Project title

**Basic scheduler with inbuilt job dispatcher within a client side simulator**

Group members

Jonathan Cawood (45887454)

Carlos Larino ()

Introduction (1/2 page)

The aim of this project is to develop a job scheduler for a distrusted system .With specific relation to Stage 1 requiring the design and implementation of a “vanilla’ version of a client-side simulator that includes a basic scheduling function with a simple job dispatcher. The simple job dispatcher upon successful implementation will send all jobs to the first one of largest server type. The largest server type will need to be determined in order for this process to work. The role of a job dispatcher within a distributed system is key for ensuring the efficient use of computer systems including distributed systems which will be highlighted within this project.

System overview (1/2 page)

**SERVER**

**Client**

The client starts by sending “HELO” to the server, then waits for the response of “OK” from the server. The client then sends “AUTH NAME” to the server which then replies with “OK” after printing out a welcome message and writing system info. The client then sends “REDY” to the server which then sends (STEP 6) need to add the specific parts about the job dispatcher and scheduler.

Design (1 page)

Implementation (2 pages)

Jonathan Cawood worked on the implementation of initial communication to the server specifically dealing with the implementation of “HELO”, “AUTH”, “REDY”, “GETS All” requests and replies. Whilst also implementing

An Array List was used to store all the information of each server using the created Server class. The Server class was created to be able to import server properties into the class and be able to locate and keep track of the imputed servers, such as the ID, State, Start Time, Cores, Memory, Disk. Therefore, enabling the use of a for loop to loop through and determine which server had the highest amounts of Cores with the use of an if statement to compare the core count with the highest core count so far, allowing for the highest to be determined.

References