International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Survey on CloudNCFS: Building Network-Coding-Based Distributed Storage System

Pranjali A. Pali[1], N. D. Kale[2]

[1]SavitriBai Phule Pune University, PVPIT College of Engineering, Bavdhan, Pune, India

[2]Professor, SavitriBai Phule Pune University, PVPIT College of Engineering, Bavdhan, Pune, India

Abstract: ***NCCloud is a proof of concept prototype of a network coding based file system that aims at providing fault tolerance and reducing data repair cost when storing files using multiple cloud storage. NCCloud is a proxy based file system that interconnects multiple (cloud) storage nodes. It can be mounted as a directory on Linux, and file uploading/downloading are done by copying files to/ from the mounted directory. NCCloud is built on FUSE, an open source programmable user space file system that provides application programmable interface (APIs) for file system operations. From the point of view of user application, NCCloud present a file system network codes for storage repair require that storage nodes encode the stored data during the repair process. However, this may not be feasible for some storage systems where nodes only provide the basic I/O functionalities but do not have the encoding capability. work is to adapt the benefits of network codes in the storage repair of a practical storage setting, by relaxing the encoding requirement of storage nodes. Yet that transparently stripes data across storage NCCloud support variety of coding schemes, in particular the functional minimum storage generating (F-MSR) code. Compared to traditional optimal erasure codes, FMSR codes maintains the same storage overhead under the same data redundancy level, but uses less repair traffic during the recovery of a single failed node. NCCloud realizes regenerating codes in a practical cloud storage system that does not require any encoding/decoding intelligence on the cloud storage nodes.***

Keywords: Encoding, Fault tolerant system, network coding, experimentation, regenerating codes recovery implementation

1. Introduction

A single cloud storage provider encounters the problem such as a single point of failure. The general solution is to distribute data across different cloud providers. The fault-tolerance can be improved by the diversity of multiple clouds. One key feature of distributed storage is data reliability, which generally refers to the redundancy of data storage. Specifically, given the pre-determined level of redundancy, the distributed storage system must sustain normal I/O operations within a tolerable number of node failures. In addition, in order to maintain the required data redundancy, the storage system must support data repair, which involves reading data from existing nodes and reconstructing essential data in the new nodes. fast data repair schemes based on network coding[2] for distributed storage systems. Such network-coding-based schemes, or called regenerating codes.

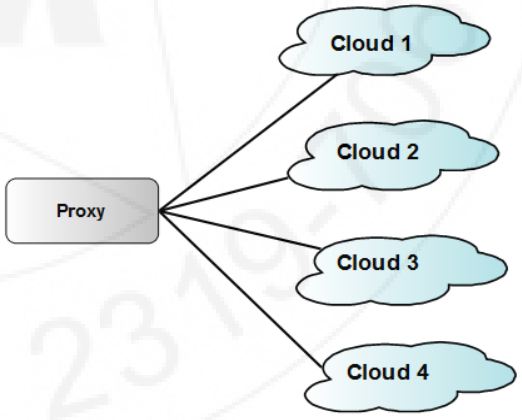
NCCloud is a proof-of-concept prototype of a network-coding-based file system that aims at providing fault tolerance and reducing data repair cost when storing files using multiple–cloud storage.

1.1 Need to repair in multiple cloud storage

The repair in multiple cloud storage is needed, especially in disastrous cloud failures. There are two types of failures: transient failure and permanent failure. Transient failure: transient failure are common, short-term and if it "failed" cloud will return to normal after some time period, no data are lost. Permanent failure: permanent failure is disastrous, long-term and if it "failed" cloud will become permanently unavailable.

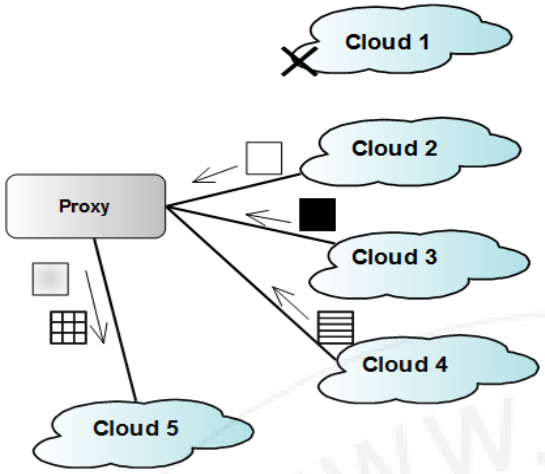
2. Techniques

The Design of NCCloud a proxy-based file system for long-term archival using multiple cloud storage providers. It Propose an implementable design of functional minimum storage generating code (F-MSR) which adapts the benefits of network coding in minimizing the cost of repairing a single-cloud failure, while preserving storage overhead as in erasure codes.



(a) Normal operation

NCCloud Maintain redundancy with Maximum Distance Separable (MDS) property Focus on double fault-tolerance called as Proxy-based design [1],[3].NCCloud serves as an interface between client & clouds. As shown in the fig (a) design interconnects multiple cloud repositories. If any cloud experience permanent failure then proxy activate the repair operation shown in fig (b).Repair operation is done. The proxy reads data from survival clouds, reconstructs new data, and writes them to a new cloud.



(b) Repair operation

3. Literature Survey

3.1 Different Coding Scheme

RAID-6 code: Let us first consider RAID-6 codes, which are double-fault tolerant. A RAID-6 code implementation based on the Reed-Solomon code as suggested in [1][54]. First divide the file into two native chunks of size M=2 each and add two code chunks formed by the linear combinations of the native chunks. Suppose that Node 1 is down. Then, the proxy must download the same number of chunks as the original file from two other nodes. It then reconstructs and stores the lost chunk on the new node. The total storage size is 2M, while the repair traffic is M.

