COMPSCI711 A1 REPORT

Intrduction:

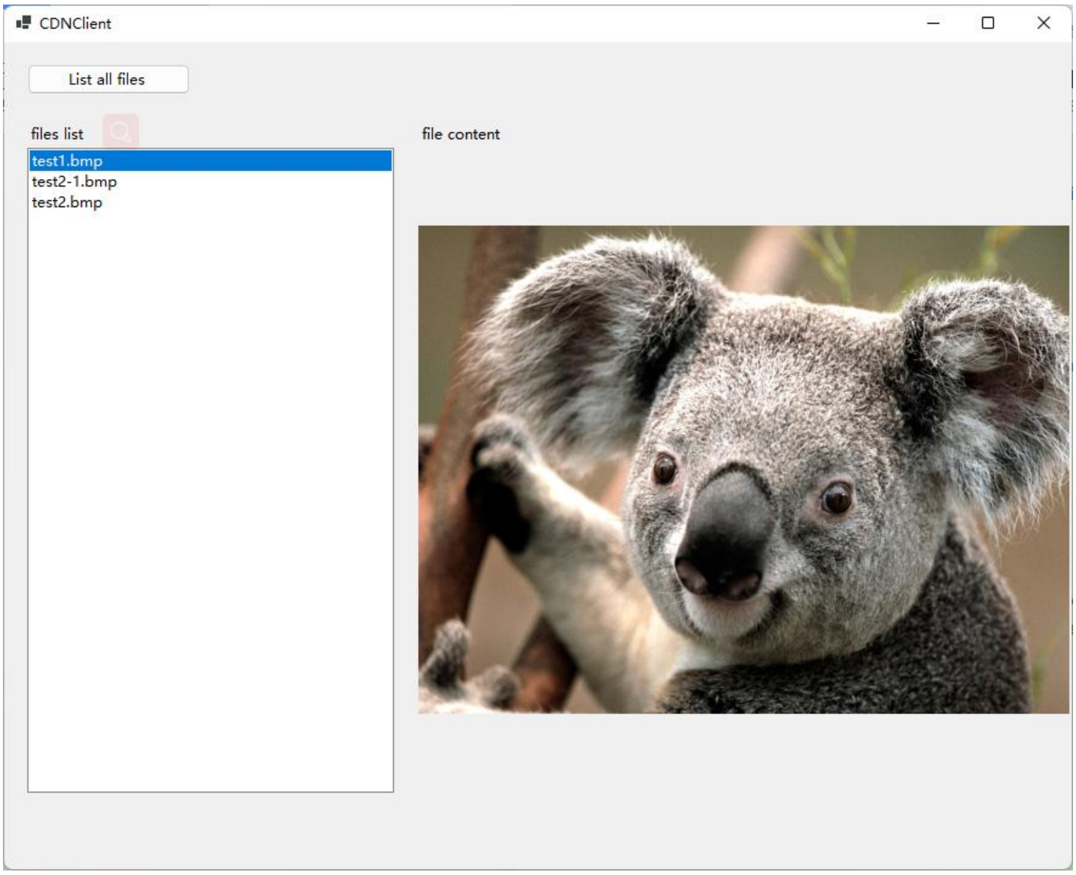
In this assignment I implemented the option2.And I have test it on a lab machine.\

Guidance for runing this program:

First you can find a batch file named strat.bat .Double click it to start the project program.
And there will display three form windows which called CDNClient,CDNCache,and CDNServer.

