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. 4, Issue 6, June 2016

A Survey on Cauchy Coding Approach to Optimize Fault Tolerance on Cloud Storage Systems Using HDFS

Krishna urf Vibha V Dambal

M. Tech Student, Dept of CSE, UBDT College of Engineering, Davanagere, Karnataka, India

ABSTRACT: Cloud storage generally provides different redundancy configuration to users in order to maintain the desired balance between performance and fault tolerance. Data availability is critical in distributed storage systems, especially when node failures are prevalent in real life. This paper explores recovery solutions based on regenerating codes, which are shown to provide fault-tolerant storage and minimum recovery bandwidth. It presents specification of algorithm for write operation and how to implement it.We implement Cauchy coding and evaluate our prototype atop of a Hadoop HDFS evaluate its performance by comparing with “Hadoop-EC” developed by Microsoft research. Our experimental results indicate that CaCo can obtain an optimal coding scheme within acceptable time. Furthermore, CaCo out performs Hadoop-EC by 26.68-40.18% in the encoding time and by 38.4-52.83% in the decoding time simultaneously.

KEYWORDS: Cloud storage, data availability, Cauchy coding, fault tolerance, XOR scheduling

I.INTRODUCTION

One of the biggest challenges in designing cloud storage systems is providing the reliability and availability that users expect. Once their data is stored, users expect it to be persistent forever, and perpetually available. Unfortunately, in practice there are a number of problems that, if not dealt with, can cause data loss in storage systems.

So, the failure protection offered by the standard RAID levels has been no longer sufficient in many cases, and storage designers are considering how to tolerate larger numbers of failures [1] Technology shifts and market forces are changing the composition and design of storage systems. Topics for this diverse issue include the emergence of nonvolatile storage technologies, virtualization technologies that reduce the distinction between storage and computing platforms, advances in tape densities, the growing use of commodity and distributed storage, and the increasing importance of error and disaster recovery, autonomic storage management, petascale file and archival storage, and long-term data preservation.[2] Cloud services inevitably fail: machines lose power, networks become disconnected, pesky software bugs cause sporadic crashes, and so on. Unfortunately, failure recovery itself is often faulty; e.g. recovery can accidentally recursively replicate small failures to other machines until the entire cloud service fails in a catastrophic outage, amplifying a small cold into a contagious deadly plague.

Cauchy Reed-Solomon (CRS) codes improve Reed-Solomon codes by using neat projection to convert Galois Field multiplications into XOR operations [3]. Currently, CRS codes represent the best performing general purpose erasure codes for storage systems. In addition, CRS coding operates on entire strips across multiple storage devices instead of operating on single words. In particular, strips are partitioned into w packets, and these packets may be large. Figure 1 illustrates a typical architecture for a cloud storage system with data coding. The redundancy configuration of the system is $k = 4$ and $m = 2$. With CRS codes, k data blocks are encoded into m coding blocks. In such a way, the system can tolerate any m disk failures without data loss. Note that those k data blocks and m coding blocks should be stored on different data nodes. Otherwise, the failure of one node may lead to multiple faults in the same group of $n = k + m$ blocks.

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. 4, Issue 6, June 2016

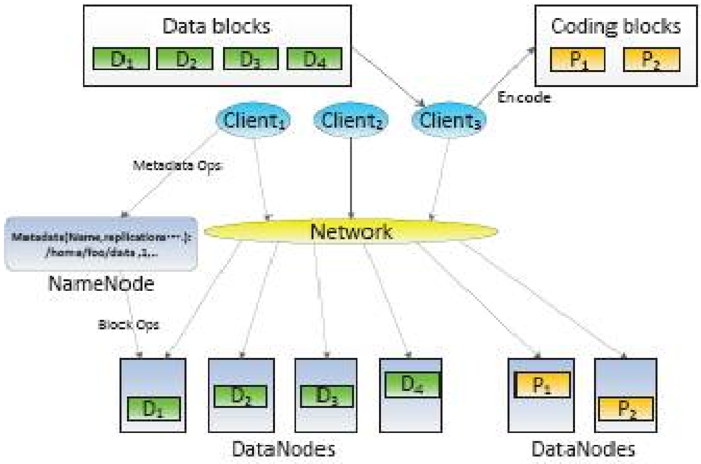

Fig. 1: A distributed architecture for a cloud storage system with Data coding, where k = 4, and m = 2.

