Maximizing the Storage Capacity of Free Cloud Storage Providers

Marisa M. Buctuanon, Kristine Joanne I.Tigue and Ralph Jay Pepito

Abstract

CThis study aims to resolve the problem on the free cloud storage services. With innumerable providers, data and files are dispersed across multiple cloud repositories. Users try to have multiple accounts on the same cloud storage provider or to sign up for an account from different providers. Hence, Fuse creates an illusion of a larger capacity of data volume that will be made available for free for public consumption. With Fuse, files are consolidated and managed in a sophisticated manner to ensure data preservation and performance efficiency. This centralized cloud service is made to utilize today's emerging cloud services such as Google Drive, Dropbox, and Box. With 1 account, users will be able to store files online and access them anywhere with larger storage capacity than what is available today.

Keywords: Abstracted cloud storage, data warehousing, centralized cloud service, free larger storage capacity

1.0 Introduction

Cloud storage has become 1 of the forefront solutions for the need of storage in lieu of the physical alternative which is more prone to data deprivation and data loss [11] [13]. It has risen in popularity among users who are in need of abundant, convenient, and secure solutions for online file storage. A lot of people are working on big size of files, such as audio, video, installer, large PDF or image file for the purpose of their work, business or personal need. This sprang the proliferation of plentiful cloud storage services and most users avail the free cloud storage services [14]. Since the service is free, a constraint of cloud storage capacity is expected. Though, there are freebies or other promos that the service provider gives to have additional storage capacity, it is still not enough for the users to store their data. As a result, most users settle for having multiple accounts on the same cloud storage provider or to sign up for an account from different providers. This creates an intersperse of files or data across multiple cloud repositories. This will somehow add up in the workload of a user in accessing files from different accounts.

There are 19 free cloud storage services that are accessible nowadays [15]. To name a few, there is MEGA which offers 50 GB of free cloud storage, pCloud with 10 GB of free online storage, and MediaFire where you can instantly get 10 GB of free online file hosting [16]. However, they are not that commonly used by the community. Google Drive, Dropbox, and Box are included in the top 10 services that offers great services for storing and sharing files with friends and families [16]. A Google user gets a 15 GB of free space that is shared with all Google services. Dropbox users start off with 2 GB of free space and can earn up to around 18 GB with simple additional requirements. Box, which is formerly Box.net, gives out a 10GB of free online storage space.

To address the problem of managing multiple cloud services, cloud storage managers have emerged. One of which is, MultCloud. This application supports more than 20 cloud drives [16]. It is a free web based app that allows cloud users to access all their online files from a single interface. The users are given the option to choose which files to transfer to another drive. This application is the same with ODrive. ODrive is a free desktop application created by Oxygen Cloud. It gives users a folder interface for browsing all their online files and syncs their chosen files, documents and photos [7] [8]. Likewise, Otixo also lets users combine and manage files on several cloud storage services using a single dashboard [9]. Unlike these cloud storage managers, Fuse automatically manages the files in the 3 drives. Additionally, Redundant Array of Cloud Storage (RACS) is another cloud storage manager that stripes data across multiple cloud storage providers [1]. The supported cloud services of RACS are for business applications like Rackspace Cloud which has 5GB storage capacity, Amazon S3 web service with 5GB storage capacity and GoGrid with 10 GB storage capacity. In total, Fuse has 25 GB storage capacity which is larger than RACS. Unlike RACS, Fuse implements the abstraction of file management using Round-robin instead of RAID. All of these multiple cloud services can be accessed via web. All these multiple cloud services can be accessed via web

Fuse aims to give light to the problem of managing multiple free cloud storage services. This application consolidates the 3 prominent cloud storage providers, namely, Google Drive, Dropbox, and Box. This makes Fuse to have larger storage capacity than what is accessible today. Choosing which drive to store the file, is abstracted. An illusion of 1 cloud service provider is offered to the users. With the use of Round-robin algorithm, the files are equally stored in each drive in a circular order.

Conceptual Framework

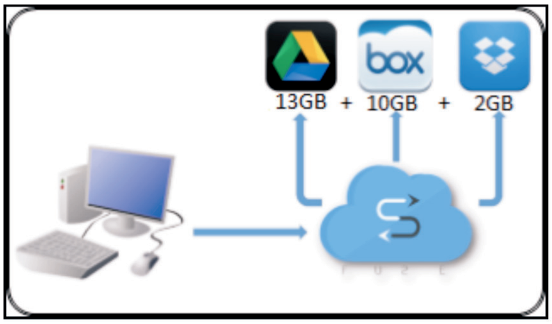


***Figure 1:** Fuse Conceptual Framework*

Figure 1 depicts the interaction of the system with the user and the APIs and SDKs that are being used by the system. The user of the application will only have a single login credential to be able to use Fuse. Upon registration, the user will be asked for the login credentials of the 3 drives. This will allow the system to manage the files in each drive. If the user doesn’t have an account in each drive, the user can create 1 and associate each account to the system. Once the user finishes the authentication process, the user will then be redirected to the *Fuse* dashboard or home page. The user can now manage his files.

