ISSN: APPLIED
2018, Volume – 1; Issue – 1;
Page 50

REPLACE RECOVERY ALGORITHM APPROACH IN RECOVERING A FAILURE IN CLOUD ENVIRONMENT

Dr.V.Dheepa,Asst Professor,Department of Computer Science,
Anna Adarsh College for Women, Chennai-40
Ms.G.Prathima,Asst Professor, Department of Computer Science,
Anna Adarsh College for Women, Chennai-40

ABSTRACT

Cloud computing is a booming technology for large data storage applications, which will be affected by data security. The data will be loss in many cases. This paper concentrate data will be loss due to cloud failure. This provides a proxy-based storage system to overcome fault-tolerant in storage over multiple cloud storage providers. Cloud can interconnect different clouds and transparently stripe data across the clouds. The existing method first provided functional minimum-storage regenerating (FMSR) codes for NC (Network Code) Cloud environment. The FMSR code implementation maintains double-fault tolerance and has the same storage cost as in traditional erasure coding schemes based on RAID-6 codes, but uses less repair traffic when recovering a single-cloud failure. But which is not providing the efficient result for this scope. To overcome this, the replace recovery algorithm is introduced for the recovering process. With the help of this algorithm, the process can easily retrieve the cloud backup when any one cloud moves to the failure to the reasons such as the disaster, attack, or data loss and corruption.

INTRODUCTION

Cloud storage provides an on-demand remote backup solution. However, using a single cloud storage vendor raises concerns such as having a single point of failure and vendor lock-ins. As suggested in, a plausible solution is to stripe data across different cloud vendors. While striping data with conventional erasure codes performs well when some clouds experience short term failures or foreseeable permanent failures, there are real-life cases showing that permanent failures do occur and are not always foreseeable. This work focuses on unexpected cloud failures. When a cloud fails permanently, it is important to activate storage repair to maintain the level

of data redundancy. A repair operation reads data from existing surviving clouds and reconstructs the lost data in a new cloud. It is desirable to reduce the repair traffic, and hence the monetary cost, due to data migration.

Recent studies propose regenerating codes for distributed storage. Regenerating codes are built on the concept of network coding. They aim to intelligently mix data blocks that are stored in existing storage nodes, and then regenerate data at a new storage node. It is shown that regenerating codes reduce the data repair traffic over traditional erasure codes subject to the same fault-tolerance level. Despite the favorable property, regenerating codes are mainly studied in the theoretical context. It remains uncertain regarding the practical performance of regenerating codes, especially with the encoding overhead incurred in regenerating codes.

NEED FOR NEW SYSTEM

In cloud computing environment unexpected failures are occurred rarely. In that situation a repair operation for retrieving the data is done. Existing system uses FMSR regenerating code, for retrieving the information from the failure cloud. Proxy based storage system for providing fault tolerance over multiple cloud service providers. Maintain the proxy system and their data is complex one. Client and cloud system communicate through the proxy system, so the overall processing time is long.

This work introduces a remote recovery concept that can be reducing the overall processing time. It has an index file of each storing data, that index can be used to easily finding the location. FMSR Regenerating code requires storage for performing encoding operation. When compared to FMSR, Index file is not occupying the large amount of space. Remote Recovery mechanism is portable for all disks.

METHOD DESCRIPTIONS

1 FILE PROCESS

The first process is to select the files for loading process. Here, the files are to be load in to different server. The directory indexing information's of each and every file can be maintained in main server. But the files are stored in cloud storage. In this process the cloud server can contains all information about the servers.

2 SYSTEM PROCESS

System process means if the file can be downloaded by some user. For

downloading the files first they give the request to the server. According to the request the file can be downloaded from different server. But the storage process is not known to the users. The files are retrieved from different server. At the time of downloading if any one of the servers shall be shutdown. For that purpose this process goes for a greedy approach.