II. RELATED WORK

In this section, we first give a general overview of *Cauchy Reed- Solomon (CRS) Coding*. Then, we provide a description in brief of the research and related work on generating Cauchy matrices and encoding with schedules. From these descriptions, we can make clearer the motivation of our work.

2.1 Cauchy Reed-Solomon Coding

Reed-Solomon (RS) codes [4] are based on a finite field, often called *Galois field*. When encoding data using RS codes, to implement a Galois filed arithmetic operation (addition or multiplication) requires many computations, so the performance is often unsatisfactory. CRS [3] codes modify RS codes and give two improvements. First, CRS codes use a Cauchy matrix instead of a Vandermonde matrix [5]. Second, CRS codes convert Galois field multiplications into XOR operations.

The key to CRS codes is construction of Cauchy matrices, and we can achieve that in the following way. Given a redundancy configuration $(k; m; w)$ where $k+m \leq 2w$, let $X = \{x1; : : : ; xm\}$, $Y = \{y1; : : : ; yk\}$, and $X \cap Y$ = _, so that each xi and yj is a distinct element of $GF(2w)$. Then we calculate the Cauchy matrix in element $(i; j)$ using $1=(xi + yj)$ (the addition and division are defined over Galois field) [3]. Since the elements of GF($2w$) are the integers from zero to $2w - 1$, each element e can be represented by a w-bit column vector, $V(e)$, using the primitive polynomial over Galois Field. Furthermore, each element e of GF($2w$) can be converted to a $(w \times w)$ binary matrix, $M(e)$, whose i-th($i = 1; : : : ; w$) column is equal to the column vector $V(e2i-1)$ [6]. Thus according to the value of w, we can transform the Cauchy matrix into a $(mw \times kw)$ binary matrix, denoted as A.

We divide every data block X and erasure codes block B into w trips. In this way, when there exists "1" in every row of A, we can do XOR operations on the corresponding data in X, to obtain the elements of B. As Figure 2 shows [7], the erasure codes require 11 XOR operations.

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. 4, Issue 6, June 2016

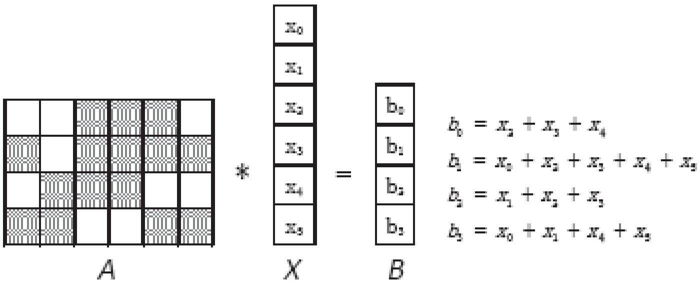

Fig. 2: Erasure Coding using Cauchy Reed-Solomon codes.

2.2 Observations and Motivation of Our Work

Cloud systems always use different redundancy configurations (i.e., (*k; m; w*)), depending on the desired balance between performance and fault tolerance. Through the preceding discussions and a number of experiments and analyses, we get some observations as follows.

• For different combinations of matrix and schedule, there is a large gap in the number of XOR operations.

• No one combination performs the best for all redundancy configurations.

• With the current state of the art, from the (2*w* *k*+*m*) (*k*+*mk*) Cauchy matrices, there is no method discovered to determine which one can produce the best schedule.

• Giving a Cauchy matrix, different schedules generated by various heuristics lead to a great disparity on coding performance.

