Applied Parallel Computing LLC
http://parallel-computing.pro
+41 78 925 90 90
contact@parallel-computing.pro

2-day CUDA training course at Universität Leipzig

(6-7 December 2017)

Exploiting the potential of GPU computing is inevitable for any modern HPC application. As the leader in the application of compute & deep learning technologies, NVIDIA sets up the quality standard for massively parallel hardware, development tools and GPU-accelerated libraries. This course provides essential practical experience for scientists to develop, debug and optimize fast and efficient research codes with NVIDIA CUDA. As a case study, the course presents a practical session on using CUDA for Deep Learning convolutional neural networks.

Hands-ons: All discussed topics will be accompanied with practical sessions, using NVIDIA CUDA 9 compiler. Exercises will be conducted either on the provided remote GPU server or on the customer's local system.

All corresponding presentations will be available to attendees in printed handouts.

Applied Parallel Computing LLC is delivering GPU training courses since 2009. Several dozens of courses have been organized all over Europe, both for commercial and academic customers. We work in close partnership with NVIDIA, CUDA Centers of Excellence and Tesla Preferred Partners. In addition to trainings, our company provides GPU porting/optimization services and CUDA certification.

Day 1: Introduction to CUDA

Morning (10:00-13:30)

10:00-11:15: lecture

- An overview of GPU performance in various applications
- Brief intercomparison of different types of accelerators
- Key programming principles to achieve high GPU performance

11:15-11:30: coffee break

11:30-12:30: lecture

- CUDA principles and CUDA implementation for C++
- Analogies between MPI+OpenMP and CUDA programming models
- The first CUDA program explained
- CUDA compute grid, examples
- Realistic CUDA application example (wave propagation code)
- Understanding GPU compute capabilities, *deviceQuery*
- Basic optimization techniques
- Overview of CUDA applications development

12:30-13:30: hands-on session

- Example of *vector addition* in CUDA, compared to OpenACC implementation
- **Hands-on:** Write & deploy a simple CUDA program
- **Hands-on:** More control on CUDA compute grid

Afternoon (14:30-18:00)

14:30-16:00: lecture

- Thrust – the C++ library of GPU-enabled parallel algorithms
- CUBLAS, MAGMA, CUBLAS-XT, CUSPARSE, CUFFT and CURAND
- CUSP and AmgX – Krylov and mutigrid solvers
- CUDNN – Deep Neural Network library

16:00-16:20: coffee break

16:20-18:00: hands-on session

- **Hands-on:** solving Poisson equation with CUFFT

Day 2: Advanced CUDA programming

Morning (09:00-12:30)

09:00-10:30: lecture

- GPU memory types
- Shared memory
- GPU caches hierarchy and mode switches
- Automatic texture cache (Kepler GK110)
- Unified virtual address space (UVA) in CUDA 7.5
- Streams and asynchronous data transfers
- Warp shuffle instruction

10:30-11:00: coffee break

11:00-12:30: hands-on session

- **Hands-on:** "fill-in" exercise on reduction with and without shared memory
- **Hands-on:** getting additional performance using warp shuffle instruction

Afternoon (13:30-17:00)

13:30-15:00: lecture

- The cost of global memory allocation
- PCI-E optimizations: streams, asynchronous data transfers
- An overview of Kepler, Maxwell, Pascal and Volta GPU architectures
- GPU optimizations: compute grid, coalescing, divergence, unrolling, vectorization, maxrregcount, aligning, floating-point constants
- CUDA C++ compiler pipeline, PTX assembler, SASS
- Understanding "-Xptxas -v" reports
- Overview of *NVIDIA Visual Profiler*
- Overview of *nvprof* (command line profiler)
- Common practices of identifying performance hazards in GPU application using NVIDIA Visual Profiler

15:00-15:20: coffee break

15:20-16:30: hands-on session

- Porting Batched Gradient Descent method for Convolutional Neural Network Deep Learning onto GPU using CUDA.

16:30-17:00: Q & A

Prerequisites

- Beamer
- Open (unfirewalled) connection to farm.parallel-computing.pro (188.44.42.33) over port 22
- PuTTY SSH client