2.0 Methodology

Fuse is a web application that was developed in October 2015 using PHP scripting language. Using the different APIs and SDKs, the system was able to connect to the cloud service providers. Choosing which cloud storage services are to be included in the system is based on the availability of APIs and SDKs, maximum free storage capacity, and their popularity or total number of subscribers. Before the developer can call or use Google Drive API, Box API, and Dropbox SDK, the developers should create an account for each drive. Google Drive API is organized by resource type and can be accessed on the Google Drive developers' reference [3] [4] [10] [12]. Although generally, Google Drive offers 15GB of storage, it can only allow Fuse to store 13GB of memory storage. This is because there are other services of Google Drive that are consuming the storage capacity. Managing the request from Box can be known from using Box developers' site [2] [5] [6]. While accessing and manipulating the Dropbox account, the developers need to install Dropbox SDK [9].

Fuse Management of Files

Round-Robin Algorithm

Round-robin is the scheduling algorithm used in distributing the files and directories across multiple cloud platforms. Round-robin was implemented so that every cloud storage would equally share the same number of files stored in *Fuse*. The circular queue of the algorithm will depend on the sequence of authentication that the user did during the registration process and the subsequent authentications. There is a scheduler in the system that caches the information on which cloud service provider the last upload happens. This will allow the system to know on which cloud platform to resume the distribution.

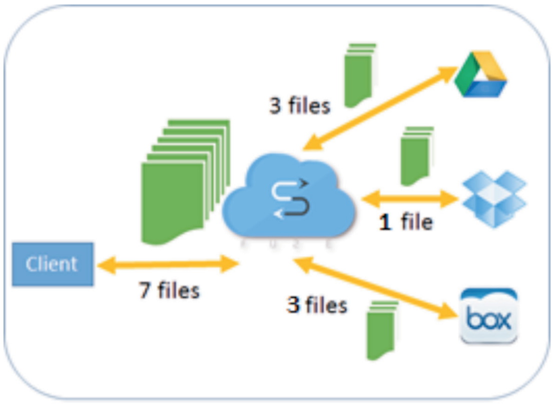


Figure 2: *Fuse Management of Files*

Figure 2 shows the internal representation of how files are distributed and accessed to/from each drive. Fuse distributes these 7 files to the 3 drives upon uploading these to the system. During the uploading of files, the system determines the number of files to be uploaded and computes the total number of files to be uploaded in a drive. Table 1 shows the percentage of uploaded files that can be stored in a particular drive.

Table 1: *Drive Files Allocation*

Cloud Service Provider	Storage Capacity	Percentage of Files to be Stored
Google Drive	13 GB	52%
Box	10 GB	40%
Dropbox	2 GB	8%

Since not all cloud storage providers have the same initial storage capacity, the system is designed to have a ratio approach of distributing the files. 52% of the files will go to Google Drive, 40% to Box, and 8% to Dropbox. If the user uploads 7 files, 3 files will go to Google Drive, 3 files will go to Box, and the 2 remaining files will go to Dropbox. The Round-robin algorithm is used to know the sequence on which drive to store the files. The order of the sequence starts with Google Drive, followed by Dropbox, then Box. Assuming that Box was the last drive where the recently added files are stored, the system thendirects the storage to Google Drive. As a result, 3 files goes to Google Drive, 2 files to Dropbox and Box. The system stores the information on which drive do these files are stored for future access.

