Parallel Computing Project Plan

David Thornton

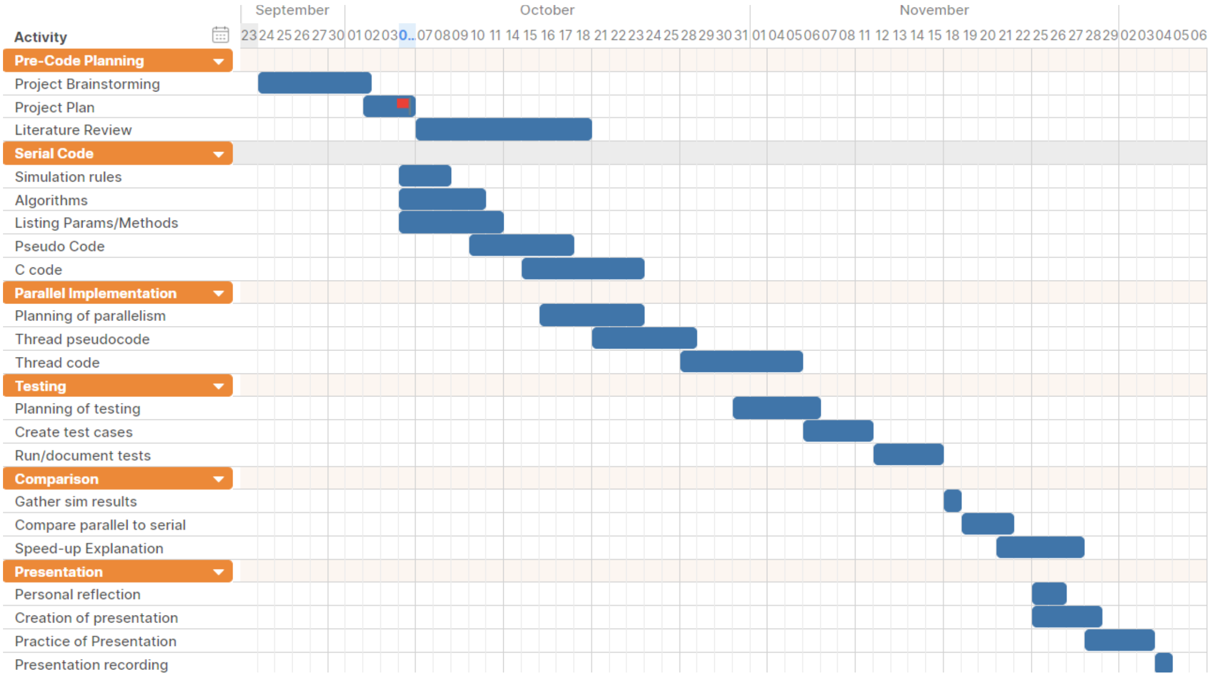
B00152842

0/10/2024

Declaration of Academic Honesty

I am aware of the University policy on plagiarism in assignments and examinations (3AS08). I understand that plagiarism, collusion, and copying are grave and serious offences in the University, and I will accept the penalties that could be imposed if I engage in any such activity. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. I declare that this material, which I now submit for assessment, is entirely of my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work.

Gantt Chart



Gantt Chart details

Pre-code Planning

This phase of the project involves the research of the problem, planning for the project timeline and the written literature review. The brainstorming began after the first lecture, by reading the Project Description and Guidance document and the When Zombies Attack paper (Munz, et al). This gave a better understanding of the assignment and raised issues I didn't know how to solve. These issues along with other questions formulated through further brainstorming were brought to Kevin during lab time to be resolved so further planning could go ahead, leading to the creation of the project plan. Further research during the development of the serial code allowed for a better understanding of the problem and how the parallelism of the project will function, which was documented in the Literature Review.

Serial code

The parallel computing lab on the week of the 30th of September began the initial planning for the serial code, with listing simulation rules, algorithms, parameters and methods that could be used in the creation of the serial program. This planning will be continued until all the logic for the program is fully accounted for, at which point the development of the pseudo code program can begin. Once the core functionality of the pseudo code program is complete the C code program can start to be started.

Parallel Implementation

Once the Serial code has been fully planned and pseudo code has begun, the initial planning for the parallelism implementation can begin. Using information form the literature review along with module lectures, the specific functions and parameters needed for the parallel part to function. Once this step is done the pseudo code for thread implementation can be written before the thread code itself.

Testing

When all parallel and serial code is fully implemented it will be tested to look for bugs and unaccounted edge cases within the code. First, these tests will need to be planned by evaluating any possible edge cases scenarios and all cases that could take place in standard running. Once these scenarios have been evaluated, the test cases to be used in testing edge cases and several standard cases can be created. These cases will then be run several times, documenting results and any flaws in code that need to be fixed.

Comparison

When code has been fully completed and all bugs have been fixed, the comparison of the serial and parallel code can begin. The test cases from the testing phase and several randomly generated cases will be ran through both the serial and parallel programs, recording results of each case. These results will then be compared to evaluate the difference in performance between the two programs and why these differences occurred (including why the parallel code was faster and why it was n times faster were n is the number of threads used).

Presentation

Finally, all relevant information gathered from all previous steps will be compiled into a PowerPoint presentation document to be used in the project presentation video. The presentation will then be rehearsed several times, tweaking the PowerPoint document as needed before recording the final version.