ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

CLOUD BASED DATA RECOVERY & RECONSTRUCTION SYSTEM USING BI METHODOLOGY ERASURE CODE IMPLEMENTATION

Kayalvizhi P. M.E, CSE,
Sri Venkateswara College ofEngineeringand Technology,Thirupachur,
Thiruvallur,India.
Email:kayalvizhi0195@gmail.com

Mr.P.Rethina Sabapathi M.Tech, Ph.D.,
Head of the Department,
Associate Professor,
Department of ComputerScience & Engineering,
Sri Venkateswara College ofEngineeringandTechnology,
Thirupachur,Thiruvallur,India.

Abstract—In order to guarantee data reliability, erasure codes have been used in distributed storage systems. Nevertheless, this mechanism suffers from the repair problem that excess data are needed to repair a single failure, causing both high bandwidth consuming for the network and heavy computing load on the replacement node. To reduce repair traffic, researchers pointed out the tradeoff between storage and repair traffic and proposed regenerating codes by combining network coding. However, the combination only focuses on the storage terminal and the construction of the codes is quite complicated. Therefore, this paper further combines network coding with network structure and proposes a repair tree model based on general erasure codes to simplify the repairprocedure. By decomposing repair computing and distributing it among the tree nodes, our model can mitigate the computing tension. The performance of repair tree is analyzed and evaluated by preliminary emulation. The result shows it can make about three times faster computing than conventional measure and the repair throughput is doubled if there are network bottlenecks. For proper topology, it can significantly reduce the repair traffic. We present algorithms to generate trees across the network topology. At last, we present the idea of extending repair tree to repair multiple failures.

Index Terms—Distributed storage system, erasure codes, decomposed repair computing, single failure, repair tree, transitive vector,network coding, intermediate node.

I. INTRODUCTION

Traditional reconstruction techniques in storage clusters advocate the pull model, where a master node initiates reconstruction by sending requests to worker nodes dedicated to the reconstruction process. The passive pull model inevitably encounters a transmission bottleneck problem that lies in rebuilding nodes. In this paper, we propose two PUSH-based reconstruction schemes—PUSH-Rep and PUSH-Sur—to improve reconstruction performance in a distributed storage cluster. At the heart of this study is the proactive PUSH technique that evenly distributes network and I/O loads among surviving nodes to shorten reconstruction times. The following three factors motivate us to propose the PUSH-based reconstruction technique for erasure-coded clustered storage. The high cost-effectiveness of erasure-coded storage, the severe impact of recovery time on reliability, and the deficiency of PULL-based reconstruction I/Os. 1. Erasure-coded storage clusters have increasingly become a cost-effective and fault-tolerant solution for archive storage data centers cloud storage and the like. Especially, Reed-Solomon (RS) codes are widely used in storage clusters to provide high data reliability. For example, Windows Azure Storage (WAS) adopts a variant of RS codes to implement a four-fault-tolerant cluster system. A detailed review on the RS-coded distributed storage is provided in Motivation 2. Ideally, erasure-coded storage clusters should protect against data loss caused by node failures, because high reliability is an indispensable requirement for building large-scale storage systems. The mean-time-to data- loss or MTTDL of a r-fault-tolerant storage system is inversely proportional to the rth power of recovery time of a storage node.

Therefore, it is extremely important to speed up the reconstruction process, which in turn can improve system reliability by shrinking vulnerability window size. The existing reconstruction schemes adopt a PULL-transmission mode, where a rebuilding node initiates the reconstruction by sending read requests to fetch/pull surviving blocks. Such a PULL mode not only raises the TCP In cast problem due to its synchronized many-to-one traffic pattern, but also yields poor reconstruction performance. When it comes to a reconstruction which relies on replacement nodes, the network traffic of replacement nodes contributes to an excessively long reconstruction time. The problem with the reconstruction among surviving nodes is that each surviving node bears extra seek time owing to the non-contiguous disk access. This problem makes the low write bandwidth become a major reconstruction performance bottleneck. In this paper, we introduce a PUSH-type transmission to speed up node-reconstruction performance. Our PUSH enables surviving nodes to accomplish reconstruction tasks in a pipelining manner. Each surviving node combines its local block with an intermediate block from another surviving node to partially generate an intermediate block forwarded to a subsequent node. Thus, PUSH can speed up the reconstruction process by maximizing the utilization of both network and I/O bandwidth of all the surviving nodes.

