Problem Statement

Design Dropbox or Google drive.

It is Remote cloud storage

Adavatages of storing data in a remote server

1)Easy to share

2)High availability

3)Scalability (since horizontally scalable)

4)Relaible (In hard disk or pendrive data might get easily corrupted) since these devices are fault tolerant and distributed.

# Features

Functional Requirements (F.R)

User should be able to upload the files

User should be able to download the files

Provide the facility of creating the local workspace folder

All the devices connected to the same dropbox account should be in sync (let us say from the laptop and mobile)

I should be able to work in the offline mode (when we have the local storage)

Non-Functional Requirements (N.F.R)

1)Scalable

2)Expect large traffic for read and write

3)Read and Write traffic is comparable

4)Try to optimize on the amount of data transferred between client and server during each sync operation. (Only the changes should get updated)

5)We can optimize the storage by (avoiding the duplicates)

# Estimations

Let say total users 1B, and active uses 20% of it i.e 200 M

Each user connects from 3 different devices (mobile, tablet, laptop)

Average each user stores = 200 files

Size of each file = 100 kb

Total space = 1 B x 200 x 100 kb = 20 Petabytes PB

Which is huge data, we should store it in the distributed storage.

# Highlevel Design

1)Anything uploaded in the local folder should get uploaded to remote.

Client Application: will transfer to remoter server

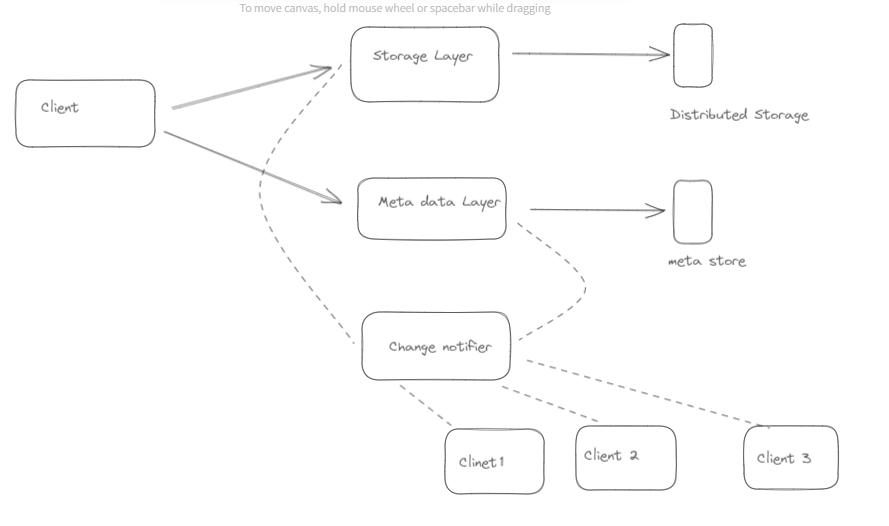
Storage Layer : wher the data is stored

2)For every file created, we can set the sharing permissions (private, public, or share to specific people etc) This can be maintained in a meta data.

Meta data storage Layer : Meta information is stored.

3)Every time there is a change this change should get reflected on all the machines connected to the same account if connected to ther internet.

Change Notifier: Notifies every change in storage layer/meta store across all the clients to be in sync.



Client :

i)Upload file

ii)Download file

iii)Detect any changes and upload to remote. (We should optimize the file transfer)

Let say every time a file ‘f’ is changed if we transfer the whole file, it is not optimal. we will be lossing a lot of time, space while transferring and also the network bandwidth.

The optimization we can do is the from the client side try to break the file into smaller chunks or blocks.

So that client should be able to figure it out which chunk(s) of the file got changed.

Client should only transfer the small diff/change.

Storage Layer

We have to store 20 PB of data, for which we need highly distributed storage like HDFS, GFS

The problem with that is the read becomes very slow, so to improve the read performance we can use the cache.

Metadata Storage Layer

Used to store info like permissions, size of the chunk, what are the devices connected, what are different local workspace folders.

This layer should be highly consistent.

We need to use RDBMS is most suited for the ACID properties.

Change notifier service

This service should keep watching for the changes the client makes to the Storage layer / Meta storage layer and as soon as it finds any change the duty of this service should be to update all the other clients.

To make this system scalable we can use messaging system for asynchronous communicaton system.

As and when the client makes some change to the Storage layer / Meta storage,it is put in the MQ,change notifier is subscribed to listen to this.

Notifier also maintains another MQ for the other clients to sync those changes.

