Providing De-Duplication And Fault Tolerance On Cloud Storage

[1]S.Maniyarasi ,[2]V.Snehalatha.M.E.,
[1]Pursuing M.E, Dept of CSE
[2]Assistant Professor(SG), Department of Computer Science and Engineering

Abstract:Cloud storage is a model of data storage in which the digital data is stored. These cloud storage providers are responsible for keeping the data available and accessible. The aim of our project to achieve fault tolerance and avoid redundant data storage in the cloud.Our proposed technique is Cauchy Coding (Caco)approaching. With this technique Cauchy matrix heuristics are used to produce a matrix set. Then for each matrix in this set, CaCo uses XOR schedule heuristics to generate a series of schedules, and then selects the shortest one from them. In this way, each matrix is attached with a locally optimal schedule. Finally, CaCo selects the globally optimal schedule from all the locally optimal schedules. This globally optimal schedule and its corresponding matrix will be stored and then used during data encoding,decoding and de-duplication process.CaCo has the ability to identify an optimal coding scheme, within the capability of the current state of the art, for an arbitrary given redundancy configuration.Our investigational results designate that CaCo can obtain an optimal coding scheme.

Keywords— Cloud storage, fault tolerance, De-Duplication,Cauchy Reed-Solomon codes, Cauchy matrix, XOR scheduling

I. INTRODUCTION

Cloud storage is a cloud computing model in which data is stored on remote servers accessed from the Internet, or "cloud." It is maintained, operated and managed by a cloud storage service provider on a storage servers that are built on virtualization techniques.Storage maintenance tasks, such as purchasing additional storage capacity, are offloaded to the responsibility of a service provider. Cloud storage can be used as natural disaster proof backup, as normally there are 2 or 3 different backup servers located in different places around the globe. Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources computer networks, servers, storage, applications and services, which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers that may be located far from the user–ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over an electricity network. It is built up of numerous inexpensive and unreliable components, which leads to a decrease in the overall mean time between failures (MTBF). As storagesystems 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 , OceanStore , DiskReduce , HAIL , and others all tolerate at least three failures.To tolerate more failures than RAID, many storage systems employ Reed-Solomon (RS) codes for fault-tolerance.

Cauchy Reed-Solomon (CRS) codes improve Reed-Solo-mon 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 singlewords. 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.

Data deduplication (often called "intelligent compression" or "single-instance storage") is a method of reducing storage needs by eliminating redundant data. The technique data de-duplication is used to store single instance of redundant data and eliminates the duplicate data in datacenter. It is used to decrease the size of datacenter and reduce the replications of data that were duplicated on cloud. The de-duplication process helps to remove any block or file that are not unique and store in smaller group of blocks.

File-level deduplication In this deduplication method, the duplicate files are identified if they have same hash value and is performed over single file. It requires less processing power since files hash numbers are relatively easy to generate. If any change is made in a file it makes to save the whole file again in file level deduplication. In file level deduplication indexes are small, and so it takes less time for computational when it identifies the duplicate copies.

Block-level deduplication Block level deduplication is performed over blocks. It first divides files into blocks and stores only a single copy of each block. It could either use fixed-sized blocks or variable-sized chunks. Block level deduplication can eliminate or delete the small redundant chunk of data when compared to whole file. Each and every file system can use same deduplication algorithm in block level deduplication.

Byte-level deduplication Byte-level deduplication is a form of block-level deduplication that understands the content, or "semantics", of the data. Byte-level deduplication understands the content of the data and the system can more efficiently deduplicate the bytes.

II. RELATED WORK

Reed-Solomon 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.

In a storage system using erasure codes, data are encoded to obtain data redundancy when data are written. Therefore, to improve the overall performance of a system, we should reduce the cost of erasure coding, i.e., the number of XOR operations. While using a binary Cauchy matrix for CRS coding, the number of XORs depends on the number of ones in the matrix. So, in order to get better performance, the number of ones in the binary Cauchy matrix should be as few as possible. To discover an optimal matrix from numerous Cauchy matrices, the simplest way is to enumerate them. The research on how to reduce the number of XOR operations in the process of erasure coding has revealed that the number of ones in a Cauchy matrix.Some heuristics such as Original , Optimizing Cauchy and Cauchy Good can generate a good matrix which contains fewer ones for larger w, but it may not be the optimal one.

It is still an open problem to derive a schedule from a Cauchy matrix that minimizes the number of XOR operations. Current algorithms for seeking schedule, such as CSHR, Uber-CSHR, and X-Sets, are heuristics, which could not guarantee to get the optimal solution.

III. PROBLEM STATEMENT

Existing system to tolerate more failures than RAID, "Redundant Arrays of Inexpensive Disks" (RAID) where batteries of small, inexpensive disks combine high storage capacity. Since then the technique has been used to design multicomputer and network file systems with high reliability and bandwidth and to design fast distributed checkpointing systems. General technique for tolerating simultaneous failures with exactly checksum devices.For that many storage systems employ Reed-Solomon (RS) codes for fault-tolerance.Reed-Solomon coding has been around for decades, and has a sound theoretical basis.

