IJETST- Vol.||02||Issue||06||Pages 2656-2659||June||ISSN 2348-9480 2015

Open access Journal International Journal of Emerging Trends in Science and Technology

Assuring Data Integrity in Cloud Using Regenerating Codes

Authors
Bhavyashree K P[1], GuruPrakash C D[2]
[1]Master of Technology, Dept of Computer Science, Shridevi Institute of Engineering and Tech., Tumkur
Email: *bhavyakpshree@gmail.com*
[2]Head & Professor, Dept of Computer science, Shridevi Institute of Engineering and Technology, Tumkur
Email: *cdguruprakash@gmail.com*

Abstract

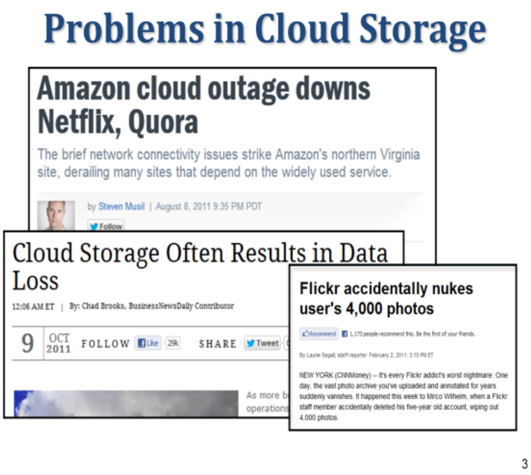
The cloud computing is the emerging technology growing rapidly with its services. Cloud storage is one of the many services it is rendering. This cloud storage not only allows the clients to store their huge volume of precious data, also provides a facility to move the data out to the other cloud storage or data centers .In these cases, the data may be corrupted or may be compromised by data centers. Hence the clients must be alert to assure the integrity and correctness of their data.

This paper aims to provide assurance of integrity, hence DIP is put forth. The issues fault tolerance and recovery procedures are addressed as additional features. Regenerating codes are introduced to give the above said features by distributing the chunks of data among different servers randomly and only the lost part of data is regenerated in to new server. Thus repair traffic is minimized and fault tolerance is provided. The tests are conducted under different parameters for the efficiency.

Keywords: *Cloud storage, Data integrity, fault tolerance, regenerating codes, repair traffic*

Introduction

Technology is ruling the world today. Cloud computing is the major well known technology all are depending most. The clients are able to do their works from the sitting place itself just by click. The services this cloud computing is offering are resource sharing, data storing, even the shopping can do on this. But the main issue to be concerned is the security of their data stored at untrusted cloud because the data may be corrupted accidently or may compromised by the cloud itself. Often the clients must verify the integrity and accuracy of their data. In this route, we have two existing techniques POR [1] (Proofs of retriveability) and PDP [2] (Proofs of data possession). These two were built under single server setting which results in a problem that if in case the server crashes, the whole of the data stored are lost and unable to recover or reconstruct. Thus extends to multi-servers. The schemes like MR-PDP [3] and HAIL [4] came to existence. MR-PDP and HAIL works on replication and erasure coding respectively. And these are suited for the small amount of data. The major limitations of these schemes are securing the large data are not possible and to recover the lost part of the data the complete data file has to be downloaded, thus consumes more time to repair the data file traffic in the network will be maximum.



Problems in Cloud Storage

Amazon cloud outage downs Netflix, Quora

The brief network connectivity issues strike Amazon's northern Virginia site, derailing many sites that depend on the widely used service.

by Steven Musil | August 8, 2011 9:35 PM PDT

Cloud Storage Often Results in Data Loss

12:06 AM ET | By: Chad Brooks, BusinessNewsDaily Contributor

9 OCT 2011

Flickr accidentally nukes user's 4,000 photos

By Laurie Segall, staff reporter February 2, 2011: 3:15 PM ET

NEW YORK (CNNMoney) -- It's every Flickr addict's worst nightmare: One day, the vast photo archive you've uploaded and annotated for years suddenly vanishes. It happened this week to Mirco Wilhelm, when a Flickr staff member accidentally deleted his five-year old account, wiping out 4,000 photos.

