

The goal of this project is for you to (1) evaluate the performance needs of a benchmark or application, (2) provide a reasonable estimate for the best possible performance for this application on either the multicore, Cray, or NVIDIA environments in the department, (3) develop a parallelization and loop transformation strategy for improving the performance of the application, (4) apply the parallelization/transformation with the help of loop transformation tools, and (5) compare what can be done in a fully automated way in one tool versus what can be done by hand and by the other tools we have discussed in the course. If you are able to go one step further and extend the tool to enable the automation of your optimization strategy, then the resulting paper from your project could be publishable. A meta-goal is for you to learn how to organize, propose, execute, and present a research project.

All of the components of the report must be written in latex. The submission should be a tar ball with all of the latex source, the figures, a Makefile that creates a pdf file, and the resulting pdf file.

There should be no spelling or grammar mistakes. Students should visit the writing center if they have some questions about their writing. It is also useful to have other students proofread your material and provide feedback.

## 1 Guidelines

Each project is to be done individually.

In class we will be working with a set of computations and a set of tools to automate parallelization and transformations. You need to identify either a different benchmark/application and/or a different automation tool than the ones we are covering in class. Possibilities for both will be available on the course resources page.

I recommend that you discuss the project with me in office hours before writing your first proposal. Insufficient proposals will need to be rewritten.

## 2 Deliverables

For each deliverable, you need to submit an electronic copy by email to [mstrout@cs.colostate.edu](mailto:mstrout@cs.colostate.edu) AND via RamCT.

### 2.1 Proposal (5% of course grade)

**Due:** Friday February 24, 2012 at NOON (used to be February 17th but was extended)

The proposal should include a descriptions of the benchmark/application you plan to study, the context of your chosen benchmark/application (who uses it, what is the main performance issue), the automation tool you will be evaluating, and a *detailed* time line for the project. At the time you write your proposal you should be familiar with the benchmark/application and the automation tool you will be using. Show this with some preliminary performance analysis results on the benchmark/application and a small example with

the automation tool. The small example should be relevant to the benchmark/application's performance issue. The proposal should be no longer than 3 pages of text.

Unsatisfactory proposals will returned to the students for revision.

## 2.2 Intermediate Report (10% of course grade)

**Due:** Wednesday March 28, 2012 at 11:59pm

At the time you write the intermediate report you should (1) already have identified the performance analysis issues in your benchmark/application in detail, (2) have performed your parallelization and loop transformations by hand and have performance graphs, and (3) have started to work with the automation tool to do parts of the transformations. In other words, your report should detail how parts of the transformations/parallelization of can be automated in the selected tool.

If your original selected tool is incapable of doing any of the by hand transformation, then you need to have identified another automation tool that can perform at least part of your planned transformation and show how that tool does some portion of your planned transformation.

The intermediate report should also (4) evaluate the effectiveness of your schedule and (5) detail how you will schedule the remaining work for the project. You should also (6) describe how you have tested that the output of the transformed code matches the output of the original code.

The intermediate report should be no more than 5 pages of text. See the Resources link for documents that describe how to write a paper in Computer Science.

## 2.3 Final Report (15% of course grade)

**Due:** Wednesday May 2, 2012 by 11:59pm

The final report should be in the form of a 6-10 page conference paper (e.g., you could use the IEEE double-column latex format). As with a conference paper, the final report should (1) describe and motivate the problem (why is the application important, what are the performance problems in the application), (2) present the approach (your transformation and automation strategy), and (3) evaluate the approach (evaluate the performance of the transformation strategy and the usability and performance of the automation tool in relation to doing the transformation by hand and also in comparison to the other tools covered in class).

## 2.4 Poster (5% of course grade)

**Due:** Thursday May 10, 2012 2-4pm

During the finals time slot for this class, you will also be presenting a poster on your project. Distance students will need to present their poster via skype during a time they schedule BEFORE the on campus poster session. The distance and on campus posters should follow the guidelines described in <http://lib.colostate.edu/sites/default/files/images/services/computers/PowerPointPostersFinalx.pdf>. The distances students do not need to actually print the posters and can just send a pdf or powerpoint document via email to the instructor.

See the Resources link on the web page for information about putting together a poster. Your poster will be graded on (1) how clear the message is presented, (2) illustrations that support the message, (3) succinct and on message text, and (4) organization and flow.