# Writeup

**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?**

Checking the performance attributes was not feasible for this implementation as there was no means to gauge each server's performance. If the number of threads a server has was specified in the health check, it might have been possible. Measuring based on performance attributes would likely increase throughput and efficiency of the load balancer as the best performance server can be selected at a given time.

**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?**

If the load balancer were able to process a client's request, it could prioritize certain requests. For example, HEAD requests are likely processed faster than GET and PUT requests. Also, parsing the request to check if it were malformed might be faster to check on the load balancer than it would be to send and wait to receive a response from a server. Implementing this feature would increase the complexity of the program and thus could lead to more points of failure for the program.

**Repeat the same experiment, but substitute one of the instances of httpserver for nc, which will not respond to requests but will accept connections. Is there any difference in performance? What do you observe?**

Experiment 1: Time with 2 httpservers: 3.8 seconds

Experiment 2: Time with 1 httpserver and nc: 4.3 seconds

The performance difference can likely be attributed to the GET request being processed by 2 httpservers in experiment 1 as opposed to 1 httpserver in experiment 2. In experiment 1, the load balancer would split the load of the requests evenly according to the given heuristic (Assuming both servers start with 0 entries and errors). In experiment 2, 1 server is solely processing requests. Also in experiment 2, several requests returned a 500 error because of it timed out.