IIT DELHI

COP 290 : DESIGN PRACTICES IN COMPUTER SCIENCE

ASSIGNMENT 2 : MYDROPBOX

Design Document(Final)

Authors:

Anmol Sood - 2013MT60587

Kartikay Garg - 2013MT60600

Rahul Kumar Rathore - 2013MT60610

February 20,2015

Dropbox

0.1 Abstract

The objective of this project is to design a cloud storage software which stores user files on a server and provides access as and when requested by the user. The software will be efficient in file transfers, secure authentication and data transfer, inter-client file transfer and sharing. A persistant storage system with proper syncing has also been implemented from scratch.

0.2 Introduction

The application makes use of TCP/IP sockets at the client side and server side and uses OpenSSL to securely send data and authenticate clients to efficiently implement the functionality provided by various cloud storage providers like "**Dropbox**". Functionalities to share files and download the files that have been shared to a user are also provided. A sync button has been provided to sync the client folder with the server folder and this has been down keeping persistent storage in mind.

0.3 Setup and Software Used

Environment : **Ubuntu 14.04**
Language : **C++**
GUI : **QT**
Communication : **TCP/IP Sockets**
Security : **OpenSSL**
DBMS : **SQLite3**
Debugger : **gdb**
Documentation : **LaTeX**

0.4 Overall Design

On the client side ,the Desktop application equips the user with the ability to upload files onto the storage server and access them when a secured connection with the server is established The user can upload files/directories using the GUI.. The application when launched for the first time on a device will generate a directory in the Home folder of the user device. The application while running, will sync the latest files from the server onto the folder or in the reverse direction depending on the situation. This is done keeping in mind that there is no extra transfer of data apart from that which is completely necessary. A timestamp based synchronization technique has been implemented which has been discussed in detail further in the Synch

The user has the liberty to share access rights with other users using this application so that they can also access the files of the first user.

0.5 Sub Components

1. **File transfer**
2. **Sharing**
3. **Syncing**
4. **Security**

0.5.1 File Transfer

To transfer files between client and the user, the file is divided into chunks of memory which are then transferred over the secured connection between the client and the users. The chunks of file

created are encrypted using SSL and then transferred over the connection to prevent any kind of data steal over the connection. The user has the option to upload files using the GUI and sync them with the server as and when he wants.

0.5.2 Sharing

- Users will be able to control the sharing permissions associated with each file/directory using the GUI of the this application. The user clicks on the share button and enters the name of the user with which he wants to share the file.
- To implement this, a SQL database(using **sqllite**) is associated with each user using this application. This database, for each file, stores the information about the others users who have been shared this file by the owner. This database also stores the information about the files that have been shared with the current user.
- All the files shared with the current user appear in the **Shared** section in the GUI of the application when launched by the user. The user can download the shared files directly from the interface and they will appear in the **Shared** section in the Home Dropbox folder of the user.

0.5.3 Syncing

Syncing forms an integral part of any cloud storage software and must function efficiently so that the account of a user remains up to date at both the client as well as the server end. We have used a timestamp based synchronization and implemented this in the following manner :

- Whenever the user tells the application to sync the storage at the client and the server end, the client sends over a database containing the information about the files present at the server along with the information about when the client device synced with the server with the current username.
- The database contains information about the files as their name, their path, when they were last modified, whether a given path depicts a directory or a file at the client end.
- The server on receiving this database as input, runs several SQL queries over the input database and the database stored at the server end corresponding to the current user.
- These queries (using suitable case analysis) can determine the fate of a file, that if this file will be deleted on the client, uploaded to the server from the client, deleted on the server or downloaded from the server to the client. Timestamps of files along with last syncing time of server/client's computer is used to handle all cases.
- The user has an option to manually download the files that are shared to the current user by other users.

0.5.4 Persistent Storage

A perfectly persistent design has not been implemented. Also however whenever the connection is made,any files that have been completely transferred are considered syncronized and will not be transferred again on the next sync unless they are modified.

0.5.5 Security

We use OpenSSL to securely transfer data through sockets.
For client authentication , the OpenSSL implmentation of the SHA-1 algorithm is used to hash the passwords and these passwords are stored on the server. We plan to salt the stored passwords to make them unbreakable against well known offline attacks like the rainbow attack. Also the server identity is checked through certificates.