Samir Cerrato

COMP332

Professor Manfredi

Febuary 29, 2024

<u>Write-Up HW #4</u>

The web client and proxy are designed to efficiently handle incoming HTTP requests from clients, forward them to the appropriate web server, cache responses to improve performance, and handle multiple clients simultaneously. Eventually, we will move away from using the web client and port 50008. Instead, we will change the port the proxy is using to 50007 and try connecting with a web browser. Safari is the recommended browser because google chrome was causing all the threads to go off and crash the code.

The program utilizes threading to handle multiple client connections at the same time. This ensures that the proxy can serve multiple clients simultaneously without blocking a request. We also implement caching to store responses from web servers, which can be reused for subsequent requests for the same resource. This helps in reducing latency and bandwidth usage, especially for frequently accessed URLs.

Basic parsing of HTTP requests and responses is performed to extract relevant information such as hostnames, paths, and headers. This allows the proxy to make informed decisions, such as whether to fetch a resource from the cache or forward the request to the web server. **However, we run into errors with parsing when we connect a web browser to our proxy. The code fails to determine a host name and cannot get the request.** The parsing works properly when using a web client and using port 50008. Ideally, the program is designed to work with websites that use UTF-8 encoding and follow standard HTTP protocol. An example of this code working is requesting the content from htttp://example.com/ through the proxy. Others include, http://eu.httpbin.org, http://info.cern.ch/, and http://www-db.deis.unibo.it/. Below is a screenshot of the code running successfully with a working test case, web client, proper caching, and using port 50008.

However, the program may not work with websites that use encoding other than UTF-8, implement non-standard HTTP protocols, have dynamic content, or require authentication or authorization to log in. Some examples of such websites include https://www.google.com, https://www.facebook.com, or https://www.amazon.com. Additionally, you might run into an error while parsing the URL. When a web browser uses a proxy, it includes the full URL including the hostname as the path in the GET line of the request. This pathname, however, is not parseable by web servers. Below is a screenshot of the program running into an error when trying to request http://info.cern.ch/hypertext/WWW/TheProject.html and http://example.com/. However, when using the web browser and port 50007 to connect to http://eu.httpbin.org, the parsing somewhat works, and the request is sent.

While repeatedly testing other websites, we get other errors saying: "error communicating with web server: [Errno 8] nodename nor servname provide, or not known". There could be a few reasons for this failure, with an invalid hostname or network connectivity issues to name a few. We also get the error "HTTP field not found" and "Cannot determine host". This means that our URL was not parse properly and therefore could not connect to the proxy.