

Cache Design

name: Li Liu ID: LL2

Cache Protocol

There are five methods will be called by Proxy.

getFile is used to download file when needed, providing length to indicate how long to read, seek to indicate from where to read. And this function will return a byte array

createFile is used to create a new file on server

writeBack is used to write things to server, providing seek to indicate from where to write, over indicate whether write is over considering that one write can be divided to many chunks, and only when over is true the version of that file will increase. this function return the latest version number

unlink is used to delete file from server

getOpenInfo is used to get information package from server when open, including if the file is dir, if the file is exist, size of the file, version of the file. By compile all these info together I can reduce the latency by reducing communication times between server and proxy.

Consistency model

I use one file semantic, each writer will has there own copied file from the cached original one, and last close one will become the latest version. And the closed one will become the cached original file.

For each reader, once she/he open a file, whenever she/he read it, it will be the original opened version, which will not affected by other's operations.

LRU replacement

I use a double LinkedList to record the freshness of a file, and use a hash map to quickly find LinkedList Node by using filename. Every time if a file is used, its corresponding LinkedListNode will go the tail.

In two cases I will tried to clean memory, one is downloading file, the other is write over. First I will try to delete all file/node of old version of this file once nobody use it, so other useful cache will not be affected by them. then I will use LRU policy to clean memory until the remain size of cache is enough for the new file.