3.0 Results And Discussion

User Interfaces

Cloud Authentication

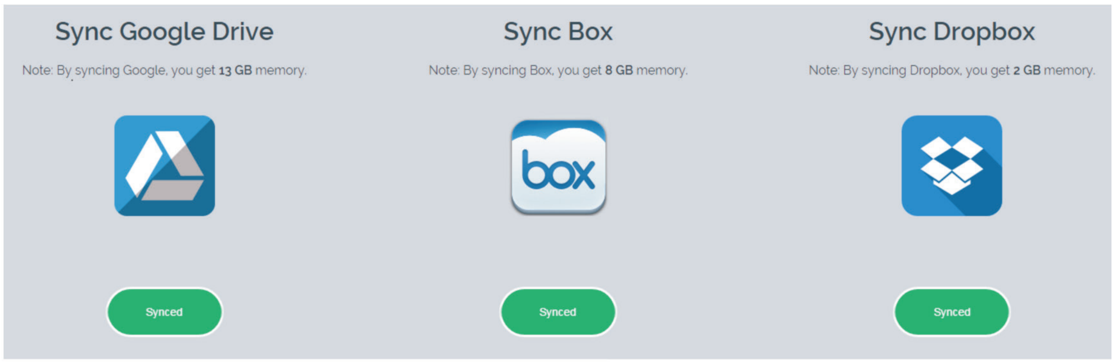
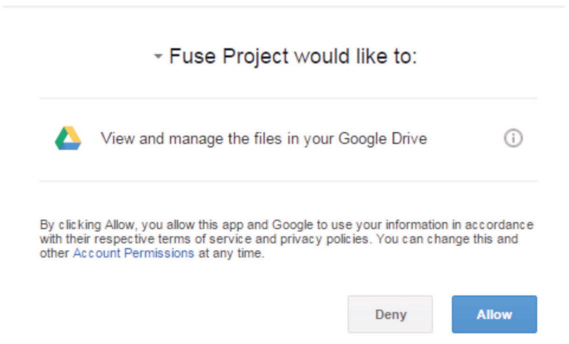

Sync Google Drive
Note: By syncing Google, you get 13 GB memory.
Synced
Sync Box
Note: By syncing Box, you get 8 GB memory.
box
Synced
Sync Dropbox
Note: By syncing Dropbox, you get 2 GB memory.
Synced

Figure 3: *Google Drive Authentication*

Figure 3 shows an example of Google Drive authentication. After the user has logged-in to his Google Drive account, the system redirects the user to the authentication page. The user needs to press the button "Allow" in order for the system to view and manage his files. This process is the same with the other 2 cloud storage providers.

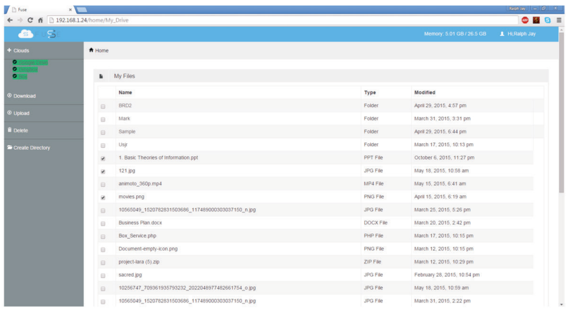

Fuse Project would like to:

View and manage the files in your Google Drive

By clicking Allow, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies. You can change this and other Account Permissions at any time.

Figure 4: *Three Synced Drives*

Figure 4 displays that the 3 drives have been synced to Fuse after authentication process. This would mean that Fuse can now manage the user's 3 accounts.



Memory: 5.01 GB / 26.5 GB

Hi,Ralph Jay

My Files

	Name	Type	Modified
☐	BRD2	Folder	April 29, 2015, 4:57 pm
☐	Mark	Folder	March 31, 2015, 3:31 pm
☐	Sample	Folder	April 29, 2015, 6:44 pm
☐	Usjr	Folder	March 17, 2015, 10:13 pm
✓	1. Basic Theories of Information.ppt	PPT File	October 6, 2015, 11:27 pm
✓	121.jpg	JPG File	May 18, 2015, 10:58 am
☐	animoto_360p.mp4	MP4 File	May 15, 2015, 6:41 am
✓	movies.png	PNG File	April 15, 2015, 6:19 am
☐	10565049_1520782831503686_117489000303037150_n.jpg	JPG File	March 25, 2015, 5:26 pm
☐	Business Plan.docx	DOCX File	March 20, 2015, 2:42 pm
☐	Box_Service.php	PHP File	March 17, 2015, 10:15 pm
☐	Document-empty-icon.png	PNG File	March 12, 2015, 10:15 pm
☐	project-lara (5).zip	ZIP File	March 12, 2015, 10:29 pm
☐	sacred.jpg	JPG File	February 28, 2015, 10:54 pm
☐	10256747_709361935793232_2022048977482661754_o.jpg	JPG File	May 18, 2015, 10:59 am
☐	10565049_1520782831503686_117489000303037150_n.jpg	JPG File	March 31, 2015, 2:22 pm

