Assignment 2
CSCI 415
Perry, the Python Proxy

Cesar Ramirez, Josh Tan

October 7, 2013

Overview

The script will start by initializing the cache and the log file. Once those are secure the first param from stdin will be read and used as the local port number. Then the start_server() method will be called to initialize the local port with the parameters given. Once we are listening a while loop is started that will be always listening for new connections, if a connection is created then it will be accepted and sent to a ProxyConn object in a separate thread. The server will always have n+1 running threads, where n is the number of active connections.

Main Classes

ProxyConn

The ProxyConn object is the entire lifecycle of a client requesting a website through this proxy. This object will recieve the socket of the client, then a ClientRequest object will be created to handle the request and once that is completed the connection with the client will be closed.

ClientRequest

The ClientRequest object handles creating a HttpRequest object from the data provided and then doing the entire request process. The most important method is the execute() method, which is where all the data relay happens between client and server and also where the cache is used.

HttpRequest

A HttpRequest is an abstraction of the http request from the client. The goal of this class is to provide a way to easily manage and handle the request line as well as the request headers. This objext is build from the raw text from the client and all the data is parsed into object.

Cache

This is an on-dick cache that follows a Least-Recently-Used model. It has a global lock that makes sure that all read/writes are thread-safe. You can also set the maximun size in bytes for the cache. In memory we will keep a reference to each entry of the cache but the data itself will be stored in a file uniquely identified by a key generated from the requested uri.

Entries as doubly linked list.