App Engine Application for Detecting Similar Files in Google Drive

Vijay Raj [1], Namita Bilagi [2], G S Nagaraja [3]

[1] Student at Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, Karnataka
[2] Student at Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, Karnataka
[3] Professor and Associate Dean at Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, Karnataka

---***---

Abstract - *Duplicate files are ruling over storage for decades. Especially when the storage is meant to backup for multiple users belonging to similar group like family or friends. In this paper we identify and list similar files present in one of the most popular clouds back up storage 'google drive'. The drive contains large amounts of different files that would ideally be stored by a family in their cloud backup storage. In this application we develop a python code and deploy it in Google App Engine, to analyze similar files in the drive. MD5 hashing approach is used to identify these similar files in the large quantity of data.*

Key Words: App Engine, Google Drive, MD5, Proximity measure, Replicate file, , Security, Unique files

I INTRODUCTION

A Similar file has many meaning, such as date modified is different, same file but copied so the date created for that file is different but content is the same or content itself is different [1][3]. These files occupy many giga to terra bytes of storage, identifying these files especially in personal storage can benefit users saving money. Let us get to know some modules and API used:

1.1 MD5

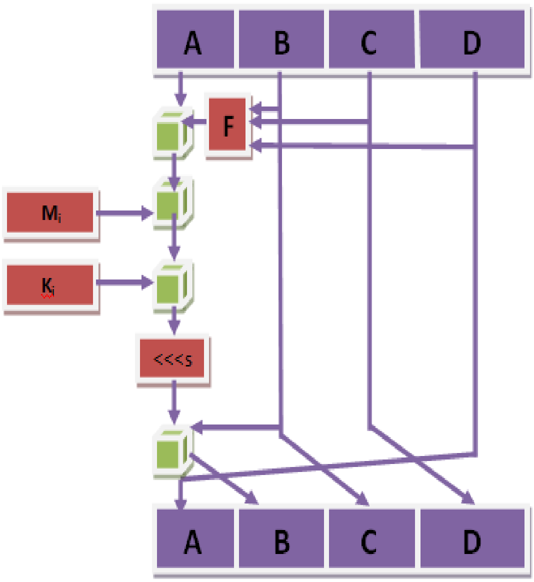
The MD5 (message-digest algorithm) is widely used hash function that is cryptographically broken.

- Input: Arbitrary length message.
- Output: Hash code of size 128-bit.

The input message is divided into 512-bit blocks (eighteen 32-bit words) for each chunk. The message must be an multiple of 512-bit blocks else padding is used.

One MD5 operation is made up of 64 of them, arranged in four rounds of 16 operations each. Each iteration of the nonlinear function F employs a different function. Ki is a 32-bit constant that is unique for each operation, and Mi is for a 32-bit block of the message input. s stands for a left bit rotation by s.

MD5 algorithm operates on a 128-bit state, then divided into four 32-bit words, denoted as A, B, C, and D.



1) Fig.1 MD5 Algorithm[4]

Below are initialized to certain fixed constants.

A = x67452301

B = 0xEFCDAB89

C = 0x98BADCFE

D = 0x10325376

The main algorithm then uses each 512-bit message A block each time to change the state. Single message block is processed in 4 steps known as rounds, each of which consists of 16 operations based on the non-linear function F, modular addition, and left rotation. The fig.1 above shows single round's of operations that message goes through. Each round uses a different one of the four potential functions F.

