# **TASK MANAGEMET FOR DISTRIBUTED SYSTEMS**

## **INTRODUCTION**

This project is about simulating real world cloud computing. The requirements include design and implementation of a smart, and relatively cost-effective job scheduler. The job scheduler must actively take into account the available resources and the run time of each job. Specifically, the CPU power, varying levels of memory and storage, and the run cost of the available resources must be considered.

## **AIMS**

To develop an effective task management system with the capability of performing exceptional job scheduling and resource allocation.

## **BACKGROUND**

This project is based upon a simulation of cloud computing, which involves using a network of remote servers to store, manage, and process data. It is an on-demand service that is scalable, cost-effective, and reliable.

By the end of this project, we will have created a job scheduler for a distributed system. Distributed Systems are networks that consist of connected autonomous computers. The system ensures efficiency by distributing tasks from a larger project to single computers within the network for completion. Hence, by creating a job scheduler we will have a complete and effective distributed system running.

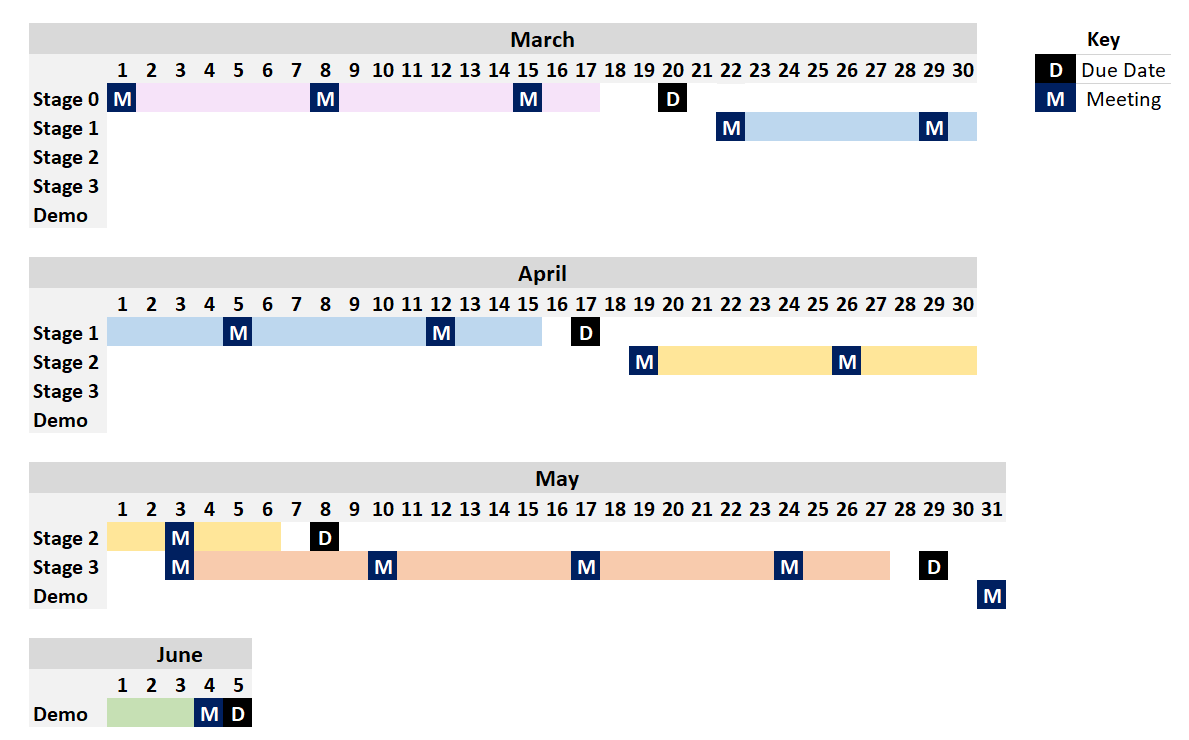
Our job scheduler will likely make use of the client-server communication model, as it is a structure that manages the workload between providers and clients. Where, upon request the server processes and stores data for the client.

## **PROJECT PLAN**

* **Group members and their main roles:**

Our group members are Matthew Long (45154929) and Siobhan O’Malley (45167648). Matthew’s solo role will be the Project Manager and Siobhan’s solo role will be Program Testing. Matthew and Siobhan will equally contribute to the Algorithm Design and Programming in this project.

* **Schedule:**

Our aim is to have each stage of the project fully completed two days before the due date, to ensure each member time to review and discuss any concerns with the group. Below is a Gantt Chart that provides a rough outline of our timeline for the project. We also plan on updating our chart to be more extensive as details of each stage are released.

* **Work arrangements (meetings and other communication channels)**

We will physically meet every Sunday to check up on progression in our respective positions and to arrange any further meetings required for collaboration on sections of the project. In these meetings we will discuss any issues faced to ensure the problems are solved effectively. We will also communicate via phone calls and text messages for anything that arises more urgently than waiting for the Sunday meeting, or for minor things that we may be able to easily help each other with.

* **Data management including source code and documentation (Bitbucket or Github; create one of these and provide the link in this document)**

We have chosen to use Github as our data management tool. We will both be collaborators on the project at the following link: <https://github.com/mattlong163/Comp3100/projects/1>

* **Choice of programming language and the justification**

Our choice of programming language for this project is Java. We are both confident in our knowledge and abilities in Java and we believe it has a user-friendly interface that will assist us in ease of development.

## **REFERENCES**

Client-Server Model - GeeksforGeeks (2020) *GeeksforGeeks*, Available from: https://www.geeksforgeeks.org/client-server-model/ (accessed 12 March 2020).

What is a Distributed System? - Definition from Techopedia (2020) *Techopedia.com*, Available from: https://www.techopedia.com/definition/18909/distributed-system (accessed 12 March 2020).

What is cloud computing? A beginner’s guide | Microsoft Azure (2020) *Azure.microsoft.com*, Available from: https://azure.microsoft.com/en-au/overview/what-is-cloud-computing/#benefits (accessed 12 March 2020).

What is the Client-Server Model? - Definition from Techopedia (2020) *Techopedia.com*, Available from: https://www.techopedia.com/definition/18321/client-server-model (accessed 12 March 2020).