

Report of Assignment 1

Chapter 1 Environment requirement

OS: it has been tested under win7 of lab machine and ubuntu14.04

JDK: it has been tested under jdk1.6.0_21 and jdk1.8.0_60

Chapter 2 Deploy steps

To make it clear in describing the steps, we assume the three machines as below

File server IP: 192.168.1.10

Cache server IP: 192.168.1.20

File Client IP: 192.168.1.30

File server port: 8091

cache server port: 8090

We can configure these parameters according to reality requirement.

In this report we use win7 environment as the example to demonstrate the deploy steps.

2.1 deploy part1

2.1.1 Steps to deploy file Server

1. Save A1.zip under one path of file server machine at IP: 192.168.1.10
2. Decompress the A1.zip
3. cd ass1\conf
4. Edit the file setting.conf as below

```
[fileServerConfig]
serverFilePath=H:\Documents\tmp\serverpath
fileServerPort = 8091
```

We should configure the two parameters according to reality requirement

5. Copy some test files under the server file path
6. Run scripts build.bat under ass1\ to compile the software
7. cd bin (this step is necessary)
8. Run script runFileServer.bat under ass1\bin\ . The whole process is like below

```
C:\Windows\system32\cmd.exe

H:\Documents\tmp>cd ass1

H:\Documents\tmp\ass1>dir
Volume in drive H is MyHome
Volume Serial Number is FE3A-855E

Directory of H:\Documents\tmp\ass1

07/09/2016  07:20 p.m.      <DIR>          .
09/09/2016  02:27 p.m.      <DIR>          ..
09/09/2016  03:04 p.m.      <DIR>          classes
07/09/2016  07:24 p.m.      <DIR>          bin
03/09/2016  08:28 p.m.      <DIR>          conf