II.RELATEDWORKS

We review related work on the repair problem for erasure coding based distributed storage system. Several researches

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

focus on minimizing the degrade effect by optimizing rowdiagonal parity (RDP) recovery and disk-oriented reconstruction (DOR) [31] recovery. These researches hardly change the natural dilemma of repair cost, especially when transplanting these methods in a more distributed environment. The outcome of RGC first clarifies the relationship between storage and repair traffic. The main feasible RGC are minimal storage regenerating codes and minimal bandwidth regenerating (MBR) codes. Subsequently, related researches propel the fast development of regenerating codes [10], [32], [33]. However, MSR and MBR hardly reduce the storage redundancy smaller than 2 for exactrepair and always need overmuch computing and I/Os for functional repair. Moreover, it is still hard to keep systematic MDS property under 2_ storage overhead [34]. Because RGC bring complicated operation, Local repair codes (LRC) sacrifice a little storage overhead to exchange for better repair performance, which make secondary MDS coding for partial strips [35], [36], [37]. Our repair approach reserves the whole property of erasure codes without introducing extra storage overhead and complicated codes design. Table 2 depict the comparison between repair tree and MSR. By referring linear network coding [14], repair treeintroduces computational nodes to XOR helping data flows to reduce network traffic. The differences between them are that network coding transmits enough information entropy to collector for decoding the original data without dropping information while repair tree compress the redundant data via the intermediate nodes and transfer the useful data to last destination. Recently, the authors of literatures [38], [39], [40] propose the similar idea by using the linear combination property of erasure code and even regenerating codes to reduce traffic. In their models, terminal helpers are treated as relays which receive data from other helpers, encode them with the own stored data and then re-send the results out till the final replacement node receives wanted data. However, these models have not yet considered thetruly important roles of network topology: routers and switches, which might lead to the data transmitted back and forth between routers (switches) and severs. Our repair tree model can effectively avoid the occurrence of such things.

III. PROPOSEDSYSTEMDESIGN

In the proposed system, we are implementing Two Techniques namely, PUSH–Rep & PUSH–Sur. In PUSH–Rep Reconstruction occurs using Replacement Nodes. Rebuilt blocks are sequentially written to the disks of replacement nodes. PUSH–Sur allows each surviving node to rebuild a subset of failed data, so all the surviving nodes accomplish the reconstruction in parallel.

SYSTEM DESIGN

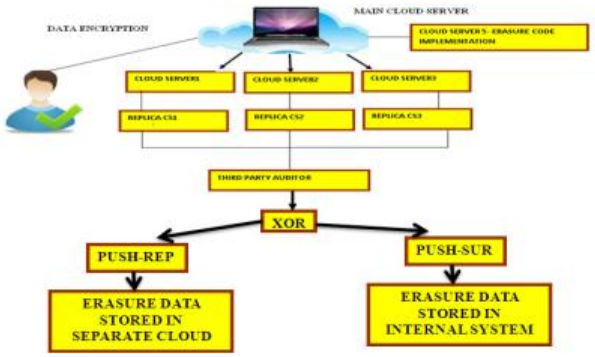

Figure 1.ARCHITECTURE

Figure.1describes theoverallscenariointheproposed system. We are deploying this Application in Cloud. In this Data is Encrypted, Splitted and stored in different Cloud. The Replica is created for data backup.Top Hash Key is stored in Separate Cloud as well in the Local Backup. We implement PUSH–Rep using reconstruction from Cloud Backup and PUSH–Sur reconstruction from Local Backup.

ALGORITHM / METHODOLOGY: Eraser Code

In this the Error Control Codes aredesigned for Ease of Encoding &Decoding (Calculation of syndrome / location of error) and Error Correction Power (Burst Errors / Low Redundancy) and the Error Correcting Codes are Most applications use hardware implemented encoding and decoding Protect against erasure of data.

Authentication M odules

In this The Authentication moduleconsistof several mainprocesses. Data owner, Main cloud server, Data splitting and Encryption, Parity bit addition and Erasure code, Trusted party auditor, Replica server.

1. DATA OWNER:

User is the person is going to see or download the data from the Cloud server. To access the data from the Cloud server, the users have to be registered with the cloud server. So that the user have to register their details like username, password and a set of random numbers. This is information will stored in the database for the future authentication. Data Owner: Data Owner is the Person who is going to upload the data in the Cloud Server. To upload the data into the Cloud server, the Data Owner have be registered in the Cloud Server. Once the Data Owner registered in cloud server, the space will be allotted to the Data Owner.

2. MAIN CLOUD SERVER:

Cloud Server is the area where the user going to request the data and also the data owner will upload their data. Once the user send the request regarding the data they want, the request will first send to the Cloud Server and the Cloud Server will

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

forward your request to the data owner. The data Owner will send the data the data the user via Cloud Server. The Cloud Server will also maintain the Data owner and Users information in their Database for future purpose.

3. DATA SPLITTING AND ENCRYPTION:

In this module, once the data was uploaded into the cloud server, the Cloud server will split the data into many parts and store all the data in the separate data servers. In techniques wasn't used in proposed system so that there might be a chance of hacking the entire data. Avoid the hacking process, we splitting the data and store those data in corresponding data server. We're also encrypting the data segments before storing into the data server.

