A modified cost-effective dynamic data replication algorithm by using copy set replication technique.

[1]Faraidoon Habibi, [2] Dr.Nagesh Kumar
[1]Masters in Technology, [2]Assistant Professor
[1]Department of Computer Science & Engineering,
[1]A P Goyal Shimla University, Shimla, India

Abstract: Data replication offers accessibility of data in numerous sites to guarantee effective application of data. Since, the extreme evolution of data procedure, the cloud computing is receiving more preferences currently. As maximum corporations are using cloud computing to stock and access data, it is obligatory to backup and replicates data offsite to confirm calm recovery of data in the experience of interruption and in case of disaster. Data replication is one of the best resolution where there is an increasing need for quicker recovery as it compromises high performance, availability, and reliability. Many researches has done to dynamically replicate data in cloud environment. Among them CIR (Cost-effective Incremental Replication) is one exclusive dynamic replication technique which focused on the cost reduction of replication of data mechanism in cloud environment. On the other hand, the probability of data loss is not completely controlled by this approach as it uses random replication approach. Copyset replication can be a better substitute of random replication in that circumstance. In this paper, we propose a modified CIR (MCIR) algorithm to balance the cost effectiveness with data reliability and consistency by dropping the possibility of data loss in a significant fashion.

IndexTerms **- Cloud,replication,data,Cost-Effective,copyset,Random.**

I. INTRODUCTION

Cloud computing is a practical method to know direct cost benefits and it has the latent to transmute a data center from a capital-intensive build up to a variable priced environment [24]. Replication is building multiple copies of a present entity. Replication enhance availability of data. It also delivers consistency and reliability by making multiple copies of the similar data on various sites. Replication also offers least cost of access, consumption of shared bandwidth and time of postponement by replicating data [11].

Data replication provides availability of data in multiple sites to ensure efficient utilization of data. Because of the extreme growth of data usage, the cloud computing is getting more preferences nowadays. As most companies are using cloud computing to store and access data, it is mandatory to backup and replicates data offsite to ensure easy recovery of data in the event of downtime and in case of disaster. And the best practice for this purpose is using data replication which allows organizations to scale their offsite storage quickly for faster backup and recovery [14].

Since now, a large variety of strategies have been followed to replicate data. Generally, we can categorize data replication techniques into two group named –static and dynamic replication. In static replication, once we decide strategy, it can't be modified while the operation is going on. To overcome this limitation, dynamic replication is employed in which the creation and deletion of replicas can be accrue dynamically according to the need. Cloud computing comprises of virtualized resources for computing and interconnected through a private network or global network (internet) [22]. To increases availability, consistency, and reliability of resources in cloud environment we used data replication and it produces multiple copies of the same data on different sites which is required in cloud [11].

Random replication is adopted as a communal method by storage of data center systems, to guarantee durability and availability. The system can restore its data by reading chunks from their replicated copies, when a node fails. But, extensive interrelated failures like power outages of cluster, a common type of data center failure situation and such events are controlled poorly by random replication.

Copysets are a great alternative to random replication. Copyset is a novel common purpose replication technique that considerably reduces the occurrence of data loss events. Copyset is built on scattering replicas placement. The aim of Copyset Replication is to decrease the probability of data loss, given any scatter width by using the slightest number of copysets. A vital property of MinCopysets is that it does not require altering the architecture of the storage system [23].

The CIR is used to control the data having unclear reliability necessity. But this algorithm does not specify the placement of the new replicas. This algorithm uses random replication and Replicas are positioned on randomly chosen data nodes [25]. As CIR applies random replication and randomly select the nodes placing the replicas and it enhance the data loss probability. But in this case, if we apply copyset replication in the place of random replication, then the data loss probability can be significantly reduced.

