ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

In Detail Comparative Study of Application of Erasure Coding on Various Cloud Storage

Ram Prakash Kota[1], Dr. Rajasekhara Rao Kurra[2]

Research Scholar, Department of CSE, ANU, India[1]

Director, Sri Prakash College of Engineering (SPCE), Tuni, India[2]

Abstract: With the increasing demand of cloud computing and hence forth the demand of cloud storage for multiple application, it is a high priority research demand for secure and efficient replication of the data. Moreover the failure is an added component of risk to cloud storage. Though replicating data over cloud storage service providers is a common task considering the low cost data recovery. However over replication of data may lead to integrity problem with un-effective cost factor. Multiple work share been addressing the same issue over a period of time to find the most effective replication algorithm. But with a specific focus on domain dependent data and service providers. Hence in this work we propose a comparative study of Erasure algorithm on various cloud service providers. This work also demonstrates a theoretical framework for cost effective storage replication and discussion the performance.

Keywords: Erasure, RAID, RAID 4, RAID 5, Array Code, Reed – Solomon Code, Azure, Amazon S3, Performance Matrix

I. INTRODUCTION

The tremendous growth in cloud storage services and the fact that is has reached to a point where loss of data due to failure is expected. The real challenge is thrown to the designer of the storage solutions for cloud services to protect the data loss during failure. The core technology behind protecting data during loss is Erasure coding. Previous works demonstrates the use of Erasure coding for the last two decades. However the true understanding of Erasure and effective use of Erasure Codings never been discussed based on different cloud service provider. Thus this leads to confusion in solution designer and developer community. Hence in this work we focus on fundamental understanding of Erasure Coding, Comparisons and analysis of Erasure performances on multiple cloud storage service providers.
The storage systems on cloud came a long way in terms of capacity and latency time improvement. All the storage hardware types are commonly failing to protect data during failures and unable to restrict data loss. The type of failure can be not having control on getting disk sectors corrupted or the entire disk is becoming unusable. The storage services have some self-protecting mechanism as extra-corrective information that can detect changing of few bits from the original data and can still retrieve the originally stored data. However there are situations when multiple bits change unexpectedly, then the self-protecting mechanism detects that as hardware failure and storage devices become un-usable. This situations lead to loss of data. To handle these types of anomalies, the storage systems depend on Erasure codes. The Erasure code deploys the mechanism of assured redundancy to overcome the failures. The most generalized way of implementing this mechanism is replication of data over multiple locations. The most popular and simplest is Redundant Array of Independent Disks or RAID. In that the most basic version of these implementations is RAID – 1, where every data byte is stored in at least two parallel disks. This way the failure may not lead to loss of data as long as a replicated copy of the data is available. This mechanism is easy to achieve, however this leads to many other overhead factors like cost of storage. The storage cost should be at least double than the actual cost. Moreover in any case if both the storage device fails then the complete solution becomes unusable. In the other hand, there are more complex solutions under Erasure methodologies such as well-known Reed-Solomon codes. Reed-Solomon code can overcome high level failures with little less extra storage. These codes provide high level of failure tolerance with reduced cost. In communication systems the Erasure coding is similar to Error Correcting Codes or ECC. Here the Erasure coding solves the similar types of problems but addresses very different types of problems. In message communication, the error is caused by changing bits of the data. Here is the different lie between Erasure and message communication as the location of the changing bits is unknown. Hence application of Erasure is restricted.

In this work we demonstrate the reasons of failure of storage systems in Section II, Compare the performance of fault tolerance mechanisms in Section III, we discuss details of Erasure Code in Section IV, Compare the performance of Erasure Codes in various cloud storage service providers in Section V, Discuss the performance measure factors in Section VI, Describe the theoretical cost effective framework in Section VII and list the conclusions and future scope in Section VIII.

