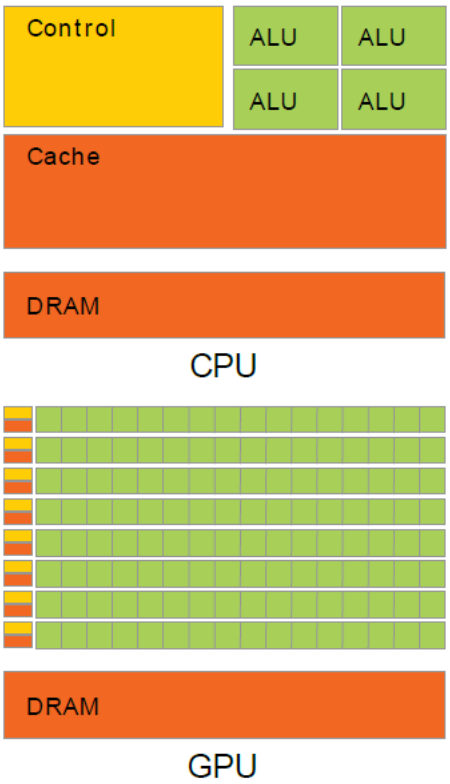
GPGPU Programming with CUDA

Duration - 4 Days

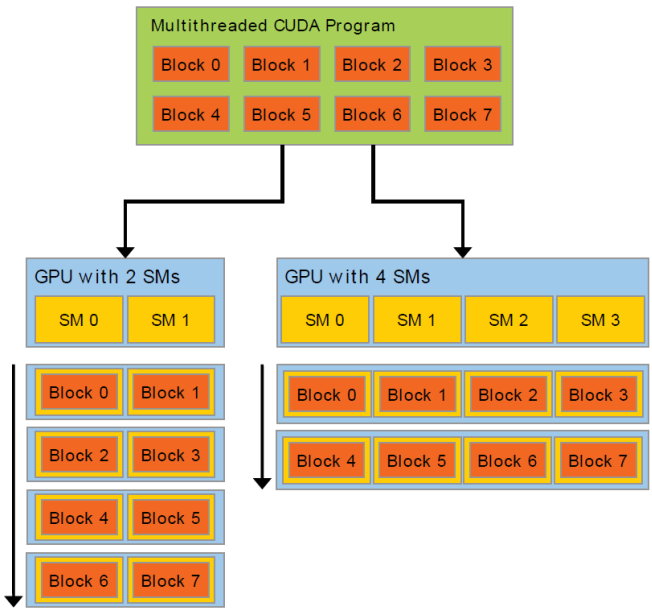
DAY 1

1. **Introduction to GPU Programming and GPU Architectures**
   - 1.1. Motivation
   - 1.2. High-level hardware specifications
   - 1.3. Brief history of GPGPU
   - 1.4. Overview of CUDA
   - 1.5. Introduction to CUDA syntax
   - 1.6. Hands-on Exercise - GPU
   - 1.7. Memory Management



2. **Data-Parallel Architectures and the GPU Programming Model**
   - 2.1. Data-parallelism
   - 2.2. GPU programming model
   - 2.3. GPU kernels
   - 2.4. Host vs. Device
   - 2.5. Memory management
   - 2.6. CUDA syntax
   - 2.7. Thread hierarchy
   - 2.8. Unified Memory
   - 2.9. Hands-on Exercise - Simple CUDA Kernels

