Dylan Hayton-Ruffner
Luca Ostertag-Hill

Checkpoint 1

Our goal for the first checkpoint was to have written a simple client in python that sends HTTP GET messages, a simple Java proxy, and have tested the proxy's ability to forward requests on to the Apache backend.

Code Progress:

In keeping with the principle of automating everything, we have created a library of bash scripts that provide several essential system functions. These scripts are able to load the Apache2 httpd source to the backend servers, configure and install httpd on the backend, automate the start of all backend Apache servers, and automate the termination of all backend Apache serves. Simple curl and Chrome requests for an index.html file were used to make sure these Apache servers were functional.

As detailed in our outline, our initial goal was to set up one frontend proxy to handle traffic. We have managed to implement a Java Socket based proxy that takes a request from a client and writes it out to one of the backend Apache servers. It also takes the response from the Apache server and writes it back out to the client. This was tested using curl. The proxy is able to stay open and process multiple requests from clients.

Complications:

We have encountered a problem with the coordination of replicas and the maintenance of the distributed index that stores the locations of the files across the servers. After reviewing the functionality of Apache, it seems that using the proxies themselves to coordinate replication and loading of files to the distributed service is not possible. Instead, we intend to create a set of Replication Managers, running on RMI, XML-RPC, or another protocol, that will upload files to servers and keep track of which servers have which file. The proxies will interact with these replication managers in order to 'find' the location of files. Our next goal is to get one Replication Manager up and running, so that we can integrate it with our frontend proxy and start testing the service.