03/09/2016  04:06 p.m.             114 build.sh
03/09/2016  05:38 p.m.             61 clean.bat
03/09/2016  03:24 p.m.      <DIR>          src
03/09/2016  04:54 p.m.             193 build.bat
03/09/2016  05:05 p.m.             70 clean.sh
               4 File(s)              438 bytes
               6 Dir(s)  37,667,672,686,592 bytes free

H:\Documents\tmp\ass1>build.bat

H:\Documents\tmp\ass1>javac src/org/cache/code/*.java src/org/cache/code/excepti
on/*.java src/org/cache/code/model/*.java src/org/cache/code/ui/*.java src/org/c
ache/code/util/*.java -d classes
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

H:\Documents\tmp\ass1>echo "build success!"
"build success!"

H:\Documents\tmp\ass1>cd bin

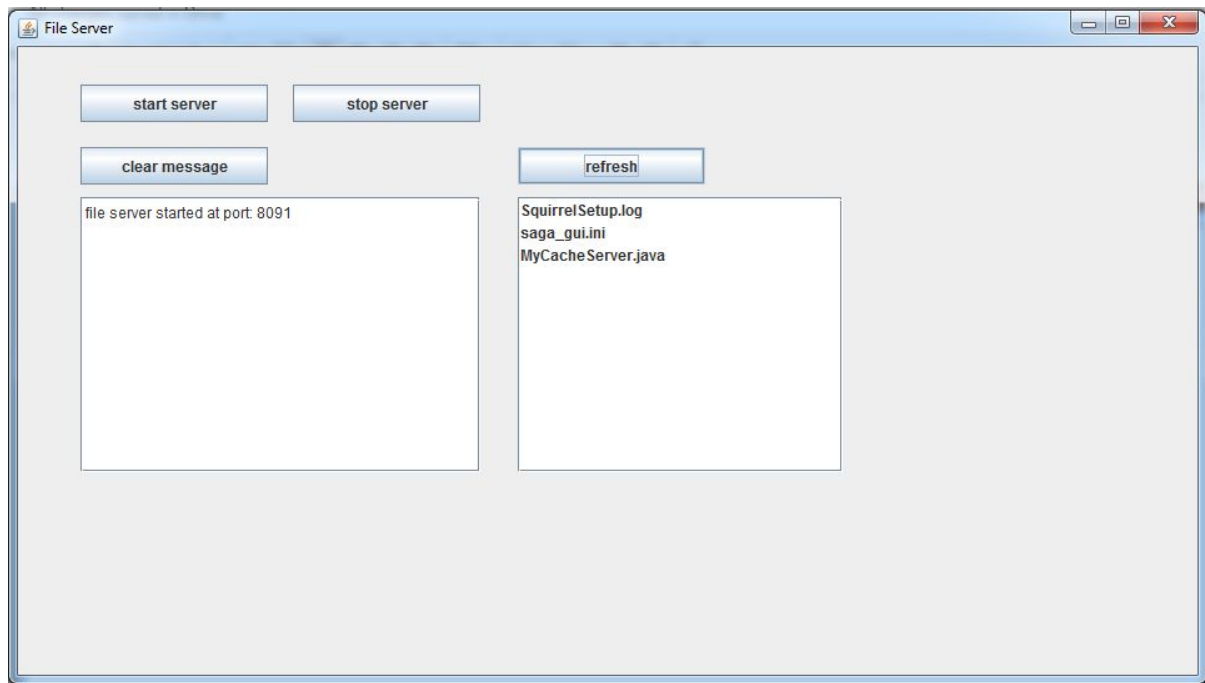
H:\Documents\tmp\ass1\bin>dir
Volume in drive H is MyHome
Volume Serial Number is FE3A-855E

Directory of H:\Documents\tmp\ass1\bin

07/09/2016  07:24 p.m.      <DIR>          .
07/09/2016  07:20 p.m.      <DIR>          ..
07/09/2016  07:23 p.m.             60 runCacheServerPart1.bat
03/09/2016  05:10 p.m.             53 runFileServer.bat
03/09/2016  04:23 p.m.             65 runFileClient.sh
03/09/2016  05:11 p.m.             53 runFileClient.bat
07/09/2016  07:21 p.m.             72 runCacheServerPart1.sh
07/09/2016  07:22 p.m.             72 runCacheServerPart2.sh
07/09/2016  07:24 p.m.             60 runCacheServerPart2.bat
03/09/2016  04:17 p.m.             65 runFileServer.sh
               8 File(s)              500 bytes
               2 Dir(s)  37,667,672,686,592 bytes free

H:\Documents\tmp\ass1\bin>runFileServer.bat_
```

