Designing and Building Applications for Extreme Scale Systems

William Gropp
www.cs.illinois.edu/~wgropp

Course Goals

- Learn a **quantitative** approach to designing and understanding the performance of applications on extreme scale systems
- Practical performance expectations (less precise than an estimate or model)
  - How much performance should you expect?
  - How can you form and test hypotheses about performance?

Course Description

Learn how to design and implement applications for extreme scale systems, including analyzing and understanding the performance of applications, the primary causes of poor performance and scalability, and how both the choice of algorithm and programming system impact achievable performance. The course covers multi-and many-core processors, interconnects in HPC systems, parallel I/O, and the impact of faults on program and algorithm design.

Course Description

Learn how to **design** and **implement** applications for extreme scale systems, including **analyzing** and **understanding** the **performance** of applications, the primary causes of poor performance and scalability, and how both the choice of algorithm and programming system impact achievable performance. The course covers multi-and many-core processors, interconnects in HPC systems, parallel I/O, and the impact of faults on program and algorithm design.

Course Topics

- Core Performance
  - Memory including cache
  - Pipelining
  - Vectorization
- Node Performance
  - Threads and OpenMP; correctness and performance hazards
- Internode Programming with MPI
  - Point to point and collective
  - Impact of network topology
  - One-sided communication
- Parallel I/O
- Hybrid Programming with MPI+OpenMP, MPI+MPI

Assignments

- Weekly Homework, with experiments
- Topics include
  - STREAM memory performance
  - Transpose and effect of cache
  - Matrix-Matrix multiply
  - Vectorization for orthogonalization
  - STREAM with Threads
  - Matrix-Matrix multiply with OpenMP
  - MPI Communication Performance
  - MPI Collective Communication
  - Parallel I/O

Class Project

- Students make project proposal
  - Recommend one close to their (research) interests
  - Blue Waters time available and encouraged but not required
  - Report on a quantitative expectation of performance, a comparison to what they observed, and a discussion of the results

Experiences

- Flipped classes
  - Lectures prerecorded
  - Range from 20 to 50 minutes
  - Include questions, pointers to other references
  - Students watch on their schedule (and their playback style)
    - Some watch at 1.2x speed (squeeky!)
    - Some watch and pause to check, test themselves

Experiences

- Class time used for discussion
  - Sometimes used to clarify lectures
  - Often used to add depth, practical experience, additional presentations
  - Discuss experiments; the complexities of real data

Course Future

- Lectures, videos will be available online
- Hope to teach the course again and soon
- Blue Waters a valuable tool
  - Students got to see **1000x** performance difference in different approaches to parallel I/O (HW#10)

Summary

- Design and performance understanding can be quantified
  - Results used to design and improve a wide variety of applications
- Similar concepts can be applied from single core to full machine
  - Different sources of cost, overhead, but often similar performance models
- Flipped class is more fun for everyone
  - I'm looking forward to the class evaluations to see if it is more effective