As an erasure code, Reed-Solomon code is widely used in the field of data storage. Given k data blocks and a positive integer m, Reed-Solomon codes can encode the content of data blocks into m coding blocks, so that the storage system is resilient to any m disk failures. Reed-Solomon codes operate on binary words of data, and each word is composed of w bits.In a storage system using erasure codes, data are encoded to obtain data redundancy when data are written.

Initially, people discover that the density of a Cauchy matrix dictates the number of XORs.For this reason, an amount of work has sought to design codes with low density. Moreover, some lower bounds have been derived on the density of MDS Cauchy matrices. In the current state of the art, the only way to discover lowest-density Cauchy matrices is to enumerate all the matrices and select the best one. When scheduling of XOR operations is introduced, the density of the matrix does not have a direct effect on the number of XORs. Addressed the issue of identifying and exploiting common sums in the XOR equations. They conjectured that deriving an optimal schedule based on common sums is NPComplete, and gave a heuristic based on Edmonds maximum matching algorithm. attacked this open problem, deriving two new heuristics called Uber-CHRS and X-Sets to schedule encoding and decoding bit-matrices with reduced XOR operations.

IV. SYSTEM MODEL

A.PROPOSED SYSTEM

Efficient Cauchy Coding approach used for cloud storage systems. CaCo uses Cauchy matrix heuristics to produce a matrix set. Then, for each matrix in this set, CaCo uses XOR schedule heuristics to generate a series of schedules, and selects the shortest one from them. In this way, each matrix is attached with a locally optimal schedule. Finally, CaCo selects the globally optimal schedule from all the locally optimal schedules. This globally optimal schedule and its corresponding matrix will be stored and then used during data encoding and decoding and avoiding redundant copies.

Incorporating all existing matrix and schedule heuristics, CaCo has the ability to identify an optimal coding scheme, within the capability of the current state of the art, for an arbitrary given redundancy configuration.

- Constructing the matrix ONES. First, CaCo constructs a matrix named ONES, whose element is defined as the number of ones contained in the binary matrix.
- Choosing the minimal element. Second, CaCo chooses the minimal element from the matrix ONES. Supposing the element is (x1, y1) we initialize X to be {x1} and Y to be {y1}
- Determining the set Y. CaCo chooses the top k minimums from row x1. Then, CaCo adds the corresponding k-1 column numbers to Y, and we have y = {y1, y2…yk)

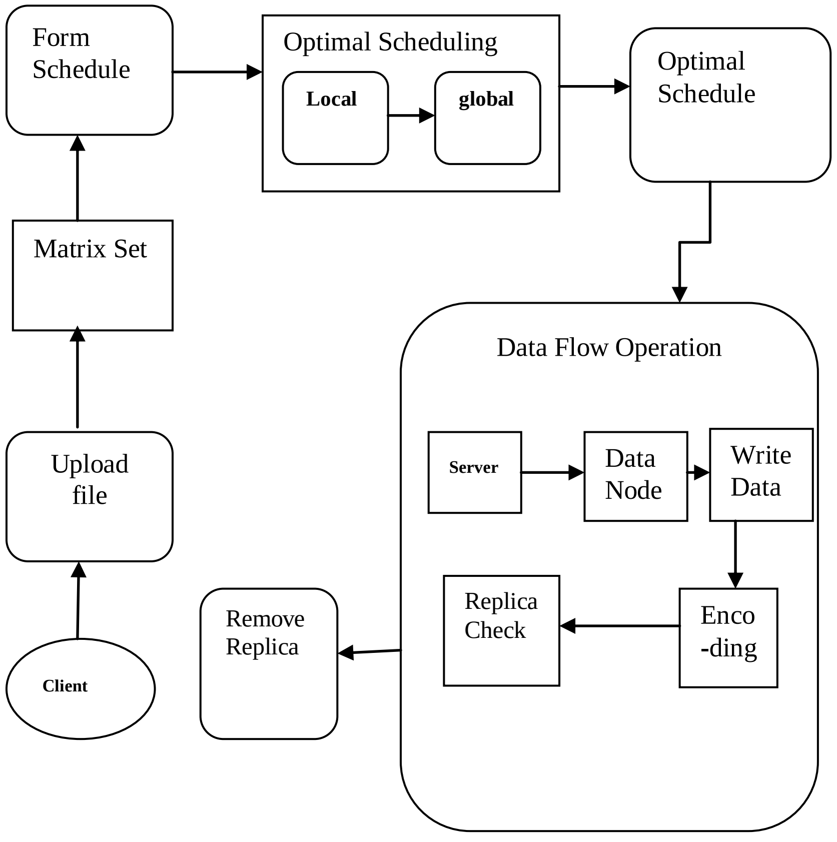

Fig 1:SYSTEM ARCHITECTURE

With erasure coding, we reduce the number of data copies from three to one. So, when a block is written, we only allocate one DataNode to the Client. After writing the data blocks to the allocated Data Nodes, we store a copy of data into DataQueue to be encoded. While there are k data blocks ready in Data-Queue, we encode them with the schedule selected by CaCo.

