README.md

CS3103 Assignment 2 - Job Scheduling (Language - Python 3)

Done by: Wong Jun Long (A0201889W) and Yu Bowei (A0205496Y)

Instructions to run the job scheduler

1. Use a Linux environment with Python 3 installed (e.g. xcne1.comp.nus.edu.sg)

2. Start the server-client simulator on a terminal using:

```
./server_client -port <port number> -prob <knowledge probability>
```

3. Launch the job scheduler by running on another terminal:

```
python3 jobScheduler.py -port <port number>
```

Ensure that the port numbers for both programs are the same.

4. To terminate the programs, press CTRL-C for both the server-client simulator and job scheduler within the CLI.

Explanation of algorithm

Our job scheduler uses the weighted least response time algorithm.

The formula used for the weighted least response time algorithm is:

Weighted Response Time = (Time to First Byte (TTFB) * Number of Active Connections) * (10000 / Weight of Server)

Start of Algorithm

Initially, each server is assigned a weighted response time of 100000. Upon receiving jobs, the job scheduler sends all servers a job in the sequence the servers are presented in the `servernames` list to get an initial gauge of their response times.

For each server, the job scheduler keeps track of the server response times, number of active connections, and the weights of the servers.

Time to First Byte (TTFB)

Firstly, after a job is completed, the job scheduler calculates the `response time` of the server. This is calculated using the formula `time result is received by job scheduler - time server is sent a job`.

Number of Active Connections

Secondly, in our algorithm, the number of active connections that any server can have is limited to 1. If all of the servers are busy, the job will be stored in a queue. Jobs in the queue will be assigned to servers once they are done with their current job.

This is done to prevent cases where servers of low computational capacity are assigned too many jobs which will worsen the 95th percentile performance.

Weight of Server

Thirdly, each server is assigned a weight according to their weighted response time ranking relative to other servers.

For example, assuming that we have 10 servers, we will assign the server with the lowest weighted response time with a server weight of 1. On the other hand, we will assign the server with the highest weighted response time with a server weight of 10.

This is done to weigh the performances of the servers relative to one another.

Weighted Response Time

The weighted response time of the server that completed the job is then computed using `response time * active connections * (100000 / (total number of servers - current index position of server in server list))`.

After the computation of the weighted response time, we will sort the servers according to their weighted response timings to update the positional rankings of the servers.

Subsequently, for each new job received by the job scheduler, the job scheduler will assign the job to the server with the lowest weighted response time.

Optimization of 50th percentile given full knowledge of job sizes

To optimize the 50th percentile performance, the queue for pending jobs is sorted in ascending order of job sizes.

This is done so that a larger quantity of jobs are completed in a shorter period of time to attain a lower 50th percentile timing.