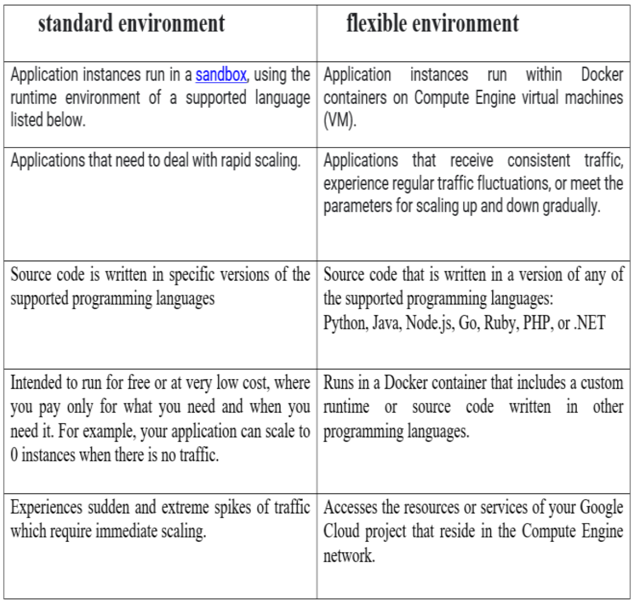
1.2 Google App Engine

Google App Engine is PaaS model. Web application development and hosting are both handled by the extremely scalable Google App Engine. The programs are built to accommodate numerous users at once without affecting overall performance. GAE can be used to construct cloud applications for service providers to offer their products [2]. App Engine characteristics [11]:

- Popular language: build application using various language such as Java, Python, Ruby, PHP etc.
- Open and flexible: Programmers can install custom libraries and framework on app engine by supplying docker container.
- Powerful application diagnostics: use Google's cloud monitoring and cloud logging tool to check the health and performance of GAE application. Fix bugs quickly using debugger and error reporting tools.
- Application versioning: host and manage different versions of same a.

You can choose between two Python language environments on App Engine[11]. Both environments allow you to utilize serving technology to build your web, mobile, and IoT applications rapidly and with no operational cost. They both scale quickly and effectively to manage increasing demand. Despite the similarities between the two environments, there are some significant differences as well.

Table 1. Standard vs flexible environment in App Engine



standard environment	flexible environment
Application instances run in a sandbox, using the runtime environment of a supported language listed below.	Application instances run within Docker containers on Compute Engine virtual machines (VM).
Applications that need to deal with rapid scaling.	Applications that receive consistent traffic, experience regular traffic fluctuations, or meet the parameters for scaling up and down gradually.
Source code is written in specific versions of the supported programming languages	Source code that is written in a version of any of the supported programming languages: Python, Java, Node.js, Go, Ruby, PHP, or .NET
Intended to run for free or at very low cost, where you pay only for what you need and when you need it. For example, your application can scale to 0 instances when there is no traffic.	Runs in a Docker container that includes a custom runtime or source code written in other programming languages.
Experiences sudden and extreme spikes of traffic which require immediate scaling.	Accesses the resources or services of your Google Cloud project that reside in the Compute Engine network.

1.3 App engine admin API

Google App Engine Admin API is a RESTful API for managing your App Engine applications. Many of the App Engine administrative tasks can be found in the Google Cloud dashboard and can be accessed to the Admin API. Admin API allow us manage App Engine applications in a way that is best suited for your environment or process.

1.4 Drive API

Google Drive is not a brand-new product by Google. It is a rebranded version of Google Docs that users may still use in their familiar ways while adding a new, optional storage capability. A marketing ploy was used to change the product's name in order to draw customers to the expanding cloud storage sector. In contrast to Google Docs' previous emphasis on applications, Google Drive's name conveys the impression that its major role is the cloud file storage service[9].

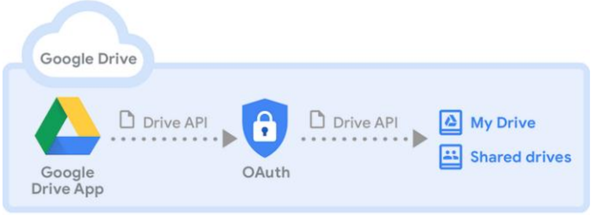


Fig 2. Drive API flowchart

1.5 Google OAuth 2.0

The OAuth 2.0 protocol is used by Google APIs for authentication and authorization. Web server, client-side, installation, and limited-input device applications are some examples of popular OAuth 2.0 events that Google supports.

