

Load Balancer Write Up

Jake Armendariz

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?

This assignment was all run on one machine, thus each "server" is just a process on my machine. If it was a real load balancer, it would send requests to a server that is geographically closer the client. This takes in performance attributes on location, but since every server is running on one machine, there is no need for geographic attributes or analysis.

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 we process the request, it could remove possible 400 requests, sending errors directly back to client. This would speed up response time on bad requests dramatically. The cost is an overhead of reading each response header on valid requests. Which is pretty heavy, but already I check for the content length everytime, so it wouldn't be that great of an additional cost.

Getting 8 400 MiB files

Load Balancer with 2 servers

```
real 0m19.144s
user 0m0.592s
sys 0m4.630s
```

```
real 0m21.857s
user 0m0.621s
sys 0m4.991s
```

1 Multi threaded

```
real 0m26.587s
```

user 0m1.091s
sys 0m6.848s

real 0m25.056s
user 0m1.005s
sys 0m6.986s

With netcast responding and not handling requests the time dropped down to:

real 0m15.330s
user 0m0.656s
sys 0m4.826s

Fairly dramatic speedup, which is to be expected when you do not need to run through data inside of the httpserver, it immediately sends back a response instead