

For this assignment, your proxy 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?

This was not used for this assignment because each HTTP server is just a process running on our computers. If it were a realistic implementation, the proxy would choose the HTTP server that has a faster connection depending on the physical location of the server to the client.

8x 419430400 bytes file test:

2 HTTP servers

real 0m14.181s

user 0m0.103s

sys 0m0.469s

1 HTTP server

real 0m15.847s

user 0m0.152s

sys 0m0.322s

Speedup = $15.847s / 14.181s = 1.1x$ speedup in performance over the 1 HTTP server

Proxy without caching:

real 0m45.512s

user 0m0.066s

sys 0m0.477s

Proxy with caching:

real 0m1.863s

user 0m0.009s

sys 0m0.000s

Speedup = $45.512s / 1.863s = 24.4x$ speedup in performance with caching on!!

What did you learn about system design from this class? In particular, describe how each of the basic techniques (abstraction, layering, hierarchy, and modularity) helped you by simplifying your design, making it more efficient, or making it easier to design.

Abstraction helped me a lot when pseudo-coding my design. I left things general at first and then focused on specifics when coding and dealing with modules concerning their functionality. I focused on implementing the functionality of the proxy one at a time such as dealing with connection forwarding first, then load balancing, and then finally caching. Modularity and layering helped me a lot in identifying which modules should interact with each other according

to their functions. One example is in `processRequest()` where I had to determine if the cached date of the file is newer or the same as the one in the priority server. I made the helper function `fileIsNewer()` and it reduced the amount of code/complexity in `processRequest()`. I'm not sure if hierarchy helped me in designing.

Testing:

I did some unit testing in the main file with caching to make sure enqueueing and dequeuing work first before adding it to my proxy.

I used my script `asgn3Test.sh` which tests concurrency, load balancing, and caching.

I also used Jess Fan's testing script `asgn3-test.py` in Discord and passed everything.