CSE 124 Fall 2010 Project
Scalable Superproxy with Distributed File System

This project is an extended SuperProxy that implements Hadoop Distributed File System (HDFS). It is a web-based proxy service that provides a distributed, scalable, and portable filesystem. It is a web-based HTTP proxy which users input target URLs to the SuperProxy website. Then, the proxy returns either cached or non-cached contents. The problem with a normal SuperProxy is that the scalability depends on the size of the cache. There are multiple solutions to this, both in hardware and software. Examples of hardware solutions are installing a huge RAM, flash drive, external hard drive or implementing Redundant Array of Independent Disks (RAID). Examples of software solutions are such as regularly deleting data or caching only some data. Another solution to this can be implementing file system across multiple machines, such as Google File System. Our solution in our project to solve cache limitations is to implement HDFS, which is very similar to Google File System.

The architecture of HDFS client request works as the following: first, the client retrieves several pieces of information from Namenode, such as the file location, the number of chunks the file should be split into, and the location of the chunks in the Datanodes. Then, the client requests the chunks from the Datanodes and merges them into a single file. The advantage of this architecture is that each file chunks can be stored into multiple Datanodes. By doing this, each Datanode is not required to have a large data storage to store the chunks of files. Additionally, there is another advantage to this architecture. By having multiple Datanodes, each chunk can be stored in multiple Datanodes to act as backups. By doing this, if any Datanode crashes, the client can still retrieve the chunks from other Datanodes. Unfortunately, we would have been able to implement the backup Datanodes but due to the lack of resources we were forced to abandon the backup feature.

Our SuperProxy uses Tomcat and HTTP Servlet. It also implements URL rewriting which rewrites HTML links and image links to request data from our server. There are three different types of work flow in our SuperProxy. The first one is for non-cached URLs. First, the client inputs the target URL into the SuperProxy website. Then, the SuperProxy performs parallel download by using multiple TCP connections. The SuperProxy then saves the URL string and the last-updated time in memory and saves the contents in HDFS. Lastly, the SuperProxy returns the data to the client. The second type and third type are for cached URLs. Similarily to non-cached, the client inputs the target URL into the SuperProxy website. However, instead of parallel downloading, the SuperProxy finds the URL string in cache and checks if contents are newer than the cached data. The second type is when the result is no, meaning the cached data is most updated. In this case, the SuperProxy loads contents from HDFS and returns the data to the client. The third type is when the result is yes, meaning the cached data is not updated. In this case, the SuperProxy requests new data from the target website and updates the URL string and the last-updated time in memory and contents in HDFS. Lastly, the SuperProxy returns data to the client.

Contributors
Tassapol Athiapinya - A50042456
Edwin Makiuchi - A07668278

In this guide, we assume that server is at 216.24.193.224.

Setup Instructions

1. SSH to 216.24.193.224 with user hadoop
2. Stop all hadoop processes by $ ./stop-all.sh
3. Format hadoop file system if necessary by $ ./hadoop namenode -format
4. Start hadoop by $ ./start-all.sh
5. Start Tomcat by $ sudo /etc/init.d/tomcat6 restart

Usage of Proxy

Point your web browser to http://216.24.193.224:8080/cse124demo/

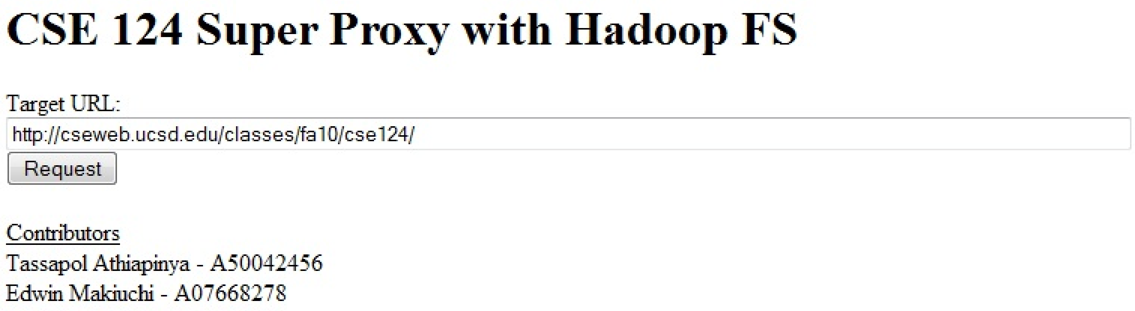
Monitoring SuperProxy’s Cache Status

Point your web browser to http://216.24.193.224:8080/cse124demo/CacheStatus

Monitoring Hadoop File System

Point your web browser to http://216.24.193.224:50070/dfshealth.jsp

Screenshots



CSE 124 Super Proxy with Hadoop FS

Target URL: http://cseweb.ucsd.edu/classes/fa10/cse124/

Contributors

Tassapol Athiapinya - A50042456
Edwin Makiuchi - A07668278