Figure 5: *Fuse Dashboard*

Figure 5 is the main view of the user once successfully log-ins to the system. On the left side pane, the user will see that the 3 drives are highlighted. This indicates that the 3 drives have successfully synced. The user can create a file directory that can also contain a subdirectory. Upon uploading of files, the user can traverse the directory as to which location the file should be. The user can remove the files by just selecting the desired files and clicking the "Delete" icon. Once the user decided to download the files, he just needs to select the directory or the files and click the "Download" button. At the upper right corner of the screen, the user can know the available storage capacity of his account. Once the system requests for file access such as viewing, the other end, cloud storage providers, will send a JSON response. This response is parsed by the system before showing the result to the user. There is no physical storage of files in Fuse server, only in the 3 drives. The way the files are shown to the user are just through the HTTP request and response data communication. Furthermore, Fuse does not expose the synched email address of the user for security purposes.

Features

Extended Memory

The key feature of Fuse is its larger memory capacity provided for the users which has a total of 25 GB. This is due to the implementation of Fuse abstraction for file storage management. The system creates a folder in each drive for the system to know which files to manage. Since there are files that are created in a certain drive, without the use of Fuse, only those files that are part of that folder is managed.

Auto Sync Files

Visually hidden from the user, whenever the user performs a task in any of the Fuse file, there is an automatic syncing of files from Fuse to a certain drive. With the use of user's provided credentials, Fuse receives a short lived access token. For every user request, a single access token will be attached. Each drive will determine if the request and the token is valid and returns a JSON response. If valid, it will return a File resource, otherwise, an error message. In case of token expiration, Fuse handles "refresh tokens" to acquire new access tokens.

Zipped Multiple Download

The user can select multiple files to download. Requests for API downloads are sent through HTTP headers. Once the request is approved, the file streaming will now get started. The file that has been fetched from each drive will be saved to the cloud server of the Fuse. Fuse creates a folder for archive to store the retrieved files. Once all files are fetched, the system prompts the user to input on which local path to store the archived files.

Distributed Multiple Upload

When user uploads files to the system, the files are distributed across different cloud storage services. The system uses Round-robin algorithm to do the process. This makes the system bypass traffic and over storing of files in a single cloud service provider. This will also prevent the user from vendor lock-in and ensures the user that whenever there are failures in a certain cloud service provider, not all files will be off track.

Limitations

Manual Creation of Cloud Accounts

Fuse is unable to support auto registration of user accounts. This is due to the fact that it is not being provided by the API or SDK because this will prevent any spam machines to generate false accounts. The user needs to personally create his account and authenticate Fuse to manage his files.

Cloud Rate Limitation

This limitation is documented online. Some cloud APIs and SDKs put limits to the amount of calls the app can make per user. Currently, Google Drive API has a courtesy limit of 1,000,000,000 queries per day. The default user rate limit is 1000 request per 100 seconds [3] [11] [10] [12]. Dropbox has no limit for requests per day but have a maximum of upload file size limit of 150MB per day [5] [10]. While Box has a limit of 50,000 requests per day [2] [6].

Fuse Upload Limitation

To be able to upload big size of files, the researcher changed the default PHP settings to allocate a bigger upload size limit. However, there are some constraints like low bandwidth connection, which would result to HTTP timeout error is inevitable. Unfortunately, Fuse is unable to support multiple file upload which are in music and video formats.

Real time Viewing of Files

Fuse cannot view files in real time due to the slow internet connection and the limited number of requests per cloud storage provider per second. This occurs when the system is bombarded with incoming and outgoing requests, such as multiple upload, deletion and download of files. However, Fuse will be providing a "Refresh" icon in order to update the view of the user.

Performance Testing

Table 2 displays the time spent for Fuse to execute a certain test case. But still, internet connection can change the result. The individual performance testing per test case of each drive is done by placing a script in certain functions. There is only file that is uploaded, deleted, and downloaded that shows the following result.

Table 2: *Fuse Performance Testing*

Test Case	Google Drive	Dropbox	Box
Login for authentication and syncing the cloud storage provider to user's *Fuse* account	6.97 seconds	4.41 seconds	7.99 seconds
Uploading of files	7.06 seconds	6.7 seconds	6.82 seconds
Deletion of files	1.53 seconds	1.49 seconds	1.3 seconds
Downloading of files	5.67 seconds	2.28 seconds	2.6 seconds
Refreshing of dashboard	3.92 seconds	3.2 seconds	5.4 seconds

Overall, the outcome presents that Google Drive has a slow response time in finishing the tasks than the 2 drives. This is due to the fact that Google Drive has more background processes while performing each task. The table also reveals that it is faster to execute the request in Dropbox. This is also known to the community that Dropbox has a faster response time in terms of managing files. This is why most users who are inclined to cloud storage services would prefer Dropbox. The result also shows that Box has slower execution time upon account authentication and refreshing of dashboard.