Figure 1: Problems in Cloud Storage

To relieve from above said problems many researches are undergone, and our paper proposes a scheme called DIP (Data Integrity Protection) to assure the integrity of the data stored and is suited for large volume of data. This DIP scheme has been proposed for the particular regenerating code which provides the fault tolerance. This code stripes the data across several servers in a randomly fashion with data encrypted. Thus integrity can be checked by verifying the random subsets of long-term archival storage from many servers. The scheme provides fault tolerance, efficient recovery, less traffic, and availability of data at low cost.

Figure 2: Cloud storage

Related work

As the days roll on, technologies are getting advance with both advantages and disadvantages. Likewise threats to our stored data are increasing and becoming more common. This results in corruption and data loss. To prevent this corruption and attacks many works are carried out. Prior the experiments were conducted under a single server setting. As said in introduction section POR and PDP are the schemes introduced to check the integrity of the data. This was considered for the long-term archival storage and static data. The drawback of these are only detection of the corrupted data are possible and recovering the data is not in the case the server is controlled by adversary fully. Thus the multi-servers are considered. In this route, HAIL is the one based on erasure coding where the whole file are needed to download to repair in case of server down. This is unable to apply directly for regenerating codes. The key differences found in these works are switching to multi-servers, but this inherits overhead of large storage due to direct adaption and for the portability and simplicity only standard read/write functionalities are supported.

Background

A.FMSR codes

FMSR (Functional minimum storage regenerating) codes are belonging to maximum separable (MDS) codes. The parameters n, k are used to define the MDS codes with the condition k < n, where k is number of servers used to store the files and n is the number of pieces the file is divided. The file of some size is encoded into few pieces each of equal size. The statement of the MDS codes are as follows: The reconstruction of the original file is possible out of n pieces from any k servers. This code divides the file into number of pieces over several servers. The size of each divided parts are equal. These small parts are called native chunks, further these are encoded to get a code chunks and are stored at several servers. During the data loss, these codes are decoded to get the original file. The goal of this is to reconstruct the lost data of the failed server in anew server, thus preserving the fault tolerance.

B. System Architecture

The architecture includes user, cloud interface and web applications. Two clouds are used for convenience but more can be deployed. Prior to upload and download, clients undergo authentication check for security purpose. Only authenticated users are permitted to perform the operations. The alterations in the data stored will notify to clients. Also the file can be generated without downloading it.

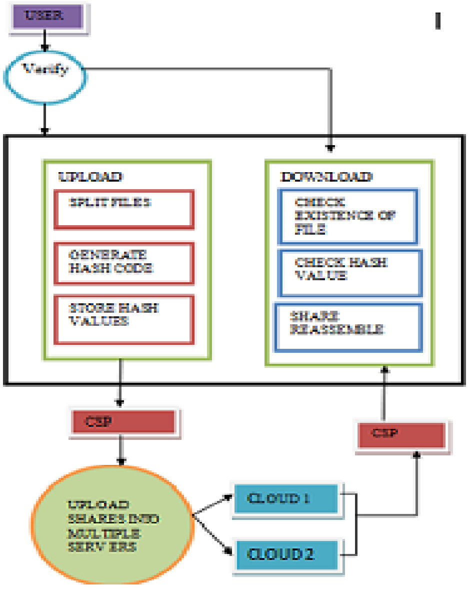


Figure 3: System Architecture

Proposed work

For the DIP feature, the different file operations like upload, download and repair are augmented. Additionally, the check operation is introduced for the verification of small chunk integrity. The consistencies of the chunks are checked by downloading the random rows from the servers.

Description of the operations is as follows:

The file is uploaded to the cloud. Prior to uploading the secret keys k_{ENC}, k_{PRF}, k_{PRF}, k_{MAC} are generated for the cryptographic primitives such as symmetric encryption, pseudorandom permutations, pseudo-andom functions and message authentication codes respectively. These keys are encrypted and are outsourced to cloud for the further security purpose. Then files are encoded by (n, k)-FMSR codes to generate the code chunks. Metadata is generated to keep the different related information. Further the code chunks are encoded with FMSR-DIP codes. The parity bits are generated by applying AECC to code chunks. AECC is applied to recover a corrupted row. Later PRF to row bytes is applied to protect the integrity of each row. Finally Mac is applied. All these parameters are updated in the metadata and are uploaded to the cloud.

