International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 2, Issue 3, March 2015, PP 13-15
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
www.arcjournals.org

Cloud Based Mobile Storage Augmentation Design with Data Deduplication

M.Tanooj Kumar

Department of Computer Science,
Krishna University, Machilipatnam.
mtanooj@gmail.com

Abstract: *Mobile systems have limited resources such as network bandwidth, battery life, storage capacity and processor performance. However, rapidly rising data volume intensifies data storage and battery limitations of mobile devices. These restrictions may be alleviated by augmenting cloud based storage. Since cloud storage is pay as a service, they need to be utilized optimally so has to accommodate maximum data. Deduplication techniques are ideal solutions for reducing both bandwidth and storage space requirements for cloud storage in data centers. This paper presents duplication less storage system over the mobile cloud computing platforms and investigates the benefits and overhead of deduplication techniques adopted.*

Keywords: *Cloud storage, deduplication, Mobile cloud computing, cloud based mobile augmentation.*

1. Introduction

A Mobile device is a portable, battery operating wireless computing entity such as smart phone, tablet, handheld/wearable computing devices and vehicle mount computers. Mobile devices have recently gained momentous ground in several communities like government agencies, enterprises, social service providers (e.g. insurance, police, fire departments), health care, education and engineering organizations. However, despite of significant improvement in mobile computing capabilities, still resource poverty of mobile devices is a major deficiency. Rapidly rising data volume intensifies data storage limitation of mobile devices. These restrictions may be alleviated by augmenting cloud based storage. Since cloud storage is pay as a service, they need to be utilized optimally by accommodating maximum data.

Deduplication techniques are ideal solutions for reducing both bandwidth and storage space requirements for cloud storage in data centers. This paper presents a deduplication system over the mobile cloud computing platform. Deduplication means that the number of the replicas of data that were traditionally duplicated on the cloud should be managed and controlled to decrease the real storage space requested for such duplications. At the front end, the system has a deduplication application implemented using android. At the back end, the parse platform is used as a mass storage system and a fast indexing system. Promising results were obtained from our system on the mobile cloud environment.

The reminder of this paper is as follows. Section 2 provides background and motivation of the work; Section 3 describes the design of our deduplication framework; Section 4 describes the implementation; Section 5 evaluates the implementation; and, finally, we give concluding remarks and future works in Section 6.

2. Back Ground

2.1 Source and Target Deduplication.

In data intensive applications, files are moved from source to the target storage devices through the network. The source end is the front-end application and the Host or the Backup server to process and store the raw data. The target end is usually ultimate storage equipment, such as disk arrays. If unnecessary or redundant data is indicated and deleted in the front-end application, it will reduce the network transmission bandwidth and time. The disadvantage of data cancelling at the front-end application is that it will cost more for duplicate data detection and deletion. If the data deduplication task is done at the target end, it will spend more resources in redundant data processing, data transmission over networks, and computing resource consumption for redundant data detection. Therefore, it is recommended that source deduplication is more feasible than the target one.

2.2 Whole-File or Chunk-Based Data Deduplication.

The two existing approaches for identifying duplications are the file level and the chunk level. On the file level, the hash function will be executed for each file, and all hash values will be kept in the index. The advantage of this approach is that it decreases the quantity of hash values significantly. The drawback is that, the system will be experiencing some lag when dealing with a large file. On the chunk level, data streams are divided into chunks, each chunk will be hashed, and all these hash values will be kept in the index. The advantage of this approach is that it is convenient for a distributed file system to store chunks, but the drawback is having an increasing quantity of hash values. It means that hash values will occupy more RAM usage and increases lookup time. Based on this consideration, in this paper, the whole-file data deduplication techniques are used for our proposed framework.

3. System Architecture and Development

The main aim in designing this scheme is to design an improved technique for storage in Cloud computing. The overall architecture of the technique is shown in Figure 1. First, the user interface is used to select the file for deduplication. For this file hash values are computed by using MD5 algorithm. It involves the detection of duplicate files by comparing hash values generated. The files are compressed using LZW algorithm, after eliminating the duplicate values. The compressed file is uploaded to cloud storage.

S.NO	File Name	File Size (Bytes)	Compressed File	
			Without deduplication +compression (Bytes)	With deduplication +compression (Bytes)
1	a.txt	4651	2053	2053
2	b. txt	4121	1764	1764
3	c. txt	2397	1169	1169
4	d. txt	2516	1203	1203
5	e. txt	4015	1760	1760

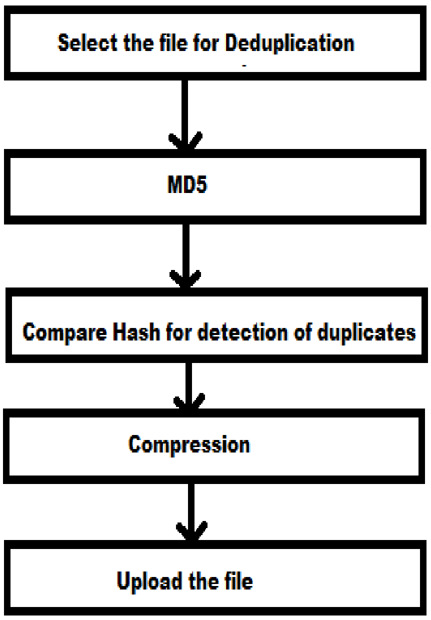

Figure 1. Overall Architecture

4. Experimental Setup

On the source side, Java language was used for implementing deduplication technique. Cloud storage was built with Parse platform. Parse Platform provides a simple web services interface that can be used to store and retrieve any amount of data at anytime from anywhere on the web. File level deduplication techniques were implemented using these techniques. The set of sample text files along with their copies are chosen for deduplication. Results are tabulated. When deduplicated segments are saved on the cloud storage and storage space is saved. Two runs were performed in this particular application. Testing was done by storing a file and then a duplicate copy of same file.

