L15: Review for Midterm

Administrative

- Project proposals due today at 5PM (hard deadline)
  - handin cs6963 prop <propfile>
- March 31, MIDTERM in class

Outline

- Questions on proposals?
- More on strsm
- Review for Midterm
  - Describe planned exam
  - Go over syllabus

Midterm Exam March 31

- Goal is to reinforce understanding of CUDA and NVIDIA architecture
- Material will come from lecture notes and assignments
- In class, should not be difficult to finish
- Open notes, but no computers

Parts of Exam

I. Definitions
- A list of 10 terms you will be asked to define

II. Constraints and Architecture
- Understand constraints on numbers of threads, blocks, warps, size of storage
- Understand basic GPU architecture: processors and memory hierarchy

III. Problem Solving
- Analyze data dependences and data reuse in code and use this to guide CUDA parallelization and memory hierarchy mapping
- Given some CUDA code, indicate whether global memory accesses will be coalesced and whether there will be bank conflicts in shared memory
- Given some CUDA code, add synchronization to derive a correct implementation
- Given some CUDA code, provide an optimized version that will have fewer divergent branches
- Given some CUDA code, derive a partitioning into threads and blocks that does not exceed various hardware limits

IV. (Brief) Essay Question
- Pick one from a set of 4

How Much? How Many?

- How many threads per block? Max 512
- How many blocks per grid? Max 65535
- How many threads per warp? 32
- How many warps per multiprocessor? 24
- How much shared memory per streaming multiprocessor? 16Kbytes
- How many registers per streaming multiprocessor? 8192 (G80)/16384 (GTX/ Tesla)
- Size of constant cache: 8Kbytes

Syllabus

L1: Introduction and CUDA Overview
- Not much there…

L2: Hardware Execution Model
- Difference between a parallel programming model and a hardware execution model
- SIMD, MIMD, SIMT, SPMD
- Performance impact of fine-grain multithreaded architecture
- What happens during the execution of a warp?
- How are warps selected for execution (scoreboarding)?

L3: Writing Correct Programs
- Race condition, dependence
- What is a reduction computation and why is it a good match for a GPU?
- What does __syncthreads () do? (barrier synchronization)
- Atomic operations
- Memory Fence Instructions
- Device emulation mode

L4 & L5: Memory Hierarchy: Locality and Data Placement
- Memory latency and memory bandwidth optimizations
- Reuse and locality
- What are the different memory spaces on the device, who can read/write them?
- How do you tell the compiler that something belongs in a particular memory space?
- Tiling transformation (to fit data into constrained storage): Safety and profitability

Syllabus

L6 & L7: Memory Hierarchy III: Memory Bandwidth Optimization
- Leftover from L5 -- Tiling (for registers)
- Bandwidth – maximize utility of each memory cycle
- Memory accesses in scheduling (half-warp)
- Understanding global memory coalescing (for compute capability < 1.2 and > 1.2)
- Understanding shared memory bank conflicts

L8: Control Flow
- Divergent branches
- Execution model
- Warp vote functions

L9: Floating Point
- Single precision versus double precision
- IEEE Compliance: which operations are compliant?
- Intrinsics vs. arithmetic operations, what is more precise?
- What operations can be performed in 4 cycles, and what operations take longer?

L10 & L11: Dense Linear Algebra on GPUs
- What are the key ideas contributing to CUBLAS 2.0 performance
- Concepts: high thread count vs. coarse-grain threads. When to use each?
- Transpose in shared memory plus padding trick

L12: Sparse Linear Algebra on GPUS
- Different sparse matrix representations
- Stencil computations using sparse matrices

L13&L14: Application case studies
- Host tiling for constant cache (plus data structure reorganization)
- Replacing trig function intrinsic calls with hardware implementations
- Global synchronization for MPM/GIMP