Cloud Job Scheduler for Distributed Systems

Lachlan Tripolone (45229287)

# Introduction

Distributed systems refers to a collection of independent machines that communicate messages towards each other in order to achieve common goals that are set by the client [1]. The main components of distributed systems are the servers. These servers are often connected together and work together to achieve the same goal. Individual servers can vary in many aspects such as size and processing power and in turn, allow for better control of resources when jobs are requested by the client. This creates a sense of ambiguity for the client-side on the number and server types that there are as the servers work concurrently perform jobs. Therefore, the goal of this project is to simulate a distributed system with a client and an unknown server as specified by the client below.



The project goal is to create a client-side simulator for the ds-sim server implementation with job scheduling algorithm in order to simulate the exchange and processes of distributed systems. In particular, the client-side simulator must implement a scheduling algorithm optimising one or more of specific performance metrics, those being the minimisation of average turnaround time, the maximisation of average resource utilisation and the minimisation of total server rental cost.

# Problem Definition

The desired performance metrics of the minimisation of average turnaround time (hereafter TT), the maximisation of average resource utilisation (hereafter RU) and the minimisation of total server rental cost (hereafter RC) are all conflicting metrics in that the optimization of one can, sometimes drastically, negatively impact another. This issue means that sacrifices is some metric must be made, and as such the algorithm must be designed with a specific metric in mind while aiming to minimise the negative impact it might have on others.

This scheduling algorithm aims to minimize the total rental cost through smarter server utilisation at minimal expense to turnaround time. Such an algorithm would be highly desirable for smaller businesses needing to run non-mission-critical jobs on a tighter budget as well as other non-mission-critical jobs where time is not a major constraint.

# Algorithm Description

The algorithm prioritizes scheduling to the smallest capable server with the least amount of jobs waiting in queue. This optimizes resource utilisation as idle servers are prioritized over all others and reduces costs through choosing smaller servers where possible.

# Implementation Detail

# Evaluation

# References

[1] Y. K. K. J. K. Young Choon Lee, "ds-sim: A Distributed Systems Simulator User Guide," Macquarie University, 18 02 2021. [Online]. Available: <https://github.com/distsys-MQ/ds-sim/blob/master/docs/ds-sim_user-guide.pdf>. [Accessed 17 April 2021].