CDNClient

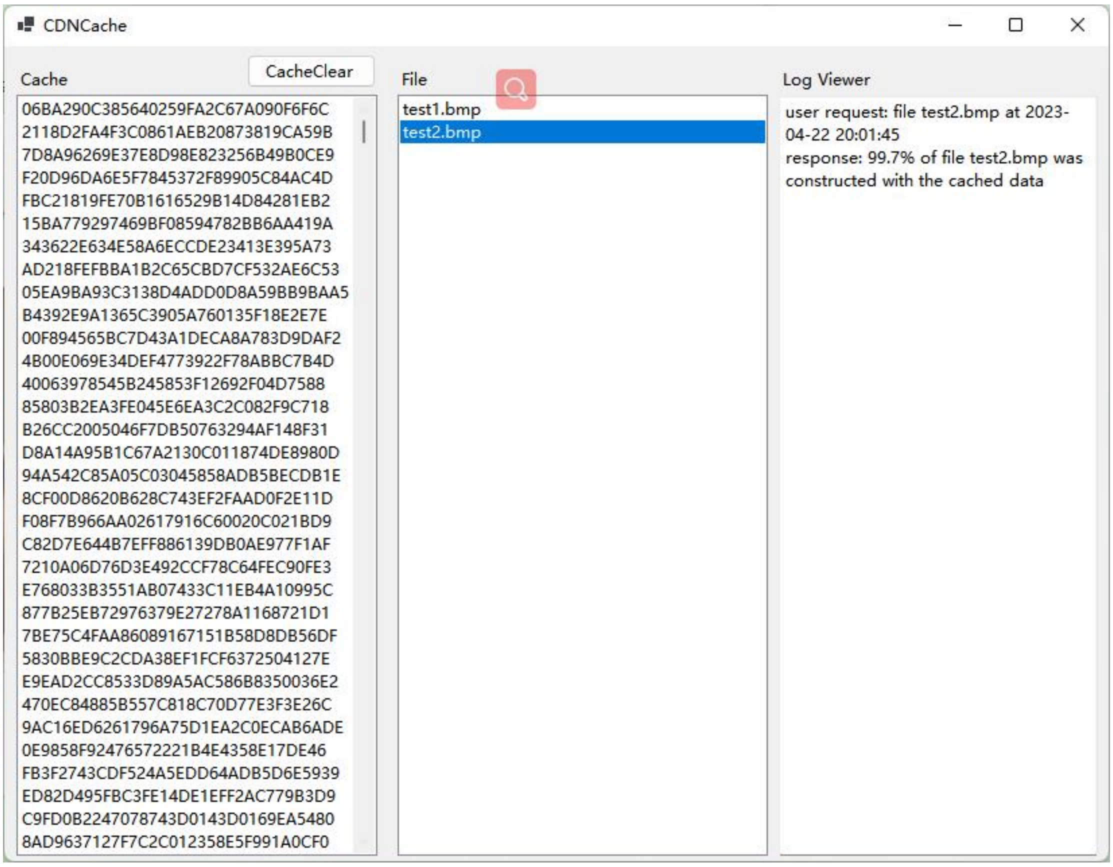
Click the "list all files",the client will display all files that are avilable on server.And double click the file name ,the picture file will display on the right side.



test1.bmp
test2-1.bmp
test2.bmp

CDNCache

The Cache side has three parts,Cache part shows the contents of the selected data fragment cached on the cache server.The file part shows the file list that requested by the client side.And couble click the file name to display the cache log .The log viwer shows the requested file name,the request time and the percent of the file that constructed by data fragments that stored on cache server.



