# Internal Virtual Job Scheduling System

## Developers

Joel Duff – 44905866

Ngoc Lan Tran – 44480636

Warren De Silva – 44912838

## Introduction

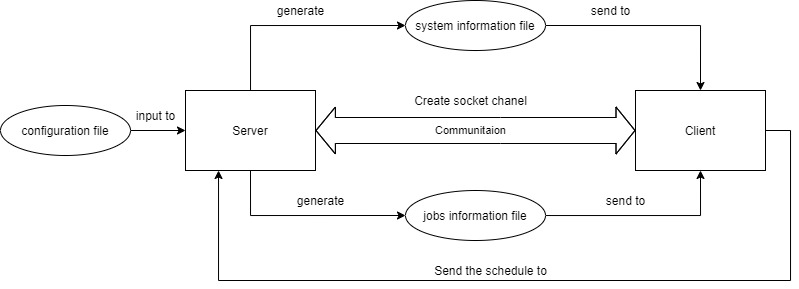
Distributed system in computing is diverse about size and types. It can be a single computer or a large network between multiple computes as data centres in the world. There are two main components to build the distributed system is server and client. Servers are connected from many single systems to implement jobs to clients. However, this project focuses on the client side when designing and developing a working model of client is job schedule.

In a system, there are many jobs from multiple clients at same time; and each job have different requirements from server to solve this job such as memory, CPU cores and disk space. This can lead to stuff at bottle neck when there are several numbers of jobs are sent at the same time. Thus, the schedule is suitable technique to solve this problem.

This project is developing depend on a discrete event simulator (ds-sim) to create the job schedule system which connects to servers, gains jobs and schedules them. This also dispatch the jobs to the largest server which is identified by the number of CPU cores.

## System Overview

The job schedule system includes two sides are server and client and base on ds-sim which is distributed systems simulator. Both sides initialise a communication by handshake and implement a loop to finish jobs which is known as discrete event simulation. After handshake between client and server to authenticate the communication’s information, server creates a list of job and transfers to client with some options i.e., JOBN, JCPL, RESF; after client finishes its job, server will run all jobs depend on the priority in client’s schedule. In client side, client receives the messages and job list from server and analyse them. In schedule policy, client will send all jobs to the largest server at the current running time, it is changing if there are amount of number of jobs are running at one server.



## Design

D

## Implementation

C

## References