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

2-day Course on GPU Computing at Rohde & Schwarz GmbH

(13–14 October 2016)

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.

All corresponding presentations and code samples will be available to attendees from the beginning of each training day.

Day 1: Introduction to GPU Computing, CUDA and GPU-enabled libraries

Morning (09:00-12:30)

09:00-10:30: lecture

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

10:30-12:30: 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 using Visual Studio 2015

Afternoon (13:30-17:00)

13:30-15:00: lecture

- Rapid GPU development with Thrust
- cuBLAS, MAGMA, cuBLAS-XT and NVBLAS for dense linear algebra operations
- cuSPARSE library for sparse linear algebra
- cuSOLVER framework for dense and sparse linear solvers

15:00-17:00: lecture

- Fast Fourier transforms on GPU using cuFFT
- cuRAND: random and pseudo-random numbers generators on GPU
- nvGRAPH library for high-performance graph traversal
- NPP: NVIDIA Performance Primitives library

Day 2: GPU memory hierarchy, CUDA architecture, GPU optimization and debugging

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); case studies
- Unified virtual address space (UVA); case studies

10:30-11:30: lecture

- An overview of Kepler, Maxwell and Pascal GPU architectures
- CUDA C++ compiler pipeline, PTX assembler, SASS
- Understanding "-Xptxas -v" reports
- Architecture-driven GPU optimizations: compute grid, coalescing, divergence, unrolling, vectorization, maxrregcount, aligning, floating-point constants
- PCI-E optimizations: streams, asynchronous data transfers

11:30-12:30: lecture

- Dynamic parallelism
- Dynamic memory allocation in CUDA threads
- Compiling & linking relocatable device code
- NVRTC runtime compilation library for CUDA C++

Afternoon (13:30-17:00)

13:30-15:30: lecture

- Principles and terminology of debugging
- CUDA-enabled GNU Debugger (cuda-gdb)
- GPU memory checker (cuda-memcheck)
- Debugging SASS without the source code
- Overview and examples of NVIDIA Nsight Visual Studio debugging
- Examples of Visual Studio Nsight Debugging

15:30-17:00: lecture

- Terminology of CUDA and OpenCL
- NVIDIA and AMD GPUs: architectural differences
- Execution workflow: offline compilation in CUDA, runtime compilation in CUDA (8.0) and in OpenCL