Project Requirement Document

Project Goals

- Create and evaluate your own algorithm(s).
- Solve a real-life problem or develop a real-life application.
- Implement research problems on Workflow Scheduling algorithms.

Project Details

A significant portion of your final course grade (20%) is the project component. For your project, you will work with your team members as it is a team based project. Each member of the team will be evaluated separately based on his or her contributions to the project. This evaluation will be determined largely from the feedback of the other team members. Although not ideal, you may also choose to work on your own if your schedule will not allow you to meet with other team members regularly. I would strongly discourage this option.

There are two tracks designed for the course project. In track 01, you can develop a course project that is to design a real-life application system that uses a new algorithm or a series of new algorithms. In track 02, you can get a feel of what computer science academic research is all about. You would be working on an academically challenging problem called the Workflow Scheduling Problem. There will be a total of 8 existing workflow scheduling algorithms provided to you with the corresponding research paper and a video demonstration of the paper. You could either:

1. implement atleast 3 of those algorithms and provide an experimental comparison to compare the performance of those algorithms.
2. implement your own workflow scheduling algorithm by getting an inspiration from the other algorithms and then compare your proposed algorithm with atleast two algorithms to show the performance advantage of your algorithm.

If your team happen to select track 02, then you get to interact with the instructor more and possibly open the door for a research manuscript submission in the future, as it is the instructor's research area.

In track 01, you have a significant amount of flexibility with what you can do with your project, so use it wisely! Is there something you wanted to create but never had time? Is there a project from other courses that you always wanted to extend? Juniors: is there an idea you have for your thesis that you want to try out first before committing to it? Seniors: did you want to explore a different aspect of your senior thesis? If the answer is ”yes” to any of these, and if you can connect your idea to algorithms, then you can do it as your project now! Of course, you cannot claim anything you have previously done as a contribution for this project, but you can certainly use your previous work, knowledge, and experience as a backbone for this project.

Hard Requirements

- Your project must be approved by me before you start working on it.
- If your team want to pursue track 01, you will follow the requirements below:

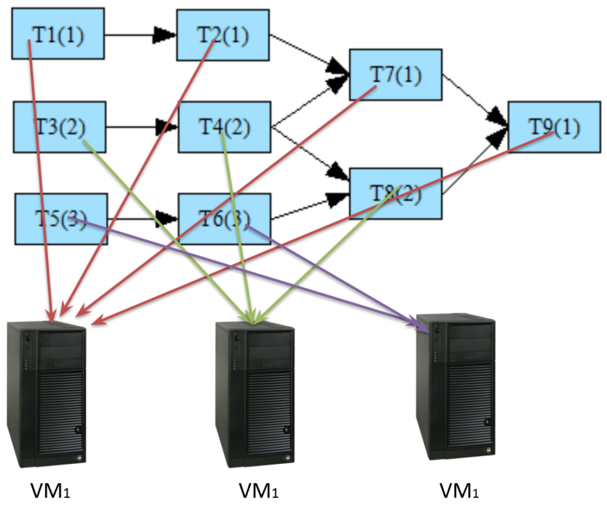
1. Your project must solve a real-life problem or develop a real-life application. You need to research the problem you select to get an idea of what has already been done. You must include references to existing work in your final report.
2. Here are some examples of projects that were done last year:
   - Improving room draw through analyzing numbers generated in previous years.
   - Improving room draw through a points system based on grades and extracurricular activities.
   - An automated, intelligent Bitcoin buyer/seller.
   - Efficient collision detection using quadtrees.
3. There are many references to all of these problems, and I'm sure as you think of your own project, you will find resources for them as well. If you're completely stumped in coming up with a project idea, you can certainly talk to me and we will set up a brainstorming session. Be creative and choose something that is interesting to you!
4. Your project must involve designing new algorithm(s). You can approach this requirement by either:
   - Creating a brand new algorithm and applying it to some problem. You must do some research to make sure your algorithm has not been proposed before.
   - Designing a significant extension to an algorithm and applying it to some problem.
   - Most likely, you will need to develop or extend a series of several algorithms.