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

II. STORAGE FAILURE

The failure is the main reasons of data loss in any cloud storage solutions. Hence to reduce the downtime, the key factor is to reduce the failure occurrences. During any disaster situation, there is no process to physically investigate the reasons by visiting the data centre and examining the storage containers. Hence it is recommended to understand the reasons for cloud storage failures and prepare according to that. Here we analyse the reasons for storage solutions failure on cloud:

A. Failure Caused by Human Fault

Major of the mission critical applications handling mission critical data can lead to failure and data loss due to miss handling by service engineers. The chance of no data loss remains on the effective man power the service provider deploys to the system. The major reasons for failure can occur by miss leading service document, service procedure or ignoring updates. It is a standard practice to write scripts to manage maintenance in all service providers. This becomes crucial, when a wrong execution of a service script leads to application and data failure. The following is an example of script, which may mislead the operation and can cause the data loss during backup operation:

```
EXECUTEdbo.DatabaseBackup@Databases= USER_DATABASES',
@URL = 'https://account.blob.core.server.net/storage',
@Credential = 'Credential_Secure',
@BackupType = 'FULL',
@Compress = 'Y',
@Encrypt = 'Y',
@EncryptionAlgorithm = 'AES_255',
@ServerCertificate = 'MyCertificate',
@Verify = 'Y'
```

Here the encryption algorithm is specified as AES_255 but the actual know algorithm to the system is AES_256. Hence after the execution of this script, the backed up data will be inaccessible.

B. Failure Caused by Bugs in Application

The applications running on cloud infrastructure are the fore front handler of the storage solutions and data. The applications are designed to manipulate data during business transactions. Hence forth the application is responsible for accepting and analysing mission critical business data during customer interactions. We understand the applications deployed by the users are well tested and highly maintained. The application may be used by company personals or directly by the customer. Hence this may lead to a situation where mishandling of that application raises reasons for failure, which cannot be handled by the application. Hence this situation is considered as bugs in the application during data validation. The example in this case demonstrates [Fig – 1] the validation bug while accepting the list price of a product. This may lead to integrity loss and failure.

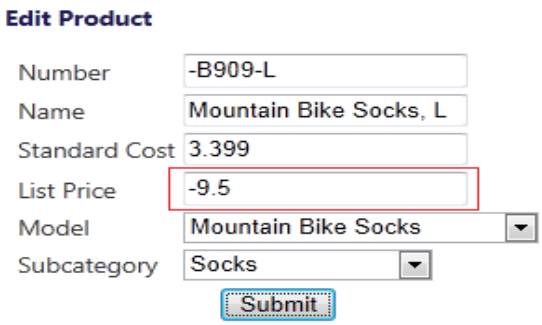

Figure 1: ERP Application for Stock Update

C. Failure Caused by Service Provider Fault

Major of the storage and cloud service providers perform their maintenance tasks. The maintenance tasks involve changing instances, storage containers, routine performance maintenance and terminating unused machines. Most of the events are well pre-notified and customers usually shutdown their applications to prevent data loss. However, sometimes a huge downtime for the applications also leads to loss of data. We analyse Amazon Cloud Services to understand the downtime and effect of applications [Table – 1].

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

TABLE I: AMAZON SERVICE AVAILABILITY

Year	Downtime per Year in Hours	Availability %
2015	5.26	99.999
2014	52.56	99.990
2013	262.8	99.950
2012	8.76	99.900
2011	87.6	99
2010	876	90

D. Failure Caused by Quality Failure

The highest priority for business customer demand is storage and streaming of video data. This depends on network latency, shifting customer requirements based on number users per node and finally the fluctuating demands. Major of the cloud service providers struggle to provide the required storage solutions, which are fast and cost effective to the customers. Hence customer applications may lead to loss of data during the process of video data streaming.

E. Failure Caused by Increasing Demands

The demand for users per node for customer is ever increasing. Hence the demand for high availability hardware resources including storage is also increasing. When the cloud service provider maintains on premises hardware resources, then there is a very less change to increase the resources immediately matching the resource demand. This may lead to loss of data during pick hours of business for customers.

