Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 16 Issue 1 Version 1.0 Year 2016
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Novel Erasure Coding based on Reed Solomon Fault Tolerance for Cloud based Storage

By Ramprakash Kota & Dr. Rajasekhara Rao Kurra

ANU, India

Abstract- In the recent years growth in usage of Erasure codes for fault tolerance is been observed. The growth in distributed storage solutions is the root cause of this growth. Multiple research is been carried out to propose the optimal fault tolerance solution for distributed storage solutions. However the recent storage solutions have shown a migration towards to the cloud based storage solutions. The growth of cloud computing and the benefits to the customer is the core of this migrations. Thus the applications managing the storage solutions have also updated with the demand. Hence the recent researches are driven by the demand of optimal fault tolerance solutions. Here in this work we propose an optimal erasure code based fault tolerance solution specific for cloud storage solutions. The work is been considered for commercial cloud based storage solution. The final outcome of this work is improvement on Bit Error Rate for the proposed Novel Erasure Coding based on Reed Solomon Fault Tolerance for Cloud based service.

Keywords: erasure, raid, raid 4, raid 5, array code, reed – solomon code, azure, amazon s3.

GJCST-B Classification : E.2

A Novel Erasure Coding based on Reed Solomon Fault Tolerance for Cloud based Storage

Ramprakash Kota [α] & Dr. Rajasekhara Rao Kurra [σ]

Abstract- In the recent years growth in usage of Erasure codes for fault tolerance is been observed. The growth in distributed storage solutions is the root cause of this growth. Multiple research is been carried out to propose the optimal fault tolerance solution for distributed storage solutions. However the recent storage solutions have shown a migration towards to the cloud based storage solutions. The growth of cloud computing and the benefits to the customer is the core of this migrations. Thus the applications managing the storage solutions have also updated with the demand. Hence the recent researches are driven by the demand of optimal fault tolerance solutions. Here in this work we propose an optimal erasure code based fault tolerance solution specific for cloud storage solutions. The work is been considered for commercial cloud based storage solution. The final outcome of this work is improvement on Bit Error Rate for the proposed Novel Erasure Coding based on Reed Solomon Fault Tolerance for Cloud based service.

Keywords: erasure, raid, raid 4, raid 5, array code, reed – solomon code, azure, amazon s3.

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 Coding is 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 [1].

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 [1] [2].

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 [3].

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 massage 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 [3] [11] [12].

The rest of the work is organized such that in Section II we discuss the fault tolerance mechanisms for Non – Cloud but distributed storage systems, in Section

Author α : Senior System Architect, USA, Research Scholar, Department of CSE, ANU, India. e-mail: ramprakash.kota@gmail.com
Author σ : Director, Sri Prakash College of Engineering (SPCE), Tuni, India. e-mail: dr.amjanshaik@gmail.com

III we realise the Reed Solomon Fault Tolerance mechanism, in Section IV we propose the Novel Erasure Coding based on Reed Solomon Fault Tolerance for Cloud Based Storage, in Section V discuss the Erasure Coding mechanisms for Cloud Storage Service Providers, in Section VI we produce the results obtained for the proposed scheme and in Section VII we conclude.

II. Fault Tolerance Mechanisms for non – Cloud Distributed Storage

The standard fault tolerance mechanism depends on the erasure codes [4]. 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

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

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

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

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

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

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

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

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

Simply requires multiplying and adding words, and decoding involves solving a set of linear equations with Gaussian elimination or matrix inversion.

Furthermore, we understand the most popular coding techniques here.

a) RAID-4 and RAID-5

The RAID – 4 and RAID – 5 [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

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

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 [6] [9] 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:

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

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 – 1.

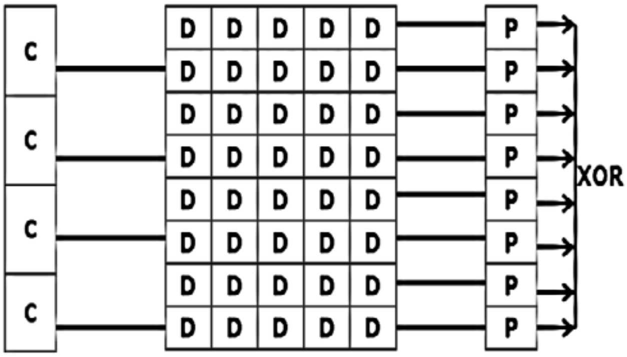

Figure 1: Array Code Storage

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.