9. The program will start with GUI as below, click button "start Server" to start it. Click button "refresh", it will show the file list under the configured server path.



10.

2.1.2 Steps to deploy cache Server (part 1)

1. Save A1.zip under one path of cache server machine at IP: 192.168.1.20
2. Decompress A1.zip
3. cd ass1\conf Edit the file setting.conf as below

```
[cacheServerConfig]  
fileServerIP = 192.168.1.10  
fileServerPort = 8091  
cacheServerPort = 8090
```

We should configure the three parameters according to real requirement.

4. Run scripts build.bat under ass\ to compile the software
5. cd bin (this step is necessary)
6. Run script runCacheServerPart1.bat under ass1\bin\ . The whole process is like below

```
C:\Windows\system32\cmd.exe
H:\Documents\tmp>cd ass1
H:\Documents\tmp\ass1>dir
Volume in drive H is MyHome
Volume Serial Number is FE3A-855E

Directory of H:\Documents\tmp\ass1

07/09/2016  07:20 p.m.    <DIR>          .
09/09/2016  03:28 p.m.    <DIR>          ..
09/09/2016  03:04 p.m.    <DIR>          classes
07/09/2016  07:24 p.m.    <DIR>          bin
03/09/2016  08:28 p.m.    <DIR>          conf
03/09/2016  04:06 p.m.             114 build.sh
03/09/2016  05:38 p.m.             61 clean.bat
03/09/2016  03:24 p.m.    <DIR>          src
03/09/2016  04:54 p.m.             193 build.bat
03/09/2016  05:05 p.m.             70 clean.sh
               4 File(s)              438 bytes
               6 Dir(s)  37,669,385,011,200 bytes free

H:\Documents\tmp\ass1>build.bat
H:\Documents\tmp\ass1>javac src/org/cache/code/*.java src/org/cache/code/excepti
on/*.java src/org/cache/code/model/*.java src/org/cache/code/ui/*.java src/org/c
ache/code/util/*.java -d classes
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

H:\Documents\tmp\ass1>echo "build success!"
"build success!"

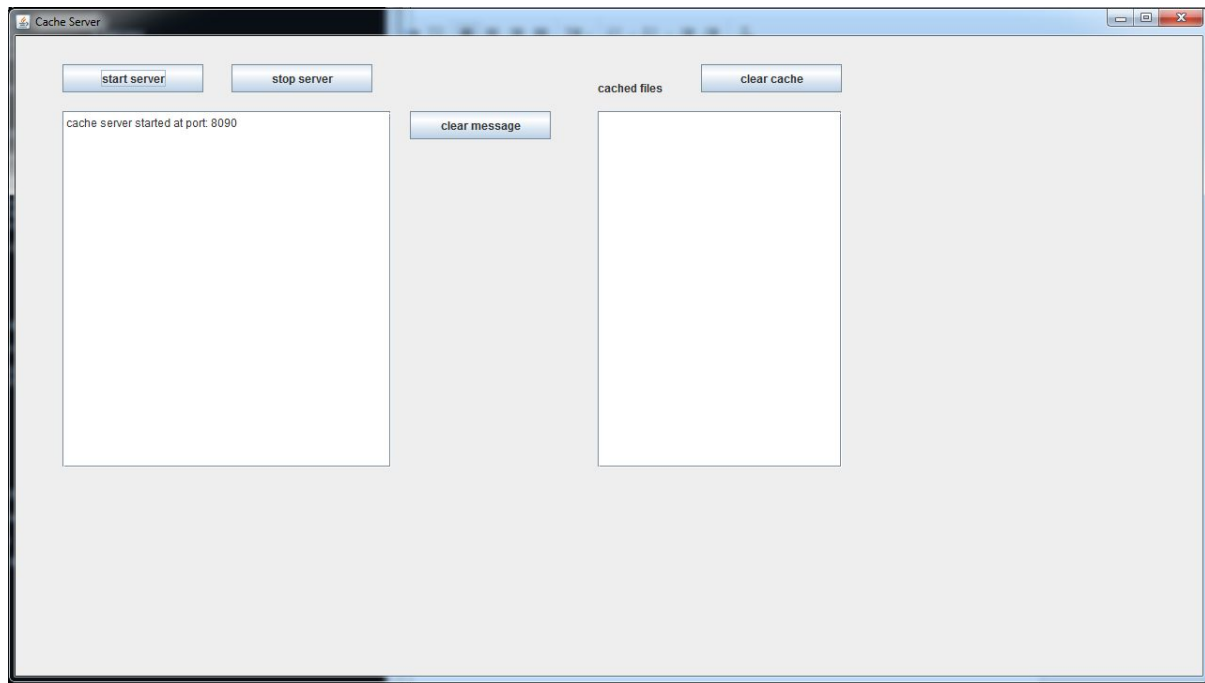
H:\Documents\tmp\ass1>cd bin
H:\Documents\tmp\ass1\bin>dir
Volume in drive H is MyHome
Volume Serial Number is FE3A-855E

Directory of H:\Documents\tmp\ass1\bin

07/09/2016  07:24 p.m.    <DIR>          .
07/09/2016  07:20 p.m.    <DIR>          ..
07/09/2016  07:23 p.m.             60 runCacheServerPart1.bat
03/09/2016  05:10 p.m.             53 runFileServer.bat
03/09/2016  04:23 p.m.             65 runFileClient.sh
03/09/2016  05:11 p.m.             53 runFileClient.bat
07/09/2016  07:21 p.m.             72 runCacheServerPart1.sh
07/09/2016  07:22 p.m.             72 runCacheServerPart2.sh
07/09/2016  07:24 p.m.             60 runCacheServerPart2.bat
03/09/2016  04:17 p.m.             65 runFileServer.sh
               8 File(s)              500 bytes
               2 Dir(s)  37,669,385,011,200 bytes free

H:\Documents\tmp\ass1\bin>runCacheServerPart1.bat_
```

