

Stage pratique de 2 jour(s)  
Réf : CDU

## Participants

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

## Pré-requis

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

## Dates des sessions

### CLASSE A DISTANCE

04 mar. 2021

PARIS

04 mar. 2021

## Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers des multiples exercices à réaliser (50 à 70% du temps).

## 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.

## 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.

• A 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.

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

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 PEDAGOGIQUES

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

### 1) Introduction

### 2) Installation de CUDA

### 3) La mise en œuvre de base

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

### 5) Les autres utilisations de CUDA

### 6) Conclusion

## 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.

## 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.

## 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.

• 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.

- 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.