<LUNCH BREAK>



3. **GPU Memory Model & Thread Cooperation**

   3.1. Task parallelism

   3.2. Thread cooperation in GPU computing

   3.3. GPU memory model

   3.4. Shared memory

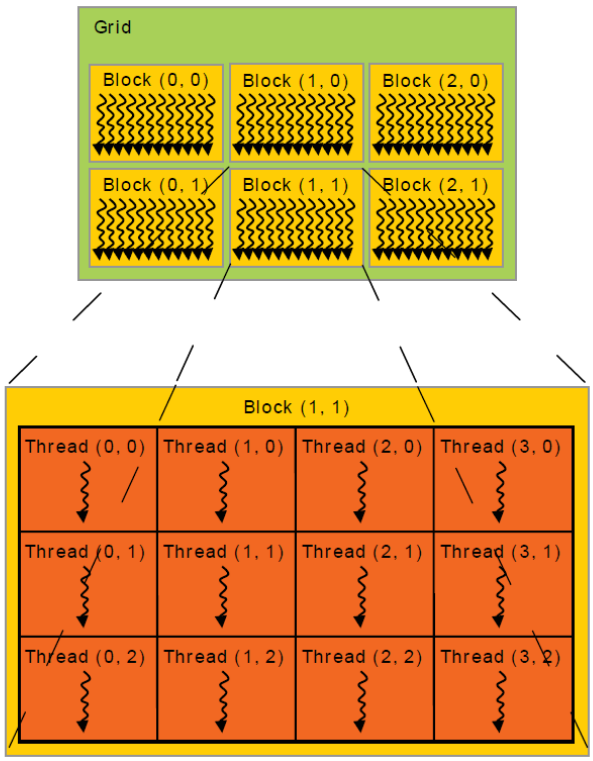
   3.5. Constant memory

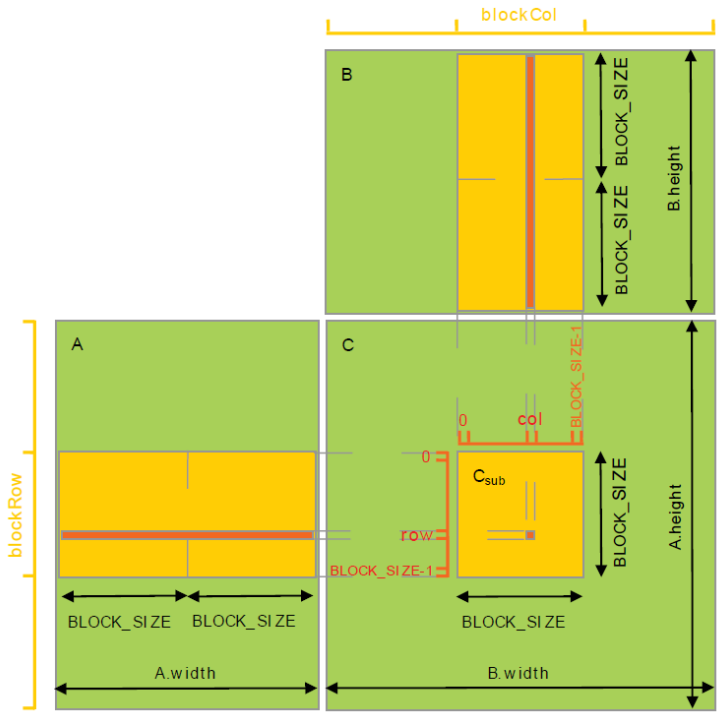
   3.6. Global memory

   3.7. Example - Matrix Multiplication without Shared Memory

   3.8. Example - Matrix Multiplication with Shared Memory

   3.9. Hands-on Exercise - Shared Memory and Constant Memory





- **Recap and Questions**

DAY 2

4. Asynchronous Operations & Dynamic Parallelism

4.1. Asynchronous vs. synchronous memory transfers

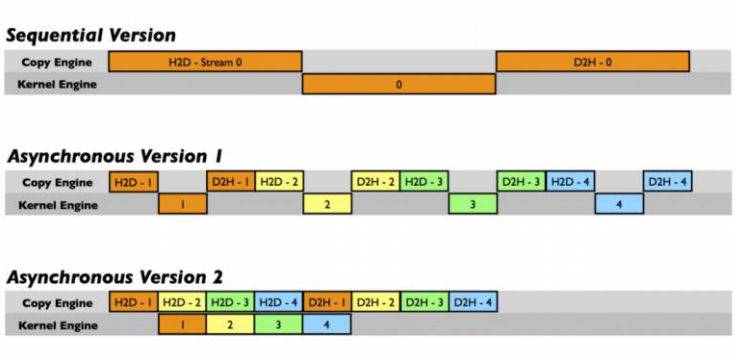
4.2. Streams and events

4.3. Page locked memory

4.4. Streams and Unified Memory

4.5. Dynamic Parallelism

4.6. Hands-on Exercise - Asynchronous Operations



5. Advanced CUDA Features

5.1. NVCC

5.2. Atomic functions

5.3. Dynamic memory allocation within kernels

5.4. Multi-GPU Programming