Users are only authenticated when they accept the terms that are provided to them on a user consent screen when using OAuth 2.0 for authentication. Applications that use OAuth 2.0 and satisfy one or more of the verification criteria are verified by Google.

II Literature Review

Sif[1] is a tool used and all similar files in a large file system are located by this tool . Even though two files may otherwise be completely different, they are deemed comparable if they share a significant number of similar pieces. For instance, a file might be a realignment of another file or it might contain another file, possibly with some alterations. Finding all groups of comparable files takes about 500MB to 1GB of processing time per hour, even for only a 25% similarity. At a quick post-processing stage, the user can choose the degree of similarity and a number of other specific factors. In file management, information collecting ,data compression, file synchronization, program reuse and plagiarism detection, the application of sif tool can be found.

At Hewlett-Packard[3], millions of technical related support documents are housed in the various repositories. Such collections are periodically merged and groomed as part of content management. During the process, it is crucial to spot and eliminate support papers that are essentially copies of newer versions. By doing this, the collection's quality is enhanced,

chaff is removed from search results, and consumer satisfaction is enhanced. They Present a method for searching across huge document repositories to locate comparable files. It works when the byte stream is chuncked to identify unique signatures that might be present in several files. Clusters of connected files can be found after an investigation of the file-chunk graph. Scalability can be considerably improved by using an optional bipartite graph splitting technique[3].

Anand Bhalerao and Ambika Pawar's [5] work discusses the data deduplication technology background study, including different forms of deduplication and its performance metrics. Various chunking strategies are also discussed with algorithms used in the data deduplication process. The comparision and taxonomy of several chunking algorithms are described in the paper. The research on chunking algorithms' benefits and drawbacks for potential future research topics is summarised in the paper's conclusion.

Mr. Pushpendra Singh Tomar, Dr. Maneesh Shreevastava[6] their work on detecting duplicates using MD5 hash on various web files like pdf, html, etc. In their paper removing duplicate documents is essential to lowering runtime and increasing search accuracy. There are billions of unique URLs that are currently being retrieved by search engine crawlers, of which hundreds of millions are copies in some way. As a result, speedy replicate detection speeds up indexing and searches. 400 million exact replicas of 1.2 billion URLs were discovered using an MD5 hash after investigation by one vendor. To support the system some amount of hardware needed is decreassed and indexing time is much reduced when collection volumes are reduced by tens of percentage points.

Researchers at Google[7] published a paper on improving google drive experience, such as UX changes where users can access recent files, most used files using machine learning techniques. They also discuss about scaling of this feature to business, But the paper majorly focuses on UX optimizations.

Most of papers focus upon finding similar files in large database from different companies. Example: HP uses chunking and matching for detecting similar files[3]. But there very few focused upon personal storage deduplication. Although the methods can be applied on user level storage but have not targeted on popular cloud storages.

III METHODOLOGY

3.1 DATA SET

The data is obtained by applying web scrapping on numerous websites.

Table 2. Data set overview

File Types	Number of files
Images	4056
PDF	779
MP3	200
MP4	162
Documents	1234
Other Files	864
Total:	7295

All these files are uploaded to google drive, some files are duplicated to get accurate results. The data scrapped are very similar to that of a back up of personal files of a family. The files varied in various parameters such as name of the files, date created, modified, author etc. but if the content of the files are the same their md5 check sum would match.

3.2 Implementation

The application is implemented using mainly 4 modules and API:

I. OAuth

II. Drive API

III. MD5

IV. App.yaml for App engine

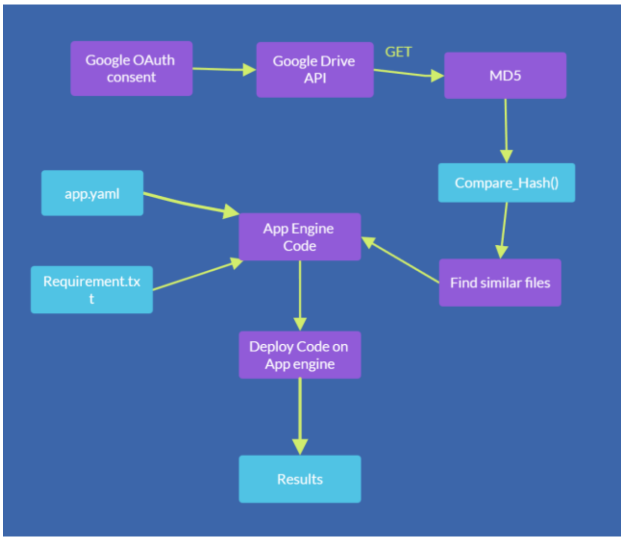


Fig 3. Workflow Flowchart

The application is implemented using following steps:

1. Google OAuth consent is needed to access google drive as prerequisite of google drive API. OAuth generates a “secret.json” file which we need for our code.

2. Google Drive API takes secret.json as input and allows user to authenticate the developer to access their drives. After a successful authentication session, drive API generates “token.json”.

3. Now we are ready to access the files in the drive. For finding similar files we need to get the md5 hash of each file. In google drive when we upload the files, google automatically generates md5 hash for each file and attaches it to the file as metadata.

4. After we access md5 hash, we use brute force approach to string match md5 hash of other files to find similar files.

5. Now to turn the code into an app engine code, we need to create two files:

App.yaml : which sets the environment of app engine server. For example: if we use python then we have to set runtime: python3. We can also set flexible environment or standard environment, number of vm scaling instances.

Requirement.txt: this file contains all the necessary packages that are used in our code.

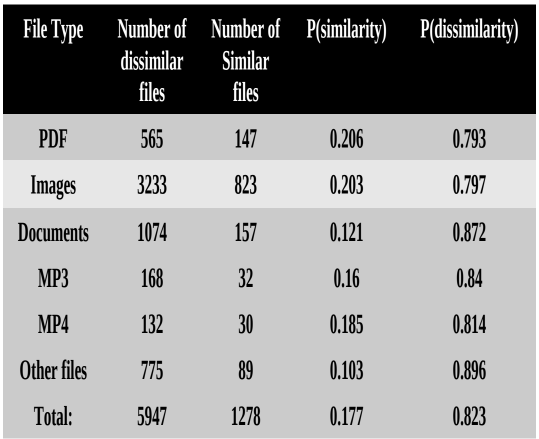
6. Now the code is ready to be deployed on app engine, but before that we need to enable few things: billing account, App Engine API and Cloud build API. Then using gcloud shell command “ gcloud app deploy” to deploy.

7. To view the results of similar files in google drive visit the URL provided in the cloud shell after deploying.

IV RESULT AND ANALYSIS

It is important to note that MD5 hash is generated by google drive itself , when user uploads the files drive automatically generated a MD5 hash for each file, this is used for various google drive user experience features.

Table 3. P value for similar and dissimilar file types



File Type	Number of dissimilar files	Number of Similar files	P(similarity)	P(dissimilarity)
PDF	565	147	0.206	0.793
Images	3233	823	0.203	0.797
Documents	1074	157	0.121	0.872
MP3	168	32	0.16	0.84
MP4	132	30	0.185	0.814
Other files	775	89	0.103	0.896
Total:	5947	1278	0.177	0.823

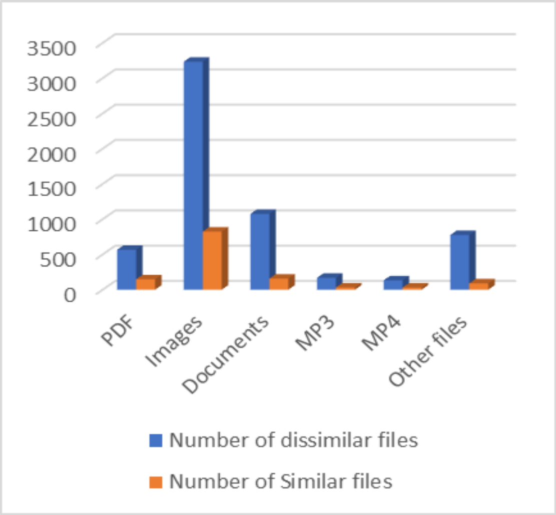
Measures for different types of similar and dissimilar files can be represented as:

P(file similarity in a storage) = Total number of similar / Total number of files in storage

P(file dissimilarity in a storage) = (Total number of files in storage - total number of similar) / Total number of files in storage.

