International Journal of scientific research and management (IJSRM)
||Volume||4||Issue||11||Pages||4929-4935||2016||
Website: www.ijsrm.in ISSN (e): 2321-3418

Efficient data administration with reed-Solomon code

Veena Mishra, Dr. R. K. Pateriya
CSE Department
Manit Bhopal
India
veenamishra038@gmail.com
pateriyark@gmail.com

Abstract: Cloud computing is a novel computing paradigm which is recognized as an arbitrary to traditional reference technology right to its intrinsic resource-sharing and low-maintenance characteristics. One of the virtually fundamental services offered by CSPs (Cloud Service Providers) is cloud storage. To increasing reliability and efficiency of data storage in the cloud the technique used is replication, but its drawback is data loss and higher space consumption. One way to increase the data reliability and reducing the storage space in the cloud is Erasure Coding. In Erasure Coding, the data is fragmented and further encoded mutually into data pieces and stored in different locations. The arbitrary benefit of the Erasure Coding is that the corrupted data can be absolutely reconstructed into separate information. Erasure code comprises of two coding techniques regenerating code and locally repairable code. Regenerating Code is used for balancing storage space and its bandwidth. The Locally repairable code is the technique used to overcome the Disk I/O overhead in the Cloud Storage. The Reed-Solomon code stored data into multiple storage node and encode the data into multiple fragments then perform decoding operation to achieve storage cost with the same level of fault tolerance and consumed time.

Keywords: Cloud storage, Replication, Erasure coding, Reed- Solomon code.

Introduction: With the decreasing of information measure and computerised data valuation, an objective has been determined that foremost IT corporations, a well known as Google, Microsoft, and Amazon, establish their services inside data centers and extend services globally over a high-bandwidth network. This new paradigm of providing computing services is termed cloud computing. which is well-known as an absolute to ancient information technology due to its intrinsic resource-sharing and low-maintenance characters [1]. one of the virtually fundamental services offered by CSPs (Cloud Service Providers) is cloud storage. By migrating the native information management directed toward the cloud, users will enjoy high-quality services and gather significant investment on their local infrastructure.

Since the clouds are sometimes operated by industrial CSPs that are very likely to be outside

of the trusted domain of the users, it's quite impendent for the cloud to produce information responsibility and confidentiality. To attain the responsibility, several proposals are planned to introduce information redundancy to avoid information unretrievable within the case of some information shares are missed accidentally.

Cloud storage:

In the framework of cloud computing, computerized information has not only been a consistent component of large-scale cloud services but furthermore been provided as a virtual storage infrastructure in a pay-as-you-go approach, a well known as Amazon S3(Simple storage service). Moreover, the volume of data stored inside data centers has been observed instant growing eventually faster than Moore’s Law[2]. It has been released that the space for storing used for icon storage only in Facebook has been around 20PB in 2011 and is increasing by 60 TB every week[3]. To approach the necessities of the substantial volume of storage, the cloud storage system needs to grow out, i.e., storing information in a very large number of artifact disks. during this plan, it becomes a significant challenge for cloud storage systems to set up data integrity, the right to both an outsized variety of disks and their artifact nature. Even though the number of disk failures is a small portion of the data centers, there can still be a large number of such failures every day due to a large number of disks. For example[4], in a Facebook cluster with 3000 nodes, there are originally at uttermost 20 repairs triggered every day. Apart from storage devices, the contrasting systems in the data center, one as the networking or thing systems, am within one area cause outages in the data center[4], making data having a full plate or even gain lost.

To increase reliability and efficiency of data storage in the cloud two technique are used :

1. Replication
2. Erasure Code

Cloud file systems transform the requirements for erasure codes because they have properties and workloads that differ from traditional file systems and storage arrays. The model for a cloud file system using erasure codes is inspired by Microsoft Azure [5]. It conforms well with HDFS [6] modified for RAID-6 [7] and Google’s analysis of redundancy coding [8]. Some cloud file systems, such as Microsoft Azure and also the Google File system, produce an append-only write workload employing a massive block size. Writes are accumulated and buffered till a block is full and so the block is sealed: it's erasure coded and also the coded blocks are distributed to storage nodes. Consequent reads to sealed blocks usually access smaller amounts information than the block size, depending upon workload [9]. To reduce storage overhead, cloud file systems are transforming from replication to erasure codes. This method has disclosed new dimensions on which to judge the performance of various coding schemes: the amount of information utilized in recovery and when performing degraded reads.

