ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

SNSC: Secure and Dependable Network Coding For the Storage Repair in a Cloud-Of-Clouds

Mohan Kumar G[1], Neelu L[2]

M.Tech(PG Scholar), Dept of CSE, RRCE, Bengaluru, India[1]

Rector, Dept of CSE, RRCE, Bengaluru, India[2]

ABSTRACT: Here, we present a proxy based storage system for fault tolerance multiple-cloud storage called NC(network coding), which overcome problems such as permanent failure and loss of data, lost data is repaired with the help of data redundancy. NC achieves cost-effective repair for a permanent single-cloud failure, it is built on top of networking coding based storage scheme called the storage regenerating code(SR) unlike traditional RAID-6 used for fault tolerance and data redundancy SR use less repair traffic and hence incur less monetary cost, and greater response time performance in normal cloud operation such as, upload/download. Implementation of a proof-of-concept prototype of NC and deploy it atop both local and commercial cloud. Proof-of-concept is designed to determine feasibility ,but does not represent deliverables is also known as proof of principle. It is used to check system requirements, such as how system can be integrated or throughput can be achieved through a given configuration. Key feature of SR code is that we release the encoding requirement of storage nodes during repair, to make regenerating code portable to any cloud storage it is desirable to assume only a thin-cloud interface, where storage node only need to support the standard read/write functionalities.

KEYWORDS: Cloud Computing, Storage regenerating code, network coding.

I. INTRODUCTION

Cloud computing denotes a family of increasingly popular on-line services for archiving, backup, and even primary storage of files, and transforming business by offering new options for businesses to increase efficiencies while reducing costs [2]. It lets user can access all applications and documents from anywhere in the world, freeing from the confines of the desktop and making it easier for group members in different locations to collaborate.
It is a model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal consumer management effort or service provider interaction. Cloud computing provides computation, software, data access, and storage resources without requiring cloud users to know the location and other details of the computing infrastructure.

Cloud storage provides data on-demand and solution. A plausible solution is to stripe data across different cloud providers, by exploiting the diversity of multiple clouds, the fault tolerance of cloud storage[3]. While striping data with conventional erasure, codes performs well when some clouds experience short-term transient failures or foreseeable permanent failures, there is real-life case showing that permanent failures do occur and are not always foreseeable. a distributed cryptographic system that permits a set of servers to prove to a client that a stored file is intact and retrievable. Storage providers charge users for outbound data, so moving an enormous amount of data across clouds can introduce significant monetary costs. It is important to reduce the repair traffic .To minimize repair traffic, storage regenerating codes have been proposed for storing data redundantly in a distributed storage system.

A proxy-based storage system is designed for providing fault-tolerant storage over multiple cloud storage providers. NC can interconnect different clouds and transparently stripe data across the clouds. On top of NC, we propose the first implementable design for the storage regenerating (SR) code[1].

The main disadvantage associated with the existing system [1] is that:

1. Storage Repair
2. fault tolerance.

The drawback of the existing system can be overcome using this proposed system. The main contribution of this paper is that Proposed system to identified the most representative fault tolerance and storage repair.. There are several systems proposed for multiple-cloud storage. HAIL provides integrity and availability guarantees for stored data. DESPKY uses erasure coding to mitigate vendor lock-ins when switching cloud vendors. It retrieves data from the cloud that is about to fail and moves the data to the new cloud. Several studies propose efficient single node failure recovery schemes that minimize the amount of data read (or I/Os) for XOR-based erasure codes. For example, the authors of propose optimal recovery for specific RAID-6 codes and reduce the amount of data read by up to around 25 percent (compared to conventional repair that downloads the amount of original data) for any number of nodes. Note that our SR codes can achieve 25 percent saving when the number of nodes is four, and up to 50 percent saving if the number of nodes increases. NC can interconnect different clouds and transparently stripe data across the clouds, 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.

The main advantage of the proposed system[1] is that:

1. Fault tolerance among clouds.
2. Data backup and recovery.
3. Regenerating codes.
4. Iterative Repairs.
5. Repair operations among the cloud.

II. ARCHITECTURE

The system design contains 6 main parts namely,
Data Owner, Proxy Server, Cloud Server, NC Cloud,
Data Consumer(End User), Threat Model (Attacker).

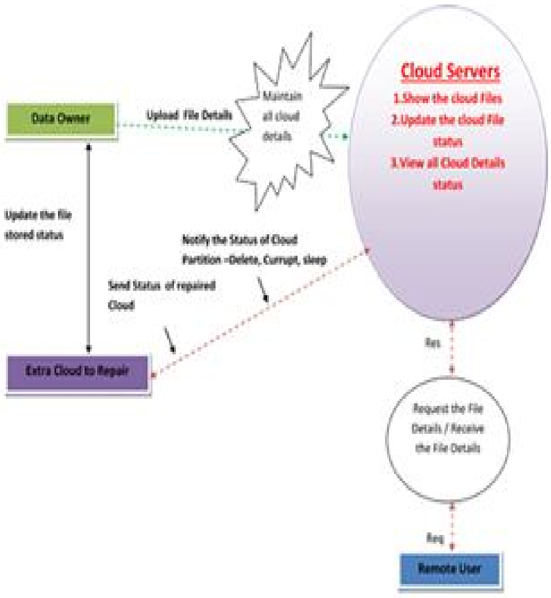
Cloud Servers
1.Show the cloud Files
2.Update the cloud File status
3.View all Cloud Details status
Data Owner
Upload File Details
Maintain all cloud details
Update the file stored status
Notify the Status of Cloud Partition –Delete, Currupt, sleep
Send Status of repaired Cloud
Extra Cloud to Repair
Res
Request the File Details / Receive the File Details
Req
Remote User

Fig 1. Architecture

- **Data Owner**

The data owner uploads, their data in the cloud server. For the security purpose the data owner splits file into four packets, encrypts the data file and then store in the multiple clouds. The Data owner can have capable of manipulating the encrypted data file.