   In any of these cases, you must provide a discussion (or proof) on the correctness of your algorithm. This discussion can include small examples and references to existing work (especially if you are extending existing algorithms.
5. You must provide a running time analysis of the algorithm.
6. Your project must have a signficiant implementation part where you will develop program(s) for your algorithm(s) for your chosen problem. You may write your code from scratch, or reuse and extend some existing code. Obviously, anything you use that is not yours must be documented. You may program in any programming language that you like.
7. Your project must be extensive enough to qualify as a project (think of work for at least 5 one-week lab assignments), but not too extensive so that you cannot finish it in the reminder of the semester (two months).

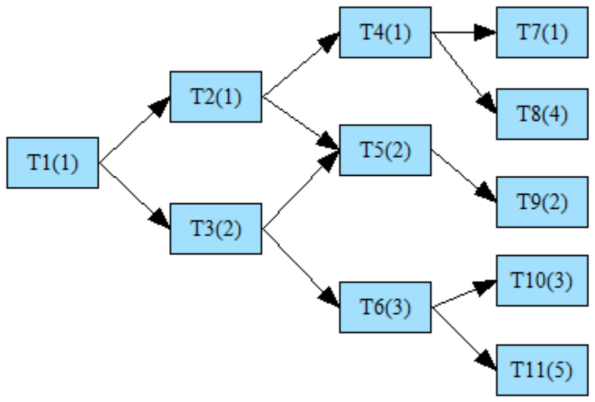
- If you want to pursue track 02, you will follow the requirements below:

  1. Basic Outline of the Problem:
     Workflows are represented using a directed acyclic graph (DAG). Each node in the graph represent a computation task and each edge in the graph represent the data flow from one task to another. From a genralized perspective, workflow executed on one machine does not require optimization strategy from a scalability perspective. However, if the workflow is executed in multiple machines then there is need to optimize the workflow from scalability perspective. There are usually two QOS (Quality of Service) constraints attached to the workflow optimization problem, which are Deadline and Budget.
     In order to develop your project in track 02, only a theoretical knowledge of the Cloud is required. You do not need any practical knowledge of how to configure and setup the Cloud. A Cloud is a framework that allows you to rent many machines. Each machine has a cost associated with it based on the time you rent the machine, usually charged on a hourly rate. The machines in the Cloud are either homogeneous or heterogeneous. Homogeneous Cloud will rent machines with similar configurations and thereby all machines have the same dollar amount for hourly rental rate. Heterogeneous Cloud will rent machines with different configurations and thereby all machines have different dollar amount for hourly rental rate. Many of existing scheduling algorithms work with Homogeneous Cloud.
     What is the Workflow Scheduling problem? The problem is widely classified as an optimization problem. Where to schedule a task in the workflow? Which machine in the Cloud. When to schedule the task? These are important questions that drives the scheduling problem. On one side, given a budget as a dollar amount, can we minimize the execution time incurred by the workflow. On the other side, given a deadline as a unit of time, can we minimize the execution cost incurred by the workflow. Both these are important questions that drives the optimization needs behind the scheduling problem.

2. There will be a total of 8 existing scheduling algorithms provided to you. The scheduling algorithms are HBCS, LOSS1, BHEFT, DBCS, DCA, GA, WRPSand IC-PCP. The research paper and a video demonstration of the scheduling algorithms are provided to you in the link below:
   `www.cs.allegheny.edu/sites/amohan/teaching/CMPSC250/track02.html`
3. You could do either of these two options:
   – Implement atleast 3 of those algorithms in Java programming language and provide an experimental comparison to compare the performance of those algorithms. Reading the paper and understanding how the algorithm works is a key component for making progress in your project.
   – Design your own workflow scheduling algorithm by getting an inspiration from the other algorithms and then implement your algorithm in Java programming language. Compare your proposed algorithm with atleast two algorithms to show the performance advantage of your algorithm. You can brainstorm with the Instructor to discuss regarding new ideas.
4. A simple example of the workflow scheduling is demonstrated using the picture below:



(a) 1 Machine.