Replication:

Although wide-scale replication has the potential to extend availableness and durability, it introduces two vital challenges to system architects. First, system architects should increase the amount of replicas to attain high durability for giant systems. Second, the increase in the range of replicas will increase the bandwidth and storage necessities of the system. Replication is the simplest redundancy scheme; here k identical copies of every data object are kept at each instant by system members. The worth of k should be set suitably depending on the desired per object inaccessibility target, (i.e., 1 − has some "number of nines"), and on the average node availableness, a. Assuming that node accessibility is independent and identically distributed (I.I.D.), and assuming we only need one out of the k replicas of the information to be accessible so as to retrieve it (this would be the case if the information is immutable and so one accessible copy is sufficient to retrieve the right object), we calculate the subsequent values for ε- ε = P(object o is unavailable) = P(all k replicas of o are unavailable) = P(one replica is unavailable) [k = (1−a)k] which upon solving for k yields [k=log/log(1−a)] •Its disadvantage is information loss and higher space consumption.

Erasure code: Before the emergence of cloud computing, erasure coding has long been proposed to observe or correct errors in storage or communication systems. Erasure codes give a storage efficient solution and ensure high information accessibility using significantly less space for storing than replication. However, once erasures occur and erased information has to be restored for long-run persistence, the repairing method of erasure coded information is a smaller amount efficient than in replication. Once replicated information is erased, repairing is simply done by replicating one in all the remaining replicas (when exists). On the opposite hand, once encoded information is erased, the repairing node first has to transfer k chunks and reclaim an entire copy of the initial file.

Erasure coding during a malicious atmosphere needs the precise identification of unsuccessful or corrupted fragments. While not the flexibility to identify corrupted fragments, there's probably a factorial combination of fragments to try to reconstruct the block; that's combinations. As a result, the system has to find once a fragment has been corrupted and discard it. A secure verification hashing theme will serve the dual purpose of characteristic and confirming every fragment. it's essentially the case that any correctly verified fragments are often wont to reconstruct the block. Such a theme is probably going to extend the bandwidth and storage requirements, however is shown to still be again and again less than replication. When examining erasure codes within the context of cloud file systems, two performance essential operations emerge. These are degraded reads to temporarily unavailable information and recovery from single failures. Though erasure codes tolerate multiple simultaneous failures, single failures represent 99.75% of recoveries [9]. Recovery performance has forever been vital. Previous work includes design support and workload optimizations for recovery [10].

3. Brief Review:

Cloud Computing is a novel computing paradigm which is recognized as an arbitrary to traditional reference technology right to its intrinsic resource-sharing and low-maintenance characterstics. One of the virtually fundamental services offered by CSPs (Cloud Service Providers) is cloud storage.

Techniques to achieve efficient data management:

The default storage policy in cloud file systems has become triplication (triple replication), implemented in the Google Filesystem [11] and adopted by Hadoop[6] and many others. Triplication has been favored because of its ease of implementation, good read and recovery performance, and reliability.

The storage overhead of triplication is a concern, leading system designers to consider erasure coding as an alternative. The performance tradeoffs between replication and erasure coding are well understood and have been evaluated in many environments, such as peer-to-peer file systems [15] and open-source coding libraries [12].

Wolfson et al. (1997)		The algorithm that changes the replication scheme as changes occur in the read–write pattern. The algorithm continuously moves the replication scheme toward an optimal one.
Moore (2002)		swiftly increasing as storage requirements are rising by 60%annually
Lamehamediet al. (2002)		presented a set of replica management services and protocols to offer high data availability, low bandwidth consumption, improved fault tolerance, and scalability of the system by considering the access cost and replication gains.
Ranganathan et al. (2002)	Dynamic and Model-driven replication strategy	Automatically produces copies in a decentralized manner whenever it is required to improve the system availability. In this model, all the peers are independent to take replication decision and they can create copies of files they store
Shafi et al. (2003)		studied real web server workloads from sports, e-commerce,financial, and internet proxy cluster and found that the average server utilization varies between 11% and 50%. The reason for the low utilization is because the system has to offer overprovision to guarantee performance at the periods of peak loads. This observation gives us opportunities to reduce the energy consumption of clusters.
Pinheiro et al. (2003)		developed a system that dynamically turns cluster nodes on/off to handle the load imposed on the system. The system makes reconfiguration decisions by considering the total workload imposed on the system, the power, and performance implications of changing the current configuration.
Elnozahy et al. (2003)		employed various combinations of dynamic voltage scaling and node vary-on/vary-off to reduce the aggregate power consumption of a server cluster during periods of reduced workload.
Park et al. (2004)		improve the network locality by replicating the files within the network region
Tang et al. (2005)	two dynamic replication algorithms	including simple bottom up and aggregate bottom up to reduce the average response time. In the proposed architecture, each node at any middle tier provides resources to the lower tier nodes as a server. A replication decision is made only at the dynamic replication scheduler which maintains information about the data access history and client access pattern.
Geet al., (2007)	MISER a run-time DVFS scheduling system	MISER is capable of providing fine-grained performance-directed DVFS power management for a power-aware cluster
Fan et al. (2007)		investigated the power consumption of a typical server. They reported that a disk drive takes 12 W. From a power standpoint, it seems the power consumption of a single disk drive is not a problem.
Yuan et al. (2007)	Dynamic data replication strategy	considering the bottleneck of the data grid storage capacity of different nodes and the bandwidth available between these nodes .
Deng and Wang (2008).		Green computing has been a hot research topic in the community of cluster computing for many years. It is more challenging for the storage clusters because

