Designing and building applications for extreme scale systems
CS598 SP2016

Project Proposal
Due March 17, 2016

Goals for the Project

- Apply what you've learned in class to an application
- Gain experience creating quantitative performance expectations
- Gain experience using extreme scale systems

Goals for the Project Proposal

- Define a well-defined project that is not too ambitious (and not too simple) for a 4-week project.
- Receive feedback on the plan before starting the project

Task:

Write a short proposal for your project. This project will be due a week before the end of class. It should demonstrate a quantitative approach to understanding and achieving performance. You are encouraged to use a problem or application that is familiar to you, for example, the subject of your research or a benchmark code that you have run.

Projects will be graded on how well they apply the principles that we've discussed in class to the project; it is not necessary to achieve substantial performance improvement in your project. Rather, you need to demonstrate an ***understanding*** of the performance of the code that you use in your project, and either demonstrate how to improve it or explain why it is difficult or impossible to improve the performance of that code.

A team of two is recommended but not required for the project proposal. Your final report should contain a description of what each team member contributed to the project.

Projects will have access to Blue Waters but are not required to use that system. Refer to the "TopicSchedule.pdf" file on the Moodle to see what we will be covering in the next few weeks.

What to turn in:

Turn in a project proposal. It should contain:

1. A brief description of the problem and why you are interested in it. What is your goal (e.g., better performance, better scalability, understanding the performance limits of a code)? (typically around 1/4 page)
2. The source of the code. Does it already exist? Will you be writing it? How long is it/do you expect it to be? Provide a URL to existing code. (1/4 page)

3. How will you establish a performance ***baseline***: the performance before you start to analyze or change the code. Will you run the code? Where and with what inputs? How will you measure the results? (1/2 page)
4. How will you analyze (quantitatively) the performance of the code? You are encouraged to use the simplest approach possible. (1/2 page)
5. What do you expect to do as a result of your analysis? Give a few examples. Your project should implement at least one modification to the code. Note that the specific changes will depend on what you learn during the project; this section is to show that you have some ideas that you will consider. (1/2 page)
6. How will you compare the results that you measure with your analysis? (1/4 page)

The total length of the project proposal should be about ***2 pages***. It may be longer or shorter, but should address these issues concisely. You may want to propose basic and stretch goals for your project (you will not be penalized if you do not reach your stretch goals – the point of the project is for you to learn as much as possible).

Example of project ideas:

1. Take one benchmark (see, for example, https://asc.llnl.gov/CORAL-benchmarks/) as the base code, and model the performance (for example, vectorization for single processors or scalability for massively parallel benchmarks). Suggest changes that might be tried to improve performance, and implement and measure at least one.
2. Take a code that you are working on, and consider one part, such as a computationally intensive loop or routine. Model the performance and investigate ways to improve the performance.
3. Same as 2, but consider I/O instead of computation (we'll be talking about I/O in early April).
4. Take a simple code, such as the matrix transpose example, but construct a more detailed performance model, and use it to design an implementation based on that performance model. Measure the performance of that implementation and explain your results.

Project topics do not need to be unique – but if several groups pick the same topic, they must work independently. If you have a question about whether an idea is appropriate, email me at wgropp@illinois.edu.