



Deep Learning using Cuda



Introduction

In the recent years the overall processing power is increasing, thanks to the new model of chip manufacturing. This model increases the overall processing power by adding additional processing cores to the microprocessor package. The processors will gradually come in heterogeneous configurations such as combination of high and low power cores, GPU's etc. These are being termed as many core architecture. Currently, only a very small proportion of developers have expertise in parallel programming. For computer science faculty this also requires a radical shift in the way computer science subjects are taught. Even though high performance computing area has programmers working in parallel applications for years, very few people have got the understanding of issues involved in developing multicore applications. What is very important is to begin the transition to "thinking in parallel" immediately, whether or not the mechanisms for doing so are ideal. This workshop will discuss why parallel programming landscape is needed, summarize the OpenMP approach to multi-threading, and illustrate how it can be used to introduce parallelism.

The Graphics Processing Unit or GPU is nowadays a mainstream component in Scientific Computing and Data Analytics. For relatively little money one can have supercomputer performance. However, some extra work has to be done to make an ordinary sequential program suitable for use on the GPU. One of the most important tools for using GPUs is currently "CUDA" (Compute Unified Device Architecture). This is basically an extension to the C programming language, which can be used to program the GPU in an easy way.

Duration: 2 day

Course Objective:

This workshop will discuss why the parallel programming landscape is needed, summarize the OpenMP approach to multi-threading, and illustrate how it can be used to introduce parallelism. Similar mechanisms using GPU's will also be discussed for Scientific Computing and Data Analytics.

Course Structure:

Session 1: Introduction and Basics

1. Why Parallel Computing? Need for parallel Computing
2. Multi-Core processors - Architecture and Design
3. Introduction to Threads, Thread Basics and Basic concepts of parallel programming
4. OpenMP- A Standard for Directive based Parallel Programming
5. Hands on / Demonstration of various programs on multicore machines

Session 2: GPU Programming

1. GPU's for parallel Computing
2. GPU's Programming Model
3. Hands on / Demonstration of various programs on GPU
4. SDK, Toolkit and Installation of environment for GPU
5. Working with various Libraries





6. Demonstration of GPU and Tools with Sample Programs and OpenACC.

Session 3: Introduction to Deep Learning

1. Intro to Deep Learning
2. Intro to Deep learning using various S/w's on GPU
3. Demos using DIGITS

Pre-requisite:

- A basic understanding of C programming Language or Java or Python.
- Interest in iterative solvers and similar workloads.
- Participants are required to bring their own laptops with Wi-Fi connectivity and chargers.