4. PARTY BIT ADDITION AND ERASURE CODE:

Once the data are stored in the corresponding data servers and the keys are stored in the key servers. Then we're adding the parity bits to the data, so that the data will be changed. Also we're applying the Erasure Code by using the XOR operation, while XORing the block data, the data will be converted in binary data.

5. TRUSTED PARTY AUDITOR:

Once added the parity added bits, then the data will be given to the Trusted Parity auditor. The Trusted Parity Auditor will generate the signature using change and response method. The data will be audited in this module, if any changes occur it will provide the intimation regarding the changes.

6. REPLICA SERVER:

We'll maintain the separate Replica Cloud server. If suppose the data in the data server was lost, then the Main Cloud server will contact the Replica Cloud server and get the data from the Replica Cloud Server. By using this concept, we can get the data if any data loss occurs. [4] discussed that the activity related status data will be communicated consistently and shared among drivers through VANETs keeping in mind the end goal to enhance driving security and solace. Along these lines, Vehicular specially appointed systems (VANETs) require safeguarding and secure information correspondences. Without the security and protection ensures, the aggressors could track their intrigued vehicles by gathering and breaking down their movement messages. A mysterious message confirmation is a basic prerequisite of VANETs. To conquer this issue, a protection safeguarding confirmation convention with expert traceability utilizing elliptic bend based chameleon hashing is proposed. Contrasted and existing plans Privacy saving confirmation utilizing Hash Message verification code, this approach has the accompanying better elements: common and unknown validation for vehicle-to-vehicle and vehicle-to-roadside interchanges, vehicle unlinkability, specialist following capacity and high computational effectiveness

IV. CONCLUSION

Nowadays a grand challenge for storage clusters is efficiently migrating data replicas to create an erasure-coded archive. To take this challenge, we are going to integrate the PUSH-type transmission into the archival migration in erasure-coded storage clusters. Moreover, since PUSH-based reconstruction schemes are sensitive to slow nodes, we plan to extend the PUSH-based reconstruction schemes for heterogeneous erasure-coded storage clusters by taking into account both load and heterogeneity of surviving nodes. To address these issues, we proposed the PUSH approach, in which a PUSH-type transmission is incorporated into node reconstruction. We developed two PUSH-based reconstruction schemes (i.e., PUSH Rep and PUSH-Sur). Compared to the PULL-based counterparts where surviving blocks are transferred in a synchronized 'M:1' traffic pattern, our PUSH-based reconstruction solutions support the '1:1' pattern, which naturally solves the In cast problem. We built performance models to investigate the reconstruction times of our PUSH-based schemes applied in large-scale storage clusters. We extensively evaluated the four schemes on a real-world cluster.

V. FUTURE ENHANCEMENT:

To take this challenge, we are going to integrate the PUSH-type transmission into the archival migration in erasure-coded storage clusters. Moreover, since PUSH-based reconstruction schemes are sensitive to slow nodes, we plan to extend the PUSH-based reconstruction schemes for heterogeneous erasure-coded storage clusters by taking into account both load and heterogeneity of surviving nodes.

REFERENCES

[1] S. Frolund, A. Merchant, Y. Saito, S. Spence, and A. Veitch, "A decentralized algorithm for erasure-coded virtual disks," in Proc. Int. Conf. Dependable Systems Networks, 2004, pp. 125–134.

[2] M. Storer, K. Greenan, E. Miller, and K. Voruganti, "Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage," in Proc. 6th USENIX Conf. File Storage Technol., 2008, p. 1.

[3] A. Thusoo, Z. Shao, S. Anthony, D. Borthakur, N. Jain, J. Sarma, R. Murthy, and H. Liu, "Data warehousing and analytics infrastructure at facebook," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2010, pp. 1013–1020.

[4] Christo Ananth, Dr.S. Selvakani, K. Vasumathi, "An Efficient Privacy Preservation in Vehicular Communications Using EC-Based Chameleon Hashing", Journal of Advanced Research in Dynamical and Control Systems, 15-Special Issue, December 2017,pp: 787-792.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 1, January 2018

[5] B. Calder et al., "Windows azure storage: A highly available cloud storage service with strong consistency," in Proc. 23rd ACM Symp. Operating Syst. Principles, 2011, pp. 143–157.

[6] O. Khan, R. Burns, J. Plank, W. Pierce, and C. Huang, "Rethinking erasure codes for cloud file systems: Minimizing I/O for recovery and degraded reads," in Proc. 10th USENIX Conf. File Storage Technol., 2012, pp. 251–264.

[7] J. Plank et al., "A tutorial on reed-solomon coding for fault-tolerance in raid-like systems," Softw. Practice Experience, vol. 27, no. 9, pp. 995–1012, 1997.

[8] M. Manasse, C. Thekkath, and A. Silverberg, "A reed-solomon code for disk storage, and efficient recovery computations for erasure- coded disk storage," Proc. Inf., pp. 1–11, 2009.