4.0 Summary of Finding

Generally, the system was able to deliver the certain specifications being set during the idea generation of the system. However, there were workarounds that requires to be done in order to simulate the complete process. Instead of auto creation of the 3 drive accounts, the user needs to manually do it because of some constraints placed in the APIs and SDKs. *Fuse* cannot support a multiple music and video file upload. This is due to fact, that with low bandwidth connection, the system will have a timeout error. The system is also restricted with the number of API calls or request per user per day. That is why *Fuse* at some point, is not able to display a result in real time. Every functionality of the system was tested by authentic accounts and data in order to validate its correctness. Initially, the authentication process of the 3 cloud storage services have minor problems. The access token usually terminates once the *Fuse* account is on standby for 1 hour. However, this has been resolved by acquiring fresh tokens from the drives upon expiration. Because of the integration of multiple APIs, some functionalities were hard to implement but still, the developers were able to find ways to resolve these issues. Thus, completed the requirements successfully. Internet connection matters really, since API calls needs to have a fast connection to have a faster result. On average, the execution time of the system is fair enough to display the expected result.

5.0 Conclusion and Future Works

The system suits best for data storage on a personal scale. The user can now allow a system to merge the prominent cloud storage providers that will suffice the need for larger storage capacity of a regular consumer in the digital era. That is why, the system is recommended for people who relies their data storage and file warehousing in the cloud. The user will no longer memorize a lot of cloud storage credentials just to access their files. *Fuse* enables the user to access the 3cloud storage providers' files in just 1 portal. This will also prevent the user to have a potential vendor lock-in.

As of now, *Fuse* is only capable of combining 3 cloud storage services. To enhance the usability of *Fuse,* the developers recommend to integrate more cloud storage platforms for larger storage capacity. The files that are managed by Fuse in each cloud storage should be encrypted to prevent user manual manipulation of files. To have a more robust experience with *Fuse*, the new version of *Fuse* should give the users the ability to acquire multiple accounts not just in different cloud storage platforms but also in the same service provider. Furthermore, it would be nice to establish a desktop version of *Fuse* in order for users to have convenience in accessing their files. This will also allow local data backup.

References

Abu-Libdeh, H., Princehouse, L., & Weatherspoon, H. (2010, June). RACS: A Case for Cloud Storage Diversity. In Proceedings of the 1st ACM symposium on Cloud computing (pp. 229-240). ACM.

Adammbalogh/box-php-sdk. (n.d.). Retrieved from https://goo.gl/8Mrczf

API Reference. (n.d.). Retrieved from https://goo.gl/mnasQQ

Butler, B. (2015, June 09). 19 free cloud storage options. Retrieved March 17, 2017, from https://goo.gl/h4KfTm

Fisher, S. (n.d.). 17 Free Cloud Storage Services - No Strings Attached. Retrieved March 18, 2017, from https://goo.gl/SHqf2T

Free APP to Combine Cloud Storage into One. (n.d.). Retrieved March 18, 2017, from https://goo.gl/votQLU

Google Drive - Cloud Storage & File Backup for Photos, Docs & More. (n.d.). Retrieved from https://www.google.com.ph/drive/

Box Developer Platform. (n.d.). Retrieved March 24, 2017, from https://docs.box.com/v2.0/reference

Odrive | Combine all your storage. (n.d.). Retrieved from https://www.odrive.com/m/

Otixo: Encryption and file manager for multiple clouds. (n.d.). Retrieved October 1, 2015.

PHP Core API - Dropbox. (n.d.). Retrieved from https://goo.gl/5jvmz7

PHP Quickstart. (n.d.). Retrieved from https://goo.gl/88cDUI

Qian, L., Luo, Z., Du, Y., & Guo, L. (2009). Cloud computing: an overview. In Cloud Computing (pp. 626-631). Springer Berlin Heidelberg.

Use Google Drive Api in Laravel 5. (n.d.). Retrieved from http://goo.gl/F7LYLw

Wu, J., Ping, L., Ge, X., Wang, Y., & Fu, J. (2010, June). Cloud storage as the infrastructure of cloud computing. In Intelligent Computing and Cognitive Informatics (ICICCI), 2010 International Conference on (pp. 380-383). IEEE.