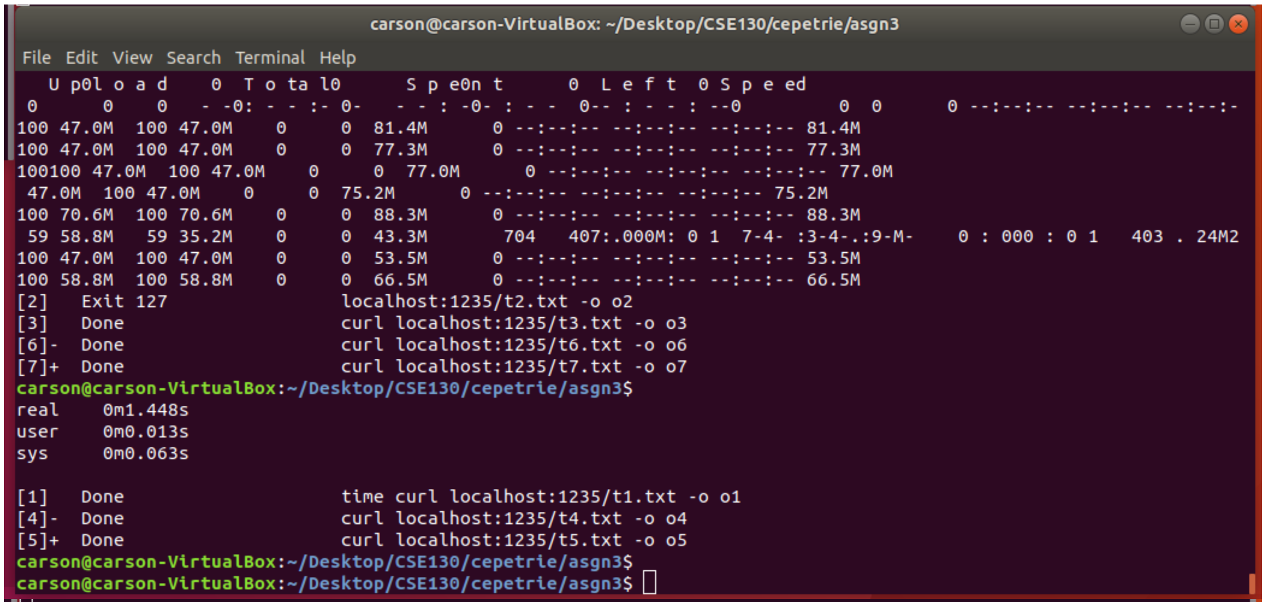
Carson Petrie
CSE 130 Assignment 3
Professor Alvaro
12/2/2021

Assignment 3 WRITEUP.pdf

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?