• For a given redundancy configuration, it is with very low probability that one coding scheme chosen by rules of thumb performs the best. In view of the problems above, it is necessary to discover an efficient coding approach for a cloud storage system. And this approach is desired to be able to identify the optimal coding scheme in the current state of the art, for an arbitrary given redundancy configuration.

Fig.3: erasure code to encode k disks of data onto m = n − k disks of coding [8]

III.EMLPOYING CaCo IN HDFS

To achieve data redundancy, HDFS requires triple replication. In CaCo, we use erasure codes instead of triple replication of data to guarantee the fault tolerance of the system. To implement CaCo, we should modify the architecture of HDFS to certain extent. For example, the illustration of write operation [7] with CaCo is shown below and we can conclude the procedure to several steps as follows.

Algorithm 1: Write Operation with CaCo

1 The Client sends a write request to the NameNode.
2 The NameNode allocates some DataNodes to the Client.
3 Write the data blocks into DataNodes.
4 Make a copy of data and put it into DataQueue.
5Encode data with the schedule selected by CaCo.
6 Write the coding blocks into DataNodes.
7Data encoding finishes.
8 Remove the copies of data from DataQueue.

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. 4, Issue 6, June 2016

The above algorithm for write operation can be shown in pictorial representation of how data will flow from name node to data node.

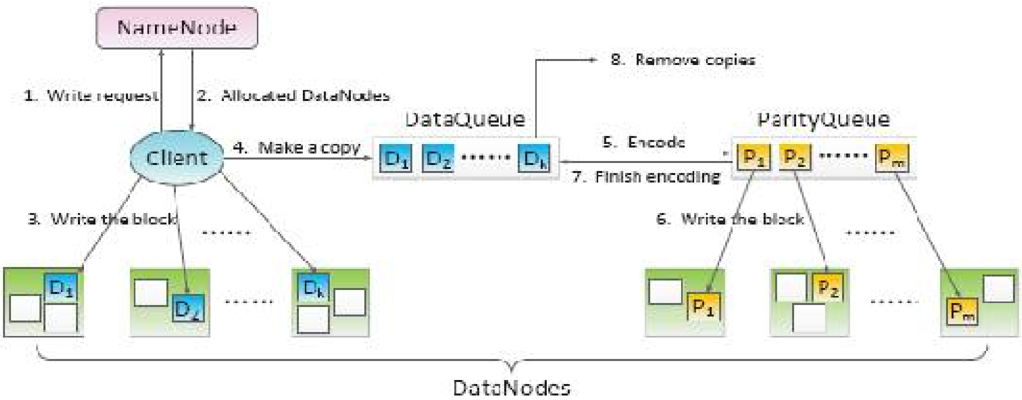

Fig. 4: Data flow of the write operation with CaCo.

Data flow diagram:

Data flow diagrams are most important and fundamental design structures in analyzing a problem and its solution. In this section, we have focused with a detailed data flow diagram design and analysis. The system consists of a server and node establishment and monitoring manager. As the system is initiated with a protocol of performing the designed task, the network registers it for deeper support.

