Checkpoint Report: Cloud-Assited File Sync System

J. Jeyakumar, C. Muthappan, P. Rajput, L. Wen
May 26, 2017

1 PROJECT GOAL

In this project, we are going to implement a Dropbox-like file sync system based on cloud. On completion, this system is capable to provide file sharing services across multiple users and devices.

2 OVERALL DESIGN

The typical workflow of file synchronization using our system is illustrated by the diagram below (Fig. 2.1):

2.1 MASTER SERVER

We are going to implement a server side application which will handle all of the meta data. In our preliminary design, a database with columns of user_id, filename, checksum, src_ip, timestamp will be maintained on the master server. Each time a client edits a file, it should push the meta-information about the change to the server. Then the server stores the meta-data and push it to the other connected clients this file is shared with.

2.2 CLIENT DEVICE

On client devices, when connected to the server, every time it receives the meta-information of the changes from the server, it will send a request to the src-ip for the detailed change data. On receiving the data, the client application will apply the changes to the corresponding file,

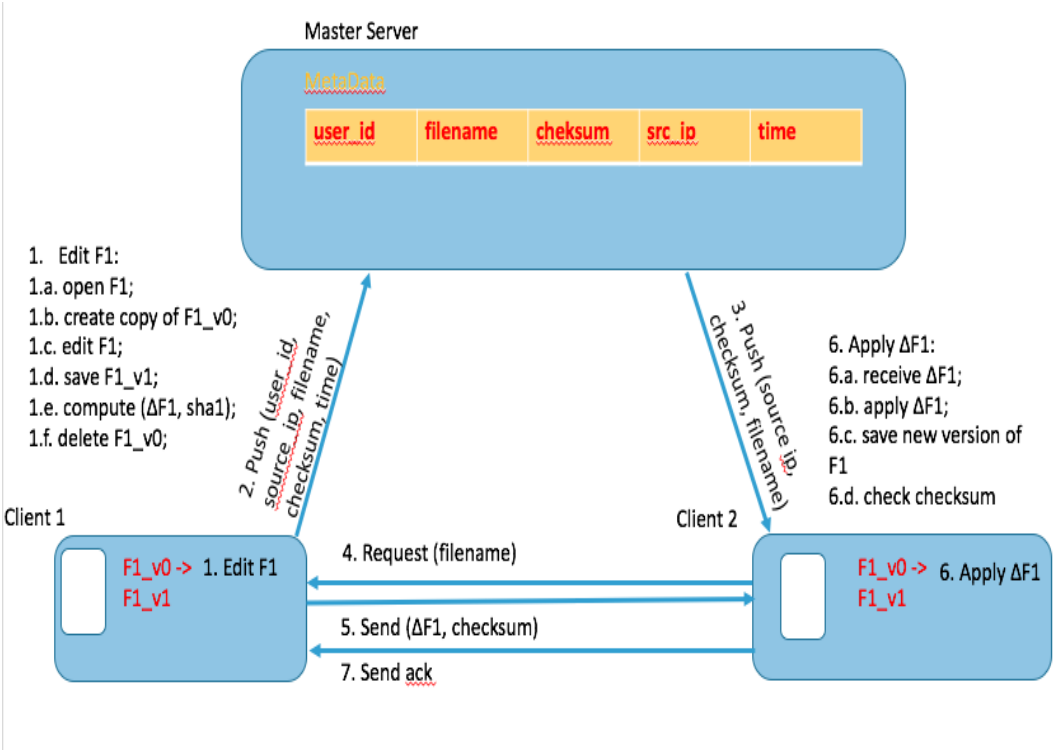

Figure 2.1: Overall Design and Typical Workflow

check the validity of the information by computing the checksum, if it is correct, the new version of the file will be saved and the client is supposed to send back the ack message; if not, the file will be rolled back to original version. After ack timeout, the client with src-ip re-sends the packet and the receipt will do this process one more time... This process will go on until the receiver succeeds to apply the correct modifications to the file.

3 What is achieved so far?

3.1 Server Side

3.2 Client Side

When a file that has to be synced is accessed in a device from any application, our system must be alerted of an incoming change. We achieved this functionality by making use of a python library called pyinotify. This package lets us watch for events that can occur in a directory. These events include many events like CREATE, MODIFY, CLOSE and DELETE. Our approach is very straight forward. When a file in the folder that has to be synced is open by any application in the client, we create a copy of the original version containing the data before modification. When the user closes the file after modification, we find the difference between the new modified file and the old original version.

4 Difficulties we are facing?

4.1 Server Side

4.2 Client Side

5 What are the next steps?

5.1 Server Side

- Lock Service

- Partitions
  Data storage partition.

5.2 Client Side