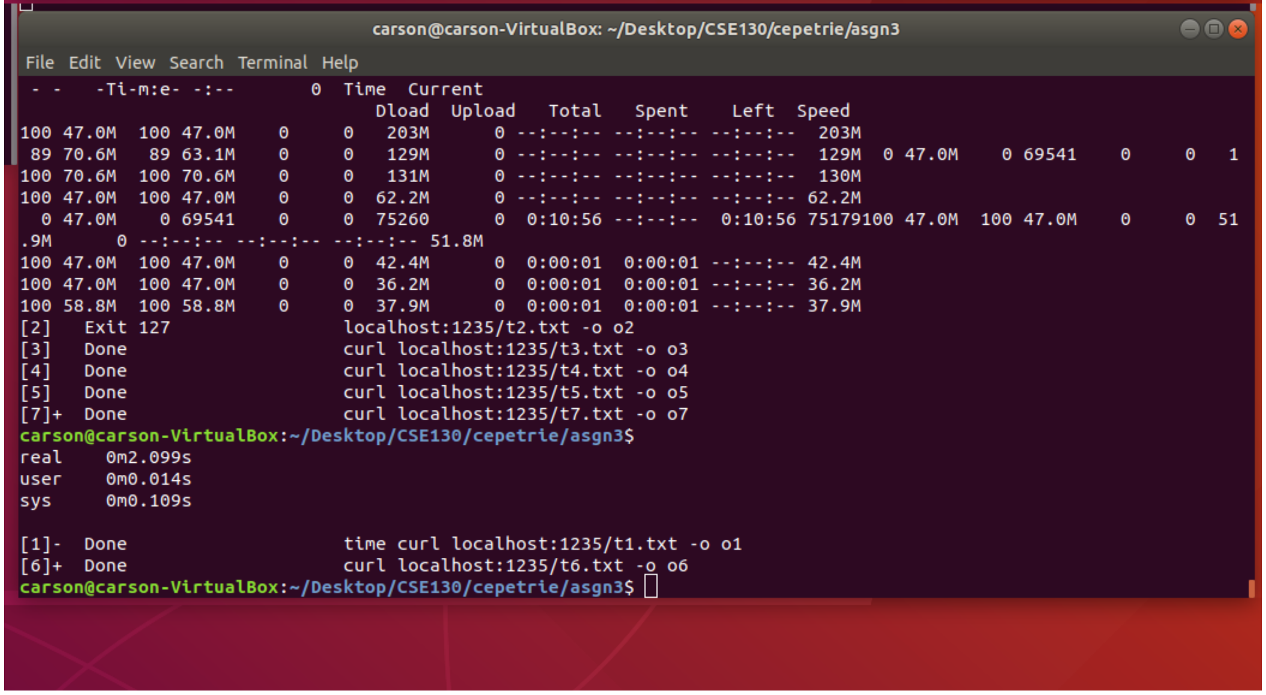
> A large purpose of this course is to work on extending our code. We went from constructing a small library of local utilities (GET, HEAD, and PUT) to creating a single-threaded web server that would support those same functionalities. We then multithreaded the server, and introduced logging and healthcheck functionality to monitor our servers performance. In this assignment, we create a proxy that allows us to redirect client requests to a multitude of multi-threaded servers. Yes, a more realistic implementation would evaluate server performance and hardware, but in terms of the codebase we have constructed, healthchecks and requests to failure ratio is the best available indicator we have of server performance.

- Using the provided httpserver, do the following:
  - Place eight different large files in a single directory. The files should be around 400 MiB long, adjust according to your computer's capacity.
  - Start two instances of your httpserver in that directory with four worker threads for each.
  - Start your httpproxy with the port numbers of the two running servers and maximum of eight connections and disable the cache.
  - Start eight separate instances of the client at the same time, one GETting each of the files and measure (using time(1)) how long it takes to get the files. The best way to do this is to write a simple shell script (command file) that starts eight copies of the client program in the background, by using & at the end.
- Repeat the same experiment, but turn off one of the servers. Is there any difference in performance? What do you observe?


carson@carson-VirtualBox: ~/Desktop/CSE130/cepetrie/asgn3
File Edit View Search Terminal Help

```
  U p0l o a d    0  T o ta l0      S p e0n t      0  L e f t  0 S p e ed
 0      0    0   - -0: - - :- 0-    - - : -0- : - -  0-- : - - : --0         0  0      0 --:--:-- --:--:-- --:--:-
100 47.0M  100 47.0M    0     0  81.4M      0 --:--:-- --:--:-- --:--:-- 81.4M
100 47.0M  100 47.0M    0     0  77.3M      0 --:--:-- --:--:-- --:--:-- 77.3M
100100 47.0M  100 47.0M    0     0  77.0M      0 --:--:-- --:--:-- --:--:-- 77.0M
 47.0M  100 47.0M    0     0  75.2M      0 --:--:-- --:--:-- --:--:-- 75.2M
100 70.6M  100 70.6M    0     0  88.3M      0 --:--:-- --:--:-- --:--:-- 88.3M
 59 58.8M   59 35.2M    0     0  43.3M      704   407:.000M: 0 1  7-4- :3-4-.:9-M-    0 : 000 : 0 1   403 . 24M2
100 47.0M  100 47.0M    0     0  53.5M      0 --:--:-- --:--:-- --:--:-- 53.5M
100 58.8M  100 58.8M    0     0  66.5M      0 --:--:-- --:--:-- --:--:-- 66.5M
[2]   Exit 127                localhost:1235/t2.txt -o o2
[3]   Done                    curl localhost:1235/t3.txt -o o3
[6]-  Done                    curl localhost:1235/t6.txt -o o6
[7]+  Done                    curl localhost:1235/t7.txt -o o7
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
real    0m1.448s
user    0m0.013s
sys     0m0.063s

[1]   Done                    time curl localhost:1235/t1.txt -o o1
[4]-  Done                    curl localhost:1235/t4.txt -o o4
[5]+  Done                    curl localhost:1235/t5.txt -o o5
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
```


carson@carson-VirtualBox: ~/Desktop/CSE130/cepetrie/asgn3
File Edit View Search Terminal Help