F. Failure Caused by Security Reasons

Hosting applications for business needs handling mission critical data might be economical on public cloud solutions but that opens the channel for security attacks. In major free solutions most of the time the security is compromised due to lack of good free source security solutions. Hence this may lead to loss of data by attacks like hacking or DDoS.

G. Failure Caused by Service Failure

Majority of the applications dealing with enterprise data which are critical in nature use third party applications to fetch data from other sources. This policy is used majorly in business process management applications on cloud. The fetched data is to be collected from multiple other services, where one part of the connection is residing on the customer application and other two parts resides outside the application. One of those two components resides on the service application from which the data needs to be collected and other component resides on the third party server to connect and match both the services. Failure of any of these services may lead to customer application failure which intern leads to loss of data.

H. Failure Caused by Container Failure

In the recent researches it is been proved that the failures of storage containers is the highest listed factor leading to data loss. The specific reasons of the storage containers failure is discussed earlier in this work.

I. Failure Caused by Bad recovery policies

As the data loss is caused by many factors listed above, hence a good recovery policy should be the highest priority for the customer applications and storage service providers. Due to lack of data management knowledge, the recovery policy is most of the time ignored in major places. Inclusion of a better recovery policy also may lead to higher cost factors. Simplest recovery policies may at least double the cost of storage and maintenance. Moreover the customer believes that the recovery services are the responsibility of the provider and storage service providers believes that the application developers must take care of the contingency plan. This leads to confusion and finally the loss of data.

As we see there are many factors influencing data loss related to application, hardware, security, policies, demands and quality of storage solutions, hence there shows requirements for understandings the fault tolerance mechanism and their performances.

III.STANDARD FAULT TOLERANCE MECHANISMS

The standard fault tolerance mechanism depends on the erasure codes. The basic mechanism can be understood if we assume a collection of n disks are partitioned into k disks. Hence there will be m disks which will hold the coding information as

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

$$m = n - \sum_{i=1}^{r<n} k_i \quad \text{....Eq 1}$$

Where r denotes number of k multiple of disks

The basic interpretation of the erasure codes can be understood as each disk must hold a z bit word to represent the customer data. If we denote them with d then the total set of codes for k number of disks are considered as

$$z_1, z_2, z_3 .... z_k \quad \text{....Eq 2}$$

Also we consider the codes stored on each every m disk with c, and then the total representation is considered as

$$c_1, c_2, c_3 .... c_k \quad \text{....Eq 3}$$

The coding and the customer data should a linear combination and can be represented as

$$\begin{aligned} c_0 &= a(_{1,0}) z_0 + ... + a(_{1,k-1}) z_{k-1} \\ c_1 &= a(_{2,0}) z_0 + ... + a(_{2,k-1}) z_{k-1} \\ &.... \\ &.... \\ c_m &= a(_{m,0}) z_0 + ... + a(_{m,k-1}) z_{k-1} \end{aligned} \quad \text{....Eq 4}$$

The coefficients “a” are also z bit words. Encoding, therefore,

Simply requires multiplying and adding words, and decodinginvolves solving a set of linear equations with Gaussian eliminationor matrix inversion.

Furthermore, we understand the most popular coding techniques here.

A. RAID-4 and RAID-5

The RAID – 4 and RAID – 5 are the simplest form of the erasure codes explained in this work earlier. RAID – 4 and RAID –5 differs from the basic framework as it employs different arrangements of data replication. The framework for RAID – 4 and RAID – 5 are explained here:

The RAID is a modification to MDS code where m=1 and z=1. The basic coding depends on a bit noted as p, where

$$p = z_0 \oplus z_1 \oplus .. \oplus z_{k-1} \quad \text{....Eq 5}$$

In case of any bit changing, the XOR code will identify it for the surviving code.