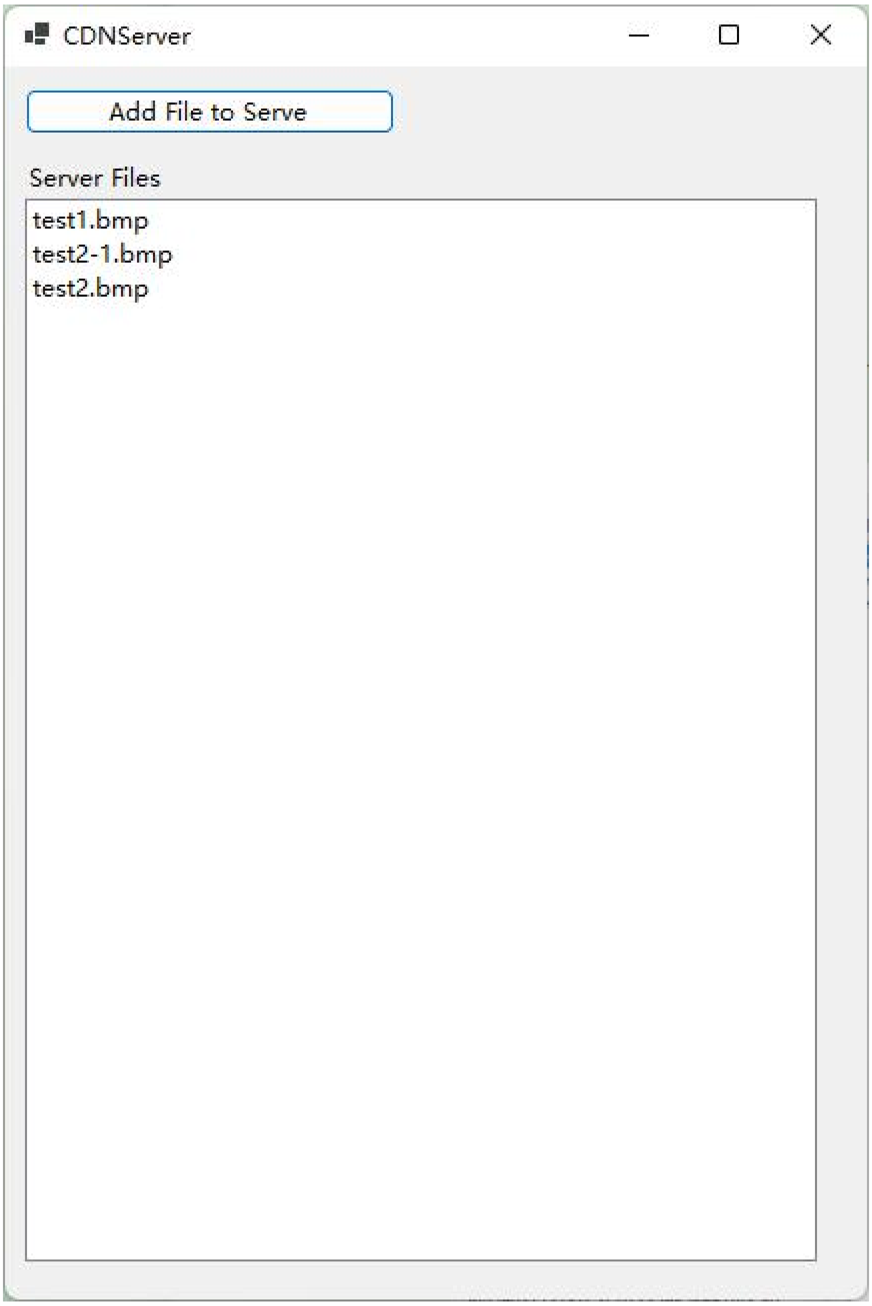
```
06BA290C385640259FA2C67A090F6F6C
2118D2FA4F3C0861AEB20873819CA59B
7D8A96269E37E8D98E823256B49B0CE9
F20D96DA6E5F7845372F89905C84AC4D
FBC21819FE70B1616529B14D84281EB2
15BA779297469BF08594782BB6AA419A
343622E634E58A6ECCDE23413E395A73
AD218FEFBBA1B2C65CBD7CF532AE6C53
05EA9BA93C3138D4ADD0D8A59BB9BAA5
B4392E9A1365C3905A760135F18E2E7E
00F894565BC7D43A1DECA8A783D9DAF2
4B00E069E34DEF4773922F78ABBC7B4D
40063978545B245853F12692F04D7588
85803B2EA3FE045E6EA3C2C082F9C718
B26CC2005046F7DB50763294AF148F31
D8A14A95B1C67A2130C011874DE8980D
94A542C85A05C03045858ADB5BECDB1E
8CF00D8620B628C743EF2FAAD0F2E11D
F08F7B966AA02617916C60020C021BD9
C82D7E644B7EFF886139DB0AE977F1AF
7210A06D76D3E492CCF78C64FEC90FE3
E768033B3551AB07433C11EB4A10995C
877B25EB72976379E27278A1168721D1
7BE75C4FAA86089167151B58D8DB56DF
5830BBE9C2CDA38EF1FCF6372504127E
E9EAD2CC8533D89A5AC586B8350036E2
470EC84885B557C818C70D77E3F3E26C
9AC16ED6261796A75D1EA2C0ECAB6ADE
0E9858F92476572221B4E4358E17DE46
FB3F2743CDF524A5EDD64ADB5D6E5939
ED82D495FBC3FE14DE1EFF2AC779B3D9
C9FD0B2247078743D0143D0169EA5480
8AD9637127F7C2C012358E5F991A0CF0
```

test1.bmp
test2.bmp

user request: file test2.bmp at 2023-04-22 20:01:45
response: 99.7% of file test2.bmp was constructed with the cached data