5. Performance Study

Test Item 1: *Different text Files with compression and Deduplication*

S.NO	File Name	File Size (Bytes)	Compressed File	
			Without deduplication +compression (Bytes)	With deduplication +compression (Bytes)
1	a.txt	4651	2053	2053
2	b. txt	4121	1764	1764
3	c. txt	2397	1169	1169
4	d. txt	2516	1203	1203
5	e. txt	4015	1760	1760

Total Size= 4651+4121+2397+2513+4015 = 17610 bytes

After Compression = 2053+1764+1169+1203+1760= 7949 bytes
Compression and Deduplication = 2053+1764+1169+1203+1760 = 7949 bytes

Total Size Reduced= 17610 – 7949 = 9661 bytes

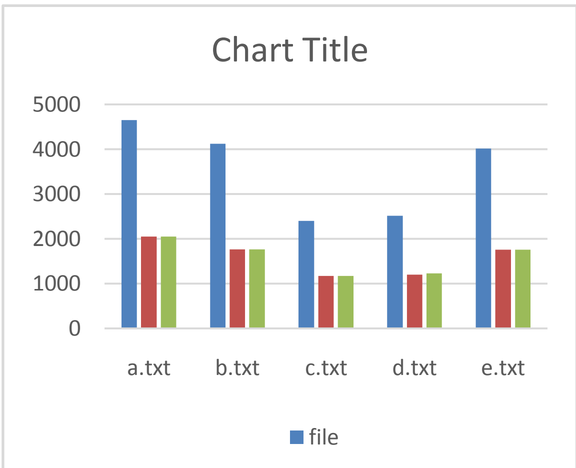

Fig.2. *Graph shows original files, files with deduplicaiton and files with deduplication and compression (Bytes) for test case1.*

TEST ITEM 2: With Different Files names And same content Files with compression and Deduplication

S.NO	File Name	File Size(Bytes)	Compressed File	
			Without deduplication +compression (Bytes)	With deduplication +compression(Bytes)
1	a.txt	4651	2053	2053
2	1. txt	4651	2053	0
3	c. txt	2397	1169	1169
4	2. txt	2397	1169	0
5	e. txt	4015	1760	1760

Total Size = 4651+4651+2397+2397+4015 = 18111 bytes

After Compression =

2053+2053+1169+1169+1760= 8204 bytes

Compression and Deduplication =

2053+0+1169+0+1760 = 4982 bytes

Total Size Reduced = 18111 – 4982 = 13129

bytes

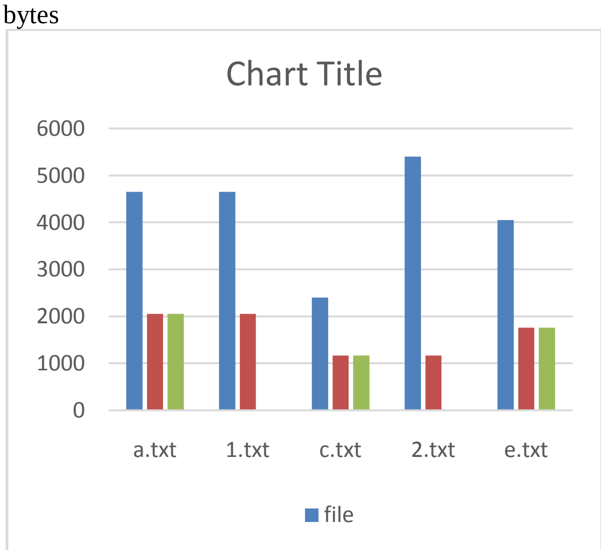

Fig.3. *Graph shows original files. files with deduplicaiton and files with deduplication and compression (Bytes) for test case2.*

6. Conclusions and Future Work

In conclusion, we have introduced a novel approach to data deduplication over the mobile cloud systems. This system is mainly useful for common users, who want to store data. Our approach exploits a file's hash value as an index saved on Parse Platform to attain high lookup performance, and it exploits 'link files' to manage mass data. In this system, the hash value is calculated at the client side prior to data transmission. When duplication is found, real data transmission will not occur. The experiments results show that the duplicated data space can be saved and the upload **performance is** not affected by the integrated schemes significantly. Currently, optimized cloud storage has been tested only for text files. In future, it can be further extended to support files of other types.

References

[1] *Saeid Abolfazli, Zohreh Sanaei, Ejaz Ahmad, Abdullah Gani, Rajkumar Buyya. Cloud-based Augmentation for Mobile Devices: Motivations, Taxonomies, and Open Challenges,IEEE communications surveys and tutorials, 2012.*

[2] *Deepu S R, Bhaskar R, Shylaja B S, Performance comparison of deduplication techniques for storage in cloud computing environment, Asian journal of computer science and Information Technology 4:5(2014) 42-46*

[3] *Ruey-Kai Sheu, Shyan-Ming Yuan, Win-Tsung Lo, and Chan-I Ku , Design and Implementation of File Deduplication Framework on HDFS, International Journal of Distributed Sensor Networks,Volume 2014, Article ID 561340.*

[4] *K.Govinda, Yuvaraj Kumar, Storage Optimization in cloud Environment using Compression Algorithms, International Journal of Emerging trends and technology in computer science , Volume 1, Issue 1, May-June 2012.*

[5] *Nazanin Aminzadeh, Zohreh Sanaei, Siti Hafizah Ab Hamid Mobile storage augmentation in mobile cloud computing: Taxonomy, approaches, and open issues, Elsevier, 2014.*