B. Linux RAID-6

The Linux system RAID – 6 is considered as additional support to RAID – 4 and RAID – 5 as it uses an alternative disk under the framework. This framework proposes an alternation to the MDS as considering the code to be stored in two disks as m=2. Hence the formulation is too simple by using an XOR code:

$$\begin{aligned} p &= z_1 \oplus z_2 \oplus ... \oplus z_k \\ q &= z_1 \oplus 2(z_2) \oplus ... \oplus 2^k (z_k) \end{aligned} \quad \text{....Eq 6}$$

Here the codes called p and q will be stored on alternative disks to ensure the Erasure code to protect the data loss.

C. Array Codes

The framework is called Array code as it is implemented using r X n array of customer data. In this framework the customer data will be stored with the arrangements as Figure – 2.

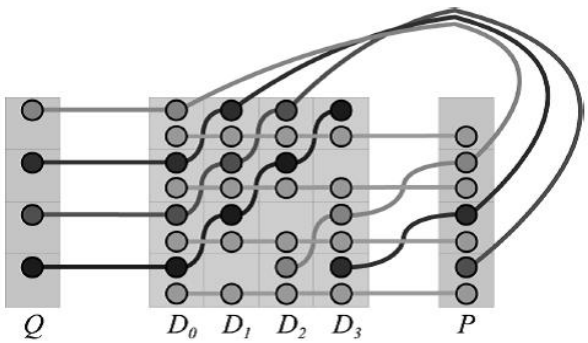

Figure 2: Array Code Storage

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

The array code with the following parameters: k=4, m=2 (RAID-6), n = k+m = 6, r=4, z=1.

D. Non-MDS Codes

The Non-MDS codes do not allow replication of m storage devices to achieve optimal fault tolerance. The replication of storage devices containing the code is higher than the other frameworks. However the efficiency provided by the Non-MDS codes compared to other frameworks in terms of performance is high.

Hence we compare all the types of code frameworks here.

IV. UNDERSTANDING REED-SOLOMON ERASURE

The most effective and popular framework under Erasure Coding is Reed-Solomon framework. The framework can be applied in case of

$n \leq 2^z$, where n denotes number of disks and z denotes number of customer data ….Eq 7

To understand the framework for 256 storage containers or disks are considered. For a 256 disks, a Reed – Solomon code can be defined and implemented using Galois Field Arithmeticor GF (2^8). The coefficient "a" can be defined in various ways. The basic implementation of Reed – Solomon is Couchy construction. To understand Couchy construction, we select any n unique numbers in the space of GF(2^Z). Hence the selected n number are distributed in two sets called X and Y, where X contains m elements and Y contains k elements. Hence:

$$a(_{i,j}) = \frac{1}{x_i \oplus y_j}$$ with the help of GF (2^z….Eq 8

The most important factor that makes Reed-Solomon framework to implement is the simplicity. In this framework selecting k and m is random and does not depend on any factors and can be selected independently. The performance can be questioned as the time complexity for performing an XOR operating is less compared to GF. However the modern processors rely on vector instruction sets for performing array based multiplication operation. Hence the reduction in time for computation can be achieved. Moreover with the improvement of latency time for the I/O devices and cache memory is also been improving to match with the highly complex Erasure Codes. The implementation of Reed – Solomon is simple as many open source solutions are readily available for storage solutions.

V. APPLICATION OF ERASURE

As the most noted fault tolerance framework is the Erasure codes, hence we understand the application of Erasure codes on various cloud storage service providers.

E. Erasure on Microsoft Windows Azure

Microsoft Windows Azure employs a Local Reconstruction Code or LRC to be implemented using Reed – Solomon Code. The LRC is shorter code, which is robust and portable to implement and store. Here we understand the application framework in detail:

We assume there are 6 data segments and 3 parity segments. Here the 3 parity segments are computed from 6 data segments stored in distinguished 9 disks. During failure any segment can be used for reconstruction. As the data and code is distributed over 9 segments, hence all the 9 segments need to be used for reconstruction. Azure define the cost of reconstruction is equal to number of data segments required for reconstruction. Hence in this case the total reconstruction cost is 6. However the main purpose of LRC is to reduce the reconstruction cost by calculating some of the codes from the local data segments. Hence to follow the same logic we have now 4 parity codes. Two of the parity codes are generated from all the data segments and should be kept globally. In the other hand the remaining two parity codes are computed from each storage data segment groups and should be kept locally [Figure – 3].