CDNServer

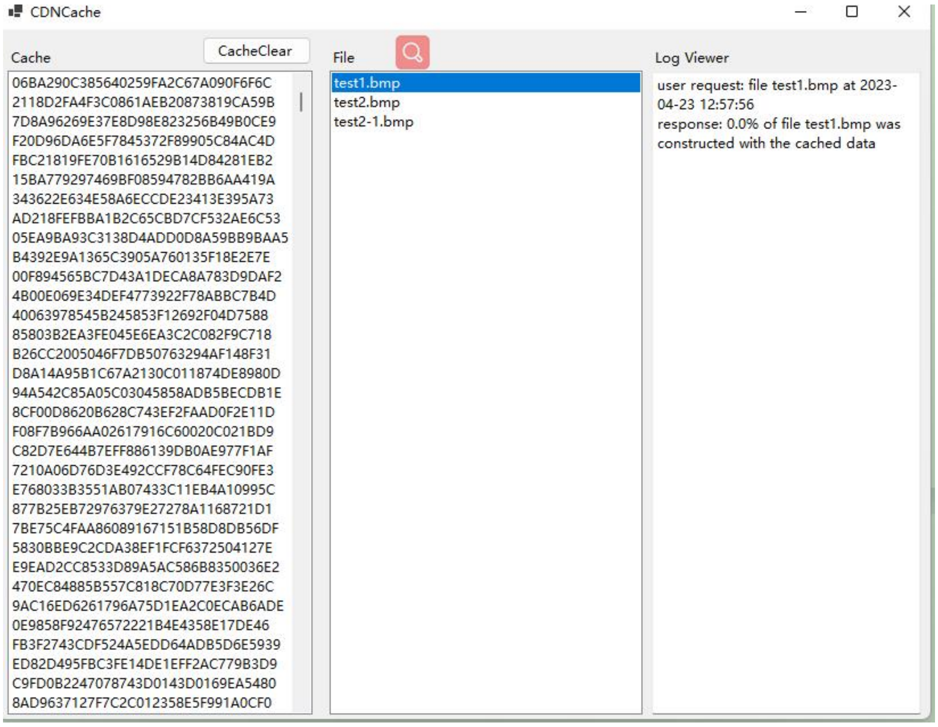
The server side displays all files that is available for user when the the server initialize.And the button “Add File to server” allows administrator to add file to server.



test1.bmp
test2-1.bmp
test2.bmp

Case test:

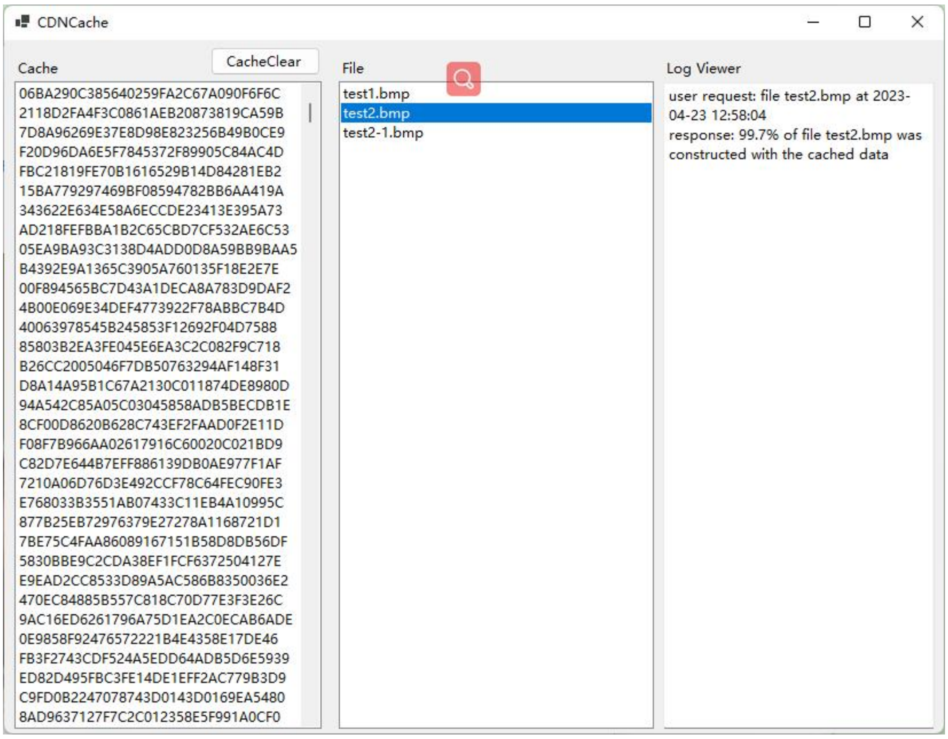
Case 1: Download files with different names but similar contents.



```
06BA290C385640259FA2C67A090F6F6C
2118D2FA4F3C0861AEB20873819CA59B
7D8A96269E37E8D98E823256B49B0CE9
F20D96DA6E5F7845372F89905C84AC4D
FBC21819FE70B1616529B14D84281EB2
15BA779297469BF08594782BB6AA419A
343622E634E58A6ECCDE23413E395A73
AD218FEFBBA1B2C65CBD7CF532AE6C53
05EA9BA93C3138D4ADD0D8A59BB9BAA5
B4392E9A1365C3905A760135F18E2E7E
00F894565BC7D43A1DECA8A783D9DAF2
4B00E069E34DEF4773922F78ABBC7B4D
40063978545B245853F12692F04D7588
85803B2EA3FE045E6EA3C2C082F9C718
B26CC2005046F7DB50763294AF148F31
D8A14A95B1C67A2130C011874DE8980D
94A542C85A05C03045858ADB5BECDB1E
8CF00D8620B628C743EF2FAAD0F2E11D
F08F7B966AA02617916C60020C021BD9
C82D7E644B7EFF886139DB0AE977F1AF
7210A06D76D3E492CCF78C64FEC90FE3
E768033B3551AB07433C11EB4A10995C
877B25EB72976379E27278A1168721D1
7BE75C4FAA86089167151B58D8DB56DF
5830BBE9C2CDA38EF1FCF6372504127E
E9EAD2CC8533D89A5AC586B8350036E2
470EC84885B557C818C70D77E3F3E26C
9AC16ED6261796A75D1EA2C0ECAB6ADE
0E9858F92476572221B4E4358E17DE46
FB3F2743CDF524A5EDD64ADB5D6E5939
ED82D495FBC3FE14DE1EFF2AC779B3D9
C9FD0B2247078743D0143D0169EA5480
8AD9637127F7C2C012358E5F991A0CF0
```

test1.bmp
test2.bmp
test2-1.bmp

```
user request: file test1.bmp at 2023-04-23 12:57:56
response: 0.0% of file test1.bmp was constructed with the cached data
```



