Design Document of File-Caching Proxy

Name: Xiangyu Sun
AndrewID: xiangyus

1. Proxy Design

Implement 6 file operation methods, including open, close, read, write, lseek and unlink.

1.1 Open-Close Semantics

In open method, synchronize files with the server to make sure the proxy has the latest cached files. Then during open-close session, we assume the cached files are the latest ones. Then in close method, we upload our modified versions to server. LRU are updated on Open method. In Close method, LRU is not updated.

All the interactions between proxy and server happens in Open method and Close method. All the other file operations are operated on cached files.

1.2 LRU Design

LRU Cache is implemented based on a HashMap and a LinkedList. While adding a cache file, if capacity exceeds, iterate all the nodes in LinkedList and evict the nodes that are not currently used to leave enough capacity for the new files.

1.3 Cache Freshness

I use a version number as an attribute of the proxy files and server files. In Open method, I compare the version number of local cached files to the version numbers of local cached files to make sure the cache is the freshest.

1.4 Reader and Writer

Readers share the same master copy. Writer keeps their own private copy to they can modify the files without other writers' interruptions. Private copies will be evicted in close methods as it will be no longer used in the future. Master Copy are recorded using a HashMap. In open method, if local master copy is no longer the newest, I check current copy is used or not. If it is not used, then evict this obsolete master copy.

2. Server Design

Server provides two major file operation for Proxy to use, UploadFile and DownloadFile. The server also use a HashMap to record the current file versions. UploadFile and Download File are implemented on chunks. So I also use a HashMap to save a RandomAccessFile instance so we can know where the file pointer is.

3. System Performance

3.1 Interaction between Server and Proxy

If cached files don't need to be updated, the proxy connects server only one time in open method to get the version number. If cached files needed to be updated, the proxy will connects server at least two times. One is to get the version number and the other one is to get the updated filed. And the number of interactions depends on how large the chunk is and how large the file is.

3.2 Time

All the methods in server are synchronized and only open and close methods in proxy are synchronized. Because all the clients that connects the same proxy will share a Server instance. The concurrent performance will be harmed which makes cost time longer. The system performance can be improved if we can make some methods in Server becomes not synchronized.

3.3 Memory

Downloading files from server and uploading files to server are implemented in chunk read and write. In this way, we don't need to consume too much memory when the file size is too large. But If we constrain the chunk size we will have more iteractions with server and this will harm the performance. We need to find an optimal chunk size and that varies in different situations.