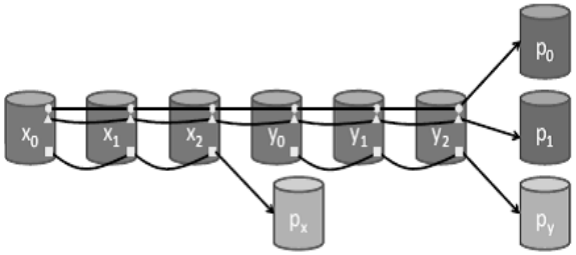

Figure 3: LRC Computation

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

Here the construction of LRC adds an additional parity code into the Reed – Solomon code. Hence it may appear as addition load on the computation, however this computation does not execute during the conventional tractions of data.

F. Erasure on Amazon S3

The basic implantation of fault tolerance of Amazon Simple Storage Service or S3 depends on the RAID framework. However rather than depending only on the storage providers, Amazon also recommends to employ application based fault tolerance mechanism. Hence this frame work should be considered as RAID – Application based framework. This is very much similar to Service Oriented Architecture or SOA model for RAID.

The fault tolerance mechanism for Amazon S3 has three major components in the framework [Figure – 4]:

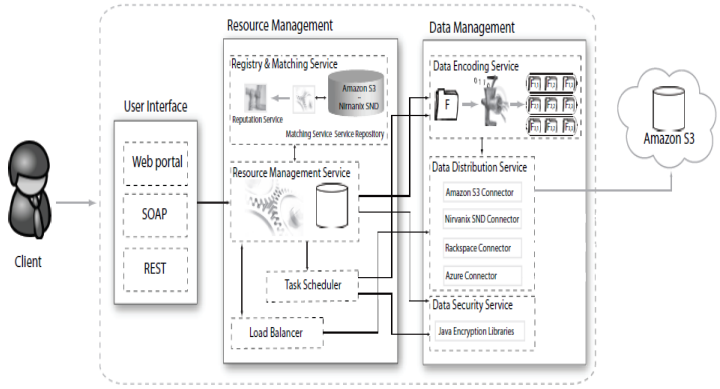

Figure 4: RAID SOA

- Module for Resource Management: The Module for resource management is responsible for data deployment considering the factors of customer location preferences, content type for storage and application performance.
- Module for Data Management: This component is responsible for handling data based on factors like encoding of data, distributions of data and security factors.
- UI Module: The UI module plays a bit of less important role in this architecture. This UI module provides the overall view of the business data for the customers.

G. Erasure on Google File Systems

The File System in Google employs an essential high load data processing and storage solutions on public storage systems. The most crucial recovery factor relies on the Google's specific algorithms using constant monitoring, replication management, automatic and chunk recovery.

Hence we understand that most of the cloud service providers use Erasure codes for their storage solutions with modifications leading to service and cost benefits.

VI. PERFORMANCE MEASURE FACTORS

The core understanding to be realized here is that only considering the storage system efficiencies and recovery speed during failure is not sufficient. To have a better realization of the performance of Erasure codes, there is a need for formulating the factors for performance measure.

Here we list the other important factors for performance measure to help selecting the better Erasure coding framework:

H. Repair Bandwidth Factor

To recover from a failure, the new data is to be stored on a new disk connected locally or remotely on the storage solution server. The process of recovery deploys a process to transmit the code to the new disk from the existing disk. Hence the new disks can locally generate the code. During the transmission of code for re-generation of data uses the same network which is used by the application for data transactions. This may lead to over load on the network and can slow down the application response time.

