Haoyang Yuan 15-640 / PIT
haoyangy 02/26/2016

Report for Project2

This report will describe the implementation of protocol between proxy and server, consistency model, LRU replacement, freshness technology, and system performance.

1. Protocol:

The server will contains several functions manipulations including create, remove, upload, download and get-file-information. The proxy will communicate with the server by RMI call of server instance. According to RMI, all the protocol message can directly be parameters of these functions.

2. Consistency Model:

I implemented this project with Check on Use propagation scheme.

a. Each time a new file is going to be open, it will first check if the file is on the server and proxy. It will try to download the cache file when the file is not existing on proxy yet or the version is older than server version.

b. **If the open call is for READ, it will read the file A1 as normal. Otherwise, the local proxy will fork a copy of the original file A1-1 to let user to create or write. During the write and seek process, all the manipulation will be done on the copy file A1-1. When the file closes, the copy file will be uploaded to server, and then be renamed as the new version of local file A2, and the be set to the cache if capacity allows. This scheme will make sure the server file will always be the newest.**

c.All the opened file will be recorded to avoid cache clean.

3.LRU policy

Generally, my LRU policy is the same as the standard LRU requirement. As mentioned previously, we will execute LRU when 1.a new open call comes to Check on Use, 2.a new copy be forked for create and write operation, 3.rename a edited file and update the cache size again.

Each time I execute the LRU cleaning process, I will first figure out how large size is needed for file A2. And then I will first remove all the unused old version file A(A1 for example) from cache(and deleted locally), as instructed by piazza@797. And I will iterate through the cache again to clean other files that is not used. Until any moment the size is large enough to store file A2, the iteration stops.

4.Freshness Technology

Due to the Check on Use scheme, all the file will always be the newest version at open process, otherwise new file will be downloaded to cache. And since I will upload the written file to server during the close process, this make sure the server will have the newest version even before the local proxy. Each time the server receives a update, it will

add the version number 1. Moreover, I use the local copy for the write process, this meets the requirement from handout that the client will have a fix view once open.

5.System Performance

According to my design, the proxy will request the file information from the server each time an open operation happens. And it will get new file only when there is no fit version file in local. And it will upload the file to server only when the local file is modified. I think the design is quiet system efficient. The only problem is that we will need to upload file when new file is created(maybe not write), the server will need to create it and write it in 2 RPCs. The other operations are as efficient as possible.