EMSR Code: Let consider double-fault-tolerant implementation of EMSR codes. First divide a file into four chunks, and allocate the native and code chunks. Suppose Node 1 is down. To repair it, each surviving node sends the XOR summation of the data chunks to the proxy, which then reconstructs the lost chunks. EMSR codes, the storage size are 2M (same as RAID-6 codes), while the repair traffic is 0:75M, which is 25 percent of saving (compared with RAID-6 codes). EMSR codes leverage the notion of network coding [2], as the nodes generate encoded chunks during repair.

FMSR Code: Consider the double-fault-tolerant implementation of FMSR codes and divide the file into four native chunks, and construct eight distinct code chunks formed by different linear combinations of the native chunks. Each code chunk has the same size M/4 as a native chunk. Any two nodes can be used to recover the original four native chunks. If Node 1 is down. The proxy collects one code chunk from each surviving node, so it downloads three code chunks of size M=4 each. Then, the proxy regenerates two code chunks formed by different linear combinations of the three code chunks. The proxy then writes in the new node.

4. Related Work

This In this paper, for implementing FMSR Code. For particular file object, specify three operations for FMSR Codes: (i) File upload operations with no failure, (ii) File download operations with node failures,(iii)Repair operations during node failures.

File upload: In the Upload operation, NCCloud generates code chunks for a file based on FMSR codes. The code chunks will be temporarily stored in the local file system instead of being uploaded to the servers as in [4].

File download: In Download operation, the idea is to treat FMSR Code as standard Reed Solomon codes, and the creating an inverse matrix technique is to decode the original data has described in tutorial[5].

Repair: The Repair operation of failed node includes three steps first is transmission of the existing blocks from survival nodes to NCCloud, second one regeneration for lost blocks of the failed node in NCCloud and third one transmission of the regenerated blocks from NCCloud to a new node[3]. If there is more than one failed node, then apply the repair operation for each failed node one by one

5. Future Implementation

We present approaches to address the challenges associated with the NCCloud. NCCloud does not support the Check operation. The Check cost is composed of the download traffic cost and the GET request cost. To minimize the download traffic cost, we can reduce the checking percentage. To minimize the GET request cost, we can set a larger check block size in order to save on the per request cost, with a trade-off of a lower probability of detecting corruptions .On the other hand, we can increase the checking percentage together with the check block size while maintaining the same Check cost.

6. Conclusion

In this paper introduce NCCloud is a proxy based file system that interconnect multiple (cloud) storage nodes. It can be mounted as a directory on Linux and file uploading/downloading are done by copying files to/ from the mounted directory. NCCloud present a file system network codes for storage repair require that storage nodes encode the stored data during the repair process. NCCloud support variety of coding schemes, in particular the functional minimum storage generating (F-MSR) code. Compared to traditional optimal erasure codes, FMSR codes maintains the same storage overhead under the same data redundancy level, but uses less repair traffic during the recovery of a single failed node

7. Acknowledgment

It is my privilege to acknowledge with deep sense of gratitude to my guide MR. N.D. Kale for his kind help and Co-operation. I am highly obliged to the entire staff of the Computer Engineering department for their kind co-operation and help. I also take this opportunity to thanks my colleague, who backed our interest by giving useful suggestions and all possible help.

References

[1] Henry C. H.Chen, Yuchong Hu,Patrick P.C.Lee, and Yang Tang. NCCloud: Applying Network Coding for the Storage System in a Cloud-of-Clouds. IEEE Trans. On computers,vol-63,no-1,January 2014
[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. IEEE Trans. on Information Theory, 46(4):1204–1216, Jul 2000.\
[3] Y. Hu, C.-M. Yu, Y.-K. Li, P.P.C. Lee, and J.C.S. Lui, "NCFS: On the Practicality and Extensibility of a Network-Coding-Based Distributed File System," Proc. Int'l Symp. Network Coding (NetCod '11), 2011.
[4] Y. Hu, H. Chen, P. Lee, and Y. Tang. NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-Clouds. In Proc. Of USENIX FAST, 2012
[5] J.S. Plank, "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-Like Systems," Software—Practice & Experience, vol. 27, no. 9, pp. 995-1012, Sept. 1997.
[6] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems," Proc. ACM Workshop Cloud Computing Security Workshop (CCSW '10), 2010.
[7] H.C.H. Chen and P.P.C. Lee, "Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage," Proc. IEEE 31st Int'l Symp. Reliable Distributed Systems (SRDS '12), 2012.
[8] A.G. Dimakis, P.B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," IEEE Trans. Information Theory, vol. 56, no. 9, pp. 4539-4551, Sept. 2010.
[9] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," Proc. IEEE INFOCOM, 2005.
[10] Y. Hu, P.P.C. Lee, and K.W. Shum, "Analysis and Construction of Functional Regenerating Codes with Uncoded Repair for Distributed Storage Systems," Proc. IEEE INFOCOM, Apr. 2013.
[11] 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.
[12] Y. Hu, C.-M. Yu, Y.-K. Li, P.P.C. Lee, and J.C.S. Lui, "NCFS: On the Practicality and Extensibility of a Network-Coding-Based Distributed File System," Proc. Int'l Symp. Network Coding (NetCod '11), 2011.
[13] A. Kermarrec, N.L. Scouarnec, and G. Straub, "Repairing Multiple Failures with Coordinated and Adaptive Regenerating Codes," Proc. Int'l Symp. Network Coding (NetCod '11), June 2011