C++, program with CUDA under Visual Studio use the GPU to improve performance

★★★★★

SII-298 **2 Days (14 Hours)**

Description

This training will allow you to learn how to use NVIDIA's CUDA SDK, a leader in GPU usage, to improve data parallelism performance. You will learn the knowledge necessary to implement CUDA, which will allow you to evaluate and manipulate this technology.

Who is this training for ?

For whom

Application designers and developers in C/C++, software architects.

Prerequisites

Good knowledge of C/C++ language and threads, experience required. Basic knowledge of C++11.

Training objectives

- At the end of the training, the participant will be able to: Understand the benefit of using the GPU as an independent computing resource
- Use the GPU with CUDA in the Visual Studio environment
- Ensure the link between C++11 threads and the use of the GPU
- Check the interest in the context of a complete project

Training program

Introduction

- Presentation of GPU usage compared to CPU.
- The CUDA SDK from NVIDIA.
- Alternatives and complements to CUDA.
- Demo Presentation of the test application and evaluation of the results on the CPU.

Installation de CUDA

- Installation of the specific CUDA driver and SDK.
- Installation of NSIGHT, the specific CUDA environment under Visual Studio.
- Exploration of application examples .
- Recovering the capacities of installed graphics cards.
- Practical work Installation of CUDA, creation of a project and validation of the installation.

La mise en oeuvre de base

- The fundamentals of executing a kernel function.
- Creating a kernel function.
- Calling a kernel function.
- Memory transfers between the host and the GPU.
- Asynchronous execution of a sequence of GPU code.
- Debugging the code executed on the GPU.
- Practical work Addition of a sequence of code to be executed on the GPU to the test application, comparison of the results with the existing one in C++11.
- Using the NSIGHT debugger.

L'utilisation des différentes options de mémoire de CUDA

- Shared memory within a block of threads, the different options.
- Optimization between the memory devoted to data and the size of the code to execute.
- The allocations mapped between host memory and graphics card memory.
- Portable memory usage between host and multiple graphics cards.
- Practical work Manipulation of the different options in the test application.
- Search for the best solution according to a case studied.

Les autres utilisations de CUDA

- The use of Streams, parallel execution on different graphics cards.
- The use of CUDA in C++ with Thrust.
- Alternatives or complements to CUDA such as C++ AMP, OpenCL, OpenAPP.
- Case study Exploration of complementary and alternative solutions, comparison using the test application.

Conclusion

- The scope of using GPU as an alternative to CPU.
- Best practices.