Network Programming Phase3 httpproxy

Learn diary

Liu Peng 280600

To fulfill the requirements in the assignment:

1. Server should serve HTTP GET/PUT request from multiple clients
2. Proxy acts as client to other servers or proxies
3. Pass queries to other proxies
4. A new header Hop-Count should be added to scale the proxy
5. Forward the request to at least two servers
6. Index should return both local files and additionally return index from two other servers
7. PUT should be passed to other two servers as well

Problems and solutions during programming:

1. How and when to redirect the get request?

Since the code is based on former httpserver, when a get request comes in, the proxy should read the route list from a single file, I name it routelist, in the routelist, there are 3 URLs, which refer to 3 netlab servers, if the file does not exist on the local folder, it forward the get request to the next server in the list until it find the file, or it will reply not found message to the original client.

2. How to make index include the directory of other servers?

In my server code, I just print the root directory to show the index, now I added some code to send the get index request to the servers in the routelist, thus I can also include the returned index to form a new one, and send it back to the client.

3. How to forward the put request?

Since the put request is only sent to the nearby servers, it is much easier than the previous two cases. Firstly I get the content which was in the body of put request, then I create a file in the local folder and store the content, when this is finished, I begin to read from the local file and send it to the nearby servers at the same time.

4. How to use the Hop-Count header?

The new Hop-Count header is required in the proxy, it can limit the scale of the proxy. Since in normal browsers, there is not such header, when the proxy can not find the header, it will add it and set the default hop count as 2. Some actions like get index and put files require this field to be set as 1, I did it in different functions separately.

When the proxy need to do some forwarding, it will firstly check if the hop count is above 0, if not, forwarding is not allowed. The hop count is also reduced by one during every forwarding.

5. When I did the forwarding part, my proxy can not understand some servers reply.

When I look into this problem I realize that different servers in netlab have slightly different headers, like the HTTP/1.0 instead of HTTP/1.1, and like Content-length instead of Content-Length, so I just adjust my code to be not so sensitive to these header names, I used wireshark to check every packet I received and tried a long time to fix the bug.

One thing I learn from this problem is that the format must be standardized, or some little mistakes will cost a lot of extra effort.

6. How to ensure the safety of the routlist?

To make sure that it is not so easy to modify the routelist from outside, I hide the file in the parent directory of the root folder.

Function test:

features work	features don’t work
Support GET request, can deliver the file on local disk to the requesting client, if file does not exist on local folder, the proxy forward the request to the nearby servers in the routelist	Only work for plain text
Support PUT request, can accept file from the client and create and save with the given file name on local disk and forward the put message to the nearby servers in the routelist as well	The time out function has not been implemented yet.
Iam header and Hop-Count header works -support to identify client by Iam header and save it in the log file -support to use Hop-Count to limit the scale of proxy	
Tested it’s working for standard URLs and other web browsers	
Large file test successfully -support put and get request to transfer a 520KB file successfully	
Support index function	

-can list all the file names under the root directory and checked with my Chrome browser -can pass the get index request to the nearby servers in the routelist and combine the index content before sending it back to client	
valgrind test passed -no leaked memory -no errors in memory operation	
Program uploaded in the netlab server and keeps running 2 days without problem	
Keep a log file that records all events that happened on the server	
Keep a daemon.lock file to indicate the running service process id -make it easier for administrator to manage the service	