Assignment 3: Load Balancer Writeup
Joshua Cheung
Cruz id: johcheun

Question 1: For this assignment, your load balancer distributed load based on 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?

Answer: The more realistic implementation would be more realistic because it makes more sense for the machine running the server to have the information. Due to the fact that we are making our own proxy servers, we have to manually tell it how many request errors and request entries there are. If we had access to the server's source code then we would be able to get this information dynamically and thus be able to get it from the machine, but since we do not have it we must do it manually.

Question 2: This load balancer does no processing of the client request. What improvements could you achieve by removing that restriction? What would be cost of those improvements?

Answer: There would be an improvement by allowing the load balancer to also process client requests because we could optimize the requests in a way where they can run concurrently and more efficiently without any delay. The cost of this on the load balancer would be that it needs some more time to figure out which requests take longer, and then distribute the requests evenly according the how long the request will take.

Question 3: Scenario
After starting eight separate instances of the client, by using time() took around 7.46 seconds. After substituting one of the instances of the http server for nc, there is an improvement in performance by about 10% percent because one of the threads is not responsible for handling requests, which takes up less resources

Testing:

Testing queueing and enqueueing:
To test my Queue structure, I would run several curl commands and print out the structure of my queue to make sure that they were being enqueued correctly in the correct order. After this, after processing the requests I would dequeue the request, and then print our the structure of the queue to make sure that the integrity of the structure was being held in place.

Testing dispatching/findingLeastLoadedServer:
To test my dispatch function in making sure that the function was distributing the requests evenly to the servers, I would print out which servers are handling requests while at the same time printing the array which tells me which servers has the most or least entries.

Testing Multithreading:

To test the multithreading of requests I would concurrently run curl commands in a shell script using the ‘&’ notation. If I saw that the client closed correctly and the responses were correct, then I would know that it is working correctly.