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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

An Efficient Coding Scheme for an Optimal Solution for Data Coding in Cloud Storage System

Supriya Datkhile[1], Prof. P. R. Ugale.[2]

M.E. Student, Department of Computer Engineering, SPCOE, Otur, India[1]

Assistant Professor, Department of Computer Engineering, SPCOE, Otur, India[2]

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 Sharing; Access Control; Privacy-Preserving.

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, pet scale file and archival storage, and long term data preservation. 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. 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.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

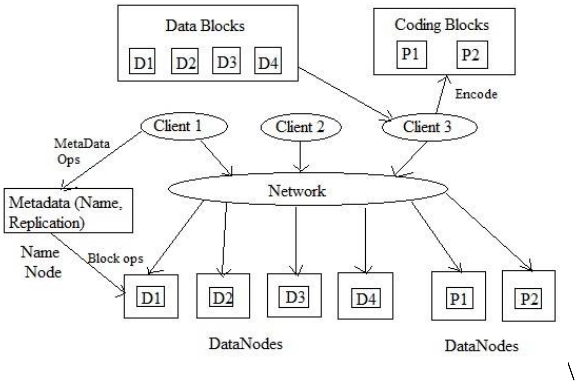

Fig. 1. Distributed Architecture for Cloud Storage System

II. OBSERVATION AND MOTIVATION

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 (2w k+m) (k+mk) Cauchy matrices, there is no method discovered to determinewhich 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.

III. RELATED WORK

In existing system cloud storage is built up of numerous inexpensive and unreliable components, which leads to a decrease in the overall MTBF (mean time between failures). As storage systems grow in scale and are deployed over wider networks, component failures have been more common, and requirements for fault tolerance have been further increased. 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. For example, Google's cloud storage, Windows Azure Storage, Ocean Store, Disk Reduce, HAIL, and others all tolerate at least three failures. To tolerate more failures than RAID, many storage systems employ Reed-Solomon codes for fault-tolerance.

Cauchy Reed-Solomon Coding Reed-Solomon (RS) codes 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 codes modify RS codes and give two improvements. First, CRS codes use a Cauchy matrix instead of a Vandermonde matrix. 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 ≤ 2w, let X = {x1; : : : ; xm}, Y = {y1; : : : ; yk}, and X ∩ Y = _, so that each xi and yjis 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). Since the elements of GF(2w) are the integers from zero to 2w

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

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

− 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×w) binary matrix, M(e), whose i-th(i = 1; : : : ;w) column is equal to the column vector V (e2i−1). Thus according to the value of w, we can transform the Cauchy matrix into a (mw × 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.

IV. PROBLEM STATEMENT

Given a redundancy configuration (k, m, w) our goal is to find a Cauchy matrix, whose schedule is desired to be the shortest. In this paper, we propose CaCo, a coding approach that incorporates all existing matrix and schedule heuristics, and therefore is able to discover an optimal solution for data coding in a cloud storage system, within the capability of the current state of the art.

V. PROPOSED SYSTEM

To provide user security for file transfer we requires proposed system. As in many roll based access system if the user have the access of the file then user can access the file any time but if the user found unauthorized then there is main challenge is revoking the access of that user. KIDS provide that facility of revoking the access of the user also signature concept for the particular file. The figure shows the Architecture of Proposed System. The system consists of four basic modules which are listed and explain below in detail.

5.1 Generating Cauchy matrices:

Selecting the best one from Cauchy matrices using the enumeration method is a combinatorial problem. Given a redundancy configuration (10; 6; 8), the magnitude of the matrices to be constructed can be up to 1029, and it is unrealistic to enumerate them. We cannot even determine which one of the matrices will produce better schedules. In the CaCo approach, we choose only a certain number of them for scheduling.

5.2 Constructing schedules for each matrix:

For each matrix mi(0 < i < p) in the set Sm, we pass the parameters including k, m, w and pointer of the matrix to the function do schedule (int k, int m, int w, int * matrix) to perform q heuristics in the function, such as Uber CSHR, X Sets, and so on. In this manner, we get a set of schedules, denoted as Ssi={s0i, s1i, . . . ,sq,1i}. If there appears a good heuristic for scheduling at a later date, we can add it to the function do schedule.

5.3 Selecting the locally optimal schedule for each matrix:

For each matrix mi(0 < i < p) in the set Sm, we select the shortest schedule from the set Ss;i, denoted as si, so that we get a set of matrices and their shortest schedules, denoted as S = {(m0, s0); (m1; s1); . . . ; (mp_1; sp_1)}. For mi in the set Sm, we can encode data in an order of XORs given by si. In this way, the times of XOR operations no longer have direct relationship with the density of the matrix. Therefore, scheduling excludes the influence of the lower limit of the number of ones in the matrix, so the performance improves significantly.

5.4 Selecting the globally optimal solution:

From the collection of combinations of Cauchy matrix and schedule, namely {(m0; s0); (m1; s1); . . . ; (mp_1; sp_1)}, we choose the combinations with the shortest schedule. On this basis, for better performance, we tend to select the one containing the fewest ones in the matrix to be (mbest; sbest). Once selected, subset can be used for encoding data.