II. LITERATURE REVIEW

D2RS (Dynamic Data Replication Strategy): This is a mathematical prototype which is expressed to show the system availability and the number of replicas correlation which was not available in previous researches [15]. The multi-tier hierarchical cloud system architecture was used to replicate data in cloud storage. In D2RS the replication of data file takes place automatically based on popularity [16].This strategy emphasizes on three parameters which are Bandwidth Expenditure, Availability& Number of Replicas. Qingsong Wei et.al [18] offered cost-effective dynamic replication management scheme which is known as CDRM, in order to obtain the relationship between the accessibility and replica number. For a designated availability requirement of a file, it computes and maintains the minimal number of the replica to fulfill the requirement. The measurement of the availability of a file is based on block locations, the current number of blocks, number of replicas, network bandwidth, etc. The replica replacement in CDRM takes place based on data nodes capacity and their blocking possibility in an efficient manner. CDRM is better than default replication technique of HDFS if popularity is small [17].

Although CDRM is an efficient algorithm the consistency of replicas was not ensured by this algorithm properly which needs to be considered in future. Bai et al (2013) proposed a response time-based replica management scheme which produces a replica for automatically raising the number of replicas depending on the average of the response time. It predicts the bandwidth of the replica servers while receiving the new request that creates the replica selection consequently and combines the number of replicas and the network transmission time [9]. RTRM technique focuses on the three issues which are the creation of the replica, selection of replica and placement of replica. This strategy defines a time interval for response time and it will enhance the number of replicas and create new replicas if the response time is higher than time intermission [19]. The important task of LRM (locality replication manager) is to obtain the user's queries, gather condition of nodes in the cluster, and finally select the best host for block placement. LRM carryout this task with the cooperation of its other components and the final decision is made by LRM. HDFS architecture is used by this strategy for replication management. Jenn-Wei Lin has proposed two QoS-aware data replication (.QADR) algorithms in order to continuously support an application QoS requirements after data corruption in cloud computing system. The first algorithm for doing the data replication pursue the idea of high-QoS first-replication (HQFR).

CIR algorithm control spontaneous replication using a threshold. While the threshold is a time period after which the file must be tested for replication. CIR pays a method for unceasingly handling the replication of the files included to the cluster. CIR check When the Replica Creation Threshold go beyond, if the present number of replicas is can meet the user specified availability requirement or not. If not, then replica is augment and this manner carry on till the extreme boundary (maximum of 3) of replica is gotten. CIR every so often check the availability and replica of the file because of which possible risk to the file can be noticed simply and minimized [16].

III. PRPPOSED MODEL

Most of the dynamic data replication focused on consistency, reliability, and performance. And they almost succeeded to provide satisfactory outcome for the above parameters. CIR was the only algorithm that mainly focused on cost and at the same time it ensures the reliability and consistency as well [17]. In the event of data loss which is one more vital portion in data management zone, the related research is outside the scope of CIR algorithm. And no specific solution for data loss prevention is provided by this algorithm [1].

In the event of data loss which is another vital part in data management zone, The CIR algorithm does not provide any specific solution for data loss prevention and the corresponding research is beyond the scope of CIR algorithm [1]. The CIR algorithm using random replication to stores the newly generated replica and stores new replicas in a randomly chosen storage unit manner. [1]

Current data centers encompass a huge volume of storage parts. These storage units all have definite life periods. The possibility of hardware letdown, henceforward data loss, rises according to the storage interval of data. In traditional theories the connection between failure rate and storage period pursues the exponential spreading with failure rate λ which equivalent to the predictable number of failures of a storage unit in a definite time: Reliability = 1- F(X) = eλT, where F(x) is the exponential increasing distribution task.

In This paper we present a modified cost-effective dynamic data replication algorithm by using copy-set replication technique in Cloud data centers. A Cost-effective dynamic data replication strategy (MCIR) implies an incremental replication technique. The aim of this technique is to prevent data loss during replication without losing the generality of CIR algorithm that reduce the number of replicas as well as satisfies reliability requirement in order to ensure the cost-effectiveness [1]. MCIR generates replicas incrementally while existing replicas cannot ensure the reliability. In this approach cost is the top priority that can be considered as an alternative solution to cloud storage. In this model we used copyset replication instead of random replication, as we mentioned in methodology section that the only problem of CIR algorithm is data loss during data replication and the reason for this problem is random replication that is used by CIR algorithm for replica placement.

In this paper we introduce the details of MCIR based on copyset replication and reliability model.

