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

GPU Supercomputing training at TÜBİTAK UZAY

25 August – 29 August 2014

Exploiting the potential of GPU computing is inevitable for any modern HPC application. Simulations for cutting-edge space research simulations have to scale up to hundreds and thousands of compute cluster nodes. Thus, the ability of researchers to efficiently combine general cluster technologies (e.g. MPI) and GPU computing nowadays becomes extremely valuable. This course will guide attendees into GPU computing & optimization, and into programming of hybrid CPU+multiGPU applications with workload balancing.

Hands-ons: All discussed topics will be accompanied with practical sessions, including image/video processing and machine learning. Exercises will be conducted either on attendee's preferred systems with NVIDIA GPU and CUDA available, or on the provided GPU server with K20 and C2050 GPUs: https://client.parallel-computing.pro/.

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 development environments

Morning:

- An overview of GPU performance in various applications
- Comparing performance of Telsa GPUs, AMD Radeon and Intel Xeon Phi at glance
- Key differences between CPU and GPU architectures
- Key programming principles to achieve high GPU performance
- 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)
- Basic optimization techniques

Afternoon:

- Overview of Eclipse Nsight Edition IDE
- Setting up remote terminal and file system synchronization for CUDA programming in cluster environment
- Understanding GPU compute capabilities, *deviceQuery*
- Checking GPU status with *nvidia-smi* and NVML library
- **Hands-on:** "fill-in" exercise on re-writing *saxpy* function

Day 2: CUDA for images and signal processing, GPU metaprogramming

Morning:

- The summary of typical steps for porting applications to GPUs with CUDA

- **Hands-on:** "fill-in" exercise on fully GPU-resident application and re-writing *Corner detection* function of *Harris algorithm* in CUDA

Afternoon:

- Rapid data processing with Thrust
  - Transforms and functors
  - Placeholders and tuples
  - Performance considerations, optimization
  - Interop between Thrust and CUDA C
  - **Hands-on:** "fill-in" exercise on image intensity histogram with Thrust (*histogram*)
  - **Hands-on:** "fill-in" exercise on implementing custom Thrust transform (*saxpy*)
- GPU libraries
  - CUBLAS, MAGMA, CUBLAS-XT, CUSPARSE, CUFFT, CURAND
  - NPP, OpenCV
  - **Hands-on:** filtering image with OpenCV (GPU-enabled version) (*salt_and_pepper*)
  - **Hands-on:** filtering sound signal with CUFFT

Day 3: Optimization and profiling

Morning:

- PCI-E optimizations: streams, asynchronous data transfers
- An overview of Fermi, Kepler and Maxwell GPU architectures

Afternoon:

- GPU optimizations: coalescing, divergence, unrolling, vectorization, maxrregcount, aligning, kernel specialization (unbranching)
- Overview of *nvprof*: GUI version
- Overview of *nvprof*: command line interface
- Common practices of identifying performance hazards in GPU application via *nvprof*
- **Hands-on:** profile and optimize the given sample GPU program

Day 4: MultiGPU

Morning:

- GPU context
- Accessing multiple GPUs from a single process/thread
- Accessing single GPU from multiple processes using MPS (Hyper-Q)
- CUDA peer-to-peer memory transfer functions, GPUDirect
- CUDA-aware MPI

Afternoon:

- **Hands-on:** using multiple GPUs in parallel from serial application
- **Hands-on:** using single GPU from MPI application with and without MPS
- **Hands-on:** using MPI_Send/MPI_Recv with GPU memory buffers in CUDA-aware OpenMPI/MVAPICH

Day 5: GPU-enabled supercomputing

Morning:

- Planning GPU port of MPI/OpenMP application: profiling with Vampir and Intel VTune
- Getting out the best from CPU and GPU, by example of MPI application with transform-reduce kernel, CPU-GPU load balancing with Threading Building Blocks
- Handling halos in MPI+GPU stencil applications
- The role of P2P and GPUDirect, checking availability of GPUDirect

Afternoon:

- **Hands-on:** MPI+GPU *Genetic Algorithm*

About us

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.