mettre en évidence l'accélération significative que le GPU peut apporter lors du traitement d'images, ce qui est essentiel dans de nombreuses applications, y compris la vision par ordinateur et la photographie numérique.