I. Repair Input-Output Factor

During the repair operations, the write and read transactions needs to be carried out on new and existing disks to write recovered data and codes respectively. The write transactions will write the re-generated data and the read transactions will read the codes from the existing disks. To write one block of data the code framework needs to read from all the

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ICRAESIT - 2015
B V Raju Institute of Technology, Narsapur
Vol. 2, Special Issue 2, December 2015

code segment blocks as discussed in the earlier part of this work. Hence the write and read transactions are unavoidable. This may lead of over use of computational power provided by the service provider and may reduce the running application performance.

J. Latency Time Factor

Few of codes employ a strategy to embed the codes into the data. Hence requirements for additional storage spaces can be ignored. Moreover due to less replication the cost of storage solutions also can be reduced. However the problem arises when the process of re-generation of data starts. The re-generated data need to same as the original data and the original data included codes as embedded data segments. Hence a iterative and time complex process need to be deployed for recovery process. This may lead to high downtime for the application.

K. Efficiency of Storage Factor

The storage efficient factor defines a ratio of size of the original data and actual data size including the codes for recovery. The MDS code frameworks are considered to be efficient in terms of increasing the storage ratio factor. However to maintain a high storage ratio factor, it is nearly impossible to sustain a high disk I/O overhead. Thus the disk I/O overhead might be compromised.

VII. PROPOSED FRAMEWORK FOR DATA REPLICATION

In this work we understand the application to Erasure to various cloud storage service providers and we also understand the types of Erasure Codes. However we identify that the replication process is unavoidable and the generation of codes are the most important process during implantation of Erasure frameworks. Hence it is the most important factor to employ some mechanism to compress the replicated data during the replication process to minimize the storage cost.
Here we propose our algorithm to compress the customer data during the process of applying Erasure Codes.
To understand the process, we assume the size of the customer data is "s" and needs to be replicated over "n" disks. The disks containing the data can be noted as

$$d_0, d_1, d_2 \ldots, d_{n-1} \qquad \ldots\text{Eq 8}$$

Here the original data must be stored on d_0 .

Moreover the codes used for recovery also needs to be stored on "m" number of disks and can be noted as:

$$d_0, d_1, d_2 \ldots, d_{m-1} \qquad \ldots\text{ Eq 9}$$

The selection of "n" and "m" is realized using Reed – Solomon framework.
Hence the process of modified algorithm is presentedhere [Figure –5].

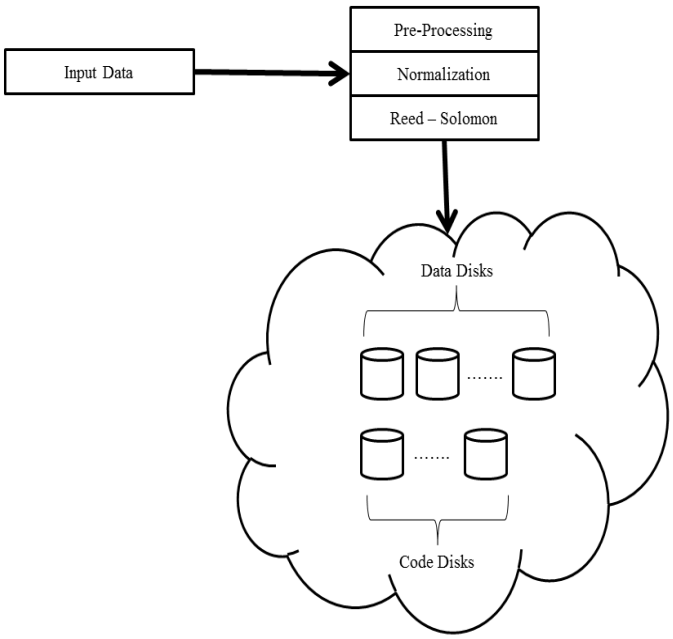

Figure 5: Proposed Algorithm

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

Step-1. **Pre-Processing of the Data**: In this phase the client data is accumulated on the storage solutions and then the pre-processing of the data is carried out. During the pre-processing phase, the data is been normalized to avoid any sort of noise and other factors that may influence the integrity problems with the failure also. Hence, a data with size "s" is normalized here, where each work of the data is noted as "w":

$$W = \sum \frac{\int_{i,j=0}^{i,j=n} w_{ij}}{\sum_{i=0}^{n} \frac{w_i}{\Delta w}} \quad \ldots \text{Eq } 10$$

