Reduce Redundancy in Server Storage using Deduplication and Compression Technique

[1]Gourav Sharma, [2]Aatif Jamshed,[3]Dr. Krishna Kant Agrawal
[1,2]Assistant Professor, [3] Professor
[1,2,3] Department of Computer Science & Engineering,
[1,2,3] Delhi Technical Campus ,Greater Noida, India.

Abstract : With the documentation process atomization document/file sharing increased this leads to increase in quantity of duplicate documents/files in the Internet and occupying HDD space in servers which is directly related with other resource i.e. hardware and increases maintenance cost making burden to sever owner and service providers. The proposed technique has been solved the problem of saving duplicate files in less space in real-time more efficiently and with high reliability and accuracy. Virtual addressing plays a vital role in to make a virtual file system for authentic users in highly secure manner where chances of real file direct access is negligible by unauthentic users. The proposed system will be useful in cloud services (like email) where hundreds to lakhs of duplicated files are shared and saved every day. The novel system saves lots of space in server by saving only new files and assigns virtual address for duplicate files.

IndexTerms **- hash function, hash key, file length, virtual address, compression.**

I. INTRODUCTION

With the world globalization, Internet speed increase and documentation process atomization document, images, videos and file sharing increased in comparison to few years back and regularly increasing day by day. This leads to increase in quantity of duplicate documents/files in the Internet and occupying HDD in servers. For example, if thousands of people are uploading the same image or copy of file it means these image or documents/files are to be stored for thousands of times which becomes very costly for the sever owner and service providers because it consumes more resource (i.e. hard disk, large no. of servers, air condition, electricity, high maintenance and replacement cost, large number of processor and more infrastructure). The proposed technique solves the problem of saving duplicate files in less space in real-time more efficiently and save money in installation of new HDD and give high accuracy (100%). The proposed system will be useful in social networking sites, blogs, email and search engines where hundreds to lakhs of duplicated files are shared and saved every day, the proposed system will save only new files and assign virtual address for duplicate files without save it on HDD which will be save lots of HDD space in server. Here, we used standard compression and hashing technique for compressing files and generate hash key respectively. Virtual addressing plays a vital role in this paper to create a logical / virtual file system where authentic users feels that they are working in their own private files in highly secure manner where chances of real file direct access is negligible by unauthentic users.

In next sections we will discuss about related works, proposed technique, results and discussion, and conclusion and feature scope.

2. Related Work

The proposed technique is not purely based of any kind of existing redundancy reduction technique. It is basically done by using any existing compress technique and hash function [2] for filtering/selecting a small group of files; which have same hash key but it may be different by content & size; out of trillions of file in real time same as antivirus software where antivirus compares hash key [13] for checking the existing file is a virus program /file or not. But antivirus doesn't compare their size and content due to database limitation and antivirus work and objective is totally different from the proposed technique. Existing techniques like real-time photo or video copy detection [3,4,5,6,7,8,9,10,11,12] are based on frames/blocks and other parameters, which have totally different concept from the proposed technique.