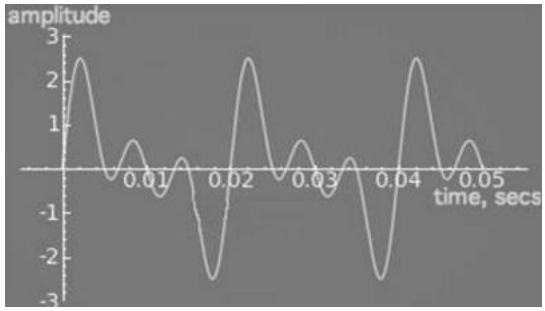
5.5. Peer-to-peer memory access

5.6. Example – Simple Linear Search using atomic operations

5.7. Hands-on Exercise - Small exercises focused on various CUDA features

<LUNCH BREAK>

6. **Libraries**



   6.1. CUFFT

   6.2. CUBLAS

   6.3. Thrust

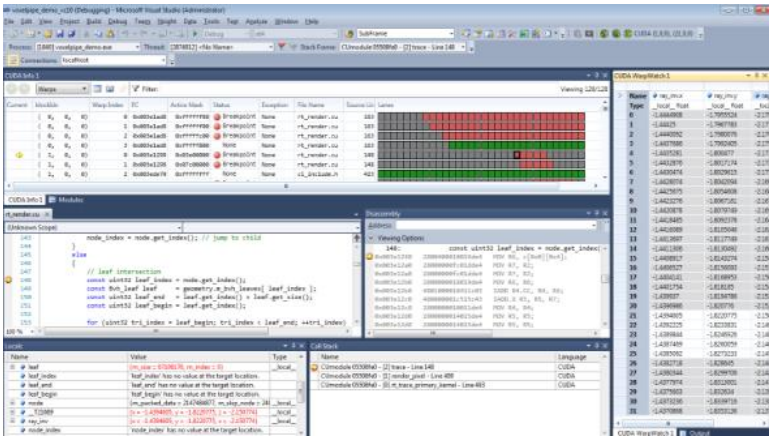
   6.4. CURAND

   6.5. NVIDIA performance primitives

   6.6. Hands-on Exercise - Experience with CUBLAS, CUFFT, Thrust and/or CURAND

- **Recap and Questions**

DAY 3



7. **Debugging GPU Programs & Numerical Accuracy**
   - 7.1. Debugging tools and techniques
   - 7.2. NVIDIA Nsight
   - 7.3. Numerical Accuracy in GPU Implementations
   - 7.4. Hands-on Exercise – Debugging
8. **Introduction to Optimizations & Profiling**
   - 8.1. High-Level Optimization Strategies
   - 8.2. Timers
   - 8.3. NVIDIA Visual Profiler
   - 8.4. Guided Performance Analysis
   - 8.5. Hands-on Exercise - Simple Profiling Functionality