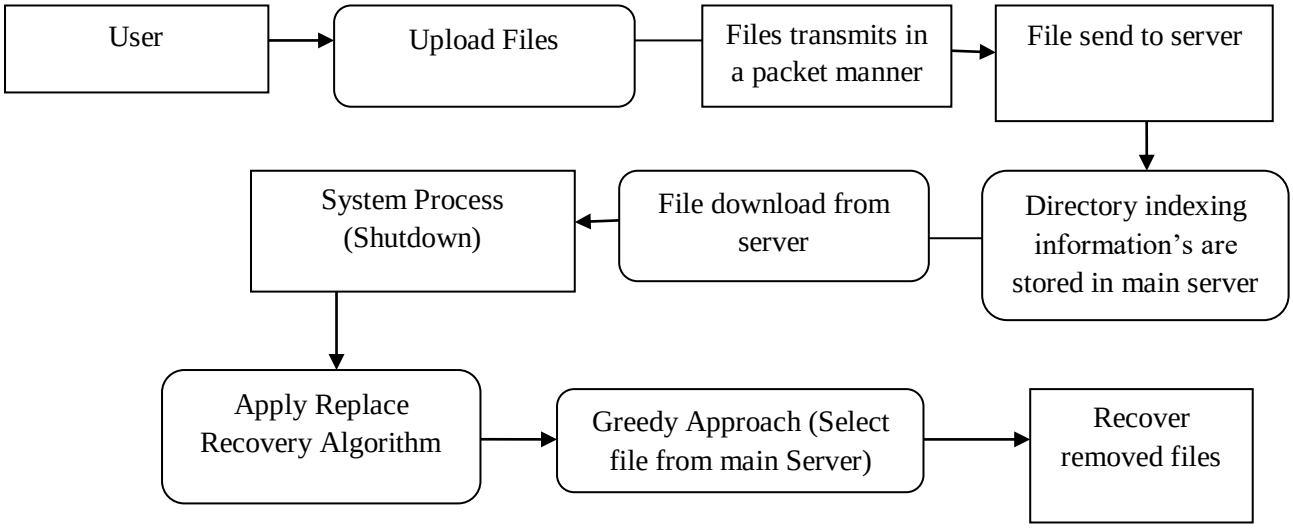
3 GREEDY APPROACH

Greedy approach mainly used for storage process. It starts with a feasible recovery solution, and incrementally replaces the current solution with another one that reads less data. This approach mainly used at the time of destroying the files from the server. Before presenting the replace recovery algorithm, a primitive function is need to determines if one server is valid to resolve the data symbols after being replaced with other parity symbols in another server. For each parity symbol, a bit encoding vector that specifies how the strip of lost data symbols is encoded to the parity symbol.

4.RECOVERY PROCESS

The recovery process use replace recovery algorithm. The directory information's are stored in every server. So the proposed system can replace the files from destroyed servers. This system simplified recovery model states that there exists a recovery solution that contains exactly parity symbols for regenerating lost data symbols for each server failure. Computationally efficient replace recovery algorithm that seeks to minimize the number of read symbols for single disk failure recovers. The whole process architecture will be shown in figure1.

Fig 1. System Architecture



ALGORITHM JUSTIFICATION

The replace recovery algorithm can be used to recover the failure of server's information's in the cloud. Using this algorithm the files are recovered which can be stored in the failure server. This replace recovery algorithm can have some objectives like search efficiency, effective recovery performance and adaptable to heterogeneous network system. So this algorithm finds recovery solution with polynomial complexity. In this process the main server can contains all servers directory indexing information's. According to the process of replace recovery algorithm the files are obtained from servers. This algorithm provides optimal recovery performance in the cloud.

A greedy algorithm is a mathematical process that looks for simple, easy-to-implement solutions to complex, multi-step problems by deciding which next step will provide the most obvious benefit. Greedy algorithms work by recursively constructing a set of objects from the smallest possible constituent parts. Recursion is an approach to problem solving in which the solution to a particular problem depends on solutions to smaller instances of the same problem. The advantage to using a greedy algorithm is that solutions to smaller instances of the problem can be straightforward and easy to understand.