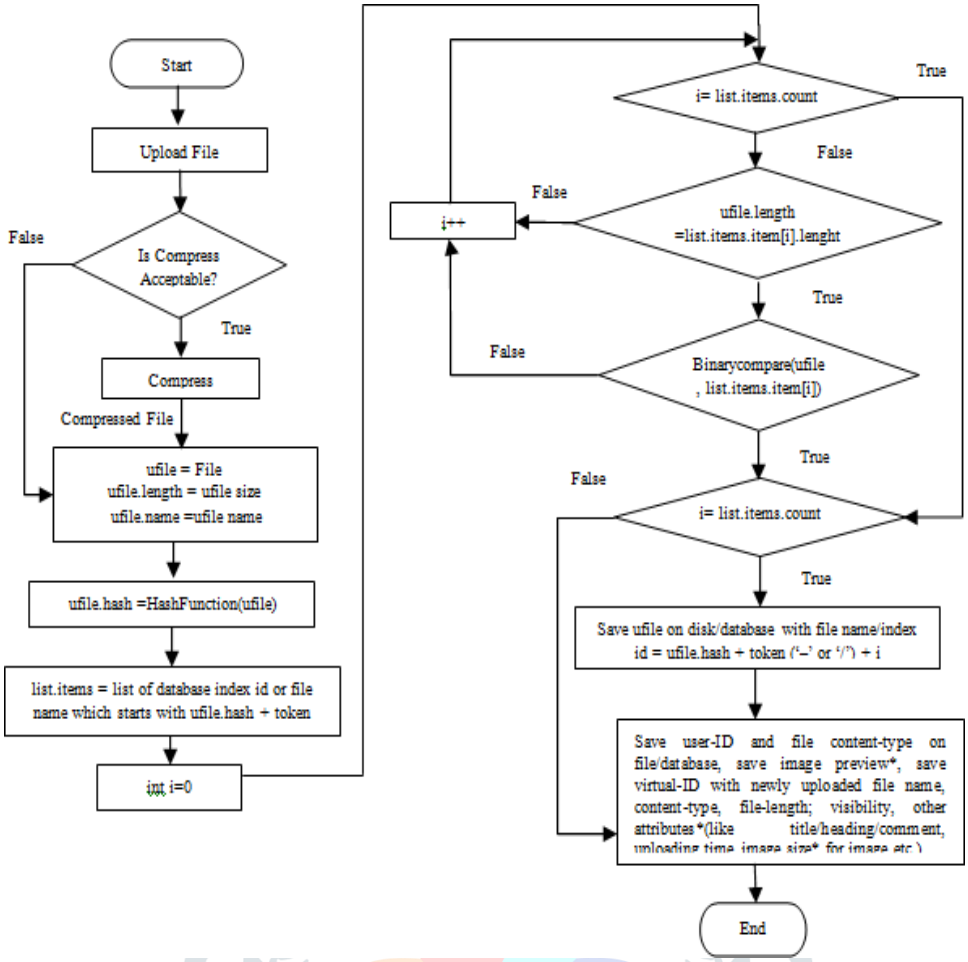
3. Proposed Technique

In the proposed technique, we are using "Real-Time Duplicate File Detection System" [1] concept to reduce redundancy in Server Storage. For achieving this we have made some adjustments in [1] using de-duplication and compression technique with virtual addressing concept (flow chart of the proposed technique is shown in Fig. 1).

I Upload File

Step 1: Upload whole file. Compress file (using any lossless data compression technique); if compress option acceptable.

Fig 1. Flow Chart Update File.



Step 2: Generate Hash key or Tag using hash function of the newly uploaded file.
Step 3: Check database index id or file name which starts with hash key plus token and list them in an array. If array is not empty then take first array element and go to Step 4 else go to Step 7 directly.
Step 4: Repeats Step 5 and Step 6 until and unless their conditions are satisfied otherwise continue with the next array element; if Step 5 and Step 6 are not satisfied until the last array element then go to Step 7.
Step 5: Compare the exiting file length; current array element; to the newly uploaded file length if both are same then go to Step 6 else go to Step 4.
Step 6: Compare exiting file content to the newly uploaded file using file binary comparison (to make the accuracy 100%) if both are same that means the newly uploaded file already exists on the server storage/hard disk; so, it should not stored on the server storage/ hard disk; and go to Step 8 else go to step 4.
Step 7: Save the newly uploaded file on the server Hard Disk/database with file name/ database index id = hash key plus token plus 0 (if duplicate file doesn't exist) / array length (if duplicate file exist) then go to Step 8.
Step 8: For security and privacy reasons, register user id/name/index and file content-type on file/ database, generate preview image of an image and save (if required), generate virtual path/virtual id and save with newly uploaded file name, content-type, file-length (size), visibility to friends/friend's friends/public (any one can view), other attributes/information (if required; like title/heading/comment on the file, time of uploading, image size if it is an image etc.).

II File Retrieval

In file retrieval user/friends/friend's friends/public could not see the actual location of the file even their thumbnail/preview image actual location (if any), they can see and use only their virtual path/virtual id for downloading/ retrieval of the file. Visibility attribute can hide file from friends/friend's friends/public eye even thumbnail/preview image.

III File Delete

In file delete only virtual path will be deleted from User's database that means actual file, thumbnail/preview file (any) and file attribute data is not deleted until any active user will be allocated the original file/registered on the file database.