<LUNCH BREAK>

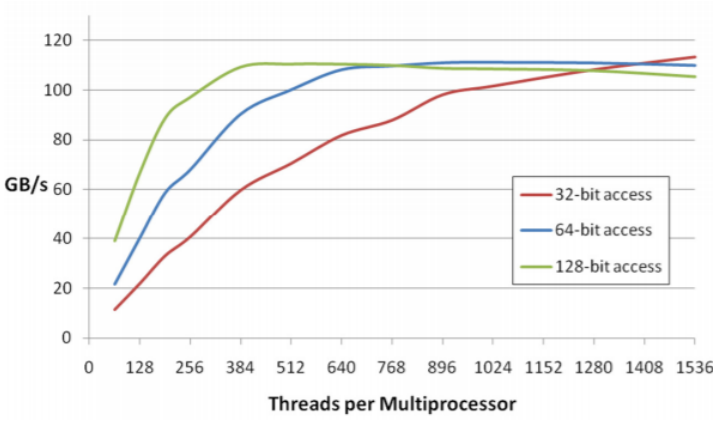
9. Resource Management, Latency, and Occupancy

9.1. GPU SM Execution

9.2. GPU Latencies and How They Impact Performance

9.3. Occupancy and Occupancy Related Optimizations

9.4. Hands-on Exercise - Occupancy Calculator and Occupancy Optimizations



10. Arithmetic Optimizations

10.1. Streaming Multiprocessor Details

10.2. Kepler

10.3. Maxwell

10.4. Instruction cost

10.5. Intrinsic functions

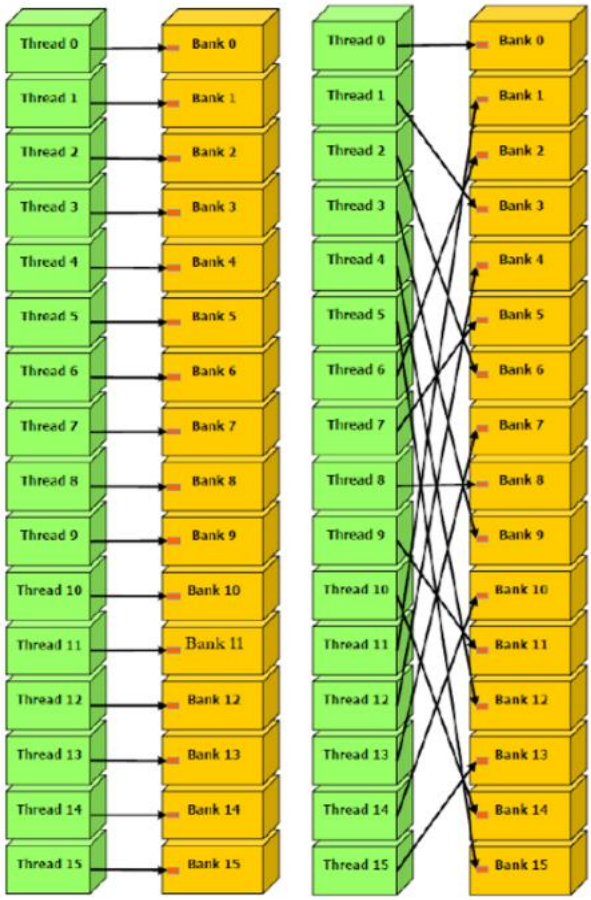
10.6. Branching efficiency

10.7. Instruction-Level Parallelism

10.8. Hands-on Exercise - Arithmetic Optimizations

- **Recap and Questions**

DAY 4



11. Memory Performance Optimizations

11.1. Review logical memory spaces

11.2. Physical implementation of memory and optimal access patterns

11.3. Global Memory Access Patterns

11.4. Shared Memory Bank conflicts

11.5. Constant Memory and Read-Only Cache

11.6. Textures and Caches

11.7. Memory usage strategies

11.8. Hands-on Exercise - Memory Optimizations

<LUNCH BREAK>

12. Graphics Interoperability

12.1. OpenGL Interop

12.2. Direct3D 11 Interop

12.3. Querying Devices

12.4. Registering and Mapping Resources

12.5. Textures

12.6. Example – Simple OpenGL and Direct3D graphics examples

13. Questions / Additional Topics

NOTE: Please install, build, and try running the provided example programs in advance. In case of issues contact support@kbvis.com