UNIVERSITÉ CÔTE D'AZUR

POLYTECH NICE-SOPHIA

PER2024-004 - Type : recherche

Étude des calculs GP-GPU sur cartes graphiques

Etudiant(s) : Julien Soto (FISE IOT-CPS), Quentin Maurois (FISE IOT-CPS)

Encadrant(s) : Sid Touati, professeur, I3S.

1. Résumé exécutif

Ce projet de recherche vise à étudier en profondeur le calcul haute performance sur cartes graphiques. L'objectif principal est de comparer les performances des processeurs graphiques (GPU) avec celles des processeurs centraux (CPU) pour des tâches de calcul intensif.

La première étape consiste à réaliser une synthèse bibliographique approfondie sur les cartes graphiques, ainsi qu'une étude comparative des deux principales librairies de programmation pour GPU : CUDA et OpenCL. Cette étude théorique permettra de mieux comprendre les fondements et les spécificités de ces technologies.

Dans un second temps, des benchmarks seront mis en place afin d'évaluer concrètement les gains de performance obtenus en utilisant un GPU par rapport à un CPU. Pour cela, des algorithmes, principalement des calculs matriciels, seront implémentés en C ou C++. Ces programmes seront optimisés pour tirer parti du parallélisme offert par les processeurs multi-cœurs grâce à la directive OpenMP. Les résultats des benchmarks permettront de quantifier les accélérations obtenues et d'identifier les types de calculs les plus adaptés à l'utilisation d'un GPU.

2. Description du projet

Contexte technologique

- Les cartes graphiques, initialement développées pour gérer les représentations graphiques 3D en temps réel, notamment dans les jeux vidéo et les logiciels de conception assistée par ordinateur, ont progressivement été adaptées pour des usages de calcul haute performance. Leur architecture, optimisée pour traiter des milliers de calculs en parallèle, a révélé un potentiel important pour des domaines comme le calcul scientifique et les applications de deep learning.
- Le calcul haute performance (HPC) repose traditionnellement sur des processeurs centraux (CPU), mais les avancées dans les processeurs graphiques (GPU) offrent de nouvelles possibilités de parallélisme et de puissance brute. Le GPU se distingue par sa capacité à traiter un grand nombre de threads simultanément, ce qui est particulièrement avantageux pour les tâches parallélisables comme les calculs matriciels.
- Les cartes graphiques
- CUDA, développée par NVIDIA et optimisée pour ses propres cartes graphiques.
- OpenCL est une norme plus ouverte supportée par diverses architectures matérielles.

Motivations

- Réaliser une étude de faisabilité des calculs sur GPU pour différentes applications au travers de différents benchmarks et l'analyse de ces derniers. Les benchmarks seront séparés selon différentes tailles de matrices, différentes opérations effectuées sur ces matrices (multiplication, addition) et en utilisant les api CUDA et OpenCL pour comparer leurs performances respectives.
- Accélération des calculs scientifiques
- Réduction de la consommation électrique

Objectifs à atteindre

- L'objectif principal est d'identifier les types d'opérations les mieux optimisés pour le calcul sur GPU et de déterminer l'API offrant la mise en œuvre la plus efficace pour ces opérations.
- Avoir des représentations graphiques des performances des benchmarks effectués.

Risques identifiés

- Peu d'articles scientifiques retraçant l'historique et les études des performances des cartes graphiques
- Difficulté de tester toutes les cartes graphiques et tous les constructeurs (carte nvidia uniquement accessibles)
- Difficultés de créer un environnement stable et réplicable pour tester
- Problèmes de compatibilité entre les bibliothèques

Scénarios

Scénario 1 : Recherche bibliographique et traitement des données

Dans ce scénario, l'objectif est de rassembler et de structurer des informations pertinentes concernant l'évolution des cartes graphiques (GPU) et leurs architectures à des fins de recherche ou d'analyse. Voici comment nous procéderons :

Étape 1 : Recherche sur le navigateur avec des mots-clés spécifiques

Nous effectuons une recherche sur un moteur de recherche en utilisant des mots-clés ciblés comme « historique des cartes graphiques », « évolution des GPU », etc.

Critère d'acceptation : Affichage d'une liste de résultats de recherche pertinents qui redirigent l'utilisateur vers des articles, blogs scientifiques ou sites spécialisés.

Étape 2 : Consultation d'articles scientifiques et universitaires

Nous sélectionnons plusieurs articles pertinents et procédons à leur lecture pour extraire des informations clés.

