Minor Project Report

on

MineCloud

Submitted by

Digvijay Bahree	110600
Amit Yadav	110269
Akash Gupta	110250

(Students of B. Tech 8th semester Computer Engineering)

Under the Guidance of
Rajeev Mohan Sharma
(Professor, Department of Computer Engineering)

Department of Computer Engineering
National Institute of Technology, Kurukshetra

Nov., 2013

Problem Statement:

Data loss on a personal computer due to hard-disk failure or any other reason costs high for a user. Therefore to avoid this, we usually backup our data on external drive and maintain a consistent backup time to time. This creates an extra overhead for a user.

This creates a need of software to remove the overhead of data backup by using a costly hard drive. This software removes these limitations from our present day systems.

Motivation:

We observe that there are a variety of vendors available which provide free cloud storage and file synchronization facilities to a normal user. Each of these vendors provides storage in small volumes as tabulated below:

1. Dropbox: 2GB upgradable up to 20GB
2. Google Drive: 15GB
3. Amazon Cloud Drive: 5GB
4. Microsoft SkyDrive: 7GB
5. Box: 10GB
6. Copy: 15GB
7. Wuala: 5GB
8. Cubby: 5GB
9. Go Aruna: 2GB
10. SpiderOak: 2GB
11. Ubuntu one: 5GB
12. mimedia: 7GB
13. symform:10GB

Total: 90GB and this list isn’t exhaustive.

Proverb “Union is Strength” fits best in this scenario i.e. we could have something that can unite each of these small volumes of storage to create an abstraction level that adds them and we have a cloud storage of order of 100GB available to our disposal, without giving a thought to file size or vendor.

Description:

MineCloud uses the free cloud storage provided by vendor (Google Drive) to backup the data of hard-disk. Basically software creates a folder on desktop and if you put any file/folder in it then it automatically syncs the data between your system and backend (cloud).

To maintain a backup of say 150GB: software uses 10 accounts of google drive storage. MineCloud maintains a variable for each of the drive accounts to determine available memory to provide such an abstraction level.

Therefore, if storage for current account overflows; then it splits the file according to size constraints, maintain a mapping in a database and store the splitted ones in different accounts at back-end (cloud).

Conclusion and Future Plan:

Thus with this we *conclude* that this application will allow users to seamlessly utilize different cloud storage resources available and provide a common window to utilize the same. All this effort is a step forward towards the common goal of cloud dominance in which we intend to pose a challenge to procure actual hardware (specifically hard drives).

The *future plan* is to extend this commutative storage to different platform like mobile, tablets.

References:

https://developers.google.com/drive/v2/reference/

msdn.microsoft.com/en-us/live/dn195579.aspx

https://www.wikipedia.org

http://www.top10cloudstorage.com