7. The cache server will start GUI like below, click button "start Server", the cache server will be started.



2.1.3 Steps to deploy file client

1. Save A1.zip under one path of file client machine at IP: 192.168.1.30
2. Decompress A1.zip
3. cd ass1\conf Edit the file setting.conf as below

```
[fileClientConfig]
cacheServerIP = 192.168.1.20
cacheServerPort = 8090
clientFilePath = H:\Documents\tmp\clientpath
```

We should configure the three parameters according to real requirement.

4. Run script build.bat under ass1\ to compile the software.
5. cd bin (this step is necessary)
6. Run script runFileClient.bat under ass1\bin\ . The whole process is like below

```
C:\Windows\system32\cmd.exe

H:\Documents\tmp>cd ass1

H:\Documents\tmp\ass1>dir
Volume in drive H is MyHome
Volume Serial Number is FE3A-855E

Directory of H:\Documents\tmp\ass1

07/09/2016  07:20 p.m.    <DIR>          .
09/09/2016  03:42 p.m.    <DIR>          ..
09/09/2016  03:04 p.m.    <DIR>          classes
07/09/2016  07:24 p.m.    <DIR>          bin
03/09/2016  08:28 p.m.    <DIR>          conf
03/09/2016  04:06 p.m.           114 build.sh
03/09/2016  05:38 p.m.           61 clean.bat
03/09/2016  03:24 p.m.    <DIR>          src
03/09/2016  04:54 p.m.           193 build.bat
03/09/2016  05:05 p.m.           70 clean.sh
               4 File(s)              438 bytes