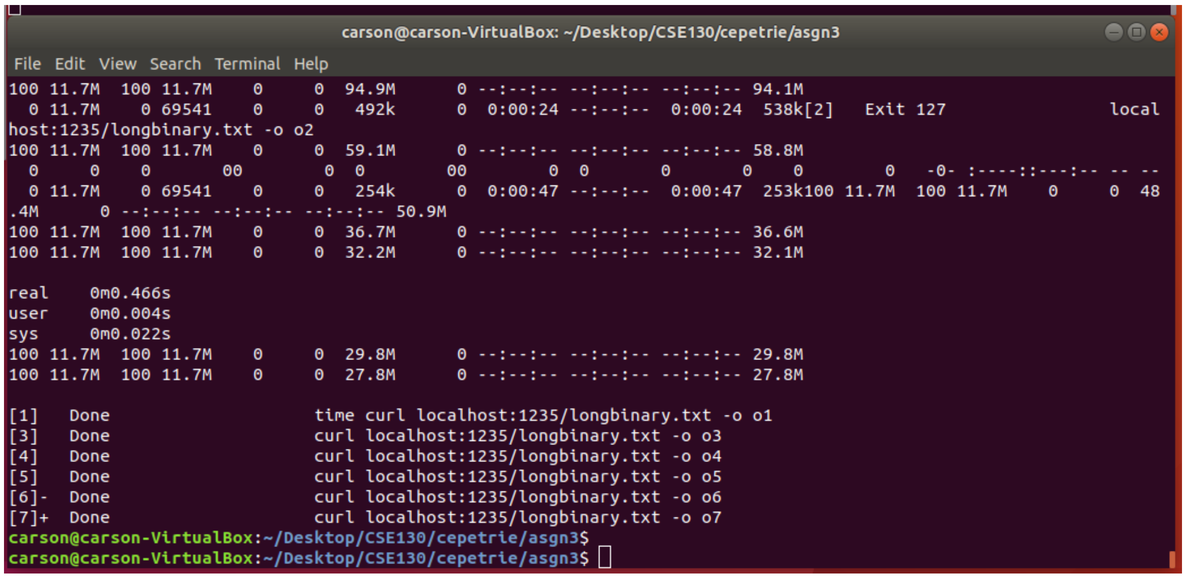
```
 - -   -Ti-m:e- -:--       0  Time  Current
                              Dload  Upload   Total   Spent    Left  Speed
100 47.0M  100 47.0M    0     0   203M      0 --:--:-- --:--:-- --:--:--  203M
 89 70.6M   89 63.1M    0     0   129M      0 --:--:-- --:--:-- --:--:--  129M  0 47.0M    0 69541    0     0   1
100 70.6M  100 70.6M    0     0   131M      0 --:--:-- --:--:-- --:--:--  130M
100 47.0M  100 47.0M    0     0  62.2M      0 --:--:-- --:--:-- --:--:-- 62.2M
  0 47.0M    0 69541    0     0  75260      0  0:10:56 --:--:--  0:10:56 75179100 47.0M  100 47.0M    0     0  51
.9M      0 --:--:-- --:--:-- --:--:-- 51.8M
100 47.0M  100 47.0M    0     0  42.4M      0  0:00:01  0:00:01 --:--:-- 42.4M
100 47.0M  100 47.0M    0     0  36.2M      0  0:00:01  0:00:01 --:--:-- 36.2M
100 58.8M  100 58.8M    0     0  37.9M      0  0:00:01  0:00:01 --:--:-- 37.9M
[2]   Exit 127                localhost:1235/t2.txt -o o2
[3]   Done                    curl localhost:1235/t3.txt -o o3
[4]   Done                    curl localhost:1235/t4.txt -o o4
[5]   Done                    curl localhost:1235/t5.txt -o o5
[7]+  Done                    curl localhost:1235/t7.txt -o o7
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
real    0m2.099s
user    0m0.014s
sys     0m0.109s

[1]-  Done                    time curl localhost:1235/t1.txt -o o1
[6]+  Done                    curl localhost:1235/t6.txt -o o6
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
```

We can observe that by turning off one of the servers, even this small scale of a test demonstrates a marginally worse performance. This is to be expected however. In my implementation of the http proxy, the heaviest working regions of the code are thread-safe, and capable of being run in parallel. This means that we can take large advantage of having multiple servers to proxy to. When we take one away, it is unexpected that the code slows down.

