



Syllabus: GPU Computing

Day 1:

- Lecture: Overview GPU Computing
 - Hardware features, system architecture, general concepts
- Exercise: First GPU program
 - Software environment, host program, data transfer
- Lecture: CUDA Programming Model
 - Kernel, kernel grid, data parallelism
- Exercise: Simple Kernels (vector op and scalar product)
 - Kernel invocation, global memory access, thread cooperation
 - Basic debugging strategies
- Lecture: CUDA Memory Hierarchy
 - Global, shared, local memory, common access patterns
- Exercise: Software managed cache (2D convolution)
 - Using shared memory

Day 2:

- Lecture: Parallel Algorithms
 - Data parallel algorithms, distributed memory parallelism
 - Estimating the performance
- Lecture: Memory optimization I - Memory coalescence
- Exercise: Reduction operations
 - Aligned data types
 - Parallel scan
- Lecture: Memory optimization II - Memory access patterns
 - Textures, atomics
- Exercise: Bilinear interpolation
 - Textures, hardware interpolation
 - Texture convolution
- Lecture: Libraries
 - cuBLAS, cuFFT, other libraries
- Exercise: Basic signal processing
 - FFT-based convolution in 1D
 - cuBLAS DGEMM

Tech-X Corporation

Day 3:

Lecture: Optimization II - system-wide optimizations

Arithmetic optimization

Asynchronous data transfer

Zero copy transfers

Exercise: Latency hiding

GPU processing of large data sets

Lecture: Day-to-day work with GPUs

Optimization strategies

Estimating theoretical performance

Tools: Profiler, Debugger

Case study: Image processing

Simple box filter

Image denoising

Future Trends and Closing Remarks