High Performance Computing Systems

- Course structure and expectations
- Overview

Doug Shook via Roger Chamberlain

Instructor

- Doug Shook
  - Office: Jolley 534
  - Office Hours: Wednesdays 2:30-4

Course Breakdown

- Website – www.cse.wustl.edu/~dshook/cse566
- Homework – 80%
  - 4 over the course of the semester
  - Groups of 2
  - C/C++
- Presentation – 20%
  - Must submit a proposal by 7/5

Homework

- Moderate amounts of programming
- More interested in answering (and asking!) questions:
  - How can we make this code run more efficiently?
  - What effects do different programming paradigms have?
  - How can we measure code performance?
- Experiment!
- Report

Policies

- Grading will be done on a straight scale
  - Curved if necessary (hint: it probably won't be)
- Class attendance is not mandatory
- Late work / Extensions

Academic Dishonesty

- Collaboration is encouraged!
- Over the line
  - Working in groups of more than 3
  - Showing your work to another group
  - Internet usage:
    - Finding sources, ideas, examples – OK
    - Copying text, ideas, code – Not OK
- Zero Tolerance

Why do we need HPC?

- Classic Example: Weather forecasting
  - Hundreds of thousands of weather stations, all collecting large amounts of data
  - Cubic Grid



Weather Modeling

- Assuming that each grid point requires 100 floating point instructions to process:
  - How many instructions for one point in time?
  - Hourly for 2 days?

- What would that look like on a...
  - Pentium 4 (2000)?
  - Core i7 (2012)?
  - NCAR Weather Supercomputer?

- And that's not even the entire planet....



Grand Challenges

- Computing goals set in the late 80s by the government:
  - Fluid dynamics
  - Nature of matter
  - Symbolic computations
  - Evolution of galaxies
  - Blood flow through a heart

Modern Examples

- IBM Sequoia – 3rd fastest super computer
  - Used for nuclear weapons simulations
    - How to dispose of old weapons?
- WATSON
  - Jeopardy!
  - Human computer interaction
- Pleiades
  - Owned and operated by NASA
  - Spaceflight simulations
  - Galaxy collisions

HPC at Wash U

- Streaming computing
  - Autopipe – Dr. Chamberlain
- Clusters
  - We're sitting in one
- Med School
  - Supercomputer with 1800 cores, 19 TFLOPS



Goals

- Parallel programs exist, but don't always perform well...
  - Why?
  - How can they be improved?
  - How can we reliably measure performance?
- Programming Paradigms
  - Shared Memory, Message Passing, Streaming
  - How does the choice of a paradigm affect performance?
- Application
  - What can we learn from existing HPC systems?
  - What problems exist with current methods?
  - What can we expect from future systems?

Tools

- Profilers
  - gprof
  - top
  - operf
  - perf
- Libraries / Frameworks
  - BLAS
  - OpenMP
  - MPI

Tools

- Languages
  - C/C++
  - FORTRAN
  - CUDA / OpenCL
  - VHDL/Verilog
- Platforms
  - CPU
  - GPU
  - FPGA
  - Hybrid
    - .