

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

-Référence: **SII-298**

-Durée: **2 Jours (14 Heures)**

## Les objectifs de la formation

- À la fin de la formation, le participant 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

## A qui s'adresse cette formation ?

### POUR QUI :

- 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 du C++11.

## Programme

- **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.
- **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.

- **La mise en oeuvre 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.

- **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.

- **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.

- **Conclusion**

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



(+212) 5 22 27 99 01



(+212) 6 60 10 42 56



Contact@skills-group.com

Nous sommes à votre disposition :  
De Lun - Ven 09h00-18h00 et Sam 09H00 – 13H00

Angle bd Abdelmoumen et rue Soumaya, Résidence Shehrazade 3, 7ème étage N° 30  
Casablanca 20340, Maroc