Step-2. **Compression of Data**: In this phase the data is been compressed before the replication process starts. This is an iterative process and executes every time where there is a significant change in data size:

$$\lceil \log_2(S) \rceil + \lceil \log_2(W) \rceil + \lceil \log_2(A) \rceil \ldots \text{Eq } 11$$

Where,
S represents the size of the data buffer;
W represents the size of the entire window
represents the size of the alphabet

Step-3. Applying Reed – Solomon Code framework: Hence during the replication process and code generation system, the algorithm needs to address only the compressed form of data.

Hence forth the raise in time complexity to apply the BF approach is reduced here and the same can be used for data compression in the system.

We evaluate the theoretical model and realize that the reduction in storage cost is achievable through this, but the increase of computational complexity increases.
We have analysed multiple data loads and understood the compression factors [Table - 2].

TABLE III: COMPRESSION RESULTS

Data Set Name	Size During Raw Storage in MB	Size After Compression in MB
Set – 1	6.27	6.20
Set – 2	9.47	9.40
Set – 3	18.5	18.4

The compression algorithm results in 0.996 compression ratio.

VIII. CONCLUSION

Hence we understand the Erasure code framework and multiple variations to the same. We also consider their applications on major cloud storage service providers. We also consider the reasons that lead to failure of storage. We realize that the Erasure codes are very effective for replication and recovery process during storage failure. However we also identify that reduction in storage cost cannot be minimized over the Erasure Codes to a maximum efficiency.
Hence in this work we also propose a theoretical framework to compress the data in order to achieve lower storage cost. We have seen a downgrade of 0.996% percent reduction of storage cost in the new algorithm. The percentage achieved may not be over helming, however considering the fact that this is achieved with minimal computational effort and this method also reduces the computational cost. Hence the total cost reduction can be considered as storage cost reduction plus computational cost reduction.

IX. FUTURE SCOPE

The work has generated satisfactory understanding and results. However application of the same process on various cloud storage systems need to be carried out. Here we list the future scope of this work. Firstly, the same algorithm needs to texted on various domain specific data. Secondly, the same compression algorithm needs to be tested on reduction of code size used during the recovery process and finally, the new proposed theoretical method needs to validated over the performance measure parameters defined in this work.