		of the explosive growth of data
Verma et al. (2008)		employed power management techniques such as dynamic consolidation and dynamic power range enabled by low power states on servers to reduce the power consumption of high-performance applications on modern power efficient servers with virtualization support.
Caulfield et al.,(2009)	Gordon	utilize slow-power processors and flash memory to reduce the power consumption and improve performance for data-centric cluster
Huang and Feng (2009)	a run-time DVFS scheduling algorithm	algorithm for a cluster system to reduce the energy consumption.β-algorithm (Hsu and Feng, 2005)is a run-time DVFS scheduling algorithm that is able to transparently and automatically reduce the power consumption while maintaining a specified level of performance.
Andersen et al.(2009)	FAWN	combines low-power CPUs with small amounts of local flash storage, and balances computation and I/O capabilities in order to offer low-power, efficient, and parallel data access on a large-scale cluster.
Khan et al., 2011	PHFS	Uses predictive techniques to predict the future usage of files and then pre-replicates the files in a hierarchal data grid on the path from source to client
Huang et al.,(2013)	ECS2	utilizes data redundancies and deferred writes to conserve energy for erasure-coded storage clusters. The parity blocks are buffered exclusively in active data nodes whereas parity nodes are placed into a low-power mode, thus saving energy

Investigations into applying RAID-6 (two faults tolerant) erasure codes in cloud file systems show that they reduce storage overheads from 200% to 25% at a small cost in reliability and the performance of large reads [14]. Microsoft research further explored the cost/benefit trade-offs and expand the analysis to new metrics: power proportionality and complexity. For these reasons, Facebook is evaluating RAID-6 and erasure codes in their cloud infrastructure [7].

Proposed work:

In this work reed, Solomon based efficient storage algorithm is proposed for data replication. Hadoop provides a solution to Big data problem. To handle big data two challenges are there:

- First is to store data.
- Second is to process data

The proposed scheme writes a full block on the primary DataNode and then performs erasure coding with Vandermonde-based Reed-Solomon algorithm that divides data into m data fragments and encode them into n data fragments (n>m), which are saved in N distinct DataNodes such that the original object can be reconstructed from any m fragments.

The Hadoop distributed file system provides a fault tolerant and reliable way of distributed storing data. First, data is divided into blocks and then each block is assigned a data node by the Namenode. As the cluster consists of commodity hardware to offer fault tolerant nature replication of blocks is done. In the latest version of Hadoop, the default block size is 128 MB. Data is put to cluster by the user. Data is divided into blocks and placed on data node. After successful placement of data block acknowledgment is sent to the master. In this way, master form metadata. This metadata will be used when the user wishes to access the data again[16].

- **Data Placement algorithm of Hadoop:**

Data_Placement

{

1. Data is put on HDFS by the user using put

DOI: 10.18535/ijsrm/v4i12.3

command.

2. First data is divided into blocks of 128 MB.
3. Each block is placed on a data node and acknowledgment is sent to master after successful placement.
4. Replication is done by data nodes and master is informed.
5. Metadata is created on Namenode about a number of blocks, the location of data nodes where blocks are placed and their replication.

}

- **Data replication process of Hadoop**

To provide fault tolerant nature Hadoop replicates every block of the file. By default, 3 replicas are formed. The first copy is placed on the data node geographically closest to the user. This is done to reduce the access cost. Then data node having an original block replicates it to other data node and this data node will again replicate the block resulting in 3 replicas.

Data_Replication

{

1. The first copy will be placed on the closest data node to the user and high priority.
2. The second replica will be formed by the above data node on the machine with moderate priority and available space.
3. The third replica will be formed by above data node on a machine having lower priority.

