Improving Storage Utilization by Image Deduplication Approach

Saurya Saurav[1], Aman Kumar Sharma[2], Raj Lakhwan[3]
[1,2,3] *Department of Computer Engineering, Bharati Vidyapeeth's College of Engineering Lavale,Pune*

***Abstract*- The rate of data growth is increasing rapidly. This requires an increase in overhead charges for hardware, data centers, and database management systems (DBMS). To overcome redundancy of similar data, technique of deduplication can be used.**

Data deduplication is a method which ensures only one unique instance of data is present in the database. Deduplication has been widely used in cloud and big server systems to improve the storage utilization and effective usage of storage systems.

This paper focuses on image deduplication. The idea is, in a chat application not downloading images that are already present in the local storage.

Index Terms- Deduplication, Perceptual Hashing, Hash Code, Data Reduction, Redundant Data

I. INTRODUCTION

We encountered with lots of repeated or duplicated images which are circulated through various social media platform, which in fact create a mess of things in our gallery. We come up with an idea why can't we develop an system which automatically removes duplicate image without downloading it or delivering to users.

Data deduplication is a method which ensures only one unique instance of data is present in the database. Deduplication has been widely used in cloud and big server systems to improve the storage utilization and effective usage of storage systems. Similarly, eliminating the redundant images from the storage or from cloud can be stated as Image Deduplication.

This system will be flexible and can be used in different scenarios. Module can be used in chat apps, it will notify if the image received from the sender is already present in the storage or not.

Our module is designed to provide on the go solution, so hashing is done on the senders side itself. Thus, Storage optimization technique is more powerful. We intended to build it for local database, which will ultimately save lots of processing and space on our device.

II. LITERATURE SURVEY

Deduplication has been widely used in backup systems and archive systems to improve storage utilization effectively. However the traditional deduplication technology can only eliminate exactly the same images, but it is unavailable to duplicate images which have the same visual perceptions but different codes. To address the above problem, this paper proposes a high-precision duplicate image deduplication approach. The main idea of the proposed approach is eliminating the duplicate images by five stages including feature extraction, high dimensional indexing, accuracy optimization, centroid selection and deduplication evaluation.[1]

In recent years, the explosion of the data such as text, image, audio, video, data centers and backup data lead to a lot of problem in both storage and retrieval process. There are two existing techniques for eliminating the redundant data in the storage system such as data deduplication and data reduction. Data deduplication is one of a technique which eliminates redundant data, reduces the bandwidth and also minimizes the disk usage and cost. This paper attempts to summarize various storage optimization techniques, concepts and categories using data deduplication. In addition to this, chunk based data deduplication techniques are surveyed in detail.[2]

III. EXISTING SYSTEM

1. From Android 8.0 duplicate images are deleted from the local gallery, but after downloading the image.

2. When uploading a duplicate image on the cloud storage drive on google, the images are automatically not uploaded. This is based on cloud.
3. In windows, apps like duplicate cleaner, has a feature to remove all the duplicate images from the drive.

Not all the above system provides on the go solution. Image has to be downloaded once. All the apps uses Post-process deduplication. Wastage of Local Memory as well as Internet Data is done.

IV. PROPOSED SYSTEM

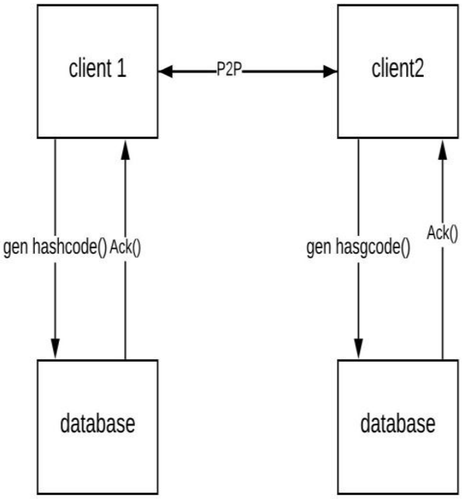


Fig 1. Communication Diagram of the System

In Fig 1, shows the basic model of how the system works. Both the device Client 1 and 2 contains the hash code of every image present in the local database and it is generated using the hashing algorithm and stored in its respective database. There is a peer to peer connection between the clients.

Algorithm: In this system, we are using perceptual hashing algorithm to generate hash codes of the images. This algorithm better suits the proposed system then any other hashing technique.

Perceptual Hashing generates a fixed length media fingerprint and that can be used to make meaningful comparison between two media files.

