

Proxy-Server

An HTTP proxy server implemented via python socket programming with caching, threading and blacklisting.

Description

Proxy Server

Generally, when the client (using curl) makes a request, the request is sent to the web server. The web server then processes the request and sends back a response message to the requesting client.

In order to improve the performance we create a proxy server between the client and the web server . A web proxy is a program that acts as an intermediary between a web client (browser or curl) and a web server.

The client requests the objects via the **proxy server**. The proxy server will forward the client's request to the web server. The web server will then generate a response message and deliver it to the proxy server, which in turn sends it to the client.

Caching

When the proxy server gets a request more than 3 times in 5 minutes, it checks whether the requested object is **cached** (i.e. server already has the request webpage or file), and if yes, it returns the object from the cache, without contacting the server.

If the object is not cached, the proxy retrieves the object from the server, returns it to you and caches a copy of this file for future requests if the **Cache-Control** header is set to **must-revalidate**. If the **Cache-Control** header is set to **no-cache**, then the proxy server does not caches the file.

In case of any further requests if the webpagepr file is already cached then, the proxy utilize the **If Modified Since** header to check if any updates have been made, and if not, then serve the response from the cache, otherwise webpage or file is again retrieves from the server.

Code Structure

- proxy.py,serverthread.py are the main proxy file.
- myserver contains the code of backend server.
- users.txt contains authentoication details.
- Blaclist.txt contains list of blacklisted servers.
- cachefiles folder will be contains the files cached by proxy server.

Features

- Checks if the cached object has been modified ? If yes, forwards the request to server and also updates the cache.
- Threaded proxy server thus able to handle many requests at the same time.

- Cache has limited size, so if the cache is full and proxy wants to store another response then it removes the least recently asked cached response.
- Certain servers (their ports) are blacklisted so that users can't access it.

How to run

Proxy

`python proxy.py` // runs proxy server on port 20100.

Server

- Run server from `myserver.py` file using `python myserver.py port_number`

Client

Client can request the webpage or a file from above running server through either browser or curl.

Using Curl

`curl -x http://host_addr:host_port http://server_addr:server_port/object_location`