Below table and graph gives the comparison for number of similar files and dissimilar files.

Fig 4. comparison of number of similar and dissimilar files



We can notice that images have most number of replicate data, this is normal to expect from a family backup since they share lot pictures from each family member. Documents and pdf files have similar number of replicate files, this maybe due to syncing activity between various application, example google cloud and MS office syncing the same files. MP3 has one of the least similar files count. MP4 has 40 similar files, MP4 files are one of the large files compared to others, making MP4 duplicate data take up lot of storage in google drive. Example single movie in MP4 format may be around 1.2 GB (depending on resolution and codecs). Other files can be classified as miscellaneous, such as the data from an mobile application back up data (.bin, .dat, etc).

V CONCLUSION AND FUTURE SCOPE

User data have comparably many similar files within the storage. So we built an application code to deploy on Google App Engine to find these similar files.

Using App engine we can build scalable web applications and along with Google drive access integration. Detecting similar files in very large amount of data files is not a barrier anymore thanks to App engine scalable computation power. MD5 is a renowned crypto authentication technique, which gives hash of file to match with other files hash to know if it's the same.

Moreover, this application can be used for deduplication purpose, applied on google drive to free large amount of storage occupied by duplicate files. There are very few application that can perform deduplication on google drive, moreover detecting similar files in popular cloud storage like google drive will save money for many users by not buying extra storage. This application can also be used on large data back up centers, where large files are chunked and hashes are generated to compare.

V REFERENCES

[1] Manber, U., 1994, January. Finding Similar Files in a Large File System. In Usenix winter (Vol. 94, pp. 1-10).

[2] Severance, C., 2009. Using Google App Engine: Building Web applications. " O'Reilly Media, Inc.".

[3] Forman, G., Eshghi, K. and Chiocchetti, S., 2005, August. Finding similar files in large document repositories. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (pp. 394-400).

[4] Gupta, S., Goyal, N. and Aggarwal, K., 2014. A review of comparative study of md5 and ssh security algorithm. International Journal of Computer Applications, 104(14t)

[5] Bhalerao, A. and Pawar, A., 2017, May. A survey: On data deduplication for efficiently utilizing cloud storage for big data backups. In 2017 international conference on trends in electronics and informatics (ICEI) (pp. 933-938). IEEE.

[6] Tomar, P.S. and Shreevastava, M., 2011. The study of detecting replicate documents using MD5 hash function. International Journal of Advanced Computer Research, 1(2), p.14.William Stalling7, Fourth Edition, Cryptography and Network Security (Various Hash Algorithms).

[7] Qinlu He, Zhanhuai Li and Xiao Zhang, "Data deduplication techniques," 2010 International Conference on Future Information Technology and Management Engineering, 2010, pp. 430-433, doi: 10.1109/FITME.2010.5656539.

[8] Chen, S.J., Qin, Z., Wilson, Z., Calaci, B., Rose, M., Evans, R., Abraham, S., Metzler, D., Tata, S. and Colagrosso, M., 2020, August. Improving recommendation quality in google drive. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2900-2908).

[9] Gallaway, T.O. and Starkey, J., 2013. Google Drive. The Charleston Advisor, 14(3), pp.16-19.

[10] 10. Fan, J. and Xie, W., 1999. Some notes on similarity measure and proximity measure. Fuzzy sets and systems, 101(3), pp.403-412.

[11] Google.com. Developer's Guide. http://code.google.com/appengine/docs/