CSE 891-Section 1
Parallel Computing: Fundamentals and Applications

Final Project

Description: Final project is an opportunity to apply the concepts discussed during lectures in a "real-world" application context. You are expected to propose your own project ideas, work in teams or individually to develop algorithms and software to solve the proposed problem, report your results and present them at the end of the class. While collaboration is highly encouraged, projects can also be done individually. Recommended team size is 2 people, but for large projects you may have 3 people on your team (with approval). The main focus needs to be on computer science related topics such as the application of algorithms & data structures, performance and scalability analysis.

Some examples: Parallelizing an application or porting an existing parallel application to some new hardware (multi-core CPUs, Intel Xeon Phi or NVIDIA GPUs), building and testing a tool for high performance computing (HPC).

Grading and timeline: Final project constitutes 40% of the total grade in this class. The project consists of 4 main parts:

- Abstracts (1 page), due Monday 11:59 pm, October 20th - (5 pts)
- Progress report (3-4 pages), due Wednesday 11:59 pm, November 19th (10 pts)
- Final report (8-10 pages), due Monday 11:59 pm, December 8th (15 pts)
- Presentation (20-25 minutes + 5 minutes Q&A), Tuesday 3-5 pm, December 9th (10 pts)

Abstract: Give an overview of the final project and identify the team members. This (roughly) 1 page document is expected to answer the following questions:

- What is the problem and why is parallel processing necessary to solve it? Try to describe the benefits that an efficient parallel implementation would have on researchers and practitioners in the field.
- What is the state of the art in terms of the existing solutions? Are you planning to parallelize a sequential code? or port an existing parallel implementation to a different platform (a multi-core CPU, a many-core/GPU accelerator)? or work on characterizing and improving the performance of an existing parallel application? Are there any similar efforts reported in the literature?
- What are the tools and libraries that you are planning to use? OpenMP, MPI, CUDA, BLAS, LAPACK, etc.? What is the scale of the problems that you are aiming to solve?
- How would you evaluate the success of your project? Comparison to an existing solution is certainly one way to do it. If that's not possible, performance analysis in terms of processing speed and/or scalability could be complementary alternatives.
- Finally, elaborate on the division of work among team members. Each team member is expected to contribute equally to the project.