}

- **Proposed data storage algorithm based on Reed-Solomon code**

Write different file using shell cmd "put" and observe the storage size acquired by that writing in both scheme. Reed-Solomon divided systems in a cluster into two parts, one having the data and other having parity bits for providing fault tolerance. It provide space efficiency of the erasure code reed soloman algorithm.

The Equation is:

$$[X+X(1/r)];$$

Where

r= m/n & n>m

X=Size of data file

r= Encoding rate

m= No of fragment data is divided into

- **Algorithm:**

Efficient data-Storage with Reed-Soloman Code

Start

Step1: In a cluster take the metadata file

Step2: If data log found in the metadata file

{

Data is cold data

{

Encode using Replication // Here we found 3X replication factor.

}

Else

Data is hot data

{

Encode using Reed-Solomon code (4,2)

}

}

Step3 : Delete that block using random generator

Step4: Recover the data using Decode function

Step 5: Calculate time AND SPACE with different entries of the file.

- **Result Analysis:**

In an example:

1. When we take the (3,1) then

Efficiency= 16.67%

2. When we take the (4,2) then

Efficiency= 33%

In the experimental result it is founded that the reduced space for the storage of data by 16.67% for (3,1) and 33% for (4,2) machine.

- **Result graph:**

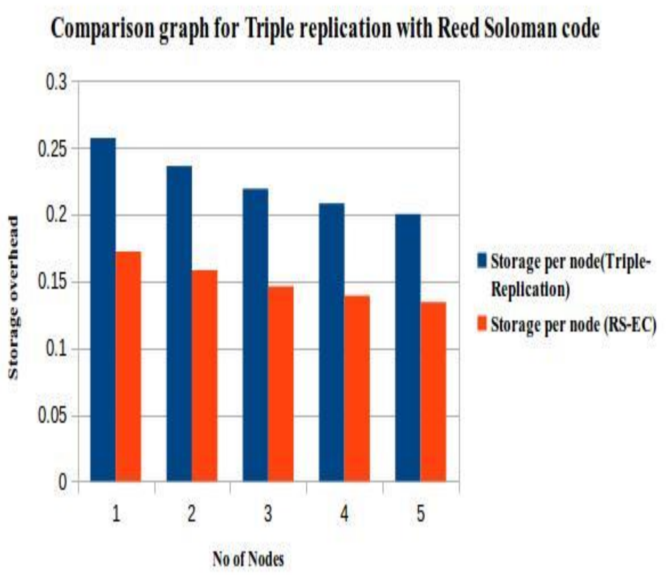

Figure: Gain efficiency in 5 nodes between storage triple replication and erasure code RS

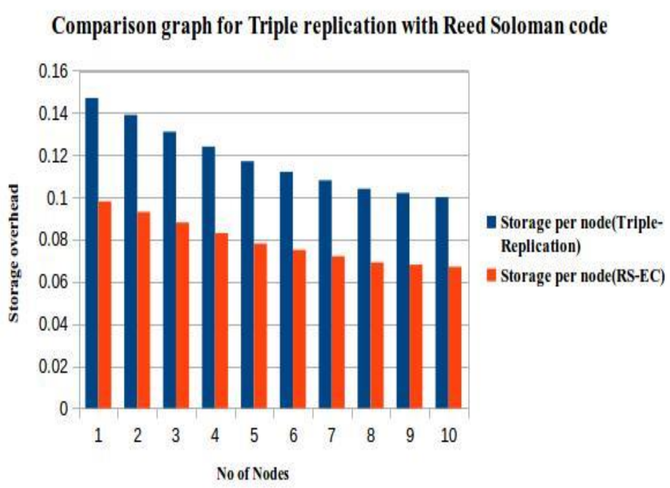

Figure: Gain efficiency in 10 node between replication and erasure code.

- **Comparison between replication and Reed-Solomon EC method:**

Techniques	Equation	Result
In Triple replication: The replication factor is 3X	X=256MB 256*3=768MB	The efficiency is less in this algorithm
In Reed-Solomon Erasure code storage overhead is in (3,1)	Storage efficiency= [X+{X*(1/r)}] When X=256MB Then Efficiency= 640MB	The efficiency is achieve by the 16.67% of 3Xreplication
In Reed-solomon Erasure code storage overhead is in (4,2)	Storage efficiency= [X+{X*(1/r)}] When X=256MB Then Efficiency= 384MB	The efficiency is achieve by the 33% of 3Xreplication

Conclusion:

- In this research work, we proposed an Erasure code with Reed-Solomon code Approach for the Cloud Computing.
- As a parameter, it minimizes the amount of storage consumed with the same level of fault tolerance and execution time. The comparative graphs are shown to be **storage consumption** between “triple replication and Reed-Solomon erasure code” techniques.