After encoding, check whether the file is already existed in Data Queue.if it is so, remove replica. Note that those data blocks and 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 blocks.In the read operation with CaCo, if there is not data corruption in the requested blocks, the Client directly retrieves the data.

When failures of some nodes occur, we need to decode the remaining valid blocks to rebuild the original data, using the optimal schedule generated by CaCo. Then, we supply the rebuilt data to the Client's read request.Finally,achieve fault tolerance and De-Duplication.

B. CODING AND TECHNIQUES

CAUCHY REED SOLOMON CODING

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.

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.The key to CRS codes is construction of Cauchy matrices.

Encoding data

- Uses of Cauchy matrix
- Using schedule

After finish encoding, can use intermediate results in the Data Queue to reduce duplication of data.

CONSTRUCTING CAUCHY MATRIX

CRS code treat each word as a number between 0 and 2^w-1.Construct W*W matrix.Each element in the matrix denotes the number of occurrence of each word. With given a redundancy configuration(k,m,w), we can construct Cauchy matrices, each of which can be used for encoding.

SELECTING A MATCHED CODING SCHEME

The steps are

- Constructing schedules for each matrix
- Selecting the locally optimal schedule
- Selecting the globally optimal solution

After selecting optimal encoding scheme, have to encode the data. If any m data blocks or coding blocks fail, we can recover the original data by decoding the remaining valid blocks. First, according to the chosen matrix m for data encoding and the specific situation of data corruption in the same group of blocks, we can generate a matrix m for decoding. Then we discover the optimal schedule for m using the same approach.

CHECK REDUNDANT

With this, chek files in the data Queue to find any duplicated copy. While performing data encoding with a given Cauchy matrix, we can use intermediate results to reduce duplication if it is so remove redundant; if not, store the file to the data node.

WRITE OPERATION

Step 1: The Client sends a write request to the Name Node

Step 2:The Name Node allocates some Data Nodes to the Client

Step 3: Write the data blocks into Data Nodes

Step 4: Make a copy of data and put it into Data Queue

Step 5: Encode data with the schedule selected by CaCo

Step 6: Write the coding blocks into Data Nodes

Step 7: Data encoding finishes

Step 8: Remove the copies of data from Data Queue

V. CONCLUSION 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 to achieved fault tolerance and remove redundant file in cloud. 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. It should also be noticed that the selection process is once for all.

Finally, we readily acknowledge that reducing XORs is not the only way to improve the performance of an erasure code.

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.

Other code properties, like the amount of data required for recovery and degraded read, may limit performance more than the CPU overhead. We look forward to addressing these challenges in the future.

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. Develop., vol. 52, pp. 413–425, Jul. 2008.

[3] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally dis-tributed storage systems," in Proc. 9th USENIX Symp. Oper. Syst. Des. Implementation, 2010, pp. 61–74.

[4] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci, J. Haridas, C. Uddaraju, H. Khatri, A. Edwards, V. Bedekar, S. Mainali, R. Abbasi, A. Agarwal, M. F. u. Haq, M. I. u. Haq, "Windows Azure storage: A highly available cloud storage service with strong consistency," inProc. 23rd ACM Symp. Oper. Syst. Principles, New York, NY, USA, 2011, pp. 143–157.

[5] 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, "Oceanstore: An architecture for global-scale persistent storage," SIGPLAN Notices, vol. 35, pp. 190–201, Nov. 2000.

[6] B. Fan, W. Tantisiriroj, L. Xiao, and G. Gibson, "Diskreduce: Raid for data-intensive scalable computing," in Proc. 4th Annu. Work-shop Petascale Data Storage, New York, NY, USA, 2009, pp. 6–10.

[7] K. D. Bowers, A. Juels, and A. Oprea, "Hail: A high-availabilityand integrity layer for cloud storage," in Proc. 16th ACM Conf. Comput. Commun. Security, New York, NY, USA, 2009,pp 187–198.

[8] G. Xu, F. Wang, H. Zhang, and J. Li, "Redundant data composi-tion of peers in p2p streaming systems using Cauchy Reed-Solomon codes," in Proc. 6th Int. Conf. Fuzzy Syst. Knowl. Dis-covery-Volume 2, Piscataway, NJ, USA, 2009, pp. 499–503.

[9] J. S. Plank, "A tutorial on Reed-Solomon coding for fault-tolerance in raid-like systems," Softw. Pract. Experience, vol. 27,pp 995–1012, Sept. 2008.

[10] 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," in Proc. 7th Conf. File Storage Technol., Berkeley, CA, USA, 2009, pp. 253–265.