IV. METHODOLOGY

Input: *FRS*: failure rate set FRS={ $\lambda_1,\lambda_2,\lambda_3 \ldots\ldots \lambda_m$ }
1 **Start** {
2. *RCTP*<-0; // initialize the record of next replica creation time point
3. while (a data storage instance be launched || the replica creation time point be reached){
4. *D*<-get the data which activated this process
5. *X*<-get the reliability requirement of *D*
6. *N=5000, R=3 , S=10// define these parameter to do the permutation in order to select nodes for replica placement*
7. if (*D.k*=1) { // if new instance
8. D.k = replica1
9. *SU*<-get the storage unit for *D randomly* // D has only one *original replica* now
10. λ_1<-get the failure rate of *SU* from *FRS*;
11. solve function $f(T_1) = e^{-\lambda 1 T1} /x$; // function (2)
12. T_1<- find the positive real root of F (T1); // obtain the storage duration
13 . *RCTP*<-current time + T_1 ;} // the first replica creation time point
14. if (*D.k*=2) {
15. D.k = replica2 // if first replica creation time point reaches
16. *SU*<-get the storage units for *D after doing the permutation according to the S/R-1* // two storage units that store replicas of D
17. λ_1, λ_2<- get the failure rate of *SU* from *FRS*;
18. solve function $F(T_2)Xa^{\lambda 1+\lambda 2} - a^{\lambda 1} - a^{\lambda 2} + 1$; // function (3)
19. *a*<- find the positive real root of f(a);
20. T_2<-*lna;* // solve the function in two steps and obtain the storage duration
21 . *RCTP* <- current time + T_2 ;} // the second replica creation time point
22. if (*D.k*=3) { // if second replica creation time point reaches
23 D.k = replica3
24. *SU*<-get the storage units for *D* // three storage units that store replicas of D
25. $\lambda_1, \lambda_2, \lambda_3$<- get the failure rate of *SU* from *FRS*;
26. solve function $f(T_3)/Xb^{(\lambda 1+\lambda 2+\lambda 3)} - b^{(\lambda 1+\lambda 2)} - b^{(\lambda 1+\lambda 3)} - b^{(\lambda 2+\lambda 3)} + b^{\lambda 1} + b^{\lambda 2} + b^{\lambda 3} - 1$;
27. *b*<- find the positive real root of f (b); // function (4)
28. T_3<-lnb;
29. *RCTP<-current time*+T_3 ;} // the time point that the storage instance ends
30. if (*k*!=3) {// create one more replica for *the data* if current replica number is lower than 3
31. *U* <-get a new storage unit for *D;*
32. *D.k<-D.k+1*; // update the replica counter
33. *D.RCTP<-RCTP*; // update the replica creation time point of *D*
34. } //end while

Figure 1. Pseudo code of MCIR algorithm

We take an instance to show how MCIR works. Consider the most common situation, a chunk of data D is newly uploaded or produced, the reliability prerequisite is initialized to a definite value, e.g. 99.999% (i.e. 5 '9s').in common every system typically has hundreds to thousands of nodes and In (line 6) we define the number of Node =5000. The number of replica is 3 by default and here we define the default number of replicas. S is defined as the scatter width that in here we defined S=10. At the start of the storage case, only the original replica of the data stocks in the system, the storage length of this 1-replica state and the first replica creation time point is resulting by resolving function (2) In ((Lines 7-13). pretend that the storage unit U1 has a failure rate of λ1, by resolving function (2), T1 is the first replica creation time point from the present time. The process for computing the second replica time point starts when the first replica creation time point is gotten. Likewise, by solving function (3) in (Lines 14-21) can be achieved and the second replica creation time point is T2 from the present time. This procedure carries on till the data has been altered or the maximum number of replicas is gotten or data loss occurs. At this phase of the procedure, there are at thoroughgoing of three replicas stored in the system. Then, at the last, by solving function (4) (Lines 30-34) the storage duration of the "3-replica phase" can be resulting in which T3 is the storage length of the data with 3 replicas, after T3 from present time, the reliability prerequisite X not able to meet any longer.