METHODOLOGY DISCUSSIONS

By using greedy approach along with replacement recovery algorithm is worked well. It reduces the recovery time and cost. The replacement recovery algorithm recovers the disk such as cloud based upon the parity values. Because of all the data's are present like binary format in the cloud. The primary objective is to minimize the amount of data read from the surviving disks for recovery and hence the overall time of the recovery operation, while the recovery solution can be quickly determined. This paper proposes a replace recovery algorithm, which uses a hill-climbing (greedy) approach to optimize the recovery solution. It starts with a feasible recovery solution, and incrementally replaces the current solution with another one that reads less data. The proposed system validate that it provides near-optimal recovery for different variants of FMSR codes. Also, it is shown to achieve polynomial complexity. Note that the replace recovery can be extended

for the setting where disks are heterogeneous with different performance costs. This implies that the replace recovery can be applied based on the current performance costs of surviving disks, while existing enumeration recovery is infeasible in doing so due to its exponential complexity.

CONCLUSION

In this work, the fault-tolerant storage was provided by a proxy based storage system. In that the functional minimum-storage regenerating (FMSR) codes used for double tolerant over multiple cloud storage. The proposed concept used replace recovery algorithm which is used for recovering the cloud data when the cloud damaged. This algorithm recovers the data in a cloud and also replaces same content. In this approach all the data are retrieved in proper manner. This algorithm is very useful for recovering important data from the natural disasters, loss and damage. The initial experimental results show that the proposed algorithm efficiently recovers the backup during the corruption and the cost reduced.

REFERENCES

1. M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A.G. Dimakis, R.Vadali, S. Chen, and D. Borthakur, "XORing Elephants: Novel Erasure Codes for Big Data," Proc. VLDB Endowment, vol. 6, pp. 325-336, 2013.
2. B. Schroeder and G.A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" Proc. Fifth USENIX Conf. File and Storage Technologies (FAST '07), Feb. 2007.
3. K. Shum, "Cooperative Regenerating Codes for Distributed Storage Systems," Proc. IEEE Int'l Conf. Communications (ICC '11), June 2011.
4. K. Shum and Y. Hu, "Exact Minimum-Repair-Bandwidth Cooperative Regenerating Codes for Distributed Storage Systems," Proc. IEEE Int'l Symp. Information Theory (ISIT '11), July 2011.
5. C. Suh and K. Ramchandran, "Exact-Repair MDS Code Construction Using Interference Alignment," IEEE Trans. Information Theory, vol. 57, no. 3, pp. 1425-1442, Mar. 2011.

6. R. Wauters,, "Online Backup Company Carbonite Loses Customers' Data, Blames and Sues Suppliers," http://techcrunch.com/ 2009/03/23/online-backup-company-carbonite-loses-customersdata- blames-and-sues-suppliers/, Mar. 2009.
7. R. Kossman, "Cloud Case Studies: Data Storage Pros Offer First-Hand Experiences," http://searchcloudstorage.techtarget.com/ feature/Cloud-case-studies-Dat a-storage-pros-offer-first-handexperiences/, 2013.
8. M. Vrable, S. Savage, and G. Voelker, "Cumulus: Filesystem Backup to the Cloud," Proc. USENIX Conf. File and StorageTechnologies (FAST '09), 2009.
9. M. Vukoli_c, "The Byzantine Empire in the Intercloud," ACM SIGACT News, vol. 41, pp. 105-111, Sept. 2010.
10. Z. Wang, A. Dimakis, and J. Bruck, "Rebuilding for Array Codes in Distributed Storage Systems," Proc. IEEE GlobeCom Workshops, 2010.
11. L. Xiang, Y. Xu, J. Lui, Q. Chang, Y. Pan, and R. Li, "A Hybrid Approach to Failed Disk Recovery Using RAID-6 Codes: Algorithms and Performance Evaluation," ACM Trans. Storage, vol. 7, no. 3, article 11, 2011.
12. J. Clark, "AWS Cloud Accidentally Deletes Customer Data," http://www.zdnet.com/aws-cloud accidentally-deletescustomer- data-30400 93665/, Aug. 2011.
13. Python, "zfec 1.4.24," http://pypi.python.org/pypi/zfec, 2013.
14. J.S. Plank, J. Luo, C.D. Schuman, L. Xu, and Z. Wilcox-O'Hearn, "A Performance Evaluation and Examination of Open-Source Erasure Coding Libraries for Storage," Proc. Seventh USENIX Conf. File and Storage Technologies (FAST '11), 2009.