               6 Dir(s)  37,665,093,189,632 bytes free

H:\Documents\tmp\ass1>build.bat

H:\Documents\tmp\ass1>javac src/org/cache/code/*.java src/org/cache/code/excepti
on/*.java src/org/cache/code/model/*.java src/org/cache/code/ui/*.java src/org/c
ache/code/util/*.java -d classes
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

H:\Documents\tmp\ass1>echo "build success!"
"build success!"

H:\Documents\tmp\ass1>cd bin

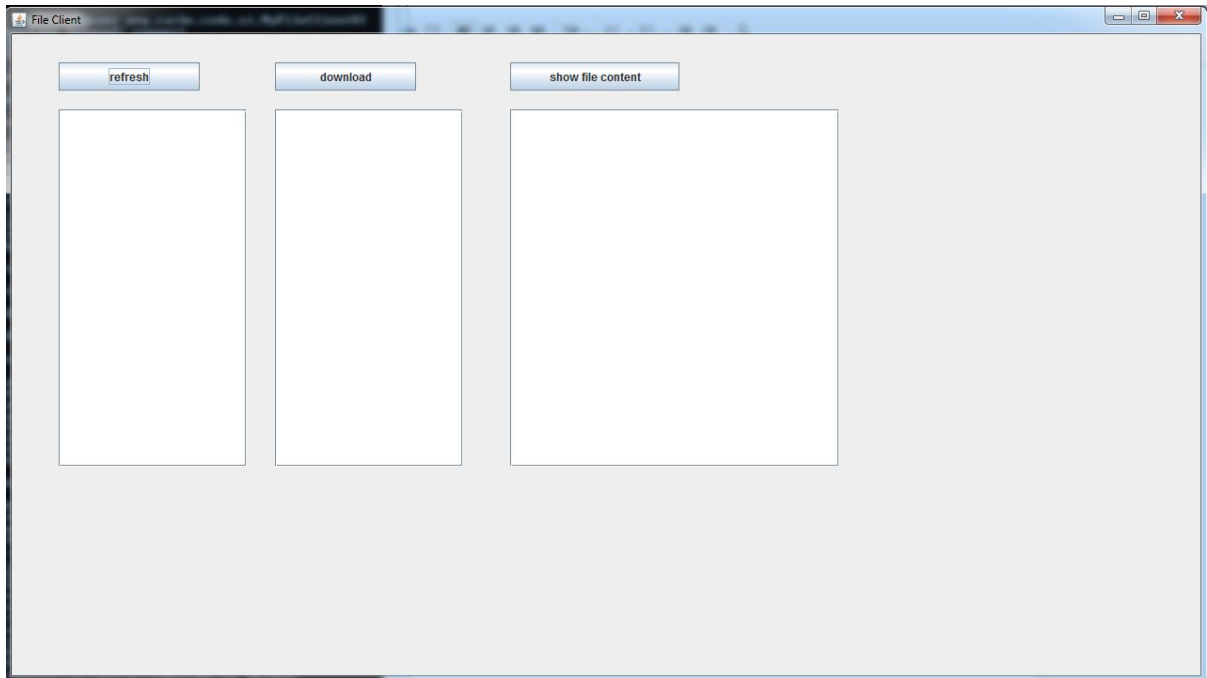
H:\Documents\tmp\ass1\bin>dir
Volume in drive H is MyHome
Volume Serial Number is FE3A-855E

Directory of H:\Documents\tmp\ass1\bin

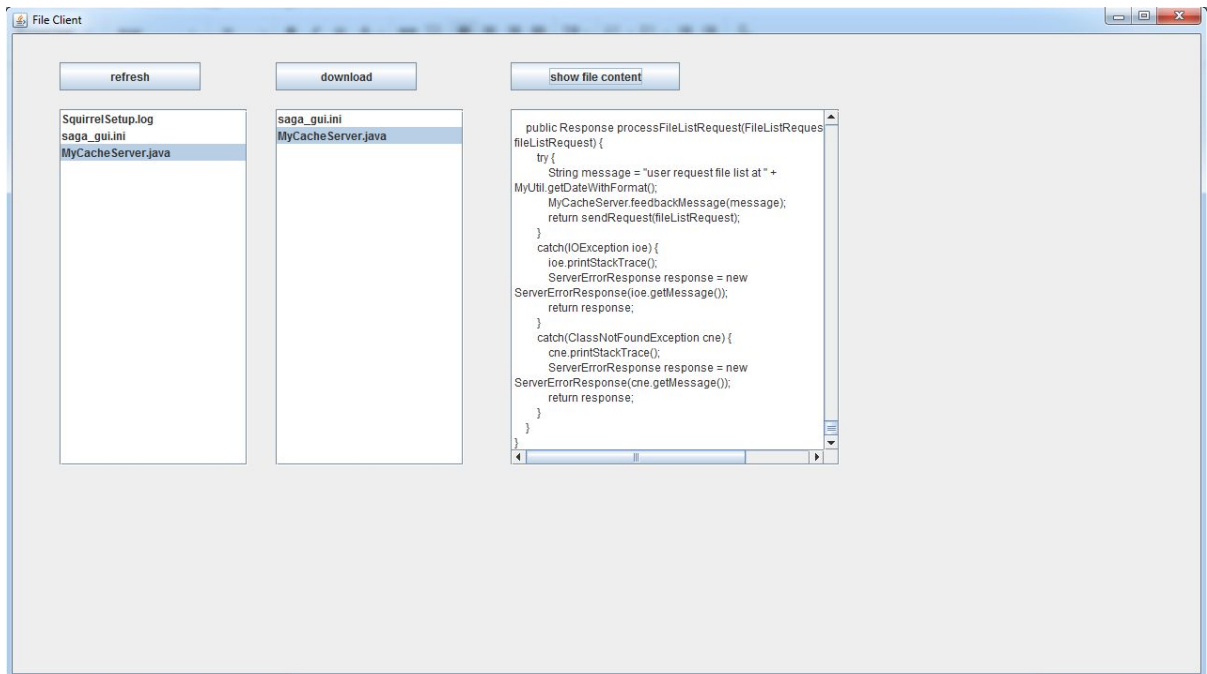
07/09/2016  07:24 p.m.    <DIR>          .
07/09/2016  07:20 p.m.    <DIR>          ..
07/09/2016  07:23 p.m.           60 runCacheServerPart1.bat
03/09/2016  05:10 p.m.           53 runFileServer.bat
03/09/2016  04:23 p.m.           65 runFileClient.sh
03/09/2016  05:11 p.m.           53 runFileClient.bat
07/09/2016  07:21 p.m.           72 runCacheServerPart1.sh
07/09/2016  07:22 p.m.           72 runCacheServerPart2.sh
07/09/2016  07:24 p.m.           60 runCacheServerPart2.bat
03/09/2016  04:17 p.m.           65 runFileServer.sh
               8 File(s)              500 bytes
               2 Dir(s)  37,665,093,189,632 bytes free

H:\Documents\tmp\ass1\bin>runFileClient.bat
```

