COMPSCI711 A2 report

Name: Shupeng Xu
Student ID: 8260026
UPI: sxu487

October 23, 2015

1 Run on a lab machine

I used Python 2.7 to do A2. It should be run and tested under Linux. I tested my program using Ubuntu 14.10 on a lab machine. To test my program, you'd better open four Terminal windows at the same time. See Figure 1.

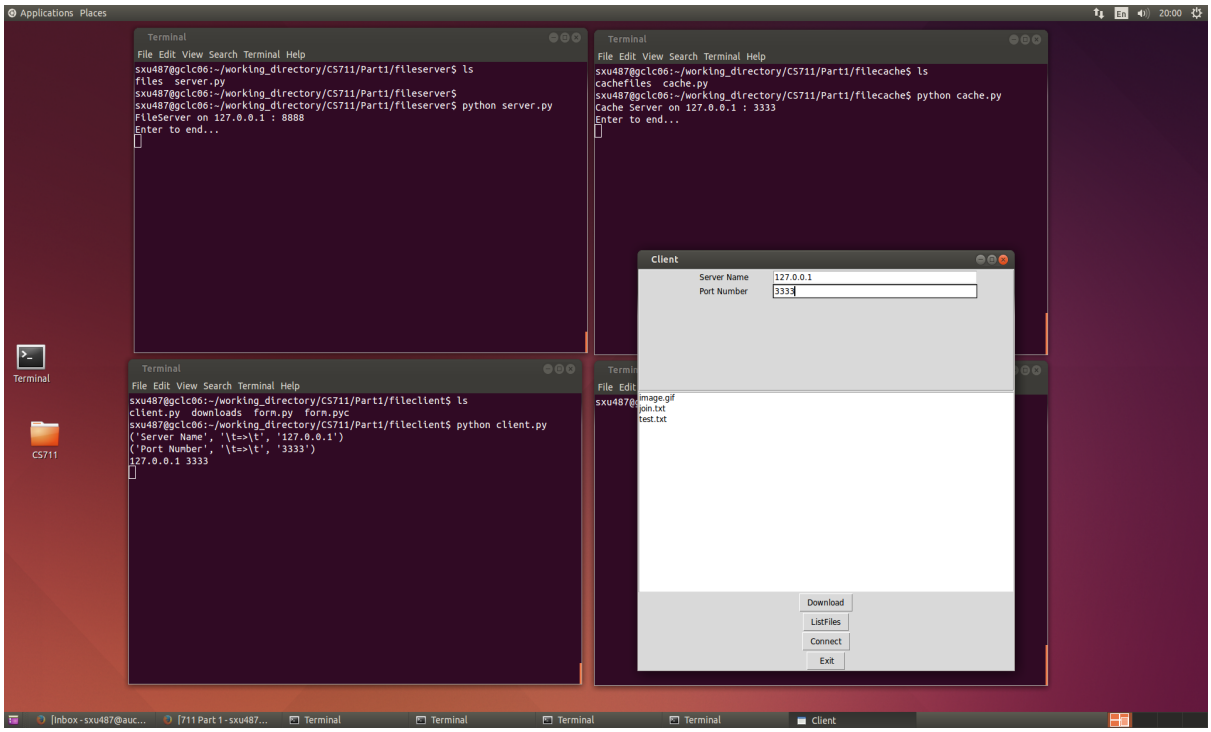

Figure 1: On a lab machine

2 How to run my program

For part 1 and part 2, the commands are the same.

1. Open four Terminal windows and navigate to fileserver, filecache, fileclient separately.
2. Put files into fileserver/files folder and use command: python server.py to run the server program.
3. Using command: python cache.py to run the cache program.

4. Using command: python client.py to run the client program.

5. Input Server Name: 127.0.0.1 and Port Number: 3333 and Click Connect.

6. Click ListFiles to list files on the server.

7. Select a file and click Download to download a file. The file downloaded can be found in folder fileclient/downloads.

8. The cache log and request info can be found in folder filecache/cachefiles. The cache log and request log info are written using JSON format. The name of cached file log is cache.log and request info is requestinfo.log.

Using the commands in Figure 2 to see the downloaded files, cache log and request log info. In part 2, you can also see the chunks in folder filecache/cachechunks and fileserver/chunks.

