**Cloud Job Scheduler**

Kelly Flett (45350043)

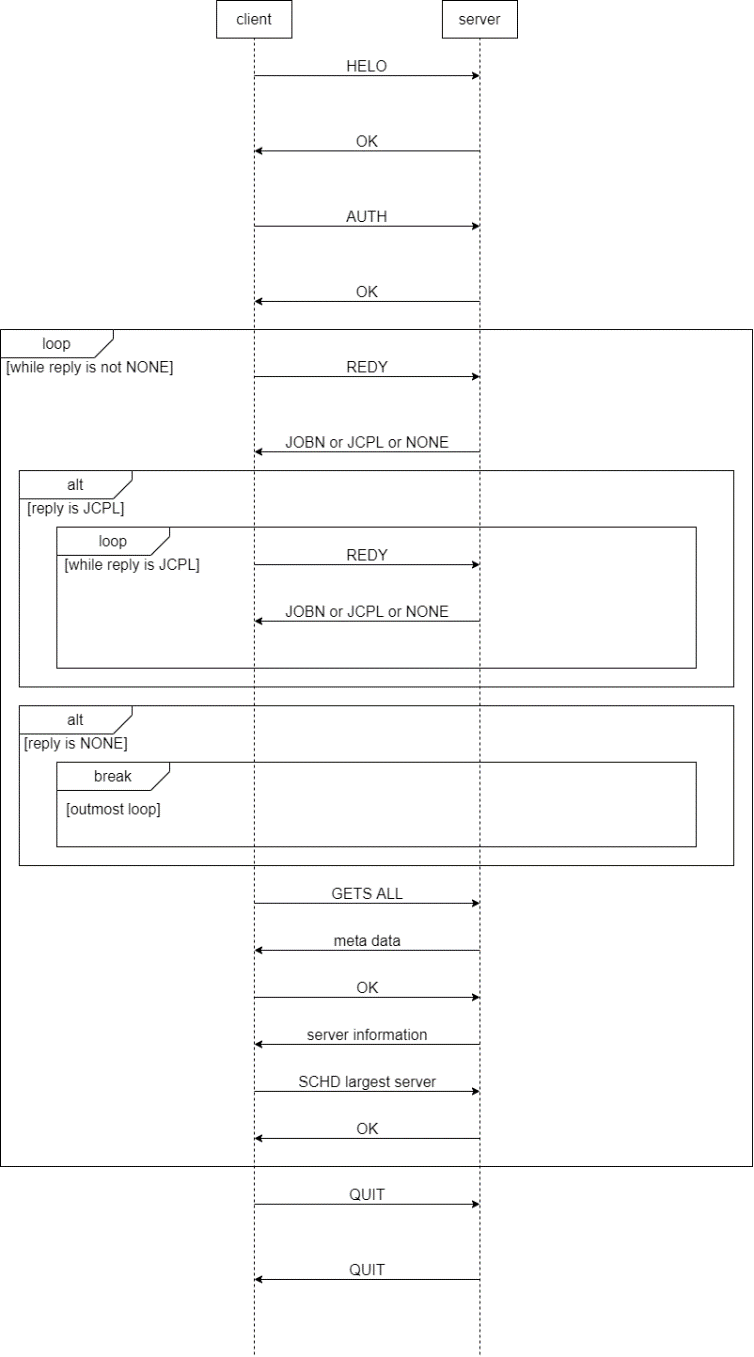
Scott Lin (45985995)

Jaime Sun (45662398)

**Introduction:**

The goal of this project is to develop a client-side simulator that acts as a simple job dispatcher. For this first stage we were tasked with implementing the ds-simulation protocol such that it successfully connects to the server-side simulator and will receive and schedule jobs. We also had to create a simple job dispatcher that sends all jobs to the first entry of largest server type.

**System Overview:**

****

Detailed steps of the protocol

1. C(lient) sends HELO to S(erver).
2. S replies with OK.
3. C sends AUTH with authentication information to S.
4. S replies with OK.
5. C sends REDY.
6. S sends one of the following:

* JOBN: a normal job for scheduling for the first time.
* JCPL: information on the latest job completion.
* NONE: when there are no more jobs to schedule.

1. C takes one of the following actions upon receiving a message from S:

* Go to step 5 for JCPL.
* Go to step 16 for NONE.
* Go to step 8 for JOBN.

1. C indexes current job.
2. C sends GETS ALL.
3. S sends DATA (the preparation message that indicates how many bytes to expect) for GETS ALL.
4. C sends OK for DATA.
5. S sends information of all servers.
6. C indexes all servers and finds the one with the largest core count.
7. C sends SCHD for current job to largest server.
8. S replies with OK.
9. C sends QUIT to S.
10. S sends QUIT to C and exits.
11. C exits.

**Design:**

The client consists of 3 classes:

* The allToLargest class
* The Servers class
* The Jobs class

The Servers and Jobs classes contain attributes in the form of Java fields for simpler referencing in the allToLargest class. allToLargest is where the exchange between client and server happens as well as the largest server algorithm. Our design philosophy is structured and comprehensive. We first make a working draft of the program then work to make it more efficient and modular until we have optimized our code as much as possible. We decided to utilize Java as our coding language as we are the more comfortable and experienced with it than other languages.

An important consideration and constraint of the project is that it must be run in Ubuntu, which we are all running on virtual machines on our devices. Utilizing Github allows us to easily store and update our project while keeping records of each team members contributions.

**Implementation:**

Current job is stored in an array with the 0-index housing the reply from the server. Job attributes are stored in the 1-5 indices. Server information is stored in an array list, with each node housing an array with indices 0-6 each containing an attribute of 1 server (each node is basically a server).

**References:**

Git Repository: https://github.com/JSFun9888/job-scheduler-assignment