IT 388 – FINAL PROJECT
Spring 2022

Important Deadlines

Project Topic and Team Names	Wed, March 02	11:55 PM
Project Proposal	Thu, March 24	11:55 PM
Project Progress discussion	Tue, April 19	5:00 PM during lecture time
Final Presentation + codes	Tue, April 26 Thu, April 28	5:00 PM (groups 1-4) 5:00 PM (groups 4-8)
Project Paper (Graduate students only)	Tue, May 3	11:55 PM

1. OVERVIEW

There will be a final course project worth 20% of your final grade. Your final course project will be to solve a compute-intensive problem with parallel processing. You will identify a compute-intensive problem, design and implement a parallel algorithm, and evaluate its performance. You may choose to parallelize an interesting application, or a problem related to a research topic in which you might be involved, or propose to further enhance the spectrum of problems and methods discussed in class. This is a group project, students are required to form teams and divide the work among the team members.

Group size:

Undergraduate students: 5 groups with 4 students per group, and 2 groups with 3 students per group.

Graduate students: 2 graduate students per group

2. CHOOSING A PROJECT

Choose some application of interest to your group and that presents significant computational challenges. Identify suitable algorithms and benchmark test cases. Implement and experiment with different possible approaches.

Some example of areas of interest from your homework among others:

1. Image processing (edge detection, compression algorithms)
2. Optimization algorithms (Genetic algorithm, gradient descent, …)
3. Graphs (shortest path algorithm)
4. Machine learning (scale up a recent learning algorithm, for example - clustering, large scale classification, deep learning)
5. Clustering algorithms
6. Physical simulation (fluid dynamics simulation, heat dissipation)
7. Protein folding
8. Scientific computing
9. Pattern recognition (pathology diagnosis)
10. Weather prediction

A few important things to consider:

- Application-driven projects can be very interesting, but it's important to remember that this is a course on parallel computing. The application should be susceptible to parallelism, but not trivially so. You should not have to spend large amounts of effort just understanding the application and the relevant algorithms. Avoid projects where you would need to write large amounts of code just to implement the application.

- It's important to scope the project so that it is neither too easy nor too hard. In terms of effort, your project is worth 20% of your total grade for the course. Since it is difficult to predict how hard or easy a given programming task will be, it helps to formulate a plan that has multiple goals, with different levels of difficulty, and with clear objectives for each.

- Your project does not need to be original research. It is perfectly acceptable to read some research papers on a new way to solve a problem, implement the ideas in the paper, and measure the performance. You do not need to invent new algorithms.

- It's allowed to build on work you are doing or have done for a different course or for a research project, as long as the previous focus was not on parallel computing.

- It's allowed to use existing code written by you or by someone else. However, the more you rely on existing code, the more I expect you to do a very thorough job conducting and evaluating experiments, examining alternative approaches, and in general being very thorough in your evaluation.

3. IMPORTANT PROJECT DEADLINES

As indicated above, there will be a series of deadlines to allow for suitable feedback and to help you manage your time.

3.1 Project Topic and Team Names (11:55 PM, Wed March 2, 2021)

There will be 5 groups with 4 students, and 2 groups with 3 students in each group. No two groups will be allowed to work on the same topic/problem. We will form the groups during class on Feb 22, 2022.

Submit the names of your group members with the selection of the two topics (order of preference), including a short description of the problems to be solved in each topic using parallel processing (2-3 sentences). If groups choose the same topic, I will randomly assign the second to one of the groups and will inform you of your final topic during lecture on March 3, 2022.

3.2 Project Proposal (11:55 PM, March 24, 2022)

Writing your ideas down forces you to organize your thoughts about your project. It gives the instructor the ability to verify that your plans are of the right scope given our expectations, and it also gives us the ability to offer suggestions and help.

Your project proposal should include the following:

Title: Please provide the title of your project

Team members: names of all team members

Team name: Not required.
Abstract: Summarize your project in no more than 2–3 sentences. Describe what you plan to do and what parallel systems you will be working with.
Problem Description: Describe the application or piece of the application you are going to implement in more detail. Include abstraction (equations, graphic, a block diagram or pseudo-code) of the basic idea.
What aspects of the problem might benefit from parallelism? Why is it important? What are the dependencies? How to partition data & computation to avoid synchronization?
Intellectual Challenge: Describe why the problem is challenging. What aspects of the problem might make it difficult to parallelize?
Resources: Describe the resources you will use. What code base will you start from? Are you starting from scratch or using an existing piece of code? Is there a book or paper that you are using as a reference? If so, provide a citation?
Work Plan Schedule: Make a work plan for your project, including a weekly meeting with your team members. List what the group plans to get done each week from now until the end of the semester in order to finish your project in a timely manner. Also, specify how the work will be divided among team members. You can use a table for this.
References: List all references you have used, make sure to cite them in the text where you used them.

