Cloud Job Scheduler & Resource allocator for distributed systems

Hayden Chan 44817002

Dac Quang Nguyen 45506698

Cuongg Nguyen 44491158

**Introduction**

**System Overview**

**Design**

The use of the string arrOfSTR = serverReply which actually finds the biggest core and compares it with the current core with the best core at index 4 and the name at index 0.

It then writes ok to the server and the server will reply to the ok with a “.” which is in the size of 1 byte.

**Implementation**

Firstly, we have implemented all the required libraries for the connection to be established. Next, we declared the initial strings, the first function is a read message function which reads the messages from the server side. Then we have included a data input and data output steam for reading and writing to the console which displays the text to the user. Next we send Helo message to the server so that we can start the communication, the client will then read the reply from the server which is 2 bytes. The client sends "REDY" message to the server to say that the client is ready to schedule jobs (if any) from the server. Then with the use of a while loop it checks if there are any existing jobs left that has not been completed. If there are no jobs in the loop then it will be ready for the next job, otherwise it will detect a job and get all the data from the job.

[**https://github.com/DacNguyen234/Comp3100\_project**](https://github.com/DacNguyen234/Comp3100_project)