CSC266/ECE206 Introduction to Parallel Computing using GPUs

Sreepathi Pai

University of Rochester

September 6, 2017
Outline

1. Organization

2. Performance Metrics

3. Program Optimization
Outline

1. Organization
2. Performance Metrics
3. Program Optimization
Lectures: Dr. Sreepathi Pai
- E-mail: sree@cs.rochester.edu
- Office: Wegmans 3409
- Office Hours: By appointment

Labs: Dr. Alex Page
- E-mail: alex.page@rochester.edu
Places

- **Class:** CSB 523
  - M, W 1650–1805
- **Course Website:**
  - [https://cs.rochester.edu/~sree/fall-2017/csc-266](https://cs.rochester.edu/~sree/fall-2017/csc-266)
- **Blackboard:**
  - Announcements, Assignments, etc.
- **Piazza:**
  - TBA
No required textbook for the class
- Useful to have a book on architecture as a reference
  - But this is not a computer architecture class
- Links to manuals, papers, etc. will be provided
  - Feel free to search for them
You will demonstrate your mastery of the course goals. Specifically, for a program, you will:

- Identify parallelization opportunities
- Implement programs on the GPU
- Optimize programs on the GPU
Outline

1. Organization
2. Performance Metrics
3. Program Optimization
Metrics we’re interested in

- **Latency**
  - Time units: 1 µs or 1,000,000 cycles
  - Lower is better

- **Throughput**
  - Rate: FLOPS or Instructions per Cycle (IPC)
  - Higher is better

- **Other interesting performance metrics**
  - Power (Watt)
  - Energy (Joule)
Applications where latency is crucial

- Audio/Video
  - MP3 players
  - MPEG4 players
  - VoIP (e.g. Skype)

- Games
  - Multi-user gameplay
  - Responsiveness

- Servers
  - Search Engines
  - Web applications
Applications where throughput is crucial

- Audio/Video
  - MP3 encoders
  - MPEG4 encoders
- Games
  - Frame rate
- Scientific Applications
  - Molecular Dynamics
  - Finite-element Code
- Servers
  - Search Engines
  - Web applications
Better performance can open up new vistas
Outline

1. Organization
2. Performance Metrics
3. Program Optimization
Principles of Optimization

- Work less
- Work cheaply
- Work concurrently

*applies to programs only*
Layers for Java

- javac
- Class Files
- Java Byte Code
- Java Virtual Machine
- Assembly
- Assembler
- Operating System
- Language Runtime
- Binary
- Process
- Instructions
- Processor
Conclusion

- Two metrics of interest
  - Latency
  - Throughput

- Unit of work is the *instruction*

- Principles of optimization
  - Use fewer instructions
  - Use cheaper instructions
  - Concurrent instruction execution

- C/C++ chosen for
  - Fewer abstractions
  - Easier understanding
Acknowledgements

Images of Toy Story and GMail from Wikipedia