COSC 462
Parallel Programming

George Bosilca and Piotr Luszczek

TA: Wyer, Austin R

All you need to know about …

- http://icl.cs.utk.edu/classes/cosc462
- Prerequisite CS360: System Programming
- C/C++, make, python/gnuplot/R

George Bosilca
Claxton 308

Piotr Luszczek
Claxton 218

Grading

- Exam 1 = 20%
- Exam 3 = 30% (cumulative)
- Homework = 30%
- Project = 20%
- Grading on the curve

Exams

- 1 midterms and 1 final
- Grading
  - On paper, multiple choice
  - Some questions will require a more detailed answer

Grading

- Exam 1 = 20%
- Exam 3 = 30% (cumulative)
- Homework = 30%
- Project = 20%
- Grading on the curve

Homework

- [Weekly] programming projects
  - 4-8 extra hours of work
  - Based on lectures
  - Incremental additions over the duration of the class
- Grading
  - Correctness
    - Of the result
    - Of the principle of the homework
  - Performance
    - Not required on all homework
    - Except when clearly specified

Grading

- Exam 1 = 20%
- Exam 3 = 30% (cumulative)
- Homework = 30%
- Project = 20%
- Grading on the curve

Project

- Team work encouraged (max 3)
- Mostly topics in parallel computing not covered in class
  - The list on the class website will be updated before the project start
  - One team per subject
    - First come / First serve
- Return 5 minutes video max
- Youtube, Vimeo, * with public visibility
- Slides / Animations / Narration / Links

Textbooks

- Generic
  - *Parallel Programming in C with MPI and OpenMP*, by Michael J. Quinn
  - *An introduction to Parallel Programming, by Peter S. Pacheco*
- Specialized
  - *Using MPI, Third Edition, by William Gropp, Ewing Lusk and Anthony Skjellum*
  - *Using Advanced MPI Modern Features of the Message-Passing Interface, by William Gropp, Torsten Hoefler, Rajeev Thakur and Ewing Lusk*
  - *Programming Massively Parallel Processors, Third Edition: A Hands-on Approach*, by David B. Kirk and Wen-mei W. Hwu
- Online documents MPI 3.1, OpenSHMEM.
- Don't hesitate to use your preferred search engine to find more information and/or examples

Homework

- Github classroom and github
- The homework will have a repo that you will fork (github interface)
  - Once a new homework has been added you will update your fork
    - git remote add XXX master
    - git pull
  - Each homework will be developed in a branch with the well-defined name (hw#) (not capitals)
    - Upon deadline I will pull the branch, test it and add a file with comments and grades
    - You will be free to merge the branch in your fork (or not)
    - I do expect you to keep things private (unless otherwise specified)
  - Each homework will generate a library
    - Automatic testing is WIP

What you will learn

- Why parallelism is important
- How to expose the parallelism available on an algorithm
- How to evaluate your algorithm scalability
- How to use parallel and distributed programming paradigms to reach your goals
  - POSIX threads, OpenMP
  - MPI, OpenSHMEM
  - CUDA
  - MPI+X