- **Proxy Server**

The Proxy server is a proxy-based design that interconnects multiple cloud repositories. The proxy serves as an interface between client applications and the clouds. If a cloud experiences a permanent failure, the proxy activates the repair operation. That is, the proxy reads the essential data pieces from other surviving clouds, reconstructs new data pieces, and writes these new pieces to a new cloud.

- **Cloud Server**

The cloud service provider manages a cloud to provide data storage service. Data owner encrypts and splits the data files and store them in the multiple clouds (cs1, cs2, cs3 and cs4) for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them.

- **NC (Network Coding)**

NC acts as a proxy server, that bridges user applications and multiple clouds. Its design is built on three layers. The file system layer is present as a mounted drive on NC Cloud which can thus, be easily interfaced with general user applications. The coding layer deals with the encoding and decoding functions. The storage layers deals with read/ write requests with different clouds. If any unauthorized user is modify the file in a cloud server then NC cloud regenerate that file and send to Remote user via newly created cloud (cloud server 5).

- **Data Consumer(End User)**

The end user can only access the data file with the encrypted key to access the file. Then Proxy based NC cloud combines all the packets and sends to Remote user. Users may try to access data files within the cloud only.

- **Threat Model (Attacker)**

Attacker can attempt to transient failure for a cloud by making Doze Off for a particular period of time. The Attacker can also attempts to permanent failure by Deleting and corrupting the cloud. Then the Unauthorized user will considered as an attacker.

III. SR CODE IMPLEMENTATION

SR codes is that we do not require lost chunks to be
exactly reconstructed, instead we generate code chunks that are not necessarily identical to those originally stored in failed node, as long as the MSD
(maximum distance separable) property holds good.

The SR implementation consist of three operations:

- ➢ File Upload.
- ➢ File Download.
- ➢ Repair.

3.1 File Upload

To upload file, we first divide the obtained file into equal size known as native chunks. Then we
encode this native chunks into code chunks, which is
formed by a liner combination of native chunks. We an encoding coefficient vector(ECV) which contain metadata i.e. native chunks. The code chunks are then evenly stored in the storage nodes.

3.2 File Download

To download a file, we first download the corresponding metadata object that contains the ECVs. Then we select any node of the storage nodes, and download the native chunks from the code chunks. the user first adopts its private key to compute a signature. Then, the user sends a data request containing to the cloud server Upon receiving the request, the cloud server checks the validity of the signature and reforms a revocation verification according to the revocation list .After successful verification, the cloud server responds to the corresponding data file and the revocation list to the user.

3.3 Repair

Unlike the traditional i.e. RAID-6, we propose two phase checking in SR codes. which ensures that the code chunks on all nodes always satisfy the MDS property, hence data is availability even after repairs. First we operate on single permanent node failure, if the new set of chunks in all storage nodes satisfies the MDS property after the repair these procedure is carried out for multiple iterations . For recovery of data from nodes we design a mutually cooperative recovery (MCR) mechanism for multiple node failures. Via a cut-based analysis of the information flow graph, we obtain a lower bound of maintenance bandwidth based on MC R. For MCR, we also propose a transmission scheme and design a linear network coding scheme based on strong-MDS code, which is a generalization of MDS code. We prove that the maintenance bandwidth based on our transmission and coding schemes matches the lower bound, so the lower bound is tight and the transmission scheme and coding scheme for MCR are optimal[4].

IV.CONCLUSION

The problem statement of the project is to provide fault tolerance for cloud storage and to study propose to stripe data across multiple cloud vendors. However, if a cloud suffers from a permanent failure and loses all its data, we need to repair the lost data with the help of the other surviving clouds to preserve data redundancy. Hence we make us of NC (network code), a proxy-based server, multiple-cloud storage system that technically addresses the reliability of cloud backup storage. NC not only provides fault tolerance in storage, but also allows cost-effective repair when a cloud permanently fails. It implements a practical version of SR codes, which regenerates new parity chunks during repair. Ensuring that the new set of stored chunks after each round of repair preserves the required fault tolerance.

REFERENCES

[1] Henry C.H. Chen, Yuchong Hu, Patrick P.C. Lee, and Yang Tang, “NCCloud: A Network-Coding - Based Storage System in a Cloud-of-Clouds, ” IEEE Tran. On Computers, Vol.63, No. 1, January 2014.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph R.H Katz,A. Konwinski, G. Lee, A.D Patterson, A. Rabkin, I Stoica, and M. Zaharia “ A View of Cloud Computing,” comm.ACM vol. 53, no. 4, pp. 50-58, April 2010

[3] K.D. Bowers, A. Juels , and A. Opera, “ HAIL: A High-Availability and Integrity Layer for Cloud Storage,” Proc. 16th ACM Conf. Computer and Comm. Security(CCS '09), 2009.

[4] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, “Cooperative Recovery of Distributed Storage Systems from Multiple Losses with Network Coding,” IEEE J. Selected Areas in Comm., vol. 28, no. 2, pp. 268-276, Feb. 2010.