•Using one of the files that you created for the previous question, start the provided server with only one thread, then do the following:

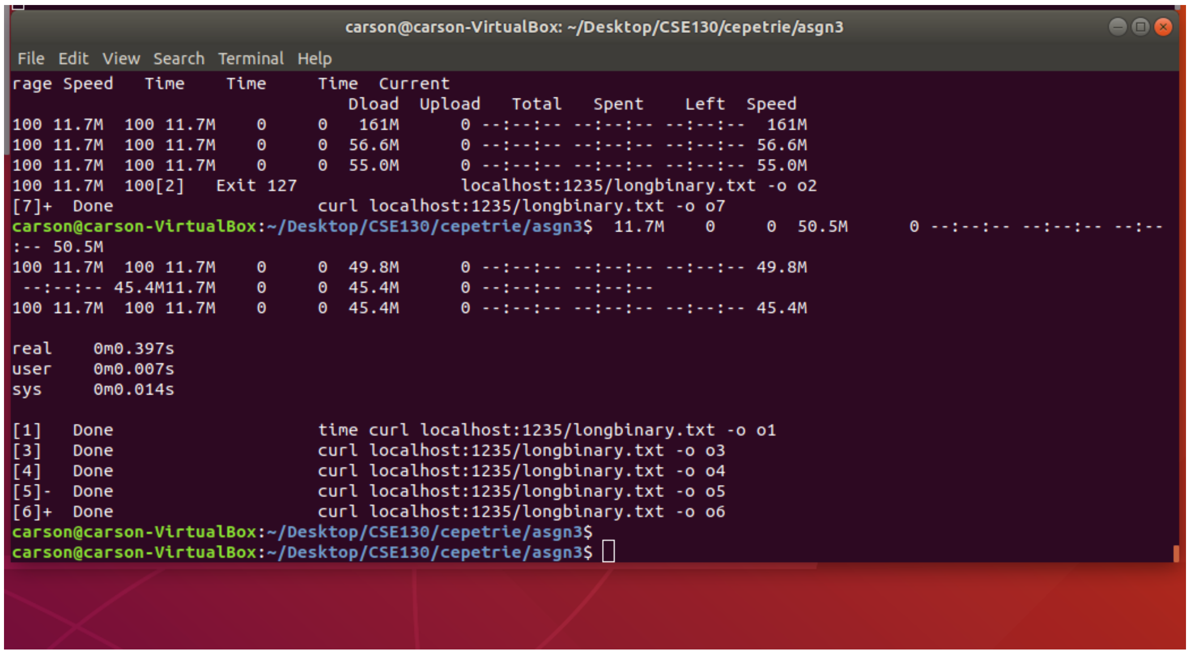
- – Start your proxy without caching. Request the file ten times. How long does it take?
- – Now stop your proxy and start again, this time with caching configured so that it can store the selected file. Request the same file ten times again. How long does it take now?


carson@carson-VirtualBox: ~/Desktop/CSE130/cepetrie/asgn3
File Edit View Search Terminal Help

```
100 11.7M  100 11.7M    0     0  94.9M      0 --:--:-- --:--:-- --:--:-- 94.1M
  0 11.7M    0 69541    0     0   492k      0  0:00:24 --:--:--  0:00:24  538k[2]   Exit 127                local
host:1235/longbinary.txt -o o2
100 11.7M  100 11.7M    0     0  59.1M      0 --:--:-- --:--:-- --:--:-- 58.8M
  0     0    0      00      0  0       00       0  0       0       0    0        0   -0- :----::---:-- -- --
  0 11.7M    0 69541    0     0   254k      0  0:00:47 --:--:--  0:00:47  253k100 11.7M  100 11.7M    0     0  48
.4M      0 --:--:-- --:--:-- --:--:-- 50.9M
100 11.7M  100 11.7M    0     0  36.7M      0 --:--:-- --:--:-- --:--:-- 36.6M
100 11.7M  100 11.7M    0     0  32.2M      0 --:--:-- --:--:-- --:--:-- 32.1M

real    0m0.466s
user    0m0.004s
sys     0m0.022s
100 11.7M  100 11.7M    0     0  29.8M      0 --:--:-- --:--:-- --:--:-- 29.8M
100 11.7M  100 11.7M    0     0  27.8M      0 --:--:-- --:--:-- --:--:-- 27.8M

[1]   Done                    time curl localhost:1235/longbinary.txt -o o1
[3]   Done                    curl localhost:1235/longbinary.txt -o o3
[4]   Done                    curl localhost:1235/longbinary.txt -o o4
[5]   Done                    curl localhost:1235/longbinary.txt -o o5
[6]-  Done                    curl localhost:1235/longbinary.txt -o o6
[7]+  Done                    curl localhost:1235/longbinary.txt -o o7
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
```


