UNIVERSIDAD COMPLUTENSE MADRID

Final Project

Your final project is to solve a data-intensive problem with parallel processing on the cloud. You will collect the data, implement the tool, and analyze the performance of an end-to-end application.

Group Size: Students are required to form groups and to partition the work among the group members. The final project must be done in groups with **3 to 5 students** each (exceptions by permission of the instructor). You can use the course forum to find prospective group members. You may also find and discuss project ideas on the forum. In general, we do not anticipate that the grades for each group member will be different. However, we reserve the right to assign different grades to each group member if it becomes apparent that one of them put in a vastly different amount of effort than the others.

Project Milestones: There are three milestones for your final project:

- Nov 4th: **Group formation**.
- Nov 16th: **Project proposal** submission and in-class presentation.
- Dec 14th: **Final project** submission and in-class presentation.

It is critical to note that no extensions will be given for any of these milestones for any reason. Projects submitted after the final due date will not be graded.

Project Proposal: Your group needs to present a project proposal (and submit the PDF of the presentation) with the following sections:

- What is the problem you are trying to solve with this application?
- What is the need for big data processing and what can be achieved thanks to large-scale parallel processing?
- Describe your model and/or data in detail: where does it come from, what does it mean, etc.
- Which tools and infrastructures are you planning to use to build the application?

The aim of the session is to review each proposal and we may suggest modifications if necessary. Our main concern is the amount of effort a given project will require; either too much or too little is unacceptable. You will have 5, and ONLY 5, minutes to briefly summarize your proposal followed by 5 minutes of discussion time. You have to prepare 2-3 slides for your proposal. We will enforce the 5-minute time limit. This presentation is a chance for you to get feedback, and to come up with ways around roadblocks you encounter. It is also a chance to ensure that your project is in the appropriate-amount-of-work range.

Project Software: Your final project can be implemented using any API or programming language you would like. Make your own repository on GitHub with a link to your project web page. Software with evaluation data sets, test cases should be available on the repository. Include a README that describes the code and application files, and how your program should be run. We will be grading these projects on a variety of platforms, so you must include detailed instructions on how to run or compile your code. If we cannot run your application from the instructions included with your submission, we will not be able to grade this portion of your project.

Project Web Site: An important piece of your final project is a public web site that describes all the great work you did for your project. The web site serves as the final project report, and needs to describe your complete project. You can use GitHub Pages, so you can easily refer to the software at the GitHub repository. You should assume the reader has no prior knowledge of your project and has not read your proposal. It should address the following ten topics:

1. Description of problem.
2. Description of the need for Big Data.
3. Description of the solution and comparison with existing work on the problem.
4. Description of your model and/or data in detail: where did it come from, how did you acquire it, what does it mean, etc.
5. Technical description of the parallel application, programming models, platform and infrastructure.
6. Links to the repository with source code, evaluation data sets and test cases.
7. Technical description of the software design, code baseline, dependencies, how to use the code, and system and environment needed to reproduce your tests.
8. Performance evaluation (speed-up, throughput, weak and strong scaling) and discussion about overheads and optimizations done.
9. Description of advanced features like models/platforms not explained in class, advanced functions of modules, techniques to mitigate overheads, challenging parallelization or implementation aspects...
10. Final discussion about goals achieved, improvements suggested, lessons learnt, future work, interesting insights… and any needed reference.

Your web page should include screenshots of your software that demonstrate how it functions.

Project Presentation: You will have 10, and ONLY 10, minutes to briefly present your project followed by 5 minutes of discussion time. You may prepare 4-5 slides for your summary, but we will enforce the 10-minute time limit. Focus the majority of your presentation on your main contributions rather than on technical details. What do you feel is the coolest part of your project? What insights did you gain? What is the single most important thing you would like to show the class? Upload the presentation to the GitHub repository.

Project Grading: Project will be graded on the depth of work undertaken, communication (web site and presentation) and participation following as criteria the ten topics described above in the Project Web Site section.

The final project grades are dependent on the following criteria:

- Attempted difficulty: Some projects are harder than others. For example, a project based off of one of the homework assignments is probably easier than a completely new application.
- Did you meet your major goals? The most important grading criteria is functionality: A working program will always garner the majority of available points; no credit will be given for non-working programs. A modest solution that works will be graded much more favorably than an ambitious "solution" that core dumps!