Critère d'acceptation : Obtenir des données factuelles et validées par des sources académiques.

Étape 3 : Extraction, vérification et structuration des informations

Après avoir rassemblé les informations, nous analysons et vérifions la véracité des données collectées en recoupant plusieurs sources. Cette étape inclut la consultation des références utilisées par les articles pour garantir la fiabilité des informations.

Critère d'acceptation : Classer et récupérer les données en fonction de leur pertinence et de leur source.

Étape 4 : Transposition des données sous forme de tableaux et visualisations

Nous transcrivons les données collectées dans des paragraphes, des tableurs ou sous forme de schémas et graphiques pour faciliter leur interprétation et leur analyse.

Critère d'acceptation : Création de tableaux et de visualisations graphiques à partir des données récupérées.

Scénario 2 : Mise en place d'un environnement de développement

Nous allons mettre en place un environnement de développement réplicable utilisant la virtualisation et le passthrough de matériel. Nous nous baserons sur un système d'exploitation stable tel que Debian ou Fedora.

Étape 1 : Utiliser le passthrough de GPU pour rendre la carte graphique accessible à la machine virtuelle.

Critères d'acceptation : la carte graphique est reconnue et accessible par la machine virtuelle.

Étape 2 : installer les différents environnements :

- CUDA
  - Drivers Nvidia (closed source)
  - cuda toolkit
  - gcc (version compatible au niveau du driver et du toolkit)
- OpenCl
  - Driver pour la carte graphique utilisée
  - gcc ou g++
  - Télécharger les bibliothèques OpenCl

Critères d'acceptation : l'environnement permet de compiler et exécuter les programmes de test tels que hello world sur la carte graphique.

Étape 3 : Sauvegarder la machine virtuelle pour pouvoir la répliquer.

Critères d'acceptation : une copie de la machine virtuelle est sauvegardée sur le cloud.

Scénario 3 : Développement des benchmarks

Dans ce scénario, l'objectif est de concevoir, exécuter et analyser des benchmarks sur différents environnements afin d'évaluer et de comparer les performances des unités de traitement graphique (GPU) et central (CPU).

Étape 1 : Définir les critères d'évaluation et de mesure

Nous commençons par établir les métriques qui seront utilisées pour évaluer les performances. Cela inclut la définition des types d'opérations à évaluer (addition, multiplication matriciels, etc.), des mesures comme le temps d'exécution, la consommation énergétique, ou encore l'utilisation des ressources (CPU/GPU).

Critère d'acceptation Liste de critères clairement définis.

Étape 2 : Création des benchmarks (types d'opérations, types de valeurs et d'ordonnancement)

Nous concevons les tests de performance en déterminant les types d'opérations à exécuter (addition, multiplication,...) ainsi que les types de données à utiliser (ordonnées, aléatoires, entiers, flottants, etc.).

Critère d'acceptation : Réalisation d'au moins 10 types de benchmarks.

Étape 3 : Exécution et comparaison des résultats selon différents environnements

Les benchmarks sont exécutés dans des environnements variés (différents matériels, configurations logicielles). Les résultats, comme les temps d'exécution, sont collectés et analysés pour comparer les performances des GPU et CPU selon les critères définis.

Critère d'acceptation : Le système exécute les benchmarks sans erreurs, quel que soit l'environnement. Les résultats sont collectés, organisés et exportables sous forme de tableaux et de graphiques.

3. Mise en en œuvre

Liste d'activités déjà réalisés avant les semaines à plein temps

- Recherches sur OpenCL et Cuda
- Etude de l'historique des cartes graphiques
- Etude des différentes architectures des cartes graphiques
- Mise en place d'un environnement de test
- Mise en place de l'état de l'art

Listes d'activités prévues pour chaque semaine à plein temps

- Effectuer différents types de benchmarks
- Adaptation et optimisation des algorithmes pour l'exécution sur GPU avec CUDA et OpenCL.
- Exécution des benchmarks sur différents types de calculs (matriciels, prédictifs), collecte et analyse des données de performances.
- Compilation des résultats, finalisation du rapport comparatif, et préparation d'une présentation des résultats obtenus.

Organisation du travail

Julien Soto : Recherches et analyse de résultats

Quentin Maurois : Implémentation des benchmarks CUDA et Opencl

Sid Touati (Encadrant) : Supervision du projet, apport de conseils techniques et validation des choix méthodologiques.