18-645: How to Write Fast Code I

Project 1:- Project Proposal

The course project is a semester long project that is meant for students to put what they have learned in class to practical use. Specifically, it allows the students to apply the knowledge from class to a particular problem from an application domain of their choice. It is also an opportunity for students to pursue a particular area (application, compiler and architecture) in greater detail.

Expected Project Outcome

The expected output of the course project is a fast and parallel implementation of the chosen problem that has been ***designed*** based on available hardware features, and the impact of the hardware features on performance.

A fast implementation where implementation choices cannot be explained will not meet the requirements of this project. Similarly, a faster implementation obtained by switching to a different algorithm is also not acceptable.

Instructions. For this project, the first deliverable is the formation of the project team, a proposal for the project, and the scheduling of a day/time for meetings to discuss your project.

A **PDF** document with the following three requirements is required to be submitted:

1. *Proposed Project.* A one page abstract for the proposed project is expected. Specifically, the abstract has answer the following:

   (a) What is the application domain / area you are proposing to work in?

   We want to know the application domain of the problem you are proposing. As there are many application domains, it is important to provide sufficient background information so that we can understand the context of your proposed project.

   (b) What is the proposed computation problem / algorithm(s) you intend to tackle?

   This is the core problem you are designing a fast implementation. Within that chosen application domain, tell us what is the computation problem you are trying to solve. Is there a specific algorithm you are addressing? Are you trying to design a particular routine(s) in a much larger application. Provide references if necessary.

   (c) What is the code base you will be using as a baseline to check for correctness and compare performance against?

   You will need to ensure that your implementation is correct. The baseline/benchmark implementation will be used to check that your fast implementation is correct. You **should not** be writing this benchmark on your own. This baseline/benchmark implementation should be running on the same computing platform as your final implementation. For example, if you are designing a CPU implementation, the baseline/benchmark should run on the same machine as your implementation. It should not be an implementation that runs on a GPU or on the cloud.

   You should ensure that you can run your chosen benchmark/baseline before selecting it for this project.

   The performance of the existing benchmark/baseline **should not** be a consideration. You will be evaluated on your design choices based on hardware available on the chosen platform.

   (d) Describe the size(s) of the dataset(s) you may need, and how you are getting the dataset(s).

Many computational problems require input data. Tell us how you will obtain the input data for your particular problem. If you are addressing a particular routine in a much larger application, how do you ensure that the input to your particular routine is obtained. If you are downloading a dataset, let us know if any preprocessing needs to be performed.

(e) What architecture do you intend to target?

Let us know the platform you are targeting. We recommend using platforms that everyone in the project team has access (e.g. ECE clusters), and are familiar with programming. If someone has a unique device which only they can program, that would be a bad choice for this project.

2. *Work Division.* While the project is a group project, it will be graded individually. Provide names of all team members and a rough division of the expected work amongst the team members.

   A good sized project would be a computational problem that can be subdivided into multiple subroutines. If it cannot be divided into sub-components, the proposed project is probably too simple.

3. *Date/Time of next meeting.* Individual Team meetings will the course staff be held to discuss your progress with your project. All team members are expected to attend the meeting. Provide three(3) distinct 1-hour time slots outside of class hours so that we can schedule to meet all the teams.

   Beyond using this time for meeting with the course-staff, these times would also be times that members of the team can meet up to discuss and work on their project.

   Order your three proposed timeslots in decreasing preference order.

Submission. As this is a group project, it is acceptable and expected that everyone contributes to the writing of the proposal. However, the same proposal must be submitted ***individually***. This serves as an agreement that everyone named on the proposal agrees to form a team, and agrees to work on the same project.

Short List of Past Projects

These are some application domains/algorithms that were tackled in past semesters. These are meant as examples and you are not restricted to either the domain or the algorithms for your project.

Project	**Benchmark**
Machine Learning	
Spiking Neural Net	Past Project
N-Gram	Existing Library
Decision tree (ID3)	Past Project
Convolution Neural Nets	Existing Library
K-nearest Neighborrs	SciKit
K- means	Existing Library
Genetic Algorithm	Existing Codebase
Graph Algorithms	
A*-pathfinding	Boost Library
Topological Sort	Multiple libraries
Single Source Shortest Path	Existing Libaries
Image Processing	
Image Filter	OpenCV Library
Corner Detection	OpenCV Library
Feature Detection (HoG)	OpenCV Library
Ray Tracer	Existing Codebase
3D space from point-cloud	Existing Codebase
Point cloud generation	Existing Library
ORB Feature Detection	OpenCV Library
Scene Carving	OpenCV Library
Other Algorithms	
Heart Beat Monitoring	Existing Codebase
Text Compression	Existing Library
Gesture Detection	Past Project