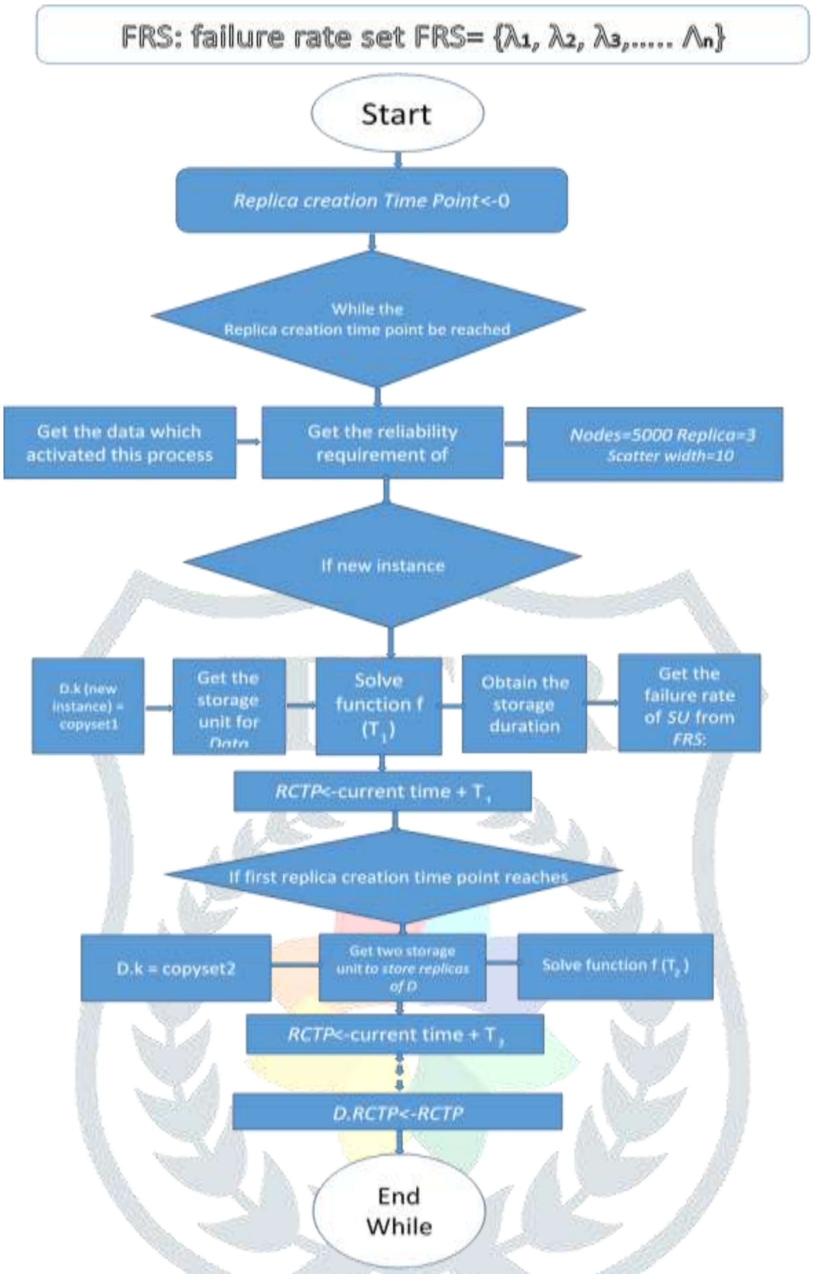

Figure 2. Flow Chart of MCIR strategy

V. COMPARISON B/W CIR & MCIR

Probability of data loss = (#Copyset)/ $^{N}P_{R}$)

Here we are using permutation mathematical formula to show how many copyset we will have if we choose the nodes randomly and as we mention that as the number of combination increases the probability of data loss will also increase. To make it better understandable let's use the formula with the help of an instance. Assume that our system contains 9 nodes and we choose three nodes among them to for replica allocation in this case let's see how many combination or how many option we have.

According to the data loss probability function, random replication 100% cause data loss if the three nodes which contains the replica of same chunk of data simultaneously. [23]

To better clarify this issue, let's walk through an example. In this example we will input our own set of data. For example, suppose a cluster with 15 servers and a replication factor of 3. If we specify a scatter width of 3, we'll need just 4 / (3 -1) = 2 permutation. [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15].