- Proposal length: between 3 and 5 pages, not including references. Format all you document with margins 1 inch, single line spacing, use one of the these font types and sizes: Arial (11 pt), Helvetica (11 pt) or Times (12 pt). Add page number at the bottom of page.

3.3 Project Progress Discussion (April 19, 2022)

The project progress discussion will be an informal discussion about how your project is going. Be prepared to:

- Summarize your project
- Report on how things are progressing relative to the schedule you provided in your project proposal. What work have you completed so far, what adjustments do you need to make to the original schedule and project goals?
- Describe how you are doing with respect to the goals stated in your proposal. Do you still believe you will be able to produce all your deliverables? If not, why?
- At this time, you should have preliminary results, be prepared to show them.

3.4 Final Presentation + Codes (Week before Finals, April 26 and April 28, 2022)

3.4.1 *Final Presentation*

We are going to have final oral presentations (15min + 5 for questions) in which the groups will have a chance to present their projects to each other and to me. Each project team will prepare and present a power point presentation given a high-level perspective on the project (at least two team members should present). Here is a list of items to guide you to prepare your presentation.

Title page
Motivation and goals
Problem to parallelized: Describe the algorithm, or application you parallelized in computer science terms. Make use of abstractions, diagrams, figures etc.

- What are the key data structures?

- What are the key operations on these data structures?
- What are the algorithm's inputs and outputs?
- What is the part that computationally expensive and could benefit from parallelization?
- Break down the workload. Where are the dependencies in the program? How much parallelism is there? Is it data-parallel?

Approach: Describe how your implementation works. Your description should be sufficiently detailed to provide a basic understanding of your approach. Again, it might be very useful to include a figure here illustrating components of the system and/or their mapping to parallel hardware.

- Describe how you mapped the problem to your target parallel machine(s).
- Did you change the original serial algorithm to enable better mapping to a parallel machine?
- If your project involved many iterations of evaluation and optimization, please describe this process as well. What did you try that did not work? How did you arrive at your solution?
- The notes you've been writing throughout your project should be helpful here. Convince us you worked hard to arrive at a good solution.
- If you started with an existing piece of code, please mention it (and where it came from) here.

Results: How successful were you at achieving your goals? Result sections can differ from project to project, but it is expected your evaluation to be very thorough (your project evaluation is a great way to demonstrate you understood topics from this course). Things to include:

- Define how you measured performance. Is it wall-clock time? Speedup? Efficiency? Could be an application specific rate? (e.g., moves per second, images/sec)

- Describe your experimental setup. What was the size of the inputs? How were benchmark data generated?

- Provide graphs of speedup or execution time. Please precisely define the configurations being compared. Is your baseline single-threaded CPU code? It is an optimized parallel implementation for a single CPU?

- Recall the importance of problem size. Is it important to report results for different problem sizes for your project? Do different workloads exhibit different execution behavior?

- What limited your speedup? Is it a lack of parallelism? (Dependencies?) Communication or synchronization overhead?

- Deeper analysis: You can break execution time of your algorithm into distinct components (regions). What percentage of time is spent in each region? Where is there room to improve?

Concluding Remarks: Provide conclusion of your project work, also what would you extend your project if you had more time.

References: Have a slide with the list of references you used for your project, cite them along the text where you have used them. Note: no need to explain this slide, just have it.

Division of Work: Have a slide with which part of the work was performed by each group member. Alternatively, you can simply state: “Equal work was performed by both project members.”

3.4.2 Codes

Make sure to comment, and format your codes nicely
Create README file with a brief description of the main functions, and instructions on how to compile and run your code
Zip your codes, readme file, expanse job script, along with any test data file you have used. and send them on Reggienet

3.4.3 Project Paper (Graduate students only, due May 3, 2022)

Graduate students, your paper should include sections similar to the ones described above for the presentation. You are also encouraged to provide more detail if you wish. Note that some of the information in your final report can be pulled directly from your proposal if it is still accurate. You will write your final project in the IEEE conference paper format, here is a link to the template https://www.ieee.org/conferences/publishing/templates.html
Your paper length should be 6-10 pages, not including references.

4. COMPUTING RECOURSES

For testing purposes, you can use our Linux servers, and for measuring performance you should use the SDSC Expanse cluster.

5. PROJECT GRADING SCHEME

Project proposal	5%
Project Progress	5%
Final presentation + codes	10%