

# COSC 3P93

## Project – Step 2

**Due date:** October 13<sup>th</sup>, 2020 at 23:55 (11:55pm)

**Delivery method:** the student needs to deliver the assignment only through Sakai.

**Delivery contents:** document with answers and [Java, C, C++] codes if applicable (see [Submission instructions](#)).

**Attention:** check the [Late Assignment Policy](#).

### Project Overview

This project is intended for students to apply design patterns, as well as strategies, for parallel systems in a sequential solution. The patterns and strategies applied in the project for the design and implementation are the ones covered in classes and course reading materials. The final output of the project is a performance comparison between a sequential program and its respective parallel design. For both programs, students will have to write the codes themselves, as well as define performance analysis parameters and metrics for the comparative analysis. It is strongly recommended that the implementation language in this project be either C or C++. Mostly likely, student will need to familiarize with Linux/Unix systems, as well as bash command line, to compile, debug, and run their code. For each of the project steps, the students are expected to employ the concepts learned in class properly and report them in the document submitted together with their code.

### Step 2 - Description

The second step of the project consists of implementing a sequential version of an algorithm in the assigned project topic (subtopic). More important than the implementation itself, this step requires a full description of the algorithm (pseudocode), as well as a discussion of the possible performance improvement opportunities through a parallel construct of the same program.

### Step 1 - Specifics

The group will need to identify an algorithm in the topic that was selected and assigned to it. The algorithm must be designed to run sequentially and be fully implemented in either C or C++. At this step, a full and detailed description of the algorithm, or pseudocode, is essential. The description needs to emphasize on the algorithmic aspects of the code instead of the its general purpose/intent. For instance, the students should not discuss about the benefits in using a bio-inspired convolutional optimization method as meta-heuristic for a Hard problem; instead, the students should detail the parts of the code that may be considered a bottleneck in the code, may be implemented only sequentially, or may lead to a future parallelization potential.

The description document should objectively describe the following:

1. The design decisions and logic of the code in algorithmic aspects.
2. An analysis of performance particularities: range of parameters and respective consequences to execution time.
3. A discussion about bottlenecks in the code and areas/regions/blocks of code where parallelism can be applied.

Also, for the implementation of the code, the student should avoid at any cost the use of specialized libraries. In other words, students should fully implement the code themselves. The reason behind this approach is to avoid the use of any optimization (parallel execution optimizations) that such specialized

library may bring. Ultimately, in case the use of the library cannot be avoided, the students will have to fully describe the functions and proprieties of the library used in their code; this description should include implementation details of the library (to show the hidden optimizations it may contain).

## **Submission Material**

The submission for this assignment will consist of two parts:

1. The C/C++ code of your implementation, well divided and commented. The code should compile and should run properly. Please provide instructions on how to compile your code in your description document. As recommended, make your project compilible and runnable from command line for marking purposes, defining all setup steps, such as environment variable setups, to make your code run in any “foreign” environment.
2. A description document that explains your implementation decisions, justifications, issues present in the code, and challenges. The comments you added in your code should serve as a good start point to generate this document. You should describe how to compile and run your code, as part of your explanation. Word processing (MS Word) or PDF are acceptable file formats for this document (style marks may be associated in case the PDF is generated using Latex).

## **Marking Scheme**

Marks will be awarded for completeness and demonstration of understanding of the material. It is important that you fully show your knowledge when providing solutions in a concise manner. Quality and conciseness of solutions are considered when awarding marks. Every code added to the originals should be well commented and explicitly indicated in the code files; lack of clarity may lead you to loose marks, so keep it simple and clear.

## **Submission**

The submission is expected to contain two parts: the proper C/C++ code well commented and a description document. The document can be in either DOC or PDF format; it should be single column, at least single spaced, and at least in font 11. All content files should be organized and put together in a ZIP file for the submission through Sakai.

## **Late Assignment Policy**

A penalty of 25% will be applied on late submissions. Late submissions are accepted until the Late Submission Date, three days after this Project Step Due Date. No excuses are accepted for missing these deadlines.

## **Plagiarism**

Students are expected respect academic integrity and deliver evaluation materials that are only produced by themselves. Any copy of content, text or code, from other students, books, web, or any other source is not tolerated. The similarity check tool, Turnitin, will be employed to identify plagiarism in the submitted document. If there is any indication that an activity contains any part copied from any source, a case will be open and brought to a plagiarism committee’s attention. In case plagiarism is determined, the activity will be cancelled, and the author(s) will be subject to the university regulations. For further information on this sensitive subject, please refer to the document below:

<https://brocku.ca/academic-integrity/>