L13: Review for Midterm

Administrative

• Project proposals due Friday at 5PM (hard deadline)
• No makeup class Friday!
• March 23, Guest Lecture
  • Austin Robison, NVIDIA
  • Topic: Interoperability between CUDA and Rendering on GPUs
• March 25, MIDTERM in class

Outline

• Questions on proposals?
  – Discussion of MPM/GIMP issues
• Review for Midterm
  – Describe planned exam
  – Go over syllabus
  – Review L4: execution model

Reminder: Content of Proposal, MPM/GIMP as Example

I. Team members: Name and a sentence on expertise for each member
   Obvious
II. Problem description
  - What is the computation and why is it important?
  - Abstraction of computation: equations, graphic or pseudo-code, no more than 1 page
   Straightforward adaptation from MPM presentation and/or code
III. Suitability for GPU acceleration
  - Amdahl’s Law: describe the inherent parallelism. Argue that it is close to 100% of computation. Use measurements from CPU execution of computation if possible
  Can measure sequential code
  Remove “history” function
  Phil will provide us with a scaled up computation that fits in 512MB
Reminder: Content of Proposal, MPM/GIMP as Example

III. Suitability for GPU acceleration, cont.
- Synchronization and Communication: Discuss what data structures may need to be protected by synchronization, or communication through host.
- Some challenges on boundaries between nodes in grid
- Copy Overhead: Discuss the data footprint and anticipated cost of copying to/from host memory.
- Some grid and patches to discover data footprint. Consider ways to combine computations to reduce copying overhead.

IV. Intellectual Challenges
- Generally, what makes this computation worthy of a project?
  - Importance of computation, and challenges in partitioning computation, dealing with scope, managing copying overhead
- Point to any difficulties you anticipate at present in achieving high speedup

Midterm Exam

- 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

Parts of Exam

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

II. Constraints
- Understand constraints on numbers of threads, blocks, warps, size of storage

III. Problem Solving
- Derive distance vectors for sequential code and use those to transform code to CUDA, making use of constant memory
- 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
- Size of constant cache: 8Kbytes
Syllabus

L1 & L2: Introduction and CUDA Overview
- Not much there...
L3: Synchronization and Data Partitioning
  - What does __syncthreads() do?
  - Indexing to map portions of a data structure to a particular thread
L4: Hardware and Execution Model
  - How are threads in a block scheduled? How are blocks mapped to streaming multiprocessors?
L5: Dependence Analysis and Parallelization
  - Constructing distance vectors
  - Determining if parallelization is safe
L6: Memory Hierarchy I: Data Placement
  - 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?
L7: Memory Hierarchy II: Reuse and Tiling
  - Safety and profitability of tiling
L8: Memory Hierarchy III: Memory Bandwidth
  - Understanding global memory coalescing (for compute capability > 1.2)
  - Understanding memory bank conflicts
L9: Control Flow
  - Divergent branches
  - Execution model
L10: Floating Point
  - Intrinsics vs. arithmetic operations, what is more precise?
  - What operations can be performed in 4 cycles, and what operations take longer?
L11: Tools: Occupancy Calculator and Profiler
  - How do they help you?

Next Time

- March 23:
  - Guest Lecture, Austin Robison
- March 25:
  - MIDTERM, in class