(b) 5 Machines.

Figure 1: Workflow Schedule.

Timeline and Deliverables

1. Proposal – 1-2 pages, Deadline March 12 (EOD)

   Develop an idea for your project. Write a 1-2 page technical description of what you propose to do for your project and submit a PDF copy of your proposal to me via Email on March 12th by end of the day. Your proposal does not need to be very detailed at this point. It should describe which track you are going to pursue, what you want to do (the real problem you will tackle and the type of algorithm(s) you will develop), and at least a couple of references to show me that you have done some research about the problem. If you pursue track 02, what exactly would you like to do for your course project? But you do not need to include any specifications on how you will solve your proposed problem. That is, you do not need to include any details about the algorithm(s) you will develop.

   Meet with me during the lab period on Feb 23rd, March 02nd, and March 09th, or schedule an appointment during my office hours during the next few weeks to discuss your proposal and to receive an approval or a modification to your project proposal.

2. Progress Report 1 – 2-3 pages, Deadline April 02nd

   By this point, you should have designed a model for your system, or a framework for solving the proposed problem; developed your algorithm(s), written them in pseudocode; and started analyzing the complexity. Also, you should have started programming.

   Include everything that you have done so far in your progress report, even if it is incomplete.

3. Progress Report 2 – 3-4 pages, Deadline April 18

   By this point, you should have made a tremendous amount of progress towards implementing the algorithm(s). Were there any unexpected challenges? Did you have to change your initial model/framework or the pseudocode? You should have also finished your mathematical analysis of the complexity by now.

   Include everything you have done so far in your progress report, even if it is incomplete. No need to include the actual code (unless you want my help with it), just describe what progress you have made with it.

4. Presentation – Deadline April 25 - 30

   By the presentation session, you should have finished implementation, run some preliminary experiements, and done some basic analysis. In the presentation, you should describe the motivation, problem definition, challenges, approaches, and results and analysis. Use diagrams and a few bullet points rather than long sentences and equations. The goal of the presentation is to convey the important high-level ideas and give intuition rather than be a formal specification of everything you did. Prepare for a $\sim$15 to $\sim$20 minute presentation. Design at least 6 to 10 slides, including a slide with the title of your project and group members.

   Every member of the group needs to contribute to the presentation talk. Also, you need to show a demo at the end of the presentation of your system.

5. Final Report – 7-10 pages, Deadline May 2

   Incorporate any feedback from the progress reports and the presentation session. Your final report should be clear and well written, which includes no typos or grammatical errors. The writing will be graded more harshly here. Your report should be written in a professional and technical manner. Your report should include the following:

   - The motivation for your project. Why is the problem you decided to solve important or useful?
   - Background for the proposed problem. What have others done for it already? Include references.
   - Detailed algorithm description and the complexity analysis. Include pseudocode, diagrams, and examples if appropriate. If you are extending existing algorithm(s), briefly describe previous work and include references to it.
   - Description of your results. Make graphs, tables, and anything else that can help me understand your results.

- Conclusion. Give a short overview of your project and its results. Describe what you learned, what were the biggest challenges and the biggest rewards.
- If you are in track 02, you want to make sure you follow the above guidelines listed in track 02 requirements carefully and also discuss with the Instructor for more clarifications on the requriements for the Final report.

Submission Details

For each deliverable, you need to submit a PDF with your report (or presentation slides). For your final report, you need to submit any supplementary material (code, data, a README file documenting what everything is, and how to run your program) to my Email.

- Proposal – 10 points
- First progress report – 15 points
- Second progress report – 15 points
- Presentation – 25 points
- Final report and implementation – 35 points

Please remember that all files that you submit should be your teams work, though you are welcome to discuss high-level topics and algorithms with classmates.