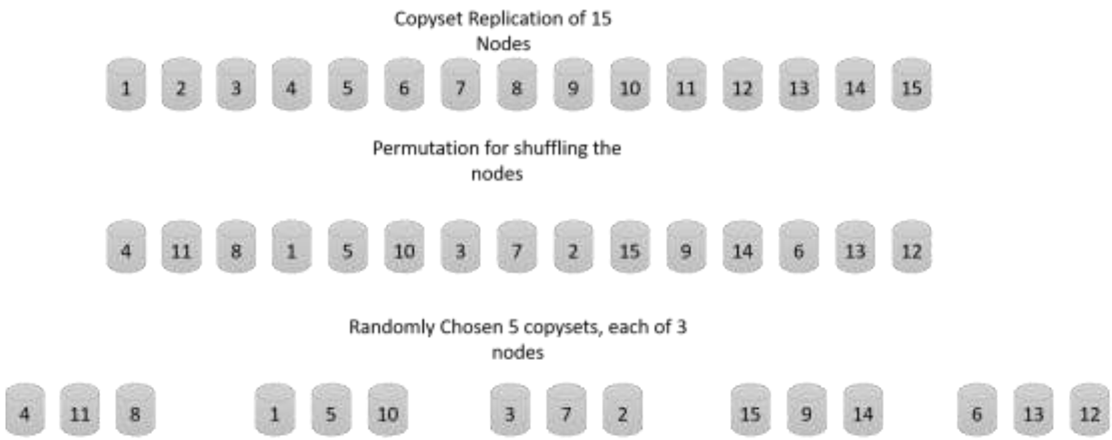

Figure 3. The above figure shows how copyset replication performs

Now to find out the data loss probability we will use the below function. Random replication in the end makes extreme number of copysets.

P(n,r)=n!(n−r)!=?

P(n,r)=P(15,3) =15!(15−3)! = 2730

Any combination of 3 nodes:

Here we found out the No of possible combination by using the above formula. The probability of data loss according to the following calculation is 100 %

Copyset replication is a little more complex. It takes as input a group of server nodes, a detailed scatter width S, and a preferred replication factor R. To create the Copysets:

- The number of permutations needed [Sc / (Rp - 1)], where Sc is the scatter width and Rp is number of replica.
- each permutation is divided into a set of size R
- For assigning a chunk of data, a server is chosen from cluster to store this chunk as the first node
- Choose secondary replicas by randomly selecting one replica assign that comprises the prime.

A number of replica sets will be created to attain the preferred replication factor while nearly upholding the wanted scatter-width.

For instance, suppose a cluster with 15 servers and a replication factor of 3. If we specify a scatter width of 3, we'll need just 4 / (3 -1) = 2 permutation. [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

Let's choice this permutation randomly:

[1, 6,12, 5,11,3,2,14,15,8, 9, 7, 4,13,10]

P (n, r)= n!/(n-r)!

P (n, r) =P (15, 3) = 15!/((15-3)!)=2730

Here, after calculation we got the result that there can be 455 combinations. We choose any combination of this but we randomly select the replica sets.

Now in this step we will divide this permuted server in two size three server groups to generate the replica sets: to find out the number of Copysets we can calculate through

n/r = 15/3=5

n is the number of nodes on our system and r is the replication factor, by dividing the No of node by the replication factor we can find the No of sets we need to create.

[1, 6, 15], [3, 10, 13], [9, 7, 12], [11,5,2], [14,8,4]

Now in case of data loss each set act as one server node and if we calculate the data loss probability then it will be:

Data loss probability=(#Copyset)/"NPR " = 5/"15P3 " =0.183%

From the above equation we can see that the probability of data loss in Copyset replication is only 0.18% which is far less than the probability of data loss in random replication which was 100%.

That proves how significantly Copyset reduces the possibility of losing any data during replication in cloud.