Results and Discussion

In graph Fig. 2. shows that in the proposed technique when a new duplicate file is uploaded or deleted by same or other user, then there is no or very small increases or decreases respectively in file folder/database size almost fixed size within few bytes which is less than 300 byte and only large increase in File Folder/database size occurs when new non existing file is uploaded. When, in traditional techniques when a new duplicate or new file is uploaded by same or other user, then there is large increase in file folder/database size. Means if you upload same X number of same file/ duplicate file of size Y (compressed) then traditional techniques takes X * Y + c space and the proposed technique takes Y + c + d space where c= space for indexing and d= space for virtual key detail.

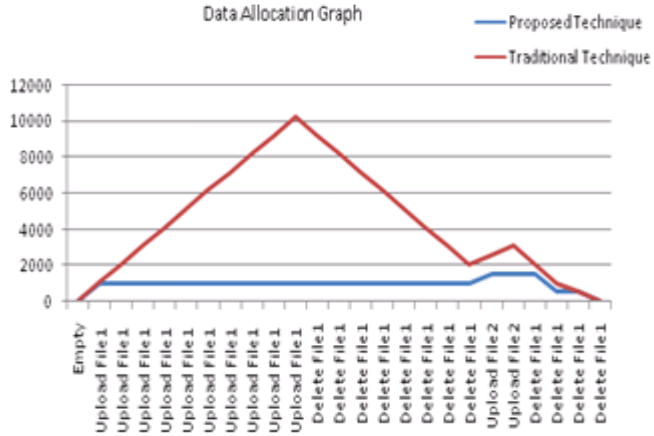


Fig 2.A comparative graph for data (in byte) allocation on server hard disk due to different file operations using the proposed technique and traditional techniques

Its means, X * Y + c > > Y + c + d i.e. X * Y >> Y here, c and d are negligible in compare to Y and X. Traditional techniques takes space >> proposed technique takes space to save X number of same file/ duplicate file of size Y.

Conclusion and Future Scope

The proposed technique solved the problem of saving duplicate files in less space in real-time more efficiently in comparison to exiting techniques [3-12] and save money in installation of new HDD and with high accuracy means 100%. The proposed system will be useful in social networking sites, blogs, email and search engines where hundreds to lakhs of duplicated files are shared and saved every day, the proposed system will save only new files and assign virtual address for duplicate files without save it on HDD which will be save lots of HDD space in server. Compression reduced storage required as well as bandwidth. Here, we used standard compression and hashing technique for compressing files and generate hash key respectively. Virtual addressing creates a logical/ virtual file system where authentic users feels that they are working in their own private files in highly secure manner where chances of real file direct access is negligible by unauthentic users.

References

1. Yaseer Ali Ahmad and Dr. Abhijit Mustafi: Real-Time Duplicate File Detection System. In: International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.55 (2015), Page No. 2984-2987, and Research India Publications. www.ripublication.com/ijaer.htm
2. Atul Kahate: Cryptography and network security 2 Ed.
3. Lifeng Shang, Linjun Yang, Fei Wang, Kwok-Ping Chan, Xian-Sheng Hua: Real-time Large Scale Near-duplicate Web Video Retrieval. In: Microsoft Research.
4. O. Chum, M. Perdoch, and J. Matas: Geometric min-Hashing: finding a (thick) needle in a haystack. In: In Proc. CVPR, 2009.
5. H. Jegou, M. Douze, and C. Schmid: Hamming embedding and weak geometric consistency for large scale image search. In: In Proc. ECCV, 2008.
6. J. Law-To, L. Chen, A. Joly, et al: Video copy detection: a comparative study. In: In CIVR, 371–378, 2007.
7. J. Law-To, V. Gouet-Brunet, B. Olivier and B. Nozha. Local behaviours labelling for content based video copy detection. In Proc. ICPR, 232–235, 2006.
8. J. Law-To, A. Joly, and N. Boujemaa: a live benchmark for video copy detection. In: Muscle-vcd, 2007.
9. D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91–110, 2004.
10. S. Maji, A. Berg, and J. Malik: Classification using intersection kernel support vector machine is efficient. In: In Proc. CVPR, 2008.
11. J. Matas, and O. Chum: Randomized RANSAC with sequential probability ratio test. In: In Proc. ICCV, 2005.
12. S. Poullot, M. Crucianu, and O. Buisson: Scalable mining of large video databases using copy detection. In: In ACM MM, 61–70, 2008.