Assignment 3 Writeup

I followed the same strategy that I used for testing on all the other assignments for this one as well. I decided to use a ton of unit testing because if I know a certain part of my code functions as it should, I wouldn't need to worry about it when I'm progressing further into the program. I first tested my multithreading to make sure that my load balancer was able to handle accepting multiple client connections. I would test it by running test scripts that would send a bunch of curl requests (PUT, GET, HEAD) to my load balancer. I used print statements after every enqueue into the queue to make sure that everything was properly stored inside of the queue and would properly dequeue as well. After that, I tested my health check function to make sure that I was able to send health checks at least once to all the servers saved in my array of servers. I checked to see if the servers stored in the array were having their information updated by printing out the fields after each health check. This allowed me to ensure that the values were being updated correctly. Making sure these two worked properly was imperative to the success of my load balancer.

1. For this assignment, your load balancer distributed load based on the number of requests the servers had already serviced, and how many failed. A more realistic implementation would consider performance attributes from the machine running the server. Why was this not used for this assignment?
    a. That method wasn't used for this assignment because the performance of a machine may be inconsistent (different hardware, how many processes are being run at the same time, etc). In a classroom environment, it'd be harder to implement something like this because of variations between student/personal hardware and school hardware.

2. This load balancer does no processing of the client request. What improvements could you achieve by removing that restriction? What would be the cost of those improvements?
    a. The load balancer doesn't process the client request because it forwards the request to an http server that is built to handle those kinds of requests. By allowing the load balancer to process the client requests, it would essentially be no different than an http server. I guess it would be faster with handling single requests simply because it does not need to send it to other http servers. The cost of these improvements would result in slower efficiency of handling many clients at once because then the load balancer would no longer send requests to other http servers. This means that less clients can be serviced at once which results in a slow down in efficiency.

3. Repeat the same experiment, but substitute one of the instances of http server for nc, which will not respond to requests but will accept connections. Is there any difference in performance? What do you observe?
    a. I’m going to guess that there is a difference in performance, and that it probably slows down because we aren’t running 2 http servers. I can’t test for this because I did not complete my assignment 3 completely, so my load balancer isn’t fully functioning. But, I do think that running less http servers would result in a slow down because there is less computing power. Only one http server would be handling connections and actually sending a response back, while the nc would only accept requests. I’m guessing there would be around a 2x slowdown with the loss of one server.