III. Understanding Reed-Solomon Erasure

The most important factor that makes Reed-Solomon framework to implement is the simplicity. Here in this work we consider the scenario to compare the performance of Reed – Solomon and Proposed Encoding technique [7][8].

We consider there will be K storage devices each hold n bytes of data such that,

$$D = \sum D_1, D_2.D_3.....D_k \qquad \text{...Eq 7}$$

Where D is the collection of storage devices

Also there will be L storage devices each hold n bytes of check sum data such that,

$$C = \sum C_1, C_2, C_3....C_L \qquad \text{...Eq 8}$$

Where C is the collection of Checksum devices

The checksum devices will hold the calculated values from each respective data storage devices.

The goal is to restore the values if any device from the C collection fails using the non – failed devices.

The Reed – Solomon deploys a function G in order to calculate the checksum content for every device in C. Here for this study we understand the example of the calculation with the values as K = 8 and L = 2 for the devices $C_1$and C_2 with G_1 and G_2 respectively.

The core functionalities of Reed – Solomon is to break the collection of storage devices in number of words. Here in this example we understand the each number of words is of u bits randomly. Hence the words in each device can be assumed as v, where v is defined as

$$v = (nbytes).\left(\frac{8bits}{byte}\right).\left(\frac{1word}{u\,Bits}\right) \qquad \text{... Eq 9}$$

Furthermore, v is defined as

$$V=\frac{8n}{u} \qquad \text{...Eq 10}$$

Henceforth, we understand the formulation for checksum for each storage device as

$$C_i = W_i.(D_1, D_2, D_3...D_k) \qquad \text{...Eq 11}$$

Where the coding function W is defined to operate on each word

After the detail understanding of the Erasure fault tolerance scheme, we have identified the limitations of the applicability to the cloud storage services and propose the novel scheme for fault tolerance in this work in the next section.

IV. Proposed Novel Fault Tolerance Scheme

With the understanding of the limitations of existing erasure codes to be applied on the cloud based storage systems as the complex calculations with erasure codes will reduce the performance of availability measures significantly. Thus we make an attempt to reduce the calculation complexities with simple mathematical operations in the standard erasure scheme.

The checksum for storage devices are considered as C_i from the Eq 11. We propose the enhancement as the following formulation for checksum calculation:

$$C_i = W_i.(D_1, D_2, D_3...D_k) = W_i(D_1 \oplus D_2 \oplus D_3 ... \oplus D_k) \qquad \text{...Eq 12}$$

Here the XOR operation being the standard mathematical operation most suitable for logical circuits used in all standard hardware makes it faster to be calculated.

Also we redefine the function to be applied on each word for the storage devices D as following:

$$W = \begin{bmatrix} w_{1,1} & . & . & . & w_{1,L} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ w_{K,1} & . & . & . & w_{K,L} \end{bmatrix}_{K\,X\,L} \qquad \text{...Eq 13}$$

The proposed matrix will be stored on one of the devices and will be recalculated only once. As the modified checksum formulation is an XOR operation, thus which will automatically notify in case of any change.

The comparative simulations is also performed in this work and the enhancement in the performance is also been exhibited.

V. Erasure Coding Mechanisms for Cloud Storage Service Providers

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 [10].

a) 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:2].

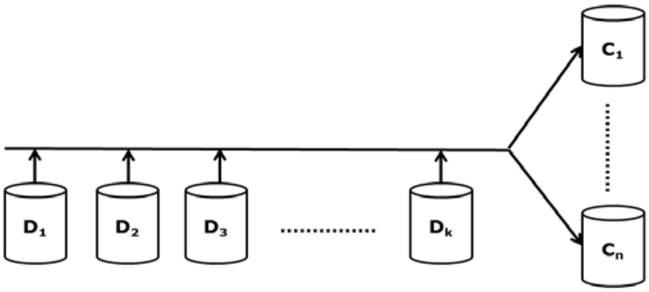

Figure 2 : LRC Computation

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.

b) 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:3]:

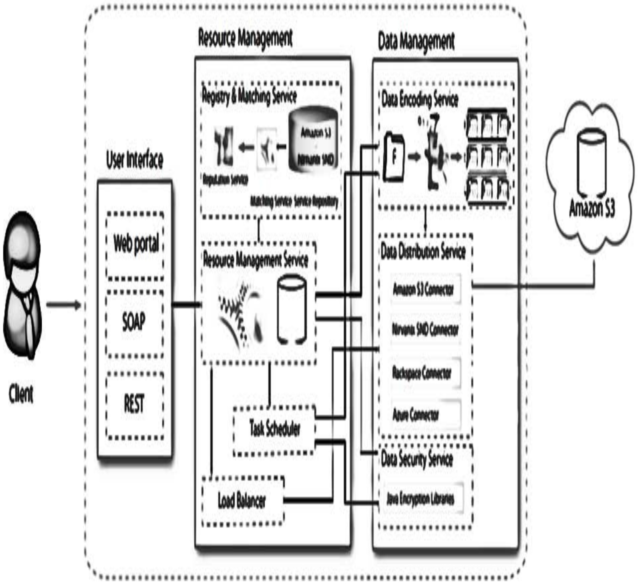

Figure 3 : Raid Soa

- 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.

c) 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. Results

The proposed fault tolerance scheme is been simulated and tested against the basic erasure fault tolerance scheme with the signal to noise ratio with Bit Error rate.

The first simulation results is the basic erasure fault tolerance code [Table – 1] shows the bit error rate for each signal to noise ranging from o to 15 decibel.

Table I : Basic Erasure Code Ber To Snr Simulation Results

Signal to Noise Ration	Bit Error Rate
0 Decibel	0.3645 %
1 Decibel	0.3362 %
2 Decibel	0.3037 %
3 Decibel	0.2674 %
4 Decibel	0.2280 %
5 Decibel	0.1868 %
6 Decibel	0.1458 %
7 Decibel	0.1070 %
8 Decibel	0.0728 %
9 Decibel	0.0452 %
10 Decibel	0.0250 %
11 Decibel	0.0120 %
12 Decibel	0.0049 %
13 Decibel	0.0016 %
14 Decibel	0.0004 %
15 Decibel	0.0001 %

The second simulation results in the proposed erasure based fault tolerance scheme [Table:II] shows the bit error rate for each signal to noise ranging from o to 15 decibel.

Table II : Proposed Fault Tolerance Scheme BER to SNR Simulation Results

Signal to Noise Ration	Bit Error Rate
0 Decibel	0.17310 %
1 Decibel	0.16220 %
2 Decibel	0.14940 %
3 Decibel	0.13490 %
4 Decibel	0.11850 %
5 Decibel	0.10060 %
6 Decibel	0.08160 %
7 Decibel	0.06210 %
8 Decibel	0.04290 %
9 Decibel	0.02530 %
10 Decibel	0.01190 %
11 Decibel	0.00410 %
12 Decibel	0.00100 %
13 Decibel	0.00010 %
14 Decibel	0.00000 %
15 Decibel	0.00000 %

Hence we realize the improvement in cloud based storage system and realized up to 59% improvement in the result [Table:III].

Table I : Basic Erasure Vs Proposed Fault Tolerance Scheme BER Comparison

Basic Erasure Scheme Bit Error Rate (%)	Proposed Scheme Bit Error Rate (%)	Improvement Percentage
0.3645	0.17310	47.5 %
0.3362	0.16220	48.2 %
0.3037	0.14940	49.2 %
0.2674	0.13490	50.4 %
0.2280	0.11850	52.0 %
0.1868	0.10060	53.9 %
0.1458	0.08160	56.0 %
0.1070	0.06210	58.0 %
0.0728	0.04290	58.9 %
0.0452	0.02530	56.0 %
0.0250	0.01190	47.6 %
0.0120	0.00410	34.2 %
0.0049	0.00100	20.4 %
0.0016	0.00010	6.3 %
0.0004	0.00000	100.0 %
0.0001	0.00000	100.0 %

The simulation results is also been generated using MATLAB simulation to observe the improvement [Figure:4].