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

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

5.5 Data Auditing:

Security monitoring on the cloud is important, because computers sharing data are most readily available to an attacker. Without mechanisms in place to detect attacks as they occur, an system may not realize its security. Therefore it is vitally important that computers residing in the cloud are carefully monitored for a wide range of audit events. The auditing in a system consists of three steps. The first step is the attack has attempted on any node in system, secondly the attack is detected by the system by hashing algorithm after detection of attack the notifications are send to data owner. Due to this security is improved.

VI. MATHEMATICAL MODEL

Our problem statement comes under the polynomial class according to definition of polynomial class; the problem is solved in P-time. So above two deterministic algorithms called P-class algorithms.

Let Sbe the set of proposed system.
Set: S={I, R, P, O}
Where,
I= Set of Inputs for our system.
R= Set of Rules that are applied while processes are performed
P= Set of Processes
O= Set of Outputs

Input Data:
I={I1, I2, I3, I4 }
Where,
I1: Upload Data
Set Of Rules:
R={R1, R2, R3}
Where,
R1= Decide the Datanodes
R2= Write Data
Set Of Processes:
P={P1, P2, P3 }
Where,
P1= Validation of required details
P2=Write Operation
Output Data:
O={O1, O2, O3, O4}
Where,
O1: Write data on DataNodes
Input={Upload data}
Output={Write data on DataNodes}
P={Fx| Iput Output}

Fx is the function which take input as upload data on cloud and give output as write data on data nodes.

Success Condition = Data written successfully.
Failure Condition = Data Nodes fail.

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

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

VII. HARDWARE/SOFTWARE REQUIREMENT

A. Hardware Requirements:

- System : Intel Core i3 2.8 GHz.
- Hard Disk : 250 GB.
- Monitor : 15" VGA Colour.
- Mouse : Logitech.
- Ram : 1 GB.

B. Software Requirements:

- Operating System : Windows 7.
- Coding Language : Java Spring MVC
- Data Base : HBase

VII. EXPERIMENTAL RESULT

In general, Table I shows an expected result of proposed system. We compare Hadoop, CaCo and Proposed result in the encoding time when recovering from failures of two data disks and one coding disk. First, the coding times of Hadoop, CaCo and Proposed result take on an upward trend as k increases along the x axis.

Table 1: Execution Time Of A Redundancy Configurations(K,M,W)

	Hadoop	CaCo	Proposed CaCo
4, 3, 4	0.03	0.01	0.01
5, 3, 4	0.04	0.022	0.02
6, 3, 4	0.04	0.033	0.025
7, 3, 4	0.05	0.036	0.025

IX. CONCLUSION

We design a secure anti-collusion data sharing scheme for dynamic groups in the cloud. In our scheme, the users can securely obtain their private keys from group manager Certificate Authorities and secure communication channels. Also, our scheme is able to support dynamic groups efficiently, when a new user joins in the group or a user is revoked from the group, the private keys of the other users do not need to be recomputed and updated. Moreover, our scheme can achieve secure user revocation; the revoked users can not be able to get the original data files once they are revoked even if they conspire with the untrusted cloud.

X. ACKNOWLEDGEMENT

With all respect and gratefulness, I would like to thanks all people who have helped me directly or indirectly for the paper presentation. I am grateful to my guide, Prof. , for his guidance and support. I wish to express my sincere thanks to the, PG Coordinator Prof. and HOD Prof for their support. Lastly I would like to thank staff member of Department of Computer, SPCOE, Otur, Pune, India for making all the requirements possible and simply available whenever required.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

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 J. Res. Dev*., vol. 52, pp. 413–425, July 2008.
[3] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," *in Presented as part of the 9th USENIX Symposium on Operating Systems Design and Implementation, (Berkeley, CA), USENIX*, 2010.
[4]. I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math*., vol.8, no. 2, pp. 300-304, 1960.
[5]. K. Manivannan, and L. Rigas, "Windows azure storage:A highly available cloud storage service with strong consistency," *in Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, SOSP '11, (New York, NY, USA)*, pp. 143–157, ACM, 2011.
[6]. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Ocean store: An architecture for global-scale persistent storage," *SIGPLAN Not*., vol. 35, pp. 190–201, Nov. 2000.
[7]. 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.
[8]. 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
[9]. J. S. Plank, "XOR's, lower bounds and MDS codes for storage," *Technical Representative. CS-11-672, University of Tennessee, May* 2011.
[10]. Bloemer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman,"An xor-based erasure-resilient coding scheme," 1995.
[11] 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.
[12]. K. D. Bowers, A. Juels, and A. Oprea, "Hail: A high-availability and integrity layer for cloud storage," *in Proc. 16th ACM Conf. Comput. Commun. Security, New York, NY, USA*, 2009, pp. 187-198.
[13]. G. Xu, F. Wang, H. Zhang, and J. Li, "Redundant data composition of peers in p2p streaming systems using Cauchy Reed-Solomon codes," *in Proc. 6th Int. Conf. Fuzzy Syst. Knowl. Discovery- Volume 2, Piscataway, NJ, USA*, 2009, pp. 499-50.
[14]. J. S. Plank, "A tutorial on Reed-Solomon coding for faulttolerance in raid-like systems," *Softw. Pract Experience*, vol. 27, pp. 995-1012, Sept. 1997

BIOGRAPHY

Supriya Datkhile is a Student in the Computer Engineering Department, SharadChandra Pawar College of Engineering, Savitribai Phule Pune University. Her research interests are Cloud Computing, Communication, Security etc.