The University of Melbourne
School of Computing and Information Systems
COMP90020 Distributed Algorithms

Term Project, 2022

Released: 28 February. Deadlines listed below.

Objectives

The objective of this project is to enable students to gain a deep and practical understanding of a distributed algorithm and its broader context. Specifically, this assessment will allow students to meet the following learning objectives:

- Describe a given problem domain in the field of distributed algorithms.
- Explain and analyze an algorithm in depth, including its assumptions and performance guarantees.
- Survey existing algorithms within a given problem domain.
- Contrast a set of algorithms in the same problem domain.
- Implement a distributed algorithm.
- Illustrate the applicability of an algorithm in a real distributed system.

Additionally, this project aims to give students the opportunity to practice cooperative research, writing, and software development.

Background and Context

This assignment is the only one for this subject. It is a semester long project intended to be completed in **teams of 3 to 4** students. The overall task is to select a distributed algorithm from the set of topics outlined below. Each team will implement the algorithm in a real application and submit a report surveying literature relevant to the selected topic and algorithm. Teams will also be required to present a summary of their project early in the semester (i.e., topic, algorithm, initial survey, and distributed application); the main purpose of this deliverable is to give teams the opportunity to receive formative feedback from the teaching team as well as from peers.

The project is worth **45%** of the final mark split across the following deliverables:

- A 5-minute presentation and a 1-page summary of your project proposal - **10%**
- A demo showcasing the implementation of your chosen algorithm in a distributed application - **15%**
- A report including a detailed discussion of the algorithm, a survey of related papers, and a comparative analysis - **20%**

The stages of your project can be summarized as: choosing a topic and algorithm, performing a background survey to select related papers, reading selected papers, drafting your report and implementation, finalizing the implementation and preparing the application demo, finishing your report.

You are welcome to seek help from the lecturers and tutors during your progress about any problems that you may face.

Key Dates

- **Friday 18 March at 23:00 (Week 3)**. Submit names of all your team members.
- **Friday 8 April at 23:00 (Week 6)**. Submit 5-minute video presentation and 1-page summary on chosen topic.
- **Sunday 8 May at 23:00 (Before Week 10)**. Source code submission.
- **Monday 9 / Wednesday 11 / Friday 15 May (Week 10)**. Live demonstration during tutorial and lecture times.
- **Sunday 22 May at 23:00 (Week 11)**. Submit final report.

Topics

Select **one** topic from below to research.

- Logical time
- Snapshots
- Failure detection
- Mutual exclusion
- Multicast
- Election
- Consensus
- Distributed transactions
- Broadcast/Convergecast*
- Replica consistency*
- Stable property detection (e.g., deadlock and termination detection)*
- Non-stable property detection (e.g., distributed debugging)*

Topics marked with an * are not covered in the subject, you are free to explore them in your project if interested. If your team is interested on working in a topic that is not part of the given list, please get in touch with the lecturers to discuss its suitability to the project.

Students should contemplate picking a topic early in the semester. A quick search on Google and a quick look at the textbook will give an idea about the topic. At the start of the semester, students are not expected to know the details of their topics, but they should have an interest in studying it.

The deadline to pick a topic is Friday 18 of March. Topics can overlap between teams as long as the reports and implementations are unique. Once you have finalized a topic, you must submit your team's name, topic, student IDs, and emails to your tutor (details to be provided). This also needs to be finalized by Friday 18 of March.

A list of distributed algorithms (not an exhaustive list) is on Wikipedia.

Project Proposal

The first deliverable of your project is a 5-minute video presentation and a 1-page summary on your chosen topic, algorithm, and distributed application. This will allow the teaching team and your peers to give you formative feedback early in the semester.

Your presentation and written summary should include the following:

- Brief introduction to the topic.
- Brief description of the selected algorithm.
- Rationale for choosing the algorithm.
- Summary of background survey done so far.
- Description of the distributed application.
- Explanation on how the algorithm benefits or supports the distributed application.

Implementation and Demonstration

The implementation is an application of your selected algorithm in a real world application. Implementation details are the choice of the students. The purpose of the demo is to demonstrate that you have implemented the chosen algorithm in a realistic setting. The setting could be of the nature of a multi-player computer game where different users can provide independent input. However, you are free to choose whatever you like for your application scenario. For example, you could be demonstrating things such as deadlock resolution or event ordering if you chose one of these topics.

The graphics of the application is not the focus of the implementation, although good graphics always help with presentations. The main focus of the implementation is the experiment with the chosen algorithm. The code used for the algorithm must be original in nature. The other parts of the application are up to your time to decide, however, proper attribution must be given where code and/or libraries are used that are not original.

Other examples of possible applications include: game server with multiple clients, logging server, state replication machine for files, auction servers (exchange), and distributed ledgers (blockchain).

Demonstration

The implementation of the algorithm and the distributed application will be demonstrated by teams during the lectures and tutorials in Week 10. During these sessions, teams will showcase the functionality of their applications and will explain the core aspects of the selected distributed algorithm.

Demos can contain a few slides as an introduction (not more than 3) prepared using an electronic presentation environment such as PowerPoint. Then your live demo will follow. You are expected to show a working system with the selected algorithm in it. You are expected to highlight where and when the algorithms play a role. It is also preferable if you can show one case in which not using the algorithm will cause a problem. Demos should last no longer than 20 minutes.

The project (source code) should also be submitted as a zip file that contains the demo code via the LMS. Code submissions are expected to have a readme.txt file explaining the details of the code structure.

Report

The report will be on a topic for a set of algorithms in the area of distributed algorithms. You will need to give a survey of papers from this topic and then give comparisons of different approaches presented in these papers. Some comparisons already exist for various topics and it is acceptable to use these existing surveys for your report, but the report has to be in your own words and is expected to be a critique from your point of view and more up-to-date than these existing surveys. In general, reports cannot be a simple re-iteration and a list of existing papers but have to be organized in an innovative manner that helps the reader understand the different categories of approaches in the topic. Reports should also contain possible application areas of the algorithm(s) as well as the details of the implementation of the chosen algorithm.

Structure

The report must be $\pm 10\%$ of **4000 words** in length. Excluding references, bibliography, appendices and figures. The following is the advised structure, note that it is not a requirement to adhere to.

- Title, Author information, Abstract, Keywords
- Introduction and background
- Survey of related and direct work
- Critical analysis of the algorithms reviewed
- Future directions and applications
- Discussion of chosen algorithm, implementation details and formal description
- Conclusions

Mark Breakdown

Project proposal (10%)

- 10/100: Presentation - technique and materials
- 30/100: Presentation - structure, flow, and contents
- 10/100: Written summary - topic description
- 20/100: Written summary - algorithm description
- 10/100: Written summary - background survey
- 20/100: Written summary - application description

Report (20%)

- 10/100: Description of problem domain
- 20/100: Synthesis and presentation of background survey
- 20/100: Comparative analysis
- 20/100: Discussion of chosen algorithm and its implementation
- 10/100: Literature selection
- 10/100: Structure and audience
- 10/100: Referencing

Algorithm and Application Implementation (15%)

- 20/100: Difficulty of algorithm
- 20/100: Quality of demonstrated application
- 40/100: Implementation of demonstrated algorithm
- 20/100: Presentation (engagement, creativity, speech of demonstration)