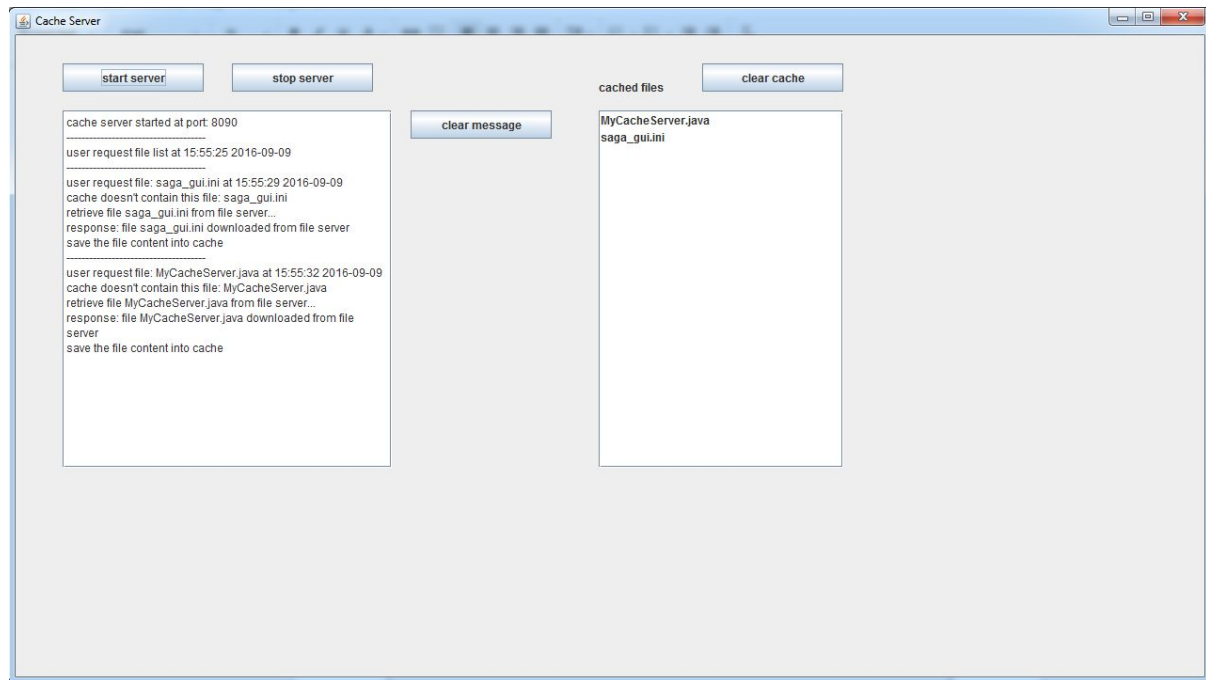
7. The client will start GUI like below.