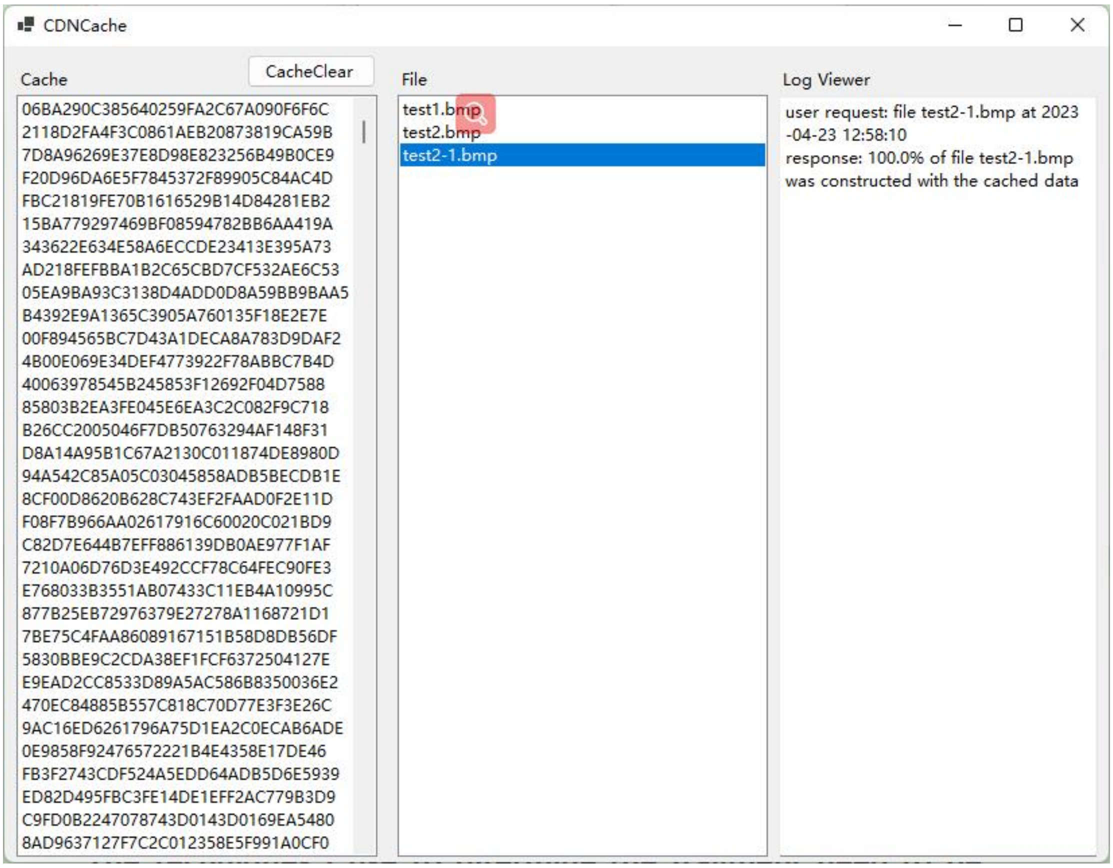
```
06BA290C385640259FA2C67A090F6F6C
2118D2FA4F3C0861AEB20873819CA59B
7D8A96269E37E8D98E823256B49B0CE9
F20D96DA6E5F7845372F89905C84AC4D
FBC21819FE70B1616529B14D84281EB2
15BA779297469BF08594782BB6AA419A
343622E634E58A6ECCDE23413E395A73
AD218FEFBBA1B2C65CBD7CF532AE6C53
05EA9BA93C3138D4ADD0D8A59BB9BAA5
B4392E9A1365C3905A760135F18E2E7E
00F894565BC7D43A1DECA8A783D9DAF2
4B00E069E34DEF4773922F78ABBC7B4D
40063978545B245853F12692F04D7588
85803B2EA3FE045E6EA3C2C082F9C718
B26CC2005046F7DB50763294AF148F31
D8A14A95B1C67A2130C011874DE8980D
94A542C85A05C03045858ADB5BECDB1E
8CF00D8620B628C743EF2FAAD0F2E11D
F08F7B966AA02617916C60020C021BD9
C82D7E644B7EFF886139DB0AE977F1AF
7210A06D76D3E492CCF78C64FEC90FE3
E768033B3551AB07433C11EB4A10995C
877B25EB72976379E27278A1168721D1
7BE75C4FAA86089167151B58D8DB56DF
5830BBE9C2CDA38EF1FCF6372504127E
E9EAD2CC8533D89A5AC586B8350036E2
470EC84885B557C818C70D77E3F3E26C
9AC16ED6261796A75D1EA2C0ECAB6ADE
0E9858F92476572221B4E4358E17DE46
FB3F2743CDF524A5EDD64ADB5D6E5939
ED82D495FBC3FE14DE1EFF2AC779B3D9
C9FD0B2247078743D0143D0169EA5480
8AD9637127F7C2C012358E5F991A0CF0
```

test1.bmp
test2.bmp
test2-1.bmp

```
user request: file test2.bmp at 2023-04-23 12:58:04
response: 99.7% of file test2.bmp was constructed with the cached data
```

We can see when we download the test2.bmp which has similar contents with test1.bmp. There is 99.7% contant is reused in cache.