Table 1 Table Type Styles

	CIR	**MCIR**
Replica Placement	CIR does not address the assignment of the new replicas. Replicas are positioned on randomly chosen data nodes.	In MCIR, nodes for replica Placement are chosen with permutation based copyset replication algorithm.
Data loss prevention	In CIR, With a random replication, data loss events happen regularly because of power failure, and numerous chunks of data are gone in each incident.	In MCIR (Modified cost-effective Incremental Replication) algorithm we used copyset replication to reduce the data loss probability.
Reliability	According to the reliability function we used in this paper, the reliability of any replication scheme is inversely proportional to the probability of data loss during replication using that corresponding replication scheme. And as we CIR use random replication which cause huge possibility that the data is lost during replication.so in case of data loss CIR is not much reliable	As MCIR use copyset replication and copyset replication significantly reduce the data loss probability. And according to the reliability model as much as the data loss probability is reduced , the reliability will increase thus it's clear that MCIR is more reliable than CIR.
Efficiency	CIR is efficient in terms of cost effectiveness.	MCIR is more efficient in terms of cost effectiveness and data loss prevention.

VI. RESULT & DISCUSSION:

In this paper we proposed MCIR (modified Cost-effectiveness Incremental Replication) which is a modified version of the original CIR algorithm. We used Copyset replication as a replica placement technique in CIR which reduces the probability of data loss significantly. In original CIR random replication is used as the replica placement which causes an assure data loss in case of power outage and node failure that we explained everything in details. As the result it is proved that MCIR reduced the probability of data loss considerably and the same time it increases the reliability of MCIR as we showed in the reliability function. We can clearly see how much we increased the reliability of CIR by modifying it. According to the reliability function we proposed in this paper, the reliability of any replication scheme is inversely proportional to the probability of data loss during replication using that corresponding replication scheme.

Therefore, it's identically proved that, if the probability of data loss is more, then the reliability of that scheme will be less. If we can reduce the probability of data loss using any other replication scheme, then the reliability will be significantly increased.

5.1 Mathematical experiment results:

After using copyset replication in MCIR algorithm we did a mathematical experiment by using data loss probability function and reliability function. In this experiment we assume the different values of failure rate and we received the data loss probability values by using the data loss probability function which we described in previous section.in this experiment we assume the system with different number of nodes but with same replication factor.in this experiment we assumed same failure for both MCIR and CIR and we got different results.

algorithm	No of nodes	Replication factor	Data loss probability	Failure rate	Reliability
CIR	15	3	100%	4.5	0.0022
MCIR	15	3	0.183	4.5	1.214
CIR	21	3	100%	3.5	0.0028
MCIR	21	3	0.087	3.5	3.284
CIR	33	3	100%	2.5	0.004
MCIR	33	3	0.033	2.5	1.333

As the results in the above table show that the reliability of MICR is significantly higher than CIR. because in MCIR the probability of data loss is considerably low than CIR. According to the received result from the experiment, it's proved that MCIR is improved in both case of data loss prevention and reliability.

5.2 Future work: As a future work of this proposed system, we can suggest real world simulation of this modified strategy in real cloud environment if possible. After that the feasibility of this scheme can be guaranteed. Through the mathematical function and pseudo code we used to prove our research study, it can also be implemented with help of computer simulator that has enough storage, virtual cloud with enough nodes.

REFERENCES

[1] W. Li, Y. Yang and D. Yuan, "A Novel Cost-Effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres," 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure

2)Erl, Thomas, Ricardo Puttini, and Zaigham Mahmood. Cloud computing: concepts, technology & architecture. Pearson Education, 2013

3) Ashraf, Imran. "An overview of service models of cloud computing." International Journal of Multidisciplinary and Current Research 2.1 (2014): 779-783

4) Krishna Murthy1, Dr. R. Selvam2 Security Issues and Challenges in Cloud Computing, IARJSET, Vol. 2, Issue 12, December 2015

5) Subashini, Subashini, and Veeraruna Kavitha. "A survey on security issues in service delivery models of cloud computing." Journal of network and computer applications 34, no. 1 (2011)

6) Aldossary, Sultan, and William Allen. "Data security, privacy, availability and integrity in cloud computing: issues and current solutions." International Journal of Advanced Computer Science and Applications 7.4 (2016):

[7] Bahareh Rahmati, Amir Masoud Rahmani , Ali Rezaei, Data Replication-Based Scheduling in Cloud Computing Environment ,Journal of Advances in Computer Engineering and Technology, 2017

[8] Nazanin Saadat, Amir Masoud Rahmani, PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids, In Future Generation Computer Systems, Volume 28, Issue 4, 2012, Pages 666-681