Future works:

In the future,

- We can extend our implementation from resource utilization as a parameter to CPU, bandwidth, RAM altogether as a parameter. This will improve efficiency of the storage system using XOR calculation.

- Minimizes storage consumption and fault tolerance, XOR system give optimized results after each iteration, so we can integrate the XOR system with Cloud server for continuous monitoring of storage during different time slots.
- Consequently, by using XOR system, we can improve the efficiency of the Cloud storage approach in future. This approach can be integrated with other existing storage algorithm for the best result which any algorithm can get[17].

References:

1. "Jun Li" and "Baochun Li", " Erasure Coding for Cloud Storage Systems: A Survey", TSINGHUA SCIENCE AND TECHNOLOGY ISSNll1007-0214ll06/11llpp259-272 Volume 18, Number 3, June 2013.

2. "S.-Y. R. Li", "R. W. Yeung", and "N. Cai", Linear network coding, IEEE Trans. on Inform. Theory, vol. 49, no. 2, pp. 371-381, 2003.

3. IDC says world's storage is breaking Moore's law, more than doubling every two years, http://enterprise.media.seagate.com/2011/06/insideit-storage/idc-says-worlds-storage-is-breaking-mooreslaw-more-than-doubling-every-two-years/, 2012.

4. "A. W. Kosner", Amazon cloud goes down Friday night, taking Netflix, Instagram And Pinterest withit, http://www.forbes.com/sites/anthonykosner/2012/06/30/amazon-cloud-goes-down-friday-night-taking-netflixinstagram-and-pinterest-with-it/, Forbes, June 30, 2012.

5. "B. Calder", "J. Wang", "A. Ogus", "N. Nilakantan", "A. Skjolsvold", "S. McKelvie" and "Y. Xu", Windows Azure storage: A highly available cloud storage service with strong consistency. In Symposium on Operating Systems Principles, 2011.

6. "D.Borthakur". The Hadoop distributed file system: Architecture and design. http://hadoop.apache.org/common/docs/current/hdfs-design.html, 2009.

7. "B. Fan"," W Tanisiriroj", "L. Xiao", and "G. Gibson", " DiskReduce: RAID for data-intensive scalable computing: In Parallel Data Storage" Workshop, 2008.

8. D. Ford, F. Labelle, F. .I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan. Availability in globally distributed file systems. In Operating Systems Design and Implementation, 2010.

9. B. Schroeder and G. Gibson. Disk failures in the real world: What does an MTTF of 1,000,000 mean to you? In Conference on File and Storage Technologies, 2007.

10. "A.L.Drapeauetal", RAID-II: A high-bandwidth network file server. In International Symposium on Computer Architecture, 1994.

11. S. Ghemawat, H. Gobioff, and S. Leung. The Google file system. In ACM SOSP, 2003.

12. J.S.Plank,J.Luo,C.D.Schuman,L.Xu,andZ.WilcoxOHearn . A performance evaluation and examination of open-source erasure coding libraries for storage. In Conference on File and Storage Technologies, 2009.

13. "Roy Friedman" and "Yoav Kantor" , "Israel and Amir Kantor" , "Replicated Erasure Codes for Storage and Repair-Traffic Efficiency" Israel 14-th " IEEE International Conference on Peer-to-Peer Computing", 978-1-4799-6201-3/14/$31.00, 2014 .

14. "Rodrigo Rodrigues" and "Barbara Liskov", "High Availability in DHTs: Erasure Coding vs. Replication", 2011 IEEE .

15. "Hakim Weatherspoon" and "John D. Kubiatowicz", "Erasure Coding vs. Replication: A Quantitative Comparison" Computer Science Division University of California, Berkeley, In Proc. IPTPS '02.

16. Qingsong Wei and Bharadwaj Veeravalli, Bozhao Gong, "CDRM: A Cost-effective Dynamic Replication Management Scheme for Cloud Storage Cluster" International Conference on Cluster Computing, IEEE, 2010 , pp 188-196.

17. Maheswaran Sathiamoorthy University of Southern California , "XORing Elephants: Novel Erasure Codes for Big Data", Proceedings of the VLDB Endowment, Vol. 6, No. 5, IEEE 2013march, pp325-336.