Case 2: Download files with the same name but slightly different contents.



```
06BA290C385640259FA2C67A090F6F6C
2118D2FA4F3C0861AEB20873819CA59B
7D8A96269E37E8D98E823256B49B0CE9
F20D96DA6E5F7845372F89905C84AC4D
FBC21819FE70B1616529B14D84281EB2
15BA779297469BF08594782BB6AA419A
343622E634E58A6ECCDE23413E395A73
AD218FEFBBA1B2C65CBD7CF532AE6C53
05EA9BA93C3138D4ADD0D8A59BB9BAA5
B4392E9A1365C3905A760135F18E2E7E
00F894565BC7D43A1DECA8A783D9DAF2
4B00E069E34DEF4773922F78ABBC7B4D
40063978545B245853F12692F04D7588
85803B2EA3FE045E6EA3C2C082F9C718
B26CC2005046F7DB50763294AF148F31
D8A14A95B1C67A2130C011874DE8980D
94A542C85A05C03045858ADB5BECDB1E
8CF00D8620B628C743EF2FAAD0F2E11D
F08F7B966AA02617916C60020C021BD9
C82D7E644B7EFF886139DB0AE977F1AF
7210A06D76D3E492CCF78C64FEC90FE3
E768033B3551AB07433C11EB4A10995C
877B25EB72976379E27278A1168721D1
7BE75C4FAA86089167151B58D8DB56DF
5830BBE9C2CDA38EF1FCF6372504127E
E9EAD2CC8533D89A5AC586B8350036E2
470EC84885B557C818C70D77E3F3E26C
9AC16ED6261796A75D1EA2C0ECAB6ADE
0E9858F92476572221B4E4358E17DE46
FB3F2743CDF524A5EDD64ADB5D6E5939
ED82D495FBC3FE14DE1EFF2AC779B3D9
C9FD0B2247078743D0143D0169EA5480
8AD9637127F7C2C012358E5F991A0CF0
```

```
user request: file test2-1.bmp at 2023
-04-23 12:58:10
response: 100.0% of file test2-1.bmp
was constructed with the cached data
```

For this case, I rename a copy file of test2.bmp as test2-1.bmp to simulate this situation.And we can see that there is 100% contant is reused in cache.

The techniques I use to ditermine the fragment need to be download from the server:

I created two dictionary on cache side and server side as following code:

```
Dictionary<string, List<string>> FileCache = new Dictionary<string, List<string>>();
Dictionary<string, byte[]> FragmentCache = new Dictionary<string, byte[]>();
```

Create a new dictionary named FileCache :
The key is the filename, and the value is a list of md5 values for each block of the file. Used to store the md5 value of the corresponding file to facilitate the file content query for the following file block dictionary.

Create a new dictionary FragmentCache :

The key is the MD5 value corresponding to the block, and the value is the file contents (stored as a byte array). The downside is that the contents of the file are stored together, which works but is not a good solution.

For the server side, when the server is initialized and a new file is uploaded to the server, the file is splited into fragments by rabin function and stored in FileCache .And during spliting the file into fragment by rabin function,the server side will store the fragment into FragmentCache for future request.

For the cache side.When a download request is processed, the cache checks to see if the requested file exists in FileCache. If not, an ST request is sent to the server to get a list of MD5 values for the file requested. And store it in FileCache on the cache side. And then the cache iterates over the file's MD5 list in FileCache. Check whether the corresponding MD5 fingerprint of each fragment has a corresponding key/value pair in FragmentCache. If there is a fragment, it is stored directly in the buffer; if not, it sends an FR request to the server to download the corresponding fragment.

Comparison with option 1:

1.Response time perceived by user:
For the brand new files ,the option2 need more time than option 1 .Because it needs extra time to split the file.For the old file ,it takes the same time as option1.And for a brand new file which is similar to a old file,It takes shorter time.

2. Network bandwidth:
For the brand new file,the bandwidth will increas because cache needs to download the md5 list for every file.For the old file ,it takes the same bandwidth with option1..And for a brand new file which is similar to a old file,It takes fewer bandwidth because it only needs to dowload the fragment that isn’t stored on the cache side.

3. Server-side and cache-side computation
Since the server side is responsible for segmenting the file and calculating the md5 digest, the amount of computation on the server side is significantly increased.