[9] Bahareh Alami Milani, Nima Jafari Navimipour, A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions, In Journal of Network and Computer Applications, Volume 64, 2016, Pages 229-238

[10] MyunghoonJeon, Kwang Ho Lim, Hyun Ahn, Byong Dai Lee, " Dynamic Data Replication Scheme In Cloud Computing Environment", Second IEEE Symposium on Network Cloud Computing and Applications, London, UK, November 2012,pp. 40-47

[11] Nagamani H Shahapure and P Jayarekha. Article: Replication: A Technique for Scalability in Cloud Computing. International Journal of Computer Applications 122(5):13-18, July 2015

[12] M.Amutha1, R.Sugumar2, "A Survey on Dynamic Data Replication System in Cloud Computing" International Journal of Innovative Research in Science, Engineering and Technology, Vol. 4, Issue 4, April 2015.

[13] Vijay. G. R1, Dr. A. Rama Mohan Reddy2, "Data Replication System in Cloud based on Data Mining Techniques", International Journal of Advanced Research in Computer and Communication Engineering,Vol. 2, Issue 11, November 2013

[14]Milani, Bahareh Alami, and Nima Jafari Navimipour. "A systematic literature review of the data replication techniques in the cloud environments." Big Data Research (2017)

[15] Sun DW, Chang GR, Gao S et al. Modeling a dynamic data replication strategy to increase system availability in cloud computing environments. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(2): 256{272 Mar. 2012. DOI 10.1007/s11390-012-1221-4

[16] Y. Huang and O. Wolfson, "A competitive dynamic data replication algorithm," Proceedings of IEEE 9th International Conference on Data Engineering, Vienna, 1993, pp. 310-317

[17] Rashmi R Karandikar and M B Gudadhe. Article: Comparative Analysis of Dynamic Replication Strategies in Cloud. IJCA Proceedings on Trends in Advanced Computing and Information Technology TACIT 2016(1):26-32, August 2016.

[18] Qingsong-Wei, Bharadwaj Veravalli, Bozhao Gong, Lingfang Zeng, Dan Feng," CDRM: A cost effective dynamic replication management scheme for cloud storage cluster", Proc. IEEE International Conference on Cluster Computing, Herakilon, Crete, Greece, September 2010, pp.188-196

[19] Bai X., Jin H., Liao X., Shi X., Shao Z. (2013) RTRM: A Response Time-Based Replica Management Strategy for Cloud Storage System. In: Park J.J..H., Arabnia H.R., Kim C., Shi W., Gil JM. (eds) Grid and Pervasive Computing. GPC 2013.

[20] Reza SOOKHTSARAEI ,Ahmad FARAAHI, Hadi ADINEH "A locality-based replication manager for data cloud, Frontiers of Information Technology & Electronic Engineer, 2015.

[21] J. W. Lin, C. H. Chen and J. M. Chang, "QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing Systems," in IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 101-115, Jan.-June 2013.

[22]"Robert Escriva and Emin Gün Sirer " hyperdex replication copysets release February 14, 2014

[23]Cidon, A., Rumble, S. M., Stutsman, R., Katti, S., Ousterhout, J. K., & Rosenblum, M. (2013, June). Copysets: Reducing the Frequency of Data Loss in Cloud Storage. In Usenix Annual [24]Technical Conference (pp. 37-48)

MOHANTY, SUNEETA, PRASANT KUMAR PATTNAIK, and GANGA BISHNU MUND. "FRAMEWORK FOR AUDITING IN CLOUD COMPUTING ENVIRONMENT." Journal of Theoretical & Applied Information Technology 65.1 (2014).

[25]Karandikar, Rashmi, and Manish Gudadhe. "Analytical Survey of Dynamic Replication Strategies in Cloud." Proc. IJCA-National Conference on Recent Trends in Computer Science and Information Technology. 2016.

[26]: Mills, K. A., Chandrasekaran, R., & Mittal, N. (2017). Algorithms for Optimal Replica Placement Under Correlated Failure in Hierarchical Failure Domains. arXiv preprint arXiv:1701.01539

[27]Cidon, A., Stutsman, R., Rumble, S., Katti, S., Ousterhout, J., & Rosenblum, M. (2013). MinCopysets: Derandomizing replication in cloud storage. In The 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)