8. click button "refresh", it will list the filenames of server side. Select one file, then click button "download", it will download this file from file server via cache server. Select one of the downloaded files, client button "show file content" the file content will be shown out.



9. Each time when we click download button, we can check the cache server's GUI, there will be new logs output.



2.2 deploy part 2

The deploy process is almost the same as part1, the only difference is when start the cache server, run script runCacheServerPart2.bat instead of runCacheServerPart1.bat.

2.3 deploy under linux

The deploy process is also the same as before, the only difference is to use .sh scripts instead of .bat scripts.

Chapter 3 Techniques description

3.1 cache mechanism of part1

3.1.1 cache data structure

Filename -----map to ----- md5 + file content (byte[])

Notes: the process to generate md5 of file is executed on file server side, not on cache server side.

3.1.2 the algorithm

1. Cache server receives a file request from client.
2. Check by filename whether the file is cached.
3. If the cache don't have the file, send a request to file server to download the file, the file server should return the file content and md5 of the file. Go to step 7.
4. If the cache has the file, send a request to file server to get the md5 of the file. File server only need to return md5 of the file. Compare the md5 with the cached md5.
5. If the md5 is not the same, that means the file has been changed, go back to step 3.
6. If the md5 is the same, that means the file has not been changed. Go to step 7.
7. Save the file content and md5 in cache, return the file content to client.

3.2 cache mechanism of part2

3.2.1 cache data structure

File name ----- *map to* ----- hash 1, hash 2, hash 3, ..., hash n (ordered hash value)
Hash 1 ----- *map to* ----- block 1 (byte[])
Hash 2 ----- *map to* ----- block 2 (byte[])
Hash 3 ----- *map to* ----- block 3 (byte[])
...
Hash n ----- *map to* ----- block n (byte[])

Notes: the process to generate blocks and hash values is executed on file server machine, not on cache server side.

3.2.2 algorithm

1. Cache server receives a file request from client.
2. Check if the file is in cache.
3. If the file is not in cache, send request to file server to get the whole file blocks. File server will split the file in blocks, each block with a hash value, return all this information to cache server. All the hash values will in a ordered list, which indicates the order of the blocks. Go to step 6.
4. if the file is in cache, send a request to file server to get the file's current hash values. File server will split the file in blocks, calculate all hash values of each block, only return the hash values in an ordered list back to cache server.
5. Cache server compare the ordered hash values with cached hash values, pick out those hash values which is not exist at cache side. If there are any new hash value, means the related blocks are changed or there are some new blocks. Cache server will send a new request to file server to get the blocks of the hash values which is not exist at cache side. File server will send back the needed blocks.
6. Cache server save all the blocks and hash values in cache side, construct the whole file content with the blocks according to the right ordered hash values, send back to client.

Chapter 4 comparison between part 1 and part 2

4.1 Theory comparison

Assume in part 2, a file will be splitted into N blocks. When client request the same file and there are M ($M \leq N$) blocks are changed, we can get a estimate comparison result as below

	Part 1	Part 2
Amount of computation on file server	1	$1+N+M$
Amount of computation on Cache server	1	N
Amount of computation on client	1	1
Bandwidth between cache and server	N	M
Bandwidth between client and cache	1	1
Response time at client side	$N+N$	$N+M$

The estimated value of the last one is based on the assumption that the distance between cache to server is the same as client to cache. If the distance between cache and server is largely greater than the distance between client and cache, then comparison of the response time at client side should be $N:M$

4.2 Test comparison

Each time client request a file, the console will print log about how long it takes to get the file. Based on log, we can get the running result of response time at client side.