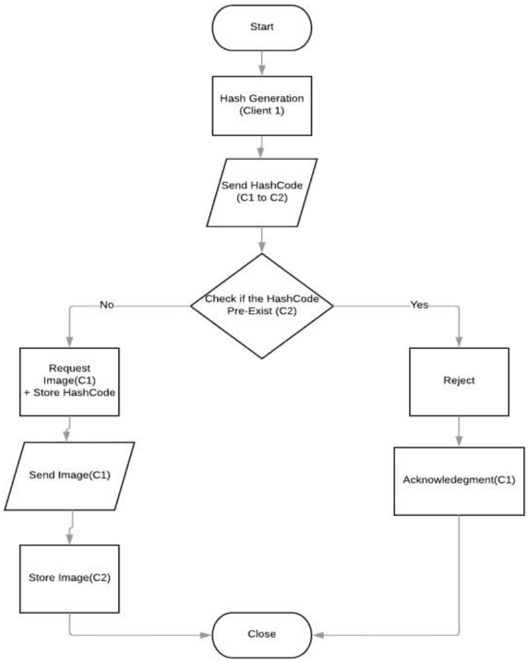

Start
Hash Generation (Client 1)
Send HashCode (C1 to C2)
Check if the HashCode Pre-Exist (C2)
No
Yes
Request Image(C1) + Store HashCode
Send Image(C1)
Store Image(C2)
Reject
Acknowledegment(C1)
Close

Fig 2. Data Flow Chart

Let’s see the flow of the system. Consider a message is to be sent from Client 1 (C1) to Client 2 (C2). Both C1 and C2 have there image hash codes already stored in respective database.

Now, C1 sends an image hash code to C2. C2 checks its own database if a similar hashcode is present in the database or not.

If, yes there is a similar hash code in C2’s database then an acknowledgment is sent to C1 that image is already present and image will not be downloaded.

If, the hash code is not present then acknowledgment is sent to C1 requesting the image and its hash code, which is sent by C1 and stored in C2’s local database.

V. RESULT

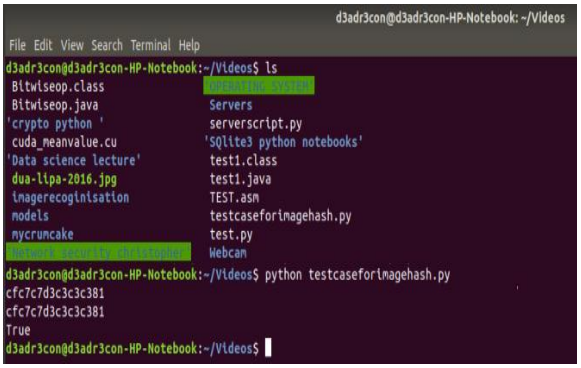
This result is based on image database caltec101 which contains 9143 images. For every image 10 individual test images with slight, randomized modifications were done.[5]

Hashing Technique	Total Error Percentage	Average Hamming Distance	Collisions	Avg calculation duration per image
aHash	25.0%	0.68	4438	2.25 ms

bHash	23.6%	1.14	711	112.70 ms
dHash	43.6%	1.44	421	0.33 ms
mHash	26.2%	0.88	4432	0.33 ms
pHash	*23.7%*	*1.12*	*483*	*60.05 ms*
wHash	21.2%	0.61	7866	0.91 ms

Table 1. Results of Different Hashing Algorithm

As the perceptual hash showed good results in overall deviations and the average Hamming distance while producing few false positives and being twice as fast as the block hash algorithm, we opted for the perceptual hash algorithm.


d3adr3con@d3adr3con-HP-Notebook: ~/Videos
File Edit View Search Terminal Help

```
d3adr3con@d3adr3con-HP-Notebook:~/Videos$ ls
 Bitwiseop.class                'OPERATING SYSTEM'
 Bitwiseop.java                  Servers
'crypto python '                 serverscript.py
 cuda_meanvalue.cu              'SQlite3 python notebooks'
'Data science lecture'           test1.class
 dua-lipa-2016.jpg               test1.java
 imagerecoginisation             TEST.asm
 models                          testcaseforimagehash.py
 mycrumcake                      test.py
'Network security christopher'   Webcam
d3adr3con@d3adr3con-HP-Notebook:~/Videos$ python testcaseforimagehash.py
cfc7c7d3c3c3c381
cfc7c7d3c3c3c381
True
d3adr3con@d3adr3con-HP-Notebook:~/Videos$ 
```

Fig 3. Snippet of Terminal Testing pHash Technique

Above snippet of terminal show the hash code generated of the image and it compares the two and return True as the hash codes are same.

VI. CONCLUSION

We developed a module which is able to rectify the images which are already present in user's local database in a chatting application..

Thus, improving the storage utilization and optimizing it. This functions are achieved using the technique of Image Deduplication approach.

REFERENCES

[1] Ming Chen, Shupeng Wang, Liang Tian ”A High precision Duplicate Image Deduplication Approach” JOURNAL OF COMPUTERS, VOL. 8, NO. 11, NOVEMBER 2013.

[2] E.Manogar, S. Abirami “A study on data deduplication techniques for optimized storage” 2014 Sixth International Conference on Advanced Computing (ICoAC).

[3] V. B. Nemirovskiy Tomsk Polytechnic University, Russia. A.K. Stoyanov Tomsk Polytechnic University, Russi ”Near-duplicate image recognition” 2014 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS).

[4] Waraporn Leesakul, Paul Townend, Jie X “Dynamic Data Deduplication in Cloud Storage” 2014 IEEE 8th International Symposium on Service Oriented System Engineering.

[5] Website: https://content-blockchain.org/research/testing-different-image-hash-functions/