WRITEUP

I tested my code in pieces. I first worked on healthcheck. I use a healthcheck thread to perform healthchecks on given servers so I first tested whether my program could probe servers for healthcheck every X seconds. I used two httpservers listening on ports 8080 and 8081 and tested if healthcheck was performed every X seconds. Then, I factored in the signaling as well for every R requests.

I checked if my loadbalancing was working by sending multiple requests and printing out the which server handled the request. I also printed the healthcheck data of the server to check if the optimal server was being chosen for the requests.

1. A more realistic implementation was not used for this assignment due to the limitations of the requests. Loadbalancer has no way of knowing the health of the server other than through the response to the healthcheck. If httpservers had another resource capable of transmitting the load, power usage, failed requests, CPU usage, and so on, perhaps loadbalancer would be more realistic. However, factoring those into the calculation of choosing a best server would be much harder.

2. The loadbalancer would perhaps send different types of requests to different servers by reading the type of request received. Processing the data, however, would use more resources and could cost time for other clients to be serviced.

3. The httpserver handles all but one of the connections. The nc connection times out, and therefore, httpserver does most of the work. The execution is therefore slower in the second scenario (nc and httpserver) when compared to two working httpservers.