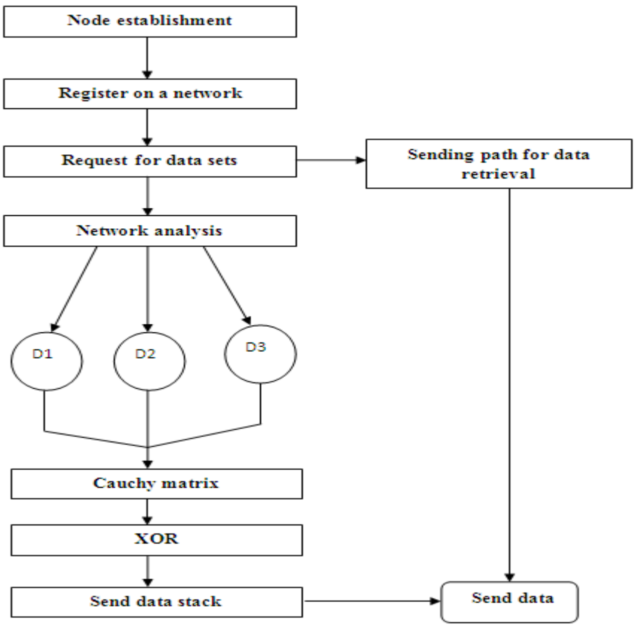

Fig. 5: Data Flow Diagram

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. 4, Issue 6, June 2016

Data is requested from the cloud and the same is retrieved via a sending path, as the module is designed to achieve higher performance gain and thus decrease the delay time in responses. The network analysis unit, consisting of server and pre-router analyzes the request and listen for responses under various data nodes. Cauchy matrix is appended in the intermediate state to reduce the complexity of the system under an overall protocol design.

IV. CONCLUSIONS AND FUTURE WORK

Cloud storage systems always use different redundancy configurations (i.e., ($k; m;w$)), depending on the desired balance between performance and fault tolerance. For different combinations of matrices and schedules, there is a large gap in the number of XOR operations, and no single combination performs best for all redundancy configurations. In this paper, we propose CaCo, a new approach that incorporates all existing matrix and schedule heuristics, and thus is able to identify an optimal coding scheme within the capability of the current state of the art for a given redundancy configuration. The selection process of CaCo has an acceptable complexity and can be accelerated by parallel computing. The experimental results demonstrate that CaCo outperforms the “Hadoop-EC” approach by 26.68-40.18% in encoding time and by 38.4-52.83% in decoding time simultaneously. Other code properties, like the amount of data required for recovery and degraded reads, may limit performance more than the CPU overhead. We look forward to addressing these challenges in the future.

In future, the Cauchy’s rule can be incubated with bandwidth of performance with respect to time for series computation of delay in network node retiring. The delay time reduction under narrow network is still a bottle neck situation. Either of this can be improvised with technical reduction of data sets and its indexing.

REFERENCES

[1] L. N. Bairavasundaram, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, G. R. Goodson, and B. Schroeder, “An analysis of data corruption in the storage stack,” Trans. Storage, vol. 4, pp. 8:1–8:28, Nov. 2008.
[2] J. L. Hafner, V. Deenadhayalan, W. Belluomini, and K. Rao, “Undetected disk errors in raid arrays,” IBM journal of research and development, vol. 52, pp. 413–425, July 2008.
[3] J. Bloemer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman,“An xor-based erasure-resilient coding scheme,” 1995.
[4] J. S. Plank, “A tutorial on reed-solomon coding for fault-tolerance in raid-like systems,” Software Practice Experience , vol. 27, pp. 995–1012, Sept.1997.
[5] J. S. Plank and Y. Ding, “Note: Correction to the 1997 tutorial on reed& ndash;solomon coding,” Software Practice Experiance, vol. 35, pp. 189–194,Feb. 2005.
[6] X. Li, Q. Zheng, H. Qian, D. Zheng, and J. Li, “Toward optimizing cauchy matrix for cauchy reed-solomon code,” IEEE Communication Letters., vol. 13,pp. 603–605, Aug. 2009.
[7] Guangyan Zhang, Guiyong Wu, Shupeng Wang, Jiwu Shu, Weimin Zheng, and Keqin Li “CaCo: An Efficient Cauchy Coding Approach for Cloud Storage Systems” IEEE Transactions On Computers, Vol. 62, No. 11, November 2015
[8] J. S. Plank, “XOR’s, lower bounds and MDS codes for storage,” Technical Representative. CS-11-672, University of Tennessee, May 2011.

BIOGRAPHY

Krishna urf Vibha V Dambal is a MTech post graduate student in computer science and engineering, UBDT college of engineering, Davanagere Karnataka, India. She received her Bachelor of Engineering in computer science from Tontadarya College of Engineering, Gadag, Karnataka, India in 2013.