CS1645/2045: Introduction to High Performance Computing (Spring 2021)

Department of Computer Science, University of Pittsburgh

Term Project: Parallel Languages and Libraries
Group Project

Release Date: Mar. 8, 2021

Phase 0 Due: 8:00 PM, Mar. 5, 2021 | Phase 1 Due: 8:00 PM, Mar. 12, 2021
Phase 2 Due: 8:00 PM, Apr. 16, 2021 | Phase 3 Due: 8: PM Apr 19 - 21, 2021

Purpose of the project

The primary goal of this project is to learn a new parallel language or library and present the weaknesses and the advantages of what we learned in the class. You need to implement at least the matrix multiplication problem on the selected language/library and submit the code and the ppt through github and as a zip file on the course website.

Phase 0: Team formation
Due: 8:00 PM, Mar. 5, 2021

Phase 1: Select your subject
Due: 8:00 PM, Mar. 12, 2021

Send an email to the instructor and the TA with 3 preferred subjects. We will follow the **FIFO** order to take the final decision regarding the subject of each team.

- The subjects should be sent in preference order. The first option will count as the most preferred.
- Subject of the email: CS 1645 Project preference.
- Add you team members again on the body of the email.

Subjects:

1. Threading Building Blocks (Intel TBB)
2. Linda
3. OpenCL
4. OpenACC
5. CnC
6. CUDA
7. Cilk
8. Cilk Plus
9. GO
10. Rust
11. Open suggestions

Phase 2: Implementation and submission through GitHub
Due: 8:00 PM, Apr. 16, 2021

We will be using the git version control system via GitHub for the development and submission of term project this term. Please visit the above URL to register an account. You will NEED a private repository on GitHub to host your project. They are provided for free to students, see https://education.github.com/ for details on how to apply.

To submit a given project phase, simply push all required code to GitHub, and send an email to the TA (Cc'ing the instructor) with the commit ID representing your submission of that phase of the project.

Organize yourselves into teams of 2 or 3 people and create a private project repository on GitHub. Add all team members as contributors to the repository. Further add your instructor (GitHub username costantinos) and your TA (GitHub username yud42) as contributors to the repository. Create a single file named team_members.txt file in your repository listing each team member's GitHub username, Pitt ID (e.g., "kok25"), and full name. Be sure to push team_members.txt file to your group's repository by the deadline. Be sure to share only 1 repository per team with the instructor/TA to avoid grading confusion.

Your project will be collected by the Instructor/TA via the GitHub repository that you have shared. To turn in a project phase, you must do three things by the deadline: Make a commit to your project repository that represents what should be graded as your group's submission for that phase. The message for this commit should be "Phase X submission" (for this phase, "X" should be "0"). Push that commit to the GitHub repository that you have shared with the instructor and TA. Send an email to the instructor and the TA with the title "[CS1645] Project phase X submission" that includes a link to your repository on GitHub and the Git commit ID for the commit that should be graded. The commit ID is a 40 character string (a SHA-1 hash) and can be found on GitHub or as part of the output of the "git log" command. Be sure to send this email to the instructor and TA by the deadline. Pushing the code to GitHub is not enough! If you do not email the instructor and TA a commit ID as specified above by the deadline, your submission for that phase will be considered late and you will receive a 0 for that phase.

Project report: The project report should be a well-presented technical report discussing your subject. You should explain how the language/library works, provide some examples of the language showing the syntax, experiments to support your finding and analyze the results. The analysis is an important component of your report. The written report should be two to ten pages in length.

The structure of the report should be as follows:

- **Introduction**: A brief introduction and motivation of the language/library.
- **Syntax**: Examples of the syntax and description of the underline properties.
- **Comparison**: Experimental evaluation of the language/library. Compare with the POSIX threads and OpenMP at least for the matrix multiplication problem.
- **Discussion**: During the discussion section you should answer the following questions: (a) What was the hardest part of the project?, (b) Is this language/library better than Pthreads and OpenMP?, (c) How much better/worse in terms of performance? Why?, (d) How easy/difficult was the implementation of a parallel program?, and (e) Is it a good idea to add the language as part of the class? Why?.
- **Conclusions**: Write your thoughts about the language/library, what you learned from the project and what you believe for the future deployments of the selected language/library.

Submission for grading should be the following files:

- **code.zip**: the code for your implementations.
- **report.pdf**: the report consisting all of your findings and answering all the questions described in this handout.
- **project.pptx**: the presentation of the selected subject.
- **Readme.txt**: the file with instructions of how to compile and run your driver program.

Phase 3: Presentations
Due: 6:30 PM, Apr. 19 - 21, 2021

In the powerpoint for your presentation (20 slides) you need to explain the syntax and the underline semantics of the language or the library. You need to implement at least the matrix multiplication problem on the selected language/library and submit the code, the report and the ppt through github and as a zip file on the course website. You should compare it with the languages that you learned during the class. Remember to add some plots to your presentation and to your report!!!

Academic Honesty

The work in this assignment is to be done *independently* by each student. Discussions with other students on the project should be limited to understanding the statement of the problem. Cheating in any way, including giving your work to someone else will result in an F for the course and a report to the appropriate University authority.

Enjoy your class project!