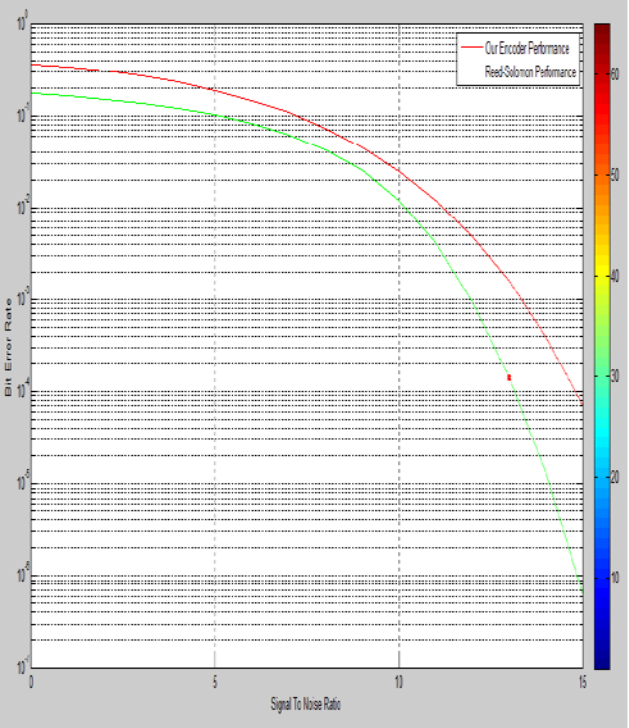

Figure 4 : Comparative Simulation

VII. Conclusion

This work compares the standard fault tolerance mechanisms for non-cloud based distributed storage solutions [9] [11] [12]. The work majorly focuses on RAID-4, RAID-5, Linux RAID-6, Array Codes and finally the Non - MDS Codes and realise the need for Erasure based codes for optimal performance. Also this work defines the parameters influencing the performance of Erasure codes in detail. Furthermore the work proposed an optimal cloud based fault tolerance code based on Erasure and evaluates the performance on multiple commercial cloud based storage solutions like Microsoft Azure, Amazon S3 and Finally Good File System. The simulation of the proposed fault tolerance scheme demonstrates up to 59% improvement in Bit Error Rate using the MATLAB simulation.

References References References

1. E. Pinheiro, W.D. Weber and L. A. Barro so, "Failure Trends in a Large Disk Drive Population," Proc. USENIX Conf. File and Storage Technologies (FAST', 07), Feb. 2007.
2. B. Schroeder and G.A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?, "Proc. USENIX Conf. File and Storage Technologies (FAST ',07), Feb. 2007
3. C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li and S. Yekhanin, "Erasure coding in Windows Azure storage", Proc. USENIX Annu. Tech. Conf. , pp.2 , 2012

4. M. Chowdhury, S. Kandula and I. Stoica, "Leveraging endpoint flexibility in data-intensive clusters", Proc. ACM SIGCOMM Conf. , pp.231 -242 , 2013
5. A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon and K.K. Rao, "A New Intra-Disk Redundancy Scheme for High-Reliability RAID Storage Systems in the Presence of Unrecoverable Errors," ACM Trans. Storage, vol. 4, no. 1, pp. 1-42, May 2008.
6. J. Plank, A. Buchsbaum and B. Vander Zanden , "Minimum density RAID-6 codes", ACM Trans. Storage, vol. 6 , no. 4 , pp.16 , 2011
7. O. Khan, R. Burns, J. S. Plank, W. Pierce and C. Huang, "Rethinking erasure codes for cloud file systems: Minimizing I/O for recovery and degraded reads", Proc. 10th USENIX Conf. File Storage Technol., 2012
8. M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen and D. Borthakur, "XORing elephants: Novel erasure codes for big data", Proc. VLDB Endowment , vol. 6, pp. 325 -336, 2013
9. Y. Zhu, P. Lee, L. Xiang, Y. Xu and L. Gao , "A cost-based heterogeneous recovery scheme for distributed storage systems with RAID-6 codes", Proc. IEEE 42nd Annu. Int. Conf. Dependable Syst. Netw., pp.1 -12 , 2012
10. B. Calder J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci, J. Hari das, C. Uddaraju, H. Khatri, A. Edwards, V. Bedekar, S. Mainali, R. Abbasi, A. Agarwal, M. F. ulHaq, M. I. ulHaq, D. Bhardwaj, S. Dayan and, A. Adusumilli, M. Mc Nett, S. Sankaran, K. Manivannan and L. Rigas, "Windows Azure storage: A highly available cloud storage service with strong consistency", Proc. 23rd ACM Symp. Operating Syst. Principles , pp.143 -157 , 2011
11. R. Li, J. Lin and P. P. Lee, "CORE: Augmenting regenerating-coding-based recovery for single and concurrent failures in distributed storage systems" , Proc. IEEE 29th Conf. Mass Storage Syst. Technol. , pp.1 -6 , 2013
12. S. Xu, R. Li, P. Lee, Y. Zhu, L. Xiang, Y. Xu and J. Lui, "Single disk failure recovery for X-code-based parallel storage systems", IEEE Trans. Comput., vol. 63, no. 4 , pp. 995 -1007, 2014

Global Journals Inc. (US) Guidelines Handbook 2016

www.GlobalJournals.org