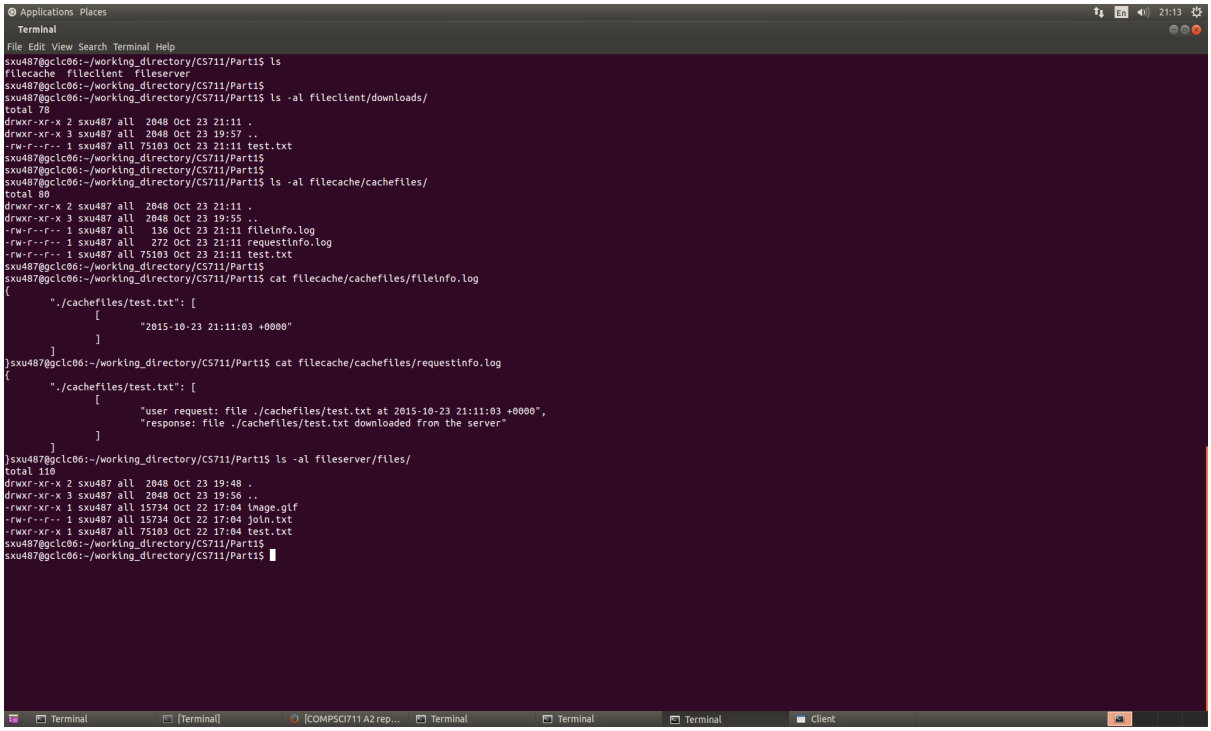

Figure 2: Show Info

3 Discussion

The Rabin-Karp rolling hash algorithm is excellent at finding a pattern in a large file or stream, but there is an even more interesting use case: creating content based chunks of a file to detect the changed blocks without doing a full byte-by-byte comparison[1]. I used this algorithm to break up a large file into chunks.

For the server part, every time it received a request, it will break up the requested file into chunks and store these chunks using digest(SHA1) of the chunks as file names. The server will keep all the digests of chunks it has in a chunk list. Next time when it received another request, it will break up the requested file into chunks and check if these chunks exist in the existing chunk list. If a chunk is in the existing chunk list, it means that this chunk has been already sent to the cache. Therefore, this chunk does not need to be sent to cache again. Otherwise, the

server will send the chunk to the cache. The server will also send the chunk list of the requested file to the cache so that the cache can join the chunks together to a complete file.

For the cache part, it will receive the chunks of a requested file from the server and join the chunks together using the chunk list received from the server. As the sever will decide whether a chunk needs to be sent to the cache, the cache only needs to receive chunk data and join them together in order to send the complete file to the client.

For the client part, it is the same in Part 1 and Part 2. The client does not need to know the details about the cache and the server.

References

[1] *http://blog.teamleadnet.com/2012/10/rabin-karp-rolling-hash-dynamic-sized.html*