carson@carson-VirtualBox: ~/Desktop/CSE130/cepetrie/asgn3
File Edit View Search Terminal Help

```
rage Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11.7M  100 11.7M    0     0   161M      0 --:--:-- --:--:-- --:--:--  161M
100 11.7M  100 11.7M    0     0  56.6M      0 --:--:-- --:--:-- --:--:-- 56.6M
100 11.7M  100 11.7M    0     0  55.0M      0 --:--:-- --:--:-- --:--:-- 55.0M
100 11.7M  100[2]   Exit 127                localhost:1235/longbinary.txt -o o2
[7]+  Done                    curl localhost:1235/longbinary.txt -o o7
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$  11.7M    0     0  50.5M      0 --:--:-- --:--:-- --:--
:-- 50.5M
100 11.7M  100 11.7M    0     0  49.8M      0 --:--:-- --:--:-- --:--:-- 49.8M
 --:--:-- 45.4M11.7M    0     0  45.4M      0 --:--:-- --:--:--
100 11.7M  100 11.7M    0     0  45.4M      0 --:--:-- --:--:-- --:--:-- 45.4M

real    0m0.397s
user    0m0.007s
sys     0m0.014s

[1]   Done                    time curl localhost:1235/longbinary.txt -o o1
[3]   Done                    curl localhost:1235/longbinary.txt -o o3
[4]   Done                    curl localhost:1235/longbinary.txt -o o4
[5]-  Done                    curl localhost:1235/longbinary.txt -o o5
[6]+  Done                    curl localhost:1235/longbinary.txt -o o6
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
carson@carson-VirtualBox:~/Desktop/CSE130/cepetrie/asgn3$
```

Here we can observe that by utilizing our caching, our servers performance is increased when repeatedly requesting a cached file. This is not unexpected because by enabling caching, after our first handling of the request we store the files contents inside our programs memory itself. We can then send it directly to the client when requested, as opposed to having to utilize a buffer and an indeterminately long while loop to proxy the request from the client to server and back. This cuts down on the computational overhead and therefore also cuts down on the processing time as well.

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.

> This class drastically changed my approach to creating design documents as well as creating a new context for me to reason about system design in. When I think back to similar courses, such as Dunne's 13S I realized that my biggest struggle in that class was being able to reason about two designs, and know why one is better than the other. Lecture topics from this course and it's readings, such as abstraction, layering, hierarchy, and modularity, are all valuable tools for me to think about designing systems with. Instead of starting out with a detailed flowchart and list of functions, I now draw large boxes, or modularize components of my designs. The next step would be to piece these boxes together, or abstract the components. These two techniques work together to help reduce propagation of effects and manage complexity in designs, whilst also making it easier to come back at later stages and rework or rewrite components.
>
> You could even go as far as to take this whole as an example of the power of systems design in regards to these techniques. We created an entire multi-threaded HTTP server, and were able to interject an entire new component between it and the client without either really being aware of it. This is why abstraction and modularity are so powerful.
>
> Layering was a key factor in the extension of our code also in a way. We could think about our initial shoulders assignment as developing a working set of utilities (GET HEAD and PUT). We then built upon this foundation by creating a single-threaded HTTP server. Upon this, we built a multi-threaded HTTP server. Although this extension is not a perfect example of layering, the idea of extending an existing base of mechanisms or modules and limiting their between to the layer immediately below it is quite relevant in a way. For multi-threading a HTTP server, I worked primarily with my single-threaded server implementation. I did not need to reference my shoulders or utility implementation themselves much at all.