Next is the Check operation to check whether the integrity i.e., the file is corrupted are not. In this operation the metadata copy is downloaded from each server and is check for its identicalness. Any corrupted data can be recovered as the metadata is replicated along different servers. And this metadata file is decrypted to retrieve the AECC, MACs, and encoding coefficients. The rows are sampled for the row verification by removing the PRFs with respect to encoding coefficients. This in turn generates the system’s equations I which by solving those, if the result got is equalities, then it is said the system is consistent. If not, then it is considered as it is having the errors in row. So any k servers are chosen and bytes are picked. The equations of system are generated from these bytes. By solving these solution found is unique. But this may not be correct always. Thus a chunk is chosen from remaining servers. The byte value and the encoding coefficients are appended and the bytes of subsets are considered and equations are solved to check the consistency. If it is found that they are equal, then it is said the system is consistent and marked correct. This is repeated for all other chunks. Finally if the bytes are marked a corrupted then the repair operation is triggered.

Next comes the Download operation. The metadata is checked for its identicalness, and the FMSR-DIP code chunks are downloaded and decoded from any k servers. Macs are used to verify the integrity. NCCloud decode the FMSR-DIP code chunks if it is not corrupted. If corrupted, AECC parity is downloaded and error correction is applied and verified with MACs. The code chunks are downloaded from all remaining servers. The subset of rows marked correct is recovered.

The last operation is the Repair operation. Here also the metadata is downloaded as in case of check operation. For repair, only the chunks which are corrupted are downloaded and decoded from any k servers. Finally the newly generated chunk is encoded and they are encrypted before they are uploaded. These are replicated to many servers.

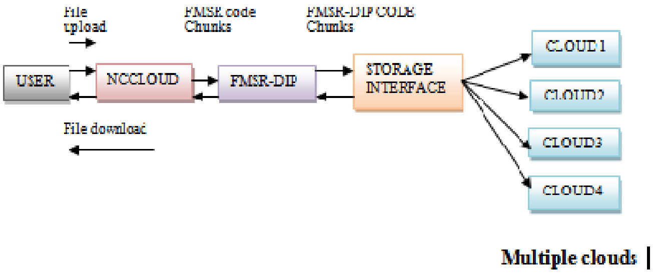


Figure 4: FMSR-DIP

Performance analysis

Service availability, attack against data intrusion and data integrity, and data integrity are provided by simulating the proposed solution. To improve the performance the user numbers are varied to two accounts. Our proposed work able to reduce the filtering and time consumption while downloading the file. Our work supported the migration of single cloud to multi-clouds because the security risk can be minimized by this at low cost.

Conclusions

It is important, that client remotely check the integrity of outsourced data in the cloud. Here FMSR-DIP is used to ensure the integrity of data which strips into multiple blocks (chunks) among multiple servers. It ensure the successful and efficient re-generation of lost data by extracting few random chunks from multiple servers (say 4 servers) where we stripe and store data into. In this paper strips (chunks) will be stored in the 4 servers. Also along with the integrity checking our work provides Fault tolerance and also allows recovering the data efficiently during data loss. In future we will extend it to more servers to upload our data. Also our work can be extended to secure the images and videos which we upload at the cloud storage.

References

1. A. Juels and B. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), 2007.
2. G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote Data Checking Using Provable Data Possession," ACM Trans. Information and System Security, vol. 14, article 12, May 2011.
3. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple- Replica Provable Data Possession," Proc. IEEE 28th Int'l Conf. Distributed Computing Systems (ICDCS '08), 2008.
4. K. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf. Computer & Comm. Security (CCS '09), 2009.
5. Y. Hu, Chen, P. Lee, and Y. Tang, "NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-Clouds," Proc.10th USENIX Conf. File and Storage Technologies (FAST '12),2012.
6. H.C.H. Chen and P.P.C. Lee, "Enabling Data Integrity Protection in Regenerating Coding-Based Cloud Storage", Proc. IEEE 31 st Symp. Reliable Distributed Systems (SRDS'12),2012.