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

REFERENCES

[1] K. Greenan, E. Miller, and T. J. Schwartz titled Optimizing Galois Field arithmetic for diverse processor architectures and applications in MASCOTS 2008: 16th IEEE Symposium on Modeling,Analysis and Simulation of Computer and Telecommunication Systems at Baltimore, MD hosted on September 2008.
[2] J. Luo, K. D. Bowers, A. Oprea, and L. Xu titled Efficient software implementations of large finite fields GF(2n) for secure storage applications in ACM Transactions on Storage hosted on February 2012.
[3] J. S. Plank titled A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems in Software—Practice & Experience hosted on September 1997.
[4] J. S. Plank, K. M. Greenan, and E. L. Miller titled Screaming fast Galois Field arithmetic using Intel SIMD instructions In FAST-2013: 11th USENIX Conference on File and Storage Technologies hosted on February 2013.
[5] H. P. Anvin titled The mathematics of RAID-6 at http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf hosted on 2009.
[6] H. Li and Q. Huan-yan titled Parallelized network coding with SIMD instruction sets at International Symposium on Computer Science and Computational Technology, IEEE, hosted on December 2008.
[7] Onion Networks. Java FEC Library v1.0.3. Open source code distribution hosted at http://onionnetworks.com/fec/javadoc/, 2001.
[8] J. S. Plank, S. Simmerman, and C. D. Schuman titled "Jerasure: A library in C/C++ facilitating erasure coding for storage applications—Version 1.2." at Technical Report CS-08-627, University of Tennessee, hosted on August 2008.
[9] L. Rizzo titled "Erasure codes based on Vandermonde matrices. Gzipped tar file posted" athttp://planete-bcast.inrialpes.fr/rubrique.php3?id_rubrique=10 hosted at 1998.
[10] The textbook by Peterson describes Reed-Solomon coding in a more classic manner.
[11] J. Blomer, M. Kalfane, M. Karpinski, R. Karp, M. Luby, and D. Zuckerman titled"An XOR-based erasure-resilient coding scheme. Technical Report TR-95-048" at International Computer Science Institute hosted on August 1995.
[12] W. W. Peterson and E. J. Weldon, Jr. titled "Error-Correcting CodesSecond Edition" at The MIT Press, Cambridge, Massachusetts hosted on 1972.
[13] M. O. Rabin titled "Efficient dispersal of information for security, load balancing, and fault tolerance" at Journal of the ACM 36(2)hosted on April 1989.
[14] M. Blaum, J. Brady, J. Bruck, and J. Menon titled "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures" at IEEE Transactions on Computing 44(2) hosted on February 1995.
[15] M. Blaum and R. M. Roth titled "On lowest density MDS codes" at IEEE Transactions on Information Theory hosted at January 1999.
[16] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J.Leong, and S. Sankar titled "Row diagonal parity for double disk failure correction. In FAST-2004: 3rd USENIX Conference on File and Storage Technologies" at San Francisco, CA hosted on March 2004.
[17] J. S. Plank, A. L. Buchsbaum, and B. T. Vander Zandentitled "Minimum density RAID-6 codes" at ACM Transactions on Storage hosted on May 2011.
[18] M.Blaum, J. Bruck, and A. Vardy titled "MDS array codes with independent parity symbols" at IEEE Transactions on Information Theory hosted on February 1996.
[19] M. Blaum titled "A family of MDS array codes with minimal number of encoding operations" at IEEE International Symposium on Information Theory at Seattle hosted on September 2006.

BIOGRAPHIES

Mr. Ram Prakash Kota is a **Senior System Architect** working for Medical Client, USA. He worked at BVRIT, Narsapur for 3 years as an **Assistant Professor** in MCA Department.He Worked as **IT Analyst** for 4 years for TCS. Currently he is pursuing his Ph.D in the area of Cloud Computing from AcharyaNagarjunaUniversity, Guntur, Andhra Pradesh, India.

Prof. Dr. Kurra Rajasekhara Rao is a Professor of Computer Science & Engineering and presently working as Director, Sri Prakash College of Engineering (SPCE), Tuni. He worked at KLCE/K.L.University for 20 years as a faculty member in various positions as HOD of CSE, HOD of IT,Vice-Principal, Principal, K L College of Engineering (Autonomous), and Dean (Administration), Dean (Faculty & Student Affairs) Dean (Exams&Evaluation) of KLU. Prof. Dr. Kurra Rajasekhara has more than 28+ years of teaching and research experience. Prof. KRR is actively engaged in the research related to Embedded Systems, Software Engineering and Knowledge Management. He had obtained Ph.D in Computer Science & Engineering from AcharyaNagarjuna University (ANU), Guntur,A.P. under the able guidance of Prof.P.Thrimurthy. He published more than 80 papers in various International/National Journals and Conferences, and produced 4 Ph.D still now. He is a member of The Institution of Electronics and Telecommunication Engineers (IETE-F 198746), Life member of The Indian Science Congress Association (India) (L16636), Life member of The Institution of Engineers (India) (M-134938-9), Life member of Indian Society for Technical Education (ISTE-LM6387) and Life member of Computer Society of India (CSI-000920).