

C++, programmer avec CUDA sous Visual Studio utiliser le GPU pour améliorer les performances

Cours Pratique de 2 jours - 14h

Réf : CDU - Prix 2024 : 1 450CHF HT

Cette formation vous propose de découvrir, d'évaluer et de manipuler le SDK CUDA de la société NVIDIA, leader en matière d'utilisation du GPU, pour améliorer les performances du parallélisme de données. Vous acquerez toutes les connaissances nécessaires à la mise en œuvre de CUDA.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Comprendre l'intérêt de l'utilisation du GPU en tant que ressource de calcul indépendante

Utiliser le GPU avec CUDA dans l'environnement de Visual Studio

Assurer le lien entre les threads du C++11 et l'utilisation du GPU

Vérifier l'intérêt dans le cadre d'un projet complet

MÉTHODES PÉDAGOGIQUES

Exposés théoriques suivis de relecture de code puis de mise en œuvre sur une application de test.

TRAVAUX PRATIQUES

Développement d'une application de test, évaluation des différentes solutions proposées et comparaison avec des traitements équivalents sur le CPU.

LE PROGRAMME

dernière mise à jour : 01/2018

1) Introduction

- Présentation de l'utilisation du GPU comparée au CPU.
- Le SDK CUDA de la société NVIDIA.
- Les alternatives et compléments à CUDA.

Démonstration : Présentation de l'application de test et évaluation des résultats sur le CPU.

2) Installation de CUDA

- Installation du driver spécifique et du SDK de CUDA.
- Installation de NSIGHT, l'environnement spécifique de CUDA sous Visual Studio.
- Exploration des exemples d'application.
- Récupération des capacités des cartes graphiques installées.

Travaux pratiques : Installation de CUDA, création d'un projet et validation de l'installation.

3) La mise en œuvre de base

- Les fondamentaux de l'exécution d'une fonction kernel.
- La création d'une fonction kernel.
- L'appel d'une fonction kernel.
- Les transferts de mémoire entre le host et le GPU.
- L'exécution asynchrone d'une séquence de code GPU.
- Le débogage du code exécuté sur le GPU.

Travaux pratiques : Ajout d'une séquence de code à exécuter sur le GPU à l'application de test, comparaison des résultats avec l'existant en C++11. Utilisation du débogueur de NSIGHT.

PARTICIPANTS

Concepteurs et développeurs d'applications en C/C++, architectes logiciels.

PRÉREQUIS

Bonnes connaissances du langage C/C++ et des threads, expérience requise. Connaissances de base de C++11.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

4) L'utilisation des différentes options de mémoire de CUDA

- La mémoire partagée à l'intérieur d'un bloc de threads, les différentes options.
- L'optimisation entre la mémoire consacrée aux données et la taille du code à exécuter.
- Les allocations mappées entre la mémoire du host et la mémoire de la carte graphique.
- L'utilisation de la mémoire portable entre le host et plusieurs cartes graphiques.

Travaux pratiques : Manipulation des différentes options dans l'application de test.

Recherche de la meilleure solution selon un cas étudié.

5) Les autres utilisations de CUDA

- L'utilisation des Streams, exécution en parallèle sur différentes cartes graphiques.
- L'utilisation de CUDA en C++ avec Thrust.
- Les alternatives ou compléments à CUDA comme le C++ AMP, OpenCL, OpenAPP.

Etude de cas : Exploration des solutions complémentaires et alternatives, comparaison à l'aide de l'application de test.

6) Conclusion

- Le champ d'application de l'utilisation du GPU comme alternative au CPU.
- Les bonnes pratiques.

LES DATES

CLASSE À DISTANCE

2024 : 01 juil., 16 sept., 19 déc.